



Learning fast, accurate, and stable closures of a kinetic theory of an active fluid

Suryanarayana Maddu^{a,*}, Scott Weady^a, Michael J. Shelley^{a,b}

^a Center for Computational Biology, Flatiron Institute, Simons Foundation, New York, NY, 10010, USA

^b Courant Institute of Mathematical Sciences, New York University, New York, NY, 10012, USA

ARTICLE INFO

Keywords:

Active fluids
Data-driven modeling
Closure modeling
Neural networks
Multiscale modeling

ABSTRACT

Important classes of active matter systems can be modeled using kinetic theories. However, kinetic theories can be high dimensional and challenging to simulate. Reduced-order representations based on tracking only low-order moments of the kinetic model serve as an efficient alternative, but typically require closure assumptions to model unrepresented higher-order moments. In this study, we present a learning framework based on neural networks that exploits rotational symmetries in the closure terms to learn accurate closure models directly from kinetic simulations. The data-driven closures demonstrate excellent *a-priori* predictions comparable to the state-of-the-art Bingham closure. We provide a systematic comparison between different neural network architectures and demonstrate that nonlocal effects can be safely ignored to model the closure terms. We develop an active learning strategy that enables accurate prediction of the closure terms across the entire parameter space using a single neural network without the need for retraining. We also propose a data-efficient training procedure based on time-stepping constraints and a differentiable pseudo-spectral solver, which enables the learning of stable closures suitable for *a-posteriori* inference. The coarse-grained simulations equipped with data-driven closure models faithfully reproduce the mean velocity statistics, scalar order parameters, and velocity power spectra observed in simulations of the kinetic theory. Our differentiable framework also facilitates the estimation of parameters in coarse-grained descriptions conditioned on data.

1. Introduction

The field of active matter has found exciting applications in many areas such as biophysics, synthetic chemistry, and material science [1–3]. Active matter systems consist of collections of agents, such as particles or macromolecules, that convert chemical energy into mechanical work, creating active forces or stresses that are communicated across the system via direct steric interactions, cross-linking proteins, or through long-ranged hydrodynamic interactions. The consequent coordination can lead to complex spatiotemporal dynamics. A central question in these systems is the relation between the microscopic interactions on the scale of individual particles and the emerging self-organization and collective dynamics at the scale of the system [4,3]. In this study, we focus on active suspensions, in which the active particles are suspended in a viscous fluid and long-ranged hydrodynamic interactions, alongside steric interactions, are important. A few prominent experimental realizations include suspensions of swimming bacteria and *in vitro* mixtures of cytoskeletal filaments and motor proteins. These systems exhibit flows characterized by large-scale vortices

* Corresponding author.

E-mail address: smaddu@flatironinstitute.org (S. Maddu).

<https://doi.org/10.1016/j.jcp.2024.112869>

Received 11 August 2023; Received in revised form 11 February 2024; Accepted 16 February 2024

Available online 23 February 2024

0021-9991/© 2024 Elsevier Inc. All rights reserved.

and jets with sizes and speed much greater than those associated with the individual swimmer or motor [5,6]. Though these spontaneous flows exhibit turbulent characteristics, energy input is due to its microscopic components. Hence, in active matter systems, the flows generated are autonomous, and the emerging pattern of energy injection is self-organized [7].

Many computational and mathematical approaches have been used to investigate active matter systems, ranging from large-scale discrete particle simulations to continuum formulations. Discrete simulations explicitly model the dynamics of active particles and their hydrodynamic and steric interactions. These models have proven invaluable and are instrumental to achieving qualitative agreements with experimental observations. However, due to their discrete nature, such descriptions cannot be easily analyzed mathematically [3] and remain constrained in the number of particles simulated. Several continuum models have been proposed for studying active suspensions based on the extension of the theory of passive liquid crystals to include activity [8], and phenomenological models constructed based on symmetries to include steric interactions [9,6]. Alternate approaches based on coupling the microscopic and macroscopic scales include kinetic theories where particle positions and their conformations, such as particle orientation, are represented through a distribution function and evolved according to the Smoluchowski equation coupled to the equations of activity-driven hydrodynamics. Even though kinetic theories involve coefficients that are well grounded in microscopic details and are amenable to analytical treatment, they are not immune from computational challenges. The particle distribution function depends on both particle position and orientation leading to $2d - 1$ degrees of freedom. In dense suspensions with strong alignment interactions, the orientation field needs to be resolved with very high accuracy, which prohibitively increases the cost of kinetic simulations, even in two dimensions. To circumvent this, approximate coarse-grained models that only track the first few orientational moments of the distribution function have been proposed. However, evolution equations for the moments are not closed as they depend on unrepresented higher-order moments of the distribution function. Several closure models have been proposed in the context of liquid crystals and particle suspensions with approximations based upon weak or strong flow or near isotropy [10,3], or by interpolating between such states [11]. In addition, accurate and efficient means exist for both polar [12] and apolar suspensions [13,14] that rely on modeling the particle distribution function by a Bingham distribution on the unit sphere. Most of these approximations either involve assuming quasi-equilibrium approximation of the distribution function or closures that apply to specific flow regimes [3].

In this paper, we propose a data-driven framework for learning closure models directly from data generated from kinetic theory. We draw inspiration from fluid dynamics applications where machine learning has been used to model discrepancies between a prescribed phenomenological closure and high-fidelity direct numerical simulations (DNS) [15]. Most of the studies involve the use of supervised learning techniques where a neural network is employed to learn corrective dynamics using DNS data as the target. Notable approaches include incorporating Galilean invariance into the Reynolds stress tensor predictions using neural networks (NN) [16,17], learning source terms from coarsely resolved quantities through NNs [18], and discovery of closed-form equations for coarse-grained ocean models using high-resolution data [19]. Similarly, the study [20] developed a field inversion and ML modeling framework to learn corrections to the Spalart–Allmaras RANS model [21]. To address the issue of compounding modeling errors during inference, ideas based on multi-agent reinforcement learning were used to locally adapt the coefficients of the eddy-viscosity closure model, with the objective to reproduce the energy spectrum predicted by DNS. In contrast to supervised learning, training in this case was not performed on a database of reference data, but through time integration of the parametric model and penalizing unstable models. While the latter approach provides stable *a-posteriori* predictions, its exploratory nature makes it computationally expensive. In this context, integrating neural network models with differential numerical solvers has enhanced the temporal stability of inferred closures, yielding significant improvements in long-term *a posteriori* statistics for a range of unsteady simulations [22–25], while maintaining computational efficiency.

By drawing parallels between the closure problem arising in active fluids and classical closure modeling in fluid dynamics, we successfully adapt ideas from ML-based closure modeling to develop data-driven closures for active matter systems. To begin, we generate high-fidelity data through kinetic theory, which serves as reference data for learning empirical closures models or comparing against. This is similar to the use of DNS data as ground truth in developing data-driven closures in fluid dynamics. Next, we identify the rotational symmetries arising in closure terms of the coarse-grained description of dense active nematics and leverage a second-order tensor-valued isotropic representation to learn an accurate and stable approximation of the closure terms. To overcome the distinction between *a priori* and *a posteriori* evaluation, we propose a strategy based on discrete time-stepping constraints that embed temporal stability into the learned closures.

The main contributions of this work are as follows. We focus on the coarse-grained description of the active apolar suspension derived from tracking only the zeroth (concentration) and second (orientation tensor) moment of the particle distribution function. By recognizing the isotropic nature of the closure terms, we utilize isotropic second-order tensor representation to model them. The proposed learning formulation incorporates invariant input features, enabling a rotationally invariant representation of the learned closures, resulting in improved accuracy and data efficiency compared to approaches without encoded symmetry. We provide a quantitative comparison between different neural architectures with varying receptive fields and mathematical representation, and also compare the data-driven approximations with commonly used closure models. To ensure stable *a-posteriori* prediction, we propose an optimization formulation based on model predictive control (MPC) with time-stepping constraints, ensuring temporal stability of the learned closures. In the last section, we assess the accuracy of the closure models for solving inverse problems, specifically inferring parameters of coarse-grained descriptions using data from kinetic theory.

2. Active suspension model

We are interested in studying the dynamics of N active and immersed particles of length ℓ and thickness b (aspect ratio $= \ell/b \gg 1$) in a volume V assumed to be a cube of linear dimension $L = |V|^{1/3}$ [10,14]. The dynamics of a concentrated suspension of such particles is modeled using kinetic theory, wherein the particle configuration is described by means of a continuum distribution function $\Psi(\mathbf{x}, \mathbf{p}, t)$ of the center of mass \mathbf{x} and orientation vector \mathbf{p} at time t . The zeroth, first, and second moments of the distribution function with respect to \mathbf{p} correspond to the concentration field $c = \langle 1 \rangle$, polarity field $\mathbf{n} = \langle \mathbf{p} \rangle / c$, and nematic order parameter $\mathbf{Q} = \langle \mathbf{pp} \rangle / c$, respectively, where $\langle f \rangle = \int_{|\mathbf{p}|=1} f \Psi d\mathbf{p}$ denotes an orientational moment. The evolution of the distribution function is governed by the Smoluchowski equation, reflecting the conservation of particle number, and is given by

$$\frac{\partial \Psi}{\partial t} + \nabla_{\mathbf{x}} \cdot (\dot{\mathbf{x}} \Psi) + \nabla_{\mathbf{p}} \cdot (\dot{\mathbf{p}} \Psi) = 0, \quad (1)$$

where the conformational fluxes $\dot{\mathbf{x}}$ and $\dot{\mathbf{p}}$ are obtained from the dynamics of a single particle in a background flow $\mathbf{u}(\mathbf{x}, t)$. The operator $\nabla_{\mathbf{p}} = (\mathbf{I} - \mathbf{pp}) \cdot (\partial/\partial \mathbf{p})$ denotes the gradient operator on the unit sphere of orientations. For a concentrated suspension, the conformational fluxes are

$$\dot{\mathbf{x}} = \mathbf{u} - d_T \nabla_{\mathbf{x}} \log \Psi, \quad (2)$$

$$\dot{\mathbf{p}} = (\mathbf{I} - \mathbf{pp}) \cdot (\nabla \mathbf{u} + 2\zeta \mathbf{D}) \cdot \mathbf{p} - d_R \nabla_{\mathbf{p}} \log \Psi. \quad (3)$$

Here d_T and d_R are dimensionless translational and rotational diffusion constants, ζ is the strength of particle alignment through steric interactions, and $\mathbf{D} = \langle \mathbf{pp} \rangle$ is the second-moment tensor. Equation (2) describes the particles being advected by the local mean-field velocity \mathbf{u} and diffusing isotropically. Similarly, in Eq. (3), the angular flux velocity of the particle is dictated by the rotation of a slender rodlike particle by the mean-field quantity $\nabla \mathbf{u} + 2\zeta \mathbf{D}$, as well as rotational diffusion. The Smoluchowski equation is coupled to the Stokes flow as

$$-\Delta \mathbf{u} + \nabla P = \nabla \cdot \boldsymbol{\Sigma}, \quad \nabla \cdot \mathbf{u} = 0, \quad (4)$$

$$\boldsymbol{\Sigma} = \alpha \mathbf{D} + \beta \mathbf{S} : \mathbf{E} - 2\zeta \beta (\mathbf{D} \cdot \mathbf{D} - \mathbf{S} : \mathbf{D}). \quad (5)$$

Here $P(\mathbf{x}, t)$ is the fluid pressure, α is the dimensionless active dipole strength, β characterizes the particle density, $\mathbf{E} = [\nabla \mathbf{u} + \nabla \mathbf{u}^T]/2$ is the symmetric rate-of-strain tensor, and $\mathbf{S} = \langle \mathbf{pppp} \rangle$ is the fourth-moment tensor. The stress tensor $\boldsymbol{\Sigma}$ in Eq. (5) is written as the sum of three contributions arising from the active dipole, particle rigidity, and stress arising from local steric torques, respectively. The system of equations (1)-(4) forms a closed system which is referred to as the kinetic theory.

To circumvent the computational complexity arising from the high dimensionality of the kinetic model, we evolve only certain moments of the distribution function Ψ . Specifically, for apolar suspensions, such as the one considered here, we evolve only the zeroth and second moments c and \mathbf{D} . By taking moments of the Smoluchowski equation (1), we can derive:

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = d_T \Delta c, \quad (6)$$

$$\mathbf{D}^\nabla + 2\mathbf{S} : \mathbf{E} = 4\zeta (\mathbf{D} \cdot \mathbf{D} - \mathbf{S} : \mathbf{D}) + d_T \Delta \mathbf{D} - 2dd_R \left(\mathbf{D} - \frac{c}{d} \mathbf{I} \right). \quad (7)$$

Here $\mathbf{D}^\nabla = \partial \mathbf{D} / \partial t + \mathbf{u} \cdot \nabla \mathbf{D} - (\nabla \mathbf{u} \cdot \mathbf{D} + \mathbf{D} \cdot \nabla \mathbf{u}^T)$ is the upper-convected time derivative and d is the spatial dimension. Equations (6)-(7) involve the higher fourth-moment of the distribution function \mathbf{S} . Hence, Eqs. (4)-(7) are not closed and one needs to employ a *closure model* to approximate terms such as $\mathbf{S} : \mathbf{D}$ and $\mathbf{S} : \mathbf{E}$ in terms of c and \mathbf{D} . We also observe that Eqs. (4) and (7) depend only on the symmetric second-order tensor $\mathbf{S} : \mathbf{T}$ with $\mathbf{T} := \mathbf{E} + 2\zeta \mathbf{D}$. In our study, we concern ourselves with the representation and learning of this tensorial quantity $\mathbf{S} : \mathbf{T}$ from kinetic simulations.

3. Closure representation and the learning problem

The term $\mathbf{S} : \mathbf{T}$ described in the previous section has the following rotational symmetry

$$\mathbf{S}' : \mathbf{T}' = \boldsymbol{\Omega} (\mathbf{S} : \mathbf{T}) \boldsymbol{\Omega}^T,$$

where $\mathbf{T}' = \boldsymbol{\Omega} \mathbf{T} \boldsymbol{\Omega}^T$. To put it simply, if we rotate the tensor arguments $(\mathbf{S}, \mathbf{D}, \mathbf{E})$ of the contraction, the resulting tensor contraction $\mathbf{S} : \mathbf{T}$ should also be rotated by the same angle. Since $\mathbf{S} : \mathbf{T}$ is computed through a linear combination of $\mathbf{S} : \mathbf{D}$ and $\mathbf{S} : \mathbf{E}$, as a consequence, both $\mathbf{S} : \mathbf{D}$ and $\mathbf{S} : \mathbf{E}$ tensors also adhere to the aforementioned relation. Any tensor-valued function that obeys the above rotational equivariance is termed an isotropic tensor-valued function. More formally, a second-order tensor valued function $\mathbf{F}(\mathbf{A}_1, \dots, \mathbf{A}_M)$ is isotropic if

$$\mathbf{F}(\boldsymbol{\Omega} \mathbf{A}_1 \boldsymbol{\Omega}^T, \dots, \boldsymbol{\Omega} \mathbf{A}_M \boldsymbol{\Omega}^T) = \boldsymbol{\Omega} \mathbf{F}(\mathbf{A}_1, \dots, \mathbf{A}_M) \boldsymbol{\Omega}^T, \quad \forall \boldsymbol{\Omega} \in \text{O}(d) \quad (8)$$

where $\{\mathbf{A}_i\}_{i=1,2,\dots,M}$ are symmetric tensors and $\text{O}(d)$ is the full orthogonal group in d -dimensions. In general, a symmetric tensor-valued isotropic function $\mathbf{F}(\mathbf{A}_1, \dots, \mathbf{A}_M)$ can be expanded as

$$\mathbf{F}(\mathbf{A}_1, \dots, \mathbf{A}_M) = \sum_{m=1}^M \varphi_m(\mathcal{I}_s) \mathbf{F}_m, \quad (9)$$

where \mathbf{F}_m are form-invariant symmetric tensor-valued isotropic functions, referred to as generators, and φ_m are scalar-valued isotropic functions of the invariants \mathcal{I}_s of the functional basis of $\{\mathbf{A}_i\}_{i=1, \dots, M}$ [26]. By representing the tensor-valued function using a set of fundamental tensors, known as an integrity basis, we can ensure tensor isotropy, as described in Eq. (8). Such a representation has been used in classical fluid turbulence closures to model the Reynolds stress anisotropy tensor, where the anisotropic tensor is expanded in the integrity basis of the strain rate tensor and the mean vorticity tensor [27,16]. For the specific case of dense active apolar suspensions, we are interested in the representation of the tensor-valued function $\mathbf{F}(\mathbf{S}, \mathbf{D}, \mathbf{E}) = \mathbf{S} : \mathbf{E} + 2\zeta \mathbf{S} : \mathbf{D}$. Under the assumption of uniform concentration $c \equiv 1$ and that the fourth-order tensor \mathbf{S} depends only on the known lower order moment \mathbf{D} , the quantity $\mathbf{S} : \mathbf{D}$ is a tensor-valued function of \mathbf{D} only, and $\mathbf{S} : \mathbf{E}$ is a tensor-valued function with tensor arguments (\mathbf{D}, \mathbf{E}) .

Let us first consider the representation of the tensor quantity $\mathbf{S} : \mathbf{D}$ in two dimensions. The independent invariant quantities associated with the orientation tensor \mathbf{D} are $\mathcal{I}_s(\mathbf{D}) = \{\text{tr}\mathbf{D}, \text{tr}\mathbf{D}^2\}$ and the corresponding tensor/generator basis is $\mathbf{F}_m = \{\mathbf{I}, \mathbf{D}\}$. Therefore, under the isotropic representation described in equation (9), we can write

$$\mathbf{S} : \mathbf{D} = \gamma_0 \mathbf{I} + \gamma_1 \mathbf{D}, \quad (10)$$

where γ_0, γ_1 are scalar-valued isotropic functions of the invariants $\mathcal{I}_s(\mathbf{D})$. In a similar fashion, the tensor quantity $\mathbf{S} : \mathbf{E}$ can be written in the expanded form,

$$\mathbf{S} : \mathbf{E} = \kappa_0 \mathbf{I} + \kappa_1 \mathbf{D} + \kappa_2 \mathbf{E}, \quad (11)$$

where $\{\kappa_i\}_{i=0,1,2}$ are scalar isotropic functions of the invariants $\mathcal{I}_s(\mathbf{D}, \mathbf{E}) = \{\text{tr}\mathbf{D}, \text{tr}\mathbf{E}, \text{tr}\mathbf{D}^2, \text{tr}\mathbf{E}^2, \text{tr}\mathbf{D}\mathbf{E}\}$. On the account of the incompressibility condition $\nabla \cdot \mathbf{u} = 0$ and the assumption of uniform concentration, we can safely omit the invariants $\text{tr}\mathbf{D}$ and $\text{tr}\mathbf{E}$ from the set $\mathcal{I}_s(\mathbf{D}, \mathbf{E})$. The detailed derivation of the representation is discussed in Appendix A. Combining equations (10) and (11) the closure tensor quantity $\mathbf{S} : \mathbf{T}$ can be written in the expanded form as,

$$\mathbf{F}(\mathbf{D}, \mathbf{E}) = \mathbf{S} : \mathbf{T} = (\kappa_0 + 2\zeta\gamma_0) \mathbf{I} + (\kappa_1 + 2\zeta\gamma_1) \mathbf{D} + \kappa_2 \mathbf{E}. \quad (12)$$

It is straightforward to show that the above representation satisfies the tensor isotropic relation discussed in equation (8), i.e.

$$\begin{aligned} \mathbf{F}(\mathbf{\Omega D \Omega}^\top, \mathbf{\Omega E \Omega}^\top) &= (\kappa_0 + 2\zeta\gamma_0) \mathbf{I} + (\kappa_1 + 2\zeta\gamma_1) \mathbf{\Omega D \Omega}^\top + \kappa_2 \mathbf{\Omega E \Omega}^\top \\ &= \mathbf{\Omega} \left((\kappa_0 + 2\zeta\gamma_0) \mathbf{I} + (\kappa_1 + 2\zeta\gamma_1) \mathbf{D} + \kappa_2 \mathbf{E} \right) \mathbf{\Omega}^\top = \mathbf{\Omega F(D, E) \Omega}^\top. \end{aligned}$$

The isotropic representation provides a relation on how the tensor contraction $\mathbf{S} : \mathbf{T}$ rotates if its input arguments are transformed by a rotation matrix $\mathbf{\Omega} \in \text{O}(2)$. More importantly, the relation (12) provides a strategy for learning closure representations that by construction satisfy properties of tensor-valued isotropic functions. In the next subsection, we discuss how a learning problem can be formulated based on the isotropic representation, where we try to learn the nonlinear dependency of the coefficients $(\gamma_0, \gamma_1, \kappa_0, \kappa_1, \kappa_2)$ on the invariant quantities $\mathcal{I}_s(\mathbf{D}, \mathbf{E})$.

3.1. Optimization formulation

We use neural networks to learn the nonlinear dependency between the scalar-valued isotropic function coefficients (γ, κ) and the invariant quantities $\mathcal{I}_s(\mathbf{D}, \mathbf{E})$. Thus inferred coefficients are used to reconstruct the closure terms using the expansion described in Eq. (12). To achieve this, we leverage the universal function approximation properties of neural networks within a supervised learning framework. In this setting, the network parameters (θ) are optimized to learn a nonlinear map between an input vector and the predefined target vector (output). The closure learning problem within a supervised setting can be written as follows,

$$\tilde{\theta} = \underset{\theta}{\text{argmin}} \int_{\Omega_d} \left(\left\| \mathbf{S} : \mathbf{T}(\mathbf{x}) - (\kappa_0(\theta) + 2\gamma_0(\theta)) \mathbf{I} - (\kappa_1(\theta) + 2\gamma_1(\theta)) \mathbf{D}(\mathbf{x}) - \kappa_2(\theta) \mathbf{E}(\mathbf{x}) \right\|_F^2 \right) d\mathbf{x},$$

where the neural network, $\mathcal{N}_\theta : \{\text{tr}\mathbf{D}^2, \text{tr}\mathbf{E}^2, \text{tr}\mathbf{D}\mathbf{E}\} \rightarrow \{\gamma_0, \gamma_1, \kappa_0, \kappa_1, \kappa_2\}$ with parameters θ , is optimized to learn the map between the invariant feature space and the coefficients of the isotropic representation in Eq. (12), and Ω_d is the spatial domain of dimension d . The reference data for the tensor quantity $\mathbf{S} : \mathbf{T}$ is generated from numerically solving the kinetic theory described in Eqs. (1)-(4). As a baseline comparison with the isotropic representation, we learn a direct nonlinear map between the components of tensor function arguments (\mathbf{D}, \mathbf{E}) and the closure tensor $\mathbf{S} : \mathbf{T}$ by solving the optimization problem

$$\tilde{\theta} = \underset{\theta}{\text{argmin}} \int_{\Omega_d} \left(\left\| \mathbf{S} : \mathbf{T}(\mathbf{x}) - \widetilde{\mathbf{S} : \mathbf{T}}_\theta(\mathbf{x}) \right\|_F^2 \right) d\mathbf{x}.$$

Here, the neural network \mathcal{N}_θ learns the map $\mathcal{N}_\theta : \{\mathbf{D}, \mathbf{E}\} \rightarrow \{\widetilde{\mathbf{S} : \mathbf{T}}_\theta\}$. This straightforward nonlinear map between tensor components is not constrained to satisfy the isotropic function property of the tensor-valued function $\mathbf{S} : \mathbf{T}$. Hence, it provides an ideal baseline for evaluating the impact of rotational invariance on the accuracy and extrapolation properties of the inferred closure models. All of this

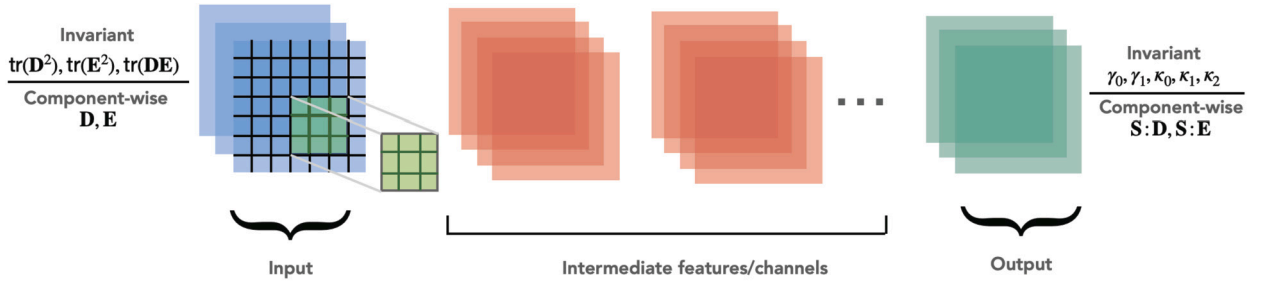


Fig. 1. Learning based on isotropic tensor representation v/s Component-wise mapping: For the isotropic representation of closure terms, the input to the neural network (CNN discussed here) are rotationally invariant scalar quantities of the orientation tensor \mathbf{D} and the strain-rate \mathbf{E} . The outputs are the coefficients $\{\gamma_0, \gamma_1, \kappa_0, \kappa_1, \kappa_2\}$ that are used to compose the closure terms through the relation $\mathbf{S}:\mathbf{T} = (\kappa_0 + 2\zeta\gamma_0)\mathbf{I} + (\kappa_1 + 2\zeta\gamma_1)\mathbf{D} + \kappa_2\mathbf{E}$. The input to the neural network is the tensor components of the orientation tensor \mathbf{D} and the strain-rate \mathbf{E} . The outputs are the tensor components of the tensors $\mathbf{S}:\mathbf{D}$ and $\mathbf{S}:\mathbf{E}$ which are then composed to approximate the closure terms, i.e. $\mathbf{S}:\mathbf{T} = \mathbf{S}:\mathbf{E} + 2\zeta\mathbf{S}:\mathbf{D}$. The gridded 3×3 box corresponds to the receptive field of the CNN.

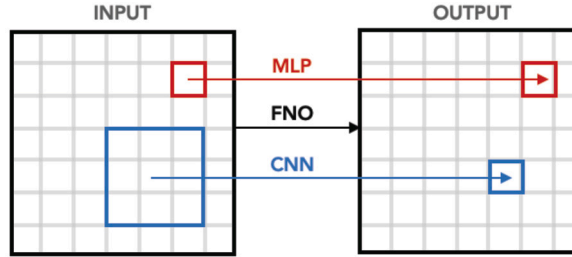


Fig. 2. Receptive field and response field of different neural architectures: The MLP architecture learns a map between the input attributes at a point in the spatial domain to output attributes at the same point. CNN architectures rely on local patch-wise convolutions to map a receptive field in the input domain to a point in the output domain. Fourier neural operators (FNO) rely on convolutions in Fourier space (global convolutions) to learn the mapping, resulting in a receptive field that spans the entire spatial domain. In the context of the closure problem, the input channel corresponds to a spatial grid containing the tensor components or its invariants.

relies on the approximation capabilities of neural networks \mathcal{N}_θ . In the next section we provide a brief description of neural networks and highlight different network architectures with varying receptive fields and internal mathematical representations (Fig. 1).

3.2. Neural network architectures

In general, a Neural Network (NN) is a multivariate compound function which contains a number of free parameters, called weights and biases, that can be learned. It maps an input vector to an output vector via successive linear (matrix multiplication) and non-linear operations. The set of real-valued n -layer neural network with input \mathbf{z} can be written as a function,

$$\mathcal{N}_\theta := \{f : \mathbb{R}^{\text{inp}} \rightarrow \mathbb{R}^{\text{out}}\},$$

$$f(\mathbf{z}) = \mathcal{T}_n \left(\rho \left(\mathcal{T}_{n-1} \left(\rho \left(\dots \mathcal{T}_1 \left(\rho \left(\mathcal{T}_0(\mathbf{z}) \right) \right) \dots \right) \right) \right) \right), \quad (13)$$

where $\rho : \mathbb{R} \rightarrow \mathbb{R}$ is an element-wise nonlinear activation function and $\mathcal{T}_k(\mathbf{z}) = \mathbf{W}_k \mathbf{z} + \mathbf{b}_k$ is the affine linear map with trainable parameters (θ) in the form of weights and biases, i.e. $\theta = \{\mathbf{W}_k, \mathbf{b}_k\}_{k=0,1,\dots,n}$. The number of hidden layers (all layers excluding the input and output layer) determines the depth of the neural network, and the number of nodes per layer are important hyperparameters that determine the expressibility and training requirements of the network. The way nodes are connected and information is passed along is encoded in the matrices \mathbf{W}_k . For instance, in a fully connected neural network (MLP) the matrices \mathbf{W}_k are dense and in Convolutional Neural Networks (CNN) the matrices are usually sparse and circulant [28].

In CNN architectures, the input attributes in the receptive field are mapped to output quantities through convolutions based on learnable filters. Such patch-wise learning with filters is used for automatic feature extraction and thus CNN architecture finds extensive applications in image recognition and object detection [29,30]. Though CNNs are usually used for feature extraction, in this study we use them to learn the nonlinear map associated with the closure. At the level of mathematical operations, CNNs are closely related to MLPs, except the global matrix multiplication is replaced by a local, multi-dimensional convolution filtering operator. The affine linear transformation in the k^{th} layer of CNN is,

$$\mathcal{T}_k(\mathbf{z}) = \mathbf{W}_k \mathbf{z} + \mathbf{b}_k, \quad (14)$$

where \mathbf{W}_k is a sparse circulant matrix that represents the convolution and depends on the size of the filter, and \mathbf{b}_k is the bias term. For a given feature, the same local convolution filter is applied to the whole input field to extract hierarchical features. Similarly, if we replace or append the local convolutions with global convolutions through spatial Fourier transforms, we arrive at Fourier Neural Operators (FNO) [31]. The affine linear transformation in the k^{th} layer of FNO is,

$$\mathcal{T}_k(\mathbf{z}) = \mathbf{W}_k \mathbf{z} + \mathbf{b}_k + \mathcal{F}^{-1}(\mathbf{R}_k \mathcal{F}(\mathbf{z})), \quad (15)$$

where $\theta = \{\mathbf{W}_k, \mathbf{R}_k, \mathbf{b}_k\}$ are the trainable weights and biases. The operators \mathcal{F} and \mathcal{F}^{-1} are the forward and inverse discrete Fourier transforms, respectively. FNOs have been used to learn PDE solution operators wherein non-local effects are captured through global convolutions [31,32]. The single layer operations of CNN and FNO are then used to compose the multivariate compound functions as shown in equation (13).

One important distinction between different neural architectures is how the input features are processed as shown in Fig. 2. In MLP, the input features are usually transformed to vectors before feeding into the neural network whereas in CNN and FNO the original multidimensional structure of the input features is preserved. Also, MLPs learn a point-wise map between input and target features, however in CNN the receptive field is a local multidimensional neighborhood around the point where the output prediction is desired. As depicted in Fig. 2, the receptive field in FNOs encompasses the entire spatial domain as it relies on global Fourier convolutions. The MLP neural architecture is agnostic to the grid resolution and domain-size as it involves point-wise mapping between input and output quantities.

4. Numerical results

In this section, we conduct *a-priori* and *a-posteriori* analyses of the closure approximations discussed in the previous sections. To generate the reference data from kinetic theory, we use a pseudo-spectral discretization of Eqs. (1)-(4) with the 2/3 anti-aliasing rule where Fourier differentiation is used to evaluate the derivatives with respect to space and particle orientation. We use a second-order implicit-explicit backward differentiation time-stepping scheme (SBDF2), where the linear terms are handled implicitly and the nonlinear terms explicitly with time-step $\Delta t = 0.0004$. The numerical simulations are performed in a periodic square domain of length $L = 10$ with 256^2 spatial modes and 256 orientational modes. We initialize \mathbf{D} with a plane-wave perturbation about the isotropic state $\mathbf{D}_0 = \mathbf{I}/2$ such that $\text{tr}\mathbf{D} = 1$ and take the concentration to be uniform so that $c(\mathbf{x}, t) \equiv 1$. In concentrated suspensions, the isotropic-to-nematic transition occurs when the dimensionless parameter ζ is increased [10]. In other words, in dense suspensions, steric interactions between particles lead to nematic ordering. For this reason, we primarily choose to probe the effects of closure approximations as a function of the alignment strength ζ , suppressing dependencies on α, β . The dipole strength was set to $\alpha = -1$ and the particle density $\beta = 0.8$, and the rotational and translation diffusion coefficients to $d_T = d_R = 0.05$.

For a chosen value of alignment strength ζ , the reference data generated from kinetic theory consists of 80 simulation snapshots separated by $\delta t = 0.25$. Each snapshot consists of information of both the tensor quantities \mathbf{D} and \mathbf{E} alongside the value of the closure tensor \mathbf{S} : \mathbf{T} recorded on a 256×256 spatial grid \mathcal{C} discretizing the domain $L \times L$. Specifically,

$$\text{Reference data} = \{\mathbf{D}_{\mathcal{K}}(\mathbf{x}, t_i; \zeta), \mathbf{E}_{\mathcal{K}}(\mathbf{x}, t_i; \zeta), \mathbf{S} : \mathbf{T}_{\mathcal{K}}(\mathbf{x}, t_i; \zeta)\}_{i=1,2,\dots,80}, \quad \mathbf{x} \in \mathcal{C},$$

and the subscript \mathcal{K} denotes data generated from kinetic theory. For the supervised problem discussed in Section 3.1, we use the first ~ 10 time ordered snapshots of the reference data as our training data. With the isotropic tensor representation, the invariant features $\{\text{tr}\mathbf{D}^2, \text{tr}\mathbf{E}^2, \text{tr}\mathbf{DE}\}$ are pre-computed before feeding into the neural network. All the hyperparameters of the neural architectures are detailed in Table 1. The neural networks are trained using Adam optimizer [33] with an initial learning rate $\eta = 0.001$.

4.1. A priori analysis of the learned closures

In this section, we perform numerical experiments to compare the prediction accuracy of data-driven closure models based on an isotropic representation and component-wise learning. For the learning based on tensor-valued isotropic function representation, we use three input features based on the invariant quantities $\mathcal{I}_s(\mathbf{D}, \mathbf{E})$ as described in Section 3. For the component-wise learning, we provide the independent components $\{d_{11}, d_{12}, d_{22}, e_{11}, e_{12}, e_{22}\}$ of each tensor \mathbf{D}, \mathbf{E} as the input to the neural network. For each value of the alignment strength parameter ζ , we generated reference data which was then used to train a neural network and make predictions. We use relative mean-square error (RMSE) as the performance metric:

$$\text{RMSE}(\zeta) = \frac{\|\mathbf{S} : \mathbf{T}_{\mathcal{K}}(\zeta) - \widetilde{\mathbf{S}} : \widetilde{\mathbf{T}}(\zeta, \theta)\|_F^2}{\|\mathbf{S} : \mathbf{T}_{\mathcal{K}}(\zeta)\|_F^2}, \quad (16)$$

where $\mathbf{S} : \mathbf{T}_{\mathcal{K}}(\zeta)$ is reference data from kinetic theory generated with parameter ζ and $\widetilde{\mathbf{S}} : \widetilde{\mathbf{T}}$ is the prediction of the closure using neural networks. The RMSE value is computed on the reference data which also includes the training set. From Fig. 3, we report that, irrespective of the network architecture, learning based on invariant representation leads to better performance in terms of prediction accuracy compared to component-wise learning. Invariant representation takes into consideration rotational symmetries during the learning process, which contributes to its superior performance. Interestingly, for a comparable number of network parameters and the same training data, going from point-wise learning (MLP) to learning based on global convolutions (FNO), the accuracy of the network predictions deteriorates. Especially, we see that FNO struggles to approximate the closure with component-wise learning and gains two orders of magnitude increase in prediction accuracy with the invariant-based representation for varying alignment strength ζ . Thus, a representation based on tensor-valued isotropic function allows for learning accurate closures from a few time snapshots of kinetic theory simulations. Interestingly, from Fig. 3 it is also evident that learning the point-wise nonlinear map using MLP yields the best *a-priori* predictions. The MLP achieves this performance without needing any additional local neighborhood information like in CNN or accounting for nonlocal effects through global convolutions as in the FNO architecture.

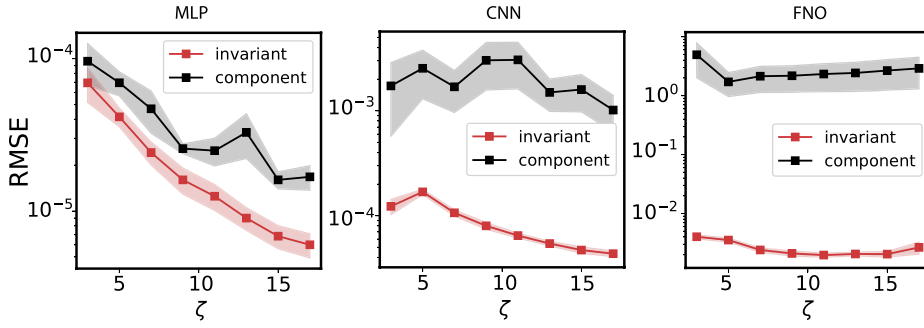


Fig. 3. Comparison between invariant based learning vs component-wise learning: The Relative Mean Square Error (RMSE) for different network architectures MLP (left), CNN (middle), FNO (right). The shaded area around the line depicts one standard deviation from the mean. The neural networks have a comparable number of parameters ~ 10000 , and architectures as shown in Table 1.

4.2. Comparison with commonly used closures

Several closures have been proposed in the context of the theory of passive and active liquid crystal polymers. In this study, we specifically compare our data-driven closure against the following established closure models:

Linear closure: One of the simplest closures is derived from truncating the particle distribution function Ψ on the basis of spherical harmonics and ignoring the coefficients of harmonics for degree greater than or equal to three [3]. In two dimensions, the resulting linear closure can be expressed using the largest eigenvalue (μ_1) of the normalized orientational order parameter $\mathbf{Q} = \mathbf{D}/c$ as,

$$S'_{1111} \approx (3/8 + (\mu_1 - 1/2))c, \quad (17)$$

where $S'_{ijkl} = \Omega_{im}\Omega_{jn}\Omega_{kp}\Omega_{lq}S_{mnpq}$ is the rotated fourth-moment tensor in the diagonal frame of \mathbf{D} , i.e. $\mathbf{D} = \mathbf{Q}\mathbf{D}'\mathbf{Q}^T$. Here, \mathbf{Q} is an orthonormal matrix and $\mathbf{D}' = \text{diag}\{\mu_i\}_{i=1}^d$ is a diagonal matrix consisting of the ordered eigenvalues of \mathbf{D} . The full tensor S_{ijkl} can be computed from S'_{1111} using the trace identities $S'_{1111} + S'_{1122} = \mu_1$ and $S'_{1122} + S'_{2222} = c(1 - \mu_1)$ and rotations.

Quadratic closure: This *ad hoc* closure proposed by Doi [34] approximates the fourth moment \mathbf{S} as the quadratic dependence of the second moment \mathbf{D} , i.e. $S'_{1111} \approx \mu_1^2$. Despite the *ad hoc* nature of the quadratic closure, it is very commonly used in theories for passive and active liquid crystals [35,36].

Bingham closure: Originally introduced by Chaubal and Leal [37] in the context of liquid crystal polymers, the Bingham closure models the particle distribution function as

$$\mathbf{S}_B[\mathbf{D}] = Z^{-1} \int_{|\mathbf{p}|=1} \mathbf{p}\mathbf{p}\mathbf{p}\mathbf{p} e^{\mathbf{B}[\mathbf{D}]:\mathbf{p}\mathbf{p}} d\mathbf{p}. \quad (18)$$

This closure relies on solving the inverse problem of determining the parameters \mathbf{B} and Z such that following moment constraints are satisfied at each point in space,

$$c(\mathbf{x}, t) = \frac{1}{Z(\mathbf{x}, t)} \int_{|\mathbf{p}|=1} e^{\mathbf{B}(\mathbf{x}, t):\mathbf{p}\mathbf{p}} d\mathbf{p}; \quad \mathbf{D}(\mathbf{x}, t) = \frac{1}{Z(\mathbf{x}, t)} \int_{|\mathbf{p}|=1} \mathbf{p}\mathbf{p} e^{\mathbf{B}(\mathbf{x}, t):\mathbf{p}\mathbf{p}} d\mathbf{p}.$$

The recent work [13] proposed an efficient method for evaluating the Bingham closure using fast and accurate Chebyshev representation to reconstruct the fourth-moment tensor to near machine precision, with accuracy and efficiency that can be finely controlled. By solving the above inverse problem over the feasible range of $\mu_1 \in [1/2, 1]$, this provides the following closure representation,

$$S'_{1111} \approx \sum_{m=0}^M c_m T_m(4\mu_1 - 3), \quad (19)$$

where $T_m(v)$ is the m^{th} Chebyshev polynomial with coefficient c_m . One can then calculate the closure $\mathbf{S}:\mathbf{T}$ by performing a contraction in the diagonal basis of \mathbf{D} and rotating back $\mathbf{S}:\mathbf{T} = \mathbf{Q} \cdot (\mathbf{S}':\mathbf{T}') \cdot \mathbf{Q}^T$.

In Fig. 4, we plot the RMSE for predictions based on the closures discussed above and data-driven closures discussed in the previous section. We note that neural networks based on point-wise (MLP) and patch-wise filter-based learning (CNN) with invariant features are able to reach prediction accuracy comparable to Bingham closure for different values of ζ (see Fig. 4). Also, the MLP and CNN architectures with invariant representation gain two orders of magnitude in prediction accuracy compared to the commonly used quadratic closure [34]. This is also evident in Fig. 5 where we show single time snapshots of the point-wise prediction error $|\mathbf{v} - \tilde{\mathbf{v}}|$

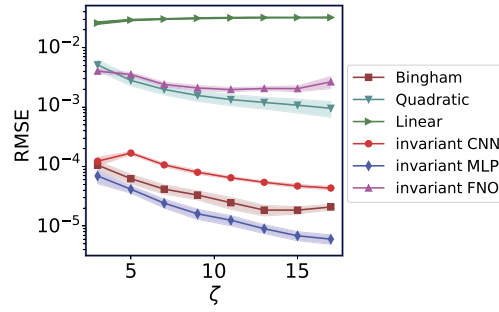


Fig. 4. Comparison of existing closures with data-driven approximation: We compare the RMSE of the linear, quadratic and Bingham closures to those based on neural networks for different values of the alignment parameter ζ . The CNN and MLP yield the highest accuracy and perform similarly to the Bingham closure, while the FNO is less accurate by two orders of magnitude.

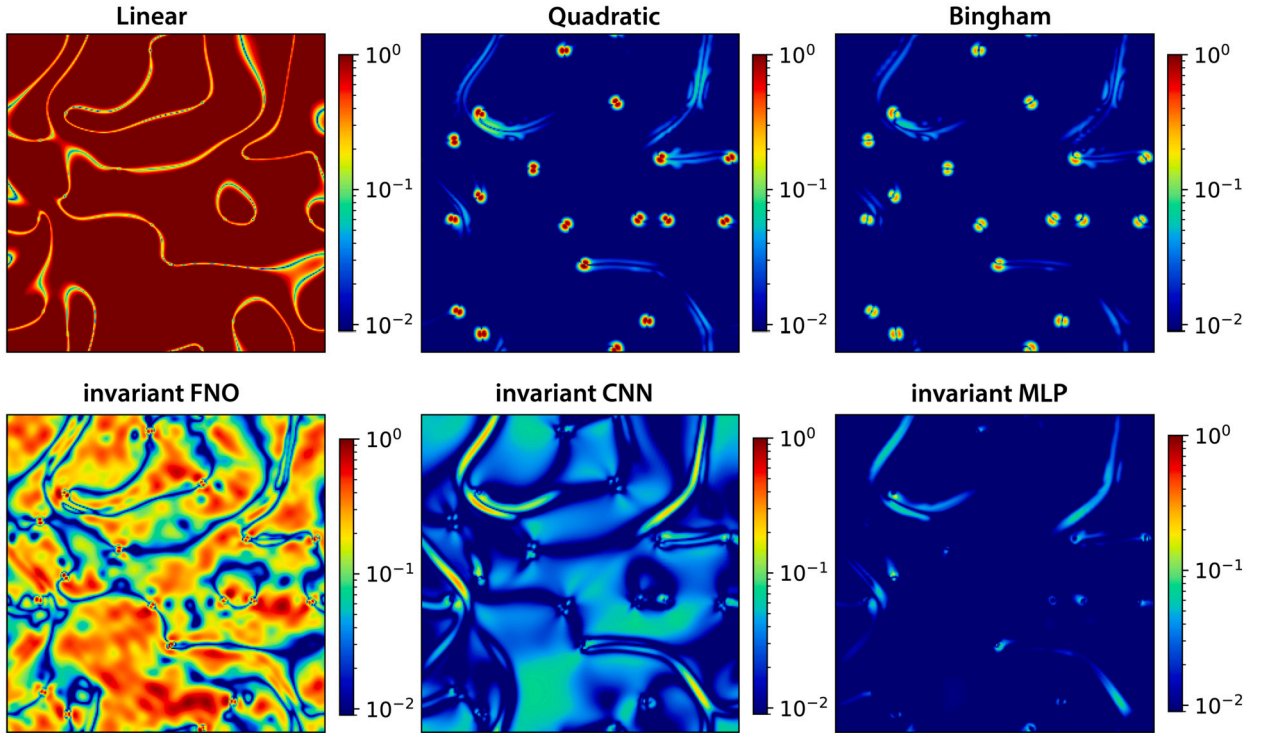


Fig. 5. Comparing point-wise absolute error: The point-wise error across the spatial domain is shown, representing the predictions of a component of the tensor quantity $\mathbf{S}:\mathbf{T}$ using different closures for $\zeta = 15$. Compared to other closures, the MLP produces the lowest error and yields accurate solutions near defects which other closures fail to capture.

using different closures for $\zeta = 13$. On closer observation in Fig. 5, we show the closure approximation using linear, quadratic and FNO closures struggle to adequately resolve regions close to defects. Given that these rich topological structures are central to the dynamics, it is important to adequately resolve the features associated with them for accurate characterization of the spatiotemporal dynamics.

In addition to comparing prediction accuracy, it is also valuable to understand how different closures vary based on their functional relationships with input arguments. However, the closure learned through neural network is a highly nonlinear multivariate function, and probing the explicit nonlinear dependency between the tensor arguments \mathbf{D}, \mathbf{E} and predicted closure $\mathbf{S}:\mathbf{T}$ is non-trivial. Nevertheless, we can use the closure predictions to plot the dependence between S'_{1111} and the largest eigenvalue μ_1 of the orientation tensor \mathbf{D} , as illustrated in Fig. 6. It can be seen that the map $\mu_1 \rightarrow S'_{1111}$, as learned through the neural network, particularly the MLP architecture, is very similar to the Bingham closure. Furthermore, the effect of varying network architectures on the learned dependency between tensor arguments and closure predictions can be clearly observed by contrasting the mappings obtained from the MLP architecture with those from the CNN, especially at low alignment interactions.

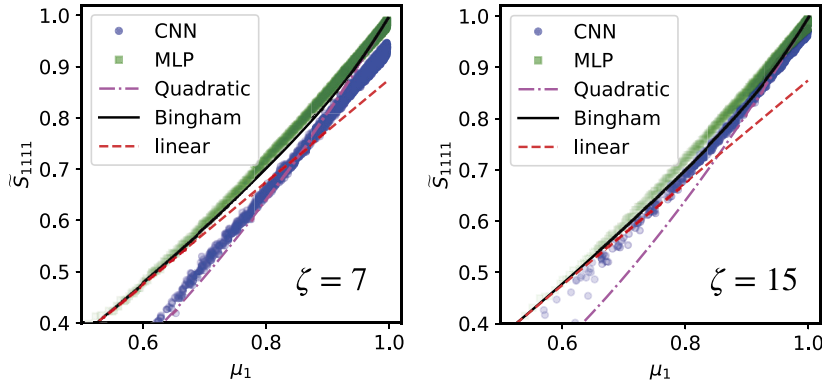


Fig. 6. Functional relationship between orientation tensor \mathbf{D} and the fourth moment tensor \mathbf{S} for different closures: The map $\mu \rightarrow S'_{1111}$ computed with different closure for alignment strengths $\zeta = 7$ (left) and $\zeta = 15$ (right).

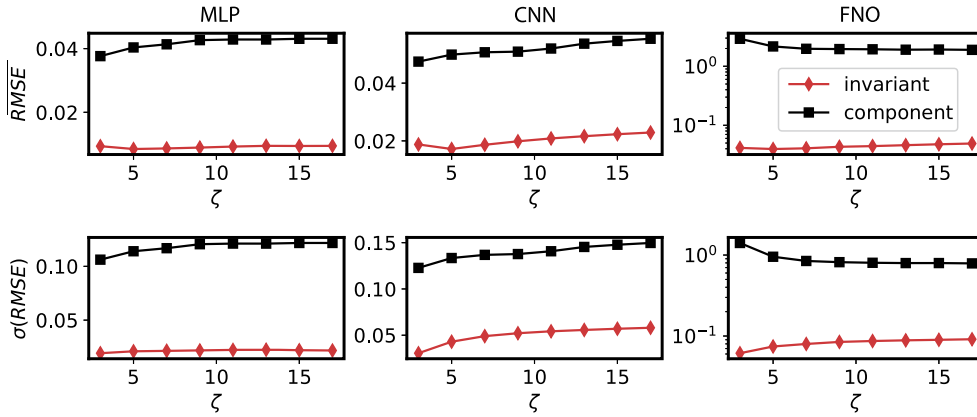


Fig. 7. Extrapolation properties of invariant based learning (red) vs. component-wise learning (black): For each value of the parameter ζ , we compare the mean (top row) and standard deviation (bottom row) of predictions made by neural networks trained using data generated from different values of ζ . (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Next, we test how different neural network architectures generalize to different parameter regimes. For this, we use a neural network trained with a specific value of the parameter ζ to make closure predictions at other values of the same parameter. To quantify the extrapolation accuracy, we use the following metrics:

$$\overline{\text{RMSE}}(\zeta) = \frac{1}{N_\zeta} \sum_j \text{RMSE}(\zeta, \theta_j), \quad \sigma(\text{RMSE}) = \frac{1}{N_\zeta} \sum_j \left(\text{RMSE}(\zeta, \theta_j) - \overline{\text{RMSE}}(\zeta) \right)^2.$$

Here, $\text{RMSE}(\zeta, \theta_j)$ is the relative mean square error associated with the prediction for parameter ζ using a neural network trained on the data generated with parameter ζ_j , and N_ζ is the number of ζ values screened for prediction. In Fig. 7, we show the mean $\overline{\text{RMSE}}$ and standard deviation $\sigma(\text{RMSE})$ associated with the predictions of neural networks trained at different values of $\zeta \in \{3, \dots, 17\}$. Once again, we consistently observe that regardless of the architecture employed, neural networks trained with invariant representation exhibit better prediction accuracy for extrapolating to a value of ζ that differs from the one used to generate the training data, in comparison to component-wise learning.

Overall, we have shown that embedding rotational equivariance through tensor-valued isotropic representation improves both the accuracy and extrapolations abilities of all neural architectures. However, comparing Figs. 3 and 7, we observe a significant drop in the prediction accuracy of neural networks. This is attributed to the dramatic change in characteristics of the fluid flow and particle orientation as the alignment strength is varied, and the over-fitting nature of neural networks. In the next section, we discuss an active learning procedure for training a neural network to predict and generalize well across a region of parameter space.

4.3. Active learning for exploring parameter space

In the previous section, we have shown performance metrics with predictions using neural networks trained at the same value of the parameter ζ . However, the goal is not to use a different neural network when the model parameters are varied, but to be able to train a single network to accurately predict the closure term for any value of the parameter in a predefined region of interest, using training data from as few samples in the parameter space as possible [38–40]. In order to achieve this, we employ an autonomous

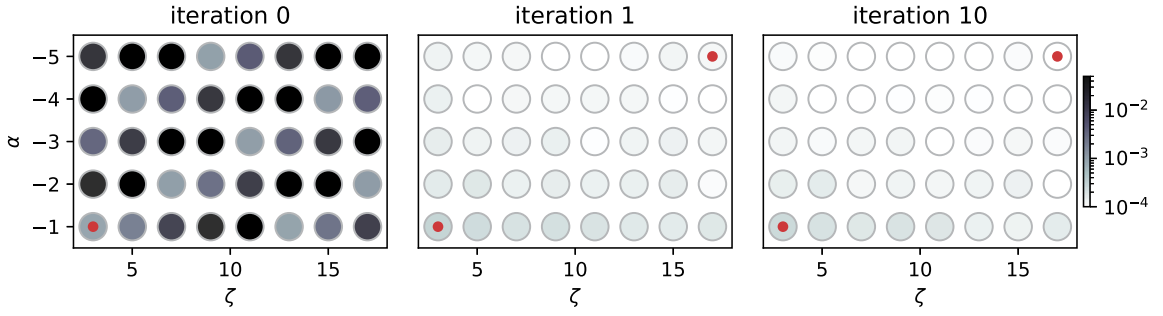


Fig. 8. Parameter sampling through active learning: Different iterations (columns) of the training procedure employing pool-based selective actives strategy are shown. The red dots indicate the locations where kinetic simulation data is solicited and included in the training set. The color coding of the grid points represents the magnitude of the RMSE associated with the network's prediction.

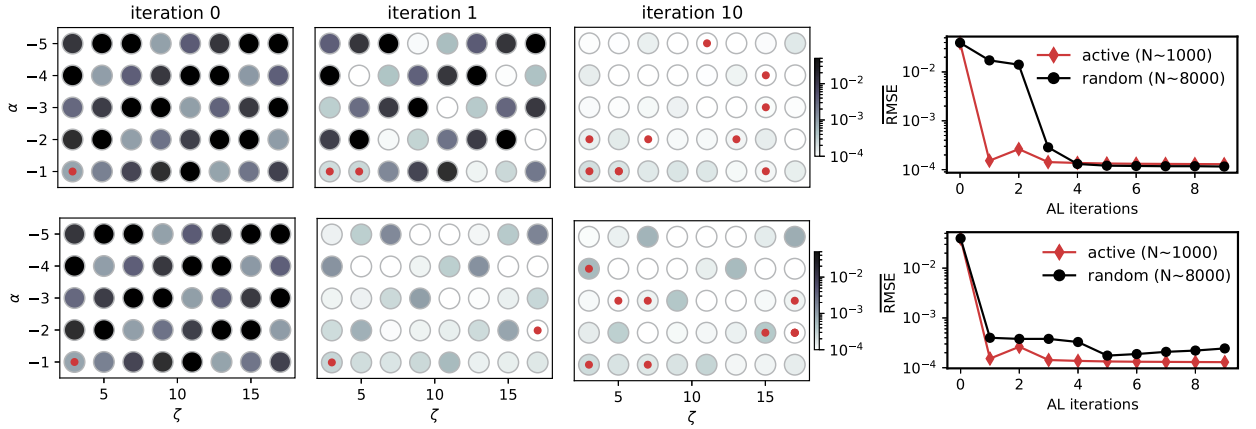


Fig. 9. Random sampling: Different iterations of the training procedure based on random parameter selection are shown for two different seeds. The red dots indicate the locations where kinetic simulation data is selected and included in the training set. The color coding of the grid points represents the magnitude of the RMSE associated with the network's prediction. The last column presents the comparison of the RMSE across the entire grid during the training process contrasting the AL and random sampling strategies.

network training procedure based on active learning. This strategy involves exploring and soliciting training data that maximally inform the learning process, thereby improving the network's ability to accurately predict the closure term for a wide range of parameter values.

In this section, using a pool-based selective sampling strategy, we efficiently explore the parameter space characterizing dipole strength α and alignment strength ζ [40]. First, we choose the region \mathcal{R} of interest in the parameter space over which we want to predict the closure terms. We discretize the region \mathcal{R} with the grid \mathcal{G} and generate kinetic simulation data at each grid point. Then, an autonomous active learning procedure is employed with the following steps: 1) Randomly choose a point on the grid and compile the training data. 2) Train the neural network on the available training data. 3) Use the trained network to make a prediction of the closure term $\widehat{\mathbf{S}} : \mathbf{T}$ at all points on the grid, regardless of where previous training data were generated. 4) Augment the training data with data generated from the point on the grid \mathcal{G} where the largest RMSE is observed and return to step 2.

In Fig. 8, we demonstrate the utility of the active learning procedure, where using just two points on the grid \mathcal{G} we achieve better predictive performance than random sampling as shown in Fig. 9. The RMSE comparison in Fig. 9 highlights the difference in performance between the active learning (AL) and random sampling strategies during the training process. As early as the second iteration of the AL procedure, the AL strategy achieves near-peak prediction performance showcasing its data efficiency and faster training process compared to random sampling. Also, the AL strategy in Fig. 8 demonstrates that selecting data points at the extrema of the parameter space (low and high) is sufficient to achieve accurate predictions across the entire parameter regime. This finding suggests that the flow and order parameters' characteristics in the intermediate parameter regime can be effectively captured by interpolating between the low and high values of the dipole and alignment strength. Unlike the active learning strategy, Fig. 9 demonstrates that random sampling does not consistently reduce prediction errors across the entire grid.

4.4. Learning stable *a-posteriori* estimates through model prediction control

In the previous sections, we demonstrated how neural networks trained with tensor isotropic representation can accurately approximate the closure terms. However, the supervised training used in the context of *a-priori* analysis cannot express the long-term

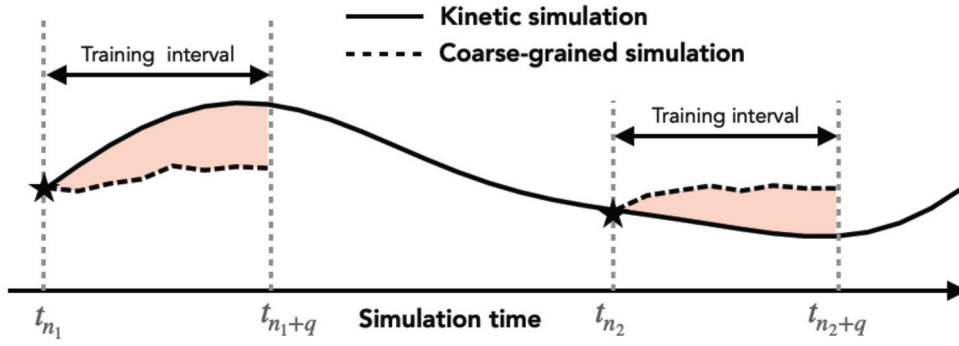


Fig. 10. Discrete Model predictive control through periodic injection: Initial conditions are provided to the pseudo-spectral solver at the injection points denoted by the set I^* and illustrated in the figure by the star symbol. Coarse-grained predictions are made within the training horizon $\{t_k, \dots, t_{k+q}\}$. The shaded region illustrates the discrepancy between the predictions and the kinetic theory that is being minimized through the optimization procedure described in Eqs. (20)-(21).

effects of the closure approximations and thus can lead to *a-posteriori* predictions that are not in agreement with kinetic theory [12]. In particular, the models learned could be over-fitted and numerically unstable for long-term temporal predictions. In this section, we demonstrate how neural networks trained using supervised learning on static reference data can lead to unstable *a-posteriori* predictions and training based on the deep integration of neural networks with a numerical solver is critical for resolving this issue.

We propose an optimization procedure based on model predictive control (MPC) that allows us to close the gap between the usual *a-priori* learning and a separate *a-posteriori* testing. In MPC, the closure approximation with the neural network is optimized during the training process such that the error between the predicted trajectory and the kinetic simulations over multiple time-steps is minimized (See Fig. 10). The method of model predictive control is akin to Neural ODEs [41–43], where unstable models are penalized using time-stepping constraints. Alternatively, we can interpret the closure approximations as *data-driven controls* that are learned to steer the system towards a desired target state set by kinetic simulations. The optimization formulation for model predictive is

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{k \in I^*} \sum_{l=1}^q \nu_l \left[\int_{\Omega_d} \left(\left\| \mathbf{D}_{\mathcal{K}}^{k,l}(\mathbf{x}) - \tilde{\mathbf{D}}^{k,l}(\mathbf{x}, \theta) \right\|_F^2 \right) d\mathbf{x} \right] + \lambda_{wd} \sum_{i=0}^{N_I} \|\mathbf{W}_i\|_F^2, \quad (20)$$

$$\text{where } \tilde{\mathbf{D}}^{k,l} = \mathcal{T}_{\mathbf{D}}^l \left(\tilde{\mathbf{D}}^{k,l-1}, \tilde{\mathbf{u}}^{k,l-1}, \mathcal{N}_{\theta} \left(\tilde{\mathbf{D}}^{k,l-1}, \tilde{\mathbf{E}}^{k,l-1} \right) \right), \quad \tilde{\mathbf{D}}^{k,0} = \mathbf{D}_{\mathcal{K}}^{k,0}. \quad (21)$$

Here, the positive integer q is referred to as the training time horizon, i.e. the number of time-integration steps considered during optimization, and θ represents the parameters of the neural networks that are being optimized. The scalars ν_l are exponentially decaying (over the training time horizon q) weights that account for the accumulating prediction error [43,42]. We optimize the trajectory over a small prediction window $q = 2, 3, 4$ as the kinetic simulations have finite Lyapunov exponents. The propagator $\mathcal{T}_{\mathbf{D}}$ is the function that updates the orientation tensor \mathbf{D} and the velocity \mathbf{u} , and encapsulates all the time-stepping formulae for \mathbf{D} and the Stokes operator used to update \mathbf{u} . The neural network $\mathcal{N}_{\theta}(\cdot)$ takes as input the invariant quantities $\{\operatorname{tr} \mathbf{D}^2, \operatorname{tr} \mathbf{E}^2, \operatorname{tr} \mathbf{DE}\}$ and outputs the closure approximation $\tilde{\mathbf{S}} : \tilde{\mathbf{T}}$ which is then passed through the propagator $\mathcal{T}_{\mathbf{D}}$ for prediction. The prediction from the spectral solver, $\tilde{\mathbf{D}}$, is then compared with the kinetic data, $\mathbf{D}_{\mathcal{K}}$. Due to the coupling between \mathbf{D} and \mathbf{u} through the Stokes equation, during optimization we compare errors in trajectories for only \mathbf{D} as described in equation (20). We also impose a penalty on the weights of the network, $\mathbf{W}_{i=1,2,\dots,N_I}$, in order to prevent over-fitting with the penalization strength controlled by the constant λ_{wd} . The overall MPC strategy is summarized in Algorithm 1. We performed the *a posteriori* analysis using the MLP architecture as it yielded the best prediction accuracy in *a-priori* analysis.

We use similar MLP architectures employed in the previous section for *a-priori* analysis. The neural network in the non-linear MPC is trained for 10000 epochs with an initial learning rate $\eta = 0.01$. The differentiable pseudo-spectral solver is integrated with time-step $\Delta t = 0.01$ with initial conditions provided from the kinetic simulations. For fast and efficient optimization, we make short window predictions with initial conditions prescribed at predefined injection points along the time axis as shown in Fig. 10. This also allows for natural batching of the data with each batch corresponding to an injection point and its corresponding prediction window. Given that during training we only optimize over short training intervals, we leverage automatic differentiation to efficiently compute the gradients of the learned model for gradient-based nonlinear optimization. Note for optimization and control over long-time windows the low-level auto-grad differentiation becomes computationally demanding and memory-intensive. In such scenarios, techniques like adjoint methods can be used to calculate derivatives efficiently, even for large simulations with thousands of control parameters (neural network parameters) [44].

In Fig. 11, we compare, kinetic simulations with the coarse-grained simulations based on mean statistics of the velocities and the scalar order parameter $s(t) = 2\langle \mu_1(\mathbf{x}, t) \rangle_{\mathbf{x}} - 1$. The angle brackets $\langle \cdot \rangle_{\mathbf{x}}$ denote averaging over the spatial domain. The predictions based on the Bingham closure and invariant MLP trained with the MPC procedure closely trace the trajectories of kinetic simulations, whereas predictions based on the quadratic closure and on the invariant MLP trained with supervised learning lead to unstable

Algorithm 1 Model Predictive Control.

Input: $\{\mathbf{D}_K^k \dots \mathbf{D}_K^{k+q}, \mathbf{E}_K^k \dots \mathbf{E}_K^{k+q}\}_{k \in I^*}$
for epoch $\in \{1, \dots, N_{\text{epoch}}\}$ **do**
 for $k \in I^*$ **do**
 $\tilde{\mathbf{D}}_K^{k,0} = \mathbf{D}_K^{k,0}, \quad \tilde{\mathbf{E}}_K^{k,0} = \mathbf{E}_K^{k,0}$ ▷ specify initial condition
 for $\ell \in \{1, \dots, q\}$ **do**
 $\{\kappa_0, \kappa_1, \kappa_2, \gamma_0, \gamma_1\} = \mathcal{N}_\theta(\tilde{\mathbf{D}}_K^{k,\ell-1}, \tilde{\mathbf{E}}_K^{k,\ell-1})$ ▷ Learning invariant coefficients
 $\tilde{\mathbf{S}} : \tilde{\mathbf{T}} = (\kappa_0 + 2\zeta\gamma_0) \mathbf{I} + (\kappa_1 + 2\zeta\gamma_1) \tilde{\mathbf{D}}_K^{k,\ell-1} + \kappa_2 \tilde{\mathbf{E}}_K^{k,\ell-1}$ ▷ Evaluate the closure
 $\tilde{\mathbf{D}}_K^{k,\ell}, \tilde{\mathbf{E}}_K^{k,\ell} = \mathcal{T}_D^l(\tilde{\mathbf{D}}_K^{k,\ell-1}, \tilde{\mathbf{E}}_K^{k,\ell-1}, \tilde{\mathbf{S}}; \tilde{\mathbf{T}})$ ▷ Solve Eq. (4) - Eq. (7)
 end for
 end for
 Evaluate Loss($\mathbf{D}_K, \tilde{\mathbf{D}}$) ▷ Compute loss using Eq. (20)
 Compute gradient $\nabla_\theta \text{Loss}$ ▷ via backpropagation
 $\theta^{\text{epoch}} = \theta^{\text{epoch}-1} - \eta \nabla_\theta \text{Loss}$ ▷ Update θ
end for

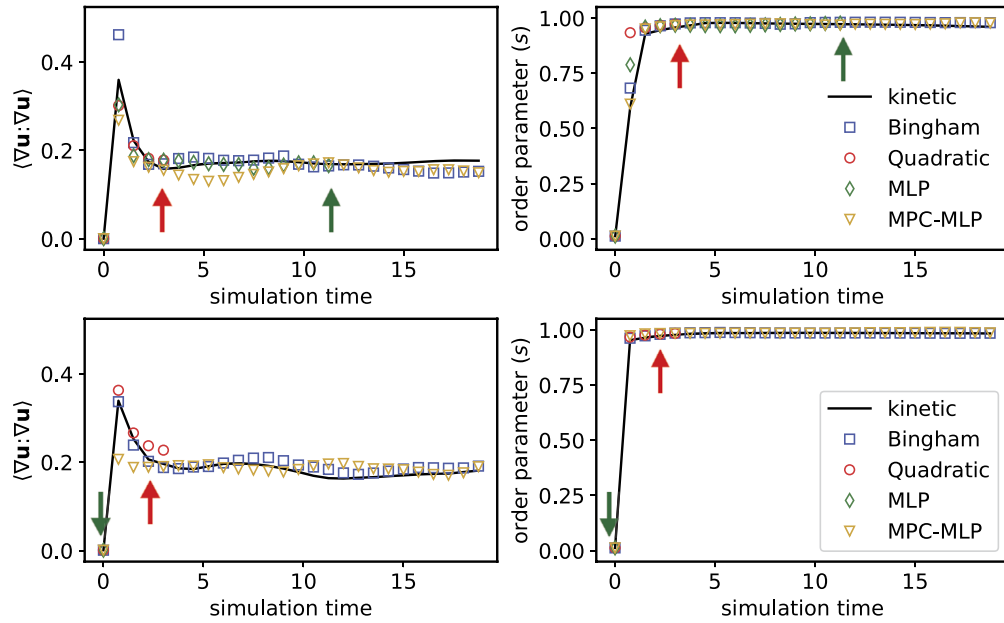


Fig. 11. Comparison of mean velocity statistics $(\nabla \mathbf{u} : \nabla \mathbf{u})_V$ and scalar order parameter (s): The *a-posteriori* tests are performed under different closure models with alignment strength $\zeta = 7$ (top row) and $\zeta = 15$ (bottom row). Different colored symbols correspond to different closure models employed as described in the inset. The colored arrows (red: quadratic, green: MLP) represent the time point at which the coarse-grained simulations become unstable. The length of prediction window was set to $q = 4$ to train neural networks based on MPC procedure.

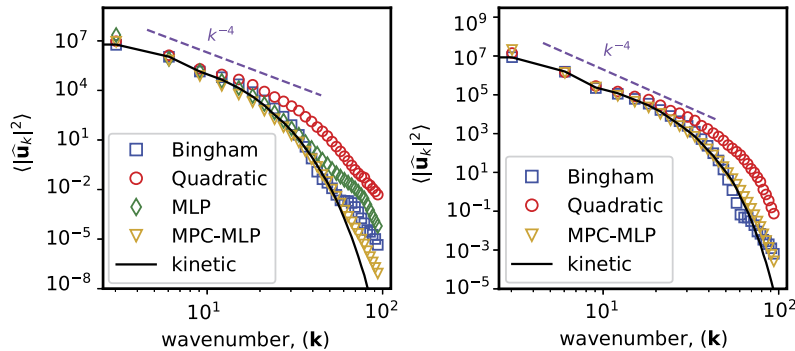


Fig. 12. Power spectrum comparison: Velocity power spectrum calculated from coarse-grained simulations under different closure models for alignment strength $\zeta = 7$ and $\zeta = 15$.

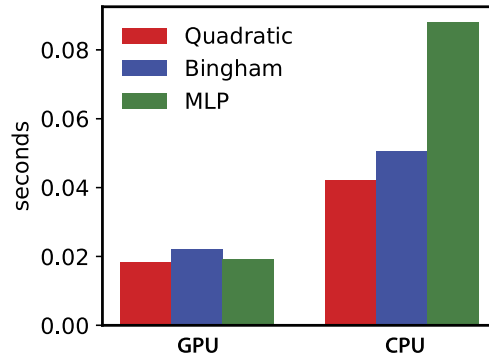


Fig. 13. Computational cost: Cost per time-step of the coarse-grained simulations under different closure models. The GPU timings were measured on a Nvidia A100 graphics card and CPU timings were measured on an Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz.

simulations. Fig. 11 illustrates that Model-Predictive Control clearly improves the temporal stability of the data-driven closures in comparison with models trained with supervised learning. This is also very much reflected in the velocity power spectrum comparison shown in Fig. 12, where the quadratic closure and MLP based on supervised learning do not adequately resolve the intermediate to large wavenumbers regime of the spectrum. We could not find any stable numerical results for the linear closure model for alignment strengths of $\zeta = 7$ and $\zeta = 15$. In Fig. 13 we report the cost per time-step of the coarse-grained simulations under different closure models. On GPUs, the computational cost shows little variation across different closure models. Despite having ~ 10000 parameters, the neural network (MLP) exhibits a similar evaluation time as the quadratic and Bingham closures. However, when running on a CPU, the quadratic and Bingham closures outperform the MLP-based predictions in terms of speed.

5. Parameter inference with nonlinear model predictive control

In the previous sections, we analyzed the learned closures in the context of *a-priori* estimates and stable and accurate *a-posteriori* predictions. However, the utility of hydrodynamic theories is not just restricted to making qualitative predictions that resemble experiments but also can be used to infer important material properties of the system. Recent studies leverage machine learning for inferring activity fields from time snapshots of director fields [45,46]. In these studies, the problem of inferring material properties is recast as a computer vision problem, and a neural network is used in a pure black-box setting to output parameter values given a huge data library of director fields. In this section, we demonstrate how accurate coarse-grained descriptions and differentiable numerical solvers can conduct parameter inference conditioned directly on the available data. More specifically, we probe how different closure models affect the accuracy of the inferred parameters.

By using nonlinear MPC and a differentiable pseudospectral solver, we explore how accurately the coefficients of the kinetic theory can be inferred from simulations of the coarse-grained description (Eq. (6), (7)) under a given closure model. The nonlinear MPC problem for inferring the alignment strength parameter ζ is

$$\tilde{\Sigma} = \underset{\Sigma}{\operatorname{argmin}} \sum_{k \in I^*} \sum_{l=1}^q v_l \left[\int_{\Omega_d} \left(\left\| \mathbf{D}_{\mathcal{K}}^{k,l}(\mathbf{x}) - \tilde{\mathbf{D}}^{k,l}(\mathbf{x}, \Sigma) \right\|_F^2 \right) d\mathbf{x} \right] + \lambda_{wd} \sum_i^{N_l} \|\mathbf{W}_i\|_F^2, \quad (22)$$

$$\text{where } \tilde{\mathbf{D}}^{k,l} = \mathcal{T}_{\mathbf{D}}^l \left(\Sigma, \tilde{\mathbf{D}}^{k,l-1}, \tilde{\mathbf{u}}^{k,l-1}, \mathcal{S} \left(\tilde{\mathbf{D}}^{k,l-1}, \tilde{\mathbf{E}}^{k,l-1} \right) \right), \text{ and } \tilde{\mathbf{D}}^{k,0} = \mathbf{D}_{\mathcal{K}}^k. \quad (23)$$

Here, \mathcal{S} is some prescribed closure model approximating $\mathbf{S} : \mathbf{T}$ and Σ is the parameter being inferred. As expected, Fig. 14 shows that both the Bingham closure and the data-driven closure lead to better parameter estimation. The MLP architecture was trained using the invariant input feature representation and the nonlinear MPC procedure discussed in the previous section. To be able to predict across different values of parameters, the neural network used for parameter estimation was trained using the active learning strategy described in section 4.3. We found the linear and quadratic closures encountered difficulties in parameter inference at low alignment strengths, which is in line with the *a-priori* estimates shown in Fig. 4. In general, we found that estimating the dipole strength α at low alignment strength leads to poor parameter estimates. The relative error is close to 20% for both Bingham and data-driven closure, 30% for the quadratic closure, and $\approx 40\%$ for the linear closure.

The poor parameter estimates can be attributed to an ill-conditioned loss landscape potentially with multiple local minima associated with the nonconvex optimization problem. These issues are further exacerbated due to the reduced accuracy of the closure models at low values of alignment strength. Future research should focus on addressing this challenge, possibly through advanced optimization techniques that can navigate the ill-conditioned landscape more efficiently, or by exploring alternate formulations that inherently lead to well-conditioned optimization problems [47,48]. This direction holds promise for improving parameter estimation and rendering the closures more accurate and reliable for inverse problems.

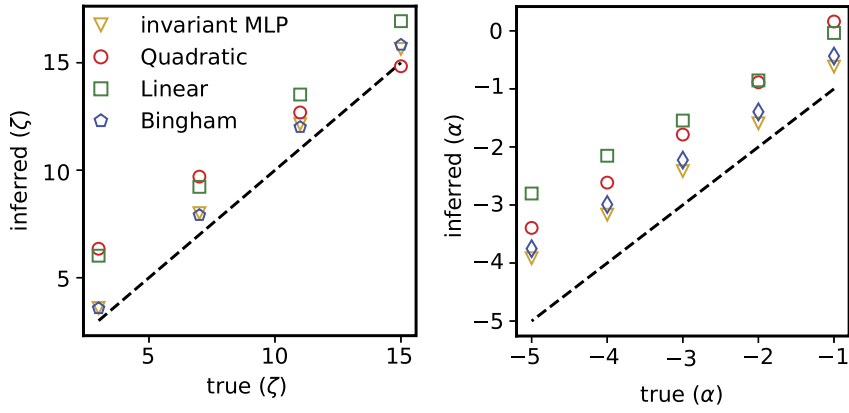


Fig. 14. Parameter inference: Left: Alignment strength ζ inferred under different closures from data generated through kinetic theory with the dipole strength fixed at $\alpha = -1$. Right: Inference of dipole strength α at fixed alignment strength $\zeta = 3$.

6. Conclusion and outlook

We have presented a learning framework based on tensor-valued isotropic function representation of the closure terms arising in continuum models of active suspensions. We conduct both *a-priori* and *a-posteriori* analyses of the learned closures and provide a quantitative comparison with existing closures. Our comparison benchmarks clearly show that, with no additional cost, neural networks trained on rotationally invariant features have better prediction accuracy and generalizing power than those trained with tensor components as input features. Furthermore, we show that the learning framework based on invariant representation results in a two-order of magnitude increase in prediction accuracy when compared to widely employed closure models in passive and active liquid crystal theories. We also provide performance benchmarks between different neural network architectures with varying receptive and response fields. The MLP neural architectures based on point-wise learning between the input and the target yielded the best predictive performance, in comparison with CNN and FNO. In particular, for a comparable number of training parameters, the FNO architecture based on global convolutions performed poorly relative to MLP and CNN. This demonstrated that nonlocal effects can be safely ignored to model the closure terms. To efficiently explore the parameter space, we use an active learning strategy, training a single neural network to predict across the parameter regime by interpolating between the low and high values of both the dipole coefficient and the alignment strength.

To bridge the gap between *a-priori* analysis and *a-posteriori* predictions, we developed a compute and memory efficient optimization procedure based on nonlinear model predictive control (MPC). By integrating neural networks with pseudo-spectral solvers, we were able to induce temporal stability prior to the learning problem. Within the discrete MPC framework, the neural network is trained to approximate the closure terms such that the discrepancy between the numerical solver's prediction and the kinetic theory is minimized within a prescribed training time horizon. In this way, we learn closures that can update the moments of the distribution function in a stable manner, which is not necessarily guaranteed from closures learned through supervised learning approaches with separate *a-posteriori* testing. When run in inference mode, the coarse-grained simulations based on the learned models trained with nonlinear MPC remained stable, and accurately reproduced the mean velocity statistics, global order parameters, and velocity spectra of the kinetic theory. We leverage our differentiable pseudo-spectral solver with the data-driven closure to conduct parameter inference. Unsurprisingly, our analysis indicated a correlation between the accuracy of the closure model and the fidelity of the inferred parameters. Consequently, Bingham and data-driven closures yielded the most reliable estimates. Furthermore, our study has demonstrated that data-driven closures based on information from both the orientation and rates-of-strain tensors are able to capture defects more accurately than commonly used closures. This suggests future directions for incorporating velocity information into existing closure models to effectively resolve defects.

In the present study, we only consider two-dimensional suspensions. However, extension of the formulation to three dimensions is straightforward. In three dimensions, new tensor bases and invariant quantities are added to the input features of the learning framework in its current form. However, generating three-dimensional kinetic simulations is computationally demanding and one would have to instead rely on highly resolved coarse-grained simulations, such as those based on the Bingham closure, as the ground truth. Future extensions where a differentiable solver that directly trains towards attaining *a-posteriori* statistics like the scalar order-parameter or velocity spectra can be envisioned [49].

In this study, the focus was solely on the characterization of apolar suspensions and the representation of closure terms that emerge in their coarse-grained models. However, many active matter systems have polar order, such as microtubule and motor protein assemblies or collections of motile bacteria. Future work should extend the current learning framework to account for polarity, thereby describing a wider range of active matter systems. In summary, data-driven techniques provide a systematic approach to ensure consistency between coarse-grained macroscopic descriptions and the microscopic details of the underlying physical process.

Code availability In the interest of reproducibility, we publish our implementation available to users: https://github.com/SuryanarayanaMK/Learning_closures/tree/master. In the same repository, we also provide the source code to simulate kinetic theory and to generate the reference data.

CRedit authorship contribution statement

Suryanarayana Maddu: Formal analysis, Writing – original draft, Investigation, Validation, Writing – review & editing, Visualization. **Scott Weady:** Writing – review & editing, Formal analysis, Investigation, Validation, Visualization. **Michael J. Shelley:** Formal analysis, Investigation, Validation, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

We thank D. Stein and B. Chakrabarti for useful discussions. The authors gratefully acknowledge funding from National Science Foundation Grant No. DMR-2004469 (M.J.S.).

Appendix A. Isotropic tensor-valued functions in two dimensions

In this section, we derive the form of the second-order tensor-valued isotropic functions $\mathbf{S}:\mathbf{D}$ and $\mathbf{S}:\mathbf{E}$, which are functions of the tensor arguments \mathbf{D} and \mathbf{D}, \mathbf{E} , respectively. To do this, we first discuss the general representation of a scalar-valued isotropic function f that has symmetric tensor-valued arguments $\{\mathbf{A}_i\}, i = 1, \dots, M$. The general representation theorem [26] states that the isotropic function f can be expressed in terms of the invariants of the functional basis of the argument tensors,

$$f(\mathbf{A}_1, \dots, \mathbf{A}_M) = f(I_s), \quad (\text{A.1})$$

where $I_s, (s = 1, 2, \dots, S)$ are the invariants of the functional basis. The Cayley-Hamilton theorem [27,26] can be used to deduce the minimal functional basis of an arbitrary set of tensors. In two-dimension, the functional basis is given as,

$$\{\text{tr}\mathbf{A}_i, \text{tr}\mathbf{A}_i^2, \text{tr}\mathbf{A}_i\mathbf{A}_j\}_{i,j=1,2,\dots,M; i < j} \quad (\text{A.2})$$

To derive the expression for $\mathbf{S}:\mathbf{E} = \mathbf{F}(\mathbf{D}, \mathbf{E})$, we introduce a symmetric tensor $\mathbf{C} = \mathbf{C}^\top$ that forms a scalar-valued isotropic function f through the inner product between the tensors \mathbf{C} and $\mathbf{F}(\mathbf{D}, \mathbf{E})$,

$$f(\mathbf{C}, \mathbf{D}, \mathbf{E}) = \text{tr}(\mathbf{C}\mathbf{F}(\mathbf{D}, \mathbf{E})). \quad (\text{A.3})$$

It can be easily demonstrated that the function $f(\mathbf{C}, \mathbf{D}, \mathbf{E})$ is an isotropic function of its two symmetric tensor arguments due to the transformation invariance of the trace operator, as shown below:

$$\begin{aligned} f(\mathbf{Q}\mathbf{C}\mathbf{Q}^\top, \mathbf{Q}\mathbf{D}\mathbf{Q}^\top, \mathbf{Q}\mathbf{E}\mathbf{Q}^\top) &= \text{tr}(\mathbf{Q}\mathbf{C}\mathbf{Q}^\top \mathbf{F}(\mathbf{Q}\mathbf{D}\mathbf{Q}^\top, \mathbf{Q}\mathbf{E}\mathbf{Q}^\top)) \\ &= \text{tr}(\mathbf{Q}\mathbf{C}\mathbf{F}(\mathbf{D}, \mathbf{E})\mathbf{Q}^\top) \\ &= \text{tr}(\mathbf{C}\mathbf{F}(\mathbf{D}, \mathbf{E})) = f(\mathbf{C}, \mathbf{D}, \mathbf{E}). \end{aligned}$$

From Eqs. (A.1)-(A.2), we see that any scalar-valued isotropic function f can be represented in terms of the invariants of the functional basis of its argument tensors. The minimal functional basis associated with the argument tensors $\mathbf{C}, \mathbf{D}, \mathbf{E}$ is

$$\{\text{tr}\mathbf{D}, \text{tr}\mathbf{C}, \text{tr}\mathbf{E}, \text{tr}\mathbf{C}^2, \text{tr}\mathbf{D}^2, \text{tr}\mathbf{E}^2, \text{tr}\mathbf{C}\mathbf{D}, \text{tr}\mathbf{C}\mathbf{E}, \text{tr}\mathbf{D}\mathbf{E}\}.$$

Since, by construction, $f(\mathbf{C}, \mathbf{D}, \mathbf{E})$ is linear in the tensor argument \mathbf{C} , only the invariants $\{\text{tr}\mathbf{C}, \text{tr}\mathbf{C}\mathbf{D}, \text{tr}\mathbf{C}\mathbf{E}\}$ are considered. This restricts the form of the scalar function f to:

$$f(\mathbf{C}, \mathbf{D}, \mathbf{E}) = \text{tr}(\mathbf{C}\mathbf{F}(\mathbf{D}, \mathbf{E})) = \text{tr}[\mathbf{C}(\kappa_0\mathbf{I} + \kappa_1\mathbf{D} + \kappa_2\mathbf{E})], \quad (\text{A.4})$$

where $\kappa_0, \kappa_1, \kappa_2$ are scalar functions of the invariants of \mathbf{D}, \mathbf{E} . Therefore, the final form of a symmetric isotropic tensor-valued function of a symmetric tensor is

$$\mathbf{F}(\mathbf{D}, \mathbf{E}) = \mathbf{S} : \mathbf{E} = \kappa_0 \mathbf{I} + \kappa_1 \mathbf{D} + \kappa_2 \mathbf{E}.$$

The expansion for the tensor-valued function $\mathbf{S} : \mathbf{D}$ can be derived through a similar procedure by omitting the dependence on \mathbf{E} .

Appendix B. Hyper-parameters of neural network architectures

Table 1

Hyper-parameters of the different network architectures used for *a-priori* and *a-posteriori* analysis. In CNN architectures a filter of size 3×3 was used.

	# input channels	#layers	# filters/nodes	# output channels
CNN invariant	3	6	16	5
CNN component	6	6	16	6
	# input channels	#layers	# modes/width	# output channels
FNO invariant	3	3	16/2	5
FNO component	6	3	14/2	6
	input-size	#layers	# nodes	output-size
MLP invariant	3	5	50	5
MLP component	6	5	50	6

References

- [1] Sriram Ramaswamy, The mechanics and statistics of active matter, *Annu. Rev. Condens. Matter Phys.* 1 (1) (2010) 323–345.
- [2] M. Cristina Marchetti, Jean-François Joanny, Sriram Ramaswamy, Tanniemola B. Liverpool, Jacques Prost, Madan Rao, R. Aditi Simha, Hydrodynamics of soft active matter, *Rev. Mod. Phys.* 85 (3) (2013) 1143.
- [3] David Saintillan, Michael J. Shelley, Active suspensions and their nonlinear models, *C. R. Phys.* 14 (6) (2013) 497–517.
- [4] Tamás Vicsek, Anna Zafeiris, Collective motion, *Phys. Rep.* 517 (3–4) (2012) 71–140.
- [5] Christopher Dombrowski, Luis Cisneros, Sunita Chatkaew, Raymond E. Goldstein, John O. Kessler, Self-concentration and large-scale coherence in bacterial dynamics, *Phys. Rev. Lett.* 93 (9) (2004) 098103.
- [6] Jörn Dunkel, Sebastian Heidenreich, Knut Drescher, Henricus H. Wensink, Markus Bär, Raymond E. Goldstein, Fluid dynamics of bacterial turbulence, *Phys. Rev. Lett.* 110 (22) (2013) 228102.
- [7] Ricard Alert, Jaume Casademunt, Jean-François Joanny, Active turbulence, *Annu. Rev. Condens. Matter Phys.* 13 (2022) 143–170.
- [8] R. Aditi Simha, Sriram Ramaswamy, Hydrodynamic fluctuations and instabilities in ordered suspensions of self-propelled particles, *Phys. Rev. Lett.* 89 (5) (2002) 058101.
- [9] John Toner, Yuhai Tu, Flocks, herds, and schools: a quantitative theory of flocking, *Phys. Rev. E* 58 (4) (1998) 4828.
- [10] Barath Ezhilan, Michael J. Shelley, David Saintillan, Instabilities and nonlinear dynamics of concentrated active suspensions, *Phys. Fluids* 25 (7) (2013) 070607.
- [11] E.J. Hinch, L.G. Leal, Constitutive equations in suspension mechanics. Part 2. Approximate forms for a suspension of rigid particles affected by Brownian rotations, *J. Fluid Mech.* 76 (1) (1976) 187–208.
- [12] Scott Weady, David B. Stein, Michael J. Shelley, Thermodynamically consistent coarse-graining of polar active fluids, *Phys. Rev. Fluids* 7 (6) (2022) 063301.
- [13] Scott Weady, Michael J. Shelley, David B. Stein, A fast Chebyshev method for the Bingham closure with application to active nematic suspensions, *J. Comput. Phys.* 457 (2022) 110937.
- [14] Tong Gao, Meredith D. Betterton, An-Sheng Jhang, Michael J. Shelley, Analytical structure, dynamics, and coarse graining of a kinetic model of an active fluid, *Phys. Rev. Fluids* 2 (9) (2017) 093302.
- [15] Steven L. Brunton, Bernd R. Noack, Petros Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* 52 (2020) 477–508.
- [16] Julia Ling, Reese Jones, Jeremy Templeton, Machine learning strategies for systems with invariance properties, *J. Comput. Phys.* 318 (2016) 22–35.
- [17] Julia Ling, Andrew Kurawski, Jeremy Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *J. Fluid Mech.* 807 (2016) 155–166.
- [18] Romit Maulik, Omer San, Adil Rasheed, Prakash Vedula, Subgrid modelling for two-dimensional turbulence using neural networks, *J. Fluid Mech.* 858 (2019) 122–144.
- [19] Laure Zanna, Thomas Bolton, Data-driven equation discovery of ocean mesoscale closures, *Geophys. Res. Lett.* 47 (17) (2020) e2020GL088376.
- [20] Eric J. Parish, Karthik Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, *J. Comput. Phys.* 305 (2016) 758–774.
- [21] Pratap Anand Singh, Shivaji Medida, Karthik Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, *AIAA J.* 55 (7) (2017) 2215–2227.
- [22] Björn List, Li-Wei Chen, Nils Thuerey, Learned turbulence modelling with differentiable fluid solvers: physics-based loss functions and optimisation horizons, *J. Fluid Mech.* 949 (2022) A25.
- [23] Justin Sirignano, Jonathan F. MacArt, Jonathan B. Freund DPM, A deep learning PDE augmentation method with application to large-eddy simulation, *J. Comput. Phys.* 423 (2020) 109811.
- [24] Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, Stephan Hoyer, Machine learning-accelerated computational fluid dynamics, *Proc. Natl. Acad. Sci.* 118 (21) (2021) e2101784118.
- [25] Jonathan F. MacArt, Justin Sirignano, Jonathan B. Freund, Embedded training of neural-network subgrid-scale turbulence models, *Phys. Rev. Fluids* 6 (5) (2021) 050502.
- [26] John Korsgaard, On the representation of two-dimensional isotropic functions, *Int. J. Eng. Sci.* 28 (7) (1990) 653–662.
- [27] Stephen B. Pope, A more general effective-viscosity hypothesis, *J. Fluid Mech.* 72 (2) (1975) 331–340.
- [28] Andrea Beck, David Flad, Claus-Dieter Munz, Deep neural networks for data-driven LES closure models, *J. Comput. Phys.* 398 (2019) 108910.
- [29] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, Lawrence Jackel, Handwritten digit recognition with a back-propagation network, *Adv. Neural Inf. Process. Syst.* 2 (1989).
- [30] Yann LeCun, Yoshua Bengio, et al., Convolutional networks for images, speech, and time series, in: *The Handbook of Brain Theory and Neural Networks*, vol. 3361 (10), 1995, p. 1995.
- [31] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, Fourier neural operator for parametric partial differential equations, *arXiv preprint*, arXiv:2010.08895, 2020.

- [32] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, Neural operator: learning maps between function spaces, arXiv preprint, arXiv:2108.08481, 2021.
- [33] Diederik P. Kingma, Jimmy Ba Adam, A method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.
- [34] Masao Doi, Molecular dynamics and rheological properties of concentrated solutions of rodlike polymers in isotropic and liquid crystalline phases, *J. Polym. Sci., Polym. Phys. Ed.* 19 (2) (1981) 229–243.
- [35] Brian J. Edwards, Antony N. Beris, Miroslav Grmela, Generalized constitutive equation for polymeric liquid crystals part 1. Model formulation using the Hamiltonian (Poisson bracket) formulation, *J. Non-Newton. Fluid Mech.* 35 (1) (1990) 51–72.
- [36] D. Marenduzzo, E. Orlandini, M.E. Cates, J.M. Yeomans, Steady-state hydrodynamic instabilities of active liquid crystals: hybrid lattice Boltzmann simulations, *Phys. Rev. E* 76 (3) (2007) 031921.
- [37] Charu V. Chaubal, L. Gary Leal, A closure approximation for liquid-crystalline polymer models based on parametric density estimation, *J. Rheol.* 42 (1) (1998) 177–201.
- [38] Christopher J. Arthurs, Andrew P. King, Active training of physics-informed neural networks to aggregate and interpolate parametric solutions to the Navier-Stokes equations, *J. Comput. Phys.* 438 (2021) 110364.
- [39] Urban Fasel, J. Nathan Kutz, Bingni W. Brunton, Steven L. Brunton Ensemble-SINDy, Robust sparse model discovery in the low-data, high-noise limit, with active learning and control, *Proc. R. Soc. A* 478 (2260) (2022) 20210904.
- [40] B. Settles, Active Learning Literature Survey, 2009.
- [41] Ricky T.Q. Chen, Yulia Rubanova, Jesse Bettencourt, David K. Duvenaud, Neural ordinary differential equations, *Adv. Neural Inf. Process. Syst.* 31 (2018).
- [42] Suryanarayana Maddu, Dominik Sturm, Bevan L. Cheeseman, Christian L. Müller, Ivo F. Sbalzarini, STENCIL-NET for equation-free forecasting from data, *Sci. Rep.* (2023).
- [43] Samuel H. Rudy, J. Nathan Kutz, Steven L. Brunton, Deep learning of dynamics and signal-noise decomposition with time-stepping constraints, *J. Comput. Phys.* 396 (2019) 483–506.
- [44] Antoine McNamara, Adrien Treuille, Zoran Popović, Jos Stam, Fluid control using the adjoint method, *ACM Trans. Graph.* 23 (3) (2004) 449–456.
- [45] Jonathan Colen, Ming Han, Rui Zhang, Steven A. Redford, Linnea M. Lemma, Link Morgan, Paul V. Ruijgrok, Raymond Adkins, Zev Bryant, Zvonimir Dogic, et al., Machine learning active-nematic hydrodynamics, *Proc. Natl. Acad. Sci.* 118 (10) (2021) e2016708118.
- [46] Anna Frishman, Kinneret Keren, Learning active nematics one step at a time, *Proc. Natl. Acad. Sci.* 118 (12) (2021) e2102169118.
- [47] Thomas Frerix, Dmitrii Kochkov, Jamie Smith, Daniel Cremers, Michael Brenner, Stephan Hoyer, Variational data assimilation with a learned inverse observation operator, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 3449–3458.
- [48] Marylou Gabrié, Grant M. Rotskoff, Eric Vanden-Eijnden, Adaptive Monte Carlo augmented with normalizing flows, *Proc. Natl. Acad. Sci.* 119 (10) (2022) e2109420119.
- [49] Guido Novati, Hugues Lascombes de Laroussilhe, Petros Koumoutsakos, Automating turbulence modelling by multi-agent reinforcement learning, *Nat. Mach. Intell.* 3 (1) (2021) 87–96.