Sandia National Laboratories

Exceptional service in the national interest

A data-driven turbulence modeling framework for the Reynolds-averaged Navier-Stokes equations via discrepancy-based tensor-basis neural networks

Eric Parish, **David Ching**, Nathan Miller, Steven Beresh, and Matt Barone

Sandia National Laboratories

SAND2022-10175 C

NASA 2022 Symposium on Turbulence Modeling: Roadblocks and the Potential for Machine Learning

July 27-29, 2022 Suffolk, VA

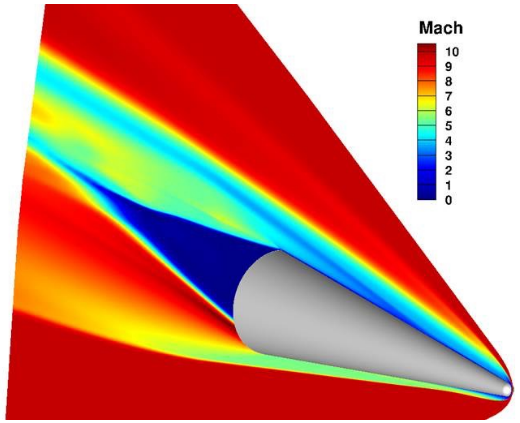
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

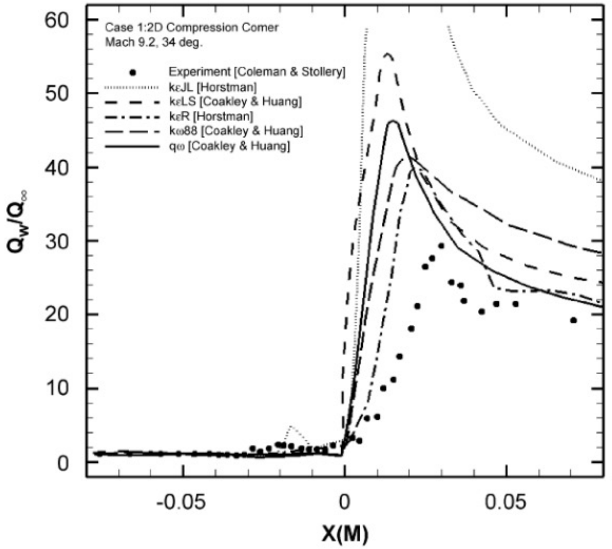
U.S. DEPARTMENT OF ENERGY

NNSA

Introduction

- Year one of a three year internal project focused on developing data-driven RANS models for turbulent flows
  - Focus on hypersonics
- Focused on three important aspects of data-driven turbulence modeling
  - Improved model forms via discrepancy tensor-basis neural networks
    - Focus on the Reynolds stress and turbulent heat flux
  - Robust ML models via extrapolation detection
  - Improved training techniques to improve feature and model consistency
- Three year project trajectory
  - Year 1: Implement discrepancy TBNNs & extrapolation detection techniques and assess on benchmark low-speed cases
  - Year 2: Extend technology to high-speed flows
    - Focus on compressibility effects and the turbulent heat flux
  - Year 3: Formalize capability and deploy on "test" configurations
- Today: Focus on details of our framework and its deployment on the NASA test cases

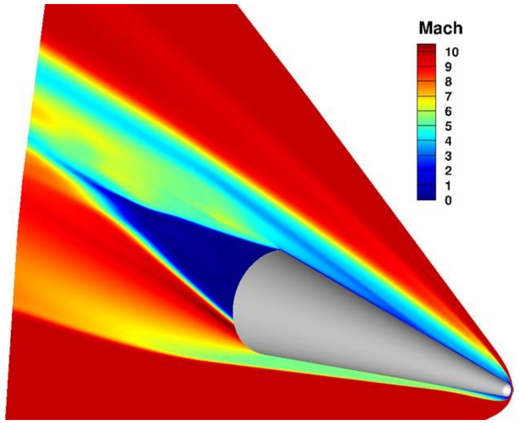




RANS predictions of thermal loadings in a shock boundary layer interaction [2]

[2] Bosco, A., *Reynolds Stress Model for Hypersonic Flows*, Ph.D. thesis, Rheinisch-Westfalischen Technischen Hochschule Aachen, 2011.

Mathematical setting



- We focus on solving the Favre-averaged compressible Navier—Stokes equations

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(u_j \rho u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j}$$

  - Reynolds stress is unclosed

- Turbulence is modeled with a k-epsilon model

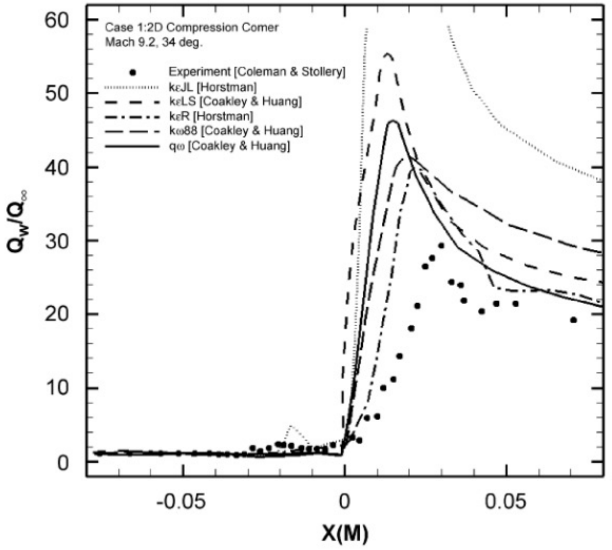
$$\frac{\partial \rho k}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i k) = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_i}\right] + P_k - \rho\epsilon + S_k$$
$$\frac{\partial \rho \epsilon}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i \epsilon) = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_i}\right] + \frac{\epsilon}{k}(C_{\epsilon 1} f_1 P_k - C_{\epsilon 2} f_2 \rho\epsilon) + S_\epsilon$$

- Reynolds stress is modeled via the Boussinesq relationship with low Reynolds number treatment

$$\tau_{ij} = 2\mu_t \overline{S}_{ij} - \frac{2}{3}\rho k \delta_{ij} \qquad \mu_t = C_\mu f_\mu \rho \frac{k^2}{\epsilon}$$

- Can be written in terms of the anisotropy tensor $a_{ij} = \frac{\overline{\rho u_i'' u_j''}}{2\overline{\rho} k} - \frac{1}{3}\delta_{ij}$

- Standard k-epsilon model is inadequate in various settings



RANS predictions of thermal loadings in a shock boundary layer interaction [2]

Tensor basis neural networks (TBNNs)

- We investigate tensor basis neural networks [1] (TBNNs) for modeling the anisotropy tensor
- TBNNs model the anisotropy tensor as:

$$a_{ij} \approx a_{ij}^{\mathrm{TBNN}} \equiv \sum_{i=1}^{10} \mathbf{g}^k(\boldsymbol{\lambda}) \mathbf{T}_{ij}^k$$

Regression model, e.g., MLP

Tensor basis, e.g., S_{ij}

Frame-invariant feature, e.g., $\mathrm{trace}\,(S_{ij} S_{jk})$

  - TBNN basis expansion and features are motivated by Pope's generalized eddy viscosity hypothesis
- Resulting model for the Reynolds stress

$$\tau_{ij} \equiv -\overline{u_i u_j} = -\frac{2}{3} k \delta_{ij} - 2k \sum_{i=1}^{10} \mathbf{g}^k(\boldsymbol{\lambda}) \mathbf{T}_{ij}^k.$$

TBNNs been demonstrated to be effective in various settings

Challenges that we aim to address in our work:

- ML model **completely** replaces the Boussinesq relationship
- May perform well where we have training data, but what happens for extrapolation?
- Data and model inconsistency can be an issue (TKE RANS != TKE DNS)

[1] Ling, J., Kurzawski, A., and Templeton, J., "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, Vol. 807, 2016, p. 155–166. https://doi.org/10.1017/jfm.2016.615.

Discrepancy tensor basis neural networks (TBNNs)

- Challenge: Vanilla TBNN completely replaces the Boussinesq relationship
- We model the Reynolds stress with a "discrepancy" TBNN [1]
  - Idea: correct a baseline model with a TBNN discrepancy term
  - Model takes the form

$$a_{ij} = a_{ij}^{\text{RANS}} + \delta a_{ij}^{\text{TBNN}} \equiv a_{ij}^{\text{RANS}} + \sum_{i=1}^{10} \mathbf{g}^k(\boldsymbol{\lambda})\mathbf{T}_{ij}^k$$

- Advantages: Builds upon, rather than replaces, an existing model
  - More explainable, more stable
- Disadvantage: Discrepancy model is tied to a specific turbulence model
  - (This is really the case for a standard TBNN as well)

Challenges that we aim to address in our work:

- ~~ML model **completely** replaces the Boussinesq relationship~~
- May perform well where we have training data, but what happens for extrapolation?
- Data and model inconsistency can be an issue (TKE RANS != TKE DNS)

[1] Peters, E. L., Balin, R., Jansen, K. E., Doostan, A., and Evans, J. A., "S-frame discrepancy correction models for data- informed Reynolds stress closure," *Journal of Computational Physics*, Vol. 448, 2022, p. 110717. https://doi.org/https: //doi.org/10.1016/j.jcp.2021.110717, URL https://www.sciencedirect.com/science/article/pii/S0021999121006124.

Out-of-distribution (OOD) detection

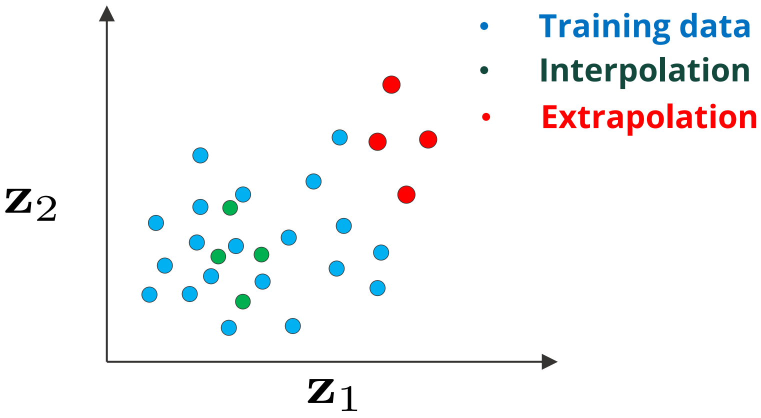
- For use in **practical** situations the ML model must be certified
- An ML model is only as good as the training data
  - Models perform well on in-distribution data
  - Models perform poorly on out-of-distribution data

Stone wall (87% confidence)

EfficientNet predictions on new images[1]

- **Our approach:** Quantify if a testing data point is "in-distribution" or "out-of-distribution"
  - Use this metric to assign confidence to a prediction
  - Use prediction confidence to make an informed decision about how to use the ML model

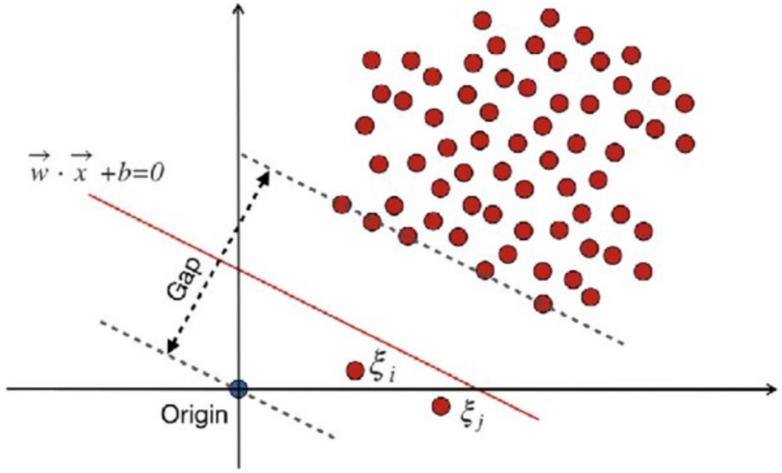
- **What are we investigating:**
  - One-class support vector machines
  - Autoencoders
  - **Gaussian mixture models**
  - **Ensemble-based neural networks**



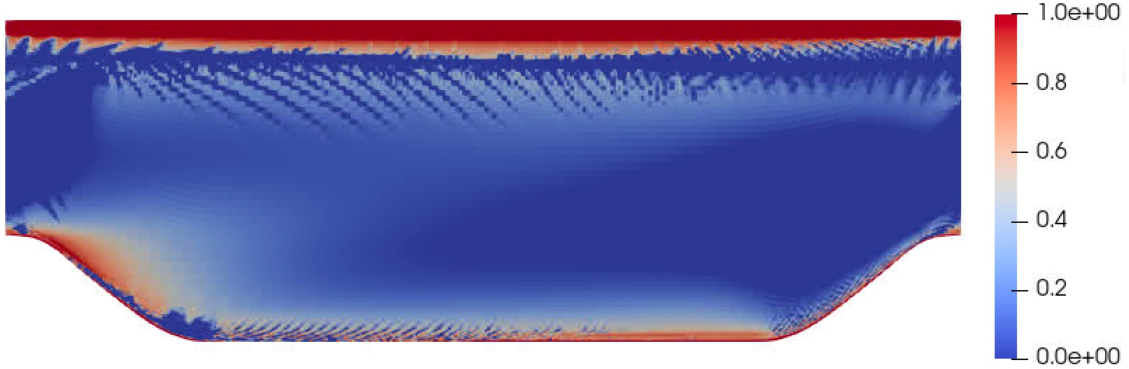
(1) E. Hullermeier and W. Waegeman, Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods, 2019, arXiv:1910.09457 [cs.LG].

Out-of-distribution (OOD) detection

- We first explored using one-class SVMs and Gaussian mixture models to detect out-of-distribution samples
  - Most other methods either scale with the training dataset size or assume radial basis functions
- Main idea: query algorithm at test time, and revert back to standard RANS model when OOD is detected
- Used in combination with artificial training data method of Rumsey et al. [1]
- Success has been mixed
- Issues:
  - Features can be highly correlated, making it difficult to identify OOD
  - OOD detection method can create instabilities when applied in flow solver



Example of using a one-class SVM for extrapolation detection [2]

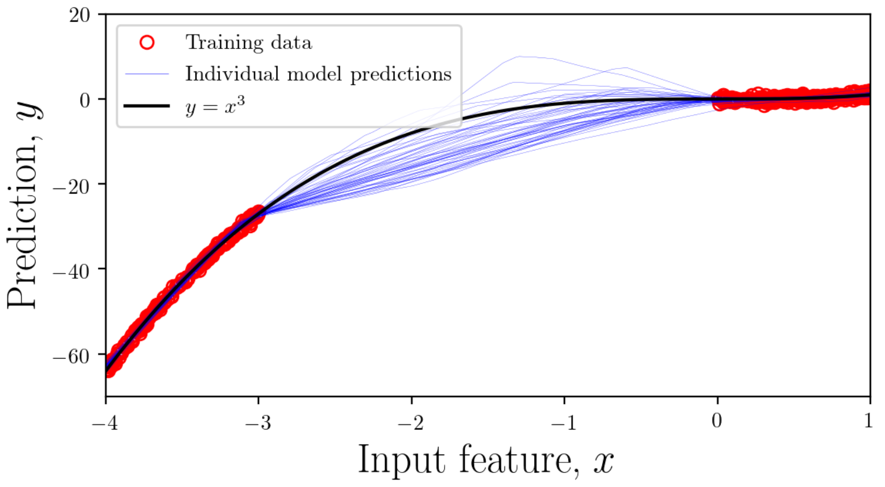


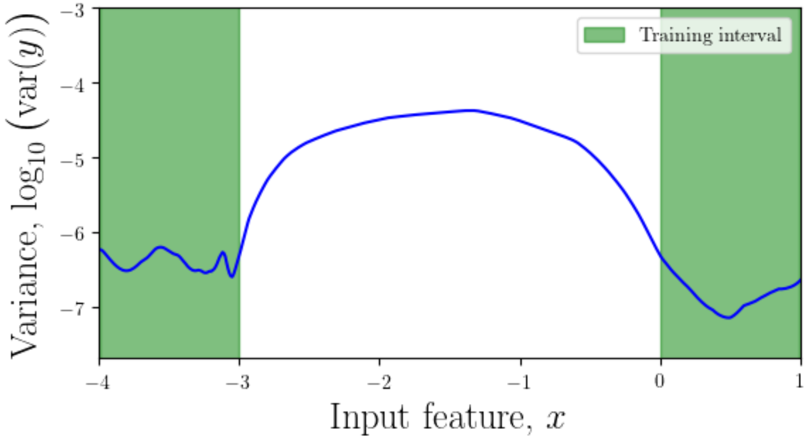
Output of Gaussian mixture model trained on wavy wall (Re=6850) applied to periodic hill (Re=10,595)

1. Rumsey, C.L., Coleman, G.N., Wang, L. "In Search of Data-Driven Improvements to RANS Models Applied to Separated Flows" AIAA SciTech Forum, 2022
2. Koo, B., and Shin, B. Using Geometry based Anomaly Detection to check the Integrity of IFC classifications in BIM Models, Journal of KIBIM, 2017

Out-of-distribution detection with neural-network ensembles

- Ensemble-based networks classify extrapolation based on an ensemble of predictions
  - **Main idea:** Identical networks with different initializations result in different predictions for out-of-distribution data
    - Relies on stochastic initialization
  - **Prediction variance** can be used to quantify network uncertainty
- **Pros:**
  - Works well for high-dimensional features space
  - Gives non-binary results that can be used to assign confidence
  - Occam's razor: idea is simple, interpretable, and easy to implement
- **Cons:**
  - **Empirical**
    - Variance of ensembles ***is not*** reflective of true uncertainty
    - Variance of ensembles depends on numerous factors (number of epochs, regularization, network architecture, etc.)
  - Each evaluation of the MLP is more expensive





Example of deep ensembles for learning y=x^3

[1] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty using deep ensembles, NEURIPS, 2017.

Out-of-distribution detection with neural-network ensembles

- Approach: select a model architecture $\mathcal{NN} : (\boldsymbol{\lambda}; \mathbf{w}) \mapsto \delta a^{\text{TBNN}}$
- Train M neural networks for weights $\boldsymbol{w_1}, \dots, \boldsymbol{w_m}$
- At run time evaluate the empirical mean and variance of the networks

$$\mathbb{E}[\delta a^{\text{TBNN}}] \equiv \frac{1}{M}\left(\sum_{i=1}^{M} \mathcal{NN}(\boldsymbol{\lambda}; \mathbf{w}_i)\right) \qquad \text{Var}[\delta a^{\text{TBNN}}] \equiv \frac{1}{M}\sum_{i=1}^{M} (\mathcal{NN}(\boldsymbol{\lambda}; \mathbf{w}_i))^2 - \mathbb{E}[\delta a^{\text{TBNN}}]^2.$$

- Employ the prediction

$$\delta a^{\text{TBNN}}_{model} = \frac{\sigma^2_{\text{prior}}}{\sigma^2_{\text{prior}} + \text{Var}[\delta a^{\text{TBNN}}]}\mathbb{E}[\delta a^{\text{TBNN}}]$$

  - Based on a Bayesian update with assumed Gaussians and a zero-mean prior discrepancy
- Prediction depends on σ_{prior} parameter
  - Tune from calibration on training/validation set

Challenges that need to be addressed:

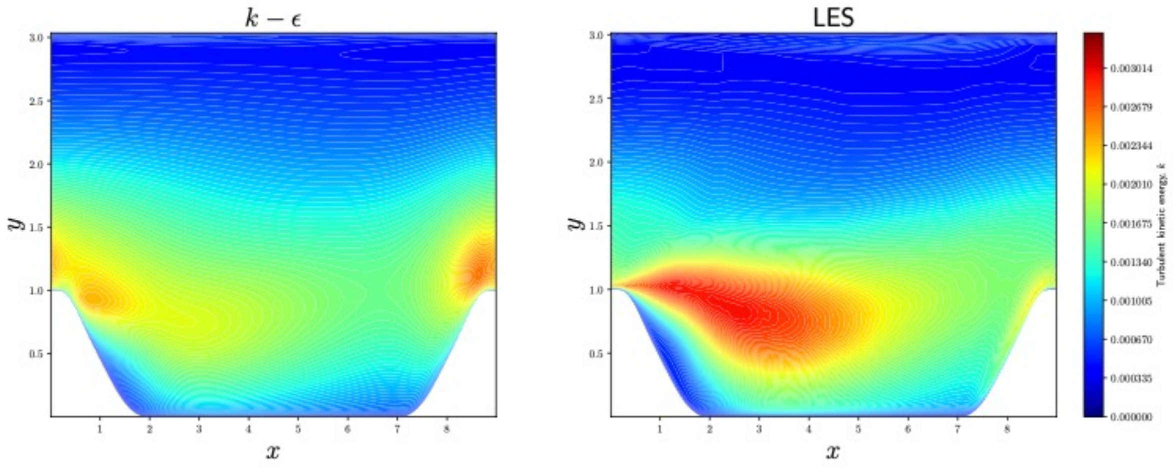
- ~~ML model **completely** replaces the Boussinesq relationship~~
- ~~May perform well where we have training data, but what happens for extrapolation?~~
- Data and model inconsistency can be an issue (TKE RANS != TKE DNS)

Data and model consistency

- Discrepancy TBNN approximates the Reynolds stress as

$$\rho\tau_{ij} \equiv -\rho\overline{u_i u_j} = -\frac{2}{3}\rho k \delta_{ij} - 2\rho k \left(a_{ij}^{\text{RANS}} + \delta a_{ij}^{\text{TBNN}}\right)$$

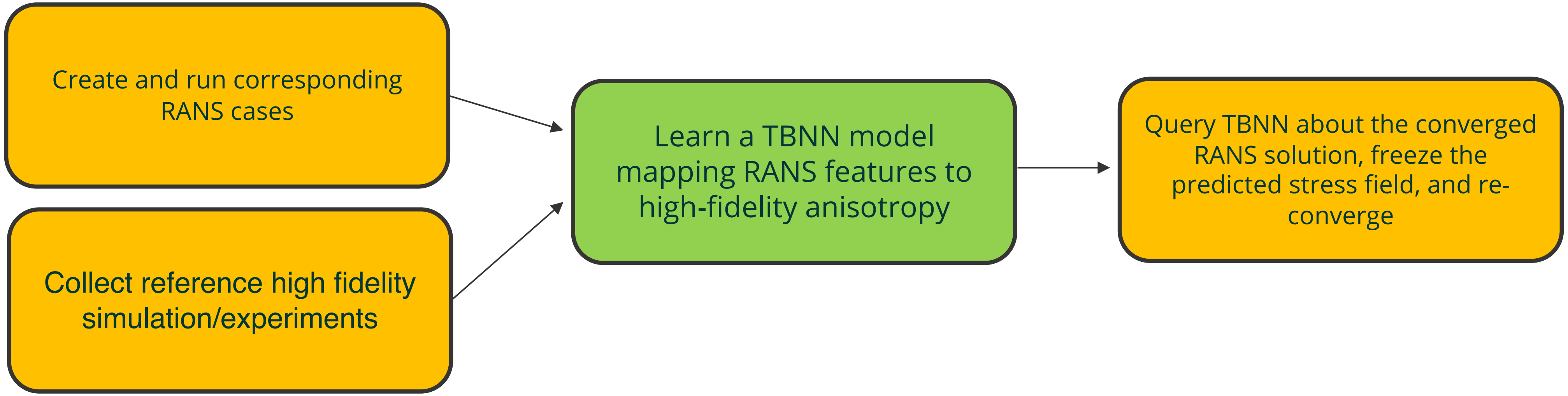
- **Significant challenge:** High-fidelity turbulence quantities are not equivalent to RANS turbulent quantities due to model form error in, e.g., the TKE equation
  - Example: Turbulent kinetic energy in flow over a periodic hill



  - What value of turbulent kinetic energy should be used to train an ML model?
  - Even with perfect Reynolds stress prediction, RANS TKE will differ from true TKE due to model form error

Training workflow

- Baseline approach employed by Ling et al.



- Significant issue:
  - Learns Reynolds stress fields from deficient features (particularly TKE)
    - Should lead to generalization errors
- Can be philosophically unsatisfying. Better to have a model that is embedded in the code and called at each iteration

Training workflow

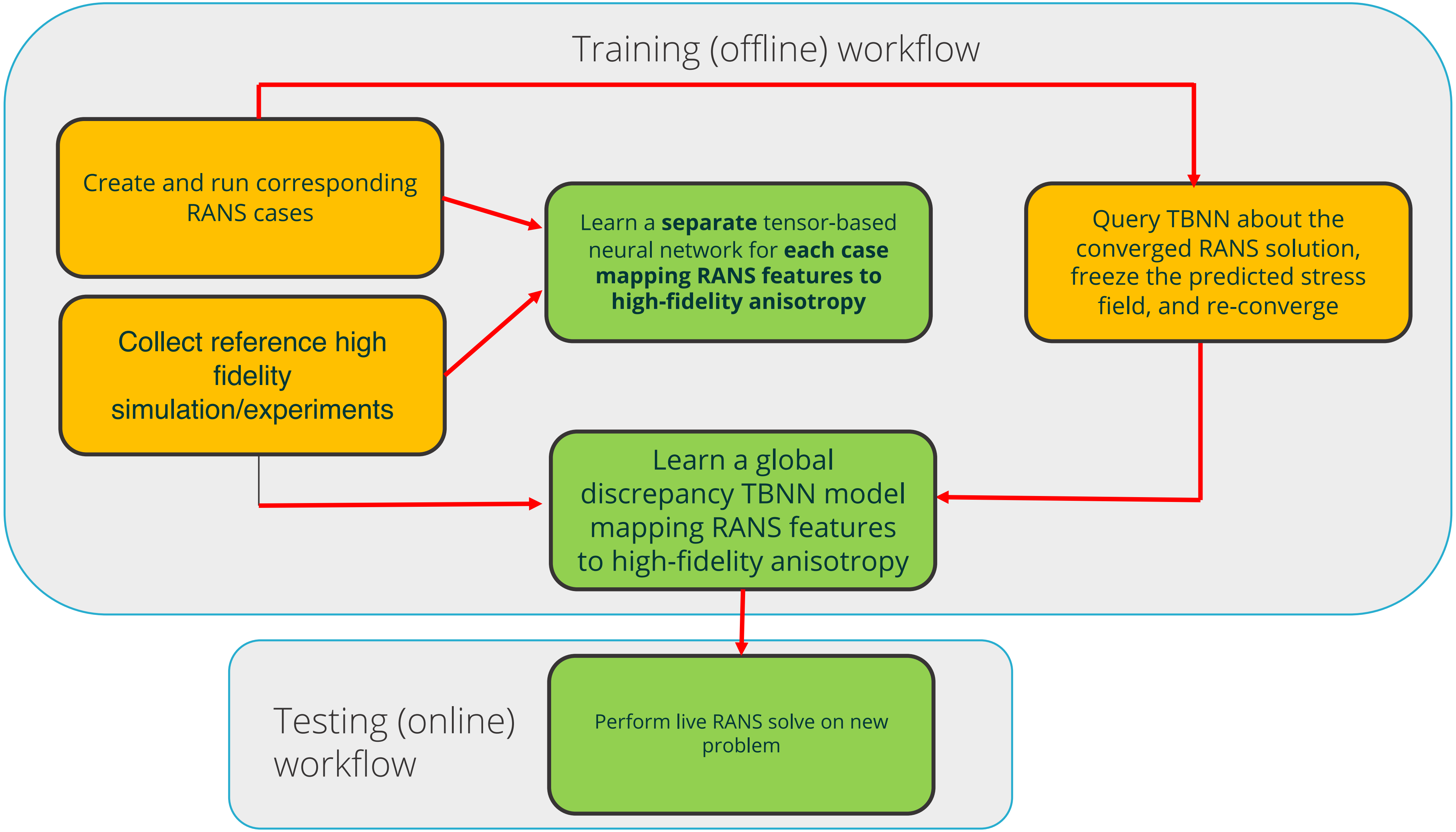
- Our first approach

Collect reference high fidelity simulation/experiments → Create and run corresponding RANS cases with exact anisotropy tensor injected → Learn a TBNN model mapping RANS features to high-fidelity anisotropy → Perform RANS solve by querying TBNN at every iteration

- Approach is still promising, but lacks robustness
  - **Hard to converge solvers** with injected anisotropy tensor
  - In practice we can never learn the true anisotropy, so we still have an inconsistency
  - Requires full-field high-fidelity data, which limits available training sets

Training workflow

- Use the TBNN within the training process to get more consistent features



Summary

- Employing discrepancy tensor-basis neural networks
  - Correct, rather than replace, a prediction for the Reynolds stress

- Detecting extrapolation based on ensembles of neural networks
  - 10 networks are better than one!

- Reducing training data inconsistencies by employing two step training process
  - Deploy a “frozen” TBNN on a given case
  - Learn mapping from resulting RANS features to true anisotropy

Results on NASA test cases

Solver and implementation details

- Discrepancy tensor-based neural networks are implemented in Sandia's Parallel Aerodynamics and Reentry code (SPARC)
  - Supports finite volume, finite difference, and finite element (CG & DG) discretizations

- Vanilla deep neural network library was developed and added to SPARC
  - Interfacing to PyTorch C++ backend was **extremely** slow due to SPARC kernels

- Discrepancy TBNNs are only queried every 10 iterations to accelerate solver performance

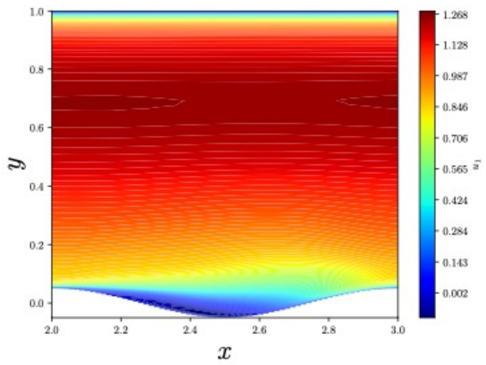
- Sensitivities to the TBNN are **not** included in the Jacobian routines
  - Cases can take a long time to converge
  - We are investigating integrating this with automatic differentiation

- TBNN-predicted Reynolds stresses are **NOT** used in the TKE and dissipation equations
  - Why? A good idea, but model constants are calibrated off of the Boussinesq relationship and should be recalibrated if we do this

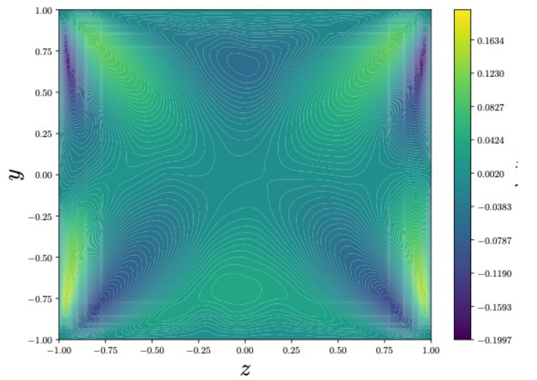
- All cases are run unsteady with BDF1 and a CFL controller to reach a steady state

Training data and their sources

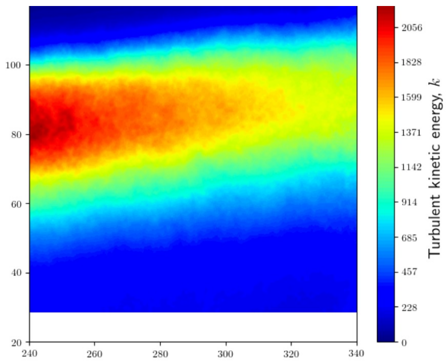
- We employ a mix of DNS, LES, and experimental data
- DNS
  - Channel flow at Re_τ = 180,395, and 590 from Jimenez [1]
  - Duct flow at Re=3500 from Pinelli et al. [3]
  - Wavy wall flow at Re=6850 [5]
- LES
  - Flow over a periodic hill from Temmerman et al. [2]
- Experimental:
  - Jet-in-crossflow from Beresh et al. [4]
- Intentionally did **not** include:
  - Experimental data from any of the test cases
  - LES over NASA wall mounted hump
  - ZPG boundary layer data
- Only **one** ensemble of TBNNs is trained on **all** the data



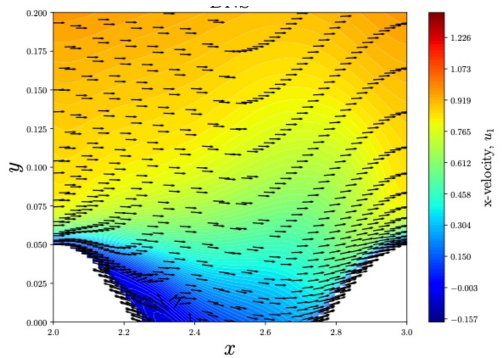
Wavy wall



Duct flow



Jet in cross flow



Periodic hill

[1] Sergio Hoyas and Javier Jimenez, (2008) "Reynolds number effects on the Reynolds-stress budgets in turbulent channels", Phys. Fluids, Vol. 20, 101511.

[2] Temmerman, L. and Leschziner, M. A., 2001. "Large Eddy Simulation of separated flow in a streamwise periodic channel construction," Int. Symp. on Turbulence and Shear Flow Phenomena, Stockholm, June 27-29.

[3] Pinelli, A., Uhlmann, M., Sekimoto, A., Kawahara, G. 2010 Reynolds number dependence of mean flow structure in square duct turbulence. J. Fluid Mech. 644, 107–122.

[4] Beresh, S. J., Henfling, J. F., Erven, R. J., & Spillers, R. W., "Penetration of a Transverse Supersonic Jet into a Subsonic Compressible Crossflow," AIAA Journal, Vol. 43, No. 2, 2005, pp. 379–389

[5] Gorlé, C., Emory, M., Larsson, J., and Iaccarino, G., "Epistemic Uncertainty Quantification for RANS Modeling of the Flow over a Wavy Wall," Center for Turbulence Research Annual Research Briefs, 2012.

Neural network details

- Model the anisotropy tensor as

$$a_{ij} = a_{ij}^{\mathrm{RANS}} + \delta a_{ij}^{\mathrm{TBNN}} \equiv a_{ij}^{\mathrm{RANS}} + \sum_{i=1}^{10} \mathbf{g}^k(\boldsymbol{\lambda})\mathbf{T}_{ij}^k$$

Regression model, e.g., MLP

Tensor basis, e.g., S_{ij}

Frame-invariant feature, e.g., $\mathrm{trace}\left(S_{ij}S_{jk}\right)$

- Employ 7 features

$$\boldsymbol{\lambda} = \begin{bmatrix} \mathrm{Tr}(\mathbf{S}^{*2}) & \mathrm{Tr}(\boldsymbol{\Omega}^{*2}) & \mathrm{Tr}(\mathbf{S}^{*3}) & \mathrm{Tr}(\boldsymbol{\Omega}^{*2}\mathbf{S}^{*}) & \mathrm{Tr}(\boldsymbol{\Omega}^{*2}\mathbf{S}^{*2}) & \log(\mu_t/\mu) & d\frac{\sqrt{k}}{50\nu} \end{bmatrix}.$$

- Employ a tensor basis of order 4:

$$\mathbf{T}^1 = \mathbf{S}^*,\ \mathbf{T}^2 = \mathbf{S}^*\boldsymbol{\Omega}^* - \boldsymbol{\Omega}^*\mathbf{S}^*,\ \mathbf{T}_3 = \mathbf{S}^{*2} - \frac{1}{3}\mathbf{I}\ \mathrm{Tr}(\mathbf{S}^{*2})\ \mathbf{T}^4 = \boldsymbol{\Omega}^{*2} - \frac{1}{3}\mathbf{I}\ \mathrm{Tr}(\boldsymbol{\Omega}^{*2})$$

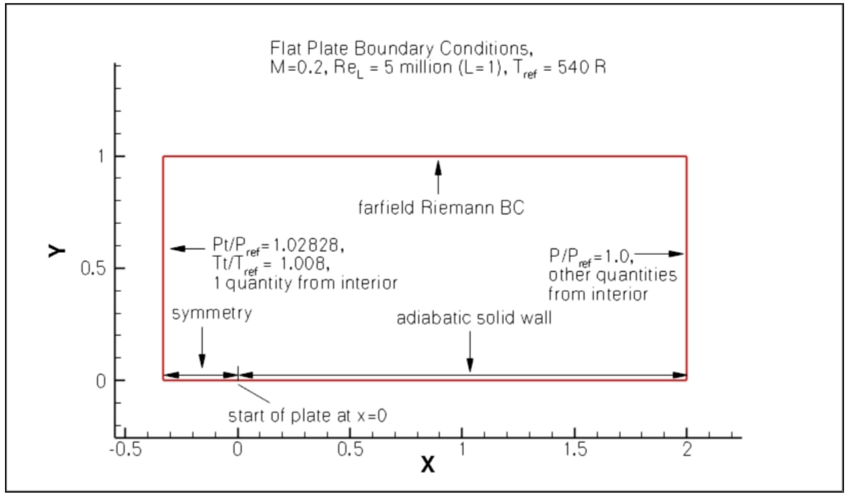
- Model architecture: MLP with depth of 5 and 40 neurons per layer
- Training: L2 penalty with regularization, 5000 epochs
- Strain rate and rotation rate were non-dimensionalized by

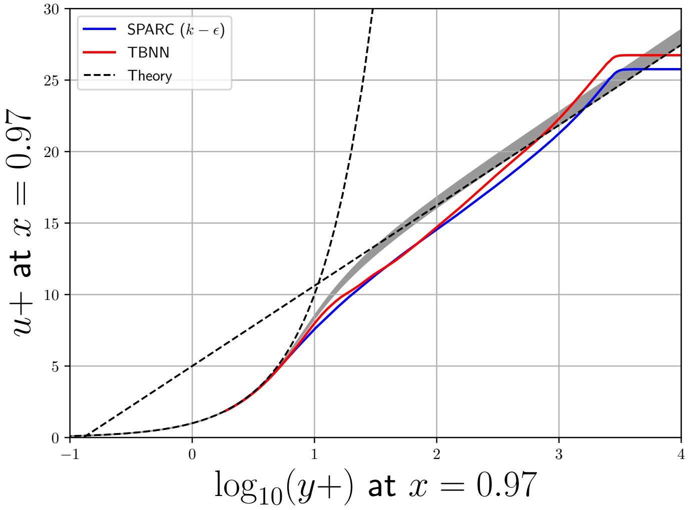
$$\mathbf{S}^* = \frac{S}{\frac{\epsilon}{k} + \|\mathbf{S}\|_2} \qquad \boldsymbol{\Omega}^* = \frac{\Omega}{\frac{\epsilon}{k} + \|\boldsymbol{\Omega}\|_2}$$

- Ensemble size of 10
- Models are trained using multistep approach outlined earlier

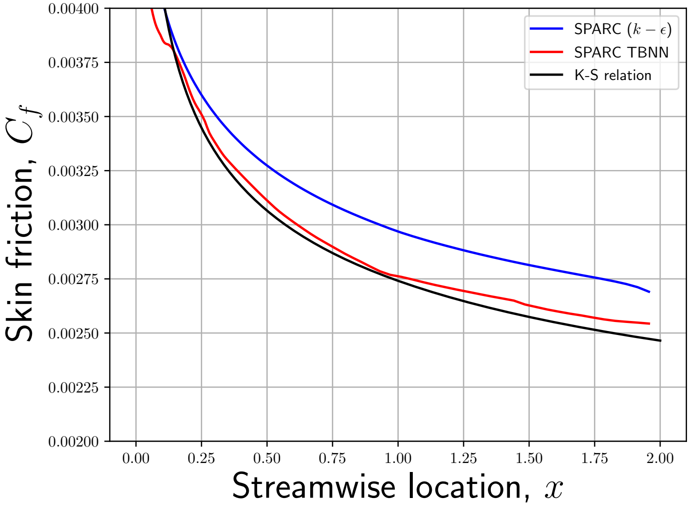
Results: Zero pressure gradient flat plate boundary layer

- Relevant training sets added:
  - Channel flow at three Reynolds numbers
  - Duct flows
- High level summary of results:
  - Improved skin friction
  - Strange “hump” in the buffer layer

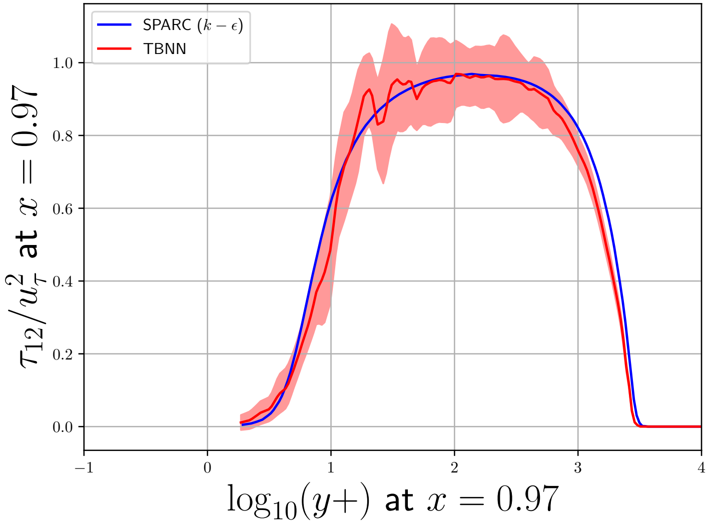




Velocity profile



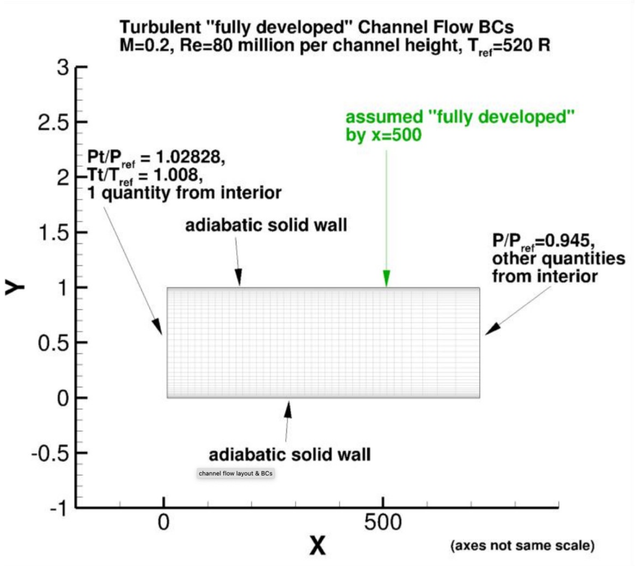
Skin friction vs. downstream location

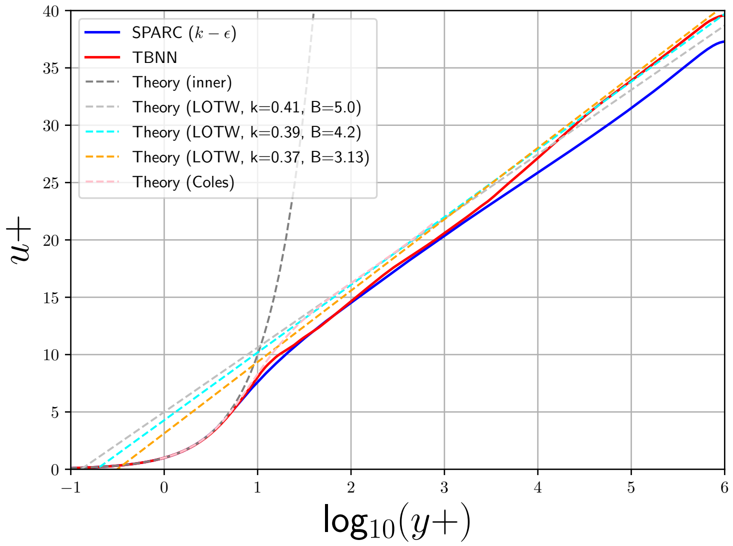


Reynolds stress w/ 1 std uncertainty

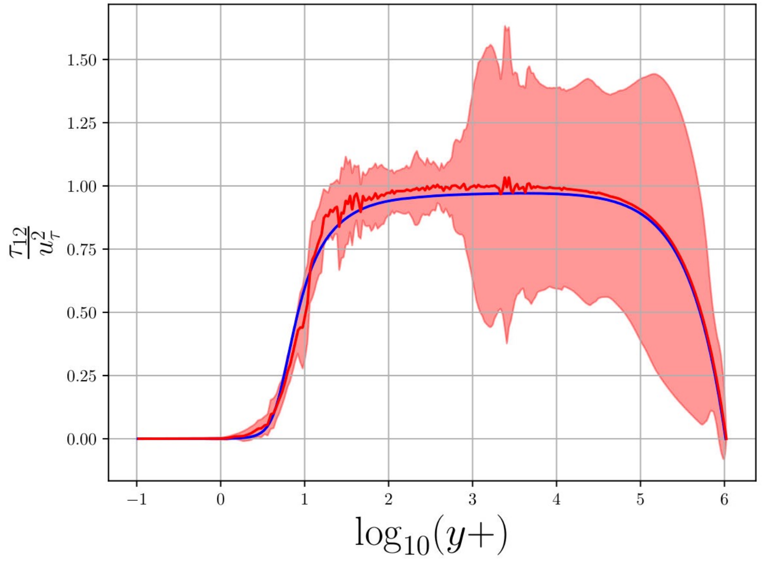
Results: Channel flow

- Relevant training sets added:
  - Channel flow at three Reynolds numbers
- Notes: Problem solved on a periodic domain
  - challenges converging regular solver (with no ML) on non-periodic domain
- High-level summary of results
  - Improved center-line velocity
  - Similar hump is observed
  - High levels of uncertainty away from the wall (no training data)





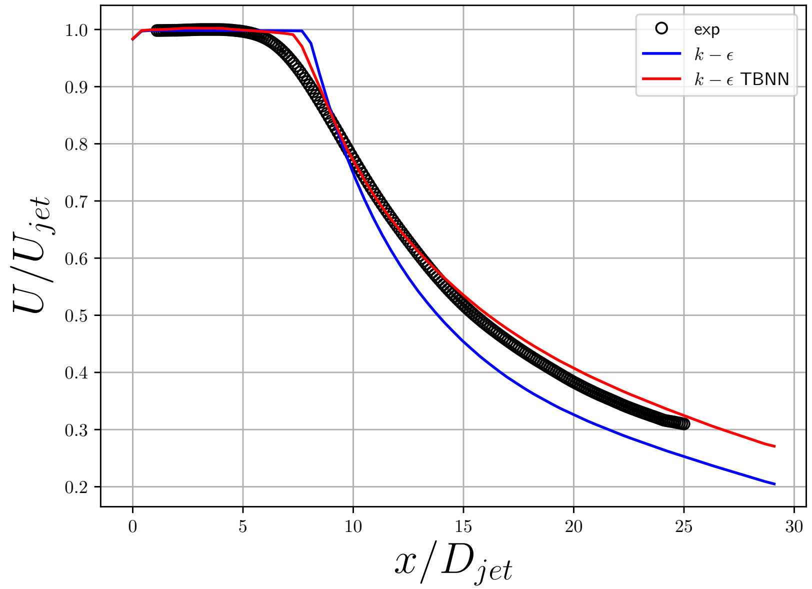
Velocity profile



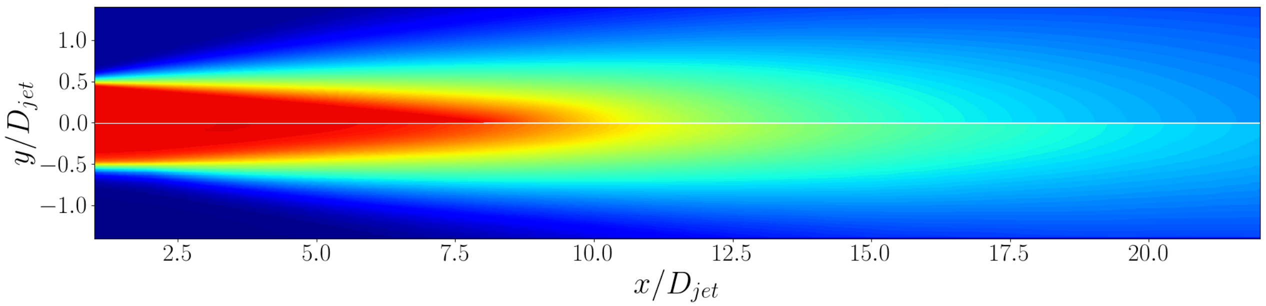
Reynolds stress w/ 1 std uncertainty

Results: Axisymmetric jet

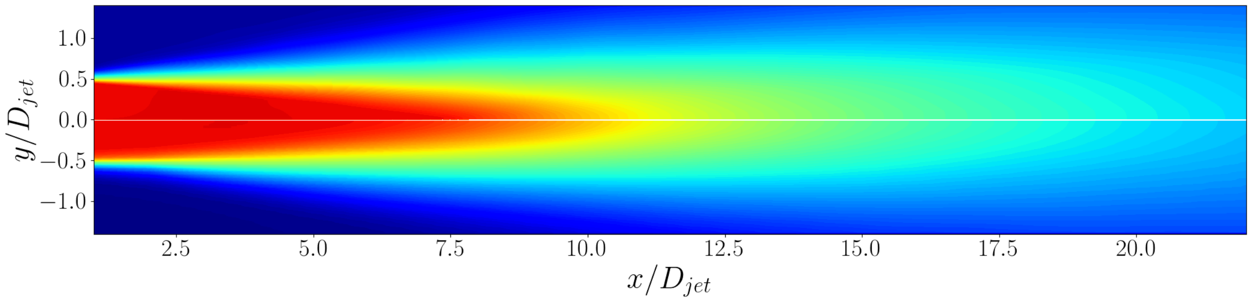
- Relevant training sets added:
  - Jet-in-crossflow (experimental)
- Notes:
  - Problem failed with out JIC data in training set
  - Difficult to fully converge the solver
- High-level summary of results
  - Improved centerline velocity
  - Mixed results on Reynolds stress
  - Overprediction of velocity at jet exterior



Centerline velocity as a function of downstream distance

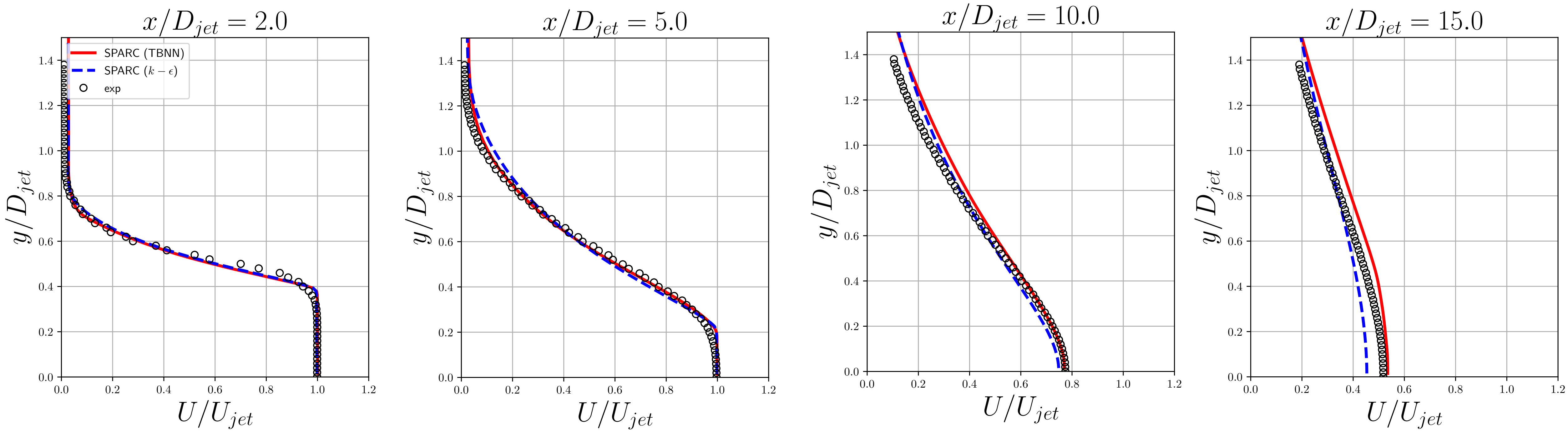


Comparison of standard k-eps (top) to experimental data (bottom)

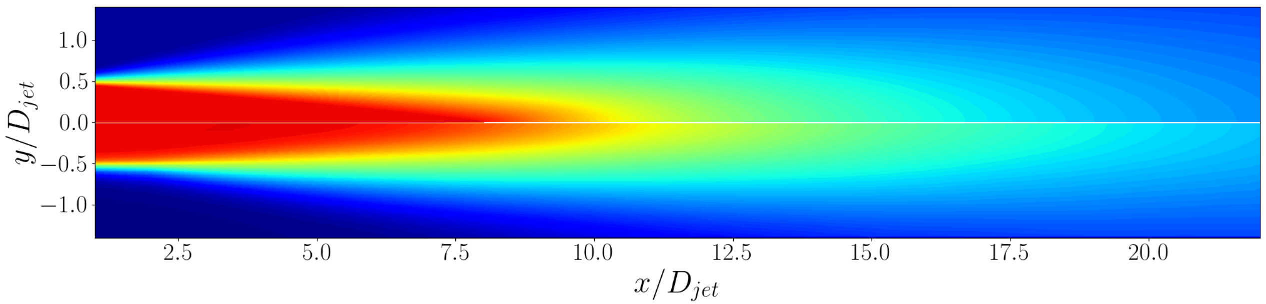


Comparison of discrepancy TBNN (top) to experimental data (bottom)

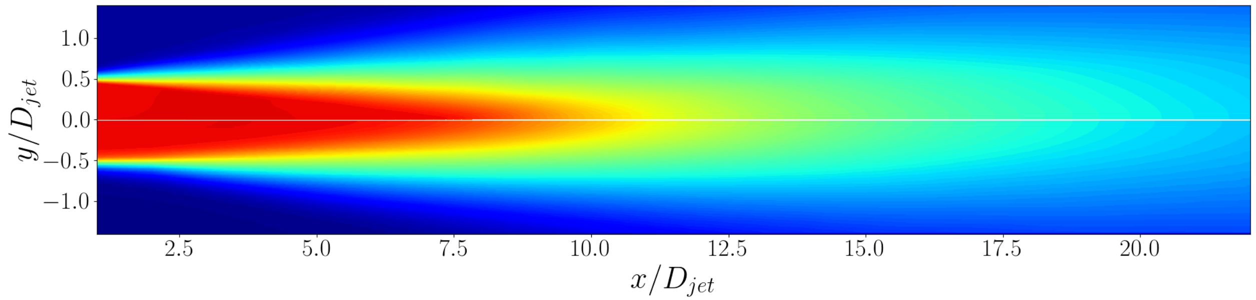
Results: Axisymmetric jet



Streamwise velocities at four different locations

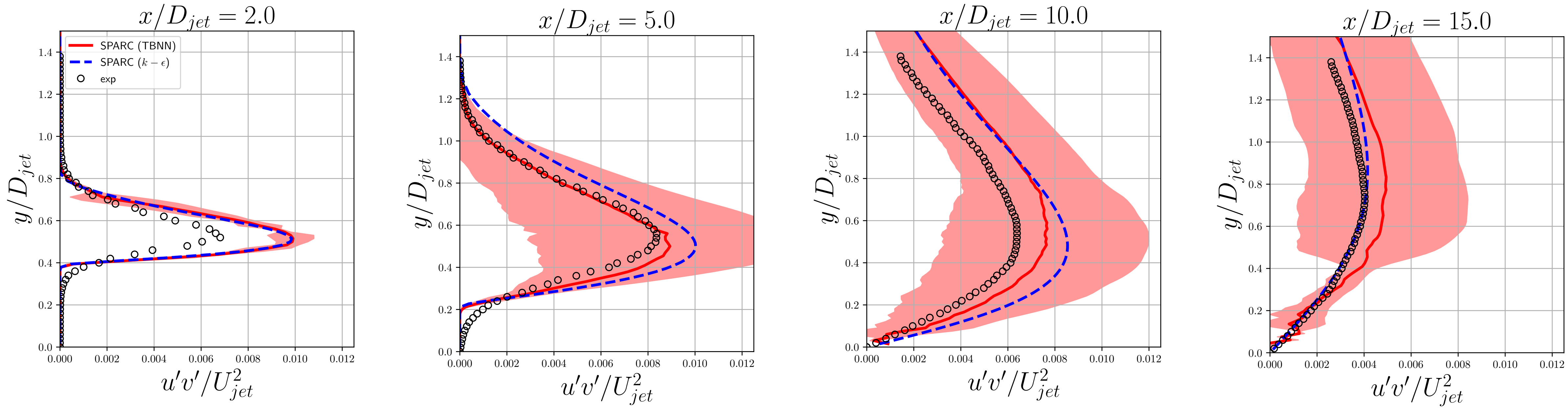


Comparison of standard k-eps (top) to experimental data (bottom)

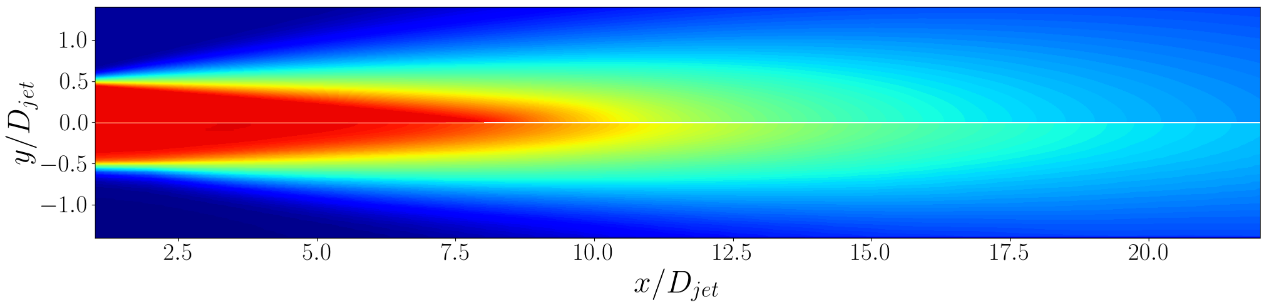


Comparison of discrepancy TBNN (top) to experimental data (bottom)

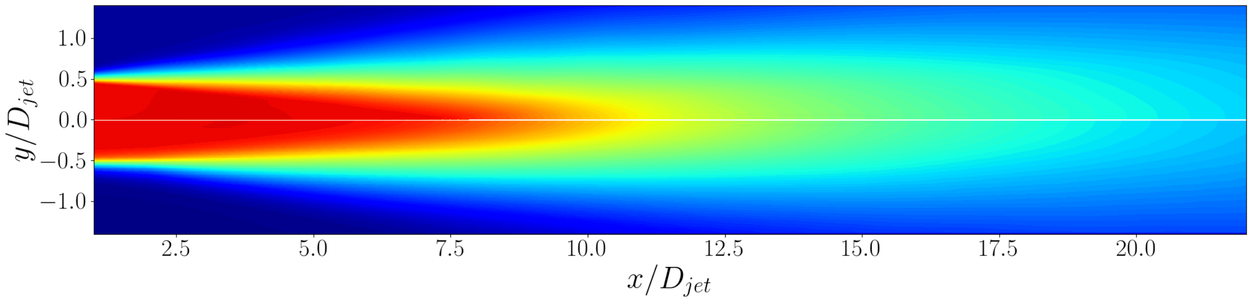
Results: Axisymmetric jet



Reynolds stresses at four different locations



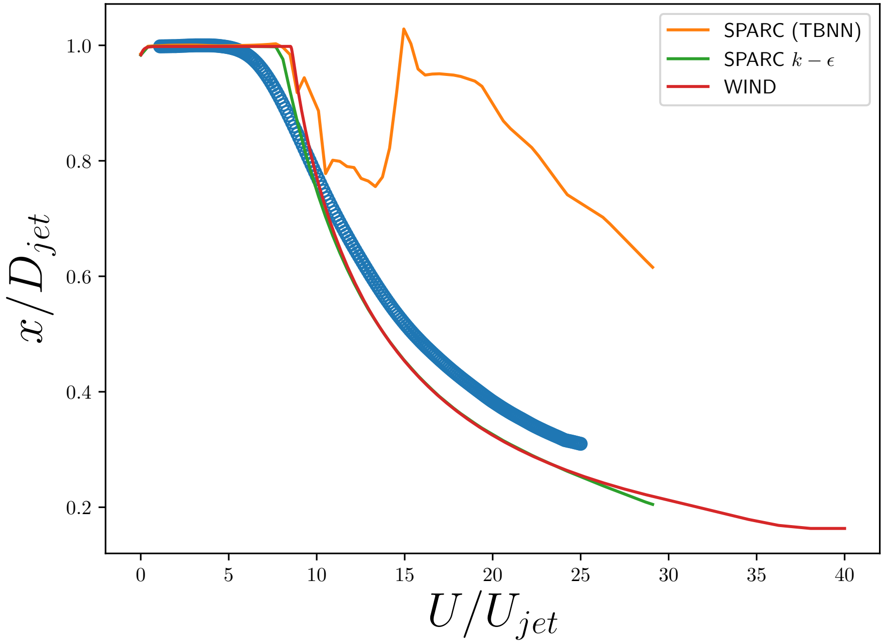
Comparison of standard k-eps (top) to experimental data (bottom)



Comparison of discrepancy TBNN (top) to experimental data (bottom)

Results: Axisymmetric jet

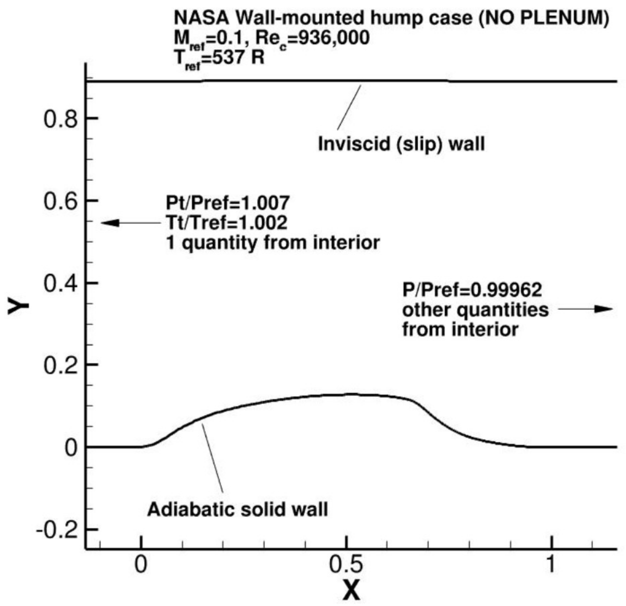
- A side notes: Problem failed without jet in crossflow data in training set
  - Ensembles were “confidently” predicting the wrong answer
    - Non-uniqueness of feature space?
  - We are still investigating this, but indicates that ensembles don’t solve all the problems

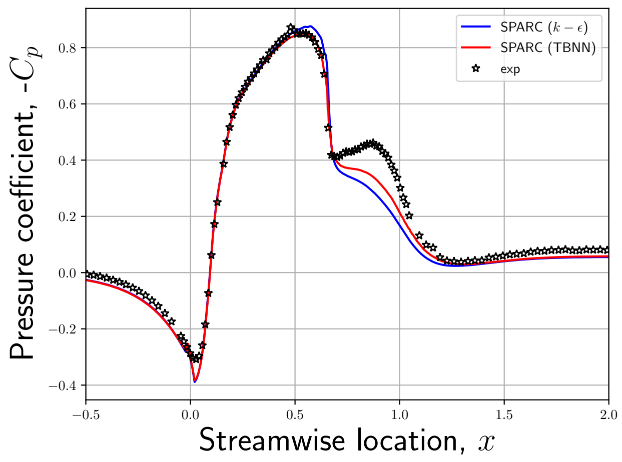


Centerline velocity as a function of downstream distance

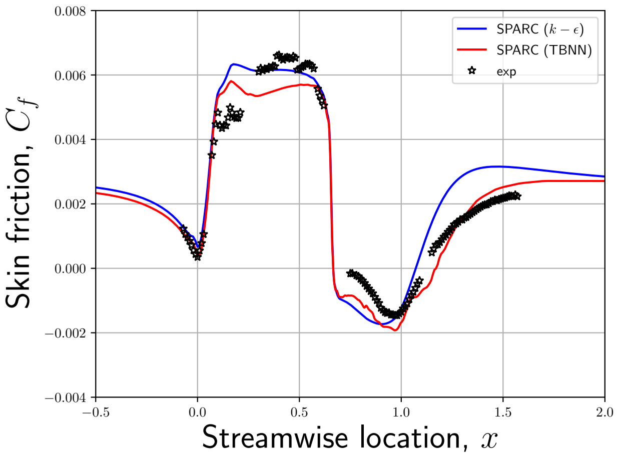
Results: Nasa hump

- Relevant training sets added:
  - Flow over periodic hill and wavy wall
- High-level summary of results
  - Improved skin friction and pressure coefficient
  - Improved prediction of separation bubble

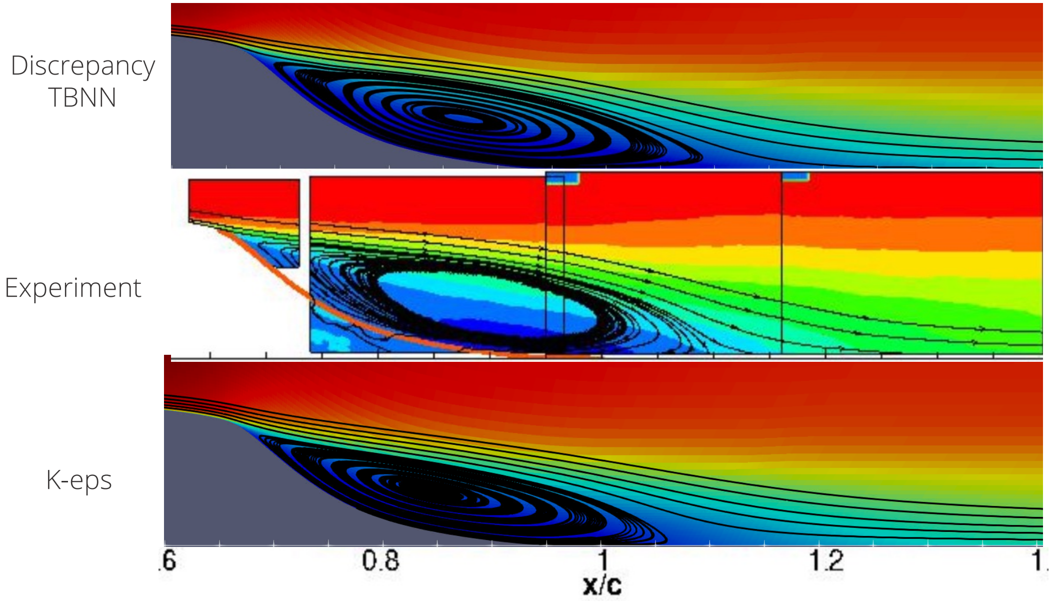




Pressure coefficient

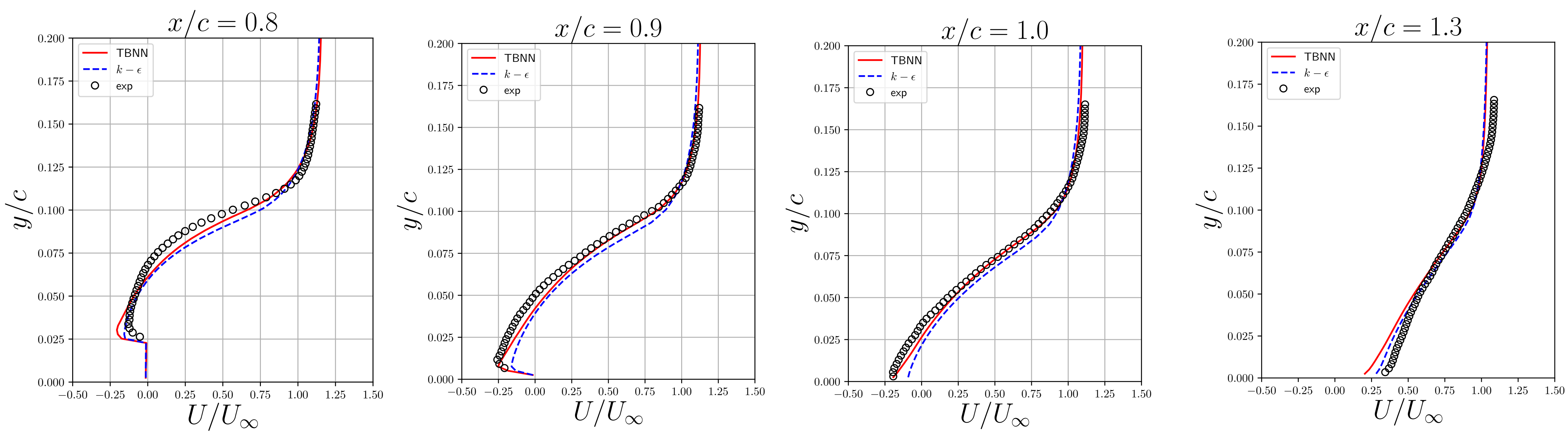


Friction coefficient

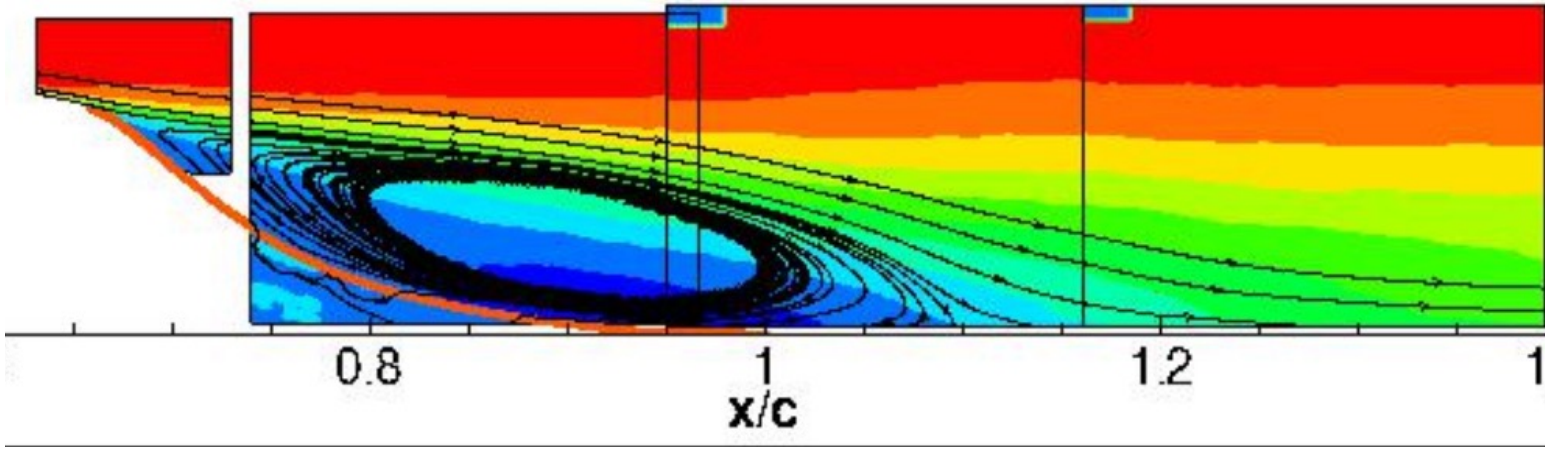


Streamwise velocity contour

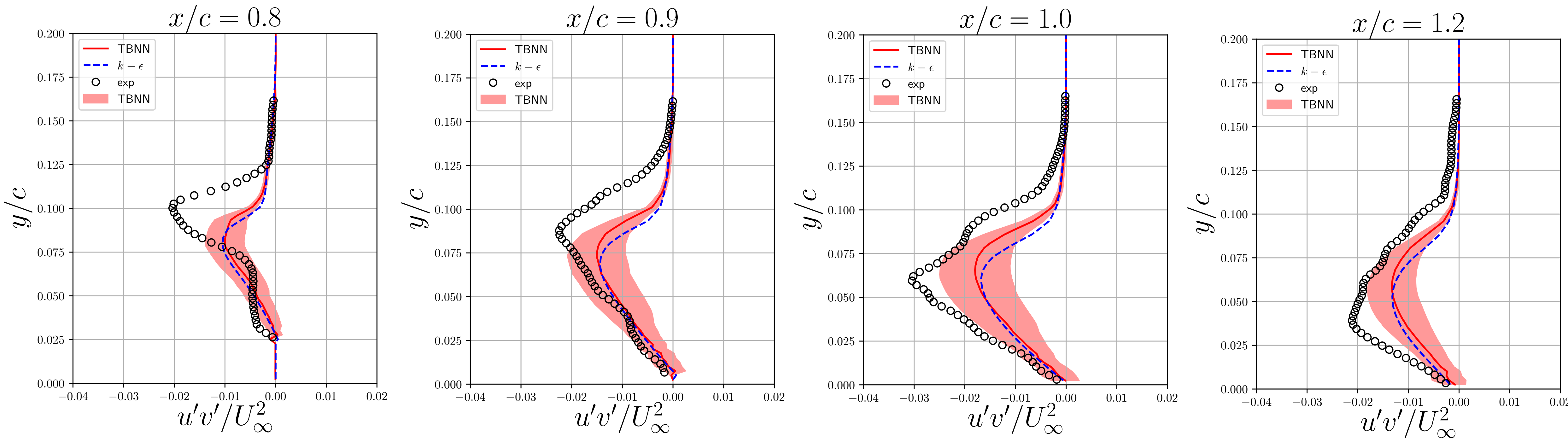
Results: Nasa hump



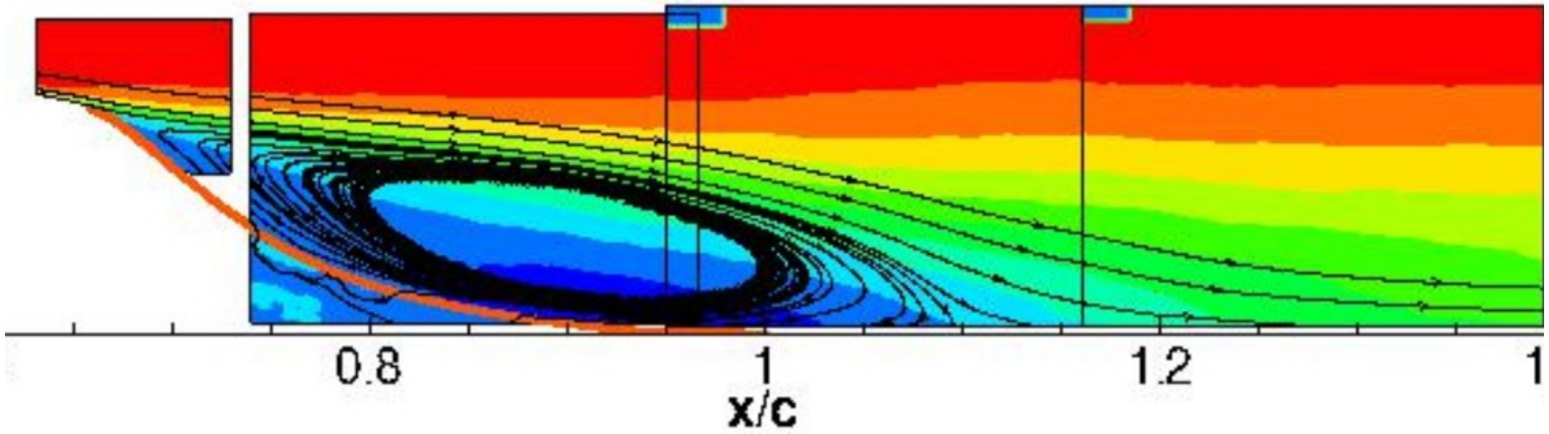
Streamwise velocity at four different locations



Results: Nasa hump

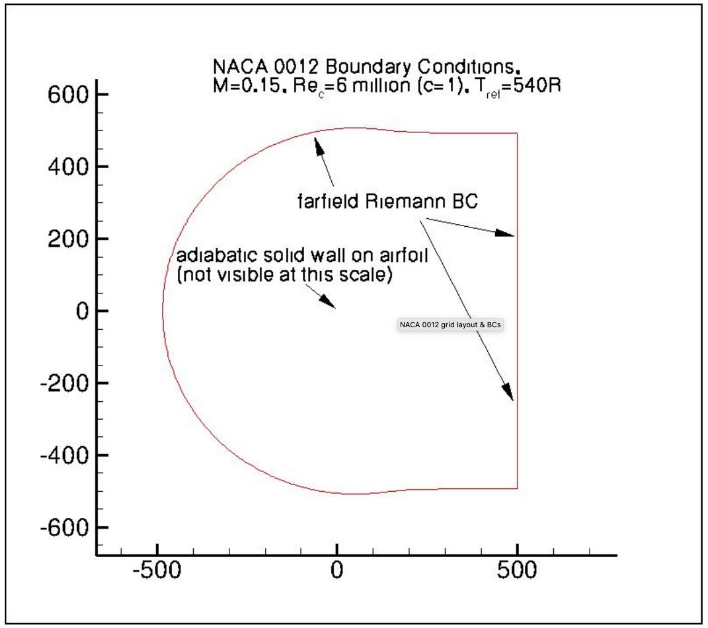


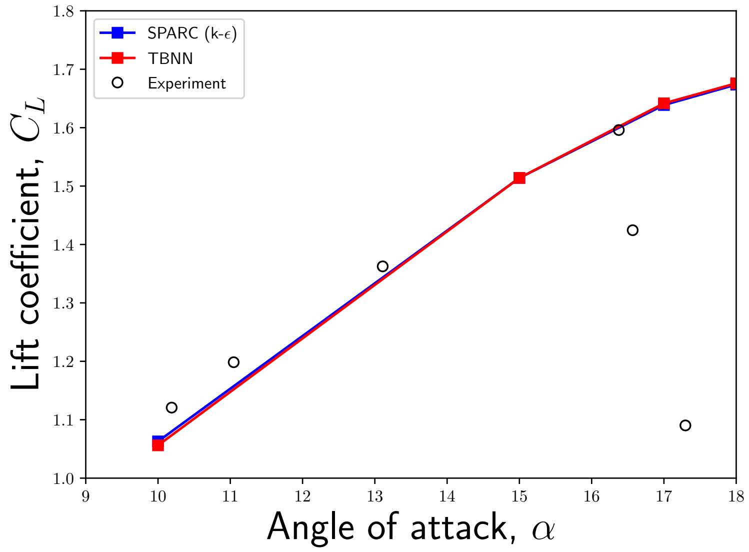
Reynolds stress at four different locations



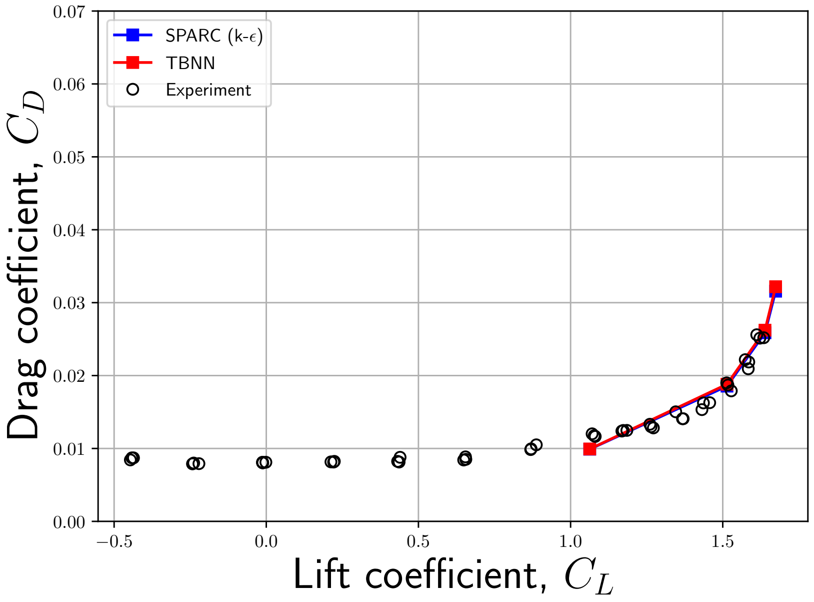
Results: NACA airfoil

- Relevant training sets added:
  - Channel flow
  - Flow over wavy wall and periodic hill
- High-level summary of results
  - Results are unchanged from the standard model
  - Indicates we need better training data for this case





Lift coefficient vs angle of attack



Drag coefficient vs. Lift coefficient

Conclusions

- Discussed our workflow for developing data-driven turbulence models for the compressible RANS equations
- Approach is based on three core concepts:
  - Discrepancy tensor-basis neural networks that correct, rather than replace, a RANS model's prediction for the Reynolds stress
  - Extrapolation detection that identifies when a model is trustworthy
  - Improved feature consistency to minimize discrepancies between the training and testing stage
- Approach was demonstrated on the NASA testing challenge
- **Positive results:**
  - QoI results tended to be better in 4/5 cases (flat plate, channel, axisymmetric jet, NASA hump)
  - QoI results were unchanged for the airfoil cases
  - Field results compared better with the PIV for the axisymmetric jet and NASA hump
  - We were able to leverage DNS, LES, and experimental data
- **Concerning results that require attention:**
  - Predicted anisotropy fields are noisy, this needs to be fixed
  - Weird bump in the transition from the inner to outer layer in wall-bounded flows (unbalanced training data set?)
  - Model failed catastrophically on axisymmetric jet without jet in crossflow training data
    - Ensembles are helpful for extrapolation detection, but don't solve the problem completely
  - Inconsistency in TKE remains an issue
  - Need more training data

Thank you!

This work was supported by Sandia LDRD 226031. This presentation describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S.
Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

[1] Ling,J.,Kurzawski,A.,andTempleton,J.,"Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, Vol. 807, 2016, p. 155–166. https://doi.org/10.1017/jfm.2016.615.

[2] Peters, E. L., Balin, R., Jansen, K. E., Doostan, A., and Evans, J. A., "S-frame discrepancy correction models for data- informed Reynolds stress closure," *Journal of Computational Physics*, Vol. 448, 2022, p. 110717. https://doi.org/https: //doi.org/10.1016/j.jcp.2021.110717, URL https://www.sciencedirect.com/science/article/pii/S0021999121006124.

[3] Pope,S.B.,"A more general effective-viscosity hypothesis," *J. Fluid Mech.*,Vol.72,No.2,1975,pp.331–340.

[4] Milani,P.M.,Ling,J.,andEaton,J.K.,"On the generality of tensor basis neural networks for turbulent scalar flux modeling," *International Communications in Heat and Mass Transfer*, Vol. 128, 2021, p. 105626. https://doi.org/https://doi.org/10.1016/j. icheatmasstransfer.2021.105626, URL https://www.sciencedirect.com/science/article/pii/S0735193321005194.

[5] BOWERSOX, R. D. W., "Extension of equilibrium turbulent heat flux models to high-speed shear flows," *Journal of Fluid Mechanics*, Vol. 633, 2009, p. 61–70. https://doi.org/10.1017/S0022112009007691.

[6] Nguyen, A., Yosinski, J., and Clune, J., "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," , 2014. https://doi.org/10.48550/ARXIV.1412.1897, URL https://arxiv.org/abs/1412.1897.

[7] Fort, S., Hu, H., and Lakshminarayanan, B., "Deep Ensembles: A Loss Landscape Perspective," , 2019. https://doi.org/10. 48550/ARXIV.1912.02757, URL https://arxiv.org/abs/1912.02757.

[8] Hoyas,S.,andJiménez,J.,"Reynolds number effects on the Reynolds-stress budgets in turbulent channels,"*Physics of Fluids*, Vol. 20, No. 10, 2008, p. 101511. https://doi.org/10.1063/1.3005862, URL https://doi.org/10.1063/1.3005862.

[9] PINELLI, A., UHLMANN, M., SEKIMOTO, A., and KAWAHARA, G., "Reynolds number dependence of mean flow structure in square duct turbulence – CORRIGENDUM," *Journal of Fluid Mechanics*, Vol. 653, 2010, p. 537–537. https://doi.org/10.1017/S0022112010001849.

[10] FRÖHLICH, J., MELLEN, C. P., RODI, W., TEMMERMAN, L., and LESCHZINER, M. A., "Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions," *Journal of Fluid Mechanics*, Vol. 526, 2005, p. 19–66. https://doi.org/10.1017/S0022112004002812.