

## Predicting unavailable parameters from existing velocity fields of turbulent flows using a GAN-based model

Linqi Yu <sup>1,\*</sup> Mustafa Z. Yousif <sup>1,\*</sup> Young-Woo Lee,<sup>1</sup> Xiaojue Zhu <sup>2</sup> Meng Zhang <sup>1</sup>  
Paraskovia Kolesova <sup>1</sup> and Hee-Chang Lim <sup>1,†</sup>

<sup>1</sup>*School of Mechanical Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan, 46241, Republic of Korea*

<sup>2</sup>*Max Planck Institute for Solar System Research, Justus-von-Liebig-Weg 3, Göttingen 37077, Germany*



(Received 18 April 2023; accepted 17 January 2024; published 20 February 2024)

In this study, an efficient deep-learning model is developed to predict unavailable parameters, e.g., streamwise velocity, temperature, and pressure from available velocity components. This model, termed mapping generative adversarial network (M-GAN), consists of a label information generator (LIG) and an enhanced super-resolution generative adversarial network. LIG can generate label information helping the model to predict different parameters. The GAN-based model receives the label information from LIG and existing velocity data to generate the unavailable parameters. Two-dimensional (2D) Rayleigh-Bénard flow and turbulent channel flow are used to evaluate the performance of M-GAN. First, M-GAN is trained and evaluated by two-dimensional direct numerical simulation (DNS) data of a Rayleigh-Bénard flow. From the results, it can be shown that M-GAN can predict temperature distribution from the two-dimensional velocities. Furthermore, DNS data of turbulent channel flow at two different friction Reynolds numbers  $Re_\tau = 180$  and  $550$  are applied simultaneously to train the M-GAN and examine its predicting ability for the pressure fields and the streamwise velocity from the other two velocity components. The instantaneous and statistical results of the predicted data agree well with the DNS data, even for the flow at  $Re_\tau = 395$ , indicating that M-GAN can be trained to learn the mapping function of the unknown fields with good interpolation capability.

DOI: [10.1103/PhysRevFluids.9.024603](https://doi.org/10.1103/PhysRevFluids.9.024603)

### I. INTRODUCTION

As a traditional topic in experimental and computational fluid dynamics (CFD), turbulence has been investigated extensively to enhance the understanding of its complicated characteristics and chaotic behavior over a hundred years. Visualizing and statistically analyzing turbulent flows requires massive data from flow fields. With various methods developed, high-fidelity turbulence data can be efficiently generated for aerodynamic simulations and other scientific applications. For instance, direct numerical simulation (DNS) has been extensively applied in the CFD field to simulate a variety of turbulent flows precisely at a specific range of Reynolds numbers ( $Re$ ), whose

---

\*These authors contributed equally to this work.

†Corresponding author: hclim@pusan.ac.kr.

main mechanism solves the Navier-Stokes equations numerically [1–3]. From the perspective of experimental measurement, one of the most widely used methods is particle-image velocimetry (PIV), which plays a crucial role in the experimental investigation of turbulent flow fields [4]. However, if an ordinary two-dimensional (2D) PIV system is employed, only two velocity components, e.g., wall-normal velocity ( $v$ ) and spanwise velocity ( $w$ ), can be obtained but without streamwise velocity ( $u$ ).

Note here that an unavailable parameter generally refers to a parameter that cannot be measured directly using 2D measurement techniques such as PIV measurement. For instance, if the normal section to the streamwise direction of the flow is measured, the streamwise component cannot be obtained directly and therefore is considered an unavailable parameter. In addition, pressure ( $p$ ) and temperature ( $T$ ) also cannot be obtained directly by those measurements. These unavailable parameters are undoubtedly important for turbulent flow analysis and engineering applications. Therefore, these three velocity components are required to analyze the instantaneous flow structures and statistics of turbulent flow [5,6]. In addition, the fluctuating pressure field over the bluff bodies under turbulent boundary layers causes severe noise and vibration [7]. Moreover, the pressure fields within the turbulent flow field are crucial to analyze the coupling mechanisms of turbulence-acoustic [8] and flow control strategies for reducing noise [9]. Temperature is also crucial in turbulence fields, as its gradient drives the heat transfer phenomenon. Additionally, temperature gradients in the near-wall region significantly impact the overall performance of many engineering systems [10,11].

Currently, several methods are applied to obtain the aforementioned unavailable parameters. Therefore, in general, acquiring the full components of turbulent flow fields requires a high-performance PIV system such as a tomographic PIV (Tomo PIV) [12], which realizes the acquisition of three-dimensional (3D) turbulent flow with full velocity components. However, in order to obtain the full velocity components, it would be necessary to incur additional costs and employ complex processes to achieve the desired level of precision. As for pressure measurement by PIV, Oudheusden [4] proposed a way to manipulate the instantaneous pressure field by combining the experimental data with the governing equations, i.e., the Poisson equation. Thermographic PIV (Thermo PIV) [13,14] has been developed to measure velocity fields and temperature based on thermographic phosphor particles, which possess temperature-dependent luminescence properties. Compared to the original PIV, Thermo PIV needs extra setup and equipment, which include an extra ultraviolet laser and two extra cameras with suitable spectral filters. Nonetheless, it is difficult to predict unavailable parameters such as  $u$ ,  $p$ , and  $T$ , based on the available data ( $v$  and  $w$ ) obtained from an ordinary 2D PIV system. In addition to the above experimental methods, this is an alternative mathematical method to generate the unavailable parameters from known flow fields, termed data assimilation. He *et al.* [15] used data assimilation to determine the pressure of turbulent velocity fields measured by PIV based on the unsteady adjoint formulation. However, implementing these methods still needs the assistance of Tomo PIV. In other words, the turbulence velocity data, e.g., wall-normal velocity  $v$  and spanwise velocity  $w$ , from the 2D PIV system are insufficient to solve the governing equation and apply data assimilation for getting pressure field because of the lack of one of the velocity components, i.e., streamwise velocity  $u$ . Therefore, the aim and innovation of this study is to find a data-driven-based method to predict the unavailable parameters mentioned earlier from existing flow fields using deep learning.

Deep-learning (DL) algorithms are rapidly developing and extensively used in various fields [16,17]. Recently, DL has been widely utilized in fluid dynamics, benefiting from its capability of highly nonlinear mapping [18]. DL is of great interest in the following problems about turbulent flows: temporal flow data generation based on turbulent modeling [19,20]; fluid flow simulation [21]; reduced-order modeling [22,23]; prediction of turbulent flow based on the information from previous temporal data [24–26]; super-resolution reconstruction of turbulent flow [27–30]. Another important application of DL in fluid dynamics is mapping parameters of the flow field, namely,

mapping some parameters obtained from a fluid flow to the other different parameters, which can be expressed as

$$f : P_a \rightarrow P_u, \quad (1)$$

where  $P_a$  is termed the set of available or already existing parameters,  $P_u$  is the set of unavailable parameters, and  $f$  is the mapping rule taking the elements in  $P_a$  to the one in  $P_u$ . The  $f$  of note is the complex nonlinear functions hidden in deep-learning neural network, which are learned by the DL model through training. Guastoni *et al.* [31] applied convolutional neural network models to map the wall-shear-stress components and the wall pressure to the 2D instantaneous velocity-fluctuation fields at different wall-normal locations in a turbulent open-channel flow. Jouybari *et al.* [32] designed a multilayer perceptron type neural network to get the mapping of various rough surface statistics to equivalent sand-grain height. Lee *et al.* [33] proposed a transfer learning method based on empirical correlations to predict the drag force on the statistics of rough surfaces. Moreover, the mapping that takes local wall-shear stresses and wall-pressure fluctuations to local heat flux was proved to be possible by Kim and Lee [34].

Furthermore, generative adversarial networks (GAN) proposed by Goodfellow *et al.* [35] have been used in various fields, e.g., the work of Shamsolmoali *et al.* [36] for synthetic image generation. Other powerful networks based on GAN are super-resolution generative adversarial network (SRGAN) [37], and the variant of SRGAN, i.e., enhanced SRGAN (ESRGAN) [38]. Recently, GAN-based DL models have been rapidly implemented to solve turbulent flow problems, particularly resolution reconstruction problems. Benefiting from deeper layers and special loss functions (e.g., perceptual loss), SRGAN and ESRGAN show better performance on the resolution reconstruction of flow fields than basic other DL models [39]. Moreover, Yousif *et al.* [40] proposed a novel 2D3DGAN based on the mechanism of ESRGAN, which could reconstruct the 3D turbulent flow fields from 2D velocity fields. Regarding flow field parameters mapping, Güemes *et al.* [41] evaluated the performance of SRGAN for reconstructing turbulent-flow quantities from coarse wall measurements. The results showed that the SRGAN could capture and rebuild the large-scale structures of the flow even for the most complicated cases. Undoubtedly, the GAN-based DL models also have the potential to tackle more turbulence-related problems.

In this paper, we attempt to predict unavailable parameters ( $P_u$ ) from existing available parameters ( $P_a$ ) using a deep-learning model that can help find a mapping rule between  $P_u$  and  $P_a$ . A novel GAN-based model is proposed, termed mapping-generative adversarial network (M-GAN), and is applied to map available parameters to unavailable parameters. Two cases of fluid flows are used to examine the performance of M-GAN. The 2D Rayleigh-Bénard (RB) flow at Rayleigh number ( $Ra$ ) =  $10^8$  is used as a demonstration, where  $u$  and  $v$  are regarded as available parameters; meanwhile,  $T$  is the unavailable parameter needing to be predicted. Furthermore, turbulent channel flow data at  $Re_\tau = 180, 395, \text{ and } 550$ , are applied to test the M-GAN model sufficiently. 2D flow fields at  $Re_\tau = 180$  and  $550$  are the data for training and testing for M-GAN, i.e.,  $v$ , and  $w$  are input data, and  $u$  and  $p$  are output data. Then channel flow at  $Re_\tau = 395$ , which has never participated in the training process, is used as an additional testing case to estimate the interpolation ability of M-GAN. Note that a label information generator (LIG) is combined with M-GAN to train and test the turbulent channel flow cases. The role of LIG is to help M-GAN decide to output  $u$  or  $p$  when the DL model can only receive the same information of  $v$  and  $w$ . All the mentioned flow data are generated using DNS.

The remainder of this paper is written as follows. Section II presents the generation of flow data using DNS. The design of M-GAN is introduced in the methodology part (Sec. III), where the training and testing procedures are also included. The results, including instantaneous contours and turbulence statistics, are plotted and discussed in Sec. IV. Finally, Sec. V summarizes the paper and proposes potential future research based on this study.

TABLE I. Simulation parameters of Rayleigh-Bénard flow.  $N_x$  and  $N_y$  are the grid resolution in the horizontal and vertical directions.  $\Gamma$  is the aspect ratio,  $\Gamma = W/L$ , where  $W$  and  $L$  are the width and depth of the domain.  $Pr$  and  $Nu$  are the Prandtl number and Nusselt number.  $\Delta t^+$  is the dimensionless time step of the simulation. The superscript “+” represents that the quantity is nondimensionalized by  $u_\tau$  and  $\nu$ .

Ra	$N_x \times N_y$	$\Gamma$	Pr	Nu	$\Delta t^+$
$10^8$	$512 \times 256$	2	1	26.1	0.1

## II. DATA GENERATION

### A. 2D Rayleigh-Bénard flow

In the Rayleigh-Bénard flow case, the governing equations with incompressibility condition can be expressed as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}_b, \quad (2)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (3)$$

As for the temperature field, an advection-diffusion equation is applied as

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \kappa \nabla^2 T, \quad (4)$$

where  $\mathbf{u}$ ,  $p$ ,  $\rho$ ,  $T$ ,  $\nu$ ,  $\kappa$ , and  $t$  represent velocity vector, pressure, density, temperature, kinematic viscosity, thermal diffusivity, and time, respectively. The Boussinesq approximation is applied for this flow and the body force  $\mathbf{F}_b$  is taken to only depend linearly on the temperature and to be in the direction of gravity. Besides, the possible dependencies of density, viscosity, and thermal diffusivity on temperature are ignored, so these parameters are considered constant.

The simulation is carried out by performing DNS using a well-validated second-order finite-difference code [11,42]. One important control parameter of Rayleigh-Bénard flow is the Rayleigh number, i.e.,  $Ra = \alpha g \Delta L^3 / (\nu \kappa)$ , where  $\alpha$  is the thermal expansion coefficient,  $g$  is the acceleration of gravity, and  $\Delta$  is the temperature difference between the upper and bottom surface with a depth  $L$ . The no-slip and constant temperature boundary conditions are applied for the bottom and top plates, and periodic boundary condition is assigned to the horizontal direction. The details of this case are listed in Table I. More information on the simulation of Rayleigh-Bénard flow can be found in these papers [11,43].

In this case, 13000 snapshots are generated and collected, where the data collection time step ( $\Delta t_c^+$ ) is equal to simulation  $\Delta t^+$ . Among the 13000 snapshots, the first 9000 snapshots are used as training data, while the rest 3000 snapshots are used to test the trained model. To save the computational expense, all the data are interpolated to reduce the resolution from  $512 \times 256$  to  $256 \times 128$ . Besides, the dissimilarity between training and testing data has been evaluated by comparing the same number of snapshots in every eddy turnover period. Thus, the testing data is valid.

### B. Turbulent channel flow

In the turbulent channel flow case, the momentum equation for an incompressible viscous fluid is expressed as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}. \quad (5)$$

The open-source CFD finite-volume code OPENFOAM-5.0x is applied to perform the quasi-DNS calculations. For convenience, the term DNS is used in the next sections of the paper. As mentioned

TABLE II. Simulations parameters of two turbulent channel flows.  $L$  and  $N$  are the domain dimension and the number of grids. The superscript “+” represents that the quantity is nondimensionalized by  $u_\tau$  and  $\nu$ .  $\Delta y_w^+$  and  $\Delta y_c^+$  are the spacing near the wall and at the center of the channel.

$Re_\tau$	$L_x \times L_y \times L_z$	$N_x \times N_y \times N_z$	$\Delta x^+$	$\Delta z^+$	$\Delta y_w^+$	$\Delta y_c^+$	$\Delta t^+$
180	$4\pi\delta \times 2\delta \times 2\pi\delta$	$256 \times 128 \times 256$	8.831	4.415	0.63	4.68	0.113
395	$4\pi\delta \times 2\delta \times 2\pi\delta$	$385 \times 257 \times 385$	12.553	6.277	0.541	5.115	0.023
550	$4\pi\delta \times 2\delta \times 2\pi\delta$	$512 \times 336 \times 512$	13.492	6.746	0.401	5.995	0.030

earlier, the current study has used three different friction Reynolds numbers, i.e.,  $Re_\tau = u_\tau\delta/\nu = 180, 395,$  and  $550$ , where  $u_\tau$  is the friction velocity, and  $\delta$  is half of the channel height. The detailed simulation parameters of each  $Re_\tau$  are listed in Table II. The periodic boundary condition is used in the streamwise ( $x$ ) and spanwise ( $z$ ) directions. The no-slip boundary condition is applied to the upper and lower walls of the channel. Moreover, the obtained turbulence statistics have been validated by comparing them with the results from Kim *et al.* [2] and Moser *et al.* [3].

We use the pressure implicit split operator algorithm to solve the coupled pressure momentum system. We discretize the convective fluxes using a second-order accurate linear upwind scheme. Additionally, all the discretization schemes employed in the simulations, including convective fluxes, have second-order accuracy. We maintain the maximum Courant-Friedrichs-Lewy (CFL) number below 1 during the simulations to ensure stability.

The 2D data of a  $y$ - $z$  plane from the 3D channel flow domain are collected. 10000 snapshots of flow velocity and pressure fields are obtained from channel flows at  $Re_\tau = 180$  and  $550$ , where the  $\Delta t_c^+$  to collect data is ten times as the simulation  $\Delta t^+$ . Among the 10000 snapshots, the front 9000 snapshots are used as training data, while the remaining 1000 snapshots are used as test data. Meanwhile, 1000 snapshots of channel flows at  $Re_\tau = 395$  are obtained to do the interpolation test. The data of the flow at  $Re_\tau = 395$  and  $550$  are interpolated to match the grid size of the DNS data of the flow at  $Re_\tau = 180$ , which is 128 and 256 in  $y$  and  $z$  directions, respectively. By applying this data processing technique, we can ensure that the flow data at all three Reynolds numbers are suitable inputs for the model.

### C. Data nondimensionalization

All the flow field parameters and time used in this study are made in nondimensional form. First, velocity is nondimensionalized using friction velocity  $u_\tau$ , represented as  $U^+ = U/u_\tau$ . The temperature  $T$  in the Rayleigh-Bénard case is nondimensionalized by  $T^+ = T/T_\tau$ , where  $T_\tau = -\kappa\partial_y\langle T \rangle_{x,t}|_{y=0}/u_\tau$  is a characteristic temperature scale like the  $u_\tau$  for the velocity [44] and  $T_b$  is the bottom plate temperature. The pressure  $p$  in the channel flow case is nondimensionalized by density  $\rho$  and  $u_\tau$ :  $p^+ = p/(\rho u_\tau^2)$ . The dimensionless wall-normal direction distance  $y^+$  is defined as  $y^+ = (yu_\tau)/\nu$ .

Moreover, the nondimensionalization of time is represented by several forms in this work. For the Rayleigh-Bénard flow case, the cycle time is defined as  $T_{\text{cycle}} = N/40$ , where  $N$  is the number of snapshots and 40 means one cycle includes 40 snapshots. Thus, the Rayleigh-Bénard data’s period is  $T_{\text{cycle}} = 300$ . For the channel flow case, flow-through time  $T_{ft}$  is defined as  $T_{ft} = tU_{\text{center}}/L_x = t/4\pi$ , where  $L_x$  is the domain length in the  $x$  direction,  $U_{\text{center}}$  is the velocity along the center line. However, for the unified representation of both Rayleigh-Bénard flow and channel flow cases, we use the  $t^+$  to represent dimensionless time expressed as  $t^+ = tu_\tau^2/\nu$ .

The detailed information of all the data sets is summarized in Table III, where data collection interval time  $\Delta t_c^+$ , number of snapshots, available parameters  $P_a$ , and unavailable parameters  $P_u$  of various cases are listed.

TABLE III. Information of data sets.

Case	$\Delta t_c^+$	Number of snapshots			Interpolation test data	Available parameters $P_a$	Unavailable parameters $P_u$
		Collected data	Training data	Test data			
Rayleigh- Bénard flow	0.1	13000	10000	3000	0	$u, v$	$T$
Channel flow at $Re_\tau = 180$	1.13	10000	9000	1000	0	$v, w$	$u, p$
Channel flow at $Re_\tau = 395$	0.23	1000	0	0	1000	$v, w$	$u, p$
Channel flow at $Re_\tau = 550$	0.3	10000	9000	1000	0	$v, w$	$u, p$

### III. METHODOLOGY

#### A. M-GAN

GAN, developed by Goodfellow *et al.* [35], has been successfully applied to tackle the image generation problem. Subsequently, based on the traditional GAN, Ledig *et al.* [37] and Wang *et al.* [38] proposed SRGAN and ESRGAN, respectively. Generally, ordinary GAN is unsupervised learning where the adversarial loss dominates the training process. However, SRGAN and ESRGAN integrate both supervised and unsupervised learning. SRGAN and ESRGAN have both supervised and unsupervised loss (adversarial loss), which can guarantee the target prediction or reconstruction is more accurate. Besides, ESRGAN has deeper and more complicated layers with a powerful ability to capture or reconstruct features of image data. Thus, ESRGAN performs more excellently in recovering high-resolution images from multiple images. The turbulent flow field has complex, multiscale, and chaotic features, making reconstructing and predicting turbulent flows more difficult. But, in the turbulence field, ESRGAN has proven its capability to reconstruct the super-resolution data of turbulent flows [28–30]. Considering the previous great mapping ability of ESRGAN for resolution reconstruction, this study intends to develop the M-GAN to map some flow field parameters to other parameters.

Like the traditional GAN, M-GAN comprises two parts, i.e., generator ( $G$ ) and discriminator ( $D$ ). As shown in Fig. 1(a),  $G$  mainly contains deep convolutional neural networks (CNNs)-based multilayers termed residual in residual dense blocks (RRDBs) [45]. The subelements of RRDBs are dense blocks (DBs) consisting of convolutional and leaky ReLU activation function layers. Skip connection is applied in RRDBs and DBs to avoid vanishing gradient. In addition,  $G$  comprises a convolution-pooling part (CPP) following the input and a multiscale part (MSP) preceding the output. The CPP module includes several convolutional and max pooling layers that extract input data information and compress the data size. Conversely, the MSP module uses upsampling layers and convolutional layers with various-sized filters to extract information from the output of RRDBs and construct the output data. More details of CPP and MSP are listed in Tables IV and V, respectively. Besides, the label information generator (LIG) is added to the input of  $G$ . LIG is used in specific situations, such as  $v$  and  $w$  are the input, and  $u$  or  $p$  are the output. Note that LIG can provide information to  $G$  for determining which parameters to output. Table VI shows the detailed structure of LIG. The label is passed into a dense layer to generate label information. After reshaping, the label information is concatenated to  $p_a$  and input into the next process. Notably, the use of LIG is optional. Sometimes LIG can be neglected when the outputting only includes one parameter in  $P_u$ . Figure 1(b) shows the detailed structure of  $D$ . The major component of  $D$  is CBL, including three different layers, i.e., convolutional, batch normalization, and LReLU activation function layers.

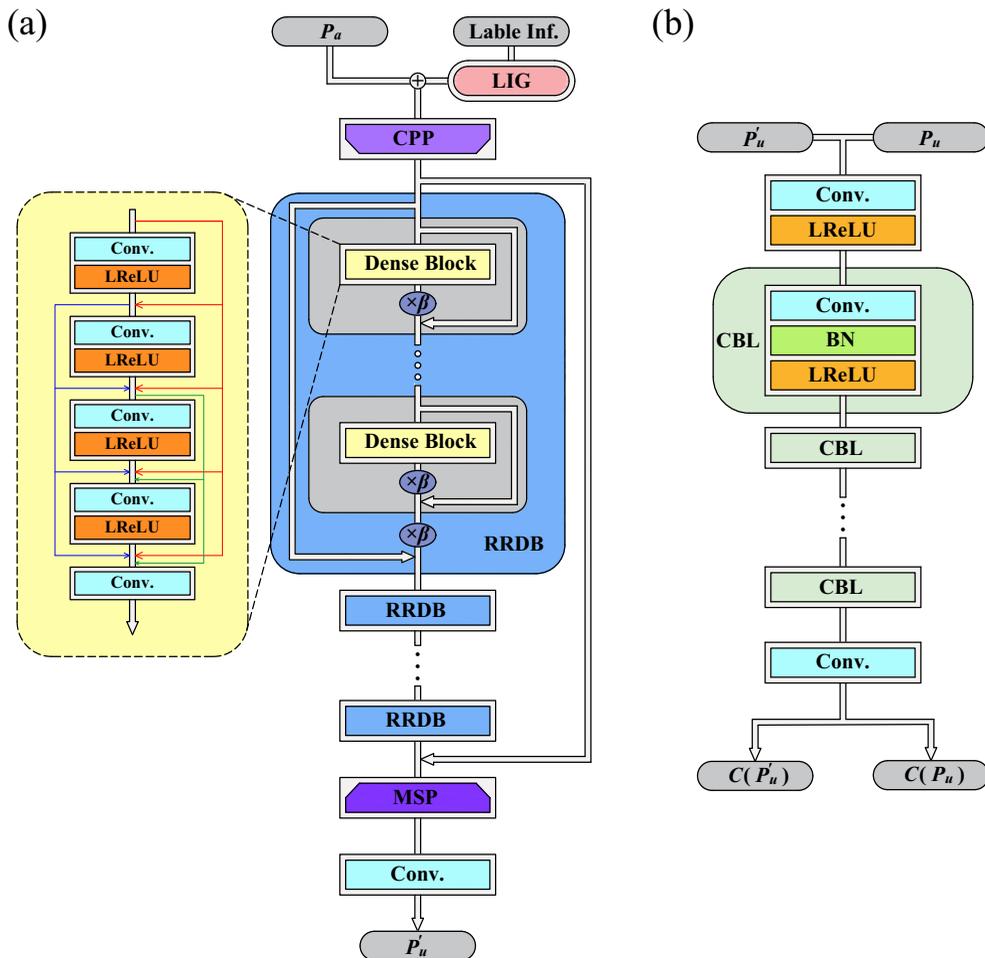


FIG. 1. Architecture of M-GAN: (a) generator and (b) discriminator, where LIG is the label information generator, CPP is the convolution-pooling part, RRDBs is the residual dense blocks, MSP is the multiscale part, and CBL is the convolutional-batch normalization-LReLU integrated layers.

The overall algorithm is described as follows. First,  $P_a$  and label information (here, we assume to have the label information) are input to  $G$ . The generated unavailable parameters ( $P'_u$ ) are obtained from the output of  $G$ . Then, both  $P_u$  (real one) and  $P'_u$  (generated one) are fed into  $D$ .

TABLE IV. CCP structure.

Type of layers	Shape
Input ( $P_a$ )	(128, 256, 2)
Conv2D.(3, 3)	(128, 256, 56)
MaxPooling(2, 2)	(64, 128, 56)
Conv2D.(3, 3)	(64, 128, 56)
MaxPooling(2, 2)	(32, 64, 56)
concat (Output of last layer and LIG)	(32, 64, 64)

TABLE V. MSP architecture.

First branch	Second branch	Third branch
Conv2D.(3, 3)	Conv2D.(5, 5)	Conv2D.(7, 7)
UpSampling(2, 2)	UpSampling(2, 2)	UpSampling(2, 2)
Conv2D.(3, 3)	Conv2D.(5, 5)	Conv2D.(7, 7)
LeakyReLU	LeakyReLU	LeakyReLU
UpSampling(2, 2)	UpSampling(2, 2)	UpSampling(2, 2)
Conv2D.(3, 3)	Conv2D.(5, 5)	Conv2D.(7, 7)
LeakyReLU	LeakyReLU	LeakyReLU
Add (first branch, second branch, third branch)		

The nontransformed discriminator value  $D_{Ra}$  is calculated from the output of  $D$ , which is formulated as

$$D_{Ra}(P_u, P'_u) = \sigma(C(P_u) - \mathbb{E}_{P'_u}[C(P'_u)]), \quad (6)$$

$$D_{Ra}(P'_u, P_u) = \sigma(C(P'_u) - \mathbb{E}_{P_u}[C(P_u)]), \quad (7)$$

where  $\sigma$  is the sigmoid function, and  $\mathbb{E}$  is the average calculating operator. Similar to ESRGAN,  $D$  is designed to predict the probability that  $P_u$  is relatively more realistic than  $P'_u$ . Using Eqs. (6) and (7), the discriminator loss  $L_D^{Ra}$  and adversarial loss  $L_G^{Ra}$  are calculated. The formulation of the discriminator loss ( $\mathcal{L}_D$ ) is expressed as

$$\mathcal{L}_D = L_D^{Ra} = -\mathbb{E}_{P_u}[\ln(D_{Ra}(P_u, P'_u))] - \mathbb{E}_{P'_u}[\ln(1 - D_{Ra}(P'_u, P_u))]. \quad (8)$$

Besides, the adversarial loss is expressed as

$$L_G^{Ra} = -\mathbb{E}_{P_u}[\ln(1 - D_{Ra}(P_u, P'_u))] - \mathbb{E}_{P'_u}[\ln(D_{Ra}(P'_u, P_u))]. \quad (9)$$

As mentioned before,  $D$  predicts the probability that the  $P_u$  is relatively more realistic than the  $P'_u$  in the training process. When  $P_u$  is more realistic than  $P'_u$ ,  $D_{Ra}(P_u, P'_u)$  tends to be 1 and  $D_{Ra}(P'_u, P_u)$  tends to be 0. Thus,  $L_D^{Ra}$  will decrease and tend to be 0. Inversely,  $G$  plays a role in generating more realistic  $P'_u$  tending to be similar to  $P_u$ . In this situation,  $D_{Ra}(P_u, P'_u)$  will decrease from 1 to 0, and  $D_{Ra}(P'_u, P_u)$  will increase from 0 to 1, which makes  $L_G^{Ra}$  tend to be 0. The above description is the adversarial process between  $G$  and  $D$ . In this process,  $G$  and  $D$  compete with each other. Meanwhile, they are also promoted mutually.

Although the adversarial loss can guide the  $G$  to generate the targeting results, it is still insufficient for  $G$  to generate accurate results. Herein, in addition to adversarial loss, the loss function of  $G$  ( $\mathcal{L}_G$ ) contains some extra parts, i.e., pixel loss ( $L_{\text{pixel}}$ ), perceptual loss ( $L_{\text{perceptual}}$ ).  $L_{\text{pixel}}$  is the pixel-based error, which is calculated by comparing the value difference between  $P_u$  and  $P'_u$ . Different from  $L_{\text{pixel}}$ ,  $L_{\text{perceptual}}$  is the error between the features of  $P_u$  and  $P'_u$ , which are extracted by the feature extractor (FE). Regarding the original ESRGAN, VGG19 [46], a very deep convolutional network, is used as FE. However, VGG19 is designed to process image data with three channels, i.e., RGB. In other words, VGG19 can only receive the flow data with three components like  $u$ ,  $v$ ,

TABLE VI. LIG structure.

Type of layers	Shape
Input (label inf.)	(1)
Dense	(32 × 64 × 8)
Reshape	(32, 64, 8)

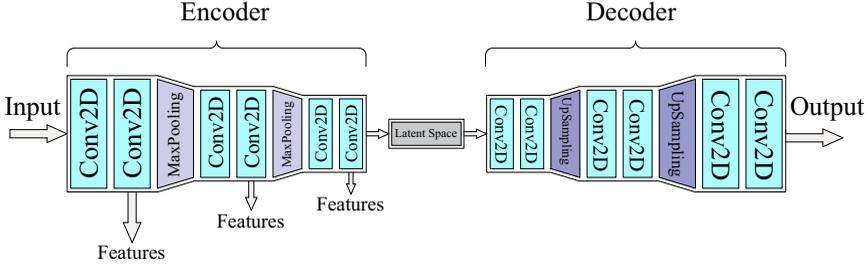


FIG. 2. Architecture of the AE-FE.

and  $w$ . In this study,  $P_u$  and  $P'_u$  only contains one component, so applying VGG19 to extract features is unsuitable. This study develops an autoencoder-based DL algorithm as a FE (AE-FE) to replace VGG19. FE is a pretrained model using  $P_u$  data. As shown in Fig. 2, after the  $P_u$  are input the AE-FE, various features data with different sizes are output from several convolutional layers. The above extra loss terms are computed using the mean-squared error (MSE). The total loss function of the  $G$  is expressed as

$$\mathcal{L}_G = L_G^{\text{Ra}} + \lambda L_{\text{pixel}} + L_{\text{perceptual}}, \quad (10)$$

where,  $\lambda$  is the coefficient used to balance the magnitude of various loss terms, which is set to be 1000.

This study uses the open-source library TENSORFLOW 2.3.0 to implement the DL model. The customized sample PYTHON code for the proposed M-GAN is available online [47].

### B. Training and testing of M-GAN

For the training process of the 2D Rayleigh-Bénard flow case, the input data, i.e.,  $P_a$  are velocity components  $u$  and  $v$ . Meanwhile, the target data, i.e.,  $P_u$  is temperature  $T$ . First,  $u$  and  $v$  are input into  $G$  to output  $P'_u$ , with a batch size of 16. Then, generator loss is calculated using  $P_u$  and  $P'_u$  based on Eq. (10). The discriminator loss is calculated by Eq. (8) through  $P_u$  and  $P'_u$ . After computing the losses, the optimization algorithm updates the weights of both  $G$  and  $D$  to minimize losses during the training period. The adaptive moment estimation (Adam) algorithm is used in this study for optimization [48]. In addition, the target data,  $P_u$ , is normalized using the min-max normalization function with a range of 0–1 to enhance the training performance. After training, the  $u$  and  $v$  values in the test data set are fed into trained M-GAN to generate predicted  $T$  ( $P'_u$ ), which is depicted in Fig. 3.

In the training process of the turbulent channel flow case, the input data  $P_a$  comprise wall-normal velocity  $v$  and spanwise velocity  $w$ , while the target data  $P_u$  consist of the streamwise velocity  $u$  and pressure  $p$ . For predicting  $u$ , a label of 0 is input into LIG, whereas a label of 1 is used for predicting  $p$ . Here, the predicted  $u$  and  $v$  are termed  $P'_u$ . In contrast to the 2D Rayleigh-Bénard flow case, we

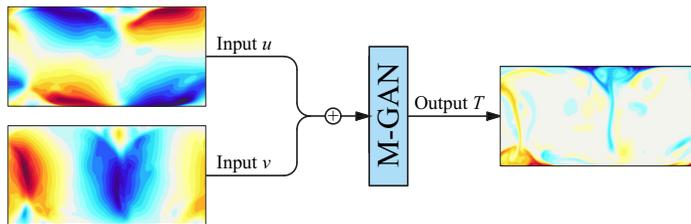


FIG. 3. Schematic of the 2D Rayleigh-Bénard flow case.

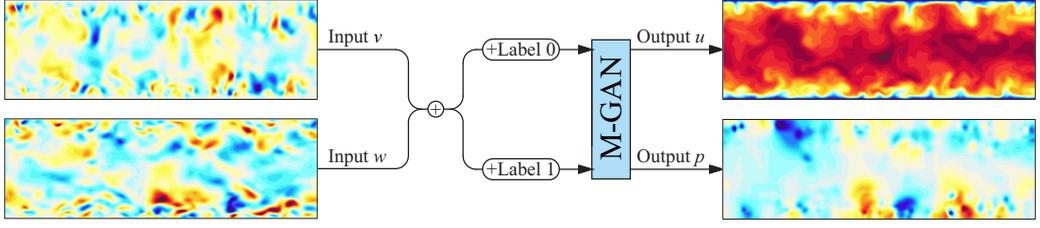


FIG. 4. Schematic of the channel flow case.

train the M-GAN using both channel flow data sets at  $Re_\tau = 180$  and  $550$  together. First,  $v$  and  $w$  are input into  $G$  to generate  $P'_u$  with a batch size of 16. Then,  $P_u$  and  $P'_u$  are used to compute the generator loss through Eq. (10). At the same time, the discriminator loss is yielded based on  $P_u$ ,  $P'_u$ , and Eq. (8). After calculating losses, the Adam optimizer changes the weights of  $G$  and  $D$  to decrease losses while the model is being trained. The data normalization strategy is also used here to make the range of  $P_u$  within 0 and 1. After training, the validation is made to check the performance of the trained M-GAN using test data. As shown in Fig. 4, we input  $v$  and  $w$  from channel flow and the corresponding label into the trained M-GAN to predict the corresponding  $u$  or  $p$ .

Here, the channel flow data at  $Re_\tau = 395$  mentioned above never attend the training process of M-GAN, and they are used to evaluate the interpolation ability of M-GAN. The  $v$  and  $w$  from channel flow at  $Re_\tau = 395$  and label are input into the M-GAN trained by channel flows at  $Re_\tau = 180$  and  $550$  to predict the  $u$  or  $p$ . Here, if the M-GAN trained by channel flows at  $Re_\tau = 180$  and  $550$  could also predict the unavailable parameters of channel flow at  $Re_\tau = 395$  from its  $v$  and  $w$ , it could be proved that the proposed M-GAN has good interpolation ability. In other words, the M-GAN might be appropriate for all the channel flows at a range of Reynolds numbers ( $Re_\tau = 180$ – $550$ ) to predict their unavailable parameters. The verification of the instantaneous and statistical results from all the above cases will be presented in due course.

## IV. RESULTS AND DISCUSSION

### A. 2D Rayleigh-Bénard flow

The capability of M-GAN is examined in this section for the case of 2D Rayleigh-Bénard flow. Figure 5 shows the predicted instantaneous temperature field from the mapping of velocity components using M-GAN. Additionally, the error contours are plotted to show the absolute error ( $\varepsilon_1$ ) between DNS and predicted results. The error is calculated as follows:

$$\varepsilon_1 = |\alpha_{\text{DNS}} - \alpha_{\text{M-GAN}}|, \quad (11)$$

where,  $\alpha_{\text{DNS}}$  and  $\alpha_{\text{M-GAN}}$  represent the DNS data and predicted results from M-GAN, respectively. Additionally,  $|\cdot|$  is the absolute value function.

From these instantaneous temperature contours, we can find that all the detailed features of predicted results using M-GAN are consistent with the corresponding DNS data. Additionally, the error between DNS and predicted results is less than 0.4 in the most areas of error contours. However, the error is relatively large in some areas where the Rayleigh-Bénard convection are located.

Moreover, the mapping ability of M-GAN is further validated by plotting and comparing the statistical results from the predicted and DNS data. The average temperature profile along the wall unit  $y^+$  and the probability density function (PDF) are plotted in Fig. 6. All the predicted results show commendable agreement with the results obtained from DNS data indicating that M-GAN can map the available parameters  $u$  and  $v$  to unavailable parameter  $T$  of the 2D Rayleigh-Bénard flow.

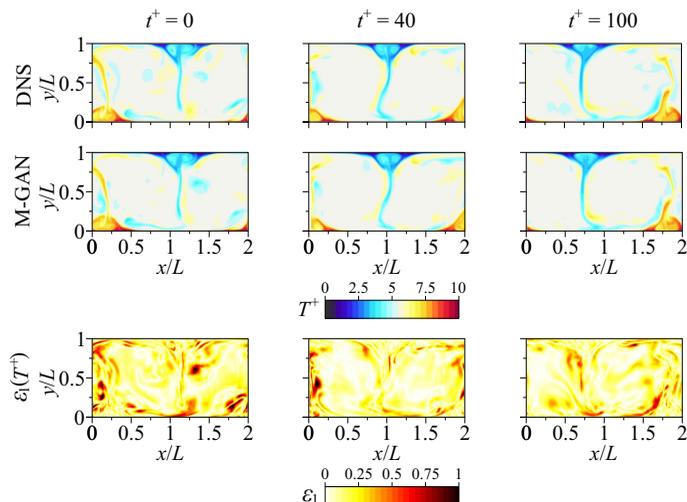


FIG. 5. Predicted instantaneous temperature field of the 2D Rayleigh-Bénard flow case.

Although the results show that M-GAN can work well for the RB flow case, the RB flow case still cannot sufficiently verify the mapping ability of M-GAs, because Rayleigh-Bénard flow does not have very complex and chaotic behavior at a low Rayleigh number ( $Ra = 10^8$ ). Thus, the next section will illustrate a more comprehensive verification using turbulent channel flow data.

### B. Turbulent channel flow

In this section, the mapping capability of M-GAN for taking the available parameters  $v$  and  $w$  to unavailable parameters  $u$  and  $p$  is examined. Figures 7 and 8 show the predicted instantaneous streamwise velocity and pressure fields of the turbulent channel flow at  $Re_\tau = 180$  and 550. The predicted results of both cases agree well with the related DNS results, where most fluctuating features of predicted results are consistent with the DNS data. Moreover, the absolute error contours calculated by Eq. (11) are also plotted in these figures. Through observation by comparing the results based on different parameters and  $Re_\tau$ , we can find that the results of channel flow at  $Re_\tau = 180$

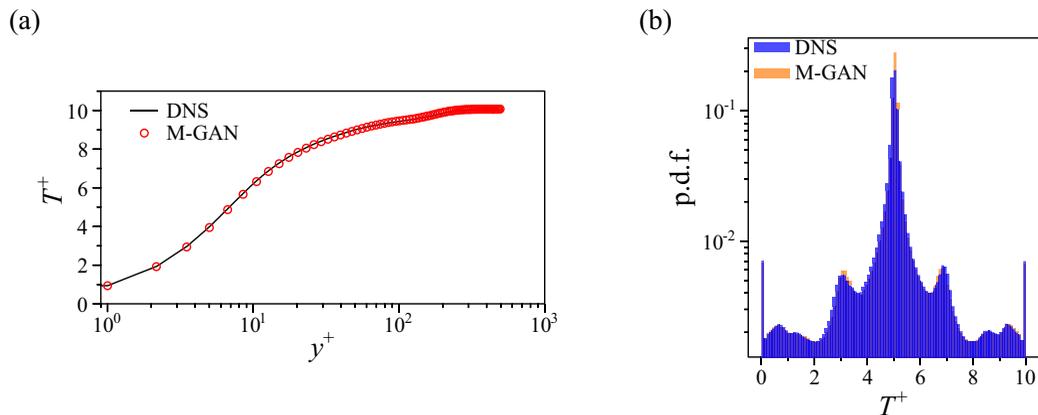


FIG. 6. Statistical results of the 2D Rayleigh-Bénard flow case: (a) average temperature profile; (b) probability density function of temperature field.

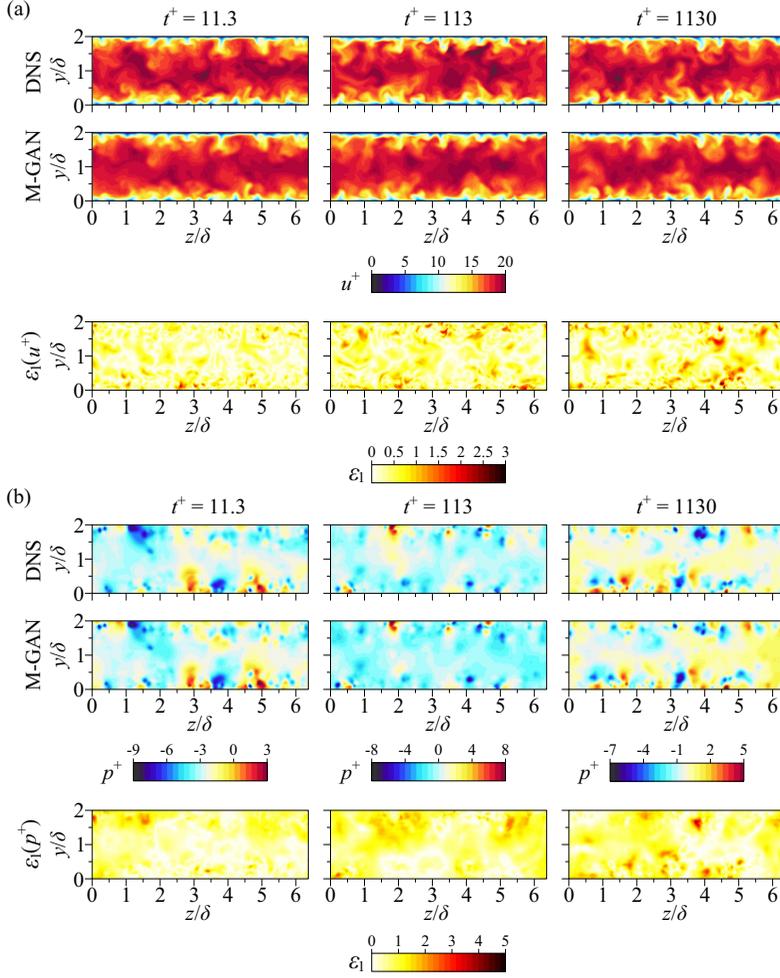


FIG. 7. Predicted instantaneous streamwise velocity and pressure fields of the turbulent channel flow at  $Re_\tau = 180$ .

have more accuracy as compared to the results of the flow at  $Re_\tau = 550$ . Besides, when comparing the results from the same channel flow case, the predicted  $u$  field is better than the results from  $p$ . Figure 9 also indicates this observation, where root-mean-square (RMS) velocity and pressure fluctuations ( $u_{\text{rms}}^+$  and  $p_{\text{rms}}^+$ ) are plotted. The reason for that could be attributed to the fact that pressure fields are more chaotic than streamwise velocity fields, and channel flow at  $Re_\tau = 550$  is more complicated than the one at  $Re_\tau = 180$ .

Figure 10 shows the probability density function (PDF) plots of the  $u$  and  $p$  fields. Note that a larger probability means that the parameter magnitude appears more frequently and dominates the flow field to a greater extent. The PDF of predicted velocity and pressure fields can basically match the reference results. However, the results obtained from channel flow at  $Re_\tau = 550$  show less accuracy than the results from channel flow at  $Re_\tau = 180$ . At the same time, the deviation shown in  $\text{PDF}(p)$  is much larger than the one in  $\text{PDF}(u)$ , which is consistent with the previous discussion in the first paragraph of this section.

In Figs. 11 and 12, the spanwise energy spectrum of  $u$  and  $p$  [ $E_{uu}^+(k_z^+)$  and  $E_{pp}^+(k_z^+)$ ] at various wall distances are plotted to evaluate the ability of M-GAN to predict the  $u$  and  $p$  fields with

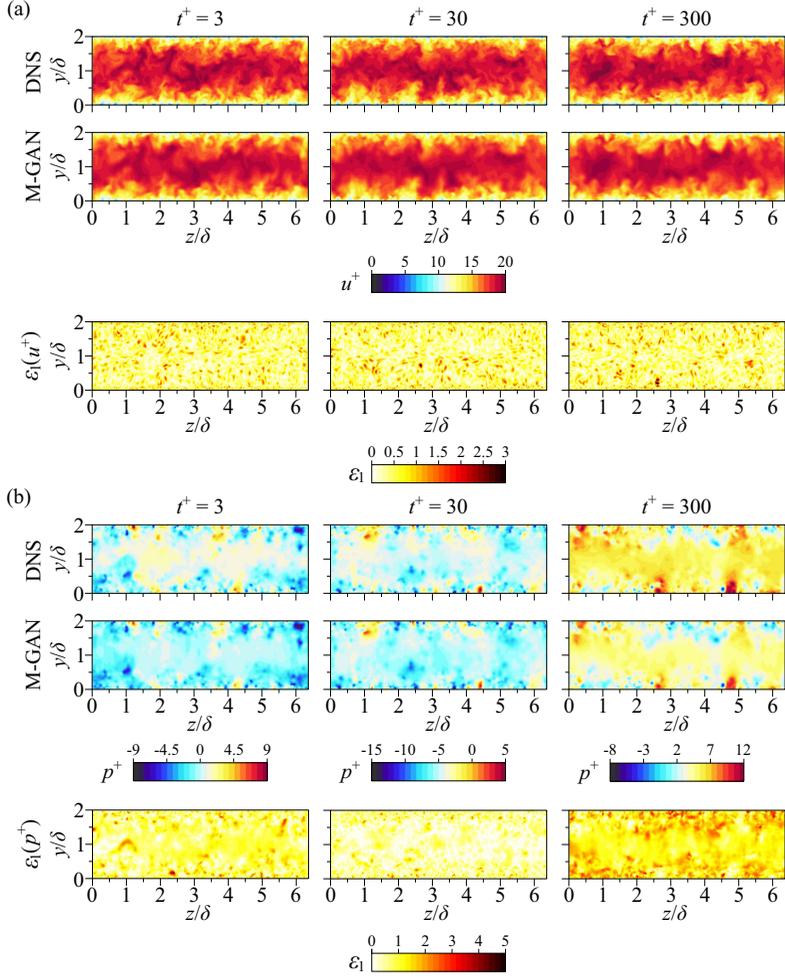


FIG. 8. Predicted instantaneous streamwise velocity and pressure fields of the turbulent channel flow at  $Re_\tau = 550$ .

realistic behavior. In these two figures, the results of  $E_{uu}^+(k_z^+)$  have more accuracy than the results of  $E_{pp}^+(k_z^+)$ . Moreover, with the  $y^+$  increasing, the deviation of the energy spectrum also increases. This is consistent with previously discussed instantaneous contours and RMS results. Another interesting thing observed from the energy spectrum results is that in the flow at  $Re_\tau = 180$ , the predicted energy in high wave numbers is higher than the DNS results. On the contrary, the predicted energy in high wave numbers is mostly lower than the DNS results for the channel flow case at  $Re_\tau = 550$ .

With the above discussion, we can observe that the M-GAN can predict the unavailable parameters from the corresponding available data based on flow data used for training. However, the interpolation ability of the model still needs to be further investigated, in other words, whether the model can work in the channel flow case at a Reynolds number between  $Re_\tau = 180$  and 550 or not. Regarding this issue, a channel flow at  $Re_\tau = 395$ , which never joins the training, is used for the interpolation ability test. Same as the previous operation,  $v$  and  $w$  are regarded as available data to predict  $u$  and  $p$  using pretrained M-GAN trained by the data of channel flows at  $Re_\tau = 180$  and 550.

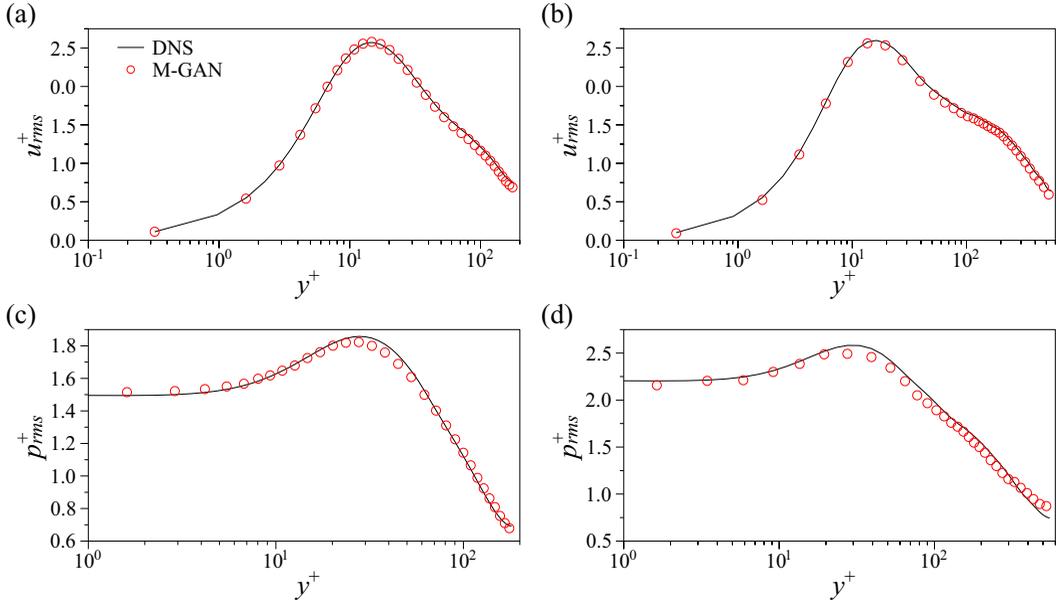


FIG. 9. Root-mean-square profiles of streamwise velocity and pressure fluctuations: (a) and (c) are  $u_{rms}^+$  and  $p_{rms}^+$  of the channel flow at  $Re_\tau = 180$ ; (b) and (d) are  $u_{rms}^+$  and  $p_{rms}^+$  of the channel flow at  $Re_\tau = 550$ .

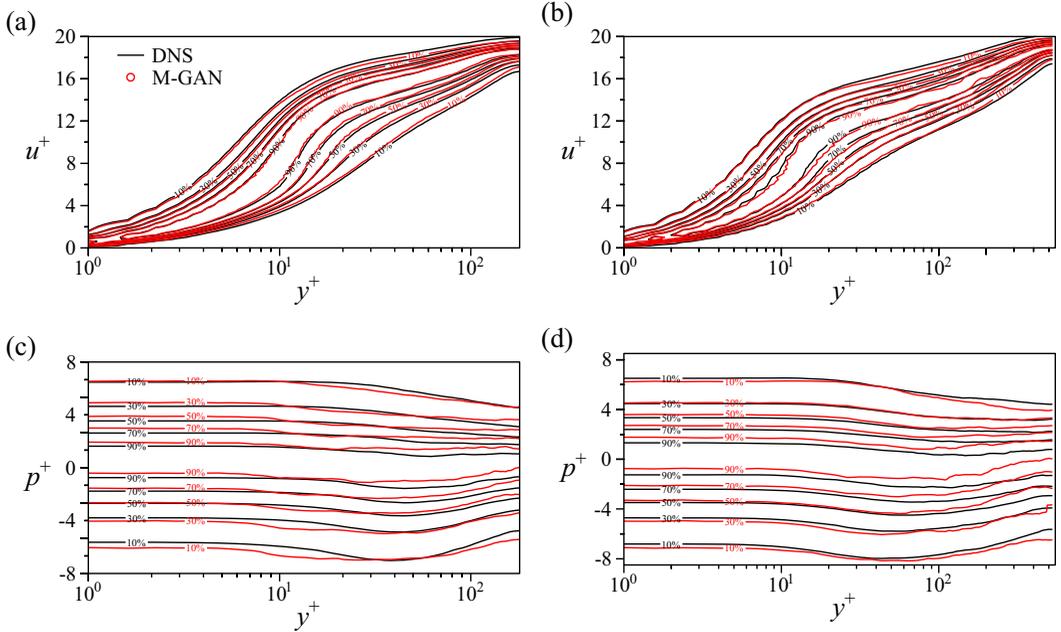


FIG. 10. Probability density functions of the streamwise velocity and pressure fields of channel flows: (a) and (c) are  $PDF(u)$  and  $PDF(p)$  of the channel flow at  $Re_\tau = 180$ ; (b) and (d) are  $PDF(u)$  and  $PDF(p)$  of the channel flow at  $Re_\tau = 550$ . The isoPDF levels are in the range of 10%–90% of the maximum PDF with an increment of 20%.

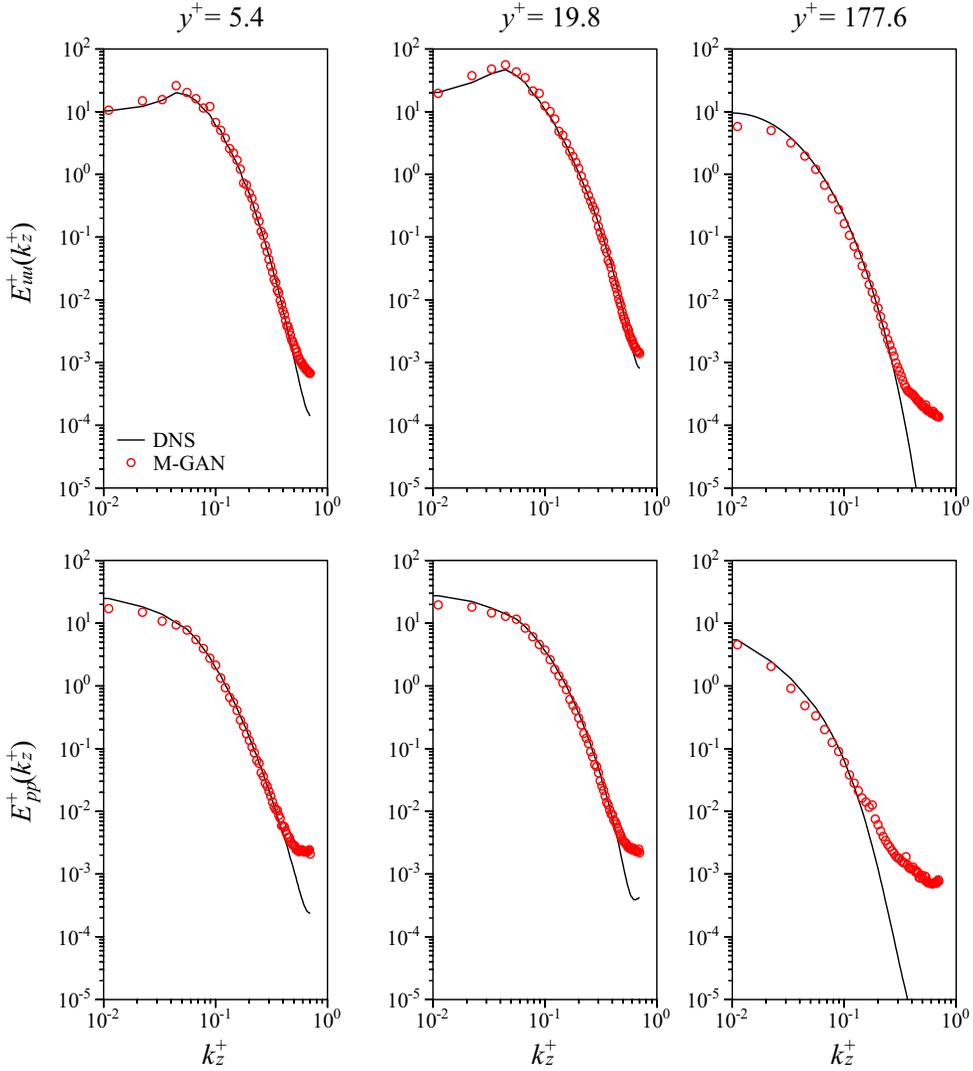


FIG. 11. Spanwise energy spectra of the predicted streamwise velocity and pressure from channel flow at  $Re_\tau = 180$ .

Figure 13 shows the predicted instantaneous streamwise, pressure fields and the absolute error of the channel flow at  $Re_\tau = 395$ . Moreover, Fig. 14 shows the statistical results of RMS velocity and pressure fluctuations. Although there are larger errors than in previous results, especially the  $p$  field cannot be predicted well, the results still are acceptable. Generally, most of the features of flow fields can be constructed and  $u_{\text{rms}}^+$  has a relatively good agreement with the DNS result. This indicates that the proposed M-GAN has a good interpolation ability to work successfully in a case different from the training data set. In other words, M-GAN can learn the mapping function between  $P_a$  and  $P_u$ .

Furthermore, the consistency among DNS  $v$ , DNS  $w$ , and predicted  $u$  of channel flow cases is verified by computing the continuity equation [Eq. (12)] using these parameters. All the computing results of channel flows at various Reynolds numbers approximate 0, indicating the predicted

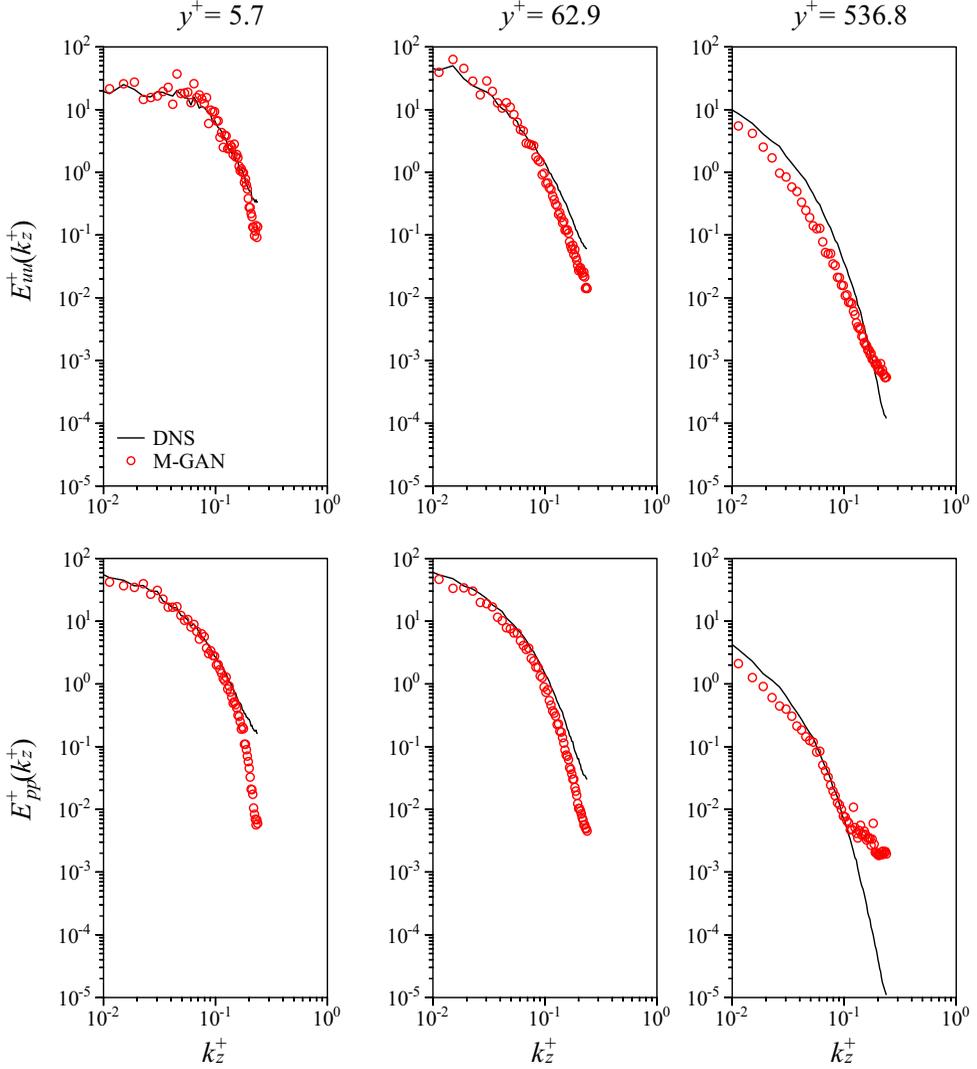


FIG. 12. Spanwise energy spectra of the predicted streamwise velocity and pressure from channel flow at  $\text{Re}_\tau = 550$ .

parameter is physically consistent with the DNS parameters. Ultimately, all the results will be summarized through an error analysis computed using  $L_2$ -norm relative error

$$\nabla \cdot \mathbf{u} = \frac{\partial u_{\text{pred.}}}{\partial x} + \frac{\partial v_{\text{DNS}}}{\partial y} + \frac{\partial w_{\text{DNS}}}{\partial z} \approx 0, \quad (12)$$

$$\varepsilon_2 = \frac{1}{N} \sum_{n=1}^N \frac{\|\alpha_n^{\text{DNS}} - \alpha_n^{\text{M-GAN}}\|_2}{\|\alpha_n^{\text{DNS}}\|_2} \times 100\%, \quad (13)$$

where,  $\alpha_{\text{DNS}}$  and  $\alpha_{\text{M-GAN}}$  represent the DNS data and predicted results from M-GAN. As observed from Fig. 15, the Rayleigh-Bénard flow case has a minimum error of around 3%. The smallest error indicates that  $T$  is easier to predict from  $u$  and  $v$  in the 2D Rayleigh-Bénard flow case because of the less chaotic and more periodic behavior the 2D Rayleigh-Bénard flow has compared to all channel

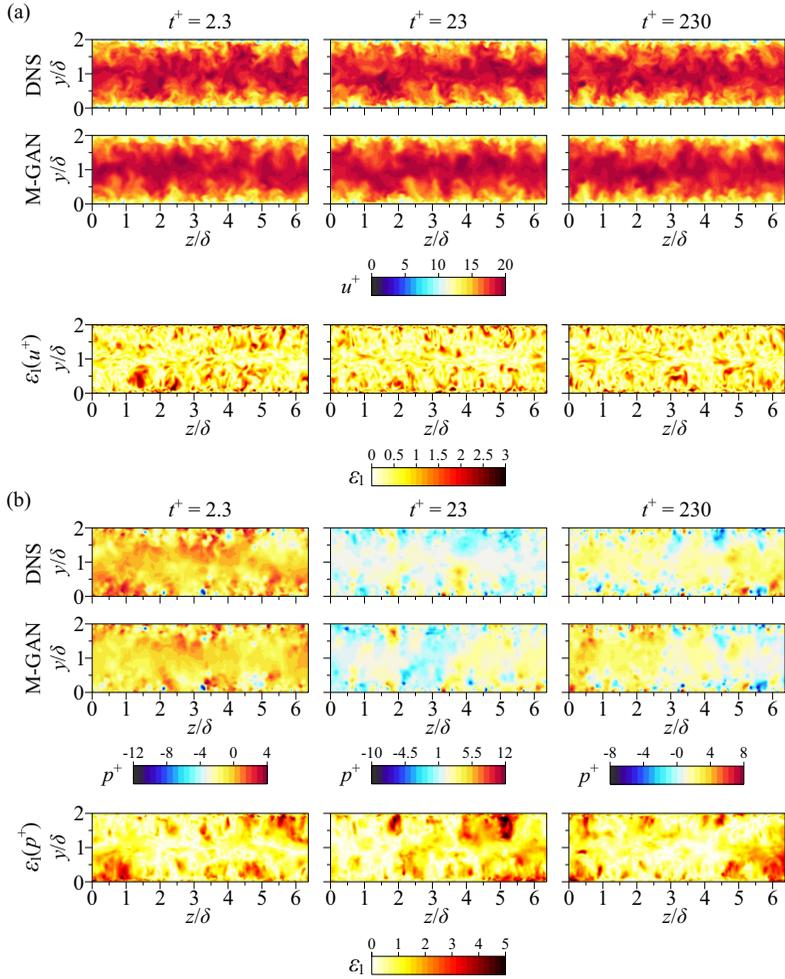


FIG. 13. Predicted instantaneous streamwise (a) velocity and (b) pressure fields of the turbulent channel flow at  $Re_\tau = 395$ .

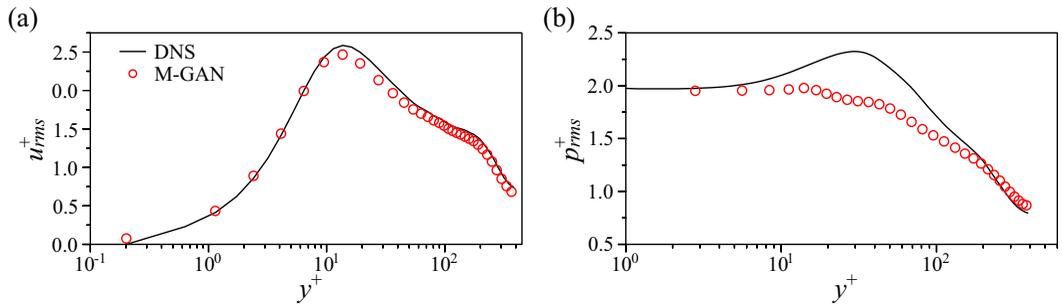


FIG. 14. Root-mean-square profile of the streamwise (a) velocity and (b) pressure fluctuations of the turbulent channel flow at  $Re_\tau = 395$ .

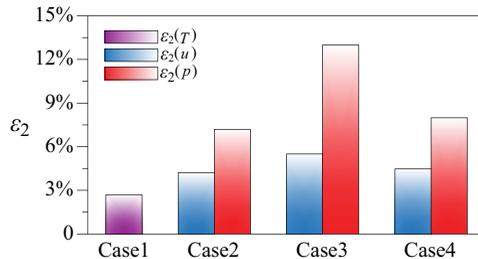


FIG. 15. Relative error of all the testing results, where case 1 is Rayleigh-Bénard flow and cases 2–4 are channel flow at  $Re_\tau = 180, 395,$  and  $550,$  respectively.

flow cases. As expected, the relative error values of both  $u$  and  $p$  for the interpolating channel flow case at  $Re_\tau = 395$  are maximal. Furthermore, the truth that pressure field and higher Reynolds number case are more difficult to reconstruct than streamwise velocity field and lower Reynolds number case is proved by analyzing the Fig. 15.

## V. CONCLUSIONS

We proposed an efficient deep-learning-based model, namely, M-GAN to predict unavailable flow parameters from existing ones. Two cases of classical fluid flows were used to validate the performance and interpolation ability of M-GAN.

First, the 2D Rayleigh-Bénard flow was used as a demonstrating case to show the capability of M-GAN for mapping available parameters  $u$  and  $v$  to unavailable parameters  $T$ . In this case, the results indicated that the M-GAN could indeed predict the temperature field of the 2D Rayleigh-Bénard flow from its velocity field.

Then, two turbulent channel flows at  $Re_\tau = 180$  and  $550$  were used together to train M-GAN and validate the predicting performance of M-GAN. In this case, wall-normal velocity  $v$  and spanwise velocity  $w$  were regarded as available data and passed into M-GAN to predict unavailable parameters that are streamwise velocity  $u$  and pressure  $p$ . Here, LIG was applied to help M-GAN decide which parameter ( $u$  or  $p$ ) should be output when the input layer received the same  $v$  and  $w$ . The prediction's instant contours and turbulence statistics were exhibited and compared to DNS data. The results illustrated that M-GAN successfully predicted  $u$  and  $p$  from the corresponding  $v$  and  $w$  in both turbulent channel flow cases. However, the pressure field and the flow fields of the channel flow at  $Re_\tau = 550$  were relatively more difficult to be predicted because of their more chaotic turbulence characteristics. Finally, turbulent channel flow at  $Re_\tau = 395$ , which was never used to train the M-GAN, was employed to test the interpolation ability of M-GAN. As the results showed, the pretrained M-GAN, which was based on the training data set of channel flows at  $Re_\tau = 180$  and  $550$ , also could predict the unavailable parameters  $u$  and  $p$  from  $v$  and  $w$  from channel flow data at  $Re_\tau = 395$ . Thus, it was proved that M-GAN learned the mapping law from  $v$  and  $w$  to  $u$  and  $p$  of the channel flows at a range of Reynolds numbers ( $Re_\tau = 180$ – $550$ ). In other words, M-GAN showed a good interpolation ability for a specific range of Reynolds numbers. In conclusion, this paper has shown that M-GAN can predict unavailable flow parameters from existing ones, i.e., velocity fields. Besides, with good interpolation ability, M-GAN can work, even for turbulent flows with a range of Reynolds number.

The data that supports the findings of this study are available within this paper.

## ACKNOWLEDGMENTS

This work was supported by Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resource from

the Ministry of Trade, Industry & Energy, Republic of Korea (Grant No. 2021400000140). In addition, this work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (Grant No. 2019R11A3A01058576).

- 
- [1] P. Moin and K. Mahesh, DIRECT NUMERICAL SIMULATION: A tool in turbulence research, *Annu. Rev. Fluid Mech.* **30**, 539 (1998).
  - [2] J. Kim, P. Moin, and R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, *J. Fluid Mech.* **177**, 133 (1987).
  - [3] R. Moser, J. Kim, and N. Mansour, Direct numerical simulation of turbulent channel flow up to  $Re_\tau = 590$ , *Phys. Fluids* **11**, 943 (1999).
  - [4] B. v. Oudheusden, Piv-based pressure measurement, *Meas. Sci. Technol.* **24**, 032001 (2013).
  - [5] C. Vanderwel, A. Stroh, J. Kriegseis, B. Frohnappel, and B. Ganapathisubramani, The instantaneous structure of secondary flows in turbulent boundary layers, *J. Fluid Mech.* **862**, 845 (2019).
  - [6] S. B. Pope, *Turbulent Flows* (Cambridge University Press, Cambridge, 2000).
  - [7] W. Blake, *Mechanics of Flow-Induced Sound and Vibration*, 2nd ed. (Academic Press, New York, 2017), Vol. 2.
  - [8] E. Manoha, B. Troff, and P. Sagaut, Trailing-edge noise prediction using Large-Eddy simulation and acoustic analogy, *AIAA J.* **38**, 575 (2000).
  - [9] A. Eltaweel, M. Wang, D. Kim, F. Thomas, and A. Kozlov, Numerical investigation of tandem-cylinder noise reduction using plasma-based flow control, *J. Fluid Mech.* **756**, 422 (2014).
  - [10] A. Toutant and F. Bataille, Turbulence statistics in a fully developed channel flow submitted to a high temperature gradient, *Int. J. Therm. Sci.* **74**, 104 (2013).
  - [11] X. Zhu, V. Mathai, R. J. A. M. Stevens, R. Verzicco, and D. Lohse, Transition to the ultimate regime in two-dimensional Rayleigh-Bénard convection, *Phys. Rev. Lett.* **120**, 144502 (2018).
  - [12] F. Scarano, Tomographic PIV: Principles and practice, *Meas. Sci. Technol.* **24**, 012001 (2013).
  - [13] S. Allison and G. Gillies, Remote thermometry with thermographic phosphors: Instrumentation and applications, *Rev. Sci. Instrum.* **68**, 2615 (1997).
  - [14] C. Abram, B. Fond, and F. Beyrau, Temperature measurement techniques for gas and liquid flows using thermographic phosphor tracer particles, *Prog. Energy Combust. Sci.* **64**, 93 (2018).
  - [15] C. He, Y. Liu, and L. Gan, Instantaneous pressure determination from unsteady velocity fields using adjacent-based sequential data assimilation, *Phys. Fluids* **32**, 035101 (2020).
  - [16] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. Reyes, M.-L. Shyu, S.-C. Chen, and S. Iyengar, A survey on deep learning: Algorithms, techniques, and applications, *ACM Comput. Surv.* **51**, 92:1 (2018).
  - [17] M. Morris, A. Rajesh, M. Asaad, A. Hassan, R. Saadoun, and C. Butler, Deep learning applications in surgery: Current uses and future directions, *Am. Surg.* **89**, 36 (2023).
  - [18] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* **52**, 477 (2020).
  - [19] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, and H. Li, An interpretable framework of data-driven turbulence modeling using deep neural networks, *Phys. Fluids* **33**, 055133 (2021).
  - [20] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, *Annu. Rev. Fluid Mech.* **51**, 357 (2019).
  - [21] R. Vinuesa and S. Brunton, Enhancing computational fluid dynamics with machine learning, *Nature Comput. Sci.* **2**, 358 (2022).
  - [22] M. Z. Yousif and H.-C. Lim, Reduced-order modeling for turbulent wake of a finite wall-mounted square cylinder based on artificial neural network, *Phys. Fluids* **34**, 015116 (2022).
  - [23] H. Eivazi, S. Le Clainche, S. Hoyas, and R. Vinuesa, Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows, *Expert Syst. Appl.* **202**, 117038 (2022).
  - [24] M. Z. Yousif, L. Yu, and H.-C. Lim, Physics-guided deep learning for generating turbulent inflow conditions, *J. Fluid Mech.* **936**, A21 (2022).

- [25] M. Z. Yousif, M. Zhang, L. Yu, R. Vinuesa, and H. Lim, A transformer-based synthetic-inflow generator for spatially-developing turbulent boundary layers, *J. Fluid Mech.* **957**, A6 (2023).
- [26] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata, Synthetic turbulent inflow generator using machine learning, *Phys. Rev. Fluids* **4**, 064603 (2019).
- [27] K. Fukami, K. Fukagata, and K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, *J. Fluid Mech.* **870**, 106 (2019).
- [28] L. Yu, M. Yousif, M. Zhang, S. Hoyas, R. Vinuesa, and H.-C. Lim, Three-dimensional esrgan for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning, *Phys. Fluids* **34**, 125126 (2022).
- [29] M. Z. Yousif, L. Yu, and H.-C. Lim, Super-resolution reconstruction of turbulent flow fields at various Reynolds numbers based on generative adversarial networks, *Phys. Fluids* **34**, 015130 (2022).
- [30] M. Z. Yousif, L. Yu, and H.-C. Lim, High-fidelity reconstruction of turbulent flow from spatially limited data using enhanced super-resolution generative adversarial network, *Phys. Fluids* **33**, 125119 (2021).
- [31] L. Guastoni, A. Güemes, A. Ianiro, S. Discetti, P. Schlatter, H. Azizpour, and R. Vinuesa, Convolutional-network models to predict wall-bounded turbulence from wall quantities, *J. Fluid Mech.* **928**, A27 (2021).
- [32] M. A. Jouybari, J. Yuan, G. Breerton, and M. Murillo, Data-driven prediction of the equivalent sand-grain height in rough-wall turbulent flows, *J. Fluid Mech.* **912**, A8 (2021).
- [33] S. Lee, J. Yang, P. Forooghi, A. Stroh, and S. Bagheri, Predicting drag on rough surfaces by transfer learning of empirical correlations, *J. Fluid Mech.* **933**, A18 (2022).
- [34] J. Kim and C. Lee, Prediction of turbulent heat transfer using convolutional neural networks, *J. Fluid Mech.* **882**, A18 (2020).
- [35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks, *Commun. ACM* **63**, 139 (2020).
- [36] P. Shamsolmoali, M. Zareapoor, E. Granger, H. Zhou, R. Wang, M. Celebi, and J. Yang, Image synthesis with adversarial networks: A comprehensive survey and case studies, *Inf. Fusion* **72**, 126 (2021).
- [37] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA (IEEE, 2017)*, pp. 105–114.
- [38] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, *ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks* (Springer, Berlin, 2019).
- [39] Z. Deng, C. He, Y. Liu, and K. Kim, Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework, *Phys. Fluids* **31**, 125111 (2019).
- [40] M. Yousif, L. Yu, S. Hoyas, R. Vinuesa, and H.-C. Lim, A deep-learning approach for reconstructing 3D turbulent flows from 2D observation data, *Sci. Rep.* **13**, 2529 (2023).
- [41] A. Güemes, S. Discetti, A. Ianiro, B. Sirmacek, and H. Azizpour, From coarse wall measurements to turbulent velocity fields through deep learning, *Phys. Fluids* **33**, 075121 (2021).
- [42] R. Verzicco and P. Orlandi, A finite-difference scheme for three-dimensional incompressible flows in cylindrical coordinates, *J. Comput. Phys.* **123**, 402 (1996).
- [43] X. Zhu, E. Phillips, V. Spandan, J. Donners, G. Ruetsch, J. Romero, R. Ostilla-Mónico, Y. Yang, D. Lohse, R. Verzicco, M. Fatica, and R. Stevens, AFiD-GPU: A versatile Navier–Stokes solver for wall-bounded turbulent flows on GPU clusters, *Comput. Phys. Commun.* **229**, 199 (2018).
- [44] A. Yaglom, Similarity laws for constant-pressure and pressure-gradient turbulent wall flows, *Annu. Rev. Fluid Mech.* **11**, 505 (1979).
- [45] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, Residual dense network for image restoration, *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 2480 (2021).
- [46] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
- [47] <https://fluids.pusan.ac.kr/fluids/65416/subview.do>
- [48] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).