FLATIRON INSTITUTE
Center for Computational Mathematics

Developing Reduced-Order PDEs With Machine Learning-Based Closure Models

Jiequn Han
Center of Computational Mathematics
Flatiron Institute

AI for Science, ICML 2022 Workshop
July 23, 2022

Collaboration with Zheng Ma, Chao Ma, Weinan E, Xu-Hui Zhou, Muhammad I. Zafar, Ruiying Xu, Richard Dwight, Heng Xiao

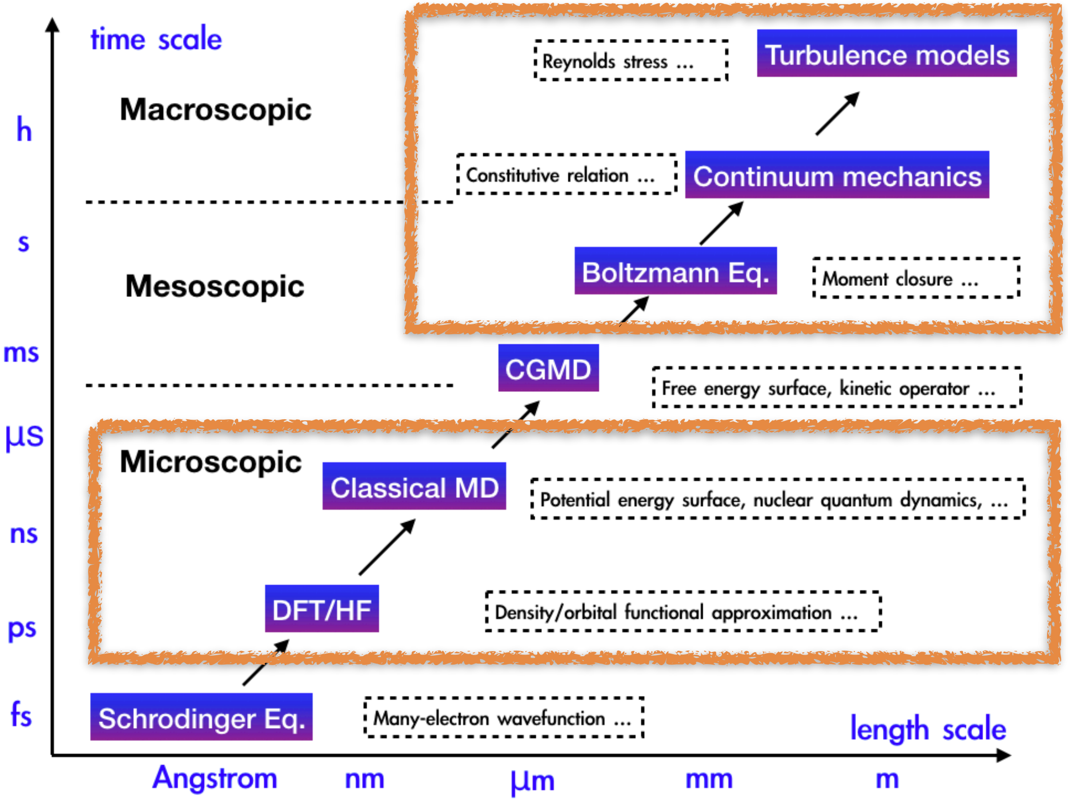
Talk Overview

1. Introduction: closure problem for reduced-order PDEs

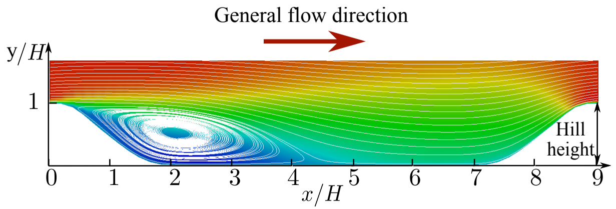
2. Principle 1: respect physical symmetry (example: Navier-Stokes equation)

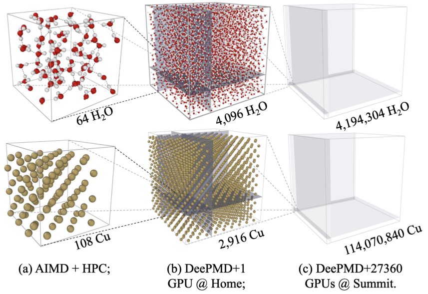
3. Principle 2: data exploration (example: Boltzmann equation)

Deep Learning for Multiscale Modeling



Machine-learning-assisted modeling, E, Han, and Zhang, Physics Today (2021)





FLATIRON INSTITUTE
Center for Computational Mathematics

PDE Example 1: Navier-Stokes Equations

Navier-Stokes equations:

(4 eqns, 4 vars)

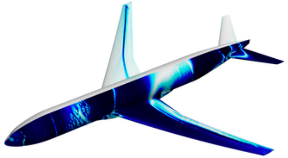
$$\nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \nabla^2 \boldsymbol{u} + \frac{1}{\rho} \nabla p = 0$$

Reynolds decomposition: $\boldsymbol{u}(t, \boldsymbol{x}) \quad = \quad \boldsymbol{U}(\boldsymbol{x}) \quad + \quad \boldsymbol{u}'(t, \boldsymbol{x})$

(time average) (fluctuation)

$$\boldsymbol{U}(\boldsymbol{x}) = \bar{\boldsymbol{u}}(t, \boldsymbol{x}) = \lim_{T \to \infty} \frac{1}{T} \int_0^T \boldsymbol{u}(t, \boldsymbol{x}) \mathrm{d}t$$



Reynolds-averaged Navier-Stokes equations (RANS)

$$\nabla \cdot \boldsymbol{U} = 0$$

(4 eqns, > 4 vars)

$$\boldsymbol{U} \cdot \nabla \boldsymbol{U} - \nu \nabla^2 \boldsymbol{U} + \frac{1}{\rho} \nabla P = \nabla \cdot \boldsymbol{\tau}, \quad \boldsymbol{\tau}_{ij} \equiv \overline{\boldsymbol{u}_i' \boldsymbol{u}_j'}$$

Reynolds stress

Turbulence closure: use mean velocity U & pressure P to express Reynolds stress τ such that the equations are closed/solvable

FLATIRON INSTITUTE
Center for Computational Mathematics

PDE Example 2: Boltzmann Equation

Number density of particles $f = f(\boldsymbol{x}, \boldsymbol{v}, t) : \mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}, \quad D \in \{1, 2, 3\}$

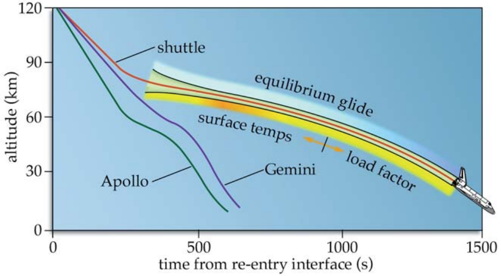
Boltzmann equations: $\partial_t f + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v} f) = \frac{1}{\varepsilon} Q(f)$

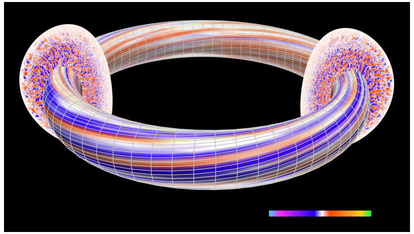
(transportation) (collision, complicated)

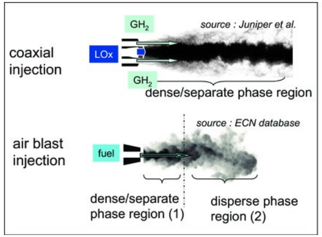
$$Q(f) = \int_{\mathbb{R}^D} \mathrm{d}\boldsymbol{v}_* \int_{\mathbb{S}^{D-1}} \mathrm{d}\boldsymbol{n} \; B(\boldsymbol{v}, \boldsymbol{v}_*, \boldsymbol{n})(f(\boldsymbol{v}')f(\boldsymbol{v}'_*) - f(\boldsymbol{v})f(\boldsymbol{v}_*))$$

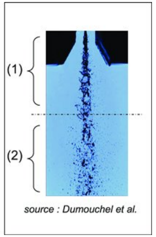
$$\boldsymbol{v}' = \frac{1}{2}(\boldsymbol{v} + \boldsymbol{v}_* + |\boldsymbol{v} - \boldsymbol{v}_*|\boldsymbol{n}), \quad \boldsymbol{v}'_* = \frac{1}{2}(\boldsymbol{v} + \boldsymbol{v}_* - |\boldsymbol{v} - \boldsymbol{v}_*|\boldsymbol{n})$$

conserving mass, momentum, and energy







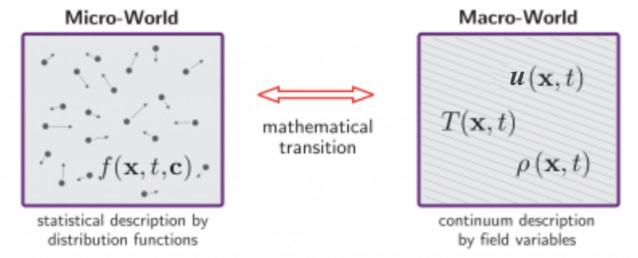


FLATIRON INSTITUTE
Center for Computational Mathematics

PDE Example 2: Boltzmann Equation

The macro-scale properties of interest can be derived from f

$$\partial_t f + \nabla_x \cdot (vf) = Q(f)$$



$\rho = \int_{\mathbb{R}^D} f \, \mathrm{d}v$ (density) $\quad u = \frac{1}{\rho} \int_{\mathbb{R}^D} f v \, \mathrm{d}v$ (bulk velocity) $\quad T = \frac{1}{D\rho} \int_{\mathbb{R}^D} f \, |v - u|^2 \, \mathrm{d}v$ (temperature)

$$\int_{\mathbb{R}^D} \partial_t f \, \mathrm{d}v + \int_{\mathbb{R}^D} \nabla_x \cdot (vf) \, \mathrm{d}v = \int_{\mathbb{R}^D} Q(f) \, \mathrm{d}v \qquad \partial_t \rho + \nabla_x \cdot (\rho u) = 0$$

$$\int_{\mathbb{R}^D} \partial_t f v \, \mathrm{d}v + \int_{\mathbb{R}^D} \nabla_x \cdot (vf) v \, \mathrm{d}v = \int_{\mathbb{R}^D} Q(f) v \, \mathrm{d}v \qquad \partial_t (\rho u) + \nabla_x \cdot (\rho u \otimes u + \rho T) = 0$$

$$\int_{\mathbb{R}^D} \partial_t f \frac{|v|^2}{2} \, \mathrm{d}v + \int_{\mathbb{R}^D} \nabla_x \cdot (vf) \frac{|v|^2}{2} \, \mathrm{d}v = \int_{\mathbb{R}^D} Q(f) \frac{|v|^2}{2} \, \mathrm{d}v \qquad \partial_t E + \nabla_x \cdot \int_{\mathbb{R}^D} (vf) \frac{|v|^2}{2} \, \mathrm{d}v = 0$$

$$\vdots$$

$\int_{\mathbb{R}^D} (vf) \, |v|^n \, \mathrm{d}v, \quad \int_{\mathbb{R}^D} Q(f) \, |v|^n \, \mathrm{d}v$??

Moment closure: use moments to express unknown flux and source terms

Machine Learning-Based Closure

Closure: find an approximating mapping from the primary variables of interest to the unresolved variables such that the reduced-order PDEs are closed/solvable.

- Turbulence closure: use mean flow velocity/pressure to express Reynolds stress
- Moment closure: use moments to express unknown flux and source terms

Find a closure model through regression. The learned closure model is a part of the reduced PDE. Powerful when the closure model input are high-dimensional.

It is NOT using ML to find a solution to a single PDE.

It is NOT (but related to) using ML to approximate a solution operator to a family of PDEs.

Two Principles

1. How to specify the ML model for closure?

respect physical symmetry

2. How to wisely collect data to train the model?

data exploration

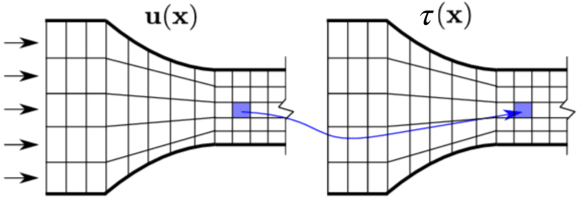
Principle 1: Model's Symmetry (Setup)

Goal of RANS closure: use (U, P) to predict the Reynolds stress τ

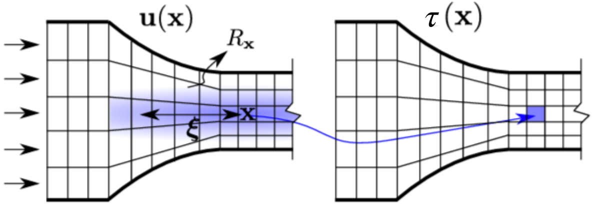
$$\nabla \cdot \boldsymbol{U} = 0$$

$$\boldsymbol{U} \cdot \nabla \boldsymbol{U} - \nu \nabla^2 \boldsymbol{U} + \frac{1}{\rho} \nabla P = \nabla \cdot \boldsymbol{\tau}$$

Local closure model: $\boldsymbol{\tau}(\boldsymbol{x}) = F(\boldsymbol{x}, \boldsymbol{U}(\boldsymbol{x}), \boldsymbol{P}(\boldsymbol{x}))$



Nonlocal closure model: $\boldsymbol{\tau}(\boldsymbol{x}) = \int_{\text{a neighborhood of } x} F(\boldsymbol{y}, \boldsymbol{U}(\boldsymbol{y}), \boldsymbol{P}(\boldsymbol{y}))\, \mathrm{d}\boldsymbol{y}$

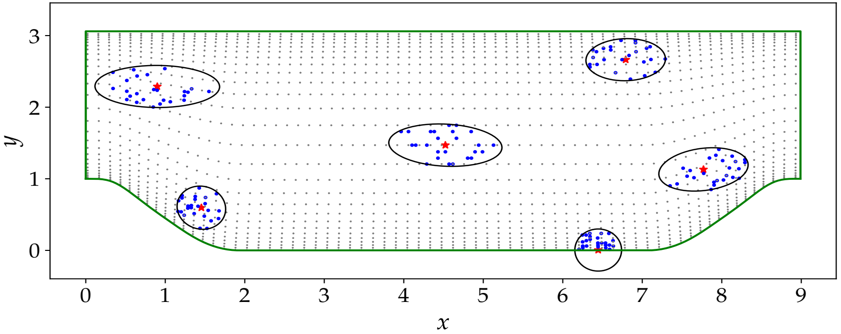


FLATIRON INSTITUTE
Center for Computational Mathematics

Principle 1: Model's Symmetry

To implement a nonlocal closure model, introduce spatial discretization:
$\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$, $\boldsymbol{U}_1 = \boldsymbol{U}(\boldsymbol{x}_1), \boldsymbol{U}_2 = \boldsymbol{U}(\boldsymbol{x}_2), \ldots, \boldsymbol{U}_n = \boldsymbol{U}(\boldsymbol{x}_n)$ *(ignoring pressure dependence for simplicity)*

$$\boldsymbol{\tau} = \boldsymbol{\tau}(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n, \boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_n)$$



Physical Symmetries

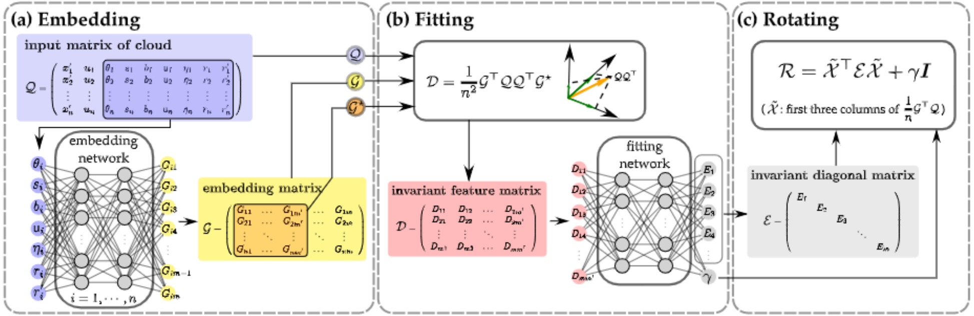
- Translation $\boldsymbol{\tau} = \boldsymbol{\tau}(\boldsymbol{x}_1 + \Delta\boldsymbol{x}, \boldsymbol{x}_2 + \Delta\boldsymbol{x}, \cdots, \boldsymbol{x}_n + \Delta\boldsymbol{x}, \boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_n)$
- Rotation $\boldsymbol{\tau} = R^\top \boldsymbol{\tau}(R\boldsymbol{x}_1, R\boldsymbol{x}_2, \cdots, R\boldsymbol{x}_n, R\boldsymbol{U}_1, R\boldsymbol{U}_2, \cdots, R\boldsymbol{U}_n)R$
- Permutation $\boldsymbol{\tau} = \boldsymbol{\tau}(\boldsymbol{x}_{\sigma(1)}, \boldsymbol{x}_{\sigma(2)}, \cdots, \boldsymbol{x}_{\sigma(n)}, \boldsymbol{U}_{\sigma(1)}, \boldsymbol{U}_{\sigma(2)}, \cdots, \boldsymbol{U}_{\sigma(n)})$

FLATIRON INSTITUTE
Center for Computational Mathematics

Vector-Cloud Neural Network (VCNN) for Symmetry

High-level ideas (similar to Deep Potential for force fields in molecular dynamics)

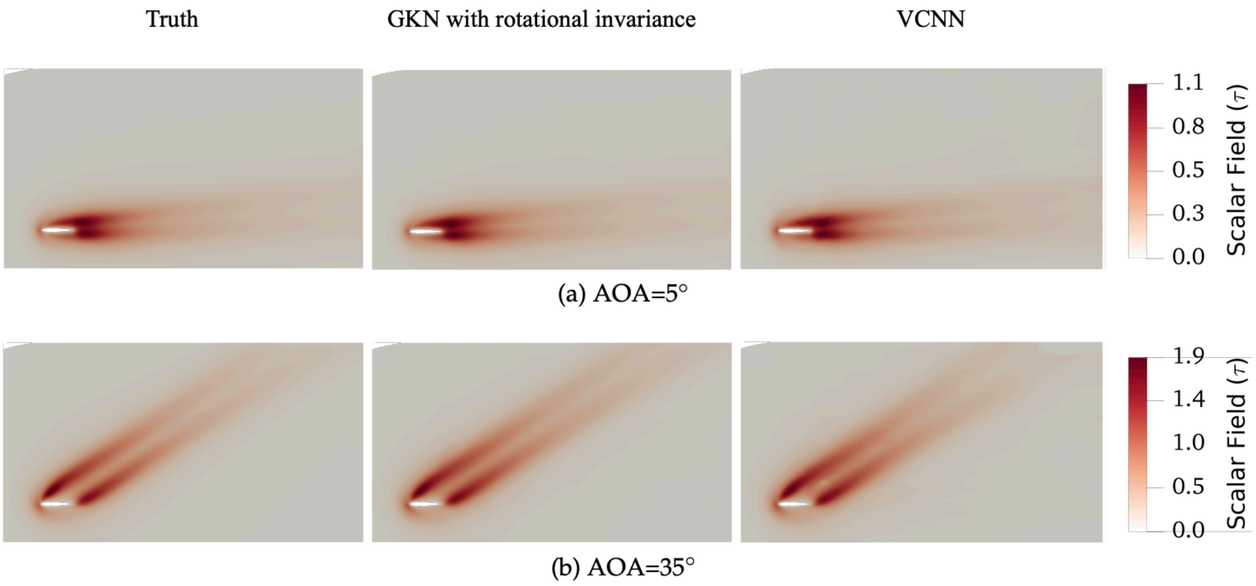
- Use relative coordinates to ensure translational invariance
- Use inner-product to ensure rotational symmetry
- Use averaged embedding to ensure permutational invariance
- Combine the above three to get symmetry features and fit the final output
- The resulting model is also adaptive to different spatial discretizations



FLATIRON INSTITUTE
Center for Computational Mathematics

VCNN-e: A vector-cloud neural network with equivariance for emulating Reynolds stress transport equations, Han, Zhou, and Xiao, arXiv (2022)

Evaluation in Different Frames

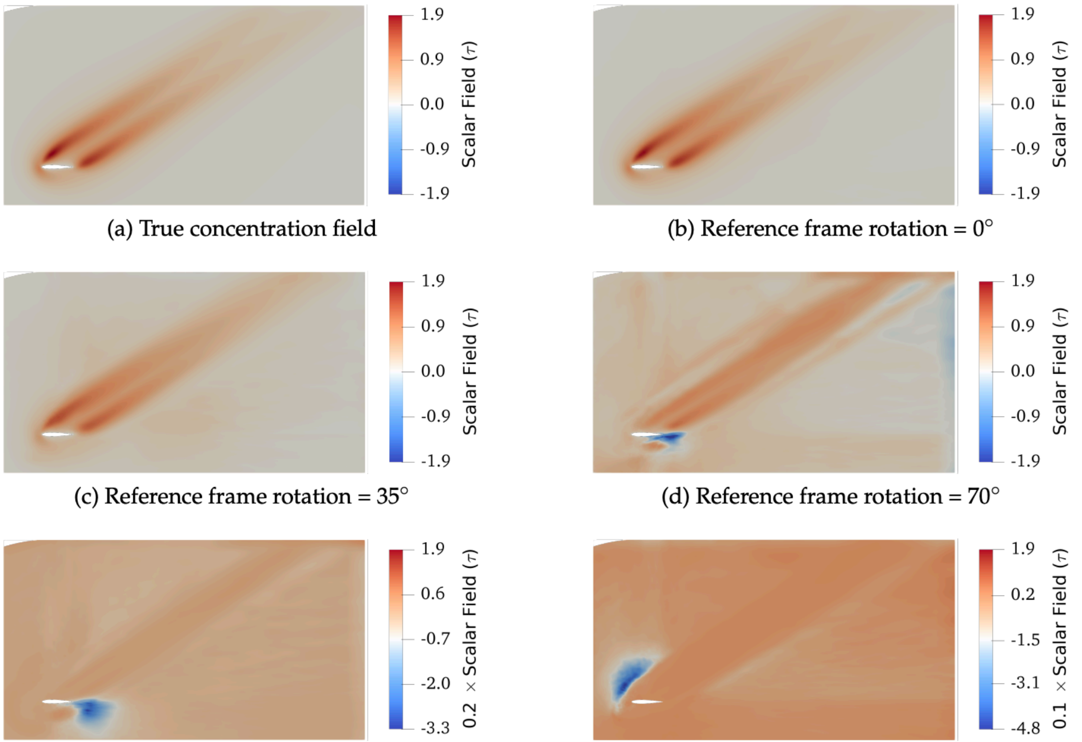


Both VCNN and Graph Kernel Network (GKN) with rotational invariance have good prediction accuracy

FLATIRON INSTITUTE
Center for Computational Mathematics

Frame invariance and scalability of neural operators for partial differential equations, Zafar, Han, Zhou, and Xiao, CiCP (2022)

Importance of Symmetry



(a) True concentration field

(b) Reference frame rotation = 0°

(c) Reference frame rotation = 35°

(d) Reference frame rotation = 70°

(e) Reference frame rotation = 90°

(f) Reference frame rotation = 180°

Prediction by an operator based on graph neural networks without rotation symmetry. The results are highly sensitive to the angles of the reference frame.

By design, VCNN gives results independent of frames.

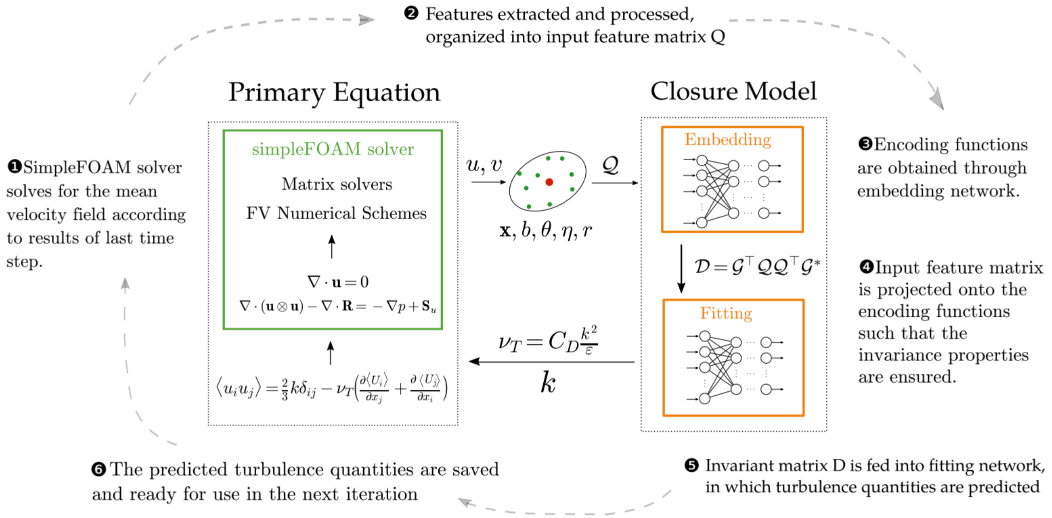
FLATIRON INSTITUTE
Center for Computational Mathematics

Frame invariance and scalability of neural operators for partial differential equations, Zafar, Han, Zhou, and Xiao, CiCP (2022)

Closure Model for RANS equations

Even if the ML closure predicts the target quantity well, how accurate is the solution to the resulting reduced PDE?

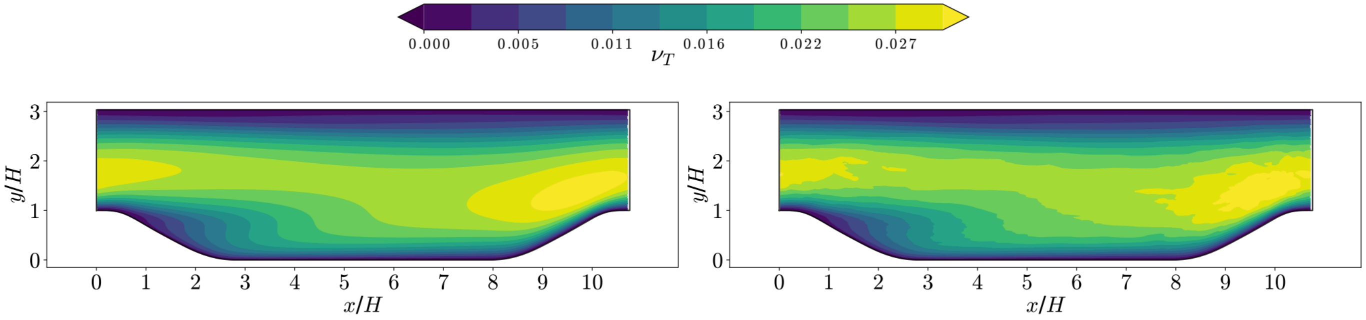
We test VCNN using the $k-\varepsilon$ model as the ground truth of RANS.



FLATIRON INSTITUTE
Center for Computational Mathematics

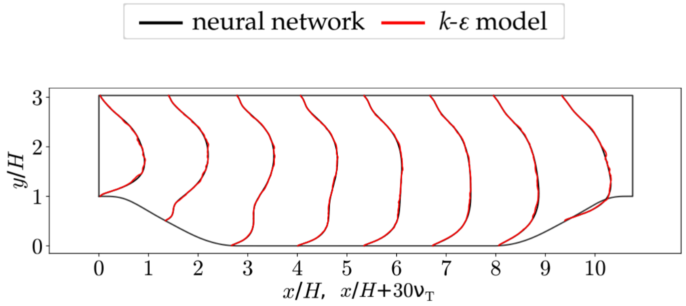
A PDE-free, neural network-based eddy viscosity model coupled with RANS equations, Xu,Zhou, Han, Dwight, and Xiao, arXiv (2022)

Closure Model for RANS equations



(a) k–ε model, $\alpha = 1.45$

(b) neural network prediction, $\alpha = 1.45$



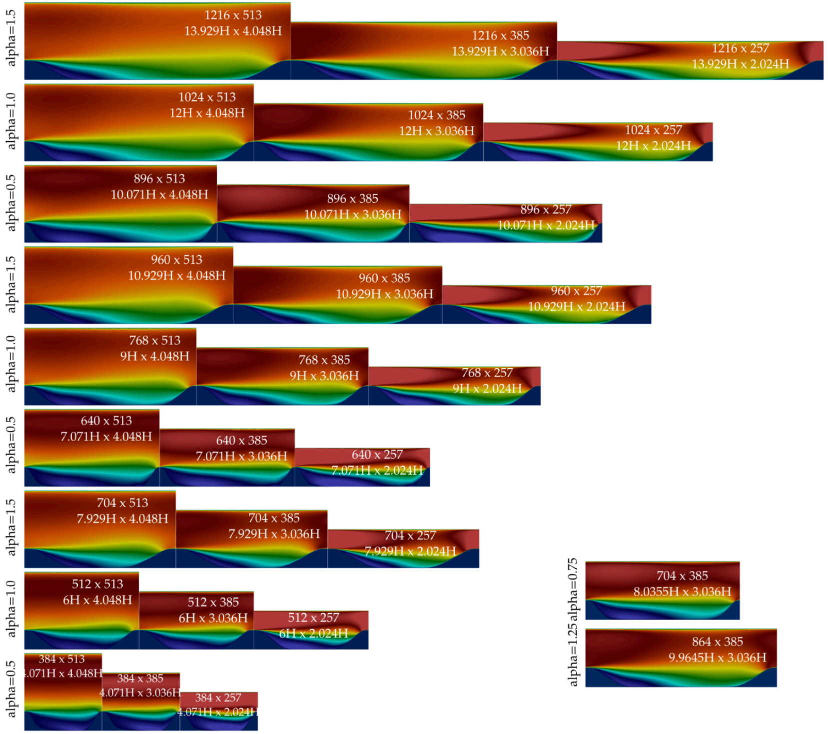
(c) neural network prediction, $\alpha = 1.45$

Reynolds number ~ 10^4

FLATIRON INSTITUTE
Center for Computational Mathematics

A PDE-free, neural network-based eddy viscosity model coupled with RANS equations, Xu,Zhou, Han, Dwight, and Xiao, arXiv (2022)

Test on DNS Data



New database with 27+2 DNS simulations:
https://github.com/xiaoh/para-database-for-PIML

(1) Reynolds number $Re = 5600$;

(2) 3 different heights, 3 different streamwise extents, and 5 different hill shapes (i.e., slope)

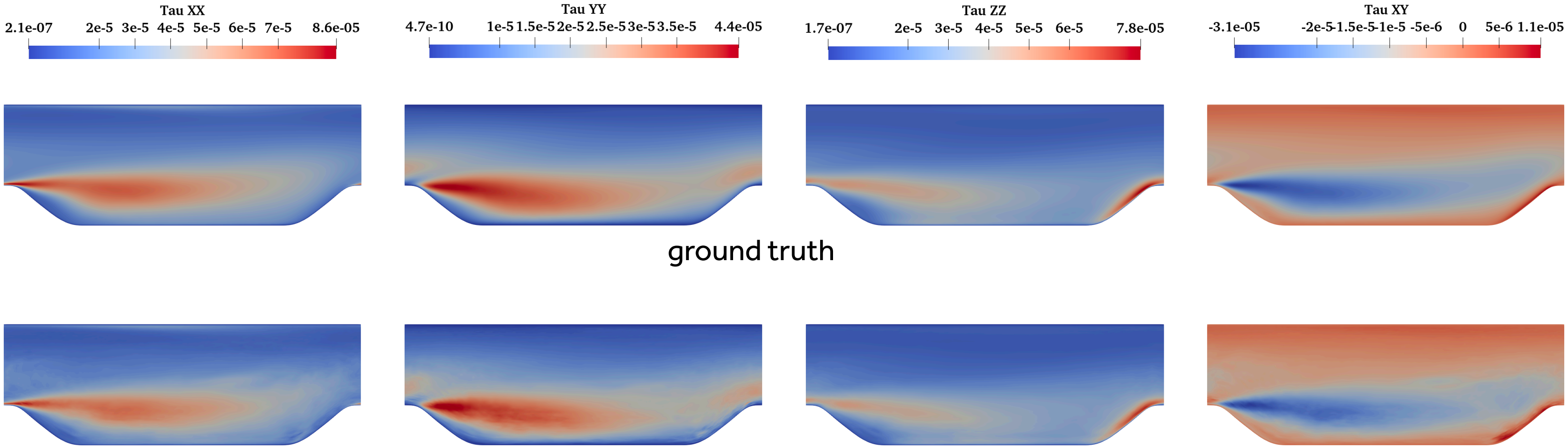
FLATIRON INSTITUTE
Center for Computational Mathematics

VCNN-e: A vector-cloud neural network with equivariance for emulating Reynolds stress transport equations, Han, Zhou, Xiao, in preparation

Test on DNS Data

Train on 4 geometries: slope = [0.5, 0.75, 1.25, 1.5]   Training error = 2.53%

Test on 1 geometry: slope = [1.0]   Testing error = 7.77%



prediction

FLATIRON INSTITUTE
Center for Computational Mathematics

VCNN-e: A vector-cloud neural network with equivariance for emulating Reynolds stress transport equations, Han, Zhou, Xiao, in preparation

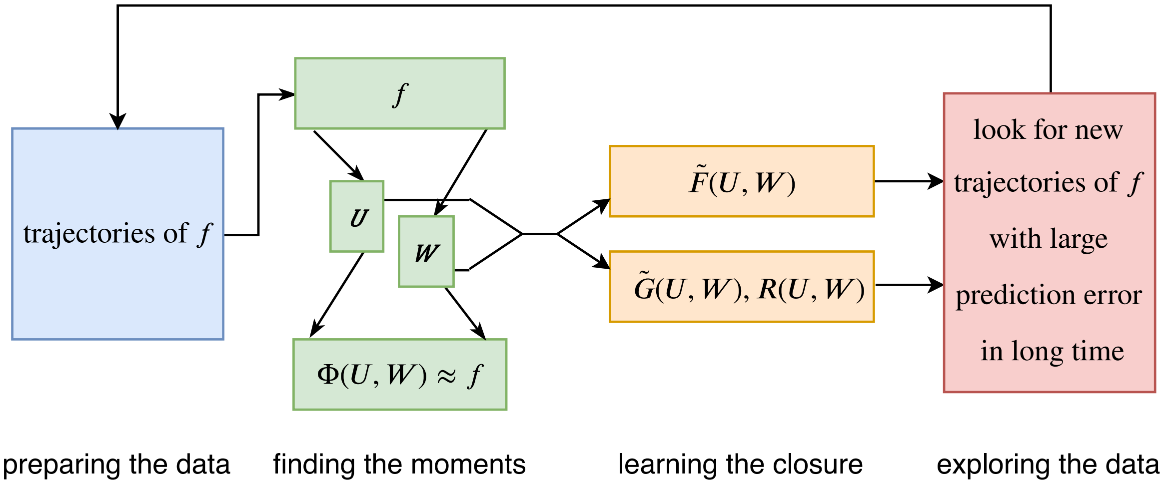
Principle 2: Data Exploration

We wish to use closure model to solve a family of PDEs with different initial conditions/boundary conditions.

Unlike most ML tasks that rely on fixed data sets, the construction of data set can be our own choice. It is an important component of the algorithm.

We want to achieve greater accuracy with fewer training data by choosing/ exploring the training data wisely.

A Scheme for Data Exploration



The "data exploration" is an iterative procedure to provide training samples of the highest quality

FLATIRON INSTITUTE
Center for Computational Mathematics

Uniformly accurate machine learning-based hydrodynamic models for kinetic equations, Han, Ma, Ma, and E, PNAS (2019)

Benefits of Data Exploration

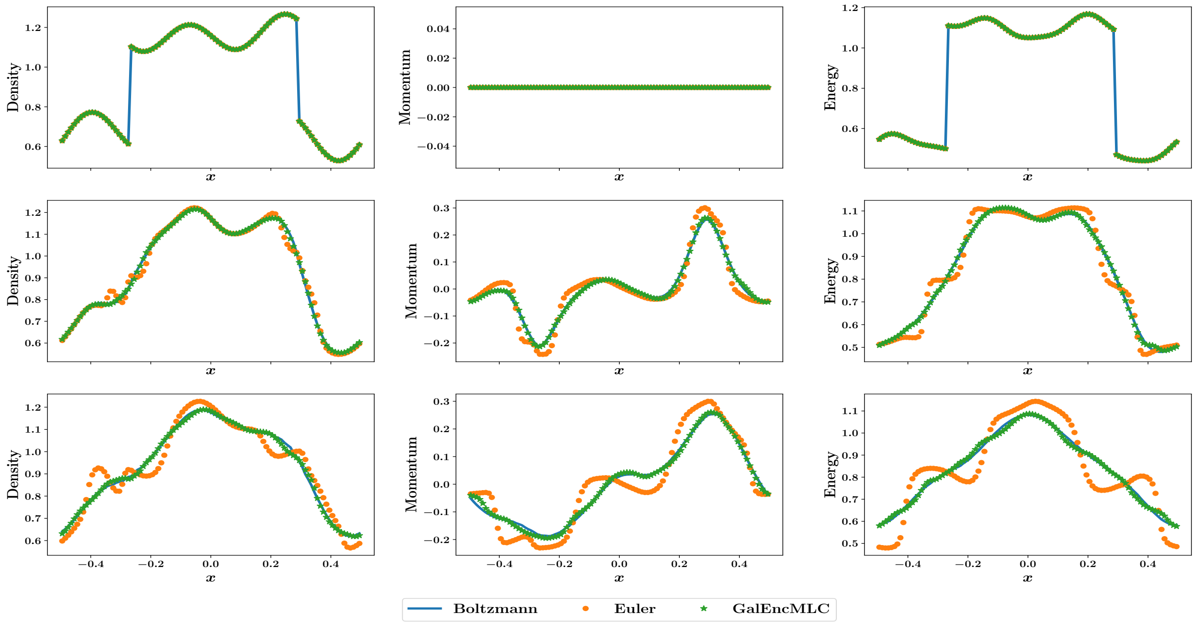
Model	*Wave*	*Mix*	*MixInTransition*
ML closure (no explor)	1.27(13), 1.60(35)	1.68(10), 2.35(12)	1.82(11), 2.49(11)
ML closure (explor)	**0.85(10), 1.01(14)**	**1.25(5), 1.75(8)**	**1.55(4), 2.06(7)**

relative errors (in percentages) of the reduced-PDE solution based on different models

Ideal scenario: given a fixed total computational budget for data generation and learning, the exploration procedure should provide training samples of the highest quality so that the best testing performance can be achieved.

Uniformly accurate machine learning-based hydrodynamic models for kinetic equations, Han, Ma, Ma, and E, PNAS (2019)

Results Based on Learned Closure



Learned from 9 moments

Sample profiles of
$\rho, \rho u, E$
(from left to right)
at t = 0, 0.05, 0.1
(from top to bottom)

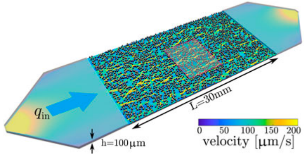
Computational cost (on a Macbook Pro): half a second using the ML-based moment method versus 5 minutes solving the original Boltzmann equation.

FLATIRON INSTITUTE
Center for Computational Mathematics

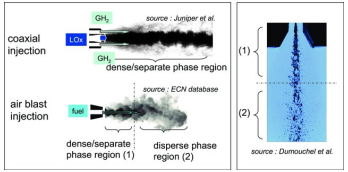
Uniformly accurate machine learning-based hydrodynamic models for kinetic equations, Han, Ma, Ma, and E, PNAS (2019)

Other Applications

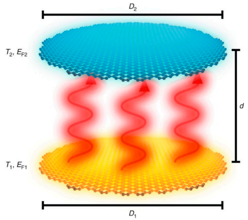
1. ML closure for non-Newtonian fluid



2. ML closure for polydisperse evaporating sprays



3. ML closure for radiative transfer equation



4. Structure preserving NN for the Boltzmann equation

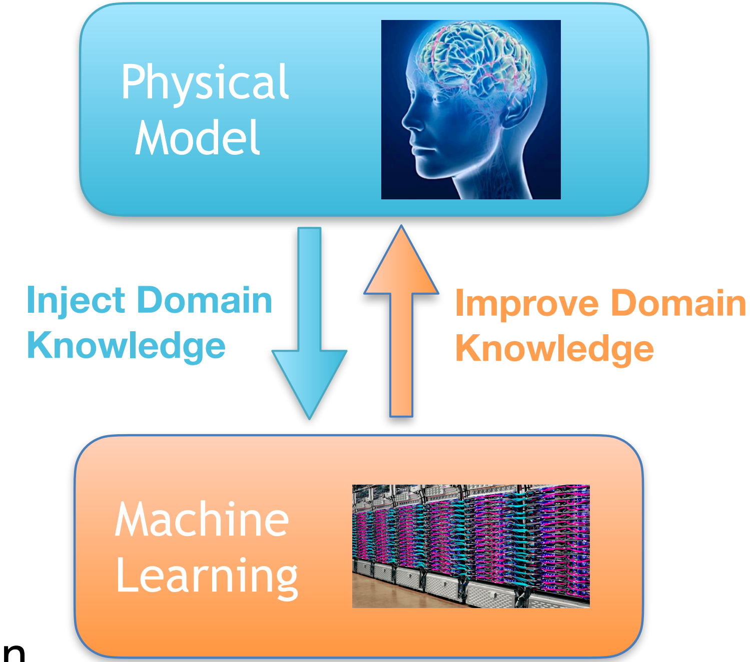
1. Machine-learning-based non-Newtonian fluid model with molecular fidelity, Lei, Wu, E, Physical Review E (2020)
2. *Machine learning moment closures for accurate and efficient simulation of polydisperse evaporating sprays, Scoggins, Han, and Massot, AIAA, (2021)*
3. *Machine learning moment closure models for the radiative transfer equation I: directly learning a gradient based closure, Huang, Cheng, Christlieb, and Roberts JCP (2022)*
4. *Structure Preserving Neural Networks: A Case Study in the Entropy Closure of the Boltzmann Equation, chotthöfer, Xiao, Frank, and Hauck, ICML (2022)*

Takeaway

1. Machine learning-based closure holds great promise to develop reduced-order PDEs.

2. Ensuring physical symmetry greatly improves training efficiency, accuracy, and generalizability.

3. Data exploration plays an important role in an interactive algorithm for realistic applications.



FLATIRON INSTITUTE
Center for Computational Mathematics

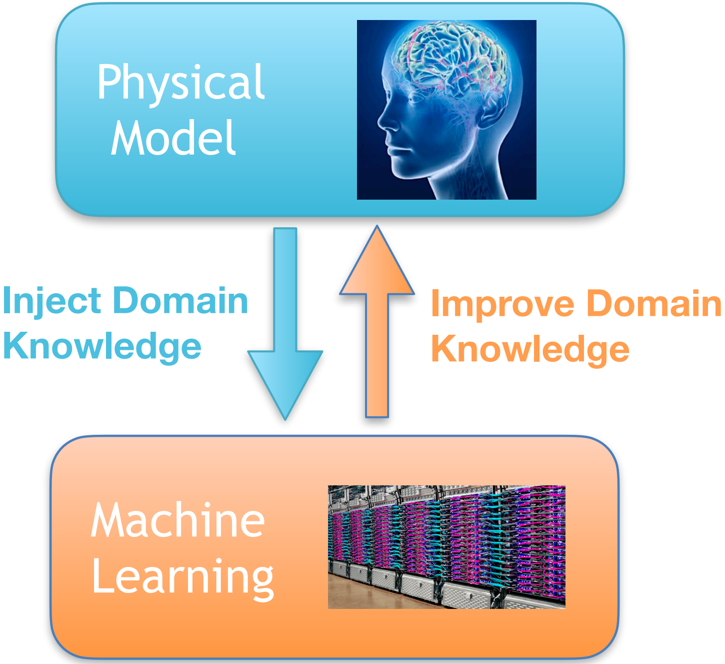
Future Work

1. Benchmark efficient models ensuring all the symmetries simultaneously.

2. More principles and guidelines to data exploration.

3. Model closure uncertainty.

4. Improve stability when coupling the ML-based closure to the reduced-order PDE.



Thank you for your attention!

FLATIRON INSTITUTE
Center for Computational Mathematics