Neural stochastic differential equations for particle dispersion in large-eddy simulations of homogeneous isotropic turbulence

J. Williams, U. Wolfram,* and A. Ozel∗

School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, UK

(Dated: 18 August 2022)

In dilute turbulent particle-laden flows, such as atmospheric dispersion of pollutants or virus particles, the dynamics of tracer-like to low inertial particles are significantly altered by the fluctuating motion of the carrier fluid phase. Neglecting the effects of fluid velocity fluctuations on particle dynamics causes poor prediction of particle transport and dispersion. To account for the effects of fluid phase fluctuating velocity on the particle transport, stochastic differential equations coupled with large-eddy simulation are proposed to model the fluid velocity seen by the particle. The drift and diffusion terms in the stochastic differential equation are modelled using neural networks (‘neural stochastic differential equations’). The neural networks are trained with direct numerical simulations (DNS) of decaying homogeneous isotropic turbulence at low and moderate Reynolds numbers. The predictability of the proposed models are assessed against DNS results through a priori analyses and a posteriori simulations of decaying homogeneous isotropic turbulence at low-to-high Reynolds numbers. Total particle fluctuating kinetic energy is under-predicted by 40% with no model, compared to the DNS data. In contrast, the proposed model predictions match total particle fluctuating kinetic energy to within 5% of the DNS data for low to high-inertia particles. For inertial particles, the model matches the variance of uncorrelated particle velocity to within 10% of DNS results, compared to 60-70% under-prediction with no model. It is concluded that the proposed model is applicable for flow configurations involving tracer and inertial particles, such as transport and dispersion of pollutants or virus particles.

* AO and UW share last authorship. Address correspondence to a.ozel@hw.ac.uk
I. INTRODUCTION

Dilute turbulent particle-laden flows are common in various natural and industrial processes, such as aerosol drug inhalation\textsuperscript{1–3}, pollutant dispersion in urban environments\textsuperscript{4}, and disease spread by cough plume dispersion\textsuperscript{5–7}. Due to low solid volume fraction and high density ratio between particles and carrier fluid (low particle mass loading), particle dynamics are mainly controlled by the carrier fluid. The fluid phase velocity fluctuations significantly alter particle transport and dispersion, and can fundamentally change how particles preferentially form in the carrier phase\textsuperscript{8}. The influence of the carrier fluid on particle dynamics is mainly characterised by the Stokes number, which is the ratio of particle drag timescale to the Kolmogorov timescale (timescale at which viscosity dominates)\textsuperscript{9–11}. Accurately capturing the influence of particle Stokes number on particle motion is a key challenge in dilute turbulent particle-laden flows\textsuperscript{12,13} due to interactions between fluid and particles ranging over several length and timescales\textsuperscript{14}. Understanding of particle transport in turbulent flows is crucial for designing engineering solutions to mitigate pollutant and virus dispersion.

Dilute turbulent particle-laden flows have been studied using computational modelling approaches such as particle-resolved direct numerical simulation (DNS)\textsuperscript{15–17}, where all temporal-spatial structures of flow are resolved without any models\textsuperscript{18}. However, particle-resolved DNS is computationally infeasible for high Reynolds number flows or tracking many particles, meaning coarser modelling approaches are required for engineering applications such as modelling aerosol inhalation. Mehrabadi et al.\textsuperscript{15} found point-particle DNS to agree well with particle-resolved DNS, particularly when mass loading was low and the particles have limited or no influence on the carrier fluid. In the point-particle approach, also called ‘unresolved’ Eulerian-Lagrangian, particles are tracked in a Lagrangian fashion by solving Newton’s equations of motion and an additional model is required for fluid-particle interactions\textsuperscript{18–21}. When modelling flows in complex geometries or at high Reynolds number, it is also infeasible to fully resolve all temporal-spatial structures of carrier fluid motion. Therefore, it is necessary to use the Reynolds-averaged Navier-Stokes or large-eddy simulation (LES) approach\textsuperscript{22,23}. In the Reynolds-averaged Navier-Stokes approach, the fluid fluctuating motion is modelled through additional transport equations for the evolution of variables such as turbulent kinetic energy and dissipation rate ($k$-$\varepsilon$)\textsuperscript{24–26}. In the LES approach, the large scales of turbulence are directly resolved and the scales smaller than the
characteristic mesh size are filtered out (see Germano\textsuperscript{27} for an overview of LES filtering). The effects of unresolved ‘subgrid’ scales are accounted for with a subgrid model\textsuperscript{23,27,28}. Reynolds-averaged Navier-Stokes and LES are both widely used for modelling fluid transport in simulations of particle-laden turbulent flows where the “resolved” fluid flow field is used to calculate the fluid-particle interactions (e.g. drag force). However, in both approaches there is an unresolved fluctuating fluid velocity that can influence the drag force acting on a particle, thereby causing an under-prediction of particle fluctuating kinetic energy for low-moderate inertia particles\textsuperscript{8,29}. As LES provides predictions of fluid phase fluctuations with higher accuracy, in this study we choose to focus on LES modelling of dilute particle-laden flows. However, neglecting the effect of subgrid fluid fluctuations on particle dynamics can cause errors in macroscopic quantities of the particle phase such as total particle fluctuating kinetic energy and preferential concentration\textsuperscript{8,29,30}. Therefore, we require an additional model to account for the effect of subgrid fluid fluctuations on the particle phase.

To account for subgrid fluid fluctuations, a Langevin-type stochastic differential equation has been developed to predict the ‘fluid velocity seen by a particle’ which is used to compute the drag force in dilute particle-laden turbulent flows\textsuperscript{31}. These models solve a stochastic differential equation (SDE) for the fluid velocity ‘seen’ by a particle, where the fluctuating component represents the randomness of the unresolved turbulence\textsuperscript{32,33}. Langevin-type stochastic models have been well developed for large-eddy simulations of single-phase reactive flows\textsuperscript{32,34–38} and Reynolds-averaged Navier-Stokes models of particle-laden turbulent flows\textsuperscript{31,39–45}. Recently, there have been studies on LES stochastic models for dilute particle-laden flows\textsuperscript{8,13,33,46–50}. However, closures in LES-SDE models have not been widely developed\textsuperscript{8,49,50}. Existing LES models are based on the simplified Langevin model for the fluid integral timescale\textsuperscript{13,48–51}, which has been used widely for Reynolds-averaged Navier-Stokes stochastic models of turbulent single-phase flow\textsuperscript{32}. However, Knorps and Pozorski\textsuperscript{50} showed in \textit{a priori} analyses of turbulent channel flow that the integral timescale approaches infinity when the particle approaches the viscous sub-layer in LES-SDE models. As the timescale approaches infinity, particles entering the viscous sub-layer will become trapped as the time taken for a particle to respond to changes in the local flow is infinitely large. As the timescale approaches infinity, particles transitioning from turbulent to laminar flow regions will maintain their initial velocity for an infinite period of time and fail to adapt
to the local fluid velocity. In decaying homogeneous isotropic turbulence, this would mean that the predicted decay of particle fluctuating kinetic energy is not consistent with the fluid as the particle velocity does not continue to decay. Knorps and Pozorski \textsuperscript{50} bypassed this by applying a near-wall damping for the timescale\textsuperscript{52}, which is not applicable in unbounded cases such as homogeneous isotropic turbulence. Therefore, new models that are consistent with the underlying LES fluid solver are needed\textsuperscript{53}, in order to avoid unphysical predictions of kinetic energy in decaying turbulent flows\textsuperscript{54}.

Data-driven methods can be used as an alternative approach to developing closures for unknown terms in LES-SDE models. The merit of data-driven turbulence models has been demonstrated for (single-phase) fluid turbulence\textsuperscript{55–59} and passive scalar dynamics\textsuperscript{60,61}, where the data-driven approaches produced improved results compared to state-of-the-art closure models from theoretical derivation. By using data from highly resolved simulations (such as point-particle DNS)\textsuperscript{56,62–64}, a neural network can learn a model for the evolution of a variable, such as the fluid velocity seen by a particle, in simulations of lower resolution. Recently, stochastic models for particle dynamics have been inferred with neural networks to create ‘neural SDEs’\textsuperscript{64–68}. Neural SDEs have been applied to learn drift and diffusion terms of a SDE model for Brownian motion\textsuperscript{65} and epidemic dynamics\textsuperscript{64} from particle observations at different time instants. Thus, neural SDEs provide a suitable framework to learn unknown physics required for model closures in LES-SDE models of dilute particle-laden flows.

Therefore, in this study we aim to develop data-driven closures for stochastic models of particle dispersion in decaying homogeneous isotropic turbulence. To do this, we first generate a \textit{ground truth} database, which is constructed on DNSs of decaying homogeneous isotropic turbulence. We then use DNS data to train neural networks to learn unknown terms in our stochastic model for the fluid velocity seen by particles. The developed model is assessed through \textit{a priori}, and \textit{a posteriori} analyses of decaying homogeneous isotropic turbulence at various Reynolds numbers, including a higher Reynolds number, which is not included in training. Our computational framework including OpenFOAM-based fluid-particle solvers and python scripts for neural network model training is publicly available online in a GitHub repository (https://github.com/jvwilliams23/turbulent-dispersion-neuralSDE). Data for training is also available online through Kaggle\textsuperscript{69}. 

4
II. MATERIALS AND METHODS

A. **Ground truth: Direct numerical simulations of decaying homogeneous isotropic turbulence**

To generate database for the neural SDE model training, we performed DNS in decaying homogeneous isotropic turbulence at $Re_t = 0$ with one-way coupling on an equispaced $N_{cell} = 256^3$ cell grid occupying a cubic domain of size $2\pi m$ in each axis, where $N_{cell}$ is the number of cells. Simulations were performed in OpenFOAM v6.70. The transport equations for the fluid phase are given as

\[
\nabla \cdot \mathbf{u}_f = 0, \\
\frac{\partial \mathbf{u}_f}{\partial t} + \mathbf{u}_f \cdot \nabla \mathbf{u}_f = -\frac{1}{\rho_f} \nabla p + \nabla \cdot \mathbf{\tau}
\]

where $\mathbf{u}_f$ is the fluid velocity, $p$ is the fluid pressure, $\rho_f$ is the fluid density and $\mathbf{\tau}$ is the viscous stress tensor, given by

\[
\mathbf{\tau} = \nu_f \left[ \nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T - \frac{2}{3} (\nabla \cdot \mathbf{u}_f) \mathbf{l} \right],
\]

where $\nu_f$ is the dynamic viscosity, and $\mathbf{l}$ is the identity tensor.

The dispersed phase is tracked in the Lagrangian reference frame by following Newton’s equations of motion. Considering only drag (gas-solid flows where $\rho_p >> \rho_f$), the evolution of a particle’s position, $\mathbf{x}_p$ and velocity $\mathbf{u}_p$ is

\[
d\mathbf{x}_p = \mathbf{u}_p \ dt \\
d\mathbf{u}_p = \frac{\mathbf{u}_{f\@p} - \mathbf{u}_p}{\tau_p} \ dt
\]

where $\mathbf{u}_{f\@p}$ is the fluid velocity at particle position and $\tau_p$ is the time for a particle to respond to changes in the velocity of the carrier fluid (called drag response time), which is defined as

\[
\tau_p = \frac{\rho_p}{\rho_f} \frac{4 d_p}{3 C_D |\mathbf{u}_r|}
\]

where $\rho_p$ is the particle density, $d_p$ is the particle diameter and $\mathbf{u}_r$ is the relative velocity ($\mathbf{u}_{f\@p} - \mathbf{u}_p$). The drag coefficient was given by Schiller and Naumann as $C_D = \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687})$, where $Re_p = |\mathbf{u}_r| d_p / \nu_f$.

The simulation timestep was based on the fluid Courant number, which was set be a maximum of 0.272. Fluid properties at the particle position were taken as the cell value,
as we observed the trilinear interpolation in OpenFOAM to have considerable numerical error (Supplementary material). The number of particles in the domain, \( N_p \), was \( N_p = 10^5 \) and the particles were forced to be stationary in time to remove noise in fluid properties at particle position when the particle drifts into a neighbouring cell. To create the initial flowfield, we created an initial divergence-free velocity field with the OpenFOAM createBoxTurb function. The initial energy spectrum followed the relationship for energy with wavenumber, \( E(\kappa) \), provided by Kang, Chester, and Meneveau\(^{73}\), with 10,000 Fourier modes. We created the spectra using a Python tool for synthetic turbulence generation named TurboGenPY\(^{74}\). The flow statistics at the beginning of the simulation are presented in Table I. The resolution of the simulation can be assessed by multiplying the largest wave number \( \kappa_{\text{max}} \) by the Kolmogorov length-scale, \( \eta_k = \nu^3/\varepsilon^{1/4} \) where \( \varepsilon \) is the resolved scale kinetic energy dissipation rate\(^{75}\). For adequate numerical resolution of the scales of motion in DNS, \( \kappa_{\text{max}} \eta_k > 1 \) is required\(^{75,76}\). In Figure 1(a) we show the kinetic energy against \( \kappa \eta_k \), which shows that \( Re_{\lambda}^{t=0} = 9 \) is well-resolved (\( \kappa_{\text{max}} \eta_k = 1.5 \)) and \( Re_{\lambda}^{t=0} = 33 \) is slightly under-resolved (\( \kappa_{\text{max}} \eta_k = 0.8 \)). The simulation is more under-resolved for \( Re_{\lambda}^{t=0} = 105 \) (\( \kappa_{\text{max}} \eta_k = 0.45 \)), where an estimated number of cells required to satisfy \( \kappa_{\text{max}} \eta_k \) is \( N_{\text{cell}} \approx 600^3 \) (\( \kappa_{\text{max}} = \sqrt{2} N_{ci}/3 \) where \( N_{ci} \) is the number of cells in one axis\(^{75}\)). We therefore did not include \( Re_{\lambda}^{t=0} = 105 \) in training and reserve it for testing our model. In Figure 1(b), we show the kinetic energy spectra for DNS and filtered data (described below). We observe that for all mesh resolutions, the large scale features of the flow are retained, including a peak energy at \( \kappa \approx 5 \).

1. **Filtering procedure for training data**

The database for LES-SDE model training was deduced by filtering the results from DNS simulations. The filtered fluid velocity is defined as\(^{77}\)

\[
\tilde{u}_f = \int_{\mathbb{R}^3} g(r - x) u_f(r, t) dr.
\]  

where \( g(r - x) \) is a weight function which satisfies \( \int_{\mathbb{R}^3} g(r) dr = 1 \). The box filter kernel is given by

\[
g(r - x) = \begin{cases} 
\frac{1}{\Delta f}, & \text{if } |r - x| \leq \frac{\Delta f}{2} \\
0, & \text{otherwise.}
\end{cases}
\]
TABLE I. Summary of flow parameters for decaying homogeneous isotropic turbulence test cases.

| Parameter                                         | Value                          |
|---------------------------------------------------|--------------------------------|
| Microscale Reynolds number, $Re_{\lambda} = 0$ [-] | 9 / 33 / 105                   |
| Large-scale kinetic energy, $k^{t=0}$ [m$^2$/s$^2$] | 0.36 / 4.7 × 10$^{-5}$ / 4.7 × 10$^{-4}$ |
| Large-scale kinetic energy dissipation rate, $\varepsilon^{t=0}$ [m$^2$/s$^3$] | 4.6 / 1.86 × 10$^{-5}$ / 1.86 × 10$^{-5}$ |
| Large-scale turbulent timescale, $\tau_L^{t=0}$ [s] | 0.079 / 25.1 / 25.1            |
| Kolmogorov timescale, $\tau_k^{t=0}$ [s]          | 0.033 / 2.9 / 0.9              |
| Kolmogorov length-scale, $\eta_k^{t=0}$ [m]       | 0.013 / 0.0068 / 0.0038        |
| Dynamic viscosity, $\nu_f$ [Pa · s]                | 0.005 / 1.57 × 10$^{-5}$ / 1.57 × 10$^{-5}$ |
| Particle diameter, $\rho_p$ [kg/m$^3$]            | 1000                           |
| Particle Stokes number, $St^{t=0} = \tau_p/\tau_k^{t=0}$ [-] | 0.1 / 1 / 5                   |

FIG. 1. Initial energy spectra in homogeneous isotropic turbulence. Panel (a) shows the effect of varying $Re_{\lambda}^{t=0}$ on the cutoff wavenumber relative to the Kolmogorov length in our DNS. Panel (b) shows effect of filter width on amount of large scale energy resolved in simulation.

We applied four widths of $\Delta/\Delta_{DNS} = \{3, 5, 7, 9\}$. Training data consisted of using 2000 particles at each timestep from the $Re_{\lambda}^{t=0} = 33$ cases (30 timesteps) and 1000 particles from the $Re_{\lambda}^{t=0} = 9$ case (20 timesteps). This means that the total number of particles
Train on 2000 particle trajectories

Simulation timeline

FIG. 2. Schematic overview of timeline for data extraction from DNS. Particle properties ($u_p$) and fluid properties at the particle position ($u_{f@p}$, $k_{sgs@p}$, $\varepsilon_{sgs@p}$, $\bar{u}_{f@p}$) were written from $t_0$ to $t_N$. At each saved timestep, we record properties at the start and end of timestep ($[t_n, t_n + \Delta t_{DNS}]$, where $n = 1, ..., N$).

included in training is only 1-2% of the total number of particles available from the DNS data. All filter widths were included in training. This gave 320,000 data points for training, of which, 70% was used for training and 30% for testing. For a priori testing, we used data from filtered DNS as model inputs and advanced the transport equation from our neural SDE one timestep ($t_n \rightarrow t_n + \Delta t_{DNS}$) to eliminate error from time integration over a large timestep. To train the neural SDE, as discussed later, we minimise the difference between neural SDE transport equation integrated over $t_n \rightarrow t_{n+1}$ (write-time interval) and the ground truth at $t_{n+1}$ (Figure 2). This is due to the practical limitation, that the change in velocity over the interval $\Delta t_{DNS}$ is small and impractical for training (discussed below).

B. LES modelling and particle dynamics

To assess particle dispersion a posteriori, we performed LES on mesh resolutions $N_{cell} = \{85^3, 51^3, 36^3, 28^3\}$, where the cell size is approximately the same as the filter width in our a priori filtering ($\Delta / \Delta_{DNS} = \{3, 5, 7, 9\}$). The particles are transported following a system of stochastic differential equations (SDEs). To focus on dilute particle-laden flows, we assume that there is no effect of particles on carrier phase dynamics (one-way coupling)$^{14}$. Therefore,
the volume-filtered equations of motion for the carrier phase are described as

\[ \nabla \cdot \tilde{\mathbf{u}}_f = 0, \]  

(9)

\[ \frac{\partial \tilde{\mathbf{u}}_f}{\partial t} + \tilde{\mathbf{u}}_f \cdot \nabla \tilde{\mathbf{u}}_f = -\frac{1}{\rho_f} \nabla \tilde{p} + \nabla \cdot (\tilde{\mathbf{\tau}} - \mathbf{\tau}_{sgs}), \]  

(10)

where bold lower-case symbols represent a vector, bold upper-case symbols represent a second-order tensor (unless stated otherwise). \((\tilde{\cdot})\) represents a volume-filtered field, \(\tilde{\mathbf{u}}_f\) is the filtered fluid velocity, \(\tilde{p}\) is the filtered fluid pressure, \(\tilde{\mathbf{\tau}}\) is the filtered viscous stress tensor, and \(\mathbf{\tau}_{sgs}\) is the subgrid viscous stress from filtering. The volume-filtering is achieved by applying the top-hat filter following (8), where \(\Delta = \sqrt[3]{V_c}\) where \(V_c\) is the volume of a computational cell\(^78,79\). We modelled \(\mathbf{\tau}_{sgs}\) by an eddy-viscosity concept\(^22\) as:

\[ \mathbf{\tau}_{sgs} = \frac{2}{3} k_{sgs} \mathbf{l} - 2 \nu_{sgs} \tilde{\mathbf{S}}, \]  

(11)

where \(\nu_{sgs}\) is the subgrid viscosity, \(\tilde{\mathbf{S}} = (\nabla \tilde{\mathbf{u}}_f + (\nabla \tilde{\mathbf{u}}_f^T))/2\) is the filtered, or ‘resolved’, strain rate tensor, and \(k_{sgs}\) is the subgrid kinetic energy. The subgrid viscosity and kinetic energy are computed using the WALE model proposed by Nicoud and Ducros\(^80\). This model is based on the filter width and invariants of the velocity gradient tensor, giving \(k_{sgs}\) as

\[ k_{sgs} = \left( \frac{C_w ^2 \Delta}{C_k} \right) ^2 \frac{\left( \left| \tilde{\mathbf{S}}^d \cdot \tilde{\mathbf{S}}^d \right| \right)^3}{\left( \left| \tilde{\mathbf{S}} \cdot \tilde{\mathbf{S}} \right| \right)^{5/2} + \left( \left| \tilde{\mathbf{S}}^d \cdot \tilde{\mathbf{S}}^d \right| \right)^{5/4}}^2, \]  

(12)

where \(\tilde{\mathbf{S}} = (\nabla \tilde{\mathbf{u}}_f^2 + [(\nabla \tilde{\mathbf{u}}_f^2)^T])/2\), \((\cdot)^d\) is the deviatoric part of a tensor, \(| \cdot |\) is the magnitude which for a second-order tensor, \(\mathbf{T}\), we find by \(|\mathbf{T}| = \sqrt{\mathbf{T} : \mathbf{T}}\), wherein : denotes the double scalar product. Modelling constants are \(C_w = 0.325\) and \(C_k = 0.094\)^\(^80\). The subgrid kinetic energy dissipation rate, \(\varepsilon_{sgs}\), was computed by

\[ \varepsilon_{sgs} = C_\varepsilon \frac{k_{sgs}^{3/2}}{\Delta}, \]  

(13)

where \(C_\varepsilon = 1\) is a model parameter\(^81\).

The transport equation for particle velocity remains as Equation (5), where \(\mathbf{u}_{f_{\partial p}}\) is replaced by \(\mathbf{u}_s\), which is the fluid velocity seen by a particle. For LES without our dispersion model, \(\mathbf{u}_s = \tilde{\mathbf{u}}_{f_{\partial p}}\), where \(\tilde{\mathbf{u}}_{f_{\partial p}}\) is the filtered fluid velocity at the particle position.
C. SDE for the fluid velocity seen by a particle

The fluid velocity seen by a particle, \( u_s \), is typically obtained by interpolating the ‘resolved’ fluid phase velocity (from either LES or Reynolds-averaged Navier-Stokes modelling) to the particle position\(^2^,\!^3,\!^30,\!^82,\!^83 \). However, this provides erroneous particle trajectories, particularly for low inertia particles whose dynamics are mainly controlled by turbulent fluctuations\(^8,\!^{13},\!^30,\!^49 \). These fluctuations can be accounted for with a stochastic model with LES modelling given as\(^49 \)

\[
du_s = -\frac{1}{\rho_f} (\nabla \tilde{p})_{\tilde{g}p} \, dt + \nu_f (\nabla \cdot \tilde{u}_f)_{\tilde{g}p} \, dt + (\tilde{u}_{p\tilde{g}p} - \tilde{u}_{f\tilde{g}p}) \cdot (\nabla \tilde{u}_f)_{\tilde{g}p} \, dt \\
- G \cdot (u_{s} - \tilde{u}_{f\tilde{g}p}) \, dt + B \cdot d\mathbf{w} 
\]

(14)

where \( \tilde{u}_{p\tilde{g}p} \) is the Eulerian particle velocity at the particle position obtained from ensemble averaging over all particles in a cell using the same filter kernel for LES filtering as shown in Equation (8). \( d\mathbf{w} \) is a vector of independent random variables sampled from a Gaussian distribution \( (N(0, dt)) \) named a Wiener process\(^84,\!^85 \). This stochastic process represents the disordered nature of the unresolved scales which occur at a timescale much smaller than the observed time interval, \( dt \).\(^41 \). \( G \) and \( B \) are drift and diffusion tensor coefficients, respectively. As the flow configuration is homogeneous isotropic turbulence, the mean velocity is zero in all directions, and the drift and diffusion parameters are isotropic\(^22 \). Therefore, the model parameters can be treated as \( G = GI \) and \( B = BI \). The drift coefficient is related to the fluid timescale, \( T \) by \( T = 1/G \) as described by Pope\(^22 \). In this study, we determine the unclosed terms \( G \) and \( B \) with a neural network for each term, \( G = G^{NN} \) and \( B = B^{NN} \).

1. Neural network model

To construct a data-driven model, we first identify the flow features that should be used as model inputs. Haworth and Pope\(^32 \) stated that the model terms \( G \) and \( B \) depend on the turbulent kinetic energy, \( k_{sgs} \), the turbulent dissipation rate, \( \varepsilon_{sgs} \), and velocity gradients which we exclude here as the flow is homogeneous with no spatial gradients. Additionally, in the LES approach there is a dependence on the filter width, \( \Delta \) (where \( \Delta = \sqrt[3]{V_c} \), \( V_c \) is the volume of a computational cell). Therefore, our model inputs are the subgrid time scale \( \tau_{s}^{*} = \tau_{s}/\tau_{k} = k_{sgs}/(\varepsilon_{sgs} \tau_{k}) \), and subgrid length scale \( \Delta^{*} = \Delta/\eta_{sgs} \). We have
normalised our model inputs using the Kolmogorov length scale \( \eta_{sgs} \) (\( \eta_{sgs} = \sqrt{\frac{\nu^3}{\varepsilon_{sgs}@p}} \)), where \( \nu_f \) is the kinematic viscosity and time scale, \( \tau_k = \sqrt{\frac{\nu_f}{\varepsilon_{sgs}@p}} \). The Kolmogorov velocity scale is \( u_k = \frac{\eta_{sgs}}{\tau_k} \).

The model can be discretised in time using the numerical scheme presented by Minier, Peirano, and Chibbaro. This scheme is an adaptation of the explicit Euler-Maruyama scheme for SDEs which uses an exponential for the ratio between the timestep and fluid or particle timescale (such as \( \exp(-\Delta t/\tau_p) \)), which makes the scheme stable for any combination of \( \Delta t/\tau_p \) or \( \Delta /T \). The scheme is given as follows:

\[
\begin{align*}
\mathbf{u}^{n+1}_{s(i)} &= \mathbf{u}^n_{s(i)} \exp(-\Delta t/T^n) + \mathbf{c}^n_{(i)} T^n \left[ 1 - \exp(-\Delta t/T^n) \right] + \gamma^n_{(i)}, \\
c^n_{(i)} T^n &= \left[ -\nabla \tilde{p} + \nu_f \nabla \cdot \tilde{\mathbf{u}}_{f@p(i)} + \left( \mathbf{u}_{p(i)} - \tilde{\mathbf{u}}_{f@p(i)} \right) \cdot \nabla \tilde{\mathbf{u}}_{f@p(i)} \right] T^n + \tilde{\mathbf{u}}_{f@p(i)}, \\
\gamma^n_{(i)} &= \sqrt{0.5 (B^{NN})^2 T^n \left[ 1 - \exp(-2\Delta t/T^n) \right]} \xi_{(i)},
\end{align*}
\]

where the notation \( (\cdot)^n \) represents the \( n \)th component of a vector, and the notation \( (\cdot)^n \) represents the \( n \)th timestep. \( \xi_{(i)} \) is a random value sampled from a distribution \( \mathcal{N}(0,1) \).

Our target is to learn the closures for \( G^{NN} \) and \( B^{NN} \) such that the \( \mathbf{u}_s \) predicted by our integration of Equation (15) is the same as \( \mathbf{u}_s \) obtained from DNS simulations (Figure 3). Therefore, we learn the increment of \( d\mathbf{u}_s \) between one timestep where simulation data has been saved, \( t = n \), and the next timestep where simulation data was saved \( t = n + 1 \). For DNS, it is infeasible to save all timesteps due to storage limitations. Therefore, simulation data was saved at intervals of 40 DNS timesteps as shown in the timeline in Figure 2. We found it to be more effective to learn using the change in \( \mathbf{u}_s \) over this large interval rather than a single timestep as the change in \( \mathbf{u}_s \) over a single DNS timestep was very small.

When training a neural network, this can cause a so-called vanishing gradients issue where the training does not converge, or converges to an incorrect solution. Therefore, we minimise the following loss function based on the evolution of the SDE (15) from timestep
\[ t = n \text{ to } t = n + 1: \]

\[
\mathcal{L} = \frac{1}{N_s} \sum_{i=1}^{N_s} \left( (u_{n+1,NN}^{s,(i)} - u_{n+1,\text{data}}^{s,(i)})^2 
+ \log \left( \frac{[u_{t=n+1,\text{data}}^{s,(i)} - (u_{n,\text{data}}^{s,(i)} \exp(-\Delta t/T^n) + c_n^{s,(i)} T^n [1 - \exp(-\Delta t/T^n)])]^2}{0.5 (B^{NN})^2 T^n [1 - \exp(-2\Delta t/T^n)]} \right) \right),
\]

(18)

where \( N_s \) is the number of samples used for training (one particle-timestep pair is one sample). The first term in the loss function compares the mean-squared error in the output of the NN with the data, which requires a sufficiently large batch size and number of samples to minimise the statistical error in the random term, \( \gamma \) (17). Therefore, to minimise statistical error we took the average of the first term in (18) over 50 independent realisations (50 separate values of \( \xi \)). The second term in (18) calculates the distance from \( u_{t=n+1,\text{data}}^{s,(i)} \) and the SDE distribution

\[
\mathcal{N}(u_{n,\text{data}}^{s,(i)} \exp(-\Delta t/T^n) + c_n^{s,(i)} T^n [1 - \exp(-\Delta t/T^n)]), 0.5 (B^{NN})^2 T^n [1 - \exp(-2\Delta t/T^n))].
\]

(19)

We take the logarithm of this to prevent huge fluctuations in the loss as the diffusion term approaches zero (exploding gradients issue\(^8^9\)).

Results created from deep learning architectures such as neural networks are sensitive to parameters such as the number of layers, number of nodes per layer, the activation functions and the initial condition of the weights and biases. We determined these so-called hyperparameters through manually tuning. From this, we found a suitable architecture to be four layers with 120 nodes per layer. The hidden layer biases were initialised to be zero, and weights were initialised as \( \mathcal{N}(0, w_h^2) \) where \( w_h = \sqrt{2/n_{\text{node},j}} \) and \( n_{\text{node},j} \) is the number of nodes in the input weight tensor for layer \( j \).\(^9^0\) The drift term neural network used ReLu activation function for hidden layers, where the output of each node in the hidden layer, \( y \), is \( y = \max(0, wx + b) \) (for a simple 1 node layer, where \( w \) is the node weight, \( x \) is the node input, and \( b \) is the node bias). The diffusion term neural network used hyperbolic tangent activation function for the hidden layers. The output layer weights for \( G^{NN} \) were initialised as \( \mathcal{U}(-w_h, w_h) \). The output layer weights for \( B^{NN} \) were initialised as \( \mathcal{U}(-w_h, w_h) \). The output layer biases were fixed at zero in all networks trained. The output layers used a softplus activation function to ensure that \( G^{NN} \) and \( B^{NN} \) were positive-definite. The network was trained stochastic gradient-descent with a learning rate of \( 10^{-4} \).
FIG. 3. Schematic overview of neural SDE training and architecture. Simulation is trained on interval $t = n$ to $t = n + 1$. Network inputs are given as green nodes, and outputs are blue nodes. The output parameters, $G^{NN}$ and $B^{NN}$ are then used to compute $\mathbf{u}_s^{NN}$ using filtered DNS data. Loss $\mathcal{L}$, Equation (18), is computed by comparing $\mathbf{u}_s^{NN}$ to DNS data $\mathbf{u}_s^{data} = \mathbf{u}_f$.

We implemented our neural network modelling framework using Keras and Tensorflow 2.4. Our code was based on the code provided on GitHub by Dietrich et al.

Finally, as training a neural network is sensitive to local minima, some steps must be taken to optimise this process. The first step being to normalise our NN inputs to below one to minimise any large ‘exploding’ gradients. Therefore, we scale the normalised timescale as $\tau_{sgs}/C_\tau$ and the length-scale as $\Delta^*/C_\Delta$, where $C_\Delta$ and $C_\tau$ were chosen such that the maximum value of the normalised length-scale and timescale in the training set was 1. Additionally, optimisation algorithms such as stochastic gradient descent converge best if the initial output is near the expected, correct value. This cannot be achieved by applying
a constant scaling value, as when the flow becomes laminar, the drift term $G^{NN}$ should approach infinity ($T^{NN} \to 0$) and $B^{NN}$ should approach zero. Through trial and error, we found the following NN scalings to produce a close initial output which converged to a correct solution:

$$G^{NN} = \sqrt{\nu} \left[ f^{NN} \left( \frac{\tau_{s\Delta}}{C\Delta}, \frac{\Delta^*}{C\Delta} \right) \right]^{-1} \frac{C_G C_{\tau}}{\tau_{s\Delta}^*},$$

(20)

$$B^{NN} = \nu f^{NN} \left( \frac{\tau_{s\Delta}}{C\Delta}, \frac{\Delta^*}{C\Delta} \right) \frac{C_B \Delta^*}{C\Delta},$$

(21)

where $f^{NN}(\cdot)$ is a non-linear function output from a neural network, that depends on the parameters enclosed in brackets. $C_G$ and $C_B$ were chosen empirically as 5 and 1000. These constants and scaling were chosen to provide a NN that produced a kinetic energy decay close to the target value (DNS data) at initialisation (epoch 0). Scaling $G^{NN}$ with $1/\tau_{s\Delta}^*$ ensured that the integral scale $T^{NN} = 1/G^{NN} \to 0$ when the flow becomes laminar.

### III. RESULTS

#### A. A priori analyses

First we assessed the model by a priori analysis at varying filter width in the $Re_{\lambda}^{t=0} = 33$ case. We quantified model performance by comparing the fluctuating kinetic energy ($k = tr[(u - \langle u \rangle) \otimes (u - \langle u \rangle)]$ where $\langle u \rangle$ is the domain-averaged velocity, which is equal to zero) for tracer particles ($u_{s} = u_p$) (Figure 4). This is compared to the fluctuating kinetic energy based on the DNS tracer particle velocity, $u_{f\theta_p}$, and based on the filtered DNS tracer particle velocity, $\tilde{u}_{f\theta_p}$. Our neural SDE recovered the DNS kinetic energy at all filter widths shown. For larger filter widths, the effect of unresolved fluctuations on the kinetic energy became more apparent. In the largest filter width, the relative error in $k$ compared to the DNS data at the first time step was 0.5% for $u_{s}^{NN}$ and 41.3% for $\tilde{u}_f$ (without the model). For the smallest filter width, the relative error in $k$ in the first time step was 0.3% for our neural SDE. The relative error in the filtered DNS data compared to the DNS data was 8% in the first time-step. These results show that the dispersion model recovers filtered out scales in simulations with a range of filter widths.

To assess generalisation to varying homogeneous isotropic turbulence flow configurations, we performed a priori tests at varying $Re_{\lambda}^{t=0}$ (Figures 4 and 5). For all three $Re_{\lambda}^{t=0}$ tested,
our developed neural SDE gave good agreement with the DNS data. The maximum error was \(< 1\%\) for \(Re^t=9\) (Figure 4c), 7\% for \(Re^t=33\) (Figure 5a), 4\% for \(Re^t=105\) (Figure 5b). In contrast, the maximum error in the filtered dataset with no model was
FIG. 5. Decay of tracer particle fluctuating kinetic energy in a priori tests of homogeneous isotropic turbulence at (a) \( Re_\lambda^t=0 = 9 \) and (b) \( Re_\lambda^t=0 = 105 \) for a filter size, \( \Delta = 7\Delta_{DNS} \). Flow configuration of homogeneous isotropic turbulence is given in Table I.

31% for \( Re_\lambda^t=0 = 9 \) and 47% for \( Re_\lambda^t=0 = 33 \) and \( Re_\lambda^t=0 = 105 \). This a priori analysis showed a significant improvement in the accuracy of fluctuating kinetic energy from using the proposed neural SDE dispersion model. Furthermore, the model is also consistent with the decay from turbulent to laminar flow suggesting that the neural SDE integral timescale is consistent with the underlying fluid solver\(^{53,54}\).

Particle fluctuating kinetic energy decay was assessed through a priori analyses at varying Stokes number, \( St^t=0 = \tau_p/\tau_k^t=0 \) (Figure 6). For low inertia particles (\( St^t=0 = 0.1 \), Figure 6a), the NN model predicted particle fluctuating kinetic energy to within 5% error throughout the simulation and the filtered DNS data under-predicts particle fluctuating kinetic energy by a maximum of 19%. For high inertia particles (\( St^t=0 = 5 \), Figure 6b), the results from both filtered tests (with and without our model) match the DNS data within 1%. This agrees with literature as larger particles with large drag response times are less sensitive to fast fluctuations in the carrier fluid, meaning that turbulent dispersion is not influential\(^{14}\).

From the presented a priori analyses, we observe that our model is independent of filter width. In contrast, the filtered kinetic energy with no model showed high mesh dependence.
FIG. 6. Decay of inertial particle fluctuating kinetic energy in a priori tests of homogeneous isotropic turbulence at (a) $St^{t=0} = 0.1$ (low inertia), (b) $St^{t=0} = 5$ (high inertia). Results show $Re_{\lambda}^{t=0} = 33$ at filter size $\Delta = 7\Delta_{DNS}$. For $St^{t=0} = 5$, all markers are overlapping.

Additionally, we have shown that our model works for a range of $Re_{\lambda}^{t=0}$, including high $Re_{\lambda}^{t=0} = 105$ which is also not included in the training data. Finally, we see the effect of the model on inertial particles, which showed that low inertia particles are influenced by the dispersion model. The influence of filtering was found to be negligible for high inertia particles, which agrees with literature$^8$.

**B. A posteriori analyses**

To assess particle dispersion a posteriori, we performed simulations of tracer dispersion to assess the performance of the neural SDE in the presence of additional errors such as subgrid modelling and numerical discretisation. This will alter the evolution of fluid properties used as network inputs and is therefore a necessary test to ensure robustness$^9$. The domain had the same dimensions and initial parameters described in Section II A. Similar to the a priori tests, the a posteriori simulations show good agreement with the DNS data for the LES with NN for all mesh resolutions, $N_{cell} = \{85^3, 51^3, 36^3, 28^3\}$ (Figure 7). The LES without the model cases also agree well with the DNS data for high mesh resolution, $N_{cell} = 85^3$
(Figure 7a). However, LES on the coarser grids show an under-prediction in $k$ (Figure 7b-d). For $9\Delta_{DNS}$ (Figure 7d), our NN under-predicted $k$ by 14% at the start of the simulation ($t/\tau_L^t = 0.1$) compared to 40% under-prediction for LES alone without the model.

We have assessed whether the $u_s$ model is mesh independent in a posteriori analyses (see Figure 7). We performed simulations on a coarse mesh at larger $Re^t_{\lambda} = 0$ (Figure 8). For all $Re^t_{\lambda} = 0$, the LES data under-estimates the $k$ predicted by DNS by around 40%. For $Re^t_{\lambda} = 9$ and 105, the LES with no model underestimates the DNS kinetic energy by 20% at $t/\tau_L^t = 1$, whereas our dispersion model over-estimated the kinetic energy by 5% (Figure 8).

For our a posteriori analysis of inertial particle dispersion, we decompose the particle velocity in two parts. One is a component that is spatially correlated (the mean Eulerian particle velocity field), and the other component is uncorrelated and quasi-random in nature, caused by crossing-trajectories. The uncorrelated component is amplified as inertia increases and the heavy particles’ trajectories will cross with the mean flow. The correlated component is dominant for low inertia particles. Moreau, Simonin, and Bédat applied this velocity partitioning analysis to decaying homogeneous isotropic turbulence, with a single-flow realisation and a large number of particles ($N_p = 8 \times 10^7$, $N_{ppc} \approx 300$, and $N_{cell} = 64^3$). For our a posteriori analyses of inertial particle dispersion, we used $N_p = 10^7$ ($N_{ppc} > 200$ when $N_{cell} = 36^3$, where $N_{ppc}$ is the number of particles per cell) to minimise statistical error in calculation of second-order velocity moments.

By following these studies, we calculated the correlated velocity in each computational cell as

$$\tilde{u}_p = \frac{1}{N_{ppc}} \sum_i^{N_{ppc}} \sum_i^N g(x^{(i)} - r)$$

and the uncorrelated velocity of a particle, $i$, was found by

$$\delta u_p^{(i)} = u_p^{(i)} - \tilde{u}_p^{(i)}.$$

With these definitions of velocity partitions, we can calculate the corresponding kinetic energy partitions. The domain-averaged particle total kinetic energy, $k_p$, can be found as

$$k_p = \frac{1}{N_p} \sum_i^{N_p} tr(u_p^{(i)} \otimes u_p^{(i)}).$$
FIG. 7. Decay of tracer particle fluctuating kinetic energy in \textit{a posteriori} tests of homogeneous isotropic turbulence at $Re_\lambda^{t=0} = 33$ for various mesh resolutions, $\Delta$, (a) $3\Delta_{DNS}$, (b) $5\Delta_{DNS}$, (c) $7\Delta_{DNS}$, (d) $9\Delta_{DNS}$. Flow configuration of homogeneous isotropic turbulence is given in Table I.
FIG. 8. Decay of tracer particle fluctuating kinetic energy in a posteriori tests of homogeneous isotropic turbulence at (a) $Re_{\lambda}^t=0 = 9$ and (b) $Re_{\lambda}^t=0 = 105$. LES simulations performed with mesh resolution $\Delta = 7\Delta_{DNS}$. Flow configuration of homogeneous isotropic turbulence is given in Table I.

The domain-averaged correlated and uncorrelated components of particle fluctuating kinetic energy are found as

$$
\tilde{k}_p = \frac{\sum_{i}^{N_{cell}} tr(\tilde{u}_p^{(i)} \otimes \tilde{u}_p^{(i)}) \tilde{n}_p^{(i)}}{\sum_{i}^{N_{cell}} \tilde{n}_p^{(i)}},
$$

(25)

$$
\delta k_p = \frac{1}{N_p} \sum_{i}^{N_p} tr(\delta u_p^{(i)} \otimes \delta u_p^{(i)}),
$$

(26)

where $\tilde{n}_p$ is the local particle number density.

The correlated and uncorrelated velocity components are sensitive to the size of the filter, $r$, used for computing $\tilde{u}_p$ in Equation (22)$^{96,97}$. When calculating the velocity partitioning, we use the adaptive filter proposed by Capecelatro, Desjardins, and Fox$^{96}$ to compute $\tilde{u}_p$, which is given as

$$
r(\alpha_p) = \sqrt[3]{\frac{N_{ppc} d_p^3}{\alpha_p}},
$$

(27)

where $\alpha_p$ is the local solid volume fraction and $N_{ppc}$ is the number of particles locally available for sampling. As an initial guess of filter width is required to calculate $\alpha_p$, we set the initial
As the same size as the cell size of the LES mesh and perform six iterations to converge to a steady value of \( r \). The same initial conditions for calculating \( r \) were used for analysing both the LES and DNS datasets.

In Figure 9, we assess the total, correlated, and uncorrelated components of particle fluctuating kinetic energy from our dispersion model. For all \( St^{t=0} \) number, the model with initial condition generator recovers the total particle fluctuating kinetic energy. We see that the model increases the uncorrelated (random) component, which is consistent as the stochastic term in our model follows a Gaussian distribution with zero mean. For all \( St^{t=0} \) numbers, we can see that \( \tilde{k}_p \) is approximately same for DNS and both LES. As particle inertia increases, the particle motion becomes increasingly uncorrelated with the mean fluid motion as shown by a larger \( \delta k_p \) which agrees with previous studies\(^{12,95}\).

In Figure 10, we assess the particle fluctuating kinetic energy decay at a low and high \( Re^{t=0} \). Similar to as shown for tracer particles (Figure 8), our model has slightly underfit to the low \( Re^{t=0} \) case. For \( Re^{t=0} = 105 \) (not in training data), we achieve good agreement with the DNS predictions, particularly for \( k_p \) and \( \delta k_p \).

Time evolution of particle fluctuating kinetic energies in Figures 9 and 10 show good agreement between DNS and the dispersion model for predicted domain-averaged kinetic energy components. To gain a deeper insight into the velocity distribution, we evaluated the instantaneous PDF of correlated and uncorrelated particle velocity (Figure 11). For \( St^{t=0} = 1 \), the variance predicted by the neural SDE agrees well with the DNS data (within 3% error), compared to the LES with no model, where the variance predicted is 70% lower than the DNS (Figure 11a). In contrast, the excess kurtosis (fourth central moment) for LES with our model was 6.4 compared to the DNS value of 0.6. The excess kurtosis for LES with no model was 6.7, which is a factor of 12 larger than the DNS. For \( St^{t=0} = 5 \) (Figure 11b), the variance predicted by the LES with our model is 6% lower than the DNS data, compared to LES with no model under-predicted the DNS variance by 60%. The excess kurtosis of \( St^{t=0} = 5 \) particles for LES with no model is significantly lower than \( St^{t=0} = 1 \) (3.1 compared to 6.7). The \( \delta u_p \) distributions qualitatively agree with results of Moreau, Simonin, and Bédat\(^{12}\), as the uncorrelated velocity for high inertia particles approaches a Gaussian distribution.

The instantaneous correlated particle velocity PDFs of both LES datasets agree well with the DNS for both \( St^{t=0} \) (Figure 11c,d). Only minor differences were observed, including that
FIG. 9. Time evolution of particle fluctuating kinetic energy components in a posteriori tests of homogeneous isotropic turbulence, with varying $St^{t=0}$ number and filter size $\Delta = 7\Delta_{DNS}$ at $Re_{\lambda}^{t=0} = 33$. Rows are $St^{t=0} = 0.1, St^{t=0} = 1, St^{t=0} = 5$ (top-to-bottom).
FIG. 10. Time evolution of particle fluctuating kinetic energy components in \textit{a posteriori} tests of homogeneous isotropic turbulence, with varying $Re^\lambda_{t=0}$ at $St^t_{t=0} = 1$. Filter size for LES was $\Delta = 7\Delta_{DNS}$. Rows are (top) $Re^\lambda_{t=0} = 9$, (bottom) $Re^\lambda_{t=0} = 105$.

The variance of the PDF predicted by LES with our model under-predicted the DNS data by 9% for $St^t_{t=0} = 1$ and 6% for $St^t_{t=0} = 5$. The PDF variance predicted by LES with no model is within 3% for both $St^t_{t=0}$. The excess kurtosis is less than 0.5 in all cases, showing the distribution is very close to Gaussian. Similar to Figures 9 and 10, these results confirm that LES filtering affects only the uncorrelated component of particle velocity. However, by examining the PDF (Figure 11) our results show that a stochastic model can recover the same variance as DNS simulations, but not higher-order moments such as kurtosis.

We have analysed the instantaneous PDF of particle velocity components for $Re^\lambda_{t=0} = 105$
FIG. 11. Probability density function of instantaneous uncorrelated (top) and correlated (bottom) particle velocity at $t/\tau^L_{t=0} = 1$. Panels show (a,c) $St^{t=0} = 1$ and (b,d) $St^{t=0} = 5$. Velocities are normalised by the standard deviation of the DNS uncorrelated velocity. Simulations are performed at $Re^L_{\lambda} = 33$ with $N_{cell} = 36^3$ ($7\Delta_{DNS}$).
FIG. 12. Probability density function of instantaneous correlated (left) and uncorrelated (right) particle velocity at $Re^\lambda = 105$ with $St^\lambda = 1$. Velocities are normalised by the standard deviation of the DNS velocity component. Simulations were performed with $N_{cell} = 36^3 (7\Delta_{DNS})$. Distribution shown from time instant $t/\tau^L = 1$.

(Figure 12), to understand the behaviour of the particle fluctuating kinetic energy (Figure 10 lower). The variance of the correlated velocity PDF predicted by LES with our model under predicts the DNS variance by 25%, compared to 1% over-prediction by LES with no model (Figure 12a). The uncorrelated velocity variance is under-predicted by 60% by LES with no model, and over-predicted by 60% for LES with our model (Figure 12b). The excess kurtosis predicted by LES with no model was 8.6 times larger than the DNS predictions. The excess kurtosis predicted by our model was 3.4 times larger than the DNS predictions. These results show that although the domain-averaged particle fluctuating kinetic energy components are well-predicted by our model on out-of-distribution data (Figure 10 lower), the moments of the uncorrelated velocity distribution show larger differences compared to the DNS data.
IV. DISCUSSION AND CONCLUSIONS

In this study, we aimed to develop a new approach for modelling turbulent dispersion in large-eddy simulation of dilute turbulent particle-laden flows. Our developed model is based on the classic Langevin-type equation for dispersion\textsuperscript{33,49,53}, where each unclosed term is inferred with a neural network (called a ‘neural SDE’). To create training data for the neural SDE, we performed DNS of decaying homogeneous isotropic turbulence. The DNS fluid velocity field was filtered in post-processing and this filtered data was used to extract subgrid turbulent properties ($k_{sgs}$ and $\varepsilon_{sgs}$), which were used for training and \textit{a priori} analysis. The trained network was then used for \textit{a posteriori} simulation in homogeneous isotropic turbulence to assess kinetic energy decay of tracer and inertial particles, which showed significant improvement over LES modelling with no dispersion model. The proposed neural SDE has been shown to provide improved predictions of particle velocity distributions and fluctuating kinetic energy decay in unbounded flows.

The proposed approach can be easily extended to include varying physics, such as shearing flow in the boundary layer of wall-bounded flows\textsuperscript{55}. Furthermore, LES-SDE models for dispersion in moderate-dense flows may be derived by using two-way coupled simulations of particle transport for training data. Stochastic models with theoretically-derived closures have been developed for two-way and four-way coupled simulations with Reynolds-averaged Navier-Stokes fluid modelling\textsuperscript{39,40}, but no such model exists for LES treatment of the fluid phase.

Limitations of this study include the limited database used for training. We have trained the model on homogeneous isotropic turbulence at two Reynolds numbers, which obtained good agreement with DNS data in \textit{a priori} and \textit{a posteriori} analysis. Our model also showed good agreement with DNS data on a high Reynolds number case which was not included in training, showing that the model can generalise to unseen problems (Figure 10). To extend the model to more complex flow configurations, such as wall-bounded turbulence, one can train using filtered DNS data from particle-laden turbulent channel flow. Additionally, the current model could be adapted to include anisotropic velocity gradients that are essential in capturing near-wall flow physics\textsuperscript{55}. The networks trained in this study can be used as a baseline, and additional networks may be added to each term, $G^{NN}$ and $B^{NN}$, to represent additional physics while satisfying the fundamental criterion that the model produces a cor-
rect decay of inertial and tracer particle fluctuating kinetic energy in homogeneous isotropic turbulence\textsuperscript{54}.

A numerical limitation of this study is the use of cell-value interpolation (Supplementary material). We have found that the trilinear interpolation scheme in OpenFOAM 6.0 does not conserve particle momentum and produces nonphysical results. We are aware that using cell values without interpolation to the particle position is generally less accurate than high-order Lagrange polynomials, cubic spline or Hermite interpolation\textsuperscript{98–101}. However, OpenFOAM is a viable framework for Eulerian-Lagrangian modelling of gas-solid flows because of its open-source aspects with applicability for wide range of applications in complex geometries. Accurate and robust interpolation schemes in OpenFOAM would be a complementary development for improved prediction of particle velocity in turbulent particle-laden flows.

The proposed model has been shown to improve predictions of particle fluctuating kinetic energy and particle velocity distributions at varying $Re_t=0$ and $St_t=0$ numbers in decaying homogeneous isotropic turbulence, compared to using no dispersion model. Applications of the model include modelling cough plume dispersion with LES for understanding disease spread\textsuperscript{6,7} or pollutant dispersion in atmospheric flows. Future efforts should extend the existing neural SDE to account for the influence of velocity gradients and validate this against DNS of turbulent channel flow\textsuperscript{102,103}. This will allow the neural SDE to be used for modelling particle dispersion in wall-bounded flows such as drug transport in lung airways.

SUPPLEMENTARY MATERIAL

See supplementary material for comparison of errors in OpenFOAM interpolation schemes.

ACKNOWLEDGMENTS

JW was funded by a 2019 PhD Scholarship from the Carnegie-Trust for the Universities of Scotland.
AUTHOR DECLARATIONS

Conflict of interest

The authors have no conflicts to disclose.

Author contributions

**Josh Williams:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – Original draft, Writing – Review and editing, Visualisation, Funding acquisition. **Uwe Wolfram:** Conceptualization, Methodology, Resources, Supervision, Writing – Review and editing, Funding Acquisition, Project administration. **Ali Ozel:** Conceptualization, Methodology, Software, Resources, Supervision, Writing – Review and editing, Funding Acquisition, Project administration.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Kaggle at http://doi.org/10.34740/KAGGLE/DSV/3998403, reference number 3998403.

REFERENCES

1. J. Williams, J. Kolehmainen, S. Cunningham, A. Ozel, and U. Wolfram, “Effect of patient inhalation profile and airway structure on drug deposition in image-based models with particle-particle interactions,” International Journal of Pharmaceutics 612, 121321 (2022).
2. A. R. Lambert, P. T. O’shaughnessy, M. H. Tawhai, E. A. Hoffman, and C.-L. Lin, “Regional deposition of particles in an image-based airway model: large-eddy simulation and left-right lung ventilation asymmetry,” Aerosol Science and Technology 45, 11–25 (2011).
3. P. Koullapis, S. C. Kassinos, J. Muela, C. Perez-Segarra, J. Rigola, O. Lehmkuhl, Y. Cui, M. Sommerfeld, J. Elcner, M. Jicha, et al., “Regional aerosol deposition in the human
airways: The siminhale benchmark case and a critical assessment of in silico methods,” European Journal of Pharmaceutical Sciences 113, 77–94 (2018).

4C. Wang, Q. Li, and Z.-H. Wang, “Quantifying the impact of urban trees on passive pollutant dispersion using a coupled large-eddy simulation–lagrangian stochastic model,” Building and Environment 145, 33–49 (2018).

5H. Calmet, K. Inthavong, A. Both, A. Surapaneni, D. Mira, B. Eguitzta, and G. Houzeaux, “Large eddy simulation of cough jet dynamics, droplet transport, and inhalability over a ten minute exposure,” Physics of Fluids 33, 125122 (2021).

6K. Liu, M. Allahyari, J. Salinas, N. Zgheib, and S. Balachandar, “Investigation of theoretical scaling laws using large eddy simulations for airborne spreading of viral contagion from sneezing and coughing,” Physics of Fluids 33, 063318 (2021).

7K. Monroe, Y. Yao, A. Lattanzi, V. Raghav, and J. Capecelatro, “Role of pulsatility on particle dispersion in expiratory flows,” Physics of Fluids 33, 043311 (2021).

8P. Fede and O. Simonin, “Numerical study of the subgrid fluid turbulence effects on the statistics of heavy colliding particles,” Physics of Fluids 18, 045103 (2006).

9J. R. Fessler, J. D. Kulick, and J. K. Eaton, “Preferential concentration of heavy particles in a turbulent channel flow,” Physics of Fluids 6, 3742–3749 (1994).

10Z. Dou, A. D. Bragg, A. L. Hammond, Z. Liang, L. R. Collins, and H. Meng, “Effects of reynolds number and stokes number on particle-pair relative velocity in isotropic turbulence: a systematic experimental study,” Journal of Fluid Mechanics 839, 271–292 (2018).

11L. Brandt and F. Coletti, “Particle-laden turbulence: progress and perspectives,” Annual Review of Fluid Mechanics 54, 159–189 (2022).

12M. Moreau, O. Simonin, and B. Bédat, “Development of gas-particle euler-euler les approach: a priori analysis of particle sub-grid models in homogeneous isotropic turbulence,” Flow, turbulence and combustion 84, 295–324 (2010).

13J. Pozorski and S. V. Apte, “Filtered particle tracking in isotropic turbulence and stochastic modeling of subgrid-scale dispersion,” International Journal of Multiphase Flow 35, 118–128 (2009).

14S. Balachandar and J. K. Eaton, “Turbulent dispersed multiphase flow,” Annual review of fluid mechanics 42, 111–133 (2010).
M. Mehrabadi, J. Horwitz, S. Subramaniam, and A. Mani, “A direct comparison of particle-resolved and point-particle methods in decaying turbulence,” Journal of Fluid Mechanics 850, 336–369 (2018).

A. Vreman and J. Kuerten, “Turbulent channel flow past a moving array of spheres,” Journal of Fluid Mechanics 856, 580–632 (2018).

P. Costa, L. Brandt, and F. Picano, “Near-wall turbulence modulation by small inertial particles,” Journal of Fluid Mechanics 922 (2021).

S. Sundaresan, A. Ozel, and J. Kolehmainen, “Toward constitutive models for momentum, species, and energy transport in gas–particle flows,” Annual review of chemical and biomolecular engineering 9, 61–81 (2018).

T. B. Anderson and R. Jackson, “Fluid mechanical description of fluidized beds. equations of motion,” Industrial & Engineering Chemistry Fundamentals 6, 527–539 (1967).

M. R. Maxey and J. J. Riley, “Equation of motion for a small rigid sphere in a nonuniform flow,” The Physics of Fluids 26, 883–889 (1983).

J. Capecelatro and O. Desjardins, “An euler–lagrange strategy for simulating particle-laden flows,” Journal of Computational Physics 238, 1–31 (2013).

S. B. Pope, Turbulent Flows (Cambridge University Press, 2000).

P. Sagaut, Large eddy simulation for incompressible flows: an introduction (Springer Science & Business Media, 2006).

W. P. Jones and B. E. Launder, “The prediction of laminarization with a two-equation model of turbulence,” International journal of heat and mass transfer 15, 301–314 (1972).

B. E. Launder and D. B. Spalding, “The numerical computation of turbulent flows,” in Numerical prediction of flow, heat transfer, turbulence and combustion (Elsevier, 1983) pp. 96–116.

P. Spalart and S. Allmaras, “A one-equation turbulence model for aerodynamic flows,” in 30th aerospace sciences meeting and exhibit (1992) p. 439.

M. Germano, “Turbulence: the filtering approach,” Journal of Fluid Mechanics 238, 325–336 (1992).

J. Smagorinsky, “General circulation experiments with the primitive equations: I. the basic experiment,” Monthly weather review 91, 99–164 (1963).

C. Marchioli, “Large-eddy simulation of turbulent dispersed flows: a review of modelling approaches,” Acta Mechanica 228, 741–771 (2017).
30. C. Marchioli, M. V. Salvetti, and A. Soldati, “Some issues concerning large-eddy simulation of inertial particle dispersion in turbulent bounded flows,” Physics of Fluids 20, 040603 (2008).

31. J.-P. Minier and E. Peirano, “The pdf approach to turbulent polydispersed two-phase flows,” Physics reports 352, 1–214 (2001).

32. D. C. Haworth and S. B. Pope, “A generalized langevin model for turbulent flows,” The Physics of fluids 29, 387–405 (1986).

33. J.-P. Minier, “On Lagrangian stochastic methods for turbulent polydisperse two-phase reactive flows,” Progress in Energy and Combustion Science 50, 1–62 (2015).

34. S. B. Pope, “Pdf methods for turbulent reactive flows,” Progress in energy and combustion science 11, 119–192 (1985).

35. S. Pope, “On the relationship between stochastic lagrangian models of turbulence and second-moment closures,” Physics of Fluids 6, 973–985 (1994).

36. L. Y. Gicquel, P. Givi, F. Jaberi, and S. Pope, “Velocity filtered density function for large eddy simulation of turbulent flows,” Physics of Fluids 14, 1196–1213 (2002).

37. P. Jenny, S. B. Pope, M. Muradoglu, and D. A. Caughey, “A hybrid algorithm for the joint pdf equation of turbulent reactive flows,” Journal of Computational Physics 166, 218–252 (2001).

38. S. B. Pope, “Simple models of turbulent flows,” Physics of Fluids 23, 011301 (2011).

39. A. Innocenti, R. O. Fox, and S. Chibbaro, “A lagrangian probability-density-function model for turbulent particle-laden channel flow in the dense regime,” Physics of Fluids 33, 053308 (2021).

40. A. Innocenti, R. O. Fox, M. V. Salvetti, and S. Chibbaro, “A lagrangian probability-density-function model for collisional turbulent fluid–particle flows,” Journal of Fluid Mechanics 862, 449–489 (2019).

41. E. Peirano, S. Chibbaro, J. Pozorski, and J.-P. Minier, “Mean-field/pdf numerical approach for polydispersed turbulent two-phase flows,” Progress in energy and combustion science 32, 315–371 (2006).

42. J. Pozorski and J.-P. Minier, “Probability density function modeling of dispersed two-phase turbulent flows,” Physical Review E 59, 855 (1999).

43. T. D. Dreeben and S. B. Pope, “Probability density function/monte carlo simulation of near-wall turbulent flows,” Journal of Fluid Mechanics 357, 141–166 (1998).
44 S. Chibbaro and J.-P. Minier, “Langevin pdf simulation of particle deposition in a turbulent pipe flow,” Journal of aerosol science 39, 555–571 (2008).
45 M. Waclawczyk, J. Pozorski, and J.-P. Minier, “Probability density function computation of turbulent flows with a new near-wall model,” Physics of Fluids 16, 1410–1422 (2004).
46 B. Shotorban and F. Mashayek, “A stochastic model for particle motion in large-eddy simulation,” Journal of Turbulence, N18 (2006).
47 P. Fede, O. Simonin, P. Villedieu, and K. Squires, “Stochastic modeling of the turbulent subgrid fluid velocity along inertial particle trajectories,” in Proceedings of the Summer Program (Center for Turbulence Research, 2006) pp. 247–258.
48 A. Berrouk, D. Laurence, J. Riley, and D. Stock, “Stochastic modelling of inertial particle dispersion by subgrid motion for les of high reynolds number pipe flow,” Journal of Turbulence, N50 (2007).
49 A. Innocenti, C. Marchioli, and S. Chibbaro, “Lagrangian filtered density function for LES-based stochastic modelling of turbulent particle-laden flows,” Physics of Fluids 28, 115106 (2016).
50 M. Knorps and J. Pozorski, “Stochastic modeling for subgrid-scale particle dispersion in large-eddy simulation of inhomogeneous turbulence,” Physics of Fluids 33, 043323 (2021).
51 M. Breuer and F. Hoppe, “Influence of a cost–efficient langevin subgrid-scale model on the dispersed phase of large–eddy simulations of turbulent bubble–laden and particle–laden flows,” International Journal of Multiphase Flow 89, 23–44 (2017).
52 E. R. Van Driest, “On turbulent flow near a wall,” Journal of the aeronautical sciences 23, 1007–1011 (1956).
53 J.-P. Minier, “A methodology to devise consistent probability density function models for particle dynamics in turbulent dispersed two-phase flows,” Physics of Fluids 33, 023312 (2021).
54 J.-P. Minier, S. Chibbaro, and S. B. Pope, “Guidelines for the formulation of Lagrangian stochastic models for particle simulations of single-phase and dispersed two-phase turbulent flows,” Physics of Fluids 26, 113303 (2014).
55 J. Ling, A. Kurzawski, and J. Templeton, “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance,” Journal of Fluid Mechanics 807, 155–166 (2016).
R. Fang, D. Sondak, P. Protopapas, and S. Succi, “Neural network models for the anisotropic reynolds stress tensor in turbulent channel flow,” Journal of Turbulence 21, 525–543 (2020).

S. Beetham and J. Capecelatro, “Formulating turbulence closures using sparse regression with embedded form invariance,” Physical Review Fluids 5, 084611 (2020).

J. Park and H. Choi, “Toward neural-network-based large eddy simulation: Application to turbulent channel flow,” Journal of Fluid Mechanics 914 (2021).

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,” Computers & Fluids 195, 104319 (2019).

P. M. Milani, J. Ling, and J. K. Eaton, “Turbulent scalar flux in inclined jets in crossflow: counter gradient transport and deep learning modelling,” Journal of Fluid Mechanics 906 (2021).

H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, and R. Lguensat, “Physical invariance in neural networks for subgrid-scale scalar flux modeling,” Physical Review Fluids 6, 024607 (2021).

Y. Jiang, J. Kolehmainen, Y. Gu, Y. G. Kevrekidis, A. Ozel, and S. Sundaresan, “Neural-network-based filtered drag model for gas-particle flows,” Powder Technology 346, 403–413 (2019).

Y. Jiang, X. Chen, J. Kolehmainen, I. G. Kevrekidis, A. Ozel, and S. Sundaresan, “Development of data-driven filtered drag model for industrial-scale fluidized beds,” Chemical Engineering Science 230, 116235 (2021).

F. Dietrich, A. Makeev, G. Kevrekidis, N. Evangelou, T. Bertalan, S. Reich, and I. G. Kevrekidis, “Learning effective stochastic differential equations from microscopic simulations: combining stochastic numerics and deep learning,” arXiv preprint arXiv:2106.09004 (2021).

L. Yang, C. Daskalakis, and G. E. Karniadakis, “Generative ensemble regression: Learning particle dynamics from observations of ensembles with physics-informed deep generative models,” SIAM Journal on Scientific Computing 44, B80–B99 (2022).

X. Chen, L. Yang, J. Duan, and G. E. Karniadakis, “Solving inverse stochastic problems from discrete particle observations using the fokker–planck equation and physics-informed neural networks,” SIAM Journal on Scientific Computing 43, B811–B830 (2021).
M. Raissi, “Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations,” arXiv preprint arXiv:1804.07010 (2018).

P. Kidger, “On neural differential equations,” arXiv preprint arXiv:2202.02435 (2022).

J. Williams, U. Wolfram, and A. Ozel, “Filtered direct numerical simulation dataset,” (2022), doi: 10.34740/KAGGLE/DSV/3998403.

H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, “A tensorial approach to computational continuum mechanics using object-oriented techniques,” Computers in physics 12, 620–631 (1998).

L. Schiller and Z. Naumann, “A drag coefficient correlation,” Z. Ver. Deutsch. Ing 77, e323 (1935).

C. Fureby, G. Tabor, H. G. Weller, and A. D. Gosman, “A comparative study of subgrid scale models in homogeneous isotropic turbulence,” Physics of Fluids 9 (1996).

H. S. Kang, S. Chester, and C. Meneveau, “Decaying turbulence in an active-grid-generated flow and comparisons with large-eddy simulation,” Journal of Fluid Mechanics 480, 129–160 (2003).

T. Saad, D. Cline, R. Stoll, and J. C. Sutherland, “Scalable tools for generating synthetic isotropic turbulence with arbitrary spectra,” AIAA journal 55, 327–331 (2017).

P.-K. Yeung and S. B. Pope, “Lagrangian statistics from direct numerical simulations of isotropic turbulence,” Journal of Fluid Mechanics 207, 531–586 (1989).

N. Mansour and A. Wray, “Decay of isotropic turbulence at low reynolds number,” Physics of Fluids 6, 808–814 (1994).

A. Ozel, J. Kolehmainen, S. Radl, and S. Sundaresan, “Fluid and particle coarsening of drag force for discrete-parcel approach,” Chemical engineering science 155, 258–267 (2016).

M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, “A dynamic subgrid-scale eddy viscosity model,” Physics of Fluids A: Fluid Dynamics 3, 1760–1765 (1991).

E. de Villiers, The Potential of Large Eddy Simulation for the Modelling of Wall Bounded Flows, Ph.D. thesis, Imperial College London (2006).

F. Nicoud and F. Ducros, “Subgrid-scale stress modelling based on the square of the velocity gradient tensor,” Flow, turbulence and Combustion 62, 183–200 (1999).

M. J. Cernick, S. Tullis, and M. Lightstone, “Particle subgrid scale modelling in large-eddy simulations of particle-laden turbulence,” Journal of Turbulence 16, 101–135 (2015).
82 M. Afkhami, A. Hassanpour, M. Fairweather, and D. O. Njobuenwu, “Fully coupled LES-DEM of particle interaction and agglomeration in a turbulent channel flow,” Computers & Chemical Engineering 78, 24–38 (2015).

83 Q. Wang and K. D. Squires, “Large eddy simulation of particle-laden turbulent channel flow,” Physics of Fluids 8, 1207–1223 (1996).

84 N. Wiener, “The homogeneous chaos,” American Journal of Mathematics 60, 897–936 (1938).

85 K. Itô, “Multiple wiener integral,” Journal of the Mathematical Society of Japan 3, 157–169 (1951).

86 A. N. Kolmogorov, “Dissipation of energy in the locally isotropic turbulence,” in Dokl. Akad. Nauk SSSR A, Vol. 32 (1941) pp. 16–18.

87 J.-P. Minier, E. Peirano, and S. Chibbaro, “Weak first-and second-order numerical schemes for stochastic differential equations appearing in lagrangian two-phase flow modeling,” Monte Carlo Methods and Applications 9, 93–133 (2003).

88 S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6, 107–116 (1998).

89 S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies,” in A Field Guide to Dynamical Recurrent Neural Networks (IEEE press, 2001).

90 K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision (2015) pp. 1026–1034.

91 F. Chollet et al., “Keras,” (2015).

92 M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” (2015), software available from tensorflow.org.
I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *International conference on machine learning* (PMLR, 2013) pp. 1139–1147.

M. Boivin, O. Simonin, and K. D. Squires, “On the prediction of gas–solid flows with two-way coupling using large eddy simulation,” *Physics of Fluids* **12**, 2080–2090 (2000).

P. Février, O. Simonin, and K. D. Squires, “Partitioning of particle velocities in gas–solid turbulent flows into a continuous field and a spatially uncorrelated random distribution: theoretical formalism and numerical study,” *Journal of Fluid Mechanics* **533**, 1–46 (2005).

J. Capecelatro, O. Desjardins, and R. O. Fox, “Numerical study of collisional particle dynamics in cluster-induced turbulence,” *Journal of Fluid Mechanics* **747** (2014).

J. Capecelatro, O. Desjardins, and R. O. Fox, “On fluid–particle dynamics in fully developed cluster-induced turbulence,” *Journal of Fluid Mechanics* **780**, 578–635 (2015).

P. Yeung and S. Pope, “An algorithm for tracking fluid particles in numerical simulations of homogeneous turbulence,” *Journal of computational physics* **79**, 373–416 (1988).

S. Balachandar and M. Maxey, “Methods for evaluating fluid velocities in spectral simulations of turbulence,” *Journal of Computational Physics* **83**, 96–125 (1989).

A. L. Rovelstad, R. A. Handler, and P. S. Bernard, “The effect of interpolation errors on the lagrangian analysis of simulated turbulent channel flow,” *Journal of Computational Physics* **110**, 190–195 (1994).

R. Garg, C. Narayanan, D. Lakehal, and S. Subramaniam, “Accurate numerical estimation of interphase momentum transfer in lagrangian–eulerian simulations of dispersed two-phase flows,” *International Journal of Multiphase Flow* **33**, 1337–1364 (2007).

C. Marchioli, A. Soldati, J. Kuerten, B. Arcen, A. Taniere, G. Goldensoph, K. Squires, M. Cargnelutti, and L. Portela, “Statistics of particle dispersion in direct numerical simulations of wall-bounded turbulence: Results of an international collaborative benchmark test,” *International Journal of Multiphase Flow* **34**, 879–893 (2008).

A. Soldati and C. Marchioli, “Physics and modelling of turbulent particle deposition and entrainment: Review of a systematic study,” *International Journal of Multiphase Flow* **35**, 827–839 (2009).
1 Interpolation error analysis

To determine the optimal scheme for interpolation of Eulerian properties to the particle position, we performed simulations of one particle flowing through a domain where the fluid velocity can be described analytically. proposed a case where the fluid velocity followed:

\[
\begin{align*}
  u_{fx} &= u_0, \\
  u_{fy} &= 0, \\
  u_{fz} &= u_0 \sin(B\pi x_x).
\end{align*}
\]

where \(u_{fx}\) is the x component of fluid velocity, \(x_x\) is the x coordinate of particle position, the parameters describing the flow are \(u_0 = 0.025\) m/s and \(B = 4\). The particle position at time, \(t\), can be obtained analytically by

\[
\begin{align*}
  x_x(t) &= x_x^0 + u_0 t, \\
  x_y(t) &= 0, \\
  x_z(t) &= x_z^0 + \frac{1}{B\pi} [\cos(B\pi x_x^0) - \cos(B\pi x_x^0 + \pi u_0 t)].
\end{align*}
\]

We evaluated the absolute relative error of particle position in the z-direction \(x_z\) obtained from interpolating the fluid velocity and integrating the equation, compared to the analytical solution. We tested interpolating the fluid velocity using trilinear interpolation (‘cellPoint’) and using the cell value (‘cell’). The computational domain was a cuboid extending from \(x = (-2, -1, -1)\) m to \(x = (2, 1, 1)\) m. The domain was discretised into \(16^3\), \(32^3\), \(64^3\), and \(128^3\) cells.

By examining particle trajectories, we observed that cellPoint interpolation scheme produces zero motion in the transverse direction on a coarse mesh and gave a maximum relative error of 165% (Figure 1b). In contrast, when using the cell value for calculating particle motion (Figure 1b), the particle shows qualitatively correct behaviour, with some over-prediction due to the coarse mesh (maximum error 43%). When \(N_{cell} = 64^3\), both interpolation schemes show an improved agreement with the analytical solution (Figure 1c,d). However, the maximum error was 25% for cellPoint and 6.4% for cell interpolation. The poor performance of cellPoint interpolation is surprising, as previous studies have shown that using linear interpolation of fluid properties produces lower error than using the cell value (??). However, some studies have discussed the limitations of trilinear interpolation compared to more advanced methods such as high-order Lagrange polynomials, cubic spline or Hermite interpolation (??). Due to the lower error of cell interpolation and an apparent implementation issue in cellPoint interpolation, we chose to use cell interpolation in this study.
Figure 1: Evolution particle position in $x$ and $z$ directions for varying interpolation scheme and mesh resolution. Columns show (a,c) cell interpolation, and (b,d) cellPoint interpolation. Rows show (a,b) $N_{cell} = 16^3$ and (c,d) $N_{cell} = 64^3$. 