UNITED STATES MILITARY ACADEMY
WEST POINT

Lessons from Data-driven Reynolds Stress and Turbulent Scalar Flux Closures

Roles of anisotropy, auxiliary equations, and model extrapolation

Andrew J. Banko

Assistant Professor
United States Military Academy

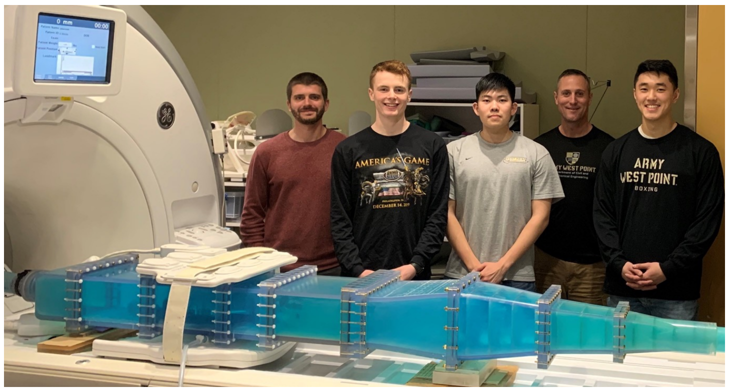
July 27-29, 2022

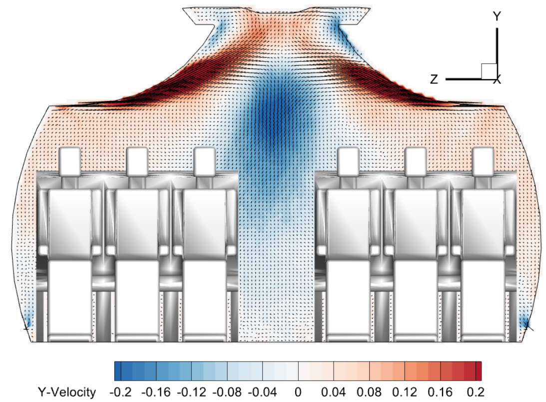
andrew.banko@westpoint.edu

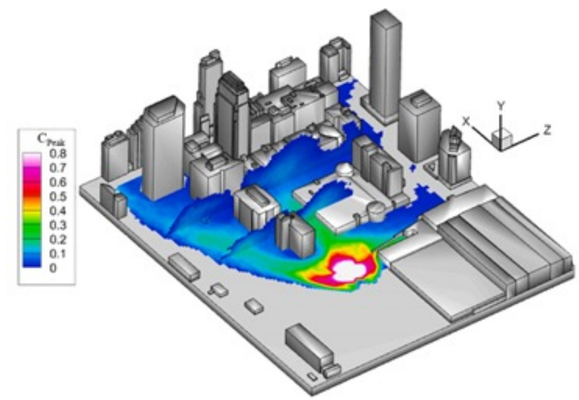
Research Objectives

Understanding heat and mass transfer in complex, 3D flows …

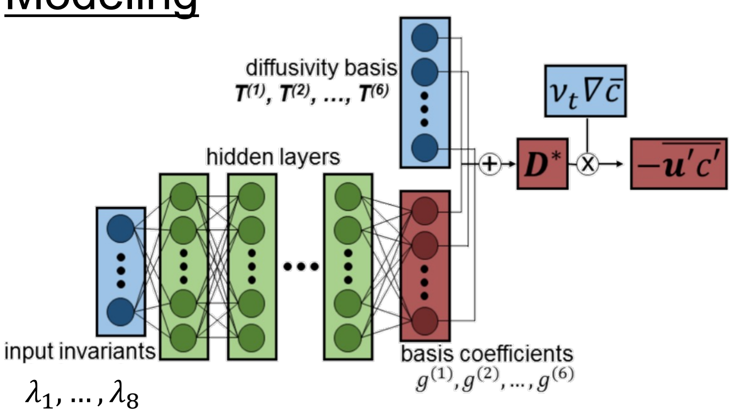
Experiments   3D mean flow measurements using magnetic resonance imaging







Modeling

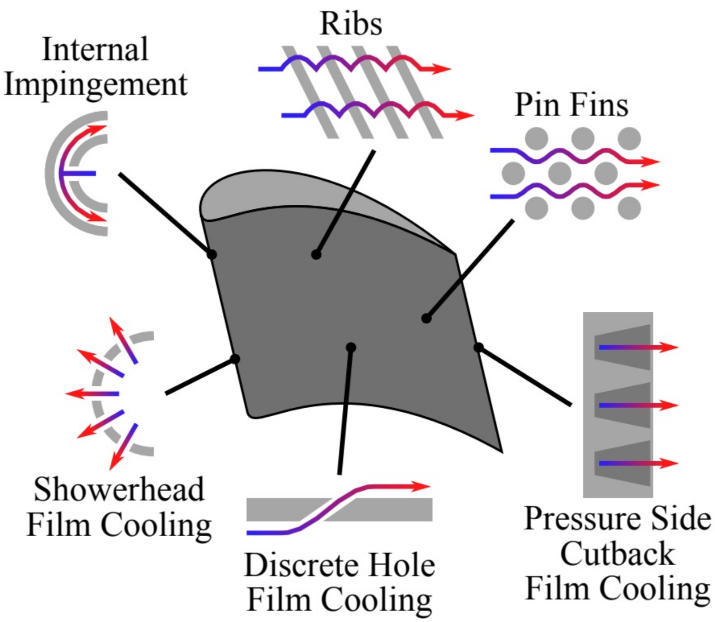


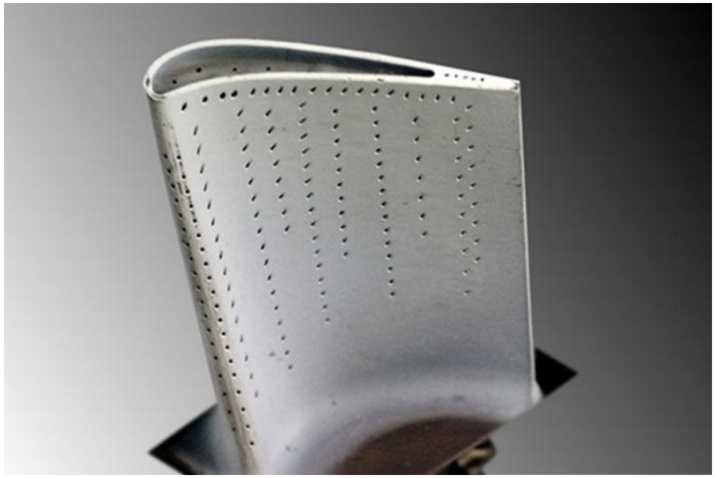
Key Questions

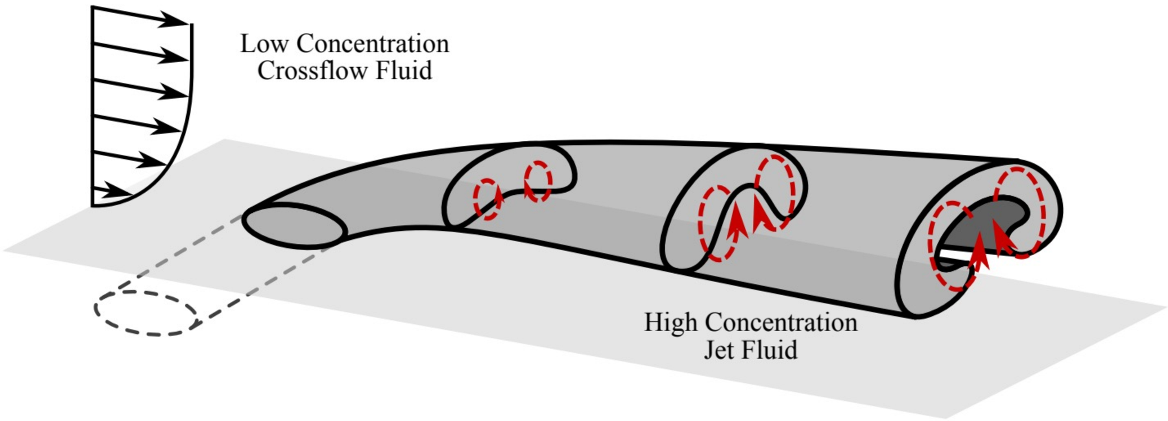
- Relative importance of anisotropy and auxiliary equations
- Performance bounds of model forms
- Data requirements and dealing with extrapolation

Gas Turbine Film Cooling

Blade surfaces are actively cooled using arrays of round and shaped holes



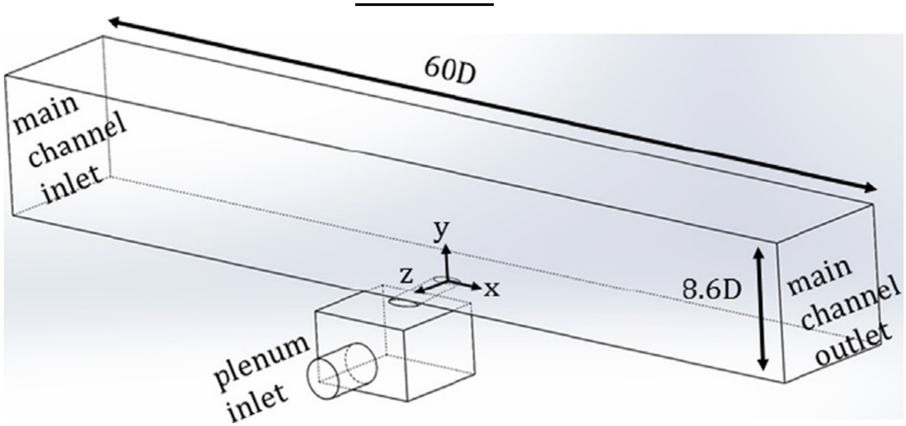




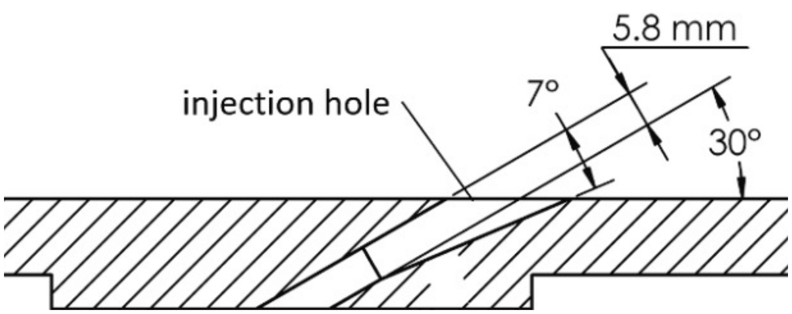
LES Database

- Family of jet in crossflows

Domain

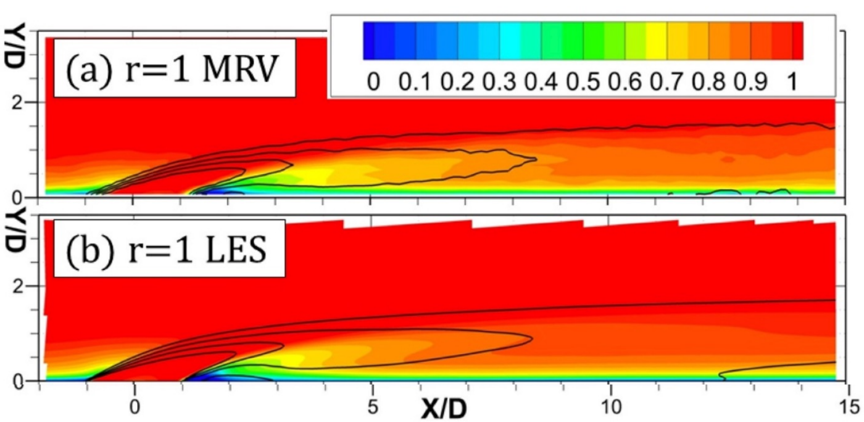


777 Diffuser Hole



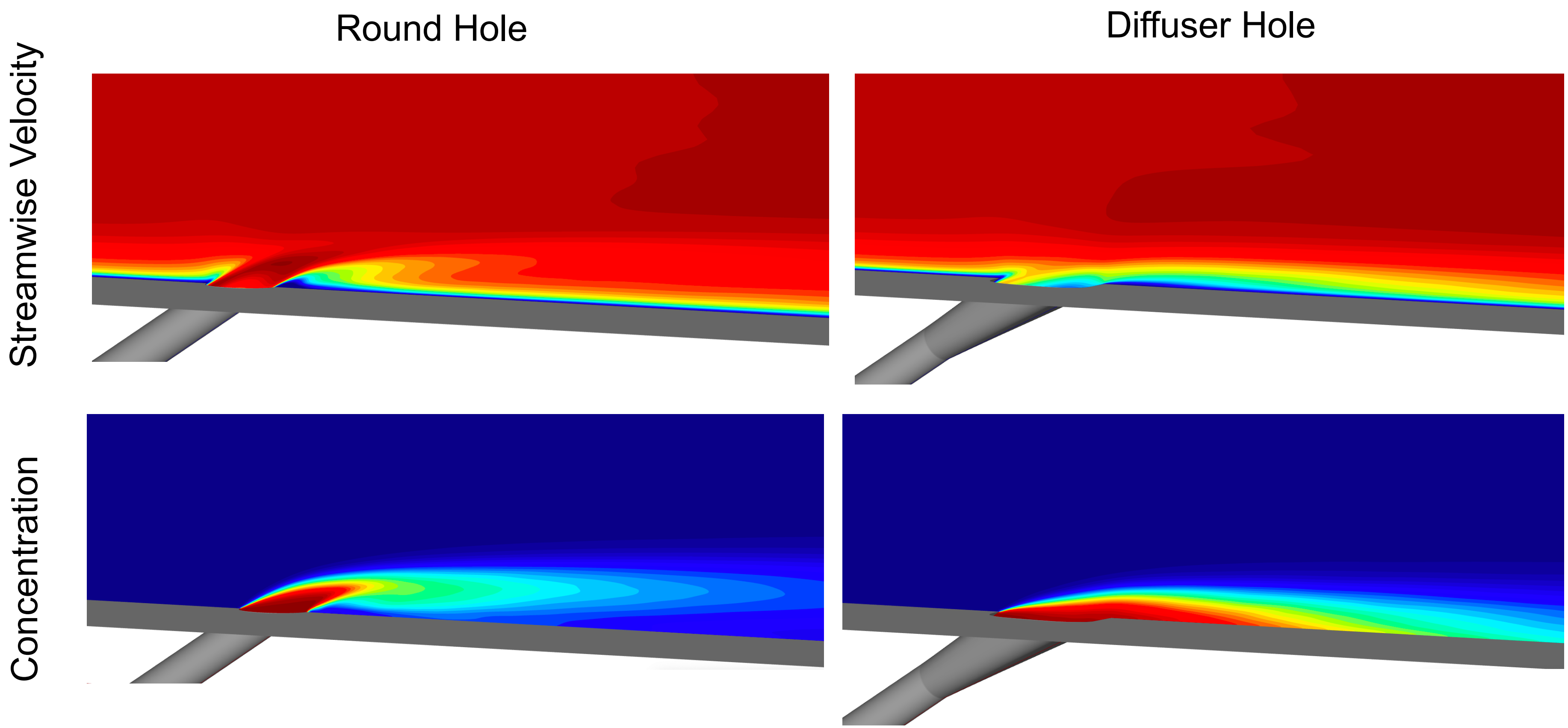
- Conditions
  - Grid: 40M cells, wall-resolved on bottom/hole
  - SGS Model: Vreman, ν_{SGS}/ν < 1
  - Main flow: Re_H = 25,000
    Re_θ = 2,500
    δ_{99}/D = 1.5
  - Jet: BR = 1 (based on metering hole)
    Re_D = 2,900

- Validation against 3D MRI data



Flow Field Overview

- Round hole has a strong counter-rotating vortex pair
- Diffuser hole bleeds low momentum fluid adjacent to the surface

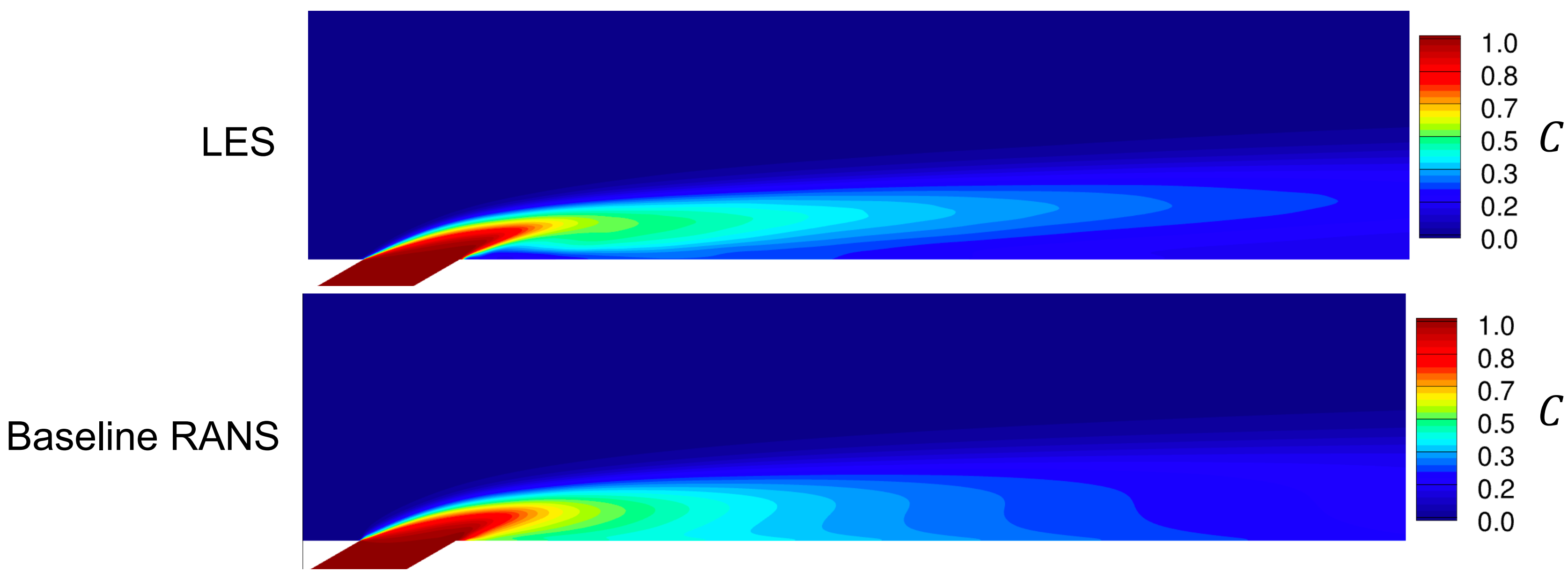


Baseline RANS

- Focus on scalar transport equation with $k-\epsilon$ as a baseline

$$\boldsymbol{U}\cdot\nabla C=\alpha\nabla^2 C-\nabla\cdot\langle c'\boldsymbol{u}'\rangle \quad \text{where} \quad \langle c'\boldsymbol{u}'\rangle=-\frac{\nu_T}{Pr_t}\nabla C \quad \text{and} \quad Pr_t\approx 0.85$$

- Isolate turbulent scalar flux model
  - Same grid as LES
  - Use LES mean velocity
  - Determine k and ϵ by solving their transport equations using the LES velocity field



ML Turbulent Scalar Flux Modeling

- Turbulent scalar flux: $\langle c' \boldsymbol{u}' \rangle = -\nu_T \boldsymbol{D}^* \nabla C$ $\quad \boldsymbol{D}^* = \boldsymbol{F}(\boldsymbol{S}, \boldsymbol{R})$ $\quad \boldsymbol{S}, \boldsymbol{R}$ normalized by k/ϵ

[1]Milani et al., JFM, 2021
[2]Banko et al., In prep.
[3]Zheng, 1994

ML Turbulent Scalar Flux Modeling

- Turbulent scalar flux: $\langle c' \boldsymbol{u}' \rangle = -\nu_T \boldsymbol{D}^* \nabla C$ $\boldsymbol{D}^* = \boldsymbol{f}(\boldsymbol{S}, \boldsymbol{R})$ $\boldsymbol{S}, \boldsymbol{R}$ normalized by k/ϵ
- Tensor basis neural network (TBNN-s)[1,2]
  - Input invariants

$$\lambda_1 = tr(\boldsymbol{S}^2) \quad \lambda_3 = tr(\boldsymbol{S}^3) \quad \lambda_5 = tr\left(\boldsymbol{S}^2\boldsymbol{R}^2\right) \quad \lambda_7 = \sqrt{k}d/\nu$$

$$\lambda_2 = tr(\boldsymbol{R}^2) \quad \lambda_4 = tr(\boldsymbol{S}\boldsymbol{R}^2) \quad \lambda_6 = tr(\boldsymbol{S}^2\boldsymbol{R}^2\boldsymbol{S}\boldsymbol{R}) \quad \lambda_8 = \nu_T/\nu$$

  - Tensor basis (derived from vector basis[3])

$$\boldsymbol{T}^{(1)} = \boldsymbol{I} \quad \boldsymbol{T}^{(3)} = \boldsymbol{R} \quad \boldsymbol{T}^{(5)} = \boldsymbol{R}^2$$

$$\boldsymbol{T}^{(2)} = \boldsymbol{S} \quad \boldsymbol{T}^{(4)} = \boldsymbol{S}^2 \quad \boldsymbol{T}^{(6)} = \boldsymbol{S}\boldsymbol{R} + \boldsymbol{R}\boldsymbol{S}$$

[1]Milani et al., JFM, 2021
[2]Banko et al., In prep.
[3]Zheng, 1994

ML Turbulent Scalar Flux Modeling

- Turbulent scalar flux: $\langle c'\boldsymbol{u}'\rangle = -\nu_T \boldsymbol{D}^* \nabla C$ $\quad \boldsymbol{D}^* = \boldsymbol{f}(\boldsymbol{S},\boldsymbol{R})$ $\quad \boldsymbol{S},\boldsymbol{R}$ normalized by k/ϵ
- Tensor basis neural network (TBNN-s)[1,2]
  - Input invariants

$$\lambda_1 = tr(\boldsymbol{S}^2) \quad \lambda_3 = tr(\boldsymbol{S}^3) \quad \lambda_5 = tr(\boldsymbol{S}^2\boldsymbol{R}^2) \quad \lambda_7 = \sqrt{k}d/\nu$$

$$\lambda_2 = tr(\boldsymbol{R}^2) \quad \lambda_4 = tr(\boldsymbol{S}\boldsymbol{R}^2) \quad \lambda_6 = tr(\boldsymbol{S}^2\boldsymbol{R}^2\boldsymbol{S}\boldsymbol{R}) \quad \lambda_8 = \nu_T/\nu$$

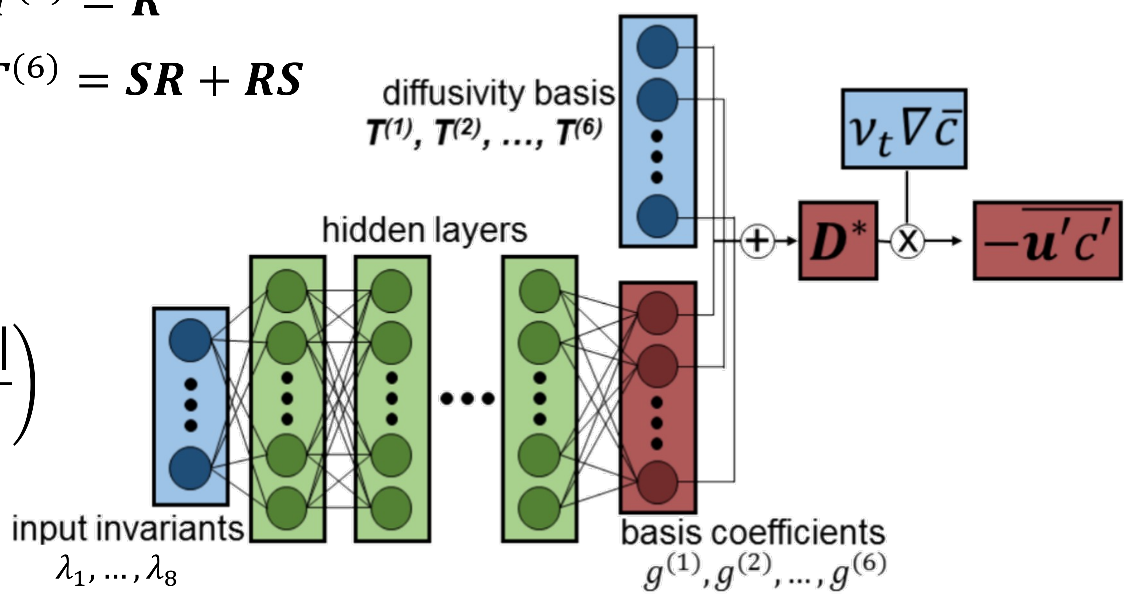
  - Tensor basis (derived from vector basis[3])

$$\boldsymbol{T}^{(1)} = \boldsymbol{I} \quad \boldsymbol{T}^{(3)} = \boldsymbol{R} \quad \boldsymbol{T}^{(5)} = \boldsymbol{R}^2$$

$$\boldsymbol{T}^{(2)} = \boldsymbol{S} \quad \boldsymbol{T}^{(4)} = \boldsymbol{S}^2 \quad \boldsymbol{T}^{(6)} = \boldsymbol{S}\boldsymbol{R} + \boldsymbol{R}\boldsymbol{S}$$

  - 8 layers, 20 nodes/layer
  - Loss function

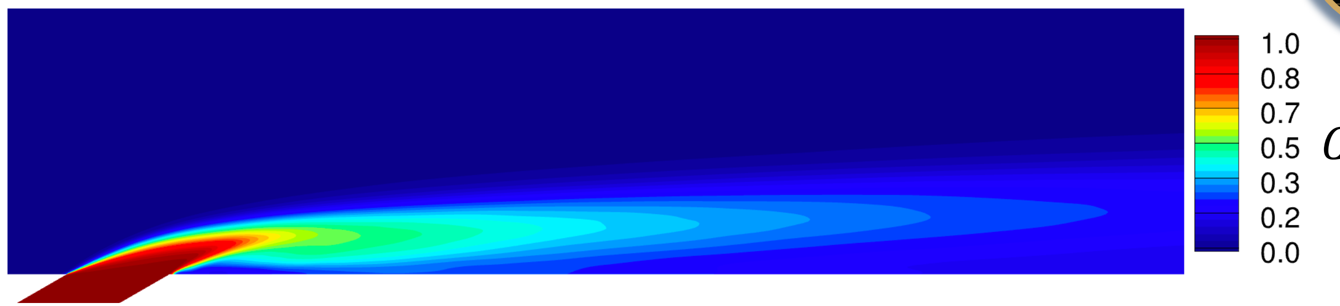
$$L = log\left(\frac{\|\langle \boldsymbol{u}'c'\rangle_{ML} - \langle \boldsymbol{u}'c'\rangle_{LES}\|}{\|\langle \boldsymbol{u}'c'\rangle_{LES}\|}\right)$$



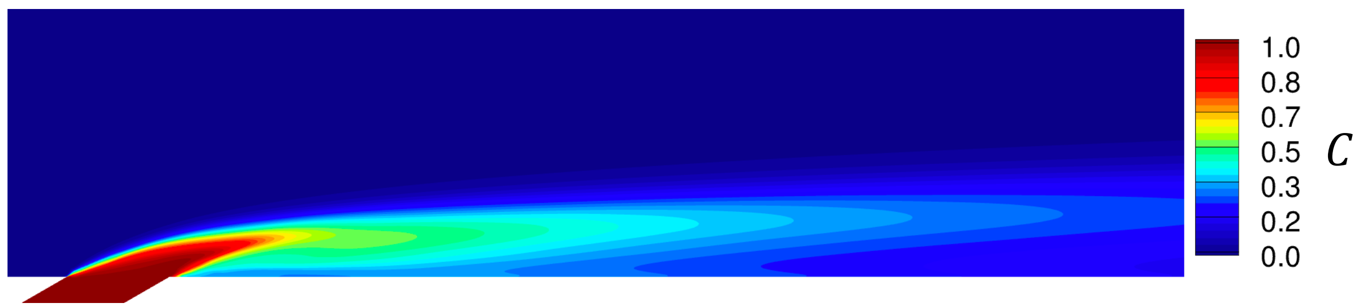
[1]Milani et al., JFM, 2021
[2]Banko et al., In prep.
[3]Zheng, 1994

Round Hole Results

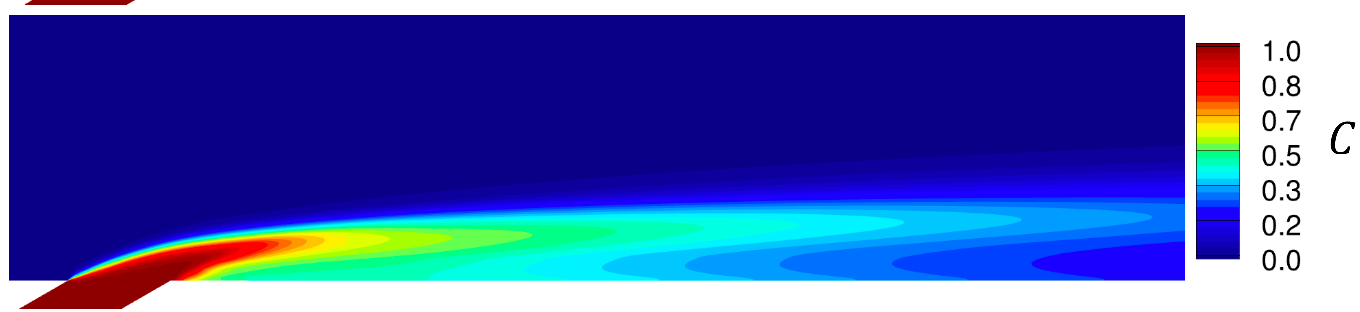
LES



Trained on Round Hole



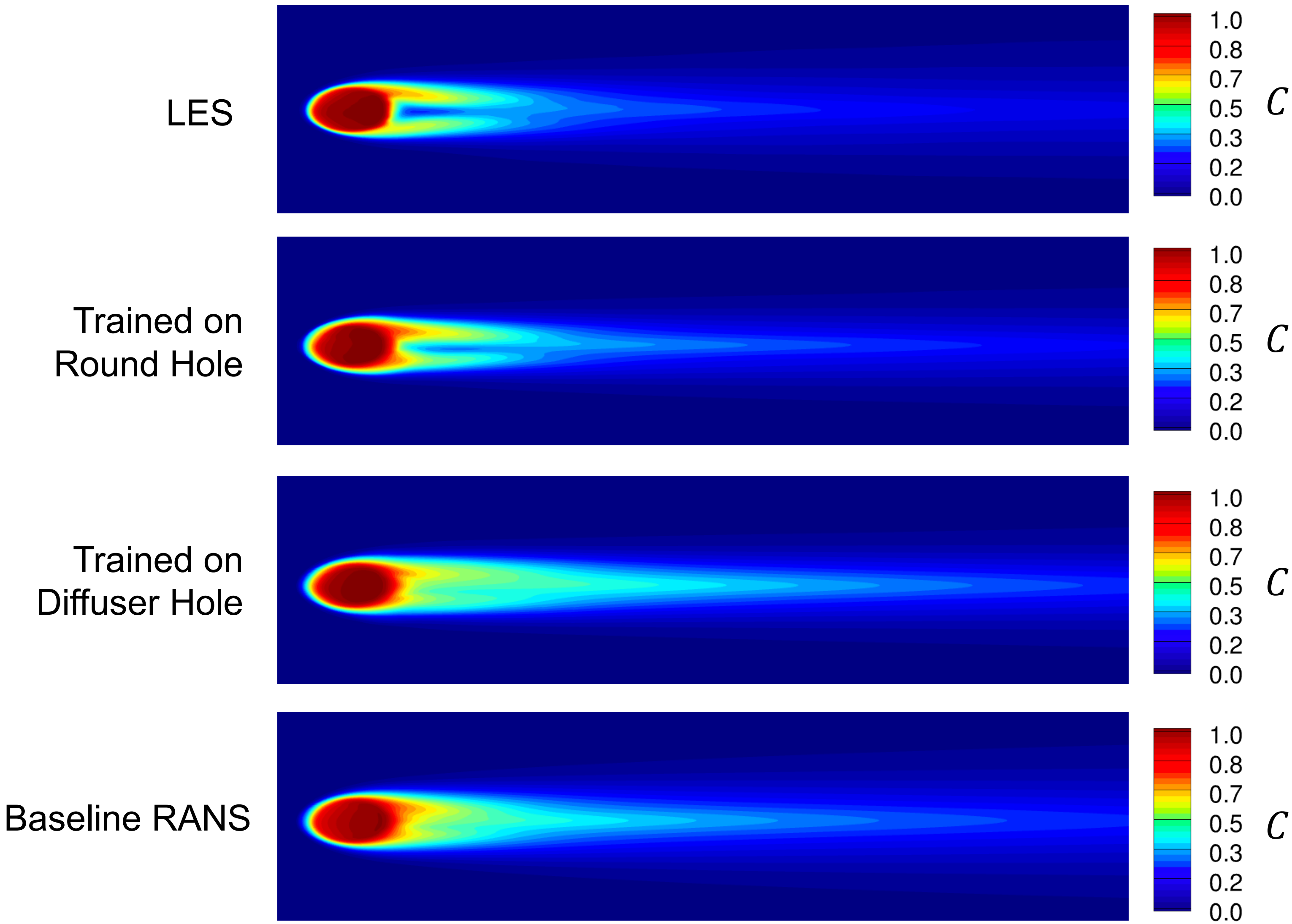
Trained on Diffuser Hole



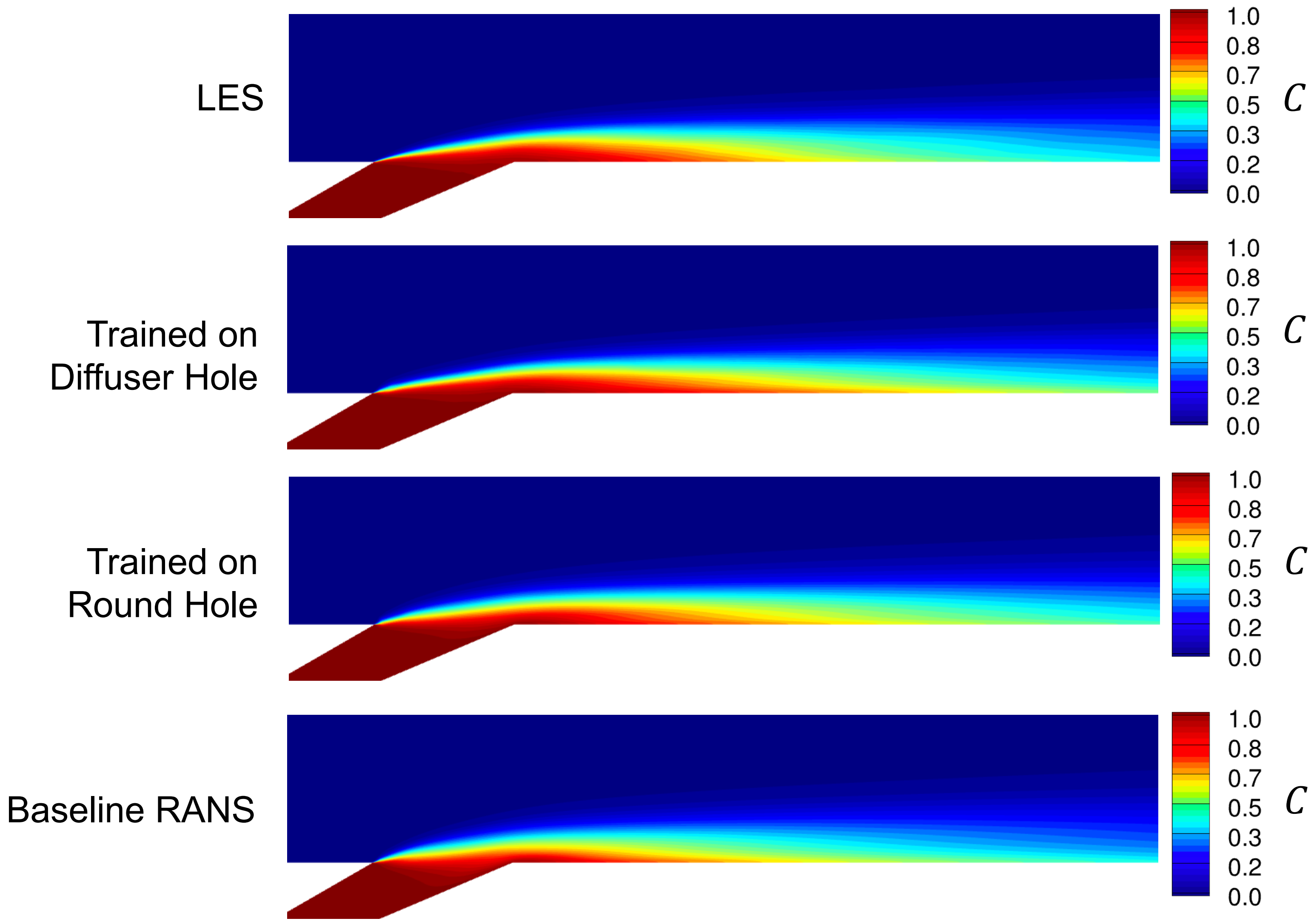
Baseline RANS

UNITED STATES MILITARY ACADEMY
WEST POINT

Round Hole Results



Diffuser Hole Results



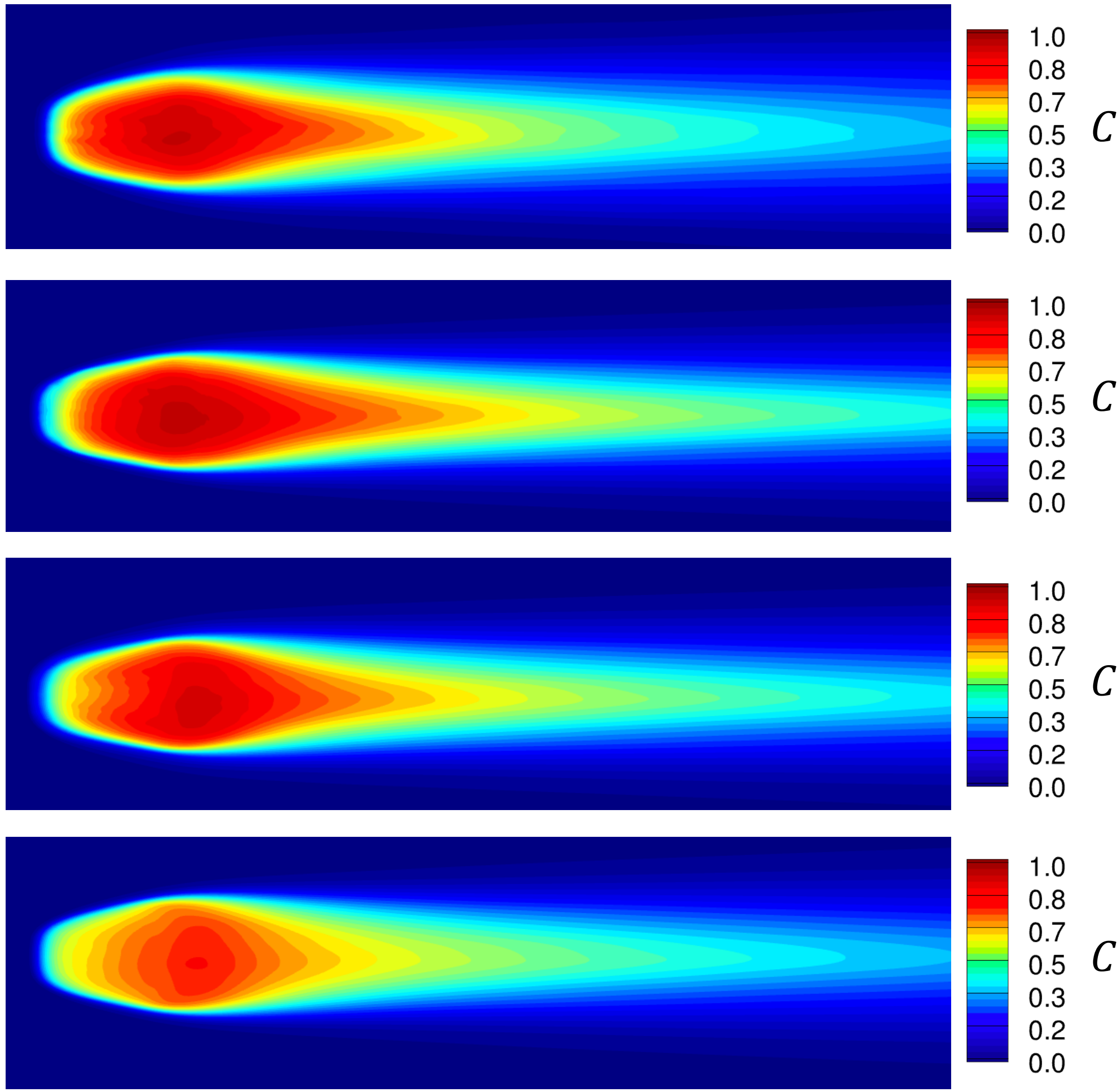
Diffuser Hole Results

LES

Trained on Diffuser Hole

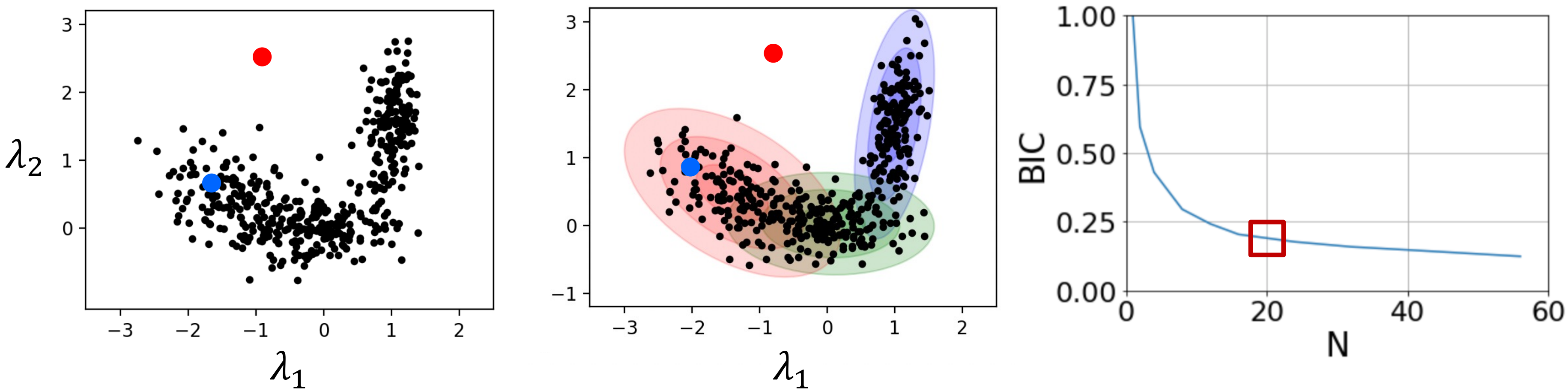
Trained on Round Hole

Baseline RANS



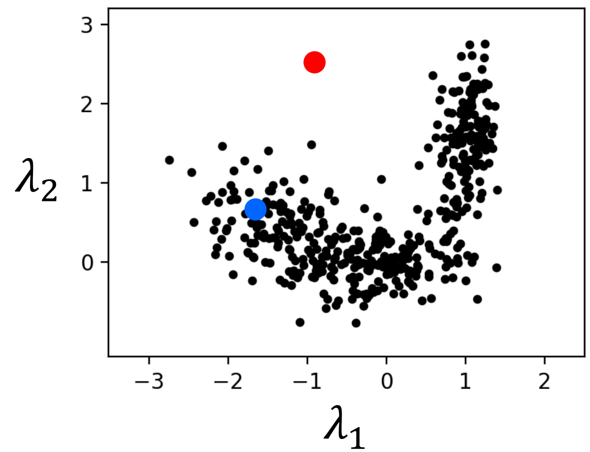
Extrapolation Detection

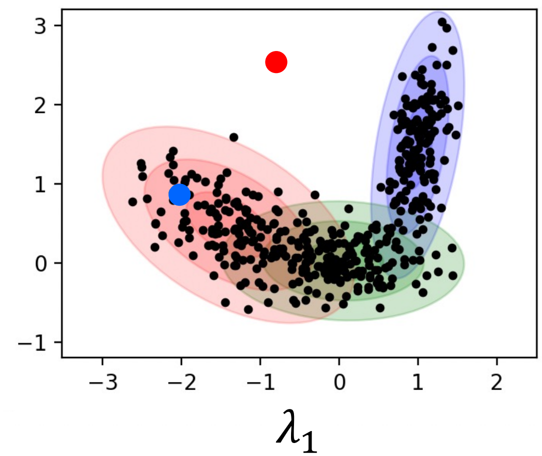
- Gaussian mixture model

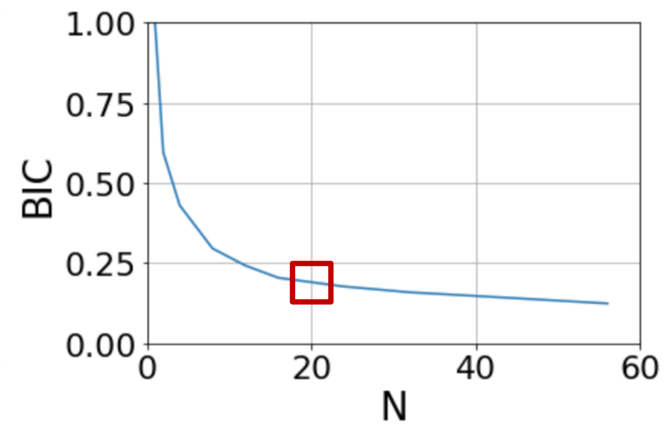


Extrapolation Detection

- Gaussian mixture model

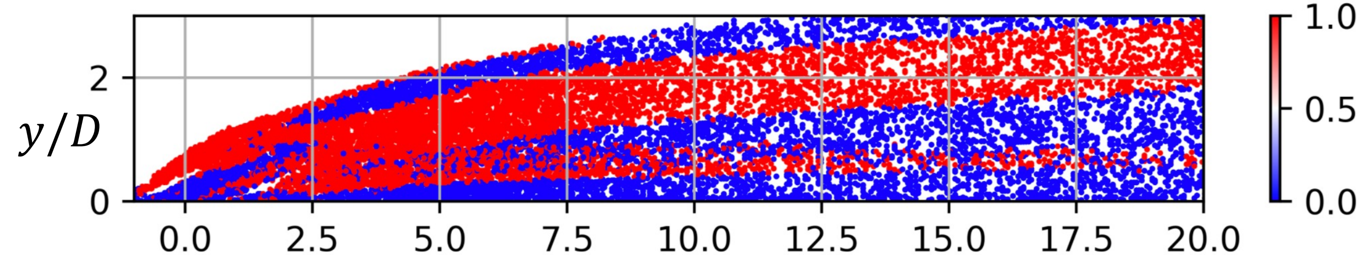




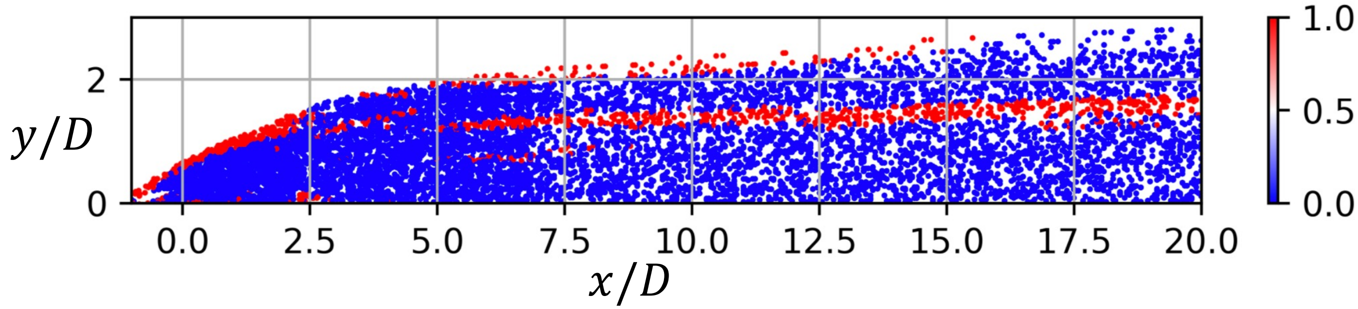


- Extrapolatory points: Blue = interpolating Red = extrapolating

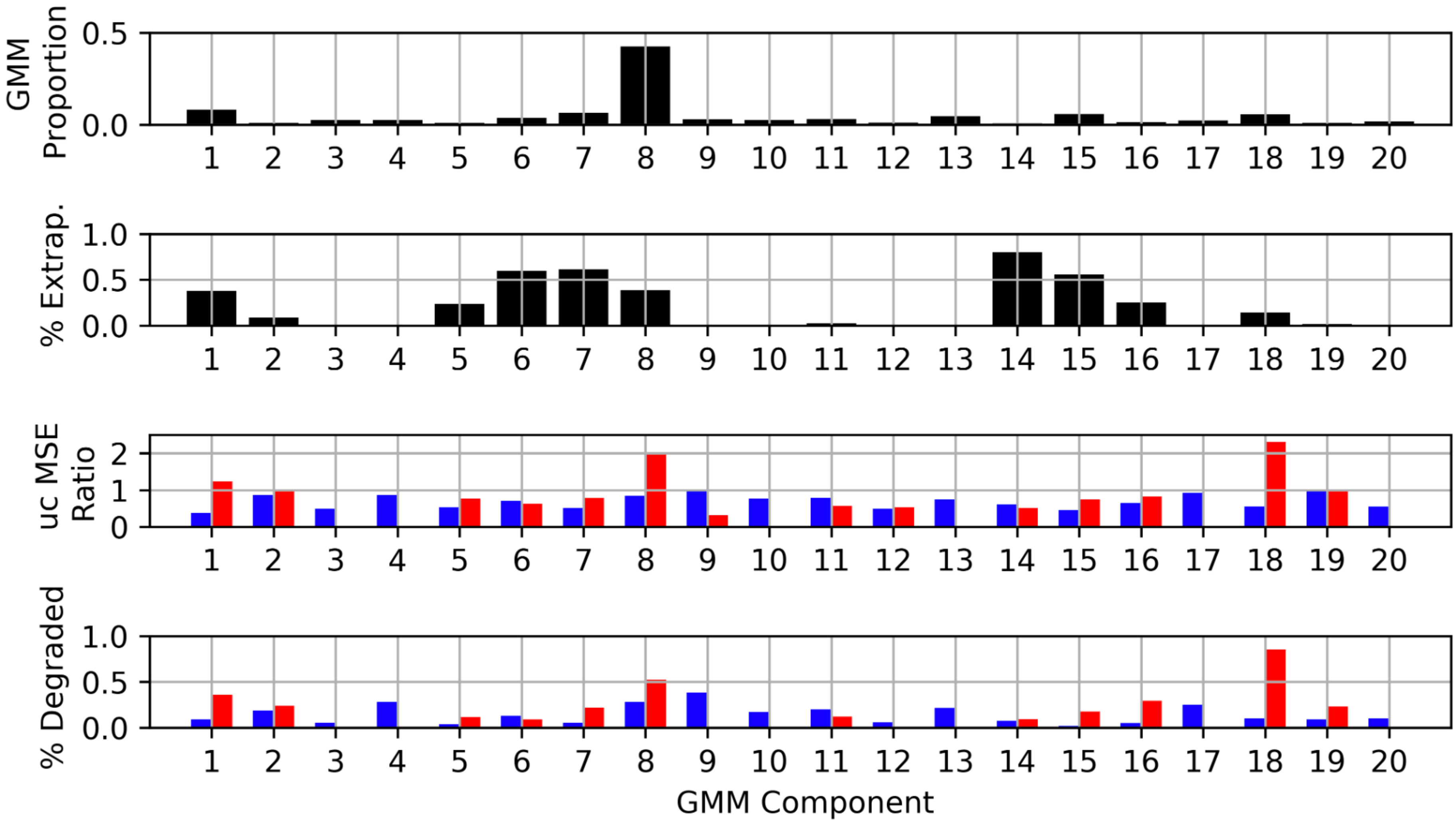
Round Hole
(Train on Diffuser)



Diffuser
(Train on Round)



Breakdown by Gaussian



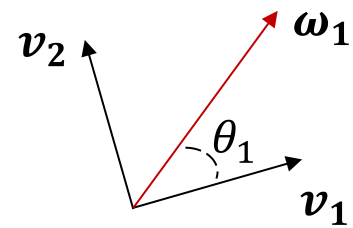
Blue = interpolating Red = extrapolating

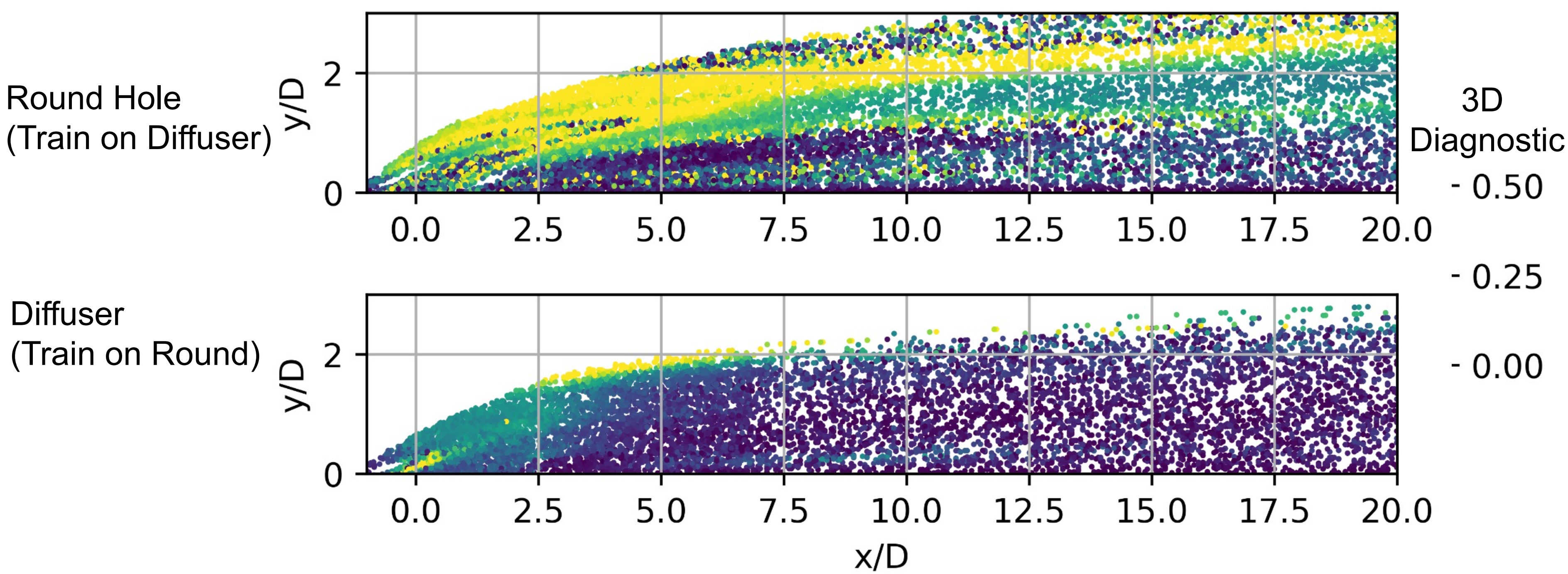
Flow Dimensionality

- Detecting statistically 2D vs. 3D flow

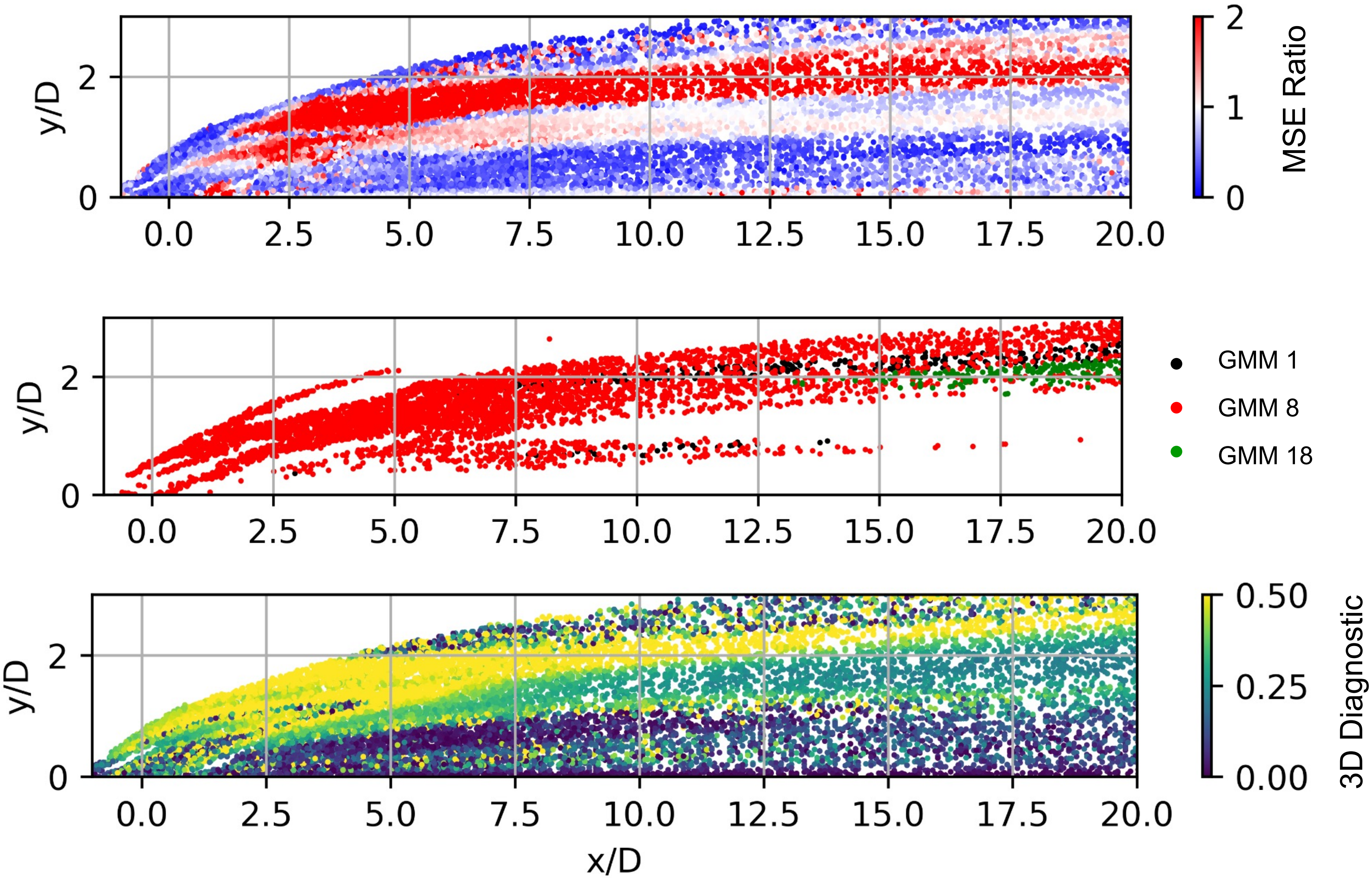
$$\lambda_5 = tr(\boldsymbol{S}^2\boldsymbol{R}^2) = \frac{1}{4}\omega^2 \sum_{i=1}^{3} \cos^2\theta_i\, s_i \;\; = \begin{cases} \neq 0 & \text{if 3D} \\ = 0 & \text{if 2D} \end{cases}$$

$$\boldsymbol{S} = \boldsymbol{V} \begin{bmatrix} \boldsymbol{s_1} & & \\ & \boldsymbol{s_2} & \\ & & \boldsymbol{s_3} \end{bmatrix} \boldsymbol{V}^T$$





"2D"-Trained Network Extrapolates in "3D"



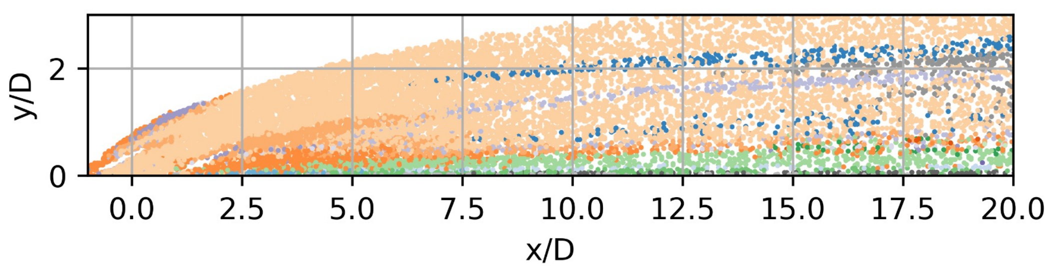
Conclusions

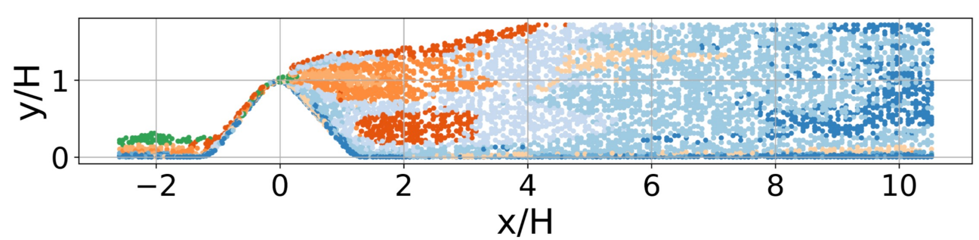
- GMM efficiently characterizes interpolation and extrapolation points
- Predictions were improved on average for interpolation cases

  Caution: High error for interpolating points is possible
- Extrapolation and error were significant when moving from quasi-2D flow to 3D flow

  3D: 5 independent invariants → 2D: 1 indp. inv. → 1D: 1 indp. inv.
- Big picture: an irony of data limitation

40 million CVs
≠
40 millions useful training points





Acknowledgments

- Collaborators
  - Jad Nasrallah (Mainspring Energy)
  - Pedro Milani (Google X)
  - David Ching (Sandia National Labs)

- Funding
  - National Science Foundation
  - Office of Naval Research