Center for Turbulence Research
Proceedings of the Summer Program 2022

Neural networks for synthesizing preferential concentration of particles in isotropic turbulence

By T. Oujia[†], S. S. Jain, K. Matsuda[¶], K. Schneider[†], J. West AND K. Maeda

Cluster and void formation are key processes in the dynamics of particle-laden turbulence. We propose different data–driven machine learning techniques for synthesizing preferential concentration fields. The database of direct numerical simulation (DNS) of homogeneous isotropic turbulence with one-way coupled inertial point particles is used to input the enstrophy and predict the particle number density fields. We compare autoencoder, U–Net, and generative adversarial network (GAN) approaches and assess the statistical properties of the generated particle number density fields. We find that the GANs are superior in predicting clusters and voids, and therefore result in best performance. This yields interesting perspectives for reducing the computational cost of expensive DNS computations by avoiding the tracking of billions of particles. We also explore the inverse problem of synthesizing the enstrophy fields using the particle density distribution as the input at different Stokes numbers. Hence, our study also provides perspectives to use neural networks to predict turbulence statistics using experimental measurements of inertial particles.

1. Introduction

Various tools and techniques from machine learning are becoming increasingly popular in the field of fluid mechanics and turbulence in particular (Duraisamy *et al.* 2019; Brunton *et al.* 2020). This rapidly evolving field is changing continuously, and evaluating the potential and capabilities, in particular of the deep neural networks, is critical to assess the benefits for science and engineering in improving and maximizing the performance. In particular, for particle-laden flow, Siddani *et al.* (2021) used a combination of convolutional neural networks and generative adversarial neural networks to recreate particle-resolved flow fields around a random distribution of particles. Machine learning was also used as described in Faroughi *et al.* (2022) to predict the drag coefficient of spherical particles translating in viscoelastic fluids and in Hwang *et al.* (2021) to model the forces on the nonspherical and irregular particles in low–Reynolds number flows. For synthesizing data, a method using harmonic wavelet phase covariance has shown high-quality results in modeling geometric structures in turbulent flows (Zhang & Mallat 2021) and inertial particle in turbulence (Brochard *et al.* 2022).

These and other previous studies use machine learning to develop models for the fine-scale dynamics of flows and particles from the macroscopic scale of information, while the present work is motivated to use machine learning to inform on the mesoscale clustering of inertial particles in turbulence from flow field data without information on an individual particle's position and velocity. Clusters of particles and void regions are characteristic

[†] Institut de Mathématiques de Marseille, Aix-Marseille Université, CNRS, France
[¶] Japan Agency for Marine-Earth Science and Technology (JAMSTEC), Japan

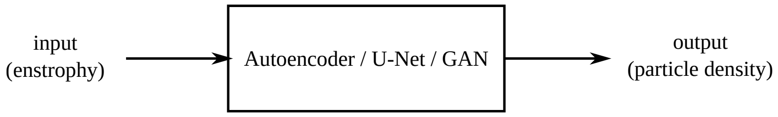

FIGURE 1. A schematic illustrating the machine learning model.

features of particle-laden turbulent flows, and their prediction and understanding are essential in various industrial applications, for instance, for modeling radiative heat transfer in particle-laden solar receivers and estimating the attenuation of radar signals in clouds.

Measuring individual particles is typically impractical. In simulations, it can be time-consuming to compute the trajectory of billions of particles. Hence, it is imperative to develop an alternative low-cost technique to predict particle distributions with various parameters, e.g., increasing the number of particles or changing the particle inertia. Prediction of particle distributions with the same parameters as given data is the first step. The goal of this work is then to develop and test machine learning tools for this task, which is illustrated in Figure 1. To this end, three different neural networks, autoencoder, U–Net, and generative adversarial network (GAN), are trained using DNS data. The aim is to synthesize one-way coupled particle densities for different Stokes numbers from snapshots of enstrophy distributions of homogeneous isotropic turbulence at high Reynolds number, obtained by three-dimensional high-resolution DNS (Matsuda *et al.* 2014). The results are then compared against the point-particle DNS data and statistically assessed. In return for high-fidelity training data, which is expensive, the present results may contribute to the development of efficient techniques that avoid expensive tracking of billions of particles in DNS computations.

A complementary approach, which is also considered here, is to invert the above procedure, i.e., particle densities are used as input for generating different flow quantities, e.g., enstrophy, as output that is of some interest for experimentalists. In fact, the enstrophy is directly related to the energy dissipation, which is a quantity of major interest. In the future, we also plan to focus on the prediction of vorticity, velocity, and pressure and more fundamental quantities, such as Reynolds and Stokes numbers.

The remainder of the report is organized as follows. Section 2 presents the neural network methodology together with details on the flow data set. Results are given in Section 3, followed by the conclusion and perspectives for future work in Section 4.

2. Methodology

In this section, we provide details on the data set used as well as a description of the neural network architecture and training. We assess different neural networks in this work, and the motivation behind using different networks is to assess if more complex architectures yield improved results by generating fields that have better quality with respect to the DNS data. However, a trade-off is certainly necessary in terms of computational cost and accuracy. The enstropy, square norm of vorticity, is used as the input to predict particle density fields. A physical explanation behind using enstrophy as the input quantity to generate the particle density fields is due to the dynamics of inertial particles in turbulence. Particles are subjected to centrifugal force in vortical regions, and therefore such regions, corresponding to high enstrophy, are characterized by lower particle density; straining regions, corresponding to low enstrophy, are characterized by larger particle density. The density distribution strongly depends on the inertia of the particles, which is quantified by the Stokes number. Other quantities related to particle dynamics, such as divergence or curl of the particle velocity, could be likewise predicted. But in this work, we choose to focus on density, which is a more fundamental quantity.

2.1. *Data set description*

The underlying data set consists of particle position and velocity data generated by DNS of particle-laden homogeneous isotropic turbulence, detailed in Matsuda *et al.* (2014). The incompressible Navier–Stokes equations are solved in a 2π-periodic cube with a fourth-order finite–difference scheme. A large-scale forcing is applied to obtain a statistically stationary flow. Uniformly distributed discrete particles are then injected into the fully developed flow and are tracked in the one–way coupled Lagrangian framework. Maxey's model (Maxey 1987) for inertial heavy point particles with Stokes drag is used, and the inertial dynamics is represented by the Stokes number, $St = \tau_p/\tau_\eta$, where τ_p is the particle relaxation time and τ_η the Kolmogorov time. The effect of gravitational settling is neglected.

State-of-the-art high-resolution DNS with $N_g^3 = 512^3$ grid points is performed for the Taylor-microscale Reynolds number $Re_\lambda = 204$, where $Re_\lambda \equiv u'\lambda/\nu$, ν is the kinematic viscosity, and λ is the Taylor microscale. The number of particles N_p is 1.5×10^7 and the considered Stokes numbers are $St = 0.05$, 0.1, 0.2, 0.5, and 1, each corresponding to a different data set. Particles with different Stokes numbers were tracked in an identical turbulent flow.

We give as input the enstrophy in log scale in order to help the neural network to converge faster by avoiding a large variation in the input values. The input data are 2D enstrophy fields; i.e., we consider 2D slices from the 3D periodic simulation domain for reasons of limited memory available on the GPU. We are also interested to do the inverse, i.e., predicting the enstrophy from the particle density for different Stokes numbers. We use 256 slices in each axis direction, i.e., a total of 3×256 slices. Each slice is obtained by merging two consecutive slices of the simulation data, and therefore a total of 256 slices are generated in each axis direction, with the thickness of each of the slices being $2\Delta x$ (where $\Delta x = 2\pi/N$, $N = 512$). This is done to have a sufficient number of particles in each slice to compute the corresponding density required for the training of the networks. Moreover, we artificially increase the size of the training data four times by transposing the 2D slices and by rotating the original and transposed slices. This yields, for one 3D simulation snapshot, in total a data set of $3 \times 256 \times 4 = 3072$ 2D images, with a resolution of 512×512 each. One snapshot is used for training and another one for computing the results, which is presented in the following.

2.2. *Description of the neural network architecture and training*

Figure 2 shows the different architectures we have tested to synthesize the number density distribution of the inertial particles for different Stokes numbers. We compare three different neural networks: autoencoder, U–Net, and GAN, each of which is a modification of the standard architecture, adapted here for the current application. An autoencoder (Rumelhart *et al.* 1986) is an artificial neural network composed of two neural networks, an encoder and a decoder. The encoder compresses/reduces the size of the data, while the decoder reconstructs the desired output using the compressed data. The autoencoder is the baseline network for this study and is represented by the blocks and arrows in blue on Figure 2. The loss function used is the binary cross-entropy between the exact and predicted values. A U–Net (Ronneberger *et al.* 2015) is an autoencoder with additional skip connections between the layers of the encoder and the decoder, which have the same shape. Skip connections bypass some of the layers in the neural network to allow

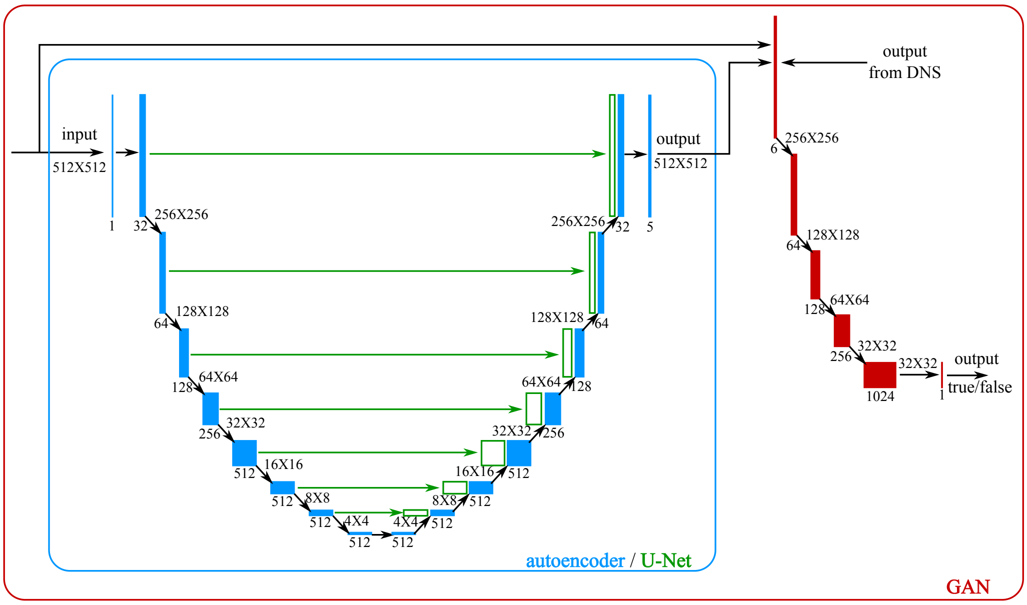

FIGURE 2. Illustration of the different neural network architectures: autoencoder (blue), U–Net (blue and green) and GAN (blue, green and red).

a connection between two distant layers, contrary to the standard connection, which are only connected to the next layer. The U–Net is represented in Figure 2 as the blue part (autoencoder), to which we have added skip connections in green. We use the same loss function as in the autoencoder. A GAN (Goodfellow *et al.* 2020) is an artificial neural network composed of two neural networks, a generator and a discriminator. The generator learns to generate new data set, and the discriminator classifies if the input data is real or generated. The goal of the generator is to generate data that will be classified as real by the discriminator. We use a U–Net as the generator, as is done in Isola *et al.* (2017). We give as input to the discriminator the enstrophy and the synthetic or exact particle density. For the discriminator, we use the binary cross-entropy between the correct and the exact classifications (generated or real image). For the generator, we use the binary cross-entropy between the exact and the predicted values to which we add the loss of the discriminator in case we try to fool it. Finally, the GAN is represented in the figure as the previous architecture to which we have added a discriminator in red.

Convolution blocks are composed of a 2D convolution using a filter of size 3×3, a batch normalization, and then a ReLU activation function. Zero padding is added so that the resolution does not change. We repeat this a second time, with the exception that the convolution has a stride of two to change the data resolution. In the case of the decoder, we use transposed convolutions. In order to encode the periodicity of the domain, we copy from each side 8 pixels to the opposite side and apply a convolution of size 5×5 without zero padding until we obtain a power of two.

We used the Adam optimizer for training the networks (Kingma & Ba 2014). Training is done on 100 epochs on a GPU Nvidia Tesla V100 32GB. The number of epochs was chosen to ensure the convergence of the different neural networks tested. The training time is 71 ms/step, 75 ms/step, and 135 ms/step for the autoencoder, U–Net, and GAN, respectively. We observe that there is no significant difference in the computation time

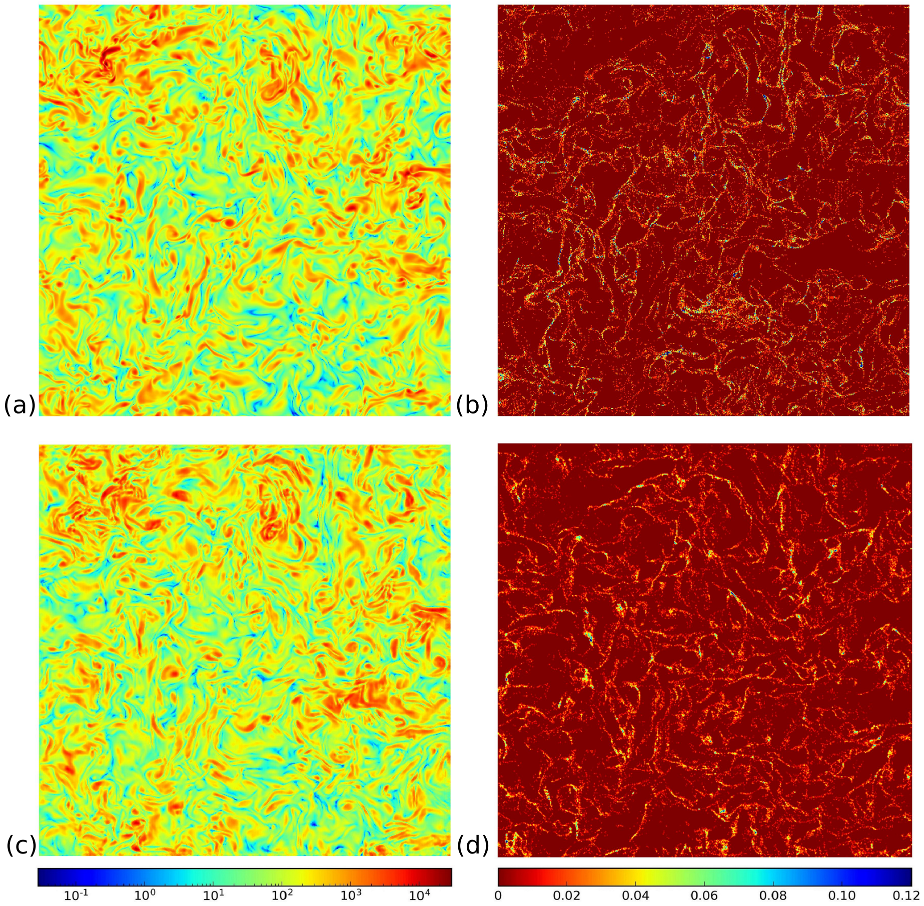

FIGURE 3. Ground truth: (a) enstrophy and (b) particle density corresponding to DNS data. Predicted fields using GAN: (c) enstrophy predicted from particle density and (d) particle density predicted from enstrophy.

for the autoencoder and the U–Net; however, the computation time is doubled for the GAN.

3. Results

In this section, we present results for different networks. We first analyze the networks using 2D visualizations in physical space and then assess them statistically, i.e., we present PDFs and energy spectra and compare them with the DNS data. The presented results were obtained from the validation data set disjoint from the training data set, i.e., another snapshot.

3.1. *Some visualizations*

Figures 3(a) and (b) shows respectively, 2D slices of the enstrophy and particle density for $St = 1$ obtained from the DNS. In the enstrophy field, we can observe vortex stretching and strong variation between regions of extreme rotation and other regions where the fluid velocity is transported. From the particle density field, we can see that the particles

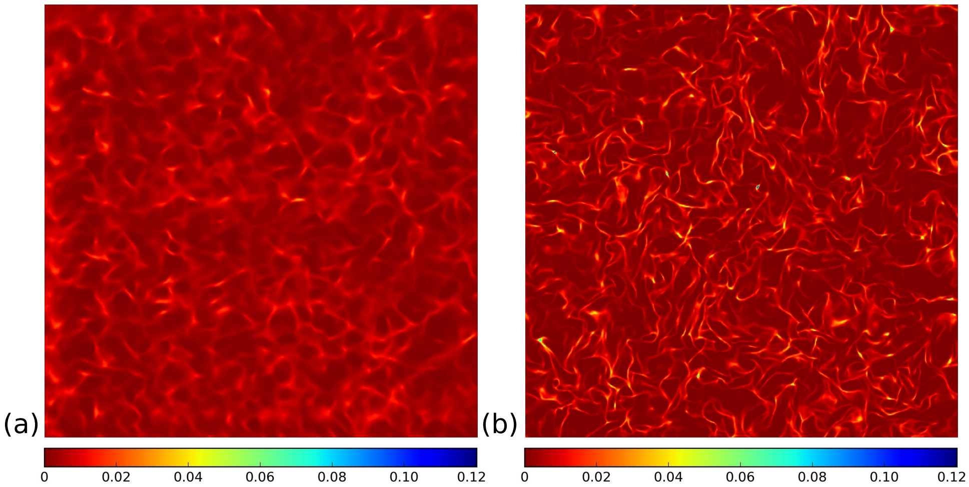

FIGURE 4. Particle density for $St = 1$ predicted from enstrophy using (a) autoencoder and (b) U–Net.

are clustered in the regions of low enstrophy and avoid regions of high enstrophy due to the centrifugal force. Note that some of the void regions do not exactly correspond to high enstrophy regions, and the void regions are larger than the regions of high enstrophy. This could be because we are looking at the 2D slices of the particle density field and the enstrophy field without considering the history effects. For a better understanding of the correspondence between the particle density field and enstrophy, the full 3D fields need to be looked at; however, 2D slices will not give a full picture, because there could be adjacent slices with different values of enstrophy that could influence the particle density in the current slice due to the history effects. This also justifies the need for a 3D network, which will be explored in the future.

Figures 3(c) and (d) show, respectively, the predicted enstrophy and particle density fields for $St = 1$ using the GAN. For the predicted enstrophy, we can observe that in some regions we have very good agreement between the DNS enstrophy and the predicted enstrophy; however, in other regions, the agreement is not very good. For large regions of high enstrophy, the particle density can be lower than expected, which could explain the inability of the neural network to reconstruct the enstrophy in these regions. The prediction of the particle density field using the GAN is overall satisfactory, i.e., a lower density of particles in the regions of high enstrophy is seen as expected. However, it also predicts particles in some regions where there were no particles in the DNS field. The predicted particle density fields also show the presence of particle clusters with thin filaments and sharp transitions from void regions to cluster regions. This is a sign that the GAN is able to generate fine-scale features accurately. This ability of GANs to predict fine-scale features has already been observed in Ledig *et al.* (2017) in the context of image processing.

Figure 4 shows the predicted particle density for $St = 1$ using autoencoder [Figure 4(a)] and U–Net [Figure 4(b)]. In the autoencoder prediction, we see the prediction of filaments of particle cluster, but the density is very diffused and blurry. The void regions cannot be clearly distinguished from the clusters. We conjecture that the information at different scales is most likely lost during the compression of the data. The U–Net prediction of particle density looks like a negative of enstrophy: When the enstrophy

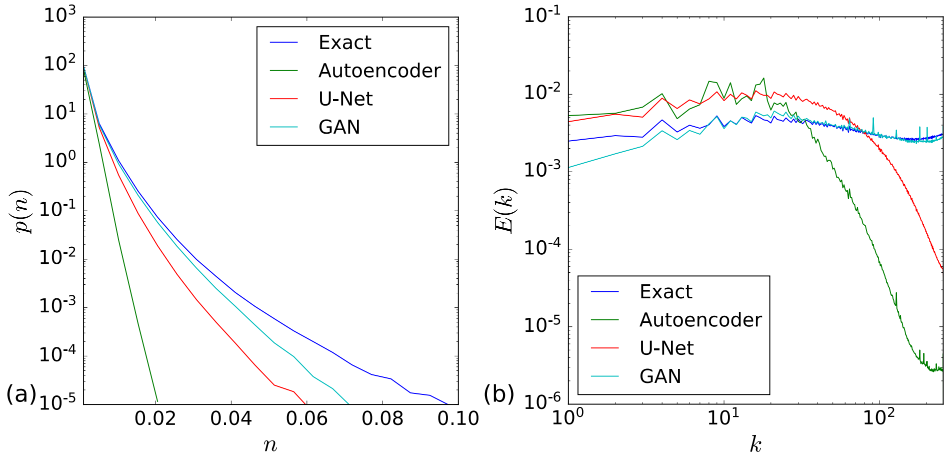

FIGURE 5. PDFs of the (a) particle density and (b) energy spectra of exact (DNS data) and predicted fields using three different architectures for $St = 1$.

value is high, the particle density is low and vice versa. We can notice that the particle density distribution predicted by the GAN is similar to the one predicted by the U–Net. However, the U–Net, unlike the GAN, has a transition between the void and cluster regions that is more smooth.

In 3D, the density of particles for each Stokes number is deterministic. However, here training is done in 2D, so the different neural networks do not have access to the adjacent slices. This lack of information can be interpreted as a randomness in the training. That may explain the performance of the GAN in comparison to the autoencoder and U–Net: The former is a generative model, which is therefore appropriate to approximate a stochastic generative process. However, the autoencoder and the U–Net are deterministic models and are good to fit deterministic functions and relationships. Similarly, replacing the autoencoder by a variational autoencoder can be interesting for its generative properties.

3.2. *Comparison of statistics*

Figure 5 shows probability density functions (PDFs) of the density [Figure 5(a)] and energy spectra of the number density [Figure 5(b)], normalized by the second-order moment, of the DNS and predicted values for $St = 1$ using different network architectures averaged on 768 slices. The energy spectrum is computed by taking the modulus square of the Fourier transformed density and then summed over concentric circles in wavenumber space. The PDF of the GAN prediction is the closest to the exact value, compared to the other networks, even if the probability at high densities is underpredicted. Compared to the GAN, the probability of having high density decreases for U–Net and even more for the autoencoder. For the energy spectrum of the density, we can see that the GAN prediction is almost on top of the exact DNS values even though we can see spikes at some frequencies. At high frequencies, spectrum drops very quickly for the U–Net, and the drop is even faster for the autoencoder. We can conclude that for both quantities, PDF and energy spectra of the density, we have better agreement between the DNS and predicted value for the GAN than for the autoencoder and U–Net. Therefore, we further analyze the results generated by the GAN for different Stokes numbers.

Figure 6 shows PDFs of the density [Figure 6(a)] and energy spectra of the density

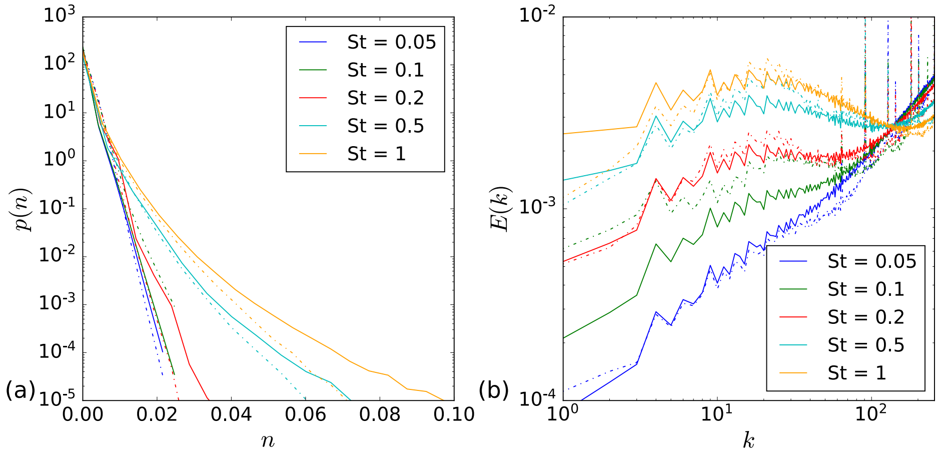

FIGURE 6. PDFs of the (a) particle density and (b) energy spectra of exact (solid lines) and predicted (dashed-dotted lines) fields using the GAN for different Stokes numbers.

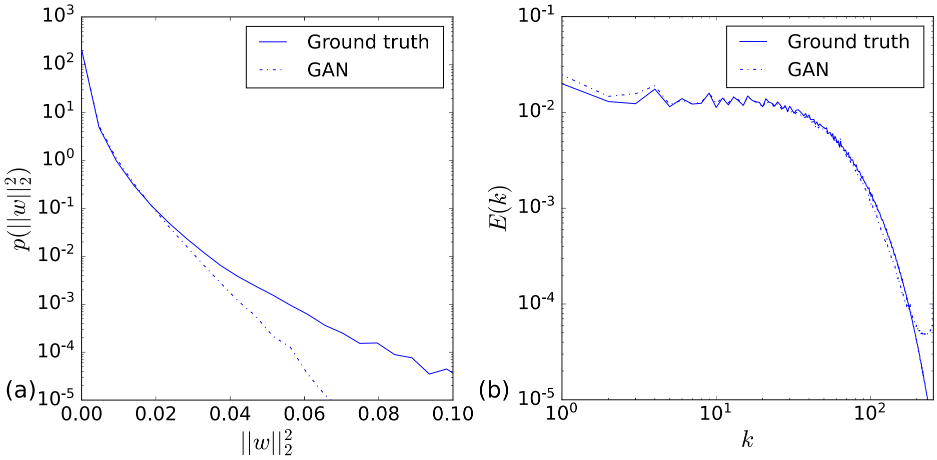

FIGURE 7. (a) PDFs and (b) energy spectra of exact (solid lines) and predicted (dashed-dotted lines) value of the enstrophy $||w||_2^2$ (not to be confused with the enstrophy spectrum).

[Figure 6(b)], normalized by the second-order moment, of exact and predicted values for St = 0.05, 0.1, 0.2, 0.5, and 1 generated by the GAN averaged on 768 slices. We can observe that, with the exception of $St = 0.1$, the predicted density is lower than the exact value, and this deviation increases as the density is higher. The energy spectrum of the predicted value, however, matches the DNS value reasonably well for all Stokes numbers, except for $St = 0.1$. In this cases the differences are much larger, particularly on large scale. This could be explained by the instability of the GAN. We can observe that all the spikes are at the same place for the different Stokes numbers. This may be due to the simple treatment to account for the periodic boundary condition.

We are also interested in knowing the feasibility of the use of the networks for the enstrophy prediction from the particle density. To this end, we try to predict the enstrophy field using the particle density fields for all five Stokes numbers available to us. Note that using particle density fields from different Stokes numbers to predict a single enstrophy field is possible only in the case of one-way coupled point-particle simulations. Indeed, in the two-way coupled case (or more), particles affect the flow, and different Stokes numbers will have different enstrophy fields. Figure 7 shows PDFs of the density [Figure 7(a)] and energy spectra of the density [Figure 7(b)], normalized by the second-order moment, of exact and predicted values of the enstrophy generated by the GAN averaged on 256 slices. For densities below 0.02, we can observe a perfect agreement between the exact and predicted values; however, for higher densities, the predicted value is underpredicted. We can see that the predicted and the DNS energy spectra are in very good agreement; however, we observe that for frequencies $k > 180$, the predicted spectrum saturates, whereas the exact value decays. This could be due either to the fact that we do not have the capacity to take the history of the particles into account or that we are working with 2D slices and this is a 3D phenomenon that we are missing. We can conjecture that using an input data set with more Stokes numbers would better resolve the high-frequency contributions. In fact, this will give a larger sampling of the density of the particles as a function of the Stokes number and thus allow the neural network to access information about the value of the gradient of the enstrophy.

4. Conclusions

Three neural networks for synthesizing preferential concentration fields of inertial particles in fully developed turbulence have been developed and assessed. Instantaneous snapshots of enstrophy fields of DNS data were taken as input, and particle concentration fields were generated using autoencoder, U–Net, and GANs. The quality of prediction is quantified by comparing PDFs and energy spectra of the synthesized fields with the DNS data. The best results were obtained with the GAN, showing similar cluster and void regions, as present in the DNS data. This can be explained by the ability of GANs to generate fine-scale features as well as by its generative properties. The computation time of the GAN is double compared to the two other test neural networks; however, the prediction is also better compared to other two networks. Inverting the input and output in the proposed procedure will allow reconstructing the enstrophy from the particle positions (e.g., obtained via PIV data). This is a physically relevant and promising application for experimentalists.

One important message of this feasibility study is that a generative model will yield better results, at the expense of higher CPU/GPU time. The limitation of the results to 2D slices is imposed by the computational cost. We plan to use wavelet decomposition to compress the vorticity field and the resulting enstrophy fields prior to training. This preprocessing should speed up the training and allow us to synthesize 3D particle density fields with a reasonable cost. Moreover, we expect that training of the network at the different levels of the wavelet decomposition should increase the quality of the results. The results could be further improved by making the architecture more complex, e.g., adding an attention mechanism. Moreover, by using a variational autoencoder, it would be possible to interpolate the value of the particle density between two Stokes numbers. All of these will be explored in future work.

REFERENCES

Brochard, A., Błaszczyszyn, B., Zhang, S. & Mallat, S. 2022 Particle gradient descent model for point process generation. *Stat. Comput.* **32**, 1–25.

Brunton, S. L., Noack, B. R. & Koumoutsakos, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508.

Duraisamy, K., Iaccarino, G. & Xiao, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377.

Faroughi, S. A., Roriz, A. I. & Fernandes, C. 2022 A meta-model to predict the drag coefficient of a particle translating in viscoelastic fluids: a machine learning approach. *Polymers* **14**, 430.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. 2020 Generative adversarial networks. *Commun. ACM* **63**, 139–144.

Hwang, S., Pan, J. & Fan, L.-S. 2021 A machine learning-based interaction force model for non-spherical and irregular particles in low Reynolds number incompressible flows. *Powder Technol.* **392**, 632–638.

Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. 2017 Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,*, pp. 1125–1134.

Kingma, D. P. & Ba, J. 2014 Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. *et al.* 2017 Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690.

Matsuda, K., Onishi, R., Hirahara, M., Kurose, R., Takahashi, K. & Komori, S. 2014 Influence of microscale turbulent droplet clustering on radar cloud observations. *J. Atmos. Sci.* **71**, 3569–3582.

Maxey, M. R. 1987 The gravitational settling of aerosol particles in homogeneous turbulence and random flow fields. *J. Fluid Mech.* **174**, 441–465.

Ronneberger, O., Fischer, P. & Brox, T. 2015 U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. 1986 Learning representations by back-propagating errors. *Nature* **323**, 533–536.

Siddani, B., Balachandar, S., Moore, W. C., Yang, Y. & Fang, R. 2021 Machine learning for physics-informed generation of dispersed multiphase flow using generative adversarial networks. *Theor. Comput. Fluid Dyn.* **35**, 807–830.

Zhang, S. & Mallat, S. 2021 Maximum entropy models from phase harmonic covariances. *Appl. Comput. Harmon. Anal.* **53**, 199–230.