Machine Learning, Scale Resolving Simulations and the Future of Predictive Computations of Engineering Flows:
A perspective

Sharath S. Girimaji
Texas A&M University

2022 Symposium on

Turbulence Modeling:

Roadblocks and the Potential for Machine Learning

NASA Langley Research Center, 27 – 29, 2022

Context of Talk

Even if ML for turbulence is **a** right thing to do, ***are we doing it right?***

- Current State of ML for turbulence modeling
  - **Instances of overselling, re-inventing the wheel, lack of physics awareness**
- But that is no reason to reject ML toolbox, instead USE IT RIGHT

Objectives of this work –

- Articulate questions many have about ML turbulence modeling
- Seeking an optimal path forward with physics awareness

Most discussion restricted to 2-Eqn RANS and SRS closures

Two-equation RANS Model

How many **closure coefficients** in a RANS model?

- 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):

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

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

All coefficients need to be compatible for optimal performance:

CCC: $G_1 \ldots G_{10}$

TCC: α, β, β^*, σ, σ^*

Is a constitutive relation always possible?

$$\frac{\partial \langle u_i u_j \rangle}{\partial t} + \langle U_k \rangle \frac{\partial \langle u_i u_j \rangle}{\partial x_k} = \mathrm{P}_{ij} - \varepsilon_{ij} + \Pi_{ij} + T_{ij} = \frac{d \langle u_i u_j \rangle}{dt}$$

In equilibrium turbulence:

$$\frac{d<u_i u_j>}{dt} = 0 \quad \rightarrow \quad < u_i u_j > = f\,(S_{ij}, W_{ij})$$

In other cases :

$$\frac{d<u_i u_j>}{dt} \neq 0 \quad \rightarrow \quad < u_i u_j > \neq f\,(S_{ij}, W_{ij})$$

In these cases, non-local space and time effects important

Questions 1 - 5

1. Why/when/where do traditional approaches fail?

2. How is turbulence different from other ML problems?

3. Are ML models truly generalizable? *Can ML extrapolate*?

4. Are current non-local ML-RANS approaches reasonable?

5. How much data is needed?

Questions 6 - 10

6. Is it okay to train ML with data from multiple flows?

7. Scale Resolution vs. ML-RANS model for complex flows?

8. Are current methods for ML-SRS modeling adequate?

9. What is minimum resolution required for a complex flow?

10. Optimal neural network architecture and parameters?

Q1: What is complex about turbulence?

Turbulence flow field →
Coherent Structures (Baby) + Stochastic field (Bath Water)

Stochastic equilibrium turbulence →
Constitutive eqn exists & unique

$$\frac{d < u_i u_j >}{dt} = 0$$

Stoch non-equil turbulence → memory effects → $\left\| \frac{d<u_i u_j>}{dt} \right\| > 0$

Constitutive possible but not be unique (memory & visco- elastic)

Steady coherent structures → non-local effects→ $\frac{d<u_i u_j>}{dt}$ is periodic

Local Constitutive Eq. may not exist (unknowable)

Transient coherent structures → NL + Memory → $\left\| \frac{d<u_i u_j>}{dt} \right\|$ is large

LOCAL CONSTITUTIVE EQUATION DOES NOT EXIST

Q2: How is turbulence different

What is different about turbulence closures?

- ML model part of a larger dynamical system with specified attractors
  - CCC and TCC must be compatible
  - Changing one as apart of ML and not others can lead to large errors
- Dynamical system must satisfy many **`Do No Harm'** constraints
  - Realizability, MFI, consistency with RDT, Log-law

Resolution:

Closed Loop ML training for RANS **(Taghizadeh et. al, NJOP, 2020)**

- Closed-loop training can improve consistency between high-fidelity data and the approximate RANS (reduced-order) model
- Additional constraints can be imposed during the looping process

Taghizadeh et. al, NJOP, 2020

Open-loop training & Computing

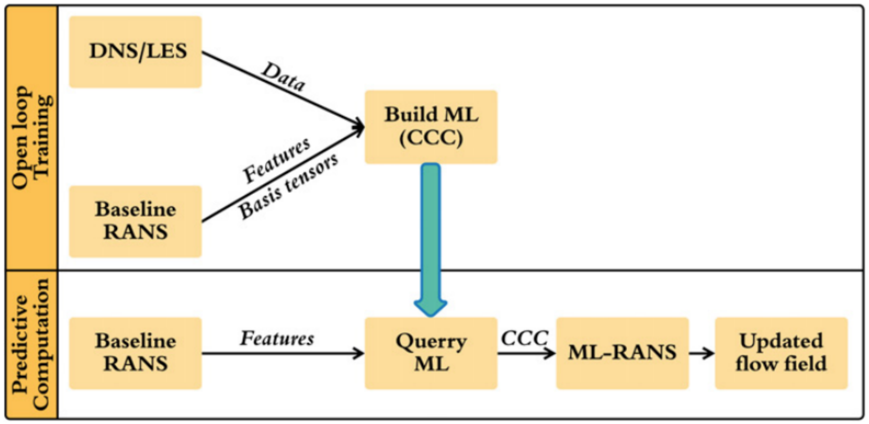
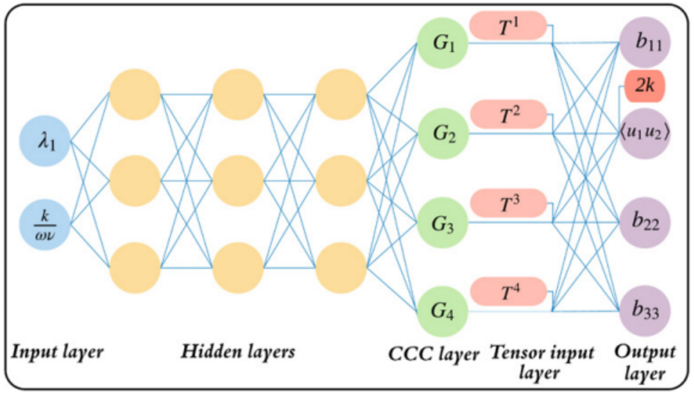


Figure 1. Open loop framework.



Closed-loop training & Computing

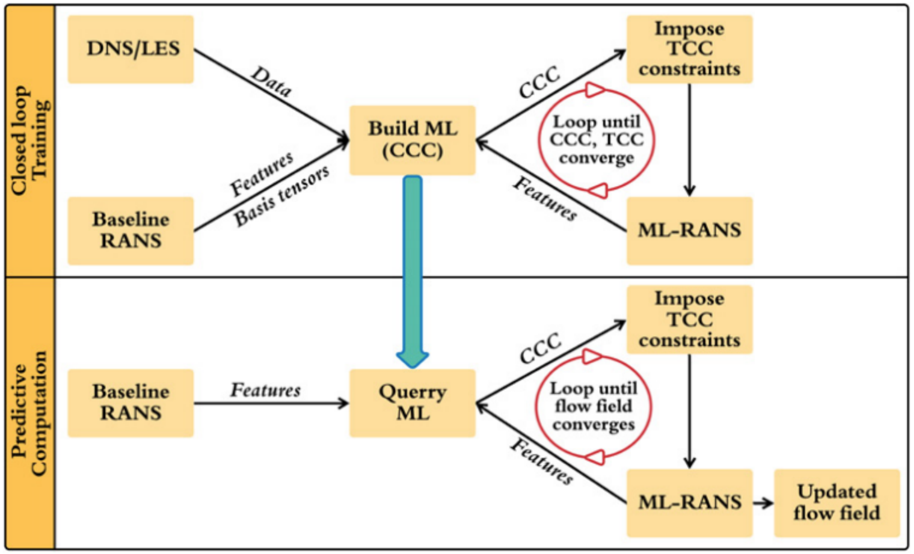


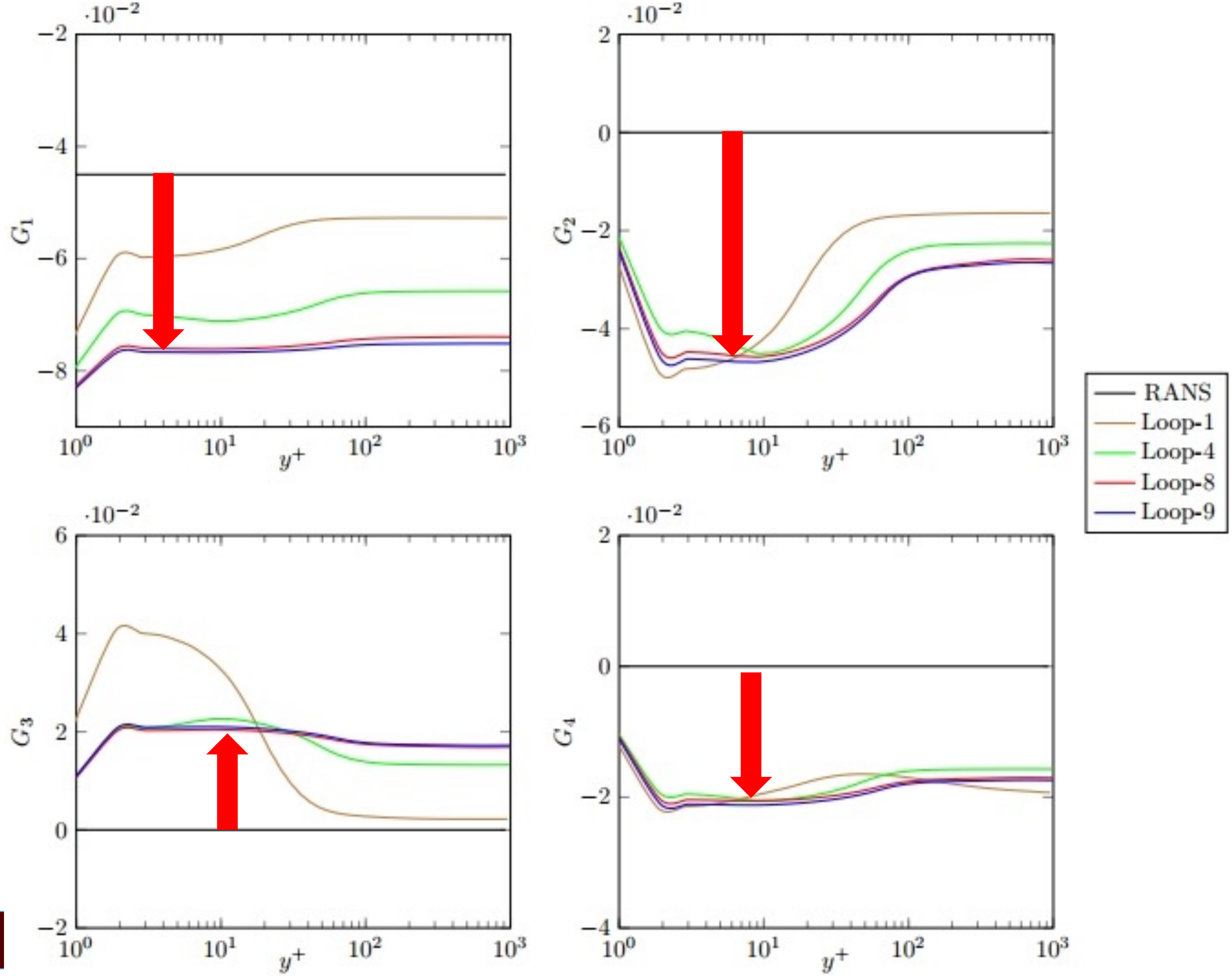
Figure 2. Closed loop framework.

Imposed TCC constraints:

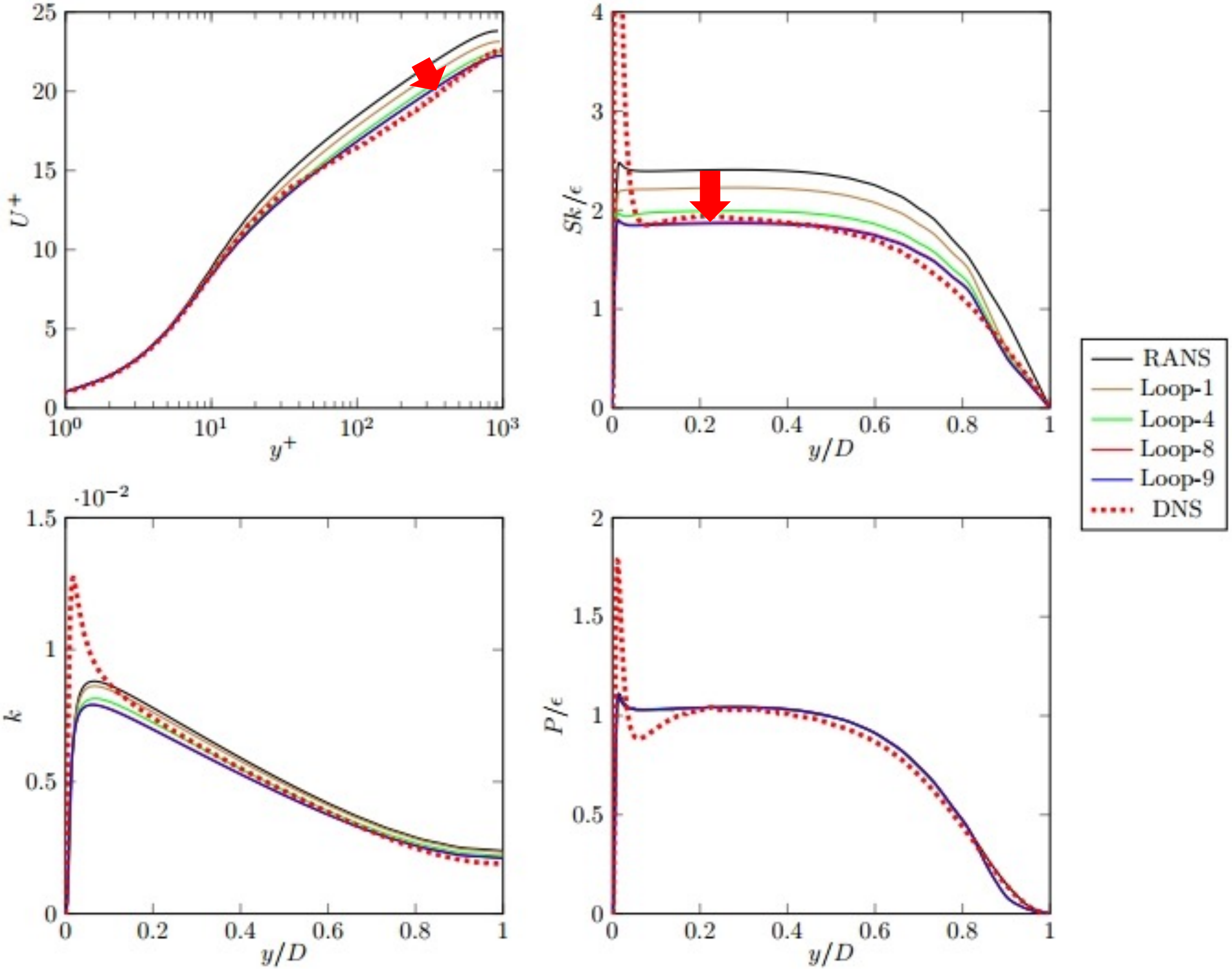
$$\sigma = \frac{\sqrt{-G_1}\left(\frac{\beta}{\beta^*} - \alpha\right)}{\kappa^2} \qquad \left(\frac{Sk}{\varepsilon}\right)^2 = \frac{\beta}{-G_1 \alpha \beta^*}$$

Do no harm constraints

Channel Flow Study:
Reset G values and see if they recover



Main Flow Variables



Q3: Is ML-RANS generalizable?

1. Turbulence statistics can exhibit strong bifurcations
2. Behavior in branches can be very different – growth vs decay
3. Can ML model trained in one branch capture behavior in another?

Test Proxy Physics Problem: ARSM cubic equation with bifurcations

$$G_1 = \begin{cases} \dfrac{L_1^0 L_2}{(L_1^0)^2 + 2\eta_2 (L_4)^2} & \text{for } \eta_1 = 0, \\ \dfrac{L_1^0 L_2}{(L_1^0)^2 - \frac{2}{3}\eta_1 (L_3)^2 + 2\eta_2 (L_4)^2} & \text{for } L_1^1 = 0, \\ -\dfrac{p}{3} + \left(-\dfrac{b}{2} + \sqrt{D}\right)^{\frac{1}{3}} + \left(-\dfrac{b}{2} - \sqrt{D}\right)^{\frac{1}{3}} & \text{for } D > 0, \\ -\dfrac{p}{3} + 2\sqrt{\dfrac{-a}{3}} \cos\left(\dfrac{\theta}{3}\right) & \text{for } D < 0, b < 0, \\ -\dfrac{p}{3} + 2\sqrt{\dfrac{-a}{3}} \cos\left(\dfrac{\theta}{3} + \dfrac{2\pi}{3}\right) & \text{for } D < 0, b > 0. \end{cases}$$

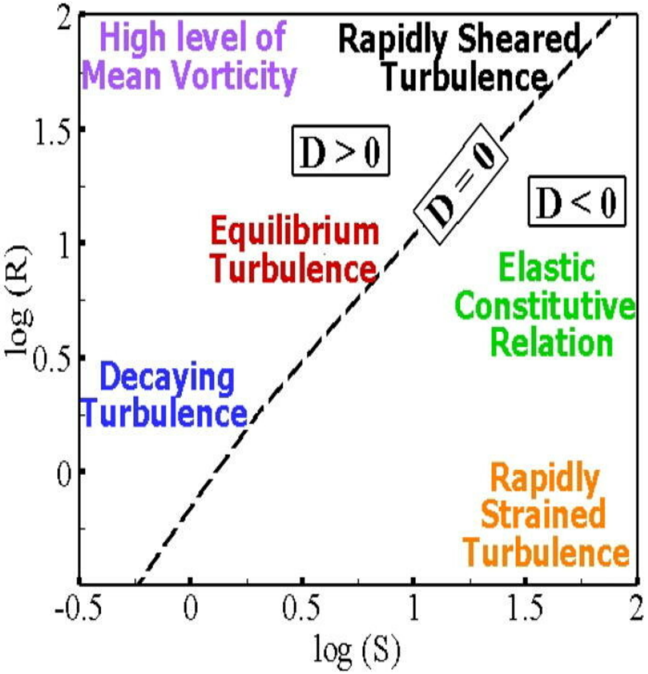
$$D = \frac{b^2}{4} + \frac{a^3}{27}$$

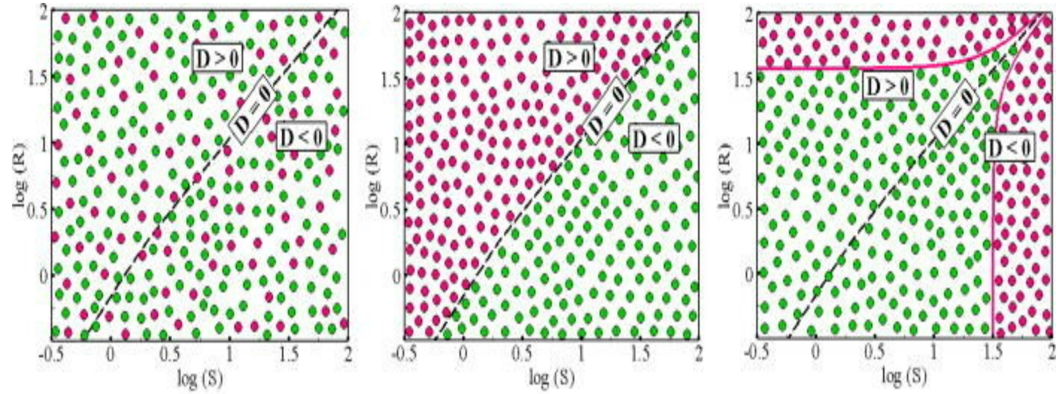
$$a = \left(q - \frac{p^2}{3}\right), \quad b = \frac{1}{27}(2p^3 - 9pq + 27r), \quad p = -\frac{2L_1^0}{\eta_1 L_1^1},$$

$$q = \frac{1}{(\eta_1 L_1^1)^2}\left[(L_1^0)^2 + \eta_1 L_1^1 L_2 - \frac{2}{3}\eta_1 (L_3)^2 + 2\eta_2 (L_4)^2\right],$$

$$r = -\frac{L_1^0 L_2}{(\eta_1 L_1^1)^2}, \quad \cos(\theta) = \frac{-b/2}{\sqrt{-a^3/27}}.$$

Training (green dots) and testing (red dots)





(a) (b) (c)

1. Trained & tested over entire domain → Excellent agreement
2. Trained in one & tested in another branch → Substantial accuracy reduction
3. Incomplete training on both sides → Substantial error in RDT regime

Conclusion: Extrapolation can be fatally inaccurate

Salar Taghizadeh; Freddie D. Witherden; Yassin A. Hassan; Sharath S. Girimaji;
Physics of Fluids **33,** 115132 (2021)

Q4: How much data do we need for a know-it-all model?

Data requirements vary significantly with locality of the flow/model:

Local models → Stress at a point depends only on the local strain field

- Small stencil size, fewer parameters to tune
- ML cannot extrapolate reliably → data needed from all bifurcation branches
- **Even for homogeneous 2D mean flows, this is a tall order**

Non-Local models → Stress at a point depends on strain field over a large domain

- Large stencil size, large number of parameters to tune, need significantly more data
- Large quantities of data from each structure type
- Many coherent structure types, strongly dependent upon flow geometry
- **Unbounded set of coherent structures → Unbounded need for training data**
- **For transient coherent structures → Need time label (dependence) as well**

Q5: Training ML models over different coherent structures?

Works in literature develop models and train over multiple coherent structures:

- Each coherent structure has a different domain of influence
- Even different locations with a coherent structure can have vastly different physics
- For same local strain rate, stress can be vastly different depending on the neighbors
- Training local model over different non-local effects will compromise the model
- Need to introduce extra features to distinguish between different flows
- But, extra features will add significantly to training efforts

Q6: Are current non-local ML models adequate?

Many non-local model still start with the following form

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

This *b (s, w)* is incomplete for non-local effects

1. ***I* & *T*** must include 2-point statistics to capture all non-local effects
2. List of all two-point scalars and tensors must be determined from representation theory
   - The list can be tediously long
   - Determining large number of G's may not be optimal

- **Much more details to be worked out**
- **Success unclear even after all tensors are included**

Q7: Non-local modeling vs. scale resolution

Non-local ML Issues:

1. Data generation can be expensive and incomplete.
2. Large upfront cost.
3. At the very end, accuracy is highly debatable

SRS Issues:

1. Low upfront cost but significant *in situ* cost
2. Computing capacity continues to grow and get cheaper
3. Accuracy of the all-important large scales reasonably guaranteed

Conclusions:

1. It is preferable to do perform scale resolving simulations.
2. How to judiciously combine the strengths of ML and SRS?

Q8: What is lowest scale resolution allowed?

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

1. How to determine the optimum degree of resolution in an unseen flows? Can we tell the baby from the bath water?
2. Markers of coherent structures and transient effects:
   - SK/ε → Resolved-to-unresolved strain rate ratio
   - P/ε → Production-to-dissipation ratio
   - F_c → Coherent-to-total kinetic energy ratio
3. Can a RANS calculation indicate latent coherent structures?

Q9: How to improve ML-SRS training?

- High-fidelity data contain rich unsteady information
- Yet, in most training we average over realizations and lose the texture of turbulence

Challenges:

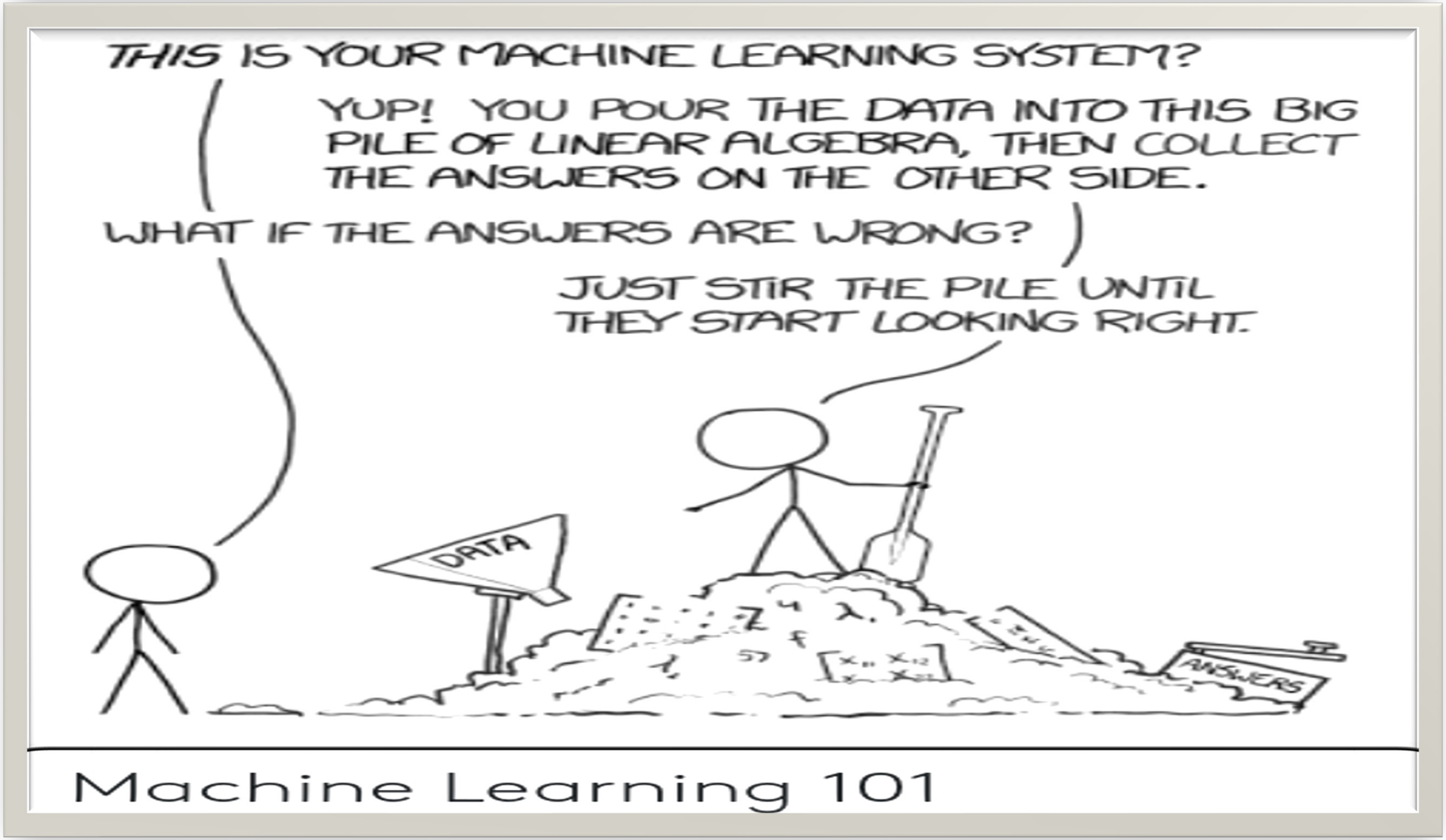
1. How to curate hi-fi data for different filter levels?

- Separate baby from bath water
- Throw away ALL the bathwater but not baby (Occam's Razor)

2. Find a way to incorporate all filtered unsteadiness into ML-SRS

Q10: Optimal network architecture and hyper-parameters

Need `best practice' so we do not have to resort to this



Parting Thoughts

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

- We need hammer & screw-driver in our tool kit

Thank you

TEXAS A&M UNIVERSITY