Knowledge-Based Systems 276 (2023) 110737

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Dynamic transformer ODEs for large-scale reservoir inflow forecasting

Xovee Xu [a,1], Zhiyuan Wang [a,1], Fan Zhou [a,*], Ying Huang [b], Ting Zhong [a], Goce Trajcevski [c]

[a] University of Electronic Science and Technology of China, China
[b] CHN Energy Dadu River Big Data Service CO., China
[c] Department of Electrical and Computer Engineering, Iowa State University, United States of America

ARTICLE INFO

Article history:
Received 2 September 2021
Received in revised form 4 April 2023
Accepted 15 June 2023
Available online 20 June 2023

Keywords:
Intelligent inflow forecasting
Hydropower industry
Time series
Self-attention
Ordinary differential equations

ABSTRACT

Forecasting incoming water demand is a critical step in efficient reservoir management and revenue optimization in large-scale cascade hydropower stations. It depends on multiple factors, such as weather conditions, grid dispatch, and electricity demand, and, in turn, facilitates a range of downstream decision-making, from natural hazards control and water ecology protection to power generation plans. Current efforts mainly rely on methodologies from statistical machine learning or deep neural networks to model the hydrological patterns from historical time series for inflow forecasting. However, existing models are restricted by short-term temporal dependencies and are prone to error accumulation issues due to the underlying autoregressive architecture. Meanwhile, most recent self-attention time-series models fail to achieve real-time inflow forecasting because of tremendous parameters and computational bottlenecks on learning long time series with fine granularity. We propose a novel framework, called DTODE (Dynamic Transformer Ordinary Differential Equations), for capturing nonlinear and non-stationary evolving patterns inherent in hydrological time series. Specifically, we present the dynamic self-attention mechanism combining transformer and ordinary differential equations that simultaneously captures long-range dependencies of observations from a dynamic system perspective. DTODE exploits a continuum of self-attention layers (instead of discrete counterparts) to learn the dynamics of multivariate time series while paying attention to the co-evolving time-related factors. Besides, our model is flexible in inferring the complex states at any time step, allowing us to forecast inflows at multiple time horizons. Comprehensive evaluations on real-world datasets show that DTODE significantly reduces forecasting errors compared to state-of-the-art inflow prediction systems.

1. Introduction

Hydroelectric power, also called hydroelectric energy or hydroelectricity, harnesses the kinetic energy of flowing water and/or waterfalls to spin the blades of a turbine which, in turn, spins rotor of the electricity generator. It is the largest producer of renewable energy in the world, catering to global decarbonization goals, while complementing variable renewables through its flexibility and storage.[2] Hydropower stations usually build dams to control the flow of water in the reservoirs – i.e., combining the storing and releasing of the water for (future) electricity generation [1]. Such reservoirs, especially the ones for large-scale hydropower stations, in addition to hydropower generation [2] also play crucial roles in water resources management enabling water supply, flood/drought alert, dam portfolios planning, agricultural and irrigation management, as well as landslide prevention [3–8].

Conventionally, reservoirs are operated based on the knowledge of experts who usually design mathematical/physical models to simulate the dynamics of inflow/outflow. For example, earlier works [9–11] design time-based rules, based upon which the operation of the reservoir system can be simulated and configured for making optimal policies. However, predefined rule-based models cannot handle sudden events (e.g., flood and dam break), which limits their applicability in real-time reservoir operation.

Effective real-time reservoir operation is a non-trivial task due to variability in natural phenomena that increase water levels (e.g., precipitation and snowmelt) and other factors such as climate and water-use changes. More importantly, reservoir operation usually confronts conflicting, if not even contradictory, optimization goals. For example, excess water in summer should have been stored for future hydropower generation. Still, the storage capacity needs to be maintained at a certain (lower)

* Corresponding author.
E-mail addresses: xovee@std.uestc.edu.cn (X. Xu), zhy.wangcs@gmail.com (Z. Wang), fan.zhou@uestc.edu.cn (F. Zhou), hy69559@163.com (Y. Huang), zhongting@uestc.edu.cn (T. Zhong), gocet25@iastate.edu (G. Trajcevski).

[1] Equal contribution.

[2] www.hydropower.org/publications/2021-hydropower-status-report

https://doi.org/10.1016/j.knosys.2023.110737
0950-7051/

level to tolerate the possible flood peaks. Water discharge may also result in a significant loss of electricity revenue, which may be considerably ameliorated if an accurate and reliable inflow forecast can be made in advance [12]. For example, an optimization of hydroelectric systems using the knowledge of inflow and water discharge based on linear dynamic programs to solve the partially observable Markov decision processes (MDP) was proposed in [13].

In recent years, machine learning methods have been widely adapted to fit complex hydrological time series data. The family of AutoRegressive Integrated Moving Average (ARIMA) models is the first attempt to exploit hydrological time series [14]. In addition, Bayesian networks and K-means clustering were applied for predicting different horizons (e.g., daily or monthly) of water inflow [15]. Other time series learning models such as support vector regression and artificial neural networks (ANN) have also been widely used for modeling nonlinear and nonstationary characteristics of hydrological data [16–18]. These approaches model either univariate water data or short-term observations, and therefore may not capture the complex dependencies among multivariate time series [19] and the nonlinear interactions between different causal factors.

The rapid developments of deep learning techniques have motivated a range of applications in reservoir inflow forecasting [20–22], primarily due to its ability in automatic feature extraction and nonlinear dependency learning. In the literature, recurrent neural network (RNN) and its well-known variants, such as LSTM [23] and GRU [24], become popular in modeling hydrological time series dependencies [25–27]. Despite the encouraging results compared to traditional statistical and machine learning approaches, RNN-based models are still limited to learning short-term temporal dependencies because of the exploding or vanishing gradient issues in training RNNs with long-term observations. Besides, the sequential output of RNN cells prevents its application in real-time training and forecasting of the large-scale hydrological time series data.

Recently, the Transformer models based on the self-attention mechanism [28] have demonstrated better performance than RNNs in capturing long-range dependency, and have attained state-of-the-art results in a range of sequential learning tasks, such as machine translation [29] and audio/speech processing [30]. Transformers rely on the multi-head attention mechanism to focus on the information from different positions, rendering them particularly suitable for learning time series data. Unfortunately, directly applying canonical Transformers to long-range sequences is computationally prohibitive because its space and time complexities grow quadratically with the sequence length. In addition, real-world time series data are often composed of both long- and short-term periodic patterns, which are difficult to capture by the vanilla Transformer. To address these issues while extending self-attention to time series forecasting, several variants of Transformers have been proposed to capture intricate temporal dependencies and improve forecasting performance. For example, LogSparse Transformer [31] first validates Transformer's capability in handling long-range dependencies and employs causal convolutions to produce queries and keys in the self-attention layer. Adversarial sparse Transformer [32] introduces a generator to learn a sparse attention map for time series and designs a discriminator to improve the prediction performance. Informer [33] enhances the prediction capacity of Transformers in long-term time series forecasting by improving self-attention with a distilling operation.

Despite the encouraging results made by the aforementioned works, prior self-attention-based models can hardly be directly applied for real-time inflow forecasting due to the large-scale parameters in the neural networks. In addition, previous endeavors did not exhibit good performance when modeling the irregular time series for multi-horizon inflow forecasting. This, however, is of particular interest for critical decision-making in operating hydropower stations, e.g., persistently optimizing long-term reservoir operation strategies such as aquatic ecosystem protection and hydroelectricity revenue maximization. Furthermore, intelligent inflow forecast systems refer to a set of continuous-time series such as water discharge, precipitation, and temperature. As a complementary observation, both RNNs and Transformers consider the continuous dynamic systems with discrete-time network layers, which may not meet the complex requirements of control systems such as reservoir operation, where high-frequency feedback is necessary to maintain system flexibility and stability.

In this paper, we propose Dynamic Transformer ODEs (DTODE) to tackle the aforementioned limitations and solve the multi-horizon inflow forecasting problem. DTODE is a novel inflow forecasting method exploiting the underlying connections between Transformer and ODEs while forecasting the multi-horizon inflow adaptively. DTODE is also an intelligent inflow forecasting system that is specifically designed for large-scale reservoirs of hydropower stations. It provides an alternative view of dam operation by modeling the statistical dependency among multivariate time series rather than the mutually independent evolution of individual sequential observations. Specifically: (1) introduce an efficient embedding method with a convolutional operation to capture the interactive relationship between different kinds of observations; (2) present a dynamic self-attention mechanism to capture the long-term temporal dependencies between the time-evolving monitoring data. It enriches the model with stacked attention layers with neural ODEs, enabling computational efficiency and fast convergence with limited memory cost for real-time inflow forecast; (3) design an evolving module for multi-horizon forecasting, which takes advantage of sequential factors for making predictions instead of forced temporal alignment and iterative forecasting in previous models. To our knowledge, DTODE is the first work bridging the gap between self-attention neural networks and neural ODEs while evolving dynamic systems with the attention mechanism.

In summary, the main contributions of this work are threefold:

- We propose a novel dynamic Transformer-based time-series model for water flow forecasting. It explicitly accounts for the locality and temporal features on the premise of capturing sequential dependencies. Moreover, our method leveraging ODE can capture complex time-series patterns efficiently and provides a dynamic perspective of modeling continuous time series.
- We introduce ODE extrapolations and aggregate sliding sequential factors for modeling the evolution of each forecasting horizon, which extrapolates the latent representation at *any* time step. This design allows us to make multi-horizon predictions without forcing temporal alignment and iterative occupation that may lead to significant error accumulation in traditional forecasting models.
- We conduct extensive experiments on real-world datasets collected from three large-scale reservoirs. Comprehensive evaluations demonstrate the superiority of our method, which not only improves the long- and short-term prediction accuracy but also provides the multi-horizon forecast results and the interpretations of the model behaviors.

We note that in our earlier work [34], we introduced an ODE-based model called FlowODE which employs neural ODE to model the multivariate hydrological data and forecast the future inflow of the reservoirs. Its main idea is to deal with multivariate flow time series in a continuous dynamic manner, by extending the

discrete state transitions in RNNs to continuous transformations. In this work, we present a substantially enhanced version of [34] with several new characteristics:

(1) We propose a novel methodology for modeling long hydrological time series by exploiting the connections between self-attention networks and numerical methods of ODEs. The DTODE model proposed in this work incorporates the numerical methods of ODEs into Transformer in a systematic manner and abandons the RNN structure completely. In contrast, FlowODE [34] is a typical RNN-based model that is still constrained by the sequential training problem.

(2) We present a consistent framework to combine multivariate time series and include external factors, considering the heterogeneous data as a whole rather than concatenating different representations as in FlowODE. This new design allows the model to better capture the complex interactions between different factors while alleviating the potential underfitting problem due to the simple representation.

(3) We introduce new datasets and conduct more comprehensive evaluations to verify the effectiveness of the newly proposed DTODE model.

The remainder of this paper is organized as follows. We review the related work in Section 2 and position our work in that context. Formal definitions and the necessary background are introduced in Section 3. The details of the proposed methodology are presented in Section 4, followed by comprehensive experimental evaluations in Section 5. We conclude the paper and outline directions of our future works in Section 6.

2. Related work

In this section, we review the related literature from three perspectives, i.e., Inflow Forecasting, Self-attention Mechanisms, and Neural Ordinary Differential Equations, and position our work in that context.

2.1. Inflow forecasting

An efficient reservoir inflow forecast is essential for making appropriate water-regulating decisions that could impact flood control, irrigation, drought prevention, and dam operation from both efficiency and safety perspectives [8,20]. However, accurate inflow prediction is a non-trivial task. It is affected by many factors, including (but not limited to) rainfall, climate, soil conditions, snow amounts, and the operations of upstream reservoirs. In the past few decades, inflow forecast has received increased attention from both academia and industry due to its importance for economic development and other societal benefits [35,36].

Earlier efforts typically relied on linear statistical methods to model the hydrological time series and forecast the future inflow. For example, the autoregressive integrated moving average (ARIMA) and its many variants have been widely utilized to model the multivariate hydrological time series representing the water flow [14,37,38]. However, those methods can only capture linear dependency and feature interactions and are not straightforwardly generalizable to large-scale reservoirs with a considerable amount of historical data. To model the nonlinear and non-stationary features of hydrological data, subsequent studies have leveraged data-driven methods for modeling and predicting the future inflow. For example, a support vector regression (SVR) to predict the discharges of monthly river flow was used in [39]. Bayesian networks have also been utilized to evaluate the stochasticity inherent in water time series and predict the inflow while considering the uncertainties [15,40]. Other machine learning techniques such as fuzzy inference and ensemble learning have also been used for inflow forecast [41,42]. In recent years, there appears some research combining more than one machine learning model that takes more factors into consideration and improves the forecastings [43]. Weighted Voting Regressor has been used as a common technique for merging machine learning models in hydrology applications for better predictions [44,45]. These models rely on typical machine learning approaches to learn the patterns of hydrological data and achieve promising results in water flow prediction. However, the nonlinear dependency and complex interactions among multivariate time series are usually ignored due to the limitations of the underlying linear learning models.

Recently, the advances in deep neural networks have inspired a number of studies using deep autoregressive models such as recurrent neural networks (RNN) for time series forecasting [46]. For example, an RNN autoencoder framework to capture the long-term dependence between multivariate time series and forecast the flood of the river in a multi-step-ahead way was proposed in [26]. The similar recurrent unit has also been used in flood forecasting in urban reservoir [47]. Besides, [21] employed RNN models for the monthly streamflow of the Yangtze River to improve accuracy and stability. [48] proposed a GRU-based model to decompose the original runoff series for mid-term runoff forecasting. Recent empirical studies [22,27] have compared various deep learning models and demonstrated that LSTM [23] – a typical RNN model – significantly outperforms other machine learning models such as ARIMA and SVR. Nevertheless, the performance of RNNs models, including LSTM and GRU [24], is constrained by highly long sequence time series data due to exploding or vanishing gradients of RNNs, which strains the model's prediction capacity. Besides, RNNs are autoregressive models relying on the output of previous steps and thus cannot be trained in parallel.

2.2. Self-attention mechanisms

The self-attention mechanism and its encoder–decoder architecture Transformer [28] have spurred extensive attention due to the impressive performance in a wide range of applications, such as language translation [49], speech recognition [50], and image generation [51]. Compared with traditional deep sequences processing models such as RNN and its variants, the Transformer network has several significant advantages. Firstly, it is a parallel-in-time architecture that allows for parallel training while avoiding recurrent computation. Secondly, it considers the sequence as a whole, enabling the model to capture the global long-distance dependency. Lastly, it addresses the gradient vanishing issue in the long sequences.

Various Transformer architectures have also been used for time-series processing. For example, in [52] a Transformer with deep layers was applied to influenza-like illness forecasting, whose performance is comparable to the state-of-the-art deep recurrent models. Hierarchical Transformer-based learning [53,54] approaches were proposed for financial time series forecasting tasks and achieved considerable improvement over previous deep learning models. Except for forecasting, Transformer-based methods also show their advantages on other time series modeling tasks such as classification [55], anomaly detection [56,57] and institution [58]. Moreover, a few attempts have combined Transformer with other popular deep architectures. For instance, the generative adversarial network (GAN) has been introduced in Transformer for time-series forecasting [32,59], which learns a sparse attention map for time series and regularizes the model with a discriminator for improving the prediction performance. Graph neural network (GNN) has been included in Transformer for spatial relations modeling in multivariate time series [60,61].

As for the Transformer itself, efforts were made to adapt the characteristic of different time series data such as enhancing the locality with a convolutional operation and breaking the memory bottleneck by proposing the LogSparse Transformer that requires fewer dot products in each attention layer [31]. A Transformer-based model called Informer to enhance the prediction capacity for long sequence time series forecasting was proposed in [33], which is a time-efficient model capturing long-range dependency between the outputs and inputs of the time series. Autoformer [62] is a decomposition architecture that progressively separates the long-term trend from the immediate latent variables, enabling the model to discover the similarity and periodicity of time series. FEDformer [63] introduces a decomposition module to capture the global profile of time series while employing Transformers to capture more detailed parts. Non-stationary Transformer [64] proposes Stationarization and De-stationary Attention mechanisms to avoid degenerating terribly on non-stationary real-world data.

Existing Transformer-based models have essentially improved both accuracy and efficiency of time-series forecasting. However, some drawbacks still exist when utilizing the Transformer for time-series forecasting. On the one hand, the self-attention mechanism has advantages in discovering long-term, but it still cannot highlight the significant correlations between adjacent time steps due to the discrete nature of the underlying neural networks. On the other hand, the representation learned by a single attention layer is limited since the multiple superimposed layers carry lots of parameters, making the model too large and too complex to fit the time series data.

2.3. Neural ordinary differential equations

Pioneering studies have found the intrinsic connections between neural networks (e.g., CNNs and RNNs) and dynamical systems [65–67]. For instance, [68] bridged the gap between the deep CNN such as ResNets [69] and the ordinary differential equations (ODEs), and demonstrated that CNNs could be interpreted as an Euler discretization step of an ODE. Simultaneously, ODE is used to alleviate the gradient vanishing problem in RNNs [67]. In the seminal work NODE [70], the discrete hidden states of neural networks are approximated with the ODE solver, and the adjoint method is introduced to compute the gradients, which essentially saves the memory cost in training the neural networks. Similarly, a stable numerical ODE solution into RNN architecture was introduced in [67], aiming to capture the long-term dependencies of the input sequences. More recently, several studies have been conducted to improve the neural ODE methods — for example: injecting noise to regularize the neural stochastic ODE networks [71], combining Conditional Latent RNN (CLORNN) with ODE solver [72], augmenting neural ODE for more expressive and stable function approximation [73], and correcting the inaccurate gradient problem [74,75]. Neural ODEs have also been applied to solve continuous dynamics in time series. [76] proposed a ODE-RNN model to model irregularly-sampled time series and [77] designed a stabilized ODE structure for long-term forecasting. As we will demonstrate in the rest of the paper, our proposed DTODE model, in which we rely on a novel Transformer and ODE extrapolation and aggregation (while eliminating the RNN), outperforms the existing models for inflow prediction.

3. Preliminaries

We now proceed with formally defining the problem studied in this paper and provide the necessary background on neural ODEs and Transformers.

Table 1
Notations.

Symbol	Description
$\mathbf{X}$	The multi-variable time series of observations
$\mathbf{Z}$	The multi-variable sequential factors
$\mathbf{Y}$	The predictive time series of inflows
N	The number of observation types in time series
M	The number of sequential factors
T	The length of historical time series
τ	The temporal horizons ahead to be predicted
$\mathbf{B}$	The mini-batch of a time-series dataset
$\mathcal{L}$	The loss in training process
θ, Θ	Learnable parameters of sub-modules and the overall model
$\mathbf{E_X}, \mathbf{E_Z}$	The embedding of time series and sequential factors
$\mathbf{S}^m$	The sinusoidal matrix of mth sequential factor
$\mathbf{W}, b$	The weights and bias parameters
$\mathbf{Q}, \mathbf{K}, \mathbf{V}$	The Query, Key, and Value of the self-attention mechanism
$C(\cdot)$	Self-attention calculation
$\mathbf{H}$	Hidden states between input and output of the attention layer
$\mathbf{H_e}, \mathbf{H_d}$	Hidden states after encoder and decoder
L	The number of layers of dynamic self-attention solver
$G(l, \mathbf{H})$	The attention ODE block with initial state $\mathbf{H}$ in layer interval l
$f(t, \mathbf{H})$	The dense ODE block with initial state $\mathbf{H}$ in time interval t

3.1. Problem definition

In inflow water forecasting tasks, we consider observations of N related multivariate time series $\mathbf{X}_{1:T} = \{x_1, \ldots, x_t, \ldots, x_T\}$, where t is the time index, T is the length of historical time series, and each x_i includes N scalars representing N related observations (e.g., inflows, temperature, and precipitation) at the same time instant. Furthermore, we take the sequential factors (e.g., day-of-the-month and hour-of-the-day) into consideration, and use $\mathbf{Z}_{1:T+\tau} = \{z_1, \ldots, z_T, \ldots, z_{T+\tau}\}$ denote them, where each z_i includes M features. Our goal is to predict the value of inflows in the next τ time steps, i.e., $\mathbf{X}^{\text{Inflow}}_{T+1:T+\tau} = \{x^{\text{Inflow}}_{T+1}, \ldots, x^{\text{Inflow}}_{T+\tau}\}$. To this end, we are going to model the following function:

$$\mathbf{Y} = \mathbf{X}^{\text{Inflow}}_{T+1:T+\tau} = \mathcal{F}(\mathbf{X}_{1:T}, \mathbf{Z}_{1:T+\tau}; \Theta)\,, \tag{1}$$

where τ is a hyperparameter representing the horizon of forecasting (which is usually set to 1 h, 1 day, or 1 week), and Θ denotes all the parameters in the model. Table 1 summarizes the notations frequently used throughout this paper.

3.2. Transformer

Transformer [28] is an encoder–decoder model with a multi-head self-attention mechanism that has been widely employed for global long-term temporal dependency learning. Here we briefly introduce the basic architecture of the Transformer and refer readers to [28] for more details. Generally, both the encoder and the decoder networks stack identical layers, each of which consists of an attention occupation and a feed-forward network (FFN). The processing of the encoder can be described as follows:

$$\begin{aligned} \mathbf{Q} &= \mathbf{W_Q}\mathbf{H} + b_{\mathbf{Q}}, \\ \mathbf{K} &= \mathbf{W_K}\mathbf{H} + b_{\mathbf{K}}, \end{aligned} \tag{2}$$

$$C(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{FFN}(\text{softmax}(\mathbf{Q}\mathbf{K}^{\text{T}}/\sqrt{d_{\text{m}}}) \times \mathbf{V})\,, \tag{3}$$

$$\text{Output} = C^L(C^{L-1}(\cdots C^1(\mathbf{H})))\,, \tag{4}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are called *query*, *key*, and *value* matrices, respectively; d_{m} is the dimension of the $\mathbf{H}$; L is the number of layers; C^i is the ith transition layer of the encoder; and $\mathbf{W}$ and b are learnable parameters. This structure is especially suitable for capturing correlations and discovering the patterns in sequences, and forming an effective latent representations [31,33,78] – mainly

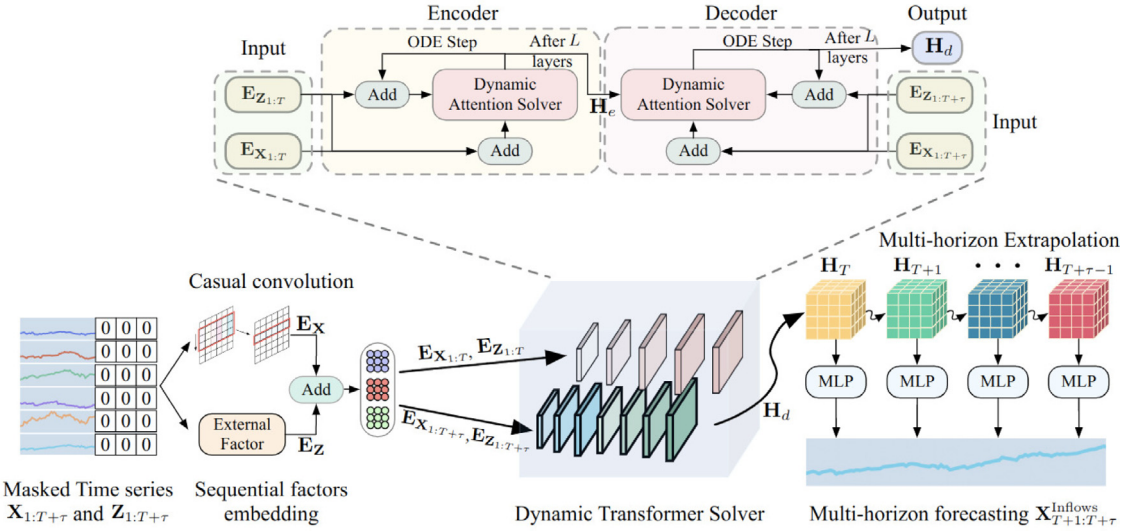


Fig. 1. Overview DTODE and its three major components: (1) convolutional sequence embedding; (2) dynamic self-attention solver; and (3) multi-horizon inflow extrapolation.

because of its ability to access any part of the historical observations, regardless of the distance. However, the representations extracted from shallow layers are limited, while deep-layer stacking may result in large-scale parameters, making the convergence of the model hard to achieve.

3.3. Neural ODE

In regular neural networks, states are transformed by a series of discrete transformations: $\mathbf{h}_{t+1} = f(\mathbf{h}_t)$, where $\mathbf{h}_t$ is the state at time t and f can be any fully connected, convolutional or recurrent layer. However, these models have difficulty when describing the events happening at continuous and irregular intervals. In such situations, Neural ODEs (NODE) [70] – a kind of equations with learned parameters – are preferred for describing plenty of dynamical systems, as follows:

$$\frac{d\mathbf{h}(t)}{dt} = g(t, \mathbf{h}(t); \theta_g)\,, \tag{5}$$

$$\mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} g(t', \mathbf{h}(t'); \theta_g)dt'\,, \tag{6}$$

where $\mathbf{h}(t)$ is the hidden state at time t; t_0 and t_1 denote the initial and final time of the process; g is the derivative function of $\mathbf{h}(t)$ with respect to t; and θ_g is the appropriate parameters of g.

NODE considers the infinite-steps hidden state update in neural networks to replace the discrete sequence of hidden state transformation. It solves the initial value problem with continuous transform and can compute the constant dynamics of hidden states $\mathbf{h}$ via ODEs. Regarding the infinite hidden state update process of the neural ODE block as solving ODEs with numerical methods (such as Euler, Runge–Kutta, and adjoint method [79]) allows obtaining the hidden states $\mathbf{h}(t)$ at *any* desired moment [70]. In the pioneering study [70], the authors propose to use an adjoint method to simulate a dynamical system. However, the dynamics of either the hidden state or the adjoint might be unstable due to numerical instability of backward ODE solve [73,74].

This work uses a neural network module to approximate g since the analytical solution is generally unsolvable (possibly even nonexistent). From the formulae, we can obtain continuous states at the posterior time by calculating the initial state $\mathbf{h}(t_0)$ and fitted parameters θ_g of g.

4. Methodologies

We now present the details of our methodology for hydrological time series modeling and reservoir inflow forecasting.

4.1. Overview of DTODE

Fig. 1 illustrates the architecture of the DTODE framework, which is composed of three main parts:

- **Time series and sequential factor embedding** – performs a 1-D causal convolution to extract meaningful representations of low-dimensional hydrological time series while enhancing local patterns and encoding the external factors such as temperature, rainfall, and power generation (see Fig. 2).
- **Dynamic self-attention solver** – exploits neural ODEs to learn the hidden states of self-attention blocks as a continuous dynamical system. It simultaneously captures long-range dependencies in time-series data and computes the query and key vectors for multi-head attention in a parameter-efficient way.
- **Multi-horizon inflow extrapolation** – extrapolates the learned latent variables that generalize the representations in previous steps to infer the latent variables at *any* time step, enabling our model to predict the future inflow at *any* scale.

4.2. Time series and sequential factor embedding

In general, there are two different kinds of hydrological data in a typical hydropower station: (1) Multivariate time series data consists of the continuous historical records spanning a long period of time, including the water inflow, temperature, rainfall, etc. (2) Other time-related categorical factors, e.g., the season, the month, and the hour in a day, which may provide external signals of the changes in the water inflow. Since the water inflow has distinct seasonal patterns, the time signals are essential features for reservoir inflow forecasting. To efficiently capture these features, we use two different schemes to embed the data for downstream learning. Specifically, we use a 1-D causal convolution network to embed the multivariate temporal sequences and encode the local context of hydrological time series. In addition, the categorical

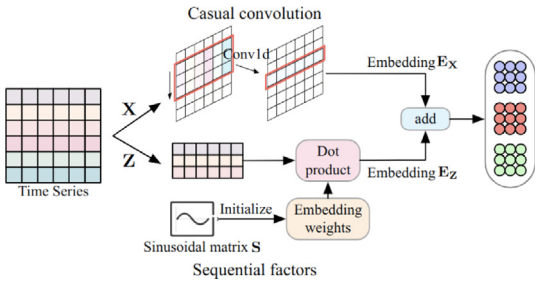


Fig. 2. Illustration of the casual convolution and sequential factors encoding.

time-related features are embedded with a position embedding technique, which allows the model to scrutinize the sequential patterns evolved with time.

4.2.1. Time series embedding with causal convolution

The significant variability in hydrological time series is due to various reasons, such as heavy rains, floods, and drainage of the upstream dam. Therefore, the trend of reservoir inflow should be accurately captured in the embedding space. However, the observation at a particular time instant/step is highly determined by the values at adjacent time steps, which can also be leveraged to identify whether an observation is a regular or change point. In traditional time series models such as ARIMA and RNNs, the sequential context is modeled by the conditional probability of previous observations. In contrast, in the typical Transformer, the sequential context is computed by the similarities between queries and keys based on their dot product values — which may not reflect the trends of inflow such as anomalies caused by extreme weather. Meanwhile, the query-key matching agnostic of local context may confuse the self-attention computation and cause optimization issues [31].

To extract more meaningful representations of hydrological time series, we design a 1-Dimension causal convolution layer to capture the local patterns and interactions between different time series. Specifically, we exploit a causal convolution on N channels, corresponding to observations represented by N different time series, to transform the original time series (with padding 1s) into low-dimensional embeddings $\mathbf{E}_\mathbf{X} \in \mathbb{R}^{N \times d_m}$ as:

$$\mathbf{E}_\mathbf{X} = \text{Conv1d}(\mathbf{X}_{1:T}; \theta_c)^\intercal, \tag{7}$$

where θ_c denotes learned weights in the convolutional kernel (we set the kernel size as 3, the stride step as 1 and the out channel as d_m).

Through the above causal convolution, the generated embedding $\mathbf{E}_\mathbf{X}$ can be more aware of the local sequential patterns and the abnormal observations. In addition, it would allow us to compute the similarities between an observation and its surrounding context, e.g., the output has paid adequate attention to multivariate observations between adjacent 3 time steps, which is beneficial for downstream dependency learning and forecasting.

4.2.2. Embedding of sequential factors

Time-related features are essential for sequence processing, as they enable the model to be more discerning of the temporal characteristic of data evolution over time. Since these features are categorical, we need a method to encode this information while retaining the awareness of the time throughout the evolution. To this end, we borrow the idea of position encoding in self-attention models to embed the sequential factors.

The canonical Transformer [28] records the index of words in a sentence and uses the sinusoidal functions to compute the position embedding, which indeed improves the performance in many NLP tasks. In our hydrological time series data, patterns that evolve with time are greatly affected by seasonal features, due to, for example, natural and man-made events. For instance, the inflow of the reservoir in summer is significantly more than in winter, and the upstream discharge water would also notably increase the inflow. We note, however, that traditional position embedding only considers the fixed index but ignores the sequential factors. To address this issue, we take the sequential factors $\mathbf{Z}_{1:T+\tau}$ into account by first generating a matrix related to each sequential factor with the sinusoidal function (due to its computational efficiency). The calculation of the mth sequential factor $\mathbf{S}^m_{i,j}$ is defined as follows:

$$\mathbf{S}^m_{i,j} = \begin{cases} \sin(i \times c^{j/d_\text{m}}) & \text{if } j \text{ is even,} \\ \cos(i \times c^{j/d_\text{m}}) & \text{if } j \text{ is odd,} \end{cases} \tag{8}$$
$$i \in \{0, 1, \ldots, \chi^m\},\ j \in \{0, 1, \ldots, d_\text{m}\}$$

where i and j respectively denote the ith row and jth column of the matrix, $c = 0.0001$ is a constant (following standard positional embedding [28]), and χ^m is the maximum value of the sequential factor, e.g., $\chi^m = 24$ when the factor denotes the hour-of-the-day. Then, we establish an embedding layer and treat the sinusoidal matrix as its initial weights. The embeddings $\mathbf{E}_Z$ of the sequential factors can be obtained by concatenating the individual features:

$$\mathbf{E}_\mathbf{Z} = \sum_{m=1}^{M} \mathbf{W}^m \cdot \mathbf{Z}^m, \tag{9}$$

where $\mathbf{Z}^m$ means the embedding of the mth feature and $\mathbf{W}^m$ denotes the corresponding learnable parameters with initial value $\mathbf{S}^m$.

4.3. Dynamic self-attention solver

Now we introduce our dynamic self-attention solver that couples the neural ODE to adapt any discrete Transformer architectures into continuous dynamic systems, as illustrated in Fig. 3.

In a typical Transformer [28] and its many variants for time series data learning [31,33,78,80,81], both the encoder and the decoder are fed with the embedding of time series, while the decoder additionally takes the hidden states $\mathbf{H}_e$ outputted by the encoder as input. However, a single self-attention layer may not fully capture the complex sequential dependencies and meaningful time series patterns. Therefore, existing Transformer-based time series models have to stack more layers, requiring significantly more parameters. This may lead to parameter redundancy while making the training process unstable and the model difficult to converge.

To address this issue, we introduce neural ODE to the self-attention layers of both the encoder and the decoder. Generally, each step of the ODE solver handles the data at a specific time index $t \in T$ when modeling the time series. Unlike this, we treat the hidden states $\mathbf{H}$ of the input and output of each attention layer as the input and output of the corresponding ODE solver's step. In other words, there is a bijective mapping between the attention layer and the continuous ODE solver. Our dynamic self-attention solver only requires the same parameters as a single attention layer. The depth of the model can be seen as a variable, which is more flexible than previous deep Transformer models.

In each step of the attention ODE solver, there are three main phases: (1) First, we add the constant sequential factors embedding to the output of the previous step, making each step has the same perception of sequential factors and positional information. (2) Next, we apply a typical attention mechanism with dot product, enabling the model to pay attention to the critical information and capture long-term dependencies. In this

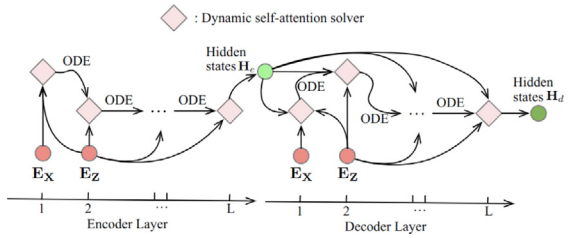


Fig. 3. Illustration of the multiple-layer dynamic self-attention via ODE solver.

way, the generated hidden states contain temporal patterns of both hydrological time series and sequential features. (3) Finally, a fully connected layer and a GeLU activation function [82] are employed to perform the non-linear transformations. Suppose $\mathbf{H}^l$ is the hidden state of the lth step, the transition procedure between two steps of the dynamic self-attention solver can be formally described as follows:

$$\frac{d(\mathbf{H}(l))}{dl} = G(l, \mathbf{H}(l) + \mathbf{E_Z}; \theta_G) \,, \tag{10}$$

$$\mathbf{H}(l+1) = \mathtt{Norm}(\mathbf{H}(l) + \int_l^{l+1} G(l', \mathbf{H}(l') + \mathbf{E_Z};\ \theta_G)\, dl'), \quad 0 \le l < L \,, \tag{11}$$

where $\mathbf{H}(0)$ equals to the input embedding $\mathbf{E_X}$, G is an attention block with parameters θ_G, $\mathtt{Norm}$ denotes normalization layer, and L is a hyperparameter representing the number of layers.

To calculate the integration term in Eq. (11), we employ the fourth-order Runge–Kutta method [79] since it has higher precision than the simple Euler method. The calculation can be described as:

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{G}(l, \mathbf{H}(l) + \mathbf{E_Z}) \,, \\ \mathbf{G}_2 &= \mathbf{G}(l + 1/2, \mathbf{H}(l) + \mathbf{G}_1/2 + \mathbf{E_Z}) \,, \\ \mathbf{G}_3 &= \mathbf{G}(l + 1/2, \mathbf{H}(l) + \mathbf{G}_2/2 + \mathbf{E_Z}) \,, \\ \mathbf{G}_4 &= \mathbf{G}(l + 1, \mathbf{H}(l) + \mathbf{G}_3 + \mathbf{E_Z}) \,, \end{aligned} \tag{12}$$

$$\int_l^{l+1} G(l', \mathbf{H}(l') + \mathbf{E_Z};\ \theta_G)\, dl' = \frac{1}{6}(\mathbf{G}_1 + 2\mathbf{G}_2 + 2\mathbf{G}_3 + \mathbf{G}_4) \,, \tag{13}$$

where $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{G}_3$ and $\mathbf{G}_4$ denote the derivative at the beginning, midpoint, and end of the interval. In this way, we approximate the integration with multi-step discrete processes.

Algorithm 1: Dynamic self-attention solver in the encoder.

Input:
The embedding of time series $\mathbf{E}_{\mathbf{X}_{1:T}}$ and corresponding sequential factors $\mathbf{E}_{\mathbf{Z}_{1:T}}$.
The number of layers L.
Output: The hidden states after the encoder $\mathbf{H}_e$.
Initialize: $\mathbf{H}_0 \leftarrow \mathbf{E}_{\mathbf{X}_{1:T}}$
Iterative transition in self-attention block of each layer:
foreach i in $[1, ..., L]$ **do**
$\mathbf{H} \leftarrow \mathbf{H}_{i-1} + \mathbf{E}_{\mathbf{Z}_{1:T}}$
$\mathbf{Q} \leftarrow \mathbf{W_Q H} + b_{\mathbf{Q}},\ \mathbf{K} \leftarrow \mathbf{W_K H} + b_{\mathbf{K}},\ \mathbf{V} \leftarrow \mathbf{W_V H} + b_{\mathbf{V}}$
$G(l, \mathbf{H}) =$
$\mathrm{FFN}(\mathbf{H} + \text{self-attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})) + \text{self-attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$
Calculate $\mathbf{H}_i$ via Eq. (11), (12) and (13)
$\mathbf{H}_i \leftarrow \mathrm{Norm}(\mathbf{H}_i)$
end for
Output hidden states: $\mathbf{H}_e \leftarrow \mathbf{H}_L$

The procedures of the proposed dynamic self-attention solver for the encoder and decoder are outlined in Algorithm 1 and Algorithm 2, respectively. In the encoder network, the original input is the embedding of historical time series before employing the self-attention mechanism. In the decoder, the input is the output of the encoder $\mathbf{H}_e$ and the embedding of masked target time series (i.e., zeros instead of real future values) to ensure that the current position would not access the future information. We observe that $\mathbf{Q}$, $\mathbf{K}$, *and*$\mathbf{V}$ are respectively the affine of input embeddings in self-attention of encoder and decoder, i.e., $\mathbf{Q}$ is the affine of $\mathbf{H}_e$, $\mathbf{K}$ and $\mathbf{V}$ are same with the setting of self-attention in the cross-attention of the decoder.

Algorithm 2: Dynamic self-attention solver in the decoder.

Input:
The embedding of masked time series $\mathbf{E}_{\mathbf{X}_{1:T+\tau}}$ and the sequential factors $\mathbf{E}_{\mathbf{Z}_{1:T+\tau}}$.
The hidden states outputted by encoder $\mathbf{H}_e$.
The number of layers L.
Output: The hidden states after the decoder $\mathbf{H}_d$.
Initialize: $\mathbf{H}_0 \leftarrow \mathbf{E}_{\mathbf{X}_{1:T+\tau}}$
Iterative transition in self-attention block of each layer:
foreach i in $[1, ..., L]$ **do**
$\mathbf{H} \leftarrow \mathbf{H}_{i-1} + \mathbf{E}_{\mathbf{Z}_{1:T+\tau}}$
$\mathbf{Q}_1 \leftarrow \mathbf{W}_{\mathbf{Q}_1}\mathbf{H} + b_{\mathbf{Q}_1},\ \mathbf{K}_1 \leftarrow \mathbf{W}_{\mathbf{K}_1}\mathbf{H} + b_{\mathbf{K}_1},\ \mathbf{V}_1 \leftarrow \mathbf{W}_{\mathbf{V}_1}\mathbf{H} + b_{\mathbf{V}_1}$
$\mathbf{P} \leftarrow \mathbf{H} + \text{self-attention}(\mathbf{Q}_1, \mathbf{K}_1, \mathbf{V}_1)$
$\mathbf{Q}_2 \leftarrow \mathbf{W}_{\mathbf{Q}_2}\mathbf{H}_e + b_{\mathbf{Q}_2},\ \mathbf{K}_2 \leftarrow \mathbf{W}_{\mathbf{K}_2}\mathbf{P} + b_{\mathbf{K}_2},\ \mathbf{V}_2 \leftarrow \mathbf{W}_{\mathbf{V}_2}\mathbf{P} + b_{\mathbf{V}_2}$
$G(l, \mathbf{H}) = \mathrm{FFN}(\mathbf{P} + \text{cross-attention}(\mathbf{Q}_2, \mathbf{K}_2, \mathbf{V}_2)) +$
$\text{cross-attention}(\mathbf{Q}_2, \mathbf{K}_2, \mathbf{V}_2)$
Calculate $\mathbf{H}_i$ via Eq. (11), (12) and (13)
$\mathbf{H}_i \leftarrow \mathrm{Norm}(\mathbf{H}_i)$
end for
Output hidden states: $\mathbf{H}_d \leftarrow \mathbf{H}_i$

4.4. Multi-horizon inflow extrapolation

Previous RNN- and Transformer-based models usually perform well on observations over fixed intervals, but may fail to model irregular samplings. That is, once the training process finishes, the forecasting time steps and intervals are fixed. Besides, existing models rely on the performance of iterative predictions for multi-horizon prediction, which are limited to error accumulation at previous steps. As an example, in the current Transformer models [31,33,78,80,81] the long-term results have been decoded from the latent representation. Nevertheless, if the forecasting horizon is different from the input sequence length, these models are easily prone to inaccurate predictions.

In the real dam operation, we would like to model flexible sampling intervals, learn the multi-horizon dependencies, and forecast the future inflow at different scales. Towards this goal, we design another ODE solver with a variable related to τ, which can effectively solve multi-horizon forecasting problems. Specifically, we reformulate the time series evolving process as follows:

$$\mathbf{H}_{T+t} = \mathbf{H}_T + \int_T^{T+t} f(t', \mathbf{H}_T + \mathbf{E}_{\mathbf{Z}_{t+1:T+t}}; \theta_f) dt' \,, \quad t \in \{1, 2, \ldots, \tau - 1\} \tag{14}$$

where f is a dense block, $\mathbf{H}_T$ equals to the output of the attention solver in the decoder $\mathbf{H}_d$, $\mathbf{E}_{\mathbf{Z}_{t+1:T+t}}$ is the sequential factors embedding of $\mathbf{Z}_{t+1:T+t}$, and θ_f denotes learned parameters of f. By considering the evolution of the hydrological time series with a continuous dynamic system, we endow the model with the ability to fit and extrapolate the future inflow.

Here we also employ the Runge–Kutta method, similar to Eq. (12) and (13), for solving this equation. During the process,

the previous latent states evolve with posterior sequential factors embedding in the ODE solver, and we can get the latent representations from time T to $T+\tau-1$. More importantly, it allows us to arbitrarily change the size of τ (i.e., the horizons) even though the model training has been accomplished since the parameters θ_f are shared in the ODE function and the time series evolution is considered as a continuous dynamic system in our method.

4.5. Inflow forecasting & model training

By now, we have obtained the expressive representations of time series from time $T+1$ to $T+\tau$, which sufficiently reflects the sequential reliance and correlations between different kinds of observations. Therefore, we use a multi-layer perceptron to transform the representations to the inflow forecasting at time $T+\tau$ as follows:

$$\begin{aligned}\mathbf{Y} &= \mathbf{X}^{\text{Inflow}}_{T+1:T+\tau} \\ &= \text{MLP}(\{\mathbf{H}_T, \mathbf{H}_{T+1}, \ldots, \mathbf{H}_{T+\tau-1}\}; \theta_m)\,, \end{aligned} \tag{15}$$

where MLP is the multi-layer perceptron and θ denotes learned parameters.

We train our DTODE model with stochastic gradient descent (SGD) using Adam optimizer [83], and we choose the mean squared error (MSE) loss function as our optimization objective. Given a batch of the time series data B, we update all parameters by minimizing the loss given by:

$$\mathcal{L} = -\frac{1}{|\mathbf{B}| \times \tau} \sum_{i=1}^{\mathbf{B}} \sum_{t=T+1}^{T+\tau} (\hat{y}_{i,t} - y_{i,t})^2\,, \tag{16}$$

where $y_{i,t}$ and $\hat{y}_{i,t}$ denote the target and prediction inflow of the ith sample at time t. During training, the loss is back-propagated from the decoder's outputs across the entire model. The whole training procedure of our DTODE is summarized in Algorithm 3.

Algorithm 3: The training procedure of DTODE.

Input:
The sequential factors $\mathbf{Z}_{1:T+\tau}$ and masked time series embedding $\mathbf{X}_{1:T+\tau}$.
Output: The inflow forecasting $\mathbf{X}^{\text{Inflow}}_{T+1,T+\tau}$.
Initialize all parameters Θ of DTODE
while not converged **do**
Obtain the input embedding $\mathbf{E_Z}$ and $\mathbf{E_X}$ via Eq. (7) and Eq. (9).
Compute $\mathbf{H}_e$ by Algorithm 1.
Compute $\mathbf{H}_d$ conditioned on $\mathbf{H}_e$ by Algorithm 2.
Evolve $\mathbf{H}_d$ and obtain $\mathbf{H}_{T:T+\tau-1}$ via Eq. (14)
Predict the future inflows $\mathbf{X}^{\text{Inflow}}_{T+1,T+\tau}$ via Eq. (15)
Update Θ by minimizing the objective in Eq. (16)
end while

4.6. Complexity analysis

There are three main components in DTODE, i.e., data embedding, dynamic self-attention solver, and multi-horizon evolution. Data embedding transforms original sequential data into representations through causal convolution and sequential factor mapping. The time complexity of 1-D convolution is $\mathcal{O}(TNd_{\text{m}})$. Also, the complexity of sinusoidal computation is linear with a constant related precision, requiring total $T \times d_{\text{m}} \times \sum_{m=1}^{M} \chi^m$ calculations. Thus, the complexity of sequential factor mapping is $\mathcal{O}(TMd_m)$ and that of data embedding is $\mathcal{O}\left(T(M+N)d_m\right)$. For the dynamic self-attention solver, there are L layers and P steps between adjacent layers. The representation feeds forward in each step via an attention block. Hence the complexity is linear with that of the attention block, i.e., $\mathcal{O}(T^2 d_{\text{m}})$. For the third part, we choose the dense layer as the block of multi-horizon evolution, which has a computational complexity linear with $\mathcal{O}(\tau d_{\text{m}}^2)$.

Fig. 4. Illustration of the studied areas. Three research stations (i.e., Pubugou, Shenxigou, Danba) distributed along Dadu River have been marked red.

5. Evaluations

In this section, we report the findings of our experiments conducted to evaluate our DTODE by comparing its performance with state-of-the-art inflow forecasting and time-series modeling methods on three real-world datasets. Specifically, we aim to address the following research (RQ) questions via extensive experiments:

- **RQ1**: How does DTODE perform when compared to existing time series forecasting models, especially recent ODE-based and Transformer-based methods?
- **RQ2**: How do the critical model components affect the performance of DTODE?
- **RQ3**: Can DTODE provide interpretable representation learning for inflow forecasting?

5.1. Experimental settings

We now describe in detail the settings of our experiments.

5.1.1. Datasets

We conduct experiments on three real-world datasets collected from two large-scale hydropower plants and a hydrometric station, as marked in Fig. 4. The three datasets are:

- **PBG**: Pubugou is an artificial dam corresponding to the largest hydropower station along the Dadu River. It houses a hydroelectric power station with 6×600 MW generators for a total installed capacity of 3,600 MW. For the reservoir of PBG, the normal storage level is 850 m, with 5.39 billion m^3 total storage capacity. Among the total storage, there are 1.06 billion m^3 for flood storage and 3.88 billion m^3 for regulating storage.
- **SXG**: Shenxigou is a smaller hydropower station located downstream of PBG, installed with 4×165 MW generators. The standard storage level of the reservoir is 660 m, with a total storage capacity of 32 million m^3. As a regulation station of PBG, its inflow is significantly affected by the drainage of the PBG dam.
- **Danba** is a hydrology monitoring station located in the upper basin of Dadu River, which covers 52,738 km^2 area, accounting for 68% of the whole basin. Due to its geographical location and essential role in monitoring the meteorology of the river, this data is a significant reference for predicting the overall downstream water flows.

Table 2
Statistics of three real-world datasets.

Dataset	PBG	SXG	Danba
Time Spanning	2 years	2 years	2 years
Time Interval	1 h	1 h	1 h
Time Series			
Water Inflow	[0.0, 7020.0]	[0.0, 5571.0]	[0.0, 4531.5]
Avg. Inflow	1549.0	1496.9	906.7
Water Outflow	[119.0, 5670.0]	[53.8, 6090.0]	NULL
Generation Flow	[119.0, 2470.0]	[55.1, 2480.0]	NULL
Temperature/°C	[−24.4, 21.7]	[−24.4, 21.7]	NULL
Power Generation	[0.0, 3587.8]	[0.0, 660.0]	NULL
Rainfall (mm)	NULL	NULL	[0.0, 34.0]
Sequential Factors			
MonthOfYear	[1, 12]	[1, 12]	[1, 12]
DayOfMonth	[1, 31]	[1, 31]	[1, 31]
HourOfDay	[0, 24)	[0, 24)	[0, 24)

Table 2 shows the statistics of the three datasets. Each dataset includes observations spanning two years — from Jan.1, 2017, to Dec.31, 2018. The two datasets from hydropower stations include three types of observed multivariate time series data — hydropower generation, water inflow, and temperature. The data from Danba contains two types of hydrological time series, i.e., water inflow and precipitation. For each dataset, we used the historical 50% data for training, 25% for model validation, and the rest 25% for testing.

5.1.2. Baseline models

The following baseline approaches are selected for comparison in the experiments:

- **History Average (HA)** – treats the average of observations of the last C time steps as the predictions, and we set $C = 5$.
- **ARIMA** – a generalization of an autoregressive moving average (ARMA) model. It is a statistical analysis method for time-series data, which can well understand the dataset and predict future trends based on past periods. It was employed in [14] for inflow prediction.
- **SVR** – a regression type of support vector machine (SVM), which has been utilized for inflow forecasting and has been shown to achieve quite good performance [84,85]. Here we use the linear kernel for SVR following [84].
- **GRU** [24] – is an RNN variant that can process events with long-term observations and is more efficient than another popular variant LSTM with comparable performance. It has been widely used for time-series forecasting, including inflow prediction [25,27,85].
- **GRU-VAE** [86] – employs GRU as the basic module for both encoder and decoder and employs VAE [87] to reconstruct the latent representations of input time series for time series forecasting.
- **Latent-ODE** [76] – generalizes RNNs to continuous-time hidden dynamic systems defined by neural ODEs. It can naturally handle arbitrary time gaps between observations and explicitly model the probability of observation times using Poisson processes.
- **AST** [32] – an Adversarial Sparse Transformer (AST) that uses a sparse Transformer as the generator to learn a sparse attention map for time series and leverages a discriminator to improve long-term dependencies.
- **Informer** [33] – a state-of-the-art Transformer-based time series forecasting model. It consists of three distinct components: a ProbSparse attention mechanism with lower time complexity, a distilling operation highlighting the dominant attention representation, and a generative decoder predicting long time-series sequences.
- **FlowODE** [34] – is an ODE-based inflow forecasting model. It models the RNNs with neural ODEs to provide a dynamic perspective of learning continuous time series data. Besides, FlowODE encodes the stochasticity of RNN hidden layers and the uncertainty of long- and short-term dependencies among temporal observations.

5.1.3. Parameter settings & implementation details

All models are tuned to the best performance with early stopping when validation loss has not declined for 100 consecutive epochs. The length of input sequence of each model is set as $T = 168$. We train all deep learning models using a dynamic learning rate with an initial value of 3×10^{-4}, which halves every 50 epoch. For our DTODE, we set the number of the layers L of both encoder and decoder as 4, and the dimension of hidden states d_m as 256. In self-attention occupation, we employ canonical dot-product for effective information extraction and layer normalization [88] to stabilize the hidden state dynamics in the neural networks. For ODE settings, we apply the adjoint method [70] for back-propagation for less resource consumption. Besides, we utilize the Runge–Kutta method as mentioned in Section 4.3 based on the compromises between computational precision and model efficiency.

5.1.4. Evaluation protocols

We evaluate the algorithms with three metrics that are generally used for assessing time series prediction models. Among them, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are two scale-dependent measures, while Mean Absolute Percentage Error (MAPE) is a deviation proportion measure.

5.2. Performance comparisons (RQ1)

We conducted two different comparisons that are meaningful for dam operation: (1) *immediate forecasting* (i.e., $\tau = 1$ hour), and (2) *multi-horizon forecasting* – where $\tau = 24$ hours (i.e., 1 day) and $\tau = 24 \times 7$ hours (i.e., 1 week), respectively.

5.2.1. On inflow forecasting

Table 3 shows the performance of different approaches in forecasting the inflow of three reservoirs. As we can see, our proposed method DTODE achieves the best performance in terms of all the metrics across three datasets.

In addition, we have the following important observations.

Firstly, the statistical time series models such as HA and ARIMA perform poorly due to their inability to model non-linear dependencies in hydrological multivariate time series. SVR, which has been widely used in inflow forecasting, performs worse than recurrent neural models such as GRU. This result demonstrates the superiority of RNNs on learning long- and short-term dependencies in time series.

Secondly, the GRU models can be improved by capturing the stochasticity of time series data, as done by GRU-VAE and LatentODE. As a variational sequential learning method, GRU-VAE combines the known priors to build a probabilistic model for latent factor learning and posterior inference. However, its improvement over recurrent neural networks is limited, largely due to the bottleneck of VAE in encoding the factors of variation — which, therefore, restricts its performance in learning useful and compact representations of the time series data.

Thirdly, both Latent-ODE and FlowODE improve inflow forecasting by modifying the recurrent neural networks with continuous hidden layers through modeling the dynamic inflow forecasting systems while solving the parameter redundancy issue. However, the two models also rely on RNNs to model the sequential dependencies of time series data, which have

Table 3
The inflow forecasting performance of baselines and our model.

Datasets	PBG			SXG			Danba		
Metrics	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
HA	685.0	514.8	0.734	582.6	374.8	0.801	39.52	19.14	0.020
ARIMA	486.7	378.1	0.705	328.2	223.3	0.555	25.58	13.31	0.014
SVR	453.1	355.9	0.684	231.5	160.4	0.344	19.85	10.54	0.011
GRU	416.1	328.9	0.665	177.6	126.8	0.278	15.41	8.593	0.009
GRU-VAE	415.1	326.2	0.665	174.6	124.8	0.261	14.01	8.166	0.009
Latent-ODE	412.4	324.5	0.659	172.7	124.9	0.258	13.57	7.872	0.008
AST	423.7	334.4	0.668	183.5	132.1	0.284	11.89	7.049	0.007
Informer	413.5	325.8	0.664	175.3	123.2	0.259	9.596	6.052	0.006
FlowODE	401.5	312.3	0.621	166.2	114.8	0.231	9.534	5.980	0.006
DTODE	**397.4**	**305.9**	**0.613**	**151.4**	**101.7**	**0.218**	**9.393**	**5.951**	**0.006**

Table 4
Multi-horizon forecasting comparisons on three datasets.

Datasets		PBG			SXG			Danba		
Metrics		RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
1 day	SVR	602.6	472.4	0.909	351.9	243.8	0.523	26.80	14.23	0.015
	GRU	553.4	437.4	0.884	270.0	192.7	0.423	20.80	11.60	0.012
	GRU-VAE	552.1	433.8	0.883	265.4	189.7	0.397	18.91	11.02	0.012
	Latent-ODE	532.0	418.6	0.850	253.0	183.0	0.378	17.91	10.39	0.010
	AST	550.8	434.7	0.868	273.4	196.8	0.423	15.81	9.375	0.009
	Informer	537.6	423.5	0.863	261.2	183.6	0.386	12.76	8.049	0.008
	FlowODE	518.9	397.7	0.801	232.0	166.4	0.348	12.42	7.803	0.008
	DTODE	**508.7**	**391.6**	**0.785**	**218.5**	**146.4**	**0.314**	**12.21**	**7.736**	**0.008**
1 week	SVR	634.3	498.3	0.958	388.9	269.5	0.578	28.78	15.28	0.016
	GRU	582.5	460.5	0.931	298.4	213.0	0.467	22.34	12.46	0.013
	GRU-VAE	577.0	453.4	0.923	293.3	209.7	0.438	20.31	11.84	0.013
	Latent-ODE	556.7	438.1	0.890	276.3	199.8	0.413	18.73	10.86	0.011
	AST	580.5	458.1	0.915	300.9	216.6	0.466	16.65	9.869	0.010
	Informer	566.5	446.3	0.910	287.5	202.0	0.425	13.43	8.572	0.009
	FlowODE	546.4	421.5	0.844	264.9	172.4	0.366	13.20	8.300	0.008
	DTODE	**536.5**	**413.0**	**0.828**	**251.7**	**164.8**	**0.353**	**13.04**	**8.260**	**0.008**

been proven to be inefficient in dealing with long-term correlations [28]. Our DTODE inherits the benefits of previous ODE-based methods, while offering several key advantages. In particular, our method totally discards the auto-regressive networks in RNNs. Instead, it models the hydrological time series with self-attention networks, which allows the model to pay attention to any part of the historical observations regardless of distance.

Lastly, DTODE significantly outperforms previous Transformer-based models such as AST and Informer, which verifies our motivation of improving self-attention networks with continuous dynamic models. AST is capable of capturing the long-term reliance and correlations between different kinds of observations. Although it may reduce the error accumulation and regularize the model at the sequence level, AST relies on adversarial networks [89] which is known as unstable during model training. Informer improves the self-attention in a typical Transformer with sparse attention and reduces the space complexity with the distilling operation. Nevertheless, Informer still needs to stack many discrete layers. More importantly, the Informer model focuses on learning long-term dependency, restricting its performance in capturing subtle local patterns in sequences (e.g., due to food and drainage of the upstream reservoir) and complex mutual influence among multivariate time series (e.g., precipitation) that are crucial in learning hydrological data. In contrast, our DTODE introduces the expressive time series embedding and dynamic mechanism for modeling the hydrological data, enabling us to capture the co-evolving patterns of multiple factors and perform free calculations and optimizations without incurring computational overhead through numerical ODE solvers.

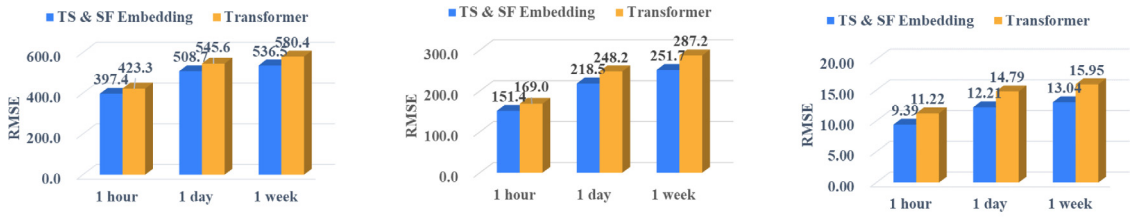
5.2.2. On multi-horizon forecasting

Multi-step-ahead inflow forecasting is of great interest for managing the dams since it can provide more reliable predictions and facilitate decision-making in front of extreme events. Table 4 compares the performance of different approaches in multi-step-ahead inflow forecasting, demonstrating the superiority of DTODE over other baselines in predicting the future inflow with distinct horizons.

RNN-based methods such as GRU and GRU-VAE predict the future inflow in an auto-regressive manner, suffering from the error accumulation issue that significantly deteriorates their forecasting performance. Transformer-based approaches achieve relatively better performance due to their ability to model and forecast long-term dependencies. However, these methods also rely on deeper stacking of self-attention layers that still accumulate errors and, more importantly, are restricted by the input sequence sampling intervals.

ODE-based methods – i.e., Latent-ODE, FlowODE, and our proposed DTODE – are free to extrapolate the evolving time series and therefore perform better than other baselines because of their nature of modeling the neural networks as dynamic systems and therefore perform better than the non-ODE baseline approaches. In other words, the continuous characteristics of these models allow better long-term inference and flexible extrapolations of multi-horizon water inflow. Compared to Latent-ODE and FlowODE, our DTODE is not restricted by the RNN architecture and, therefore, avoids the error accumulation problem raised by autoregressive forecasting. Additionally, DTODE is capable of learning the critical long-term dependencies due to the inherent self-attention mechanism.

To further validate the robustness of the proposed DTODE, we run our model 5 times on each horizon across three datasets with different random seeds. Table 5 reports the statistical results in terms of RMSE, MAE, and MAPE with standard deviations, which supports the effectiveness and stability of our model.



(a) PBG (b) SXG (c) Danba

Fig. 5. Ablation study: the effect time series and sequential factors embedding (TS & SF Embedding).

Table 5
Statistical results on three datasets.

Datasets		PBG	SXG	Danba
1 h	RMSE	397.4 ± 3.3	151.4 ± 2.0	9.393 ± 0.151
	MAE	305.9 ± 2.4	101.7 ± 1.3	5.591 ± 0.082
	MAPE	0.613 ± 0.005	0.218 ± 0.002	0.006 ± 0.000
1 day	RMSE	508.7 ± 8.3	218.5 ± 10.1	12.21 ± 0.17
	MAE	391.6 ± 3.5	146.4 ± 6.5	7.736 ± 0.108
	MAPE	0.785 ± 0.062	0.314 ± 0.030	0.008 ± 0.000
1 week	RMSE	535.5 ± 9.4	251.7 ± 9.5	13.04 ± 0.18
	MAE	413.0 ± 5.8	164.8 ± 5.7	8.260 ± 0.147
	MAPE	0.828 ± 0.103	0.353 ± 0.054	0.008 ± 0.000

5.3. Ablation studies (RQ2)

Recall that DTODE consists of three main components: data embedding, dynamic self-attention solver, and multi-horizon extrapolation. To study the effect of each part, we perform a series of ablation studies by replacing the specific module with the previous best approaches, and we report our observations next.

5.3.1. Effect of time series and sequential factors embedding

In previous Transformer-based models [31,33,78,80,81], multivariate time series are embedded and learned with self-attention layers in an end-to-end manner. In contrast, we propose to learn the representations of time series and external factors separately. To validate our method, we replace the data embedding method in DTODE with a typical Transformer model.

Fig. 5 compares the performance across three datasets, which proves the effectiveness of our method in learning data embeddings. The improvement lies in the time series causal convolution and positional factor embedding in our model. The two mechanisms encourage DTODE to simultaneously capture the patterns of time series and the complex temporal dependencies while being aware of the time-related features crucial to future time series (e.g., inflow here) forecasting.

Besides, we find that DTODE reaps more gains on Danba because precipitation is more informative than other features (e.g., water discharge and power generation), implying that our method is more efficient in capturing the interactions between different time series. This can also be understood by the strategy of the canonical Transformer in embedding time series data. That is, the Transformer employs dot-product self-attention to match the queries against keys, making it insensitive to local context and prone to anomalies. In contrast, DTODE learns sequential patterns with causal convolution, which naturally focuses on the local patterns and change points. The long-term dependency learning is left to the dynamic self-attention solver in the successive module.

Moreover, we can observe evident advantages of our method in multi-horizon forecasting. This phenomenon can be explained by the differences in time series embedding between DTODE and Transformer. The latter uses a constant sinusoidal matrix for sequence embedding. Nevertheless, our DTODE generates time series representation dynamically, allowing us to fit the different horizons and predict multi-step-ahead inflow flexibly.

5.3.2. Effect of dynamic self-attention solver

Next, we investigate the effect of our dynamic self-attention solver (DSAS) by replacing it with state-of-the-art self-attention designs. Three counterparts are considered in this experiment, including canonical self-attention (SA) used in Transformer [28], sparse self-attention (SSA) proposed in AST [32], and ProbSparse self-attention (PSSA) in Informer [33]. Considering that the attention mechanism is mainly used to discover the reliance and obtain corresponding representations, we only report the short-term forecasting results. Specifically, we set different layers ($L = 2, 4, 6, 8$) of different mechanisms and fix all other settings. The experimental results are shown in Fig. 6, where we can observe that the RMSE scores of SA, SSA, and PSSA decrease first and then increases with the number of layers. However, our DSAS can continuously optimize the model performance as the layers increase. This result suggests that stacking more layers does not guarantee better forecasting performance for baseline approaches since their model parameters would increase significantly with L – and, therefore, make them suffer from the overfitting problem. In contrast, the performance of DSAS is stable and consistently outperforms other approaches. DSAS is based on numerical ODE solvers and thus only requires the same parameters as a one-layer network, which allows us to continuously optimize the model without incurring extra computational overhead.

5.3.3. Effect of multi-horizon extrapolation

Lastly, we present quantitative observations regarding the effect of multi-horizon extrapolation in DTODE. Recall that we incorporate the numerical ODE solvers in DTODE, enabling it to learn and predict the inflow at arbitrary multi-steps ahead. This characteristic eases the efforts of previous works in aligning the input and output of time series in the encoder and decoder, respectively. Fig. 7 shows the accumulation errors in multi-horizon forecasting. Clearly, our method successfully reduces errors increasing with the time horizon. This result proves the advantages of our method in multi-step-ahead forecasting since it can, to a large extent, resist error accumulations. Although we only report 1-day and 1-week results following the above empirical evaluations, it is worthwhile noting that our method can extrapolate the forecasting results in any time horizon, due to the continuous dynamic system modeling in the DTODE.

5.4. Model interpretability (RQ3)

We now present our observations regarding the third question that our experiments are attempting to address — the potential benefits of DTODE in terms of interpretability.

5.4.1. Visualization of the sequential factors embedding

Sequential time-related factors play important roles in forecasting future inflow, as they provide additional features implying the involving patterns of the hydrological time series. For example, the inflow exhibits certain daily periodicity, and the volume in the summer is significantly larger than in the winter. Therefore, it is highly desirable to embed the temporal factors and

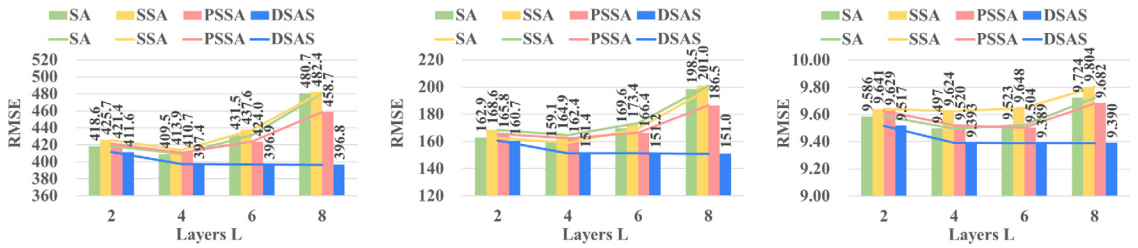


Fig. 6. Ablation study: the effect of dynamic self-attention solver (DSAS). We replace DASA in DTODE with an alternative self-attention mechanism, including canonical self-attention (SA), sparse self-attention (SSA), and ProbSparse self-attention (PSSA).

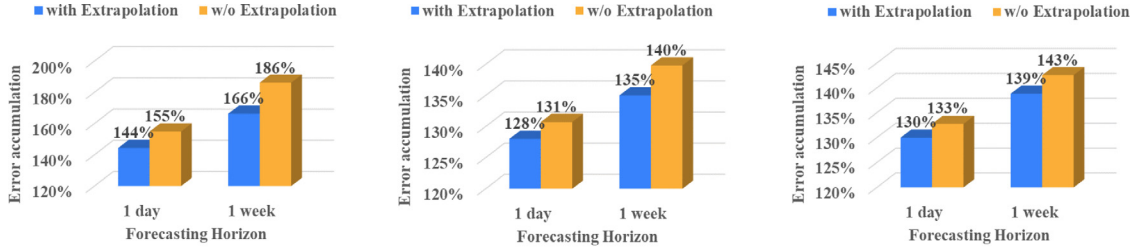


Fig. 7. Ablation study: the effect of multi-horizon extrapolation.

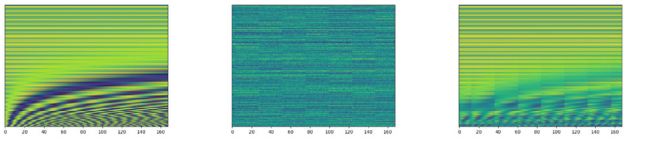


Fig. 8. Visualization of the time-related factor embedding for DTODE and two typical sequential time embedding methods. The *x*-axis denotes the index of each time step (i.e., 1-hour interval).

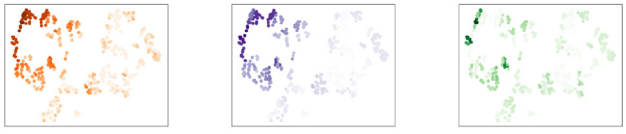


Fig. 9. Visualization of the latent space learned on the SXG dataset.

make the time series aware of the corresponding time-related regularization.

Fig. 8 compares our method with two regular sequential time embeddings in the Transformer. Sinusoidal embedding relies on a constant matrix for sequence learning, which obtains a "smooth" embedding as the distance between the neighboring time steps are symmetrical and decays fluently with time. In contrast, positional embedding with random initialization learns a segmented latent representation without obvious evolving patterns. Since Transformer is originally designed for natural language embedding, positional embedding injects the input words with their relative information. However, the evolving patterns and the subtle local changes are ignored by this method. Our model is initialized with the sinusoidal matrix and optimized by the time-related categorical features, assisting DTODE to concomitantly attend to the time series patterns and natural time intervals.

5.4.2. Visualization of the learned latent representation

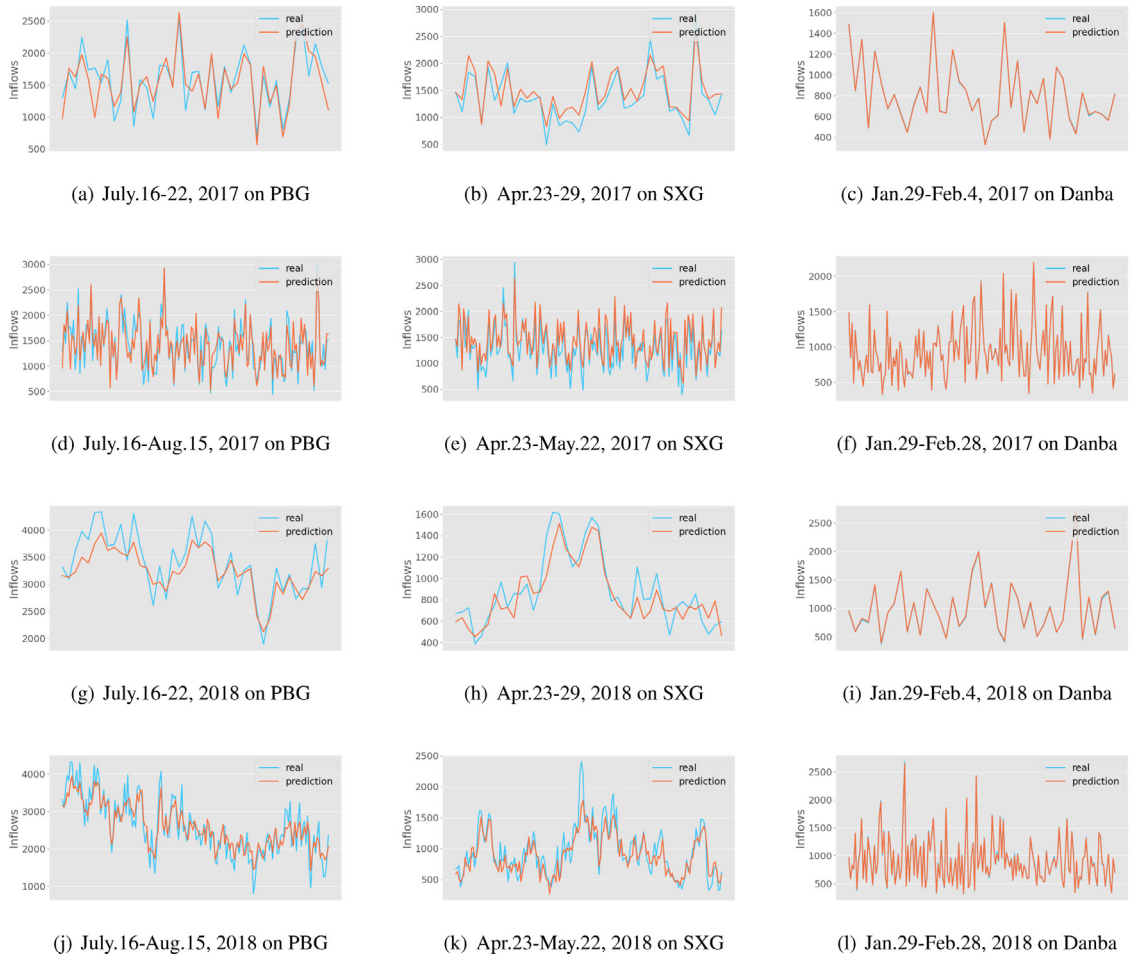
Next, we demonstrate that our DTODE model can successfully learn the nonlinear and non-stationary patterns of time series by visualizing the learned latent representations. We project the latent encoding vectors $\mathbf{H}_{T+\tau}$ of inflow time series to the 2-dimension space using *t-SNE* [90] algorithm and plot the results in Fig. 9, where each point denotes an inflow time series colored by a specific feature of that time series. Here we select three features, i.e., the *predictive* value of inflow, the *average*, and the *variance* of history inflow. Besides, we use their relative sizes as coloring standards, i.e., the higher the value, the darker the color. We can observe that the latent representations correspond closely to these standards, reflecting the trends and patterns of the time series to a certain extent. Therefore, the representations learned by our model are expressive enough to facilitate inflow forecasting.

5.5. Quantitative results

Finally, we investigate the qualitative forecasts made by the DTODE. Specifically, we randomly select the data in two different periods (a week and a month) for each dataset and show the fit of DTODE against the actual inflow values, as illustrated in Fig. 10. We can clearly see that our method can accurately predict the trend of time series even when the inflow shows frequent fluctuations and sharp turns. Moreover, DTODE performs better on Danba than on PBG and SXG. Because PBG and SXG are artificial dams, their operations are affected by many other unpredictable factors, such as water drainage or turbine maintenance in upstream stations. In addition, the rainfall of the areas surrounding the Danba basin is clearly recorded, which is the essential factor affecting the river's inflow. This result also suggests that our method can successfully adapt to the dynamic and co-evolving multivariate hydrological time series patterns.

5.6. Deployment

We close this section with a note that our model has been successfully deployed on the Intelligent Decision-making Support



(a) July.16-22, 2017 on PBG (b) Apr.23-29, 2017 on SXG (c) Jan.29-Feb.4, 2017 on Danba

(d) July.16-Aug.15, 2017 on PBG (e) Apr.23-May.22, 2017 on SXG (f) Jan.29-Feb.28, 2017 on Danba

(g) July.16-22, 2018 on PBG (h) Apr.23-29, 2018 on SXG (i) Jan.29-Feb.4, 2018 on Danba

(j) July.16-Aug.15, 2018 on PBG (k) Apr.23-May.22, 2018 on SXG (l) Jan.29-Feb.28, 2018 on Danba

Fig. 10. Quantitative results on PBG, SXG, and Danba datasets. We randomly select a few samples in two periods (a week and a month) and report the forecast results of DTODE against the true inflow. Other methods are not plotted for visibility.

(a) The IDSS platform. (b) Real-time inflow forecasting with DTODE.

Fig. 11. Snapshots of the IDSS platform and the running of the DTODE model.

System (IDSS) platform of a large-scale hydropower station and is continuously optimized with new hydrological observations. Fig. 11(a) shows a snapshot of the IDSS platform, where both natural factors (e.g., weather, rainfall, and water level) and social factors (e.g., power generation schedule) are displayed. An illustration of the real-time inflow forecasting made by DTODE is shown in Fig. 11(b).

6. Concluding remarks

In this paper, we presented DTODE – a novel multi-horizon reservoir inflow forecasting model for large-scale hydropower stations. DTODE extends the Transformer-based time series method with a numerical ODE solver that allows flexible multi-horizon extrapolation. Our method provides a dynamical perspective of learning hydrological multivariate time series, which requires significantly fewer parameters than previous works. We conducted extensive experiments on three real-world datasets, and the results validated the superiority of our proposed DTODE in terms of both forecasting accuracy and interpretable model behavior.

The proposed model has been successfully deployed on the intelligent inflow forecasting system in a large-scale hydropower generation company. Our algorithm is continuously optimized with new hydrological observations.

As part of our future work, we are interested in improving the extrapolation of ODE solver with more complex and flexible approaches (e.g., improving efficiency by a stable discretization [91]). Another interesting direction is to stabilize the training process of DTODE with alternative ODE solvers, such as ANODE [74]. Complementary to this, modeling stochastic latent representations with variational inference may enable the model to make probabilistic inflow inference and forecasting, which is also important for interpreting the model behavior and forecast results. Last but not the least, we plan to expand DTODE framework to deal with larger-scale spatial extents, involving multiple rivers [92].

CRediT authorship contribution statement

Xovee Xu: Methodology, Experiments, Writing – review & editing. **Zhiyuan Wang:** Experiments, Validation, Visualization, Writing. **Fan Zhou:** Conceptualization, Methodology, Data curation, Funding acquisition. **Ying Huang:** Resources, Methodology, Deployments. **Ting Zhong:** Funding acquisition, Review & editing. **Goce Trajcevski:** Funding acquisition, Review & editing.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled “Dynamic Transformer ODEs for Large-Scale Reservoir Inflow Forecasting”.

Data availability

The authors do not have permission to share data

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62176043 and 62072077, and in part by the Natural Science Foundation of Sichuan Province, China, under Grant 2022NSFSC0505, and in part by the NSF SWIFT under Grant 2030249.

References

[1] M. Jahandideh-Tehrani, O.B. Haddad, H.A. Loáiciga, Hydropower reservoir management under climate change: the Karoon reservoir system, Water Resour. Manag. 29 (3) (2015) 749–770.
[2] M.F. Allawi, O. Jaafar, F.M. Hamzah, S.B. Koting, N.S.B. Mohd, A. El-Shafie, Forecasting hydrological parameters for reservoir system utilizing artificial intelligent models and exploring their influence on operation performance, Knowl.-Based Syst. 163 (2019) 907–926.
[3] Y. Yin, B. Huang, W. Wang, Y. Wei, X. Ma, F. Ma, C. Zhao, Reservoir-induced landslides and risk control in three gorges project on Yangtze river, China, J. Rock Mech. Geotechn. Eng. 8 (5) (2016) 577–595.
[4] F.-J. Chang, M.-J. Tsai, A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques, J. Hydrol. 535 (2016) 256–269.
[5] R.J.P. Schmitt, S. Bizzi, A. Castelletti, J.J. Opperman, G.M. Kondolf, Planning dam portfolios for low sediment trapping shows limits for sustainable hydropower in the mekong, Sci. Adv. 5 (10) (2019) eaaw2175.
[6] Z. Moshe, A. Metzger, G. Elidan, F. Kratzert, S. Nevo, R. El-Yaniv, HydroNets: Leveraging river structure for hydrologic modeling, in: International Conference on Learning Representations ICLR, 2020.
[7] M. Ehteram, H. Karami, S.-F. Mousavi, A. El-Shafie, Z. Amini, Optimizing dam and reservoirs operation based model utilizing shark algorithm approach, Knowl.-Based Syst. 122 (2017) 26–38.
[8] Z.-k. Feng, S. Liu, W.-j. Niu, B.-j. Li, W.-c. Wang, B. Luo, S.-m. Miao, A modified sine cosine algorithm for accurate global optimization of numerical functions and multiple hydropower reservoirs operation, Knowl.-Based Syst. 208 (2020) 106461.
[9] O. Sigvaldson, A simulation model for operating a multipurpose multireservoir system, Water Resour. Res. 12 (2) (1976) 263–278.
[10] J. Cuena, The use of simulation models and human advice to build an expert system for the defense and control of river floods, in: International Joint Conference on Artificial Intelligence, IJCAI, 1983, pp. 246–249.
[11] A.P. Georgakakos, D.H. Marks, A new method for the real-time operation of reservoir systems, Water Resour. Res. 23 (7) (1987) 1376–1390.
[12] S.K. Ahmad, F. Hossain, A generic data-driven technique for forecasting of reservoir inflow: Application for hydropower maximization, Environ. Model. Softw. 119 (2019) 147–165.
[13] M. Petrik, S. Zilberstein, Linear dynamic programs for resource management, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 25, 2011.
[14] W.-c. Wang, K.-w. Chau, D.-m. Xu, X.-Y. Chen, Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition, Water Resour. Manag. 29 (8) (2015) 2655–2675.
[15] P. Noorbeh, A. Roozbahani, H.K. Moghaddam, Annual and monthly dam inflow prediction using Bayesian networks, Water Resour. Manag. 34 (9) (2020) 2933–2951.
[16] M. Aboutalebi, O. Bozorg Haddad, H.A. Loáiciga, Optimal monthly reservoir operation rules for hydropower generation derived with SVR-NSGAII, J. Water Res. Plan. Man. 141 (11) (2015) 04015029.
[17] Y. Tikhamarine, D. Souag-Gamane, A.N. Ahmed, O. Kisi, A. El-Shafie, Improving artificial intelligence models accuracy for monthly streamflow forecasting using grey wolf optimization (GWO) algorithm, J. Hydrol. 582 (2020) 124435.
[18] Z.-k. Feng, W.-j. Niu, Hybrid artificial neural network and cooperation search algorithm for nonlinear river flow time series forecasting in humid and semi-humid regions, Knowl.-Based Syst. 211 (2021) 106580.
[19] P. Zhou, K.C.C. Chan, A feature extraction method for multivariate time series classification using temporal patterns, in: Pacific-Asia Conference, PAKDD, 2015.
[20] L.-C. Chang, F.-J. Chang, S.-N. Yang, F.-H. Tsai, T.-H. Chang, E.E. Herricks, Self-organizing maps of typhoon tracks allow for flood forecasts up to two days in advance, Nature Commun. 11 (1) (2020) 1983.
[21] S. Ha, D. Liu, L. Mu, Prediction of Yangtze river streamflow based on deep learning neural network with El Niño–southern oscillation, Sci. Rep. 11 (1) (2021) 11738.
[22] S.D. Latif, A.N. Ahmed, E. Sathiamurthy, Y.F. Huang, A. El-Shafie, Evaluation of deep learning algorithm for inflow forecasting: A case study of Durian Tunggal Reservoir, peninsular Malaysia, Nat. Hazards (2021) 1–19.
[23] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[24] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.
[25] S. Yang, D. Yang, J. Chen, B. Zhao, Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model, J. Hydrol. 579 (2019) 124229.
[26] I.-F. Kao, Y. Zhou, L.-C. Chang, F.-J. Chang, Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting, J. Hydrol. 583 (2020) 124631.
[27] H. Apaydin, H. Feizi, M.T. Sattari, M.S. Colak, S. Shamshirband, K.-w. Chau, Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting, Water 12 (5) (2020) 1500.
[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 30, 2017, pp. 6000–6010.
[29] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2019, pp. 4171–4186.
[30] H. Kameoka, W. Huang, K. Tanaka, T. Kaneko, N. Hojo, T. Toda, Many-to-many voice transformer network, IEEE ACM Trans. Audio Speech Lang. Process. 29 (2021) 656–670.
[31] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, X. Yan, Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 32, 2019, pp. 5243–5253.
[32] S. Wu, X. Xiao, Q. Ding, P. Zhao, Y. Wei, J. Huang, Adversarial sparse transformer for time series forecasting, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 33, 2020, pp. 17105–17115.
[33] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021.
[34] F. Zhou, L. Li, Forecasting reservoir inflow via recurrent neural ODEs, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 15025–15032.
[35] H. Kling, Climate variability risks for electricity supply, Nature Energy 2 (12) (2017) 916–917.

[36] E.P. Anderson, C.N. Jenkins, S. Heilpern, J.A. Maldonado-Ocampo, F.M. Carvajal-Vallejos, A.C. Encalada, J.F. Rivadeneira, M. Hidalgo, C.M. Cañas, H. Ortega, N. Salcedo, M. Maldonado, P.A. Tedesco, Fragmentation of andes-to-amazon connectivity by hydropower dams, Sci. Adv. 4 (1) (2018) eaao1642.
[37] J.D. Salas, Applied Modeling of Hydrologic Time Series, Water Resources Publication, 1980.
[38] M. Valipour, M.E. Banihabib, S.M.R. Behbahani, Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of dez dam reservoir, J. Hydrol. 476 (2013) 433–441.
[39] J.-Y. Lin, C.-T. Cheng, K.-W. Chau, Using support vector machines for long-term discharge prediction, Hydrol. Sci. J. 51 (4) (2006) 599–612.
[40] A.M. Ticlavilca, M. McKee, Multivariate Bayesian regression approach to forecast releases from a system of multiple reservoirs, Water Resour. Manag. 25 (2) (2011) 523–543.
[41] A.K. Lohani, N. Goel, K. Bhatia, Improving real time flood forecasting using fuzzy inference system, J. Hydrol. 509 (2014) 25–41.
[42] H.S. Lee, Y. Liu, J. Ward, J. Brown, A. Maestre, H. Herr, M.A. Fresch, E. Wells, S.M. Reed, E. Jones, Nationwide validation of ensemble streamflow forecasts from the hydrologic ensemble forecast service (HEFS) of the US national weather service, in: AGU Fall Meeting Abstracts, Vol. 2017, 2017, pp. H41A–1416.
[43] K.S.M.H. Ibrahim, Y.F. Huang, A.N. Ahmed, C.H. Koo, A. El-Shafie, A review of the hybrid artificial intelligence and optimization modelling of hydrological streamflow forecasting, Alex. Eng. J. 61 (1) (2022) 279–303.
[44] M. Troin, J.-L. Martel, R. Arsenault, F. Brissette, Large-sample study of uncertainty of hydrological model components over north america, J. Hydrol. 609 (2022) 127766.
[45] M. Rajesh, S. Anishka, P.S. Viksit, S. Arohi, S. Rehana, Improving short-range reservoir inflow forecasts with machine learning model combination, Water Resour. Manag. 37 (1) (2023) 75–90.
[46] H. Wan, S. Guo, K. Yin, X. Liang, Y. Lin, CTS-LSTM: LSTM-based neural networks for correlatedtime series prediction, Knowl.-Based Syst. 191 (2020) 105239.
[47] B. Cai, Y. Yu, Flood forecasting in urban reservoir using hybrid recurrent neural network, Urban Clim. 42 (2022) 101086.
[48] X. Wang, S. Zhang, H. Qiao, L. Liu, F. Tian, Mid-long term forecasting of reservoir inflow using the coupling of time-varying filter-based empirical mode decomposition and gated recurrent unit, Environ. Sci. Pollut. Res. 29 (58) (2022) 87200–87217.
[49] J. Zhang, H. Luan, M. Sun, F. Zhai, J. Xu, M. Zhang, Y. Liu, Improving the transformer translation model with document-level context, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP, 2018, pp. 533–542.
[50] L. Dong, S. Xu, B. Xu, Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2018, pp. 5884–5888.
[51] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, D. Tran, Image transformer, in: International Conference on Machine Learning, ICML, PMLR, 2018, pp. 4055–4064.
[52] N. Wu, B. Green, X. Ben, S. O'Banion, Deep transformer models for time series forecasting: The influenza prevalence case, 2020, arXiv preprint arXiv:2001.08317.
[53] Q. Ding, S. Wu, H. Sun, J. Guo, J. Guo, Hierarchical multi-scale Gaussian transformer for stock movement prediction, in: International Joint Conference on Artificial Intelligence, IJCAI, 2020, pp. 4640–4646.
[54] X. Zhan, L. Kou, M. Xue, J. Zhang, L. Zhou, Reliable long-term energy load trend prediction model for smart grid using hierarchical decomposition self-attention network, IEEE Trans. Reliab. (2022).
[55] B. Zhao, H. Xing, X. Wang, F. Song, Z. Xiao, Rethinking attention mechanism in time series classification, Inform. Sci. (2023).
[56] W. Zhang, C. Zhang, F. Tsung, Grelen: Multivariate time series anomaly detection from the perspective of graph relational learning, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, 2022, pp. 2390–2397.
[57] C. Ding, S. Sun, J. Zhao, MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection, Inf. Fusion 89 (2023) 527–536.
[58] W. Du, D. Côté, Y. Liu, Saits: Self-attention-based imputation for time series, Expert Syst. Appl. (2023) 119619.
[59] X. Li, V. Metsis, H. Wang, A.H.H. Ngu, Tts-gan: A transformer-based time-series generative adversarial network, in: Artificial Intelligence in Medicine: 20th International Conference on Artificial Intelligence in Medicine, AIME 2022, Halifax, NS, Canada, June 14–17, 2022, Proceedings, Springer, 2022, pp. 133–143.
[60] X. Geng, X. He, L. Xu, J. Yu, Graph correlated attention recurrent neural network for multivariate time series forecasting, Inform. Sci. 606 (2022) 126–142.
[61] S. Guan, B. Zhao, Z. Dong, M. Gao, Z. He, GTAD: Graph and temporal neural network for multivariate time series anomaly detection, Entropy 24 (6) (2022) 759.
[62] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Adv. Neural Inf. Process. Syst. 34 (2021) 22419–22430.
[63] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: International Conference on Machine Learning, PMLR, 2022, pp. 27268–27286.
[64] Y. Liu, H. Wu, J. Wang, M. Long, Non-stationary transformers: Exploring the stationarity in time series forecasting, in: Advances in Neural Information Processing Systems, 2022.
[65] W. E, A proposal on machine learning via dynamical systems, Commun. Math. Stat. 5 (1) (2017) 1–11.
[66] Y. Lu, A. Zhong, Q. Li, B. Dong, Beyond finite layer neural networks - bridging deep architectures and numerical differential equations, in: International Conference on Machine Learning, ICML, 2018, pp. 3282–3291.
[67] B. Chang, M. Chen, E. Haber, E.H. Chi, Antisymmetricrnn: A dynamical system view on recurrent neural networks, in: International Conference on Learning Representations, ICLR, 2019.
[68] R.T.Q. Chen, J. Behrmann, D.K. Duvenaud, J.-H. Jacobsen, Residual flows for invertible generative modeling, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 32, 2019, pp. 9916–9926.
[69] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2016, pp. 770–778.
[70] T.Q. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 31, 2018, pp. 6572–6583.
[71] X. Liu, T. Xiao, S. Si, Q. Cao, S. Kumar, C.-J. Hsieh, Neural SDE: Stabilizing neural ODE networks with stochastic noise, 2019, arXiv Preprint arXiv:1906.02355v1.
[72] F. Zhou, L. Li, K. Zhang, G. Trajcevski, F. Yao, Y. Huang, T. Zhong, J. Wang, Q. Liu, Forecasting the evolution of hydropower generation, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, 2020, pp. 2861–2870.
[73] E. Dupont, A. Doucet, Y.W. Teh, Augmented neural ODEs, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 32, 2019.
[74] A. Gholami, K. Keutzer, G. Biros, ANODE: Unconditionally accurate memory-efficient gradients for neural ODEs, in: International Joint Conference on Artificial Intelligence, IJCAI, 2019, pp. 730–736.
[75] T. Zhang, Z. Yao, A. Gholami, K. Keutzer, J. Gonzalez, G. Biros, M. Mahoney, ANODEV2: A coupled neural ODE evolution framework, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 32, 2019.
[76] Y. Rubanova, T.Q. Chen, D.K. Duvenaud, Latent ordinary differential equations for irregularly-sampled time series, in: Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 5321–5331.
[77] A.J. Linot, J.W. Burby, Q. Tang, P. Balaprakash, M.D. Graham, R. Maulik, Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems, J. Comput. Phys. 474 (2023) 111838.
[78] N. Kitaev, L. Kaiser, A. Levskaya, Reformer: The efficient transformer, in: International Conference on Learning Representations, ICLR, 2020.
[79] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (2–3) (1997) 151–167.
[80] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, 2021, arXiv preprint arXiv:2106.13008.
[81] S. Wang, B.Z. Li, M. Khabsa, H. Fang, H. Ma, Linformer: Self-attention with linear complexity, 2020, arXiv preprint arXiv:2006.04768.
[82] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), 2016, arXiv preprint arXiv:1606.08415.
[83] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, ICLR, 2015.
[84] M. Aboutalebi, O. Bozorg Haddad, H.A. Loáiciga, Optimal monthly reservoir operation rules for hydropower generation derived with SVR-NSGAII, J. Water Resour. Plan. Manag. 141 (11) (2015) 04015029.
[85] M. Babaei, R. Moeini, E. Ehsanzadeh, Artificial neural network and support vector machine models for inflow prediction of dam reservoir (case study: Zayandehroud dam reservoir), Water Resour. Manag. 33 (6) (2019) 2203–2218.
[86] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837.
[87] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: International Conference on Learning Representations, ICLR, 2014.
[88] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, arXiv preprint arXiv:1607.06450.

[89] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 2672–2680.
[90] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).
[91] J. Zhang, A. Mokhtari, S. Sra, A. Jadbabaie, Direct Runge-Kutta discretization achieves acceleration, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 31, 2018, pp. 3904–3913.
[92] F.M. Fan, D. Schwanenberg, W. Collischonn, A. Weerts, Verification of inflow into hydropower reservoirs using ensemble forecasts of the TIGGE database for large scale basins in Brazil, J. Hydrol.: Regional Stud. 4 (2015).