Physics of Fluids

Three-dimensional ESRGAN for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning

Cite as: Phys. Fluids **34**, 125126 (2022); https://doi.org/10.1063/5.0129203
Submitted: 03 October 2022 • Accepted: 18 November 2022 • Accepted Manuscript Online: 23 November 2022 • Published Online: 12 December 2022

Linqi Yu (俞霖奇), Mustafa Z. Yousif, Meng Zhang (张梦), et al.

Physics of Fluids ARTICLE scitation.org/journal/phf

Three-dimensional ESRGAN for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning

Cite as: Phys. Fluids **34**, 125126 (2022); doi: 10.1063/5.0129203
Submitted: 3 October 2022 · Accepted: 18 November 2022 ·
Published Online: 12 December 2022

Linqi Yu (俞霖奇),[1] Mustafa Z. Yousif,[1] Meng Zhang (张梦),[1] Sergio Hoyas,[2] Ricardo Vinuesa,[3] and Hee-Chang Lim (임희창)[1,a)]

AFFILIATIONS

[1]School of Mechanical Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan 46241, Republic of Korea
[2]Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, València 46022, Spain
[3]FLOW, Engineering Mechanics, KTH Royal Institute of Technology, Stockholm 10044, Sweden

[a)]**Author to whom correspondence should be addressed:** hclim@pusan.ac.kr

ABSTRACT

Turbulence is a complicated phenomenon because of its chaotic behavior with multiple spatiotemporal scales. Turbulence also has irregularity and diffusivity, making predicting and reconstructing turbulence more challenging. This study proposes a deep-learning approach to reconstruct three-dimensional (3D) high-resolution turbulent flows from spatially limited data using a 3D enhanced super-resolution generative adversarial networks (3D-ESRGAN). In addition, a novel transfer-learning method based on tricubic interpolation is employed. Turbulent channel flow data at friction Reynolds numbers $Re_\tau = 180$ and $Re_\tau = 500$ were generated by direct numerical simulation (DNS) and used to estimate the performance of the deep-learning model as well as that of tricubic interpolation-based transfer learning. The results, including instantaneous velocity fields and turbulence statistics, show that the reconstructed high-resolution data agree well with the reference DNS data. The findings also indicate that the proposed 3D-ESRGAN can reconstruct 3D high-resolution turbulent flows even with limited training data.

I. INTRODUCTION

High-resolution flow data are required to visualize and analyze turbulent flows in experimental and computational fluid dynamics (CFD). With the rapid development of various methods, high-fidelity turbulent flow data have been produced more effectively than before. In the field of CFD, direct numerical simulation (DNS) has been used to accurately simulate various cases of turbulent flows at a specific range of Reynolds numbers, where the Navier–Stokes equations are solved numerically.[1–3] As for experimental measurements, some practical measuring techniques, such as particle-image velocimetry (PIV),[4] tomographic PIV,[5] and the four-dimensional time particle-tracking velocimetry (PTV),[6,7] have been successfully employed to visualize and quantify the intricate instantaneous structure of turbulence, producing multidimensional turbulent flow data spatiotemporally. However, in turbulent flows, the flow velocity varies significantly and irregularly in both position and time.[8] Furthermore, turbulent flows have multiple spatiotemporal flow scales. Thus, a large prior investment in experimental and computational equipment is a prerequisite for generating high-resolution turbulent-flow data using state-of-the-art methods such as DNS and PIV.

Deep learning (DL), a subfield of machine learning in artificial intelligence, is extraordinarily focused on by researchers from various fields. Because of its ability to solve highly nonlinear problems using deep neural networks (DNNs), DL has been implemented in various fields, including computer vision, language processing, automatic driving, and medical diagnosis.[9–11] In addition, with the rapid development of graphics-processing units (GPUs), the training and prediction processes of DL algorithms have been significantly accelerated, benefiting from the parallel-computing capabilities in GPUs.[12] In other words, the GPU guarantees efficiency when DL is applied to complex problems.

Regarding fluid dynamics, with the highly nonlinear mapping ability of DNNs, various DL-based methods exhibit a considerable

potential to deal with problems regarding turbulent flows,[13,14] such as generating the temporal data of turbulent flows based on turbulence modeling,[15,16] simulating fluid flows,[17] reduced-order modeling,[18–20] prediction of turbulent flows based on information from flow fields at previous time steps,[21–26] turbulent-flow control for drag reduction,[27,28] and reconstruction of turbulent flows.[29–31,63] Another important application that needs to be focused on is the high-resolution reconstruction of turbulent flows using DNNs. DL-based approaches have outperformed conventional reconstruction methods in this field, such as the bicubic interpolation method.[32,33] Fukami *et al.*[34] used models based on convolutional neural networks (CNNs) and multiscale CNNs with skip connections to map the coarse flow fields to super-resolution fields. Their results showed that the models could remarkably improve the spatial resolution of extremely low-resolution flow data in both flow regimes (laminar and turbulent). Liu *et al.*[35] proposed a CNN-based model with multiple temporal paths to reconstruct high-resolution turbulent flow data. Their results revealed that this method has a better reconstruction effect than static CNN-based models. Kim *et al.*[36] used a cycle-consistent generative adversarial network (CycleGAN) to reconstruct turbulent flow from low-resolution DNS and large-eddy simulation data, where they used unpaired low- and high-resolution data in the training process. Their results illustrated that CycleGAN outperforms the bicubic interpolation method and CNN-based models.

Furthermore, another generative adversarial networks (GANs)-based model named enhanced super-resolution generative adversarial network (ESRGAN)[37] has shown a greater capability to reconstruct high-resolution data than other models in the fields of computer vision and fluid dynamics, which benefits from the help of deeper layers and special loss functions like the perceptual loss. Deng *et al.*[38] compared the high-resolution turbulent-flow reconstruction from a super-resolution GAN (SRGAN) and ESRGAN using PIV measurement data. Their reconstruction results agreed well with the actual measurement data, and ESRGAN exhibited a better reconstruction ability than SRGAN.

Yousif *et al.*[39] recently used an ESRGAN-based model with a multiscale convolution part (MS-ESRGAN) to reconstruct high-fidelity turbulent flows from spatially limited data. Meanwhile, they also employed this DL model to reconstruct high-resolution data based on turbulent flows at various Reynolds numbers.[40] In addition to its remarkable ability to reconstruct high-resolution turbulent flows from low-resolution data, they found that MS-ESRGAN could reconstruct turbulent flows at the specific range of Reynolds numbers, even if the flow data at these Reynolds numbers were not included in the training process.

In summary, the above-mentioned previous literature shows that the GANs-based models have a powerful ability to reconstruct the resolution of turbulent flows. However, all the turbulent flow cases mentioned in the above studies are based on two-dimensional (2D) data. Compared to 2D data, three-dimensional (3D) turbulent flows are more complicated and practical. On the other hand, the general DL-based models are designed for 2D data like images, which is unsuitable for processing 3D turbulent flow data. In this study, we aim to use a specific DL-based model to reconstruct 3D high-resolution turbulent-flow data from low-resolution data. This DL-based model is a 3D form of the original MS-ESRGAN, termed 3D-ESRGAN. This method is particularly used in reconstructing high-resolution flow fields from spatially limited 3D experimental data. In addition, it is expected to improve the detail of the 3D flow obtained from CFD, where the computational mesh is very coarse. Moreover, inspired by the idea of transfer learning (TF) employed to tackle turbulence problems,[19,41–43] we propose a transfer-learning method for the DL-based model based on tricubic interpolation. Generally, the traditional transfer-learning method can save some training times but still need huge paired training data to conduct pre-training. Regarding this issue, we propose tricubic interpolation-based transfer learning. We aim to apply tricubic interpolation to prepare the pre-training data for transfer learning so that the training process can be more efficient. To evaluate the performance of 3D-ESRGAN and the effect of the proposed transfer-learning method, we have used 3D turbulent channel-flow data at two different Reynolds friction numbers generated by performing DNS with their low-resolution data. The results have also been plotted to show the performance of the DL model and the transfer-learning method based on the channel flow's instantaneous velocity fields and physical statistics.

The remainder of this paper is organized as follows: Sec. II introduces how to generate turbulent flow data through DNS and the method for obtaining the low-resolution data of turbulent flows. The methodology of super-resolution reconstruction using 3D-ESRGAN and tricubic interpolation-based transfer learning is explained in Sec. III. In Sec. IV, the testing results of the model are discussed. Finally, the conclusions of this study are summarized in Sec. V.

II. GENERATION OF TURBULENT-FLOW DATA

In this study, we use fully developed incompressible turbulent channel flow data at $Re_\tau = 180$ and 500 generated by performing DNS as a test case. The governing equations are the incompressible continuity and momentum equations expressed as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re_\tau} \nabla^2 \mathbf{u}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where $\mathbf{u}$, t, and p represent velocity, time, and pressure, respectively. The variables in both equations are non-dimensionalized using the channel half width δ, friction velocity u_τ calculated as $u_\tau = \sqrt{\tau_w/\rho}$, where τ_w is the wall-shear stress, and ρ is the fluid density. The friction Reynolds number is calculated as $Re_\tau = u_\tau \delta/\nu$, where ν represents the fluid kinematic viscosity. The LISO code[44] is used to perform the DNS. This code has been developed to solve the incompressible Navier–Stokes equations, similar to the one used by Lluesma-Rodríguez *et al.*[45] This code has been successfully used to simulate wall-bounded turbulent flows.[46–48] The code uses the same strategy as that described by Kim *et al.*[2] but employs a seven-point compact-finite-difference scheme in the wall-normal direction y direction with fourth-order consistency and extended spectral-like resolution.[49] The temporal discretization is a third-order semi-implicit Runge–Kutta scheme.[50] The wall-normal grid spacing is set to $\triangle y \approx 1.5\eta$, which is adjusted so that the grid is constant in terms of the local Kolmogorov scale $\eta = (\nu^3/\varepsilon)^{0.25}$, where ε represents the isotropic dissipation of turbulent kinetic energy.

The dimensions of the computational domain for flows at the two Reynolds numbers are set to $8\pi\delta$, 2δ, and $3\pi\delta$ in the streamwise (x), wall-normal (y), and spanwise (z) directions, respectively.

TABLE I. Some salient parameters for the DNS cases, where N represents the number of grid points in the streamwise (x), wall-normal (y), and spanwise (z) directions. The superscript "+" means that the quantity is dimensionalized using u_τ and ν. Furthermore, Δy_w^+ and Δy_c^+ denote the distance of the first mesh point away from the wall and the maximum spacing at the centerline of the channel, respectively.

Re_τ	$N_x \times N_y \times N_z$	Δx^+	Δz^+	Δy_w^+	Δy_c^+	Δt^+
180	$576 \times 432 \times 201$	8.55	4.27	0.53	2.53	0.07
550	$1536 \times 1152 \times 251$	8.33	4.16	0.74	5.45	0.09

The other parameters for the simulations are summarized in Table I. The streamwise and spanwise grid points are uniformly distributed, and a non-uniform mesh is used in the wall-normal direction. Periodic boundary conditions are imposed in the x and z directions. In addition, the no-slip condition is applied to the top and bottom walls of the channel. Furthermore, 1000 3D snapshots are separately collected at two different Reynolds numbers.

To reduce the computational cost of the training process and increase the training and testing data, each 3D flow snapshot is split into 12 parts (a total of 12 000 snapshots) with sizes of $2\pi\delta$, 2δ, and $\pi\delta$. The grid size of each part is reduced from the original one to $64 \times 48 \times 48$ by interpolation, which is more suitable for training the DL model. For the flow at the two Reynolds numbers, 80% and 20% of the data are used to train the DL model and evaluate its performance. We generate low-resolution data artificially by selecting discrete points in the flow domain, as shown in Fig. 1. The points are selected uniformly with an interval of four grid points along the x and z directions, and uniformly along the y direction (more points are selected near the walls). After selecting these points, the grid size of low-resolution data is reduced from $64 \times 48 \times 48$ to $16 \times 12 \times 12$. To improve the training process of the DL model, all data are normalized using the min-max normalization function to scale the values between 0 and 1.

III. METHODOLOGY

A. 3D-ESRGAN

Since the development of the GAN architecture by Goodfellow *et al.*,[51] various GAN-based models have been developed to solve image-based problems such as synthetic image generation and resolution improvement problems.[37,52,53] Compared with the traditional GAN, the ESRGAN has shown an excellent capability to reconstruct high-resolution turbulent flows from low-resolution data.[38–40] A GAN typically includes two adversarial networks: generator (G) and discriminator (D). The working process of G and D is called a two-player minimax game with a value function that can be expressed as

$$\min_G \max_D V(D,G) = \mathbb{E}_{\chi_r \sim P_{data}(\chi_r)}[\log D(\chi_r)] + \mathbb{E}_{\xi \sim P_\xi(\xi)}[\log(1 - D(G(\xi)))], \quad (3)$$

where χ_r represents the real image, and ξ represents the random noise input into G to generate artificial images $G(\xi)$. Here, $P_{data}(\chi_r)$ and $P_\xi(\xi)$ represent the distributions of the real image data and the input noise variables, respectively, where χ_r denotes the real image data, and ξ represents a random vector. Furthermore, $D(\chi_r)$ represents the probability (from 0 to 1) that an image is real rather than a generated artificial image. Inversely, $D(G(\xi))$ represents the probability of whether $G(\xi)$ comes from real images or not. We attempt to make G generate images more similar to the real data in the training process. Meanwhile, we also train D, enabling D to distinguish real images from generated artificial images. In this case, the value of $D(\chi_r)$ will increase (tends to be 1), and the value of $D(G(\xi))$ will decrease (tends to be 0) so that the value of $V(D, G)$ would increase.

The present study develops a 3D-ESRGAN based on two-dimensional MS-ESRGAN[39] to reconstruct 3D turbulent flow fields from coarse data. Figure 2 depicts the architecture of the 3D-ESRGAN. As illustrated in Fig. 2(a), G mainly includes a deep CNN called residual in residual dense blocks (RRDBs),[54] where several dense blocks are used, consisting of convolutional and leaky ReLU (LReLU) activation-function layers with skip connection. In addition, the structure of 3D-ESRGAN comprises a multiscale part (MSP), which maps features from RRDBs to more complex features through a series of convolutional layers with filters of various sizes.[55] More details about the MSP are summarized in Table II. Figure 2(b) shows the architecture of D. The main part of D is represented by CBLs, consisting of three layers: convolutional, batch normalization, and LReLU activation-function layers.

Regarding the calculation process, first, low-resolution data are introduced to G and passed through a convolutional layer, RRDBs, and MSP. Then, the output from MSP is passed through a convolutional layer to generate the reconstructed high-resolution data called artificial data (x_a), whereas the real data (x_r) are the original data from DNS, as mentioned in Sec. II. After generating x_a, both real and artificial data are fed into D and passed through the corresponding layers of D to output the non-transformed discriminator output: $C(x_r)$ and $C(x_a)$. Based on $C(x_r)$ and $C(x_a)$, we can calculate the relativistic average discriminator value D_{Ra}, which is formulated as

$$D_{Ra}(x_r, x_a) = \sigma(C(x_r)) - \mathbb{E}_{x_a}[C(x_a)], \quad (4)$$

$$D_{Ra}(x_a, x_r) = \sigma(C(x_a)) - \mathbb{E}_{x_r}[C(x_r)], \quad (5)$$

where σ represents the sigmoid function. Unlike the general D in the standard GAN, which estimates the probability that input data are real or artificial, the D in ESRGAN attempts to predict the probability that real data are relatively more realistic than artificial data.[37,56] The discriminator loss is then defined as

$$L_D^{Ra} = -\mathbb{E}_{x_r}[\log(D_{Ra}(x_r, x_a))] - \mathbb{E}_{x_a}[\log(1 - D_{Ra}(x_a, x_r))]. \quad (6)$$

The adversarial loss of the generator can be expressed in a symmetrical form as

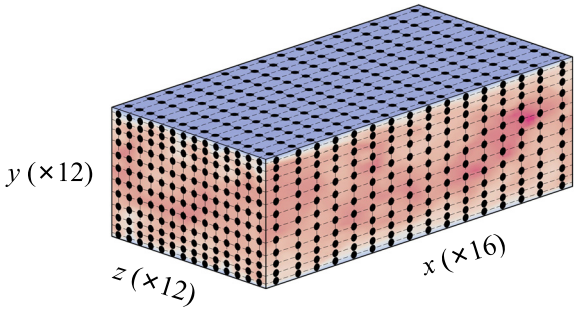

FIG. 1. Distribution of selected points in the 3D flow domain.

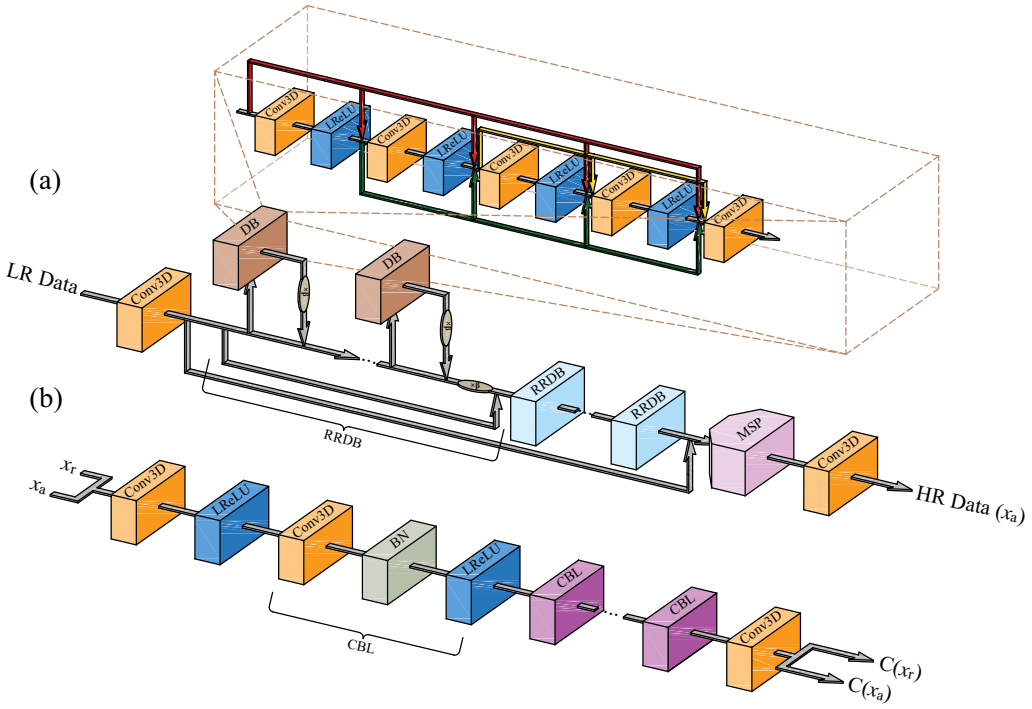

FIG. 2. The architecture of 3D-ESRGAN: (a) the architecture of the generator, where β represents the residual scaling parameter set to 0.2, and (b) the architecture of the discriminator.

$$L_G^{Ra} = -\mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_a))] - \mathbb{E}_{x_a}[\log(D_{Ra}(x_a, x_r))]. \quad (7)$$

Note that D is trained to predict the probability that the real data are relatively more realistic than the artificial data during the training process. In this case, $D_{Ra}(x_r, x_a)$ tends to be 1, and $D_{Ra}(x_a, x_r)$ tends to be 0, which decreases L_D^{Ra} to 0. On the other hand, G is trained to generate more realistic artificial data, based on which real data are less realistic than artificial data. Then, $D_{Ra}(x_r, x_a)$ starts to decrease from 1 to 0, and $D_{Ra}(x_a, x_r)$ begins to increase from 0 to 1, which decreases L_G^{Ra} to 0. In this adversarial process, D and G compete with each other even though they also improve each other. In the end, G will generate the constructed high-resolution data, which are very similar to the real DNS data.

TABLE II. MSP architecture.

First branch	Second branch	Third branch
Conv3D.(3, 3)	Conv3D.(5, 5)	Conv3D.(7, 7)
UpSampling(2, 2)	UpSampling(2, 2)	UpSampling(2, 2)
Conv3D.(3, 3)	Conv3D.(5, 5)	Conv3D.(7, 7)
LeakyReLU	LeakyReLU	LeakyReLU
UpSampling(2, 2)	UpSampling(2, 2)	UpSampling(2, 2)
Conv3D.(3, 3)	Conv3D.(5, 5)	Conv3D.(7, 7)
LeakyReLU	LeakyReLU	LeakyReLU
UpSampling(2, 2)	UpSampling(2, 2)	UpSampling(2, 2)
Conv3D.(3, 3)	Conv3D.(5, 5)	Conv3D.(7, 7)
LeakyReLU	LeakyReLU	LeakyReLU
Add (first branch, second branch, third branch)		

In addition to the adversarial loss, the loss function of G ($\mathscr{L}_G$) includes four additional loss terms: voxel loss (L_{voxel}), perceptual loss ($L_{\text{perceptual}}$), continuity loss ($L_{\text{continuity}}$), and momentum loss (L_{momentum}). L_{voxel} represents the error calculated by comparing the voxel (volume pixel) difference between the reconstructed high-resolution data and real DNS data. $L_{\text{perceptual}}$ represents the error in features extracted from the real and artificial data by the feature extractor (FE). This study develops a 3D FE to extract features from 3D data using a pre-trained 3D convolutional auto-encoder. As shown in Fig. 3, data are passed through the encoder part. Then, various features

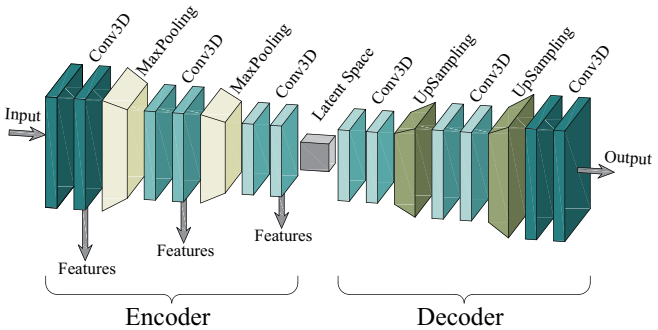

FIG. 3. Schematic of the feature extractor based on 3D convolutional auto-encoder.

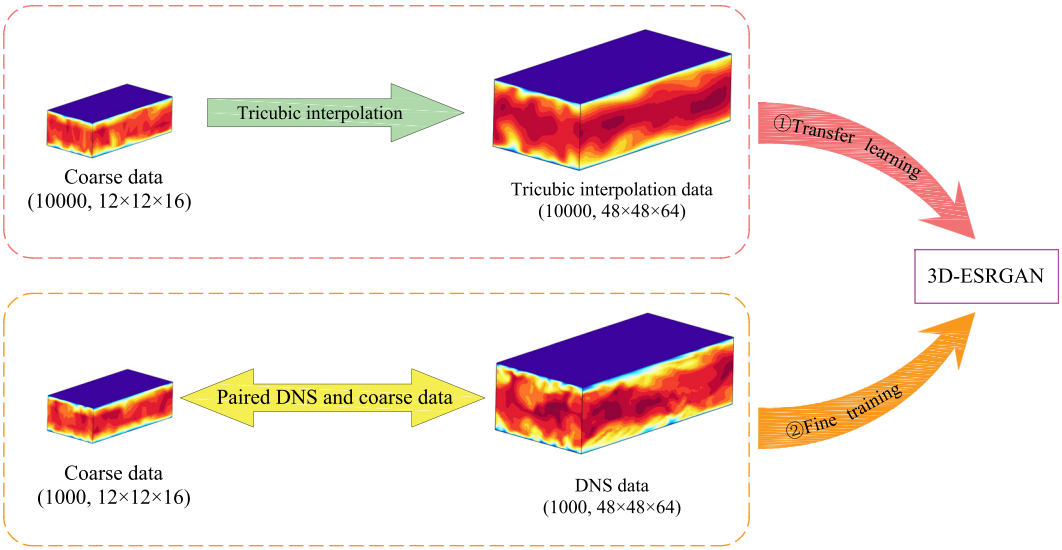

FIG. 4. Overview of tricubic interpolation-based transfer learning process.

are extracted and output from several convolutional layers. $L_{\text{continuity}}$ and L_{momentum} indicate errors calculated using the continuity and momentum equations. The above additional loss terms are computed using the mean-squared error

$$L_{\text{voxel}} = \frac{1}{N}\sum_{n=1}^{N}(x_r - x_a)_n^2, \tag{8}$$

$$L_{\text{perceptual}} = \frac{1}{N}\sum_{n=1}^{N}(\mathscr{F}_{FE}(x_r) - \mathscr{F}_{FE}(x_a))_n^2, \tag{9}$$

$$L_{\text{continuity}} = \frac{1}{N}\sum_{n=1}^{N}(\nabla\cdot\mathbf{u}_r - \nabla\cdot\mathbf{u}_a)_n^2, \tag{10}$$

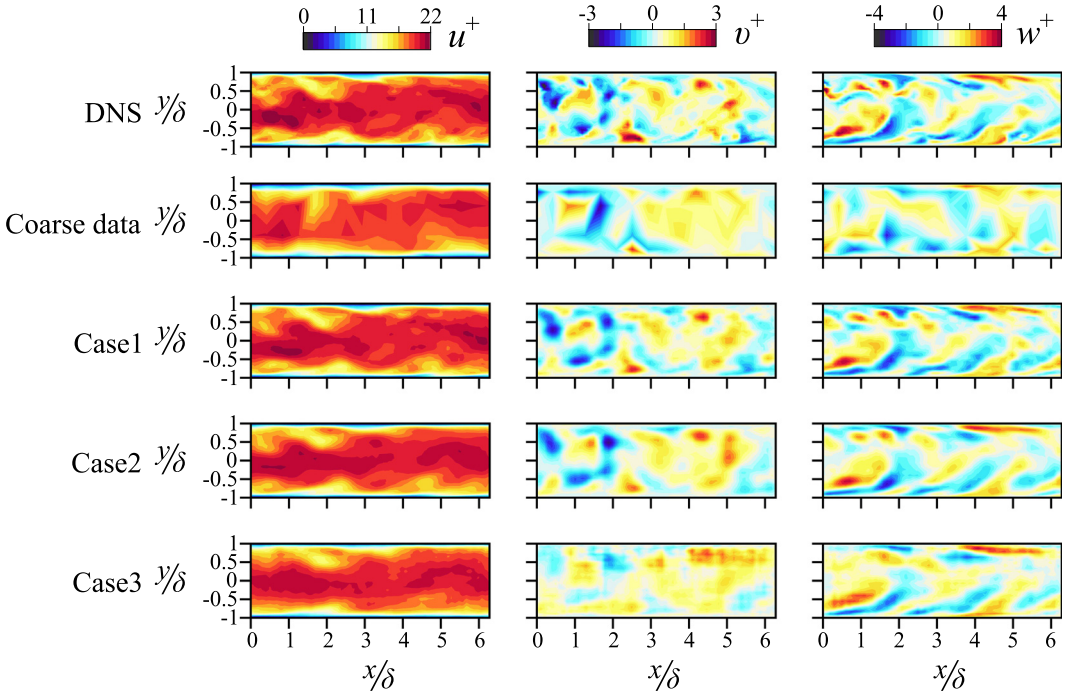

FIG. 5. Reconstructed instantaneous velocity fields of channel flow at $Re_\tau = 180$ (x–y plane at $z = 0.5\pi\delta$).

$$L_{\text{momentum}} = \frac{1}{N}\sum_{n=1}^{N}\left(\left(\frac{\partial \mathbf{u}_r}{\partial t} + \mathbf{u}_r \cdot \nabla \mathbf{u}_r\right) - \left(\frac{\partial \mathbf{u}_a}{\partial t} + \mathbf{u}_a \cdot \nabla \mathbf{u}_a\right)\right)_n^2, \tag{11}$$

where N represents the number of the paired training samples in a training mini-batch, set to 8 in this study, and $\mathscr{F}_{FE}$ of Eq. (9) is a mapping function of the FE. Finally, $\mathscr{L}_G$ is defined as

$$\mathscr{L}_G = \beta_1 L_G^{Ra} + \beta_2 L_{\text{voxel}} + \beta_3 L_{\text{perceptual}} + L_{\text{continuity}} + L_{\text{momentum}}, \tag{12}$$

where β_1, β_2, and β_3 denote the coefficients used to balance the loss terms, the values of which are set to 10, 1000, and 2000, respectively.

The training process of 3D-ESRGAN includes the following steps. First, the batched low-resolution data are inputted into G to output x_a, where the batch size is set to 8 in this study. Then, x_a and x_r are calculated to get a generator loss based on Eqs. (7)–(12). Second, the x_a and x_r pass through the D and are calculated to get a discriminator loss based on Eqs. (4)–(6). After obtaining losses, the optimization algorithm will update the model weights attempting to decrease losses during training, where the optimization algorithm used in this study is adaptive moment estimation (Adam).[57] In addition, this study uses the open-source library TensorFlow 2.3.0 to implement the deep-learning model. The customized sample Python code for the proposed 3D-ESRGAN is available on the web page (https://fluids.pusan.ac.kr/fluids/65416/subview.do).

B. Tricubic interpolation-based transfer learning

The capability, generalization, and computational cost of the DL model are greatly affected by the amount of training data. Developing a DL model with high capability, great generalization, and less computational cost is a hot pursuit of researchers. Transfer learning (TF)[41]

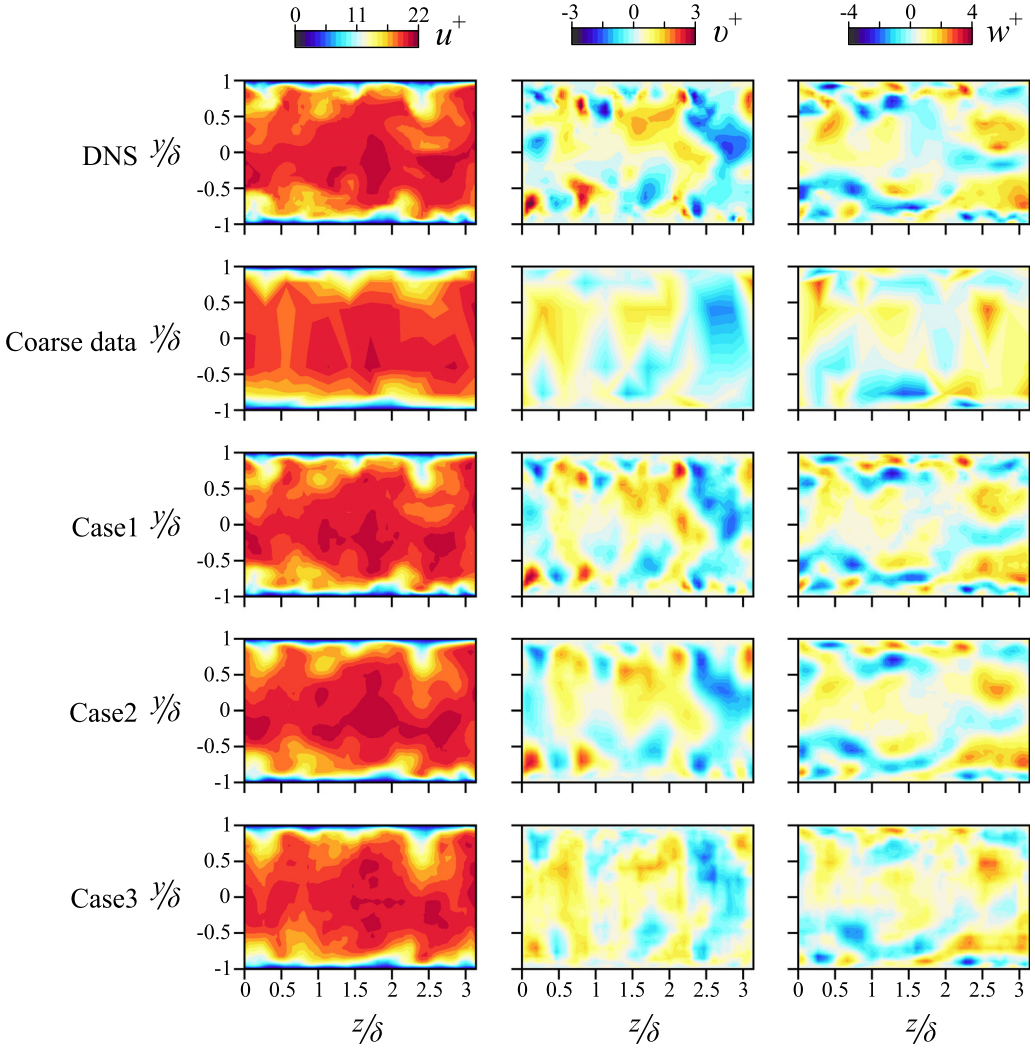

FIG. 6. Reconstructed instantaneous velocity fields of channel flow at $Re_\tau = 180$ (y–z plane at $x = \pi\delta$).

has been widely applied in turbulent-flow research.[19,42,43] TF also shows a good advantage when successful training results are obtained but with fewer data and less computational time. The basic idea of TF could be understood as enabling a new DL model to learn some knowledge transferred from some pre-trained models in terms of their weights. On the other hand, regular DL training starts with the initialization of random values of weights.

This study employs two methods to train and test the 3D-ESRGAN. The first one is developed classically. As mentioned in Sec. II, 9600 snapshots of paired low-resolution and DNS data are used to train the DL model. Then, we use 2400 extra snapshots of low-resolution flow data to evaluate the model performance. Another method based on tricubic interpolation and TF is used when paired DNS training data are insufficient. For instance, only 1000 snapshots of paired DNS training data are available, whereas 10,000 snapshots of low-resolution data are without any corresponding high-resolution DNS data. The second method is denoted tricubic interpolation-based transfer learning.

Tricubic,[58] also called tricubic interpolation, is a numerical approximation for the mathematical problem of obtaining values at arbitrary points in a 3D space of a function defined on a regular grid, which is expressed as follows:

$$f(x,y,z) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} a_{ijk} x^i y^j z^k. \tag{13}$$

This formulation has 64 coefficients a_{ijk}, which require the function to have a given value or directional derivative at some points, such as the peaks around a cube with $3 \times 3 \times 3$ grids, so that a single linear constraint can constrain the 64 coefficients.

In terms of the principle, bicubic interpolation is widely applied to image-resolution enhancement.[59,60] Tricubic interpolation can also reconstruct the resolution of 3D data, such as magnetic-resonance imaging data.[61] The reconstruction obtained using tricubic is less accurate than that obtained using DL-based models. Nevertheless, as previously mentioned, tricubic interpolation is a purely mathematical method without any data-based training work. Thus, we can directly use tricubic to reconstruct the 10 000 snapshots of low-resolution flow data in our work and obtain tricubic interpolation-based high-resolution data.

As Fig. 4 shows, in the transfer-learning process, the 3D-ESRGAN model is trained with a large dataset consisting of 10 000 snapshots of low-resolution data and the corresponding tricubic high-resolution data obtained by tricubic interpolation. After the pre-training process, the model can gain a rough knowledge

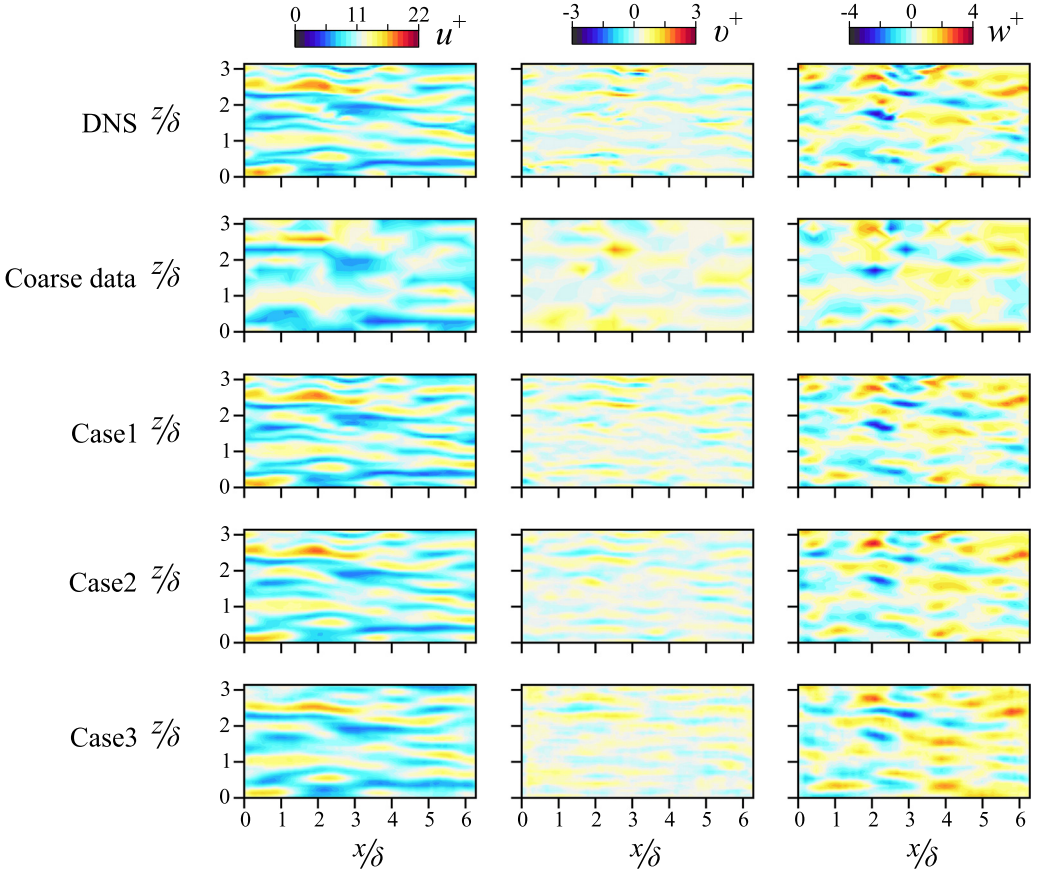

FIG. 7. Reconstructed instantaneous velocity fields of channel flow at $Re_\tau = 180$ (x–z plane at $y^+ = 16.78$).

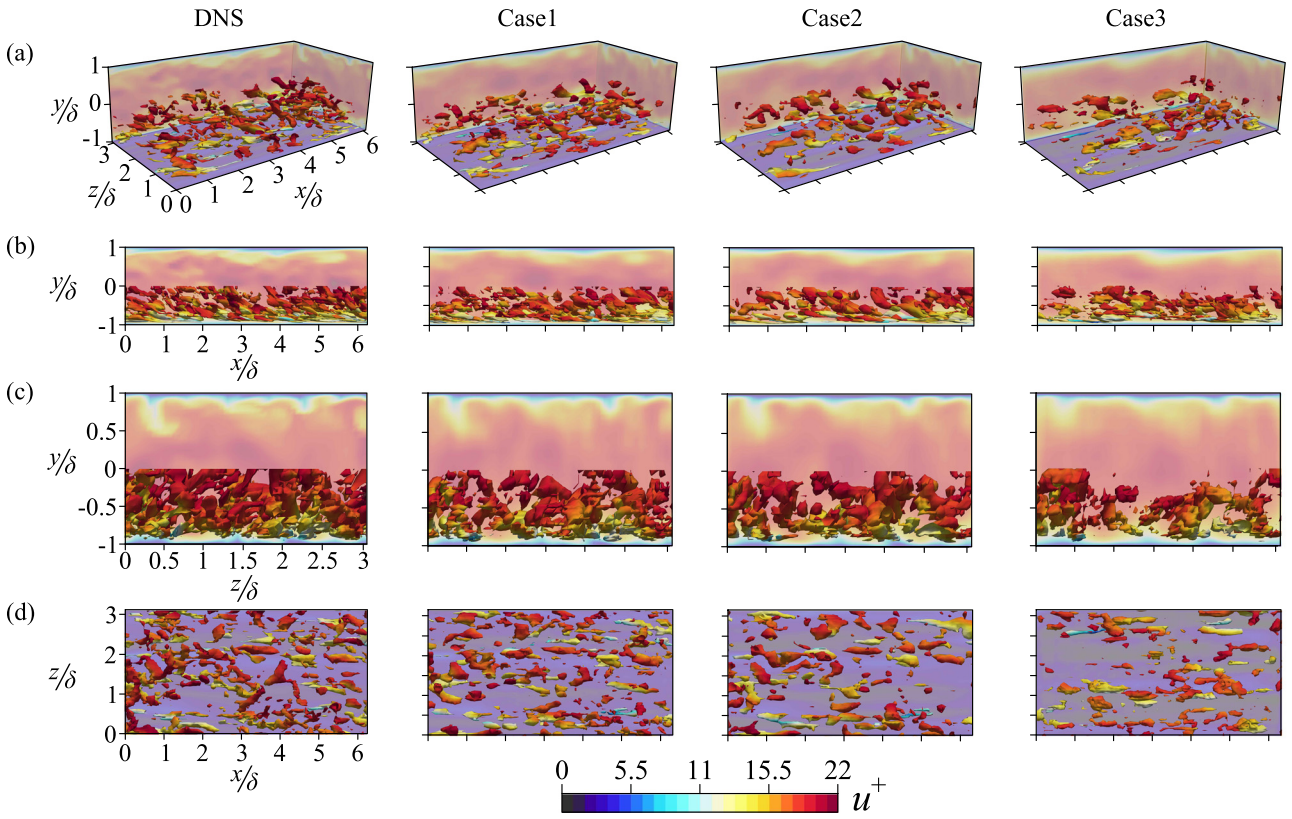

FIG. 8. Isosurfaces of the instantaneous flow structure ($Q^+ = 0.006$) of channel flow at $Re_\tau = 180$ colored by u^+. (a) 3D orthogonal view. (b) Front view. (c) Side view. (d) Top view.

of the approximate mapping of low-resolution data to high-resolution data. In the fine-training process, the model is trained to learn a high-fidelity mapping based on the limited DNS dataset (1000 snapshots), enabling the model to reconstruct high-resolution turbulent-flow data from coarse data more accurately.

In Sec. IV, we show that the model with tricubic interpolation-based transfer learning (case 2) performs better than the one without tricubic interpolation (case 3) when the DNS training datasets are identical (1000 snapshots for each case). Meanwhile, the reconstruction performance in case 2 is similar to the normal training result, whose training DNS database includes 9600 snapshots (case 1).

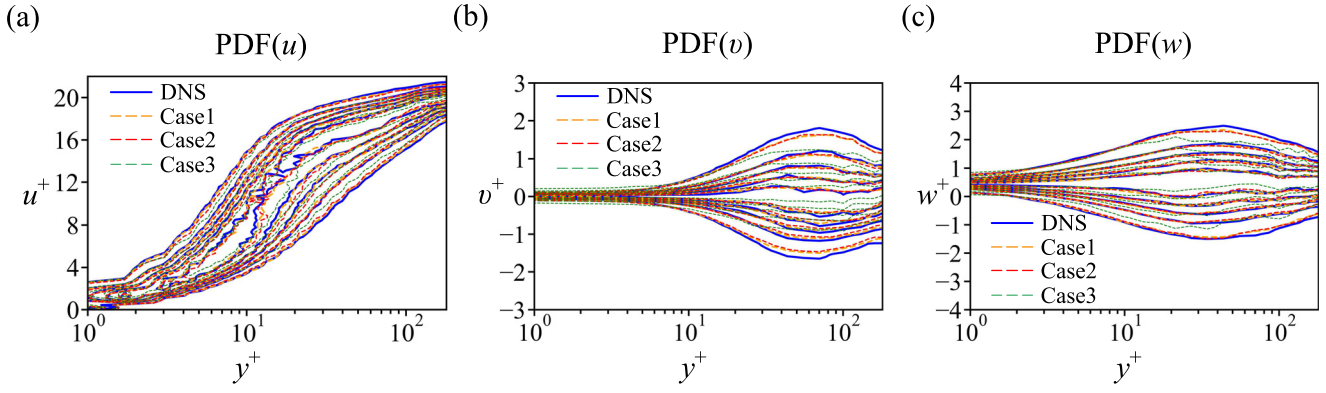

FIG. 9. Probability density functions of the velocity components of channel flow at $Re_\tau = 180$ as a function of wall-normal distance. The isoline levels are in the range 10%–90% of the maximum PDF with an increment of 20%. Each figure indicates the probability density functions of velocity component (a) u, (b) v, and (c) w.

IV. RESULTS AND DISCUSSION

In this section, the turbulent channel flow data will be validated to demonstrate the ability of the 3D-ESRGAN, which would be able to reconstruct high-resolution wall-bounded turbulent flow fields from low-resolution turbulent flow fields. The results are presented and discussed. Figures 5–7 show the reconstructed instantaneous fields of the streamwise velocity (u^+), wall-normal velocity (v^+), and spanwise velocity (w^+) from different sections in the 3D domain of the channel flow at $Re_\tau = 180$. As illustrated in the figures, the reconstructed velocity fields of cases 1 and 2 agree well with the DNS data, with most of the detailed flow structures reconstructed. Nevertheless, the reconstruction performance in case 3 is worse than in the other cases. In particular, the reconstructed velocity of the streamwise component (u) in case 3 is widely different from the DNS data, which is also observed in another figure (see Fig. 8). Figure 8 shows the turbulent-flow structure in the channel by applying the Q criterion for vortex identification.[62] As shown in this figure, the reconstructed instantaneous velocity fields of cases 1 and 2 represented in terms of the vortical structure (Q^+) are identical to those obtained from DNS data, whereas the reconstruction data in case 3 are less similar to the DNS one. This indicates that the proposed model can successfully reconstruct flow fields with an actual structure for both cases 1 and 2.

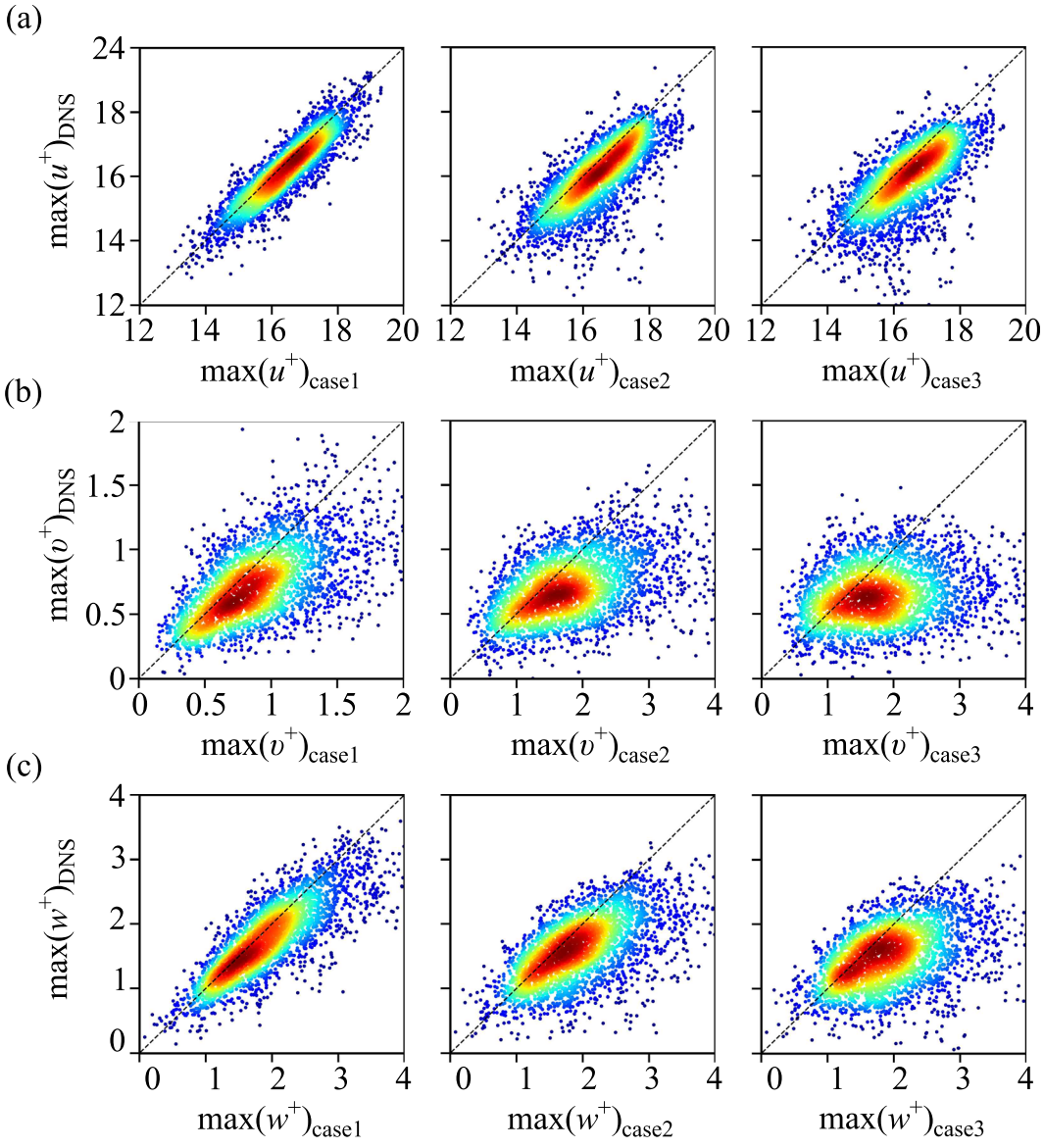

FIG. 10. Scatter plots of the maximum values within the x–z plane at $y^+ = 16.78$ of the (a) streamwise velocity, (b) wall-normal velocity, and (c) spanwise velocity based on DNS, for the various reconstruction cases in channel flow data at $Re_\tau = 180$. The contour colors (from blue to red) are proportional to the density of points in the scatter plot.

In Fig. 9, the probability density function (PDF) of each velocity component is plotted as a function of the wall-normal location to evaluate the capability of the 3D-ESRGAN model to reconstruct velocity fields with accurate spatial distributions. In terms of cases 1 and 2, the PDF plots of the reconstructed velocity components agree considerably with the results obtained from DNS data. In case 3, the PDF plots significantly deviated from the DNS ones.

Figure 10 shows the scatter plots of the extreme values of velocity fields based on the x–z plane at $y^+ = 16.78$. As can be observed from the figure, the results from cases 1 and 2 correlate well with the DNS data. Thus, the 3D-ESRGAN model with tricubic interpolation-based transfer learning can adequately reconstruct the true extreme values of turbulent flows.

Moreover, the turbulence statistics of reconstructed velocity fields are compared with the fields obtained from the DNS data in Fig. 11. In Fig. 11(a), the mean streamwise velocity profiles from all the cases agree with the results obtained from DNS data within the considered wall-normal range. The root mean square (RMS) of the velocity components is defined as: $\alpha_{rms} = \overline{\alpha'\alpha'}^{1/2}$, where α represents the velocity component. The statistically averaged quantities in x and z denoted by an overbar can be defined as

$$\bar{\phi} = \frac{1}{L_{x_i}(t_1 - t_0)} \int_{t_0}^{t_1} \int_0^{L_{x_i}} \phi \, dx_i \, dt, \tag{14}$$

whereas fluctuating quantities are denoted by primes, that is, $u = \bar{u} + u'$. As shown in Figs. 11(b)–11(d), the results obtained from cases 1 and 2 agree well with the DNS results. However, results from case 3 show obvious offsets. The above results indicate that, in case 3, the 3D-ESRGAN model can roughly reconstruct high-resolution flow fields but cannot reconstruct detailed turbulent features, such as the fluctuation part when y^+ is greater than 15. Figure 11(e) depicts the Reynolds shear-stress profile $-\overline{u'v'}^+$. Although the results of cases 1 and 2 are in better agreement than the results of case 3, there exists a slight deviation near $y^+ = 45 - 125$. Similarly, the RMS streamwise vorticity profile ($w^+_{x,rms}$) also shows similar results to RMS velocity fluctuation and mean shear stress, which means that in cases 1 and 2, 3D-ESRGAN can reconstruct the turbulence statistics of flows with remarkable precision.

In addition, this study uses premultiplied streamwise and spanwise wavenumber spectra to investigate the capability of the model to reconstruct the premultiplied power-spectral densities of these flows, that is, $k_x^+\Phi^+_{\alpha\alpha}$ and $k_z^+\Phi^+_{\alpha\alpha}$, where k_x and k_z represent the streamwise and spanwise wavenumbers, respectively; $\Phi_{\alpha\alpha}$ denotes the corresponding power-spectral density; and α represents the velocity component. Figure 12 shows $k_x^+\Phi^+_{\alpha\alpha}$ and $k_z^+\Phi^+_{\alpha\alpha}$ as a function of y^+ and the corresponding wavelength λ^+. Except for case 3, the energy spectra of the reconstructed velocity fields match well with the DNS results even though offsets exist at high-wavenumber regions. However, the results

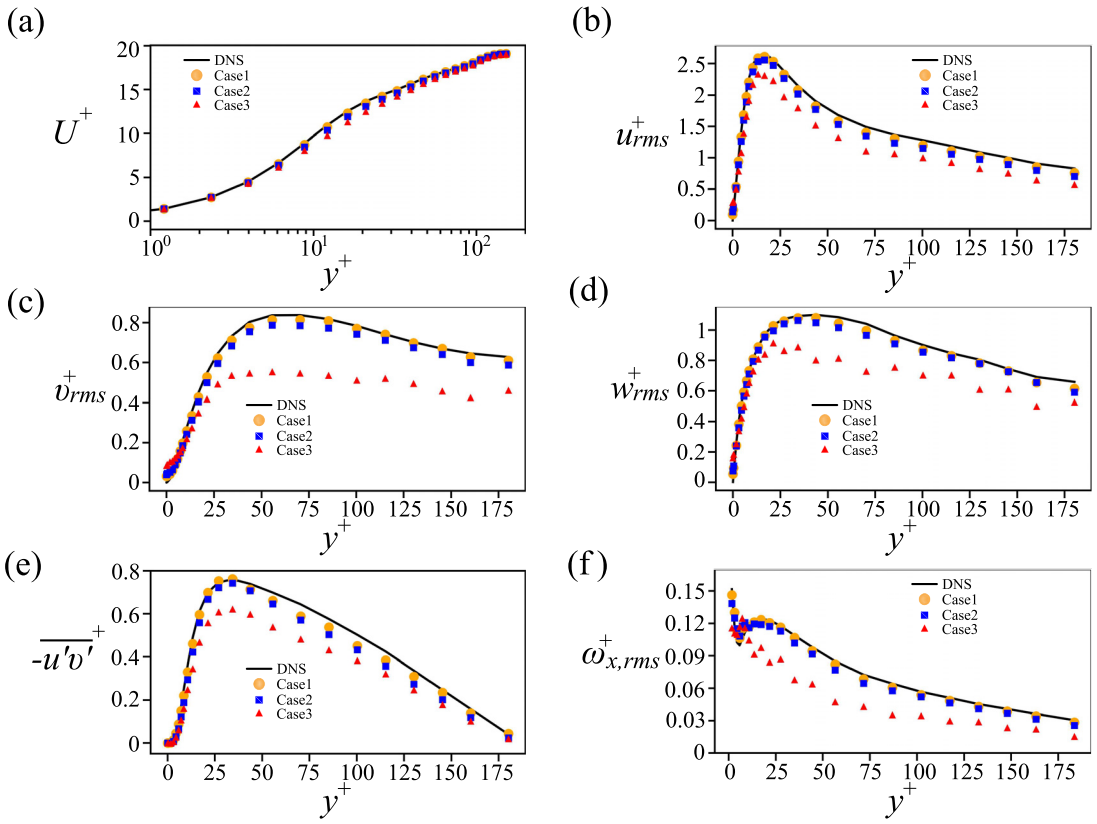

FIG. 11. Turbulence statistics of the turbulent channel flow at $Re_\tau = 180$. (a) Mean streamwise velocity profile, (b) RMS profile of the streamwise velocity, (c) RMS profile of the wall-normal velocity, (d) RMS profile of the spanwise velocity, (e) mean shear stress profile, and (f) RMS profile of the streamwise vorticity.

from case 3 show extremely poor agreement with the DNS results, particularly at the wavelength range of less than 10^2. This figure also shows that the DL-based model can reproduce energy spectra well and accurately reconstruct the spatial distribution of turbulence velocity fields.

The flow at high Reynolds number ($Re_\tau = 500$) is more chaotic than the lower Reynolds number. The high-resolution flow data can be reconstructed using the 3D-ESRGAN model with commendable accuracy. Figure 13 shows turbulence statistics for the flow at $Re_\tau = 500$. Generally, the statistical results from cases 1 and 2 are in acceptable agreement with the DNS results; however, similar to the previous discussion, in case 3 the performance is worse than in the other two cases.

Finally, the performance of the proposed 3D-ESRGAN model and different training methods are further evaluated using the L_2-norm relative error

$$\varepsilon = \frac{1}{N}\sum_{n=1}^{N}\frac{||\alpha_n^{\mathrm{DNS}} - \alpha_n^{\mathrm{REC}}||_2}{||\alpha_n^{\mathrm{DNS}}||_2} \times 100\%, \tag{15}$$

where α_n^{DNS} and α_n^{REC} represent the DNS and reconstructed instantaneous velocity components at a specific wall-normal distance, y^+, and N represents the number of the test flow data. In Figs. 14(a) and 14(b), we can find that the L_2-norm relative errors of cases 1 and 2 are similar, even if the error of case 1 is slightly lower than that of case 2. However, the error of case 3 shows a rapid increase. This result also indicates that tricubic interpolation-based transfer learning method can help the DL-based model to reconstruct high-resolution flow fields well, even using limited DNS training data. The figure shows that the errors at $y^+ = 25$–75 of the case at $Re_\tau = 180$ and $y^+ = 10$–100 of the case at $Re_\tau = 500$ are larger than in other positions. In these wall-normal regions, beyond the buffer layer, the flow is more difficult to reconstruct because the velocity fluctuations are associated with fluctuations of many different scales. In addition, the error of the case at $Re_\tau = 500$ is larger than that of the case at $Re_\tau = 180$ because the turbulent flow at $Re_\tau = 500$ is more chaotic than that at $Re_\tau = 180$.

V. CONCLUSIONS

This study proposed a DL-based method to reconstruct 3D high-resolution turbulent flows from low-resolution data. We developed a DL neural network named 3D-ESRGAN, a 3D version of the general MS-ESRGAN. The 3D-ESRGAN was used to reconstruct 3D channel flows from spatially coarse data at two different friction Reynolds numbers ($Re_\tau = 180$ and 500). Unlike previous two-dimensional resolution–reconstruction works, continuity and momentum equations were included in the physics-based loss terms of G so that the DL-based model could achieve better and more physical reconstruction results. Moreover, the pre-trained network VGG19 could not be used

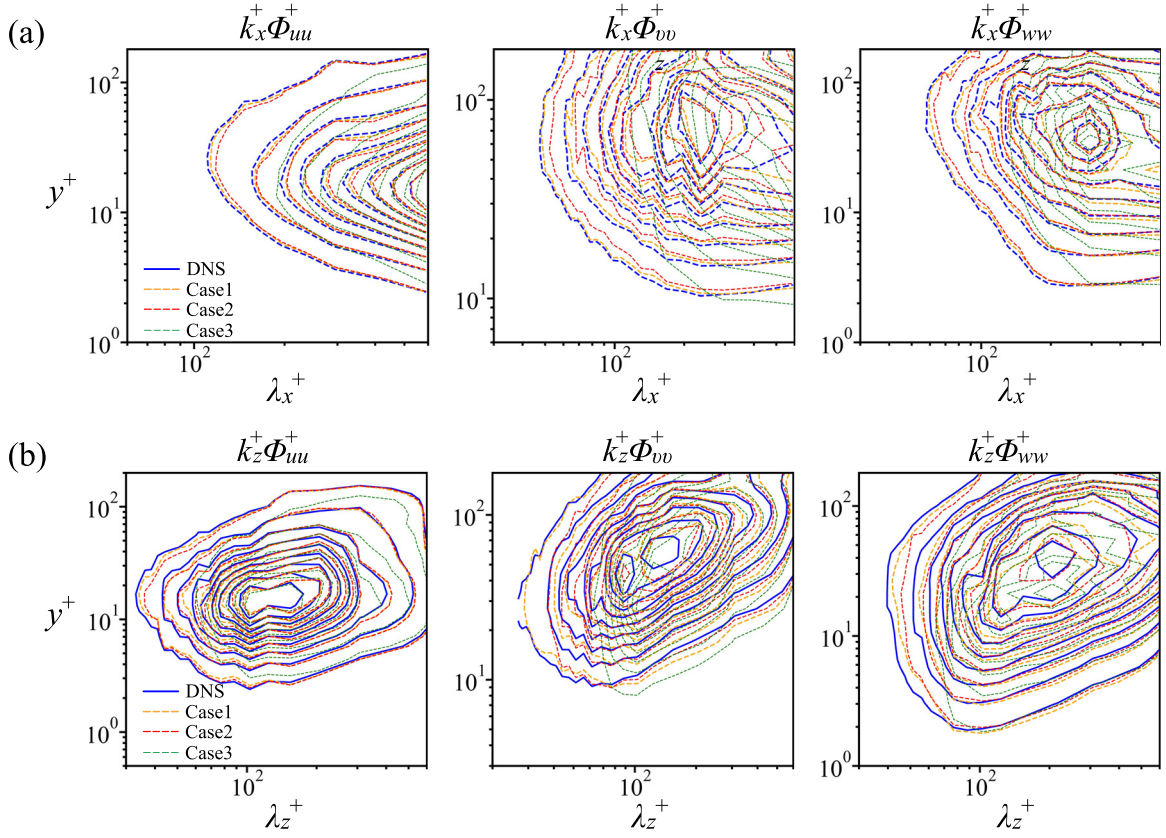

FIG. 12. Premultiplied streamwise (a) and spanwise (b) power-spectral density of velocity components from the turbulent channel flow case $Re_\tau = 180$ as a function of wall-normal distance and wavelength. The isoline levels are in the range 10%–90% of the maximum $k_x^+\Phi_{\alpha\alpha}^+$ and $k_z^+\Phi_{\alpha\alpha}^+$, with increments of 10%.

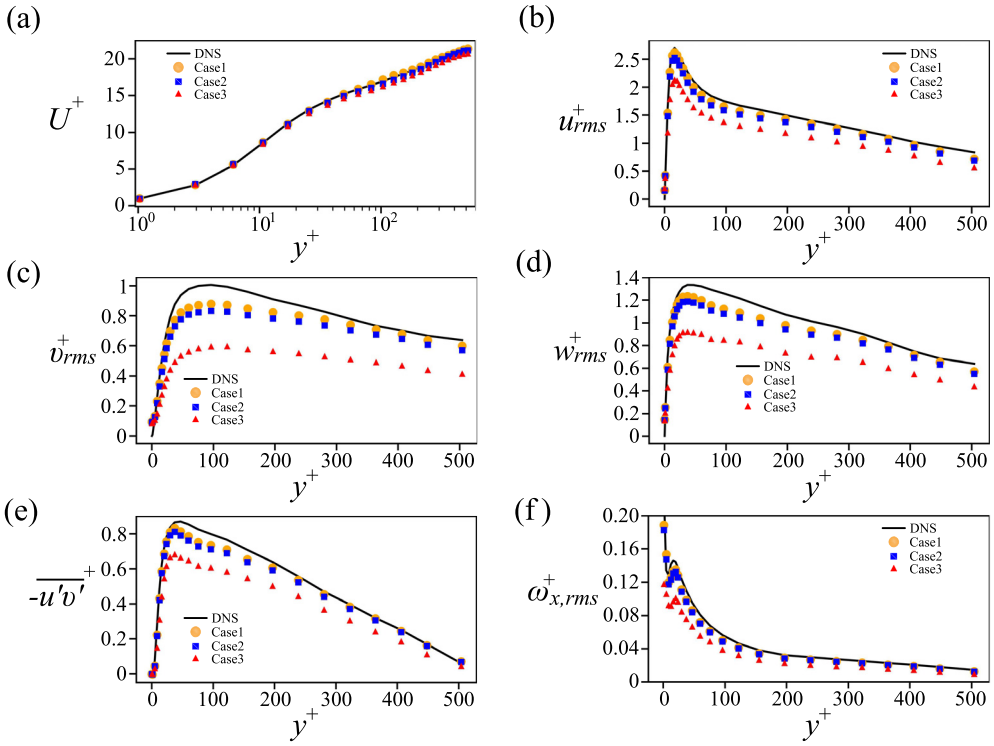

FIG. 13. Turbulence statistics of the turbulent channel flow at $Re_\tau = 500$. (a) Mean streamwise velocity profile, (b) RMS profile of the streamwise velocity, (c) RMS profile of the wall-normal velocity, (d) RMS profile of the spanwise velocity, (e) mean shear stress profile, and (f) RMS profile of the streamwise vorticity.

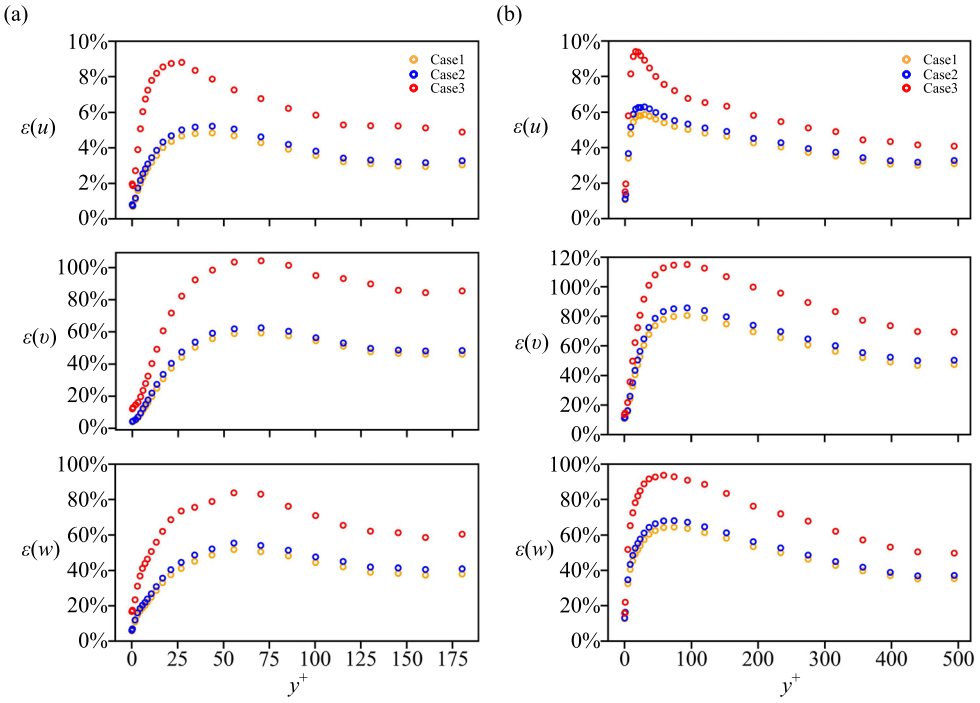

FIG. 14. Profiles of L_2-norm error of reconstructed velocity fields based on inner-scaled wall-normal distances y^+. (a) Turbulent channel flow at $Re_\tau = 180$, and (b) turbulent channel flow at $Re_\tau = 500$.

in this study because of its limited capacity to handle 3D data. Thus, we trained an FE using a 3D convolutional auto-encoder to extract features from real and artificial data. Finally, we proposed a new training method similar to transfer learning, that is, tricubic interpolation-based transfer learning. This method reconstructed low-resolution data roughly and was used to train the 3D-ESRGAN model to improve its generalization. Then, limited DNS data were used to train the model finely, enabling the DL-based model to achieve more accurate results.

We used turbulent channel flows at two different Reynolds numbers to evaluate the reconstruction ability of 3D-ESRGAN based on three different training methods. The results from instantaneous velocity fields, isosurfaces of the instantaneous flow structure, and turbulence statistics demonstrated that with limited paired DNS training data 3D-ESRGAN could effectively reconstruct high-resolution channel-flow data at $Re_\tau = 180$ from spatially limited flow data. In addition, 3D-ESRGAN could reconstruct high-resolution channel flows at $Re_\tau = 500$ even though the Reynolds number is higher and flows are more chaotic than at $Re_\tau = 180$. Moreover, the L_2-norm relative error was investigated in this study. In addition to the previously mentioned results, fluctuating region of channel flows was more difficult to reconstruct than non-fluctuating region.

This study has demonstrated that the 3D-ESRGAN model can reconstruct high-resolution turbulent-flow data efficiently and accurately. By applying tricubic interpolation-based transfer learning, we can train the DL-based model with limited paired data while maintaining the results and the regular training method.

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 (No. 20214000000140). In addition, this work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2019R1I1A3A01058576). This work was also supported by the National Supercomputing Center with supercomputing resources including technical support (No. KSC-2022-CRE-0282). R.V. acknowledges the financial support from the ERC Grant No. 2021-CoG-101043998, DEEPCONTROL. S.H. was funded by Contract No. PID2021–128676OB-I00 of Ministerio de Ciencia, innovación y Universidades/FEDER.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Linqi Yu: Conceptualization (equal); Data curation (lead); Formal analysis (equal); Investigation (lead); Methodology (equal); Software (lead); Validation (lead); Writing – original draft (lead). **Mustafa Z. Yousif:** Conceptualization (lead); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (lead); Validation (equal); Writing – review & editing (lead). **Meng Zhang:** Data curation (equal); Methodology (equal); Software (equal); Validation (equal). **Sergio Hoyas:** Data curation (equal); Writing – review & editing (equal). **Ricardo Vinuesa:** Investigation (equal); Writing – review & editing (equal). **Heechang Lim:** Conceptualization (equal); Funding acquisition (lead); Investigation (lead); Project administration (lead); Resources (lead); Supervision (lead); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available within this article.

REFERENCES

[1]P. Moin and K. Mahesh, "Direct numerical simulation: A tool in turbulence research," Annu. Rev. Fluid Mech. **30**, 539–578 (1998).
[2]J. Kim, P. Moin, and R. Moser, "Turbulence statistics in fully developed channel flow at low Reynolds number," J. Fluid Mech. **177**, 133–166 (1987).
[3]R. Moser, J. Kim, and N. Mansour, "Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$," Phys. Fluids **11**, 943–945 (1999).
[4]R. J. Adrian, "Scattering particle characteristics and their effect on pulsed laser measurements of fluid flow: Speckle velocimetry vs particle image velocimetry," Appl. Opt. **23**, 1690–1691 (1984).
[5]F. Scarano, "Tomographic PIV: Principles and practice," Meas. Sci. Technol. **24**, 012001 (2013).
[6]D. Schanz, S. Gesemann, and A. Schröder, "Shake-the-box: Lagrangian particle tracking at high particle image densities," Exp. Fluids **57**, 70 (2016).
[7]N. Neeteson, S. Bhattacharya, D. Rival, D. Michaelis, D. Schanz, and A. Schröder, "Pressure-field extraction from Lagrangian flow measurements: First experiences with 4D-PTV data," Exp. Fluids **57**, 102 (2016).
[8]S. B. Pope, *Turbulent Flows* (Higher Education from Cambridge University Press, 2000).
[9]L. Deng and D. Yu, "Deep learning: Methods and applications," Found. Trends Signal Process. **7**, 197–387 (2014).
[10]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**, 1–36 (2019).
[11]K. Munir, H. Elahi, A. Ayub, F. Frezza, and A. Rizzi, "Cancer diagnosis using deep learning: A bibliographic review," Cancers **11**, 1235 (2019).
[12]M. Garland, S. L. Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, and V. Volkov, "Parallel computing experiences with CUDA," IEEE Micro **28**, 13–27 (2008).
[13]J. Kutz, "Deep learning in fluid dynamics," J. Fluid Mech. **814**, 1–4 (2017).
[14]S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," Annu. Rev. Fluid Mech. **52**, 477–508 (2020).
[15]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).
[16]K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," Annu. Rev. Fluid Mech. **51**, 357–377 (2019).
[17]R. Vinuesa and S. Brunton, "Enhancing computational fluid dynamics with machine learning," Nat. Comput. Sci. **2**, 358–366 (2022).
[18]T. Nakamura, K. Fukami, K. Hasegawa, Y. Nabae, and K. Fukagata, "Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow," Phys. Fluids **33**, 025116 (2021).
[19]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).
[20]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).
[21]S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," J. Fluid Mech. **879**, 217–254 (2019).

[22]M. Z. Yousif, L. Yu, and H.-C. Lim, "Physics-guided deep learning for generating turbulent inflow conditions," J. Fluid Mech. **936**, A21 (2022).
[23]R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu, "Towards physics-informed deep learning for turbulent flow prediction," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (KDD, 2020), pp. 1457–1466.
[24]P. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, "Predictions of turbulent shear flows using deep neural networks," Phys. Rev. Fluids **4**, 054603 (2019).
[25]H. Eivazi, L. Guastoni, P. Schlatter, H. Azizpour, and R. Vinuesa, "Recurrent neural networks and Koopman-based frameworks for temporal predictions in a low-order model of turbulence," Int. J. Heat Fluid Flow **90**, 108816 (2021).
[26]D. Schmekel, F. Alcántara-Ávila, S. Hoyas, and R. Vinuesa, "Predicting coherent turbulent structures via deep learning," Front. Phys. **10** (published online 2022).
[27]B.-Z. Han and W.-X. Huang, "Active control for drag reduction of turbulent channel flow based on convolutional neural networks," Phys. Fluids **32**, 095108 (2020).
[28]R. Vinuesa, O. Lehmkuhl, A. Lozano-Duran, and J. Rabault, "Flow control in wings and discovery of novel approaches via deep reinforcement learning," Fluids **7**, 62 (2022).
[29]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," arXiv.2208.05754 (2022).
[30]M. Buzzicotti, F. Bonaccorso, P. C. Di Leoni, and L. Biferale, "Reconstruction of turbulent data with deep generative models for semantic inpainting from TURB-Rot database," Phys. Rev. Fluids **6**, 050503 (2021).
[31]K. Fukami, B. An, M. Nohmi, M. Obuchi, and K. Taira, "Machine-learning-based reconstruction of turbulent vortices from sparse pressure sensors in a pump sump," J. Fluids Eng. **144**, 121501 (2022).
[32]S. M. A. Bashir, Y. Wang, M. Khan, and Y. Niu, "A comprehensive review of deep learning-based single image super-resolution," PeerJ Comput. Sci. **7**, e621 (2021).
[33]R. Keys, "Cubic convolution interpolation for digital image processing," IEEE Trans. Acoust., Speech, Signal Process. **29**, 6 (1981).
[34]K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," J. Fluid Mech. **870**, 106–120 (2019).
[35]B. Liu, J. Tang, H. Huang, and X.-Y. Lu, "Deep learning methods for super-resolution reconstruction of turbulent flows," Phys. Fluids **32**, 025105 (2020).
[36]H. Kim, J. Kim, S. Won, and C. Lee, "Unsupervised deep learning for super-resolution reconstruction of turbulence," J. Fluid Mech. **910**, A29 (2021).
[37]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," arXiv:1809.00219 (2018).
[38]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).
[39]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).
[40]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).
[41]K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," J. Big Data **3**, 9 (2016).
[42]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).
[43]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).
[44]S. Hoyas and J. Jiménez, "Scaling of the velocity fluctuations in turbulent channels up to $Re_\tau = 2003$," Phys. Fluids **18**, 011702 (2006).
[45]F. Lluesma-Rodríguez, F. Alcantara-Avila, M. J. Perez-Quiles, and S. Hoyas, "A code for simulating heat transfer in turbulent channel flow," Mathematics **9**, 756 (2021).
[46]M. Oberlack, S. Hoyas, S. Kraheberger, F. Alcántara-Ávila, and J. Laux, "Turbulence statistics of arbitrary moments of wall-bounded shear flows: A symmetry approach," Phys. Rev. Lett. **128**, 024502 (2022).
[47]S. Hoyas, M. Oberlack, F. Alcántara-Ávila, S. Kraheberger, and J. Laux, "Wall turbulence at high friction Reynolds numbers," Phys. Rev. Fluids **7**, 014602 (2022).
[48]F. Alcántara-Ávila, S. Hoyas, and M. J. Pérez-Quiles, "Direct numerical simulation of thermal channel flow for $Re_\tau = 5000$ and $\mathrm{Pr} = 0.71$," J. Fluid Mech. **916**, A29 (2021).
[49]S. Lele, "Compact finite difference schemes with spectral-like resolution," J. Comput. Phys. **103**, 16–42 (1992).
[50]P. Spalart, M. Moser, and R. D. Rogers, "Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions," J. Comput. Phys. **96**, 297–324 (1991).
[51]I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (NIPS, 2014), Vol. **2**, pp. 2672–2680.
[52]J. Islam and Y. Zhang, "GAN-based synthetic brain PET image generation," Brain Inf. **7**, 3 (2020).
[53]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," arXiv:1609.04802 (2017).
[54]Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," arXiv:1802.08797 (2018).
[55]X. Du, X. Qu, Y. He, and D. Guo, "Single image super-resolution based on multi-scale competitive convolutional neural network," Sensors **18**, 789 (2018).
[56]A. Jolicoeur-Martineau, "The relativistic discriminator: A key element missing from standard GAN," arXiv:1807.00734 (2018).
[57]D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2017).
[58]F. Lekien and J. Marsden, "Tricubic interpolation in three dimensions," Int. J. Numer. Methods Eng. **63**, 455–471 (2005).
[59]J. Liu, Z. Gan, and X. Zhu, "Directional bicubic interpolation," in *Proceedings of 3rd International Conference on Multimedia Technology (ICMT-13)* (Atlantis Press, 2013), pp. 463–470.
[60]V. Patel and K. Mistree, "A review on different image interpolation techniques for image enhancement," Int. J. Emerg. Technol. **3**, 7409–7414 (2013).
[61]Y. Zhang, I. Aganj, A. v d Kouwe, and M. Tisdall, "Effects of resolution and registration algorithm on the accuracy of EPI vNavs for real time head motion correction in MRI," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, 2016), pp. 583–591.
[62]A. Lozano-Durán and J. Jiménez, "Time-resolved evolution of coherent structures in turbulent channels: Characterization of eddies and cascades," J. Fluid Mech. **759**, 432–471 (2014).
[63]A. Güemes, S. Discetti, A. Ianiro, B. Sirmacek, H. Azizpour, and R. Vinuesa, "From coarse wall measurements to turbulent velocity fields through deep learning," Phys. Fluids **33**, 075121 (2021).