Physics-Informed Machine Learning for Predictive Turbulence Modeling:

Status, Perspectives, and Case Studies

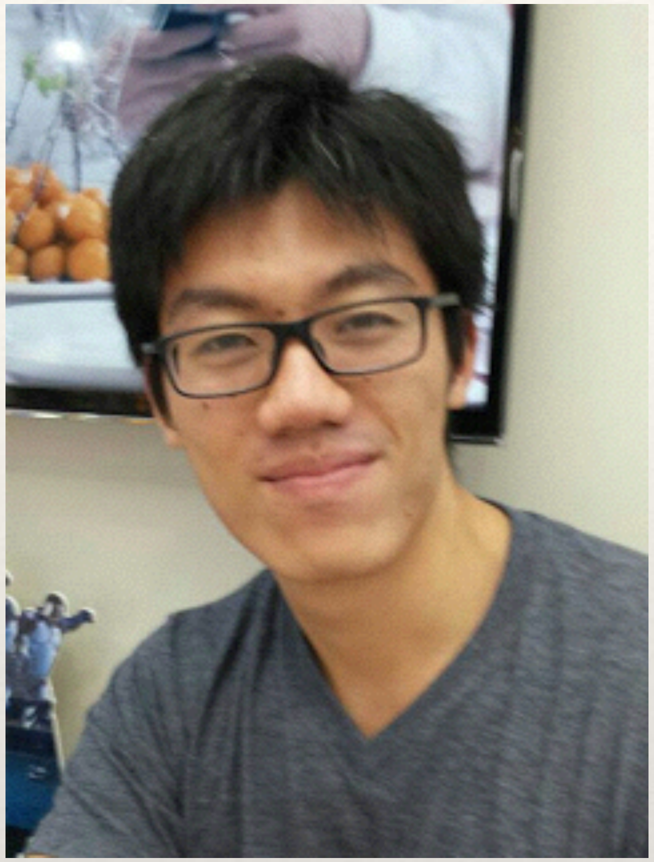
Heng Xiao

Assistant Professor
Department of Aerospace & Ocean Engineering
Virginia Tech

Machine Learning Technologies and Their Applications to Scientific and Engineering Domains Workshop, August 17, 2016

Acknowledgment of Collaborators

Jianxun Wang, VT



Jinlong Wu, VT

Dr. Julia Ling, SNL

VirginiaTech Invent the Future

AEROSPACE & OCEAN ENGINEERING AT VIRGINIA TECH

Publications Related to This Talk

The presentation slides will be made available publicly via ResearchGate, or sent me an email: hengxiao@vt.edu

https://sites.google.com/a/vt.edu/hengxiao/home
Or google: Heng Xiao, VT

Turbulence Modeling with Machine Learning (*offline data*)

J.-L. Wu, J.-X. Wang, H. Xiao, J. Ling. Physics-informed machine learning for predictive turbulence modeling: A priori assessment of prediction confidence. 2016.

J.-X. Wang, J.-L. Wu, and H. Xiao. Physics-informed machine learning for predictive turbulence modeling: Using data to improve RANS modeled Reynolds stresses. Submitted. Also available at: arxiv: 1606.07987

H. Xiao, J.-L. Wu, J.-X. Wang, E. G. Paterson. Are discrepancies in RANS modeled Reynolds stresses random? Submitted. Also available at: arxiv: 1606.08131

Other Related Publications

Turbulence Modeling with Data Assimilation (*online data*)

H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, and C. J. Roy. Quantifying and reducing model-form uncertainties in Reynolds averaged Navier-Stokes equations: An data-driven, physics-based Bayesian approach. Journal of Computational Physics, 115-136, 2016. Also available at arxiv:1508.06315

J.-X. Wang, J.-L. Wu, and H. Xiao. Incorporating prior knowledge for quantifying and reducing model-form uncertainty in RANS simulations. Accepted IJUQ, 2016. Accepted. Also available at arxiv:1512.01750

J.-L. Wu, J.-X. Wang, and H. Xiao. A Bayesian calibration-prediction method for reducing model-form uncertainties with application in RANS simulations. Flow, Turbulence and Combustion, 2016. DOI: 10.1007/s10494-016-9725-6 Also available at arxiv: 1510.06040

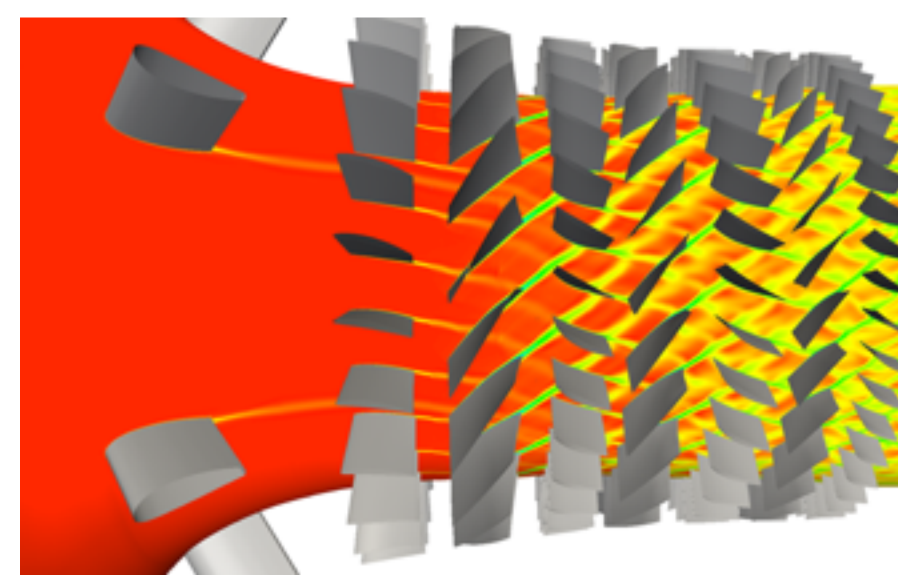
J.-L. Wu, J.-X. Wang, and H. Xiao. Quantifying Model Form Uncertainty in RANS Simulation of Wing–Body Junction Flow. Submitted to FTC.

Predictive Modeling & Model Discrepancy

- Reynolds Averaged Navier-Stoke (RANS) simulations are widely used in design, optimization, and reliability assessment of aero and space vehicles and gas turbines relevant to NASA missions.

- However, it remains challenging to **predict** system performance with confidence.

- **Model discrepancy is a major obstacle in predictive modeling with RANS models.**



Origin of Model Discrepancy in Low Fidelity Models

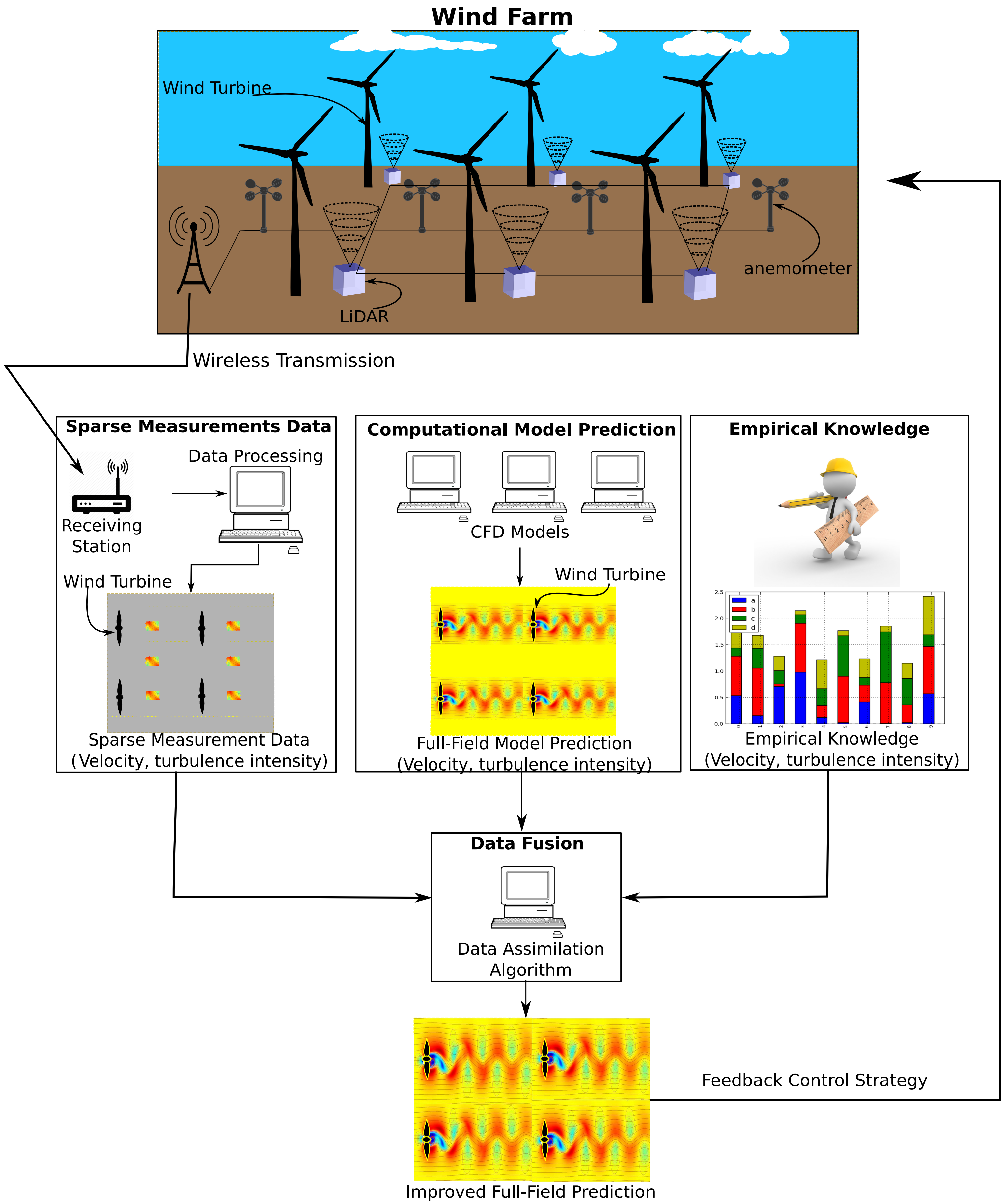
1. We do not **understand** the physics well enough to describe/model them.

2. We cannot **afford** the computational cost to adequately resolve the physics.

- In many cases, model discrepancy originates from the combination of two.

- The second reason is dominant for RANS based turbulence modeling, but it also depends on the interpretation.

Using data to complement low fidelity models!



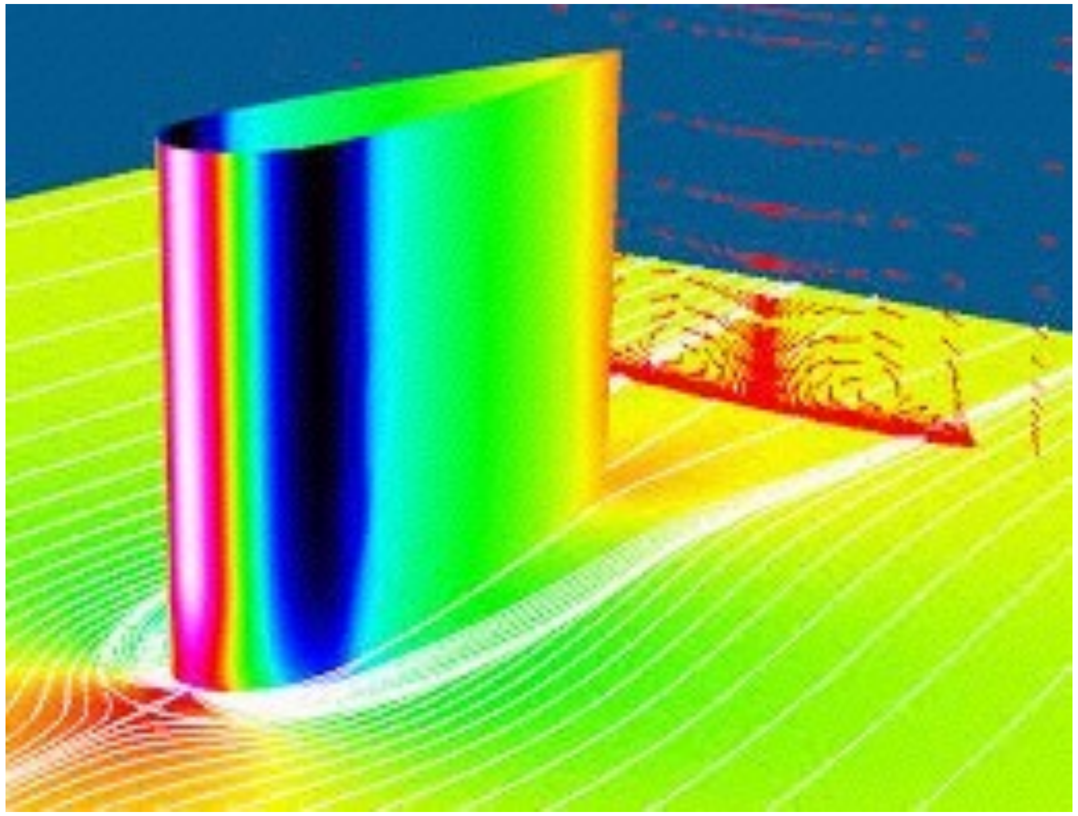
Simulation in Support System Monitoring

Online data:

Real-time streamed sensor data are available but are sparse

Data assimilation

Simulation in Support of Design and Optimization

Calibration Cases (offline data)	Prediction Cases (no data)
A few configuration with data (DNS or measurements) 	Similar configuration with different: • Twist • Sweep angles • Airfoil shape Machine learning

Scope of This Presentation

- Proposition: *Machine learning in conjunction with (offline) data can be used to reduce model discrepancy of* ***low-fidelity models****, which are often used in engineering design.*
- Here, I share my perspectives and experiences of using physics-informed machine learning to assist modeling of complex physical systems.
- RANS turbulence modeling, a typical low fidelity CFD model for turbulent flows, is used as example.
- However, the approach is general enough to be relevant for researchers of many other domains, e.g., structures, materials, combustion (flow and chemistry).

Unique Challenges in ML for Computational Physics

- Why can't we take the usual approach and simply use ML to learn what we want to know? Pressure, drag, lift, velocity, failure probability etc. May violate physics laws!
- There are many "**hard constraints**" originating from physics laws, e.g., velocity field is divergence free for incompressible fluid; pressure and velocity fields must be consistent (related via PDE); elasticity tensors must be positive definite, etc.
- Popular applications of ML has mostly "**soft constraints**". Sentiment analysis in reviews: great, pleasant=5; terrible=1. Targeted advertisement: diapers go with infant toys. Scientific document analysis: abstract, introduction, methodology, results, conclusion.

Physics-Informed Machine Learning: Clarification

- Algorithm development has drawn inspirations from physics/biological systems, e.g., simulated annealing, particle swarm, genetic algorithm: **Not** what I mean.
- Our interpretation of PIML: using ML to solve physical problems (mechanics of fluids, solids, materials, combustion).

☑ **Incorporate physical constraints (e.g., conservation laws, realizability) in every aspect of ML:**

- ☑ formulation of the learning problem
- ☑ choice/normalization of features and responses
- ☑ choice/development of learning algorithm.

☐ Co-design in (1) formulation of physical problem for learning; (2) ML algorithm development; (3) hardware.

Physics-Informed Machine Learning: Perspectives

Assist but respect models: Machine learning should be used to correct/improve existing models, not to replace them. Thus, we learn the model discrepancy, not the model output directly. (*consensus*)

1. Choose quantities that have physical bounds/constraints/interpretation to learn (allow for anchoring to physics).

2. Learned quantities should be **universal** to some extent: same functional form in training and prediction flows! Note the limitation of universality though…

3. **Obey physical constraints** in the learning as much as possible (e.g., hard constraints such as positive semi-definiteness of Reynolds stress; conservations of mass).

Case Study: RANS-Based Turbulence Modeling

- Description of the challenge
- Problem formulation
- Procedure:

$$y = f(q) \text{ or } f : q \mapsto y$$

    - Choice of features and responses
    - Choice of machine learning algorithm
- Results
- Possible extension to other systems

RANS as Work-Horse Tool in CFD

- RANS (Reynolds Averaged Navier-Stokes) solvers with turbulence closures are still the **low-fidelity** work-horse tool in industrial CFD simulations.
- High-fidelity methods such as LES and DNS are still too expensive for practical flows.
- The drawback of RANS: poor performance in flows with *separation, mean pressure gradient, curvature, or swirling …*
- Need to quantify and reduce model discrepancy in RANS simulations.

Source of Uncertainty in RANS Models

- **Reynolds stress closure** is the source of model form uncertainty in RANS simulations.

$$\boxed{\frac{\partial U_i}{\partial t} + \underbrace{\frac{\partial (U_i U_j)}{\partial x_j}}_{\text{convection}} + \underbrace{\frac{1}{\rho}\frac{\partial p}{\partial x_i}}_{\text{pressure grad.}} - \nu \underbrace{\frac{\partial^2 U_i}{\partial x_j \partial x_j}}_{\text{diffusion}}} = \boxed{\text{Div. Reynolds Stresses } \nabla \cdot \boldsymbol{\tau}}$$

$\boldsymbol{\tau}$: hub of turbulence models

Physics-based prior knowledge

Source of Uncertainty in RANS Models

- **Reynolds stress closure** is the source of model form uncertainty in RANS simulations.

$$\frac{\partial U_i}{\partial t} + \underbrace{\frac{\partial (U_i U_j)}{\partial x_j}}_{\text{convection}} + \underbrace{\frac{1}{\rho}\frac{\partial p}{\partial x_i}}_{\text{pressure grad.}} \underbrace{- \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j}}_{\text{diffusion}} = \boxed{\text{Div. Reynolds Stresses } \nabla \cdot \boldsymbol{\tau}}$$

$\boldsymbol{\tau}$: hub of turbulence models

k-ω model

$$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U}) \longrightarrow \boxed{\nu_t}$$

Physics-based prior knowledge

Source of Uncertainty in RANS Models

- **Reynolds stress closure** is the source of model form uncertainty in RANS simulations.

$$\frac{\partial U_i}{\partial t} + \underbrace{\frac{\partial (U_i U_j)}{\partial x_j}}_{\text{convection}} + \underbrace{\frac{1}{\rho}\frac{\partial p}{\partial x_i}}_{\text{pressure grad.}} \underbrace{- \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j}}_{\text{diffusion}} = \boxed{\text{Div. Reynolds Stresses } \nabla \cdot \boldsymbol{\tau}}$$

$\boldsymbol{\tau}$: hub of turbulence models

k-ω model

$$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U})$$

SA model

$$\frac{D\tilde{\nu}_t}{Dt} = P(\tilde{\nu}_t, \mathbf{U}) - D(\tilde{\nu}_t, \mathbf{U}) + T(\tilde{\nu}_t, \mathbf{U})$$

......

$$\boxed{\nu_t}$$

Physics-based prior knowledge

Source of Uncertainty in RANS Models

- **Reynolds stress closure** is the source of model form uncertainty in RANS simulations.

$$\frac{\partial U_i}{\partial t} + \underbrace{\frac{\partial (U_i U_j)}{\partial x_j}}_{\text{convection}} + \underbrace{\frac{1}{\rho}\frac{\partial p}{\partial x_i}}_{\text{pressure grad.}} - \underbrace{\nu \frac{\partial^2 U_i}{\partial x_j \partial x_j}}_{\text{diffusion}} = \boxed{\text{Div. Reynolds Stresses } \nabla \cdot \boldsymbol{\tau}}$$

$\boldsymbol{\tau}$: hub of turbulence models

k-ω model

$$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U})$$

$\boxed{\nu_t}$

SA model

$$\frac{D\tilde{\nu}_t}{Dt} = P(\tilde{\nu}_t, \mathbf{U}) - D(\tilde{\nu}_t, \mathbf{U}) + T(\tilde{\nu}_t, \mathbf{U})$$

......

Reynolds Stress Transport Model (RSTM)

$$\frac{D\boldsymbol{\tau}}{Dt} = P(\boldsymbol{\tau}, \mathbf{U}) - D(\boldsymbol{\tau}, \mathbf{U}) + T(\boldsymbol{\tau}, \mathbf{U})$$

Physics-based prior knowledge

Injecting Uncertainty in RANS Modeling

$$\frac{\partial U_i}{\partial t} + \underbrace{\frac{\partial (U_i U_j)}{\partial x_j}}_{\text{convection}} + \underbrace{\frac{1}{\rho}\frac{\partial p}{\partial x_i}}_{\text{pressure grad.}} - \nu \underbrace{\frac{\partial^2 U_i}{\partial x_j \partial x_j}}_{\text{diffusion}} = \text{Div. Reynolds Stresses } \nabla \cdot \boldsymbol{\tau}$$

$$\boldsymbol{\tau} + \delta\boldsymbol{\tau}$$

k-ω model

$$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U}) + \boxed{\delta} \longrightarrow \boxed{\nu_t}$$

......

- Injecting uncertainties directly to the Reynolds stresses: **output** of the turbulence closure [Xiao et al.] Our approach.

- Injecting uncertainties to turbulence model transport equations: **form** of the turbulence closure [Duraisamy et al.]

Critical Questions in Physics-Informed Machine Learning

Objective: Reduce RANS model discrepancy by learning from data.

- Where does the training data come from?
- What are the quantities to learn (responses, targets, dependent variables)? Are they universal, at least to some extent?
- What are the features (predictors, independent variables)?
- What learning algorithm should be used?

Critical Questions in PIML

Objective: Reduce RANS model discrepancy by learning from data.

- Where does the training data come from?
- What are the quantities to learn (responses, targets, dependent variables)? Are they universal, at least to some extent?
- What are the features (predictors, independent variables)?
- What learning algorithm should be used?

Formulation: Inference (optional) + Machine Learning

Data Pressure, Skin friction, Velocity

Inverse Modeling

$$\mathcal{N}(U) = \nabla \cdot (\boldsymbol{\tau}_{\text{rans}} + \delta\boldsymbol{\tau})$$

Xiao et al.

or $$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U}) + \delta$$

Duraisamy et al.

↓ Information Spatial discrepancy

Machine Learning

↓ Knowledge functional discrepancy

Embedding

Prediction : Injection into solver

[Duraisamy et al., 2015. AIAA]

Formulation: Inference (optional) + Machine Learning

Data — Pressure, Skin friction, Velocity

Inverse Modeling

Xiao et al.

$$\mathcal{N}(U) = \nabla \cdot (\boldsymbol{\tau}_{\mathrm{rans}} + \delta\boldsymbol{\tau})$$

or $$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U}) + \delta$$

Duraisamy et al.

Information — Spatial discrepancy

$$\delta\boldsymbol{\tau}(x) \quad \text{or} \quad \delta(x)$$

Machine Learning

Knowledge — functional discrepancy

Embedding

Prediction : Injection into solver

[Duraisamy et al., 2015. AIAA]

Formulation: Inference (optional) + Machine Learning

Data — Pressure, Skin friction, Velocity

Xiao et al.

Inverse Modeling

$$\mathcal{N}(U) = \nabla \cdot (\boldsymbol{\tau}_{\mathrm{rans}} + \delta\boldsymbol{\tau})$$

or
$$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U}) + \delta$$

Duraisamy et al.

Information — Spatial discrepancy

$$\delta\boldsymbol{\tau}(x) \quad \text{or} \quad \delta(x)$$

Machine Learning

Knowledge — functional discrepancy

$$\hat{\delta\boldsymbol{\tau}}(q) \quad \text{or} \quad \hat{\delta}(q)$$

Embedding

Prediction : Injection into solver

[Duraisamy et al., 2015.AIAA]

Formulation: Inference (optional) + Machine Learning

Data Pressure, Skin friction, Velocity

Inverse Modeling

Xiao et al.

$$\mathcal{N}(U) = \nabla \cdot (\boldsymbol{\tau}_{\text{rans}} + \delta\boldsymbol{\tau})$$

or $$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U}) + \delta$$

Duraisamy et al.

Information Spatial discrepancy

$\delta\boldsymbol{\tau}(x)$ or $\delta(x)$

Machine Learning

Knowledge functional discrepancy

$\delta\hat{\boldsymbol{\tau}}(q)$ or $\hat{\delta}(q)$

Embedding

$$\mathcal{N}(U) = \nabla \cdot (\boldsymbol{\tau}_{\text{rans}} + \delta\hat{\boldsymbol{\tau}}(q))$$

or $$\frac{D\omega}{Dt} = P - D + T + \hat{\delta}(q)$$

Prediction : Injection into solver

[Duraisamy et al., 2015. AIAA]

Formulation: Inference (optional) + Machine Learning

Data Pressure, Skin friction, Velocity

DNS Data of Reynolds Stress for Elementary Flows

ncy

Xiao et al.

$$\mathcal{N}(U) = \nabla \cdot (\boldsymbol{\tau}_{\text{rans}} + \delta\boldsymbol{\tau})$$

$$\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U}) + \delta$$

Duraisamy et al.

$$\delta\boldsymbol{\tau}(x) \quad \text{or} \quad \delta(x)$$

↓

Machine Learning

Knowledge functional discrepancy

$$\hat{\delta\boldsymbol{\tau}}(q) \quad \text{or} \quad \hat{\delta}(q)$$

↓

Embedding

$$\mathcal{N}(U) = \nabla \cdot (\boldsymbol{\tau}_{\text{rans}} + \hat{\delta\boldsymbol{\tau}}(q))$$

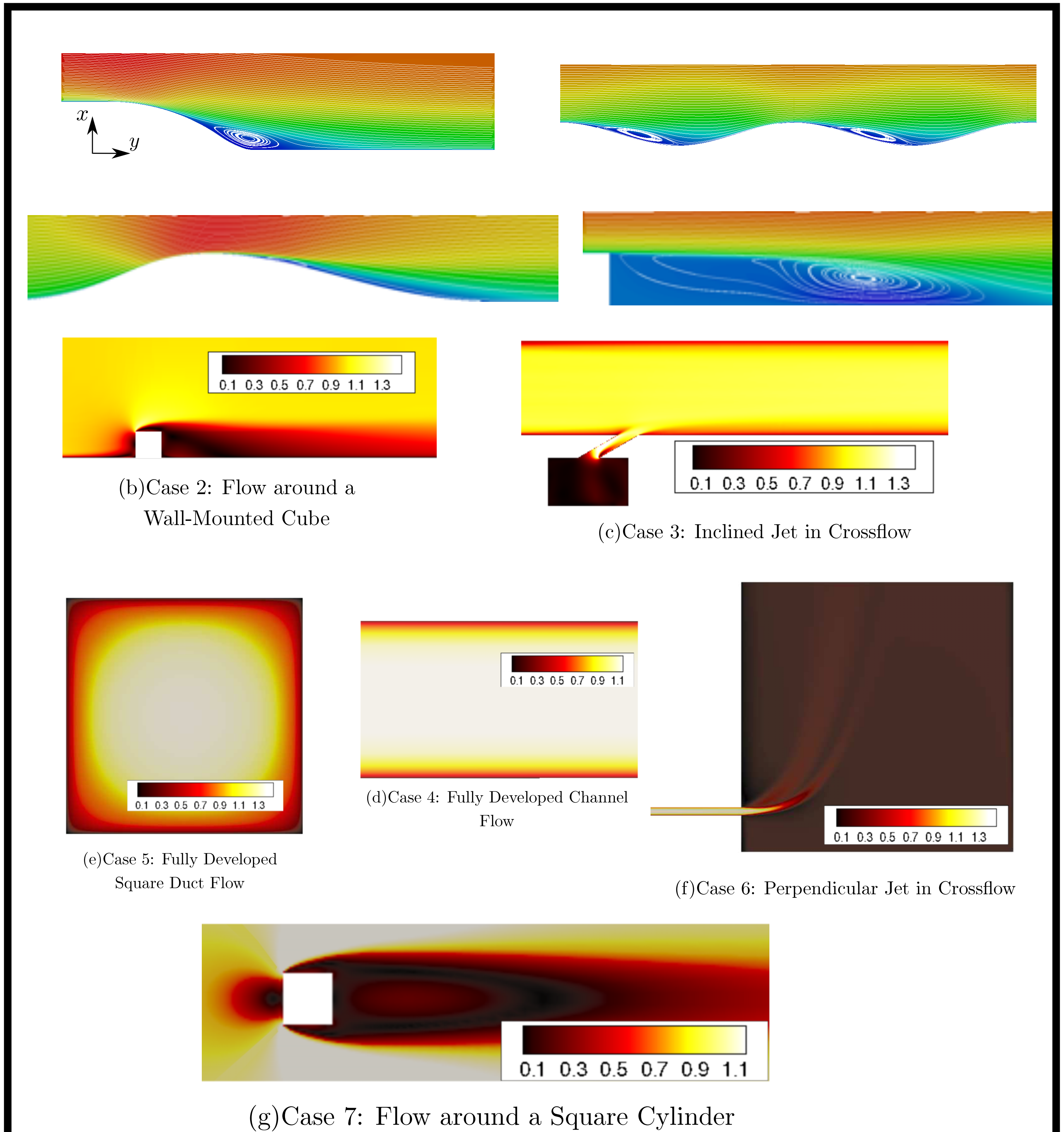
$$\text{or} \quad \frac{D\omega}{Dt} = P - D + T + \hat{\delta}(q)$$

Prediction : Injection into solver

[Duraisamy et al., 2015.AIAA]

Reynolds Stresses Obtained from DNS

Training: zoo of elementary flows



(b)Case 2: Flow around a Wall-Mounted Cube

(c)Case 3: Inclined Jet in Crossflow

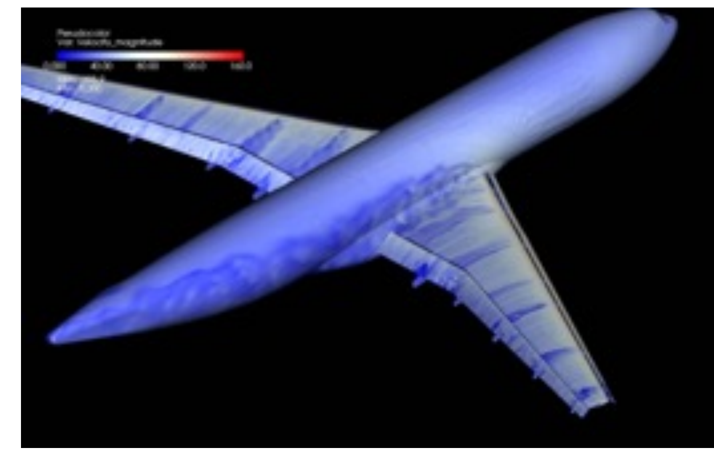
(d)Case 4: Fully Developed Channel Flow

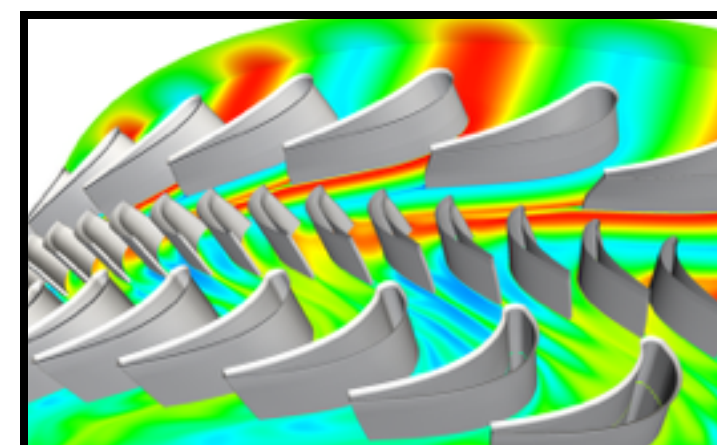
(e)Case 5: Fully Developed Square Duct Flow

(f)Case 6: Perpendicular Jet in Crossflow

(g)Case 7: Flow around a Square Cylinder

Prediction: Industrial flows





Some figures adopted from Ling et al. POF 2015; www.turbostream-cfd.com; youtube.com

Critical Questions in PIML

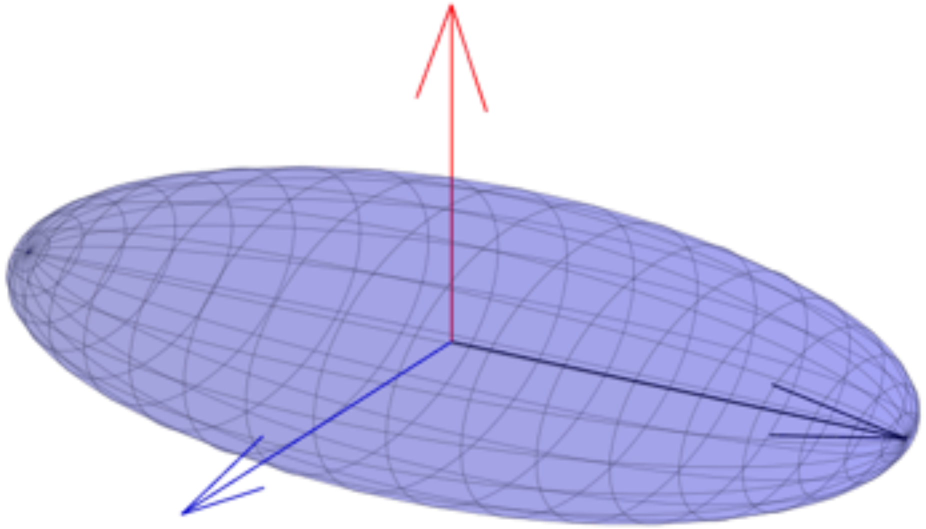
Objective: Reduce RANS model discrepancy by learning from data.

- Where does the training data come from?
- What are the quantities to learn (responses, targets, dependent variables)? Are they universal?
- What are the features (predictors, independent variables)?
- What learning algorithm should be used?

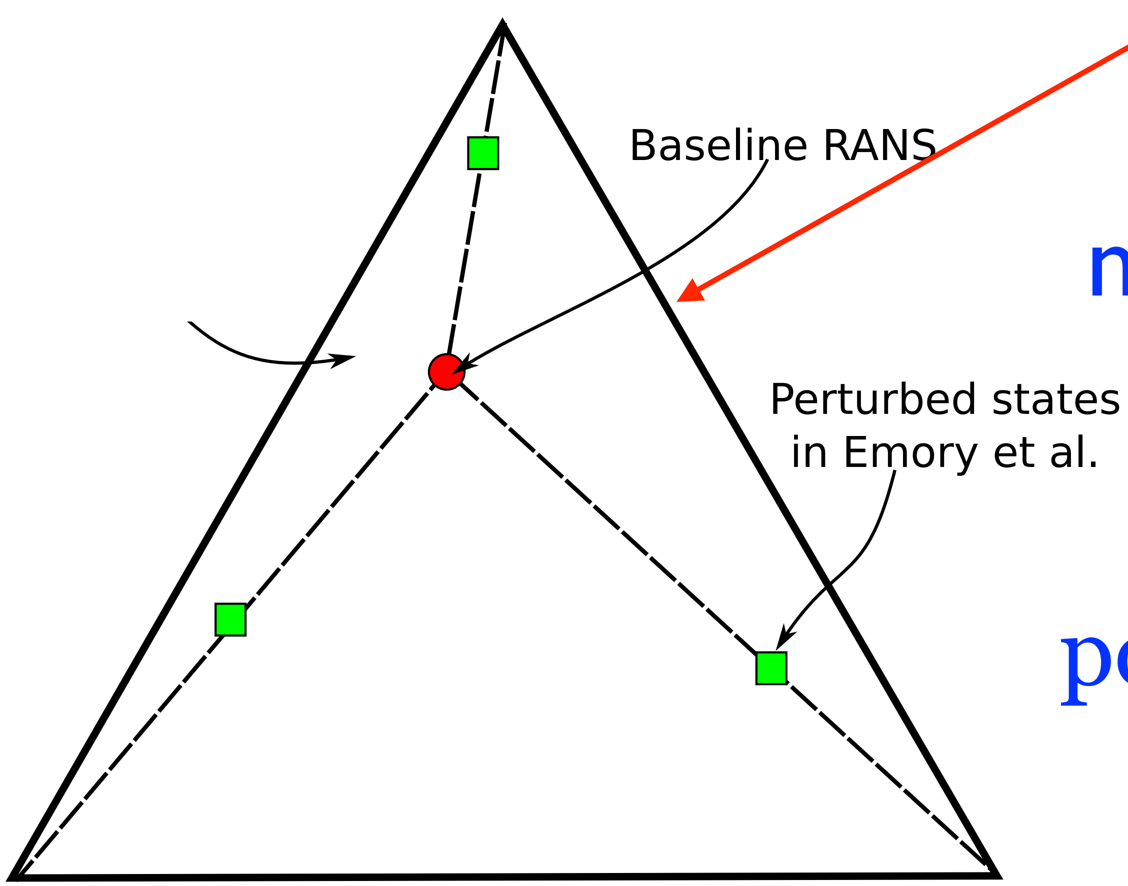
Injecting Uncertainty into Reynolds Stresses

- Iaccarino et al. perturbed towards three limiting states in Barycentric triangle (realizability map) for uncertainty estimation

[Iaccarino et al.]



$$\boldsymbol{\tau} = 2k\left(\frac{1}{3}\mathbf{I} + \mathbf{a}\right) = 2k\left(\frac{1}{3}\mathbf{I} + \mathbf{V}\Lambda\mathbf{V}^T\right)$$



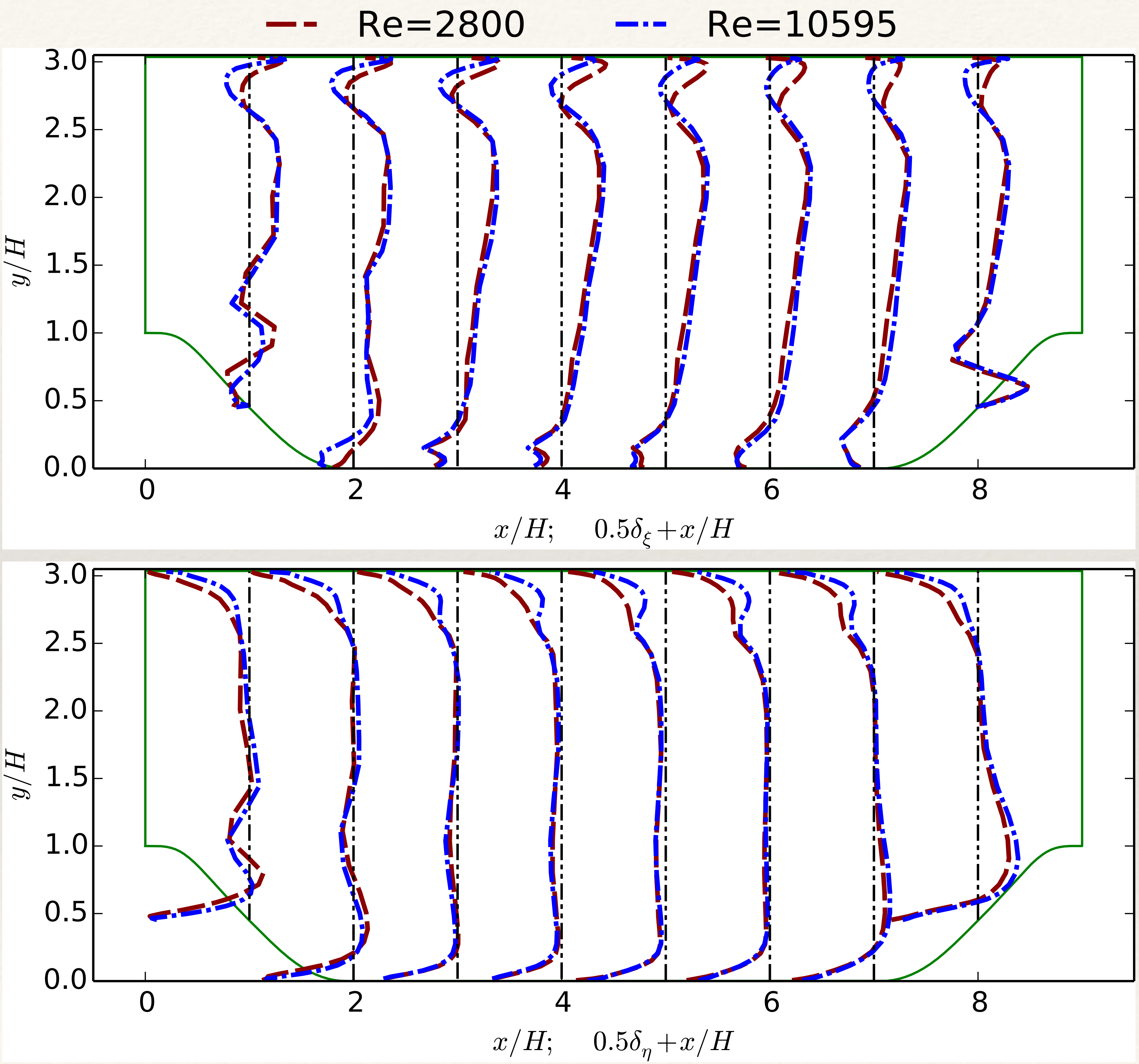
$$\boldsymbol{\tau} \longrightarrow (k, \xi, \eta, \varphi_1, \varphi_2, \varphi_3)$$

magnitude, aspect ratio, orientation

Physics-based "normalization":
potential to be universal quantities;

Physical constraints respected.

Is The Discrepancy of Anisotropy Universal?



Probably!

δ^ξ

δ^η

J.-L. Wu, J.-X. Wang, and H. Xiao. A Bayesian calibration-prediction method for reducing model-form uncertainties with application in RANS simulations. *Flow, Turbulence and Combustion*, 2016.

From Physical Space to Feature Space: Learning

- Construct discrepancy function based on "mean flow features" ***q***!

$$\delta \boldsymbol{\tau}(\mathbf{x})$$

Inferred or DNS, not universal (specific to the geometry)

$$\Rightarrow \quad \delta \boldsymbol{\tau}(\mathbf{q})$$

Machine learning

- Responses are **discrepancies** in TKE (log), eigenvalues and eigenvectors.

$$\boldsymbol{\tau} = 2k\left(\frac{1}{3}\mathbf{I} + \mathbf{a}\right) = 2k\left(\frac{1}{3}\mathbf{I} + \mathbf{V}\Lambda\mathbf{V}^T\right)$$

$$\boxed{\delta \boldsymbol{\tau}_i(\mathbf{q})}$$

$$\delta \log(k)(\mathbf{q}), \delta\xi(\mathbf{q}), \delta\eta(\mathbf{q}), \delta\varphi_1(\mathbf{q})$$

Critical Questions in PIML

Objective: Reduce RANS model discrepancy by learning from data.

- Where does the training data come from?
- What are the quantities to learn (responses, targets, dependent variables)? Are they universal, at least to some extent?
- What are the features (predictors, independent variables)?
- What learning algorithm should be used?

Construction of Feature Space

$$\{\boxed{S, \Omega, \nabla p, \nabla k,} Re_d, \mathcal{P}/\varepsilon, k/\varepsilon, \kappa\}$$

4 tensors/vectors; 47 invariants (integrity bases)

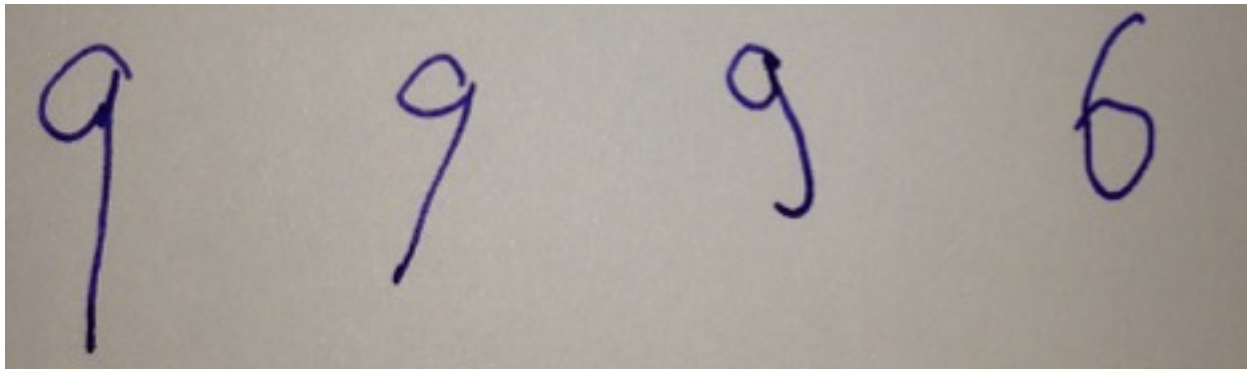
- Invariants of 4 tensors/vectors: strain rate (S), rotation rate (Ω), pressure (p) gradient, TKE (k) gradient;
- 4 scalars: streamline curvature (κ), wall-distance based Reynolds number (Re_d), turbulent time scale
- (Normalized) feature vector **q** has a length of ~50.
- Very high dimension feature space: beyond human comprehension: interpretation in progress.

Objective: train discrepancy functions $\delta\boldsymbol{\tau}_i(\mathbf{q})$

(Ling et al. 2016)

Should Feature Variables Be Invariant?

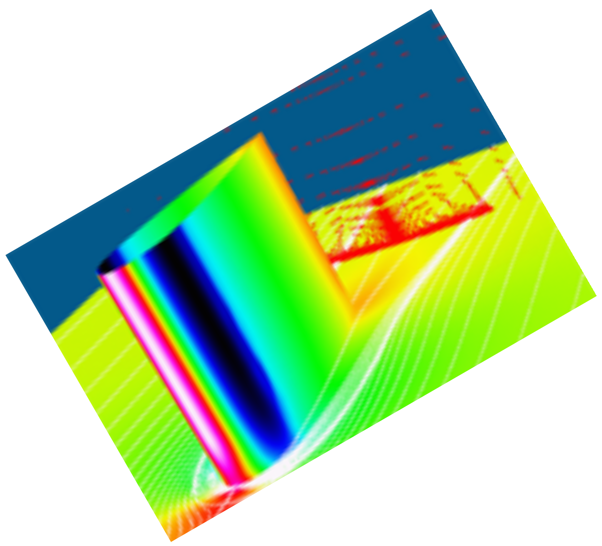
- Invariants is not only desirable, but essential!
- Very different from other applications of ML, e.g., handwritten digits recognition.

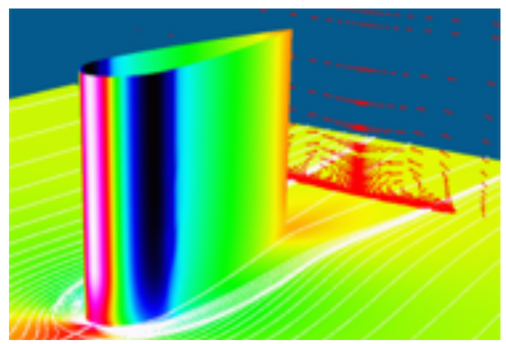


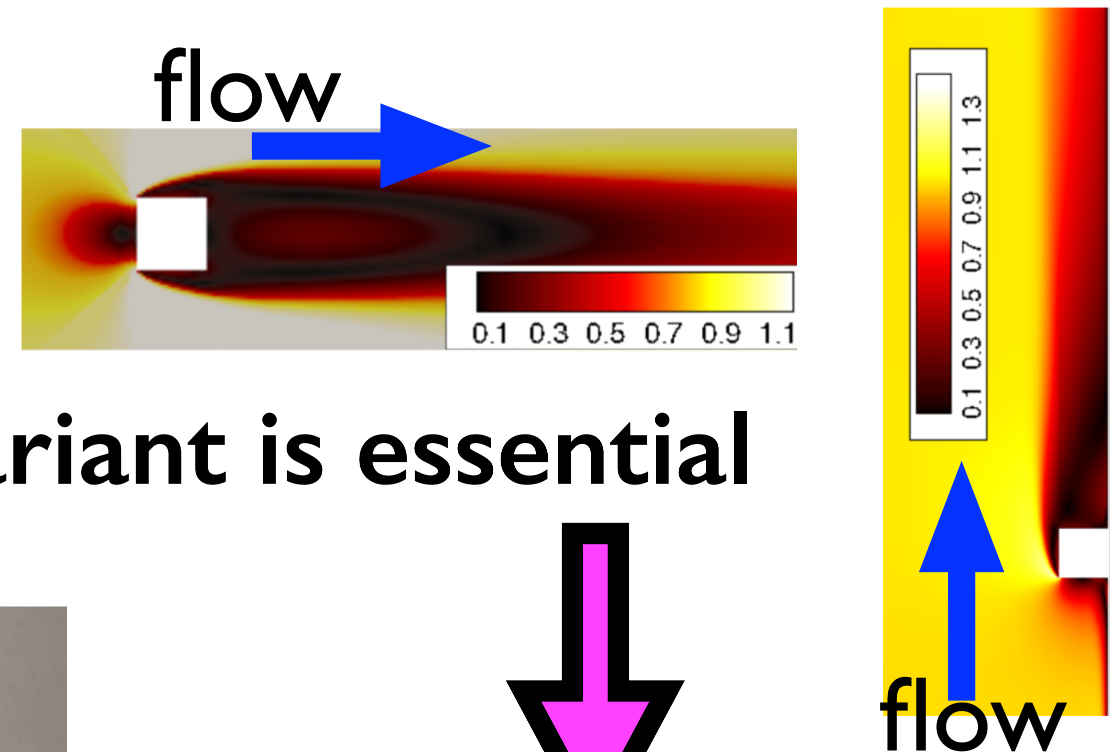
Features should **not** be fully invariant here!

Should Feature Variables Be Invariant?

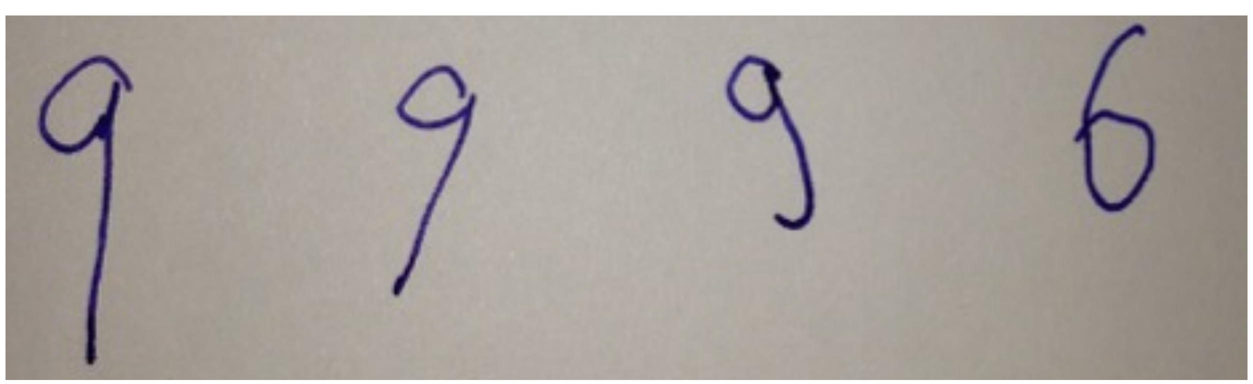
- Invariants is not only desirable, but essential!
- Very different from other applications of ML, e.g., handwritten digits recognition.



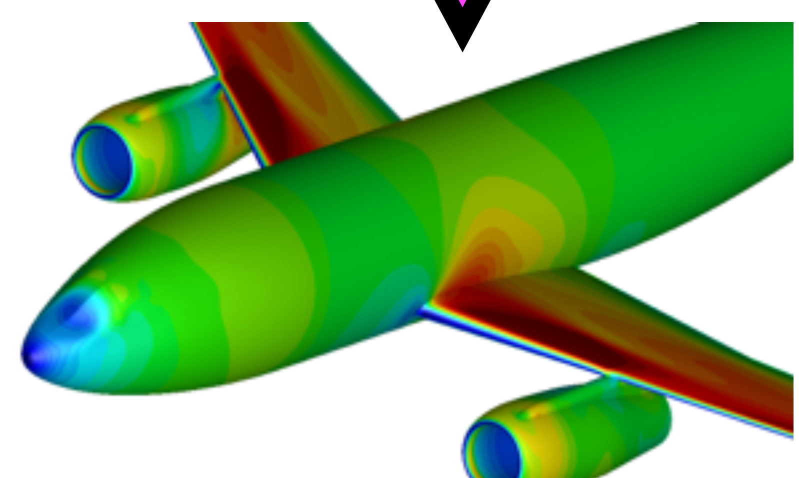




Fully invariant is essential



Features should **not** be fully invariant here!



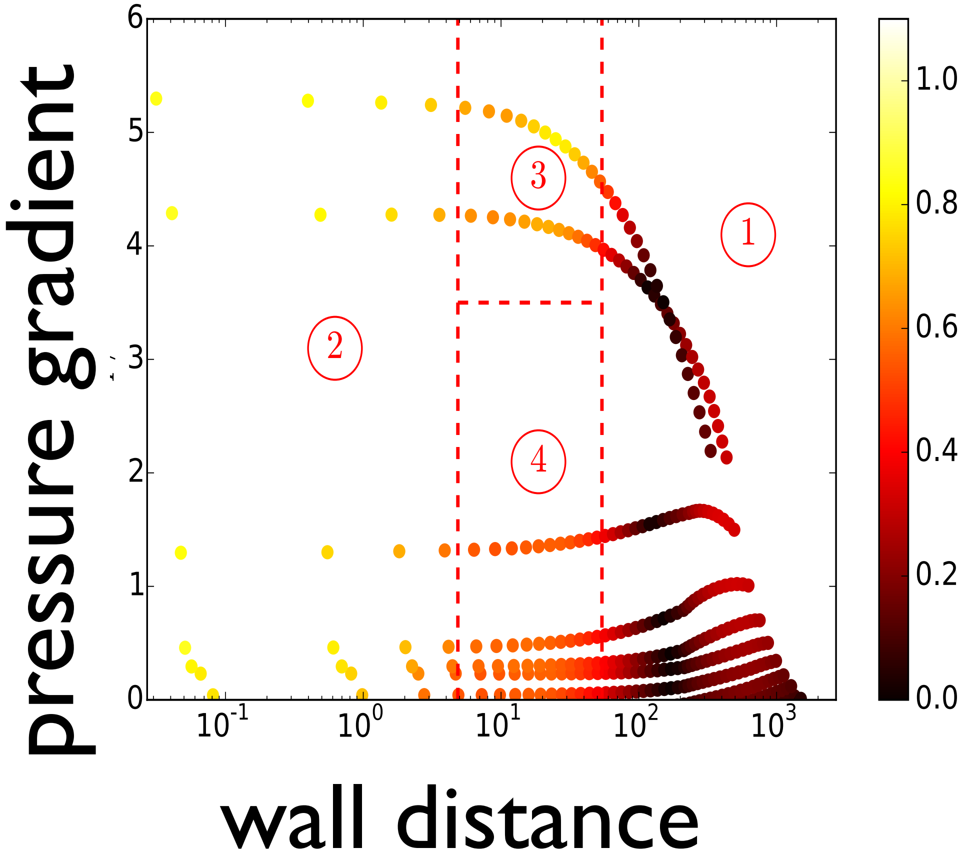
Critical Questions in PIML

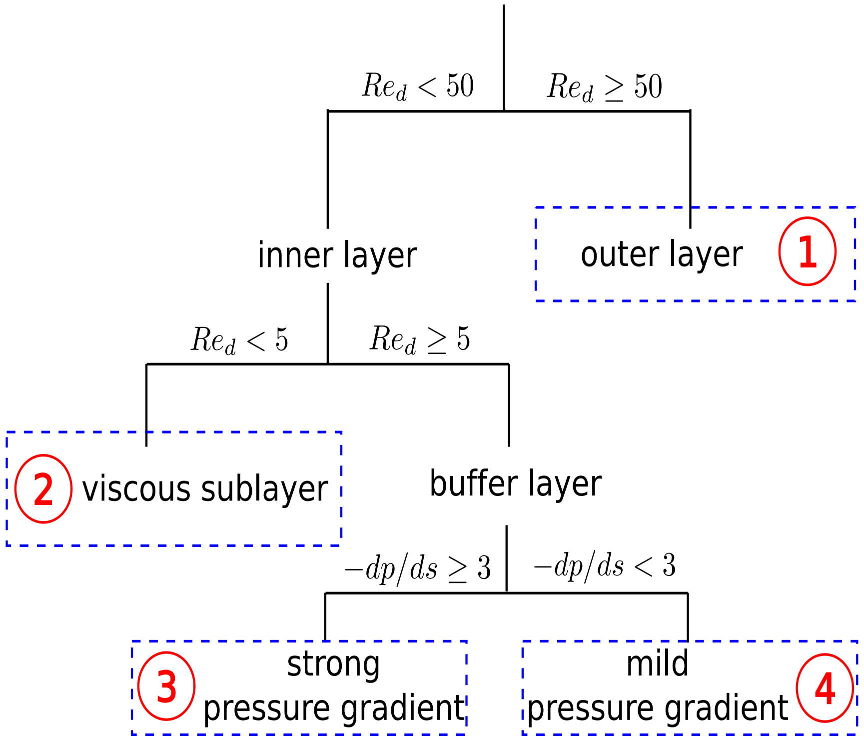
Objective: Reduce RANS model discrepancy by learning from data.

- Where does the training data come from?
- What are the quantities to learn (responses, targets, dependent variables)? Are they universal, at least to some extent?
- What are the features (predictors, independent variables)?
- What algorithm should be used?

Machine Learning Algorithm: Random Forests

- Machine learning is an umbrella term for many algorithms.
- We used **Random Forests regression**: (1) suitable for high-dimension feature space and (2) robust in tolerating unimportant features; no linear regression = more robust
- Key lesson: choice of algorithm is dictated by the physical problem.



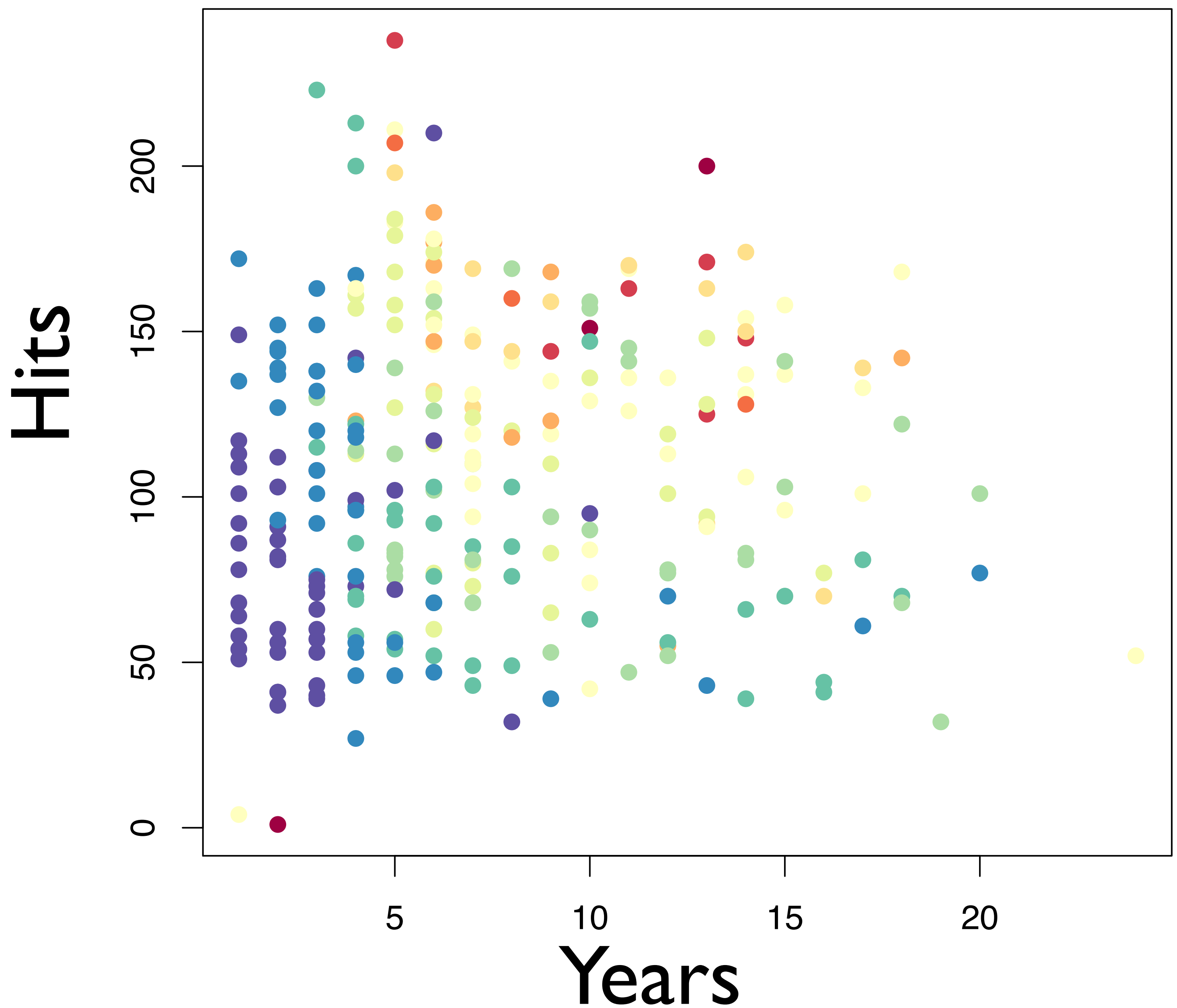


(b) regression tree

Physics-informed machine learning

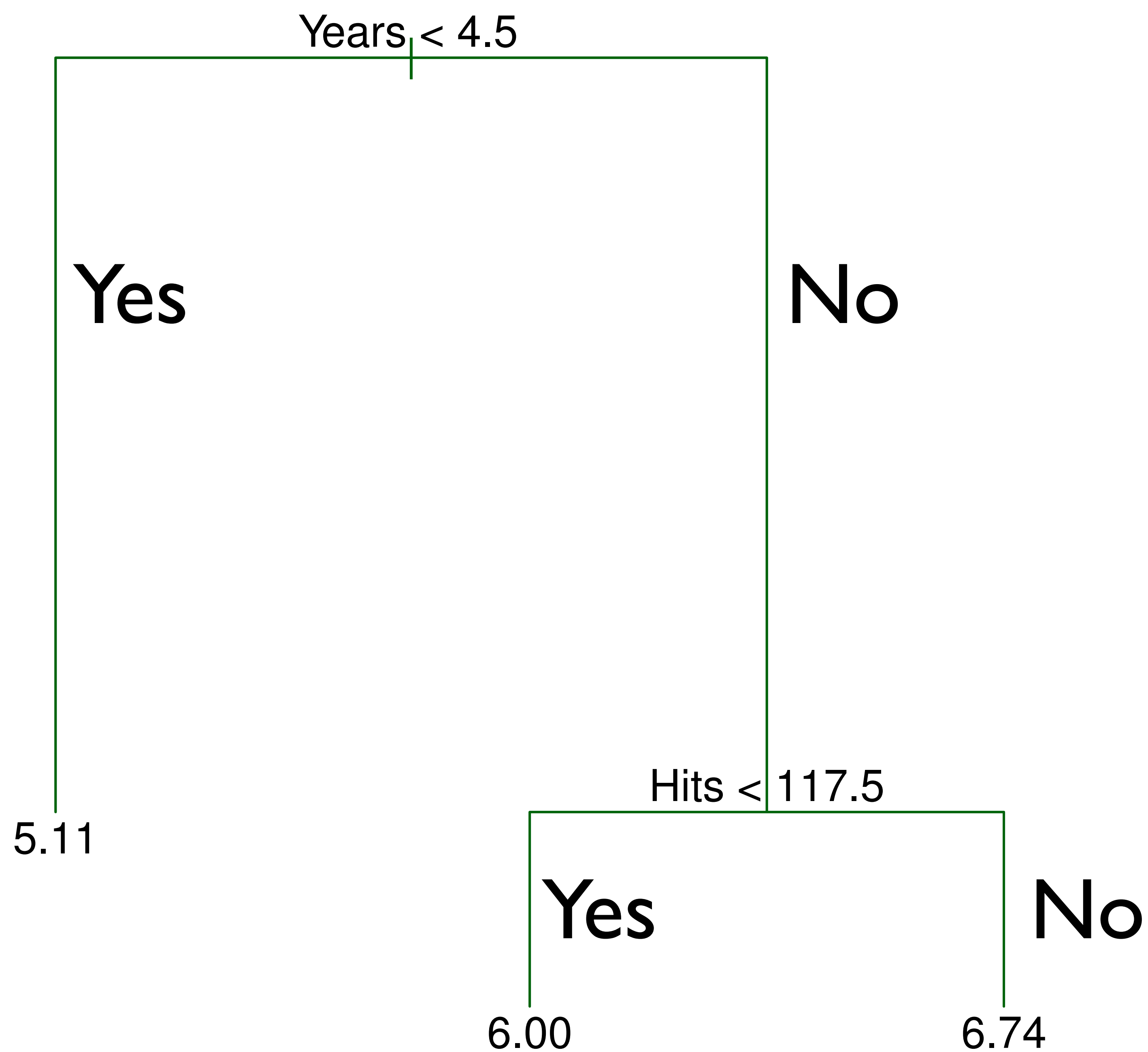
Decision Tree Example: Prediction of Salary

- Stratifying baseball player salary data (color coded from low (blue, green) to high (yellow, red)



(Figure credit: Hastie and Tibshirani, Introduction to Statistical Learning)

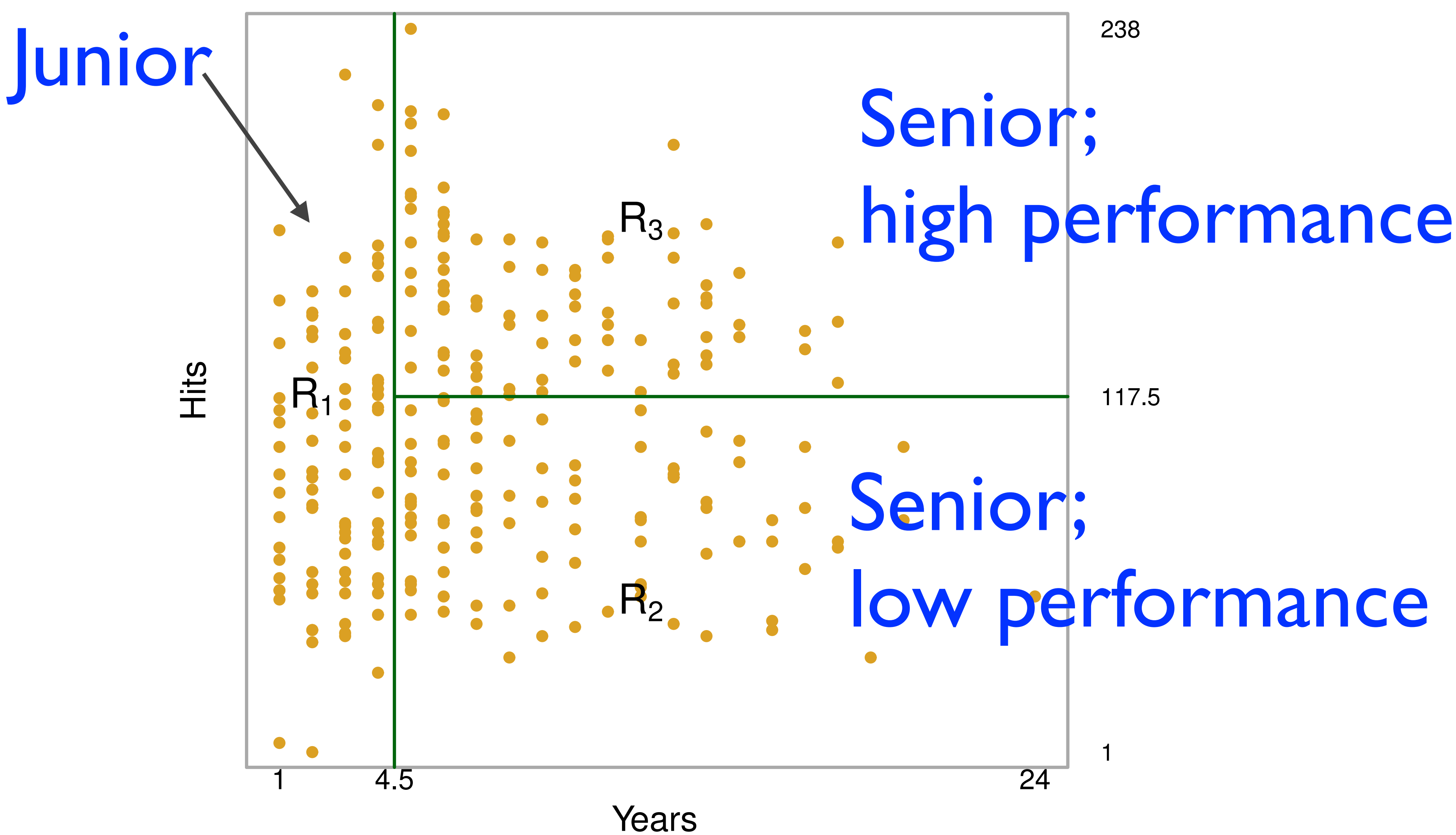
Decision tree for these data



(Figure credit: Hastie and Tibshirani, Introduction to Statistical Learning)

Results

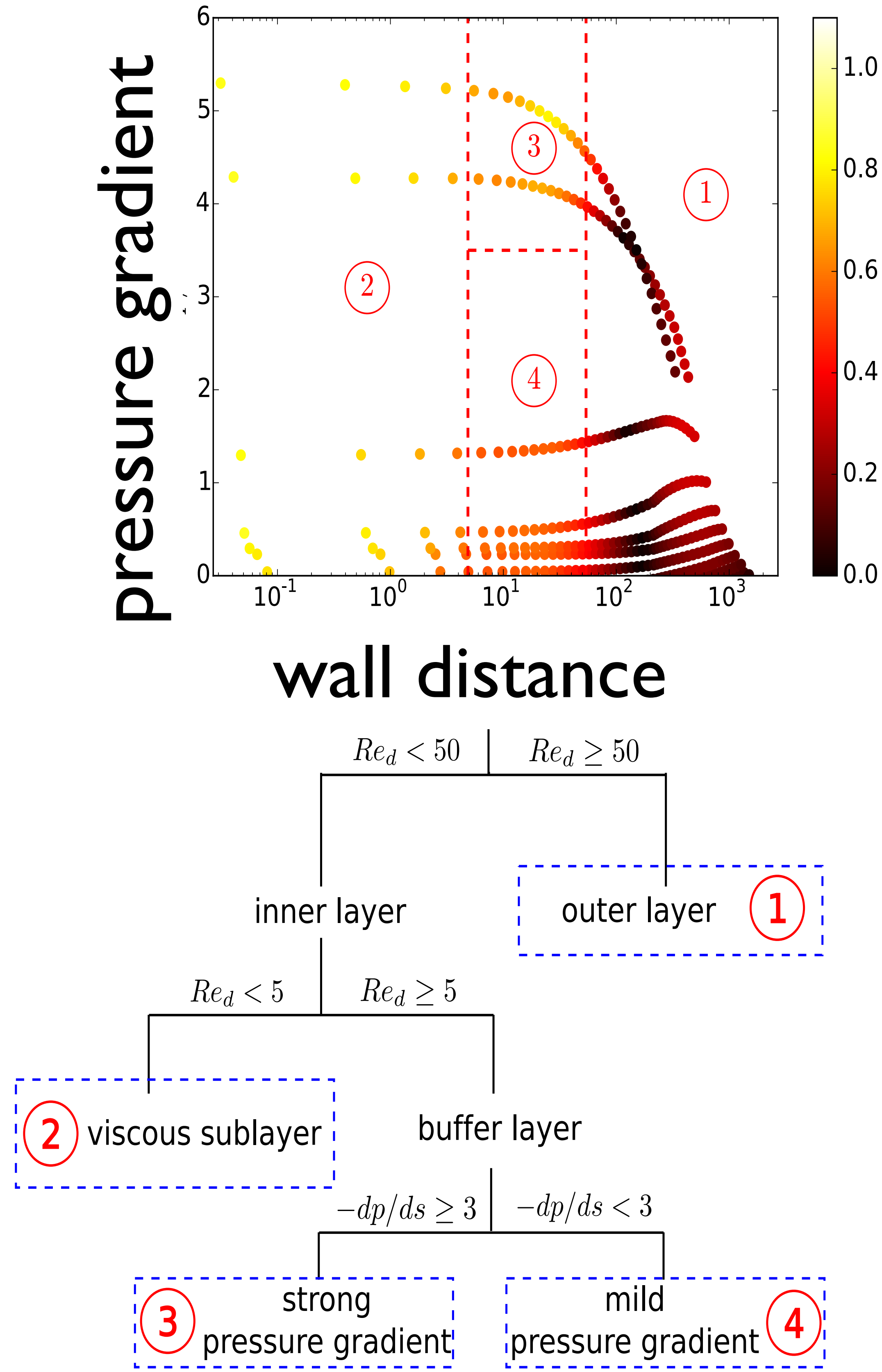
- Overall, the tree stratifies or segments the players into three regions of predictor space: R_1 ={X | `Years`< 4.5}, R_2 ={X | `Years`>=4.5, `Hits`<117.5}, and R_3 ={X | `Years`>=4.5, `Hits`>=117.5}.



(Slide credit: Hastie and Tibshirani, Introduction to Statistical Learning)

From Decision Tree to Random Forests

- Individual decision trees are usually bad decision makers: greedy algorithm may miss globally optimal stratification.
- Random forests: an ensemble of trees built from bootstrap samples.
- Use only a subset of features to de-correlate the trees.
- Physical intuition!



(b) regression tree

Other Machine Learning Algorithms

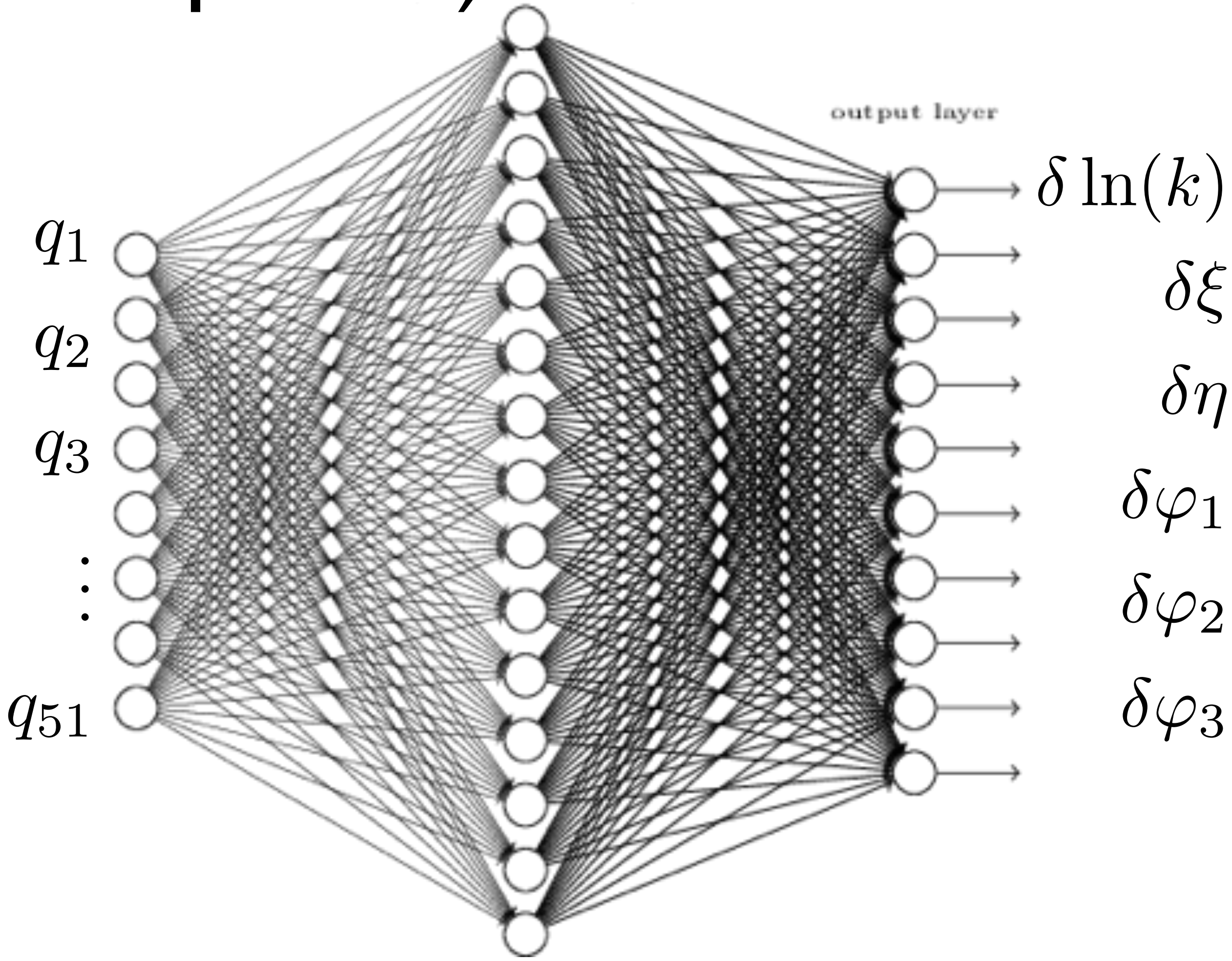
- With a feature space dimension of 50, many ML algorithms susceptible to "*curse of dimensionality*" are ruled out: e.g., linear regression and its variants; Gaussian Process.

- Neural network seems to be viable choice with several potential benefits (yet to be explored):

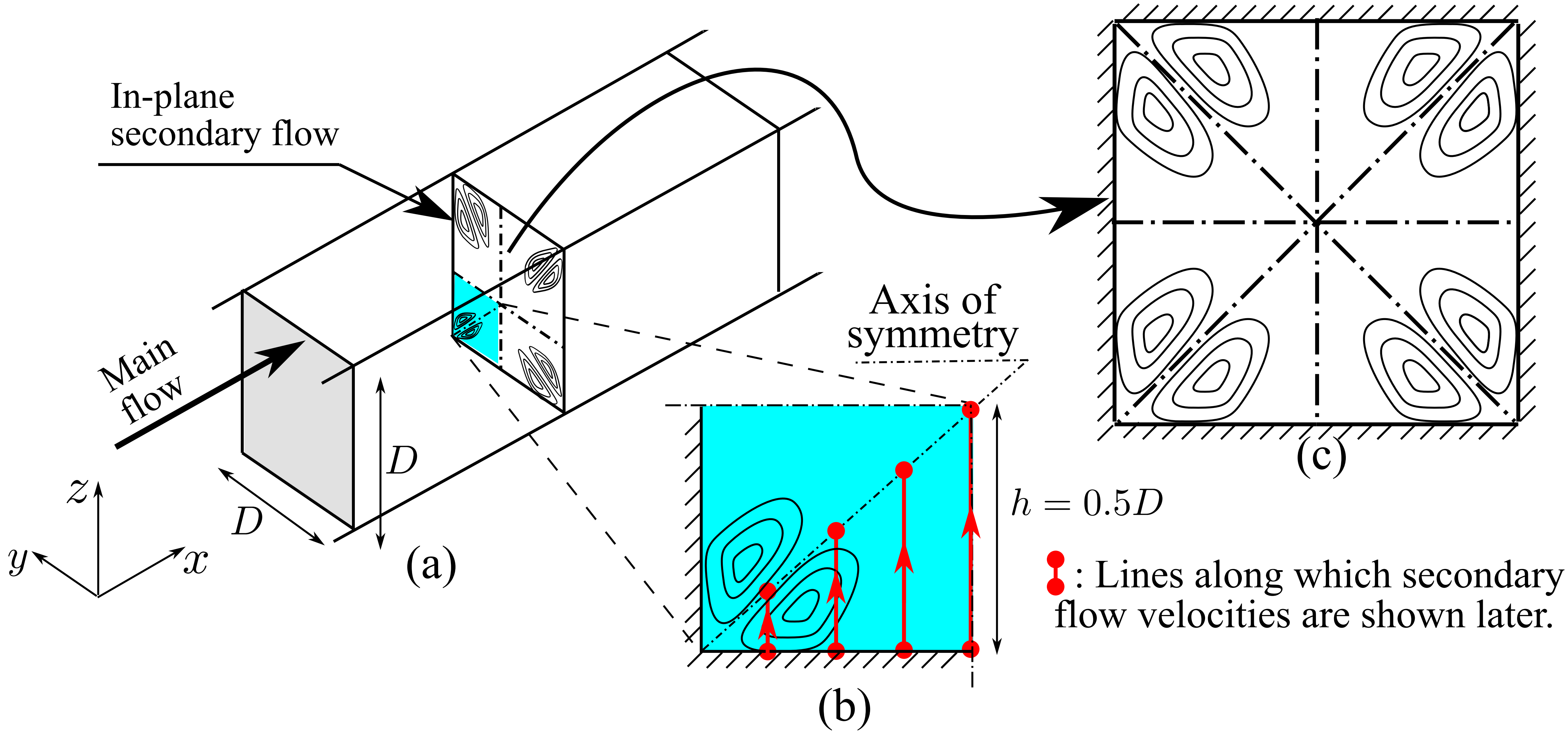
  ✓ More natural for coupled regression.

  ✓ More flexible and possibly better predictive skills.

  ✓ Co-design?

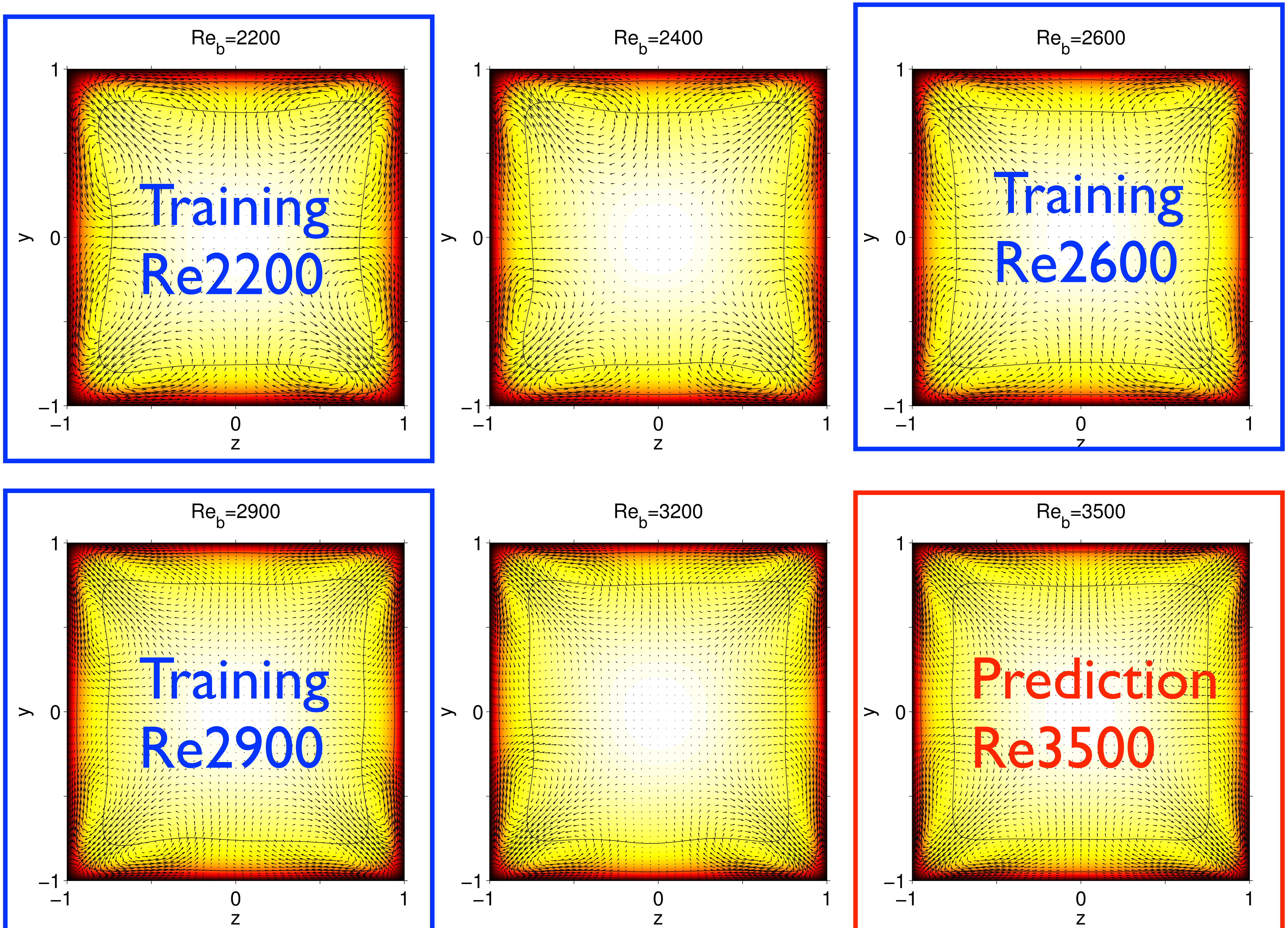


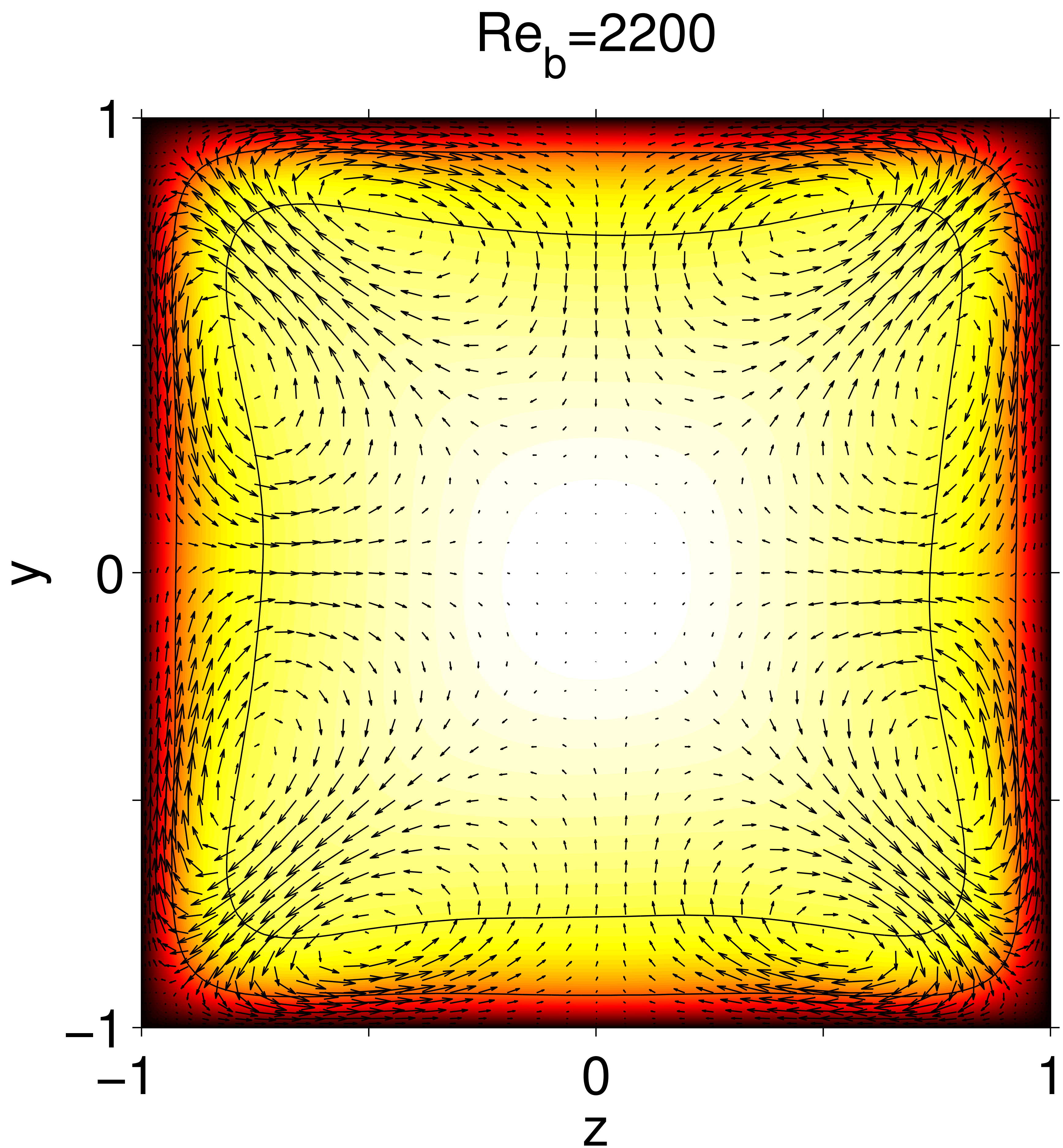
Case Study: Flow in a Square Duct



The flow features in-plane secondary flow vortexes, which cannot be predicted by standard RANS models.

DNS Data for Duct Flows

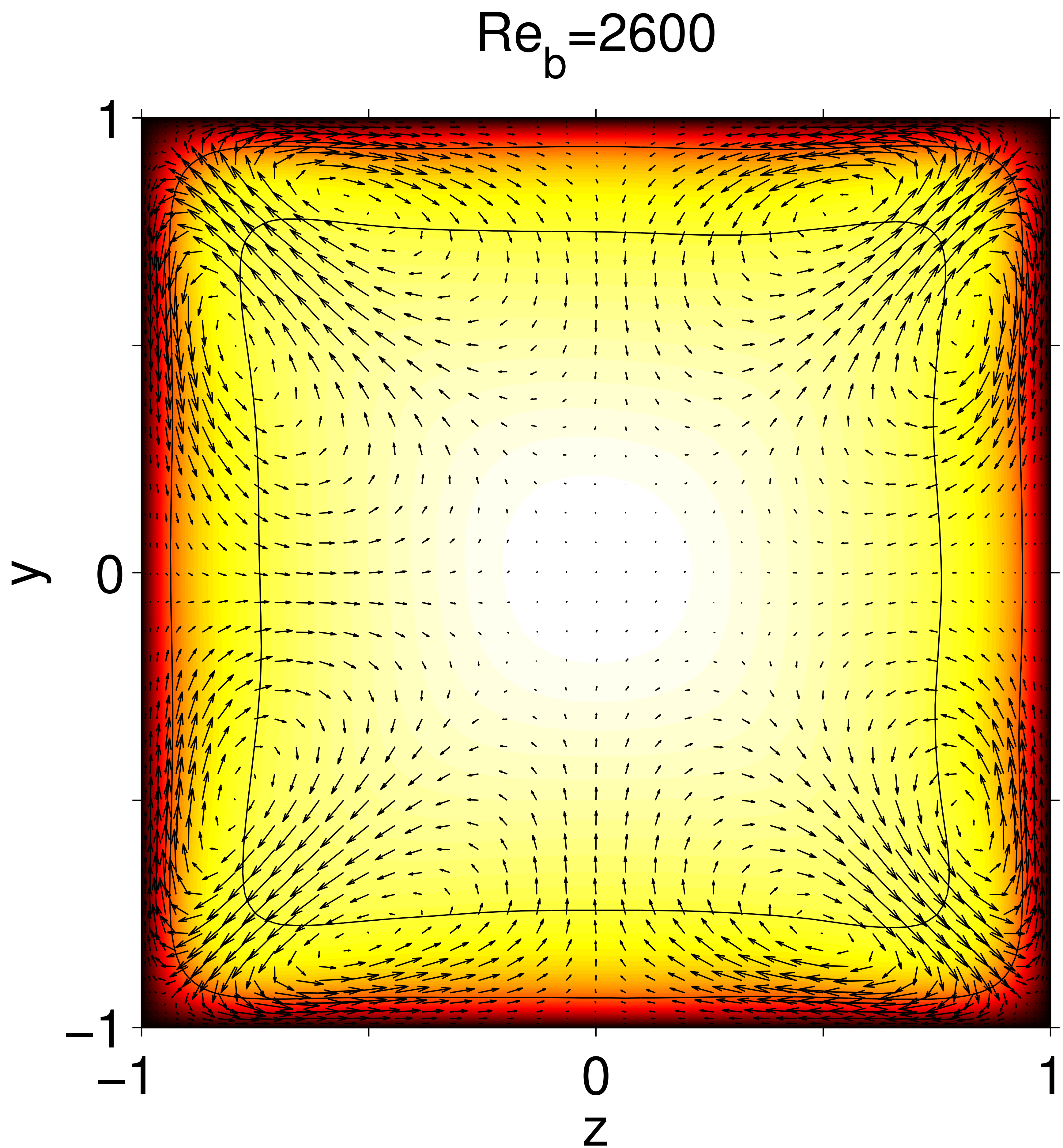




Arrows:
In-plane
velocity

Color:
streamwise
velocity

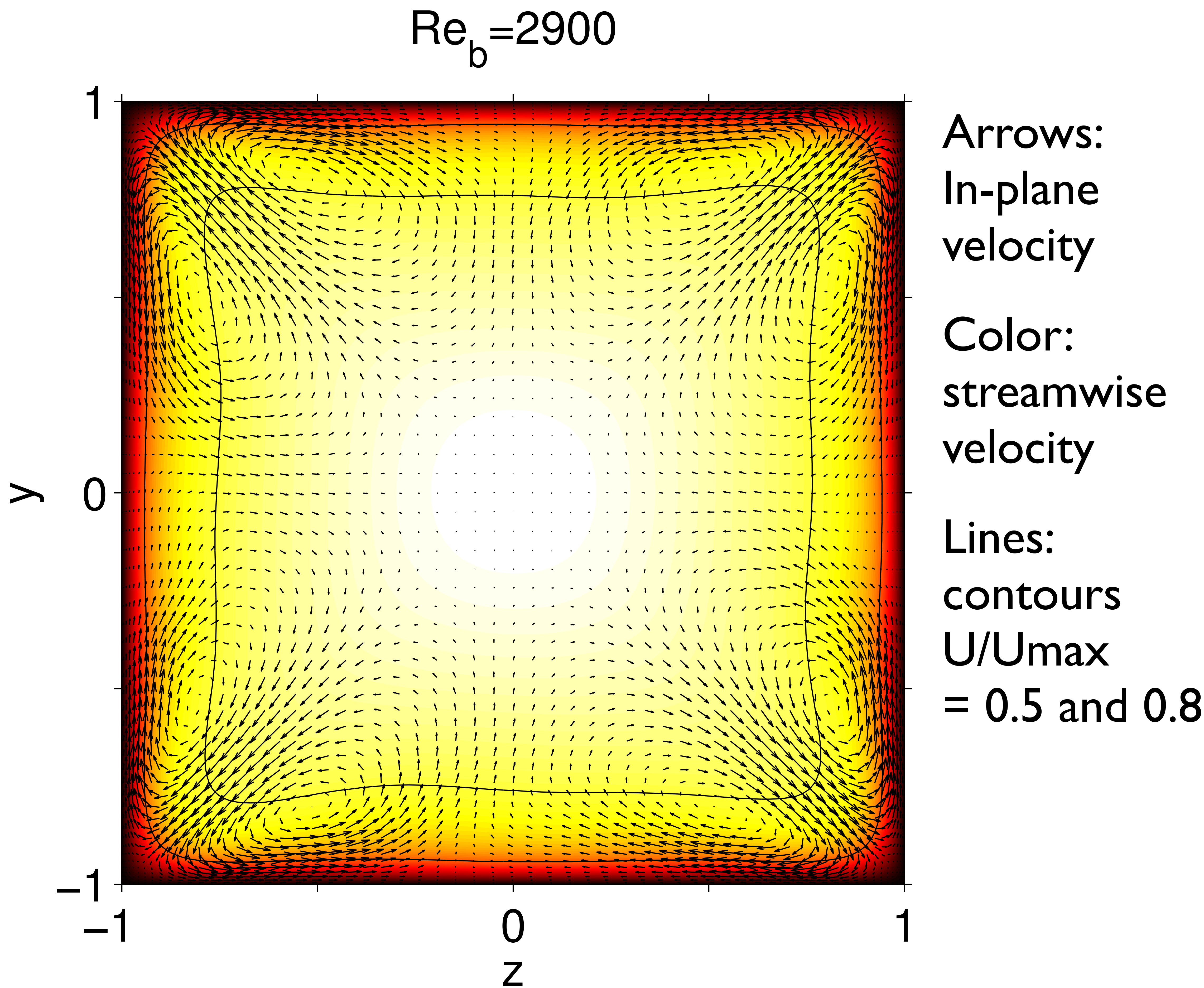
Lines:
contours
U/Umax
= 0.5 and 0.8



Arrows: In-plane velocity

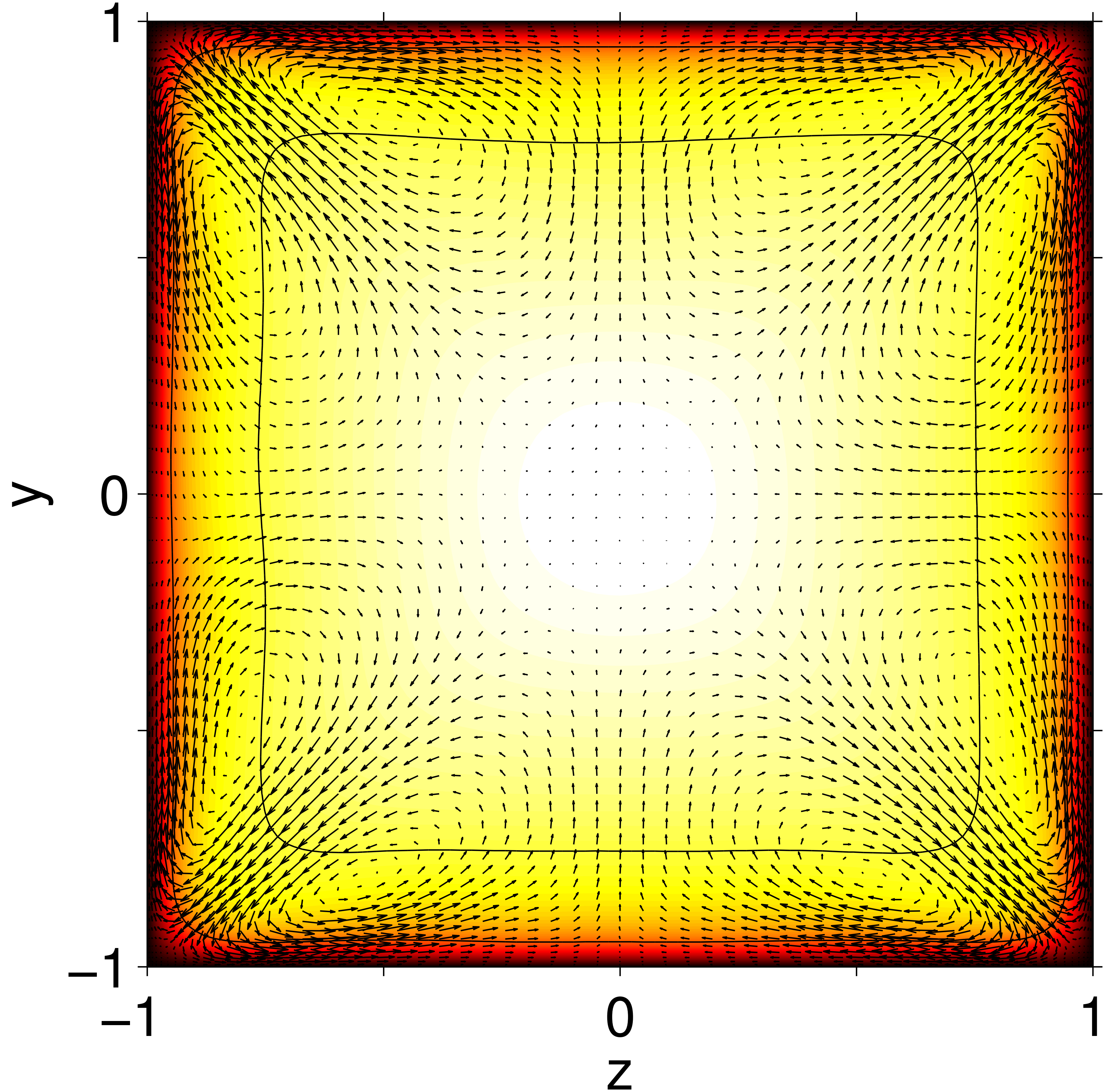
Color: streamwise velocity

Lines: contours U/Umax = 0.5 and 0.8



Re$_b$=2900
1
0
−1
y
−1
0
1
z
Arrows: In-plane velocity
Color: streamwise velocity
Lines: contours U/Umax = 0.5 and 0.8

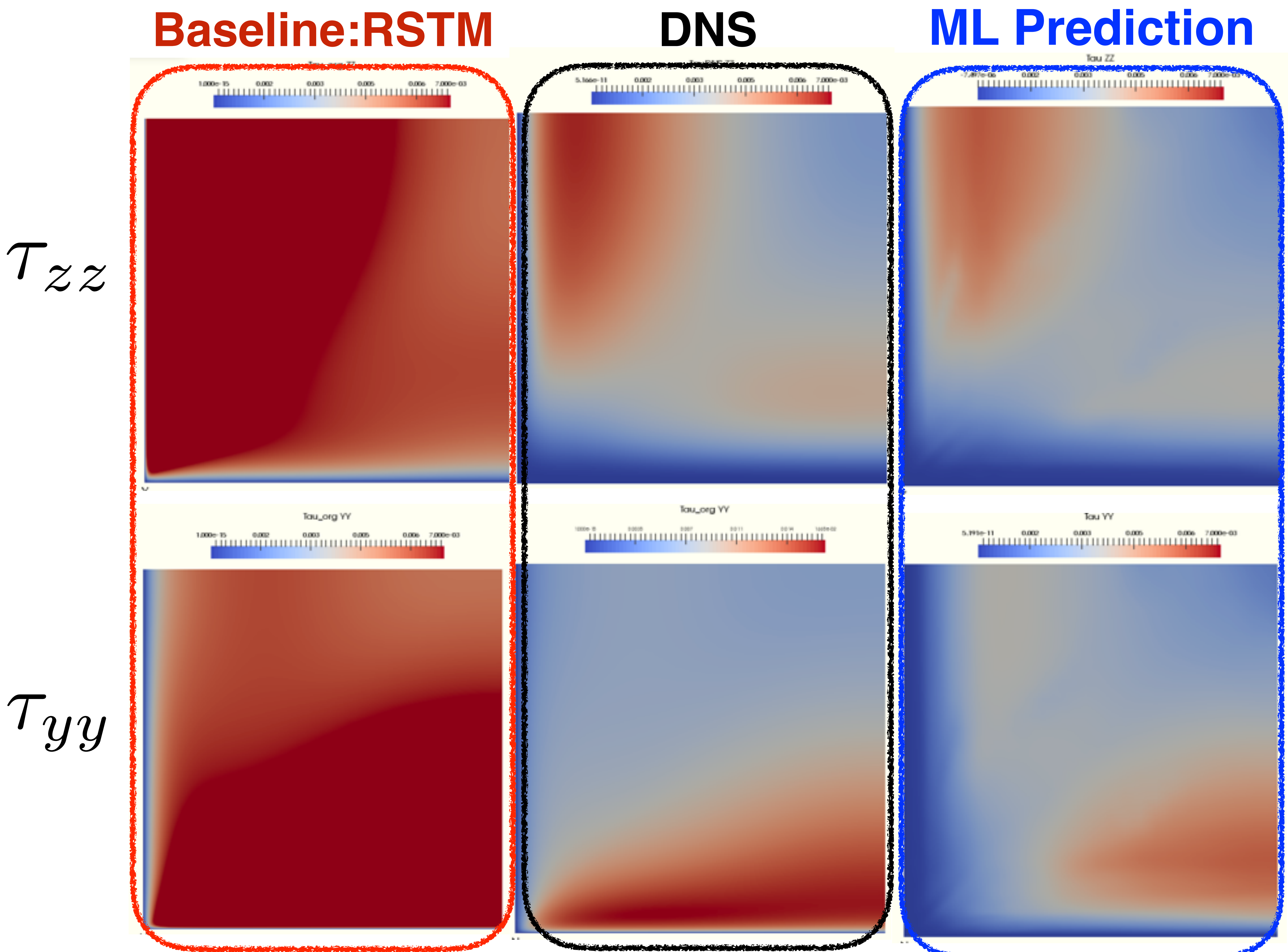
Re_b=3500



Arrows:
In-plane
velocity

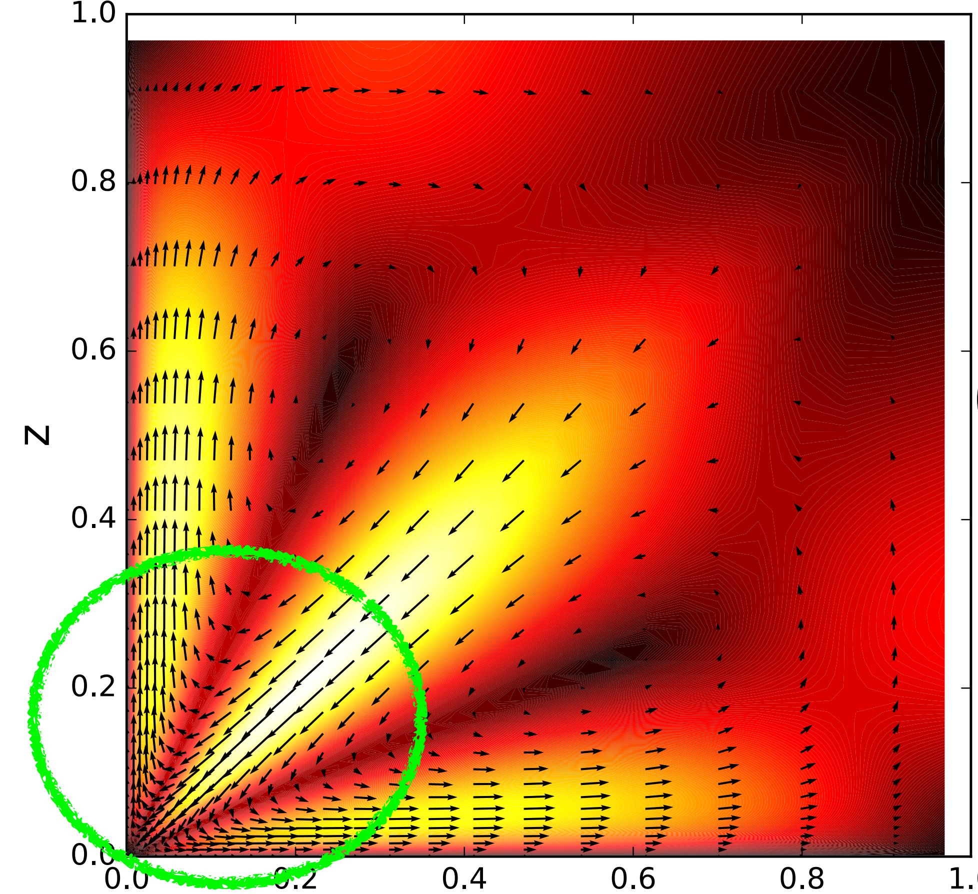
Color:
streamwise
velocity

Lines:
contours
U/Umax
= 0.5 and 0.8

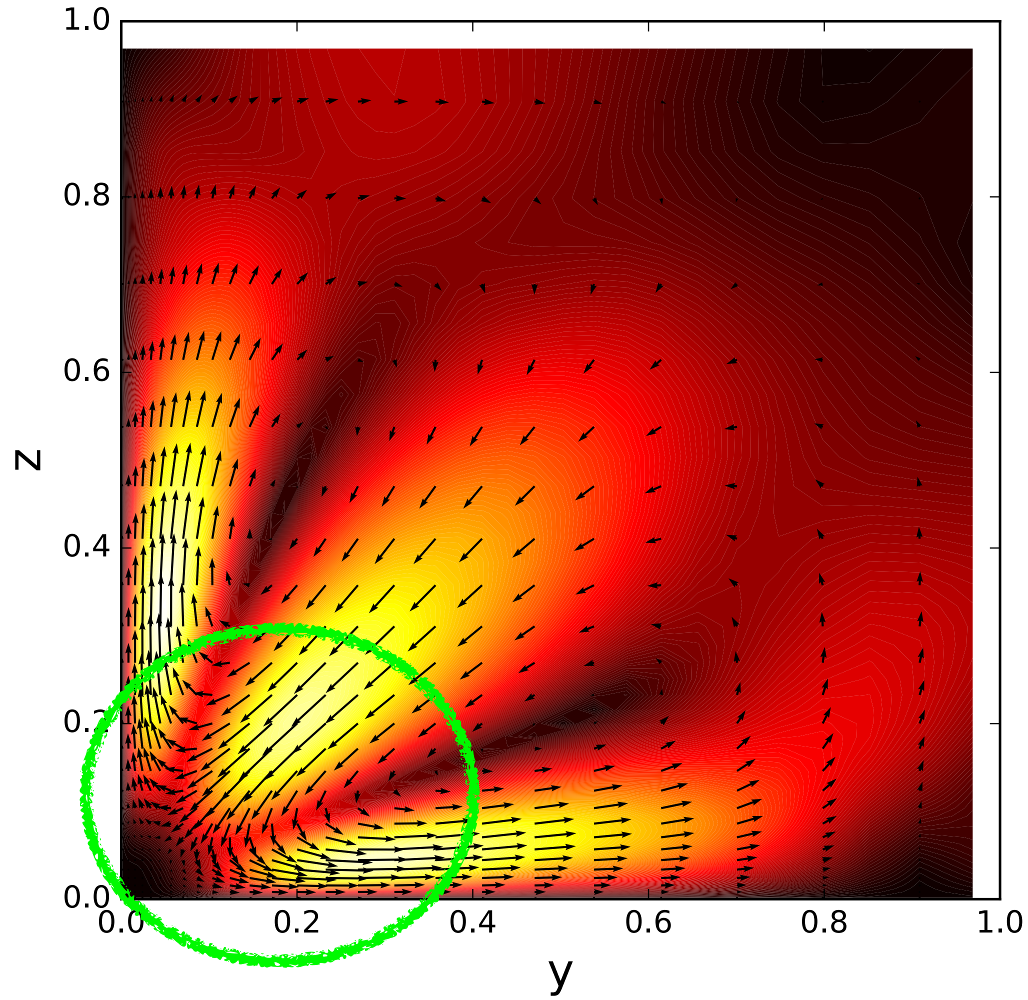


Baseline:RSTM
DNS
ML Prediction
τ_{zz}
τ_{yy}
Tau ZZ
Tau_org YY
Tau YY

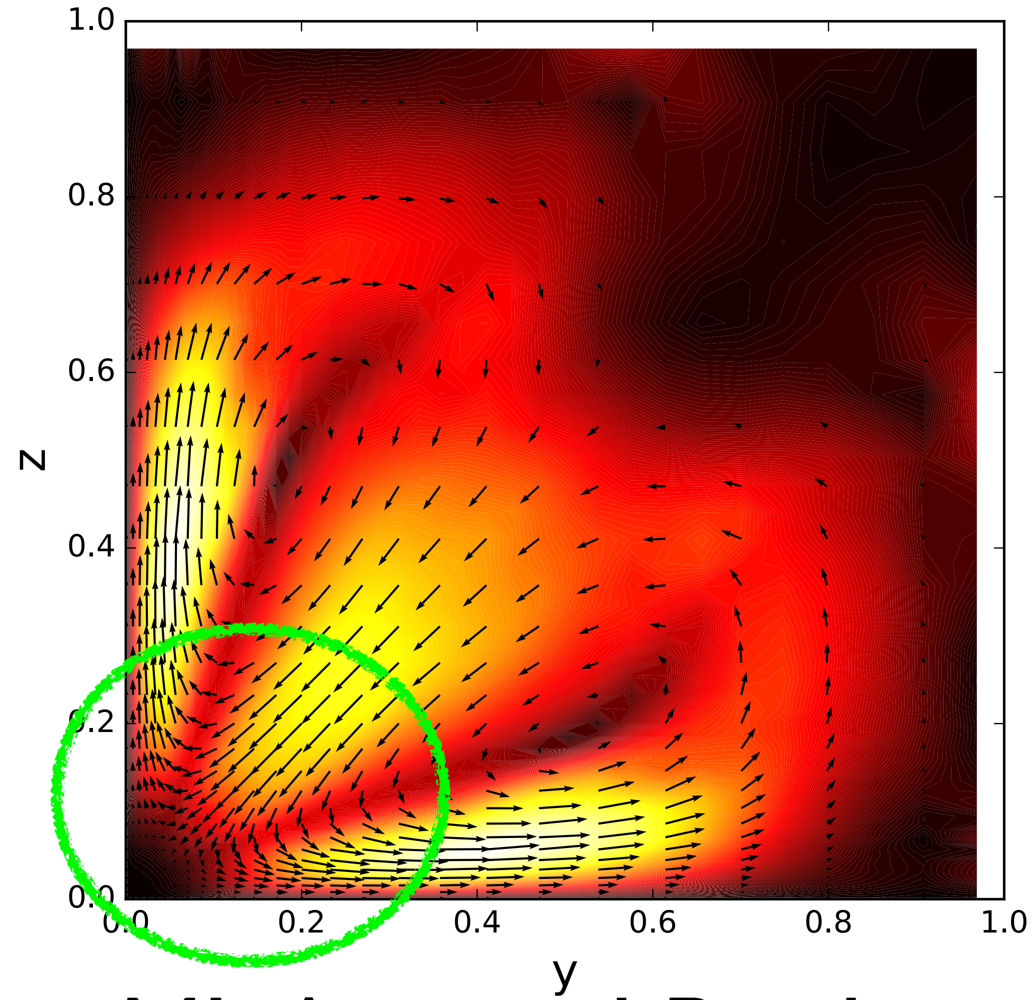
Velocity *Prediction* with Machine Learning *Corrected* Reynolds Stresses



Baseline (RSTM)



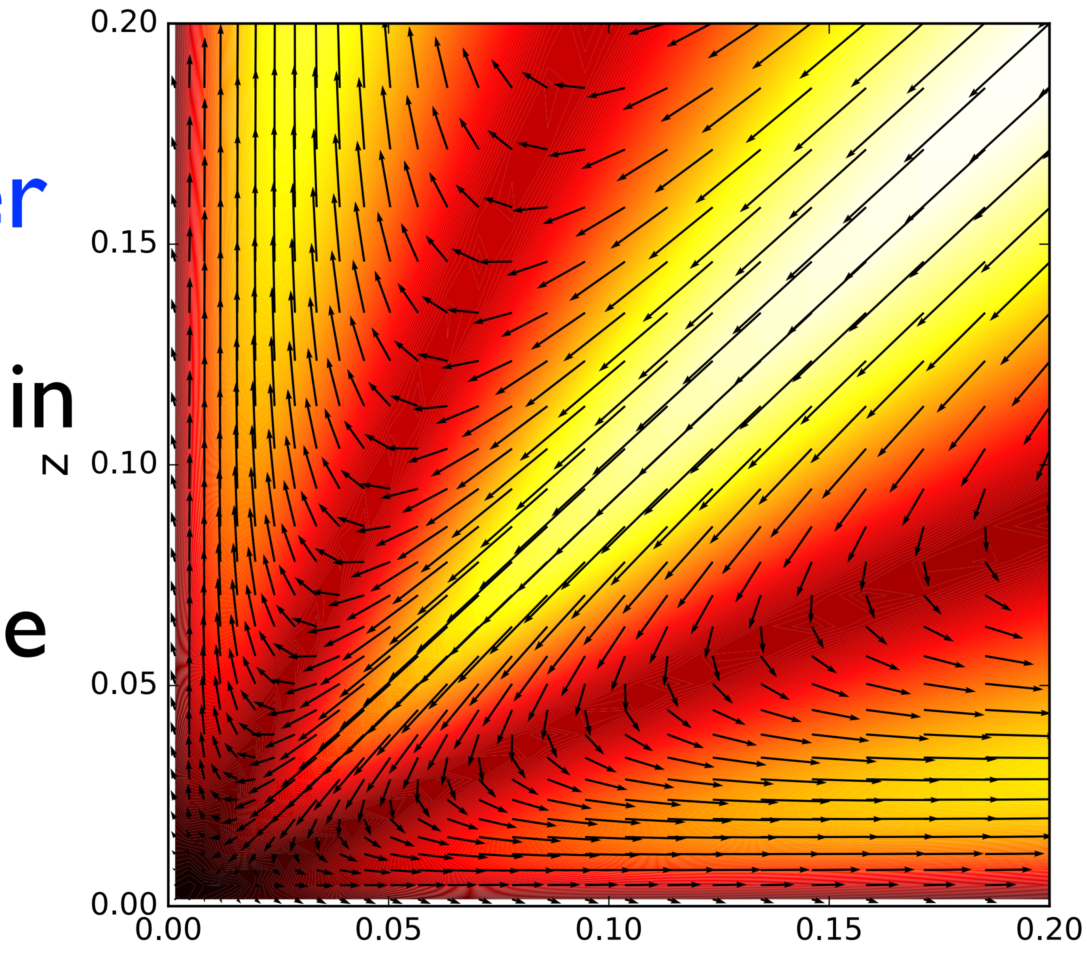
DNS



ML Assisted Prediction

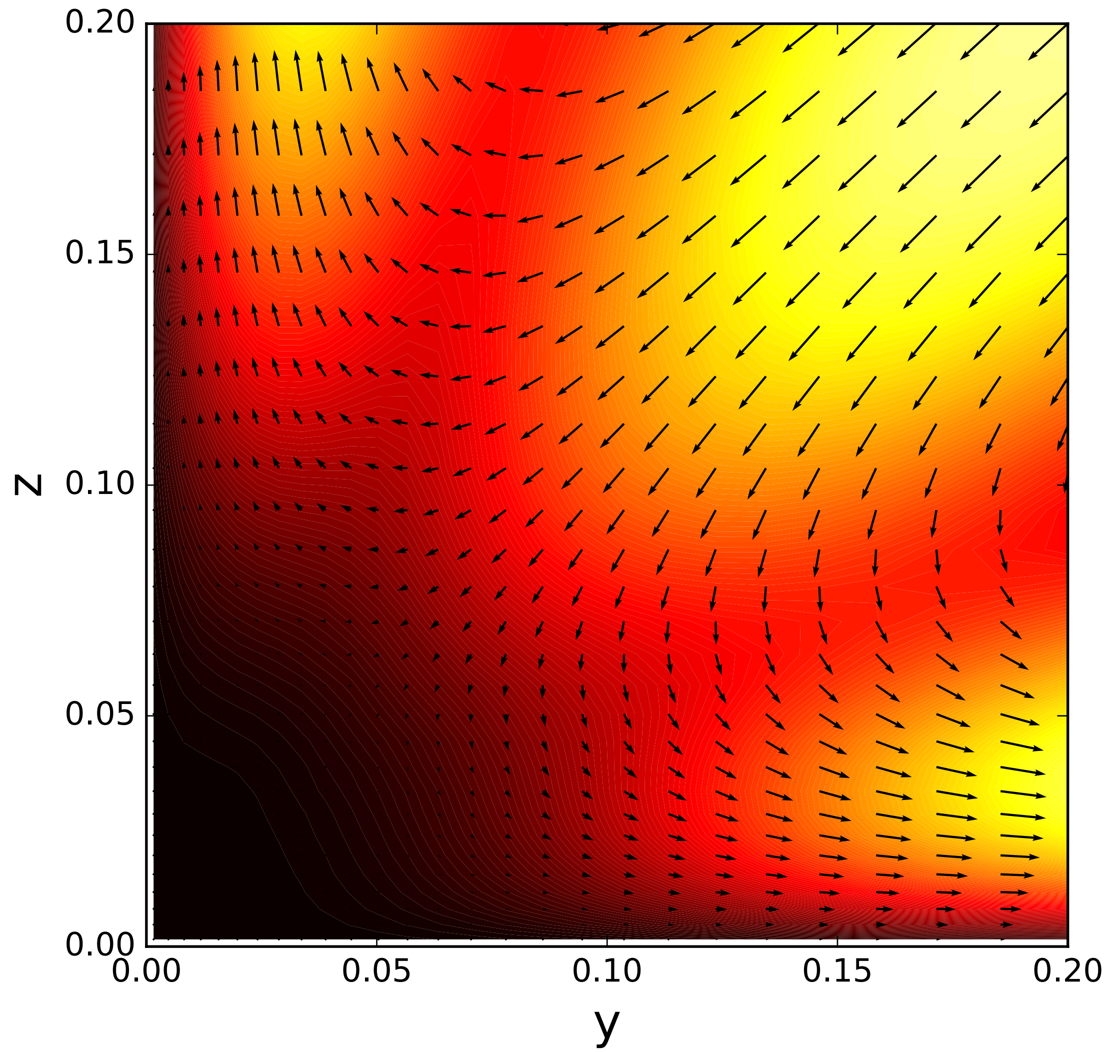
Secondary velocity pattern near corner

Significance success in using ML towards predictive turbulence modeling.

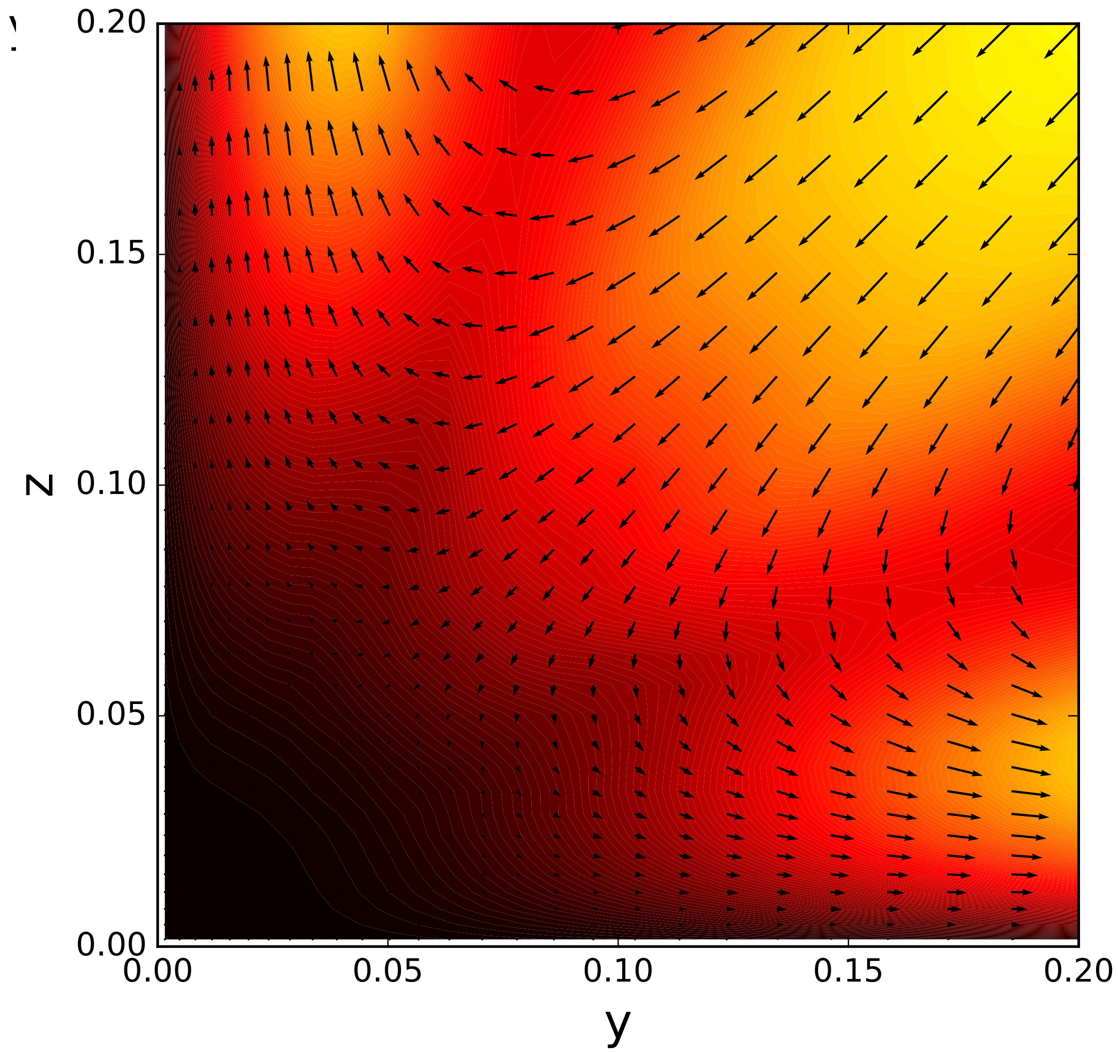


Baseline (RSTM)

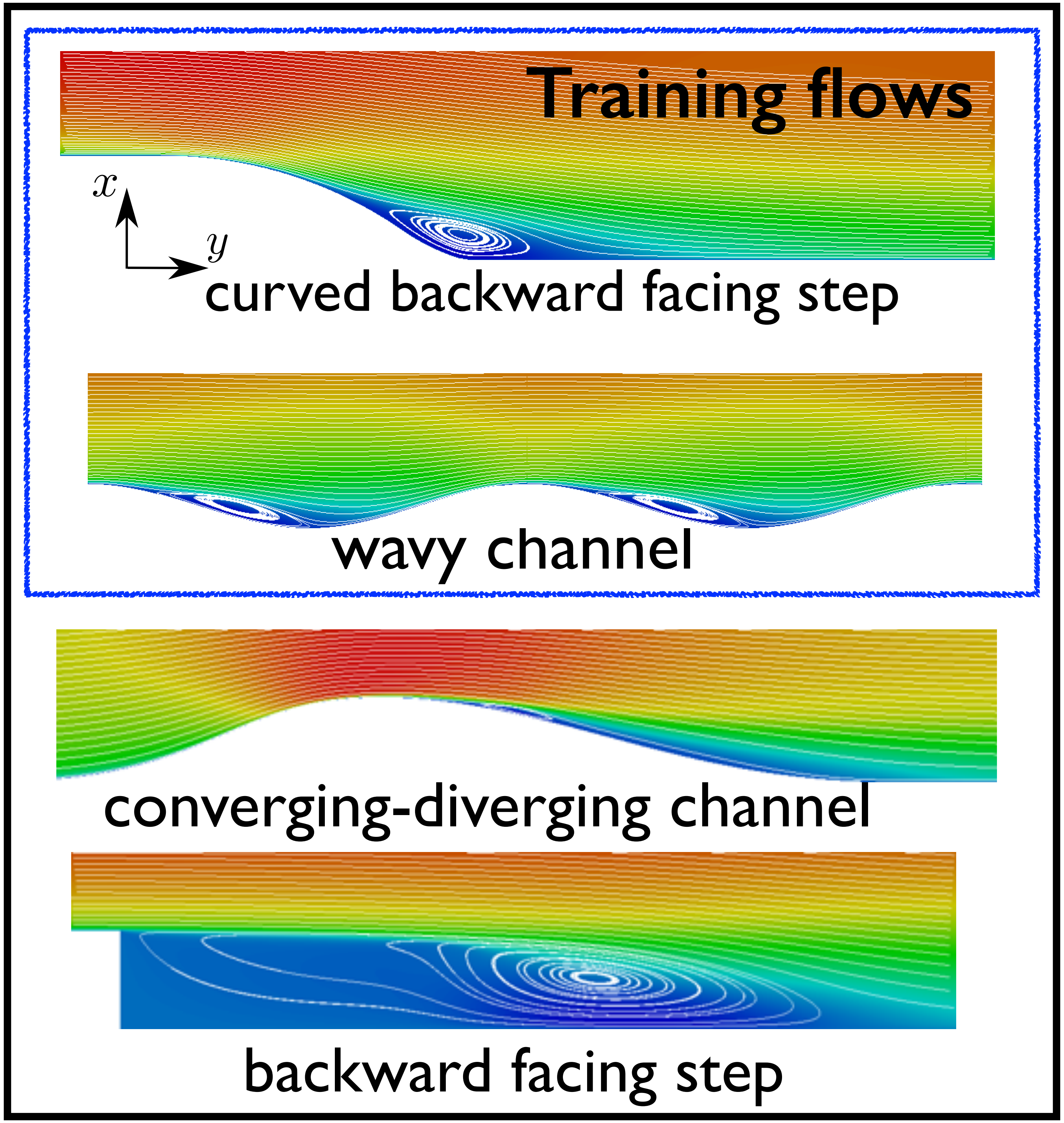
DNS

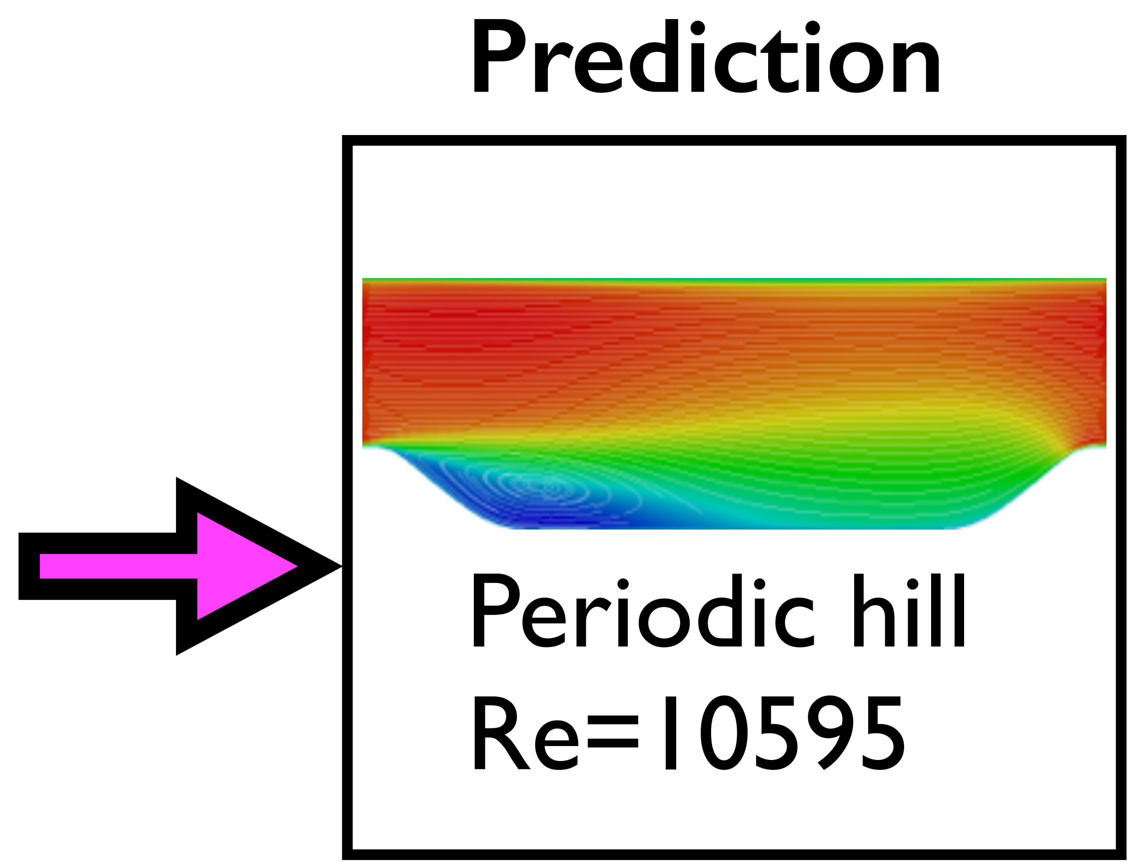


ML-Assisted Prediction



Case Study:
Separated Flows in Different Geometries



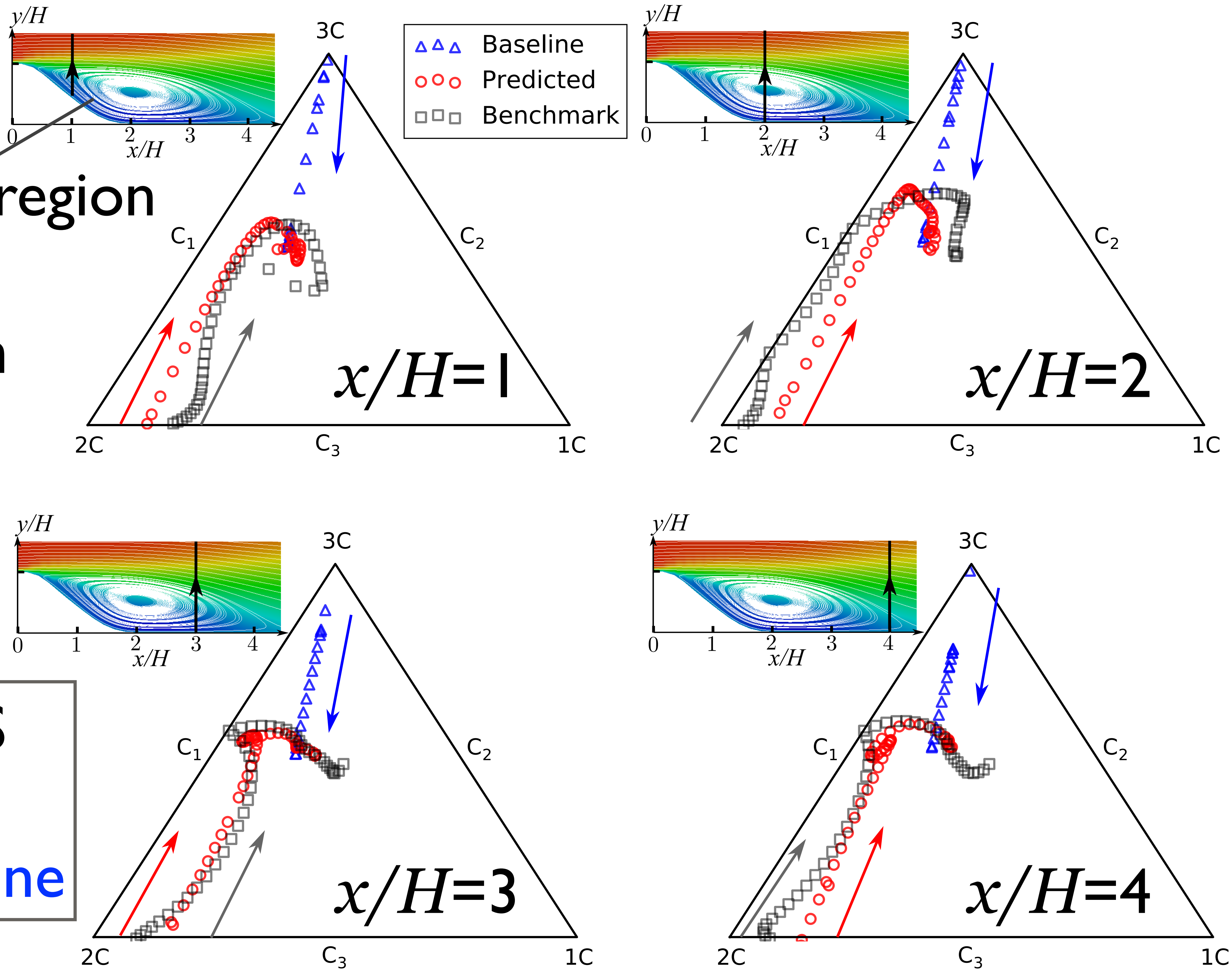


Source: http://turbmodels.larc.nasa.gov/other_dns.html

Predicted Anisotropy in Separated Region

Separation region

Trained on cases with **different geometry**



Black: DNS
Red: ML
Blue: Baseline

Beyond Turbulence Modeling

- **Constitutive modeling of complex materials**
- Dynamics of atmospheric, ocean, and climate system
- Combustion
- … …

Similar challenges to turbulent flows:

1. We do not **understand** the physics well enough to describe/model them (e.g., chemical reactions)
2. We cannot **afford** the computational cost to adequately resolve the physics (e.g., micro-fibers, grains; cloud, ABL, terrain)

Analogy between turbulent flows and dynamics of complex materials

Analogy Between Turbulence & Solid Mechanics

- Turbulence can be considered "a fluid with complex constitutive behavior".

$$\boldsymbol{\tau} = 2\nu_t \mathbf{S}$$

$$\mathbf{S} = \frac{1}{2}(\nabla U + \nabla^t U)$$

U = velocity

ν_t = turbulent viscosity

- Complex materials can be considered "a solid with complex constitutive behavior (stress/strain relation)".

$$\boldsymbol{\tau} = \mathbf{E}\boldsymbol{\varepsilon}$$

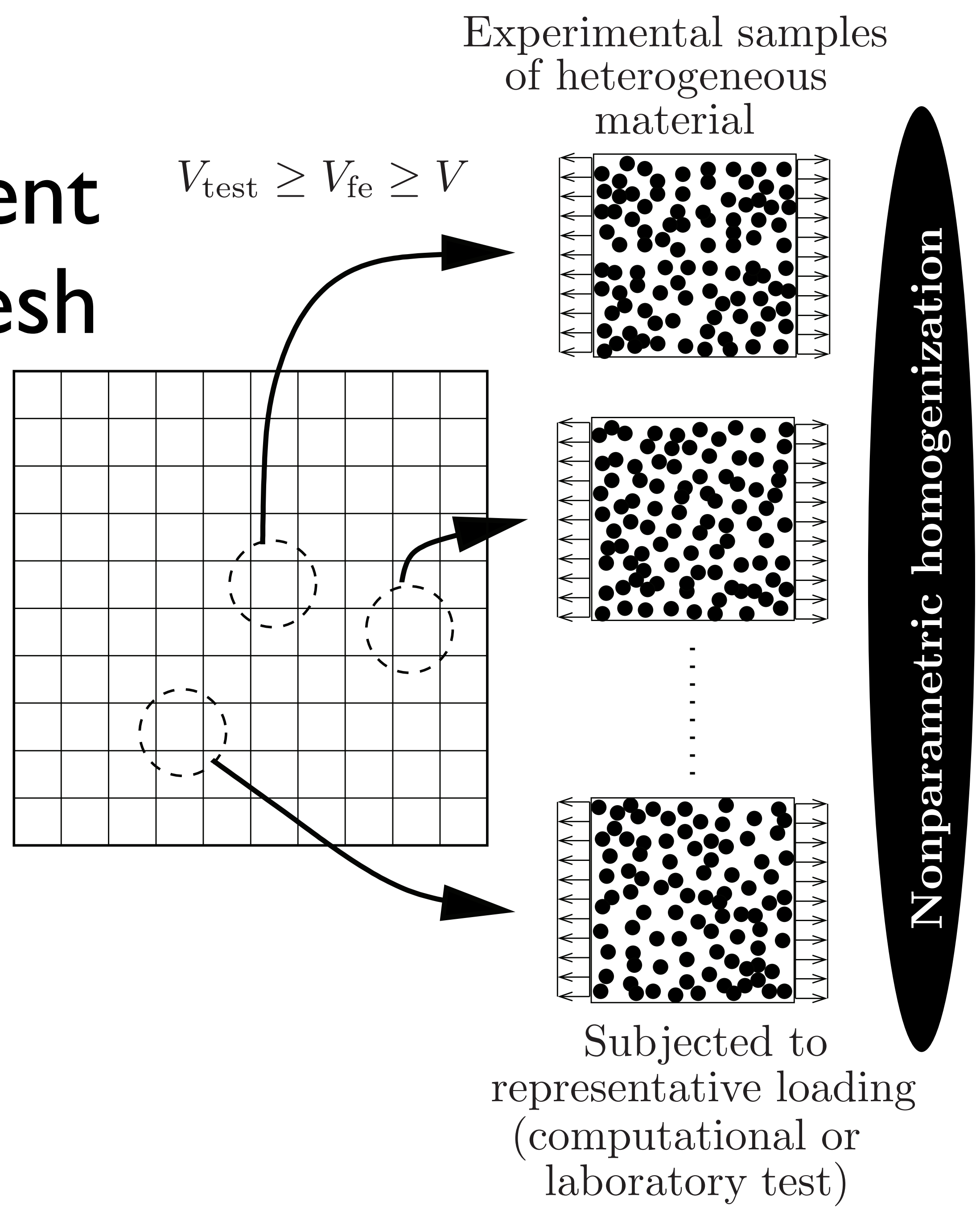
$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla U + \nabla^t U)$$

U = displacement

E = effective modulus

Complex Heterogeneous Material

Finite element mesh



apparently stress-dependent material property at the macro-scale

[Das & Ghanem, 2009]

Where does the **Complex Behavior** come from?

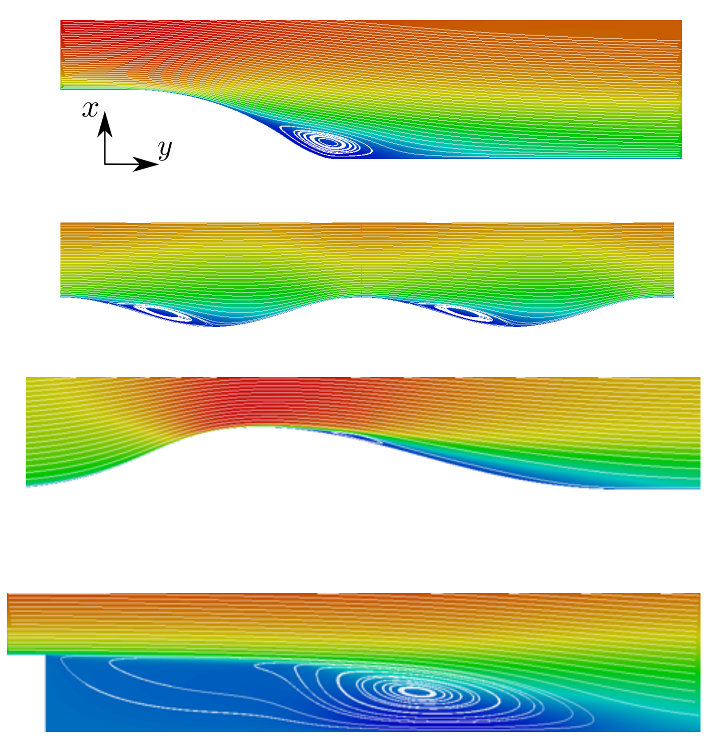
- The "complex behaviors" in both problems do not really exist if they were fully resolved, i.e.,
  - directly resolve all scales in turbulent flows (Direct Numerical Simulation).
  - directly resolve all meso-scale constituents (fully resolved FEM)
- The "apparently" complex constitutive behavior is due to the **modeling** of *unresolved scales.*
- As a result the constitutive coefficients (ν_t or **E**) are properties of the flow dynamics or structural dynamics, and not the property of the materials (fluid or solid).

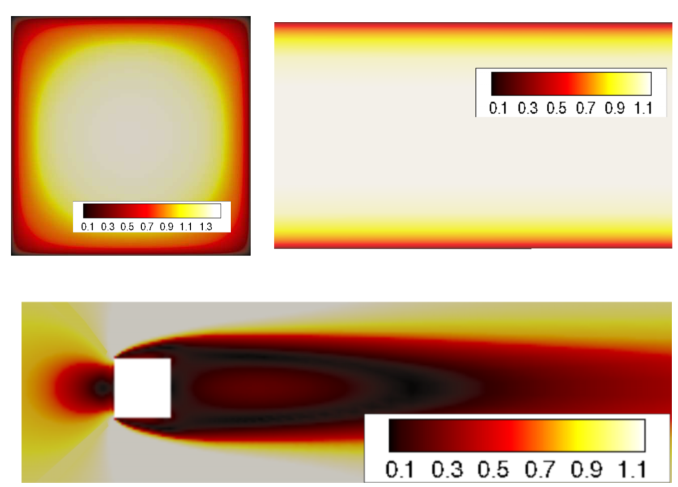
Summary

- Using ML in computational physics has unique challenges.
- In physics-informed machine learning, we utilize physical constraints in all aspects of machine learning to address these challenges.
- Choose universal quantities based on physical prior knowledge.
- Preliminary success in RANS based turbulence modeling. The objective is co-design of ML algorithm and problem formulation.
- Has potential well beyond turbulence modeling.

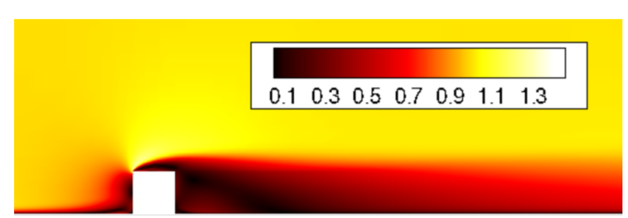
Collaborations Ideas

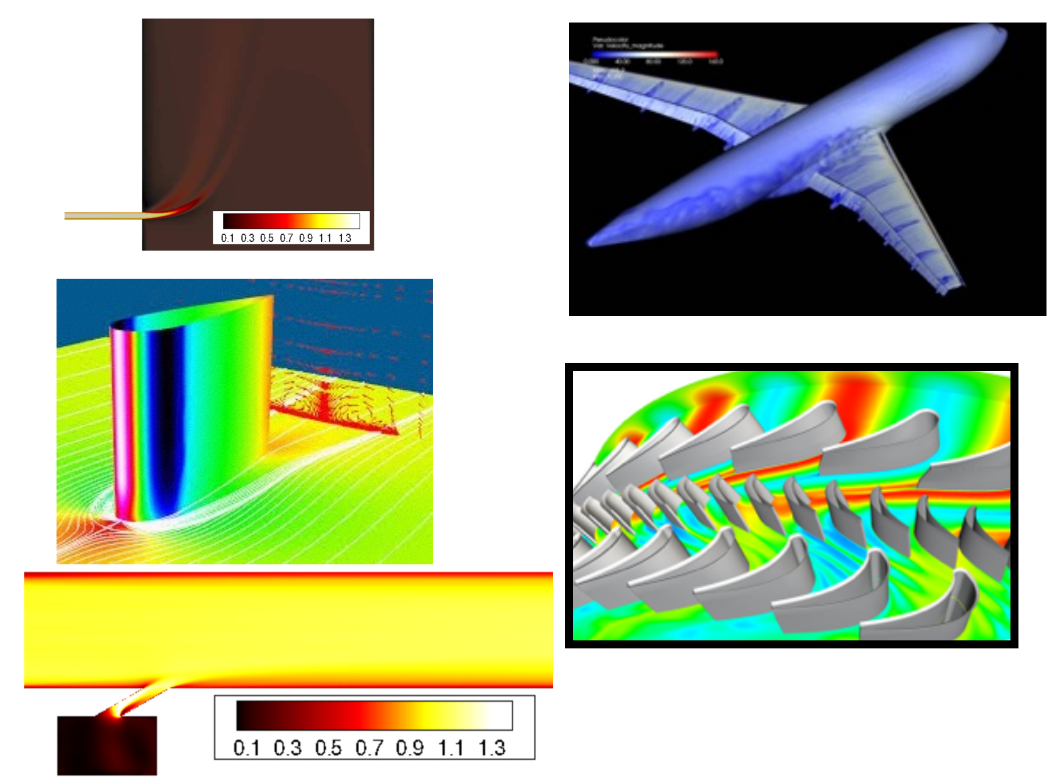
- **Now**: separate functions for each flow class; **Eventually**: ML algorithm choose data automatically.
- Need benchmark database: elementary flows (free shear, plane channel), flows of medium complexity (separation, airfoil), to realistic flows (wing-body junction) and industrial flows.





(g)Case 7: Flow around a Square Cylinder





Collaboration Ideas

- Evaluation of the PIML method in turbulence models relevant to NASA (SA, k-ω SST, maybe an EARSM/RSTM)
- Dissemination by implementing/distributing in NASA codes (e.g., CFL3D, FUN3D, OVERFLOW). Current implementations are in open-source code OpenFOAM with Python scripts.
- Extensions beyond RANS-based turbulence modeling.

Thank you!

Heng Xiao, Virginia Tech

hengxiao@vt.edu

https://sites.google.com/a/vt.edu/hengxiao/home

Physics-Informed Machine Learning: Perspectives

Assist but respect models: Machine learning should be used to correct/improve existing models, not to replace them. Thus, we learn the model discrepancy, not the model output directly. (*consensus*)

1. Choose quantities that have physical bounds/constraints/interpretation to learn (allow for anchoring to physics).

2. Learned quantities should be **universal** to some extent: same functional form in training and prediction flows! Note the limitation of universality though…

3. **Obey physical constraints** in the learning as much as possible.