

Deep Learning Closure Models for Large-eddy Simulation

Justin Sirignano

University of Illinois at Urbana-Champaign

Presentation based upon research with Jonathan MacArt (University of Notre Dame) and Jonathan Freund (University of Illinois at Urbana-Champaign).

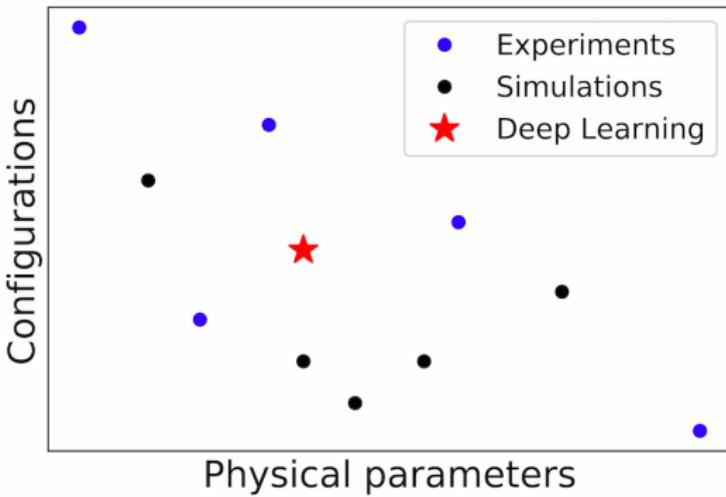
February 11, 2020

Deep learning (DL) in Scientific Modeling

- ▶ Although the physics are typically known, they cannot be **resolved** on a grid which is computationally tractable.
- ▶ Example: Navier-Stokes equations in turbulent flows
- ▶ **Engineering** requires reduced-order models which:
 - can be solved on a **coarse grid** at low computational cost,
 - but still **accurately represent** the physics.

Ultimate goal: accelerate analysis and design

- ▶ Train on datasets across a variety of physical parameters and configurations
- ▶ Predictions for new physical parameters and configurations



Filtered Navier-Stokes equations

► Direct Numerical Simulation (DNS)

- Resolves the exact physics
- Computationally expensive
- Computationally intractable for many real-world engineering problems

► Large Eddy Simulation (LES)

- Filter Navier-Stokes equations to remove the small scales
- Solve on coarse grid at low computational cost
- Unclosed term which must be modeled
- Finite difference error due to coarse grid

Overview

- ▶ Evaluate sources of error in LES
- ▶ Derivation of an adjoint PDE for optimization of the deep learning closure model
- ▶ Develop scalable multi-GPU code to solve adjoint PDEs
- ▶ Train and test on DNS datasets

DNS datasets

We have generated several DNS datasets to train and test on:

- ▶ Decaying isotropic turbulence across several viscosities and initial pseudo-integral scale lengths (1024^3 grid at Re 3,300)
- ▶ Decaying isotropic turbulence (2048^3 grid at Re 7,805)
- ▶ Turbulent jet ($1024 \times 1280 \times 768$ grid at Re 6,000)
- ▶ Turbulent double jet ($1024 \times 1792 \times 768$ grid at Re 6,000)

Out-of-sample *a posteriori*: Isotropic Turbulence

- ▶ Trained on filtered DNS data at $\text{Re } 3,300$ (1024^3 grid)
- ▶ Tested on $\text{Re } 7,805$

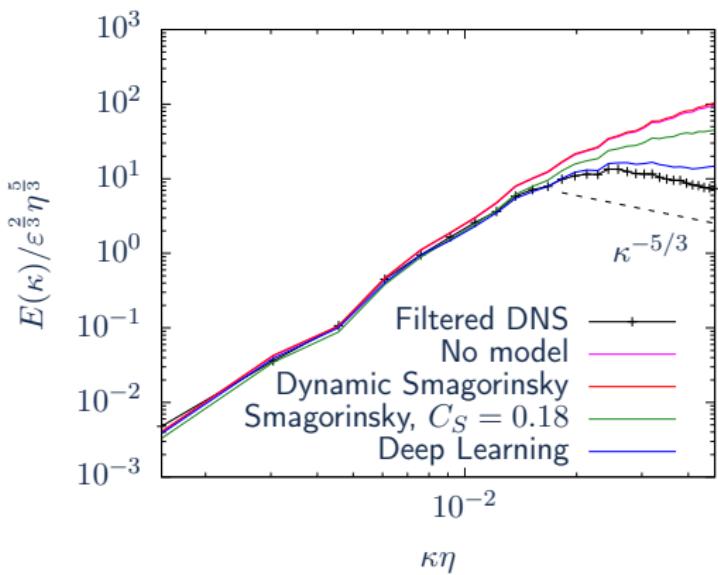


Figure: Resolved Kinetic Energy spectrum for different closure models. The DNS grid was $N = 2048^3$ and the filter-to-grid ratio is $32\times$. Shown for $t/t_0 = 0.28$.

Out-of-sample *a posteriori*: Isotropic Turbulence

Vorticity magnitude ($\times 10$) in decaying isotropic turbulence, normalized by the eddy-turnover time scale $t_{\ell,0} = k/\varepsilon$:

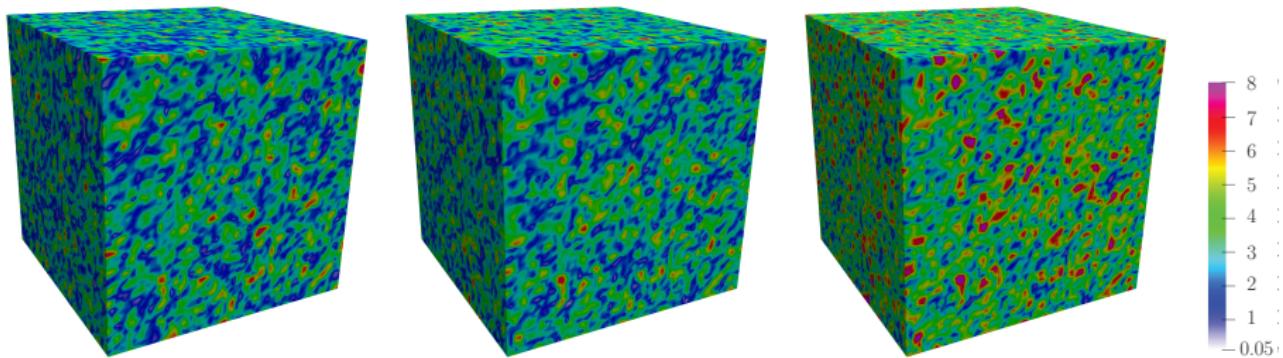
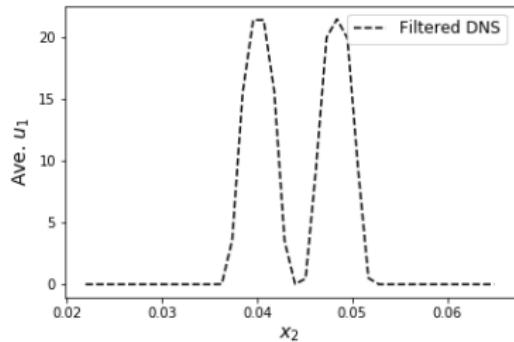
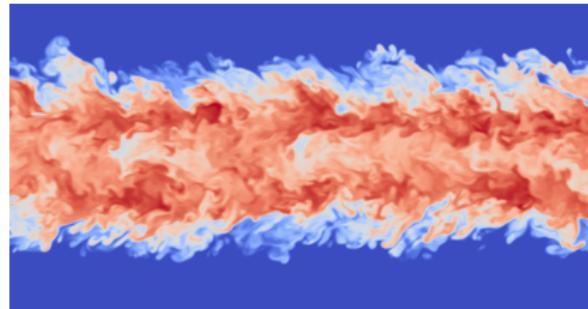


Figure: **Left:** Filtered DNS. **Middle:** LES with deep learning closure model. **Right:** Dynamic Smagorinsky.

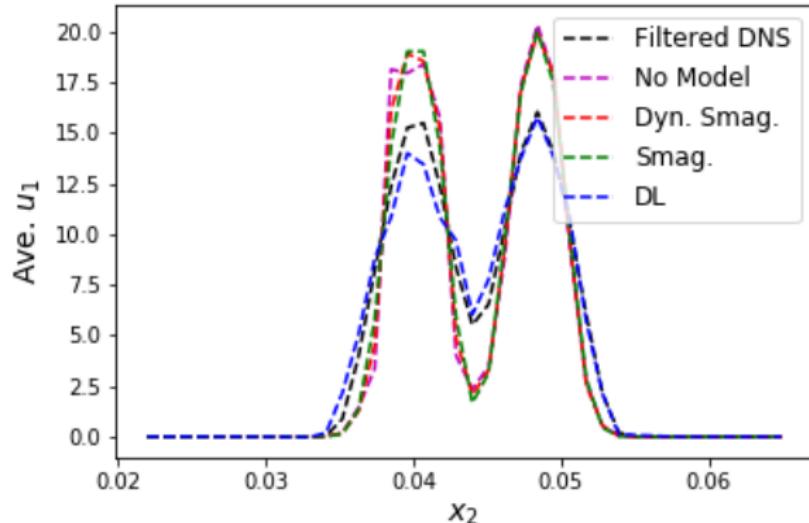
Out-of-sample *a posteriori*: Double Jet

- ▶ DL model trained on filtered DNS from single jet
- ▶ Tested on double jet
- ▶ Two parallel jets that eventually merge



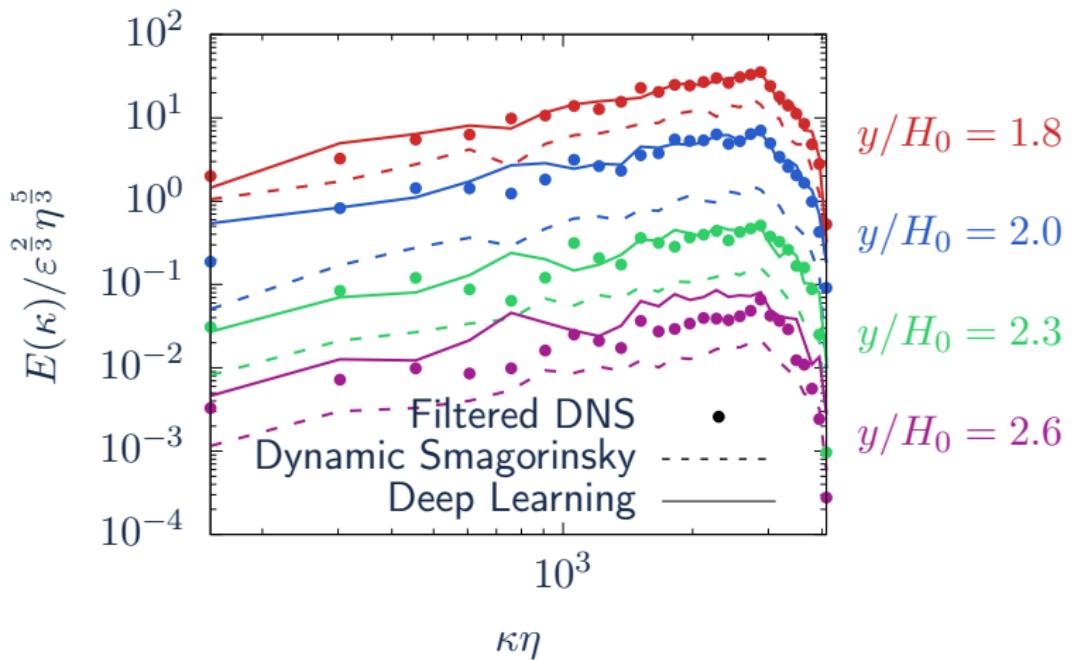
Out-of-sample *a posteriori*: Double Jet

- ▶ DL model trained on filtered DNS from single jet
- ▶ Tested on double jet



Out-of-sample *a posteriori*: Double Jet

- DL model trained on filtered DNS from single jet
- Tested on double jet



Out-of-sample *a posteriori*: Double Jet

- ▶ DL model trained on filtered DNS from single jet
- ▶ Tested on double jet

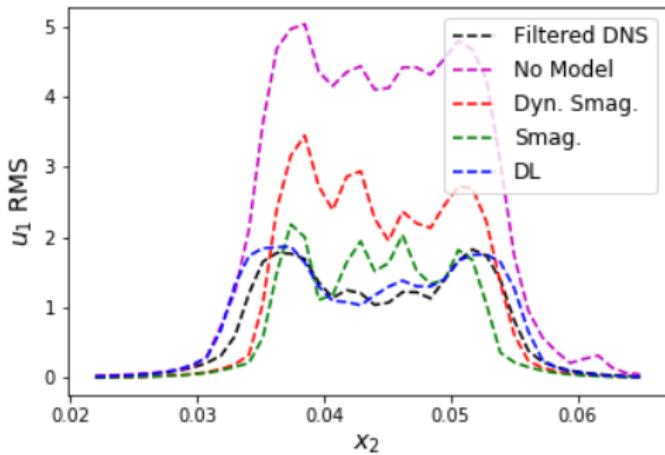


Figure: Root-mean-square (RMS) deviation of u_1 from its mean profile.

Related Literature on DL for scientific computing

- ▶ DL for PDEs: Menon et al. (2010); Raissi, Perdikaris, and Karniadakis (2019); Brunton, Noack, Koumoutsakos (2019); Popov, Buchta, Anderson, Massa, Capecelatro, Freund (2019); Berg and Nystrom (2017); Jentzen, Weinan E. et al. (2018-2020); Brenner (2019); and Sirignano and Spiliopoulos (2018).
- ▶ DL for Reynolds-averaged Navier-Stokes equations (RANS): Ling, Jones, and Templeton (2016); Ling, Kurzawski, and Templeton (2016); and Holland, Baeder, Duraisamy (2019)
- ▶ DL for Large Eddy Simulation (LES): Wang, Luo, Li, Tan, and Fan (2018)
 - Decouple estimation of the unclosed term from the PDE and then substitute DL model into PDE, while we optimize over the entire PDE.

Incompressible Navier-Stokes equations

$$\begin{aligned}\frac{\partial u_i}{\partial t} &= -\frac{1}{\rho} \frac{\partial p}{\partial x_i} - \frac{\partial u_i u_j}{\partial x_j} + \nu \frac{\partial^2 u_i}{\partial x_j^2} \\ \frac{\partial u_k}{\partial x_k} &= 0.\end{aligned}\tag{1}$$

where ρ is the fluid density, u_i is the i^{th} velocity component, p is the hydrodynamic pressure, μ is the constant viscosity, and $\nu = \frac{\mu}{\rho}$.

Direct Numerical Simulation (DNS)

- ▶ Turbulent flows are computationally costly to resolve
- ▶ Solve the Navier-Stokes equations on a large grid to resolve all physics
- ▶ Computationally infeasible for large-scale, **real** engineering applications

Filtered Navier-Stokes equations

We define the filtered quantity $\bar{\phi}$

$$\bar{\phi}(\mathbf{x}, t) \equiv \int_{\Omega} G(\mathbf{r}, \mathbf{x}) \phi(\mathbf{x} - \mathbf{r}, t) d\mathbf{r}, \quad (2)$$

where the filter kernel G is a Gaussian or box filter.

The filtered quantity satisfies the **Large Eddy Simulation** (LES) equation

$$\begin{aligned} \frac{\partial \bar{u}_i}{\partial t} &= -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial \tau_{ij}^r}{\partial x_j}, \\ \frac{\partial \bar{u}_k}{\partial x_k} &= 0. \end{aligned} \quad (3)$$

The residual subgrid-scale stress $\tau_{ij}^r \equiv \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j$ cannot be evaluated using the resolved flow state \bar{u} .

Closure Problem

The LES equation satisfies

$$\begin{aligned}\frac{\partial \bar{u}_i}{\partial t} &= -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial \tau_{ij}^r}{\partial x_j}, \\ \frac{\partial \bar{u}_k}{\partial x_k} &= 0.\end{aligned}\tag{4}$$

- ▶ Learn a closure model for τ^r using deep learning (**DL**)
- ▶ I.E., replace $-\frac{\partial \tau_{ij}^r}{\partial x_j}$ in the LES equation with $h(\bar{u}, \bar{u}_x, \bar{u}_{xx}; \theta)$.

Unresolved physics

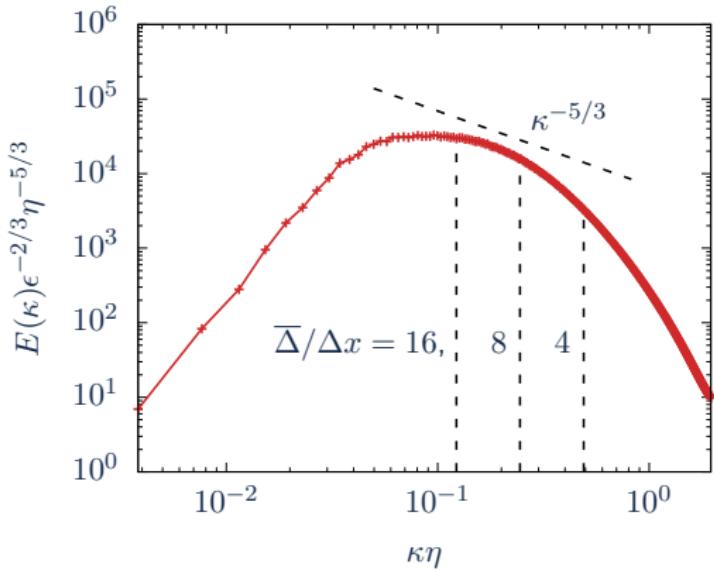


Figure: Energy spectrum of the $N = 1024^3$ isotropic turbulence DNS ($\mu/\mu_0 = 1.5$). The cutoff wave numbers for filter sizes $\bar{\Delta}/\Delta x = 4, 8$, and 16 are indicated by vertical dashed lines.

The LES equation with the deep learning closure model is

$$\begin{aligned}\frac{\partial \bar{u}_i}{\partial t} &= -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} + \frac{\partial h_{ij}}{\partial x_j}(\bar{u}, \bar{u}_x, \bar{u}_{xx}; \theta), \\ \frac{\partial \bar{u}_k}{\partial x_k} &= 0.\end{aligned}\tag{5}$$

- ▶ The parameters θ of the deep learning model $h(\bar{u}, \bar{u}_x, \bar{u}_{xx}; \theta)$ must be estimated from trustworthy data.
- ▶ Let the variable q describe the configuration, i.e.
 $q = (\text{physical parameters, boundary conditions, geometry}).$
- ▶ Denote the solution of (5) for configuration q as $\bar{u}(t, x; q).$
- ▶ Trustworthy data $V(t, x; q)$ (either DNS or experimental) is available for certain values of $q.$

θ is estimated by minimizing the objective function

$$\begin{aligned} L(\theta) &= \underbrace{\sum_{n=1}^N \int_{[0,T]} \int_{\Omega} \| \bar{u}(t, x; q_n) - V(t, x; q_n) \| dt dx}_{\text{Simulations}} \\ &+ \underbrace{\sum_{n=N+1}^{N+M} \sum_{i,j=1}^K \| \bar{u}(t_i, x_j; q_n) - V(t_i, x_j; q_n) \|}_{\text{Experiments}}. \end{aligned} \quad (6)$$

Optimization challenges:

- ▶ Complex function: neural network $h(\cdot; \theta) \rightarrow$ PDE $\bar{u} \rightarrow L(\theta)$.
- ▶ θ is high-dimensional (hundreds of thousands of parameters).

Optimization

- Gradient descent:

$$\theta^{(k+1)} = \theta^{(k)} - \alpha^{(k)} \nabla_{\theta} L(\theta^{(k)}) \quad (7)$$

- One could solve the PDE $w = \nabla_{\theta} \bar{u}$. However, this requires solving a large number of PDEs since θ is high-dimensional!
 - If θ has 100,000 parameters, system of 100,000 PDEs must be solved to calculate $L(\theta^{(k)})$ for a single k !
- We derive an adjoint PDE to efficiently calculate $\nabla_{\theta} L(\theta)$.
 - Number of adjoint PDEs is the same as the dimension of \bar{u} (**not** θ).
 - Thus, only 3 adjoint PDEs need to be solved!

Adjoint PDE

For isotropic turbulence, the LES equation with the DL closure model is:

$$\begin{aligned}\frac{\partial \bar{u}_i}{\partial t} &= -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} + \frac{\partial h_{ij}}{\partial x_j}(\bar{u}, \bar{u}_x, \bar{u}_{xx}; \theta), \\ \frac{\partial \bar{u}_k}{\partial x_k} &= 0,\end{aligned}\tag{8}$$

with a periodic solution, $h(u, v, w; \theta) : \mathbb{R}^3 \times \mathbb{R}^9 \times \mathbb{R}^9 \rightarrow \mathbb{R}^3 \times \mathbb{R}^3$, and where h_{ij} is the (i, j) -th element of the output of h .

- ▶ Let (u, p) be the solution to (8) for a DL model with parameters θ .
- ▶ Let $(u_i + \epsilon u_i^\epsilon, p + \epsilon p^\epsilon)$ be the solution to (8) under the perturbation $\theta \rightarrow \theta + \epsilon e$, where e is an arbitrary vector.
- ▶ The objective function is $L(\theta) = \int_{\Omega} \|u(T, x) - V(T, x)\| dx$.
- ▶ We are interested in $\frac{\partial L}{\partial \epsilon}(\theta + \epsilon e)|_{\epsilon=0}$.

Adjoint PDE

$(u_i^\epsilon, p^\epsilon)$ satisfies

$$\begin{aligned}\frac{\partial u_i^\epsilon}{\partial t} &= -\frac{1}{\rho} \frac{\partial p^\epsilon}{\partial x_i} - \frac{\partial}{\partial x_j} [u_i^\epsilon u_j + u_i u_j^\epsilon] + \nu \nabla^2 u_i^\epsilon \\ &+ \frac{\partial^2 h_{ij}}{\partial x_j \partial u} (u, u_x, u_{xx}; \theta) \cdot u^\epsilon + \frac{\partial^2 h_{ij}}{\partial x_j \partial v} (u, u_x, u_{xx}; \theta) \cdot u_x^\epsilon \\ &+ \frac{\partial^2 h_{ij}}{\partial x_j \partial z} (u, u_x, u_{xx}; \theta) \cdot u_{xx}^\epsilon + \frac{\partial^2 h_{ij}}{\partial x_j \partial \theta} (u, u_x, u_{xx}; \theta) \cdot e \\ &+ \mathcal{O}(\epsilon),\end{aligned}\tag{9}$$

combined with the continuity equation $\frac{\partial u_k^\epsilon}{\partial x_k} = 0$ and periodic boundary conditions.

Let (\hat{u}_i, \hat{p}) be the solution to the adjoint PDE

$$\begin{aligned}
 -\frac{\partial \hat{u}_i}{\partial t} &= \frac{1}{\rho} \frac{\partial \hat{p}}{\partial x_i} + u_j \frac{\partial \hat{u}_i}{\partial x_j} + u_j \frac{\partial \hat{u}_j}{\partial x_i} + \nu \nabla^2 \hat{u}_i \\
 &\quad + \hat{u}_k \frac{\partial^2 h_{kj}}{\partial x_j \partial u_i} - \frac{\partial}{\partial x_m} \left[\hat{u}_k \frac{\partial^2 h_{kj}}{\partial x_j \partial v_{im}} \right] + \frac{\partial^2}{\partial x_m^2} \left[\hat{u}_k \frac{\partial^2 h_{kj}}{\partial x_j \partial z_{im}} \right], \\
 \frac{\partial \hat{u}_k}{\partial x_k} &= 0, \\
 \hat{u}_i(T, x) &= \nabla_{u_i} \|u(T, x) - V(T, x)\|. \tag{10}
 \end{aligned}$$

We assume the adjoint PDE has a periodic solution, and we have used the notation $v_{km} = \frac{\partial v_k}{\partial x_m}$ and $z_{km} = \frac{\partial^2 z_k}{\partial x_m^2}$.

$$\begin{aligned}
& \sum_{i=1}^3 \int_{\Omega} \int_0^T \hat{u}_i \frac{\partial u_i^\epsilon}{\partial t} dt dx = \sum_{i=1}^3 \int_{\Omega} \int_0^T \hat{u}_i \left(-\frac{\partial p^\epsilon}{\partial x_i} - \frac{\partial}{\partial x_j} [u_i^\epsilon u_j + u_i u_j^\epsilon] \right. \\
& + \nu \nabla^2 u_i^\epsilon + \frac{\partial^2 h_{ij}}{\partial x_j \partial u}(u, u_x, u_{xx}; \theta) \cdot u^\epsilon + \frac{\partial^2 h_{ij}}{\partial x_j \partial v}(u, u_x, u_{xx}; \theta) \cdot u_x^\epsilon \\
& \left. + \frac{\partial^2 h_{ij}}{\partial x_j \partial z}(u, u_x, u_{xx}; \theta) \cdot u_{xx}^\epsilon + \frac{\partial^2 h_{ij}}{\partial x_j \partial \theta}(u, u_x, u_{xx}; \theta) \cdot e \right) dt dx + \mathcal{O}(\epsilon).
\end{aligned}$$

Using integration by parts and the definition of the adjoint PDE,

$$\begin{aligned}
& \int_{\Omega} \nabla_u \|V(T, x) - u(T, x)\| \cdot u^\epsilon(T, x) dx \\
& = \sum_{i=1}^3 \int_0^T \int_{\Omega} e \cdot \hat{u}_i \frac{\partial^2 h_{ij}}{\partial x_j \partial \theta}(u, u_x, u_{xx}; \theta) dx dt + \mathcal{O}(\epsilon). \quad (11)
\end{aligned}$$

Let $\epsilon \rightarrow 0$ and using the fact that e is an arbitrary vector:

$$\nabla_{\theta} L(\theta) = \sum_{i=1}^3 \int_0^T \int_{\Omega} \hat{u}_i \frac{\partial^2 h_{ij}}{\partial x_j \partial \theta}(u, u_x, u_{xx}; \theta) dx dt. \quad (12)$$

Stochastic gradient descent

- ▶ Solving the adjoint PDE over the entire time length of the training data is computationally expensive.
- ▶ In practice, we randomly sample sub-intervals $[t, t + \tau] \subset [0, T]$.
- ▶ The adjoint PDE is solved on $[t, t + \tau]$.
- ▶ A stochastic gradient descent step is then taken.

Out-of-sample *a posteriori*: Isotropic Turbulence

- ▶ Trained on Filtered DNS data at $\text{Re} = 3,300$ (1024^3 grid)
- ▶ Tested on Filtered DNS data at $\text{Re} = 7,805$ (2048^3 grid)

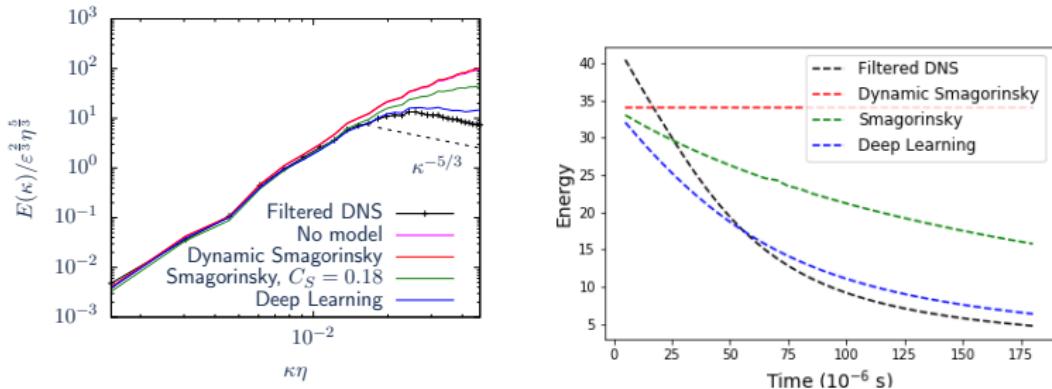


Figure: **Left:** Resolved Kinetic Energy spectrum for different closure models.
Right: Energy decay. The DNS grid was $N = 2048^3$ and the filter-to-grid ratio is $32\times$. Shown for $t/t_0 = 0.28$.

Out-of-sample *a posteriori*: Isotropic Turbulence

- ▶ Train: decaying isotropic turbulence at $\text{Re} = 3,300$ (1024^3 grid)
- ▶ Test: $\text{Re} = 3,300$ but with **different** length scale and viscosity

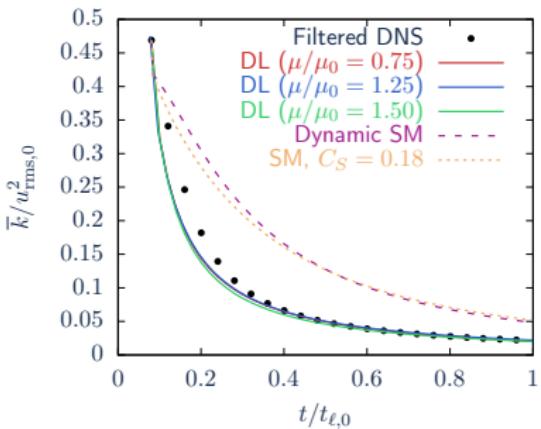
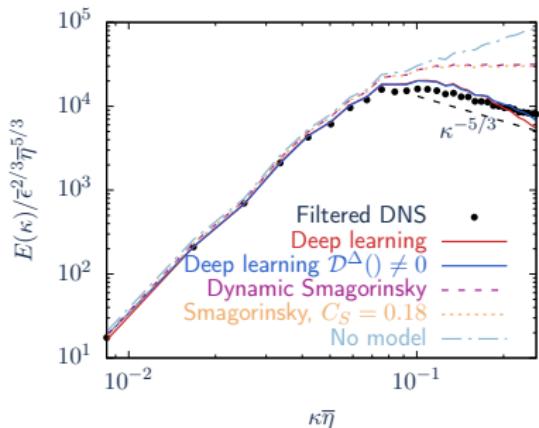


Figure: **Left:** Resolved Kinetic Energy spectrum for LES models. **Right:** Energy decay vs. time. The DNS grid was $N = 1024^3$ and the filter-to-grid ratio is $16\times$.

In-sample *a posteriori*: Turbulent jet

- ▶ Turbulent jet at $\text{Re} = 6,000$ on a $1024 \times 1280 \times 768$ grid
- ▶ LES comparisons for $16 \times$ filter size
- ▶ We study the mean profile:

$$\text{Ave}_i(t, x_2) = \int_0^{L_3} \int_0^{L_1} \bar{u}_i(t, x_1, x_2, x_3) dx_1 dx_3. \quad (13)$$

- ▶ We also study the root-mean-squared fluctuations around the mean profile.

Mean Profile

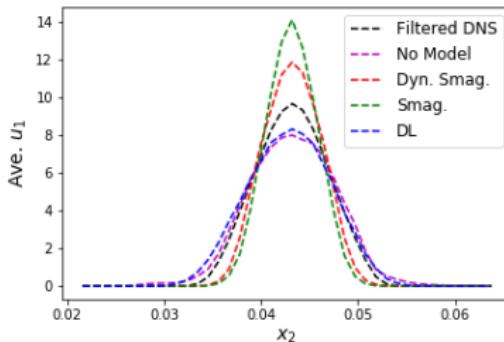
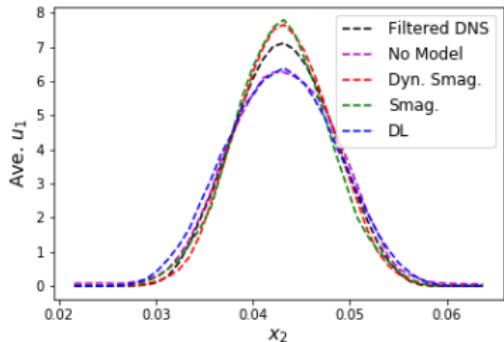
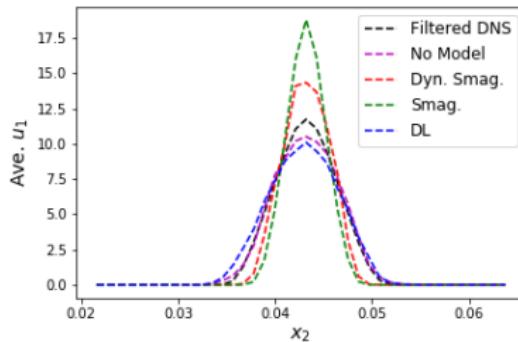
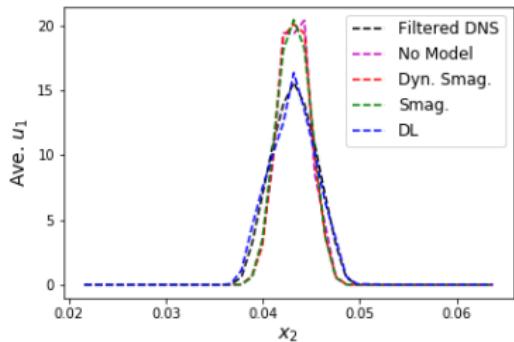
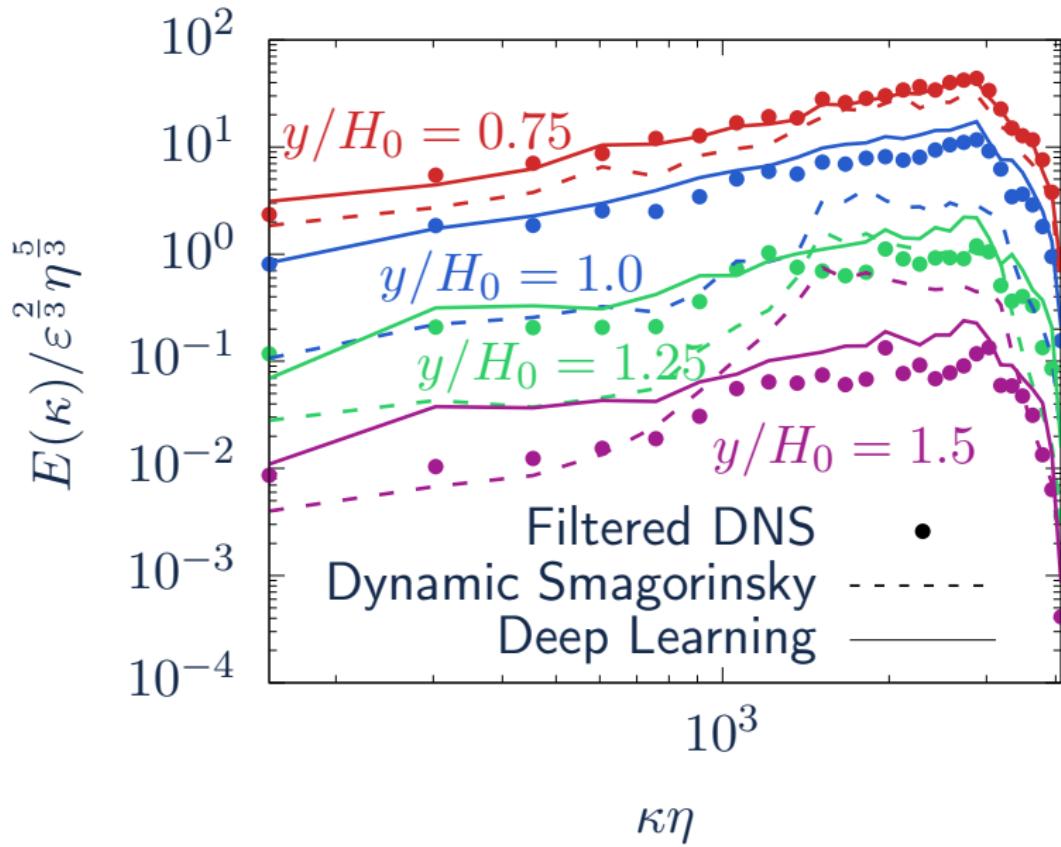


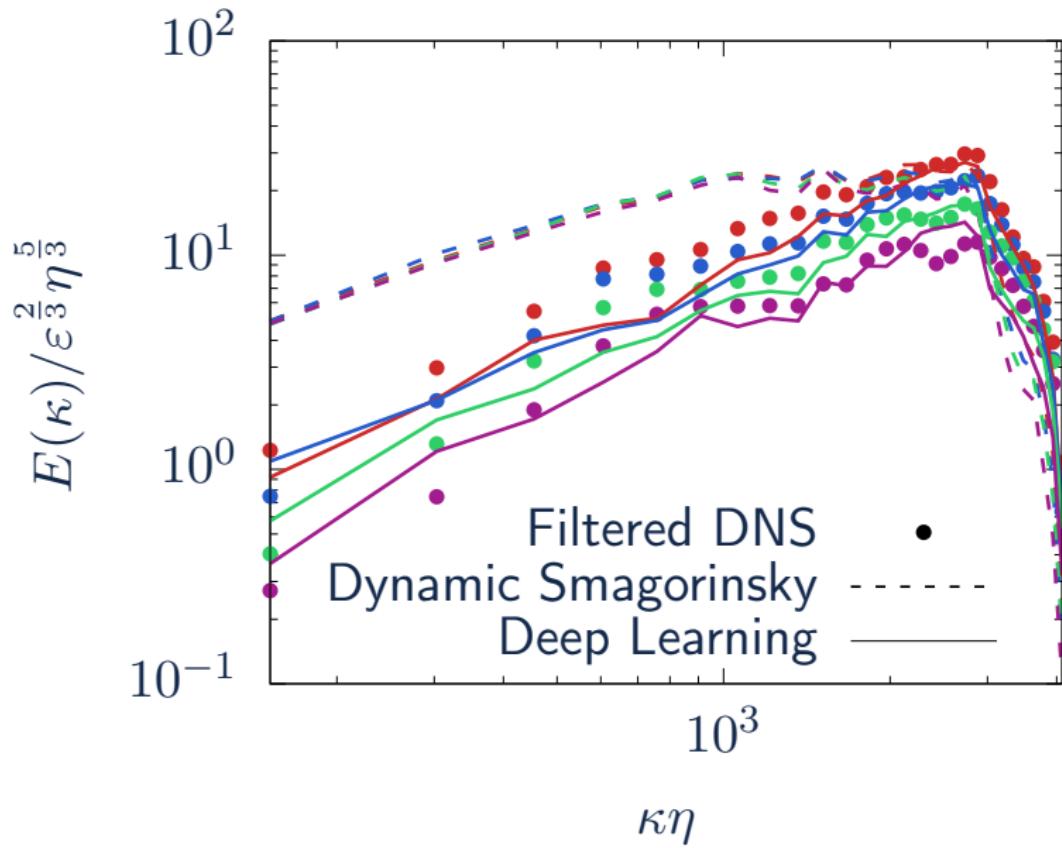
Figure: Clockwise from early to later times:

$t = 3.00 \times 10^{-3}, 6.00 \times 10^{-3}, 8.50 \times 10^{-3}$, and $t = 1.35 \times 10^{-2}$ seconds.

Single Jet Spectrum at $t = 3 \times 10^{-3}$ (s)



Single Jet Spectrum at $t = 1.35 \times 10^{-2}$ (s)



RMS velocity fluctuations for u_1

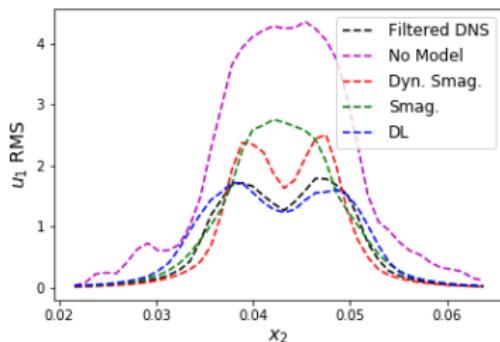
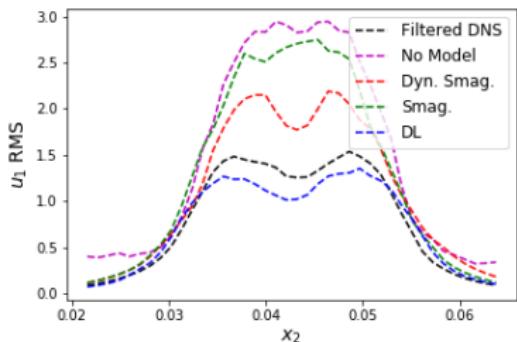
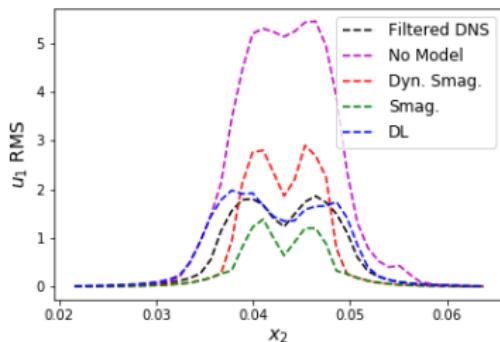
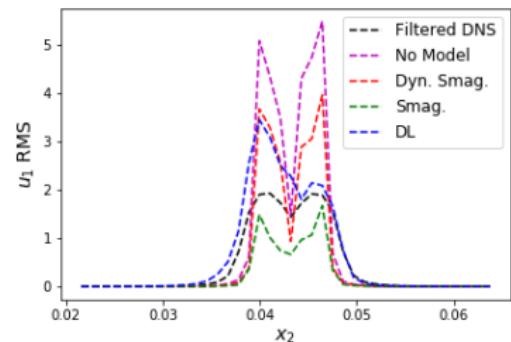


Figure: Clockwise from early to later times:

$t = 3.00 \times 10^{-3}, 6.00 \times 10^{-3}, 8.50 \times 10^{-3}$, and $t = 1.35 \times 10^{-2}$ seconds.

RMS velocity fluctuations for u_2

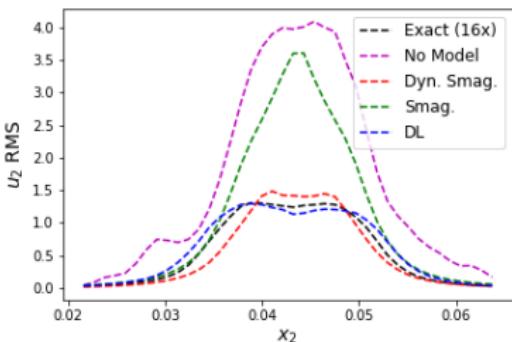
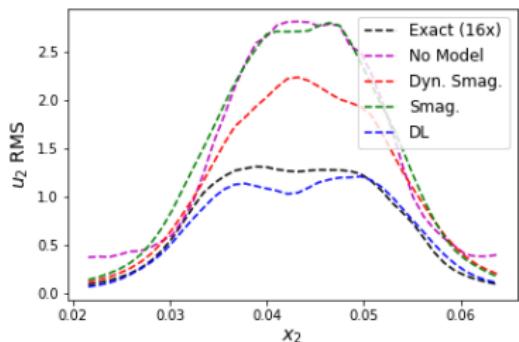
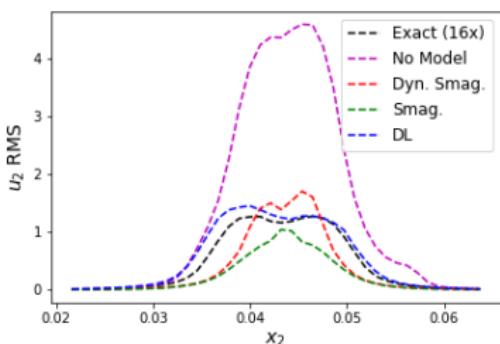
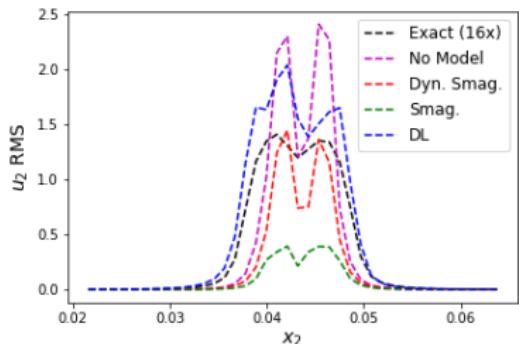


Figure: Clockwise from early to later times:

$t = 3.00 \times 10^{-3}, 6.00 \times 10^{-3}, 8.50 \times 10^{-3}$, and $t = 1.35 \times 10^{-2}$ seconds.

Out-of-sample *a posteriori*: Double jet

- ▶ Two parallel turbulent jets at $\text{Re} = 6,000$ on a $1024 \times 1792 \times 768$ grid
- ▶ DL closure model trained only on single jet DNS dataset, and tested **out-of-sample** on double jet.

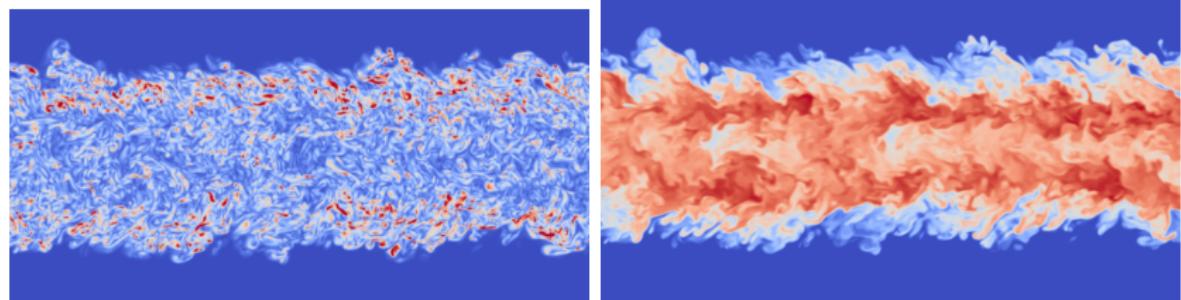


Figure: **Left:** Vorticity. **Right:** Velocity.

Mean profile for double jet

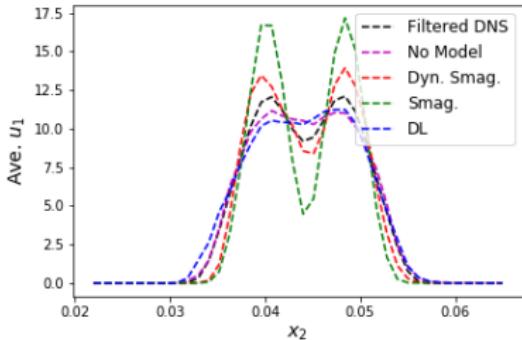
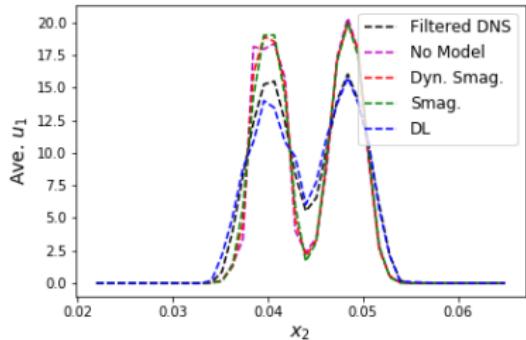


Figure: Left: $t = 3.00 \times 10^{-3}$ s. Right: $t = 6.00 \times 10^{-3}$ s.

Double jet: RMS for u_1

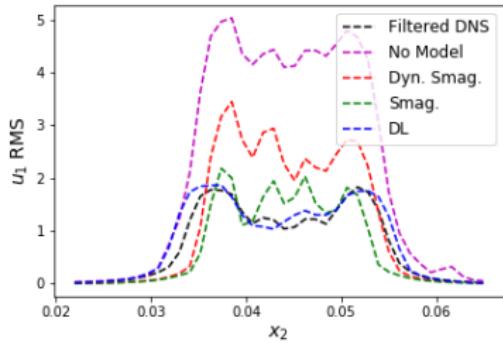
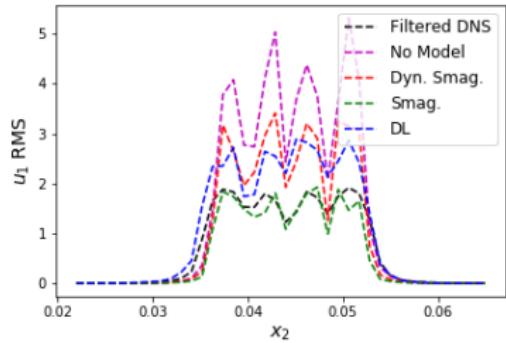


Figure: Left: $t = 3.00 \times 10^{-3}$ s. Right: $t = 6.00 \times 10^{-3}$ s.

Double jet: RMS for u_2

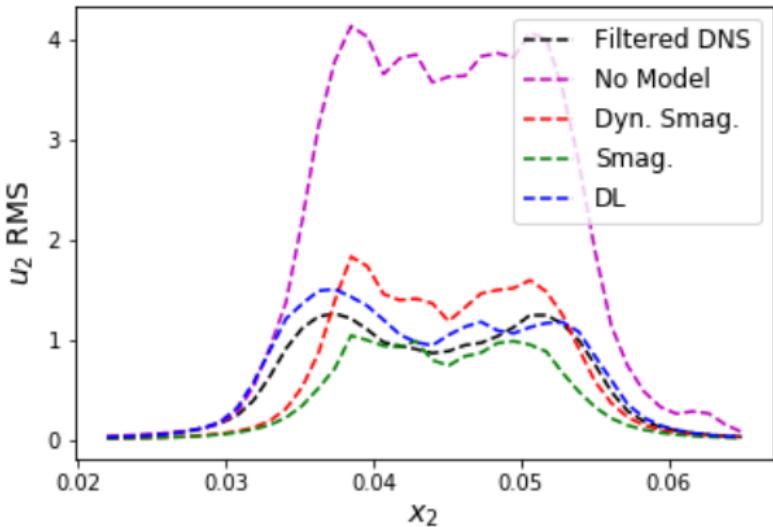


Figure: $t = 6.00 \times 10^{-3}$ s.

Double jet: Reynolds Stress $\tau_{1,2}$

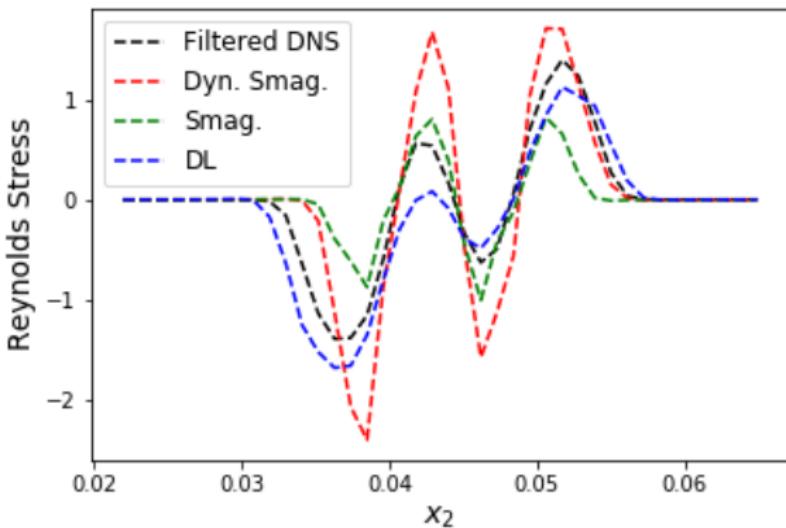
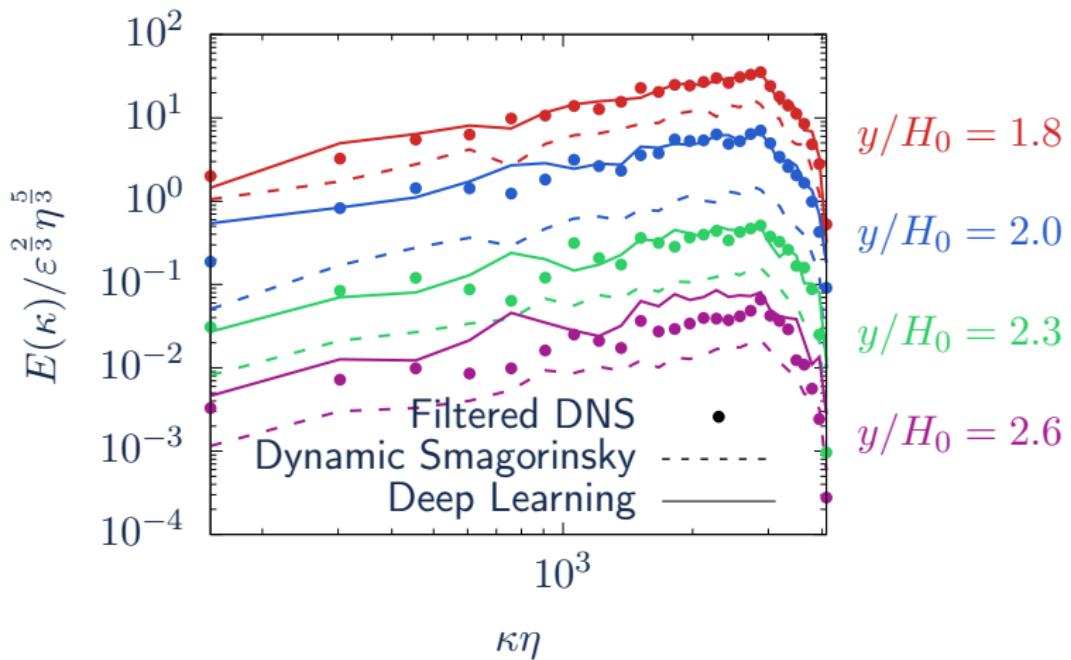


Figure: Reynolds Stress(x_2) = $\int_0^{L_3} \int_0^{L_1} (\bar{u}_1 - \text{Ave}_1)(\bar{u}_2 - \text{Ave}_2) dx_1 dx_3$ at $t = 6.00 \times 10^3$ (s).

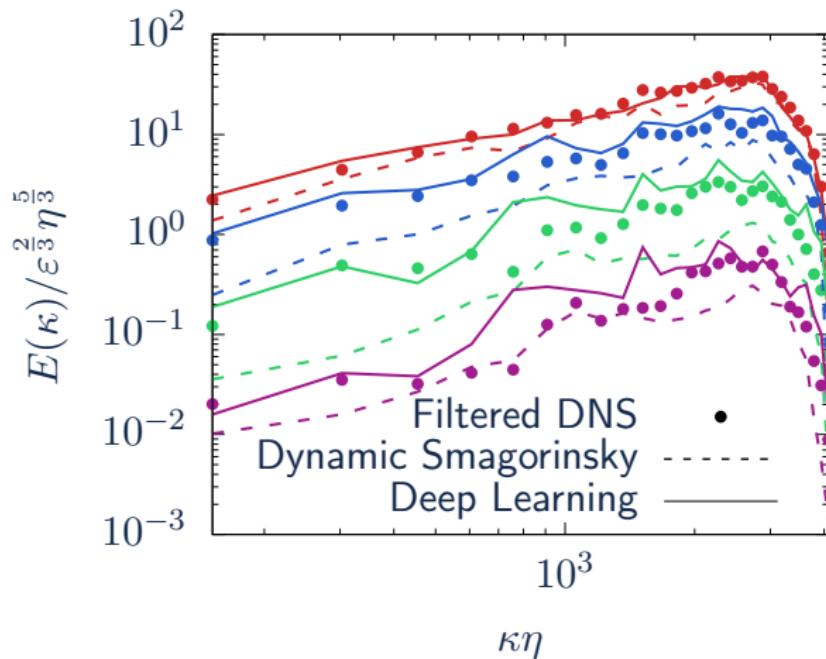
Double Jet: Spectrum at $t = 3 \times 10^{-3}$

- DL model trained on filtered DNS from single jet
- Tested on double jet



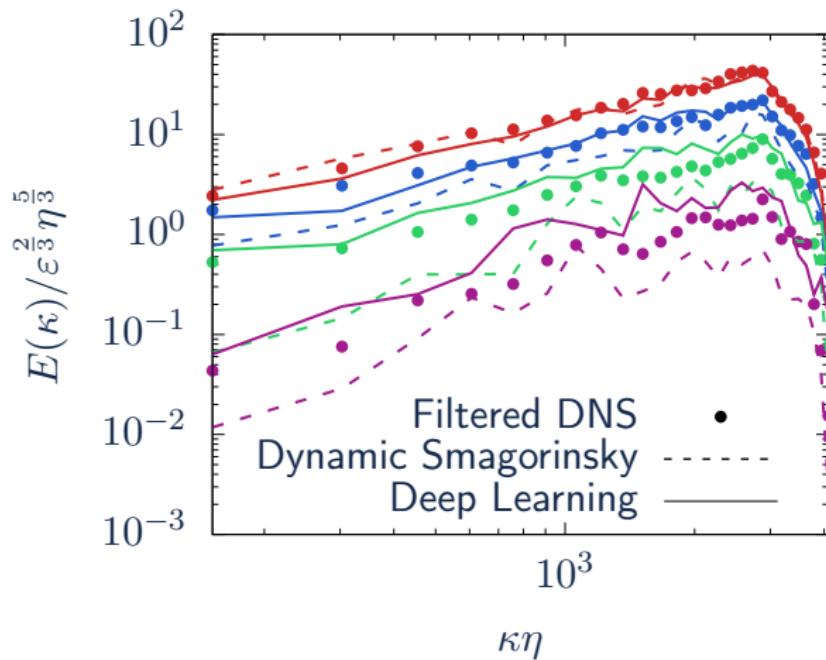
Double Jet: Spectrum at $t = 4.5 \times 10^{-3}$

- DL model trained on filtered DNS from single jet
- Tested on double jet



Double Jet: Spectrum at $t = 6.0 \times 10^{-3}$

- DL model trained on filtered DNS from single jet
- Tested on double jet



Sources of error in LES

- ▶ Closure error
- ▶ Finite difference error

LES is typically solved with mesh size $\Delta = \bar{\Delta}$. The coarse grid reduces the computational cost, but introduces

- ▶ Large finite difference errors in the momentum equation
- ▶ Large residual errors in the divergence equation

These errors are of the same magnitude (or larger) as the closure error!

Finite difference error on the (coarse) LES grid

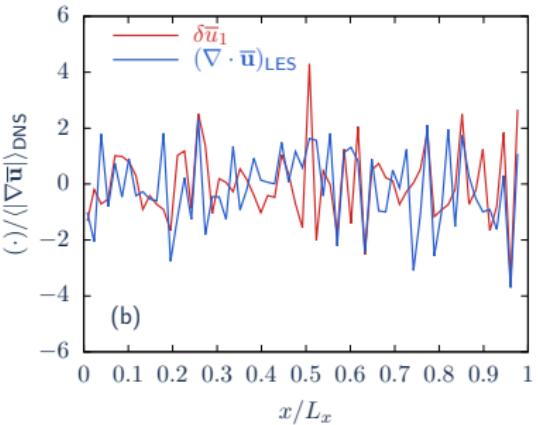
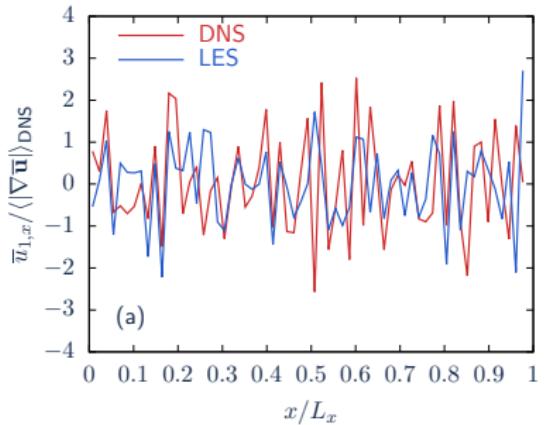


Figure: **Left:** finite-difference approximations to the filtered velocity gradient $\bar{u}_{1,x}$ evaluated on DNS ($N = 2048^3$) and LES ($N = 64^3$) grids. **Right:** $\delta \bar{u}_1 = \bar{u}_{1,x,LES} - \bar{u}_{1,x,DNS}$ and the normalized velocity divergence on the LES grid.

Finite difference error on the (coarse) LES grid

Filtering	$\frac{\bar{\Delta}}{\Delta_{x\text{DNS}}}$	$\frac{\Delta}{\Delta_{x\text{DNS}}}$	$\frac{\langle \delta\bar{u}_1 \rangle}{\langle \nabla\bar{u} \rangle_{\text{DNS}}}$	$\max(\mathcal{D}^\Delta(\bar{u}))_{\text{LES}}$
Implicit	8	8	0.601	2.257×10^5
	16	16	0.854	1.216×10^5
	32	32	1.076	0.579×10^5
Explicit	32	16	0.654	4.734×10^4
	32	8	0.325	2.855×10^4
	32	4	0.141	1.325×10^4

Table: Error in the finite-difference approximation to the filtered velocity gradient $\delta\bar{u}_1 = \bar{u}_{1,x,\text{LES}} - \bar{u}_{1,x,\text{DNS}}$, evaluated on LES grids of varying mesh size Δ . The maximum of the discrete filtered-velocity divergence is also evaluated on the LES grid.

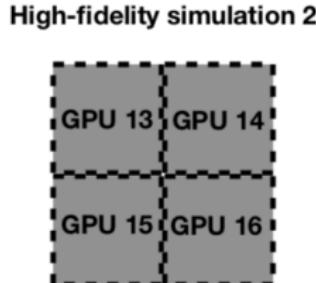
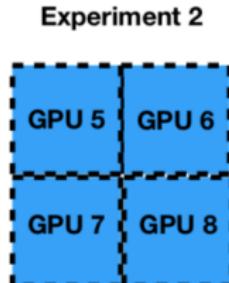
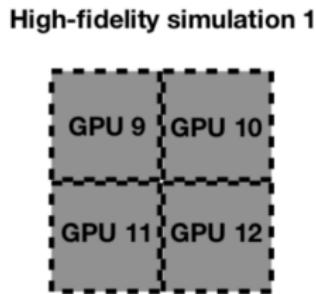
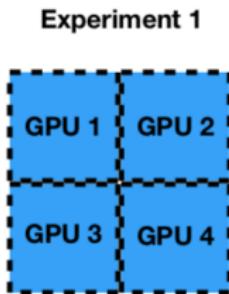
Scalable multi-GPU code

- ▶ Solves Navier-Stokes equation and its adjoint PDE
- ▶ Written in Python
 - Can be rapidly adapted to new configurations
 - Seamless integration with PyTorch deep learning library
- ▶ All calculations on GPU (accelerates solution time by $\sim 10\times$)
- ▶ Parallelization:
 - Domain decomposition
 - Data parallelism

Scalable multi-GPU code

► Parallelization:

- Domain decomposition
- Data parallelism



Numerical Solution of Adjoint PDE

The LES equation and its adjoint PDE can both be solved via Chorin's projection method:

$$u(t, x) \longrightarrow \text{Mom. Eqn.} \longrightarrow \text{Poiss. Eq.} \longrightarrow u(t + \Delta t, x),$$

$$\hat{u}(t, x) \longleftarrow \text{Adj. Mom. Eqn.} \longleftarrow \text{Adj. Poiss. Eq.} \longleftarrow \hat{u}(t + \Delta t, x).$$

Our implementation uses:

- ▶ A staggered grid for both LES and the adjoint PDE
- ▶ Runge-Kutta 4 for the momentum equations
- ▶ PyTorch for fast evaluation of the derivative of the neural network, which appears in the adjoint PDE momentum equation.

Conclusion

- ▶ Derived adjoint PDEs for optimization of the deep learning closure model
- ▶ Developed scalable multi-GPU code to solve adjoint PDEs
- ▶ Evaluated on DNS datasets for decaying isotropic turbulence and turbulent jets

Acknowledgements

1. This material is based in part upon work supported by the Department of Energy, National Nuclear Security Administration, under Award Number DE-NA0002374.
2. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the State of Illinois.