PNAS RESEARCH ARTICLE | APPLIED PHYSICAL SCIENCES OPEN ACCESS

Lagrangian large eddy simulations via physics-informed machine learning

Yifeng Tian[a,1], Michael Woodward[b,c], Mikhail Stepanov[b], Chris Fryer[c], Criston Hyett[b,c], Daniel Livescu[c,1], and Michael Chertkov[b,1]

Edited by Daniel Price, Monash University, Melbourne, Australia; received August 11, 2022; accepted May 25, 2023 by Editorial Board Member William H. Press

High-Reynolds number homogeneous isotropic turbulence (HIT) is fully described within the Navier–Stokes (NS) equations, which are notoriously difficult to solve numerically. Engineers, interested primarily in describing turbulence at a reduced range of resolved scales, have designed heuristics, known as large eddy simulation (LES). LES is described in terms of the temporally evolving Eulerian velocity field defined over a spatial grid with the mean-spacing correspondent to the resolved scale. This classic Eulerian LES depends on assumptions about effects of subgrid scales on the resolved scales. Here, we take an alternative approach and design LES heuristics stated in terms of Lagrangian particles moving with the flow. Our Lagrangian LES, thus L-LES, is described by equations generalizing the weakly compressible smoothed particle hydrodynamics formulation with extended parametric and functional freedom, which is then resolved via Machine Learning training on Lagrangian data from direct numerical simulations of the NS equations. The L-LES model includes physics-informed parameterization and functional form, by combining physics-based parameters and physics-inspired Neural Networks to describe the evolution of turbulence within the resolved range of scales. The subgrid-scale contributions are modeled separately with physical constraints to account for the effects from unresolved scales. We build the resulting model under the differentiable programming framework to facilitate efficient training. We experiment with loss functions of different types, including physics-informed ones accounting for statistics of Lagrangian particles. We show that our L-LES model is capable of reproducing Eulerian and unique Lagrangian turbulence structures and statistics over a range of turbulent Mach numbers.

physics-informed machine learning | large eddy simulation | Lagrangian particles | turbulence modeling

Significance

Accurately simulating high-Reynolds number turbulent flows is computationally challenging. Traditional turbulence modeling methodologies, e.g., large eddy simulation (LES), aim to resolve large-scale turbulence in the Eulerian setting, while modeling the subgrid contributions using the resolved-scale flow; the approach usually involves various physics assumptions and parameter tuning. In this work, we develop LES turbulence models, stated in terms of Lagrangian particles moving with the flow, using the physics-informed machine learning framework. We propose generalized equations to model the evolution of a Lagrangian particle cloud with physics-informed parameterization and functional freedom based on a representation using Neural Networks. We show that the proposed Lagrangian–LES model is capable of reproducing turbulence structure and statistics over a range of turbulent Mach numbers.

Accurate direct numerical simulations (DNS) of turbulent flows in physical sciences and engineering applications are, in general, prohibitively expensive due to the existence of a wide range of lengths and time scales. This challenge has motivated the development of the reduced order models (ROM) which achieve fast and efficient solutions in practical applications. In the field of turbulence simulations, Reynolds-averaged Navier–Stokes (RANS) and large eddy simulations (LES) have been widely adopted as alternatives for the DNS of practical turbulent flows (1). In these traditional methods, turbulent flows are viewed from the Eulerian frame where the computational domain is discretized into small elements or units via meshes. In LES, a spatiotemporal low-pass filter is applied to the Navier–Stokes (NS) equations, therefore removing small-scale fluctuations from the consideration. LES results in the reduction of the computational complexity, simply because the range of scales that need to be resolved (1) is reduced.

Eulerian LES has achieved great success in scientific and engineering applications. However, when solving turbulent problems with complex geometries, moving interfaces, and multiphase materials, the Eulerian description suffers from deteriorating numerical accuracy due to deforming mesh, fluid-particle coupling, etc. To resolve these issues, resulting from Eulerian description of the flow, mesh-free methods have been proposed (2, 3), such as smoothed particle hydrodynamics (SPH), the vortex method, and the mesh-free Galerkin method. Simulating turbulent flows using the mesh-free, Lagrangian approximation of NS equations is appealing because it provides a unique perspective for understanding the transport processes, such as the mixing and dispersion of passive scalars. Such processes are dominated by the advective motion of velocity fluctuations in time and space, which is naturally a part of the Lagrangian representation of the fluid via trajectories of representative fluid particles (4). Among the mesh-free methods for solving fluids problems, SPH (5–7) has been adopted in a wide range of scientific applications, e.g., astrophysics, computer graphics, free-surface flows, fluid–structure interaction, biofluids,

This article is a PNAS Direct Submission. D.P. is a guest editor invited by the Editorial Board.

[1]To whom correspondence may be addressed. Email: yifengtian@lanl.gov, livescu@lanl.gov, or chertkov@arizona.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2213638120/-/DCSupplemental.

Published August 16, 2023.

geological flows, magnetohydrodynamics (8–20). Recently, there has been a number of studies adopting SPH to model turbulence (21–29). SPH framework was also integrated into RANS modeling and, specifically, into RANS two-equation models, such as the $k-\epsilon$ type (30–32). A series of studies explored similarity of the filtering procedure of LES and the smoothing procedure of SPH to develop LES models in the SPH framework (33–35). These papers have helped to establish SPH as a promising framework for modeling turbulence which accounts for advection explicitly, through integration over Lagrangian trajectories. However, the papers have also highlighted significant challenges of the SPH modeling related to resolving the stress tensor term in the Lagrangian frame and also the multiparticle nature of the formulation, making derivation and tuning parameters within the SPH models time-consuming and difficult (34).

This manuscript addresses the challenge of adopting the Lagrangian approach, and specifically generalizing, i.e., adApting—NOT adOpting, the SPH approach, to reduced-order modeling of NS turbulence directly. Here, we mention several works that attempt to adapt the SPH approach based on different numerical procedures, such as mesh-less finite volume (36) and adaptive smoothing kernel (37), and differentiate them from the more general formulation proposed in the work, with physics-informed and interpretable parameterization that can be learned from data. We achieve this goal by utilizing the power of machine learning (ML), and more generally artificial intelligence (AI).

Modern ML and, specifically, deep learning, leveraging efficient computational tools such as automatic differentiation and sensitivity analysis of forward and backward propagation, have resulted in multiple success stories in classic AI disciplines, such as computer vision (38), speech recognition (39), and natural language processing (40). However, the biggest achievements of AI and ML in sciences so far have been limited, until recently, to approaches that are data-driven but agnostic to traditional scientific modeling of the underlying physics. Integrating the breakthrough of the modern AI and ML with physical modeling, and specifically for this manuscript with Lagrangian modeling of turbulence, is the major challenge of what we call today physics-informed machine learning (PIML).* In the field of turbulence modeling, there has been a surge of PIML activity in recent years. In particular, major efforts have been devoted to the development of closure models for RANS and LES in the Eulerian frame using innovative neural network architectures. Some of the contributions, prioritized here due to their significance for the methodology of this manuscript, are: tensor basis neural network (TBNN) embedding physical constraints, such as Galilean invariance and rotational invariance, into the closure model (46), and PIML models infusing physical constraints into the neural networks (47–49). A comprehensive overview of these and other related contributions to the field of turbulence closure modeling in the Eulerian frame can be found in ref. 50. Other than developing closure models for RANS and LES, researchers have been experimenting with novel ML approaches to learn turbulence dynamics. In this regard, and just to name a few, we mention refs. 51 and 52, where a convolutional long short-term memory (ConvLSTM) neural network was developed to learn spatial-temporal turbulence dynamics; studies of superresolution allowing reconstruction of turbulence field using underresolved data (53); and neural ordinary differential equation (Neural ODE) for turbulence forecasting (54). PIML models for the Lagrangian description of turbulence have also been actively developed. Authors of ref. 55 have used a regression forest to approximate the behavior of particles in Lagrangian fluid simulations. A differentiable fluid solver accounting for Lagrangian dynamics has been developed in ref. 56. A continuous convolution network was constructed to learn SPH-inspired Lagrangian simulation in ref. 57. Three of us (Y.T., D.L., and M.C.) have reported in ref. 58 development of a PIML approach to design a closure model for the Lagrangian (single particle) dynamics of the velocity gradient tensor (VGT).

We have also developed in ref. 59 a hierarchy of reduced Lagrangian multiparticle models of the PIML/SPH type trained on ground truth (GT) data of two types, as described below. First, we train the SPH model on synthetic multiparticle simulations imitating multiscale Lagrangian turbulence. Then, we also extend it to the Lagrangian GT data derived from the fully resolved Eulerian DNS of HIT.

In this manuscript, we further develop the ideas from the Lagrangian modeling of the PIML/SPH type we have started to explore in ref. 59. In ref. 59, we took advantage of the degrees of freedom in the definition of SPH and then ad**O**pted SPH to the GT data, i.e., adjusted the degrees of freedom within SPH to fit the data. On the contrary, in this manuscript, we generalize SPH, i.e., ad**A**pt SPH. This generalization allows us to introduce a general Lagrangian LES (L-LES) model, simulating turbulence at the resolved scales, and train this model on the Lagrangian data derived from the fully resolved Eulerian DNS.

We describe the L-LES framework in the remainder of the manuscript in steps. The stage is set in Section 1 where we introduce fictitious fluid particles and describe their Lagrangian dynamics, i.e., effective advection by a collective velocity field, and effective interaction imposed by the collective contribution of the particles to the interparticle forces, reconstructed from the velocity field. Parametric, as well as functional, degrees of freedom in the resulting nonlinear system of multiparticle ODEs, stated in terms of coordinates and velocities of the particles, are discussed both in intuitive physical but also formal mathematical terms. We argue about why and in which sense the system of equations governing dynamics of interacting Lagrangian particles is a good LES heuristics. In Section 2, the connection between the Lagrangian and Eulerian descriptions of turbulence is established through smoothing kernel (SK). We introduce parameterization of the SK via a neural network (NN) and then set an SK loss function (LF), based on the turbulence statistics (connecting both Lagrangian and Eulerian descriptions) to train the SK–NN.

We provide a comprehensive description of how we "learn" the PIML/L-LES model in Section 3. We discuss the choice of the L-LES LF, respective L-LES NN architecture, and relevant technical details. Section 4 is devoted to details of the training and validation experiments. We explain the GT DNS model, the process of data generation, and the significance of validating/testing the underlying physics in detail. We summarize results and discuss the path forward, respectively, in Sections 5 and 6.

1. Lagrangian LES

In this section, we describe the main principles of the proposed L-LES methodology for modeling turbulent flows using a system of interacting Lagrangian particles, whose evolution is governed

*It may be appropriate to mention here that PIML, as a term, was coined in 2016 in the name of the LANL workshop in Santa Fe, NM, which later became a biannual event with the 4th PIML taken place in May of 2022 (41) The workshops brought together ML/AI experts and statistical physicists to present their work where either principal ideas from statistical physics were utilized in the design and training of ML models, including Neural Networks, or ML techniques were integrated into physical models to better describe applications in various quantitative disciplines of sciences and engineering. However, the term was also used later by the authors of ref. 42 to describe Neural Network modification of the classic collocation method (for the numerical solution of ODEs and PDEs; see ref. 43), originally introduced in ref. 44 and then popularized in ref. 45 as the physics-informed neural networks.

by a set of parameterized equations with embedded physical constraints.

In the classic mesh-free Lagrangian turbulence models (2, 3), the dynamics of acceleration, density, and pressure fields evaluated at the particle positions, are approximated following the Lagrangian form of the NS equations. This system of interacting particles provides an excellent platform for simulating turbulent flows in the Lagrangian frame, especially for resolving large-scale structures in the context of LES. This is because the dynamics of a Lagrangian particle cloud accurately represent the sweeping of an eddy associated with the mean particle distance by larger eddies. The interactions among particles in the cloud span over a wide range of scales, which enables the system to capture the multiscale turbulence dynamics even at the scales which are smaller than the filtering length (subgrid scale) because the distances between nonuniform Lagrangian particles may become smaller than the filtering length. However, additional closure models for the emerged subfilter contributions are needed to formally derive a set of dynamic equations that describe Lagrangian particles in the spatially filtered turbulence field. In the classic LES, which are exclusively Eulerian to the best of our knowledge, these closure models have been historically formulated based on heuristics and hand-tuned parameters. Very recently, the PIML framework has been used to improve the parameter tuning, or even to discover new functional forms of the closure for Eulerian LES (50, 60). In this study, we aim to leverage the PIML framework in the Lagrangian, multiparticle framework to learn L-LES models from filtered DNS data, which alleviates us from the exhaustive process of model tuning.

A. Pair-Wise Particle Interaction. A straightforward phenomenological model for the evolution of a system of Lagrangian particles is through the interactions among the particles. In this work, we utilize the main physics idea of the particle-based methodology—i.e., express the flow, e.g., velocity and density fields evolving in time—in terms of particles. Specifically, we consider particles "interacting" with each other in a pair-wise way, where the force exerted on a given particle by the flow is expressed as a sum of terms, each dependent on the current position of the particle and another particle (thus a pair) but also dependent on the velocities and densities evaluated at positions of the two particles. The pair-wise expression for the force acting on any particle in the volume is common for all particle-based methods (PBM), including arguably the most popular example of the PBM—the SPH. Other PBM examples include the family of the mesh-free Galerkin methods and molecular dynamics (MD) methods; see refs. 2, 3, 61, and references therein.

Notice that the prime use of the particle-based, Lagrangian methods, is for either description of subviscous scale phenomena in materials (largely MD methods) and flows or for descriptions of the large scale (energy-containing) part of the flows. In the subviscous setting, particles are associated with the actual physical particles or small patches of flows resolved at the scales which are of the order or smaller than the viscous (Kolmogorov) scale of the actual flow. In the case of large-scale flow modeling, the PBM (primarily SPH) have been used for a sketchy (imprecise) description of the underlying physics at the largest (often astronomical) scales. The main goal of this manuscript is to bridge the gap between the smallest (viscous) and the largest (energy-containing) scales in a continuum description of the flow. We focus here on building L-LES by resolving the range of scales that covers a significant portion of the large-scale part of the inertial range of scales in a turbulent flow.

The traditional approach of the PBM describing continuum-level flows consists of an intuitive derivation of the interparticle interaction by analogy with the NS equations. In this work, we choose to generalize the approach. We do not derive the interactions by analogy with the NS equations but rather model them in the most general way. This general way of modeling is inclusive in the sense that it allows us to include many other modeling ideas used in the LES and PBM so far.

For an N particles system in three-dimensional domain $\Omega \subset \mathbb{R}^3$, we introduce a vector, $\boldsymbol{\phi}_i = [\boldsymbol{v}_i, \rho_i]^T$, built from velocity and density fields evaluated at the particle locations, $\boldsymbol{x}_i \in \Omega$. (Given that our focus here is on the weekly compressible limit of fully compressible flows, we do not account for explicit dependence on the pressure field, expressed via density according to the barotropic equation of state.) Then, the most general equations we consider describing the evolution of the system of particles are

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{x}_i\\ \boldsymbol{\phi}_i\end{bmatrix} = \begin{bmatrix}\boldsymbol{v}_i\\ \sum_{j=1, j\neq i}^{N} \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{\phi}_i, \boldsymbol{\phi}_j)\end{bmatrix}, \forall\, i \in 1, ..., N, \quad [1]$$

where $\boldsymbol{f}$ is (yet to be learned/discovered) 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 $\boldsymbol{\phi}$ evaluated at the positions of the two particles.

In general, we have the freedom to choose the functional form of $\boldsymbol{f}$ in Eq. **1** for modeling, by either using heuristic physical/mathematical arguments or inferring it from data. For example, in the specific choice made within the context of SPH modeling, $\boldsymbol{f}$ is modeled by approximating the right-hand side (RHS) of the Lagrangian NS equations using the reconstructed field by a parameterized smoothing kernel. SPH is formulated to ensure that the physical constraints, such as conservation of momentum, Galilean and rotational invariance, are satisfied. Readers interested in the details of the SPH modeling and learning of NS turbulence are advised to check ref. 59. In this manuscript, we do account for the relevant physical constraints; however, we do not follow the PIML/SPH learning path of ref. 59. Instead, we take here a more general approach, described in the following.

B. Physics-Informed Choice of the Pair-Wise Interaction. Despite the existence of a wide range of functions that might describe the data, a neural network, as a general function approximator with overparameterization, provides an excellent data-driven platform for modeling pair-particle interaction. In the PIML framework, which aims to leverage the capabilities of NNs in advancing scientific applications, the mathematical and physical laws and constraints are either built into the architectures of NN or integrated into the LFs. Various studies have shown that the PIML framework is advantageous in improving robustness, efficiency, and interpretability over a physics-blind one (52, 58). In this section, we describe how the physics constraints are included in (otherwise generic) pair-wise function $\boldsymbol{f}$ in Eq. **1**.

We need to enforce invariance of the dynamics [**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 $\mathcal{T}$. Then, we require that the pair-wise force is $\mathcal{T}$-invariant, i.e., formally,

$$\mathcal{T}\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \boldsymbol{f}(\mathcal{T}\boldsymbol{x}_i, \mathcal{T}\boldsymbol{x}_j, \mathcal{T}\boldsymbol{\phi}_i, \mathcal{T}\boldsymbol{\phi}_j). \quad [2]$$

By using specially designed mathematical structures and features, we can construct the L-LES model with the above-mentioned invariance properties automatically embedded. See *SI Appendix* for more details.

Another important aspect in developing our PIML model is the capability of generalizing over a wider range of parameters that have not been used in the model training. In the context of weakly compressible NS turbulence, we are interested in developing a L-LES model that can describe turbulence with different turbulent Mach numbers, M_t,[†] at fixed Reynolds number.[‡]

Our next key step is to leverage the enormous progress achieved during the recent decade in using NNs as universal function approximators and represent the remaining freedom left in the functions on the RHS of Eq. **1** via NNs. This path leads us to the following NN version of Eq. **1** consistent with the physical symmetries:

$$\frac{d\boldsymbol{\phi}_i}{dt} = \frac{d}{dt}\begin{bmatrix}\rho_i\\ \boldsymbol{v}_i\end{bmatrix} = \sum_{j=1,j\neq i}^{N} \boldsymbol{f}_{NN}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$$
$$= \sum_{j=1,j\neq i}^{N}\begin{bmatrix}\mathcal{NN}_\rho(I_{ij,m};\lambda_\rho)\\ \sum_{k=1}^{2}\mathcal{NN}_{v,k}(I_{ij,m};\lambda_v)\boldsymbol{b}_{ij,k} + \Pi_{ij}\boldsymbol{b}_{ij,1}\end{bmatrix} + \begin{bmatrix}0\\ F_i\end{bmatrix}, \quad [3]$$

where N is the number of particles placed inside of the computational domain, Ω. In all the numerical experiments, reported in this manuscript, Ω is a three-dimensional cube of the volume, $(2\pi)^3$, i.e., $\Omega = [0, 2\pi]^3$. There are three main components in the proposed L-LES model: neural networks $\mathcal{NN}_\rho$ and $\mathcal{NN}_v$, which are parameterized by λ_ρ and λ_v, an eddy viscosity term that models the subgrid contributions Π_{ij}, and an external forcing term F_i. See *SI Appendix* for more details. Here, in Eq. **3**, I and $\boldsymbol{b}$ denote, respectively, the scalar invariants and vector bases of the bases expansion approach of refs. 46 and 58 and are defined using pair-particle information:

$$\boldsymbol{x}_{ij} = \frac{(\boldsymbol{x}_i - \boldsymbol{x}_j)}{d},\ \boldsymbol{v}_{ij} = \frac{(\boldsymbol{v}_i - \boldsymbol{v}_j)}{v_{rms}},\ \rho_{ij} = \frac{1}{2}\frac{(\rho_i + \rho_j)}{\rho_{rms}},$$
$$I_{ij,1} = \frac{\rho_i}{\rho_{rms}},\ I_{ij,2} = \frac{\rho_j}{\rho_{rms}},\ I_{ij,3} = |\boldsymbol{x}_{ij}|,\ I_{ij,4} = |\boldsymbol{v}_{ij}|,$$
$$I_{ij,5} = \boldsymbol{x}_{ij}\cdot\boldsymbol{v}_{ij},\ \boldsymbol{b}_{ij,1} = \boldsymbol{x}_{ij},\ \boldsymbol{b}_{ij,2} = \boldsymbol{v}_{ij}. \quad [4]$$

To ensure the generalizability of the L-LES model, we formulate it in nondimensional form, as shown above. In this work, we use the filtered RMS velocity v_{rms} and average pair-particle distance d as the reference velocity and length scales, while the density fluctuations are normalized by ρ_{rms}. In particular, the latter makes the features independent of M_t to the leading order. The reference length scale d, which represents the resolved turbulence field, is chosen to be within the inertial subrange. With this normalization of features, we expect the learned dynamics can be generalized over different d values within the inertial range, as this is assumed to be universal (64). On the other hand, the forcing term is used in the GT data generation to control M_t and Re_λ and contains an explicit dependency on M_t. Thus, through the nondimensionalization of the input features of NNs, we separate the modeling of pair-particle interaction into M_t independent (represented by NN) and M_t dependent (reference scales) parts. When generalizing over different M_t, the same NN can be employed, while the overall magnitude of the acceleration learned from the GT data can be used to rescale the NN output to the correct magnitude.

A number of comments about the future use of the L-LES model [**3**] in the remainder of the manuscript are in order.

First, it is important to emphasize that all the parameters contributing to the L-LES model are subject to learning, i.e., optimization minimizing the so-called LF expressing the mismatch between DNS data and the L-LES model. There are many plausible and also physics-informed choices for the LF, e.g., based on Lagrangian trajectories, Eulerian field, and statistics, which we discuss in detail in Section 4.

Second, let us recall that the L-LES model is set as a phenomenology/heuristics which is stated in terms of the particles that are not representing exactly the actual particles advected by the velocity fields from the DNS (our GT data). Indeed, we aim to construct L-LES as a statistical model—i.e., model reproducing correctly only important statistical features of turbulence but not exact (deterministic) trajectories. To achieve this statistical goal, we will introduce in the next section a mapping between Lagrangian particles and the Eulerian field, which will then be used to formulate the learning problem and then perform training, validation, and statistical analysis. This mapping from the Lagrangian trajectories to the Eulerian fields has also been adopted in other mesh-free methods, such as the field reconstruction in SPH using the smoothing kernel. Notice, however, that this particles-to-field mapping was used in SPH in a much more targeted way (than in how we use it here). Specifically, the SPH mapping was tuned to approximate the spatial derivatives of the field quantities for approximating the RHS of NS equations.

Finally and third, if all the functions (represented by NNs) and constants are fixed by fitting Eq. **3** from the Lagrangian DNS data, the equations are closed. Notice that this simple closure is in contrast with what is built into the methodology of the SPH and of some other mesh-free methods. Indeed, the SPH scheme also includes reconstruction of the pressure field from the density field, according to the equation of state, then followed by reconstruction of the velocity field from pressure by solving (in the incompressible case) the Poisson equation. Even though we do not have the density-to-pressure-to-velocity reconstruction step explicitly embedded into the L-LES Eq. **3**, we still reconstruct the fields from particles because the reconstructed fields will be used in the following: 1) to build the Eulerian field–based LF to learn parameters of the L-LES model and 2) to test the results via statistical a posteriori analysis.

2. From Lagrangian Trajectories to Eulerian Fields with Smoothing Kernel

Reconstruction of the fields (e.g., of velocity, density, and pressure) defined at an Eulerian grid, which is not changing in time, from the Lagrangian trajectories is one of the key elements

[†] M_t is the turbulent Mach number, which is estimated as the ratio of the typical velocity at the energy-containing scale to the speed of sound, $M_t = v_L/c_s$. In the limit of low Mach number, for single component flows, density can be expanded as $\rho \approx \rho_0 + \rho_1$, where $\rho_1 \sim M_t^2\rho_0$ (62), so that M_t^2 is proportional to the fluid density deviation from the uniform distribution. Note that any particle-based modeling of turbulence requires introducing, discussing, and analyzing compressibility simply because any distribution of particles translates into fluid density which is always spatially nonuniform, even if slightly. Therefore, even if we model fully incompressible turbulence, we should still introduce an effective turbulent Mach number when discussing a particle-based approximation, as $M_t^2 \sim |\rho - \rho_0|/\rho_0$.

[‡] Re number measures a relative strength of the self-advection term to the viscous term evaluated at the integral scale of turbulence. It is thus estimated as $Re = v_L L/\nu = \varepsilon^{1/3}L^{4/3}/\nu$ (63). However, as a part of G.I. Taylor legacy, we often measure the strength of turbulence in terms of the so-called, $Re_\lambda = v_L\lambda/\nu$, where $\lambda = L\sqrt{15/Re}$ is the so-called Taylor microscale, which is sandwiched in between the energy-containing scale, L, and the Kolmogorov (viscous) scale, η, i.e., $\eta < \lambda < L$ (64).

of the SPH methodology where the temporally evolving positions and velocities of the particles are considered interpolation points for the fields evaluated at any other points of the domain. We follow the same basic approach and utilize the smoothing kernel method to reconstruct fields from the positions and velocities of the particles.

Consider an exemplary scalar field, $A(\boldsymbol{x})$, which may be one (of the three) components of the velocity field, the density field, or the pressure field. Image of the field convoluted against the smoothing kernel, $W_\theta(\boldsymbol{x})$, parameterized by the vector of parameters, θ is

$$\langle A(\boldsymbol{x})\rangle = \int_\Omega A(\boldsymbol{x}')\,W_\theta(\boldsymbol{x}-\boldsymbol{x}')d\boldsymbol{x}' \qquad [5]$$

$$= \sum_{j=1}^{N} \Delta V_j A(\boldsymbol{x}_j)\,W_\theta(\boldsymbol{x}-\boldsymbol{x}_j) \qquad [6]$$

where ΔV_j stands for a volume element associated with the particle j. The smoothing kernel, $W_\theta(\boldsymbol{x})$, is assumed to be a continuous and sufficiently smooth function of compact support—nonzero if $|\boldsymbol{x}-\boldsymbol{x}'| \le h = cd$, where c is a sufficiently large, $c > 1$, constant. This assumption, that the kernel is compactly supported, allows a more efficient implementation. Notice that the smoothing kernel also depends on the resolved scale, d. The smoothing kernel is also normalized to guarantee the zeroth order consistency of the integral representation of continuum functions (3).

Eq. **6** allows us to reconstruct $\langle A\rangle$ or its gradient $\langle A(\boldsymbol{x})\rangle$ at a field grid position $\boldsymbol{x}$ from the values of A evaluated at the particle locations $\boldsymbol{x}_i$. More details on the smoothing kernels can be found in *SI Appendix*.

A classic SPH approach consists of postulating the form of the kernel function, $W_\theta(\cdot)$. On the contrary, the approach of this manuscript [which was also developed in our earlier paper (59)] aims at learning the kernel. Specifically, we model the parameterized kernel, subject to the finite-support and normalization constraints described above, using a Neural Network as follows

$$W_\theta(\boldsymbol{r}) = \alpha_n \frac{\mathcal{NN}_\theta(r/h)}{1+\exp(C(r/h-1))}. \qquad [7]$$

$\mathcal{NN}_\theta(r)$ stands for a neural network with scalar input, $r = |\boldsymbol{r}| = |\boldsymbol{x}_{ij}|$, and scalar output; and α_n is the normalization coefficient enforcing the normalization condition. This specific choice of the functional form of $W_\theta(\cdot)$ ensures that the kernel is of compact support, by multiplying a sigmoid function, and that it is properly normalized. C is a parameter that determines the sharpness of the decay to zero within the compact support domain.

The process of learning the smoothing kernel, $W_\theta(\cdot)$, i.e., training NN entering [7], depends on two additional constructs described below. First, in Section A, we introduce the finite neighborhood list approximation, which is a technique that allows us to reduce computational complexity of the convolution involving the smoothing kernel and thus to make the learning problem tractable. Then, in Section B, we discuss the LF used to train the smoothing kernel.

A. Finite Neighborhood List Approximation. Formally, the sum over particle pairs contributing the RHS of Eq. **6** evaluated at the i-th particle location, $\boldsymbol{x} = \boldsymbol{x}_i$, contains N terms. However, for any x, there will be only $O(N_b)$ terms contributing significantly, where N_b is the number of particles that are $O(h)$ close to the i-th particle. To improve the efficiency of this evaluation, we prepare and update dynamically the finite neighborhood list for each particle in the system (i.e., the list consisting of the finite $O(1)$ number of neighbors for each particle) and then use it to truncate the number of terms contributing Eq. **6** to $O(1)$.

B. SK LF. To describe the reconstruction of the velocity field in the Eulerian frame, we introduce a constant, i.e., frozen in time, grid defined on N_E locations, $\boldsymbol{x}_a$, $a = 1,\cdots,N_E$, which we index by the letters from the first three letters of the English alphabet, *a, b, c* (that is in contrast with the Lagrangian particles, changing locations in time, which are indexed by, $i,j,k = 1,\cdots N$).

To learn the smoothing kernel that maps Lagrangian particles to Eulerian fields, we build the SK LF, consisting of three terms,

$$L_{SK} = c_v L_v + c_g L_g + c_n L_n. \qquad [8]$$

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:

$$L_v = \frac{1}{N_E}\sum_{a=1}^{N_E} \left|\langle \boldsymbol{v}(\boldsymbol{x}_a)\rangle - \boldsymbol{v}(\boldsymbol{x}_a)\right|^2, \qquad [9]$$

$$L_g = \frac{1}{N_E}\sum_{a=1}^{N_E}\sum_{\alpha=1}^{3} \left|\langle \nabla \boldsymbol{v}^\alpha(\boldsymbol{x}_a)\rangle - \nabla \boldsymbol{v}^\alpha(\boldsymbol{x}_a)\right|^2, \qquad [10]$$

where superscript $\alpha \in 1,2,3$ denotes the component of the velocity vector; and values of $\boldsymbol{v}$, evaluated at the Lagrangian positions and locations of the Eulerian grid, are assumed taken from the GT data extracted from properly filtered Lagrangian DNS (see Section 4 for details).

There are, obviously, many possible choices one can make in designing the LF. Specific choices of L_v and L_g in Eqs. **9**–**11** were guided by consideration of simplicity and also accepted practices of the SPH community. In particular, we choose to work with the L^2-norm, and we utilize the finite-particle asymmetric approximation for gradients from ref. 6, which, according to refs. 61 and 65, helps to improve numerical accuracy.

The role of the third contribution to the SK-LF, L_n, is to enforce the SK normalization condition and thus learn the normalization coefficient α in Eq. 7.

$$L_n = \left[\int_\Omega W_\theta(\boldsymbol{r})d\boldsymbol{r} - 1\right]^2. \qquad [11]$$

More details on the formulation of the SK-LF can be found in *SI Appendix*.

The smoothing kernel NN [7] is constructed using PyTorch (66) open-source ML library. We build the neural network from four fully connected layers with 20 nodes each and with the hyperbolic tangent activation functions. The parameters (weights and biases of the NN layers) are randomly initialized using the Glorot normal initialization method. The parameters are updated by minimizing L_{SK} using Adam optimizer (67). The initial

learning rate is set to 10^{-3}, and it is then reduced gradually to 10^{-6} throughout the training process. Training is terminated when both training and testing losses are saturated.

3. LF Enforcing Multiphysics in L-LES

To train the PIML L-LES model [**3**] and verify that the model can reproduce realistic turbulence fields and statistics at the resolved scales, we need to introduce a LF whose minimization embodies the training. There are a number of options for the LF to choose from. We present below some of these options and then rely on their linear combination in accomplishing the training task.

Since the L-LES model [**3**] is principally Lagrangian, it is natural to consider a LF stated in terms of particles. Therefore, let us, first of all, construct a trajectory-based LF, L_t, which compares predictions of the L-LES model [**3**] and respective GT data evaluated along the Lagrangian trajectories:

$$L_t = \frac{1}{NM}\sum_{i=1}^{N}\sum_{m=1}^{M}\left|\frac{\boldsymbol{\phi}_i^{(m+1)}-\boldsymbol{\phi}_i^{(m)}}{\Delta} - \left(\sum_{j=1,j\neq i}^{N}\boldsymbol{f}_{NN}(\boldsymbol{x}_i^{(m)},\boldsymbol{x}_j^{(m)},\boldsymbol{\phi}_i^{(m)},\boldsymbol{\phi}_j^{(m)})\right)\right|^2, \quad [12]$$

where $\boldsymbol{\phi}_i^{(m)}$ stands for $\boldsymbol{\phi}_i(t)$ observed (as a Lagrangian part of the GT data) at the discretized moments of time t_m, $m = 0, 1, \cdots, M$; and we use a simple Euler method with time step $\Delta = t_{m+1} - t_m$ to find components of $\boldsymbol{\phi}$ integrating the L-LES model Eq. **3** in discrete time.

However, we may also build a LF based on a comparison of the observed and predicted values of the Eulerian field, thus resulting in

$$L_f = \frac{1}{N_E M}\sum_{a=1}^{N_E}\sum_{m=1}^{M}\left|\langle\boldsymbol{\phi}^{(m+1)}(\boldsymbol{x}_a)\rangle - \boldsymbol{\phi}^{(m+1)}(\boldsymbol{x}_a)\right|^2, \quad [13]$$

Note that $\boldsymbol{\phi}^{(m+1)}(\boldsymbol{x}_a)$ denotes the Eulerian field $\boldsymbol{\phi}$ measured at the grid point $\boldsymbol{x}_a$ at time t_{m+1}.

Finally, we construct the LF which aims to match the Lagrangian particle statistics—predicted and observed. We choose to work with statistics of the single-particle Lagrangian velocities and accelerations—which are the simplest possible Lagrangian characteristics testing the resolved scale. Therefore, we aim to compare respective probability distribution functions (PDFs) represented via histograms.

To compare statistics predicted by the model and statistics of the corresponding GT data, we construct the Kullback–Leibler (KL) LF for velocity (and similarly for acceleration):

$$L_{KL;v} = \sum_{k=1}^{K} P_{GT}(\mu_k)\log\left[\frac{P_{GT;v}(\mu_k)}{P_{pred;v}(\mu_k)}\right], \quad [14]$$

where $P(\mu_k)$ denotes the constructed histogram to represent the PDFs of velocity and acceleration, and μ_k, $k = 1, \ldots, K$ denotes the center of the bin for the histogram. More details on the mathematical formulation and implementation can be found in *SI Appendix*.

We combine the LFs, representing different physical inputs into the structure of L-LES, in one expression,

$$L = c_t L_t + c_f L_f + c_{KL;v} L_{KL;v} + c_{KL;dv/dt} L_{KL;dv/dt}, \quad [15]$$

where c_t, c_f, $c_{KL;v}$, and $c_{KL;dv/dt}$ are constants which can be adjusted (e.g., during training for faster convergence).

4. GT Data

Our "GT" Lagrangian data are generated from the Eulerian NS DNS of weakly compressible, thus low Mach number, stationary HIT. Our numerical implementation of the Eulerian DNS is on a $1{,}024^3$ mesh over the three-dimensional box, $\Omega = [0, 2\pi]^3$. We use sixth-order compact finite differences for spatial discretization (68), the 4th-order Runge–Kutta scheme for time advancement, also imposing triply periodic boundary conditions over the box (69, 70). The velocity field is initialized with 3D Gaussian spectral density enforcing zero mean condition for all components. A large-scale quasi-solenoidal linear forcing term is applied to the simulation at wavenumber $|k| < 2$ to prevent turbulence from decaying (69). The linear forcing term is deterministic and mimics the natural shear turbulence production mechanism in the NS equations (71, 72), in contrast to stochastic-type forcing, e.g., refs. 73 and 74, which may lead to larger temporal fluctuations of energy and other large-scale quantities (75). The forcing method allows the specification of the Kolmogorov scale at the onset and ensures that it remains close to the specified value. The simulations presented here have $\eta/\Delta x = 0.8$, where Δx is the grid spacing. Compared to a standard (well-resolved) spectral simulation with $\eta k_{max} = 1.5$, where k_{max} is the maximum resolved wavenumber, which has $\eta/\Delta x = 1.5/\pi$, the contraction factor is $\approx$0.6 (69, 76) and the maximum differentiation error at the grid (Nyquist) scale is less than 3.5%. Compared to a spectral method with $\eta k_{max} = 1$, which has $\eta/\Delta x = 1/\pi$, the contraction factor is $\approx$0.4 (69, 76), and the maximum differentiation error at the Nyquist scale is less than 0.2%. The initial temperature field is set to be uniform, and the initial pressure field is calculated by solving the Poisson equation. The numerical code is an in-house code developed at Los Alamos National Laboratory (77) to perform high-fidelity simulations of a wide range of turbulent flows, such as compressible isotropic turbulence (69), Rayleigh-Taylor instability (78, 79), shear-driven turbulence (76), shock-turbulence interaction (70), etc. More details about the numerical method and specific setup used here can be found in refs. 69 and 70. The simulations are conducted until the turbulence becomes statistically stationary, which is verified based on the evolution of the kinetic energy and dissipation (69, 70).

Once a statistically steady state of HIT is achieved, we remove the forcing term in the Eulerian DNS scheme and apply a Gaussian filter to the spatiotemporal Eulerian data to obtain the velocity field at the resolved scale, d, and then inject in the filtered flow 64^3 noninertial Lagrangian fluid particles. We remove forcing to make sure that the flow and, consequently, Lagrangian dynamics of passive particles are not masked by nonuniversal details of the large-scale forcing. We use a Gaussian filter, which is commonly used in LES, with a filtering width of the order or larger than the scale d that can be resolved for the $64^3 = 262{,}144$ particles. (In dimensionless units, where the energy-containing scale, L, which is also the size of the box, is $L = 2\pi$, the smallest scale d we can resolve with this number

Table 1. Three use cases considered in training and validation of the L-LES model

Case number	1	2	3
Turbulent Mach number M_t	0.08	0.16	0.04
Taylor Reynolds number Re_λ	200	200	200
Kolmogorov timescale t_η	0.41	0.28	0.73
Usage	Training & validation	Validation	Validation

of particles is $\pi/16$, i.e., 32 times smaller than the size of the domain.)

The particles are placed in the computational domain at random, and then, we follow trajectories of the passively advected particles for the time, τ, which is of the order of (or longer) than the turbulence turnover time of an eddy of size comparable to the resolved scale, d, i.e., $\tau = O(d^{2/3}/\varepsilon^{1/3})$, where ε is the estimate of the energy flux transferred downscale within the inertial range of turbulence.[§]

In this work, we consider three turbulence cases for training and testing the model with comparable Reynolds numbers, $Re_\lambda \approx 200$, and the turbulent Mach number estimated as $M_t = 0.04$, 0.08, and 0.16, respectively. (Table 1) For fare comparison of different approaches to learning L-LES, we extract the filtered Eulerian velocity and velocity gradient snapshot at the same time as the Lagrangian particles. Particle positions and velocities $\{\boldsymbol{x}_i, \boldsymbol{v}_i\}$ are then used to reconstruct velocities at the points of the Eulerian grid according to Eq. **6**. We follow Lagrangian trajectories for as long as the (largest) eddy turnover time $\tau_{eddy} \sim (2\pi)^{2/3}/\varepsilon^{1/3}$ and utilize the data to compute and optimize the particle-based loss. The Eulerian fields are also recorded to compute and optimize the field-based loss.

5. Results and Analyses

This section presents results testing the quality of training of the smoothing kernel, described by Eq. **7**, and of the L-LES model, described by Eq. **3**. We present the SK results first in the A section. Here, the focus is on analyzing the shape of the SK and testing the quality of the Eulerian field predictions for different choices of the coefficients in the SK-LF [**8**]. We then analyze in Section B the L-LES model by investigating the generalization errors of the trained L-LES model and testing the ability to reproduce turbulence statistics collected by simulating the trained L-LES model.

A. Smoothing Kernel. The learning of the SK is discussed first because it is a prerequisite for learning the L-LES model [**3**] and for the following statistical analyses. We experiment with training LF correspondent to different choices of the coefficients in Eq. **8**. Fig. 1 shows the resulting shape of SK and its derivative as a function of r/h for two different choices of LF. We also show as a reference a cubic spline shape of SK popular in the SPH studies (80). We observe that the shape of the SK depends strongly on whether the gradient loss, L_g, is included in the LF. Without L_g, the learned SK does not show the decay property [that is empirically established and thus widely adopted in the SPH studies (3)] and shows a peak at $r/h \approx 0.4$. On the other hand, when L_g is included in the LF, the anticipated decay of the SK with increase in r is observed. This observation is significant because it explains that decay of the SK is instrumental for an accurate prediction of the gradient field.

[§]We remind that d is bounded from above by the size of the box, i.e., $L = 2\pi$ in the dimensionless units of our DNS setting, and from below by the Kolmogorov (viscous) scale, $\eta = O(\nu^{3/4}/\varepsilon^{1/4})$, where ν is the (kinematic) viscosity coefficient.

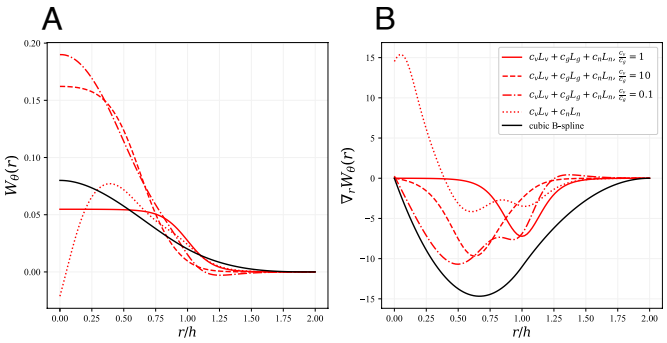

Fig. 1. The learned smoothing kernels (*A*) and their derivatives to r/h (*B*) for different combinations of LFs.

To examine qualitatively the reconstructed fields using the learned SK, we show in Fig. 2 the two-dimensional contours of the reconstructed velocity component $\langle \boldsymbol{v}^\alpha \rangle$ and the velocity gradient, $\langle \nabla_\beta \boldsymbol{v}^\alpha \rangle$ using out-of-sample testing data. Comparing Fig. 2 *A*–*C*, we observe that the large-scale features of the flow can be reproduced using models trained with or without the gradient-based LFs. On the other hand, the qualities of the reconstructed gradient field vary depending on the LF. As shown in Fig. 2 *D* and *E*, the gradient field can not be correctly reproduced by models without accounting for the gradient term in the LF. In general, we observe that even though the profile of the SK has a weak effect on the reconstruction quality of the field, it influences very strongly the profile of higher derivatives. This observation emphasizes the significance of choosing the correct SK. A naive, i.e., not properly trained, choice of the smoothing kernel has a significant deterioration effect on predicting coarse-grained flows and their statistics.

B. Training and Testing L-LES Model.

B.1. Optimal LF: A priori test. In this section, we follow the discussion of the 3 section and discuss the training and testing of the L-LES model using the LF in Eq. **15**. The training procedure is set as follows: a) We start with the trajectory-based LF, i.e., with all coefficients in Eq. **15** but c_t set to zero. This choice of the starting point is motivated by our empirical observation that the trajectory-based LF L_t shows a faster training (decay rate of the LF). b) When the decay rate saturates, we add to the LF all the other field and statistics-based contributions, L_f and L_{KL}s. Notice that convergence of the field-based LF is significantly slower (than that observed initially with the trajectory-based loss). We attribute this observation to the additional layer of spatial averaging (summation over multiple particles) in Eq. **13**. c) The training is stopped once the decay of the combined LF is saturated.

We run a series of diagnostic tests of the L-LES model [**3**]. As custom in ML, we test the quality of the model on the validation set that is unseen during training. Specifically, we test whether the trained model is capable to predict evolution from unseen sections of the flow field (interpolation) or flows with different characteristics (extrapolation). Fig. 3*A* shows the comparison of normalized L^2 errors across different cases.

To test the interpolation abilities of the model, we apply it to the same turbulent model ($M_t \approx 0.08$), however predicting flows at times that are delayed to a different degree with respect to the flow segment used for training. We observe that prediction errors of the model, when applied with a delay, are very similar to

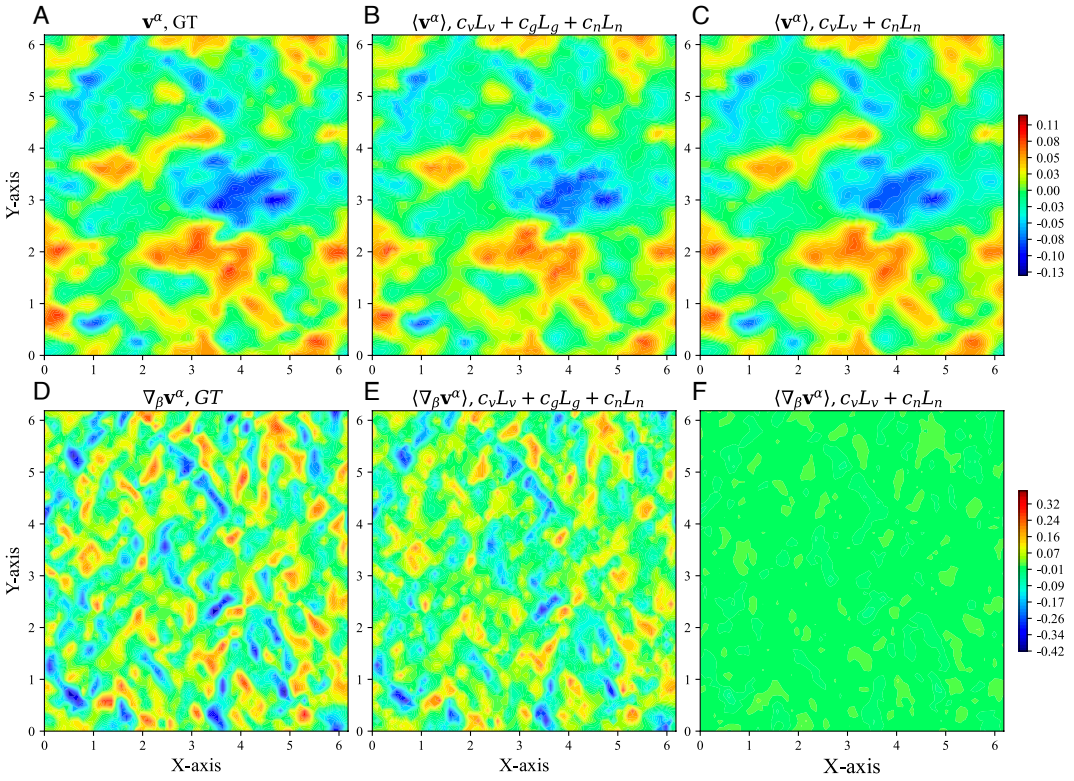

Fig. 2. Two-dimensional contours of the three-dimensional velocity and velocity gradient field at $z = \pi$ reconstructed with the trained SK. The *Top* row shows the x-component of the velocity vector ($\alpha = 1$), and the *Bottom* row shows spatial derivatives of velocity in the x-direction ($\beta = 1$). The GT DNS (*A*) are compared with the reconstructed fields using different combinations of LFs (*B* and *C*). The same comparison is also shown for the derivative of velocity (*D*, *E*, and *F*).

the training errors. This indicates that the models do not overfit, and we can use it to interpolate in time.

Our second test is more demanding. We test whether the model allows extrapolation, i.e., generalizes well to new turbulence regimes with different M_t. Note that the L-LES model in the form of Eq. **3** is dimensionless. After selecting the proper velocity, density, and acceleration scales based on the corresponding M_t, we are able to generalize the dynamics to different turbulence intensities and resolved scales. Results, testing different M_t, corresponding to different values of the resolved scale are also shown in Fig. 3*A*. We observe that the model extrapolates very well to the regimes with different M_ts.

Fig. 3*B* reports a comprehensive comparison of the acceleration statistics for the aforementioned interpolation and extrapolation experiments. We observe that the PDFs predicted by the trained L-LES model match the *GT* data very well.

B.2. Ultimate test of the L-LES model. Once our choice of the LF is validated (see the preceding subsection), we naturally turn to an in-depth, robust, and thus ultimate (we may also call it a posteriori) test of the Lagrangian features and prediction capabilities of the L-LES model.

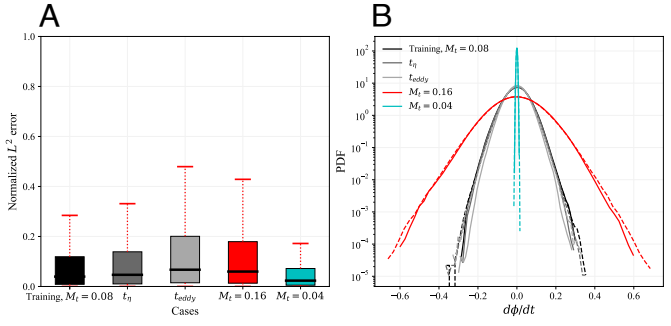

Fig. 3. Performance on the trained L-LES model in interpolation and extrapolation errors over different timescales away from the training data and different levels of turbulent Mach number, respectively. We show (*A*) Normalized L_2 error; and (*B*) PDFs of the prediction.

Consider the following setup: 1) A random snapshot from the filtered *GT* DNS simulation, which has not been used in training, is selected. In this ultimate test, we choose $M_t = 0.16$, a turbulent Mach number never used in the training of the L-LES model. 2) Trained L-LES model Eq. **3** is initialized with 64^3 particles dispersed (at random) at the locations $\boldsymbol{x}_i$ with their properties $\boldsymbol{\phi}_i$ interpolated using the respective Eulerian portion of the GT data. 3) The external force in the L-LES model (Eq. **3**) is chosen similar to the one implemented in the GT DNS Eulerian frame—the energy injection rate is set to match the dissipation rate extracted from the decaying turbulence simulation. 4) We utilize the Varlet integration method to calculate the properties and positions of the Lagrangian particles at the next step, $(\boldsymbol{x}_i^{(m+1)},\ \boldsymbol{\phi}_i^{(m+1)} | i = 1, \cdots, N)$, given input from the previous step, $(\boldsymbol{x}_i^{(m)},\ \boldsymbol{\phi}_i^{(m)} | i = 1, \cdots, N)$. 5) This dynamic multiparticle simulation is run for t_{eddy}, which is the turnover time of the turbulence energy-containing scale. Then, we stop the simulation and perform dynamic and statistical analysis, some requiring interpolation of the Lagrangian particles to the Eulerian grid using the previously learned smoothing kernel.

Fig. 4 shows the comparison of the two-dimensional contours at $z = \pi$ of GT and predicted velocity fields throughout the dynamic process just explained: the GT Eulerian fields at different moments of time are shown in the first row, and the reconstructed fields from L-LES model predictions are shown in the second row. We observe that early in the process, e.g., at the Kolmogorov scale, the contours match each other perfectly. Then, at the times corresponding to roughly one-half of the eddy turnover time, the Lagrangian simulations start to deviate from the exact trajectory of the GT field at the smaller scales.

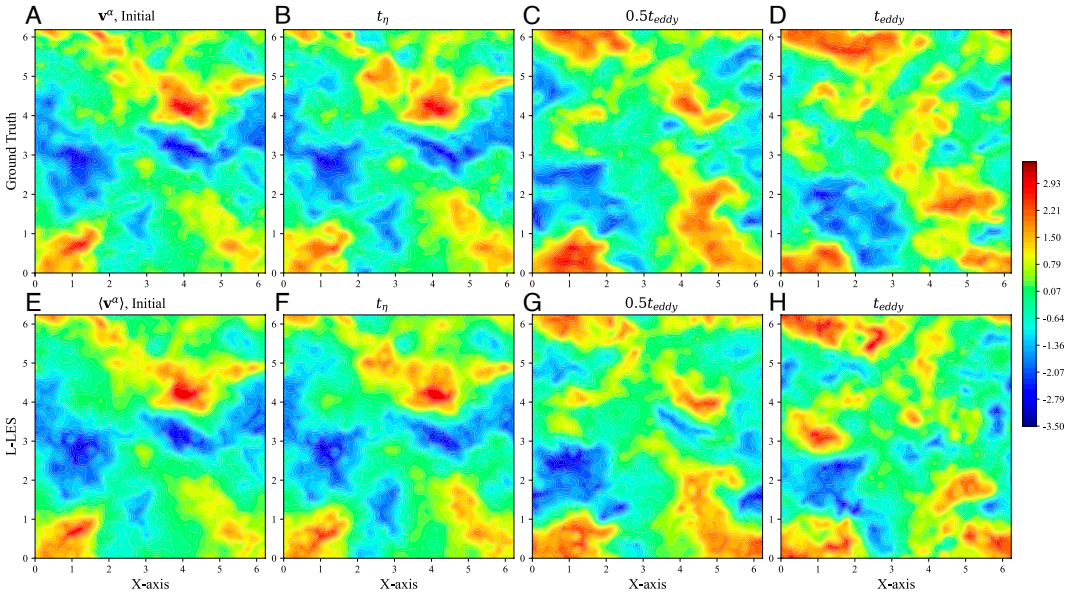

Fig. 4. Two-dimensional contours of the three-dimensional velocity field at plane $z = \pi$ and turbulent Mach number $M_t = 0.16$. The *Top* row shows the evolution of the x component of the velocity vector from the initial snapshot (*A*) to different times scales, including Kolmogorov time scale (*B*) and eddy turnover time (*C* and *D*). The *Bottom* row (*E*, *F*, *G*, and *H*) shows the reconstructed field by the learned SK using predictions from the L-LES model from the same initial condition.

However, the large-scale structures of the two flows (predicted vs GT) still match each other reasonably well. When the time stamp approaches the eddy turnover time (*Right*-most panel in Fig. 4), we start to observe a mismatch. In fact, the mismatch at the longest time stamp is not surprising for the following two reasons. First, our L-LES is a reduced-order model, and as such it is not supposed to reproduce the dynamics of the truly chaotic system one-on-one. We do expect deviation growth in time eventually reaching the magnitude comparable to the flow itself (just as seen in Fig. 4). Second, the linear forcing is used in L-LES and is an approximation of the DNS forcing, but not exactly the same, therefore causing not very fast but still accumulation of the mismatch with time.

Results of another statistical analysis of the ultimate test are shown in Fig. 5. Here, the two-dimensional contours of the vorticity and the turbulent kinetic energy (TKE) derived from the test are juxtaposed with those extracted from the GT data. We observe a reasonable qualitative agreement between the two.

Qualitative analysis of the velocity contours, reported in Figs. 4 and 5, shows the principal capability of the L-LES approach to predict flows fatefully. However, such capabilities are also readily offered by the classic Eulerian LES. To demonstrate that L-LES can capture more physics, we turn now to a more quantitative analysis also emphasizing the Lagrangian perspective of the approach.

Fig. 6 reports the results of the analysis capturing Lagrangian statistics of the turbulent flow. We show here the PDFs of the particle velocity and acceleration as captured by L-LES (vs GT) at the different time stamps. We observe in Fig. 6*A* that the PDF of the particle velocity (coarse-grained at the resolved scale) is close to Gaussian, which is expected for turbulent flows, even for highly compressible cases (81, 82). We also observe that L-LES predictions of the particle velocity at different time stamps match the respective GT data very well even if checked with a delay time (up to an eddy turnover time later). Fig. 6*B* shows a similar comparison for the particle acceleration. Here again, we report a very good quantitative reproduction of the acceleration statistics by L-LES even when checked after evolving for one eddy turnover time.

Our statistical analysis so far was limited to relatively simple objects, e.g., velocity and acceleration statistics at the resolved scale. Let us now turn to more advanced diagnostics and test the so-called Q–R plane statistics of the coarse-grained velocity gradient (58, 83). Statistics in the Q–R plane, where Q and R are second- and third-order invariants of the VGT coarse-grained at the resolved scale, d, is known to determine a variety of important characteristics of turbulence, such as the flow topology, deformation of material volume, energy cascade, and intermittency.

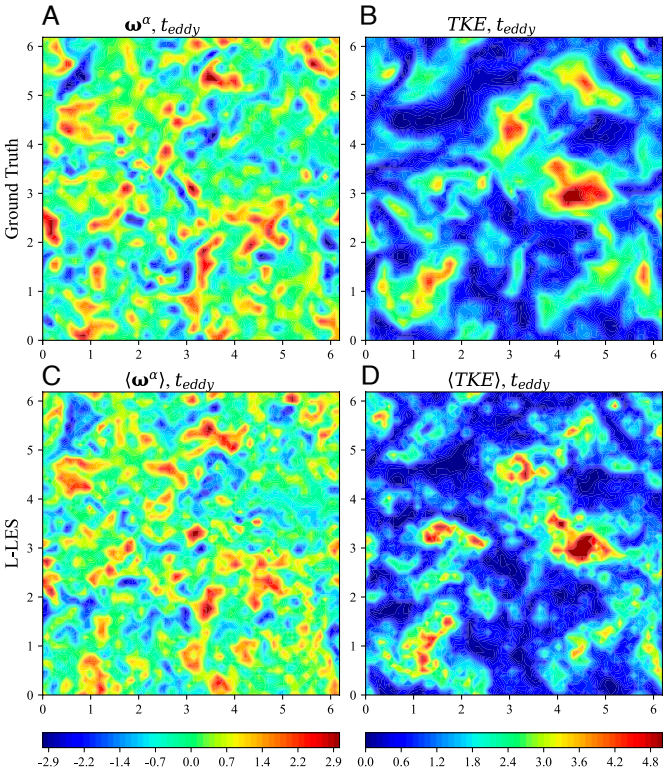

Fig. 5. Two-dimensional contours of the three-dimensional vorticity (*A* and *C*) and turbulent kinetic energy (*B* and *D*) field at the plane $z = \pi$ and turbulent Mach number $M_t = 0.16$. The *Top* row shows the GT DNS field at the resolved scale d at eddy turnover time, and the *Bottom* row shows the fields constructed by the learned smoothing kernel using predictions of the L-LES model from the same initial condition.

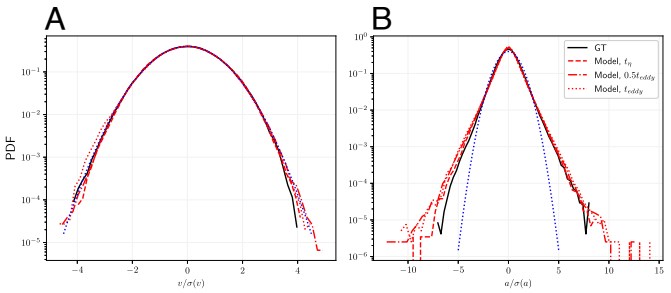

Fig. 6. Normalized PDFs of the Lagrangian particle velocity (*A*) and acceleration (*B*) at different time scales. A reference Gaussian distribution (blue dotted line) is used for comparison.

To test L-LES ability to reconstruct the joint PDF of Q and R, we extract respective statistics following these steps: 1) We obtain $\boldsymbol{v}_i$ by running L-LES model [**3**] and then compute $\nabla \boldsymbol{v}_i$ applying the smoothing kernel [**5**, **7**]. 2) We calculate the anisotropic part of the VGT according to, $\nabla \boldsymbol{v}_i^* = \nabla \boldsymbol{v}_i - \frac{1}{3}\mathrm{Tr}(\nabla \boldsymbol{v}_i)$. Then, the second and third invariants of the anisotropic part of the velocity tensor are computed from

$$Q_i = -\frac{1}{2}\mathrm{Tr}(\nabla \boldsymbol{v}_i^* \times \nabla \boldsymbol{v}_i^*), \qquad [16]$$

$$R_i = -\frac{1}{3}\mathrm{Tr}(\nabla \boldsymbol{v}_i^* \times \nabla \boldsymbol{v}_i^* \times \nabla \boldsymbol{v}_i^*),\ \forall i \in 1, \ldots, N, \qquad [17]$$

where $\times$ denotes matrix multiplication.

We compute the second and third invariants for the particles at different time stamps from the L-LES predictions and aggregate them in a joint PDF, summing over particles. The results are shown in Fig. 7 *A* and *B* in the form of the isolines of the joint PDF of the invariants. We observe the distinctive "tear-drop" shape, expressing complex dynamics of turbulence, in all the subfigures of Fig. 7, consistent with what is referred to in the literature as the statistical geometry of turbulence (see ref. 83 and references therein). Specifically, Fig. 7*A* shows the (Q,R) PDF from the GT, filtered DNS results. Fig. 7*B* shows the (Q,R) joint PDF reconstructed by L-LES predictions at eddy turnover time. It is evident that the tear-drop shape of the joint (Q,R) PDF is reproduced by L-LES very well even at much later times, indicating the L-LES model is capable of reproducing detailed long-time statistics of Lagrangian turbulence.

The ability of L-LES to predict Lagrangian trajectories allows for the study of much more intimate features of turbulence, in particular statistics of the VGT as seen from the Lagrangian frame, i.e., PDF of the VGT accumulated over Lagrangian particles (58, 84). The Lagrangian dynamics of the VGT projected to the (Q,R) plane is shown in Fig. 7 *C* and *D*. Specifically, we show in Fig. 7 the mean (over Lagrangian particles) vector of the time derivatives of the VGT invariants, $(\frac{dR}{dt}, \frac{dQ}{dt})$, conditioned to the values of the invariants. We observe that the vector field prediction of the L-LES, shown in Fig. 7*D*, agrees very well with the results extracted from the GT/DNS data coarse-grained at the resolved scale, d, shown in Fig. 7*C*. The agreement is especially impressive given that we did not enforce the vector field matching in the LF used to train the L-LES. In other words, the agreement between L-LES and DNS confirms that the former learns detailed Lagrangian features of turbulence from the latter accurately.

Fig. 8 shows the evolution of the energy spectrum for two different settings, corresponding to $M_t = 0.16$ and 0.08. Following our general logic, we train L-LES on the DNS/GT data correspondent to $M_t = 0.08$ and then compare the energy spectra predictions at both $M_t = 0.08$ and $M_t = 0.16$ with the results extracted from the DNS and filtered DNS. The L-LES energy spectra results are extracted from the Lagrangian data in two steps. First, we apply the smoothing kernel to Lagrangian data to reconstruct the snapshot of the velocities over the Eulerian grid, and then, we compute the energy spectrum (via standard Fourier transform), also truncating it at the wave vector correspondent to the resolved scale, d. At small M_t, the energy from the compressible component of the velocity field is negligible compared to the solenoidal component (81), so we compute the total energy spectrum instead of decomposing the velocity field and computing them separately. From Fig. 8, we observe the DNS datasets show an inertial range scaling that is slightly steeper than the classical $-5/3$ scaling law, which could be attributed

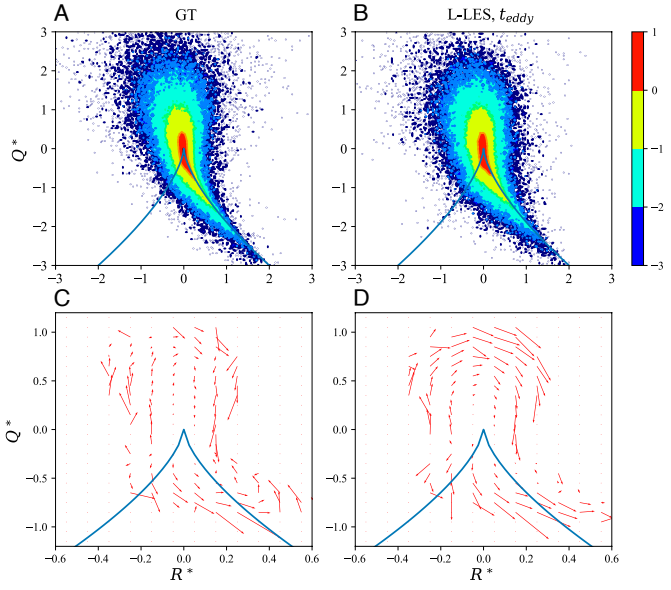

Fig. 7. Statistical geometry of turbulence, as revealed by the joint PDF and vector field of the invariants of the coarse-grained velocity gradient (Q and R). Subfigure (*A*) shows the GT (filtered DNS). Subfigure (*B*) shows the PDF reconstructed from the L-LES model prediction at the time τ_{eddy} from the initial condition. Subfigures (*C* and *D*) show the corresponding vector field plot of the mean $(dQ/dt, dR/dt)$ conditioned to the values of the VGT invariants, Q, R, and averaged over Lagrangian particles, as extracted from the DNS data coarse-grained over, d (*C*), and predicted by L-LES model (*D*). The VGT statistics is computed using the learned smoothing kernel (see text for details).

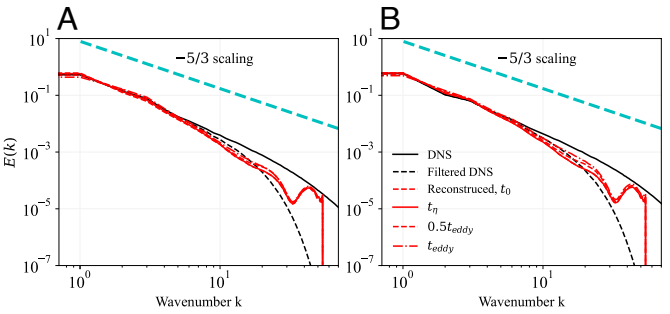

Fig. 8. The energy spectrum of turbulence field as extracted from the Eulerian DNS/GT data vs reconstructed by L-LES at different time stamps. Subfigures (*A*) and (*B*) show comparisons at $M_t = 0.16$ and $M_t = 0.08$, respectively, while the training for both cases is done at $M_t = 0.08$.

to the intermittency corrections (74, 85), and compressibility (81, 86). The filtered DNS data further deviate from the scaling. The reconstructed spectra at the initial time stamp t_0 agree very well with the filtered DNS at large scales. At smaller scales, reconstructed spectra are slightly more intense than those of the filtered DNS. We explain this overestimation of the turbulence intensity at the smaller scales by the fact that the interparticle distance fluctuates, therefore leading to additional energy stored at scales smaller than d. After evolving the Lagrangian particles using the trained L-LES model, we observe that the spectra remain the same over one eddy turnover time, suggesting that the L-LES model can reproduce long-time stationary turbulence statistics.

B.3. Computational cost of the L-LES model. In this study, we trained the PIML smoothing kernel and L-LES model using a single Nvidia A100 GPU with 40 GB of memory. It took approximately 6 h for the PIML-SK to converge and 12 h for the L-LES model to converge. When performing forward simulation using the learned L-LES model, we use the same Nvidia A100 GPU and each iteration takes approximately 0.5 s, in contrast to 3.3 s for the $1{,}024^3$ resolution DNS using 8192 AMD EPYC 7H12 CPUs.

6. Conclusions

Representing turbulent flows with particles is advantageous because it potentially offers the ultimate model reduction—underlying physics becomes transparent and mathematics simpler. In the case of passive scalar turbulence, it allows us to express n-th order correlation functions via dispersion of n particles. In the case of "Burgulence," i.e., turbulence described by Burgers equations, it allows us to focus attention on the dynamics and interaction of shocks.

This manuscript presents an attempt to carry over this Lagrangian logic to the much more challenging case of NS turbulence. We show how to build a reduced model of HIT in terms of a system of N coevolving Lagrangian markers/particles. Specifically, we claim fateful representation of HIT within the larger scale portion of the inertial range, which extends upscale from the resolved scale, d, to the energy-containing scale, L, in terms of at least N particles, where $N = O((L/d)^3) \gg 1$.

We call this approach L-LES because it can be viewed as a Lagrangian version of the classic and highly popular LES approach, which was thus far solely Eulerian.

One may as well argue that L-LES extends the growing list of particle-based approaches to modeling complex dynamics of fluids and materials. Insofar the SPH, which is arguably one of the most popular PBM in fluid mechanics, is the L-LES closest ancestor.

L-LES is also an advanced instance of the PIML approach because it fits the GT data with the dynamic model which is sufficiently loose but also consistent with our current physical interpretation of the geometry and dynamics of HIT.

L-LES, considered a learning problem, is split into two parts. First, we learn/train the SK, represented by Eq. **7**, which maps positions and velocities of the Lagrangian particles to the velocity and density fields evaluated over the Eulerian grid. Then, we fit the L-LES model itself, represented by Eq. **3**. In both cases, the training is done on the DNS data, and we rely on neural networks to provide the best fit to the degrees of freedom remaining after all the physical considerations and mathematical constraints, such as normalization of the SK, translational and rotational invariance, are taken into account. Physics also enters into the design of the LF.

The SK-LF is built from a combination of terms enforcing SK normalization, and also matching the velocity field and the velocity gradient field with the SK applied to the respective GT Lagrangian data, we obtained by advecting particles placed in the filtered (at the resolved scale) DNS. Our experiments show that combining all three terms is critical for learning the SK.

The L-LES Eq. **3**, describing Lagrangian dynamics of the probe particles at the resolved scale consistent with the filtered NS equations and thus accounting for the filtered stress tensor, subfilter, viscous, and forcing contributions, are at the core of our modeling. We show that given the capability of a deep NN used to represent the remaining (and sufficiently large) freedom in the L-LES equations, the L-LES model is more inclusive than all previously considered particle-based models of turbulence, including SPH. Three types of LFs were introduced in order to train the L-LES model – trajectory-based, field-based, and statistics-based. The model is trained on the Eulerian and Lagrangian GT data extracted from DNS simulations at turbulent Mach number $M_t = 0.08$. Various a priori tests of the L-LES model all show that the model is capable to predict accurately multiparticle dynamics in unseen portions of the data at $M_t = 0.08$, and it also generalizes (extrapolates) well to regimes with other values of M_t within the weakly compressible limit.

We also conducted a posteriori tests of advanced Lagrangian physics of turbulence where the model was not enforced directly. These tests/experiments included seeding Lagrangian particles into unseen (in training) parts of the GT data and collecting advanced statistics, e.g., on the energy spectrum and invariants of the filtered velocity gradients. All of the tests have reported a quality agreement between the model and the GT.

Data, Materials, and Software Availability. The source code for our models, including the smoothing kernel and L-LES model, and training datasets are available at Github: https://github.com/lanl/Lagrangian-LES (87).

ACKNOWLEDGMENTS. This work was performed under the auspices of DOE. Financial support comes from Los Alamos National Laboratory (LANL), Laboratory Directed Research and Development (LDRD) project "Machine Learning for Turbulence," 20180059DR. LANL, an affirmative action/equal opportunity employer, is managed by Triad National Security, LLC, for the National Nuclear Security Administration of the US Department of Energy under contract 89233218CNA000001.

Author affiliations: [a]Information Sciences Group, Computer, Computational and Statistical Sciences Division (CCS-3), Los Alamos National Laboratory, Los Alamos, NM 87545; [b]Graduate Interdisciplinary Program in Applied Mathematics and Department of Mathematics, University of Arizona, Tucson, AZ 85721; and [c]Computational Physics and Methods Group, Computer, Computational and Statistical Sciences Division (CCS-2), Los Alamos National Laboratory, Los Alamos, NM 87545

Author contributions: Y.T. designed research; Y.T. performed research; Y.T. contributed new reagents/analytic tools; Y.T., M.W., M.S., C.F., C.H., D.L., and M.C. analyzed data; and Y.T., D.L., and M.C. wrote the paper.

The authors declare no competing interest.

1. P. Sagaut, *Large Eddy Simulation for Incompressible Flows: An Introduction* (Springer Science & Business Media, 2006).
2. G. R. Liu, Y. T. Gu, *An Introduction to Meshfree Methods and Their Programming* (Springer Science & Business Media, 2005).
3. M. Liu, G. Liu, Smoothed particle hydrodynamics (SPH): An overview and recent developments. *Arch. Comput. Methods Eng.* **17**, 25–76 (2010).
4. P. K. Yeung, Lagrangian investigations of turbulence. *Annu. Rev. Fluid Mech.* **34**, 115–142 (2002).
5. R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *MNRAS* **181**, 375–389 (1977).
6. J. Monaghan, Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.* **30**, 543–574 (1992).

7. J. J. Monaghan, Smoothed particle hydrodynamics and its diverse applications. *Annu. Rev. Fluid Mech.* **44**, 323–346 (2012).
8. M. Herant, W. Benz, W. R. Hix, C. L. Fryer, S. A. Colgate, Inside the supernova: A powerful convective engine. *ApJ* **435**, 339 (1994).
9. W. Benz, E. Asphaug, Catastrophic disruptions revisited. *Icarus* **142**, 5–20 (1999).
10. M. R. Bate, I. A. Bonnell, N. M. Price, Modelling accretion in protobinary systems. *MNRAS* **277**, 362–376 (1995).
11. S. Rosswog *et al*., Mass ejection in neutron star mergers. *Astron. Astrophys.* **341**, 499–526 (1999).
12. C. S. Frenk *et al*., The Santa Barbara cluster comparison project: A comparison of cosmological hydrodynamics solutions. *ApJ* **525**, 554–582 (1999).
13. V. Bromm, P. S. Coppi, R. B. Larson, Forming the first stars in the universe: The fragmentation of primordial gas. *ApJ Lett.* **527**, L5–L8 (1999).
14. V. Springel, The cosmological simulation code GADGET-2. *MNRAS* **364**, 1105–1134 (2005).
15. G. Stinson *et al*., Star formation and feedback in smoothed particle hydrodynamic simulations–I. Isolated galaxies. *MNRAS* **373**, 1074–1090 (2006).
16. D. J. Price, Smoothed particle hydrodynamics and magnetohydrodynamics. *J. Comput. Phys.* **231**, 759–794 (2012).
17. M. S. Shadloo, G. Oger, D. Le Touzé, Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges. *Comput. Fluids* **136**, 11–34 (2016).
18. S. J. Lind, B. D. Rogers, P. K. Stansby, Review of smoothed particle hydrodynamics: Towards converged Lagrangian flow modelling. *Proc. R. Soc. A* **476**, 20190801 (2020).
19. 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).
20. S. Rosswog, Modelling astrophysical fluids with particles. *Proc. Natl. Acad. Sci. U.S.A.* **16**, 382–397 (2022).
21. J. J. Monaghan, SPH compressible turbulence. *MNRAS* **335**, 843–852 (2002).
22. E. Y. Lo, S. Shao, Simulation of near-shore solitary wave mechanics by an incompressible SPH method. *Appl. Ocean Res.* **24**, 275–286 (2002).
23. R. A. Dalrymple, B. Rogers, Numerical modeling of water waves with the SPH method. *Coast. Eng.* **53**, 141–147 (2006).
24. D. J. Price, C. Federrath, A comparison between grid and particle methods on the statistics of driven, supersonic, isothermal turbulence. *MNRAS* **406**, 1659–1674 (2010).
25. D. J. Price, Resolving high Reynolds numbers in smoothed particle hydrodynamics simulations of subsonic turbulence. *MNRAS: Lett.* **420**, L33–L37 (2012).
26. A. Mayrhofer, D. Laurence, B. Rogers, D. Violeau, DNS and LES of 3-D wall-bounded turbulence using smoothed particle hydrodynamics. *Comput. Fluids* **115**, 86–97 (2015).
27. T. S. Tricco, D. J. Price, C. Federrath, A comparison between grid and particle methods on the small-scale dynamo in magnetized supersonic turbulence. *MNRAS* **461**, 1260–1275 (2016).
28. D. J. Price *et al*., Phantom: A smoothed particle hydrodynamics and magnetohydrodynamics code for astrophysics. *Public. Astron. Soc. A.* **35**, e031 (2018).
29. D. Rennehan, Mixing matters. *MNRAS* **506**, 2836–2852 (2021).
30. D. Violeau, R. Issa, Numerical modelling of complex turbulent free-surface flows with the SPH method: An overview. *Int. J. Numer. Methods Fluids* **53**, 277–304 (2007).
31. D. De Padova, M. Mossa, S. Sibilla, SPH numerical investigation of the velocity field and vorticity generation within a hydrofoil-induced spilling breaker. *Environ. Fluid Mech.* **16**, 267–287 (2016).
32. A. Leroy, D. Violeau, M. Ferrand, C. Kassiotis, Unified semi-analytical wall boundary conditions applied to 2-D incompressible SPH. *J. Comput. Phys.* **261**, 106–129 (2014).
33. G. Bicknell, The equations of motion of particles in smoothed particle hydrodynamics. *SIAM J. Sci. Stat. Comput.* **12**, 1198–1206 (1991).
34. A. Di Mascio, M. Antuono, A. Colagrossi, S. Marrone, Smoothed particle hydrodynamics method from a large eddy simulation perspective. *Phys. Fluids* **29**, 035102 (2017).
35. M. Antuono, S. Marrone, A. Di Mascio, A. Colagrossi, Smoothed particle hydrodynamics method from a large eddy simulation perspective. Generalization to a quasi-Lagrangian model. *Phys. Fluids* **33**, 015102 (2021).
36. P. F. Hopkins, A new class of accurate, mesh-free hydrodynamic simulation methods. *MNRAS* **450**, 53–110 (2015).
37. J. M. Owen, J. V. Villumsen, P. R. Shapiro, H. Martel, Adaptive smoothed particle hydrodynamics: Methodology. II. *Astrophys. J. Suppl. Ser.* **116**, 155 (1998).
38. N. Sebe, I. Cohen, A. Garg, T. S. Huang, *Machine Learning in Computer Vision* (Springer Science & Business Media, 2005), vol. 29.
39. L. Deng, X. Li, Machine learning paradigms for speech recognition: An overview. *IEEE Trans. Audio Speech Lang. Process.* **21**, 1060–1089 (2013).
40. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (MIT Press, 2016).
41. See online links and materials from the series of LANL conferences, in Santa Fe, NM, in 2018, 2020, 2022 on "Physics Informed Machine Learning" (2022).
42. G. E. Karniadakis *et al*., Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
43. L. Lapidus, G. F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering* (John Wiley & Sons, 2011).
44. I. Lagaris, A. Likas, D. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.* **9**, 987–1000 (1998).
45. M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
46. J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166 (2016).
47. R. King, O. Hennigh, A. Mohan, M. Chertkov, From deep to physics-informed learning of turbulence: Diagnostics. arXiv [Preprint] (2018). http://arxiv.org/abs/1810.07785 (Accessed 12 May 2018).
48. J. X. Wang, J. L. Wu, H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids* **2**, 034603 (2017).
49. A. T. Mohan, N. Lubbers, M. Chertkov, D. Livescu, Embedding hard physical constraints in neural network coarse-graining of three-dimensional turbulence. *Phys. Rev. Fluids* **8**, 014604 (2023).
50. K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377 (2019).
51. A. T. Mohan, D. Daniel, M. Chertkov, D. Livescu, Compressed convolutional LSTM: An efficient deep learning framework to model high fidelity 3D turbulence. arXiv [Preprint] (2019). http://arxiv.org/abs/1903.00033 (Accessed 3 May 2019).
52. A. T. Mohan, D. Tretiak, M. Chertkov, D. Livescu, Spatio-temporal deep learning models of 3D turbulence with physics informed diagnostics. *J. Turbul.* **21**, 484–524 (2020).
53. K. Fukami, K. Fukagata, K. Taira, Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid. Mech.* **870**, 106–120 (2019).
54. G. D. Portwood *et al*., Turbulence forecasting via Neural ODE. arXiv [Preprint] (2019). http://arxiv.org/abs/1911.05180.
55. L. Ladickỳ, S. Jeong, B. Solenthaler, M. Pollefeys, M. Gross, Data-driven fluid simulations using regression forests. *ACM Trans. Graph.* **34**, 1–9 (2015).
56. C. Schenck, D. Fox, "SPNets: Differentiable fluid dynamics for deep neural networks" in *Conference on Robot Learning* (PMLR, 2018), pp. 317–335.
57. B. Ummenhofer, L. Prantl, N. Thuerey, V. Koltun, "Lagrangian fluid simulation with continuous convolutions" in *International Conference on Learning Representations (ICLR)* (2019).
58. Y. Tian, D. Livescu, M. Chertkov, Physics-informed machine learning of the Lagrangian dynamics of velocity gradient tensor. *Phys. Rev. Fluids* **6**, 094607 (2021).
59. M. Woodward *et al*., Physics informed machine learning with smoothed particle hydrodynamics: Hierarchy of reduced Lagrangian models of turbulence. *Phys. Rev. Flu.* **8**, 054602 (2023).
60. G. D. Portwood, B. T. Nadiga, J. A. Saenz, D. Livescu, Interpreting neural network models of residual scalar flux. *J. Fluid Mech.* **907**, A23 (2021).
61. G. R. Liu, M. B. Liu, *Smoothed Particle Hydrodynamics: A Meshfree Particle Method* (World Scientific, 2003).
62. D. Livescu, Turbulence with large thermal and compositional density variations. *Annu. Rev. Fluid Mech.* **52**, 309–341 (2020).
63. U. Frisch, A. Kolmogorov, *Turbulence: The Legacy of A. N. Kolmogorov* (Cambridge University Press, 1995).
64. H. Tennekes, J. Lumley, *A First Course in Turbulence* (MIT Press, 1978).
65. M. Liu, G. R. Liu, Restoring particle consistency in smoothed particle hydrodynamics. *Appl. Numer. Math.* **56**, 19–36 (2006).
66. A. Paszke *et al*., "Pytorch: An imperative style, high-performance deep learning library" in *Advances in Neural Information Processing Systems, 32*, H. Wallach *et al*., Eds. (Curran Associates, Inc., 2019), pp. 8024–8035.
67. D. P. Kingma, J. Ba, A method for stochastic optimization. arXiv [Preprint] (2014). https://arxiv.org/abs/1412.6980 (Accessed 30 January 2017).
68. S. K. Lele, Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **103**, 16–42 (1992).
69. M. Petersen, D. Livescu, Forcing for statistically stationary compressible isotropic turbulence. *Phys. Fluids* **22**, 116101 (2010).
70. J. Ryu, D. Livescu, Turbulence structure behind the shock in canonical shock–vortical turbulence interaction. *J. Fluid Mech.* **756**, R1 (2014).
71. T. S. Lundgren, Linearly forced isotropic turbulence. *Center Turb. Res. Annu. Res. Briefs* **2003**, 461–473 (2003).
72. C. Rosales, C. Meneveau, Linear forcing in numerical simulations of isotropic turbulence: Physical space implementations and convergence properties. *Phys. Fluids* **17**, 095106 (2005).
73. V. Eswaran, S. Pope, An examination of forcing in direct numerical simulations of turbulence. *Comput. Fluids* **16**, 257–278 (1988).
74. C. Federrath, R. S. Klessen, L. Iapichino, J. R. Beattie, The sonic scale of interstellar turbulence. *Nat. Astron.* **5**, 365–371 (2021).
75. M. Overholt, S. Pope, A deterministic forcing scheme for direct numerical simulations of turbulence. *Comput. Fluids* **27**, 11–28 (1998).
76. J. R. Baltzer, D. Livescu, Variable-density effects in incompressible non-buoyant shear-driven turbulent mixing layers. *J. Fluid Mech.* **900**, A16 (2020).
77. D. Livescu, J. Mohd-Yusof, M. R. Petersen, J. W. Grove, "CFDNS: A computer code for Direct Numerical Simulation of turbulent flows" (Technical Report, Los Alamos National Laboratory, 2009).
78. D. Livescu, T. Wei, P. Brady, Rayleigh–Taylor instability with gravity reversal. *Physica D: Nonlinear Phenom.* **417**, 132832 (2021).
79. D. Aslangil, D. Livescu, A. Banerjee, Effects of Atwood and Reynolds numbers on the evolution of buoyancy-driven homogeneous variable-density turbulence. *J. Fluid Mech.* **895**, A12 (2020).
80. J. J. Monaghan, J. C. Lattanzio, A refined particle method for astrophysical problems. *Astron. Astrophys.* **149**, 135–143 (1985).
81. J. Wang, T. Gotoh, T. Watanabe, Spectra and statistics in compressible isotropic turbulence. *Phys. Rev. Fluids* **2**, 013403 (2017).
82. C. Federrath, On the universality of supersonic turbulence. *MNRAS* **436**, 1245–1257 (2013).
83. M. Chertkov, A. Pumir, B. I. Shraiman, Lagrangian tetrad dynamics and the phenomenology of turbulence. *Phys. Fluids* **11**, 2394–2410 (1999).
84. Y. Tian, F. A. Jaberi, D. Livescu, Density effects on post-shock turbulence structure and dynamics. *J. Fluid Mech.* **880**, 935–968 (2019).
85. Z. S. She, E. Leveque, Universal scaling laws in fully developed turbulence. *Phys. Rev. Lett.* **72**, 336 (1994).
86. A. G. Kritsuk, P. Padoan, R. Wagner, M. L. Norman, "Scaling laws and intermittency in highly compressible turbulence" in *AIP Conference Proceedings* (American Institute of Physics, 2007), vol. 932, pp. 393–399.
87. Y. Tian, Lagrangian Large Eddy Simulations vis Physics-Informed Machine Learning. Lagrangian-LES. https://github.com/lanl/Lagrangian-LES. Deposited 14 August 2023.