PHYSICAL REVIEW FLUIDS **9**, 034606 (2024)

Improving prediction of preferential concentration in particle-laden turbulence using the neural-network interpolation

Jiajun Hu [1] and Zhen Lu [*]
State Key Laboratory for Turbulence and Complex Systems, College of Engineering, Peking University, Beijing 100871, China

Yue Yang [†]
*State Key Laboratory for Turbulence and Complex Systems, College of Engineering, Peking University, Beijing 100871, China
and HEDPS-CAPT, Peking University, Beijing 100871, China*

(Received 24 October 2023; accepted 26 February 2024; published 13 March 2024)

We develop a neural-network interpolation (NNI) to improve the prediction of preferential concentration in simulations of particle-laden turbulence. The NNI uses the particle position and velocity on neighboring grid points to estimate the fluid velocity at the particle position via fully connected neural networks. By avoiding the requirement for superresolution of the entire field and additional interpolations, the NNI offers computational efficiency and simplifies implementation. To evaluate the effectiveness of NNI and compare it with other interpolation methods, we conduct simulations on two-dimensional homogeneous isotropic turbulence subjected to high-wave-number forcing. This specific turbulent flow has a long inertial range and rich small-scale structures, posing a challenge for velocity interpolations and subsequently accurate prediction of preferential concentration. We compare the results on flow fields, energy spectra, and preferential concentration against the reference data obtained from direct numerical simulations at a range of the Stokes number from 0.1 to 5.0. The comparison demonstrates that the NNI can recover the effect of small-scale motion on particle distribution, so it improves the prediction accuracy of the preferential concentration from *a priori* test results of the large-eddy simulation on coarse grids.

DOI: 10.1103/PhysRevFluids.9.034606

I. INTRODUCTION

Particle-laden turbulence is commonly observed in both natural phenomena and industrial processes [1,2], with examples including cloud droplets, sandstorms, liquid fuel combustion, aerosols, and fluidized beds. The preferential concentration of particles, i.e., particles accumulate in low-vorticity regions due to their inertial bias [3–7], is a critical and characteristic feature of the particle-laden turbulence.

Accurate simulations of the preferential concentration depend on the proper modeling of the interactions between particles and the turbulent flow field. One essential modeling quantity is the flow velocity $\mathbf{u}(\mathbf{x}_p)$ at the particle position $\mathbf{x}_p$. Since the velocity is solved on a computational grid, an interpolation is necessary to obtain $\mathbf{u}(\mathbf{x}_p)$ because the particles are not always located at grid points,

*zhen.lu@pku.edu.cn
†yyg@pku.edu.cn

The direct numerical simulation (DNS) that resolves all flow scales is a powerful tool to simulate particle-laden turbulence. It has been extensively used to investigate the preferential concentration in two-dimensional (2D) [8–12] and three-dimensional (3D) [13–23] homogeneous isotropic turbulence (HIT). The high-resolution (HR) velocity field in the DNS allows for the accurate determination of $\mathbf{u}(\mathbf{x}_p)$. In the Eulerian-Lagrangian approach, classical interpolation methods based on neighboring grid points of $\mathbf{x}_p$ have been commonly used [8,10,23]. However, DNS suffers by its demanding computational cost.

In recent decades, large-eddy simulation (LES) has been employed in the investigation of particle-laden turbulence. In LES, the large-scale motions are solved directly on a low-resolution (LR) grid, and the effect of small-scale or subgrid-scale (SGS) motions on large-scale ones are modeled. The preferential concentration of particles is significantly influenced by SGS motions at intermediate Stokes numbers, so it requires additional SGS modeling [24–31]. At the same time, interpolation of $\mathbf{u}(\mathbf{x}_p)$ based on the LR fields imposes extra challenges.

Although the use of a LR grid significantly reduces the accuracy of the polynomial-based interpolation on $\mathbf{u}(\mathbf{x}_p)$ [32,33], the advancement of machine learning (ML) [34] may facilitate achieving accurate and efficient interpolation of $\mathbf{u}(\mathbf{x}_p)$. For example, superresolution (SR) generates a HR field from a LR input, and then the classical interpolation is employed to obtain $\mathbf{u}(\mathbf{x}_p)$ based on the reconstructed HR field. Since Fukami *et al.* [35] explored convolutional-neural-network (CNN)-based models for SR in 2D decaying HIT, several investigations [36–39] have demonstrated the applicability of SR to turbulence. The CNN models have also been employed in simulating particle-laden turbulence. Shirzadi *et al.* [40] introduced a CNN model to generate the velocity and pressure field around particles for Lagrangian simulation of particle motions, showing more accurate solutions and lower computational cost than classical methods. However, the SR typically operates and stores information at grid points for the entire flow field, while particles are usually not at grid points. This inconsistency can hinder an efficient implementation of SR-based interpolation for $\mathbf{u}(\mathbf{x}_p)$ in particle-laden turbulence.

Instead of using SR, we seek to directly interpolate $\mathbf{u}(\mathbf{x}_p)$ from the LR field with the aid of neural networks (NNs). Raissi *et al.* [41] introduced the physics-informed NNs (PINNs) as an approach for solving partial differential equations. PINNs take arbitrary spatiotemporal coordinates as inputs and output the physical values. Bezgin *et al.* [42] combined the weighted essentially nonoscillatory (WENO) scheme with a NN, proposing the WENO-NN. Both WENO-NN and WENO share identical inputs and outputs. Validation in a one-dimensional shocktube simulation confirmed that WENO-NN achieves higher accuracy than WENO. The PINNs and WENO-NN offer insights into designing NNs that can generate output values at random input positions. This feature is particularly suitable for the application of particle tracking, as it allows for the direct acquisition of $\mathbf{u}(\mathbf{x}_p)$ and avoids additional interpolations.

We propose a NN interpolation (NNI) to combine the advantages of ML-assisted modeling and SR, aiming to improve the prediction of preferential concentration in simulations of particle-laden turbulence. The NNI uses the particle position and velocity on neighboring grids to calculate $\mathbf{u}(\mathbf{x}_p)$ via fully connected NNs (FC-Nets). This approach avoids the need for SR of the entire field and additional interpolations, reducing computational cost and simplifying implementation.

We evaluate NNI using a series of challenging cases of 2D particle-laden HIT. Although most previous studies employed a decaying or low-wave-number forcing HIT for validation, we chose a 2D HIT with high-wave-number forcing [43]. This HIT contains rich small-scale structures within a long inertial range. The LR field obtained by applying a cutoff filter in the inertial range retains little small-scale motion from the HR field, so it is very challenging for velocity interpolations and ML methods to reproduce $\mathbf{u}(\mathbf{x}_p)$ and consequently to predict the preferential concentration accurately.

The outline of this paper is as follows. Section II gives an overview of the simulation of 2D HIT and particle motions. Section III describes the NNI. Section IV compares the results of different velocity interpolation methods on predicting the preferential concentration in particle-laden turbulence. Conclusions are drawn in Sec. V.

TABLE I. Parameters of the 2D HIT

Grid size	N	1024
Time step	Δt	5×10^{-4}
Forcing length scale	k_f	300
Forcing strength	α	0.1
Hyperviscosity	ν_u	7.5×10^{-40}
Hyperviscosity power	p_u	8
Hypoviscosity	ν_i	2.5
Hypoviscosity power	p_i	1

II. DATASETS

A. Flow field

We obtained datasets from the DNS of 2D particle-laden HIT with the hyperviscosity to train ML-based methods and evaluate different velocity interpolations. For the flow field, the standard pseudospectral method [43] was employed to solve the vorticity equation:

$$(\partial_t + \nu_i k^{-2p_i} + \nu_u k^{2p_u})\widehat{\omega}(\mathbf{k}) + [\widehat{(\mathbf{u} \cdot \nabla)\omega}](\mathbf{k}) = \widehat{f}(\boldsymbol{k}), \tag{1}$$

in the Fourier space, where $\mathbf{u}$ is the velocity, $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is the vorticity, the hat $(\widehat{\ })$ denotes the Fourier transform, $\mathbf{k}$ and $k = |\mathbf{k}|$ are the wave number vector and wave number, respectively, and $\widehat{f}(\mathbf{k})$ is the external forcing term. The forcing is set as $\widehat{f}(\mathbf{k}) = \alpha G(k - k_f)$, where α is the forcing strength, $G(k - k_f)$ is the Gaussian smoothing function, and k_f denotes the wave number where the forcing is injected. The computational domain is a periodic box with the size of $[0, 2\pi]^2$, and it is discretized on a uniform grid of N^2. The phase-shift method [44] is applied to dealiase the solution. The second-order Adam-Bashforth scheme is used to advance in time.

Table I summarizes the 2D HIT setup. The number of grid points is set to 1024^2. The forcing wave number $k_f = 300$ is the largest possible one while still being constrained by the $\frac{2}{3}$ rule in dealiasing and $k_f \leqslant N/3$ [8]. The hyperviscosity and hypoviscosity terms dissipate energy at small and large scales, respectively. These viscosity parameters are selected based on previous studies [8,45]. As a result of the inverse cascade [46], the high-wave-number forcing produces a turbulent flow with a long inertial range, which has been widely adopted in past 2D turbulence simulations [45,47,48].

Important statistics in the DNS of 2D HIT are listed in Table II, including the total kinetic energy $E_{\text{tot}} = \int E(k)dk$, root-mean-square (rms) velocity fluctuation $u' = \sqrt{E_{\text{tot}}}$ [49], integral length scale $l_T = 2\pi \int k^{-1}E(k)dk/E_{\text{tot}}$, eddy turnover time $\tau_e = l_t/u'$, rms vorticity ω' [10], Kolmogorov length scale $\eta = 2\pi/k_f$ [8], and Kolmogorov time scale $\tau_\eta = 1/\omega'$ [10]. The current simulation with a long inertial range and rich SGS motions poses significant challenges for the velocity interpolation methods.

TABLE II. Statistics of the 2D HIT with hyperviscosity.

Total kinetic energy	$E_{\text{tot}} = \int E(k)dk$	0.280
rms velocity fluctuation	$u' = \sqrt{E_{\text{tot}}}$	0.529
Integral length scale	$l_t = 2\pi \int k^{-1}E(k)dk/E_{\text{tot}}$	0.626
Eddy turnover time	$\tau_e = l_t/u'$	1.18
rms vorticity	ω'	38.2
Kolmogorov length scale	$\eta = 2\pi/k_f$	0.0209
Kolmogorov time scale	$\tau_\eta = 1/\omega'$	0.0262

We have also tested on a 2D particle-laden HIT with normal viscosity and low-wave-number forcing. When the cutoff wave number is close to the dissipation range, the moderate- and small-scale eddies remain unfiltered, so that such a case only has marginal room for improving the interpolation methods [28,50].

In the 2D HIT simulations, a total of 3×10^4 time steps were saved after the flow field reached the statistically stationary state. The saved data covers a period of $t = 15$, $\sim$12.7 eddy turnover times. The initial $\frac{2}{3}$ and the final $\frac{1}{3}$ of the time steps were used for training and validating the ML-based methods, respectively. To obtain the LR flow field, we downsampled the HR field from DNS using an average pool. The average pool downsamples from the HR field with a size of $(H_{\mathrm{DNS}}, W_{\mathrm{DNS}})$ to $(H_{\mathrm{DNS}}/M, W_{\mathrm{DNS}}/M)$ LR field as $u_{i,j}^{\mathrm{LR}} = \frac{1}{M^2}\sum_{p=1}^{M}\sum_{q=1}^{M} u_{i\times M+p,\, j\times M+q}^{\mathrm{DNS}}$, where the pooling factor M is set to powers of two.

B. Particle motion

We consider the motion of small spherical particles described by the simplified Basset-Boussinesq-Oseen (BBO) equation [4,51]:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p, \quad \frac{d\mathbf{u}_p}{dt} = \frac{\mathbf{u}(\mathbf{x}_p) - \mathbf{u}_p}{\tau_p}, \tag{2}$$

where $\mathbf{u}_p$ denotes the particle velocity, and τ_p the particle response time scale. The fourth-order Lagrangian interpolation (LGI) was used to interpolate $\mathbf{u}(\mathbf{x}_p)$ from the HR flow field in DNS. Equation (2) is advanced using the second-order Runge-Kutta scheme.

We conducted an extensive investigation with a range of the Stokes number $St = \tau_p/\tau_\eta$ from 0.1 to 5.0. In each case, we tracked 10^5 particles that are initially distributed uniformly in the flow. The particles were initialized at $t = 0$ and were then tracked over a time period of $t = 15$. The particle motion calculated using LGI based on the HR DNS is considered as the ground truth.

III. NNI

We propose a NNI to obtain $\mathbf{u}(\mathbf{x}_p)$ from the LR flow fields. The NNI has the same input and output as the LGI. As illustrated in Fig. 1(a), the input into the NNI consists of the particle position and the velocities at the surrounding 4×4 grid points in the LR field. To ensure generality, the particle position $\mathbf{x}_p = (x_p, y_p)$ is normalized as $(\xi, \zeta) = [(x_p - x_1)/h, (y_p - y_1)/h]$ with the grid spacing $h = 2\pi/N$. The NNI outputs a flow velocity component $u(\mathbf{x}_p)$ or $v(\mathbf{x}_p)$ at the particle position.

As sketched in Fig. 1(b), the NNI architecture consists of three FC-Nets to estimate $u(\mathbf{x}_p)$ from the input. FC-Net 1 and FC-Net 2 take the velocities and particle positions as inputs, respectively. The outputs of FC-Net 1 and FC-Net 2 are two n-dimensional vectors $\mathbf{a}$ and $\mathbf{b}$, respectively, with $n = 10$. The intermediate array $(a_1 b_1, a_2 b_2, \cdots, a_n b_n)^{\intercal}$ is then fed into FC-Net 3 to yield $u(\mathbf{x}_p)$. Unlike the SR methods used in turbulence simulation, the NNI does not read the entire LR flow field to reconstruct the entire HR field. Instead, its input is the particle location, and its output is the flow velocity at the particle position. This can greatly reduce the computational cost for the SR of the LR field and additional Lagrangian interpolations in tracking particles.

The mean-square-error loss function $L = \frac{1}{m}\sum_{i=1}^{m}[u^{\mathrm{NNI}}(\mathbf{x}_i^s) - u^{\mathrm{DNS}}(\mathbf{x}_i^s)]^2 + \lambda\|\mathbf{w}\|_2^2$ is adopted to optimize the NNI, where $(\mathbf{x}_1^s, \mathbf{x}_2^s, \cdots, \mathbf{x}_m^s)$ are m points on the DNS grid, superscripts NNI and DNS denote NNI and DNS results, respectively, and $\mathbf{w} = (w_1, w_2, \cdots, w_{N_w})$ are the N_w weights in NNs. The L2 regularization $\lambda\|\mathbf{w}\|_2^2 = \sum_{i=1}^{N_w} w_i^2$ avoids overfitting, where λ serves as a regularization parameter [52].

Each learning batch contains $m = 1000$ points $(\mathbf{x}_1^s, \mathbf{x}_2^s, \cdots, \mathbf{x}_m^s)$, together with the ground truth $\mathbf{u}^{\mathrm{DNS}}(\mathbf{x}_i^s)$, where $\mathbf{x}_i^s$ is the particle position. The m points are randomly selected from the 1024^2 HR field over the 2×10^4 time steps. Then the velocities on the adjacent 4×4 LR grid points and the normalized particle position (ξ, ζ) were used to generate $\mathbf{u}^{\mathrm{NNI}}$ at each selected point. The Kaiming

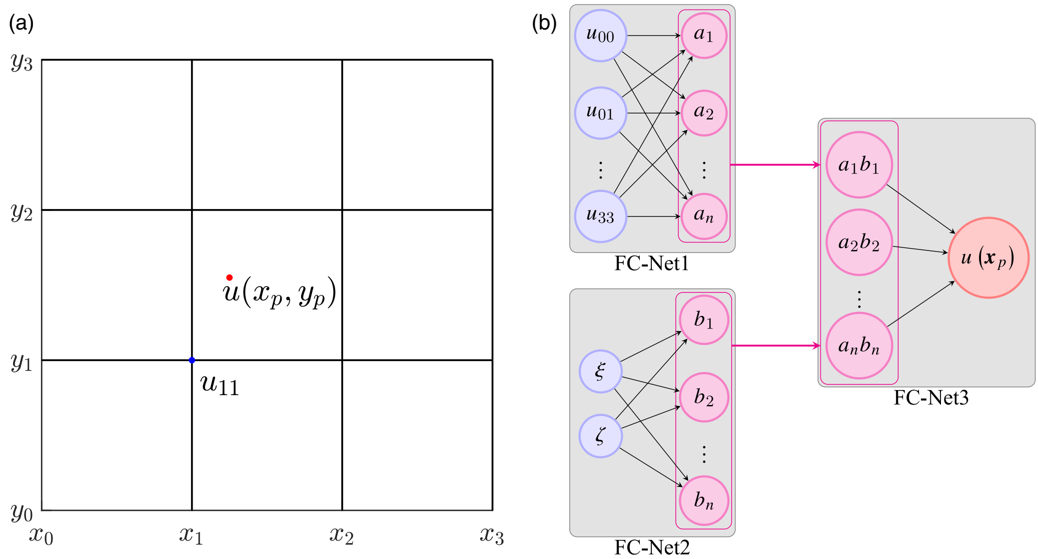

FIG. 1. (a) Schematic of the input and output of neural-network interpolation (NNI) and Lagrangian interpolation (LGI). (b) Flowchart of NNI.

method [53] initializes all the NNI parameters, and the parameters are updated using the Adam optimizer [54]. The initial learning rate is 0.01, which is subsequently reduced to $\frac{1}{5}$ of its preceding value after every 2000 epochs.

IV. RESULTS

To assess the performance of NNI, we compare the results from four methods (see Fig. 2), DNS with LGI, the LR field with LGI, the LR field with CNN SR and LGI, and the LR field with NNI. Here, the DNS results with $\mathbf{u}(\mathbf{x}_p)$ interpolated from HR fields using the LGI serve as the ground truth. The LR field with regular LGI serves as an *a priori* test for the LES. The CNN-assisted LGI interpolates $\mathbf{u}(\mathbf{x}_p)$ from the SR field generated from the LR field. We employed a CNN architecture

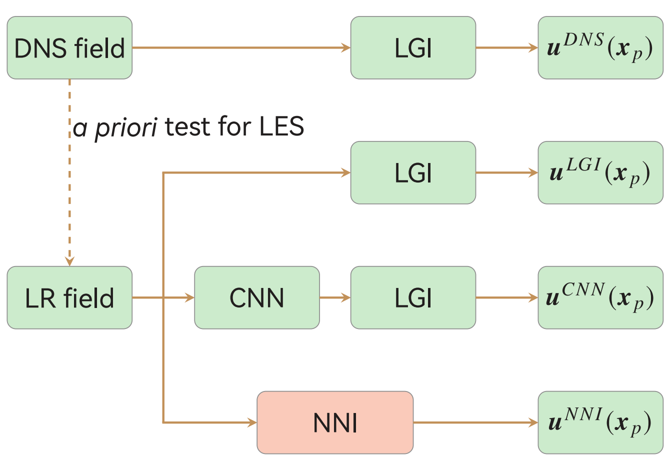

FIG. 2. Summary of different approaches of tracking particles.

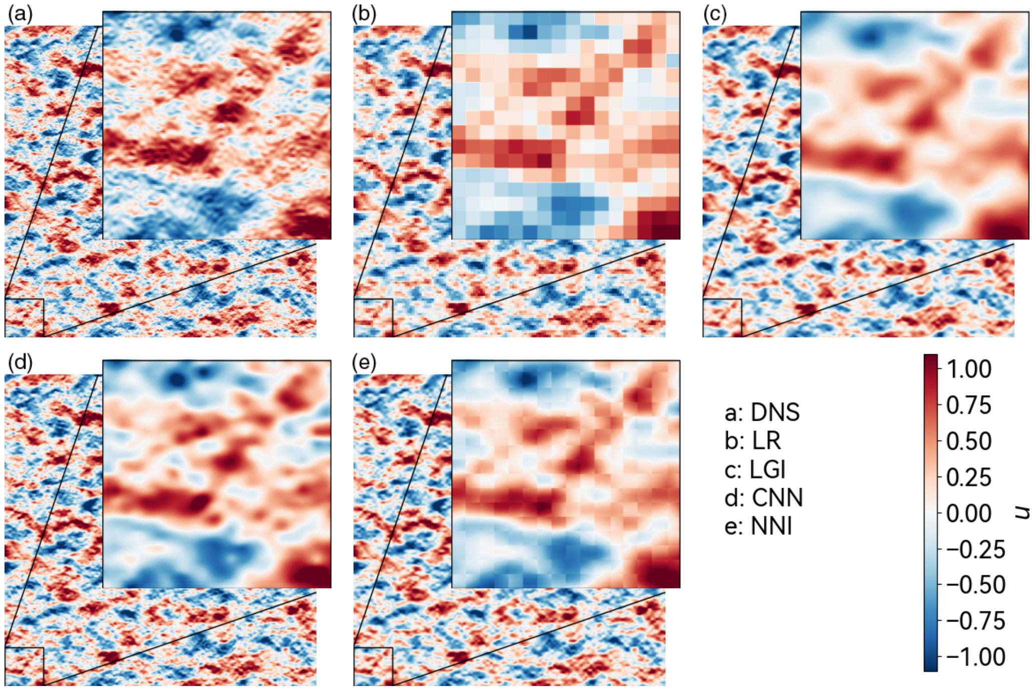

FIG. 3. Instantaneous velocity contours of the (a) high-resolution (HR) field with resolution 1024^2 by direct numerical simulation (DNS), (b) low-resolution (LR) field with resolution 128^2, and reconstructed fields with resolution 1024^2 by (c) Lagrangian interpolation (LGI), (d) convolutional neural network (CNN), and (e) neural-network interpolation (NNI).

validated for SR of HIT [39], consisting of eight residual blocks and three upscale modules with a kernel size of 3×3, and the CNN-based SR was trained using the same datasets and tools as for the NNI. The NNI directly interpolates $\mathbf{u}(\mathbf{x}_p)$ from the LR field without the additional SR for the HR field.

For convenience, we refer to the direct application of LGI on LR fields as LGI and the CNN-assisted LGI as CNN below. If not specified, NNI, CNN, and LGI are used for the particle computation in a 128×128 LR flow field. The LR flow field is obtained by an 8×8 averaging pooling of the 1024×1024 HR flow field in DNS. Moreover, an *a posteriori* test based on the LES of 2D HIT is provided in Appendix.

A. Flow field

We first compare the accuracy of reconstructing the HR velocity field using different interpolation methods from the LR field. Figure 3 plots snapshots of the contour of u obtained from the 1024^2 HR field by DNS, 128^2 LR field, and 1024^2 reconstructed fields by the LGI, CNN, and NNI at $t = 15$, along with closed-up views to show details of the resolved flow structures. The LR field in Fig. 3(b) only retains blurred large-scale structures. Although the LGI increases the spatial resolution, it smooths out small-scale structures. This limitation arises from the incapability of low-order polynomial functions for capturing complex flow features. Furthermore, simply increasing the interpolation order can lead to the Runge phenomenon.

Remarkably, both NNI and CNN reconstruct finer flow structures than those from the LGI. We observe that the flow field reconstructed by NNI shows weak discontinuities at the LR grid boundary. This is because the interpolation parameters of NNI are obtained by optimization of the global flow

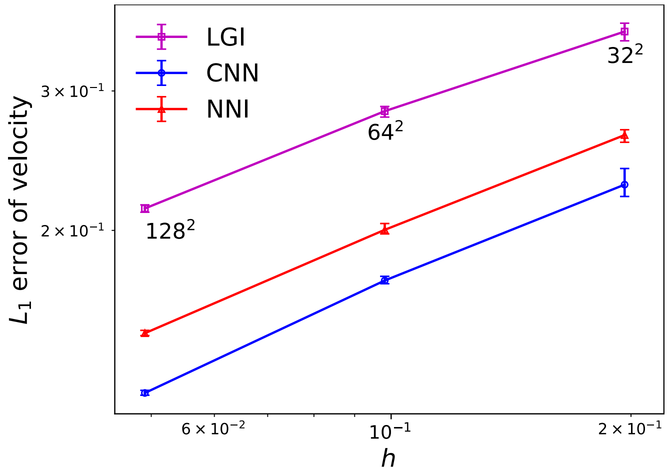

FIG. 4. L_1 error of the velocity field in Lagrangian interpolation (LGI), convolutional neural network (CNN), and neural-network interpolation (NNI) cases, with resolutions 128^2, 64^2, and 32^2 of the low-resolution (LR) field.

field and do not guarantee the zeroth-order continuity through polynomial functions as in the case of LGI. Meanwhile, the NNI used only 4^2 neighboring points for local interpolations compared with CNN. Such discontinuities were also observed in early CNN SR work [35] when the convolution layers are shallow.

The L_1 error $L_1 = \sum_{i=1}^{N}\sum_{j=1}^{N}|\phi_{ij}^{\mathrm{R}} - \phi_{ij}^{\mathrm{DNS}}|$ is used to measure the discrepancy between the reconstructed field and the ground truth at each time step, where ϕ_{ij}^{R} denotes a reconstructed quantity and ϕ_{ij}^{DNS} the DNS ground truth. Figure 4 compares the time-averaged L_1 error of the velocity from $t = 10$ to 15 from LGI, CNN, and NNI. The error bars representing the range of L_1 during $t = 10$–15 indicate a small fluctuation of L_1 for different time steps. The results are based on LR fields with three different grid resolutions 128^2, 64^2, and 32^2. The LR field with a higher resolution or smaller h preserves more details from the HR fields, leading to better performance for all interpolation methods. The NNI and CNN outperform the LGI, and the CNN, which reads in the entire LR field for SR, exhibits the highest accuracy in reconstructing the entire flow field.

Figure 5 plots the energy spectra of the HR field by DNS, the LR field by the average pool, and the reconstructed fields by the LGI, CNN, and NNI. The CNN and NNI outperform the LGI in recovering high-frequency data. However, the ML-based methods fail to recover the full small-scale DNS field at very high wave numbers because the small-scale turbulence is highly nonlinear, and most ML-based methods suffer [55,56]. The present 2D HIT has a long inertial range, and the cutoff at k_c removes a wide range of small-scale structures, which poses a more significant challenge in reconstructing SGS motion than those reported in 3D turbulence [28,50]. Additionally, both CNN and NNI do not adequately capture the rapid decay of the spectrum at $k > k_f$. Thus, both CNN and NNI only partially recover the SGS motion when the LR field is significantly underresolved.

Table III presents the turbulence statistics of the LR field and the reconstructed fields by LGI, CNN, and NNI, compared with the DNS results. The 128^2 LR field contains 78.2% of the total energy in the HR field. While LGI recovers total energy slightly closer to DNS, CNN and NNI exhibit clear improvements in these statistics. Due to the SR of the entire field, CNN outperforms all methods in most large-scale statistics. Note that NNI performs best in recovering small-scale motions, as indicated by the Kolmogorov time scale. Figure 6 compares the vorticity distribution from various methods. Consistent with the spectra, the CNN and NNI partially reconstruct small-scale vortical structures from the large-scale LR field.

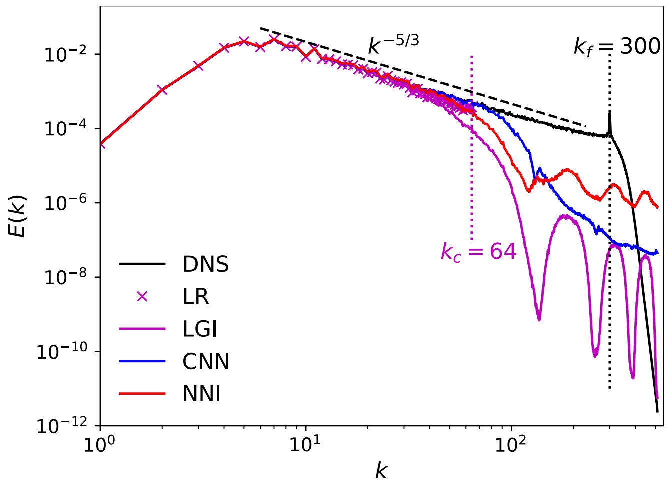

FIG. 5. Energy spectra of the high-resolution (HR) field by direct numerical simulation (DNS), the low-resolution (LR) field by average pool, and the reconstructed fields by Lagrangian interpolation (LGI), convolutional neural network (CNN), and neural-network interpolation (NNI). The purple dotted line marks the cutoff wave number, and the black dotted line marks the wave number where the forcing is injected.

B. Particle distribution

Next, we examine the effect of different interpolation methods in predicting particle distributions in turbulence. Figure 6 plots the particle distributions obtained using different methods in the left domain at $t = 15$. The particles are concentrated in regions of low vorticity, which is referred to as the preferential concentration [2–5]. Moreover, the particle distributions in the CNN and NNI simulations are more dispersed than the LGI results, as the ML-based methods preserve more small-scale motions.

The preferential concentration of particles is sensitive to the small-scale motion [5]. Figure 7 compares the particle distributions obtained via DNS and simulations based on the LR field with LGI, CNN, and NNI at $t = 15$ for the case with $St = 1.0$. The particles are color-coded by $|\omega|$. It confirms that particles tend to accumulate at low-vorticity regions at the statistically stationary state. In particle simulations, $\mathbf{u}(\mathbf{x}_p)$ is interpolated at the particle position, illustrated in the zoom-in subplots with arrows. The DNS results show the velocity vectors circling along two major vortices. However, these vectors do not strictly adhere to the swirling direction of the large vortex due to small-scale motions. In the LR field with LGI, the small-scale motions are filtered. Consequently, the velocities tend to align with the direction of large-scale motions, resulting in a higher preferential concentration. Both CNN and NNI recover small-scale motions in the turbulence fields, which

TABLE III. Turbulence statistics of the HR field by DNS, LR field by average pool, and reconstructed fields by LGI, CNN, and NNI.

Statistics		DNS	LR	LGI	CNN	NNI
Total kinetic energy	E_{tot}	0.280	0.219	0.222	0.252	0.246
rms velocity fluctuation	u'	0.529	0.468	0.471	0.502	0.496
Integral length scale	l_t	0.626	0.768	0.758	0.682	0.711
Eddy turnover time	τ_e	1.18	1.64	1.61	1.36	1.43
rms vorticity	ω'	38.2	8.579	11.0	18.3	19.8
Kolmogorov time scale	τ_η	0.0262	0.117	0.0905	0.0547	0.0504

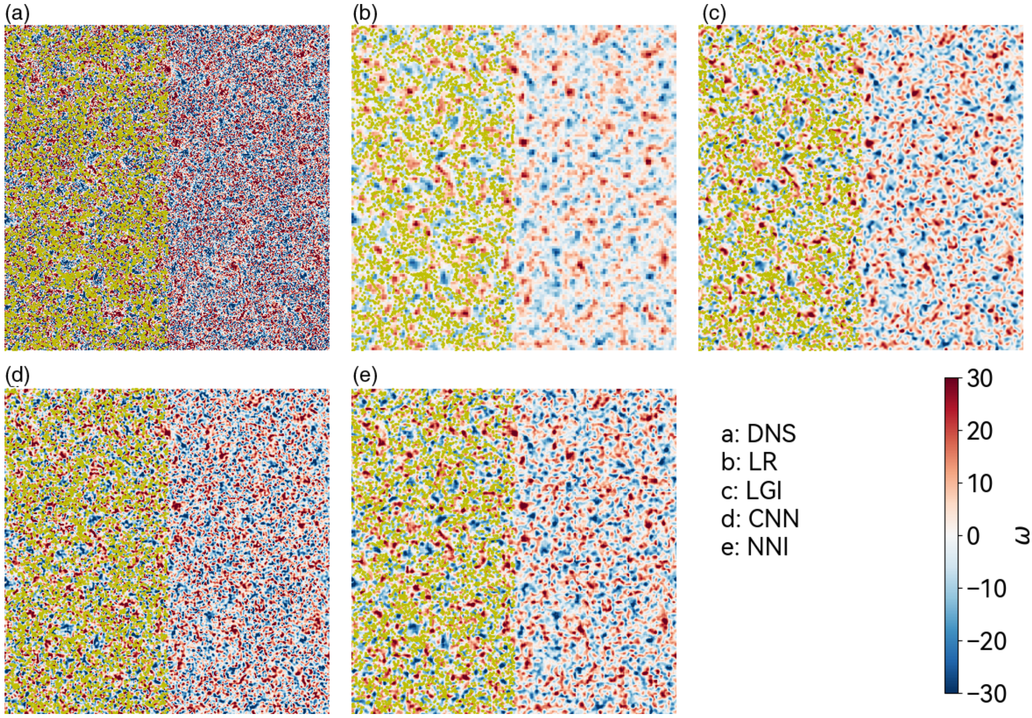

FIG. 6. Instantaneous vorticity contours and particle distributions (marked in the left half domain using yellow dots) obtained from the (a) high-resolution (HR) field (1024^2) by direct numerical simulation (DNS), (b) low-resolution (LR) field (128^2), and reconstructed fields (1024^2) by (c) Lagrangian interpolation (LGI), (d) convolutional neural network (CNN), and (e) neural-network interpolation (NNI).

perturb the velocity and add high-wave-number energy to the turbulence field. Therefore, the velocity directions become more stochastic, and the particles are more scattered than in the LGI case.

We apply the D method [57] to quantify the preferential concentration of particles. As illustrated in Fig. 8(a), this method measures the L_2 distance D_C between the probability density function (PDF) of the particle number N_c in each cell and the Poisson distribution. The uniform distribution of particles at $t = 0$ corresponds to the Poisson distribution of N_c and $D_C = 0$. The preferential particle concentration causes the growth of D_C. The evolution of D_C in DNS with $St = 0.3$, 1.0, and 4.0 in Fig. 8(b) shows that the particle distribution reaches the statistically stationary state around $t/\tau_e = 5$.

In Fig. 9, $\langle D_C \rangle$, the time-averaged D_C over $t = 10$–15, obtained via LGI, CNN, and NNI based on the 128^2 LR fields are compared with the DNS results for different values of St. The error bars denote the range of D_C. Consistent with the former results [4,23,25,57,58], $\langle D_C \rangle$ peaks around $St \approx 1$, indicating the strongest preferential concentration. As St approaches zero, the particles behave as passive tracers, resulting in a uniform distribution with $D_C \approx 0$. With increasing St, the particles become less responsive to the fluid motion, and the distribution begins to deviate from uniform with growing $\langle D_C \rangle$. At very large St, the particles are less affected by the fluid motion, and the preferential concentration is reduced with moderate $\langle D_C \rangle$.

The LR field, as an *a priori* test of LES, only preserves the large-scale eddies. The absence of SGS motion mitigates the particle dispersion. In Fig. 7, particles tend to be more concentrated at the edges of large eddies than that in DNS. As a result, the profile of $\langle D_C \rangle$ against St by LGI is significantly overpredicted. Additionally, the LR field increases the effective Kolmogorov time

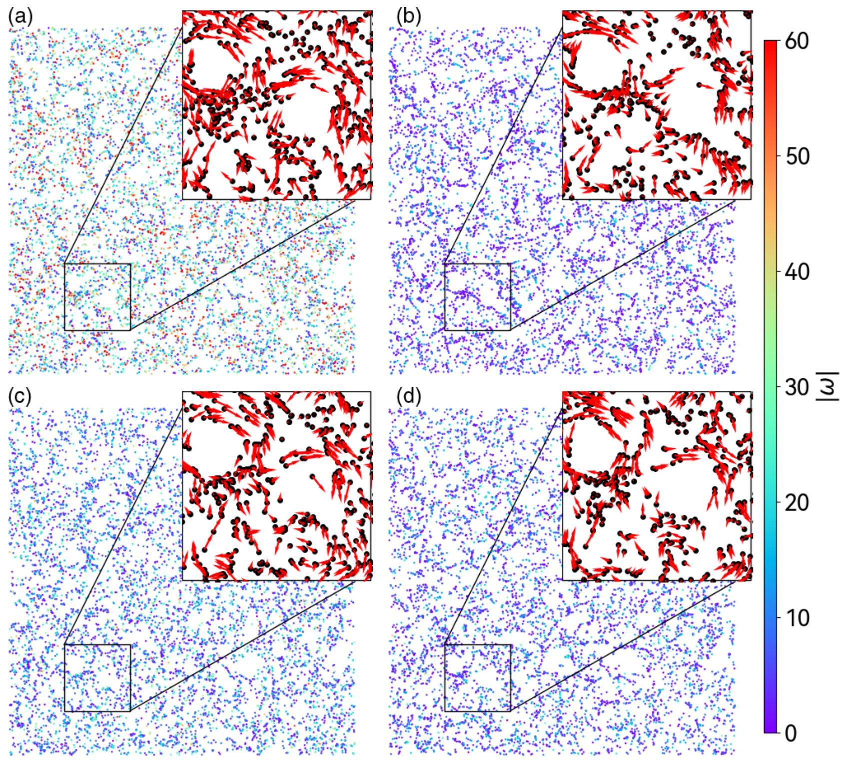

FIG. 7. Particle distributions obtained via (a) direct numerical simulation (DNS) and simulations based on the low-resolution (LR) field with (b) Lagrangian interpolation (LGI), (c) convolutional neural network (CNN), and (d) neural-network interpolation (NNI) at $t = 15$ for the case with $St = 1$. The particle is color-coded by $|\omega|$. The arrows in zoom-in subplots denote $\mathbf{u}(\mathbf{x}_p)$ for each particle.

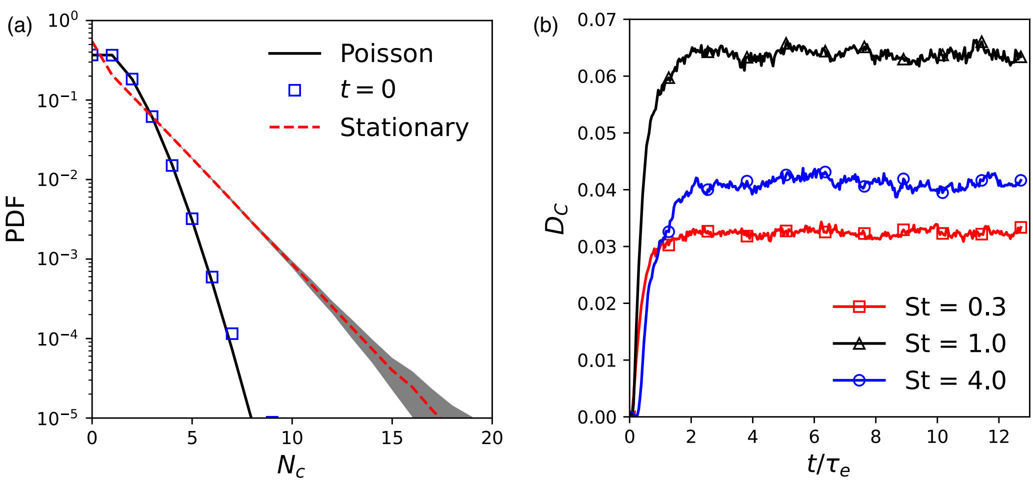

FIG. 8. Direct numerical simulation (DNS) results on (a) the probability density function (PDF) of the particle number in each cell at $t = 0$ and the average over $t = 10$–15, when the particle distribution reaches a stationary state, along with the Poisson distribution. The gray zone denotes the variance of the PDF at $t =$ 10–15. (b) Evolution of D_C in DNS cases with $St = 0.3$, 1.0, and 4.0.

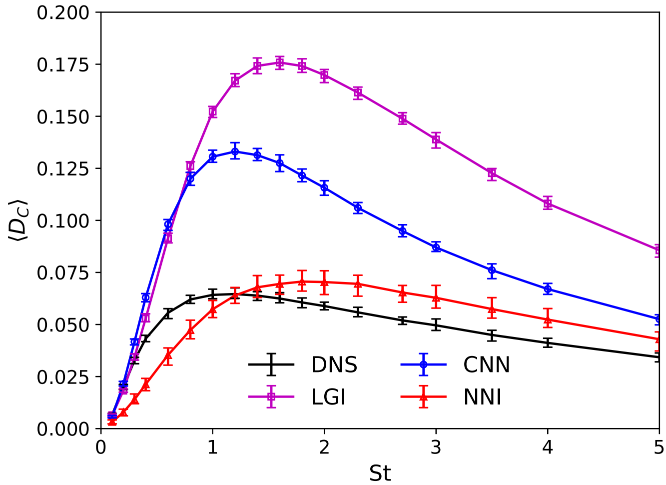

FIG. 9. Comparison of $\langle D_C \rangle$ in direct numerical simulation (DNS), Lagrangian interpolation (LGI), convolutional neural network (CNN), and neural-network interpolation (NNI) based on the low-resolution (LR) field (128^2) for different St. The error bars denote the range of D_C.

scale and then decreases the effective Stokes number [25], shifting the predicted peak of D_C toward larger St values.

As shown in the energy spectrum in Figs. 5 and 6, both CNN and NNI can partially recover small-scale eddies in the inertial range, outperforming LGI. This enables the CNN method to predict the profile of $\langle D_C \rangle$ for the preferential concentration more accurately.

The NNI method, on the other hand, recovers more high-wave-number information than the other methods in Fig. 5, allowing it to capture the small-scale motions in the turbulence field. Although NNI has the same input and output as LGI, the FC-Nets in NNI have a more sophisticated expression than the polynomial basis function in LGI, enhancing the fitting capability of the NNI. Compared with CNN, NNI focuses on reconstructing near the local region of a LR grid cell, in which the NNI predicts the velocity over the space continuously, while CNN only gives values at discretized points. As a result, the preferential concentration in the NNI case is most consistent with the DNS result.

We also compare the results of different interpolation methods for three LR fields with 128^2, 64^2, and 32^2 grid points in Fig. 10(a). In general, $\langle D_C \rangle$ decreases with increasing grid resolution because the higher-resolution fields preserve more small-scale turbulence and lead to a more dispersed particle distribution. The NNI generally has the smallest L_1 error for all three LR grids, indicating its best performance in predicting the preferential concentration.

Furthermore, Fig. 10(b) shows the sum of the D_C deviations $\sum_{St} |\langle D_C \rangle_{St}^{\Theta} - \langle D_C \rangle_{St}^{\mathrm{DNS}}|$ over the 18 Stokes numbers of the LGI, CNN, and NNI results from the DNS results for different resolutions. The results confirm that the NNI is the most accurate among the three interpolation methods, showing the first-order accuracy with the resolution of the flow field.

C. Computational cost

We evaluate the computational cost of different methods by separately considering two main parts: fluid flow evolution and particle tracking. According to Fig. 2, we divide the computational cost of each method into

$$\begin{aligned} T^{\mathrm{DNS}} &= T_f^{\mathrm{DNS}} + T_p^{\mathrm{LGI}}, \qquad & T^{\mathrm{LGI}} &= T_f^{\mathrm{LR}} + T_p^{\mathrm{LGI}}, \\ T^{\mathrm{CNN}} &= T_f^{\mathrm{LR}} + T_f^{\mathrm{SR}} + T_p^{\mathrm{LGI}}, \quad & T^{\mathrm{NNI}} &= T_f^{\mathrm{LR}} + T_p^{\mathrm{NNI}}, \end{aligned} \tag{3}$$

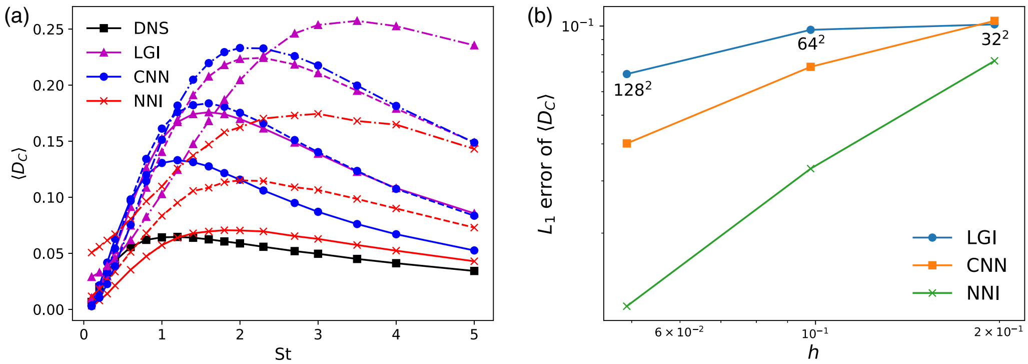

FIG. 10. (a) Comparison of $\langle D_C \rangle$ in direct numerical simulation (DNS), Lagrangian interpolation (LGI), convolutional neural network (CNN), and neural-network interpolation (NNI) based on low-resolution (LR) fields with different grid resolutions 128^2 (solid lines), 64^2 (dashed lines), and 32^2 (dash-dotted lines) for different St. (b) L_1 error of $\langle D_C \rangle$ of the LGI, CNN, and NNI results with different grid resolutions.

where T_f^{DNS} and T_f^{LR} denote the costs of flow simulations on HR and LR grids, respectively, T_f^{SR} is the cost for performing SR, T_p^{LGI} and T_p^{NNI} correspond to the cost of particle tracking using the LGI and NNI, respectively. Here, T_f^{DNS} and T_f^{LR} depend on the number of grid points, with $T_f^{\mathrm{LR}}/T_f^{\mathrm{DNS}} \approx 0$ for the present cases, and T_p^{LGI} and T_p^{NNI} depend on the number N_p of particles.

Our DNS was performed on the Tianhe-2 at the National Super Computer Center in Guangzhou, China (NSCC-GZ), with the Intel Xeon Gold 6150 processor. We ran 3×10^4 steps on 1024^2 grid points. The CNN SR was conducted on a NVIDIA Tesla V100 SXM2 GPU at NSCC-GZ. All computational costs are normalized by the running time of DNS $T_f^{\mathrm{DNS}} = 703$ s for the flow field.

Figure 11 plots the time costs of different methods for simulations of 2D particle-laden turbulence, where N_p varies from 1×10^4 to 5×10^5. The computational costs of both NNI and LGI scale linearly with N_p, and the cost of LGI is only slightly lower than the cost of NNI. For DNS and

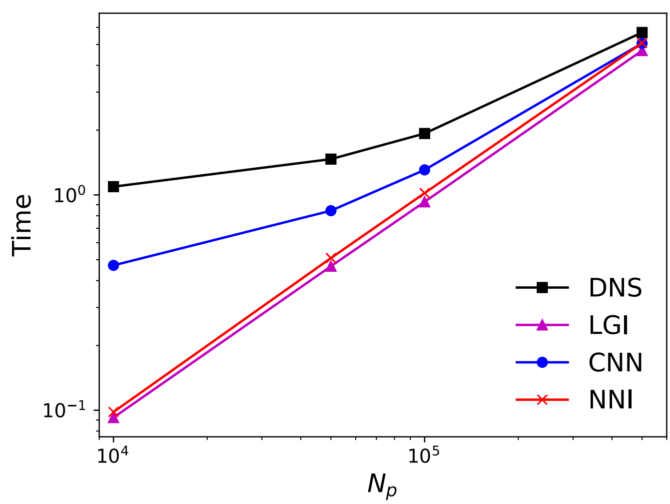

FIG. 11. Computational costs of different methods for simulating particle-laden two-dimensional (2D) homogeneous isotropic turbulence (HIT).

CNN, T_f^{DNS} and T_f^{SR} took a large part of the cost when N_p was not very large, e.g., $N_p = 10^4$. For both accuracy and efficiency, the NNI performs well in particle tracking in LR simulations.

V. CONCLUSIONS

We developed the NNI to improve the prediction of preferential concentration in the simulation of particle-laden turbulence. The NNI utilizes the particle position and velocity on neighboring grid points to calculate $\mathbf{u}(\mathbf{x}_p)$ via three FC-Nets. It avoids the SR of the entire flow field and additional interpolations, reducing the computational cost and simplifying the numerical implementation.

The NNI was trained and validated based on the DNS of particle-laden 2D HIT, and it was compared with the fourth-order Lagrangian interpolation and the LGI assisted with the CNN SR. The LR fields downsampled from the HR DNS fields are taken as the input flow field for the cases of LGI, CNN, and NNI. The present 2D turbulence exhibits a long inertial range, as the LR field filters a significant portion of small-scale motion at high wave numbers. It is challenging for the velocity interpolation and SR to accurately reproduce $\mathbf{u}(\mathbf{x}_p)$ and consequently predict the preferential concentration accurately.

Both CNN and NNI demonstrate good performance in the SR of flow field, recovering the majority of small-scale motions lost in the LR field. On the particle tracking, the NNI has proven to be effective in improving the prediction of preferential concentration. Despite the challenging case without most high-wave-number information in the LR fields, the NNI leverages the capabilities of NNs to estimate $\mathbf{u}(\mathbf{x}_p)$ with recovering SGS motions. This is critical to achieve an accurate prediction of the preferential concentration in particle-laden turbulent flows. The results confirm that the NNI results quantitatively reproduce particle distributions closer to the DNS results than the CNN results over a range of St from 0.1 to 5.0.

One of the key advantages of NNI over CNN is its flexibility. The NNI only requires the particle position and adjacent data to calculate the velocity at the particle position. This not only reduces the computational resources but also simplifies the implementation. The straightforward implementation of NNI using FC-Nets, which rely on matrix multiplication, further enhances its versatility and integration into regular simulation codes.

The NNI can be further improved in future work. The functional form in NNI does not ensure continuity between LR cells, which may result in discontinuities on the edges of a LR cell. This is also related to the limited number of 4×4 points to interpolate in NNI. The development of CNN SR indicates that expanding the interpolation range or integrating a convolutional network can mitigate this issue. Moreover, additional constraints can be applied to ensure that the NNI outcomes are more physical.

ACKNOWLEDGMENTS

Numerical simulations were performed on the TH-2A supercomputer in Guangzhou, China. This paper has been supported in part by the National Natural Science Foundation of China (Grants No. 11988102, No. 11925201, No. 52306126, and No. 92270203), the National Key R&D Program of China (Grant No. 2020YFE0204200), and the Xplore Prize.

APPENDIX: A POSTERIORI TEST

We validate the interpolation methods *a posteriori*. The LES of the 2D HIT is performed using the pseudospectral method [43] on a computational domain of $[0, 2\pi]^2$, with 128^2 grid points. In the LES, the spectral eddy hyperviscosity:

$$\nu_T = 1.5 \times 10^{-25}\left[0.267 + 9.21 \exp\left(-\frac{40.03 k_f^{\mathrm{LES}}}{k}\right)\right]\sqrt{\frac{E\left(k_f^{\mathrm{LES}}\right)}{\left(k_f^{\mathrm{LES}}\right)^{4p_u-3}}}, \tag{A1}$$

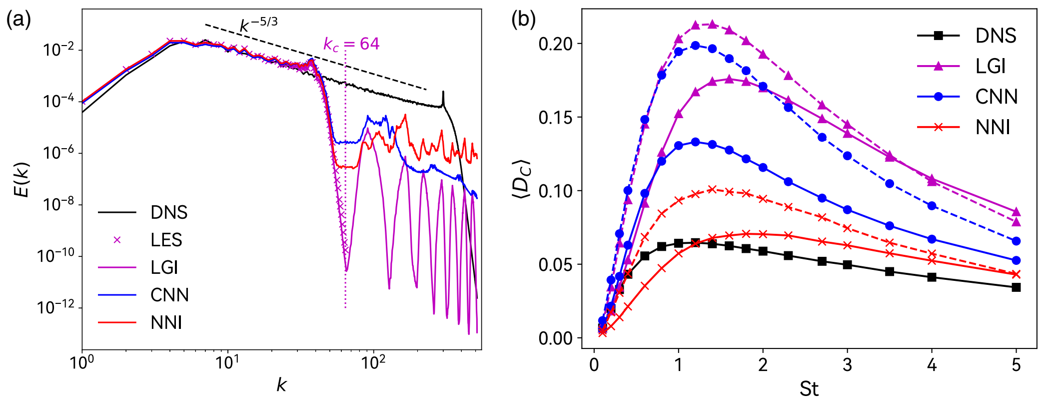

FIG. 12. (a) Energy spectra of the high-resolution (HR) field in direct numerical simulation (DNS), low-resolution (LR) field in large-eddy simulation (LES), and reconstructed LES fields with Lagrangian interpolation (LGI), convolutional neural network (CNN), and neural-network interpolation (NNI). The purple dotted line marks the cutoff wave number k_c. (b) Comparison of $\langle D_C \rangle$ obtained via LGI, CNN, and NNI based on the LR fields by the average pool (solid lines) in the *a priori* test and LES (dashed lines) in the *a posteriori* test for different St, along with the DNS results (black solid line).

is tuned based on the Chollet-Lesieur model [59]. A linear mapping $k_f^{\mathrm{LES}}/N^{\mathrm{LES}} = k_f^{\mathrm{DNS}}/N^{\mathrm{DNS}}$ determines the forcing wave number in the LES, where the superscripts denote the LES or DNS quantities. We apply the LGI, CNN, and NNI to the LES field for SR and particle simulations. The NNI and CNN are trained using the DNS data in Sec. II.

Reconstructing unresolved turbulence based on LES of 2D HIT poses a significant challenge [8]. Figure 12(a) plots the energy spectra of the HR field by DNS, the LR field by LES, and reconstructed fields by LGI, CNN, and NNI. Both CNN and NNI partially recover the SGS motion at $k > k_c$. On the other hand, the LES spectrum shows a steep decay near the cutoff wave number k_c, distinct from the LR field obtained by the average pool. Consequently, all three methods underestimate the energy spectrum at high wave numbers.

We assess the performance of different methods in predicting particle distributions. The $\langle D_C \rangle$ values obtained from LES with LGI, CNN, and NNI are compared with the DNS results for different St in Fig. 12(b). The NNI consistently outperforms the LGI and CNN in predicting the preferential concentration in the LES of particle-laden turbulence. In addition, all three *a posteriori* results have larger discrepancies from the DNS than the *a priori* results (see Fig. 9) due to the underestimated energy spectra at high wave numbers in the reconstructed LES fields [see Fig. 12(a)].

[1] L. Brandt and F. Coletti, Particle-laden turbulence: Progress and perspectives, Annu. Rev. Fluid Mech. **54**, 159 (2022).

[2] S. Balachandar and J. K. Eaton, Turbulent dispersed multiphase flow, Annu. Rev. Fluid Mech. **42**, 111 (2010).

[3] K. D. Squires and J. K. Eaton, Preferential concentration of particles by turbulence, Phys. Fluids **3**, 1169 (1991).

[4] L.-P. Wang and M. R. Maxey, Settling velocity and concentration distribution of heavy particles in homogeneous isotropic turbulence, J. Fluid Mech. **256**, 27 (1993).

[5] G. He, G. Jin, and Y. Yang, Space-time correlations and dynamic coupling in turbulent flows, Annu. Rev. Fluid Mech. **49**, 51 (2017).

[6] J. K. Eaton and J. R. Fessler, Preferential concentration of particles by turbulence, Int. J. Multiphase Flow **20**, 169 (1994).

[7] M. Obligado, T. Teitelbaum, A. Cartellier, P. Mininni, and M. Bourgoin, Preferential concentration of heavy particles in turbulence, J. Turbul. **15**, 293 (2014).

[8] S. Goto and J. C. Vassilicos, Particle pair diffusion and persistent streamline topology in two-dimensional turbulence, New J. Phys. **6**, 65 (2004).

[9] L. Chen, S. Goto, and J. C. Vassilicos, Turbulent clustering of stagnation points and inertial particles, J. Fluid Mech. **553**, 143 (2006).

[10] S. Goto and J. C. Vassilicos, Self-similar clustering of inertial particles and zero-acceleration points in fully developed two-dimensional turbulence, Phys. Fluids **18**, 115103 (2006).

[11] T. Faber and J. C. Vassilicos, Turbulent pair separation due to multiscale stagnation point structure and its time asymmetry in two-dimensional turbulence, Phys. Fluids **21**, 015106 (2009).

[12] R. Onishi and J. C. Vassilicos, Collision statistics of inertial particles in two-dimensional homogeneous isotropic turbulence with an inverse cascade, J. Fluid Mech. **745**, 279 (2014).

[13] M. Gitterman and V. Steinberg, Memory effects in the motion of a suspended particle in a turbulent fluid, Phys. Fluids **23**, 2154 (1980).

[14] M. R. Maxey, The gravitational settling of aerosol particles in homogeneous turbulence and random flow fields, J. Fluid Mech. **174**, 441 (1987).

[15] J. Bec, A. Celani, M. Cencini, and S. Musacchio, Clustering and collisions of heavy particles in random smooth flows, Phys. Fluids **17**, 073301 (2005).

[16] J. Chun, D. L. Koch, S. L. Rani, A. Ahluwalia, and L. R. Collins, Clustering of aerosol particles in isotropic turbulence, J. Fluid Mech. **536**, 219 (2005).

[17] L. I. Zaichik and V. M. Alipchenkov, Refinement of the probability density function model for preferential concentration of aerosol particles in isotropic turbulence, Phys. Fluids **19**, 113308 (2007).

[18] S. W. Coleman and J. C. Vassilicos, A unified sweep-stick mechanism to explain particle clustering in two- and three-dimensional homogeneous, isotropic turbulence, Phys. Fluids **21**, 113301 (2009).

[19] Y. Tagawa, J. M. Mercado, V. N. Prakash, E. Calzavarini, C. Sun, and D. Lohse, Three-dimensional Lagrangian Voronoï analysis for clustering of particles and bubbles in turbulence, J. Fluid Mech. **693**, 201 (2012).

[20] A. D. Bragg and L. R. Collins, New insights from comparing statistical theories for inertial particles in turbulence: I. Spatial distribution of particles, New J. Phys. **16**, 055013 (2014).

[21] A. D. Bragg, P. J. Ireland, and L. R. Collins, Mechanisms for the clustering of inertial particles in the inertial range of isotropic turbulence, Phys. Rev. E **92**, 023029 (2015).

[22] A. D. Bragg, P. J. Ireland, and L. R. Collins, On the relationship between the non-local clustering mechanism and preferential concentration, J. Fluid Mech. **780**, 327 (2015).

[23] X. Wang, M. Wan, Y. Yang, L.-P. Wang, and S. Chen, Reynolds number dependence of heavy particles clustering in homogeneous isotropic turbulence, Phys. Rev. Fluids **5**, 124603 (2020).

[24] Y. Yang, G. He, and L.-P. Wang, Effects of subgrid-scale modeling on Lagrangian statistics in large-eddy simulation, J. Turbul. **9**, N8 (2008).

[25] Y. Yang, G. He, and G. Jin, Prediction of particle distribution in isotropic turbulence by large-eddy simulation, in *New Trends in Fluid Mechanics Research*, edited by F. G. Zhuang and J. C. Li (Springer, Berlin, 2007), pp. 127–130.

[26] Z. Zhou, J. Chen, and G. Jin, Prediction of Lagrangian dispersion of fluid particles in isotropic turbulent flows using large-eddy simulation method, Acta Mech. **228**, 3203 (2017).

[27] Z. Y. Wan and T. P. Sapsis, Machine learning the kinematics of spherical particles in fluid flows, J. Fluid Mech. **857**, R2 (2018).

[28] Z. Zhou, G. He, S. Wang, and G. Jin, Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network, Comput. Fluids **195**, 104319 (2019).

[29] T. Davydzenka and P. Tahmasebi, High-resolution fluid-particle interactions: A machine learning approach, J. Fluid Mech. **938**, A20 (2022).

[30] B. Ray and L. R. Collins, Preferential concentration and relative velocity statistics of inertial particles in Navier-Stokes turbulence with and without filtering, J. Fluid Mech. **680**, 488 (2011).

[31] B. Ray and L. R. Collins, Investigation of sub-Kolmogorov inertial particle pair dynamics in turbulence using novel satellite particle simulations, J. Fluid Mech. **720**, 192 (2013).
[32] P. K. Yeung and S. B. Pope, An algorithm for tracking fluid particles in numerical simulations of homogeneous turbulence, J. Comput. Phys. **79**, 373 (1988).
[33] S. Balachandar and M. R. Maxey, Methods for evaluating fluid velocities in spectral simulations of turbulence, J. Comput. Phys. **83**, 96 (1989).
[34] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, Machine learning-accelerated computational fluid dynamics, Proc. Natl. Acad. Sci. USA **118**, e2101784118 (2021).
[35] K. Fukami, K. Fukagata, and K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, J. Fluid Mech. **870**, 106 (2019).
[36] B. Liu, J. Tang, H. Huang, and X.-Y. Lu, Deep learning methods for super-resolution reconstruction of turbulent flows, Phys. Fluids **32**, 025105 (2020).
[37] K. Fukami, K. Fukagata, and K. Taira, Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows, J. Fluid Mech. **909**, A9 (2019).
[38] H. Kim, J. Kim, S. Won, and C. Lee, Unsupervised deep learning for super-resolution reconstruction of turbulence, J. Fluid Mech. **910**, A29 (2021).
[39] Z. Zhou, B. Li, X. Yang, and Z. Yang, A robust super-resolution reconstruction model of turbulent flow data based on deep learning, Comput. Fluids **239**, 105382 (2022).
[40] M. Shirzadi, T. Fukasawa, K. Fukui, and T. Ishigami, Prediction of submicron particle dynamics in fibrous filter using deep convolutional neural networks, Phys. Fluids **34**, 123303 (2022).
[41] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. **378**, 686 (2019).
[42] D. A. Bezgin, S. J. Schmidt, and N. A. Adams, WENO3-NN: A maximum-order three-point data-driven weighted essentially non-oscillatory scheme, J. Comput. Phys. **452**, 110920 (2022).
[43] Z. Xiao, M. Wan, S. Chen, and G. L. Eyink, Physical mechanism of the inverse energy cascade of two-dimensional turbulence: A numerical investigation, J. Fluid Mech. **619**, 1 (2009).
[44] M. Y. Hussaini and T. A. Zang, Spectral methods in fluid dynamics, Annu. Rev. Fluid Mech. **19**, 339 (1987).
[45] W. J. T. Bos and J.-P. Bertoglio, Large-scale bottleneck effect in two-dimensional turbulence, J. Turbul. **10**, N30 (2009).
[46] R. H. Kraichnan, Inertial ranges in two-dimensional turbulence, Phys. Fluids **10**, 1417 (1967).
[47] A. G. Lamorgese, D. A. Caughey, and S. B. Pope, Direct numerical simulation of homogeneous turbulence with hyperviscosity, Phys. Fluids **17**, 015106 (2005).
[48] U. Frisch, S. Kurien, R. Pandit, W. Pauls, S. S. Ray, A. Wirth, and J.-Z. Zhu, Hyperviscosity, galerkin truncation, and bottlenecks in turbulence, Phys. Rev. Lett. **101**, 144501 (2008).
[49] A. van Kan, B. Favier, K. Julien, and E. Knobloch, Spontaneous suppression of inverse energy cascade in instability-driven 2-D turbulence, J. Fluid Mech. **952**, R4 (2022).
[50] Q. Wu, Y. Zhao, Y. Shi, and S. Chen, Large-eddy simulation of particle-laden isotropic turbulence using machine-learned subgrid-scale model, Phys. Fluids **34**, 065129 (2022).
[51] M. R. Maxey and J. J. Riley, Equation of motion for a small rigid sphere in a nonuniform flow, Phys. Fluids **26**, 883 (1983).
[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (The MIT Press, Cambridge, 2016).
[53] K. He, X. Zhang, S. Ren, and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015 IEEE International Conference on Computer Vision (ICCV) (Santiago, Chile, 2015), pp. 1026–1034.
[54] D. P. Kingma and J. L. Ba, Adam: A method for stochastic optimization, 3rd International Conference on Learning Representations (ICLR) (San Diego, 2015), p. 4.
[55] Y. Zhang, Z.-Q. J. Xu, T. Luo, and Z. Ma, A type of generalization error induced by initialization in deep neural networks, Proceedings of Machine Learning Research, 2020, Vol. 107, pp. 144–164.
[56] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, Frequency principle: Fourier analysis sheds light on deep neural networks, Commun. Comput. Phys. **28**, 1746 (2020).

[57] R. C. Hogan and J. N. Cuzzi, Stokes and Reynolds number dependence of preferential particle concentration in simulated three-dimensional turbulence, Phys. Fluids **13**, 2938 (2001).

[58] G. Jin, Y. Wang, J. Zhang, and G. He, Turbulent clustering of point particles and finite-size particles in isotropic turbulent flows, Ind. Eng. Chem. Res. **52**, 11294 (2013).

[59] M. Lesieur, *Turbulence in Fluids: Stochastic and Numerical Modelling* (Springer, Dordrecht, 1990).