Supplementary Information for

Lagrangian large eddy simulations via physics-informed machine learning

Yifeng Tian, Michael Woodward, Mikhail Stepanov, Chris Fryer, Criston Hyett, Daniel Livescu, Michael Chertkov

To whom correspondence should be addressed. E-mail: yifengtian@lanl.gov, chertkov@arizona.edu, or livescu@lanl.gov

This PDF file includes:

- Supplementary text
- Fig. S1 (not allowed for Brief Reports)
- SI References
Supporting Information Text

1. Lagrangian Large Eddy Simulation

A. Physics-Informed Choice of the Pair-Wise Interaction. In the main text, we introduced general equations for describing the evolution of the system of particles as:

$$\frac{d}{dt} \left[ \mathbf{x}_i \right] = \sum_{j=1, j \neq i}^{N} f(x_i, x_j, \phi_i, \phi_j)$$

where $f$ is function expressing the pair-wise force imposed on a particle $i$ by a particle $j$ and dependent on the positions of the two functions as well as the vector $\phi$ evaluated at the positions of the two particles.

Here we discuss the mathematical construction of $f$ and feature selections to the invariance of (1) under the Galilean, translational, and rotational transformations of the underlying frame of reference. Let us denote the operator, representing an element of the union of the physical transformations, as $T$. Then, we require that the pair-wise force is $T$-invariant, i.e. formally,

$$T f(x_i, x_j, \phi_i, \phi_j) = f(T x_i, T x_j, T \phi_i, T \phi_j).$$

If $T$ is limited to the translational invariance, then transformation of $x_i$ can be expressed as $T x_i = x_i + x'$, whereas $\phi_i$ stays invariant, i.e. $T \phi_i = \phi_i$. Therefore, a straightforward approach to enforce translational invariance of $f$ is to replace the general dependence on $x_i$ and $x_j$ on dependence on only the difference of the two, $x_i - x_j$. Similarly, the Galilean invariance, which is associated with the transformation to a different inertial frame, $T v_i = v_i + v'$, $T x_i = x_i + v' t$, can be enforced by replacing the general dependence of the two-particle velocities, $v_i$ and $v_j$, by the relative velocity, $v_i - v_j$. To enforce the rotational invariance, symmetry transformations of the scalar (density) and the vector (velocity) components of $\phi$, $v_i$, should be dealt with differently. Let us decompose $f$ into its scalar, $f_s$, and vector, $f_v$, components. Then rotational invariance of the scalar component of the force requires that, $T f_s = f_s$. The respective constraint which needs to be imposed on the transformed vector component of $f_v$, $T f_v$, is that it becomes a linear combination of the rotation-aware vector bases and scalar functions of rotational invariant scalars (we follow here the so-called Tensor Basis (TB) approach of (1, 2)).

B. Lagrangian LES model. In the main text, we proposed the following L-LES model:

$$\frac{d}{dt} \left[ \phi_i \right] = \sum_{j=1, j \neq i}^{N} f_{NN}(x_i, x_j, \phi_i, \phi_j)$$

$$= \sum_{j=1, j \neq i}^{N} \sum_{k=1}^{2} N^N_{\nu}(I_{ij,m}; \lambda_{\nu}) b_{ij,k} + \Pi_{ij} b_{ij,1} + \left[ 0 \right]_{F_i}$$

where $N$ is the number of particles placed inside of the computational domain, $\Omega$. In all the numerical experiments, reported in this manuscript, $\Omega$ is a three-dimensional cube of the volume, $(2\pi)^3$, i.e. $\Omega = [0, 2\pi]^3$. Here in Eq. (3) $I$ and $b$ denote, respectively, the scalar invariants and vector bases of the bases expansion approach of (1, 2) and are defined using pair-particle information:

$$x_{ij} = (x_i - x_j)/d, \quad v_{ij} = (v_i - v_j)/v_{rms}, \quad \rho_{ij} = \frac{(\rho_i + \rho_j)}{2 v_{rms}},$$

$$I_{ij,1} = \frac{\rho_{ij}}{v_{rms}}, \quad I_{ij,2} = \frac{\rho_{ij}}{v_{rms}}, \quad I_{ij,3} = \| x_{ij} \|, \quad I_{ij,4} = | v_{ij} |,$$

$$I_{ij,5} = x_{ij} \cdot v_{ij}, \quad b_{ij,1} = x_{ij}, \quad b_{ij,2} = v_{ij}.$$

Let us discuss other details of the parameters and terms contributing to Eqs. (3):

- Neural Networks: We introduce two distinct NNs, $N^N_{\nu}$ and $N^N_{\rho}$, to approximate respective components of the function $f$. As customary, the NNs will over-parameterize to fit the data. We will utilize in the following, specifically in Section 3 where numerical validation of the L-LES approach is discussed, the same Neural Network for both $N^N_{\rho}$ and $N^N_{\nu}$ – containing 5 scalar features as input and 4 hidden layers, each with 100 neurons. The output of $N^N_{\rho}$ is a scalar that accounts for the RHS of the continuity equation. The outputs of $N^N_{\nu}$ are the scalar coefficients of the two vector bases.

- Eddy Viscosity: We make sure that the eddy diffusivity, $\Pi$-dependent, term in Eq. (3) satisfies physical symmetries (see discussion above) and otherwise we use the following form introduced in (3), which has been used as the standard artificial viscosity in SPH literature. The artificial viscosity term has been widely adopted in SPH formulation to prevent particle collision, ensure numerical stability, and model shock waves in supersonic flows.

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \mu_{ij} + \beta \mu_{ij}^2}{\rho_{ij}}, & \text{if } \mathbf{v}_{ij} \cdot \mathbf{x}_{ij} < 0, \\ 0, & \text{if } \mathbf{v}_{ij} \cdot \mathbf{x}_{ij} \geq 0 \end{cases}$$

$$\mu_{ij} = \frac{\partial v_{ij} \cdot x_{ij}}{|x_{ij}|^2 + c^2}$$
The eddy viscosity term associated with a pair of particles activates (becomes significant) when the two particles are separated from each other at a distance that is smaller than the resolved scale, $d$. We naturally link the resolved scale, $d$, to the number of particles, $N$, thus setting $d = 2\pi/N^{1/3}$ in our L-LES model over the $[0, 2\pi]^3$ cube. Therefore the eddy-viscosity term leads, effectively, to repulsion elongated with $b_{ij,1}$, and it can thus be interpreted (according to the name) as modeling effective energy transfer from the resolved scales to smaller scales. The eddy viscosity term also prevents particles from clustering and collisions, therefore, providing overall numerical stability of the scheme, consistent with the common reasoning well documented in the SPH literature (see e.g. (4, 5)). The coefficients, $\{\lambda_\rho, \lambda_v, \alpha, \beta\}$, contributing to the eddy viscosity term are not fixed (as custom in the classic particle-based method) but are learnable, i.e. subject to optimization together with the parameters of the aforementioned NNs. A more recent study (6) proposed a method to dynamically update parameters in the artificial viscosity term to get rid of the diffusive parameter tuning. However, in this work, we set them to be learnable parameters using the traditional formulation of artificial viscosity.

The eddy viscosity term could be absorbed into the above-mentioned NN term since NN can be used as a universal function approximator, but here we separate the parameterization of the eddy viscosity term to inject physical intuition and mathematical structures. This may also alleviate the burdens on learning NN parameters so that the learning process becomes easier and smoother.

- **External Force:** The last term $F_i$ in (3) models the external force, which is large-scale, i.e. it injects energy into the system largely at the scales comparable to the size of the box, $L$. We choose the forcing term in the L-LES model (3) to be linear in $v_i$

$$F_i = \alpha_F \sum_{j=1}^{N} \chi_F(x_i - x_j) v_i,$$

[6]

consistently with the structure of the forcing term used in our ground truth DNS data to ensure that the DNS (and thus our L-LES model too) reaches a statistically-steady state. Here in Eq. (6), $\alpha_F$ is a parameter, which is set to match the energy injection rate from external forcing with the dissipation rate of the turbulent flow; and $\chi_F(x_i - x_j)$ is the forcing weight function, dependent on the inter-particle radius-vector in the form prescribed by the DNS. In all of our experiments the large-scale forcing term is fixed, i.e. it is the only term on the RHS of Eq. (3) which is not subject to fitting/learning.

2. From Lagrangian Trajectories to Eulerian Fields with Smoothing Kernel

In the main text, we introduced the smoothing kernel $W_\theta(x)$ for mapping from Lagrangian description of a scalar field $A(x)$ to the Eulerian description:

$$\langle A(x) \rangle = \int_\Omega A(x') W_\theta(x - x') dx'.$$

[7]

Other than the compact support condition, the smoothing kernel also needs to satisfy the normalization condition,

$$\int_\Omega W_\theta(x - x') dx' = 1,$$

[8]

to guarantee the zerth order consistency of the integral representation of continuum functions (7).

We reconstruct the field $\langle A \rangle$ and its gradient $\langle \nabla A(x) \rangle$ from the values of $A$ evaluated at the particle locations $x_i$ using the following equations:

$$\langle A(x) \rangle = \sum_{j=1}^{N} \Delta V_j A(x_j) W_\theta(x - x_j),$$

[9a]

$$\langle \nabla A(x) \rangle = \sum_{j=1}^{N} \Delta V_j \left[ A(x_j) - A(x) \right] \nabla_r W_\theta(x - x_j),$$

[9b]

where $\Delta V_j$ stands for a volume element associated with the particle $j$, and $\nabla_r W_\theta$ denotes the gradient of $W_\theta$ to $r$ ($r = |x - x_j|$).

A schematic illustration of the reconstruction of the field ($A(x)$) is shown in Fig. S1 (a). We can also reconstruct the field of gradient of $A$, $\nabla \langle A(x) \rangle$ (Fig. S1 (b)) using Eq. (9b) (3).

A. Smoothing Kernel Loss Function. We build the SK loss function, consisting of three terms,

$$L_{SK} = c_v L_v + c_g L_g + c_s L_s.$$

[10]
where velocities, \( L \) of Yifeng Tian, Michael Woodward, Mikhail Stepanov, Chris Fryer, Criston Hyett, Daniel Livescu, Michael Chertkov

Fig. S1. Schematic illustration of (a) the reconstruction of the image, \( \langle A \rangle \), of the field, \( A \), defined as a convolution of the kernel, \( W_\theta(\cdot) \), with the field evaluated at the positions of the Lagrangian particles and (b) its gradient \( \nabla \langle A \rangle \). Width of the kernel, \( h \), shown in the Figure, should be viewed as one of the components of the (tunable) vector of the kernel parameters, \( \theta \).

The first two terms in the SK-LF represent a mismatch between Lagrangian (particle) data and Eulerian (field) data for the velocities, \( L_v \), and the velocity gradients, \( L_g \), respectively:

\[
L_v = \frac{1}{N_E} \sum_{a=1}^{N_E} |\langle v(x_a) \rangle - v(x_a) |^2,
\]

\[
\langle v(x_a) \rangle = \sum_{i=1}^{N_L} v_i W_\theta(x_a - x_i) \Delta V_i,
\]

\[
L_g = \frac{1}{N_E} \sum_{a=1}^{N_E} \sum_{\alpha=1}^{3} |\langle \nabla v^\alpha(x_a) \rangle - \nabla v^\alpha(x_a) |^2,
\]

\[
\langle \nabla v^\alpha(x_a) \rangle = \sum_{i=1}^{N_L} \Delta V_i (v_i^\alpha - v^\alpha(x_a)) \nabla r_{ai} W_\theta(r_{ai}) r_{ai},
\]

where \( \Delta V_i = m_i/(\sum_{j=1}^{N} m_j W_\theta(x_j - x_i)) \) is the volume element associated with the (Lagrangian) particle \( i \) (8); superscript \( \alpha \in 1, 2, 3 \) denotes the component of the velocity vector; and values of \( v \), evaluated at the Lagrangian positions and locations of the Eulerian grid, are assumed taken from the ground truth (GT) data extracted from properly filtered Lagrangian DNS.

The role of the third contribution to the SK-LF, \( L_n \), is to enforce the SK normalization condition (8), and thus learn the normalization coefficient \( \alpha \). Since the SK is modeled via a NN, whose integral over \( \Omega \) cannot be computed analytically, we consider spherically symmetric SK, split the compact spherical domain of the integration described by Eq. 7 into \( N_r \) shells, of radii \( r_k \), \( k = 0, 1, \ldots, N_r \), and then approximate the remaining one-dimensional integral according to the trapezoidal rule, thus arriving at the following expression for the normalization enforcing component of \( L_{SK} \):

\[
L_n = (I - 1)^2, \quad I = \int_\Omega W_\theta(r) dr \approx 4\pi \Delta r \sum_{k=1}^{N_r-1} W_\theta(r_k) r_k^2
\]

3. Loss Function Enforcing Multi-Physics in L-LES

A. Field-based loss function. For the field-based loss function, the loss function that compares the predicted field and ground-truth is computed using:

\[
L_f = \frac{1}{N_E M} \sum_{a=1}^{N_E} \sum_{m=1}^{M} \left| \langle \phi^{(m+1)}(x_a) \rangle - \phi^{(m+1)}(x_a) \right|^2,
\]

\[
\langle \phi^{(m+1)}(x_a) \rangle = \sum_{i=1}^{N_L} \tilde{\phi}_i^{(m+1)} W_\theta(x_a - x_i^{(m+1)}) \Delta V_i,
\]

\[
\tilde{\phi}_i^{(m+1)} = \phi_i^{(m)} + \Delta \cdot \sum_{j=1, j \neq i}^{N} f_{NN}(x_i^{(m)}, x_j^{(m)}, \phi_i^{(m)}, \phi_j^{(m)}),
\]
where $\phi_i^{(m)}, x_i^{(m)} \forall i = 1, \cdots, N, \forall m = 1, \cdots, M$ is available as the Lagrangian part of the GT data and $\hat{\phi}_i^{(m+1)}$ is the L-LES prediction given $(\phi_i^{(m)}, x_i^{(m)})$. Note that $\phi^{(m+1)}(x_a)$ denote the Eulerian field $\phi$ measured at the grid point $x_a$ at time $t_{m+1}$.

**B. Statistics-based loss function.** To build a histogram of a single-particle instantaneous velocity, $(v_i^{(m)} )^\alpha$, we map the GT data, aggregated over all the particles, $i = 1, \cdots, N$, all the directions, $\alpha = 1, 2, 3$, and all the available moments of time, $t_m$, $\forall m = 1, \cdots, M$, to a histogram $P_{v}(\mu_k)$ build of $K$ bins, $k = 1, \cdots, K$, $[\mu_k, \mu_{k+1}] = \mu_k + w$, each of size, $w$:

$$\forall k: P_v(\mu_k) = \frac{1}{3NMK} \sum_{i=1}^N \sum_{\alpha=1,2,3} \sum_{m=1}^M \mathbb{1} (v_i^{(m)})^\alpha \in [\mu_k, \mu_{k+1})$$.

[17]

Histogram of the particle acceleration, $P_{dv/dt}$ is built similarly. Notice that the samples-to-histogram functions, $P_v(\mu_k)$ and $P_{dv/dt}(\mu_k)$, allow efficient representation via a Convolutional Neural Network (CNN), with some fixed weights and biases (9), which is designed specifically to enable efficient backpropagation over parameters (entering Eq. (17) via $v_i$). To compare statistics predicted by the model and statistics of the corresponding GT data we construct the Kullback-Leibler (KL) loss function for velocity (and similarly for acceleration):

$$L_{KL,v} = \sum_{k=1}^K P_{GT}(\mu_k) \log \frac{P_{GT,v}(\mu_k)}{P_{pred,v}(\mu_k)}$$.

[18]

**References**

1. J Ling, A Kurzawski, J Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* 807, 155–166 (2016).

2. Y Tian, D Livescu, M Chertkov, Physics-informed machine learning of the lagrangian dynamics of velocity gradient tensor. *Phys. Rev. Fluids* 6, 094607 (2021).

3. JJ Monaghan, JC Lattanzio, A refined particle method for astrophysical problems. *Astron. Astrophys.* 149, 135–143 (1985).

4. T Ye, D Pan, C Huang, M Liu, Smoothed particle hydrodynamics (SPH) for complex fluid flows: Recent developments in methodology and applications. *Phys. Fluids* 31, 011301 (2019).

5. TS Tricco, DJ Price, C Federrath, A comparison between grid and particle methods on the small-scale dynamo in magnetized supersonic turbulence. *MNRAS* 461, 1260–1275 (2016).

6. DD Meringolo, S Marrone, A Colagrossi, Y Liu, A dynamic $\delta$-sph model: How to get rid of diffusive parameter tuning. *Comput. & Fluids* 179, 334–355 (2019).

7. M Liu, G Liu, Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Arch. Comput. Methods Eng.* 17, 25–76 (2010).

8. JJ Monaghan, Smoothed particle hydrodynamics and its diverse applications. *Annu. Rev. Fluid Mech.* 44, 323–346 (2012).

9. Z Wang, H Li, W Ouyang, X Wang, Learnable histogram: Statistical context features for deep neural networks in *European Conference on Computer Vision*. (Springer), pp. 246–262 (2016).