

# A robust autoregressive long-term spatiotemporal forecasting framework for surrogate-based turbulent combustion modeling via deep learning

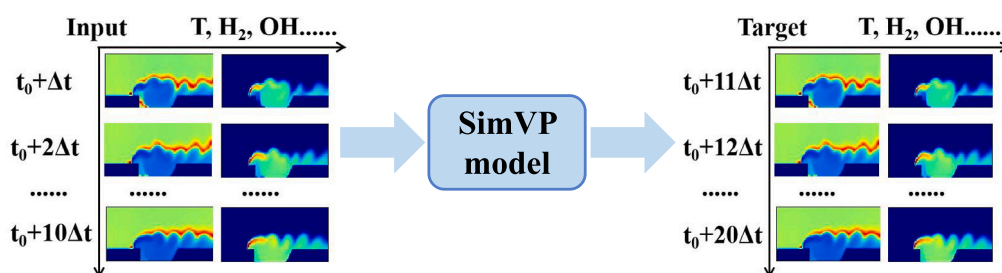
Sipei Wu<sup>a</sup>, Haiou Wang<sup>b,\*</sup>, Kai Hong Luo<sup>a,c,\*</sup>

<sup>a</sup> Center for Combustion Energy, Key Laboratory for Thermal Science and Power Engineering of Ministry of Education, Department of Energy and Power Engineering, Tsinghua University, Beijing 100084, China

<sup>b</sup> State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou 310027, China

<sup>c</sup> Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom

## GRAPHICAL ABSTRACT



## HIGHLIGHTS

- Using spatiotemporal forecasting neural network for turbulent combustion surrogates.
- Proposing two training techniques for robust autoregressive long-term prediction.
- Comparing the effects of model architectures and hyper-parameters.
- Achieving spatially and temporally consistent results for turbulent reacting flows.

## ARTICLE INFO

### Keywords:

Turbulent combustion  
Detailed reaction mechanism  
Transient simulation  
Deep neural network  
Spatiotemporal series prediction  
Long-term forecast stability

## ABSTRACT

This paper systematically develops a high-fidelity turbulent combustion surrogate model using deep learning. We construct a surrogate model to simulate the turbulent combustion process in real time, based on a state-of-the-art spatiotemporal forecasting neural network. To address the issue of shifted distribution in autoregressive long-term prediction, two training techniques are proposed: unrolled training and injecting noise training. These techniques significantly improve the stability and robustness of the model. Two datasets of turbulent combustion in a combustor with cavity and a vitiated co-flow burner (Cabra burner) have been generated for model validation. The effects of model architecture, unrolled time, noise amplitude, and training dataset size on the long-term predictive performance are explored. The well-trained model can be applicable to new cases by extrapolation and give spatially and temporally consistent results in long-term predictions for turbulent reacting flows that are highly unsteady.

\* Corresponding authors.

E-mail addresses: [wanghaiou@zju.edu.cn](mailto:wanghaiou@zju.edu.cn) (H. Wang), [k.luo@ucl.ac.uk](mailto:k.luo@ucl.ac.uk) (K.H. Luo).

<https://doi.org/10.1016/j.egyai.2023.100333>

Received 13 September 2023; Received in revised form 15 December 2023; Accepted 20 December 2023

Available online 27 December 2023

2666-5468/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

High-fidelity turbulent combustion simulation is an essential tool for rapid prototyping, design optimization, and real-time control of combustion systems. However, the complex interactions between turbulence and combustion pose significant challenges in accurately simulating turbulent combustion. The high costs associated with high-fidelity computational fluid/flame dynamics (CFD) have hindered its practical implementation in combustion-related applications. This study aims to develop transient surrogate models for turbulent combustion using transient simulation data and deep neural networks with relatively low computational costs.

In recent years, many investigations have emerged in constructing surrogate models using deep learning architectures to achieve fast solutions [1]. Considering the computationally expensive nature of reconstructing turbulence and combustion models in a traditional way [2], the data-driven method that can implicitly capture large-scale dynamics on coarse grids and use coarse time resolutions is one of the most promising approaches [3]. These data-driven methods integrate local and global information instead of relying solely on single-point data to make autoregressive predictions of the physical properties and species distribution.

In the field of fluid mechanics, there have been many attempts to construct surrogate models for laminar or turbulent flows. Guo et al. [4] proposed a general and flexible approximation model for real-time prediction of non-uniform steady laminar flow in a 2D or 3D domain based on convolutional neural networks (CNNs), which is two orders of magnitude faster than a GPU-accelerated CFD solver. Similar methods were employed to reconstruct steady flow fields around different shapes, including cylinders [5], airfoil wings, [6,7], and automotive bodies [8], yielding promising outcomes. To predict the unsteady flow fields over a circular cylinder, Lee et al. [9] trained four deep neural networks and found that generative adversarial networks (GANs) without physical loss functions is the best for achieving resemblance to the ground truth flow field during recursive predictions. To simulate the transient dynamics of air around the cross-section of an aircraft wing, Pfaff et al. [10] introduced MeshGraphNets, a framework for learning mesh-based simulations using graph neural networks (GNNs).

Due to the inherent fluctuations in turbulence, it is challenging to achieve high-precision alignment at each frame of the predicted long physical field sequence. Therefore, surrogate models are usually trained with the goal of statistical similarity or reconstructing flow fields based on low-fidelity information. For turbulent flow over a backward facing step at different Reynolds numbers and turbulent wake behind an array of bluff bodies, Geneva et al. [11] introduced a novel multi-fidelity deep generative model with recurrent LSTM connections that is able to generate unique yet physically accurate turbulent fluid flows conditioned on an inexpensive low-fidelity solution. To model the Rayleigh-Bénard convection flow, Wang et al. [12] proposed Turbulent-Flow Net (TF-Net) that combines trainable spectral filters and Unet. TF-Net predicted the next 60 frames from the starting one frame and achieved significant reductions in error. For the solution of the Navier-Stokes equation, the Fourier Neural Operator (FNO) method proposed by Li et al. [13] successfully predicted turbulent flows with zero-shot super-resolution. More improved versions have been proposed to improve accuracy [14], adapt to irregular physical domain [15], and scale to deeper networks [16]. Stachenfeld et al. [3] designed Dilated ResNet, a dilated convolutional network that is capable of learning a range of challenging chaotic and turbulent dynamics at low resolution. They found that using training noise and temporal downsampling improved the stability and accuracy of extended rollouts. Yousif et al. [17] developed a transformer-based model to generate turbulent inflow conditions for spatially developing turbulent boundary layer (TBL) simulations. The predicted instantaneous velocity fields exhibit detailed fluctuations and reproduce the turbulence statistics and spatial and

temporal spectra with commendable accuracy compared with the direct numerical simulation (DNS) results.

There are only limited results for turbulent combustion surrogates. An et al. [18] designed an optimized CNN inspired by a Unet architecture and inception module, CFDNN, and trained on the simulation results of hydrogen combustion in a cavity with different inlet velocities. The CFDNN results of spatial distributions and temporal dynamics show excellent agreement with OpenFOAM results. Ren et al. [19] proposed a CNN-LSTM model that could accurately predict the evolution of freely propagating and boundary layer flames. The predicted fuel mass fraction and reaction rate statistics agree well with the DNS data. Wang et al. [20] developed a common kernel-smoothed proper orthogonal decomposition (CKSPOD)-based surrogate model for emulation of large eddy simulation (LES) of turbulent flows and combustion at supercritical pressure. The CKSPOD-based framework can faithfully capture the salient features of its LES counterpart.

The multi-component and multi-scale nature of turbulent combustion simulation incorporating detailed reaction mechanisms leads to complex dynamics. To lower short-term prediction errors, the neural network architecture should fully extract spatiotemporal features of a given input sequence. When dealing with a turbulent combustion simulation dataset that contains thousands of physical field snapshots, the accumulation of errors during long-term predictions may produce inaccurate and unrealistic predictions. Additionally, the efficacy of deep learning techniques still needs to be verified on turbulent combustion simulation data comprising intricate geometries and rapid fluctuations.

In this study, a surrogate model is constructed based on a state-of-the-art spatiotemporal forecasting neural network to simulate the turbulent combustion process in real time. Two training techniques are proposed to address the shifted distribution in autoregressive long-term prediction. We generate two datasets of turbulent combustion for model validation: one for a combustor with cavity [21] and another for a vitiated co-flow burner (Cabra burner) [22]. The study focuses on the robustness of long-term predictions and the generalization ability of extrapolation cases for unsteady turbulent reacting flows.

## 2. Data preparation

### 2.1. Configuration of the combustor with cavity

First, hydrogen combustion in the 2D combustor configuration with cavity is modeled using unsteady Reynolds-averaged Navier-Stokes (URANS). The configuration serves as a simplified prototype for gas turbine combustors and scramjet engines [23], with broad applications in energy conversion and aerospace propulsion systems. The recirculation zone in the cavity decelerates the inflow velocity, achieving flame stability and facilitating successful ignition. This non-intrusive geometry improves combustion efficiency, reduces total pressure losses, minimizes aerodynamic heating, and converts fuel chemical energy into useful flow enthalpy within an appropriate combustor length [21].

As shown in Fig. 1(a), the geometric characteristics include five parameters [24]:  $L_{inj}$  (distance between the upstream nozzle and the cavity's leading edge),  $D$  (cavity depth),  $L/D$  (length-to-depth ratio),  $\theta$  (rear wall 'closeout' angle relative to the cavity floor), and  $D_r$  (depth of the rear wall). For this study, we set  $D = 10$  mm,  $\theta = 45^\circ$ , and  $D_r = 10$  mm for upstream injection. The hydrogen nozzle width is 2 mm, and the air inlet width is 30 mm. Our objective is to create a turbulent combustion dataset for the upstream injection of hydrogen fuel in the combustor with cavity for various geometries and nozzle positions. We vary the geometry by changing  $L/D$  within the range of [2.5, 7] and the nozzle position by altering  $L_{inj}$  within  $[0.25D, 1.5D]$ . We randomly generate 64 operating conditions, with 54 cases for training and 10 cases for testing.

The inflow boundary conditions used in our study are presented in Table 1. All simulations are performed using reactingFoam of the

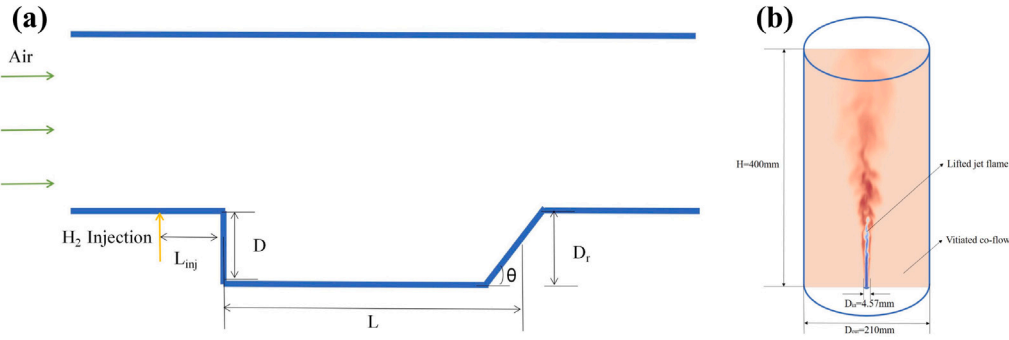


Fig. 1. The computational domain of (a) the combustor with cavity and (b) the Cabra burner.

**Table 1**  
Inflow boundary conditions for URANS of the combustor with cavity.

	$Y_{H_2}$	$Y_{O_2}$	$Y_{N_2}$	Velocity (m/s)	Temperature (K)	Pressure (MPa)
Air	–	0.23	0.77	200	1600	1.6
Fuel	1	–	–	500	300	1.6

OpenFOAM platform [25]. The chosen turbulence model is the k- $\Omega$  SST model, ideal for high-speed compressible flows due to its accurate representation of flow inside and outside the boundary layer. The turbulence-chemistry interaction is modeled using the Eddy-Dissipation-Concept (EDC) combustion model. The GRI-Mech 3.0 mechanism [26] is employed to simulate nitrogen oxides accurately. The grid number is  $600 \times 160$  in the streamwise and wall-normal direction, and the grid is refined in the near-wall region. A time step of  $5 \times 10^{-8}$  s is used to ensure a maximum Courant number below 0.3. Each case runs for 4 ms, resulting in 800 frames for the analysis. Raw data is then resampled to a uniform grid of  $128 \times 64$  for subsequent training.

## 2.2. Configuration of the vitiated co-flow burner

Next, a more challenging configuration of the Cabra burner is considered using LES. The Cabra burner was experimentally studied to understand the mechanism of flame stabilization in high-temperature environments. Note that in gas turbine combustion chambers, hot recirculating combustion products contribute to flame stabilization [27]. The experimental setup of the Cabra burner was detailed in [22]. The central jet consists of a lifted  $H_2/N_2$  jet flame, while the co-flow consists of hot combustion products from a lean premixed  $H_2$ /air flame. This simplified burner allows for a decoupling of chemical kinetics, heat transfer, and molecular transport from the complex recirculating flow, enabling a study of reaction flow under well-defined homogeneous boundary conditions.

The simulation parameters are listed in Table 2. To create a turbulent combustion dataset for different boundary conditions, 14 temperatures are randomly selected between 1000 K and 1100 K as the inlet temperature for the co-flow. Ten cases are used for training, and the remaining 4 cases are reserved for testing. Simulations were performed using reactingFoam of the OpenFOAM platform [25], with the Smagorinsky model for subgrid turbulent kinetic energy estimation and the EDC combustion model for turbulence-chemical interaction. A 9-species mechanism for hydrogen combustion [28] was employed. The computational domain is a coaxial cylinder with a height of 400 mm, inner diameter of 4.57 mm, and outer diameter of 210 mm, as shown in Fig. 1(b). The grid number is 63, 40, and 200 in the radial ( $r$ ), azimuthal ( $\theta$ ), and axial ( $x$ ) directions, respectively, and the grid is refined near the central axis. The time step is set to  $1 \times 10^{-6}$  s to ensure a maximum Courant number below 0.3. The simulation duration is 0.1 s, equivalent to approximately 30 convective cycles. Each case generates 1000 snapshots for the analysis. Twenty  $x-r$  planes are sampled to generate a series of matrices with a grid number of  $64 \times 64$ , which serve as training data for the neural network.

## 3. Deep learning framework

### 3.1. Model architecture

In this study, spatiotemporal sequence prediction models are employed and trained using transient physical field sequences obtained from URANS/LES simulations. This approach is a data-driven method that has recently shown its broad application potential in various domains, such as video prediction [29], traffic flow prediction [30], precipitation prediction [31], weather forecasting [32], and sea surface temperature prediction [33]. However, its utilization in turbulent combustion is scarce. Various models, including the SimVP model, Unet model, ConvLSTM model and Fourier Neural Operator (FNO) model, are considered in the present work, which are described below.

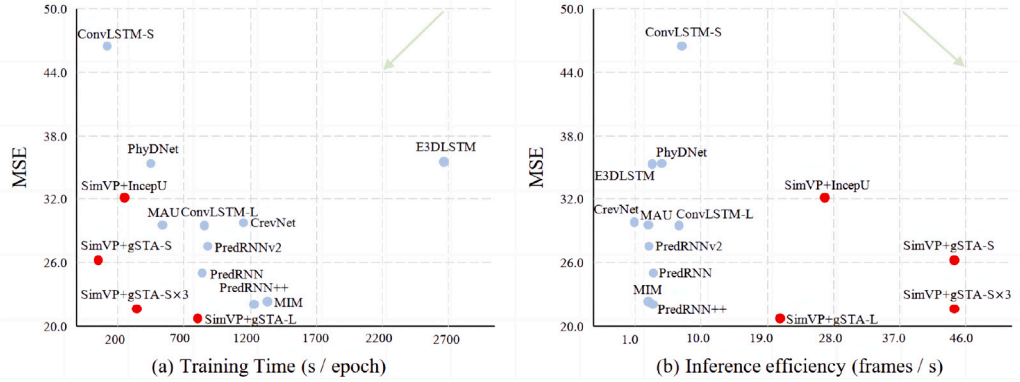
#### 3.1.1. SimVP model

Tan et al. [34,35] compared the testing errors, training time, and inference time on the benchmark dataset Moving MNIST [36] using different models. It demonstrates that SimVP can achieve relatively high testing accuracy with relatively small training costs, making it a preferable choice for spatiotemporal sequence prediction models, as shown in Fig. 2 [35]. The superior performance of SimVP for more complex tasks, including traffic flow forecasting, climate prediction, road driving, and human motion prediction, is also demonstrated through extensive experiments [35]. Note that turbulent combustion is featured by multi-component and multi-scale, necessitating the neural network to have multiple channels and a global receptive field. SimVP, with large kernel convolution, effectively extracts features from both local and distant receptive fields and conducts channel-wise interactions to address the challenges posed by multi-channel and multi-scale complexities. Dividing the large kernel convolution operation into three parts allows SimVP to extract features from local and distant receptive fields and perform channel-wise interactions more efficiently. In contrast, other models typically employ smaller  $3 \times 3$  convolution kernels. It should be noted that these scatter plots only illustrate the accuracy of short-term predictions. The potential for error accumulation in an uncontrolled manner during long-term predictions is not clear yet.

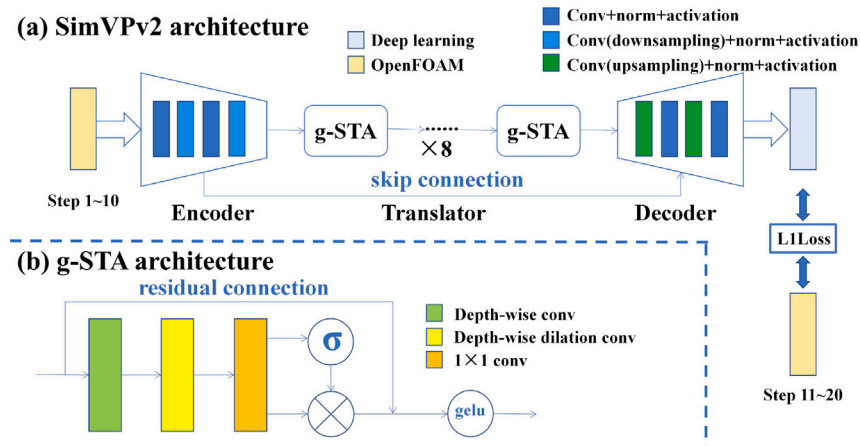
The SimVP model consists of three main components: the Encoder, Translator and Decoder, as illustrated in Fig. 3(a). The Encoder-Decoder architecture is commonly employed when dealing with similar input and output dimensions. In the context of the transient combustion surrogate modeling problem, where the input and output comprise physical fields with identical dimensions at different time steps, an Encoder-Translator-Decoder structure is adopted, which includes Encoder, Translator and Decoder. The Encoder extracts spatial features from the input physical fields of several preceding time steps, the Translator integrates spatiotemporal features, and the Decoder reconstructs the physical fields of several subsequent time steps. The Encoder and Decoder comprise four convolutional and deconvolutional modules with a hidden channel size of 64. Each convolutional module is composed of a convolutional layer, a normalization layer and an activation

**Table 2**  
Inflow boundary conditions for LES of the vitiated co-flow burner.

	$Y_{H_2}$	$Y_{O_2}$	$Y_{N_2}$	$Y_{H_2O}$	Velocity (m/s)	Temperature (K)	Pressure (Pa)
Central jet	0.17095	–	0.97618	0.06452	107	305	101 325
Co-flow	–	0.17095	0.76453	–	3.5	1045	101 325



**Fig. 2.** The performance of SimVPs on the Moving MNIST dataset. The variants of SimVP are denoted in red color. For the training time, the less the better. For the inference efficiency (frames per second), the more the better. The light green arrow indicates the direction of model optimization [35]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 3.** SimVP (version 2) architecture.

layer. Additionally, the model incorporates skip connections between the Encoder and Decoder, inspired by the Unet architecture [37]. The same encoding and decoding networks are utilized for the input and output 10-step fields.

The innovative aspect of the SimVP model lies in the Translator component, which introduces a series of gated spatiotemporal attention modules (g-STA). The structure of a single g-STA module is shown in Fig. 3(b). This design is inspired by recent findings that large kernels in CNNs outperform smaller ones, particularly in downstream tasks, offering superior results to vision transformers (ViTs) with enhanced inference speed [38]. In contrast to small-kernel CNNs, large-kernel CNNs have much larger effective receptive fields and higher shape bias rather than texture bias [38]. In other words, large-kernel CNNs can capture more global structure and shape information in the input image, as opposed to merely local texture information. For the transient combustion physical fields, the value of each point in the combustion fields to be predicted is related to the entire combustion fields in the preceding time step rather than being influenced by only a few local points. Large kernels provide a more efficient way to integrate global information and reconstruct the structure of the combustion physical fields. However, using large kernel convolution directly imposes a computational inefficiency and results in numerous parameters. To

address this, the large kernel convolution is decomposed into (1) a depth-wise convolution to capture local receptive fields within a single channel, (2) a depth-wise dilation convolution to capture distant receptive fields within a single channel, (3) a  $1 \times 1$  convolution to perform channel-wise interactions. Further, the output of the above large kernel convolution operation is split into two parts and takes one of them with a sigmoid function as an attention gate. The Translator provided in the official code of the paper consists of eight g-STA modules, and the hidden layer channel number of g-STA is 512. After testing, we found that this network structure can also efficiently capture long-range correlations in both spatial and temporal perspectives in our turbulence combustion datasets.

### 3.1.2. Unet model

The classic Unet model [37] is adopted in this study as a baseline model, which has been widely applied not only in the field of image segmentation but also in future frames regression tasks [12,18]. Specifically, the SimVP model is transformed into the Unet model by removing the middle Translator and keeping only the Encoder-Decoder structure. To simultaneously extract features from 10 steps of the physical fields, the channel axis is merged with the time axis, resulting in a tenfold increase in the number of channels. The encoding

and decoding parts consist of eight convolutional and deconvolutional layers, with a hidden layer channel number of 256.

### 3.1.3. ConvLSTM model

The ConvLSTM model, originally proposed by Shi et al. [31], is widely applied in spatiotemporal sequence prediction. By incorporating a convolutional process into the LSTM module, the ConvLSTM model can exploit the spatiotemporal characteristics of the input sequence. In this study, a surrogate model is built using four layers of ConvLSTM modules, each featuring 64 hidden units.

### 3.1.4. FNO model

In addition, the recently proposed Fourier Neural Operator (FNO) [13] is studied. This model's architecture, which combines Fourier transforms with linear mapping and inverse Fourier transforms, is distinct from conventional CNNs or RNNs. We use the FNO-3D model, which integrates temporal (1D) and spatial (2D) dimensions, to capture spatiotemporal dynamics. Both the vanilla FNO [13] and UFNO models [14] use 3D Fourier transform and inverse Fourier transform for temporal prediction. Our surrogate model uses six Fourier operator module layers, each with 36 hidden units, and maintains 5, 20, and 20 modes along the  $t$ ,  $x$ , and  $y$  directions, respectively.

## 3.2. Data preprocessing and model training

For data preprocessing, maximum absolute normalization is initially applied, constraining each field within the range of 0~1. Then, exponential transformation is implemented across all components, assigning a relatively higher weight to radicals to prevent them from being overlooked. This process is expressed as follows.

$$\widetilde{u}_{i,j,k,l} = (u_{i,j,k,l})^\alpha, 0 < \alpha < 1, \quad (1)$$

$$i = 1 \dots C, j = 1 \dots X, k = 1 \dots Y, l = 1 \dots T$$

Here, the fourth-order tensor  $u_{i,j,k,l}$  is the raw data and  $\widetilde{u}_{i,j,k,l}$  is the normalized data, where  $C$  is the number of channels (i.e., the number of physical fields),  $X$  and  $Y$  indicate the spatial resolutions and  $T$  is the number of time steps.  $\alpha$  is an exponential factor that ranges from 0~1. This transformation mirrors the Box-Cox transformation in its aim to normalize the distribution. The work of Zhang et al. [39] indicates that the Box-Cox transformation can be applied to capture the multi-scale distributions of different species and species at different times. In this study, the exponential transformation is designed to balance the weights of minor and major components, negating significant magnitude differences. Tests suggested that an exponent of 0.5 is optimal for our tasks.

The physical fields predicted by deep learning are exactly the same as those solved by OpenFOAM, namely the fields of velocity, temperature, pressure, mass fraction of each component. This study uses the Mean Absolute Error (L1Loss) as the loss function, commonly used in regression training tasks. We train the models using the Adam optimizer [40] with the OneCycle learning rate scheduler [41]. As for the training cost, the training process for the whole dataset requires about ten hours on one NVIDIA Tesla V100 GPU, and the inference process for one case costs less than five seconds. In contrast, each URANS/LES case described in Section 2 requires about 400 CPU hours on AMD EPYC 7742 processors. A well-trained neural network can perform inference three orders of magnitude faster than parallel simulations, meeting the requirements of real-time simulation.

## 3.3. Methods for long-term prediction

### 3.3.1. Shifted distribution problem

The present study emphasizes the long-term prediction accuracy of the models, specifically for ultra-long-term dynamics in combustion fields. Traditional tasks, like video sequence prediction and precipitation forecasting, are short-term due to high randomness in the real

world. In contrast, the turbulent combustion data of the present work is obtained by numerically solving a series of governing equations, which can be considered a deterministic forecasting problem. Therefore, by selecting appropriate training methods, the spatiotemporal sequence prediction model can learn the dynamic behavior of various physical fields in turbulent combustion.

Using the original one-step training strategy, neural network models perform poorly in long-term forecasts, even yielding unphysical results. The possible reasons for this phenomenon are mainly twofold:

First, the dynamic trends of the physical field data under different operating conditions vary significantly, with highly uneven spatiotemporal distributions and pronounced oscillations. For example, some rapidly generated and consumed radicals and LES data with rapid fluctuations. For such data, the error in short-term prediction is already significant, making it challenging to perform longer-term forecasts. Reducing the deep learning step size could enhance prediction accuracy.

Second, there are also some slowly evolving fields that, despite having high accuracy in short-term prediction, experience a drastic decrease in accuracy after dozens of steps. Suitable training methods could minimize this long-term prediction error.

While performing long-term forecasting using an autoregressive approach, the inputs of the surrogate model are based on previous predictions, which can lead to cumulative prediction errors. When the error reaches a critical level, there will be distributions that differ significantly from the training set. This unfamiliar input can lead to incorrect predictions by the surrogate model, a phenomenon known as the shifted distribution problem [42]. Therefore, it is necessary to adopt more reasonable training methods to enhance the stability and robustness of neural networks in long-term forecasting. This article proposes two training methods, namely, unrolled training and injecting noise training, to improve the stability and robustness of surrogate models in long-term prediction, which are discussed in detail in the following.

### 3.3.2. Unrolled training

To address the shifted distribution problem in long-term prediction, we propose a direct approach to imitate the process of repeatedly using the output as input for prediction. This training approach requires the neural network to predict the simulated results accurately after multiple iterations.

We propose an unrolled training approach using model parameter sharing [43]. The training process for  $n$  unrollings is illustrated in Fig. 4(a) and involves the following steps. First, a segment of  $(n+1) \times 10$  steps is randomly selected from a simulation data sequence containing hundreds of snapshots. Then, the neural network model takes steps 1~10 of the segment as inputs and steps 11~20 as labels to calculate the first part of the loss function. Next, the predicted steps 11~20 from the neural network become new inputs fed into the same neural network. Steps 21~30 of the segment now serve as the new labels to compute the second part of the loss function. This process is repeated  $n$  times, resulting in a  $n$ -stage loss function. Finally, the loss function is backpropagated using the gradient descent algorithm to update the parameters of each layer in the neural network.

During the inference process (Fig. 4(b)), the model predicts the physical field sequence for steps 11~20 using the data from steps 1~10 as input. Subsequently, the predicted results for steps 11~20 become the new input to predict the sequence for steps 21~30. This iterative process continues until the maximum time step of the simulation labels is reached.

In practical tests, it has been observed that this multi-stage loss function can be complex, causing slow convergence and a higher risk of getting stuck in local optima when unrolling multiple times. This study suggests using a pre-trained model with one-step training as the initial weights to address this issue. Adopting this approach reduces training costs significantly, and satisfactory results can be achieved through fine-tuning via unrolled training for no more than five epochs.

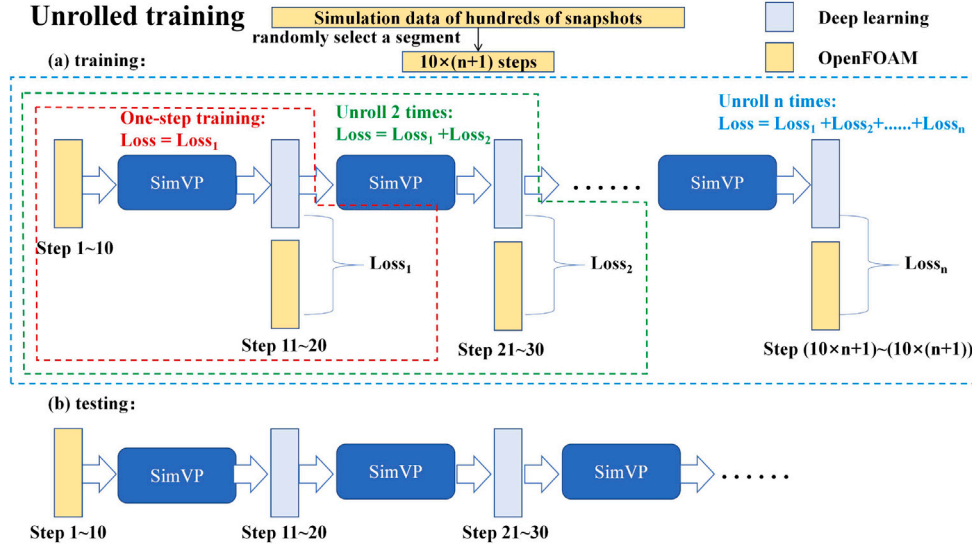


Fig. 4. Illustration of (a) training and (b) testing procedures for unrolled training.

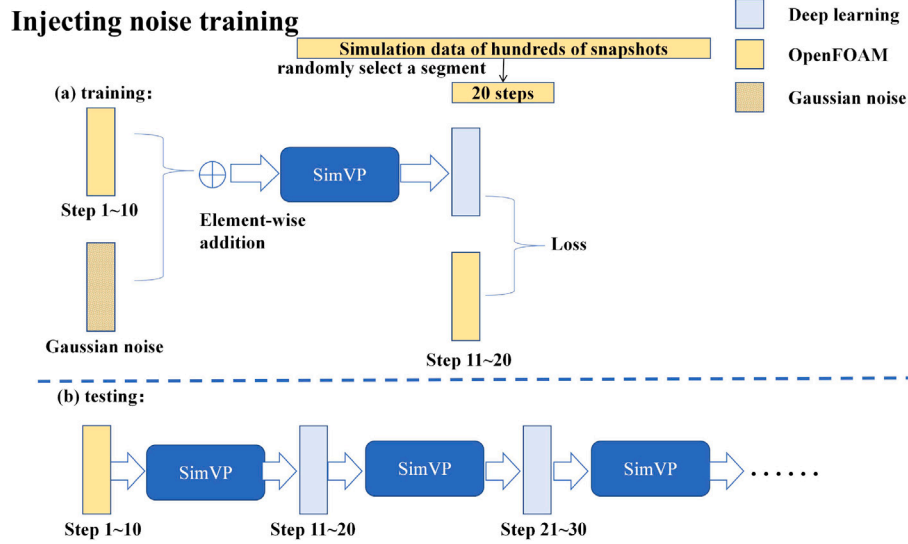


Fig. 5. Illustration of (a) training and (b) testing procedures for injecting noise training.

### 3.3.3. Injecting noise training

Another training method addresses the shifted distribution problem by injecting Gaussian noise into the data. During training, Gaussian noise is added to the input while keeping the output unchanged, requiring the neural network to produce correct outputs based on noisy inputs. This approach makes the training data distribution closer to the distribution with accumulated error that occurs during autoregressive inference, enabling deep learning models to achieve better performance in long-term predictions.

Noise injection is a widely used training technique in neural networks. It has been applied to various areas, including early denoising autoencoders [44], addressing over-smoothing in graph neural networks [45], and improving prediction accuracy in molecular simulation data [46]. Additionally, noise injection has been utilized in domains such as airfoil flow fields [47] and turbulent simulation data [3].

The training and testing procedures for injecting noise are depicted in Fig. 5. Independently and identically distributed Gaussian noise tensors are generated and element-wisely added to each input tensor. The processed tensor is then fed to the neural network for prediction. The prediction results are compared with the simulation labels to

calculate the loss function, and model parameters are updated through backpropagation.

There are several ways to add Gaussian noise, as mentioned in [48], including:

- Uniform noise

$$u_{i,j,k,l} = u_{i,j,k,l} + \varepsilon, \quad \varepsilon \sim N(0, \sigma) \quad (2)$$

- Channel-wise noise

$$u_{i,j,k,l} = u_{i,j,k,l} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma_i) \quad (3)$$

- Element-wise noise

$$u_{i,j,k,l} = u_{i,j,k,l} + \varepsilon_{i,j,k}, \quad \varepsilon_{i,j,k} \sim N(0, \sigma_{i,j,k}) \quad (4)$$

Applying uniform noise to all physical fields may mask the original values of minor components. In addition, applying element-wise noise would require preserving a larger number of preprocessing parameters. However, test results indicate that this approach does not enhance long-term prediction accuracy. The difference between combustion surrogate models and flow field surrogate models lies in the fact that many components must be predicted, and each physical field has its unique

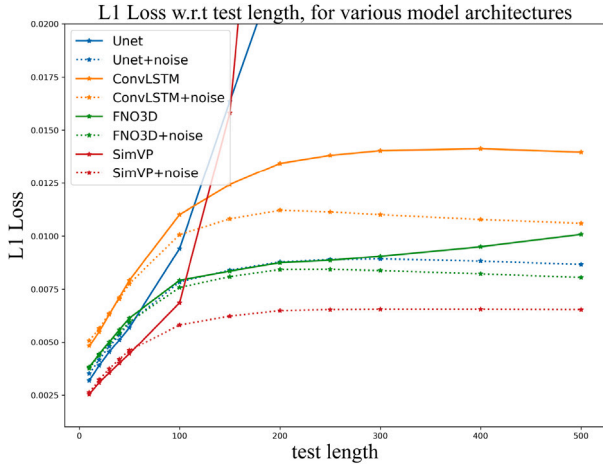


Fig. 6. Long-term predictive performance under various model architectures.

dynamic trend. Therefore, channel-wise noise is the most appropriate approach.

The standard deviation for injecting Gaussian noise at the  $i$ th field is calculated as  $\sigma_i = k \cdot S_i$ , where  $S_i$  represents the standard deviation for all moments and pixels of the  $i$ th field and  $k$  is a constant. The choice of an appropriate noise amplitude is closely related to the model structure and the dataset. This study compares the long-term predictive performance with three values for  $k$ , i.e. 0.002, 0.02, and 0.2, for conceptual demonstration purposes.

## 4. Results and discussion

### 4.1. 2D URANS of the combustor with cavity

This section presents comparative experiments to investigate the impact of different training schemes and hyperparameter settings on the long-term predictive performance of ten unseen test cases. The primary factors studied include model architecture, the number of unrollings, the noise amplitude, and the training dataset's size. We select steps 20~30, 100~110, and 180~190 as inputs to predict the subsequent 500 steps of the physical fields. The mean absolute error is calculated and averaged on ten unseen cases.

#### 4.1.1. The predictions with different model architectures

Fig. 6 illustrated the long-term predictive capabilities of four models (SimVP, Unet, FNO-3D, and ConvLSTM) using one-step training (solid lines) and injecting noise training (dotted lines). Each model uses an optimal noise magnitude (0.02 for SimVP and ConvLSTM, 0.2 for Unet, and 0.002 for FNO-3D) to ensure a fair comparison. Overall, each

model achieves stable long-term predictions by incorporating training techniques that address shifted distribution issues. Furthermore, sensitivity to distribution shifts varies among architectures: convolutional architecture is the most sensitive, then convolution plus recurrent, followed by Fourier neural operator, whose unique Fourier transform capabilities mitigate distribution shifts. SimVP demonstrates the best short- and long-term performance, owing to its robust feature extraction capabilities and physical field reconstruction abilities. By dividing the large kernel convolution operation into three parts, SimVP extracts the features of local and distant receptive fields and performs channel-wise interactions more efficiently. In contrast, Unet and ConvLSTM employ smaller  $3 \times 3$  convolution kernels, and FNO is essentially a linear mapping of the Fourier space. Due to its superior performance, SimVP is chosen for further hyperparameter comparisons.

#### 4.1.2. The predictions with different unrolled times

Unrolled time is the most crucial training parameter for unrolled training. Fig. 7(a) shows long-term predictive performance for various unrolled times, and labels 'train10' to 'train50' denote one to five unrollings, respectively. With a 10-frame spatiotemporal sequence model, unrolling  $n$  times predicts the next  $10 \times n$  steps during training. More than three unrollings stabilize long-term predictions, while fewer unrollings, although achieving better short-term results, deteriorate quickly beyond 100 steps. Over-unrolling reduces short-term accuracy due to constraints on mid- to long-term predictions during training, but enhances long-term stability. Five unrollings offer the lowest mean absolute error across 500 steps despite the higher computational cost. Multiple unrollings lead to a complex computational process and a multi-stage optimization objective function, making it easier to converge to local optima during training. This might limit short-term prediction performance, but it significantly slows the error accumulation rate in later stages, which is beneficial for stable ultra-long snapshot sequence predictions.

#### 4.1.3. The predictions with different noise amplitudes

In this study, a method of channel-wise Gaussian noise injection is used during training, which is controlled by a predefined scaling coefficient ( $k$ ). Fig. 7(b) examines the long-term predictive performance for different values of  $k$ : 0.002, 0.02, and 0.2. While this training method proves to be suitable, the noise magnitude should not be too small, as even with a noise standard deviation of 0.2% of the field standard deviation, long-term predictions still collapse. Conversely, excessive noise, such as a standard deviation of 20%, complicates training and introduces errors that can obscure the original distribution, leading to slightly faster error accumulation. A noise standard deviation of 2% provides the lowest prediction errors in the short and long term. Note that this method reduces the training time per epoch to one-fifth compared to using five unrollings, facilitating sufficient training. We demonstrate that injecting noise during one-step training achieves similar short- and long-term predictive performance as unrolling five

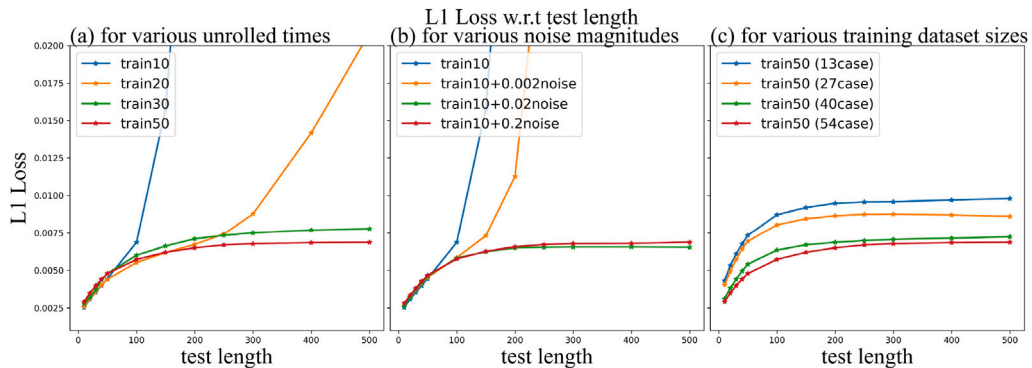


Fig. 7. Long-term predictive performance under various (a) unrolled times, (b) noise amplitudes, (c) training dataset sizes.

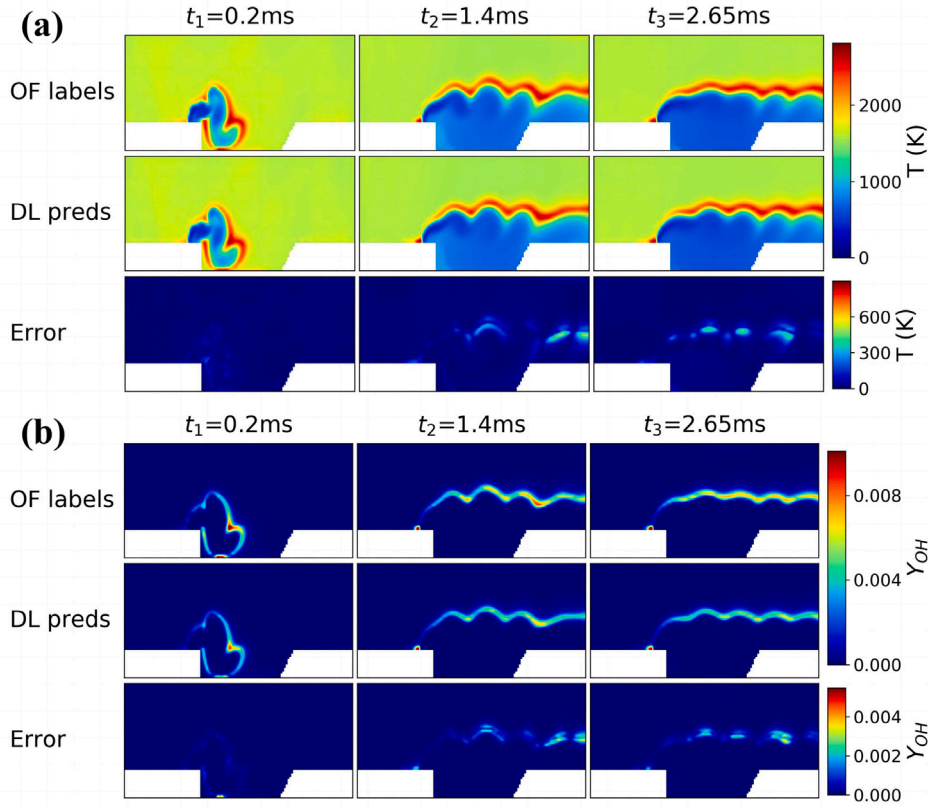


Fig. 8. Comparisons of (a) temperature and (b) OH mass fractions distributions between OpenFOAM and SimVP results at a random cavity geometry.

times for the URANS data. However, combining the two strategies, i.e. injecting noise with several unrollings, leads to slow convergence and high training costs, making it unsuitable for further adoption.

#### 4.1.4. The predictions with different training dataset sizes

In the context of deep learning, matching the dataset size with the model parameters is often desirable. Thus, we investigated the amount of training data required for a given model using fractions of the available training data: 1/4 (13 cases), 1/2 (27 cases), 3/4 (40 cases), and all (54 cases). Fig. 7(c) demonstrates the predictive performance under various training dataset sizes with five unrollings. The effectiveness of the proposed training strategy does not show a significant correlation with the amount of data. Surprisingly, even small-scale training data produces stable predictions over 500 steps by incorporating our training methods. Increasing the amount of data covers a wider feature space, leading to improved generalization and reduced errors on the test set. Utilizing a large-scale dataset (54 cases) enhances capturing the physical evolution and chaotic dynamics of turbulent combustion processes. However, there are limited performance gains between 40 and 54 cases. Therefore, 54 cases are sufficient to train the selected neural network model for this study.

#### 4.1.5. Qualitative and quantitative analysis

This section presents snapshots for several representative physical fields and plots the corresponding temporal evolution curves of the field average (after inverse normalization) and the field relative mean absolute error, aiming to analyze the long-term predictive performance field-wisely. The results presented in this section are obtained from the predictions of the SimVP model on 10 test cases. The training approach incorporates noise injection, with a selection of  $k$  equal to 0.02.

We randomly select a test case to plot the snapshots of various physical fields for further analysis. The snapshots for the next 500 steps are acquired by predicting 50 times autoregressively from steps

21~30. The snapshots of temperature and OH mass fractions for three different time steps are shown in Fig. 8(a) and (b). In each figure, the first row shows the simulation results from OpenFOAM, the second row shows the results predicted by the SimVP model, and the third row displays the absolute error, with all data inverse normalized. These figures show that the deep learning model exhibits good consistency with the OpenFOAM results regarding spatial distribution and temporal dynamics for the predicted values of the main physical quantities. More significant deviations primarily occur near the leading and rear walls of the cavity, as well as in regions with substantial fluctuations.

From a quantitative point of view, the temporal evolution curves of the field averages (after inverse normalization) and the field relative mean absolute errors for temperature, major components ( $H_2$ ,  $H_2O$ ), and interested minor components (OH, NO) fields are plotted. The calculation method for the simulated field average value (OF), predicted field average value (DL), and relative mean absolute error (Error) for the  $c$ th channel at time step  $t$  is given by the following equations:

$$OF_{c,t} = \frac{1}{X \times Y} \sum_{i=1}^X \sum_{j=1}^Y u_{i,j} \quad (5)$$

$$DL_{c,t} = \frac{1}{X \times Y} \sum_{i=1}^X \sum_{j=1}^Y \hat{u}_{i,j} \quad (6)$$

$$Error_{c,t} = \frac{\sum_{i=1}^X \sum_{j=1}^Y |u_{i,j} - \hat{u}_{i,j}|}{\sum_{i=1}^X \sum_{j=1}^Y u_{i,j}} \quad (7)$$

where the subscripts  $c$ ,  $t$  stand for the  $c$ th physical field, the  $t$ th time step.  $u_{i,j}$  represents the label value and  $\hat{u}_{i,j}$  represents the predicted value.

Fig. 9(a) shows that the SimVP model exhibits consistent dynamic trends with the OpenFOAM results for the major physical quantities. Additionally, each field's relative mean absolute errors over time are plotted as shown in Fig. 9(b). It is observed that the errors of the



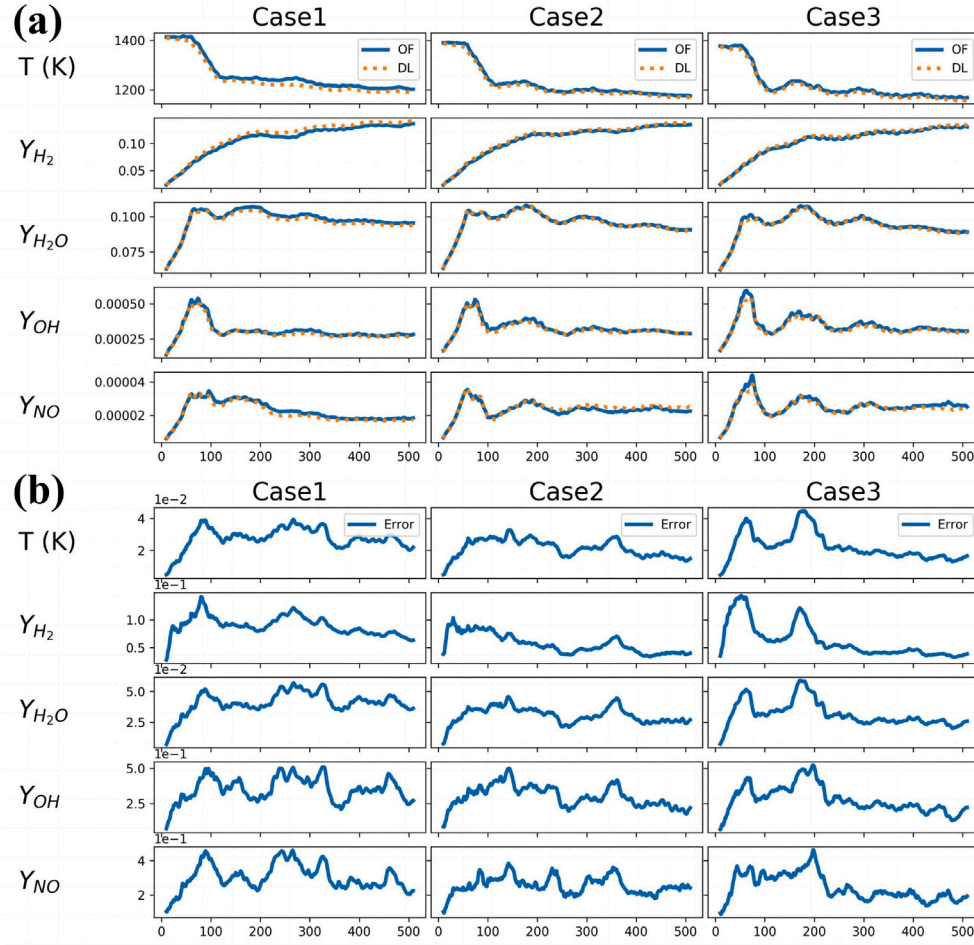


Fig. 9. Dynamic trends of (a) field average values of OpenFOAM simulation and SimVP prediction and (b) relative mean absolute errors for each field over time of three test cases.

main physical quantities are relatively small, while those of minor components are relatively larger. However, these errors remain stable or decrease to low levels as the autoregressive prediction reaches the stage of numerical stability. Overall, SimVP exhibits a reasonable accuracy for predicting physical fields at geometries not used in the URANS training dataset.

#### 4.2. 3D LES of the vitiated co-flow burner

##### 4.2.1. Comparison with the experimental results

Fig. 10(a) displays the computed axial profiles of the mean temperature for LES results (blue line) along the central line compared to the measurements (blue dots). The LES results demonstrate a remarkable concurrence with the measurements. After a distance of several nozzle diameters, the temperature undergoes a linear increment as the jet advances.

Fig. 10(b) exhibits the comparisons of scalars at various axial locations between the LES (blue lines) and the experiment (blue dots). In each row, the scalar's Favre mean and root mean square (rms) properties at two axial locations of  $x/d = 11$  and  $14$  are presented. The selected scalar includes temperature, mixture fractions,  $H_2$ , and OH mass fractions, respectively. The mixture fraction measures the local fuel/oxidizer ratio and is defined as [49].

The general agreement between the LES results and those of the experimental measurements indicates that the present numerical approach applied in the LES code can reasonably reproduce the laboratory flame. This gives us more confidence in applying simulation data with slightly different inlet jet conditions to train the surrogate model.

##### 4.2.2. Qualitative and quantitative analysis

In Section 4.1.1, the SimVP model demonstrates the best short- and long-term performance owing to its robust feature extraction capabilities and physical field reconstruction abilities. Consequently, we select the SimVP model for LES predictions. During training, we unrolled three times with no noise for better long-term predictive performance, as validated in Section 4.1.2. Considering the inherent stochastic turbulence and rapid fluctuations in LES data, the trained model using unrolled training captures more small-scale structures compared to that of injecting noise training.

For a randomly chosen unseen test case, we predict the subsequent 500 steps using steps 191~200 as input. Fig. 11(a) and (b) show the distributions of temperature and OH mass fractions for six different time steps. In each figure, the first row shows the simulation results from OpenFOAM, the second row shows the results predicted by the deep learning model, and the third row displays the absolute error, with all data inverse normalized. The plots show that points with more significant prediction errors mainly locate near the central jet. The substantial gradients and fluctuations of various physical quantities at the interface of the central jet and the co-flow make accurate predictions more challenging.

Due to the stochasticity and uncertainty of turbulent inlet conditions, expecting a perfect alignment between the deep learning predicted frames and OpenFOAM is unreasonable. However, achieving similarity in statistical quantities is a more realistic objective. Fig. 12(a) illustrates the comparisons of scalars at various axial locations between the LES (blue lines) and the SimVP predictions (orange dots). Each row

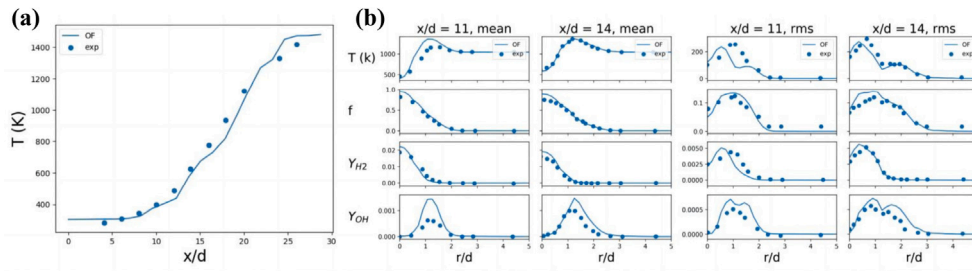


Fig. 10. Comparisons of (a) axial profiles of the mean temperature along the central line and (b) scalars at various axial locations between the LES and the experiment.

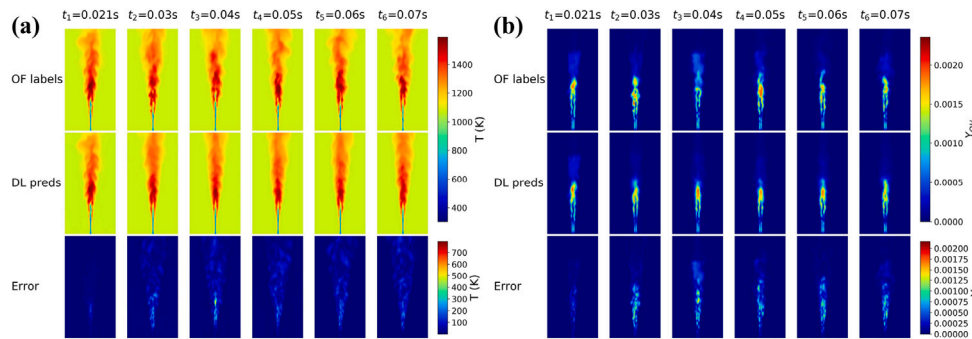


Fig. 11. Comparisons of (a) temperature and (b) OH mass fractions distributions between OpenFOAM and SimVP results at a random inlet temperature.

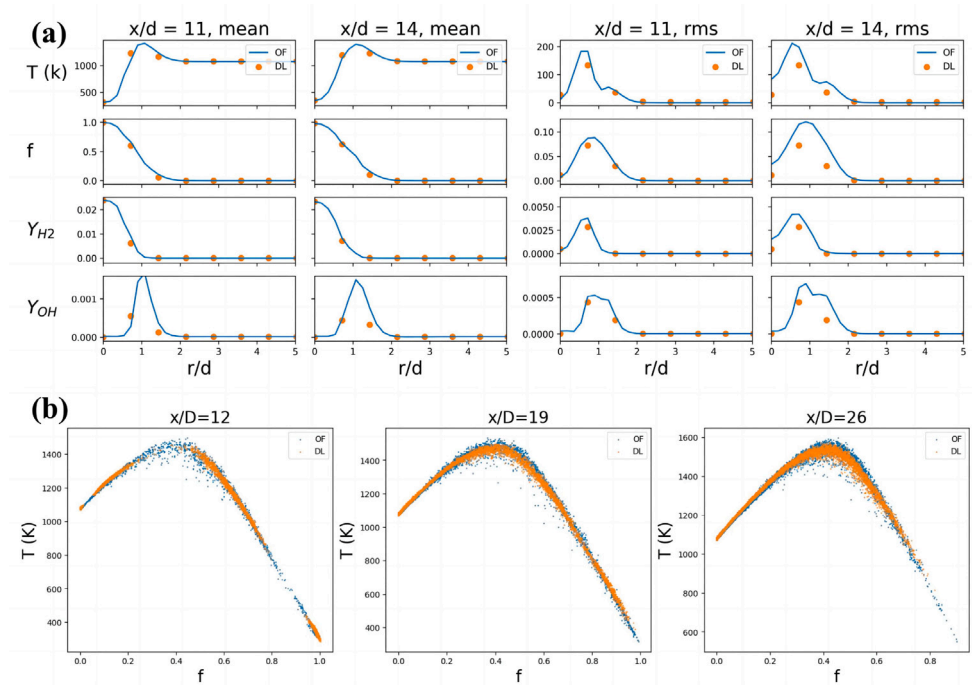


Fig. 12. (a) Comparisons of scalars at various axial locations between OpenFOAM and SimVP results. (b) Scatter plots of temperature and mixing fraction at three axial sections  $x/D = 12, 19,$  and  $26$  using both OpenFOAM simulation and SimVP prediction.

presents one scalar’s Favre mean and root mean square (rms) properties at two axial locations,  $x/d = 11$  and  $14$ . It can be observed that the Favre mean aligns remarkably well, while there is a slight deviation in rms, indicating an overall tendency for the proposed method to underestimate turbulent fluctuations in the LES data. This discrepancy can be attributed to the significant variations in the input data distribution under different inlet conditions and at various time frames. From a statistical perspective, this introduces increased variability in the data distribution of high-dimensional input physical fields, consequently

diminishing long-term prediction accuracy. Moreover, due to the lack of detailed information of inlet velocity fluctuations, it is difficult for neural network to accurately maintain the turbulence fluctuation for a long time, which eventually leads to the underestimation of rms. Future work can focus on domain adaptation of data distribution and incorporation of inlet velocity fluctuation information when training neural networks.

To illustrate the flame structure, the temperature ( $T$ ) and mixture fraction ( $f$ ) are depicted on scatter plots at three specifically chosen

cross sections:  $x/D = 12, 19,$  and  $26,$  as shown in Fig. 12(b). Blue dots represent OpenFOAM simulations, while orange dots depict SimVP results. After several hundred steps, the predictions exhibit certain deviations in the flame structure at the fuel-rich side ( $f$  close to 1) but align better at the fuel-lean side ( $f$  close to 0). This phenomenon bears similarities to the deviations in predicting the central jet upstream region shown by the snapshots. The highest points on the scatter plot indicate that the SimVP predictions closely mirror the OpenFOAM simulation results. As the axial distance increases, the peak of the scatter plot gradually shifts to the right.

## 5. Implications for long-term turbulent combustion prediction

In the URANS case, the proposed surrogate model framework can accurately perform long-term forecasting. The predictions demonstrate high spatial distribution and temporal dynamics consistency with OpenFOAM simulations, even for unseen geometries and boundary conditions.

In the LES case, the proposed surrogate model framework provides physics-consistent snapshots, statistical values, and flame structures over extended durations. However, finer grids will exhibit more detailed and small-scale structures, posing more challenges for the long-term prediction of high-fidelity turbulent combustion data.

The combustor with cavity exhibits a complex geometry, whereas the Cabra burner has a simple axisymmetric one. Despite this complexity difference, the long-term predictive performance of the combustor with cavity surpasses that of the Cabra burner across various metrics. This highlights a significant disparity between slowly-evolving URANS data and fast-evolving LES data, indicating that long-term prediction of LES and DNS data with rapid fluctuations is notably more challenging than that of URANS data.

## 6. Conclusion and further scope

The development of high-fidelity turbulent combustion surrogate models is a fundamental infrastructure for downstream tasks such as rapid prototyping, design optimization, and digital twins. The deep learning model offers a viable solution for efficient turbulent combustion surrogate models, as it can implicitly capture large-scale dynamics on coarse grids and with coarse time resolutions, resulting in bottleneck inference speed.

Two datasets of turbulent combustion, comprising a combustor with cavity and a Cabra burner, are generated for model validation. For the accurate short-term prediction of the turbulent combustion process with detailed reaction mechanisms, we adopt the state-of-the-art spatiotemporal predictive neural network, which outperforms other mainstream convolutional, recurrent neural networks and neural operator models. The main challenge in long-term prediction stems from the adaptive handling of shifted distributions during autoregressive forecasting to avoid unphysical prediction outcomes. This study proposes two training approaches, namely unrolled training and injecting noise training. It compares the impact of model architecture, unrolled time, noise amplitude, and training dataset size on long-term predictive performance. The well-trained model can generalize to extrapolation cases and give spatially and temporally consistent results in long-term predictions for turbulent reacting flows that are highly unsteady.

Future work is needed to explore methods for predicting the fast-evolving combustion field with multi-scale information, such as integrating transfer learning techniques and designing deep neural networks using the frequency principle. Furthermore, attempts should be made to incorporate physical constraints to constrain the predictive outcomes of the models, making them more realistic and consistent with the laws of physics.

## CRediT authorship contribution statement

**Sipei Wu:** Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Haiou Wang:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Writing – review & editing, Formal analysis. **Kai Hong Luo:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Writing – review & editing, Formal analysis.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

The authors gratefully acknowledge the support from the National Natural Science Foundation of China (Grant No. 52250710681 and 52022091). Support from the UK Engineering and Physical Sciences Research Council under the project “UK Consortium on Mesoscale Engineering Sciences (UKCOMES)” (Grant No. EP/X035875/1) is also acknowledged. This work made use of computational support by CoSeC, the Computational Science Centre for Research Communities, through UKCOMES.

## References

- [1] Zhang X, Wang L, Helwig J, Luo Y, Fu C, Xie Y, Liu M, Lin Y, Xu Z, Yan K, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. 2023, arXiv preprint arXiv:2307.08423.
- [2] Ihme M, Chung WT, Mishra AA. Combustion machine learning: Principles, progress and prospects. *Prog Energy Combust Sci* 2022;91:101010.
- [3] Stachenfeld K, Fielding DB, Kochkov D, Cranmer M, Pfaff T, Godwin J, Cui C, Ho S, Battaglia P, Sanchez-Gonzalez A. Learned coarse models for efficient turbulence simulation. 2021, arXiv preprint arXiv:2112.15275.
- [4] Guo X, Li W, Iorio F. Convolutional neural networks for steady flow approximation. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, p. 481–90.
- [5] Ribeiro MD, Rehman A, Ahmed S, Dengel A. DeepCFD: Efficient steady-state laminar flow approximation with deep convolutional neural networks. 2020, arXiv preprint arXiv:2004.08826.
- [6] Sekar V, Jiang Q, Shu C, Khoo BC. Fast flow field prediction over airfoils using deep learning approach. *Phys Fluids* 2019;31(5).
- [7] Bhatnagar S, Afshar Y, Pan S, Duraisamy K, Kaushik S. Prediction of aerodynamic flow fields using convolutional neural networks. *Comput Mech* 2019;64:525–45.
- [8] Chen F, Akasaka K. 3D flow field estimation around a vehicle using convolutional neural networks. In: *Proceedings of the 32nd british machine vision conference*, no. 936. 2021.
- [9] Lee S, You Z. Data-driven prediction of unsteady flow over a circular cylinder using deep learning. *J Fluid Mech* 2019;879:217–54.
- [10] Pfaff T, Fortunato M, Sanchez-Gonzalez A, Battaglia PW. Learning mesh-based simulation with graph networks. 2020, arXiv preprint arXiv:2010.03409.
- [11] Geneva N, Zabarar N. Multi-fidelity generative deep learning turbulent flows. 2020, arXiv preprint arXiv:2006.04731.
- [12] Wang R, Kashinath K, Mustafa M, Albert A, Yu R. Towards physics-informed deep learning for turbulent flow prediction. In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, p. 1457–66.
- [13] Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A. Fourier neural operator for parametric partial differential equations. 2020, arXiv preprint arXiv:2010.08895.
- [14] Wen G, Li Z, Azizzadenesheli K, Anandkumar A, Benson SM. U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Adv Water Resour* 2022;163:104180.
- [15] Li Z, Huang DZ, Liu B, Anandkumar A. Fourier neural operator with learned deformations for pdes on general geometries. 2022, arXiv preprint arXiv:2207.05209.

- [16] Tran A, Mathews A, Xie L, Ong CS. Factorized fourier neural operators. 2021, arXiv preprint arXiv:2111.13802.
- [17] Yousif MZ, Zhang M, Yu L, Vinuesa R, Lim H. A transformer-based synthetic-inflow generator for spatially developing turbulent boundary layers. *J Fluid Mech* 2023;957:A6.
- [18] An J, Wang H, Liu B, Luo KH, Qin F, He GQ. A deep learning framework for hydrogen-fueled turbulent combustion simulation. *Int J Hydrogen Energy* 2020;45(35):17992–8000.
- [19] Ren J, Wang H, Chen G, Luo K, Fan J. Predictive models for flame evolution using machine learning: A priori assessment in turbulent flames without and with mean shear. *Phys Fluids* 2021;33(5).
- [20] Wang X, Chang Y-H, Li Y, Yang V, Su Y-H. Surrogate-based modeling for emulation of supercritical injector flow and combustion. *Proc Combust Inst* 2021;38(4):6393–401.
- [21] Barnes FW, Segal C. Cavity-based flameholding for chemically-reacting supersonic flows. *Prog Aerosp Sci* 2015;76:24–41.
- [22] Cabra R, Myhrvold T, Chen J, Dibble R, Karpetis A, Barlow R. Simultaneous laser Raman-Rayleigh-LIF measurements and numerical modeling results of a lifted turbulent H<sub>2</sub>/N<sub>2</sub> jet flame in a vitiated coflow. *Proc Combust Inst* 2002;29(2):1881–8.
- [23] Wang Z, Wang H, Sun M. Review of cavity-stabilized combustion for scramjet applications. *Proc Inst Mech Eng G* 2014;228(14):2718–35.
- [24] Kim KM, Baek SW, Han CY. Numerical study on supersonic combustion with cavity-based fuel injection. *Int J Heat Mass Transfer* 2004;47(2):271–86.
- [25] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput Phys* 1998;12(6):620–31.
- [26] GRI-Mech 3.0. URL <http://combustion.berkeley.edu/gri-mech/version30/text30.html>.
- [27] Domingo P, Vervisch L, Veynante D. Large-eddy simulation of a lifted methane jet flame in a vitiated coflow. *Combust Flame* 2008;152(3):415–32.
- [28] Marinov N, Westbrook C, Pitz W. Detailed and global chemical kinetics model for hydrogen. *Transp Phenom Combust* 1996;1(118–129):80.
- [29] Ionescu C, Papava D, Olaru V, Sminchisescu C. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans Pattern Anal Mach Intell* 2013;36(7):1325–39.
- [30] Zhang J, Zheng Y, Qi D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In: *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31. 2017.
- [31] Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: *Advances in neural information processing systems*, Vol. 28. 2015.
- [32] Rasp S, Dueben PD, Scher S, Weyn JA, Moutadid S, Thuerey N. WeatherBench: a benchmark data set for data-driven weather forecasting. *J Adv Modelling Earth Syst* 2020;12(11):e2020MS002203.
- [33] De Bézenac E, Pajot A, Gallinari P. Deep learning for physical processes: Incorporating prior scientific knowledge. *J Stat Mech Theory Exp* 2019;2019(12):124009.
- [34] Gao Z, Tan C, Wu L, Li SZ. Simvnp: Simpler yet better video prediction. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, p. 3170–80.
- [35] Tan C, Gao Z, Li S, Li SZ. Simvnp: Towards simple yet powerful spatiotemporal predictive learning. 2022, arXiv preprint arXiv:2211.12509.
- [36] Srivastava N, Mansimov E, Salakhudinov R. Unsupervised learning of video representations using lstms. In: *International conference on machine learning*. PMLR; 2015, p. 843–52.
- [37] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18. Springer; 2015, p. 234–41.
- [38] Ding X, Zhang X, Han J, Ding G. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, p. 11963–75.
- [39] Zhang T, Yi Y, Xu Y, Chen ZX, Zhang Y, Weinan E, Xu Z-Q. A multi-scale sampling method for accurate and robust deep neural network to predict combustion chemical kinetics. *Combust Flame* 2022;245:112319.
- [40] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.
- [41] Smith LN, Topin N. Super-convergence: Very fast training of neural networks using large learning rates. In: *Artificial intelligence and machine learning for multi-domain operations applications*, Vol. 11006. SPIE; 2019, p. 369–86.
- [42] Brandstetter J, Worrall D, Welling M. Message passing neural PDE solvers. 2022, arXiv preprint arXiv:2202.03376.
- [43] Long Z, Lu Y, Ma X, Dong B. Pde-net: Learning pdes from data. In: *International conference on machine learning*. PMLR; 2018, p. 3208–16.
- [44] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A, Bottou L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* 2010;11(12).
- [45] Godwin J, Schaarschmidt M, Gaunt A, Sanchez-Gonzalez A, Rubanova Y, Veličković P, Kirkpatrick J, Battaglia P. Simple gnn regularisation for 3d molecular property prediction & beyond. 2021, arXiv preprint arXiv:2106.07971.
- [46] Sanchez-Gonzalez A, Godwin J, Pfaff T, Ying R, Leskovec J, Battaglia P. Learning to simulate complex physics with graph networks. In: *International conference on machine learning*. PMLR; 2020, p. 8459–68.
- [47] Li Z, Meidani K, Farimani AB. Transformer for partial differential equations' operator learning. 2022, arXiv preprint arXiv:2205.13671.
- [48] Zhu Y, Zabarar N. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *J Comput Phys* 2018;366:415–47.
- [49] Luo K, Wang H, Yi F, Fan J. Direct numerical simulation study of an experimental lifted H<sub>2</sub>/N<sub>2</sub> flame. Part 1: validation and flame structure. *Energy Fuels* 2012;26(10):6118–27.