15th World Congress on Computational Mechanics (WCCM-XV)
8th Asian Pacific Congress on Computational Mechanics (APCOM-VIII)
Virtual Congress: 31 July – 5 August 2022
S. Koshizuka (Ed.)

A WAVELET-BASED THREE-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORK FOR SUPERRESOLUTION OF TURBULENT VORTICITY

Tomoki Asaka[1], Katsunori Yoshimatsu[2] and Kai Schneider[3]

[1] Department of Materials Physics, Graduate School of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan
asaka.tomoki@e.mbox.nagoya-u.ac.jp

[2] Institute of Materials and Systems for Sustainability, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
yoshimatsu@nagoya-u.jp

[3] Institut de Mathématiques de Marseille (I2M), Aix Marseille Université, CNRS
39 rue F. Joliot-Curie, 13453 Marseille Cedex 13, France
kai.schneider@univ-amu.fr

Key words: superresolution, wavelets, isotropic turbulence, vortex structures

Abstract.

We develop a wavelet-based three-dimensional convolutional neural network (WCNN3d) for superresolution of coarse-grained data of homogeneous isotropic turbulence. The turbulent flow data are computed by high resolution direct numerical simulation (DNS), while the coarse-grained data are obtained by applying a Gaussian filter to the DNS data. The CNNs are trained with the DNS data and the coarse-grained data. We compare vorticity- and velocity-based approaches and assess the proposed WCNN3d method in terms of flow visualization, enstrophy spectra and probability density functions. We show that orthogonal wavelets enhance the efficiency of the learning of CNN.

1 INTRODUCTION

In recent years, deep learning has been extensively applied to laminar and turbulent flows. For an overview we refer to review articles, e.g., Refs. [1, 2, 3]. Turbulent flows have many degrees of freedom, they are multi-scale, and have coherent vortices, e.g., vortex tubes, to which strong intermittency in space and time is attributed. Because direct numerical simulation (DNS) of turbulent flows needs high computational cost, many simulations using turbulence models, in which the small-scale motions are modeled, are performed. To get deep insight into the simulation data and to improve turbulence models, it is beneficial to reconstruct unresolved-scale data. To this end it is important to develop superresolution (SR) techniques which efficiently handle this task. One of the essential topics is hereby superresolution reconstruction of turbulent flow fields from low-pass filtered or coarse-grained turbulent fields, i.e., low-resolution (LR) turbulent fields. SR is originally an image processing technique reconstructing information that cannot be appropriately expressed by LR images. Dong *et al.* [4] proposed a SR method using convolutional neural networks (CNN) with deep learning. CNN learns characteristic spatial structures by convolution of training data via compression and extraction of the data. Application of SR using two-dimensional (2D) CNN, so-called CNN2d, to turbulence has been proposed for

2D freely-decaying homogeneous turbulence [6], urban turbulence [5], three-dimensional (3D) forced homogeneous isotropic turbulence (e.g., Refs. [7, 8]), and 3D turbulent channel flows (e.g., Refs. [7, 8]). We here focus on 3D forced homogeneous isotropic turbulence which is one of the most canonical turbulent flows. In LR fields, small scales in the dissipation range are filtered out and are thus not available. Lui *et al.* [7] showed that convolutional neural networks using a time series of LR data as an input can indeed reconstruct statistics of velocity better than SR using a single time input data. Generative adversarial networks (GAN) for SR were introduced by Ledig *et al.* [9]. The learning by GAN progresses such that the probability density function (PDF) of the generated data is well superimposed on that of the total data. Therefore, GAN can reconstruct the statistics of the total data. Kim *et al.* [8] showed that though only CNN2d does not reconstruct statistics of velocity and vorticity well, the SR using a kind of GAN, CycleGAN, well preserves these statistics. It is to be noted that their SR based on CycleGAN does not reconstruct strong vortex structures and position of vortices very well (see Fig. 5 in Ref. [8]).

Turbulence exhibits significant spatial and temporal intermittency in particular in the dissipation range. Wavelets, which are well-localized functions in both space and scale, provide a suitable tool to analyse spatially intermittent multi-scale fields [10]. Readers interested in the application of wavelets to turbulence can refer to review articles, e.g., Refs. [11, 12]. Coherent vorticity is extracted out of turbulence, by using orthonormal wavelets, which have a fast transform [13, 14, 15, 16]. It was shown that the coherent vorticity well preserves the turbulence statistics, and consists of few percents of the degrees of freedom of turbulence. Furthermore, wavelet analysis is employed in deep learning. For example, Guo *et al.* [18] proposed an image SR method using CNN2d and orthonormal wavelets, which well reconstruct edges of structures.

In this paper, we propose a SR method based on the orthonormal wavelet transform and 3D convolutional neural networks (CNN3d) for 3D homogeneous isotropic turbulence. CNN3d is a neural network that learns local area of 3D structure via a 3D convolution filter. Taking the results [13, 14, 15, 16], mentioned above, we conjecture that the use of wavelets enhances the efficiency of learning turbulent data in deep learning. We assess the proposed wavelet-based SR method by using flow visualization, PDFs of vorticity, and enstrophy spectra and compare the results obtained without using wavelets.

The remainder of the manuscript is organized as follows. In section 2 we describe the DNS approach and briefly recall wavelet analysis of turbulent flow. A wavelet-based SR method is presented in section 3. Numerical results are given in section 4 and finally conclusions are drawn in section 5.

2 DIRECT NUMERICAL SIMULATION AND WAVELET REPRESENTATION

In the following the governing equations with the DNS approach are briefly described and the orthogonal wavelet decomposition of a 3D vector field is presented.

2.1 GOVERNING EQUATIONS AND NUMERICAL SIMULATION

We consider 3D incompressible homogeneous isotropic turbulence in a periodic cube $\Omega = [0, 2\pi]^3$. The turbulent motion is governed by the Navier–Stokes equations

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

with the divergence-free condition $\nabla\cdot\boldsymbol{u} = 0$. Here, $\boldsymbol{u}(\boldsymbol{x},t)$ is the velocity, $p(\boldsymbol{x},t)$ is the pressure, $\boldsymbol{F}(\boldsymbol{x},t)$ is the external forcing, ρ is the constant density, ν is the constant kinematic viscosity, t is the time,

Table 1: Statistics of the DNS data at different time instants, where R_λ is the Taylor microscale Reynolds number. The interval $t = 1 \approx 0.48T$ for the DNS at 512^3 grid points, where T is a large-eddy turnover time defined as $T = \mathcal{L}/u'$, L is an integral length scale defined as $\mathcal{L} = \pi/(2u'^2)\int_0^{k_{\max}} E(k)/k dk$ at $t = 9$, $E(k)$ is the energy spectrum, and $u' = \sqrt{2\mathcal{E}/3}$, and $\mathcal{E} = 0.5$ here. The resolution of the DNS are either $n_g^3 = 512^3$ or 1024^3 where $k_{\max}\eta \approx 1.0$ and 1.2, respectively. Here, η is the Kolmogorov microscale, and $k_{\max}$ is the maximum wavenumber retained by the DNSs.

	n_g^3	(t, R_λ)
training data	512^3	$(10, 277)$, $(11, 278)$
data 1	512^3	$(9, 279)$
data 2	1024^3	$(12.5, 373)$

$\boldsymbol{x} = (x, y, z)$, and $\nabla = (\partial/\partial x, \partial/\partial y, \partial/\partial z)$. The arguments $\boldsymbol{x}$ and t are omitted at will. The spatial average of $\boldsymbol{u}$ was set to zero.

We carried out a series of DNSs with 512^3 grid points and 1024^3 grid points, using a Fourier spectral method and a fourth-order Runge–Kutta method for time integration. The aliasing errors are removed by a phase shift method and a spherical cutoff filter. The Fourier modes satisfying $k < \sqrt{2}n_g/3$ are retained, where $k = |\boldsymbol{k}|$ is the modulus the wavevector, and n_g is the number of grid points in each Cartesian direction. For the details on the DNS method, readers may refer to e.g., Ref. [19]. The kinematic viscosity is set to $\nu = 2.8 \times 10^{-4}$ for $n_g = 512$, while $\nu = 1.5 \times 10^{-4}$ for $n_g = 1024$. The initial fields of the DNSs at $t = 0$ are divergence-free random velocity fields whose energy spectra $\propto k^2 \exp(-k^2/4)$, and the energy per unit volume, $\mathcal{E}$, is set to 0.5. For forcing, negative viscosity was used as $\boldsymbol{f}$ at $k < 2.5$ such that the energy remains 0.5. We analyse the DNS data after a statistically stationary regime was obtained. Statistics of the different flow simulations are summarized in Table 1.

2.2 WAVELET ANALYSIS

Let $v(\boldsymbol{x})$ be a component of a 3D vector field in a sub-domain $\Omega_s = [L_0, L_0 + L]^3 \subseteq \Omega = [0, 2\pi]^3$, where (L_0, L_0, L_0) is an appropriate position in the periodic box, $0 \le L_0 \le 2\pi$, and $v \in L^2(\Omega)$. The field in Ω_s is sampled on 2^{j_L} grid points in each Cartesian direction, where $j_L (\le J)$ is a natural number, and J is the number of octaves in each space direction in the domain $[0, 2\pi]$ (e.g., $J = 9$ for 512 grid points). It has to be noted that periodic boundary conditions are present at the boundary of Ω. For the sub-domains Ω_s we impose mirror symmetry in each direction of the sub-domain. The wavelet functions $\psi_{\mu,j,\boldsymbol{i}}(\boldsymbol{x})$ yield an orthogonal basis at scale 2^{-j}, where $\mu (= 1, \cdots, 7)$ denotes directions and $\boldsymbol{i} = (i_x, i_y, i_z)$ denotes the position.

The field $v(\boldsymbol{x})$ can then be decomposed into an orthogonal wavelet series through the application of a periodization or folding technique [10]:

$$v(\boldsymbol{x}) = \langle v \rangle + \sum_{j=0}^{J-1} v_j(\boldsymbol{x}), \tag{2}$$

$$v_j(\boldsymbol{x}) = \sum_{\mu=1}^{7} \sum_{i_1, i_2, i_3 = 0}^{2^j - 1} \langle v, \psi_{\mu,j,\boldsymbol{i}} \rangle \psi_{\mu,j,\boldsymbol{i}}(\boldsymbol{x}), \tag{3}$$

where $\langle\cdot\rangle = \int_{\Omega_s} \cdot d\boldsymbol{x}$, and $\langle\cdot,\cdot\rangle$ denotes the L^2-inner product defined as $\langle g_1, g_2\rangle = \int_{\Omega_s} g_1(\boldsymbol{x})\, g_2(\boldsymbol{x}) d\boldsymbol{x}$. Note that $\langle \boldsymbol{v}_j \rangle = \boldsymbol{0}$. Using the scaling function $\phi_{j,\boldsymbol{i}}(\boldsymbol{x})$ based on $\phi(\boldsymbol{x})$, we obtain

$$\sum_{m=0}^{j-1} v_m(\boldsymbol{x}) = \sum_{i_1,i_2,i_3=0}^{2^j-1} \langle v, \phi_{j,\boldsymbol{i}}\rangle \phi_{j,\boldsymbol{i}}. \tag{4}$$

At scale 2^{-j} we have 7×2^{3j} wavelet coefficients, $\langle v, \psi_{\mu,j,\boldsymbol{i}}\rangle$, and 2^{3j} coefficients, $\langle v, \phi_{j,\boldsymbol{i}}\rangle$. Here we use Coiflet 12 wavelets which have filter length 12, 4 vanishing moments and compact support. In PyWavelets [17] which we use here they are denoted by 'coif2'.

3 A WAVELET BASED SUPERRESOLUTION METHOD

We develop a SR methodology for 3D homogeneous isotropic turbulence, using CNN3d and orthogonal wavelet analysis. We denote the methodology WCNN3d. In the following, we describe the training data and its preprocessing in section 3.1, and then present the procedure of WCNN3d in section 3.2. In section 3.3 we discuss the output data after the training.

3.1 TRAINING DATA

In the training of SR, we use two snapshots of x-components of high-resolution (HR) DNS data and low-pass filtered HR data (LF data). The HR data are the DNS data at 512^3 grid points described in Sec .2, while the LF data computed at 512^3 are obtained by application of an isotropic Gaussian filter to the HR DNS data. The low-pass filter is given by $\widehat{G}(k) = \exp\left\{-\pi^2 k^2/(24k_c^2)\right\}$, where we set k_c to be 64 for the training data and the data 1 computed at 512^3. For resolution 1024^3, data 2 (cf. Table 1), we use $k_c = 128$. We consider here three types of the training data, called type 1, type 2 and type 3: type 1 uses HR and LF data of ω_x, the x-component of vorticity $\boldsymbol{\omega}(=\nabla\times\boldsymbol{u})$, while type 2 and type 3 use HR and LF data of u_x, the x-component of velocity $\boldsymbol{u}$.

In the type 1 of WCNN3d which learns HR and LF data from ω_x, we divided the periodic box data with resolution 512^3 into 16^3 subcubes with 32^3 grid points, having side length $L = \pi/8$. Each subcube is decomposed into a 3D orthogonal wavelet series by using FWT only on one level. We then obtain wavelet coefficients in seven directions and scaling coefficients on the coarser level. For the analysis of the subcube data we used the option 'symm' for symmetric boundary conditions in PyWavelets. This option needs 5 extra elements of the array in each direction, and thus the number of the elements for the subcubes are finally 21^3. For simplicity type 1 is denoted here by 'WCNN3d(ω)'.

Type 2 learning uses HR and LF data of u_x and we first decompose u_x given in a periodic box into 3D orthogonal wavelet series using the option 'periodization'. After multiplying with the weight 2^j at scale 2^{-j} for all j, we construct a weighted field by using the inverse FWT (IFWT). Then, we divide the weighted velocity into the subcubes at 32^3 grid points, and then follow the procedure of type 1. Type 2 is denoted by 'w-WCNN3d(u)'. The weight 2^j comes from a representative wavenumber of wavelets at scale 2^{-j} and is motivated by the norm equivalence of wavelets [10]. Weighting the wavelet coefficients of the velocity with 2^j yields gradient information of the velocity and is thus comparable with the vorticity-based type 1 approach, which only uses the anti-symmetric part of the velocity gradient.

To compare with w-WCNN3d(u), we also consider type 3 learning HR and LF data of u_x where in contrast to type 2 no weight is imposed. The remaining procedure follows the one of type 1. Type 3 is denoted here by 'WCNN3d(u)'.

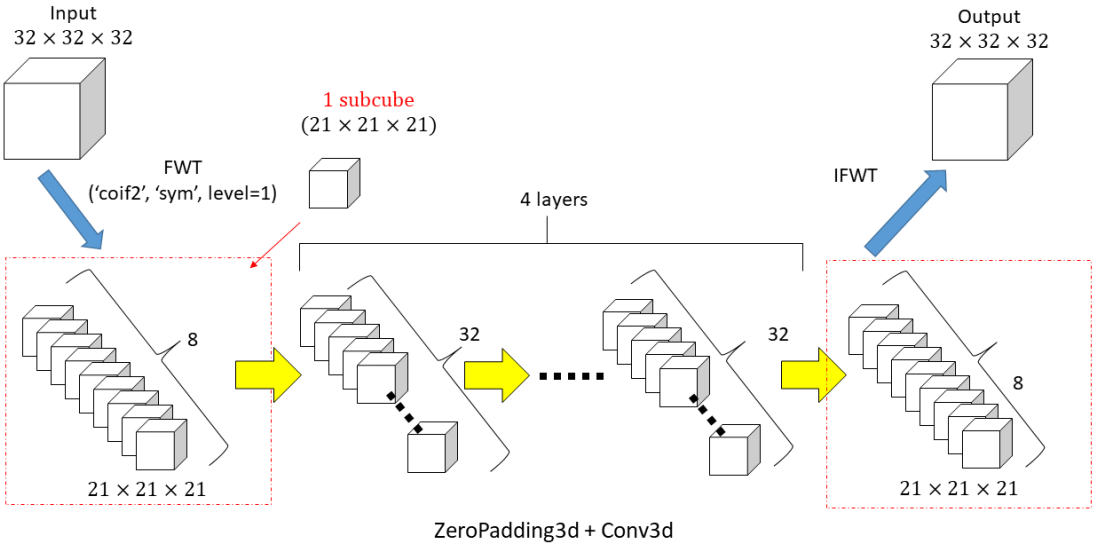

Figure 1: Illustration of SR network structure.

3.2 MACHINE LEARNING MODEL

We illustrate the procedure of WCNN3d in Fig. 1. The training scheme is implemented using the TensorFlow open-source library together with Python 2.7.5 interaction interface. We use Conv3D layer (keras.io) for 3D CNN. Our deep learning model has six layers; one input layer, four middle layers and one output layer. In these layers, we use 3D kernels with their sizes being $3 \times 3 \times 3$ in order to catch local features of the turbulence data. The input and the middle layers consist of thirty-two filters of $3 \times 3 \times 3$ 3D kernels. The output layer consists of eight filters of $3 \times 3 \times 3$ kernels. Zero-padding is employed at each layer, to keep the size of the output subcubes' data being the same as those of the input subcubes' data. As an activation function, we use a leaky rectified linear unit (ReLU) defined as

$$f(x) = \begin{cases} x & (x \geq 0), \\ \alpha x & (x < 0), \end{cases} \tag{5}$$

where $\alpha = 0.2$. The error was measured by the use of the mean-square of the difference between the output and correct data. The L^2-regularization was used to suppress overfitting. Here, the Adam optimizer [20] is used. Our batch sizes are 32 and the number of epochs is set to 30.

To show the influence of the use of the wavelets, we also trained a CNN3d without using wavelets. In this CNN3d, the training is performed in physical space.

3.3 TEST DATA AND OUTPUT DATA

After the training was completed, we input the LF data of the DNS data at another time instant (another snapshot), corresponding to data 1 and data 2 in Table 1, to the model described in section 3.2. It has to be noted that we use x, y and z components of either the LF vorticity or the LF velocity, which depends on what type of the learning model we use as input data. In type 1, WCNN3d(ω), and CNN3d without

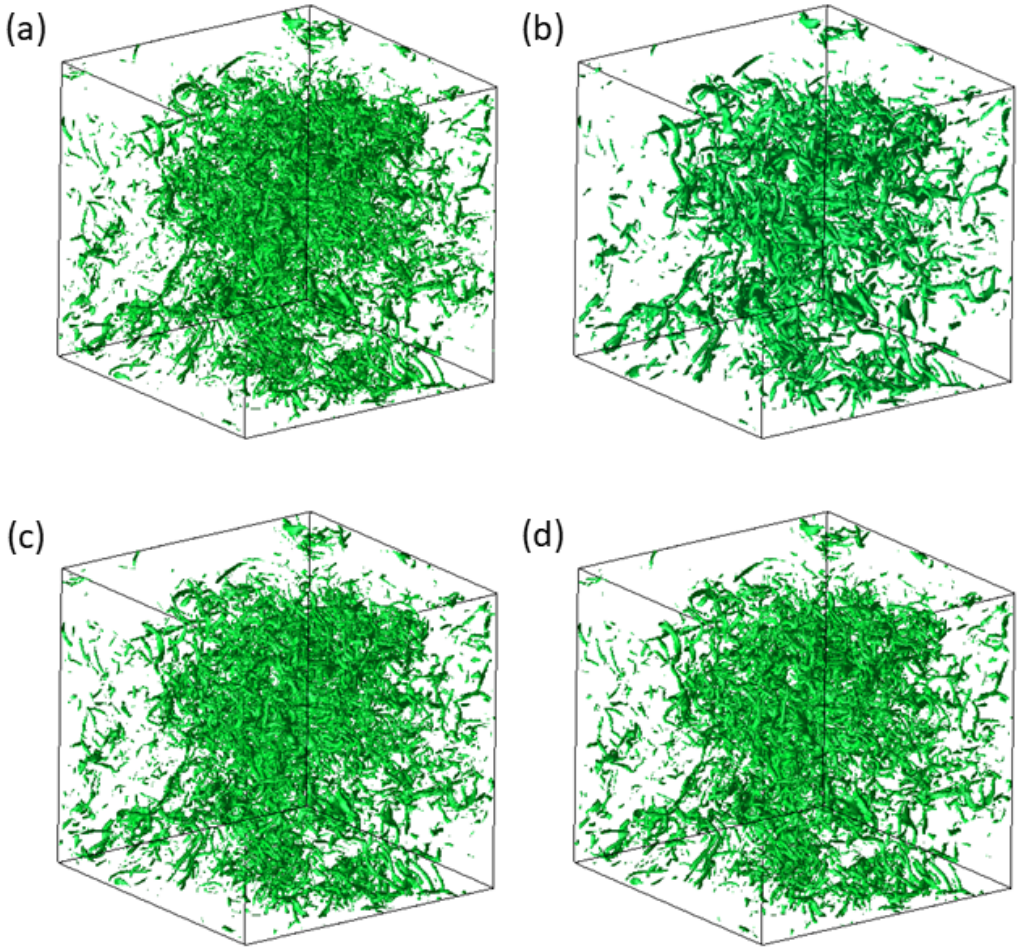

Figure 2: The visualization of the modulus of vorticity $|\omega|$ for the isosurface, $m+3\sigma$, for subcubes at size 256^3: (a) total data, (b) LF data, (c) field predicted by WCNN3d(ω), and (d) field predicted by w-WCNN3d(u). Here, m and σ are the mean value and the standard deviation of the corresponding vorticity modulus $|\boldsymbol{\omega}|$.

wavelets, we use $\boldsymbol{\omega}$, while in type 2, w-WCNN3d(u), and type 3, WCNN3d(u), we use $\boldsymbol{u}$. Then, our deep learning model predicts the eight sets of the coefficients, i.e., 7 directions of wavelet coefficients and the scaling coefficient. Using IFWT, we obtain 16^3 subcubes with 32^3 grid points in physical space. All of the subcubes are subsequently merged into the data yielding a flow field with 512^3 grid points. In w-CNN3d(u), after the merging, we apply the FWT, divide the resulting wavelet coefficients by 2^j for all j, and then use IFWT to get the output data. Note that no thresholding is applied in the use of wavelet decomposition.

Moreover, we use the DNS data at higher Re_λ with 1024^3 grid points, cf. data 2 in Table 1. We input the LF data of data 2 where $k_c = 128$ to our models, WCNN(ω) and w-WCNN(u). The subcube size is 32^3 which is the same as what our model used for data 1. For the high resolution case we show here only the results using w-WCNN(u). WCNN(ω) shows better results than w-WCNN(u).

Table 2: Mean values m and standard deviation σ for the total data, the LF data, the field predicted by WCNN3d(ω), and the field predicted by w-WCNN3d(u).

	m	σ
total	12.8	10.4
LR	8.9	6.37
WCNN3d(ω)	12.5	10.0
w-WCNN3d(u)	12.3	9.66

4 NUMERICAL RESULTS

Figures 2(a), (b), (c) and (d) respectively show the visualization for the total data, i.e., the HR field of data 1, the LF field of data 1, the field predicted by WCNN3d(ω), and the field predicted by w-WCNN3d(u). The visualizations show isosurfaces satisfying $|\omega| = m + 3\sigma$ in a subcube with 256^3 grid points, where m and σ are the mean and the standard deviation of $|\omega|$, respectively. The corresponding values are presented in Table 2. In Fig. 2(a), the total field, we can see well-pronounced vortex tubes. As expected the LF field, shown in Fig. 2(b), exhibits much less tubes, which is due to the application of the low-pass filter to the total field. For the predicted vorticity fields using either WCNN3d(ω) or w-WCNN3d(u), it can be observed in Figs. 2(c) and (d) that the visualizations of the reconstructed fields are in good agreement with those of the total field. These observations are confirmed by 2D cuts of ω_x in the plane $z = 0$, shown in Fig. 3. The effect of the low-pass filtering is nicely seen in Fig. 3(b), where the scale of the coherent structures seems larger than that in Fig. 3(a). In contrast, the structures in the predicted field, shown in Figs. 3(c) and (d), agree well with those in the total field, shown in Fig. 3(a). Hence we can conclude from the 2D and 3D visualizations in Figs. 2 and 3 that the deep learning model recovers the structure of the total field. We remark that in our models, we divide the data in the periodic box into subcubes, as mentioned in section 3.2. However, in the visualizations shown in Figs. 2 and 3 the influence of the division seems negligible.

To quantify the above observations statistically, we computed the PDFs of ω_x and enstrophy spectra. In this paper, we focus on small-scale behavior which can be super-resolved, and hence we consider the statistics of $\boldsymbol{\omega}$ instead of $\boldsymbol{u}$. Figure 4 shows the PDFs of ω_x and the enstrophy spectrum $Z(k)$ for the for the total vorticity, the LF vorticity, the vorticity predicted by WCNN3d(ω) using data 1 with resolution 512^3. For comparison we also performed SR of the LF data using CNN3d without wavelets. In Fig. 4(a), we observe that the PDF of the LF data is much narrower than the total one, which is expected and in accordance with the values of σ in Table 2. We can also see that the PDF of the vorticity reconstructed by WCNN3d(ω) well overlaps with the total PDF. For CNN3d without using wavelets, the PDF of ω_x is somewhat skewed, though the PDF seems to overlap the total PDF well. From the enstrophy spectra in Fig. 4(b), we can deduce that WCNN3d(ω) well preserves $Z(k)$ of the total field, though there is a small departure of the predicted spectrum from the total one for $k\eta \gtrsim 0.7$. Concerning $Z(k)$ for CNN3d, it can be seen that the enstrophy spectrum has somewhat smaller values than the total one. It can be noted that the pile up of $Z(k)$ for the total field around $k\eta = 1$ is due to the use of the Fourier pseudo-spectral method and its resolution. From the enstrophy spectra we can conclude that the use of the wavelets enhances the deep learning of turbulent flows at least for the case considered here.

The PDFs of ω_x and the enstrophy spectra $Z(k)$ for the cases w-WCNN3d(u) and WCNN3d(u) are presented in Fig. 5 together with the total field and LF. We find that w-WCNN3d(u) preserves the total

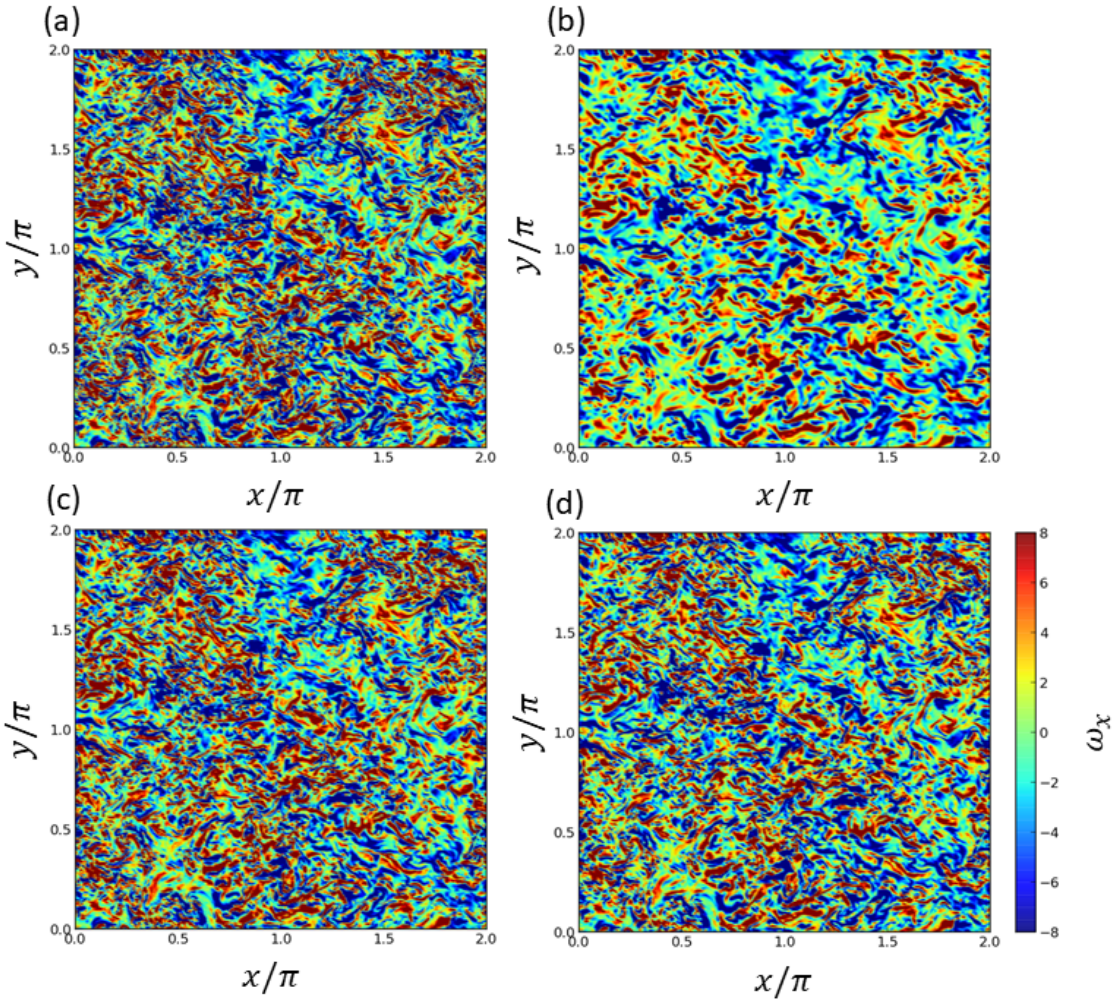

Figure 3: Visualization of the x-component of vorticity ω_x in a 2D cut at $z = 0$: (a) the total data, (b) the LF data, (c) the field predicted by WCNN3d(ω), and (d) the field predicted by w-WCNN3d(u).

PDF and the total enstrophy spectrum much better than WCNN3d(u). This shows that the weight 2^j in w-WCNN3d(u) is essential to our SR using the LF velocity field. Small magnitude of spikes in $Z(k)$ at high k range can be seen for w-WCNN3d(u) and WCNN3d(u). These spikes are probably due to the division of the total field into the subcubes.

Strictly speaking, the output fields generated by our model are not divergence-free. We thus examine whether the divergence of $\boldsymbol{\omega}$ predicted by WCNN3d(ω) and the divergence of $\boldsymbol{u}$ predicted by w-WCNN3d(u) are negligible or not. On the basis of the Helmholtz decomposition, these fields are decomposed into $\boldsymbol{\omega} = \nabla \times \boldsymbol{A}_\omega + \nabla\theta_\omega$ and $\boldsymbol{u} = \nabla \times \boldsymbol{A}_u + \nabla\theta_u$, where $\boldsymbol{A}_\omega$ and $\boldsymbol{A}_u$ are vector potentials and θ_ω and θ_u are scalar potentials. The divergent parts are given by $\nabla\theta_\omega$ and $\nabla\theta_u$. Figure 6 shows the spectrum of $\nabla\theta_\omega$ and the k^2 weighted spectrum of $\nabla\theta_u$. The latter means that the spectrum of $\nabla\theta_u$ is multiplied by k^2. We find that these spectra have values much smaller than the enstrophy spectrum $Z(k)$ predicted by WCNN3d(ω) and also the spectrum $k^2E(k)$ predicted by w-WCNN3d(u). Here, $E(k)$ is the

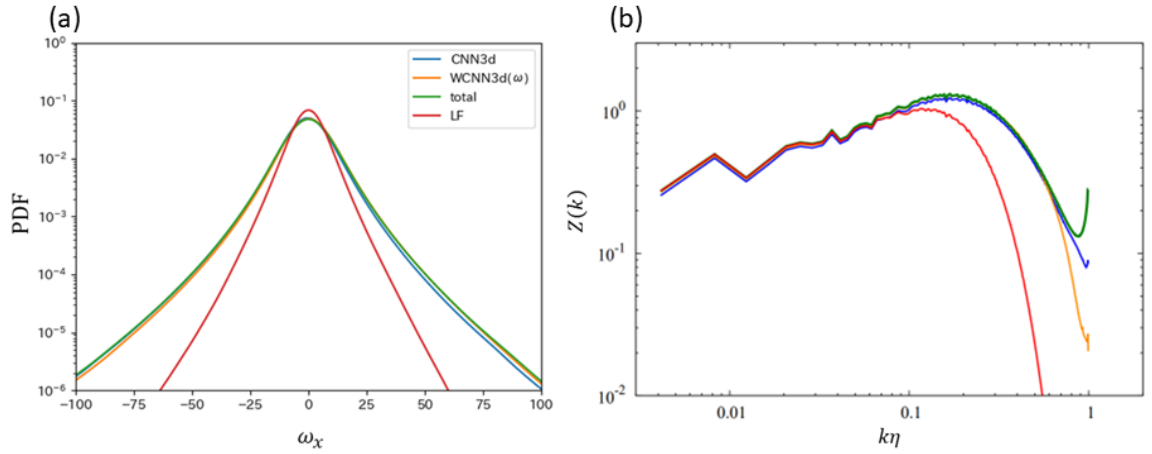

Figure 4: (a) PDFs of ω_x and (b) enstrophy spectra $Z(k)$ for the total data, the LF data, the field predicted by WCNN3d(ω) together with the field predicted by CNN3d without wavelets. The wavenumber k is normalized by η of the total field.

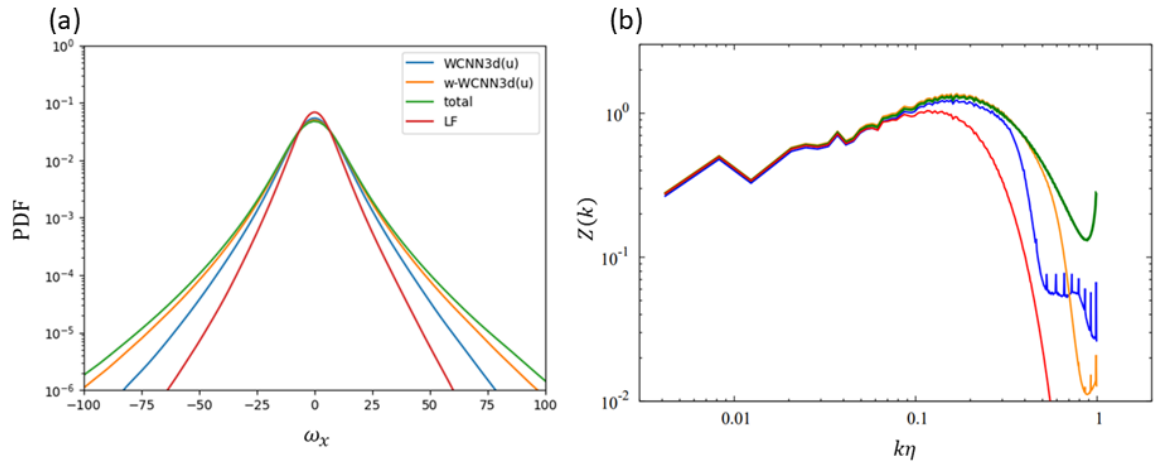

Figure 5: (a) PDFs of ω_x, and (b) enstrophy spectra $Z(k)$ of the field predicted by w-WCNN3d(u), and the field predicted by WCNN3d(u), together with the total data and the LF data. The wavenumber k is normalized by η of the total field.

energy spectrum predicted by w-WCNN3d(u). Therefore, the influence of the divergence is negligible. In addition, we observe some spikes in the spectra of the divergence parts. We confirm that the influence of the subcube division is likewise negligible, as already mentioned above.

Furthermore, we evaluate whether our model w-WCNN3d(u), which was obtained by using the training data with 512^3 cubes, can well super-resolve the LF data obtained by the application of the Gaussian filter to data 2 computed with resolution 1024^3 (see the end of section 3.3). In Fig. 7(a), we can observe that the PDF of ω_x predicted by w-CNN3d(u) is somewhat narrower than that of the total field. However, the enstrophy spectrum is well reconstructed by the SR, as shown in Fig. 7(b). We also confirmed that

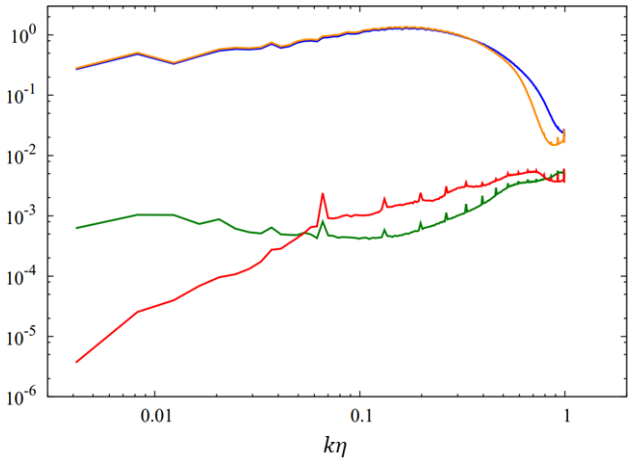

Figure 6: The spectrum of the divergence part of the vorticity field predicted by WCNN3d(ω) (green), and the k^2-weighted spectrum of the divergence part of the velocity field predicted by w-WCNN3d(u) (red). For comparison, $Z(k)$ for WCNN3d(ω) (blue) and $k^2E(k)$ for w-WCNN3d(u) (orange) are also plotted. The wavenumber k is normalized by η of the total field.

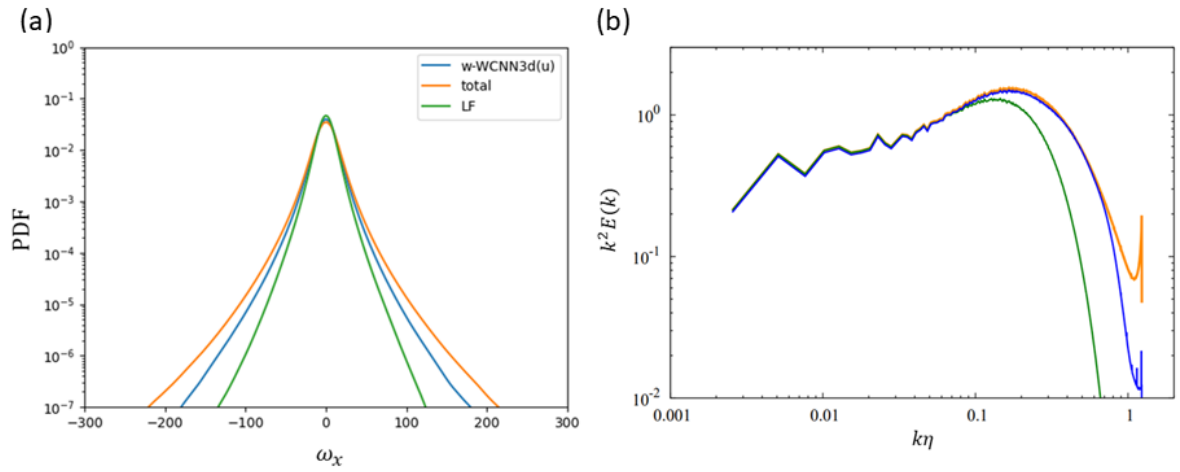

Figure 7: The results using the LF data of data 2 with 1024^3 grid points as input data. (a) PDFs of ω_x, and (b) the enstrophy spectra $Z(k)$ for the total data, the LF data, and the field predicted by w-WCNN3d(u). The wavenumber k is normalized by η of the total field of data 2.

the divergence issue is not crucial (figure omitted for brevity).

5 CONCLUSIONS

We have developed wavelet-based 3D CNN (WCNN3d) for SR of coarse-grained data of homogeneous isotropic turbulence. Different approaches have been proposed, vorticity- and velocity-based models using orthogonal wavelets. For the later we also investigated a version with weighted wavelet coefficients which yields velocity gradient information. As input data of the WCNN3d, we used DNS data

of isotropic turbulence in a periodic box and the coarse-grained data are obtained by the application of a Gaussian low-pass filter to the DNS data. We assessed the WCNN3d in terms of 3D visualization of vorticity, PDF of vorticity, and enstrophy spectra. We found that WCNN3d well reproduces vorticity statistics and the positions of the vortices from coarse-grained vorticity fields. For the vorticity-based approach, the use of wavelets enhances deep learning of turbulent flows considered here thanks to the sparsity of the wavelet representation which improves deep learning. For the velocity-based approach, we showed that weighting the wavelet coefficients of velocity, which yields velocity gradient information due to norm equivalence, improves the accuracy and yields results similar to the vorticity-based model. Furthermore, we assessed the divergence issue of the predicted fields and showed that its impact is negligible. We demonstrated the capability to predict a turbulent flow whose Reynolds number is higher than the flows used for the training.

In this paper, our emphasis is put on SR when the intensity of data in wavenumber space is reduced, especially in the high wavenumber range. It would be interesting whether the use of wavelets can improve performance of the SR for the data where the modes in a high wavenumber range are completely cut off. However, this is beyond the scope of this study and remains an issue to be pursued.

Acknowledgement The computations presented here were carried out on the supercomputer "Flow" at Information Technology Center, Nagoya University. KS acknowledges partial support by the French Federation for Magnetic Fusion Studies (FR-FCM) and the Eurofusion consortium, funded by the Euratom research and training programme 2021-2022 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

REFERENCES

[1] Duraisamy, K., Iaccarino, G. and Xiao H., *Annu. Rev. Fluid Mech.* (2019) **51**:357–377.
[2] Brunton, S. L., Noack, B. R. and Koumoutsakos, P., *Annu. Rev. Fluid Mech.* (2020) **52**:477–508.
[3] Pandey, S., Schumacher, J. and Sreenivasan, K. R., *J. Turbul.* (2020) **21**:567–584.
[4] Dong, C., Loy, C., C., He, K., Tang, X., *Computer Vision, ECCV* (2014):184–199.
[5] Onishi, R., Sugiyama, D. and Matsuda, K., *SOLA* (2019) **15**:178–182.
[6] Fukami, K., Fukagata, K. and Kunihiko, T., *J. Fluid Mech.* (2019) **870**:106–120.
[7] Liu, B., Tang, J., Huang, H and Lu, X., *Phys. Fluids* (2020) **32**:025105.
[8] Kim, H., Kim, J., Won S. and Lee, C., *J. Fluid Mech.* (2021) **910**:A29.
[9] Ledig, C., Theis, L., Huszar, F, Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. and Shi, W., *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017):105-114.
[10] Mallat, S., *A Wavelet Tour of Signal Processing: The Sparse Way*, Third Edition, Academic Press (2009).
[11] Farge, M., *Annu. Rev. Fluid Mech.* (1992) **24**:395–457.
[12] Schneider, K. and Vasilyev, O., *Annu. Rev. Fluid Mech.* (2010) **42**:473–503.
[13] Farge, M., Pellegrino, G. and Schneider, K., *Phys. Rev. Lett.* (2001) **87**:45011–45014.
[14] Farge, M., Schneider, K., Pellegrino, G, Wray, A. and Rogallo, R., *Phys. Fluids* (2003) **15**:2886–2896.
[15] Schneider, K., Farge, M., Pellegrino, G., and Rogers, M., *J. Fluid Mech.* (2005) **534**:39–66.
[16] Okamoto, N., Yoshimatsu, K., Schneider, K., Farge, M. and Kaneda, Y., *Phys. Fluids* (2007) **19** :115109.

[17] Lee, G. R., Gommers R., Wasilewski F., Wohlfahrt K., O'Leary A., J. Open Source Software (2019) **4**:1237.
[18] Guo, T., Mousavi, H. S., *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017):1100–1109.
[19] Kaneda, Y, Ishihara, T, Yokokawa, M., Itakura, K. and Uno, A., *Phys. Fluids* (2003) **15**:L21–24.
[20] Kingma, D., and Ba, J., arXiv:1412.6980 (2017).