Turbulence Closure Modeling with Differentiable Physics

JULY 28, 2022

Venkat Viswanathan*, Varun Shankar, Vedant Puri

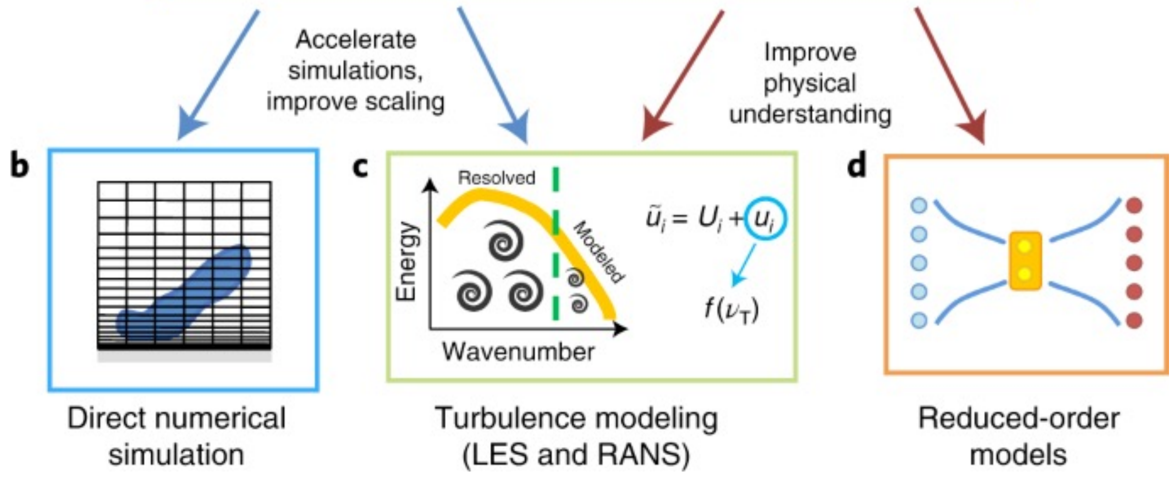
*venkvis@cmu.edu

Agenda

1. Motivation and background
2. Differentiable Neural Closures
   - Learning more accurate closure dynamics
3. Turbulence predictions with equivariant GNNs
   - Reducing time to solution for complex flows
4. Conclusions and future directions

State of ML in CFD

Machine Learning



Solver integration

CNN surrogates

Super-resolution

Open-loop turbulence corrections

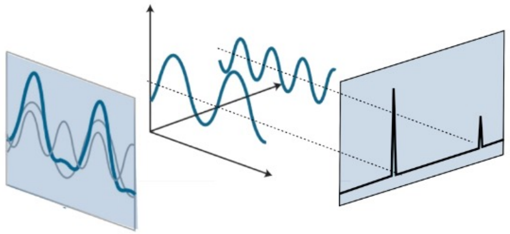
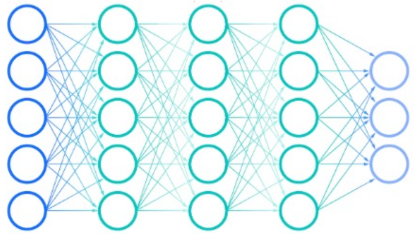
Equation discovery

Operator learning

Vinuesa, R., Brunton, S.L. "Enhancing computational fluid dynamics with machine learning". *Nat Comput Sci* **2,** 358–366 (2022)
Kochkov, D. et al. "Machine learning–accelerated computational fluid dynamics". Proceedings of the National Academy of Sciences 118. 21(2021): e2101784118.
Ling, Julia, et al. "Reynolds Averaged Turbulence Modelling Using Deep Neural Networks with Embedded Invariance." Journal of Fluid Mechanics, vol. 807, 2016, pp. 155–166., doi:10.1017/jfm.2016.615.
Fukami, Kai, et al. "Sparse Identification of Nonlinear Dynamics with Low-Dimensionalized Flow Representations." Journal of Fluid Mechanics, vol. 926, 2021, p. A10., doi:10.1017/jfm.2021.697.
Zongyi Li, et al. "Fourier Neural Operator for Parametric Partial Differential Equations." (2021).
Shankar, V., et al. "Validation and parameterization of a novel physics-constrained neural dynamics model applied to turbulent fluid flow." (2021).

Approximation Theory and Curse of Dimensionality

$$f = \tilde{f} + \mathcal{O}(h)$$

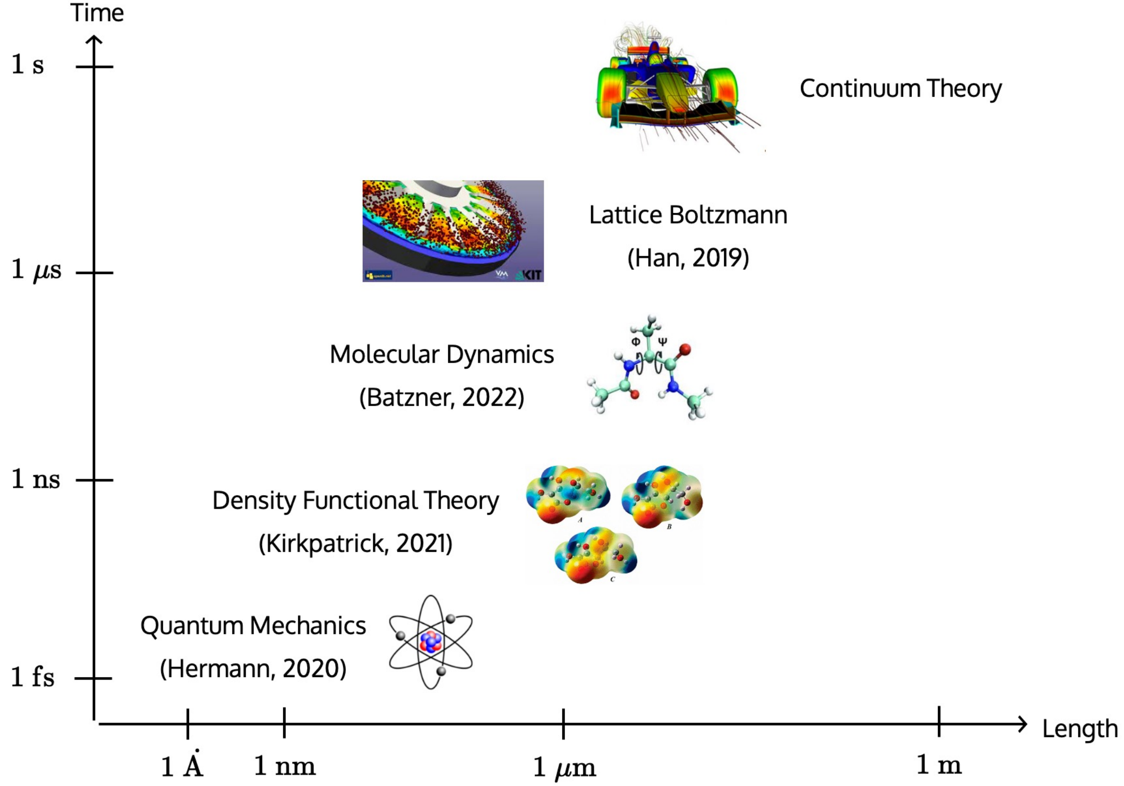
Orthogonal Functions	Deep Neural Networks
$\tilde{f} = \Sigma_{i=1}^{N} f_i \phi_i(x)$ 	$\tilde{f} = W_L \circ \sigma(W_{L-1} \circ (\ldots(\sigma W_0 x)) + b_{L-1})$ 
$h \sim N^{-c/d}$	$h \sim 1/N$ (for 2 layer networks)

Curse of dimensionality

Dimension independent

Weinan E, . "Machine Learning and Computational Mathematics". Communications in Computational Physics 28. 5(2020): 1639–1670.

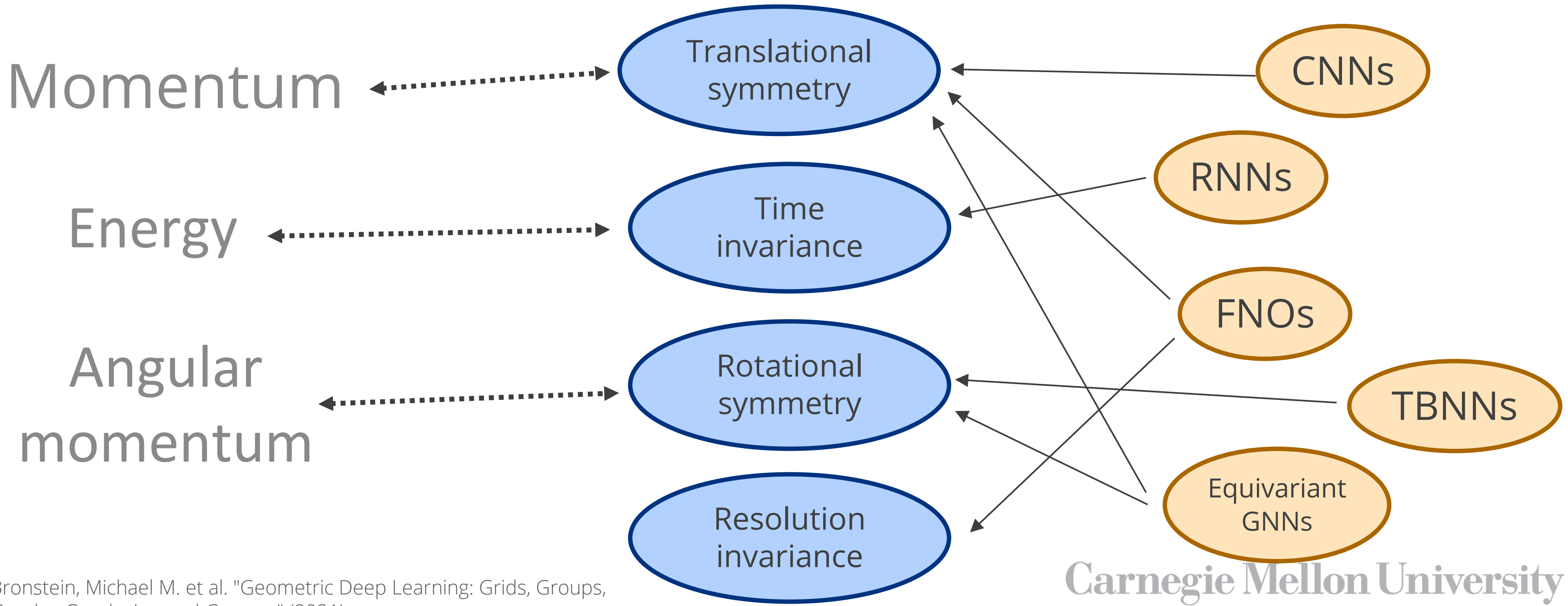
Machine Learning in Physics



Hermann, J., et al. "Deep-neural-network solution of the electronic Schrödinger equation". Nat. Chem. 12, 891–897 (2020),
Kirkpatrick, J., et al. "Pushing the frontiers of density functionals by solving the fractional electron problem". Science 374. 6573(2021): 1385-1389.
Batzner, S., et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. Nat Commun 13, 2453 (2022).
Jiequn Han, et al. "Uniformly accurate machine learning-based hydrodynamic models for kinetic equations". Proceedings of the National Academy of Sciences 116. 44(2019): 21983-21991.

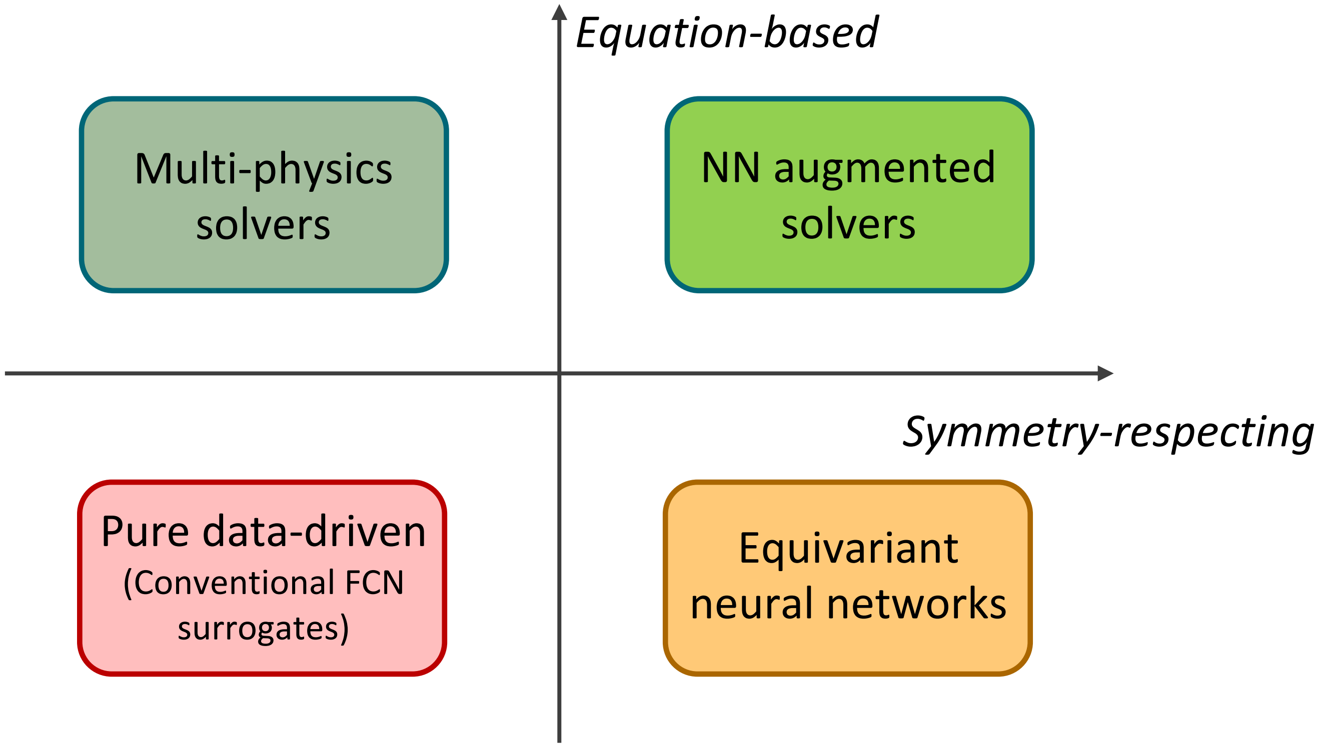
Conservation laws as embedded symmetries

Conservation laws ⟺ Symmetries (Noether's theorem)



Bronstein, Michael M. et al. "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges." (2021).

Data-driven physics represented as symmetries and equations



Differentiable Neural Closures

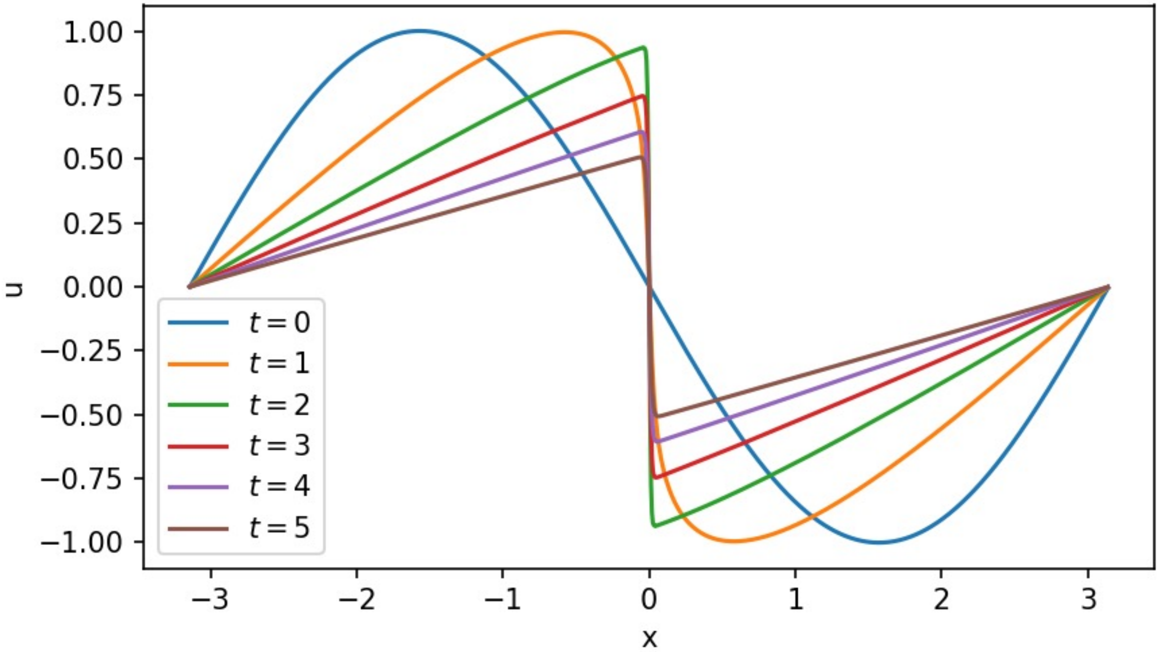
Carnegie Mellon University

1D Viscous Burgers' Equation

$$\frac{\partial}{\partial t} u(x,t) = \nu \frac{\partial^2}{\partial x^2} u - u \frac{\partial}{\partial x} u$$

$$\bar{u}_t = \nu \bar{u}_{xx} - \bar{u}\bar{u}_x + \eta$$

$$\eta = \frac{\partial}{\partial x}\left(\nu_T \frac{\partial}{\partial x} \bar{u}\right)$$



Adding structure to the model increases interpretability

~~$\eta = f_\theta(\bar{u}; x, t)$~~

$$\dot{\bar{u}} = \nu\nabla^2\bar{u} - \bar{u}\nabla\bar{u} + \nabla(\nu_T\nabla\bar{u})$$

$$\dot{\nu}_T = \alpha_\theta(\bar{u}_x; x, t)\cdot\nu\nabla^2\nu_T - \beta_\theta(\bar{u}_x; x, t)\cdot\bar{u}\nabla\nu_T$$

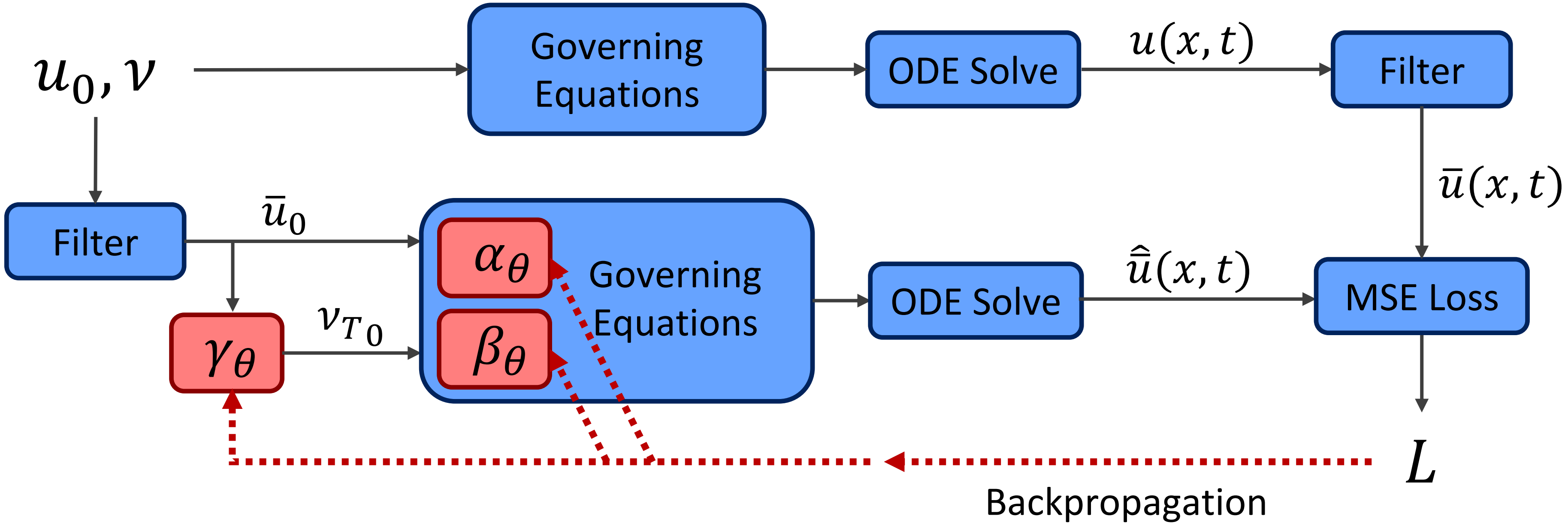
$$\nu_T(x, 0) = \gamma_\theta(\bar{u}_0, \nu; x)$$

*NNs operate in Fourier space

The need for a differentiable solver

- We're learning a dynamical equation
- The simpler approach
  - Generate high-fidelity solutions
  - Filter solutions and compute closure term
  - Learn the function mapping $\bar{u}$ to the closure term
- Want to learn a-posteriori velocity → evolve coupled system together during training

The need for a differentiable solver

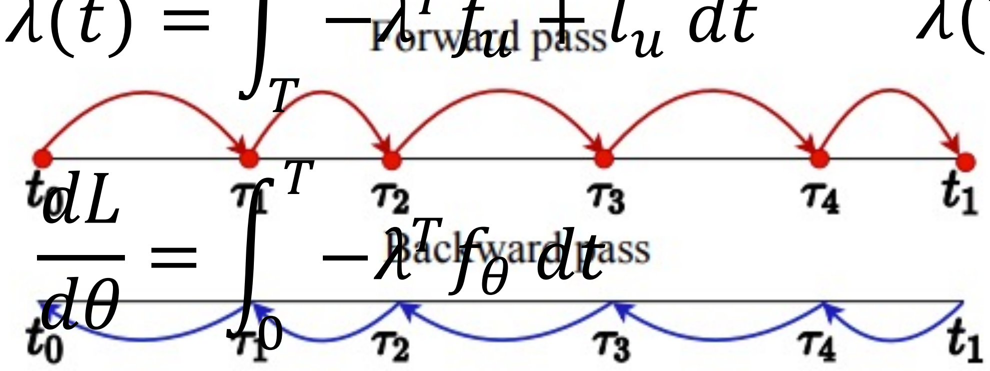


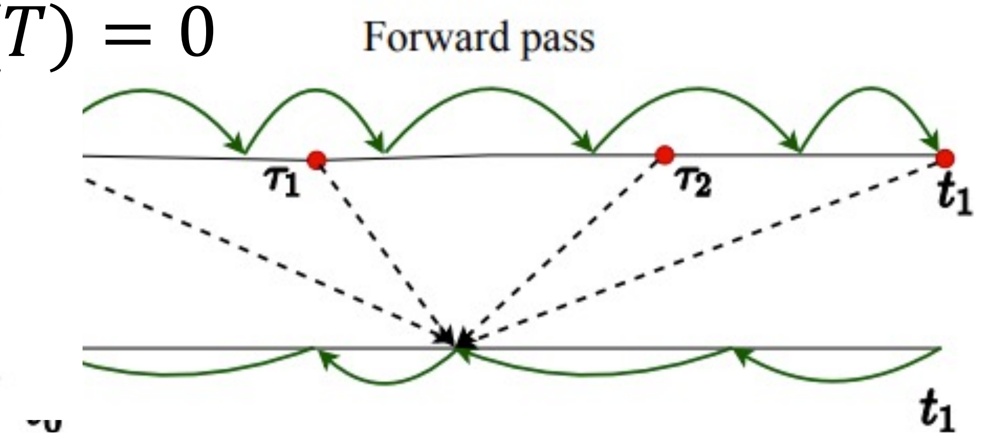
Adjoint method

$$\dot{u} = f(u,\theta) \qquad L(u) = \int_0^T l(u,t)\,dt$$

$$\lambda(t) = \int_T^t -\lambda^T f_u + l_u \, dt \qquad \lambda(T) = 0$$

$$\frac{dL}{d\theta} = \int_0^T -\lambda^T f_\theta \, dt$$



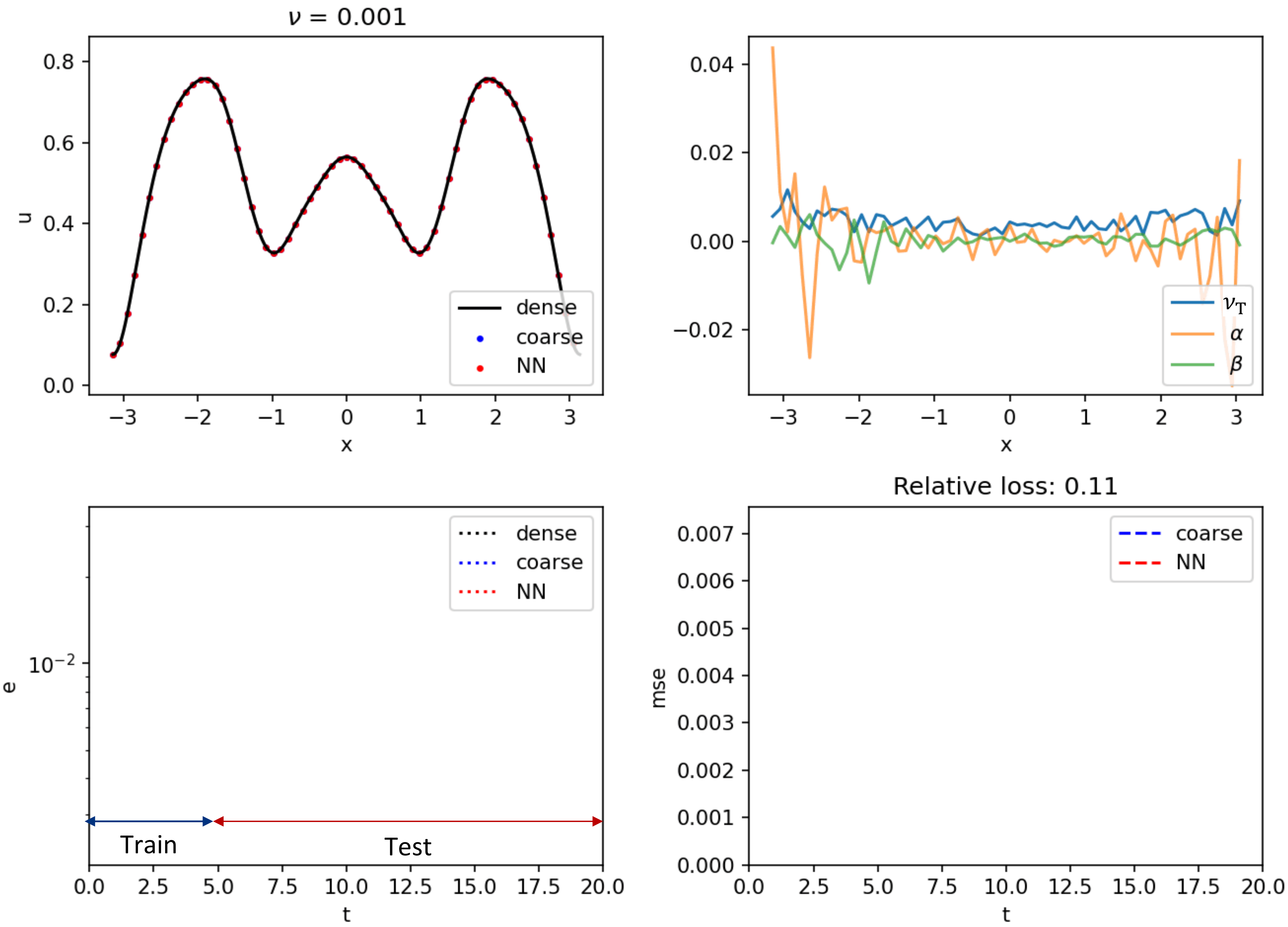


Learning methodology

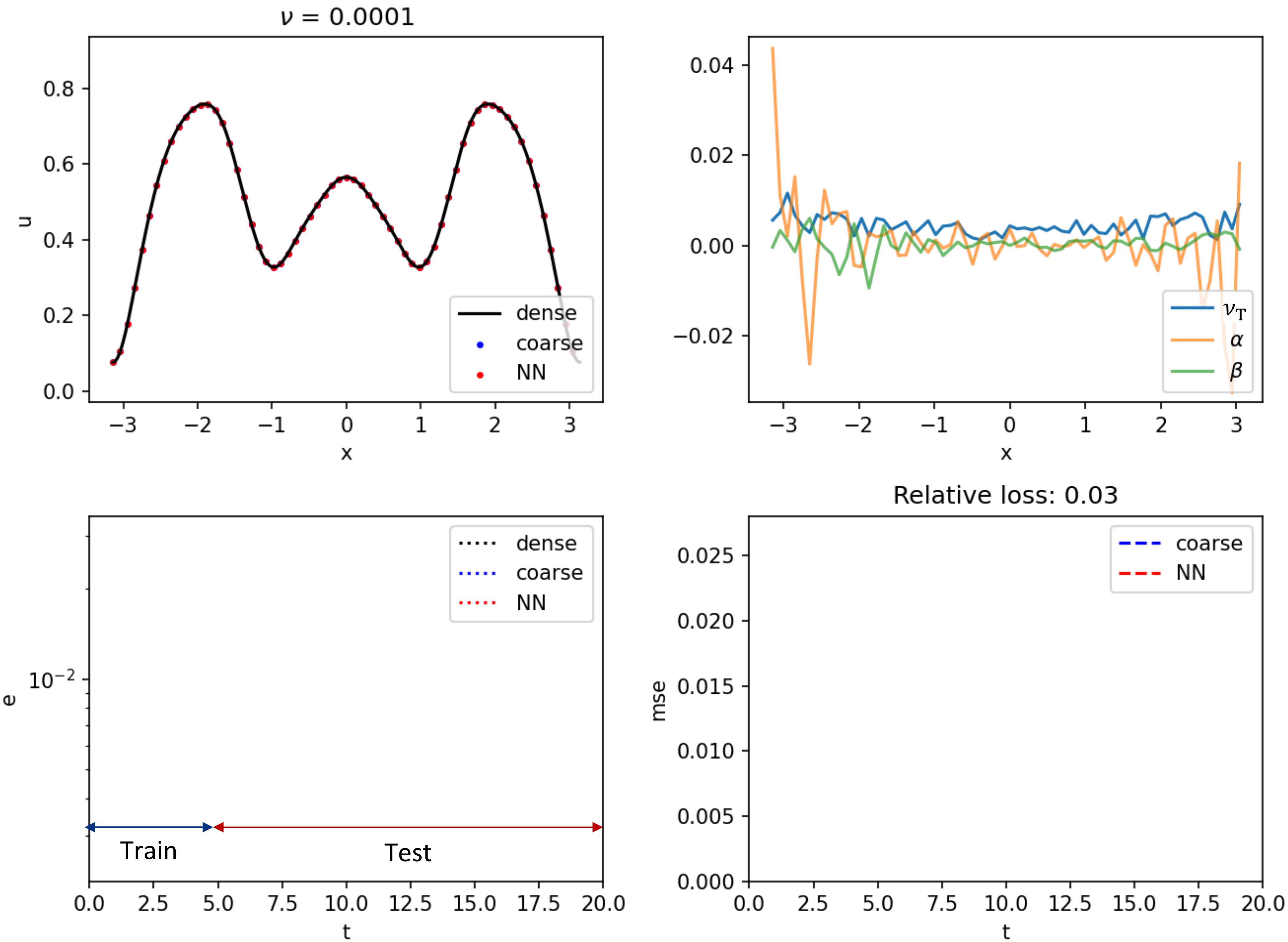
- Desire to learn a one-equation closure transport model for Burgers' equation
  - Neural network approximates ν_T IC and convection/diffusion coefficients of the transport equation in Fourier space
- Data generated on fully-resolving dense grid
- ODE system integrated to produce approximate velocity profiles
- Loss is backpropagated via adjoint method to compute NN parameter gradients

Results

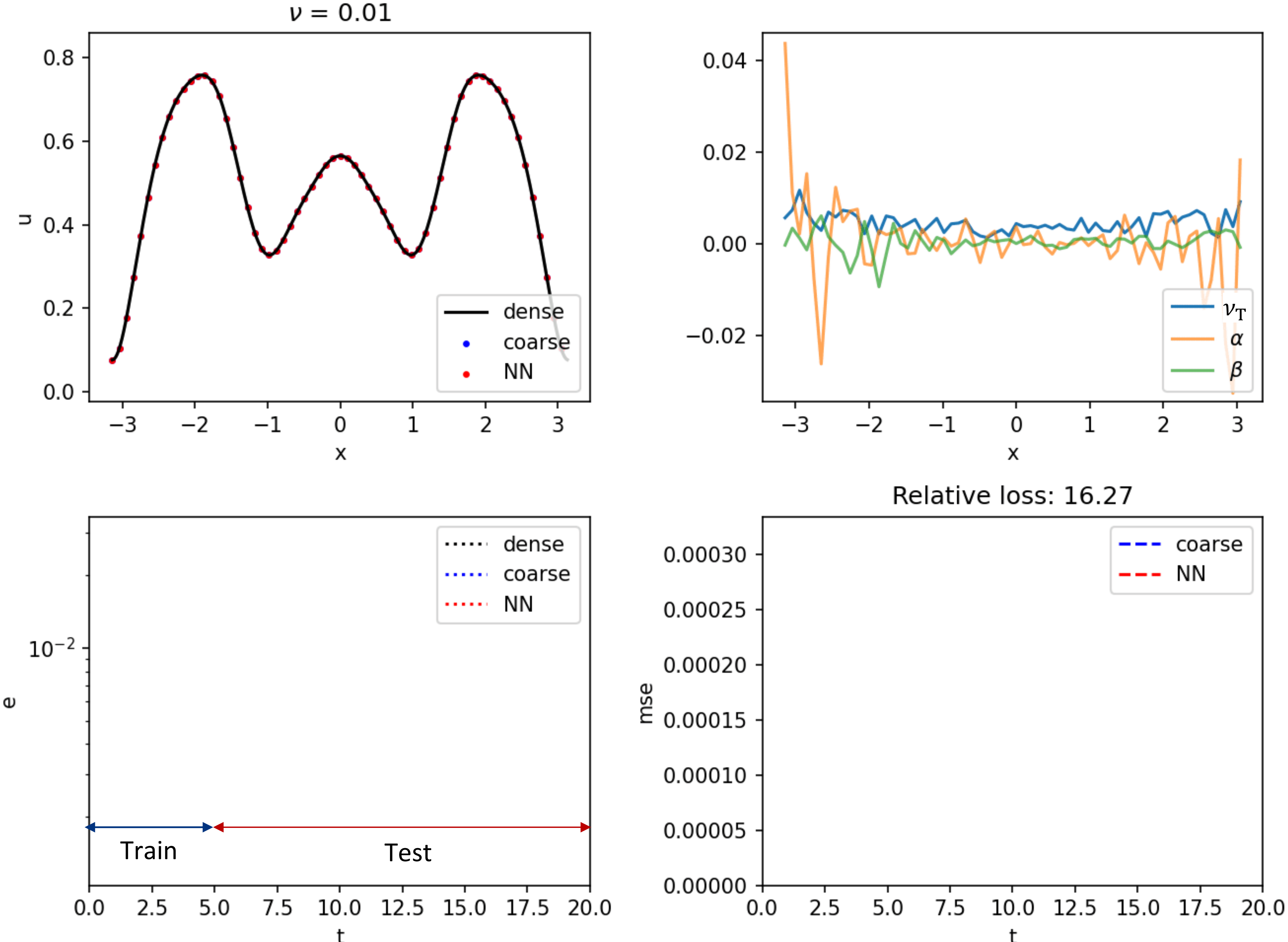
- Model training:
  - Random initial condition
  - $t = 0 \text{ to } 5$
  - $\nu = 5\times10^{-2} \text{ to } 5\times10^{-4}$



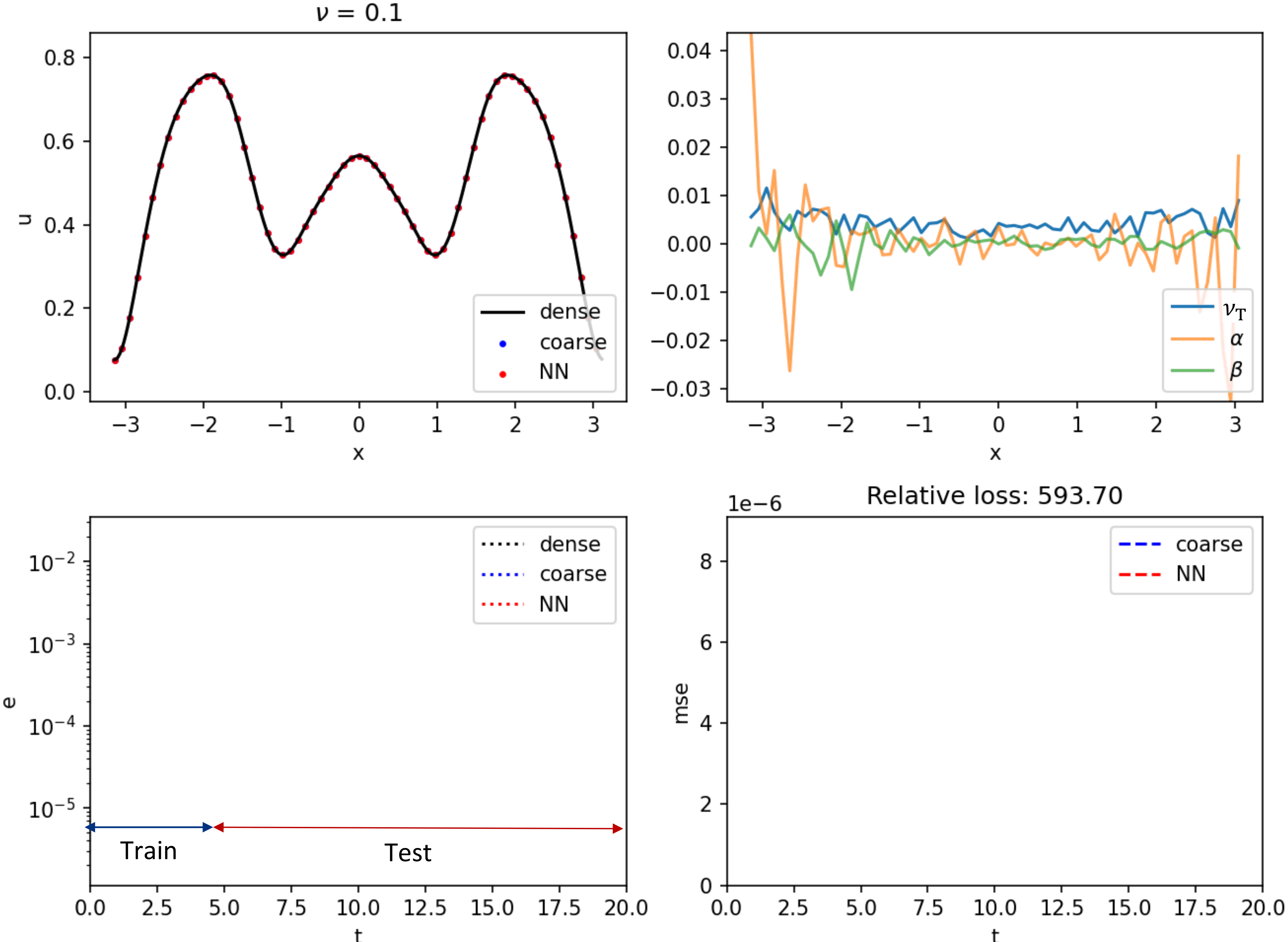
ν = 0.001
u
x
dense
coarse
NN
x
νT
α
β
Relative loss: 0.11
e
t
Train
Test
dense
coarse
NN
mse
t
coarse
NN



ν = 0.0001
u
x
dense
coarse
NN
ν_T
α
β
Relative loss: 0.03
e
t
Train
Test
mse



ν = 0.01
u
x
dense
coarse
NN
ν_T
α
β
x
e
t
dense
coarse
NN
Train
Test
Relative loss: 16.27
mse
t
coarse
NN


ν = 0.1
u
x
dense
coarse
NN
ν_T
α
β
e
t
Train
Test
Relative loss: 593.70
1e−6
mse

Discussion

- 1-eqn learned transport model for eddy viscosity effective for sub-grid modeling of Burgers' equation
- Interpretable model → improve physical understanding
  - Stationary eddy viscosity for time-varying system
- Model fails at high viscosities → can we learn to "switch off?"
- Scaling to higher dimensions, Navier-Stokes

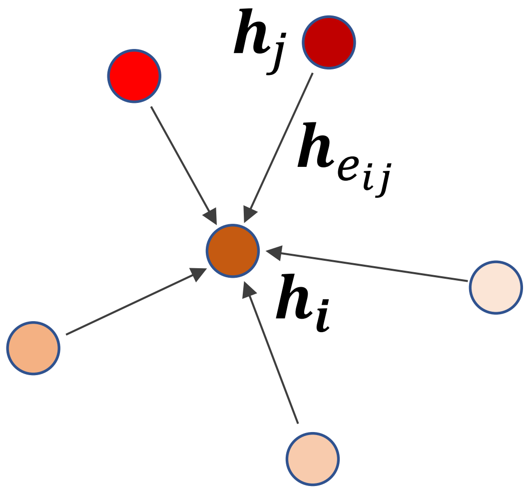
Equivariant Turbulence Predictions

Carnegie Mellon University

Scaling up introduces challenges

Issue	Solver not differentiable	Expensive forward simulations	Unstructured grids, FCNs intractable	Equiv. for vector valued fields
Solution	a-priori learning of eddy viscosity	Learn steady-state quantities	Graph neural networks	Tensor-field equivariant NN

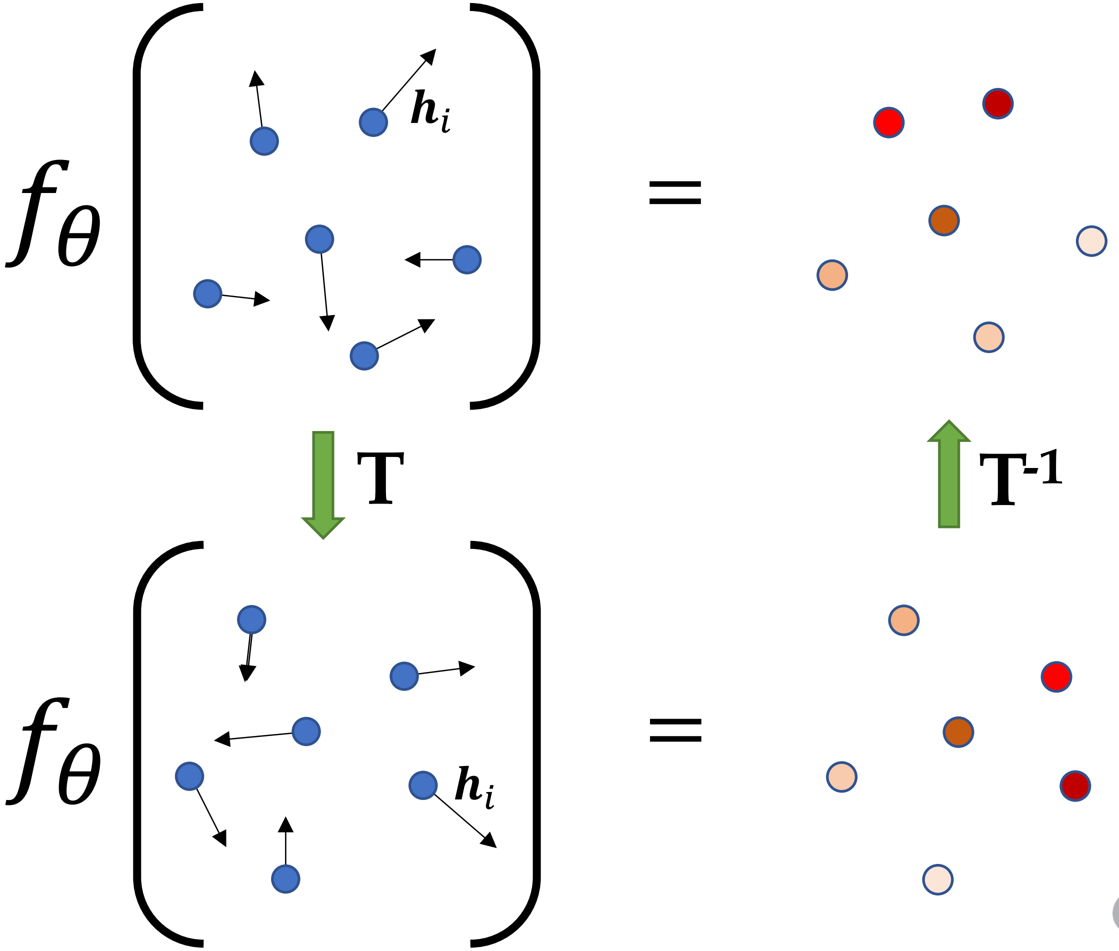
Graph Message-Passing Neural Networks



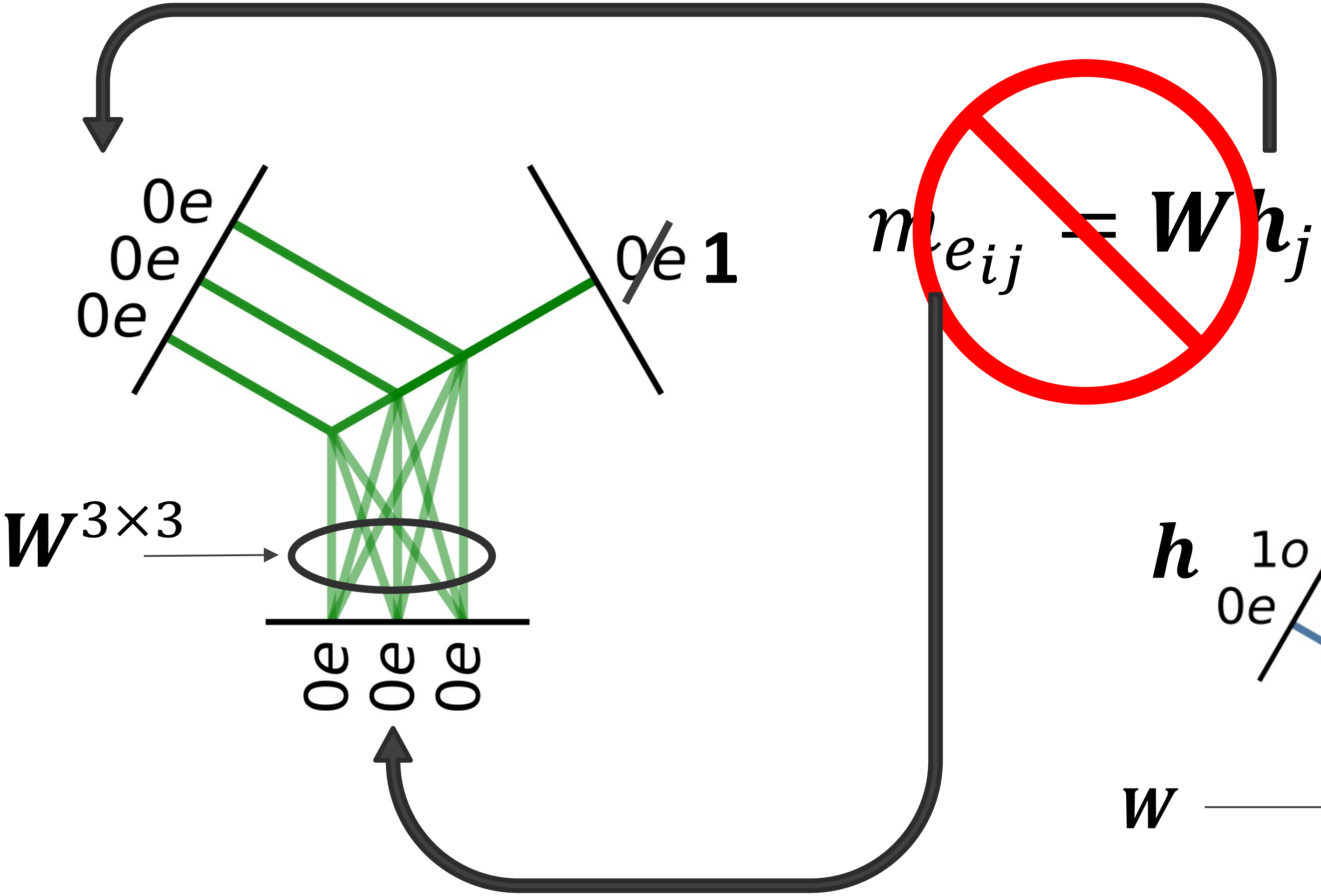
$$\boldsymbol{h}_{e_{ij}}^{n+1} = f_\theta(\boldsymbol{h}_{e_{ij}}^{n}, \boldsymbol{h}_i^n, \boldsymbol{h}_j^n)$$

$$\boldsymbol{h}_i^{n+1} = g_\theta\left(\sum_{e\in\mathcal{N}} \boldsymbol{h}_{e_{ij}}^{n+1}, \boldsymbol{h}_i^n\right)$$

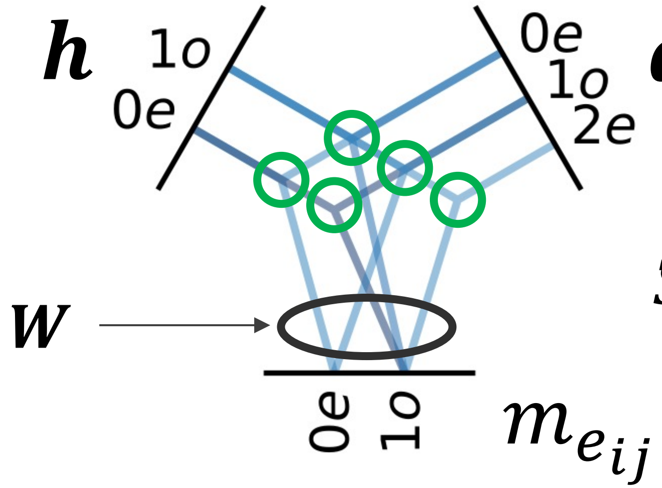
Equivariance



Tensor Products

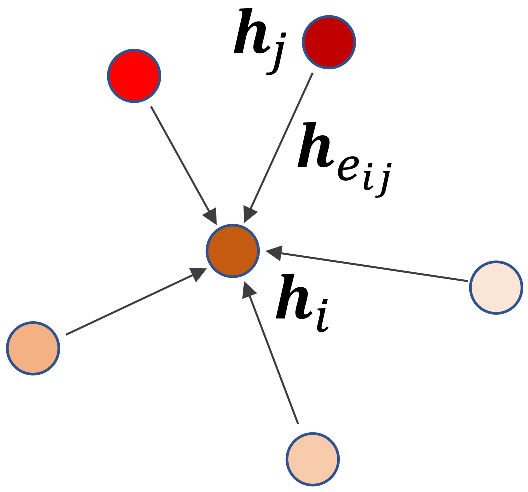


$$m_{e_{ij}} = \boldsymbol{h}_j \otimes^W \boldsymbol{a}_{ij}$$



 Tensor product visualization from e3nn package: https://e3nn.org/

Equivariant GNNs

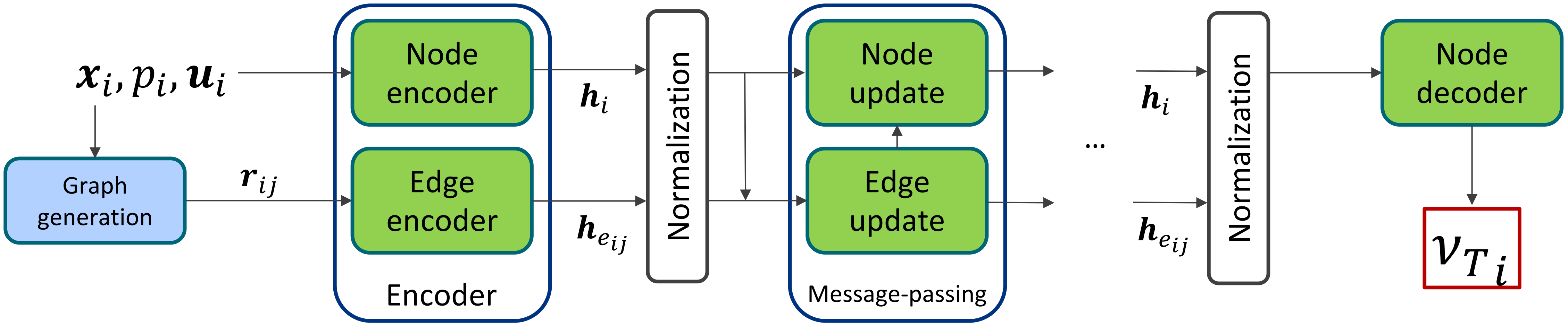


$$\boldsymbol{h}_{e_{ij}}^{n+1} = f_\theta(\boldsymbol{h}_{e_{ij}}^n, \boldsymbol{h}_i^n, \boldsymbol{h}_j^n)$$

$$f_\theta = [\boldsymbol{h}_{e_{ij}}, \boldsymbol{h}_i, \boldsymbol{h}_j] \otimes^{\boldsymbol{W}(r_{ij})} Y_{sh}(\boldsymbol{r}_{ij})$$

$$\boldsymbol{W}(r_{ij}) = MLP(\|\boldsymbol{r}_{ij}\|)$$

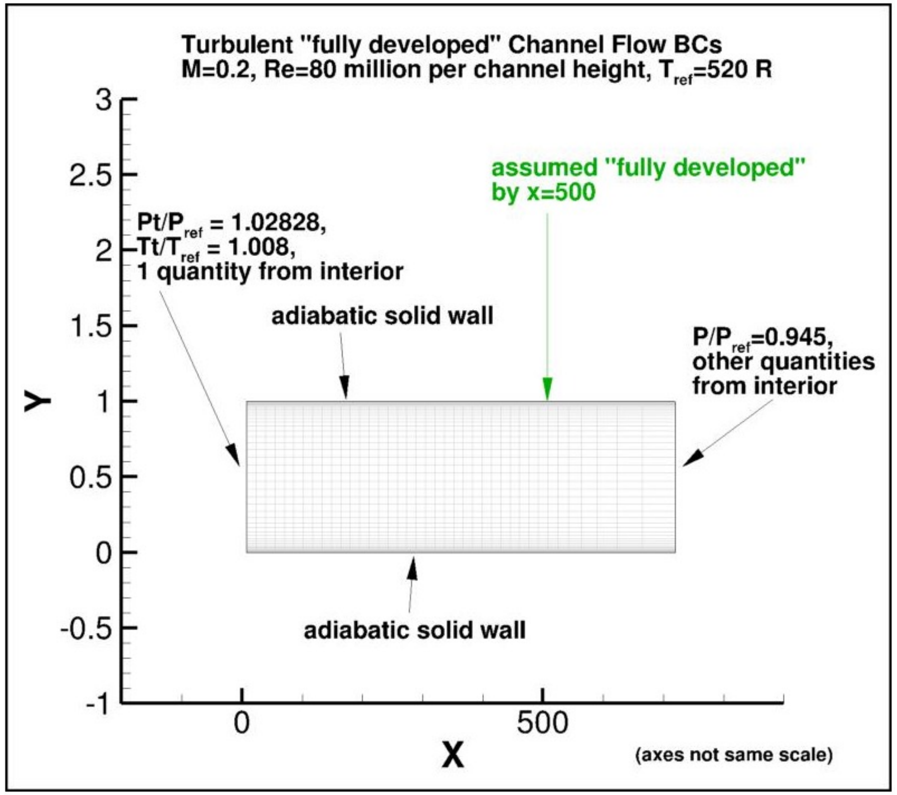
Architecture

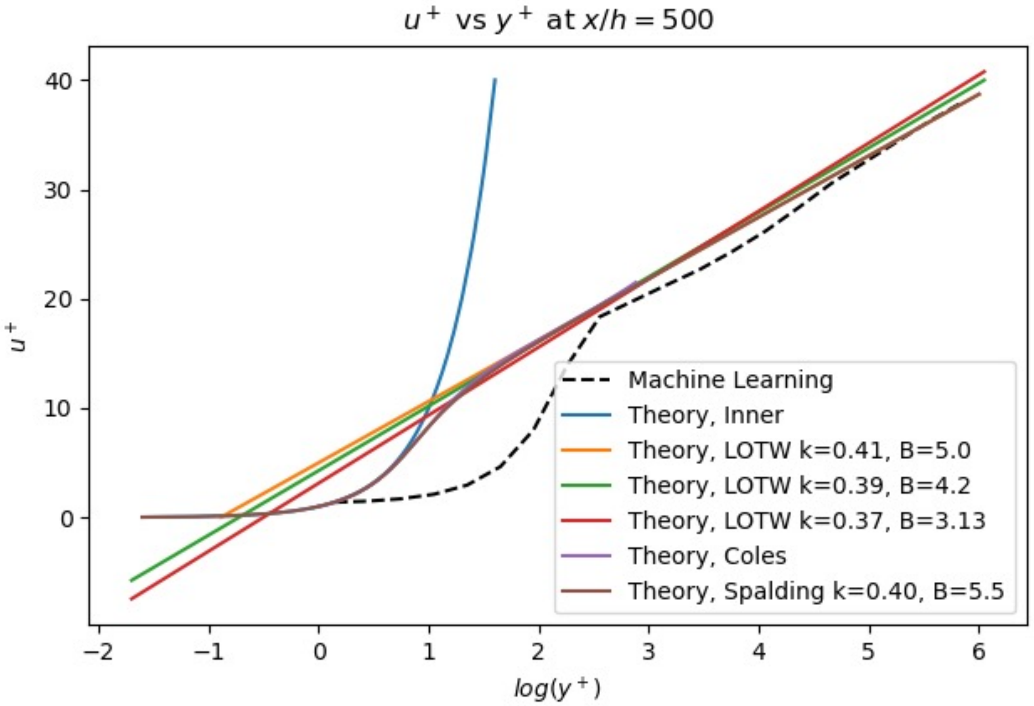


Learning Methodology

- Desire to learn steady-state eddy viscosity from mesh point cloud and initial pressure/velocity fields
  - Neural network approximates ν_T using equivariant graph network
- Data generated using Spalart-Allmaras turbulence model
- Approximate ν_T field used in incompressible solver
  - Only pressure/velocity equations solved
- A-posteriori analysis of solution fields

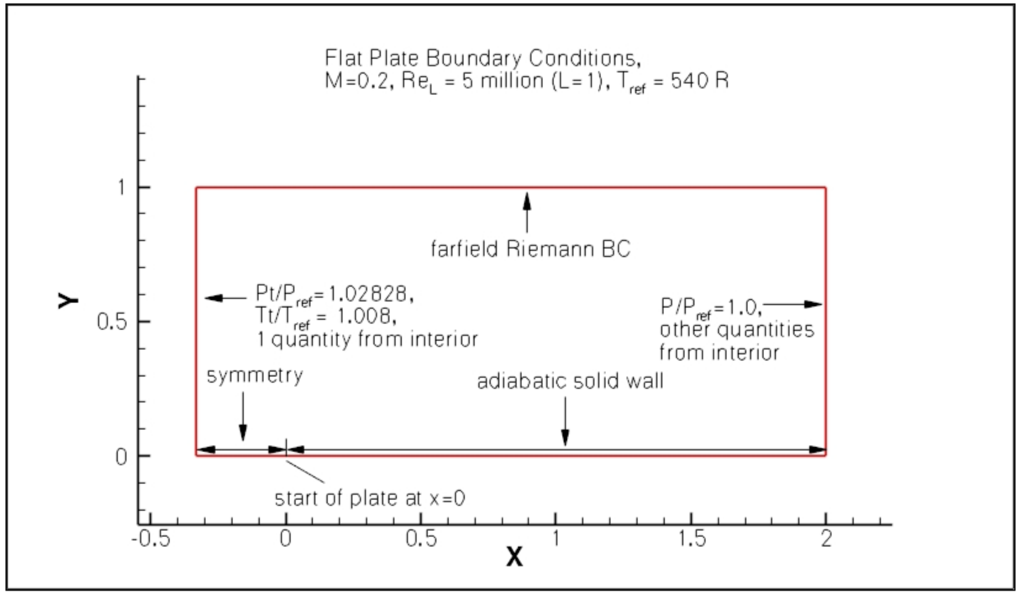
2D Fully Developed Channel Flow (Re_h=80m)

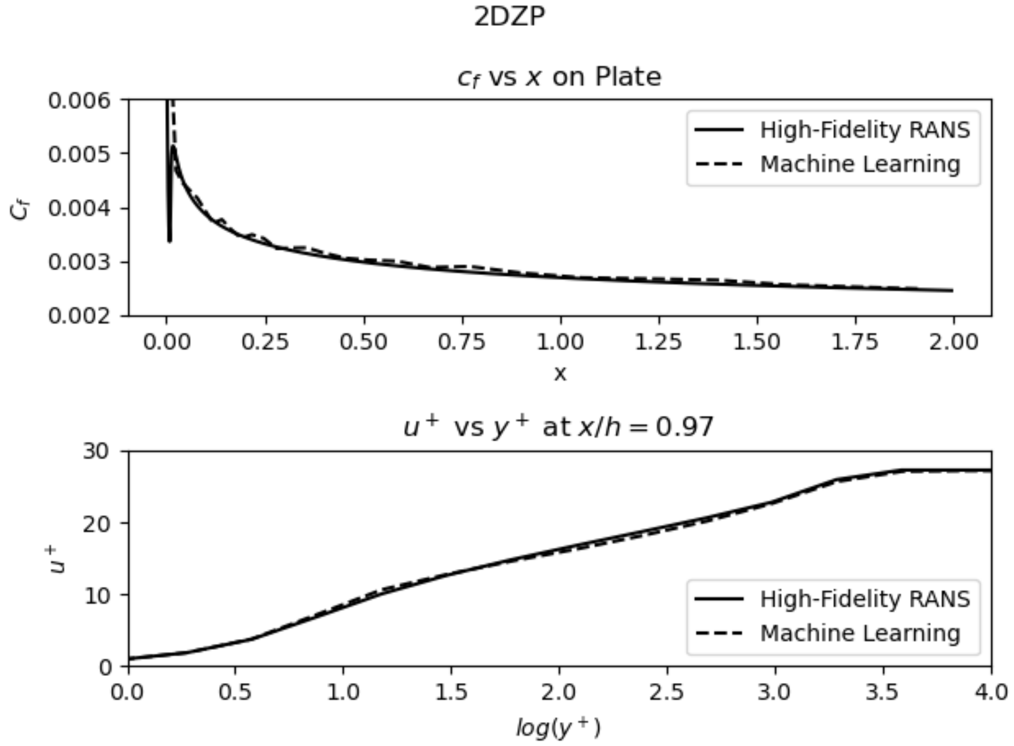




https://turbmodels.larc.nasa.gov/channelflow_val.html

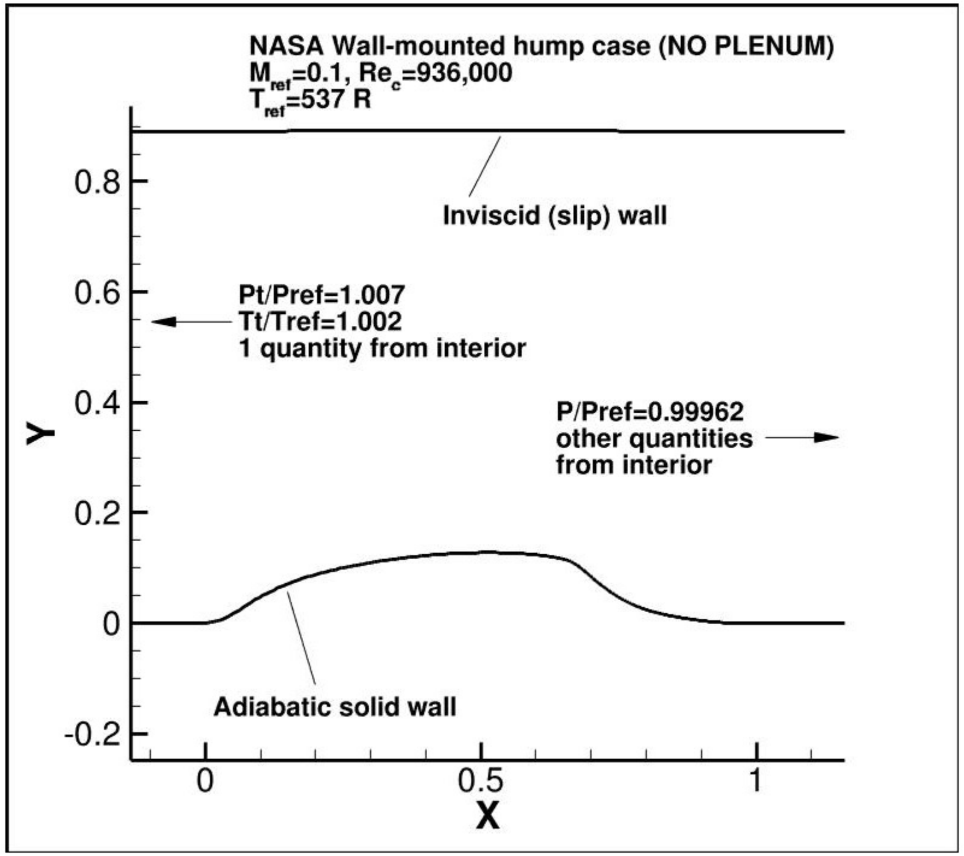
2D Zero Pressure Gradient Flat Plate (Re_x=5m)





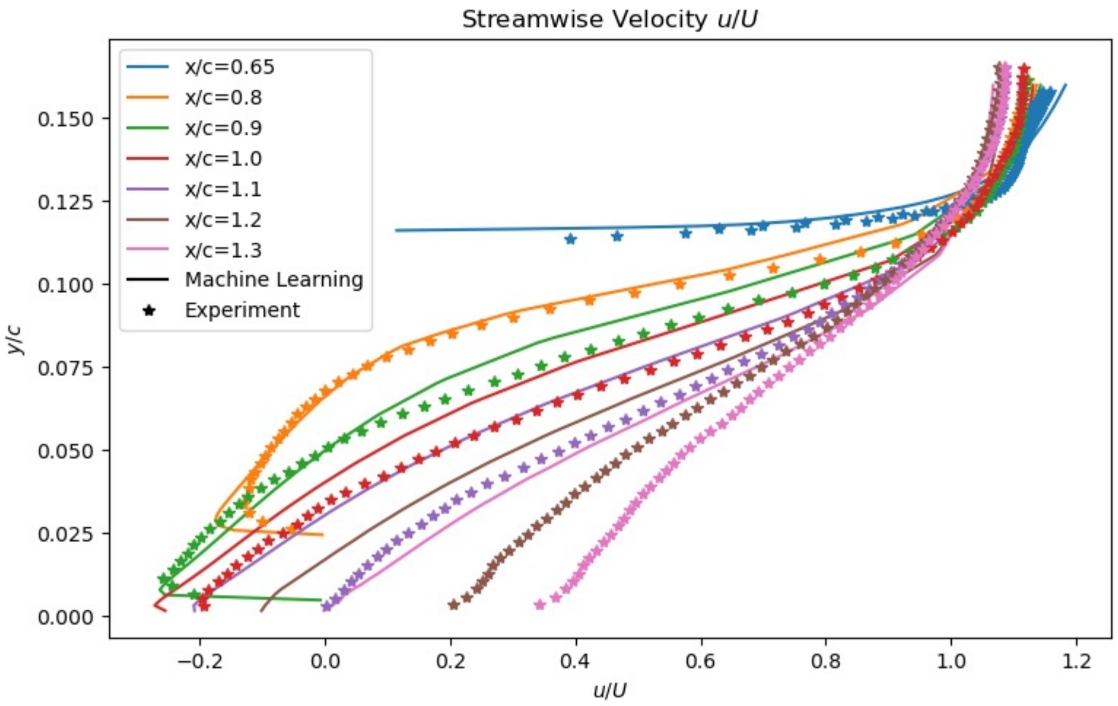
https://turbmodels.larc.nasa.gov/flatplate_val.html

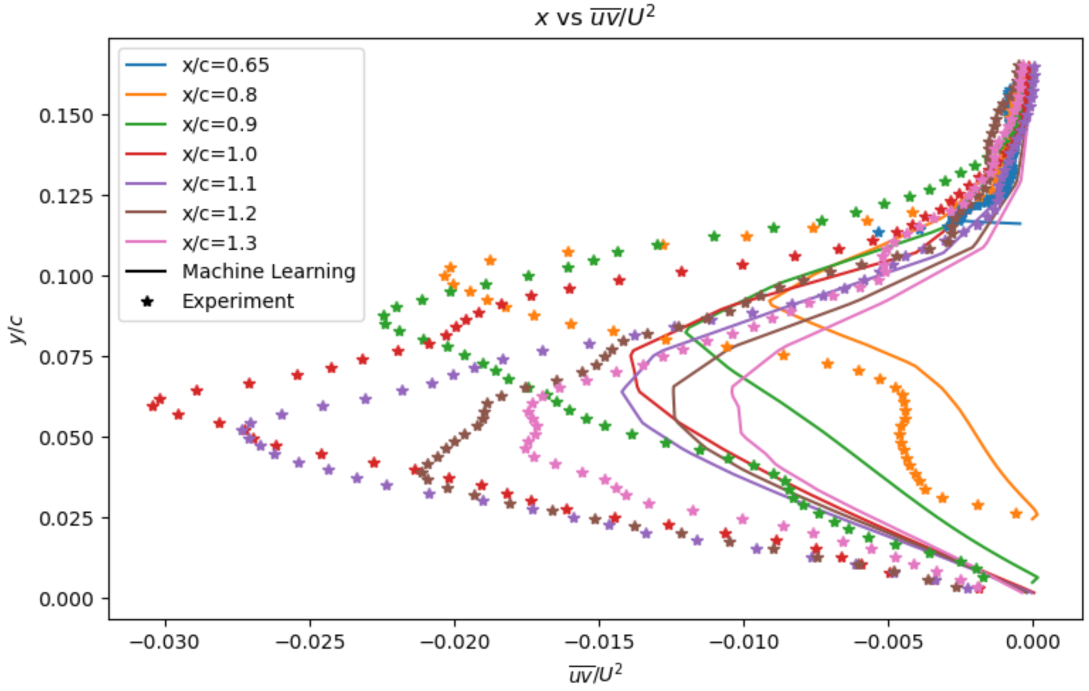
2D Wall Mounted Hump Re_c=936k



 https://turbmodels.larc.nasa.gov/nasahump_val.html

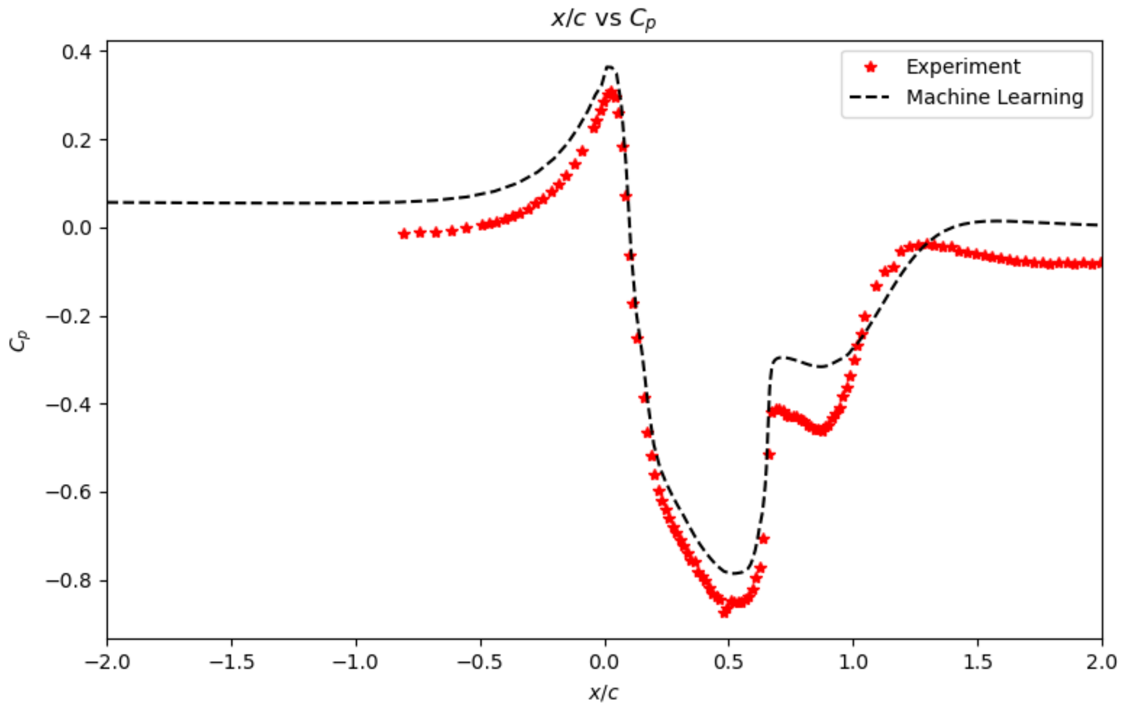
2D Wall Mounted Hump Re_c=936k

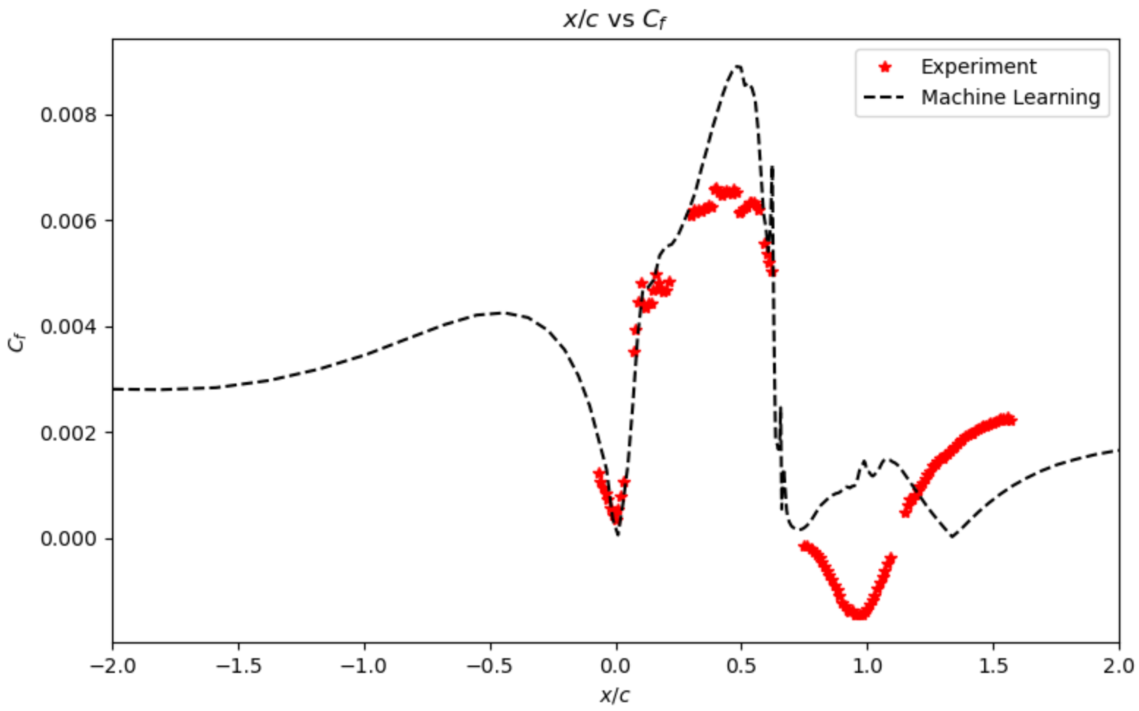




https://turbmodels.larc.nasa.gov/nasahump_val.html

2D Wall Mounted Hump Re_c=936k





https://turbmodels.larc.nasa.gov/nasahump_val.html

Conclusions and future directions

- Differentiable physics → more interpretable data-driven methods
- Symmetry-respecting architectures as best candidates for learning in the low-data regime
- Can we make strides towards fully differentiable N-S solvers?
- Can we leverage the advantages of DNN scaling and equivariant architectures to learn more robust turbulence models?

Acknowledgements

Mobility21
A USDOT NATIONAL UNIVERSITY TRANSPORTATION CENTER

ORACLE
for Research

NSF

XSEDE