



## Letter

## Fourier neural operator approach to large eddy simulation of three-dimensional turbulence

Zhijie Li<sup>a,b,c</sup>, Wenhui Peng<sup>a,b,c,d</sup>, Zelong Yuan<sup>a,b,c</sup>, Jianchun Wang<sup>a,b,c,\*</sup><sup>a</sup> Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen 518055, China<sup>b</sup> Southern Marine Science and Engineering Guangdong Laboratory (Guangzhou), 511458, China<sup>c</sup> Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications, Southern University of Science and Technology, Shenzhen 518055, China<sup>d</sup> Department of Computer Engineering, Polytechnique Montreal, Montreal H3T1J4, Canada

## ARTICLE INFO

## Article history:

Received 28 August 2022

Accepted 22 September 2022

Available online 13 October 2022

## Keywords:

Fourier neural operator

Large eddy simulation

Data-driven simulation

Incompressible turbulence

## ABSTRACT

Fourier neural operator (FNO) model is developed for large eddy simulation (LES) of three-dimensional (3D) turbulence. Velocity fields of isotropic turbulence generated by direct numerical simulation (DNS) are used for training the FNO model to predict the filtered velocity field at a given time. The input of the FNO model is the filtered velocity fields at the previous several time-nodes with large time lag. In the *a posteriori* study of LES, the FNO model performs better than the dynamic Smagorinsky model (DSM) and the dynamic mixed model (DMM) in the prediction of the velocity spectrum, probability density functions (PDFs) of vorticity and velocity increments, and the instantaneous flow structures. Moreover, the proposed model can significantly reduce the computational cost, and can be well generalized to LES of turbulence at higher Taylor-Reynolds numbers.

© 2022 The Author(s). Published by Elsevier Ltd on behalf of The Chinese Society of Theoretical and Applied Mechanics.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

In recent years, data-driven approaches based on machine learning (ML) have been widely used to accelerate computational fluid dynamics (CFD) [1,2]. These studies can be mainly classified as ML-assisted modeling approaches which aim to fit the unclosed term of classical turbulence models [3–7], and pure data-driven approaches which aim to approximate the Navier-Stokes equations directly by deep learning methods [8,9]. Ling et al. [10] used a deep learning method to predict the Reynolds stress tensor with the Galilean invariance. Beck et al. [11] presented a method of using the convolution neural network (CNN) and residual neural network (RNN) to build more accurate large eddy simulation (LES) models. Park and Choi [12] developed a subgrid-scale (SGS) model using a fully connected neural network in LES of channel turbulence. Wang et al. [13] proposed a semi-explicit deep learning framework to reconstruct the SGS stress in LES of incompressible turbulence.

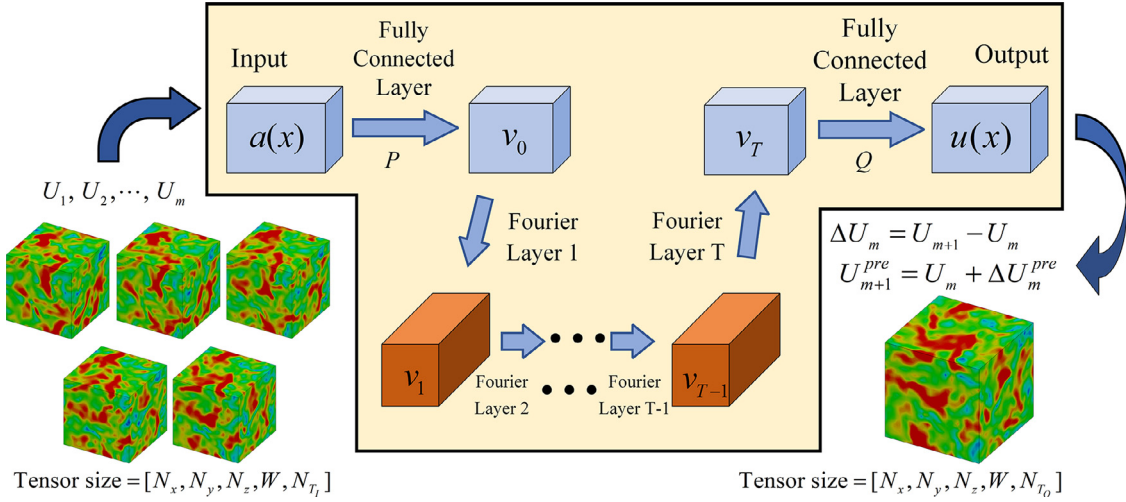
Although ML-assisted modeling approaches have the potential to achieve more accurate results than classical turbulence models, it is still hard to significantly reduce the computational cost. In

contrast, the pure data-driven methods, like a “black-box” model, can make predictions very quickly and efficiently compared with traditional methods [14–16]. Han et al. [17] proposed a hybrid deep neural network model to learn the spatial-temporal features of the unsteady flow fields. Ren et al. [18] combined the long short-term memory and convolution neural network to predict the temporal evolution in turbulence. Guan et al. [19] used a deep convolutional neural network and transfer learning to improve the accuracy and stability of the prediction in turbulence. Li et al. [20] proposed a Fourier neural operator (FNO) framework which can directly learn the mapping between infinite dimensional spaces from a series of input-output pairs. The FNO can perform better than the state-of-the-art models in two-dimensional turbulence prediction tasks on account of its efficient and expressive architecture. Peng et al. [21] introduced an attention-enhanced FNO model which can accurately reconstruct various statistical properties and instantaneous flow structures of two-dimensional (2D) turbulence at high Reynolds numbers. In recent years, FNO has received more and more attention, but it has not been used in the study of three-dimensional (3D) turbulence.

In this paper, we apply the FNO approach to LES of three-dimensional turbulence. We take the filtered velocity fields at previous several time-nodes as inputs of the FNO model and expect

\* Corresponding author.

E-mail address: [wangjc@sustech.edu.cn](mailto:wangjc@sustech.edu.cn) (J. Wang).



**Fig. 1.** Architecture of the Fourier neural operator (FNO) for 3D turbulence prediction. Note that the black box shows the FNO architecture, the lower left corner shows the input of the model, and the lower right corner shows the result calculated from the model output. Here,  $N_x = N_y = N_z = 32$  denotes grid resolutions,  $W = 3$  is number of dimensions,  $N_{t_i} = 5$  and  $N_{t_o} = 1$  are the numbers of time-nodes.  $T = 4$  is the number of Fourier layer and each layer contain 20 Fourier modes,  $m = 5$ .

that the FNO model can learn the evolution of turbulence from the dataset. It is shown that the FNO model can perform with a higher accuracy, more efficiency, and good generalization ability in comparison with traditional SGS models in the *a posteriori* studies of LES.

Most neural network methods are used to learn the mapping between finite-dimensional Euclidean spaces. They can perform well for particular governing equations but are difficult to generalize when the parameters of the equations or conditions change [22–24]. The Fourier neural operator learns the mapping relationships between infinite dimensional spaces from a series of input-output pairs, meaning that FNO learns the rule of an entire family of PDE by using data-driven methods [20]. The FNO model converts the convolution operation in the neural network into a multiplication operation through discrete Fourier transform, which can greatly improve the computational efficiency.

Denote the target non-linear map as  $G^\dagger: \mathcal{A} \rightarrow \mathcal{U}$  where  $\mathcal{A} = \mathcal{A}(D; \mathbb{R}^{d_a})$  and  $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$  are separable Banach spaces of function taking values in  $\mathbb{R}^{d_a}$  and  $\mathbb{R}^{d_u}$ , respectively [25]. Here  $D \subset \mathbb{R}^d$  is a bounded, open set.  $\mathbb{R}$  is real number space, and  $\mathbb{R}^{d_a}$  and  $\mathbb{R}^{d_u}$  are the values sets of  $a(x)$  and  $u(x)$ , respectively. Here,  $d_a$ ,  $d_u$  and  $d_v$  is the dimensions of input  $a(x)$ , output  $u(x)$  and  $v_T(x)$ , respectively. By constructing a parametric map  $G: \mathcal{A} \times \Theta \rightarrow \mathcal{U}$ , the Fourier neural operators can learn to approximate  $G^\dagger$ . By using data-driven methods, FNO will adjust the optimal parameter  $\theta^\dagger \in \Theta$  so that  $G(\cdot, \theta^\dagger) = G_{\theta^\dagger} \approx G^\dagger$ . The input  $a(x)$  is first transformed to a higher dimensional representation  $v_0(x) = P(a(x))$  by a shallow fully-connected layer  $P$ . Then  $v_0(x)$  is updated iteratively by Eq. (1),

$$v_{t+1}(x) = \sigma(Wv_t(x) + (\mathcal{K}(a; \phi)v_t)(x)), \quad \forall x \in D. \quad (1)$$

Here, Fourier integral operator  $\mathcal{K}: \mathcal{A} \times \Theta_{\mathcal{K}} \rightarrow \mathcal{L}(\mathcal{U}(D; \mathbb{R}^{d_v}), \mathcal{U}(D; \mathbb{R}^{d_v}))$  maps to bounded linear operators on  $\mathcal{U}(D; \mathbb{R}^{d_v})$  and is parameterized by  $\phi \in \Theta_{\mathcal{K}}$ .  $W$  is a linear transformation,  $\sigma$  is a non-linear activation function and  $D \subset \mathbb{R}^d$  is a bounded, open set.

Equation (1) shows an iterative architecture  $v_0 \mapsto v_1 \mapsto \dots \mapsto v_T$  where  $v_i$  for  $i = 0, 1, \dots, T-1$  is a sequence of function. Finally, the output  $u(x)$  is projected from  $v_T$  by applying the local transformation  $u(x) = Q(v_T(x))$ , where  $Q: \mathbb{R}^{d_v} \rightarrow \mathbb{R}^{d_u}$  [21]. Here,  $\mathbb{R}^{d_v}$  and  $\mathbb{R}^{d_u}$  are the real numbers sets of  $v_T(x)$  and  $u(x)$ , respectively. An illustration is shown in the black box in Fig. 1. Replacing the kernel integral operator  $(\mathcal{K}(a; \phi)v_t)(x)$  in Eq. (1) by a Fourier convolution

operator  $\mathcal{F}^{-1}(\mathcal{F}(\kappa_\phi) \cdot \mathcal{F}(v_t))(x)$ , and applying the convolution theorem, the Fourier kernel integral operator can be rewritten as Eq. (2).

$$(\mathcal{K}(\phi)v_t)(x) = \mathcal{F}^{-1}(R_\phi \cdot (\mathcal{F}v_t))(x), \quad \forall x \in D, \quad (2)$$

where  $\mathcal{F}$  is the Fourier transform of a function and  $\mathcal{F}^{-1}$  is its inverse transform,  $R_\phi$  is the Fourier transform of a periodic function  $\kappa$  parameterized by  $\phi \in \Theta_{\mathcal{K}}$ .

Since assuming  $\kappa$  is periodic, it admits a Fourier series expansion, meaning that the frequency mode  $k \in D$  can be discretized as  $k \in \mathbb{Z}^d$ . Finite-dimensional parameterization can be obtained by truncating the Fourier series at a maximal mode  $k_{\max} = |Z_{k_{\max}}| = |\{k \in \mathbb{Z}^d: |k_j| \leq k_{\max,j}, \text{ for } j = 1, 2, \dots, d\}|$ .  $\mathcal{F}(v_t) \in \mathbb{C}^{n \times d_v}$  can be obtained by discretizing the domain  $D$  with  $n \in \mathbb{N}$  points,  $v_t \in \mathbb{R}^{n \times d_v}$  [20].  $\mathcal{F}(v_t) \in \mathbb{C}^{k_{\max} \times d_v}$  is obtained by truncating the higher modes. Here,  $\mathbb{C}$  denotes the complex space. Multiplication by the weight tensor  $R_\phi \in \mathbb{C}^{k_{\max} \times d_v \times d_v}$  gives rise to Eq. (3).

$$(R_\phi \cdot (\mathcal{F}v_t))_{k,l} = \sum_{j=1}^{d_v} R_{\phi,k,l,j} (\mathcal{F}v_t)_{k,j}, \quad k = 1, 2, \dots, k_{\max}, \quad j = 1, 2, \dots, d_v. \quad (3)$$

When the flow is uniformly discretized with resolution  $s_1 \times s_2 \times \dots \times s_d = n$ , Fast Fourier Transform (FFT) can be used to replace  $\mathcal{F}$ . For  $f \in \mathbb{R}^{n \times d_v}$ ,  $l = 1, 2, \dots, d_v$ ,  $k = (k_1, k_2, \dots, k_d) \in \mathbb{Z}_{s_1} \times \mathbb{Z}_{s_2} \times \dots \times \mathbb{Z}_{s_d}$ , and  $x = (x_1, x_2, \dots, x_d) \in D$ , the FFT  $\hat{\mathcal{F}}$  and its inverse  $\hat{\mathcal{F}}^{-1}$  are defined as

$$(\hat{\mathcal{F}}f)_l(k) = \sum_{x_1=0}^{s_1-1} \dots \sum_{x_d=0}^{s_d-1} f_l(x_1, x_2, \dots, x_d) e^{-2i\pi \sum_{j=1}^d \frac{x_j k_j}{s_j}}, \quad (4)$$

$$(\hat{\mathcal{F}}^{-1}f)_l(x) = \sum_{k_1=0}^{s_1-1} \dots \sum_{k_d=0}^{s_d-1} f_l(k_1, k_2, \dots, k_d) e^{2i\pi \sum_{j=1}^d \frac{x_j k_j}{s_j}}. \quad (5)$$

A filtering technique can be introduced to separate the physical quantities in turbulence into resolved large-scale and sub-filter small-scale quantities [26,27]. For any fields  $f$  in Fourier space, a filtered field is defined as  $\hat{f}(\mathbf{k}) = \hat{G}(\mathbf{k})f(\mathbf{k})$ . In this study, the sharp spectral filter is used in Fourier space which is defined as  $\hat{G}(\mathbf{k}) = H(k_c - |\mathbf{k}|)$ . Here, cutoff wavenumber  $k_c = \pi/l$ , and  $l$  denotes the filter width. The Heaviside step function  $H(x) = 1$  if  $x \geq 0$ ; other-

**Table 1**  
Parameters of the 3D FNO model.

Layer structure	Epochs	Fourier modes	learning rate	batch size
2 Fully connected layer + 4 Fourier layers	30	20	0.0001	2

wise  $H(x) = 0$  [28]. The filtered incompressible Navier-Stokes equations for the resolved fields can be rewritten as [27,29]

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \quad (6)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial (\bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} + \bar{F}_i. \quad (7)$$

Here, an overbar represents the filtering operation,  $u_i$  denotes  $i$ th velocity component,  $p$  is the pressure,  $\nu$  is the kinematic viscosity,  $\bar{F}_i$  is the  $i$ th large-scale force applied to the fluid moment and  $\tau_{ij}$  is the SGS stress tensor defined by  $\tau_{ij} = \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j$ .

In this research, a pseudo-spectral method is applied to numerical simulation of three-dimensional incompressible homogeneous isotropic turbulence in a cubic box of  $(2\pi)^3$  with periodic boundary conditions on a uniform grid [30]. The large-scale forcing is applied by fixing the velocity spectrum within the two lowest wavenumber shells in the velocity field [31]. A two-step Adams-Bashforth scheme is applied for time marching with the second-order accuracy [32]. The aliasing error is eliminated by truncating the high wavenumbers of Fourier modes by the two-thirds rule [33]. The CFL number is calculated by:  $CFL = \Delta t \max \{|\bar{u}_1|, |\bar{u}_2|, |\bar{u}_3|\} / \Delta x = 0.218$ .

The direct numerical simulation (DNS) is performed with the grid resolutions of  $256^3$  at the Taylor Reynolds number  $Re_\lambda \approx 100$ . The DNS data is filtered into large-scale flow fields at grid resolutions of  $32^3$  by the sharp spectral filter with cutoff wavenumber  $k_c = 10$ . The advanced time step is set to be  $10^{-3}$  and the numerical solution is recorded every 200 steps as a time-node. In this numerical simulation, 45 groups of different random fields are used as the initial fields for subsequent calculations and 600 time-nodes for each group of calculations are stored. Here, we stored the DNS data after the flow fields become statistically steady. The filtered direct numerical simulation (fDNS) data of size  $45 \times 600 \times 32 \times 32 \times 32 \times 3$  can be obtained as the training and testing dataset, which contains 45 groups, each group has 600 time-nodes, and each time-node is a velocity field of  $32^3$  with three directions. 595 input-output pairs of velocity fields can be generated from 600 time-nodes. Therefore, 45 groups contain a total of 26775 sample pairs of velocity fields, where 80% of the samples are used for training and 20% for testing.

Denotes the velocity field of  $m$ th time-node as  $U_m$ . The  $m$ th evolution increment field  $\Delta U_m = U_{m+1} - U_m$  represents the difference of velocity field between two adjacent time-nodes. In this deep learning model, the velocity field of the previous five time-nodes ( $U_1, U_2, U_3, U_4, U_5$ ) are taken as input and the difference between the sixth and fifth velocity field ( $\Delta U_5 = U_6 - U_5$ ) is taken as output, as shown in Fig. 1. When the predicted evolution increment  $\Delta U_5^{pre}$  is obtained from the trained model, then the predicted sixth velocity field can be calculated by  $\Delta U_6^{pre} = U_5 + \Delta U_5^{pre}$ . With the same method,  $\Delta U_7^{pre}$  can be predicted by  $(U_2, U_3, U_4, U_5, U_6^{pre})$  and so on.

The parameters of the 3D FNO model are listed in Table 1. In the process of optimizing the number of modes and the number of Fourier layers, we found that when choosing 4 Fourier layers with 20 modes for each layer, the model performance is better. When the numbers of Fourier layers and modes are set to large values, it will lead to too many model parameters and the calculation consumption is too large. When the numbers of Fourier layers

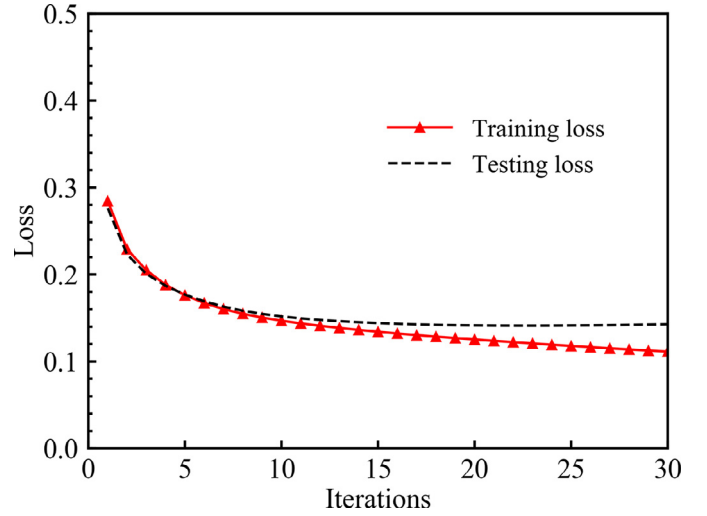


Fig. 2. Learning curves of the FNO model for LES of 3D turbulence.

and modes are small, the model accuracy will be insufficient. The ReLU function and adaptive moment estimation (Adam) are chosen as the activation function and optimizer, respectively [34]. The FNO model is trained for 30 epochs with a learning rate of 0.0001 and batch size of 2. The learning curves of the FNO model for LES of 3D turbulence are shown in Fig. 2. It can be seen that the curves of training loss and testing loss are well convergent and quite close to each other.

To ensure the generalization of the model, five more groups of data from different initial fields are generated and used for the *a posteriori* test. In the following *a posteriori* test, the FNO model compares with the results from fDNS data and classical LES models, including the dynamic Smagorinsky model (DSM) and the dynamic mixed model (DMM) [35–37]. All simulations are initialized by the same flow field of filtered DNS data. The spectra of velocity and velocity error field (error compared with fDNS data) at fifteenth time-nodes ( $t/\tau = 4.29$ ) are shown in Fig. 3. Here  $\tau \equiv L_l/u^{rms}$  denotes the large-eddy turnover time [38] and is about 0.7.  $L_l$  denotes the integral scale and  $u^{rms}$  is the root-mean-square (RMS) of velocity. The interval between every two adjacent time-nodes is  $\Delta t = 0.2$  and the dimensionless time is  $\Delta t/\tau = 0.286$ . For different models, the errors of the velocity spectrum increase as  $k$  increases. Both classical LES models can agree well with fDNS in the low-wave number region and deviate significantly from fDNS near the cut-off wavenumber. In contrast, the FNO model can well reconstruct the velocity spectrum at different flow scales. Meanwhile, the spectra of the velocity error field predicted by FNO are less than those predicted by DSM and DMM near the cut-off wavenumber region. Therefore, the velocity spectrum predicted by the FNO model is very close to that of fDNS.

Figure 4 shows the probability density functions (PDFs) of the normalized vorticity and the normalized velocity increment  $\delta_r \bar{u} / \bar{u}^{rms}$ , where  $\delta_r \bar{u} = [\bar{\mathbf{u}}(\mathbf{x} + \mathbf{r}) - \bar{\mathbf{u}}(\mathbf{x})] \cdot \hat{\mathbf{r}}$  represents the longitudinal increment of the velocity at the separation  $\mathbf{r}$ . Here,  $\hat{\mathbf{r}} = \mathbf{r}/|\mathbf{r}|$ . The vorticity and the velocity increment are normalized by the RMS values of vorticity  $\bar{\omega}^{rms}$  and velocity  $\bar{u}^{rms}$ , respectively. It can be seen that both FNO and DSM models can predict the PDF of

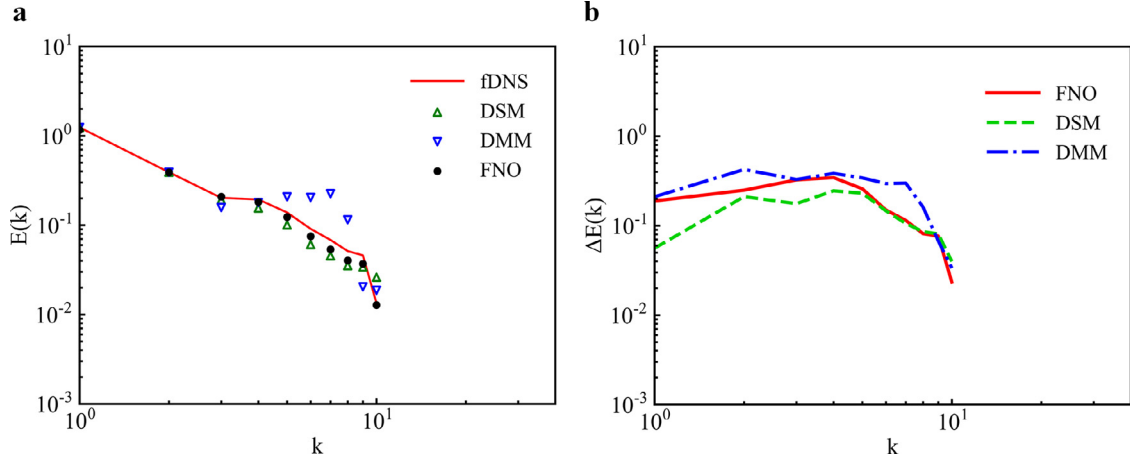


Fig. 3. Spectra of velocity and velocity error field for LES at  $t/\tau = 4.29$ : **a** velocity spectrum, **b** velocity error field spectrum.

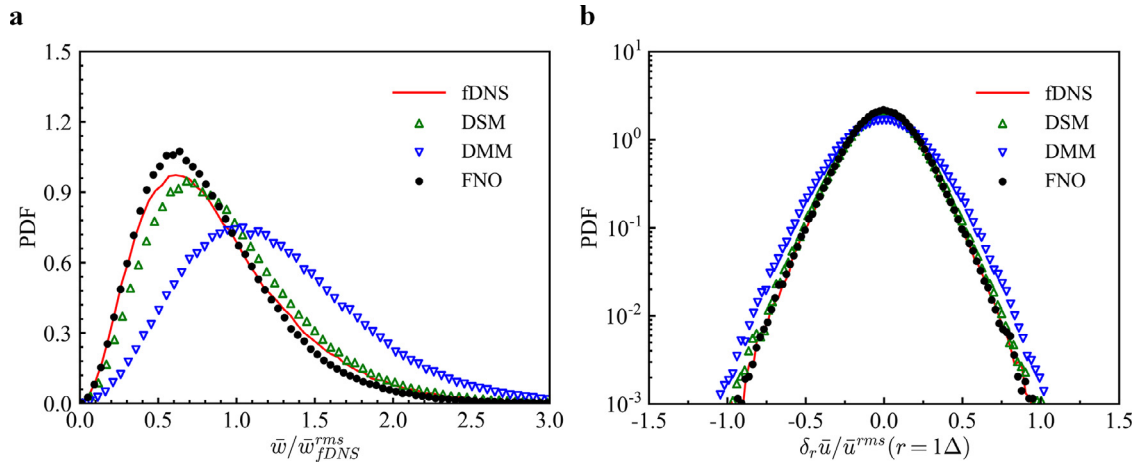


Fig. 4. PDFs of normalized vorticity and normalized increments of the velocity for LES at  $t/\tau = 4.29$ : **a** normalized vorticity, **b** normalized increments of the velocity.

$\bar{\omega}/\bar{\omega}^{rms}$  better than that of the DMM model, and the prediction by FNO model is closer to that of fDNS than DSM model for  $\bar{\omega}/\bar{\omega}^{rms}$  less than 0.5 and between 0.8 to 1.5, which indicates that the FNO model can reconstruct the vortical coherent structures more accurately in this case. The PDF of  $\delta_r \bar{u}/\bar{u}^{rms}$  predicted by the DMM model is significantly wider than that of fDNS. In contrast, the PDFs predicted by the FNO and DSM models are in good agreement with that of fDNS.

The spatial structures of vorticity of fDNS and LES with DSM and FNO models at the instant  $t/\tau \approx 1.0$  (third time-node) are shown in Fig. 5. It can be seen that the large vortex structures from fDNS can be caught by both DSM and FNO models. Moreover, the overall instantaneous vorticity structures predicted by FNO model are much closer to those of fDNS, as compared to those of DSM model.

Figure 6 illustrates the comparison of the predictions of velocity spectrum and PDFs of normalized vorticity at different time  $t/\tau$ . It can be seen that the velocity spectrum and PDFs of normalized vorticity predicted by FNO model are significantly better than those of DSM and DMM models at  $t/\tau \approx 1.0$ . As time increases, the error of velocity spectrum predicted by different models will increase. However, the velocity spectrum predicted by FNO is still better than those of DSM and DMM models at  $t/\tau \approx 5.71$ . The PDFs of normalized vorticity predicted by FNO and DSM models have a good agreement with fDNS at  $t/\tau \approx 5.71$ .

Moreover, the generalization ability of the 3D FNO model is tested. The well-trained FNO model can be directly used to pre-

dict the fDNS data at higher Taylor Reynolds numbers. Figure 7 compares the velocity spectrum and PDFs of normalized vorticity of isotropic turbulence at Taylor Reynolds numbers  $Re_\lambda \approx 160$  and  $Re_\lambda \approx 250$ , respectively. It can be seen that the PDFs of normalized vorticity predicted by DSM and DMM models have large discrepancy with that of fDNS. Because the computational grid is very coarse, the dissipation of the traditional model is insufficient at high Taylor Reynolds numbers. In the Fig. 7, the legend named 'DSM G64' and 'DMM G64' denote the results of DSM and DMM models computed with a resolution of  $64^3$  grids. It can be seen that increasing the number of grids can improve the performance of the DSM and DMM models, but they are still slightly worse than the FNO model in terms of velocity spectrum and PDFs of normalized vorticity. However, in the case of low Taylor Reynolds numbers, due to the large physical viscosity, better results can also be obtained by traditional models. In contrast, the velocity spectrum and PDFs of normalized vorticity predicted by the FNO model are in good agreement with fDNS at high Taylor Reynolds numbers, which are significantly better than those of DSM and DMM models. Since turbulent flow data for high Taylor Reynolds numbers are not readily available in many situations, it is important to train the FNO model using data with lower Reynolds numbers and apply the well trained model to turbulence at high Taylor Reynolds numbers. Considering the self-similarity of turbulence, the large-scale statistical features and flow structures in some regions are insensitive to the Taylor Reynolds numbers, which provides a theoretical basis for developing such a model.



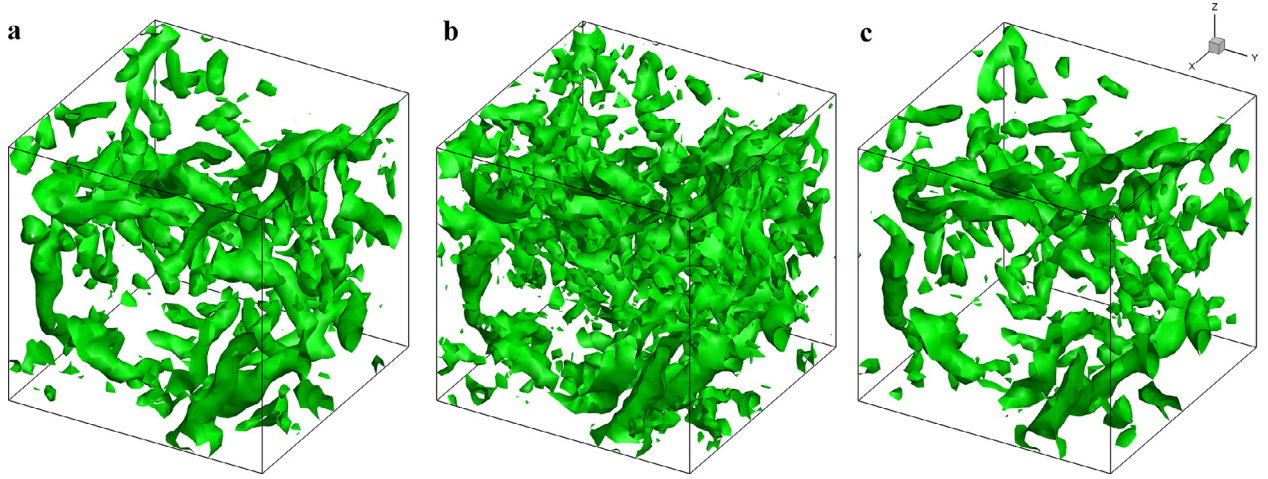


Fig. 5. Isosurface of the normalized vorticity  $\bar{\omega}/\bar{\omega}_{fDNS}^{rms} = 1.5$  at  $t/\tau \approx 1.0$ : **a** fDNS, **b** DSM, **c** FNO.

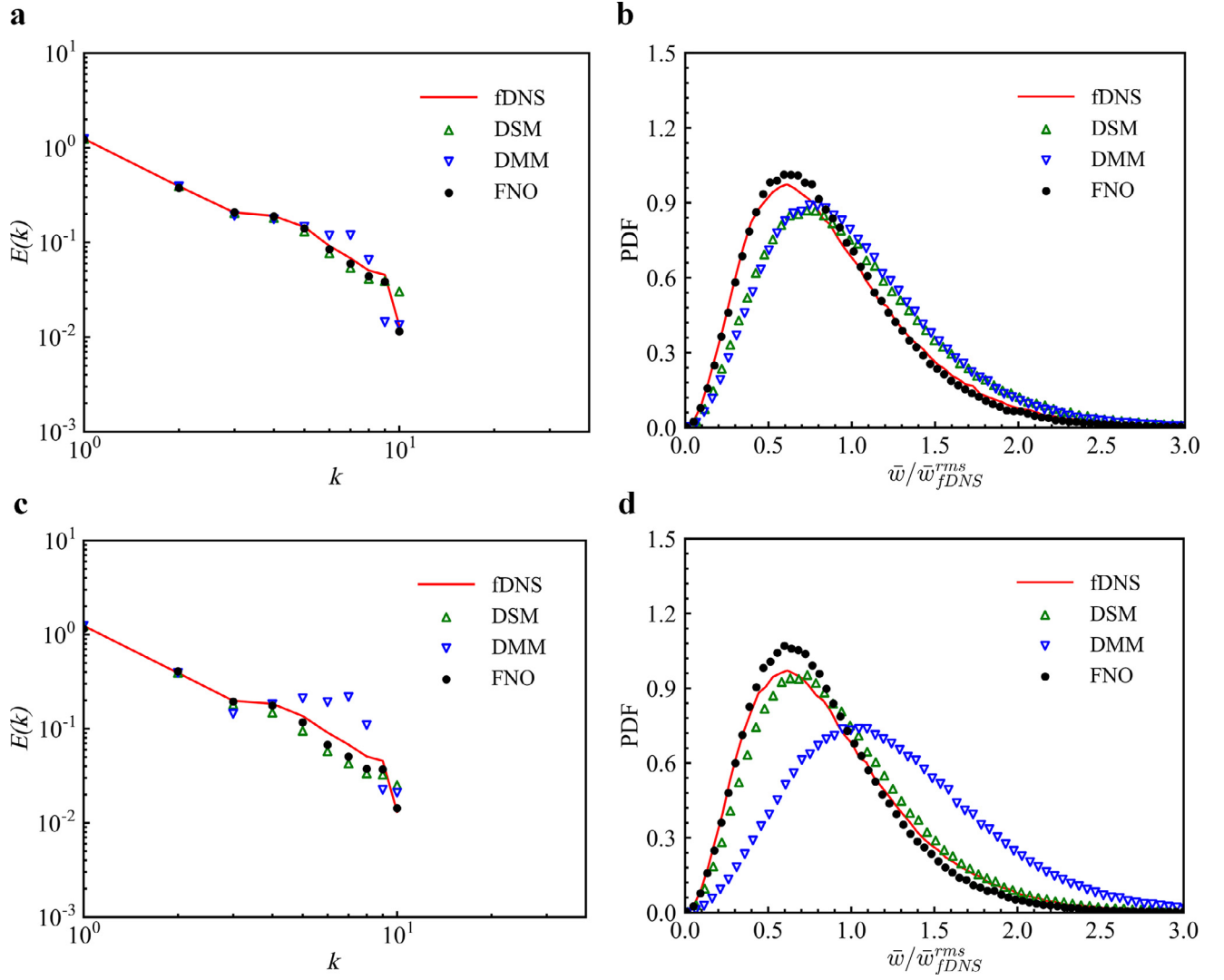
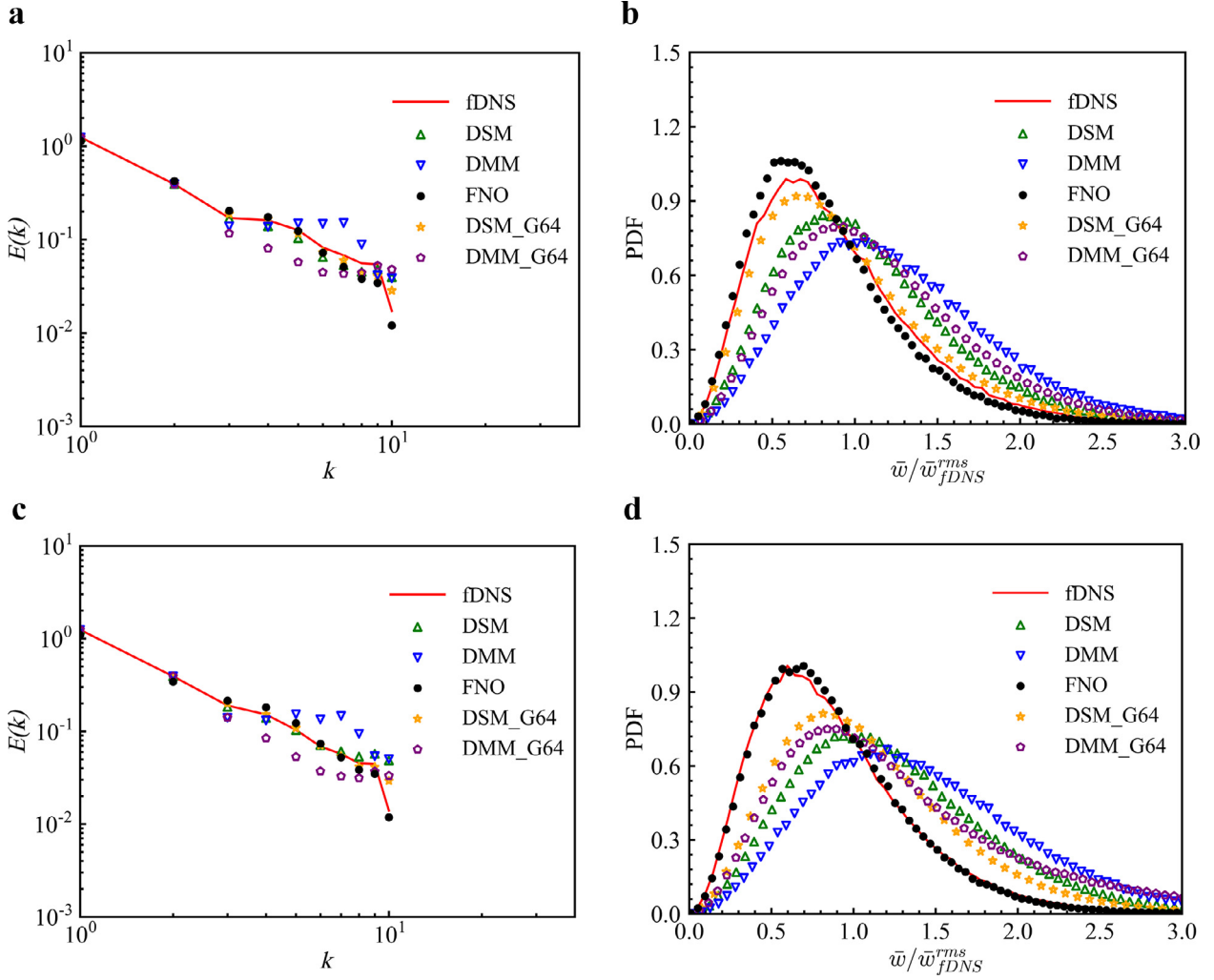


Fig. 6. Velocity spectrum and PDFs of normalized vorticity at different  $t/\tau$ : **a** Velocity spectrum at  $t/\tau \approx 1.0$ , **b** PDFs of normalized vorticity at  $t/\tau \approx 1.0$ , **c** Velocity spectrum at  $t/\tau \approx 5.71$ , **d** PDFs of normalized vorticity at  $t/\tau \approx 5.71$ .



**Fig. 7.** Velocity spectrum and PDFs of normalized vorticity with different Taylor Reynolds numbers at  $t/\tau = 4.29$  : **a** Velocity spectrum at  $Re_\lambda \approx 160$ , **b** PDFs of normalized vorticity at  $Re_\lambda \approx 160$ , **c** Velocity spectrum at  $Re_\lambda \approx 250$ , **d** PDFs of normalized vorticity at  $Re_\lambda \approx 250$ . Specially, the DSM G64 and DMM G64 represent the results computed with a resolution of  $64^3$  grids.

Another great advantage of the FNO model is its computational efficiency. The FNO model is trained on the Pytorch and MindSpore open-source deep learning frameworks. The DSM and DMM simulations are implemented on a virtual machine where the type of CPU is Intel Xeon Gold 6148 CPU @2.40 GHz. Using 32 cores to parallel accelerate the computation, 0.204s and 0.271s are taken for calculating per time-node of DSM and DMM models, respectively. In contrast, the prediction of the FNO model only takes 0.0311s for the same tasks which are conducted on a GPU type of NVIDIA GeForce RTX 2080 Ti. It is worth noting that the DSM and DMM models with  $64^3$  grids take 6.23s and 9.94s to calculate per time-node, respectively, which are 199 and 330 times slower than FNO model.

Different from the traditional neural network that learns the mappings between input and output in the physical space, the FNO model replaces the kernel integral operator with a convolution operator defined in Fourier space and converts it into a multiplication operation by using discrete fast Fourier transform (DFFT). Therefore, compared with the traditional convolutional neural network (CNN) and U-Net models, the FNO model can perform a higher computational efficiency and well capture the multi-scale flow structure of convection-diffusion [20].

The FNO model can directly predict the evolution process of the flow field with a large time step, indicating that this model has learned the integral of the dynamic equation in this time period. However, it cannot be achieved by traditional discrete methods based on differential equations, due to the strong nonlinearity of the dynamical evolution in such a time period. This also explains the advantage of the FNO model in terms of computational efficiency from another perspective.

In this research, we develop an FNO approach to large eddy simulation of three-dimensional turbulence. Filtered DNS flow fields of isotropic turbulence at different times are used for training FNO models. In the *a posteriori* test of LES, it is shown that FNO performs better than the DSM and DMM models in the prediction of velocity spectrum, instantaneous flow structures, and PDFs of vorticity and velocity increments. The present FNO model can be used to predict large scale dynamics of isotropic turbulence at higher Taylor Reynolds numbers. Moreover, the FNO model shows its great potential in terms of computational speed. Our future work aims to improve the generalization ability of the model by using local neural operators avoiding periodic condition, incorporating physical constraints, and hybridizing with transitional methods.

## Declaration of Competing Interest

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

## Acknowledgments

This work was supported by the [National Natural Science Foundation of China](#) (Nos. 91952104, 92052301, 12172161, and 12161141017), by the National Numerical Windtunnel Project (No. NNW2019ZT1-A04), by the [Shenzhen Science and Technology Program](#) (No. KQTD20180411143441009), by Key Special Project for Introduced Talents Team of Southern Marine Science and Engineering Guangdong Laboratory (Guangzhou) (No. GML2019ZD0103), by CAAI-Huawei MindSpore open Fund, and by Department of Science and Technology of Guangdong Province (No. 2019B21203001). This work was also supported by Center for Computational Science and Engineering of Southern University of Science and Technology.

## References

- [1] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* 52 (2020) 477–508.
- [2] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, *Annu. Rev. Fluid Mech.* 51 (2019) 357–377.
- [3] Z. Yuan, C. Xie, J. Wang, Deconvolutional artificial neural network models for large eddy simulation of turbulence, *Phys. Fluids* 32 (2020) 115106.
- [4] Z. Wang, K. Luo, D. Li, et al., Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation, *Phys. Fluids* 30 (2018) 125101.
- [5] C. Xie, Z. Yuan, J. Wang, Artificial neural network-based nonlinear algebraic models for large eddy simulation of turbulence, *Phys. Fluids* 32 (2020) 115101.
- [6] X. Yang, S. Zafar, J.-X. Wang, et al., Predictive large-eddy-simulation wall modeling via physics-informed neural networks, *Phys. Rev. Fluids* 4 (2019) 034602.
- [7] Z. Zhou, G. He, S. Wang, et al., Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network, *Comput. Fluids* 195 (2019) 104319.
- [8] B. Lusch, J.N. Kutz, S.L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, *Nat. Commun.* 9 (2018) 1–10.
- [9] J. Sirignano, K. Spiliopoulos, Dgm: a deep learning algorithm for solving partial differential equations, *J. Comput. Phys.* 375 (2018) 1339–1364.
- [10] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *J. Fluid Mech.* 807 (2016) 155–166.
- [11] A. Beck, D. Flad, C. Munz, Deep neural networks for data-driven LES closure models, *J. Comput. Phys.* 398 (2019) 108910.
- [12] J. Park, H. Choi, Toward neural-network-based large eddy simulation: application to turbulent channel flow, *J. Fluid Mech.* 914 (2021) A16.
- [13] Y. Wang, Z. Yuan, C. Xie, et al., Artificial neural network-based spatial gradient models for large-eddy simulation of turbulence, *AIP Adv.* 11 (2021) 055216.
- [14] D. Kochkov, J.A. Smith, A. Alieva, et al., Machine learning-accelerated computational fluid dynamics, *Proc. Natl. Acad. Sci.* 118 (2021) 2101784118.
- [15] A. Mohan, D. Daniel, M. Chertkov, et al., Compressed convolutional LSTM: an efficient deep learning framework to model high fidelity 3D turbulence, *arXiv preprint arXiv:1903.00033* (2019).
- [16] M.R. Jovanović, From bypass transition to flow control and data-driven turbulence modeling: an input-output viewpoint, *Annu. Rev. Fluid Mech.* 53 (2021) 311–345.
- [17] R. Han, Y. Wang, Y. Zhang, et al., A novel spatial-temporal prediction method for unsteady wake flows based on hybrid deep neural network, *Phys. Fluids* 31 (2019) 127101.
- [18] J. Ren, H. Wang, G. Chen, et al., Predictive models for flame evolution using machine learning: a priori assessment in turbulent flames without and with mean shear, *Phys. Fluids* 33 (2021) 055113.
- [19] Y. Guan, A. Chattopadhyay, A. Subel, et al., Stable a posteriori LES of 2D turbulence using convolutional neural networks: backscattering analysis and generalization to higher re via transfer learning, *J. Comput. Phys.* 458 (2022) 111090.
- [20] Z. Li, N. Kovachki, K. Azizzadenesheli, et al., Fourier neural operator for parametric partial differential equations, *arXiv preprint arXiv:2010.08895* (2020).
- [21] W. Peng, Z. Yuan, J. Wang, Attention-enhanced neural network models for turbulence simulation, *Phys. Fluids* 34 (2022) 025111.
- [22] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [23] S. Pan, K. Duraisamy, Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability, *SIAM J. Appl. Dyn. Syst.* 19 (2020) 480–509.
- [24] H. Xu, D. Zhang, J. Zeng, Deep-learning of parametric partial differential equations from sparse and noisy data, *Phys. Fluids* 33 (2021) 037132.
- [25] B. Beauzamy, Introduction to Banach Spaces and Their Geometry, Vol. 68, North-Holland Mathematics Studies, 1982.
- [26] M. Lesieur, O. Metais, New trends in large-eddy simulations of turbulence, *Annu. Rev. Fluid Mech.* 28 (1996) 45–82.
- [27] C. Meneveau, J. Katz, Scale-invariance and turbulence models for large-eddy simulation, *Annu. Rev. Fluid Mech.* 32 (2000) 1–32.
- [28] S.B. Pope, Turbulent Flows, Cambridge University Press, 2000.
- [29] P. Sagaut, Large Eddy Simulation for Incompressible Flows: An Introduction, Springer Science & Business Media, 2006.
- [30] T. Ishihara, T. Gotoh, Y. Kaneda, Study of high-Reynolds number isotropic turbulence by direct numerical simulation, *Annu. Rev. Fluid Mech.* 41 (2009) 165–180.
- [31] J. Wang, M. Wan, S. Chen, et al., Effect of flow topology on the kinetic energy flux in compressible isotropic turbulence, *J. Fluid Mech.* 883 (2020) A11.
- [32] S. Chen, G.D. Doolen, R.H. Kraichnan, et al., On statistical correlations between velocity increments and locally averaged dissipation in homogeneous turbulence, *Phys. Fluids A Fluid Dyn.* 5 (1993) 458–463.
- [33] G. Patterson Jr, S.A. Orszag, Spectral calculations of isotropic turbulence: efficient removal of aliasing interactions, *Phys. Fluids* 14 (1971) 2538–2541.
- [34] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
- [35] M. Germano, U. Piomelli, P. Moin, et al., A dynamic subgrid-scale eddy viscosity model, *Phys. Fluids A Fluid Dyn.* 3 (1991) 1760–1765.
- [36] B. Vreman, B. Geurts, H. Kuerten, On the formulation of the dynamic mixed subgrid-scale model, *Phys. Fluids* 6 (12) (1994) 4057–4059.
- [37] C. Yu, Z. Xiao, X. Li, Scale-adaptive subgrid-scale modelling for large-eddy simulation of turbulent flows, *Phys. Fluids* 29 (2017) 035101.
- [38] Z. Yuan, Y. Wang, C. Xie, et al., Dynamic iterative approximate deconvolution models for large-eddy simulation of turbulence, *Phys. Fluids* 33 (2021) 085125.