

RESEARCH

Open Access



# Deep learning method for super-resolution reconstruction of the spatio-temporal flow field

Kairui Bao<sup>1,2</sup>, Xiaoya Zhang<sup>2</sup>, Wei Peng<sup>2</sup> and Wen Yao<sup>2\*</sup>

\*Correspondence:  
wendy0782@126.com

<sup>1</sup> College of Aerospace  
Science and Engineering,  
National University of Defense  
Technology, Changsha 410073,  
China

<sup>2</sup> Defense Innovation Institute,  
Chinese Academy of Military  
Science, Beijing 100071, China

## Abstract

The high-resolution (HR) spatio-temporal flow field plays a decisive role in describing the details of the flow field. In the acquisition of the HR flow field, traditional direct numerical simulation (DNS) and other methods face a seriously high computational burden. To address this deficiency, we propose a novel multi-scale temporal path UNet (MST-UNet) model to reconstruct temporal and spatial HR flow fields from low-resolution (LR) flow field data. Different from the previous super-resolution (SR) model, which only takes advantage of LR flow field data at instantaneous (SLR) or in a time-series (MTLR), MST-UNet introduces multi-scale information in both time and space. MST-UNet takes the LR data at the current frame and the predicted HR result at the previous moment as the model input to complete the spatial SR reconstruction. On this basis, a temporal model is introduced as the inbetweening model to obtain HR flow field data in space and time to complete spatio-temporal SR reconstruction. Finally, the proposed model is validated by the spatio-temporal SR task of the flow field around two-dimensional cylinders. Experimental results show that the outcome of the MST-UNet model in spatial SR tasks is much better than those of SLR and MTLR, which can greatly improve prediction accuracy. In addition, for the spatio-temporal SR task, the spatio-temporal HR flow field predicted by the MST-UNet model has higher accuracy either.

**Keywords:** Super-resolution, Flow field, UNet, Deep learning

## 1 Introduction

The demand for HR data in the flow field has always been a major pursuit in computational fluid dynamics (CFD). Many CFD methods, such as the finite volume (FV) [1], the finite difference (FD) [2] and the lattice Boltzmann (LBM) [3, 4], can be performed for obtaining HR turbulence data. Unlike the CFD methods that directly solve the Navier-Stokes (NS) equations on a macroscopic scale, the LBM simulates the flow field based on the lattice Boltzmann equation on a mesoscopic scale. Compared to FD and FV solvers, LBM is simpler in modeling and implementation, has parallel processing capabilities, and is more successful in processing complex boundaries [5, 6]. However, for HR turbulent flow data, LBM usually requires a large number of lattice numbers with a very small time step, which makes the acquisition of HR data time-consuming and computationally expensive [7]. Due to the high data cost of LBM results for the flow field, the rapid development of data-driven models provides an effective way to obtain HR flow

data in real-time [8–10]. In this study, we focus on obtaining HR turbulent flows from LR spatio-temporal data in real-time using deep learning (DL) models.

With the mushroom growth of deep learning technology, deep neural networks have been widely used in flow field SR and reconstruction tasks [11–13]. Various deep learning models have been applied in turbulent flow field super-resolution [14–17]. Fukami et al. [18] proposed a hybrid downsampled skip-connection/multi-scale (DSC/MS) model to reconstruct two-dimensional homogeneous turbulence. Considering the multi-scale information of the flow field, Liu et al. [19] designed a multi-resolution convolutional autoencoder (MrCAE) SR architecture that leverages the multigrid method and transfer learning. MrCAE can dynamically capture different scaled flow information at different network depths. Given the difficulty of obtaining HR label information, Gao et al. [20] proposed the CNN-SR model, which is trained without any high-resolution labels in a physics-driven way. Han et al. [21] proposed a novel deep learning solution that recovers temporal super-resolution (TSR) of three-dimensional vector field data (VFD) for unsteady flow. In addition to the CNN model, the generative adversarial network (GAN) [22] model is also widely used in flow field super-resolution. Güemes et al. [23] used the super-resolution generative adversarial network (SRGAN) to reconstruct the turbulent flow measured by the coarse wall. The results show that SRGAN can obtain perfect reconstruction even at very low resolution. In addition, Deng et al. [24] compared the results of two models, SRGAN and enhanced-SRGAN (ESRGAN), on super-resolution reconstructions of complicated wake flows behind cylinders. Their results found that ESRGAN has better reconstruction ability than SRGAN in both mean and fluctuation flow fields. To reconstruct HR turbulent flow by minimal flow field data, Yousif et al. [25] proposed the multi-scale ESRGAN with a physics-based loss function. Considering the great performance of reduced-order models in flow field prediction, the combination of reduced-order models with deep learning can also be well applied to flow field super-resolution tasks. Guastoni et al. [26] combined a fully convolutional neural network (FCN) with proper orthogonal decomposition (POD) [27]. They proposed the FCN-POD model, which has great performance in reconstructing the flow field from coarse wall measurements. The flow field SR technology based on deep learning can accurately obtain the HR flow field in real-time, which is of great significance for obtaining high-precision flow field information.

Most of the models mentioned above only use the spatial information of the low-resolution flow field in the SR reconstruction of the flow field, which will increase the prediction error of the model. In order to improve the quality of the SR reconstruction, Liu et al. [28] proposed multiple temporal paths convolutional neural network (MTPC). The LR information of the current moment and two adjacent moments is adopted as the input of the MTPC network, and then the output of the three moments is weighted to predict the HR image of the current moment. The results indicate that the MTPC has a better reconstruction capability than only using LR information at the current moment. In addition, Fukami et al. [29] further proposed a spatio-temporal SR model based on the temporal SR task. They constructed a temporal model and a spatial model, respectively, to reconstruct the HR information over a period of time, only using the LR information at the first and last moments. Their work has made important contributions to reconstructing the spatio-temporal flow field.

In this paper, to further improve the prediction accuracy of the HR flow field, we propose the MST-UNet model for flow field SR tasks with spatio-temporal combination. Since the flow field is usually spatially and temporally coupled, the MST-UNet model not only uses the LR spatial information of the flow field at the current frame but also incorporates the HR information predicted at the previous frame. The information input of this spatio-temporal combination can further improve prediction accuracy. In addition, we apply the MST-UNet model to the spatio-temporal SR task, which can reconstruct the HR information of a series of intermediate frames only from the beginning and end frames of LR information.

The remainder of this paper is structured as follows. Section 2 introduces the MST-UNet model and spatio-temporal SR task. The construction of datasets and the results of the MST-UNet model in spatial SR and spatio-temporal SR tasks are presented in Section 3. Section 4 concludes the paper.

## 2 Methods

The purpose of this study is to reconstruct the spatio-temporal HR flow field from LR data in real time using CNN. In order to predict the HR flow field at the current frame, we use the LR information at the current frame and the HR information at the previous frame. In this way, we can obtain HR flow field information by learning the end-to-end mapping function  $F$ ,

$$P_t^{SR} = F(P_{t-1}^{HR}, P_t^{LR}, \theta), \quad (1)$$

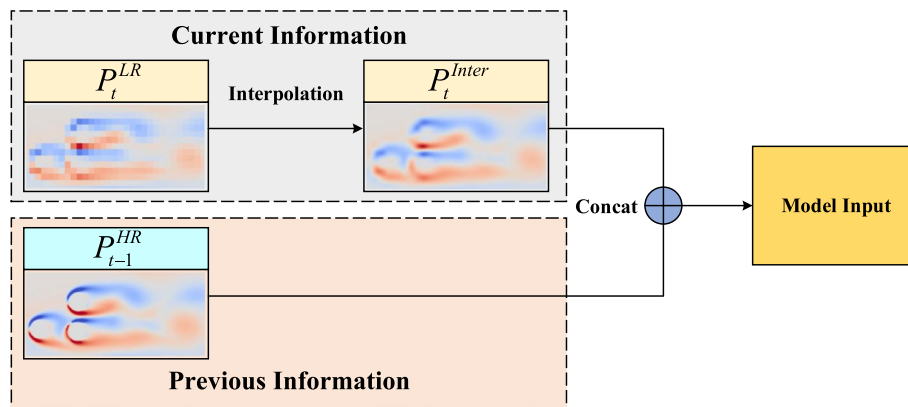
where  $P_t^{SR}$  is the result of SR reconstruction, and  $\theta$  means the learnable network parameters in CNN,  $P_t^{LR}$  and  $P_{t-1}^{HR}$  represent LR information at the current frame and HR information at the previous frame, respectively. It should be noted that  $P_{t-1}^{HR}$  is not the label information but the model output result at the previous frame instead. We can take advantage of the LR information at the current moment only, which is a normal single-input-single-output situation. Compared with this case, using the previous time information can improve the data utilization efficiency and prediction accuracy. In this multiple-input-single-output case, the shared explicit redundancy from adjacent frames will provide more constraints.

In the current work, the MST-UNet model is first introduced for the SR tasks. Then, we apply the MST-UNet model in the spatio-temporal SR task.

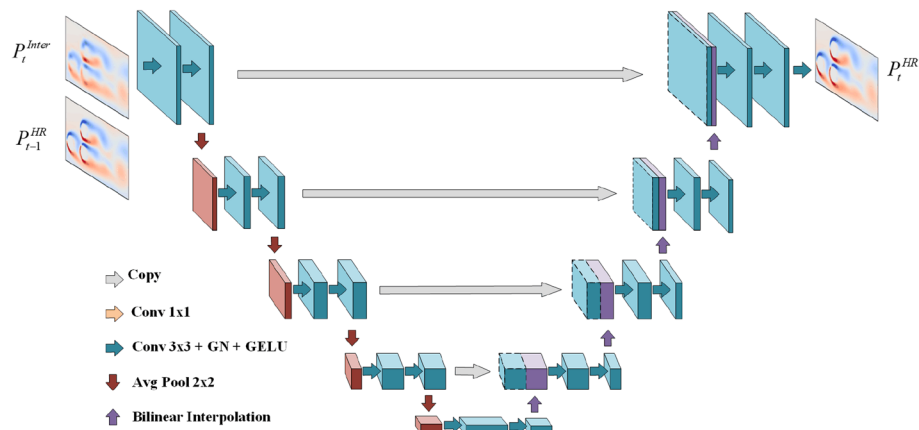
### 2.1 MST-UNet model

In the MST-UNet model, the HR images of the previous frame ( $P_{t-1}^{HR}$ ) and the LR images of the current frame ( $P_t^{LR}$ ) are combined as the model input, which is multi-scale information. To fuse this multi-scale information as model input, the  $P_t^{LR}$  is first up-sampled to the target resolution  $P_t^{inter}$  using the bicubic interpolation. Then, the  $P_{t-1}^{HR}$  and  $P_t^{inter}$  are concatenated into a two-channel image as the input of the model (Fig. 1).

In this paper, the UNet model [30] is adopted as the basic model of the SR task, which is one outstanding image-to-image CNN model. The UNet model is an encoder-decoder architecture consisting of a contracting path that captures global information (encoder) and an asymmetric expanding path to reconstruct HR information (decoder).



**Fig. 1** Data fusion for the input of the model. The HR images of the previous frame ( $P_{t-1}^{HR}$ ) and the LR images of the current frame ( $P_t^{LR}$ ) are combined as the input of the model, and the interpolation method is used to make two inputs the same size



**Fig. 2** The UNet architecture for SR tasks. The input of the model is a two-channel image consisting of  $P_{t-1}^{HR}$  and  $P_t^{inter}$ , and the output of the model is the HR image at the current frame

In addition to combining LR information during downsampling and HR information during upsampling, the UNet model fills in the underlying information through a feature fusion operation of skip connections to improve prediction accuracy. Due to this particular framework, UNet has gained great success on computer vision tasks.

The super-resolution task we deal with is a typical image-to-image regression problem, and the classic UNet structure is modified for better application to this assignment, as shown in Fig. 2. The input of the model is a two-channel image consisting of  $P_{t-1}^{HR}$  and  $P_t^{inter}$ , and the output of the model is the HR image at the current frame. The blue arrow is a basic block made up of three parts, followed by a  $3 \times 3$  convolutional layer, Group-Norm (GN) [31] and ReLU activation functions ( $\phi(x) = \max(x, 0)$ ). Since the batch normalization method is relatively dependent on the batch size, when the batch size is too small, the calculated mean and variance are insufficient to represent the entire data distribution. The GN method is employed in this problem, independent of the batch size. The UNet model consists of the contracting path (on the left) and the expansive path (on the right). As for the contracting path, four downsampling blocks (red arrow) using

average pooling with  $2 \times 2$  filters and stride 2 are adopted for feature extraction. Correspondingly, the expansive path consists of four identical  $2 \times$  upsampling blocks, using bilinear interpolation instead of the transpose convolution. For our modified UNet model, the contracting path and the expansive path are completely symmetric, which makes the input and output of the model the same size.

In the training process, in order to predict the HR image at the  $t$  frame, the LR information of the  $t$  ( $P_t^{LR}$ ) and  $t - 1$  ( $P_{t-1}^{LR}$ ) frames is used. As shown in Fig. 3, in order to obtain the HR results of the previous frame, we use interpolation to obtain an initial value in the first epoch of the training process. In the remaining epochs, the idea of the memory bank [32] is adopted. The prediction results of the previous epoch will be stored and used as the HR images of the previous frame. The L1 loss is chosen as the loss function,

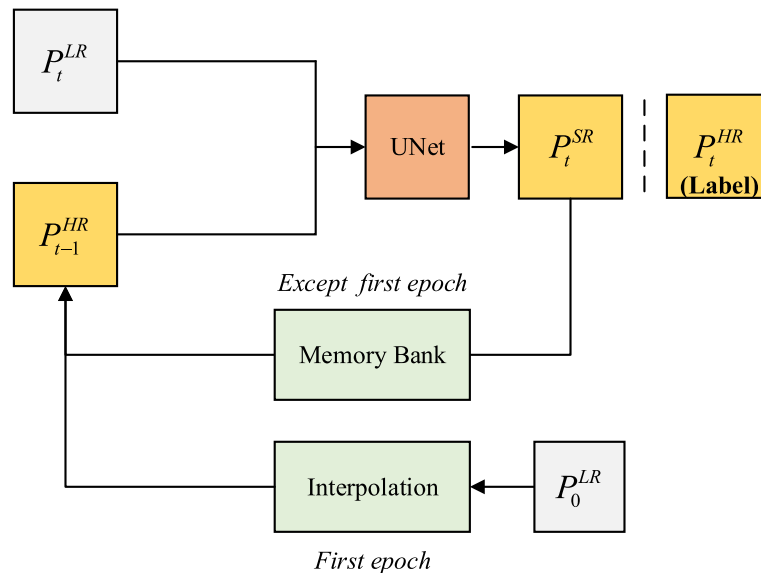
$$Loss_{mst} = \frac{1}{N} \sum_{t=1}^N \|P_t^{SR} - P_t^{HR}\|. \quad (2)$$

$Loss_{mst}$  is the loss function of the MST-UNet. According to (2), the network parameters can be updated using the back-propagation algorithm.

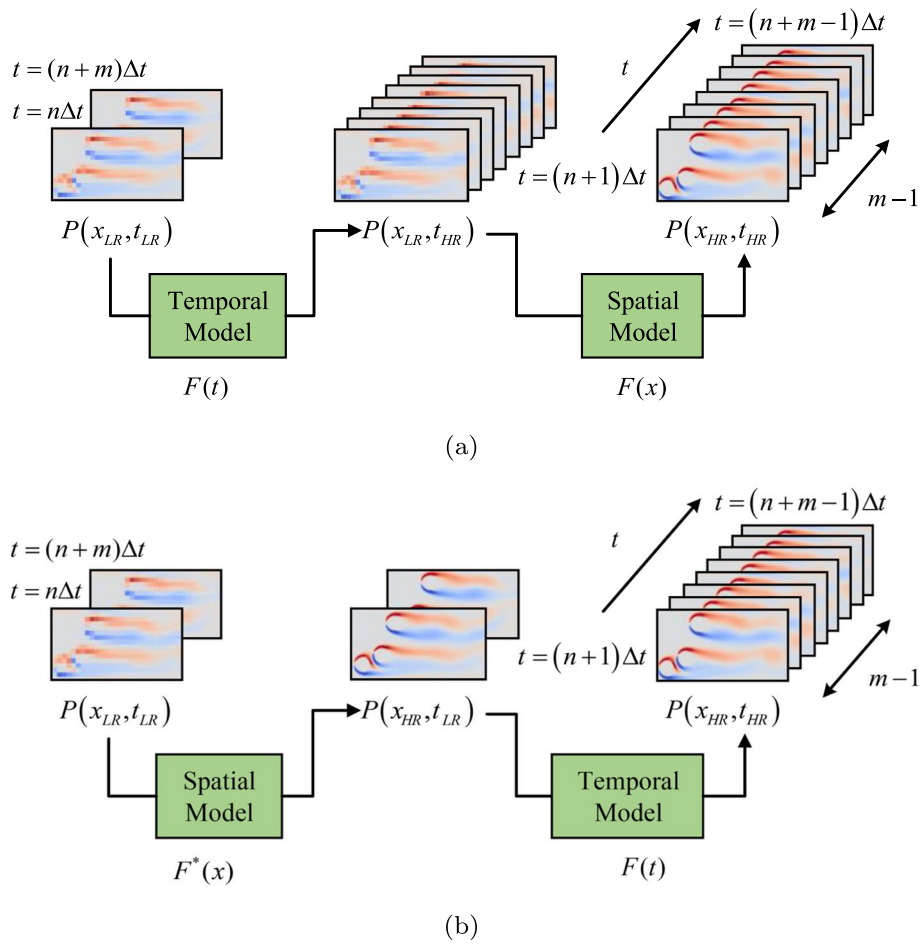
## 2.2 Spatio-temporal super resolution

For reconstructing the spatio-temporal flow field, Fukami et al. [29] proposed two methods. Both methods include temporal and spatial models, but in different orders.

As shown in Fig. 4(a), for LR spatial and temporal information ( $P(x_{LR}, t_{LR})$ ), we apply the inbetweening temporal model ( $F_t^* : \mathbb{R}^{n_{LR} \times m_{LR}} \rightarrow \mathbb{R}^{n_{LR} \times m_{HR}}$ ) to obtain ( $P(x_{LR}, t_{HR})$ ), which has HR temporal information. Then the spatial model ( $F_x$ ) is employed to gain HR spatial and temporal information ( $P(x_{LR}, t_{HR})$ ).



**Fig. 3** Schematic diagram of the multi-scale temporal path UNet model



**Fig. 4** Spatio-temporal SR reconstruction of the flow field using deep learning. (a) shows the combine model using the temporal model as the inbetweening model while (b) demonstrates the combine model using the spatial model

$$\begin{aligned}
 P(x_{LR}, t_{HR}) &= F_t^*(P(x_{LR}, t_{LR})), \\
 P(x_{HR}, t_{HR}) &= F_x(P(x_{LR}, t_{HR})), \\
 P(x_{HR}, t_{HR}) &= F_x(F_t^*(q(x_{LR}, t_{LR}))).
 \end{aligned} \tag{3}$$

Correspondingly, we can use the spatial model as the inbetweening model to obtain HR spatial information (\$P(x\_{HR}, t\_{LR})\$), and then use the temporal model to obtain HR spatial and temporal information (Fig. 4(b)).

$$\begin{aligned}
 P(x_{HR}, t_{LR}) &= F_x^*(P(x_{LR}, t_{LR})), \\
 P(x_{HR}, t_{HR}) &= F_t(P(x_{HR}, t_{LR})), \\
 P(x_{HR}, t_{HR}) &= F_t(F_x^*(q(x_{LR}, t_{LR}))),
 \end{aligned} \tag{4}$$

where  $n$  is the dimension of spatial information and  $m$  is the dimension of temporal information.  $P(x_{LR}, t_{LR})$  represents LR spatial and temporal information, which is the model input.  $P(x_{LR}, t_{HR})$  represents HR temporal information while LR spatial information. Similarly,  $P(x_{HR}, t_{LR})$  represents HR spatial while LR temporal information, and

$P(x_{HR}, t_{HR})$  represents HR temporal and spatial information, which is the output of the model.

In this study, the first model is selected as the spatio-temporal SR model, which uses the temporal model as the inbetweening model. In the temporal model, we still use the UNet model, taking the first and last two frames of the LR image as input, and the output is a series of LR images in the middle frame. In this way, the MST-UNet model proposed in Section 2.1 can be used as a spatial model to improve the model prediction accuracy further. The loss function of the spatio-temporal SR model is as follows,

$$\begin{aligned} Loss_{ts} &= \|\hat{P}(x_{HR}, t_{HR}) - F_x(F_t^*(P(x_{LR}, t_{LR})))\|, \\ &= \|\hat{P}(x_{HR}, t_{HR}) - P(x_{HR}, t_{HR})\|. \end{aligned} \quad (5)$$

Among them,  $\hat{P}(x_{HR}, t_{HR})$  represents the label information of a series of frames.

### 3 Numerical results

To evaluate the ability of the MST-UNet model to reconstruct HR flow fields, the flow around cylinders is used as our test [33, 34]. The entire flow field domain is shown in Fig. 5.

In Fig. 5,  $D = 0.2$  m and  $u_0 = 0.2$  m/s represent the cylinder diameter and initial velocity, respectively. The LBM is applied in the construction of the data set.

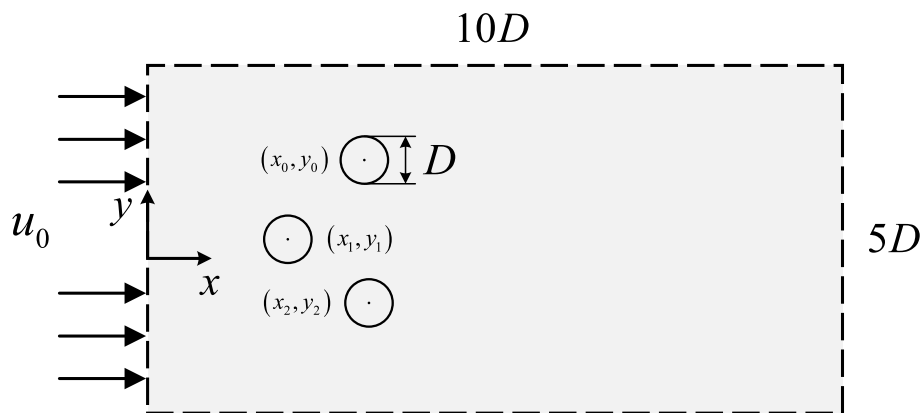
#### 3.1 The lattice Boltzmann method

The LBM is performed on a square lattice where at each node there is a discrete number of directions in which the fluid particles can move. In this study, a 2D model with nine lattice velocities - the D2Q9, will be discussed and applied.

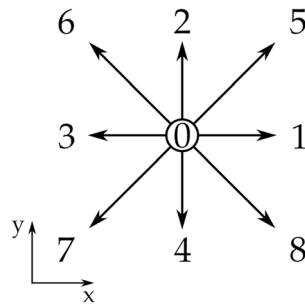
A general form of the LBM equation can be written as follows [35],

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = \Omega_i^{\text{coll}}(\mathbf{x}, t), \quad (6)$$

where  $f_i$  is a discrete distribution function at position  $\mathbf{x}$ , with average particle lattice velocity  $\mathbf{e}_i$ , pointing in direction  $i$  (see Fig. 6), at time  $t$ .  $\Omega^{\text{coll}}$  is the collision operator, and will be discussed later.  $\mathbf{e}_i = (e_{x,i}, e_{y,i})$  is the directional vector and it takes the following values:



**Fig. 5** The illustration of the flow domain around three cylinders



**Fig. 6** D2Q9 basic lattice velocity directions

$$\mathbf{e} = \begin{pmatrix} e_{x,0} & e_{x,1} & \dots & e_{x,8} \\ e_{y,0} & e_{y,1} & \dots & e_{y,8} \end{pmatrix} = \begin{pmatrix} 0, 1, 0, -1, 0, 1, -1, -1, 1 \\ 0, 0, 1, 0, -1, 1, 1, -1, -1 \end{pmatrix}. \quad (7)$$

The local fluid density  $\rho$  is computed as the sum of all  $f_i$  on site. The expression for the D2Q9 lattice model is the following,

$$\rho(\mathbf{x}, t) = \sum_{i=0}^8 f_i(\mathbf{x}, t). \quad (8)$$

The local fluid velocity  $\mathbf{u} = (u_x, u_y)$  can be computed from the expression for the local momentum.

$$\rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^8 f_i(\mathbf{x}, t)\mathbf{e}_i. \quad (9)$$

The Bhatnagar-Gross-Krook (BGK) model has the simplest form of the presented collision operators, which reads as follows [35, 36],

$$\Omega_i^{\text{coll}} = -\omega(f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)). \quad (10)$$

$f_i^{\text{eq}}$  is the local equilibrium distribution function, and is computed from the local macroscopic velocity  $\mathbf{u}$ , and the local density  $\rho$ .

$$f_i^{\text{eq}}(\mathbf{u}, \rho) = w_i \rho(\mathbf{x}, t) \left[ 1 + 3\mathbf{e}_i \cdot \mathbf{u}(\mathbf{x}, t) + \frac{9}{2}(\mathbf{e}_i \cdot \mathbf{u}(\mathbf{x}, t))^2 - \frac{3}{2}\mathbf{u}(\mathbf{x}, t) \cdot \mathbf{u}(\mathbf{x}, t) \right], \quad (11)$$

where  $w_i$  is the equilibrium weight factor in the  $i$ -th direction. Its values add up to 1, and in D2Q9 they equal:

$$\mathbf{w} = (w_0, w_1, \dots, w_8) = \left( \frac{4}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36} \right). \quad (12)$$

$\omega$  is the relaxation rate and is the inverse of the relaxation time  $\tau = \frac{1}{\omega}$ . From the relaxation rate, the fluid's kinematic viscosity  $\nu$  can be calculated,

$$\nu = \frac{1}{3} \left( \frac{1}{\omega} - \frac{1}{2} \right). \quad (13)$$



There are three cylinders in the domain, with center coordinates  $(x_0, y_0)$ ,  $(x_1, y_1)$ , and  $(x_2, y_2)$ . In our problem, the range of six coordinates is shown in Table 1. Two datasets of different sizes are generated to illustrate the effectiveness of our model. One is a small dataset with the size of 5000, denoted as *Dataset<sub>s</sub>*: 500 groups of different cylindrical positions are generated using the Latin hypercube sampling method. For each set of positions, we generate 10 frames of data at consecutive moments. The other one is a large dataset completed with the size of 19000, which has 1000 groups of different cylindrical positions and 19 frames of data at each position, denoted as *Dataset<sub>l</sub>*. In each dataset, 80% of the samples are used for training, and the remaining 20% for testing. The position of the cylinder changes, and the flow field at each position varies with time, which increases the difficulty of the SR task. In our problem, the resolution of HR and LR flow fields is  $128 \times 256$  and  $16 \times 32$  separately with an upscaling factor of 8.

In the experiments, we first investigate the spatial SR task using the MST-UNet model. Then, on the basis of the spatial super-resolution task, the spatial-temporal SR task is further studied. The mean absolute error (MAE) and  $L_2$  error norms ( $\varepsilon$ ) are adopted as the metrics to evaluate the performance of the MST-UNet model and spatial-temporal SR task, where  $\varepsilon$  is defined as

$$\varepsilon = \left\| P^{HR} - F(P^{LR}, \theta) \right\|_2 / \left\| P^{HR} \right\|_2. \quad (14)$$

### 3.2 Spatial super-resolution task

In this task, our MST-UNet model combines the HR information of the previous moment with the LR information of the current moment. By contrast, models with LR information as inputs only are introduced,

$$P^{SR} = F(P_{[t-d:t]}^{LR}, \theta), \quad (15)$$

where  $P_{[t-d:t]}^{LR}$  represents a sequence of LR data and  $\theta$  represents model parameters. In the case of  $d = 0$ , the mapping between input and output becomes  $P^{SR} = F(P_t^{LR}, \theta)$ , which is the case of general single-input-single-output, denoted as SLR-UNet. Under the circumstances, the model input contains only instantaneous spatial information, not temporal information. When  $d > 0$ , the model input combines LR information from multiple moments, as shown in Fig. 7. Besides the case of  $d = 0$ , the two cases  $d = 1$  and  $d = 2$  are also known as our comparative experiments, denoted as MTLR1-UNet and MTLR2-UNet, respectively. The loss function for these models is as follows,

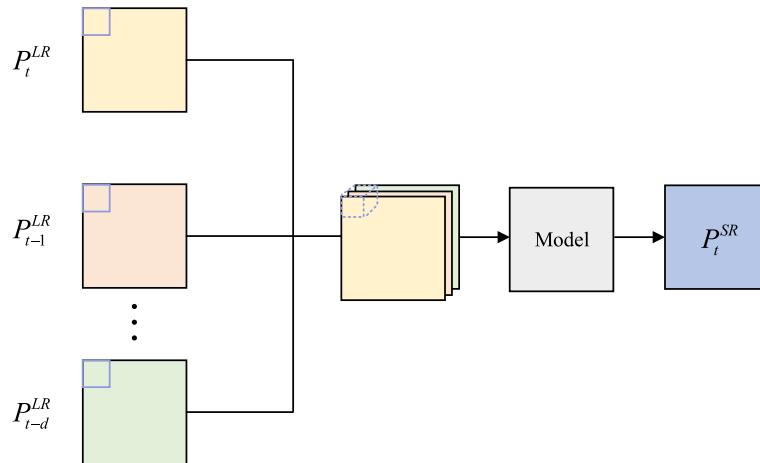
**Table 1** The value range of six coordinates

Parameter	Lower Bound/m	Upper Bound/m
$x_0$	0.6	0.8
$y_0$	0.6	0.8
$x_1$	0.2	0.4
$y_1$	0.2	0.8
$x_2$	0.6	0.8
$y_2$	0.2	0.4

$$Loss_{MTLR} = \left\| F\left(P_{[t-d:t]}^{LR}, \theta\right), P_t^{HR} \right\|_p. \quad (16)$$

It is worth noting that in order to ensure the fairness of the comparison experiments, the different models used for comparison only have different input information while with the same network structure.

We perform SR on the curl of the flow field. These models are evaluated on the test set, and the results of MAE and  $L_2$  error norms on two different dataset sizes are shown in Table 2. Where the MST-UNet model is the model proposed in this paper, SLR-UNet, MTLR1-UNet and MTLR2-UNet correspond to the model in (15) where  $d$  is 0, 1 and 2 respectively. It can be seen from the results that the error decreases significantly as the dataset size increases. In the comparison of different models, on both datasets the spatial model containing time information (MTLR1-UNet and MTLR2-UNet) is better than the spatial model using the LR information at the current frame (SLR-UNet) only. For models with different frames of LR information as input, i.e. MTLR1-UNet and MTLR2-UNet, we find that MTLR2-UNet does not perform well, compared to the MTLR1-UNet model with less input LR information. The main reason is that the LR information of multiple frames as input is redundant or invalid, which will affect the prediction accuracy to a certain extent. In the performance of MST-UNet model, it is obvious from Table 2 that the error of the MST-UNet model proposed in this paper is significantly lower than the rest of the models in each indicator. In addition, this advantage is more obvious as the size of the dataset increases. For the results of  $Dataset_l$ , it can be seen that the model prediction errors of the three models SLR-UNet, MTLR1-UNet and MTLR2-UNet, have little difference. From the results, we can find that in the case of sufficient data, the increase of LR temporal information has little effect on the prediction accuracy of the model. However, the MST-UNet model that uses the HR information of the previous frame can greatly improve the prediction accuracy compared to the model which only uses the LR temporal information, since the MST-UNet model uses the HR information of the previous frame in addition to the LR information of the current frame. Generally speaking, the HR information of the previous frame can contain most of the information before the current frame, which allows us to use the information of



**Fig. 7** Schematic diagram of the multi-scale time path network (MST-UNet)

**Table 2** Results of the four models on the test set

Dataset	Model	MAE ( $\times 10^{-4} s^{-1}$ )	$L_2$ error norms
$Dataset_s$	MST-UNet	2.44	0.046
	SLR-UNet	4.00	0.065
	MTLR1-UNet	2.99	0.055
	MTLR2-UNet	3.52	0.062
$Dataset_l$	MST-UNet	1.24	0.031
	SLR-UNet	1.94	0.039
	MTLR1-UNet	1.76	0.038
	MTLR2-UNet	1.81	0.038

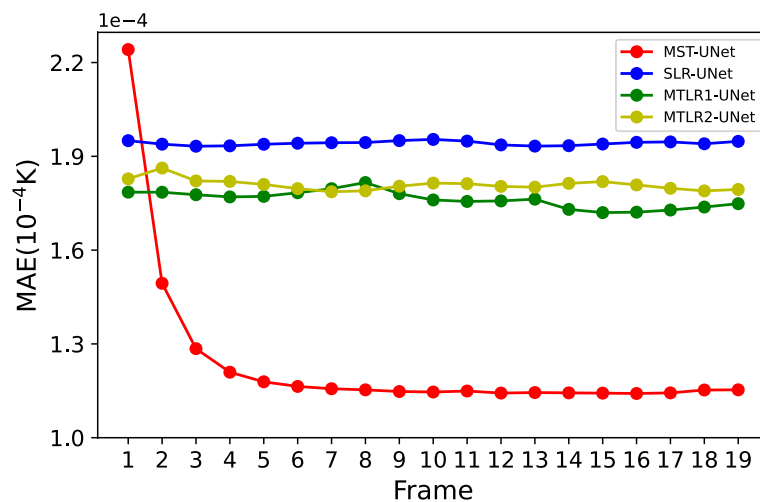
**Table 3** The MAE ( $\times 10^{-4} s^{-1}$ ) of the four models in the test set on  $Dataset_{5000}$

Frame number	1	2	3	4	5	6	7	8	9	10
MST-UNet	3.53	2.49	2.34	2.31	2.31	2.29	2.28	2.27	2.29	2.32
SLR-UNet	4.03	4.41	5.53	7.16	3.82	3.06	2.98	2.97	2.97	3.04
MTLR1-UNet	3.02	2.94	2.97	2.99	3.00	2.99	2.98	2.98	2.98	3.02
MTLR2-UNet	3.61	3.58	3.45	3.46	3.48	3.49	3.49	3.50	3.53	3.61

the previous frame very efficiently compared to the MTLR1-UNet and MTLR2-UNet models.

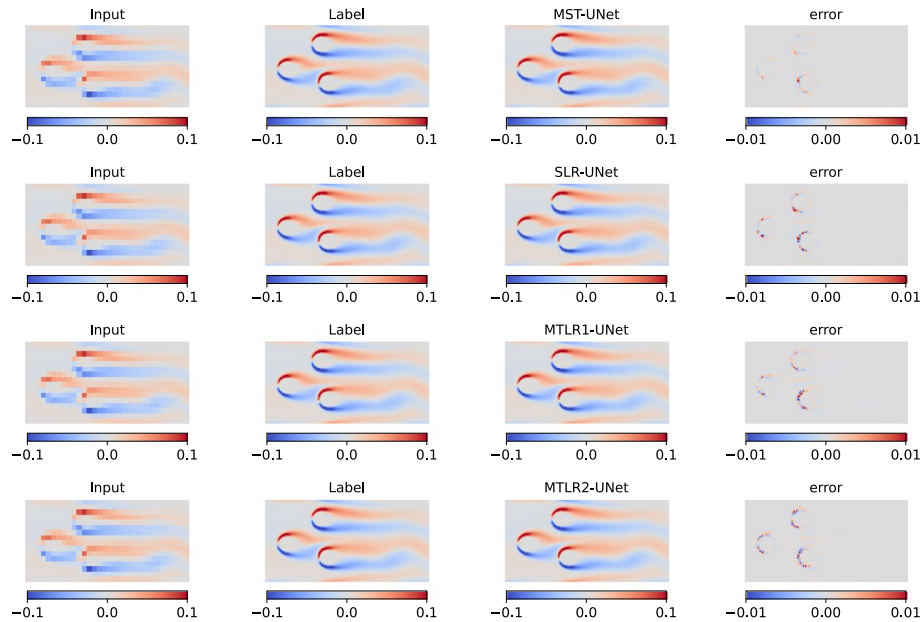
For the spatio-temporal flow field, since we usually pay more attention to the flow field of the last frame, the overall low average error does not reflect the prediction error of the model at a certain moment, and it is very likely that the error accumulates with time. We calculate the average error over 10 frames across all different cylinder positions in the test set, and the MAE of  $Dataset_s$  and  $Dataset_l$  are shown in Table 3 and Fig. 8, respectively.

From the results, it can be seen that the prediction error of the MST-UNet model is much lower than that of the SLR-UNet, MTLR1-UNet and MTLR2-UNet models on both  $Dataset_s$  and  $Dataset_l$  except for the first frame. Furthermore, the prediction

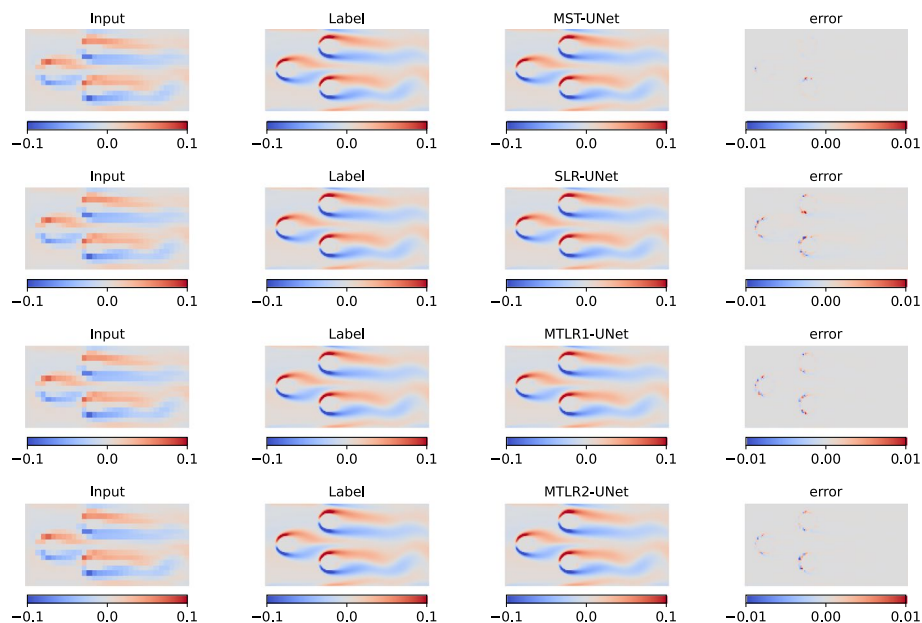


**Fig. 8** The MAE ( $\times 10^{-4} s^{-1}$ ) of the four models in the test set on  $Dataset_{19000}$

errors eventually stabilize over time and do not increase. The reason for the high prediction error of the first frame is that the HR input of the previous frame is obtained by interpolation. In order to improve the prediction accuracy of the first frame, we can also build a separate model for the first frame to improve the prediction accuracy. However, for the space-time flow field, we generally focus on the information at later moments. The errors of the last frame at one position on the test sets are shown in Figs. 9 and 10, respectively. The results show that the performance of MST-UNet is greatly improved compared to the other three models.



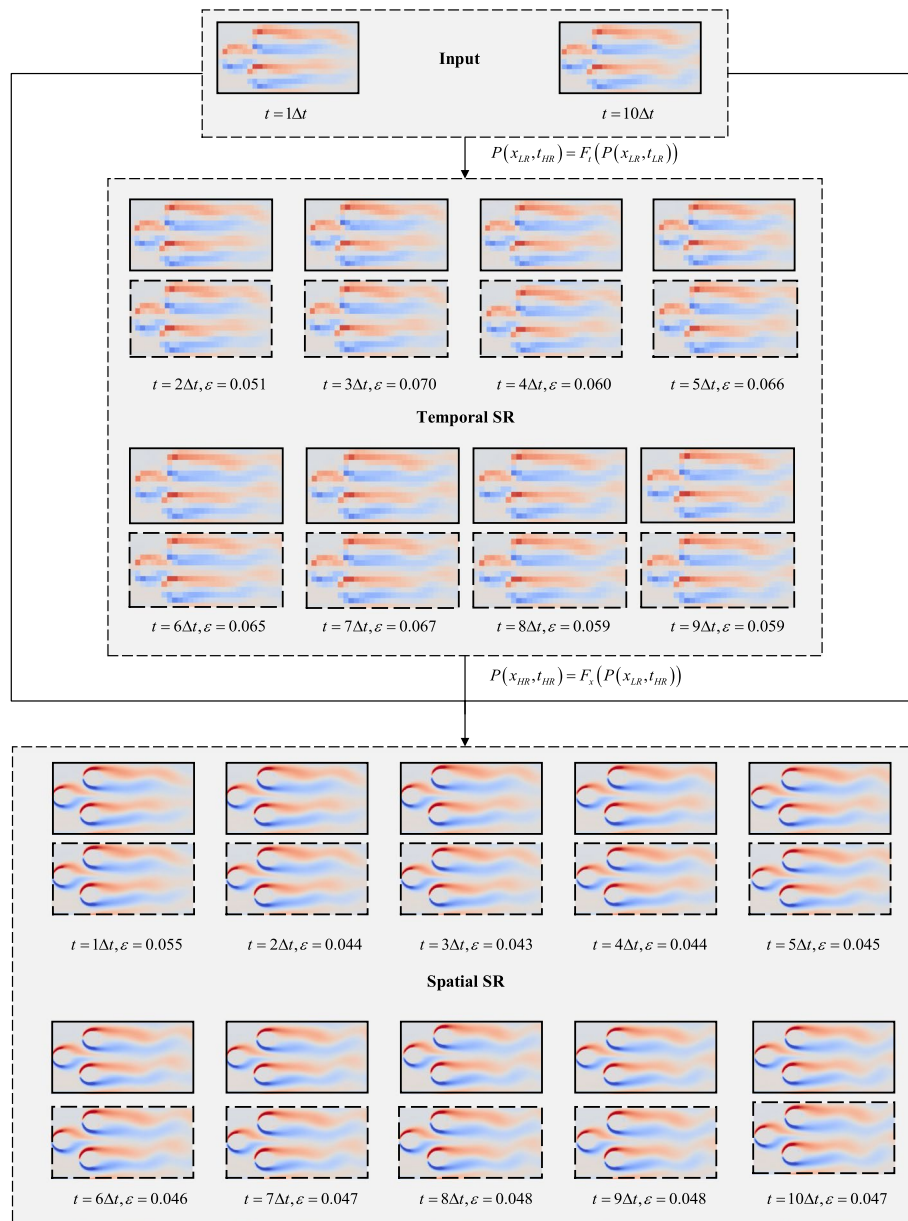
**Fig. 9** Performance of different models in the test set on *Dataset<sub>5</sub>*



**Fig. 10** Performance of different models in the test set on *Dataset<sub>7</sub>*

### 3.3 Spatio-temporal super-resolution task

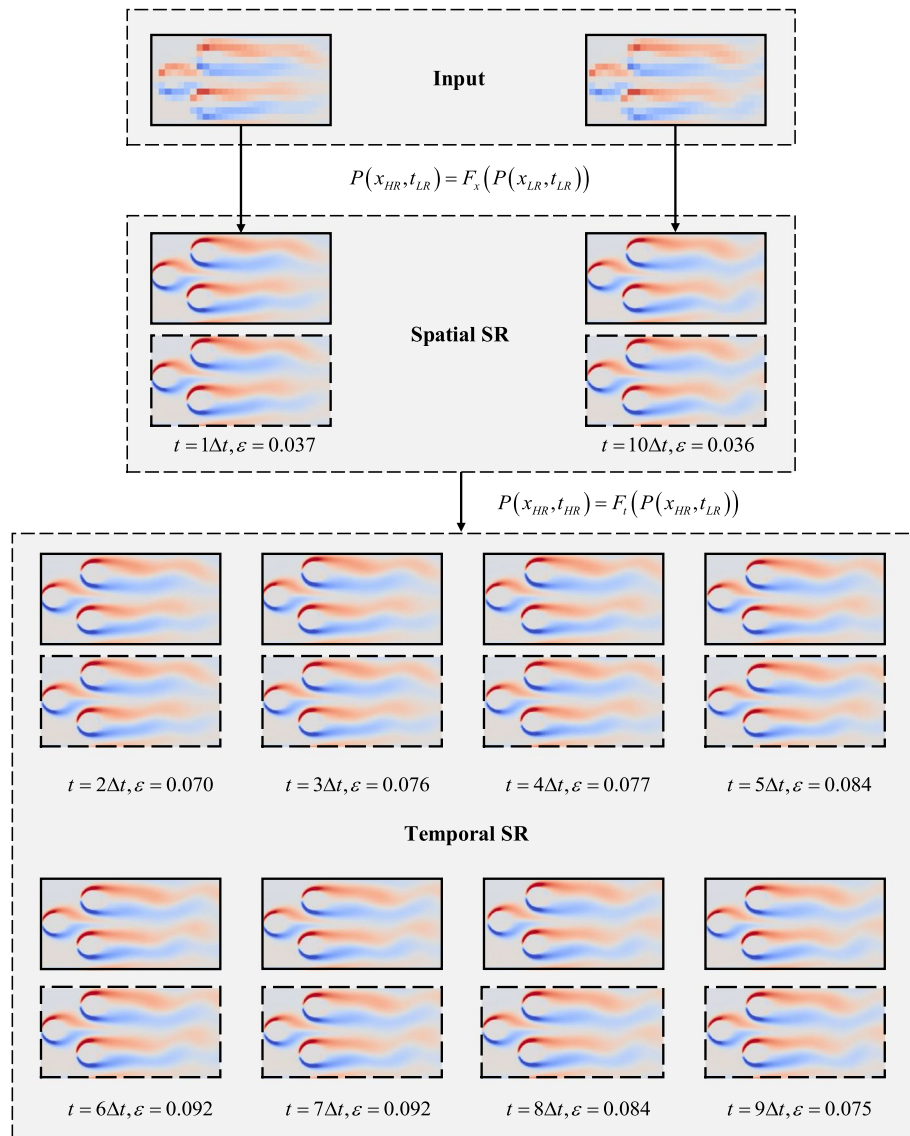
As shown in Section 2.2, the purpose of the spatio-temporal SR task is to reconstruct HR information of a time series from the LR information at the first and last two frames. As shown in Section 2, two models can be used in the spatio-temporal SR task. One is to obtain HR temporal information through the temporal model first, and then gain HR spatial and temporal information through the spatial model, denoted as TS; the other model is in reverse order, taking the spatial model as the inbetweening model, which acquires the HR spatial information first, denoted as ST. In order to facilitate comparison, for different cylinder positions in  $Dataset_l$ , we select the first ten frames. In our spatio-temporal SR task, we predict the HR information of eight frames through the LR information at the first and last two frames (frame 1 and frame 10).



**Fig. 11** Performance of the TS model on the spatio-temporal super-resolution task

Our model adopts the TS model. For the temporal model, we employ the UNet model in Fig. 2, which takes the LR images of the 1st and 10th frames as input and the output is a series of LR images between the two frames. Since the temporal model has output a series of LR images, on the spatial model, we adopt the MST-UNet model proposed in Section 2.1. A set of results for the TS model is shown in Fig. 11. The comparison model adopts the ST model. On the spatial model, since there are only two frames of LR images, we use the single model in Section 3.2. The temporal model in ST adopts the UNet model too, while the input and output are HR images. Figure 12 presents a set of results for the ST model.

We evaluate the results of both models on the test set, and the results of MAE and L2 error norm are shown in Table 4. The results show that the TS model we use is lower than the ST model in all indicators, which shows that the TS containing the MST-UNet model has good performance on the spatio-temporal SR task.



**Fig. 12** Performance of the ST model on the spatio-temporal super-resolution task

**Table 4** Results of the two models on the test set

Dataset	Model	MAE ( $\times 10^{-4} s^{-1}$ )	$L_2$ error norms
Dataset <sub>s</sub>	TS	5.20	0.059
	ST	7.40	0.076
Dataset <sub>t</sub>	TS	3.02	0.039
	ST	4.32	0.048

## 4 Conclusions

In this paper, we propose the MST-UNet model for SR reconstruction of spatio-temporal flow fields. In order to obtain the HR result of the current frame, the MST-UNet model combines the LR information of the current moment and the HR information of the previous moment. In addition, we apply the MST-UNet model to the task of flow field reconstruction around cylinders and verify the proposed model on two tasks: spatial SR and spatio-temporal SR. The experimental results show that the MST-UNet model is much better than the other models that only use LR information as input in the spatial SR task. In the spatio-temporal SR task, the results of the model with MST-UNet are significantly improved compared with the model that only uses low-resolution information.

Although we explore a novel model for the spatio-temporal SR reconstruction problem, many topics still exist for further investigation. Our model adopts a data-driven approach, which requires a lot of label data. We will adopt a physics-informed approach to reduce data costs in future work.

### Acknowledgements

Not applicable.

### Authors' contributions

Conceptualization: KB, WP, XZ and WY; methodology: KB, XZ and WP; software: KB and WP; validation: KB, WY and XZ; formal analysis: WY and XZ; investigation: KB and WP; resources: KB, WY and WP; data curation: KB and WP; writing—original draft preparation: KB; writing—review and editing: KB, XZ and WP; visualization: KB and WP; supervision: XZ and WY; project administration: WY. All authors read and approved the final manuscript.

### Funding

Not applicable.

### Availability of data and materials

The code of the neural network and data generator can be downloaded at: <https://github.com/baokairui/MST-UNet>.

## Declarations

### Competing interests

The authors declare that they have no competing interests.

Received: 20 December 2022 Accepted: 6 April 2023

Published online: 01 June 2023

## References

1. Chang RY, Yang WH (2001) Numerical simulation of mold filling in injection molding using a three-dimensional finite volume approach. *Int J Numer Methods Fluids* 37(2):125–148
2. Fakhari A, Lee T (2014) Finite-difference lattice Boltzmann method with a block-structured adaptive-mesh-refinement technique. *Phys Rev E* 89(3):033310
3. Strniša F, Urbic T, Plazl I (2020) A lattice Boltzmann study of 2D steady and unsteady flows around a confined cylinder. *J Braz Soc Mech Sci Eng* 42:103
4. Koda Y, Lien FS (2015) The lattice Boltzmann method implemented on the GPU to simulate the turbulent flow over a square cylinder confined in a channel. *Flow Turbulence Combust* 94(3):495–512



5. Wang Y, Shu C, Teo CJ et al (2015) An immersed boundary-lattice Boltzmann flux solver and its applications to fluid-structure interaction problems. *J Fluids Struct* 54:440–465
6. Zheng HW, Shu C, Chew YT (2006) A lattice Boltzmann model for multiphase flows with large density ratio. *J Comput Phys* 218(1):353–371
7. Hennigh O (2017) Lat-Net: compressing lattice Boltzmann flow simulations using deep neural networks. arXiv preprint [arXiv:1705.09036](https://arxiv.org/abs/1705.09036)
8. Carrillo M, Que U, González JA (2016) Estimation of Reynolds number for flows around cylinders with lattice Boltzmann methods and artificial neural networks. *Phys Rev E* 94(6):063304
9. Rabbani A, Babaei M (2019) Hybrid pore-network and lattice-Boltzmann permeability modelling accelerated by machine learning. *Adv Water Resour* 126:116–128
10. Janssens N, Huysmans M, Swennen R (2020) Computed tomography 3D super-resolution with generative adversarial neural networks: implications on unsaturated and two-phase fluid flow. *Materials* 13(6):1397
11. Kim H, Kim J, Won S et al (2021) Unsupervised deep learning for super-resolution reconstruction of turbulence. *J Fluid Mech* 910:A29
12. Sekar V, Jiang Q, Shu C et al (2019) Fast flow field prediction over airfoils using deep learning approach. *Phys Fluids* 31(5):057103
13. Han J, Tao J, Wang C (2018) FlowNet: A deep learning framework for clustering and selection of streamlines and stream surfaces. *IEEE Trans Vis Comput Graph* 26(4):1732–1744
14. Srinivasan PA, Guastoni L, Azizpour H et al (2019) Predictions of turbulent shear flows using deep neural networks. *Phys Rev Fluids* 4(5):054603
15. Fukami K, Nabae Y, Kawai K et al (2019) Synthetic turbulent inflow generator using machine learning. *Phys Rev Fluids* 4(6):064603
16. Nakamura T, Fukami K, Hasegawa K et al (2021) Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys Fluids* 33(2):025116
17. Linot AJ, Graham MD (2022) Data-driven reduced-order modeling of spatiotemporal chaos with neural ordinary differential equations. *Chaos* 32(7):073110
18. Fukami K, Fukagata K, Taira K (2019) Super-resolution reconstruction of turbulent flows with machine learning. *J Fluid Mech* 870:106–120
19. Liu Y, Ponce C, Brunton SL et al (2020) Multiresolution convolutional autoencoders. arXiv preprint [arXiv:2004.04946](https://arxiv.org/abs/2004.04946)
20. Gao H, Sun L, Wang JX (2021) Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Phys Fluids* 33(7):073603
21. Han J, Wang C (2022) TSR-VFD: Generating temporal super-resolution for unsteady vector field data. *Comput Graph* 103:168–179
22. Goodfellow I, Pouget-Abadie J, Mirza M et al (2020) Generative adversarial networks. *Commun ACM* 63(11):139–144
23. Güemes A, Discetti S, Ianiro A et al (2021) From coarse wall measurements to turbulent velocity fields through deep learning. *Phys Fluids* 33(7):075121
24. Deng Z, He C, Liu Y et al (2019) Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Phys Fluids* 31(12):125111
25. Yousif MZ, Yu L, Lim HC (2021) High-fidelity reconstruction of turbulent flow from spatially limited data using enhanced super-resolution generative adversarial network. *Phys Fluids* 33(12):125119
26. Guastoni L, Güemes A, Ianiro A et al (2021) Convolutional-network models to predict wall-bounded turbulence from wall quantities. *J Fluid Mech* 928:A27
27. Towne A, Schmidt OT, Colonius T (2018) Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *J Fluid Mech* 847:821–867
28. Liu B, Tang J, Huang H et al (2020) Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys Fluids* 32(2):025105
29. Fukami K, Fukagata K, Taira K (2021) Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J Fluid Mech* 909:A9
30. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W et al (eds) *Medical image computing and computer-assisted intervention – MICCAI 2015. Lecture notes in computer science*, vol 9351. Springer, Cham, pp 234–241
31. Wu Y, He K (2018) Group normalization. In: Ferrari V, Hebert M, Sminchisescu C et al (eds) *Computer vision – ECCV 2018. Lecture notes in computer science*, vol 11217. Springer, Cham, pp 3–19
32. Wu Z, Xiong Y, Yu SX et al (2018) Unsupervised feature learning via non-parametric instance discrimination. In: *Proceedings of the 2018 IEEE/CVF conference on computer vision and pattern recognition*, Salt Lake City, 18–23 June 2018, pp 3733–3742
33. Zhang K, Haque MdN (2022) Wake interactions between two side-by-side circular cylinders with different sizes. *Phys Rev Fluids* 7(6):064703
34. Morimoto M, Fukami K, Zhang K et al (2022) Generalization techniques of neural networks for fluid flow estimation. *Neural Comput Appl* 34(5):3647–3669
35. Succi S (2001) *The lattice Boltzmann equation for fluid dynamics and beyond*. Clarendon Press, Oxford
36. Lee C, Yang W, Parr RG (1988) Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys Rev B* 37(2):785

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.