Machine learning for turbulence modeling: A (turbulence traditionalist's) perspective

Sharath S. Girimaji

Collaborators: Salar Taghizadeh and Freddie Witherden

Texas A&M University

NSF WORKSHOP ON EXUBERANCE OF MACHINE LEARNING IN TRANSPORT PHENOMENA
FEBRUARY 10 — 11, 2020, DALLAS, TX

Context of Talk

- Data-Driven Modeling (DDM) / Machine Learning (ML) has been very successful in many areas of science and engineering
- Can DDM/ML help to `solve' the age-old problem of turbulence

The purpose of this talk:

1. Ask questions of ML – as an agnostic
2. Seek answers – as an optimistic pragmatist

Some preliminary comments

Three-part talk

1. Opinions about the amenability of turbulence phenomenon to Data-Driven Modeling (DDM) or Machine Learning (ML)

2. ML for different level of turbulence closures

3. Rudimentary ML computations intended for illustrations of concepts

`Rise and Fall of turbulence theories’

Many `promising’ approaches have been applied to turbulence

1. Renormalization Group (Ken Wilson, 1980s – Nobel Prize)
   - Extremely successful for Quantum Electro Dynamics
2. Lattice Gas Automata (Steve Wolfram, 2000s)
   - Successful in many areas of biological process modeling
3. Many mathematical tools: POD, wavelets, fractals etc.

- Each approach has added important value to turbulence research, but not solved the problem
- These investigations have only added to the mystique of turbulence

Soul Searching in Field of Turbulence

1. Stanford, 1968: Turbulence Olympics
    - Review of various turbulence models
    - DuPont Donaldson laid the foundation for formal closure modeling
2. Cornell, 1990: Whither Turbulence? Turbulence at Cross-Roads
    - \`Traditional modeling' vs. DNS vs. Coherent Structures vs. Lattice Gas
    - Lumley's ***\`Tortoise vs. Hare' analogy*** for ***\`Traditional vs. Trendy'*** methods
    - Role of funding agencies in promoting one approach vs other

Where do we stand now? (Lumley and Yaglom, 2001)

- We believe that even after 100 years, turbulence studies are still in their infancy
- We do have a crude, practical, working understanding of many turbulence phenomena but certainly *nothing approaching a comprehensive theory, and nothing that will provide predictions of an accuracy demanded by designers.*

Turbulence Phenomena - Challenges

- Non-linearity and large number of degrees of freedom
- More importantly, non-locality with long-range interactions
  - Elliptic nature of pressure
- Spatio-temporal chaos
- `Complex' phenomenon
  - Emergent behavior
  - Some self-organization
- Intermittency

Can ML help

Physics-based modeling

- Longstanding approach leading to important theories
- *Approximate representation of a `larger truth'*
- Imprecise but holistic

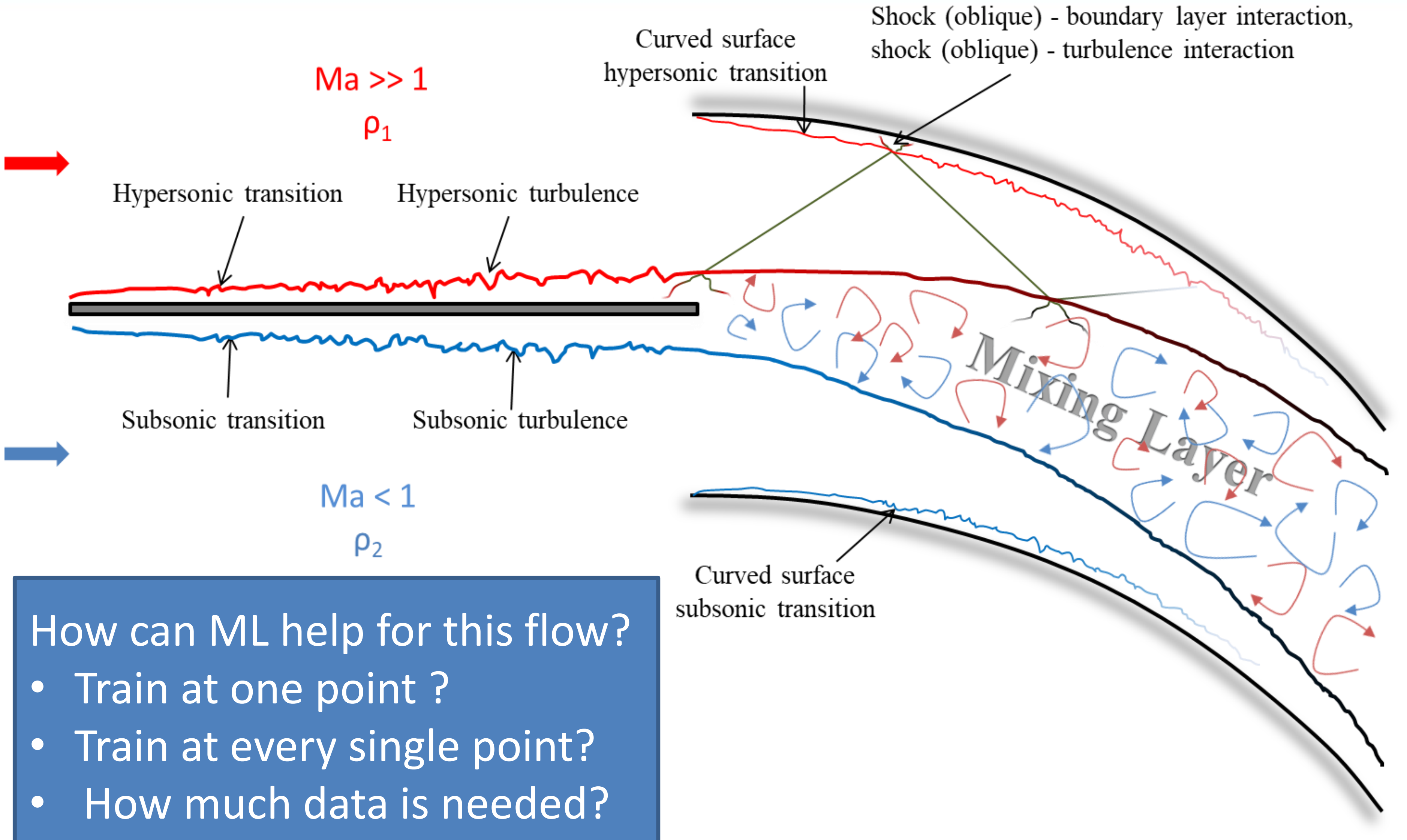
Machine Learning (ML)

- Recent success in many areas of science and engineering
- *Precise quantification of observed data*
- Accurate but inherently incomplete → not easily generalizable

Can the weaknesses of physics-based models be overcome with ML

- Will we get an all-encompassing theory?
- Will designers get the accurate predictions for practical flows?

Simple Application

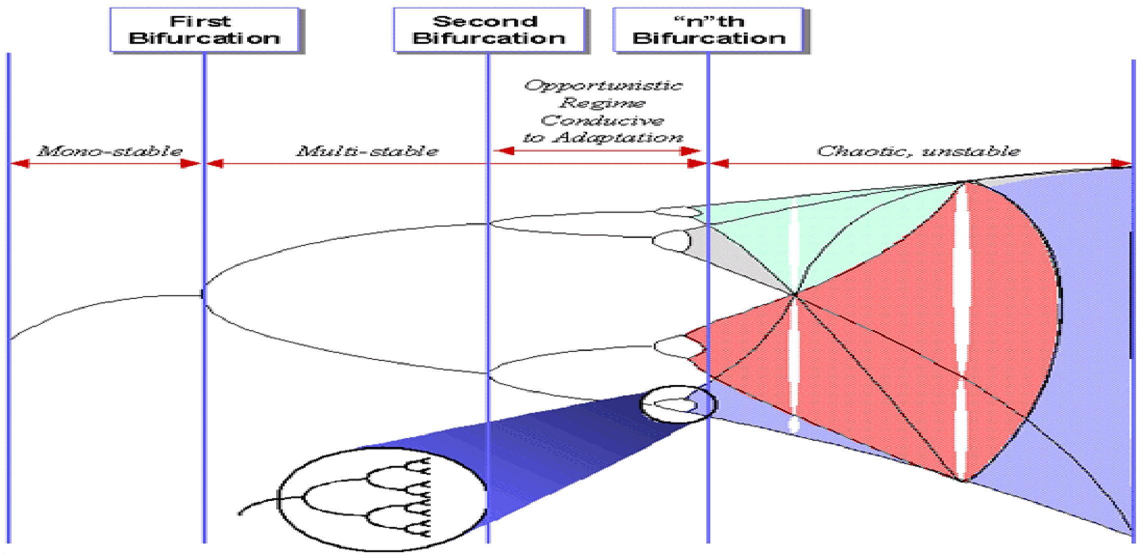


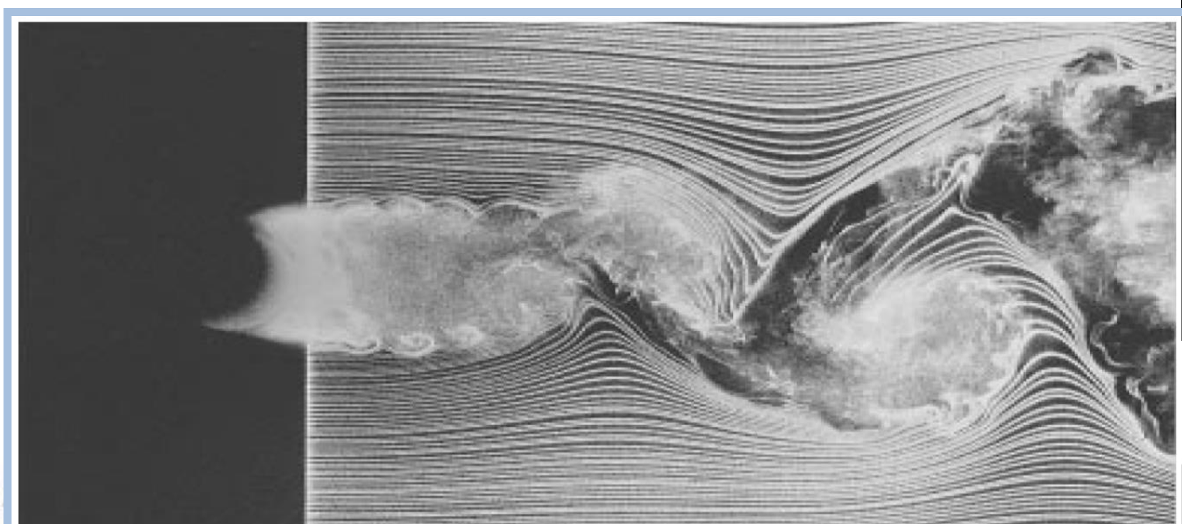
How can ML help for this flow?

- Train at one point ?
- Train at every single point?
- How much data is needed?

Flows with spatially developing structures

Breakdown from one state of turbulence to another





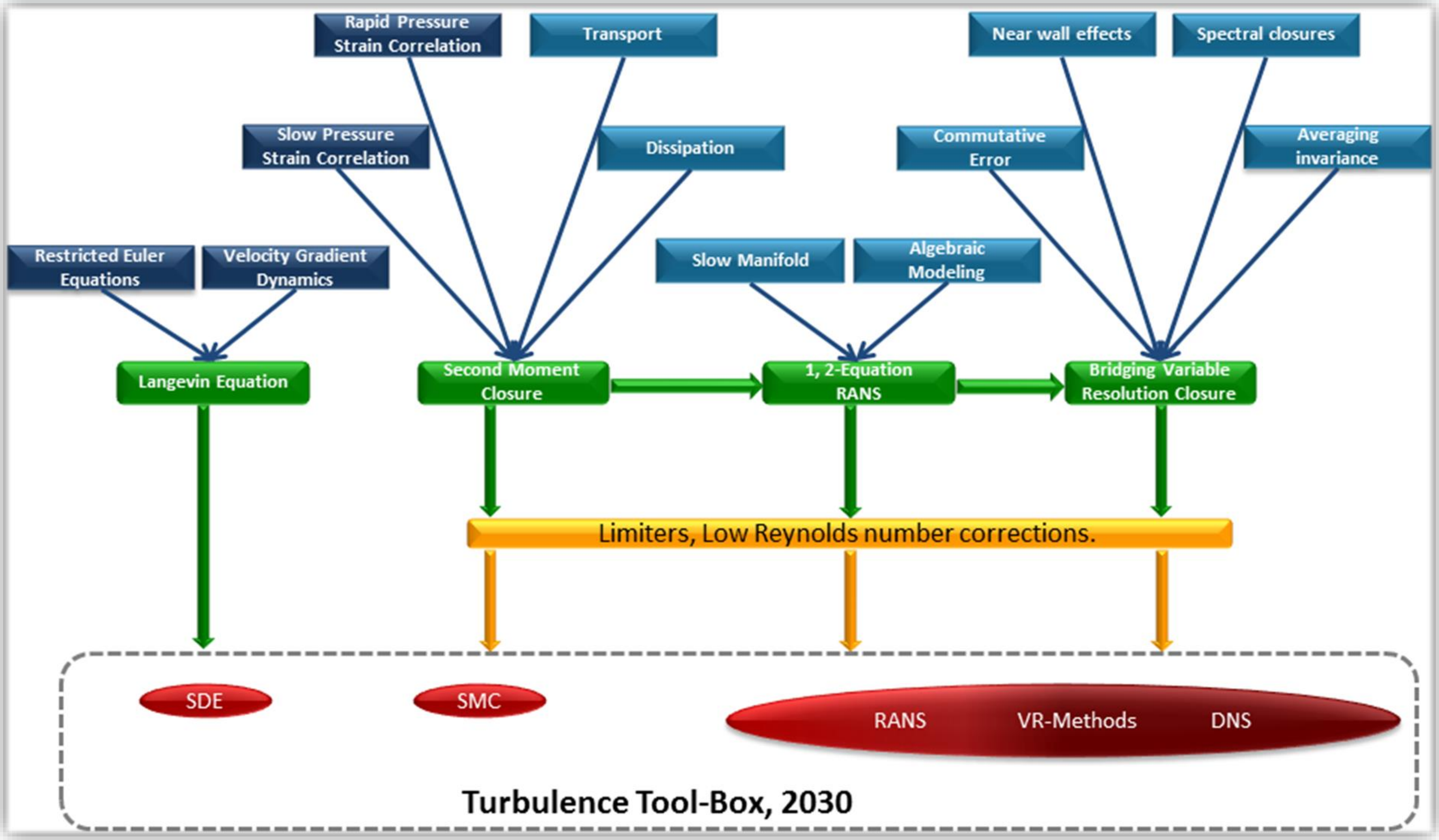
- Resolve what we cannot model
- Model what physics allows
- Have the wisdom to know the difference

Top turbulence modeling challenges

1. Non-linear/anisotropic viscous constitutive relation
2. Spatio-temporal non-locality of stress dependence on strain field
   - *Rapid-distortion → Viscous vs. visco-elastic behavior*
   - *Non-equilibrium turbulence*
3. Multiple production mechanisms
   - *Shear; stratification; magnetic field, etc*
4. Multi-physics effects including flow-chemistry interactions
5. Change in equation of state
   - Comp. effects; flow-therm interactions, thermodynamic non-eqbm
   - Physics different with increasing Ma as thermodynamic interactions change
6. Spatially-evolving flows with multiple equilibrium states
7. Large-scale unsteadiness and coherent structures

The onus on the turbulence model can be reduced by resolving more scales

Turbulence Tool-Box



Modeling methods: Attributes and Limitations

1. **2-equation RANS model**
   - Needs non-linear constitutive relation for many flows
   - Cannot capture non-eqbm effects and instabilities/coherent structures
2. **7-equation RSCM**
   - Need models for pressure-strain correlation
   - Can capture simple non-eqbm effects but not instabilities/structures
3. **Scale Resolving simulations & Large-eddy simulations**
   - Need subgrid models; lower degree of modeling difficulty than RANS models
   - Computationally very expensive but can potentially capture relevant physics
4. **Stochastic differential equations**
   - Based on probability distribution function of flow variables
   - Least developed (and least understood) of all turbulence modeling methods
   - Has potential to overcome many shortcomings of moment-based methods

How can ML help RANS and LES?

- Improve constitutive relation & transport equation coefficients
- Still may not capture effects of non-eqbm, instabilities, structures
- Generalization to different class of flows still a major challenge

Do current methods make effective use of data?

- Velocity field is Gaussian & does not reflect turbulence complexity
- Averaging the data further eliminates key physics
- Physics incumbent in pressure and velocity gradient pdf ignored
- Overall, the richness of information available in data is under-utilized

ML for SRS and LES

- Lower degree of closure difficulty: complexity is eliminated due to flow resolution
- Much more data needed as all models must be **conditioned on the state of resolved flow field**

How can ML help 7-equation RSCM?

$$\frac{D\langle u_i u_j\rangle}{Dt} = \frac{\partial\langle u_i u_j\rangle}{\partial t} + \langle U_k\rangle \frac{\partial\langle u_i u_j\rangle}{\partial x_k} = P_{ij} - \epsilon_{ij} + \Pi_{ij} + T_{ij}$$

$$Production:\ P_{ij} = -\langle u_i u_k\rangle \frac{\partial\langle U_j\rangle}{\partial x_k} - -\langle u_j u_k\rangle \frac{\partial\langle U_i\rangle}{\partial x_k}$$

$$Disipation:\ \ \epsilon_{ij} = 2\nu \left\langle \frac{\partial u_i}{\partial x_k} \frac{\partial u_j}{\partial x_k} \right\rangle$$

$$PSC:\ \Pi_{ij} = 2\nu\langle p S_{ij}\rangle$$

$$Transport:\ T_{ij} = \frac{\partial}{\partial x_l}\left[-\langle p u_i\rangle \delta_{jl} - \langle p u_j\rangle \delta_{il} + \nu \frac{\partial\langle u_i u_j\rangle}{\partial x_l} - \langle u_i u_j u_l\rangle\right]$$

- ML-enhanced Pressure-strain correlation models can be developed
  - PSC is the game-changing effect in RSCM
  - Realizability, RDT consistency can be applied with greater fidelity
  - Greater potential of generalizability than RANS
- Turbulent transport model can also be developed
  - Strong anisotropy effects and secondary flows can be better captures
- Overall, much better utilization of information incumbent in data
  - Higher order and mixed moments of data used

How can ML help Stochastic Closures?

Velocity-gradient evolution eqn

$$\frac{d}{dT}\left(\frac{\partial u_i^+}{\partial x_j}\right) + \mathcal{N}_{ij} = [\mathcal{P}_{ij} + \mathcal{V}_{ij}](x^+, T)$$

N → Non-linear; P → pressure; V → Viscous effects

Langevin Equation

$$dh_{ij} = [-N_{ij} + M_{ij}]dt + D_{ijkl}\, dW_{kl}$$

PDF Equation

$$\frac{df_m}{dt} = -\frac{\partial}{\partial h_{ij}}[f_m(M_{ij} - N_{ij})] + \frac{1}{2}\frac{\partial^2}{\partial h_{kl}\,\partial h_{pq}}[D_{ijkl}\, D_{ijpq}\, f_m]$$

- Mij and Dijkl require closure modeling
- The full pdf of data can be used in ML-enhanced modeling
- **Best utilization of all information incumbent in hi-fidelity data**
- **Promising ML-based approach, but in preliminary stage**

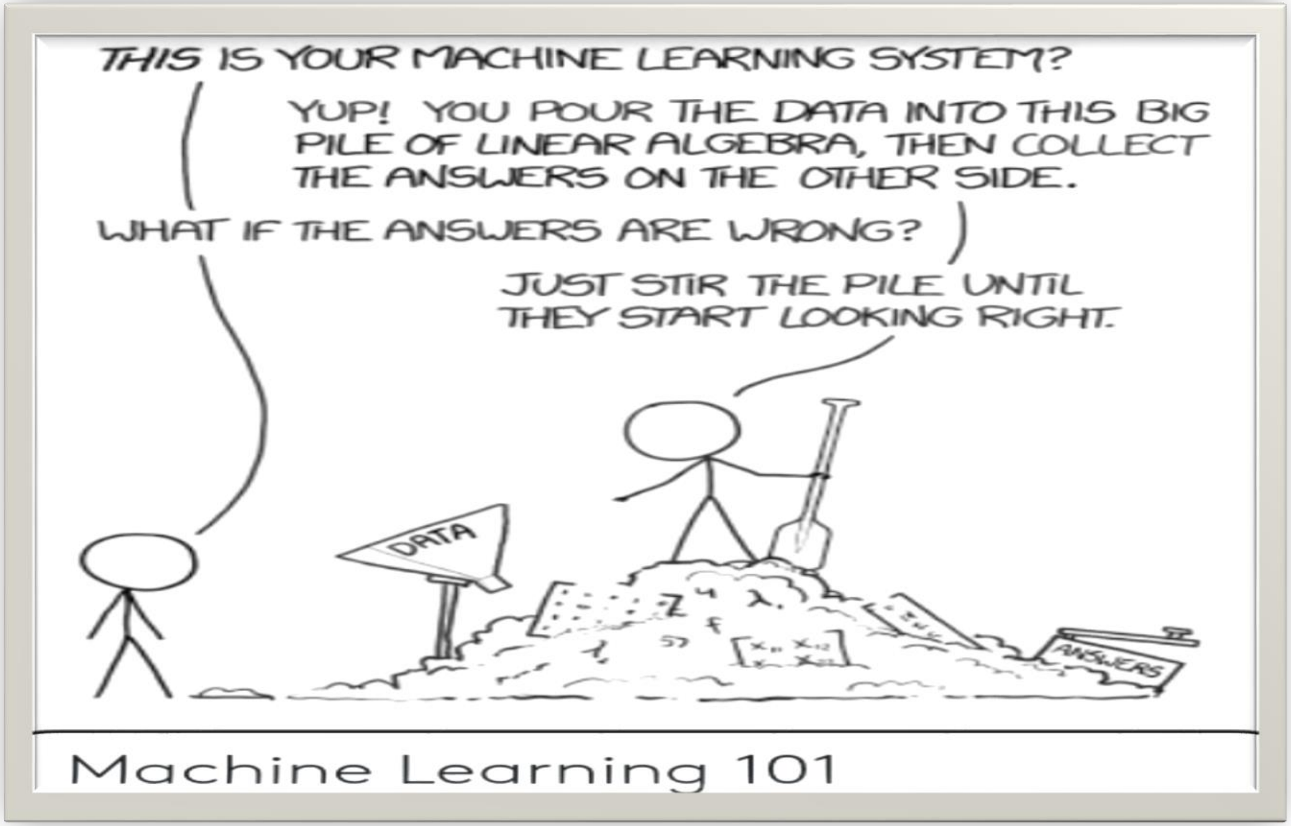
Part 2: Current ML-enhanced RANS

- Physical consistency & generalizability of certain current approaches
- Different flow scenarios
  - When underlying RANS model is reasonable
  - When underlying RANS model is incorrect
    - Can ML help to yield reasonable results
- Open vs. Closed-loop training
- Computations from closed-loop training

ML for 2-equation Model – Questions?

Can we standardize the training procedure?

- Which Neural Network Architectures? How many features?
- What is the right objective function?



DDM/ML for RANS

Constitutive coefficients: Algebraic Equations

- Use of ML best developed for this piece of turbulence modeling
- Representation theory used for Feature Selection
- But in many instances, constitutive equation is not weakest link

Transport equations: Weakest links

- Can ML help modeling production and destruction of dissipation?
- How can ML help in turbulent transport modeling?
- Representation theory is not useful as these are scalar equations
- Objective functions may be integro-differential equations!

Internal consistency of traditional RANS

- Constitutive Closure Coefficients (CCC):

$$\langle u_i u_j \rangle = -\tau_{ij} = 2k b_{ij}(s_{ij}, w_{ij}) + \frac{2}{3}k\delta_{ij}, \qquad \boldsymbol{b}(\boldsymbol{s}, \boldsymbol{w}) = \sum_{\lambda=1}^{10} G_\lambda (I_{1:5}) \boldsymbol{T}^\lambda$$

- Transport eqn. Closure Coefficients (TCC):

$$\rho \frac{\partial k}{\partial t} + \rho \langle U_j \rangle \frac{\partial k}{\partial x_j} = \tau_{ij} \frac{\partial \langle U_i \rangle}{\partial x_j} - \beta^* \rho k \omega + \frac{\partial}{\partial x_j}\left[(\mu + \sigma^* \mu_t) \frac{\partial k}{\partial x_j}\right]$$

$$\rho \frac{\partial \omega}{\partial t} + \rho \langle U_j \rangle \frac{\partial \omega}{\partial x_j} = \alpha \frac{\omega}{k} \tau_{ij} \frac{\partial \langle U_i \rangle}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j}\left[(\mu + \sigma \mu_t) \frac{\partial \omega}{\partial x_j}\right]$$

- These coefficients need calibration,

CCC: $G_1 \ldots G_{10}$

TCC: $\alpha, \beta, \beta^*, \sigma, \sigma^*$

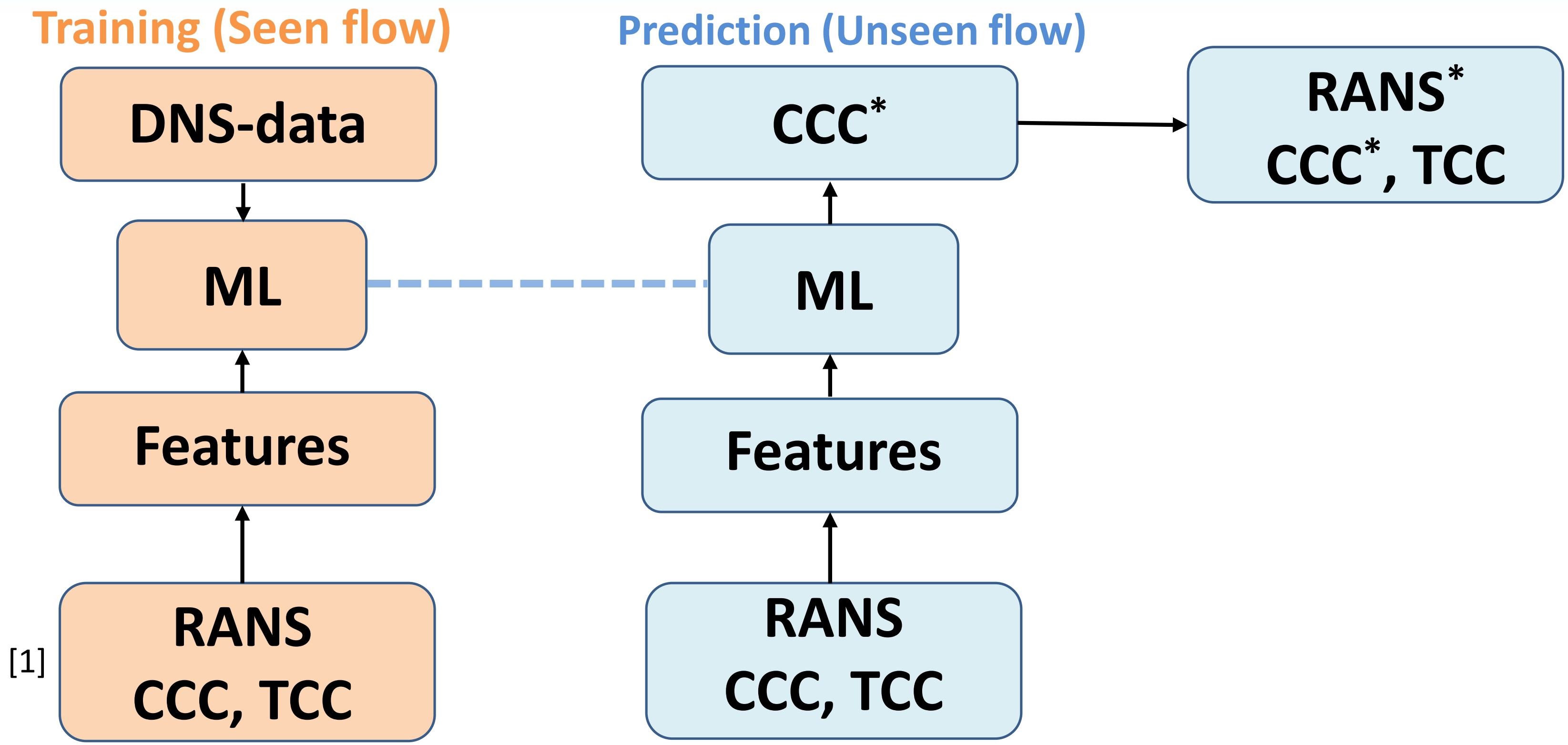
- Turbulence physics requires CC to be **self-consistent** and satisfy: [1,2]

  - Fixed point behavior, realizability, rapid distortion limit

- In traditional TM, **self-consistency** is guaranteed using **dynamical system analysis**

[1]Girimaji, S.S., 2000. *Journal of Fluid Mechanics*.
[2]Girimaji, S.S., 1996. *Theoretical and Computational Fluid Dynamics*.

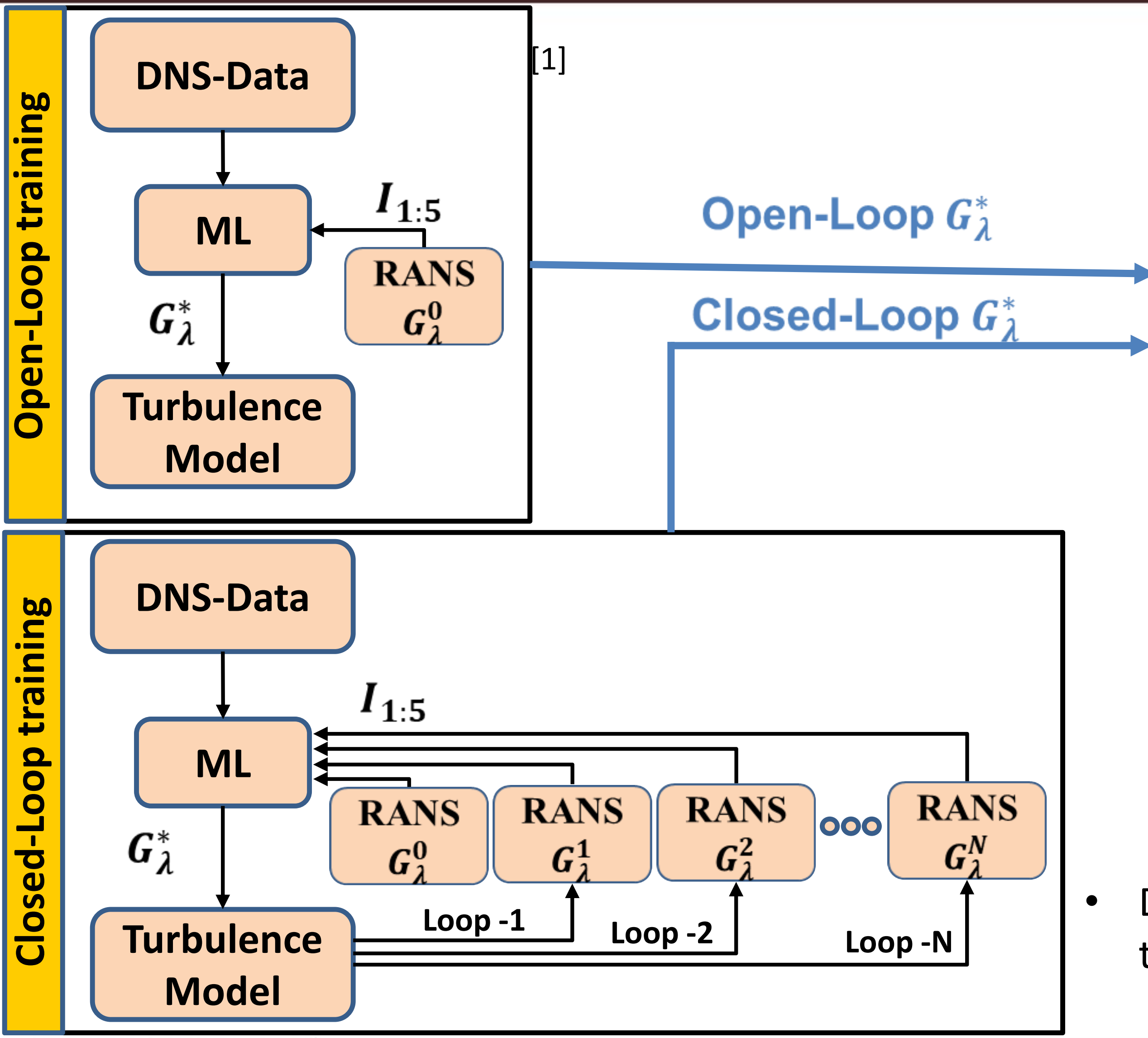
How do we use ML for TM?

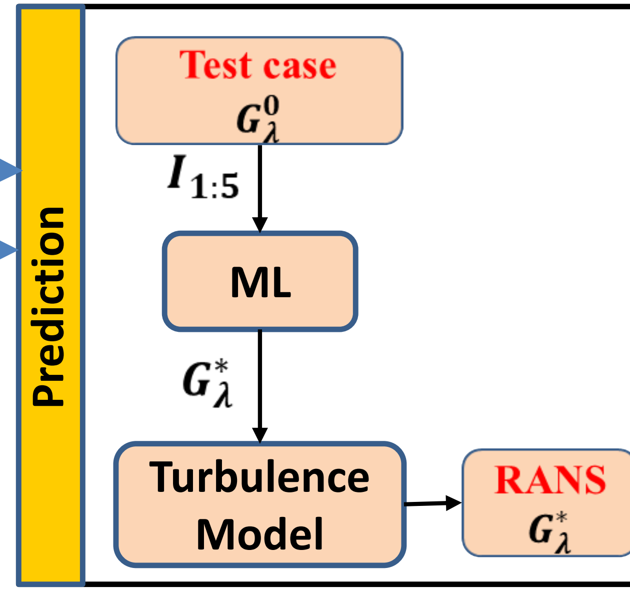


- A priori CCC and A posteriori CCC* are not **self-consistent**
- Further TCC and CCC* are not compatible
- Inconsistency and incompatibility will affect generalization to unseen flow
- **Dynamical system analysis for ML-based features is not developed.**
- **How can we improve internal consistency?**

[1]Ling, J., Kurzawski, A. and Templeton, J., 2016. *Journal of Fluid Mechanics*.

Closed loop training





- Different from the CFD-driven training that is introduced in [2].

[1]Ling, J., Kurzawski, A. and Templeton, 2016. *Journal of Fluid Mechanics.*
[2]Zhao, Y., Akolekar, H.D., Weatheritt, J., Michelassi, V. and Sandberg, R.D., 2019. *arXiv*

Part 3: Closed-loop training

Objectives:

- demonstrate internal inconsistency in current approaches
- Demonstrate closed-loop training better when `original' model is wrong

Proof of concept in simplest flow possible

- Channel flow in which current models already perform well
- To simulate unseen flow conditions, change model coefficients
- Examine if ML leads to recovery of original coefficients

Test cases

- Study 1 - Standard k-ω

G_1	G_2	G_3	G_4	α	β	β^*	σ	σ^*
-0.09	0	0	0	0.52	0.072	0.09	0.5	0.5

- Study 2 - Modified CCC model

G_1	G_2	G_3	G_4	α	β	β^*	σ	σ^*
~~-0.09~~ **-0.045**	0	0	0	0.52	0.072	0.09	~~0.5~~ **0.23**	~~0.5~~ **0.23**

(Equilibrium boundary layer analysis)

- Study 3 - Modified TCC model

G_1	G_2	G_3	G_4	α	β	β^*	σ	σ^*
-0.09	0	0	0	0.52	~~0.072~~ **0.054**	0.09	~~0.5~~ **0.143**	~~0.5~~ **0.143**

- Preliminary computation for channel flow $Re_\tau = 1000$
- Reference DNS data obtained form, Lee & Moser.

Lee, M. and Moser, R.D., 2015. JFM.

Desired behavior & Label Selection

Desired behavior in channel flow

1. Accurate log-law velocity profile.
2. Maintain equality $\langle u_1 u_1 \rangle + \langle u_2 u_2 \rangle + \langle u_3 u_3 \rangle = 2k$, $\langle u_\alpha u_\alpha \rangle \geq 0$.
3. Accurate anisotropy (b_{ij})

Label selection

- If select $\langle u_i u_j \rangle$ [1] → 1 is satisfied, 2-3 are violated
- If select b_{ij} [2] → 2&3 are satisfied, 1 is violated
- If select $\langle u_1 u_2 \rangle$, b_{11}, b_{22}, b_{33} → 1-3 are satisfied

[1]Geneva, N. and Zabaras, N., 2019. *Journal of Computational Physics*.
[2]Ling, J., Kurzawski, A. and Templeton, J., 2016. *Journal of Fluid Mechanics*.

Implementation

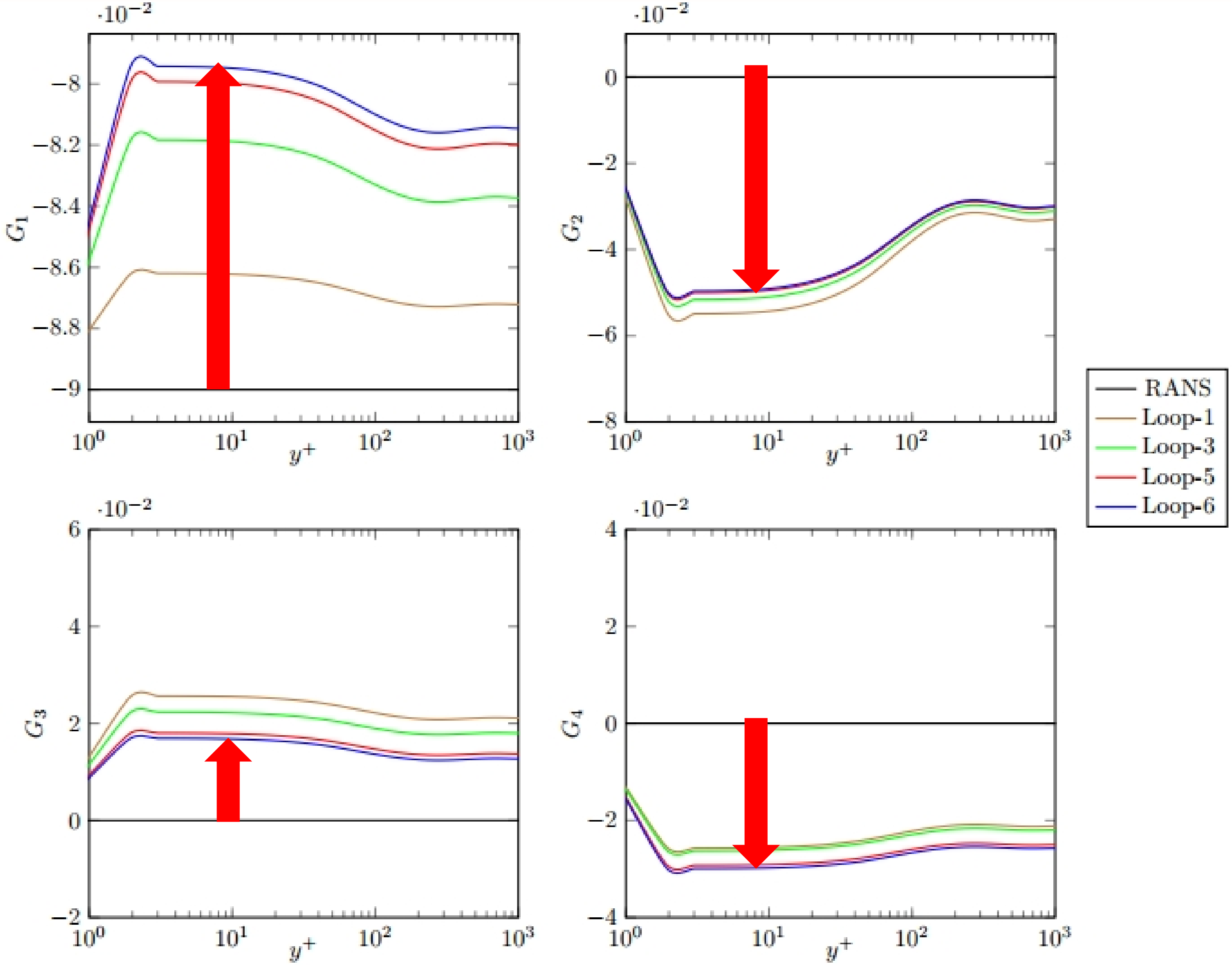
- TensorFlow is linked to OpenFOAM CFD code via C API
- **Labels** are true accurate quantities that we get form **DNS**, b_{ij}, $\langle u_i u_j \rangle$
- **Features** are the input parameters extracted from **RANS** simulations.
- Input features used for the 2D channel test case: $I_i = \left\{ tr(s^2), \frac{k}{\nu\omega} \right\}$
- Loss function definition:

$$MSE = \frac{1}{4N_{data}} \sum_{m=1}^{N_{data}} \left[\sum_{\alpha=1}^{3} \left(b_{\alpha\alpha}^{Predicted} - b_{\alpha\alpha}^{DNS} \right)^2 + \frac{1}{u_\tau^4} \left(\langle u_1 u_2 \rangle^{Predicted} - \langle u_1 u_2 \rangle^{DNS} \right)^2 \right]$$

- Optimized hyperparameters for neural network:

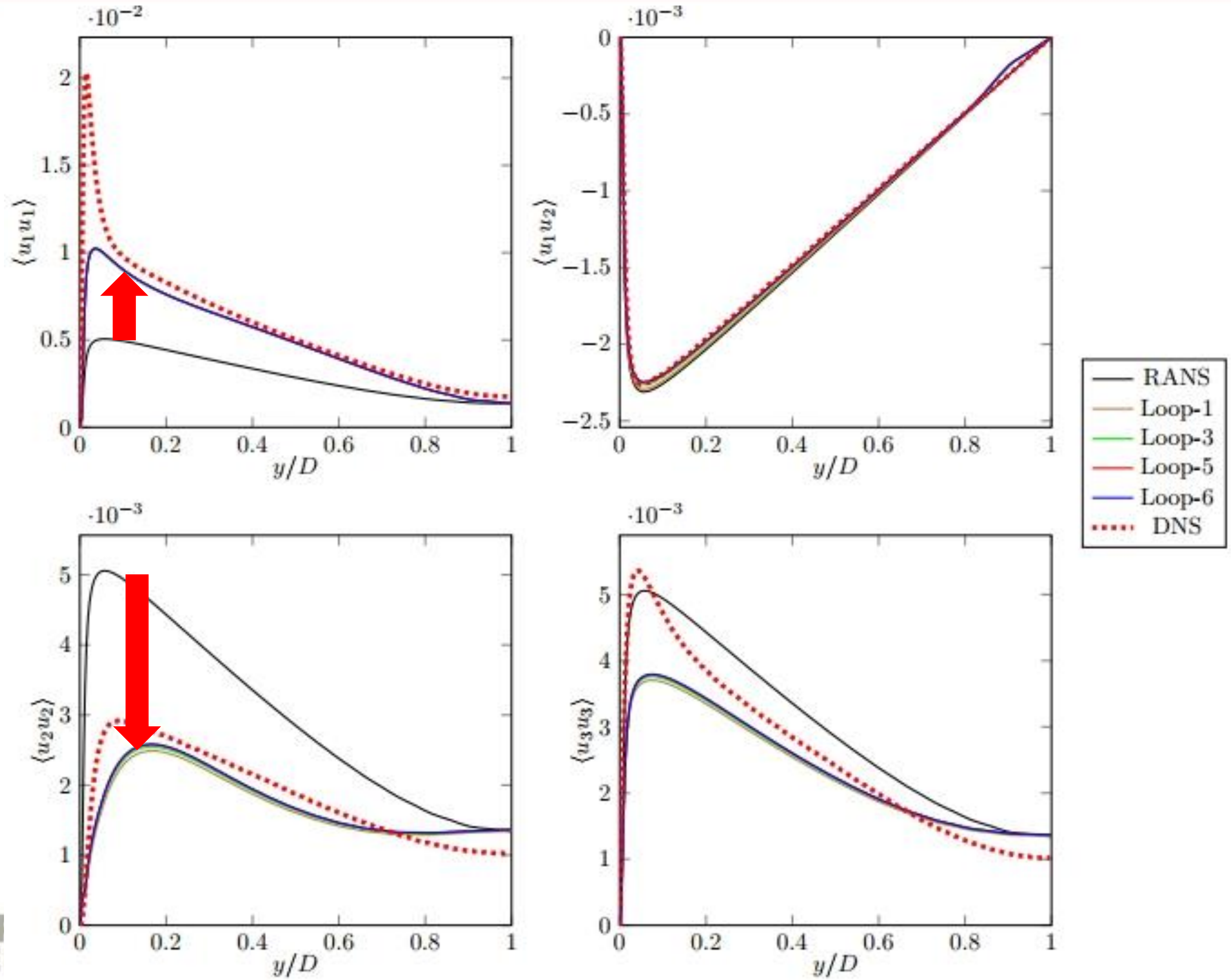
# layers	# nodes per layer	Activation function	Optimization Algorithm
3	3	Elu	Adam

Study-1



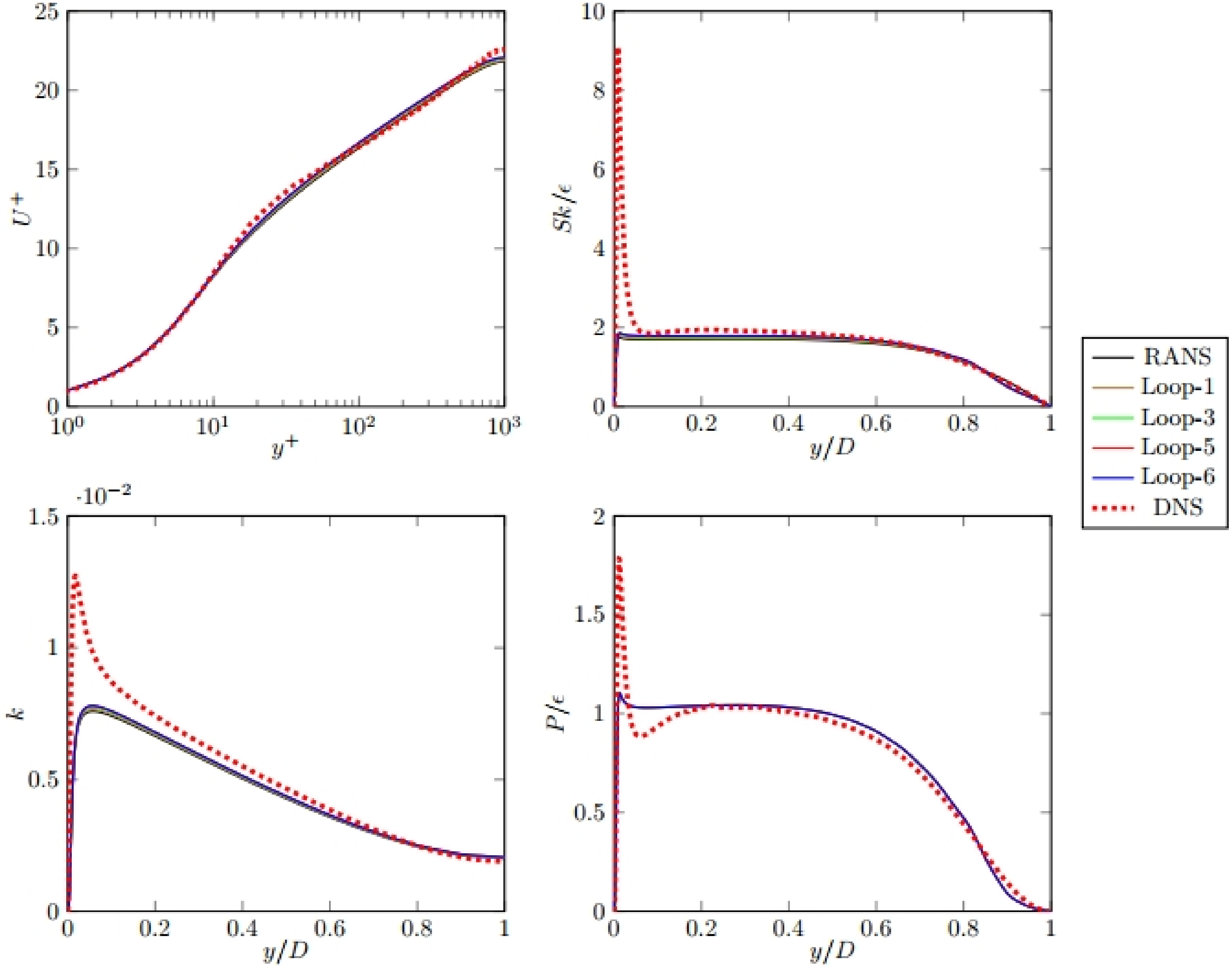
$\cdot 10^{-2}$
-8
-8.2
-8.4
-8.6
-8.8
-9
G_1
10^0
10^1
10^2
10^3
y^+
$\cdot 10^{-2}$
0
-2
-4
-6
-8
G_2
10^0
10^1
10^2
10^3
y^+
RANS
Loop-1
Loop-3
Loop-5
Loop-6
$\cdot 10^{-2}$
6
4
2
0
-2
G_3
10^0
10^1
10^2
10^3
y^+
$\cdot 10^{-2}$
4
2
0
-2
-4
G_4
10^0
10^1
10^2
10^3
y^+

Study-1



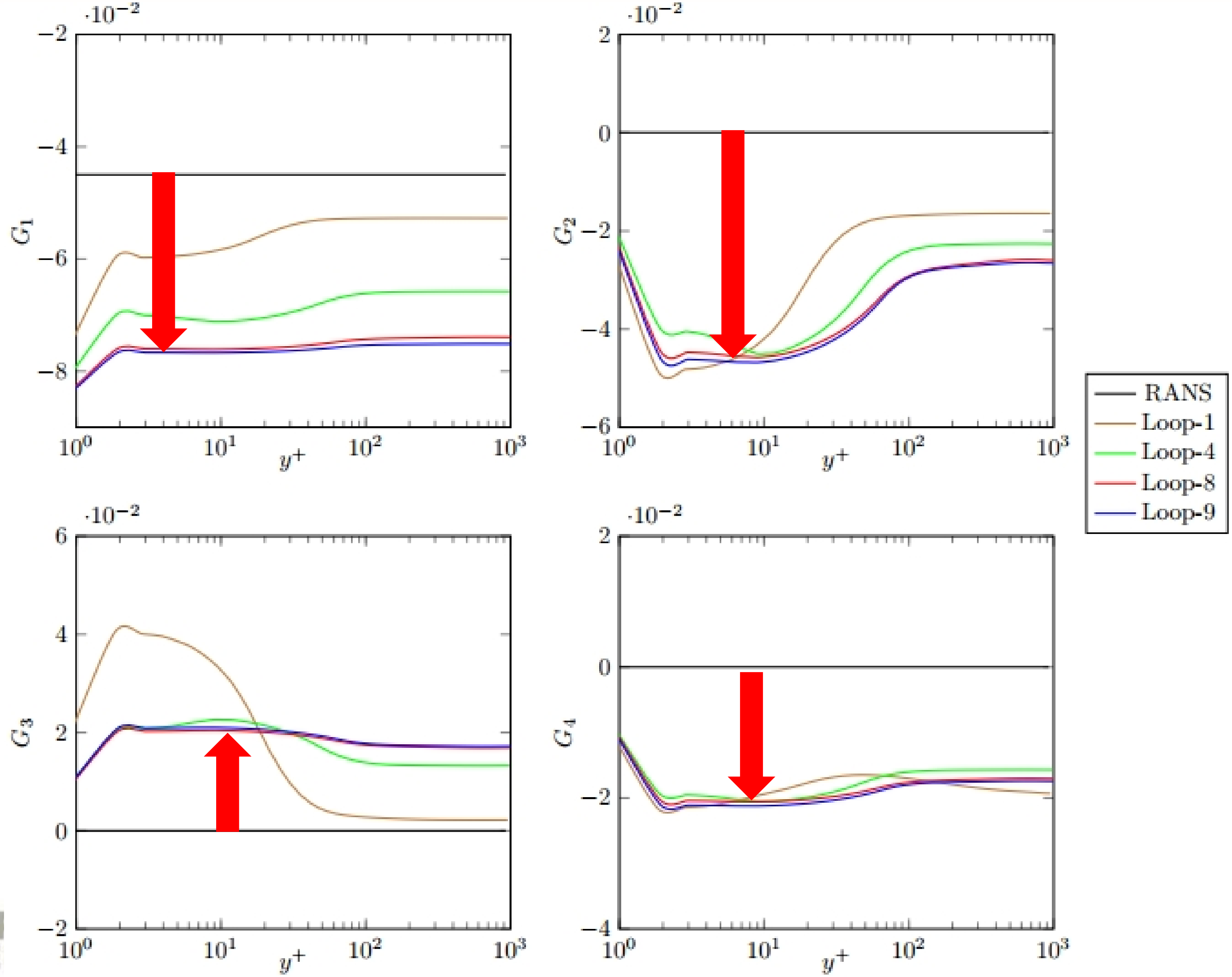
$\cdot 10^{-2}$
$\langle u_1 u_1 \rangle$
y/D
$\cdot 10^{-3}$
$\langle u_1 u_2 \rangle$
y/D
$\cdot 10^{-3}$
$\langle u_2 u_2 \rangle$
y/D
$\cdot 10^{-3}$
$\langle u_3 u_3 \rangle$
y/D
RANS
Loop-1
Loop-3
Loop-5
Loop-6
DNS

Study-1

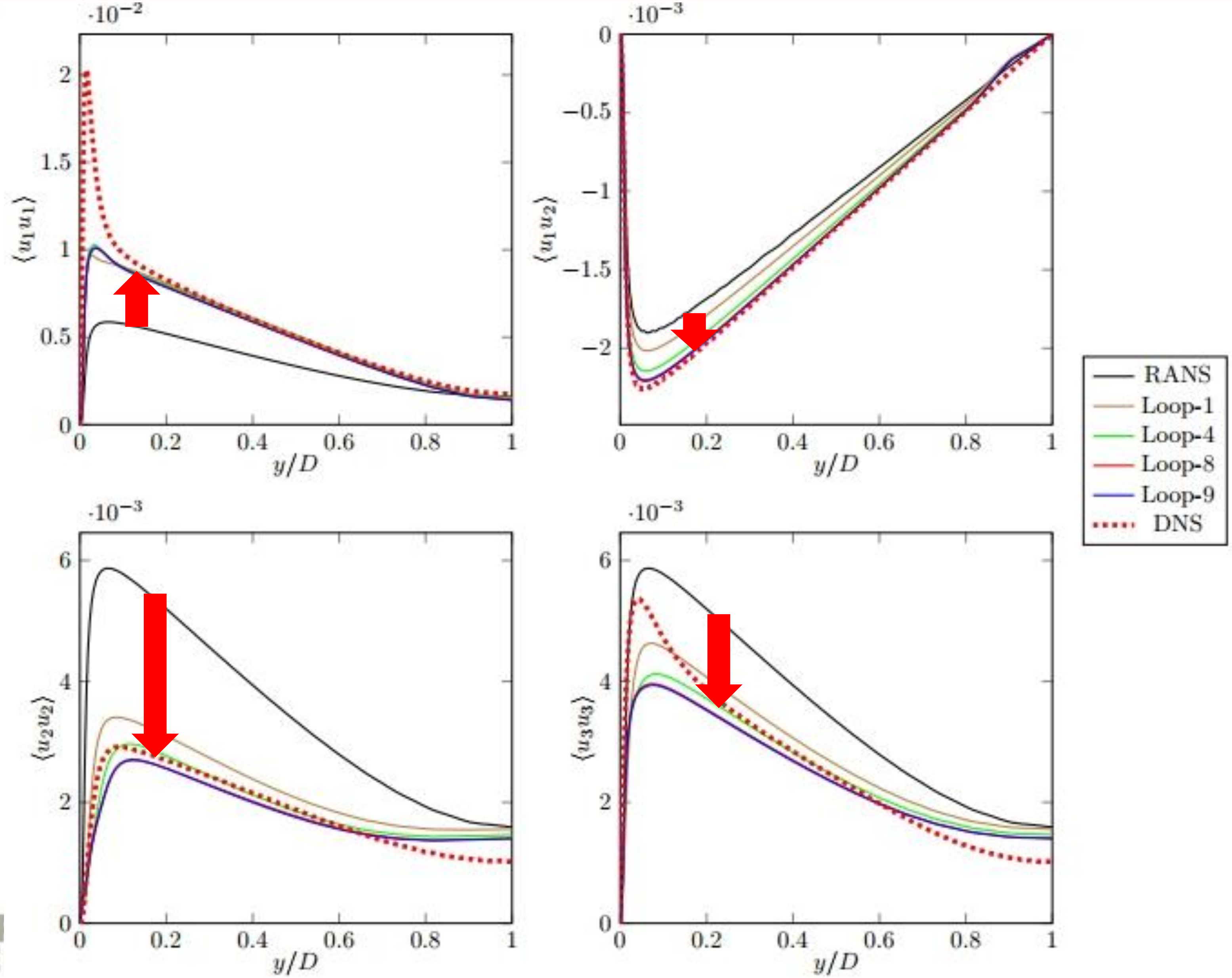


U^+
y^+
Sk/ϵ
y/D
$\cdot 10^{-2}$
k
y/D
P/ϵ
y/D
RANS
Loop-1
Loop-3
Loop-5
Loop-6
DNS

Study-2

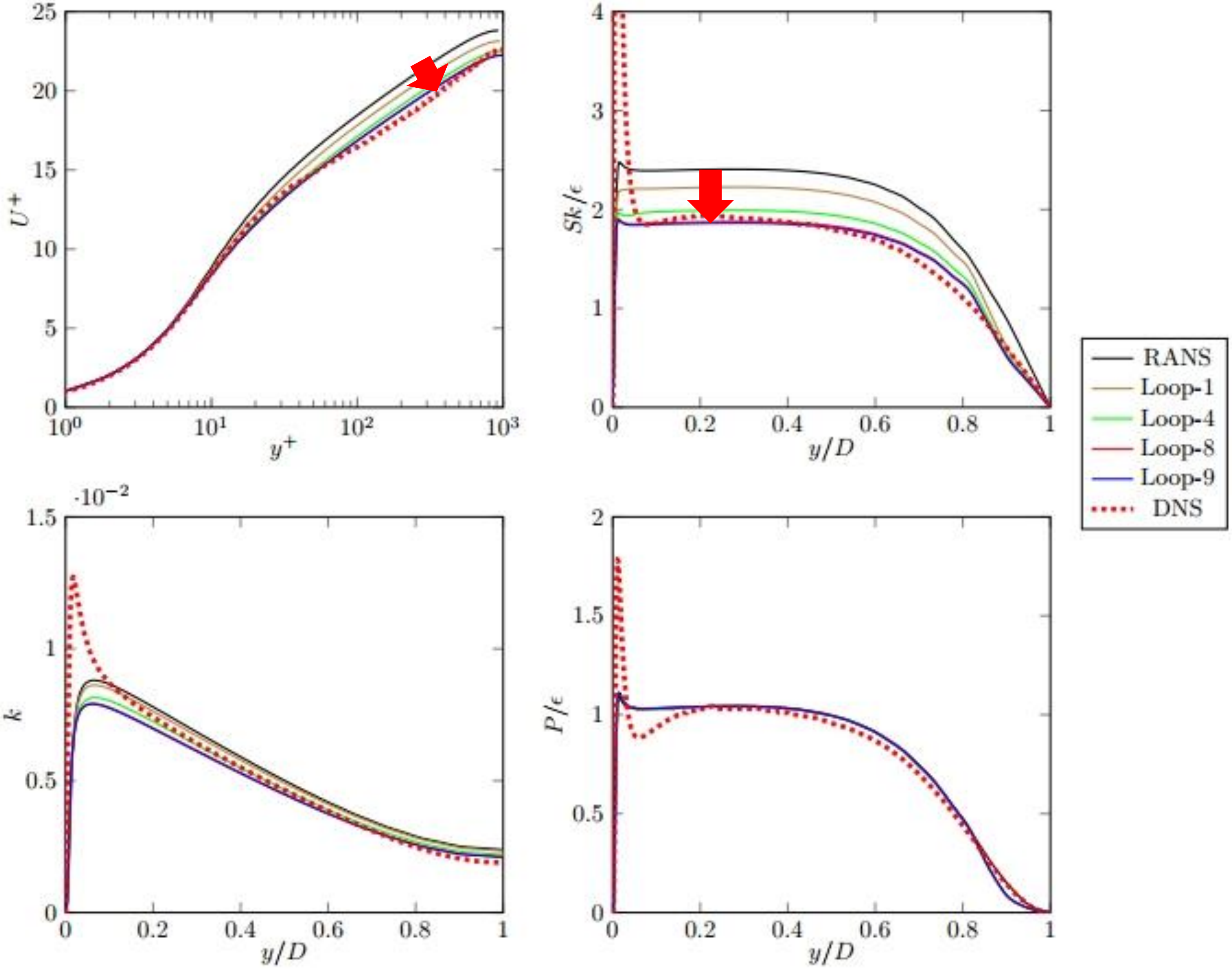


Study-2



$\cdot 10^{-2}$
$\langle u_1u_1 \rangle$
y/D
$\cdot 10^{-3}$
$\langle u_1u_2 \rangle$
y/D
$\cdot 10^{-3}$
$\langle u_2u_2 \rangle$
y/D
$\cdot 10^{-3}$
$\langle u_3u_3 \rangle$
y/D
RANS
Loop-1
Loop-4
Loop-8
Loop-9
DNS

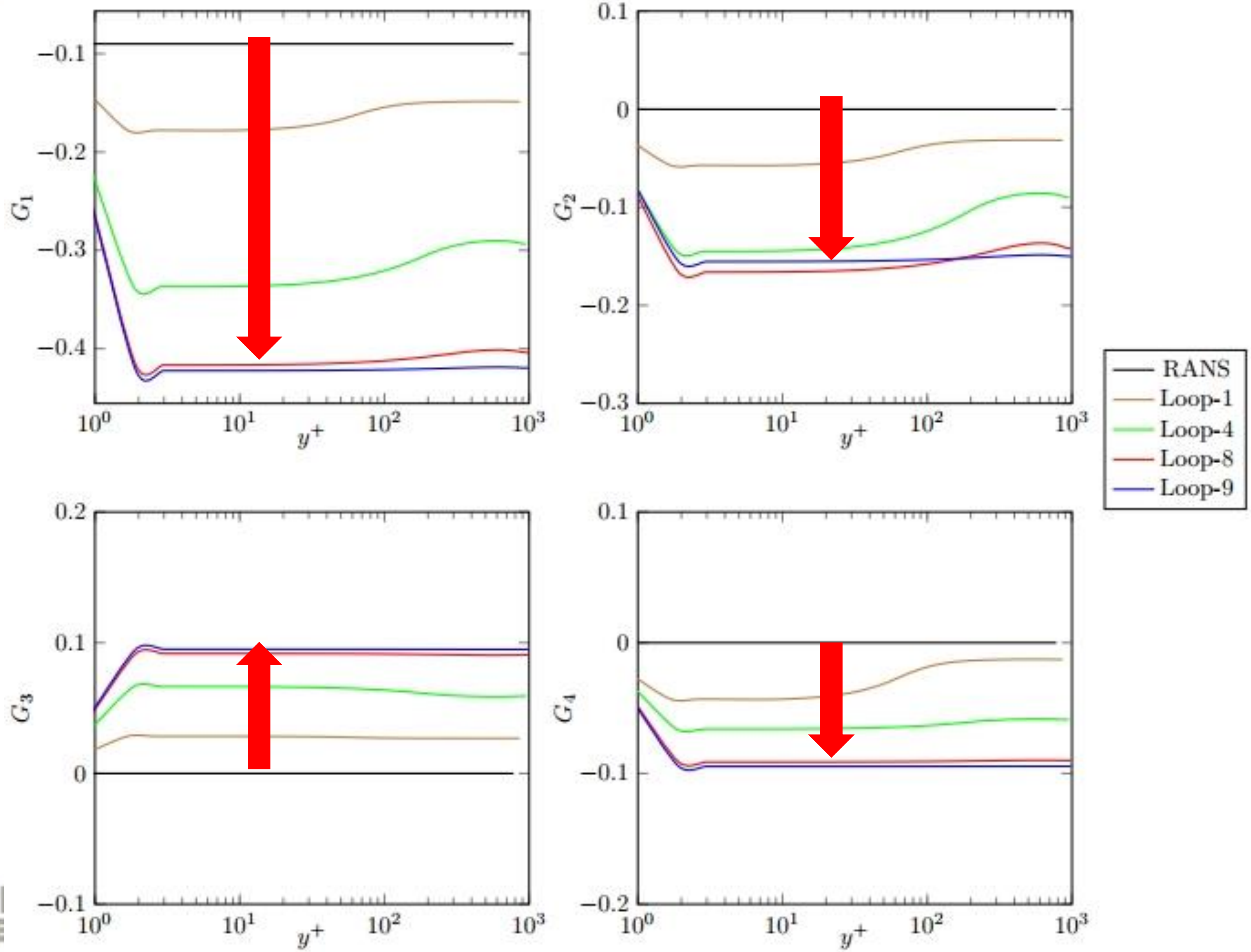
Study-2



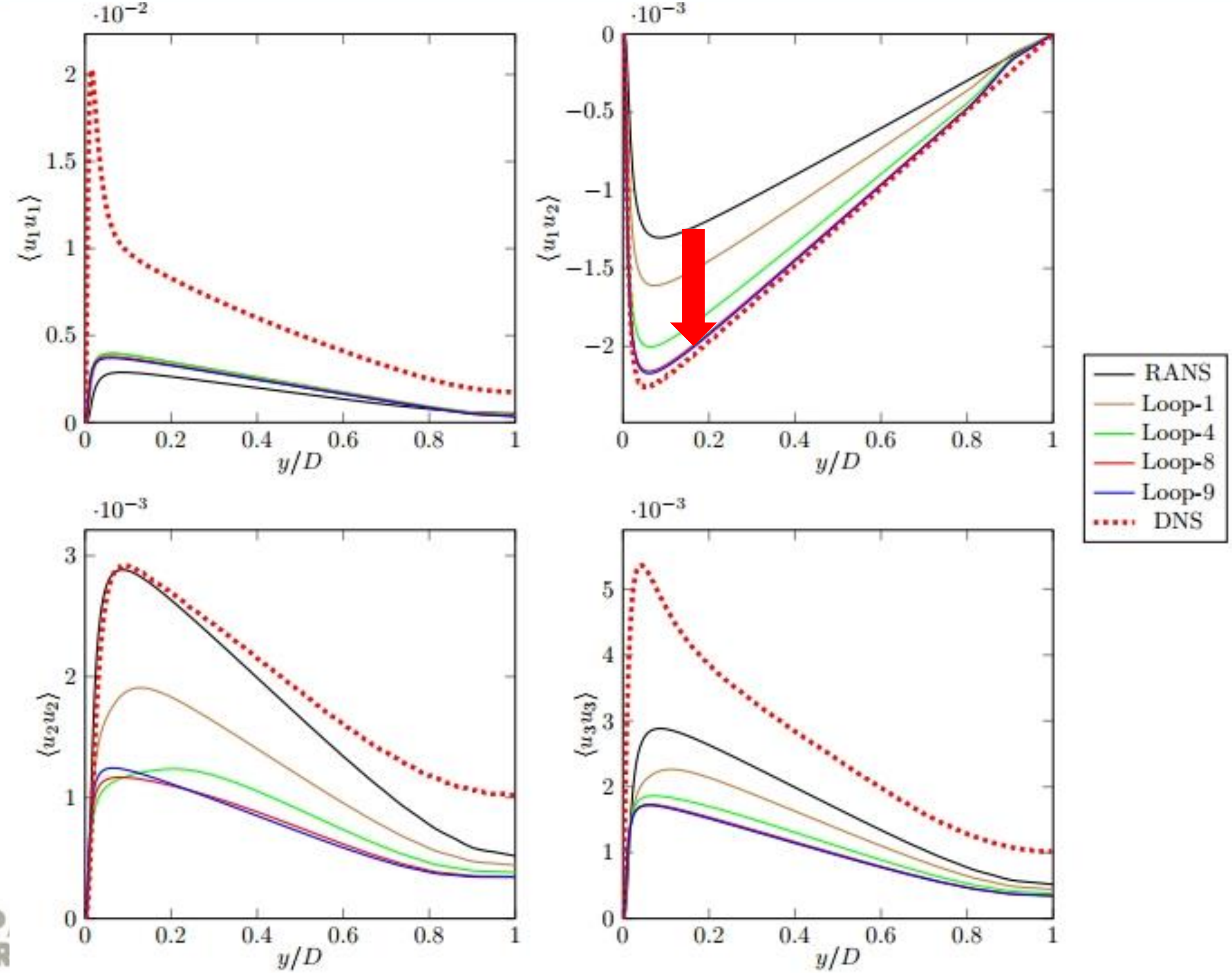
AEROS
ENGINE

A&M

Study-3

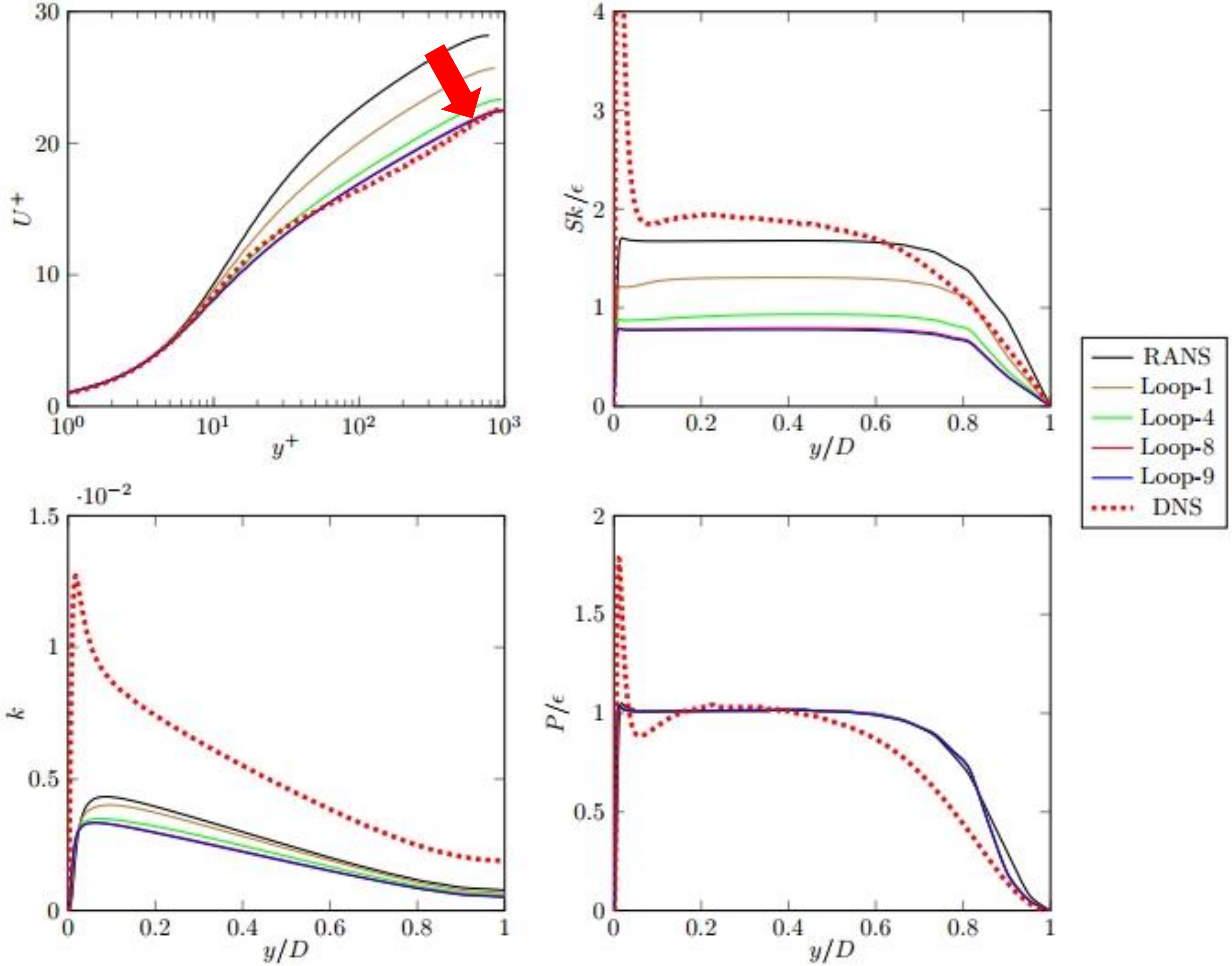


Study-3



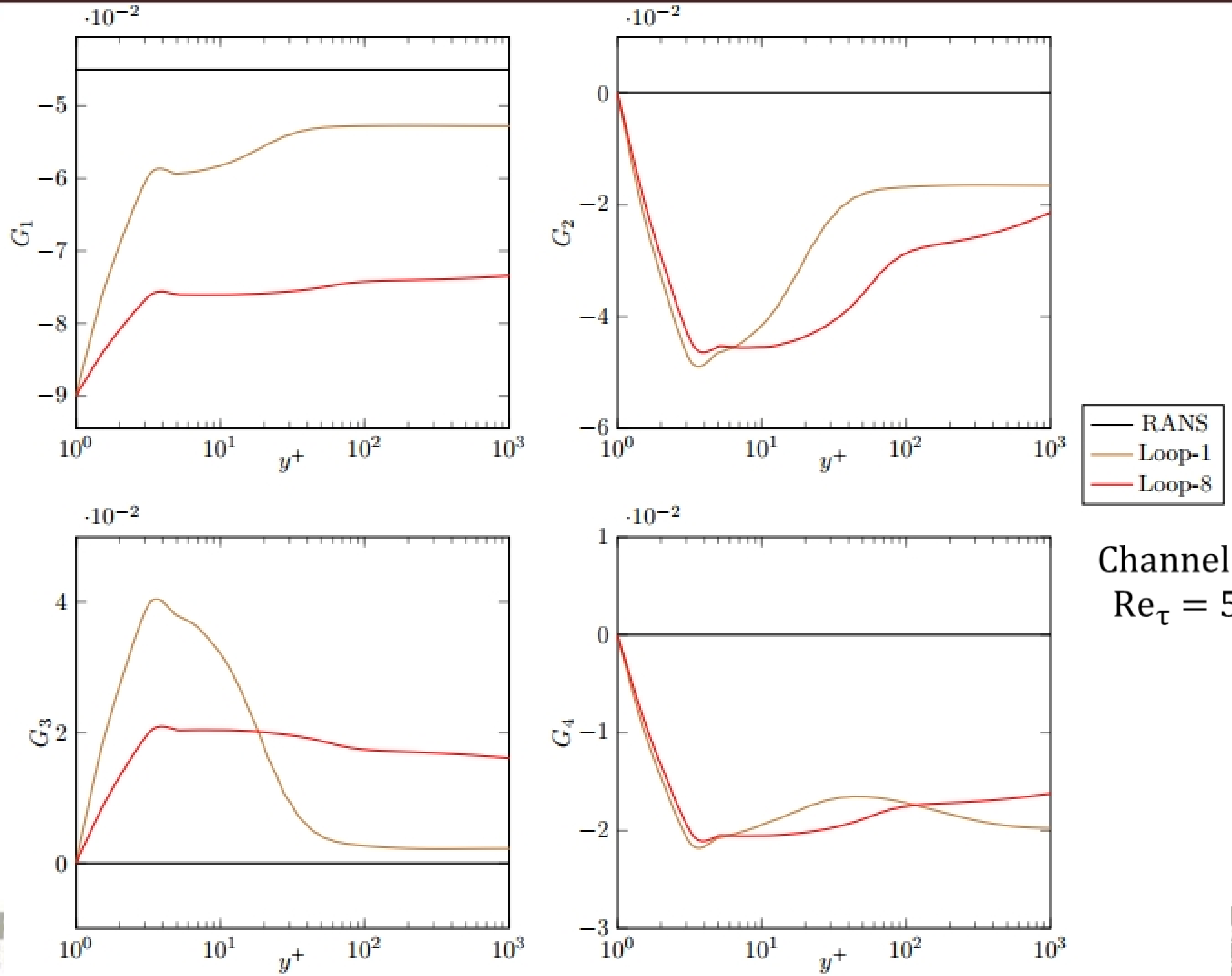
$\cdot 10^{-2}$
$\langle u_1 u_1 \rangle$
0
0.5
1
1.5
2
$\cdot 10^{-3}$
$\langle u_1 u_2 \rangle$
0
−0.5
−1
−1.5
−2
$\cdot 10^{-3}$
$\langle u_2 u_2 \rangle$
0
1
2
3
$\cdot 10^{-3}$
$\langle u_3 u_3 \rangle$
0
1
2
3
4
5
0
0.2
0.4
0.6
0.8
1
y/D
RANS
Loop-1
Loop-4
Loop-8
Loop-9
DNS

Study-3



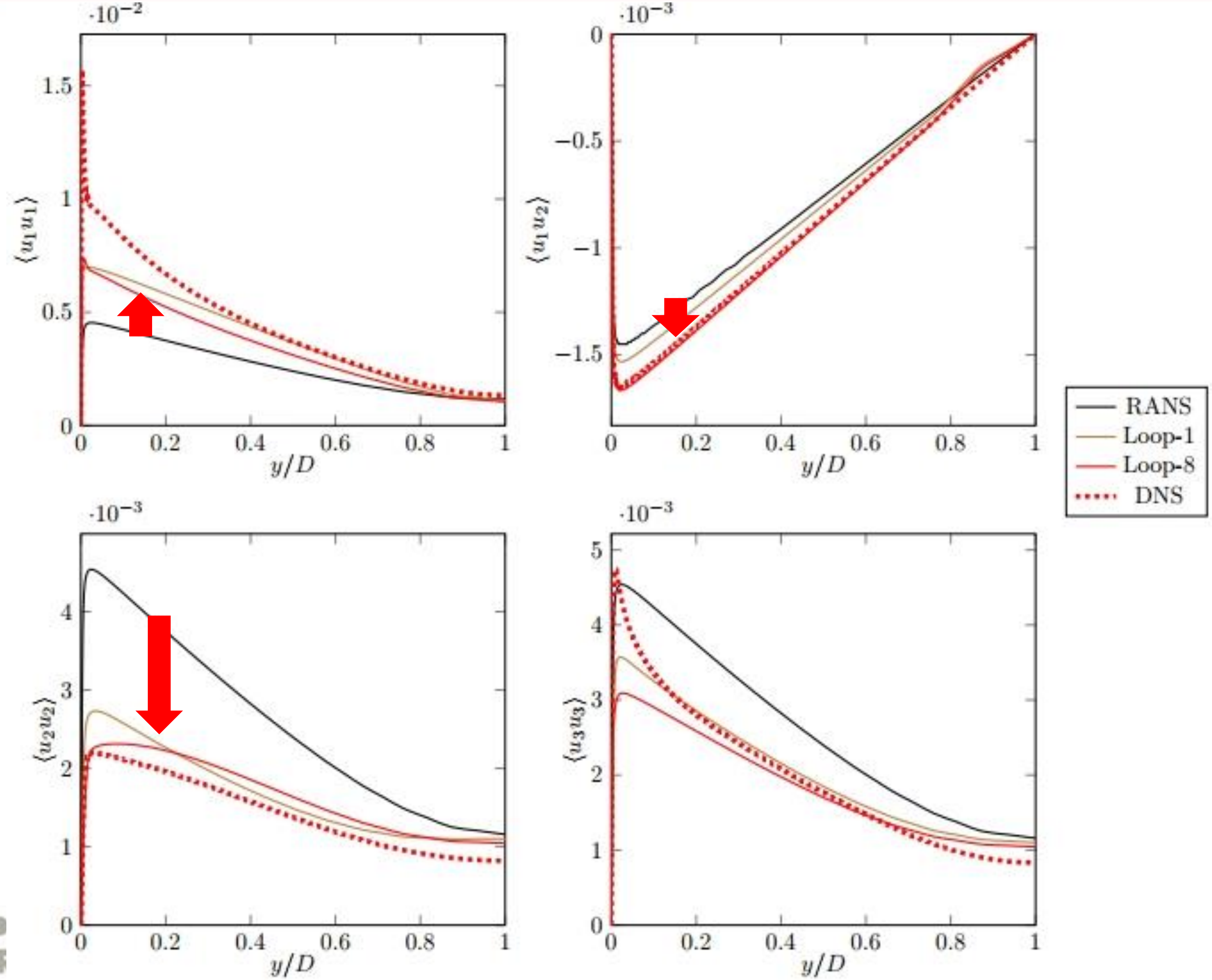
30
20
10
0
U^+
10^0
10^1
10^2
10^3
y^+
4
3
2
1
0
Sk/ϵ
0
0.2
0.4
0.6
0.8
1
y/D
RANS
Loop-1
Loop-4
Loop-8
Loop-9
DNS
$\cdot 10^{-2}$
1.5
1
0.5
0
k
0
0.2
0.4
0.6
0.8
1
y/D
2
1.5
1
0.5
0
P/ϵ
0
0.2
0.4
0.6
0.8
1
y/D

Study-2-CV



Channel flow
$Re_\tau = 5200$

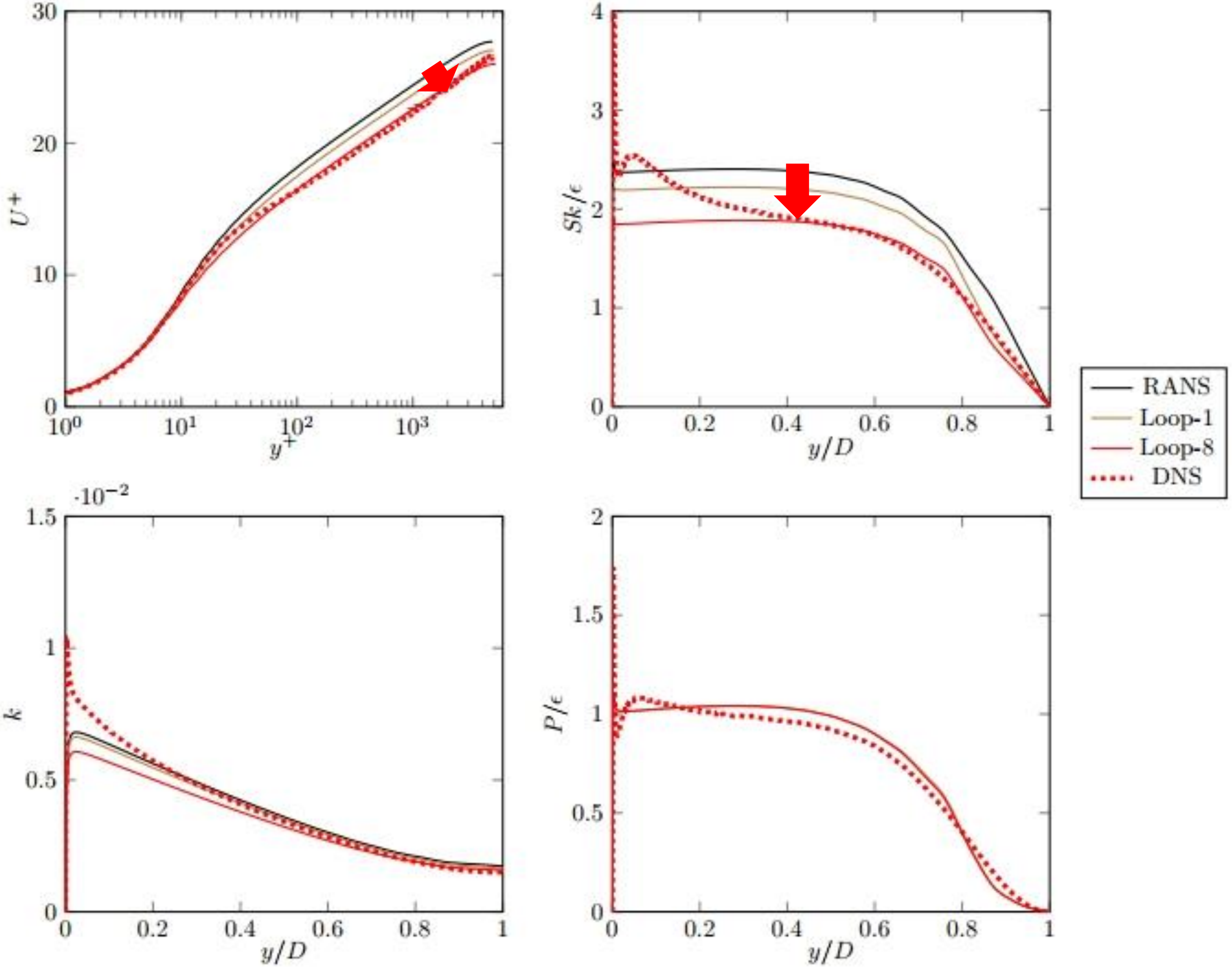
Study-2-CV



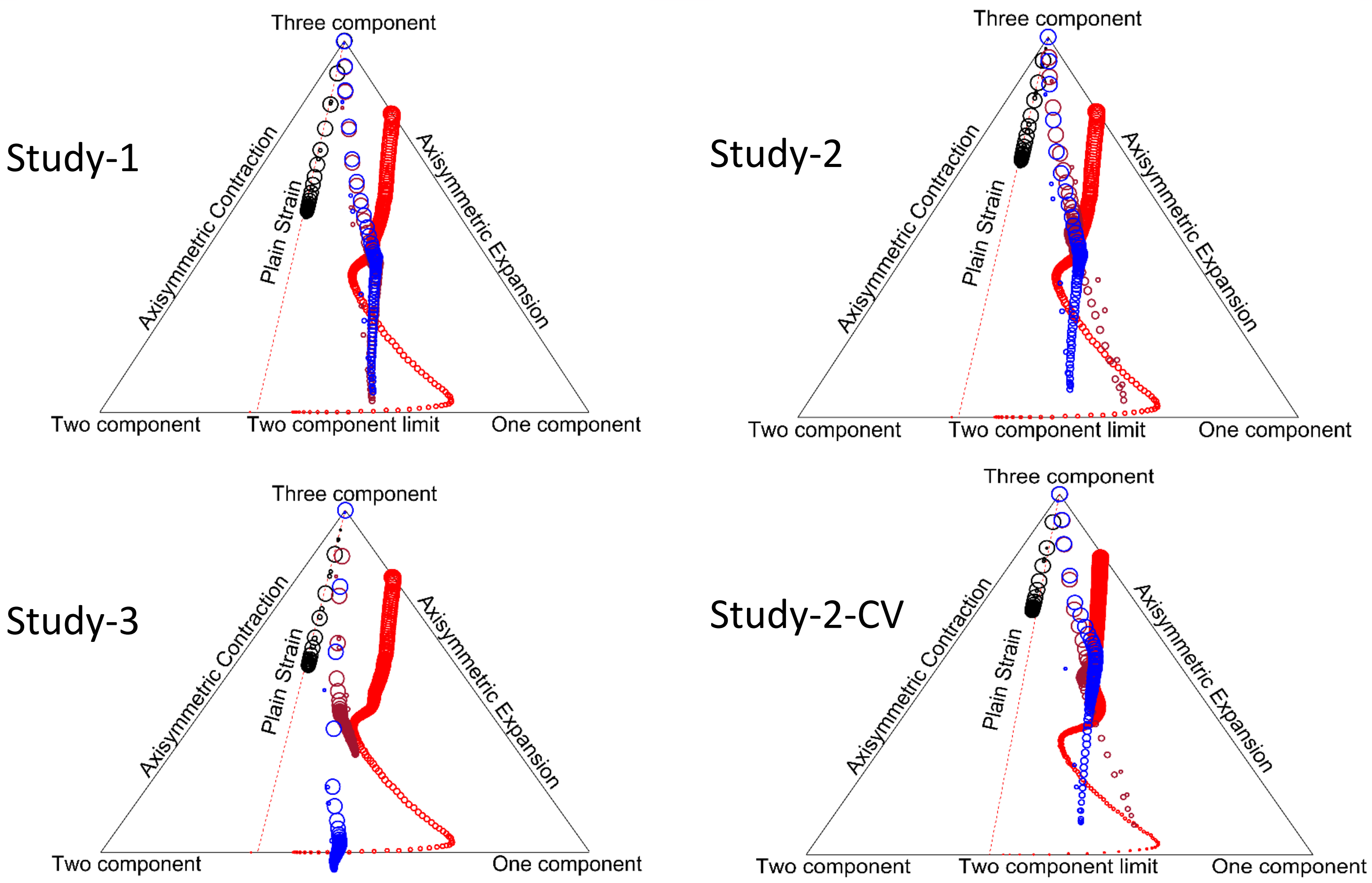
AEROSP
ENGINEER

A&M

Study-2-CV



Barycentric maps



Red: DNS, Black: RANS, Brown: Open loop, Blue: Closed loop
Distance from the wall is shown by size increase of symbols in plots.

Conclusions

- DDM/ML cannot make up all deficiencies in modeling
  - NN recovers from errors in G1, G2 and G3
  - NN cannot recover from errors in other coefficients
- Training practices and type of Neural Network have to standardized
- Physics-based modeling + DDM/ML can lead to improved predictive modeling
- Much more physics-based concepts are needed to correctly implement DDM/ML

Outcome of test study

- DDM/ML is reasonable for statistics included in objective function (OF)
- Statistics not included in objective function (OF)   – are worse than good `physics-based' model
- Challenge is to construct objective function (OF) and select Features that simultaneously optimizes:
  - *Mean flow, Reynolds stress, mean scalar, scalar variance, heat release, etc ?*
- Need physics-based analysis for construction objective functions and feature
  - **Need for physics merely takes a different form**

Summarizing Opinions

- **ML for RANS**
  - Can lead to improvements if be done right
  - Generalizability to flows with coherent structure????
  - Does not fully utilize the insight in data
- **ML for RSCM**
  - More generalizable as it uses more physics incumbent in data
  - Not popular yet
- **Stochastic Models**
  - Best suited for ML and highly generalizable in principle
  - Need to develop new turbulence physics vocabulary
- ***Perfectly suite for rarefied constitutive equations based on Chapman-Enskog analysis***

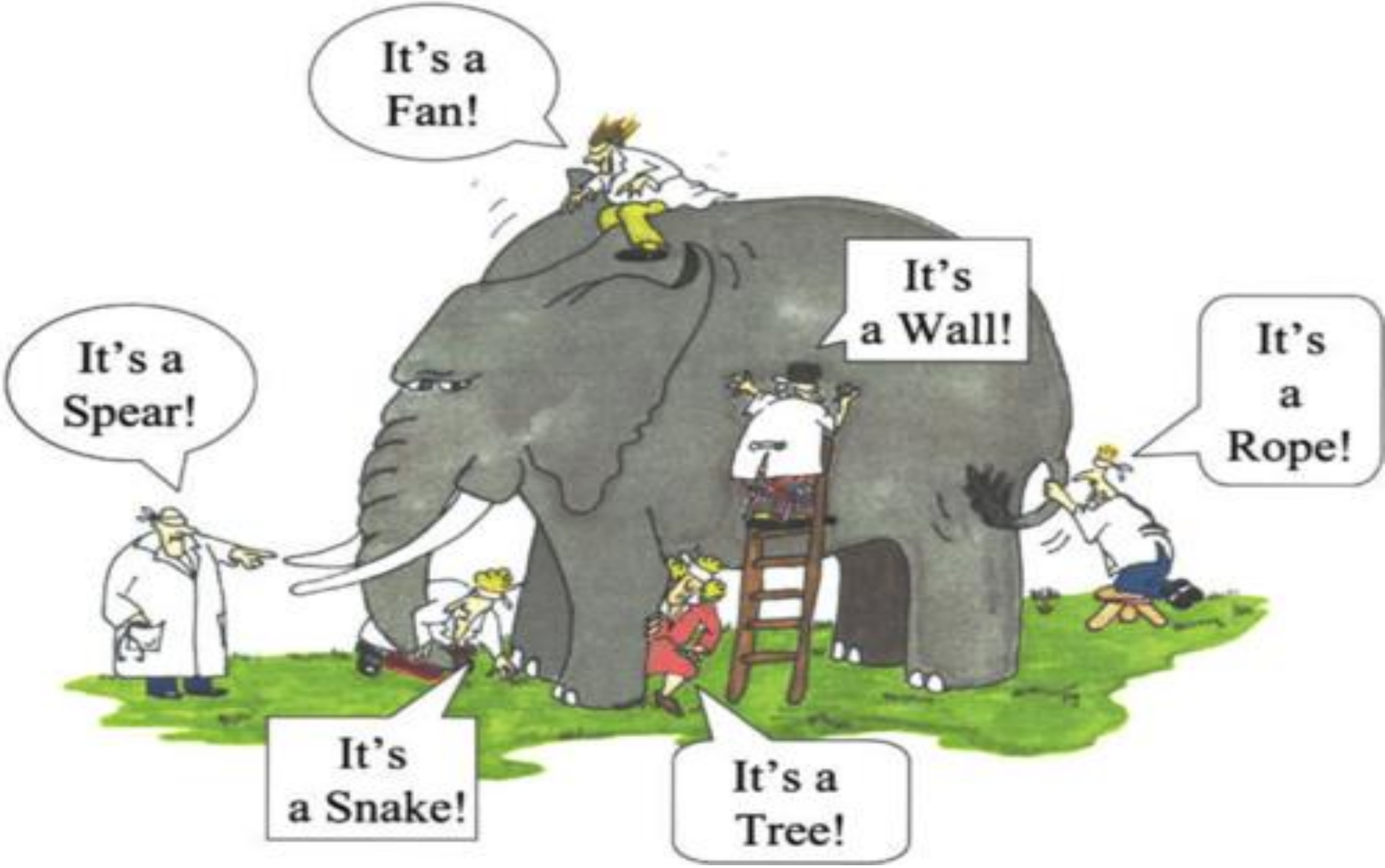
Parting Thoughts

- DDM/ML →a big hammer looking for a nail
- Turbulence modeling → Part Nail; Part Screw

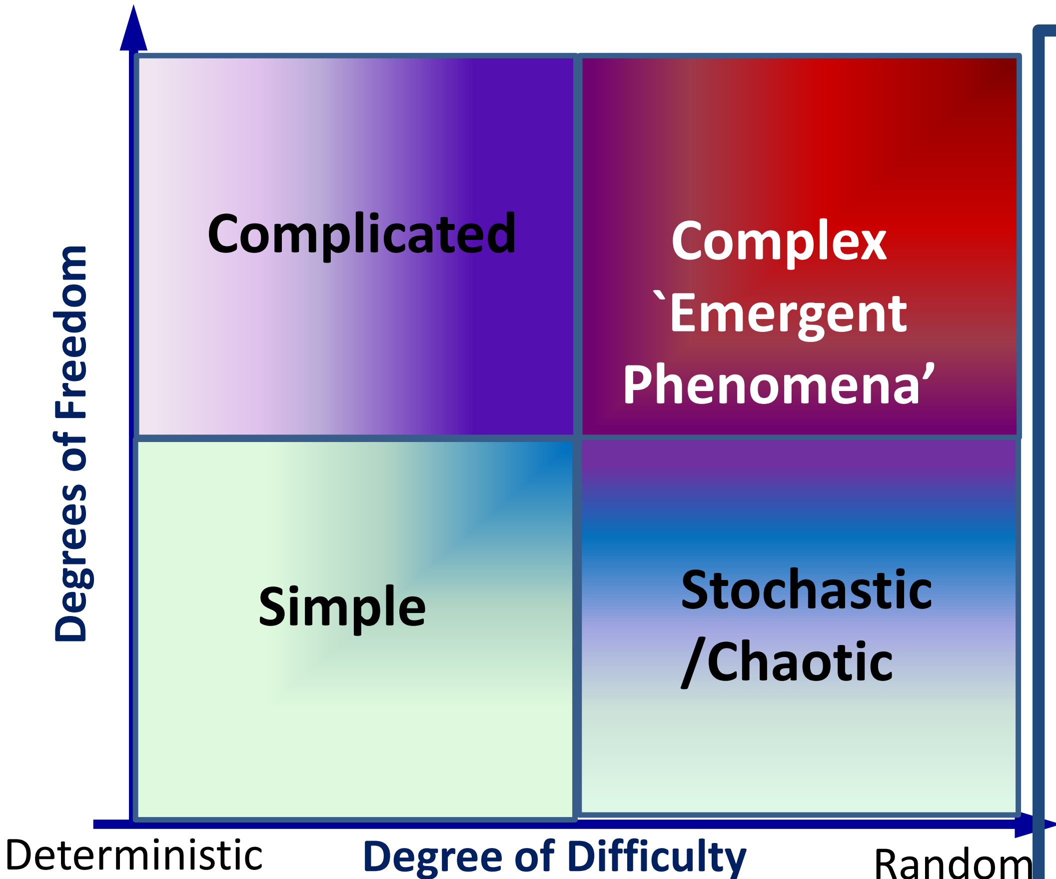
- Both DDM/ML (Hammer) and Physics-Based Methods (Screw-Driver) needed


Thank you
TEXAS A&M
UNIVERSITY

Traditional turbulence modeling



Turbulence: A complex dynamical system



Mathematical Approaches

1. Simple → Most present methods
2. Chaotic → Probabilistic and dynamical systems
3. Complicated system → DDM/ML appears to be well suited
4. Complex or `emergent phenomena' → Is DDM/ML adequate