

Advanced Modeling and simulation Seminar Series

NASA Ames Research Center, December 8th, 2022

Bayesian machine learning for data-enhanced CFD

Paola CINNELLA (paola.cinnella@sorbonne-universite.fr)

SCIENCES SORBONNE UNIVERSITÉ

∂'Alembert Institut Jean le Rond d'Alembert

Overview

- Promises and challenges of machine learning for CFD
- Bayesian machine learning
- Application to turbulence modeling
- Conclusions and outlook

Motivation

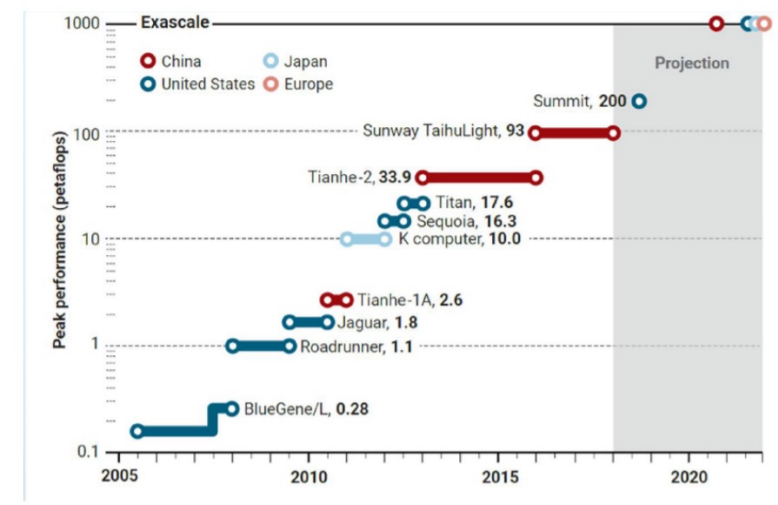
- Environmental constraints require drastically reduced energy consumption and carbon emissions
- Increasing pressure for breakthrough designs
- Intensive use of Computational Fluid Dynamics (CFD)



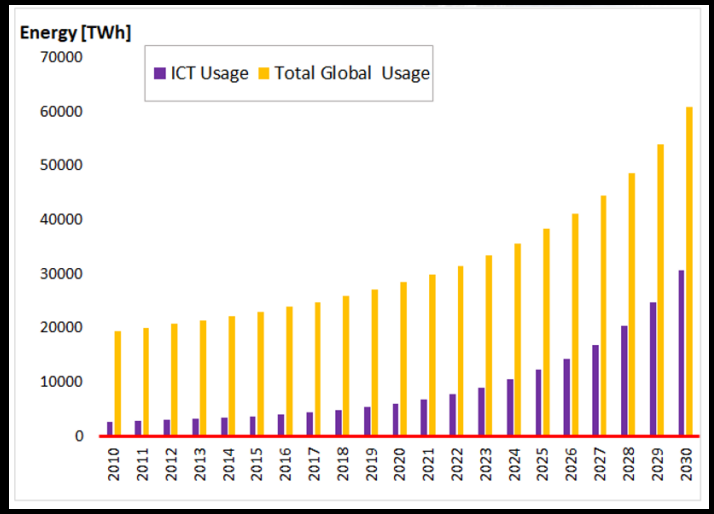


Motivation

- Very detailed, high-fidelity CFD models require HPC
- Progress made, but still unfeasible for routine Engineering use
- Power consumption of supercomputers O(10) Megawatts
- HPC facilities not easily accessible to SME



Top Supercomputing Platforms (2019)



Worst-case forecast for ICT Energy Consumption

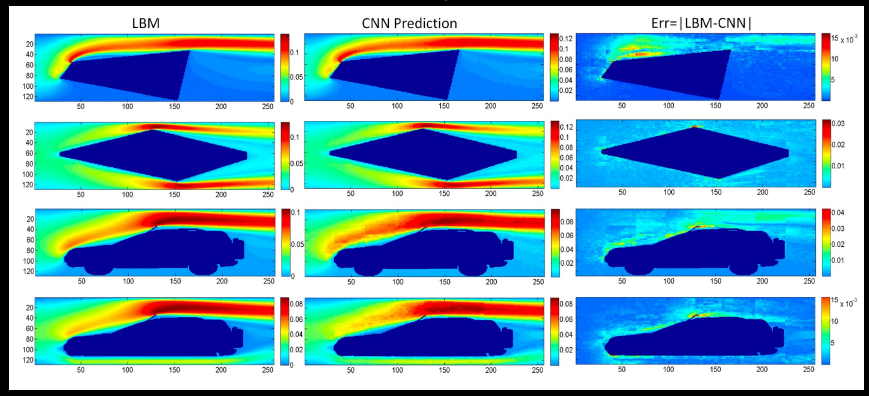
Rank	System	Power (MW)
1	**Frontier** DOE/SC/Oak Ridge National Laboratory United States	21,1
2	**Supercomputer Fugaku** RIKEN Center for Computational Science Japan	29,9
3	**LUMI** EuroHPC/CSC Finland	6,0
4	**Leonardo** EuroHPC/CINECA Italy	5,6
5	**Summit** DOE/SC/Oak Ridge National Laboratory United States	10,1

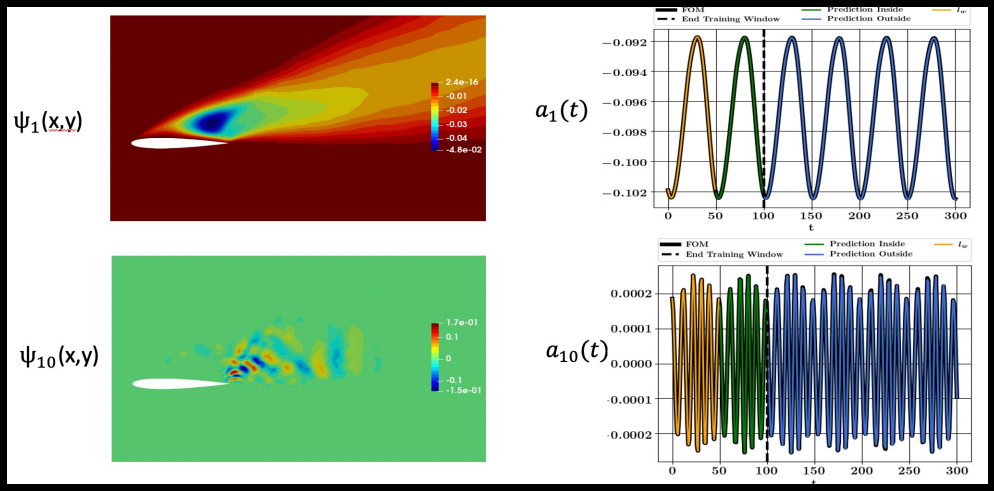
Top500.org

HiFi-quality CFD at the cost of LoFi or less

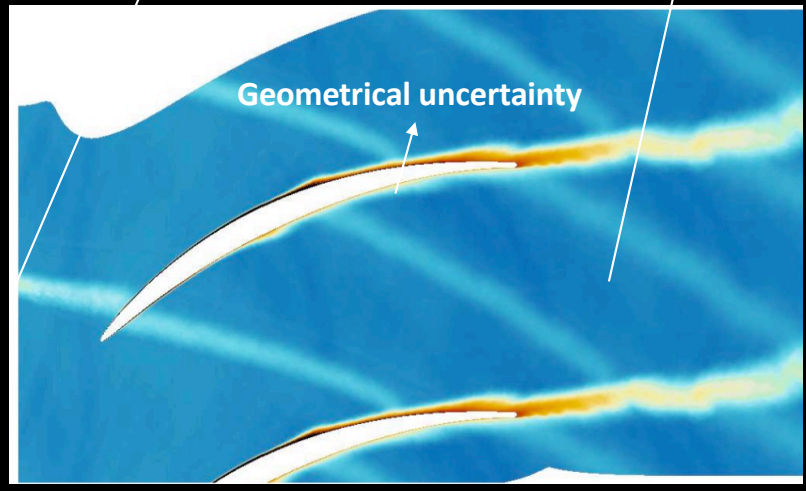
- Automated design and optimization
- Uncertainty quantification
- Digital Twins and real-time simulation

Guo et al., 2016





Sayadi et al., 2020

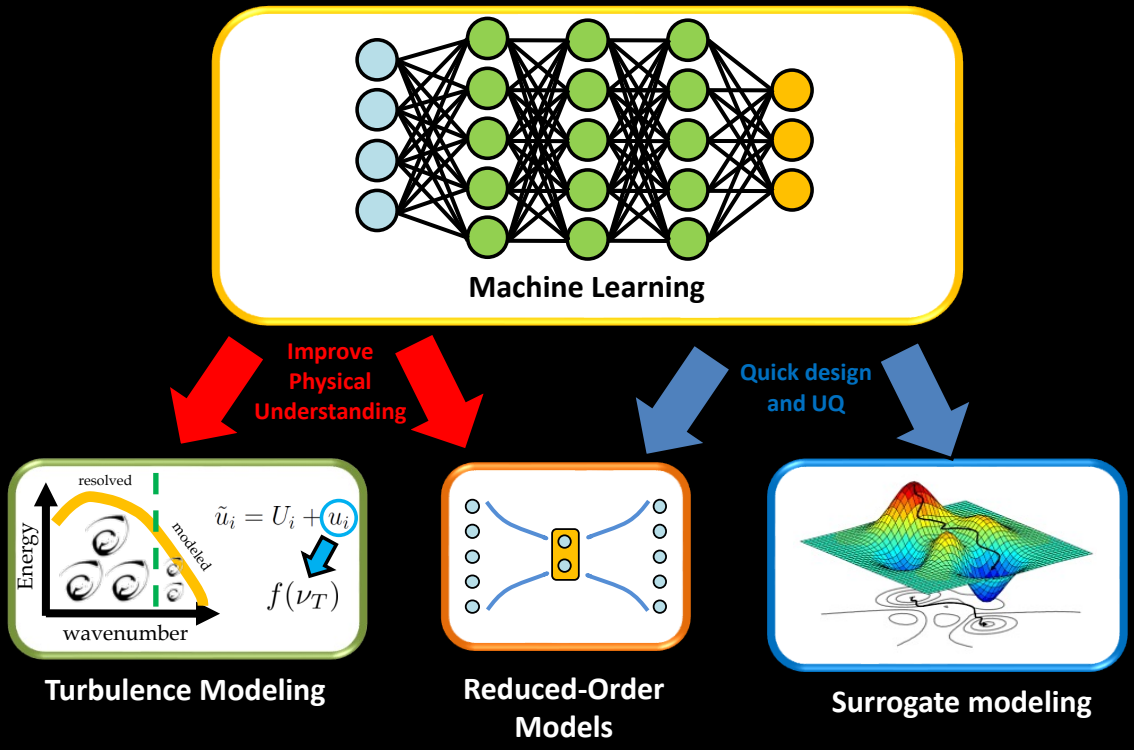


De Zordo Banliat et al., 2020

Promises

Potentially disruptive impact of artificial intelligence/**machine learning (ML)** techniques:

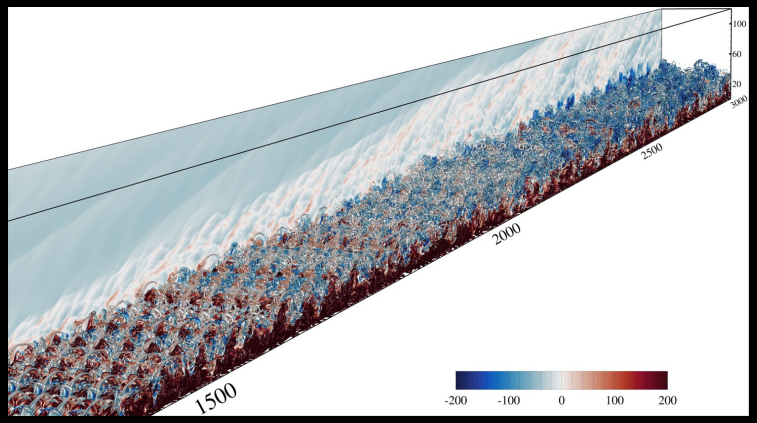
- Abundant HiFi databases → model augmentation, digital twins, control, surrogate modeling, classification…



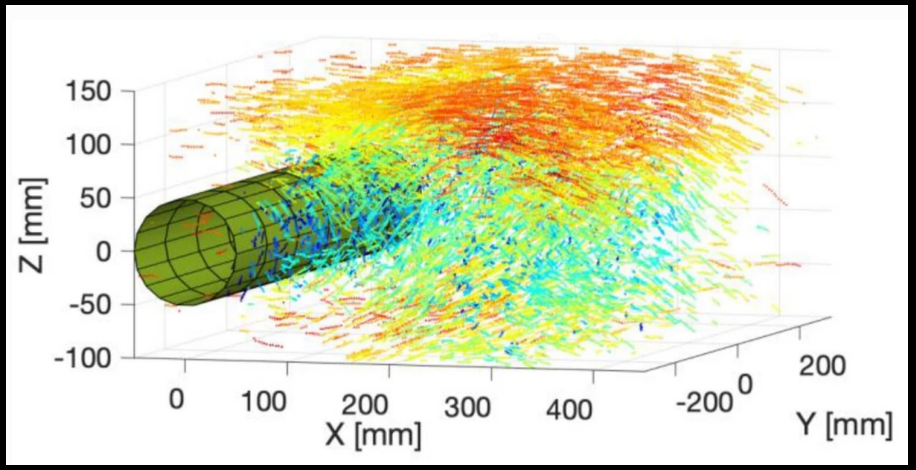
Adapted from Vinuesa & Brunton, 2022

Challenges

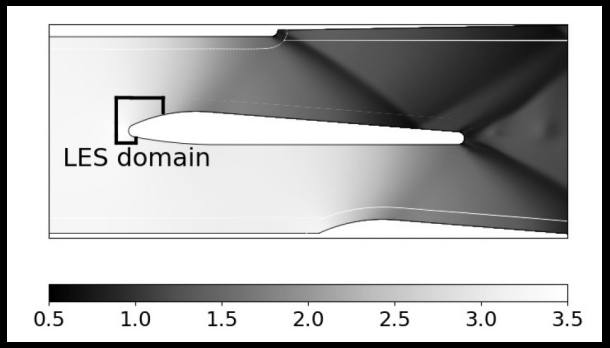
- Large, but scarce and extremely costly data bases
- HiFi-CFD limited to low/moderate-Reynolds numbers
- Experimental databases: higher-Re, but noisy/incomplete
- ML out-of-sample performance?
- ML uncertainty quantification?

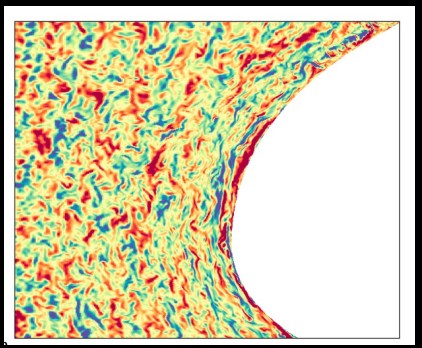


Mach 12.48 boundary layer in thermochemical nonequilibrium 2.5x10⁹ points, 10 equations, 9th-order finite differences, 4x10⁶ CPUh (Passiatore et al., 2022)



Time-resolved tomographic PIV of incompressible flow past a cylinder at Re_D=27000 (Scarano et al., 2022)





Free-stream transition of a dense-gas boundary layer (Novec649 advanced, ozone-friendly working fluid) around a turbine leading edge at Mach 0.9, Re_{L11}~11000, 2.15x10⁹ points, CFL~5, ~10⁶ CPUh (Bienner et al., 2022)

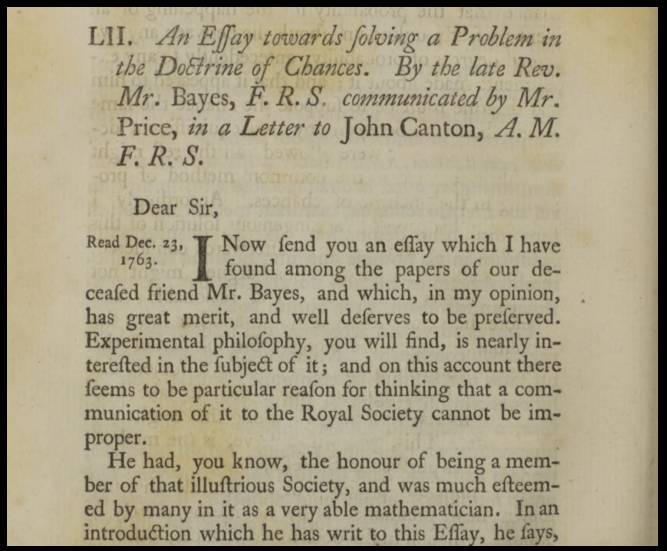
Strategy

- Large, but scarce and extremely costly data bases
  - Parsimonious approaches
- HiFi-CFD limited to low/moderate-Reynolds numbers
- Experimental databases: higher-Re, but noisy/incomplete
  - CFD-in-the-loop training, data assimilation
- Out-of-sample performance?
  - Model mixtures
- Uncertainty quantification?
  - Stochastic Machine Learning

Bayesian Machine Learning



Rev'd Thomas Bayes (1702-1761)



LII. *An Eſſay towards ſolving a Problem in the Doctrine of Chances. By the late Rev. Mr.* Bayes, *F. R. S. communicated by Mr.* Price, *in a Letter to* John Canton, *A. M. F. R. S.*

Dear Sir,

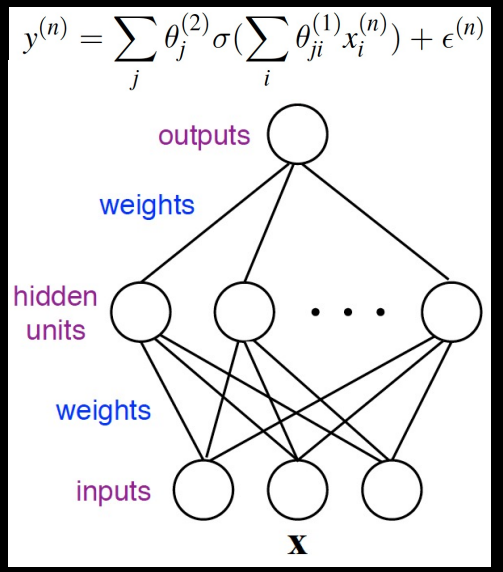
Read Dec. 23, 1763. I Now ſend you an eſſay which I have found among the papers of our deceaſed friend Mr. Bayes, and which, in my opinion, has great merit, and well deſerves to be preſerved. Experimental philoſophy, you will find, is nearly intereſted in the ſubject of it; and on this account there ſeems to be particular reaſon for thinking that a communication of it to the Royal Society cannot be improper.

He had, you know, the honour of being a member of that illuſtrious Society, and was much eſteemed by many in it as a very able mathematician. In an introduction which he has writ to this Eſſay, he ſays,

Deterministic vs Bayesian machine learning

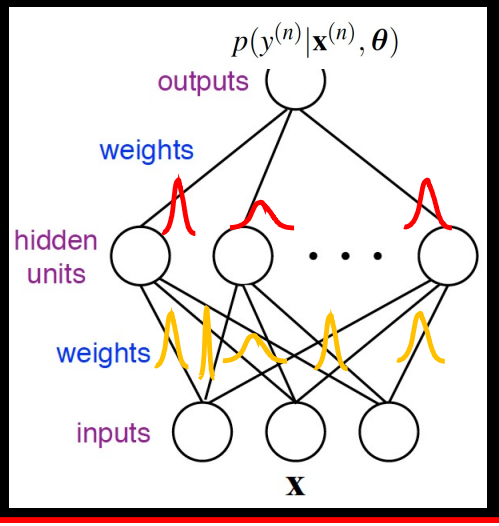
- ML relies on **deterministic regression**:
  - Under given data, optimal model structure + parameter set is selected
  - Data hungry, **no uncertainty estimation** (or difficult)

Neural Network



- **Bayesian ML**
  - Accounts for uncertainties in machine architecture/hyperparameters
  - Well-suited for noisy data, sparse data
  - Provides a confidence measure on predictions
  - Examples: Bayesian neural networks, Mondrian Forests, Sparse Bayesian Learning…

Bayesian Neural Network



Modeling framework

- Consider a model

$$\mathbf{y} = M(\mathbf{x}; \boldsymbol{\theta})$$

$\mathbf{x}$=vector of model inputs; $\mathbf{y}$=vector of model outputs; $\boldsymbol{\theta}$=vector of model parameters

- Choose a functional space for M
  - To fix ideas, we focus on generalized linear models

$$M(\mathbf{x}; \boldsymbol{\theta}) = \sum_{m=1}^{M} \theta_m \Phi_m(\mathbf{x})$$

  - $\Phi_m(\mathbf{x})$=basis functions (e.g. radial-basis functions)

- Consider a set of N data, we may write the design matrix:

$$\boldsymbol{\Phi} = \begin{bmatrix} \Phi_1(\mathbf{x}_1) & \dots & \Phi_M(\mathbf{x}_1) \\ \dots & \ddots & \dots \\ \Phi_1(\mathbf{x}_N) & \dots & \Phi_M(\mathbf{x}_N) \end{bmatrix}_{N \times M}$$

Deterministic regression

Least mean square (LMS) regression

- minimize mean square error (MSE) to the data $\mathbf{Y}$ for some inputs $\mathbf{X}$

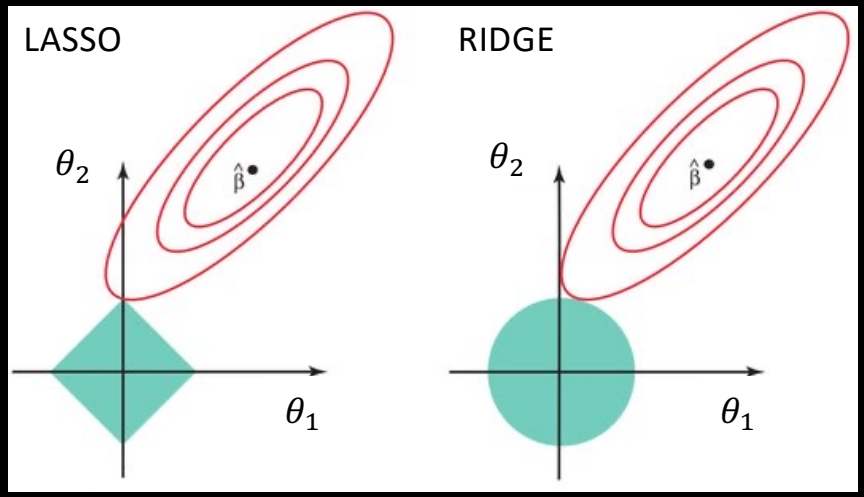
$$N \gg M$$

$$\varepsilon_D^2(\boldsymbol{\theta}) = \left|\left|\mathbf{Y} - \boldsymbol{\Phi}\boldsymbol{\theta}\right|\right|_2^2 \rightarrow \boldsymbol{\theta} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{Y}$$

- Complexity control via regularization

$$\hat{\varepsilon}^2(\boldsymbol{\theta}) = \varepsilon_D^2(\boldsymbol{\theta}) + \lambda\varepsilon_\theta$$

- Penalty on model parameters
  - L2-norm penalty $\varepsilon_\theta = \left|\left|\boldsymbol{\theta}\right|\right|_2^2$
    → Ridge (Tikhonov regularization)
  - L1-norm penalty $\varepsilon_\theta = \left|\left|\boldsymbol{\theta}\right|\right|_1$
    → LASSO (Least Absolute Shrinkage and Selection Operator)



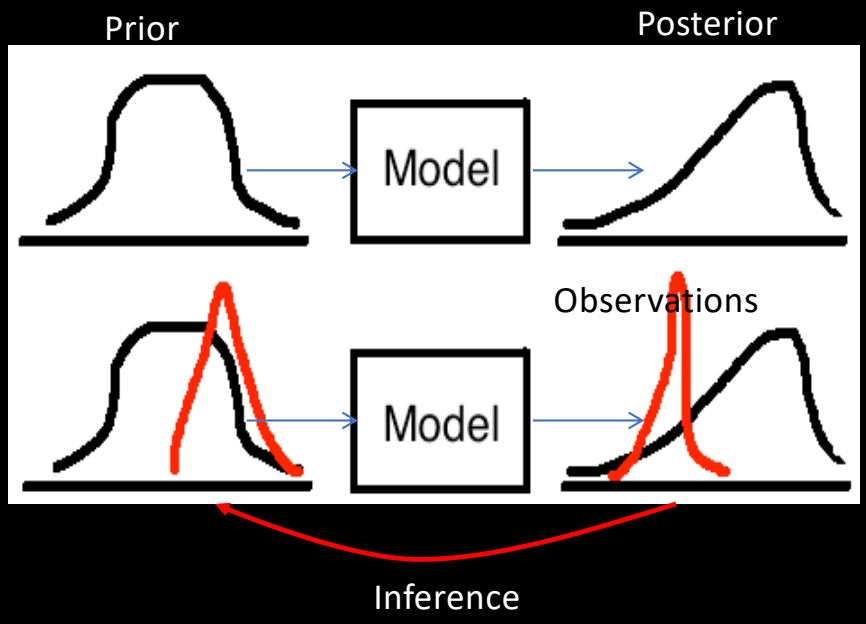
Outcome of LMS: 1 model structure + 1 set of optimal parameters

How good are they?

Probabilistic modeling

- Express all forms of uncertainty and noise associated with the model via probability theory

- Law of inverse probability (**Bayes' rule**) allows to infer unknown quantities, adapt models to new observations, make predictions and learn from data



Bayesian inference

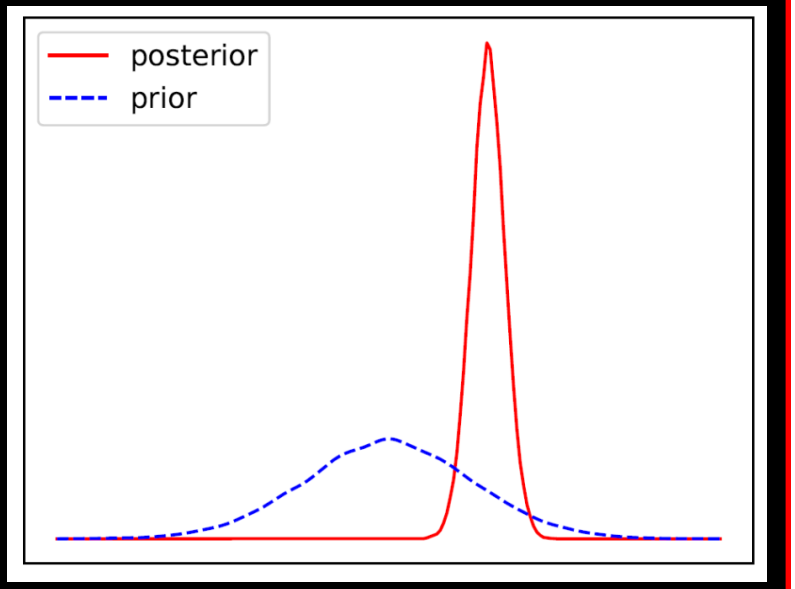
- **Bayes'rule** allows to do inference about hypothesis given some data (observations)
- $\theta \rightarrow$ parameters
- $\mathbf{Y} \rightarrow$ observations

Likelihood Prior

Posterior

$$p(\boldsymbol{\theta}|\mathbf{Y}, M) = \frac{p(\mathbf{Y}, \boldsymbol{\theta}|M)}{p(\mathbf{Y}|M)} = \frac{p(\mathbf{Y}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta})}{p(\mathbf{Y}|M)}$$

Evidence



- Prior: summarizes hypothesis before we observe data
- Likelihood: probability of observing the data if the hypothesis is true
- Evidence or « Marginal likelihood »: probability that randomly selected parameters from the prior would generate the data
- Posterior: hypothesis updated after observing the data

Bayesian regression vs LMS

- Regression of noisy data:

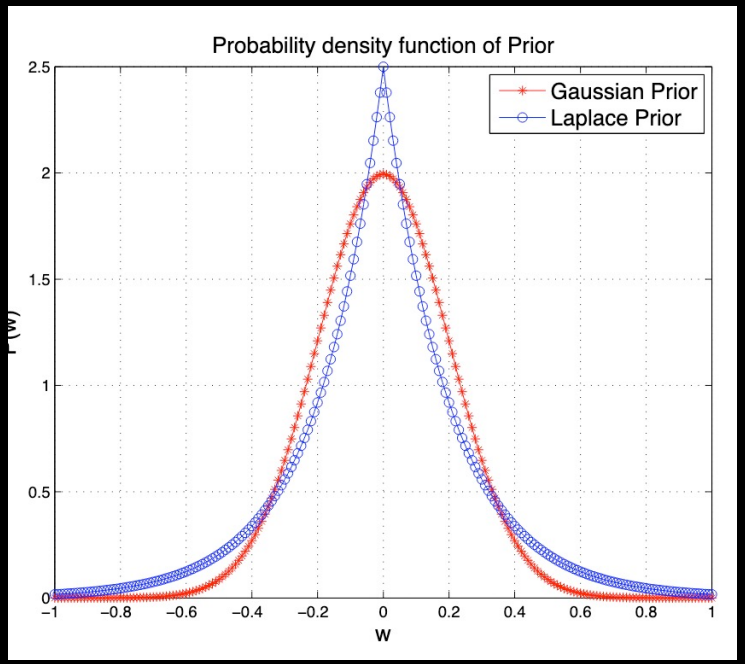
$$Y_n = M(X_n; \boldsymbol{\theta}) + e_n$$

  - e_n i.i.d. Gaussian noise with variance σ^2

$$p(e_n|\sigma^2) = \mathcal{N}(0, \sigma^2)$$

- The maximum likelihood estimate of $\boldsymbol{\theta}$ corresponds to minimizing the MSE to the data
- The maximum a posteriori estimate of $\boldsymbol{\theta}$ corresponds to:
  - **Gaussian** prior → ridge-type regularization with $\lambda = \sigma^2\alpha$
  - **Laplace** prior → LASSO-type regularization

More peaked at 0 → sparser



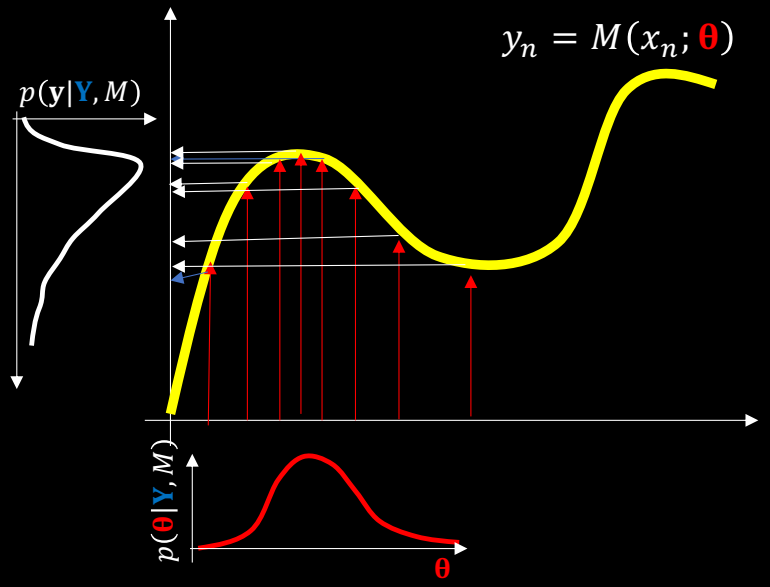
So Bayesian regression equivalent to LMS?

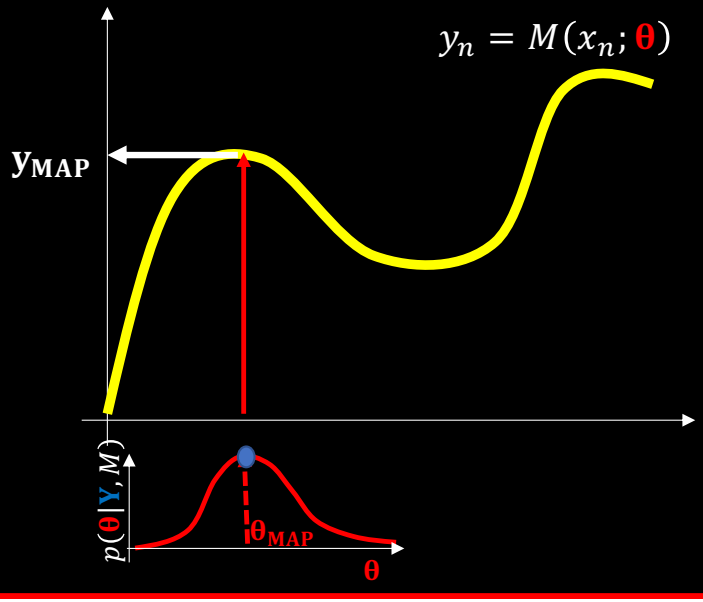
- **No**: now we have **posteriors**!
- Posterior predictive distribution (P.P.D.):

$$p(\mathbf{y}|\mathbf{Y},M) = \int p(\mathbf{y}|\mathbf{Y},\boldsymbol{\theta},\alpha,\sigma^2,M)p(\boldsymbol{\theta},\alpha,\sigma^2|\mathbf{Y})d\boldsymbol{\theta}d\alpha d\sigma^2$$

$\mathbf{y}$=model output at a new input $\mathbf{x}$

- Prediction with **quantified uncertainty**



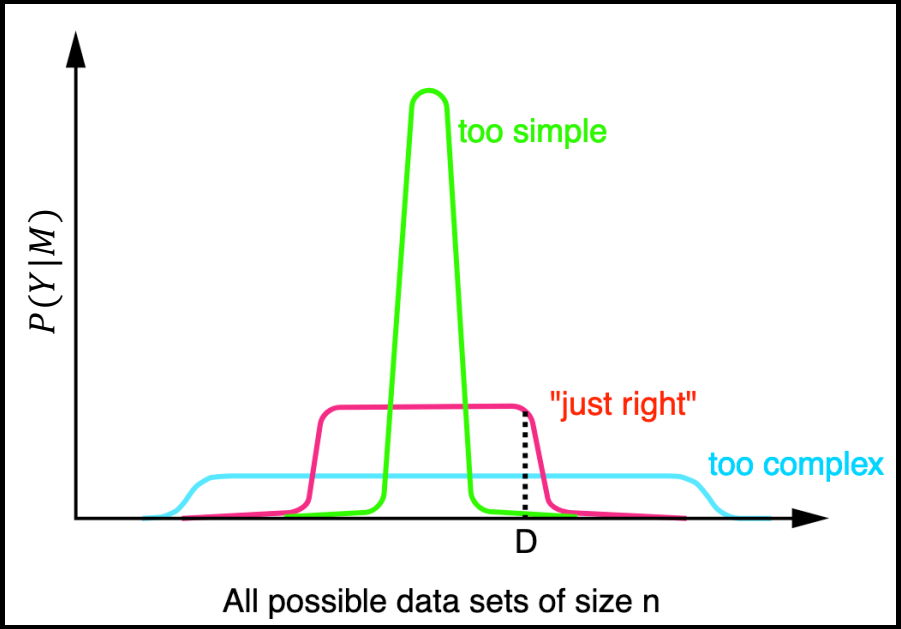


Bayesian Ockham's razor and model comparison

- Automatic model selection through $p(\mathbf{Y}|M)$
  - Compare model classes, e.g. M and M', using posterior model probabilities

$$P(M|\mathbf{Y}) = \frac{p(\mathbf{Y}|M)P(M)}{p(\mathbf{Y})}; \quad P(M'|\mathbf{Y}) = \frac{p(\mathbf{Y}|M')P(M')}{p(\mathbf{Y})}$$

  - Too simple classes unlikely to generate the data set
  - Too complex can generate many possible data sets → unlikely to generate one particular data set at random



From Z. Ghahramani

Pros and cons of Bayesian approach

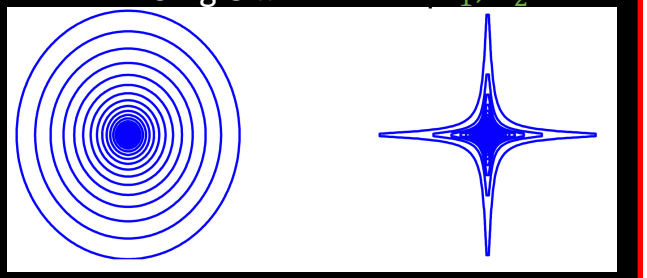
- Pros
  - ☺ Expert knowledge or preferences of the models easily incorporated into the model by prior distribution.
  - ☺ Probabilistic outputs with confidence intervals
  - ☺ Sparsity promoting via automatic implementation of « Ockham's razor »

- Cons
  - ☹ Analytically intractable multidimensional integrals

Model <u>evidence</u>: $p(\mathbf{Y}|M) = \int p(\mathbf{Y}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|\alpha)p(\sigma^2)p(\alpha)d\boldsymbol{\theta}d\alpha d\sigma^2$

<u>Posterior predictive distribution</u>: $p(\mathbf{y}|\mathbf{Y}, M) = \int p(\mathbf{y}|\mathbf{Y}, \boldsymbol{\theta}, \alpha, \sigma^2)p(\boldsymbol{\theta}, \alpha, \sigma^2|\mathbf{Y}, M)d\boldsymbol{\theta}d\alpha d\sigma^2$

Sparse Bayesian Learning [Tipping 2001]



- Redundant dictionary of functions
- Gaussian priors with individual precisions α_m
- Objective: find $p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2|\mathbf{Y}) = \underbrace{p(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\alpha}, \sigma^2)}_{\text{Analytically computable}} \times \underbrace{p(\boldsymbol{\alpha}, \sigma^2|\mathbf{Y})}_{\text{Approximate}}$
- Do iterative procedure:
  1. Initialise prior hyperparameters $\boldsymbol{\alpha}, \sigma^2$
  2. Posterior maximization: analytically update the optimal weight vector $\boldsymbol{\theta}$ by maximizing the posterior of weights $p(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\alpha}, \sigma^2)$ given $\boldsymbol{\alpha}, \sigma^2$
  3. Evidence maximization: update $\boldsymbol{\alpha}, \sigma^2$ by approximately maximizing the evidence $p(\mathbf{Y}|\boldsymbol{\alpha}, \sigma^2)$ → $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{MAP}, \sigma^2 = \sigma^2_{MAP}$

     if $\alpha_m \to \infty$, the component is discarded (automatic selection)
  4. Loop steps (2) and (3) until converged

”Black-box” vs “open-box” machine learning

- **Black-box** → highly expressive, hardly interpretable, data-hungry

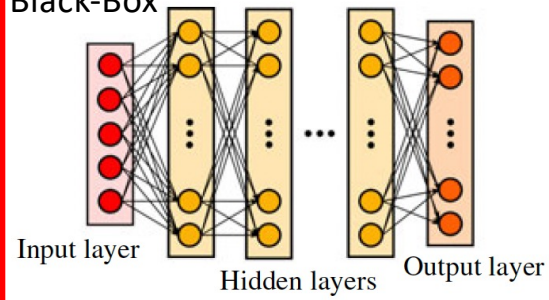
- **Grey-box or “physics-informed”**
  - TBNN (Ling et al. 2016), (Wu et al., 2017)
  - TBRF (Kandoorp&Dwight, 2020)

- **Field inversion and machine learning (**Parish&Duraisamy 2016)

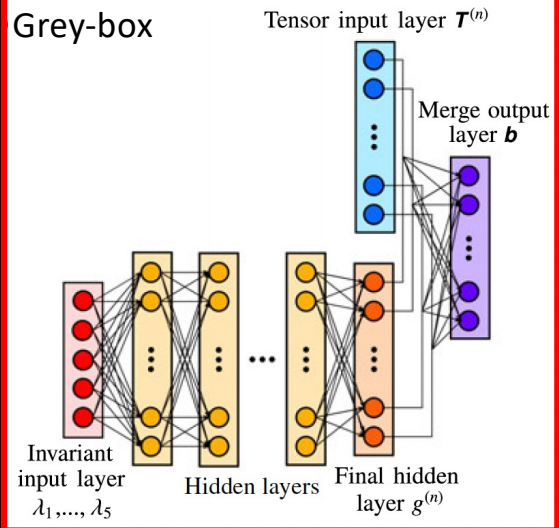
  → optimal “corrective” field + ML extrapolation

- **Open-box** (symbolic identification) → dictionary of explicit operators
  - Gene-Expression Programming (Weatheritt&Sandberg, 2016)
  - SpaRTA (Schmelzer et al. 2020)
  - SBL-SpaRTA (Cherroud et al. 2022)

Black-Box



Grey-box



Open-box

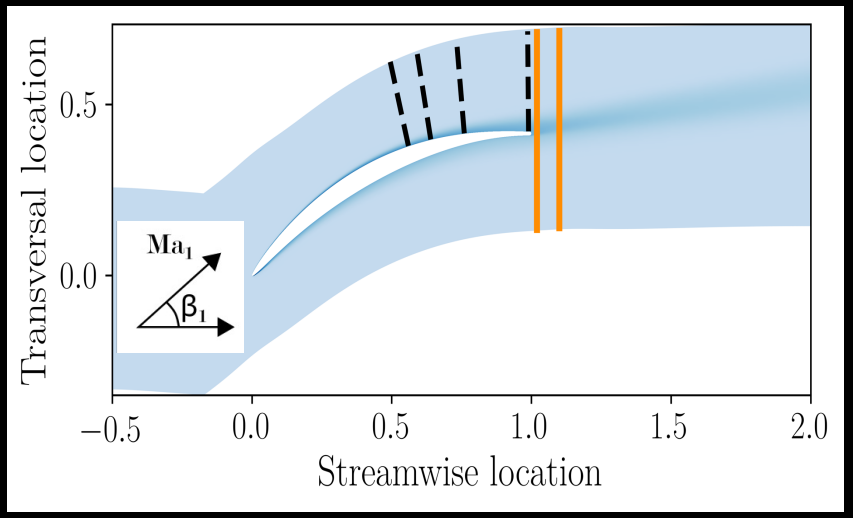
$$M(\mathbf{x},\boldsymbol{\theta}) = \sum_{m=1}^{M} \theta_m \Phi_m(\mathbf{x})$$

First Bayesian exercise: model calibration

[De Zordo-Banliat et al., 2020]

Flow through the NACA65-V103 compressor cascade

Scenario	S_1	S_2	S_3	S_4
β_1	36.99°	39.97°	44.09°	49.2°
Ma_1	0.654	0.674	0.666	0.65
Re_1	302K	302K	298K	289K
Tu (%)	2.9	3.4	3.4	3.5

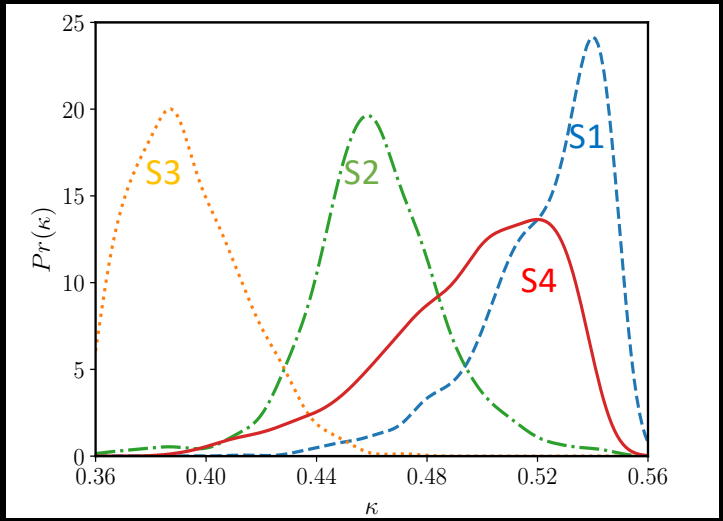


Model: Spalart-Allmaras (7 closure coefficients)

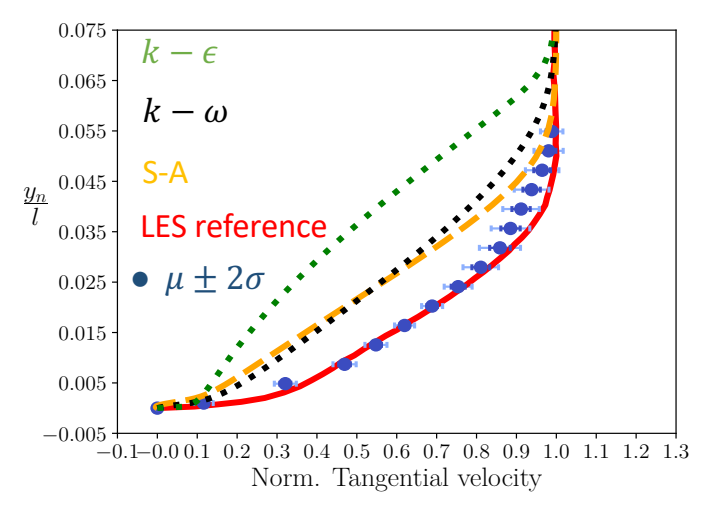
Data : LES, 4 velocity profiles and 2 total pressure profiles [Leggett et al., 2016]

Posterior : MCMC sampling from Kriging surrogate of CFD solver

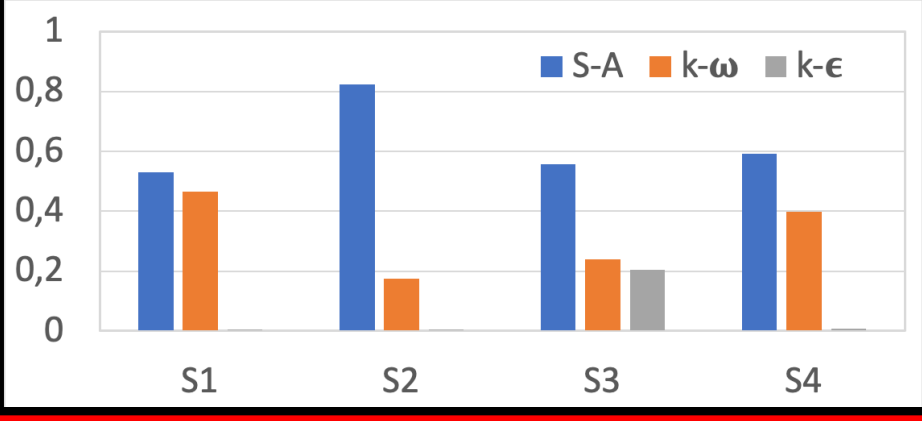
Posteriors and predictive distribution



Von Karman « constant » κ



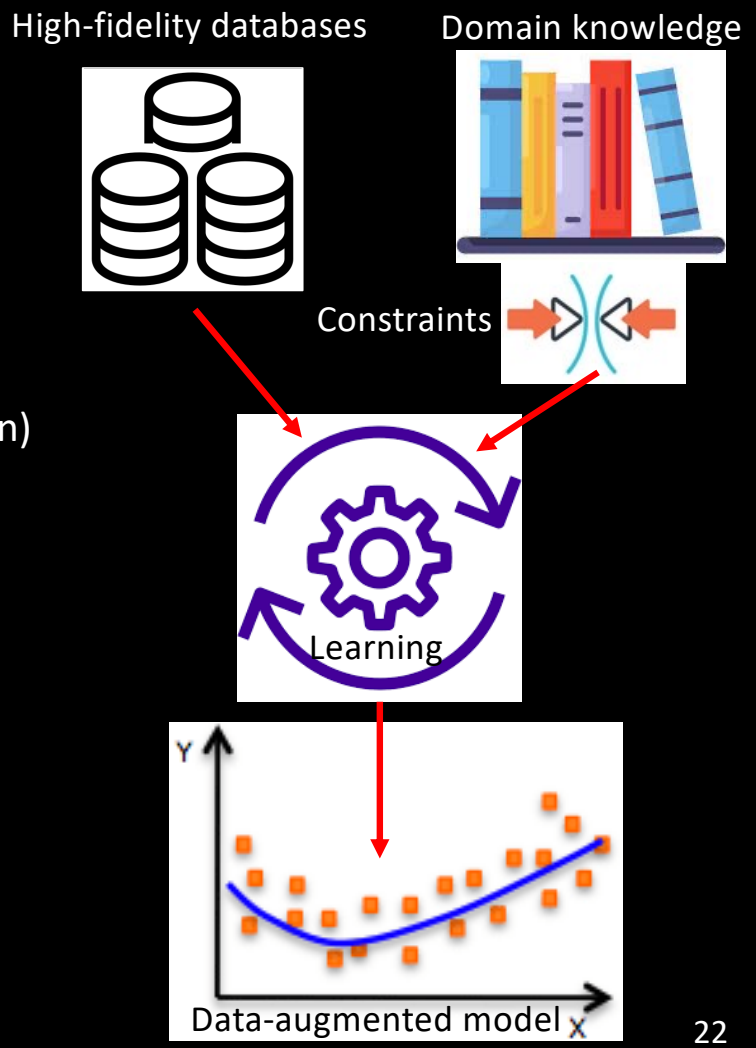
P.P.D., velocity profile, S2



Posterior model probabilities for various scenarios

Data-augmented turbulence modeling

- General philosophy
  - No « universal » model
  - Customized models for flow classes
  - Mostly formulated as a “correction” (data-driven augmentation)
    1. Collect data
    2. Choose a functional basis
    3. Enforce physical constraints (whenever possible)
    4. Train against data
    5. Validate for a test set



Data-driven discovery of turbulence models

[Schmelzer et al., 2020]

SpaRTA = Sparse Regression of Turbulent-stress Anisotropy

- Start with linear eddy viscosity model (here, Menter's $k-\omega$ SST)

$$\tau_{ij} = 2k\left(b_{ij} + \frac{1}{3}\delta_{ij}\right); \quad b_{ij} = -\frac{\nu_t}{k}S_{ij}; \quad \nu_t = f(k,\omega)$$

+ transport equations for k and ω

- Internal additive corrections of Reynolds stress anisotropy (b_{ij}^{Δ}) and turbulent transport equations (R):

$$b_{ij} = -\frac{\nu_t}{k}S_{ij} + b_{ij}^{\Delta} \qquad \frac{Dk}{Dt} = P + P^{\Delta} + D + T + R \qquad \frac{D\omega}{Dt} = P_{\omega} + P_{\omega,\Delta} + P_{\omega,R} + D + T$$

- **Learn b_{ij}^{Δ} and R from high-fidelity data**

Data-driven discovery of turbulence models

- Ansatz for b_{ij}^{Δ} [inspired from Pope, JFM, 1975]:

$$b_{ij}^{\Delta} = \sum_{l=1,\dots,10} \alpha_l(I_1, I_2, I_3, I_4, I_5) T_{ij}^l$$

- Ansatz for R:

$$R = 2k b_{ij}^{R} \frac{\partial \bar{u}_i}{\partial x_j}$$

- Dictionary of function libraries $\mathcal{B}_l = [f_{0l}, f_{1l}, \dots f_{Nl}]$ with $f_{kl} = f_{kl}(I_1, \dots, I_5)$

$$b_{ij}^{\square} = \mathbf{C}_{\square} \cdot \mathbf{\Theta}_{\square} \qquad (\square = \Delta, R)$$

$\mathbf{C}_{\square} = \mathcal{B}_1 T_{ij}^1 \sqcup \mathcal{B}_2 T_{ij}^2 \sqcup \cdots \sqcup \mathcal{B}_{10} T_{ij}^{10}$ → candidate tensor functions

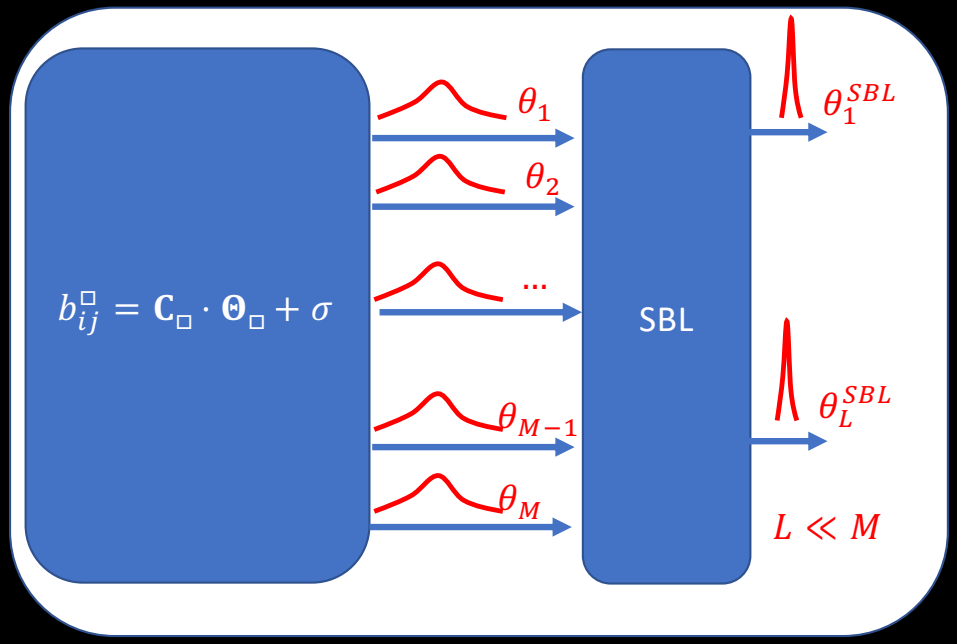
$\mathbf{\Theta}_{\square} = \Theta_1 \sqcup \Theta_2 \sqcup \dots \sqcup \Theta_{10}$ → vector of coefficients

- Find Θ by solving a **regularized regression problem** so that most coefficients are zero

OUTCOME: sparse data-driven Explicit Algebraic Reynolds-Stress Model (EARSM)

SBL-SpaRTA [Cherroud et al., 2022]

- Find $p\left(\boldsymbol{\Theta}_{\square}, \boldsymbol{\alpha}, \sigma^2 \middle| b_{ij}^{\square}\right)$ using the SBL algorithm
  - Select functions in $\mathbf{C}_{\square}$ dictionary and infer posteriors



Ref	Abr.	Data
D_1	ZPG	DNS of turbulent boundary layer, $670 \leq Re_\theta \leq 4060$ [1]
D_2	FDC	DNS of turbulent channel flow, $180 \leq Re_\tau \leq 590$ [2]
D_3	ANSJ	PIV of near sonic axisymmetric jet [3]
D_4	APG	LES of adverse pressure-gradient TBL [4] $Re_\theta \leq 4000$, $\beta = 4$, 5 different pressure gradients
D_5	SEP	LES of Periodic Hills at Re=10595 [5] DNS of converging-diverging channel at Re=13600 [6] LES of curved backward facing step at Re = 13700 [7]

Example of discovered model:

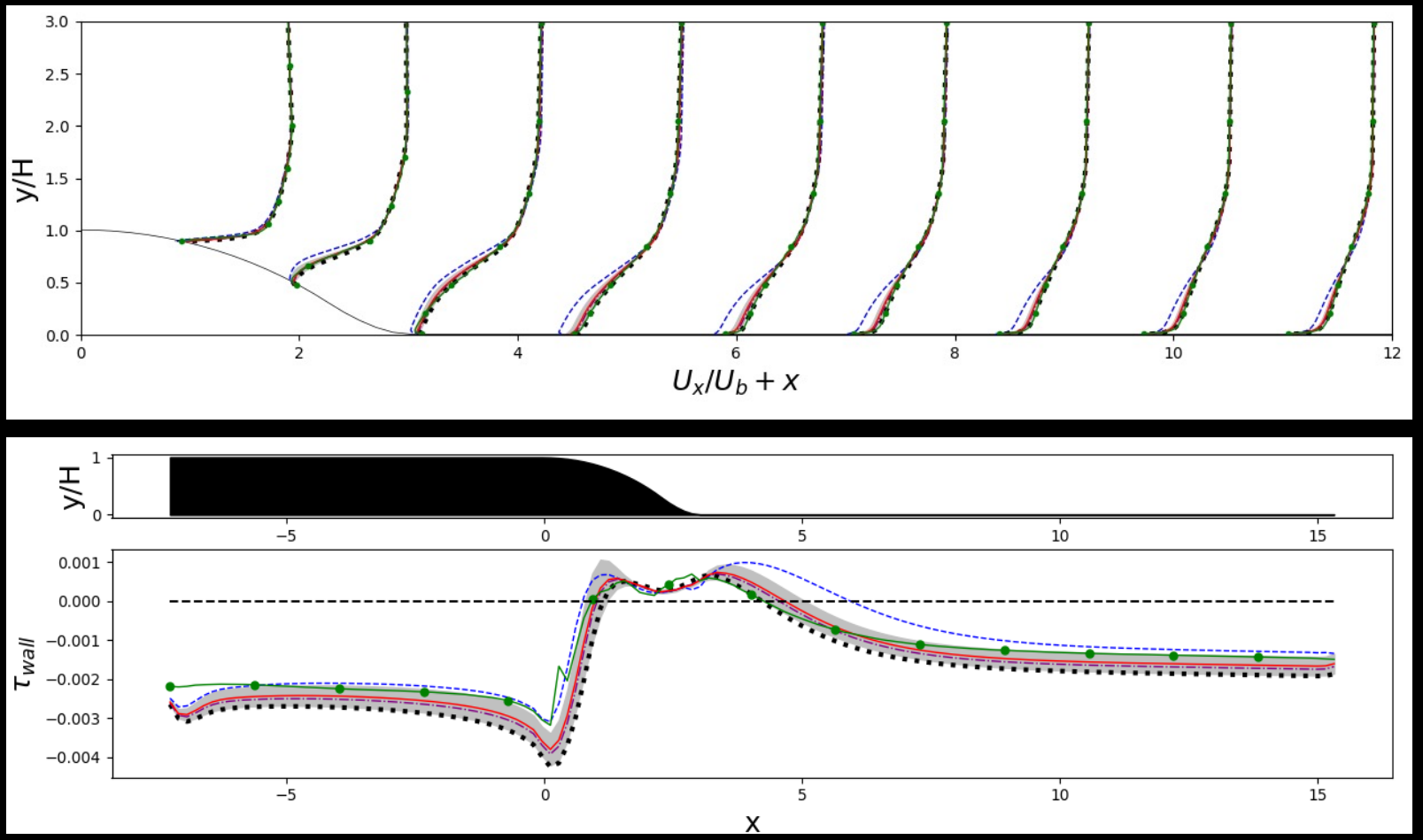
$$\begin{cases} \mathbf{M}_{\mathbf{b}^{\Delta}}^{(SEP)} = [(5.21 \pm 0.0173)]\mathbf{T}^{(2)} + \pm 0.0348 \\ \mathbf{M}_{\mathbf{b}^{R}}^{(SEP)} = [(0.681 \pm 0.02)]\mathbf{T}^{(1)} \pm 0.0318 \end{cases}$$

$E[\theta]$ $std[\theta]$ σ

ZPG: the discovered model correction is 0!

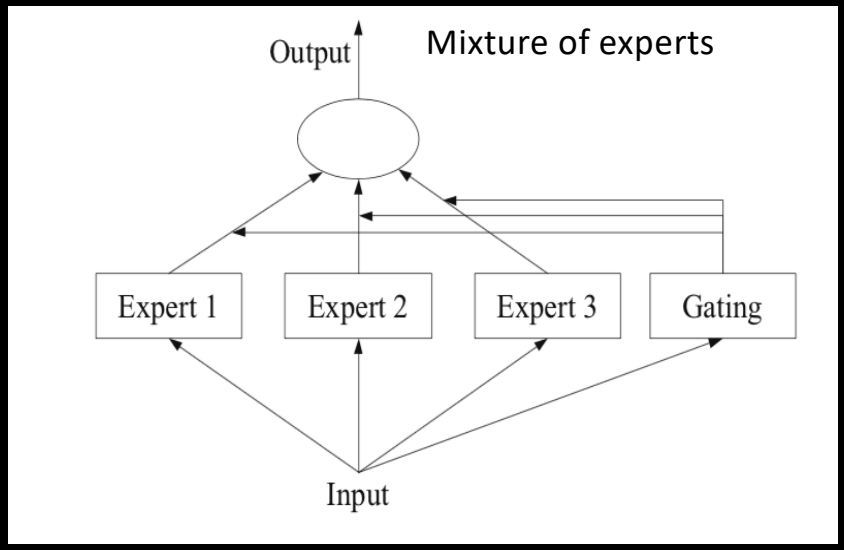
SBL-SpaRTA

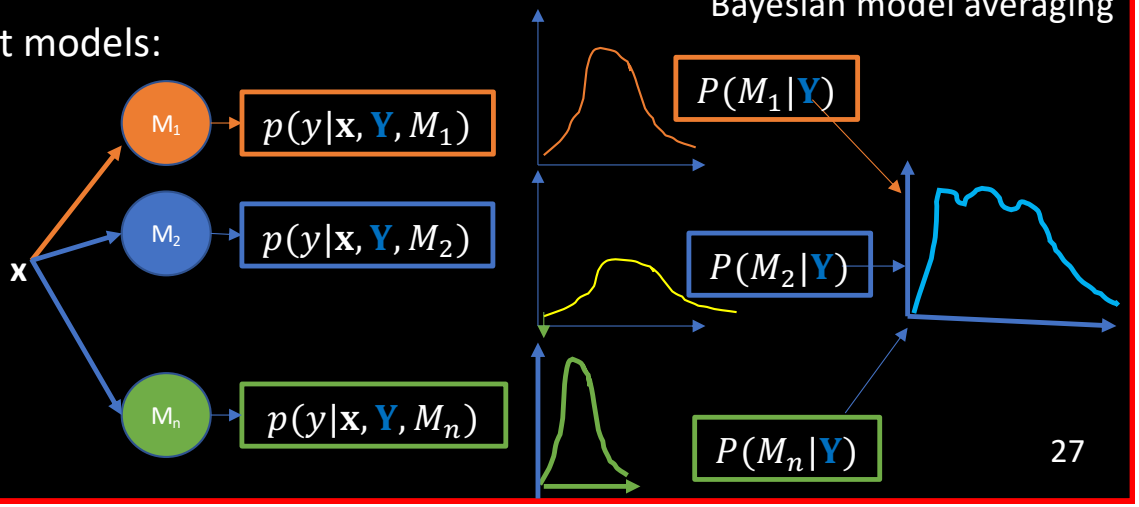
- Curved backward-facing step flow at Re=13700



Bayesian model committees

- How good is our model at predicting another data sets?
- Large data sets: combining models trained on subsets better than single model trained over all data
- Out of sample predictions: uncertainty on which model (among those at hand) is better
- Generate hypermodels by combining component models:
  - Mixtures of experts (regional weights)
  - Bayesian model averaging (global weights)
  - Clustered Bayesian averaging





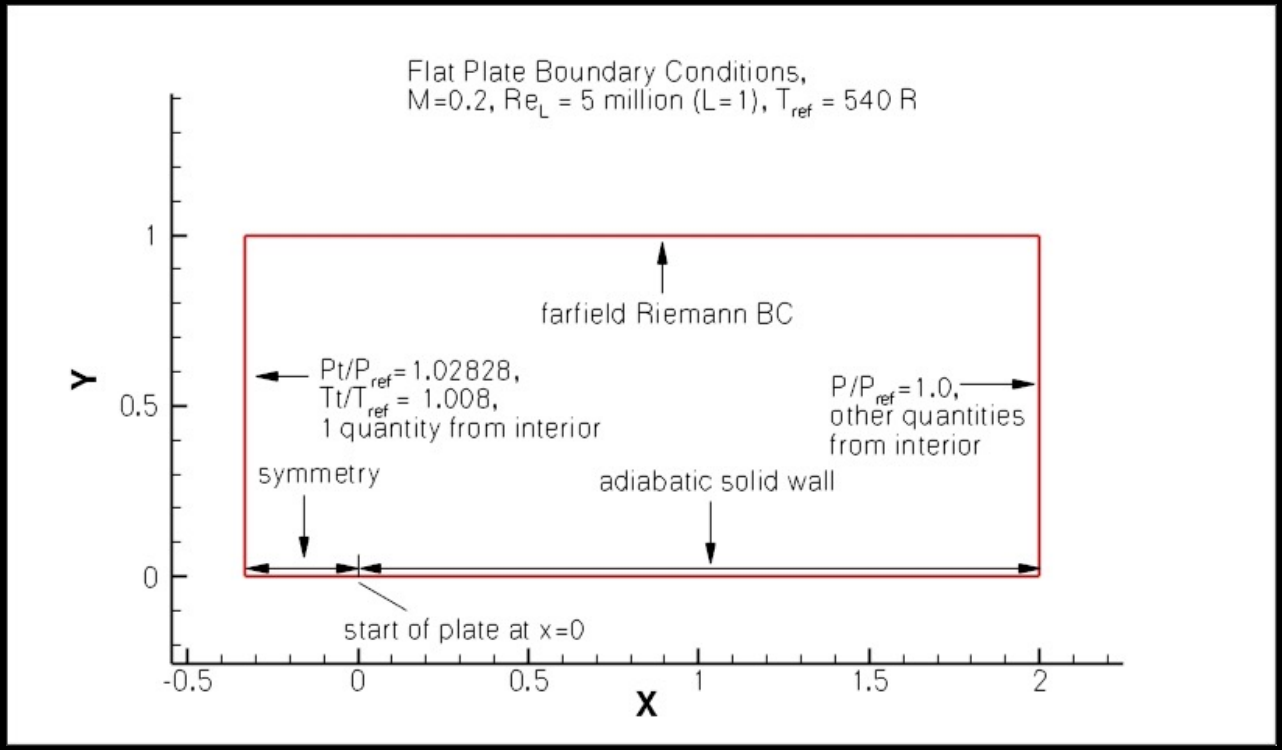
Spatial model aggregation (X-MA) of turbulence models [de Zordo et al., 2021]

- Consider a set of K competing models $\mathcal{M} = \{\boldsymbol{M_1}, \boldsymbol{M_2}, \dots, \boldsymbol{M_K}\}$
- « Hypermodel »: $M_{hyp}(\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} w_k M_K(\mathbf{x}; \boldsymbol{\theta})$, with
$$w_k = w_k(\mathbf{x}) = w_k(\boldsymbol{\eta}(\mathbf{x}))$$
- Regress $w_k(\boldsymbol{\eta}(\mathbf{x})|\mathbf{Y})$ from data as a function of features
  - Predict local model weights for a new case $w_k(\boldsymbol{\eta}(\mathbf{x})|\mathbf{Y})$ and use them to aggregate individual model predictions

Feature	Description	Formula	Feature	Description	Formula
η_1	Normalized Q criterion	$\frac{\|\|\boldsymbol{\Omega}\|\|^2 - \|\|\mathbf{S}\|\|^2}{\|\|\boldsymbol{\Omega}\|\|^2 + \|\|\mathbf{S}\|\|^2}$	η_6	Viscosity ratio	$\frac{\nu_T}{100\nu + \nu_T}$
η_2	Turbulence intensity	$\frac{k}{0.5U_iU_i + k}$	η_7	Ratio of pressure normal stresses to normal shear stresses	$\frac{\sqrt{\frac{\partial P}{\partial x_i}\frac{\partial P}{\partial x_i}}}{\sqrt{\frac{\partial P}{\partial x_j}\frac{\partial P}{\partial x_j}} + 0.5\rho\frac{\partial U_k^2}{\partial x_k}}$
η_3	Turbulent Reynolds number	$\min\left(\frac{\sqrt{k}\lambda}{50\nu}, 2\right)$	η_8	Non-orthogonality marker between velocity and its gradient [28]	$\frac{\left\|U_kU_l\frac{\partial U_k}{\partial x_l}\right\|}{\sqrt{U_nU_nU_i\frac{\partial U_i}{\partial x_j}U_m\frac{\partial U_m}{\partial x_j}} + \left\|U_iU_j\frac{\partial U_i}{\partial x_j}\right\|}$
η_4	Pressure gradient along streamline	$\frac{U_k\frac{\partial P}{\partial x_k}}{\sqrt{\frac{\partial P}{\partial x_j}\frac{\partial P}{\partial x_j}U_iU_i} + \left\|U_l\frac{\partial P}{\partial x_l}\right\|}$	η_9	Ratio of convection to production of k	$\frac{U_i\frac{\partial k}{\partial x_i}}{\|\overline{u_j'u_l'}S_{jl}\| + U_l\frac{\partial k}{\partial x_l}}$
η_5	Ratio of turbulent time scale to mean strain time scale	$\frac{\|\|\mathbf{S}\|\|k}{\|\|\mathbf{S}\|\|k + \varepsilon}$	η_{10}	Ratio of total Reynolds stresses to normal Reynolds stresses	$\frac{\|\|\overline{u_i'u_j'}\|\|}{k + \|\|\overline{u_i'u_j'}\|\|}$

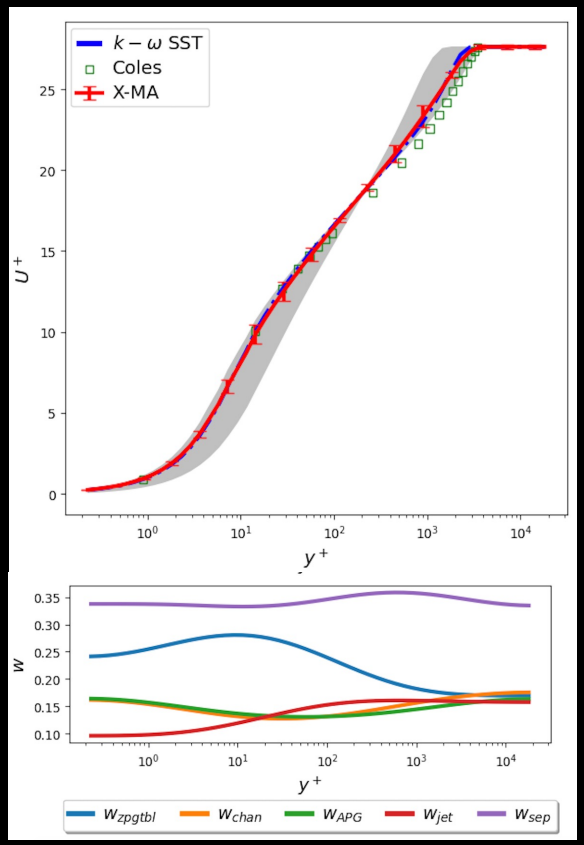
NASA Turbulence Modeling Testing Challenge

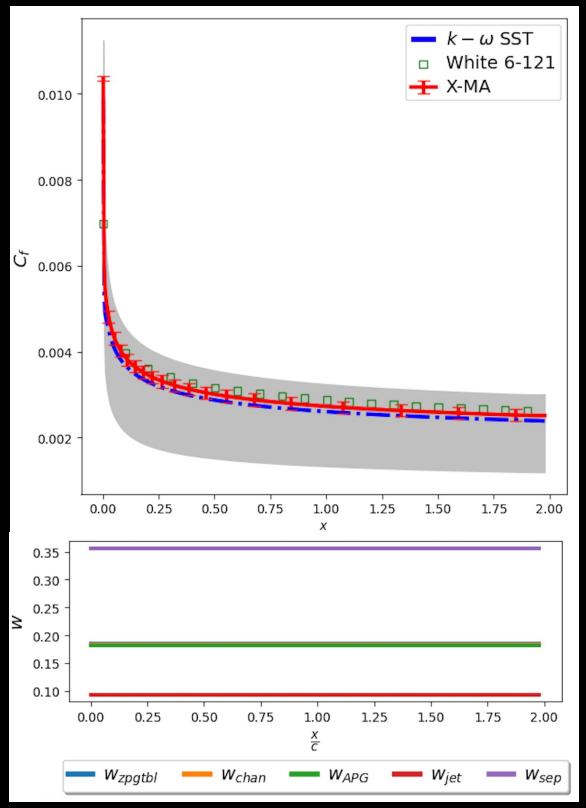
- Application to Test Case 1 2DZP: 2D Zero Pressure Gradient Flat Plate
  - Show (1) Cf vs. x and (2) u+ vs. log(y+) at x=0.97; compare with theory



NASA Turbulence Modeling Testing Challenge

- Application to Test Case 1 2DZP: 2D Zero Pressure Gradient Flat Plate
  - Show (1) Cf vs. x and (2) u+ vs. log(y+) at x=0.97; compare with theory

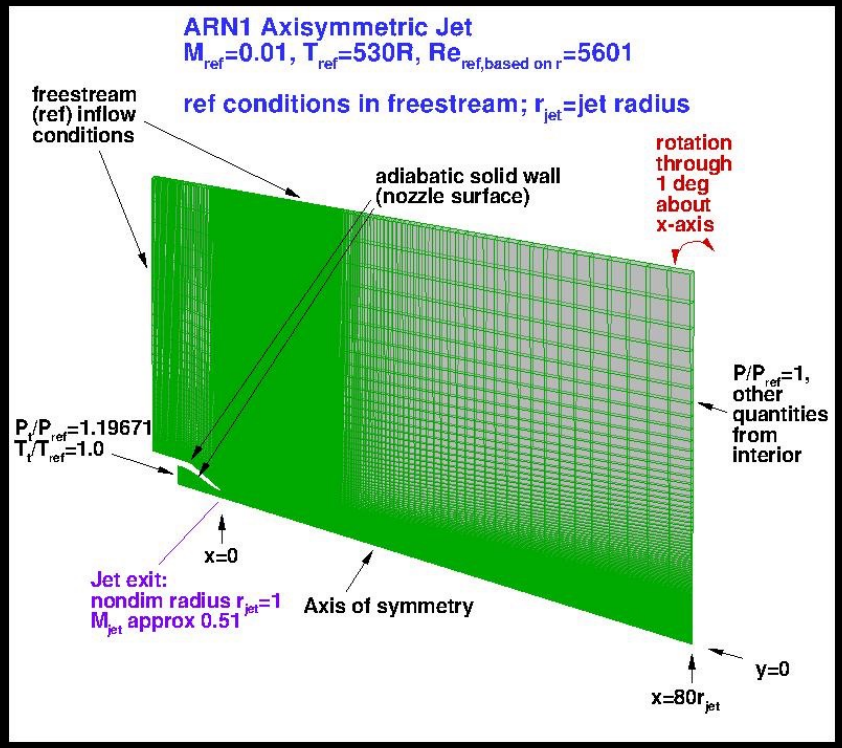




Accurate prediction of the 2DZP

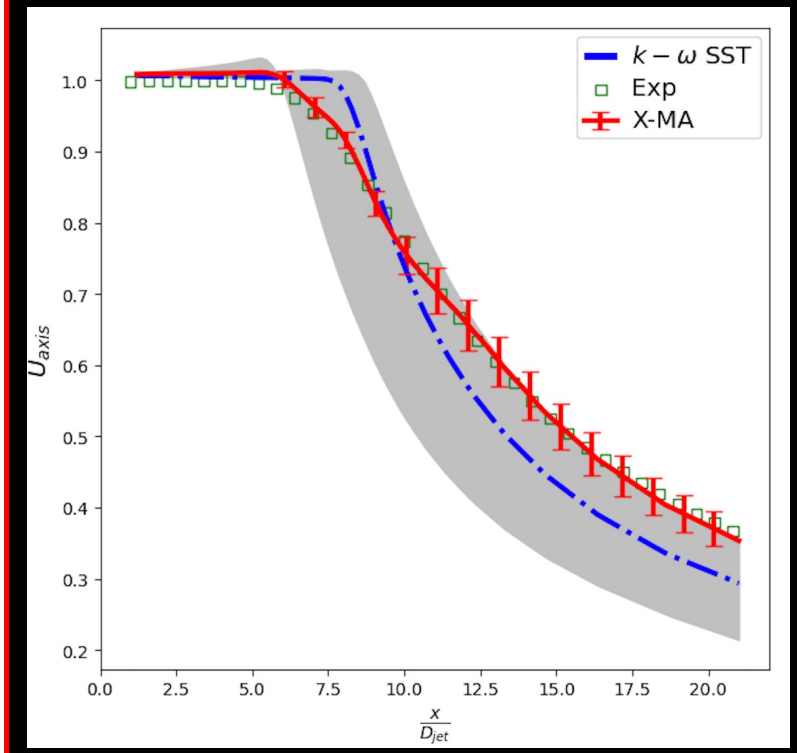
NASA Turbulence Modeling Testing Challenge

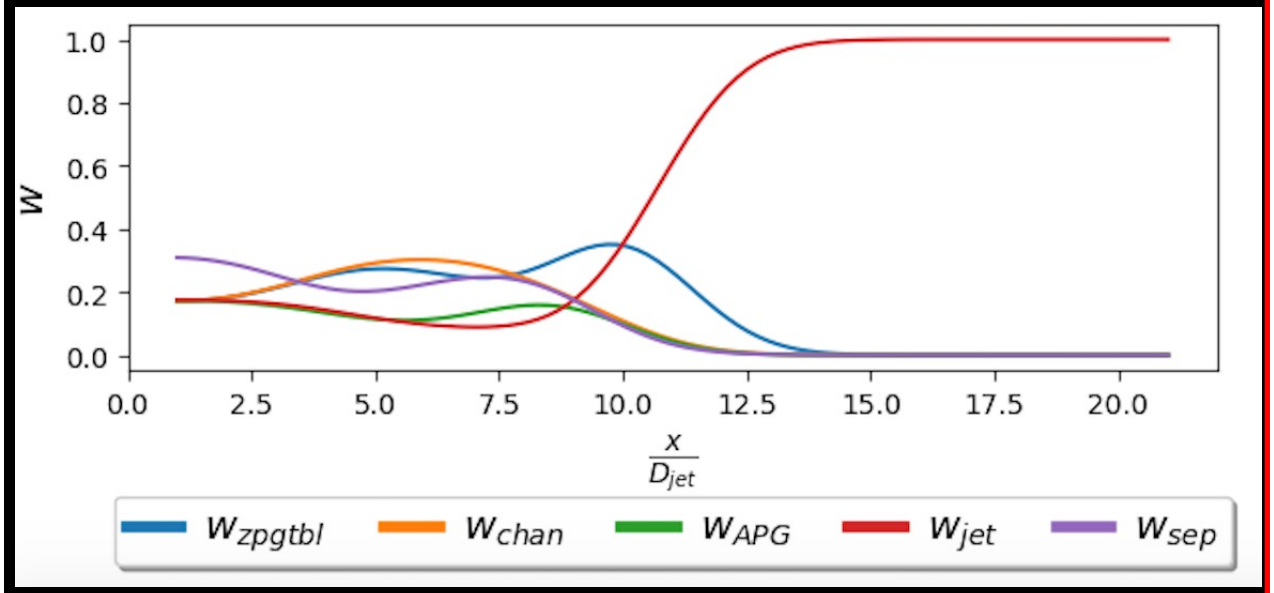
- Application to Test Case 3 ASJ: Axisymmetric Subsonic Jet



NASA Turbulence Modeling Testing Challenge

- Application to Test Case 3 ASJ: Axisymmetric Subsonic Jet

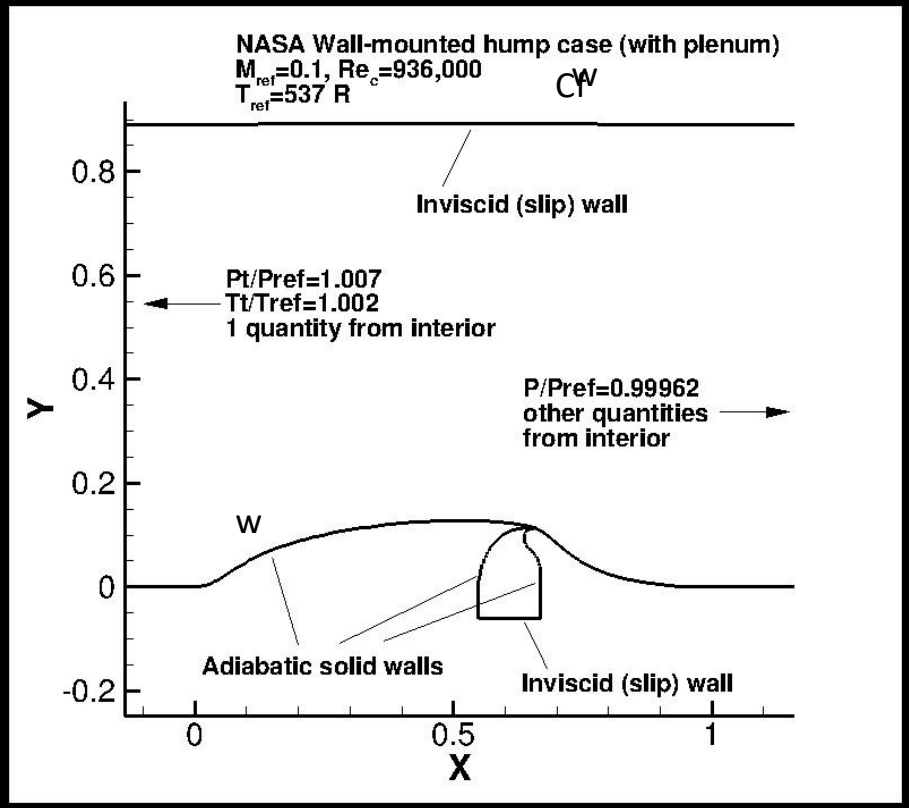




X-MA puts higher weight on the 'jet model'

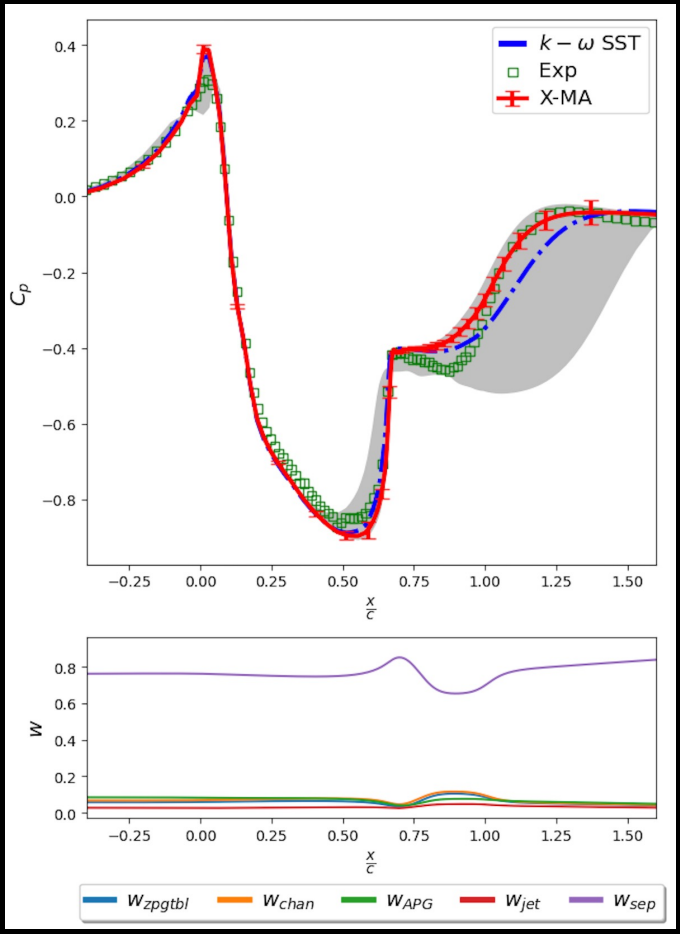
NASA Turbulence Modeling Testing Challenge

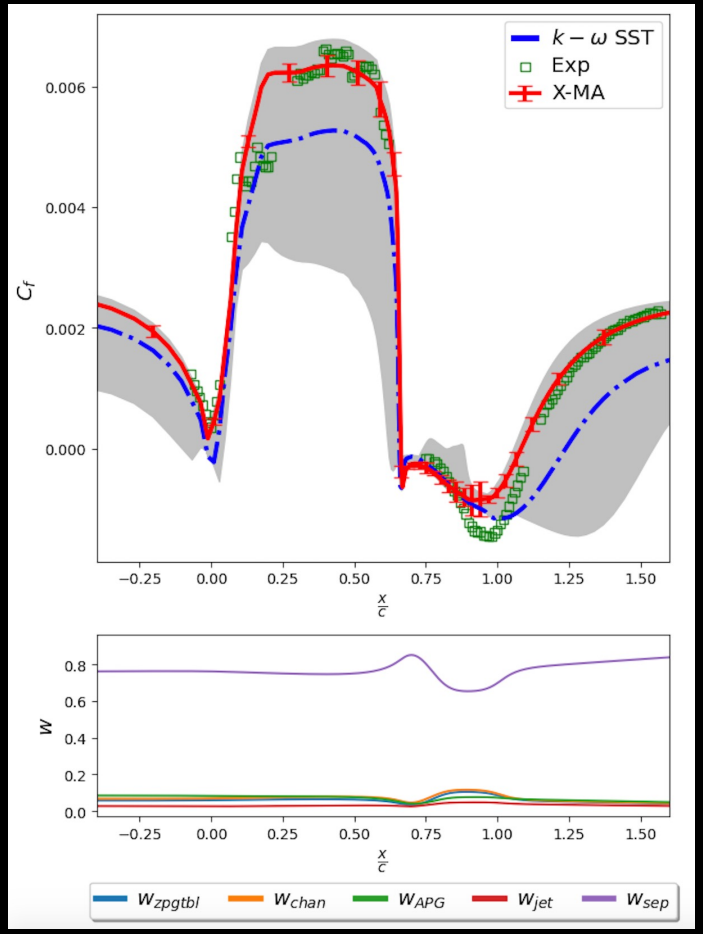
- Application to Test Case 4 2DWMH: 2D NASA Wall-Mounted Hump Separated Flow Validation Case



NASA Turbulence Modeling Testing Challenge

- Application to Test Case 4 2DWMH: 2D NASA Wall-Mounted Hump Separated Flow Validation Case

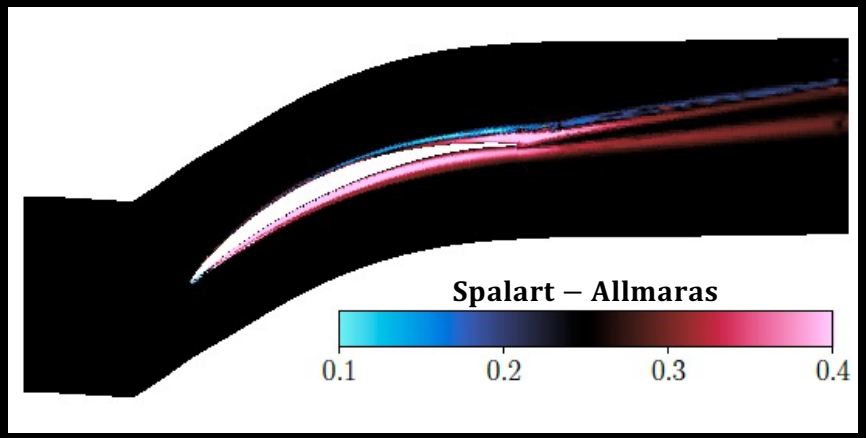


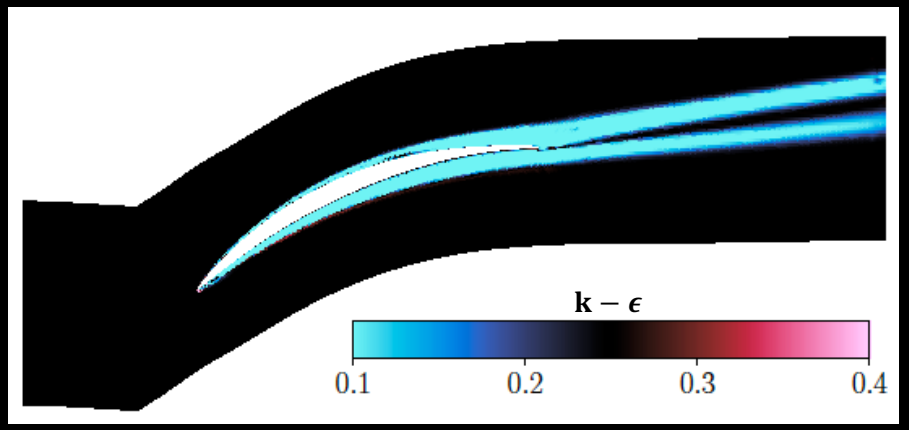


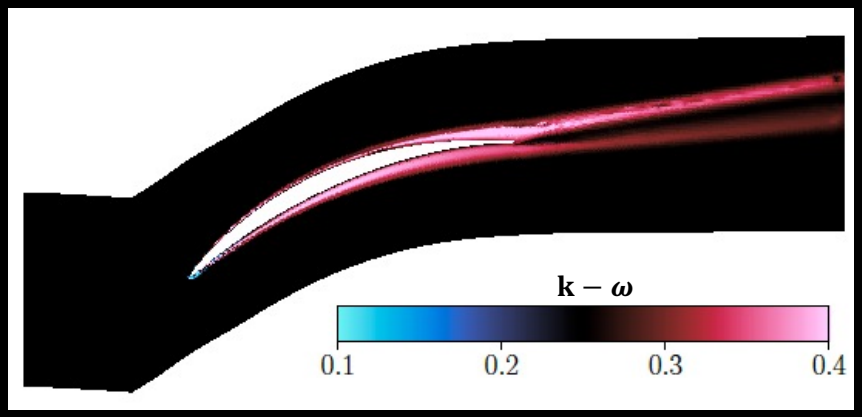
X-MA better than individual models for all cases + improvement over the baseline

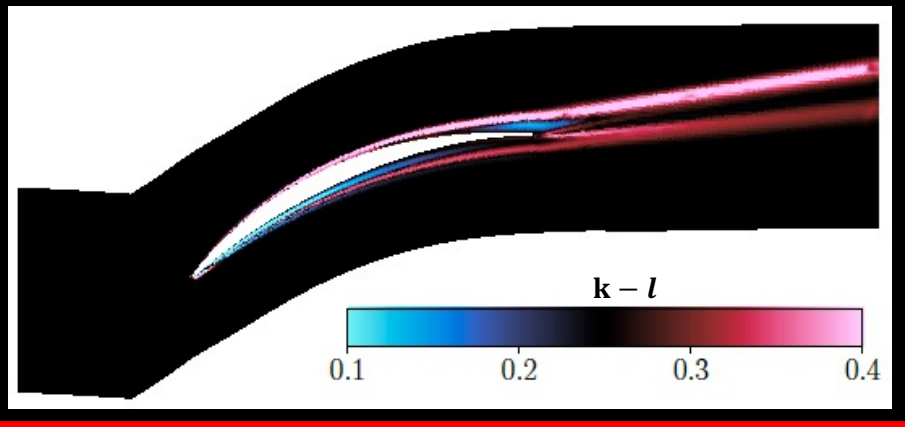
Application to the NACA 65 V103 cascade

- **Experts**: Spalart-Allmaras, $\mathbf{k-\epsilon}$, $\mathbf{k-\omega}$ and $\mathbf{k-l}$ models of turbulence
- Training: synthetic total pressure data generated from k-kl EARSM model for S1,S3 and S4; prediction for S2





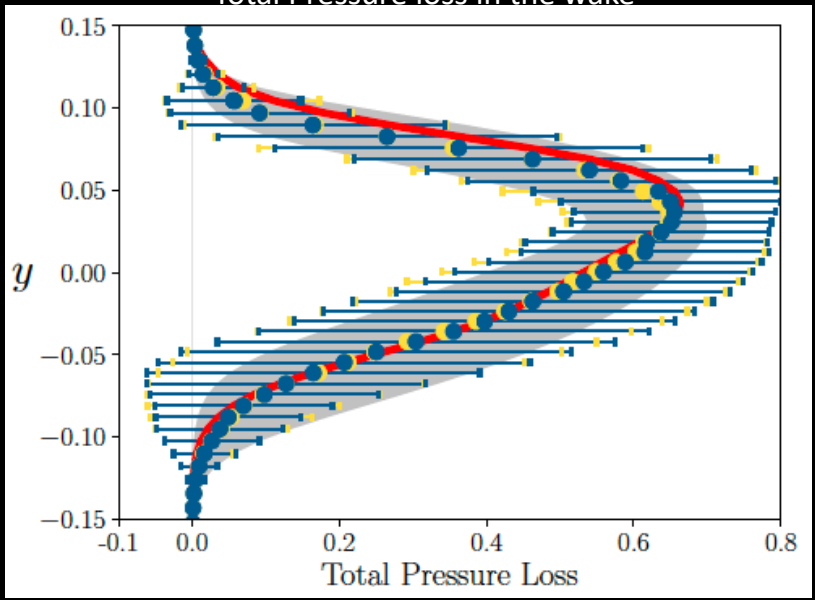




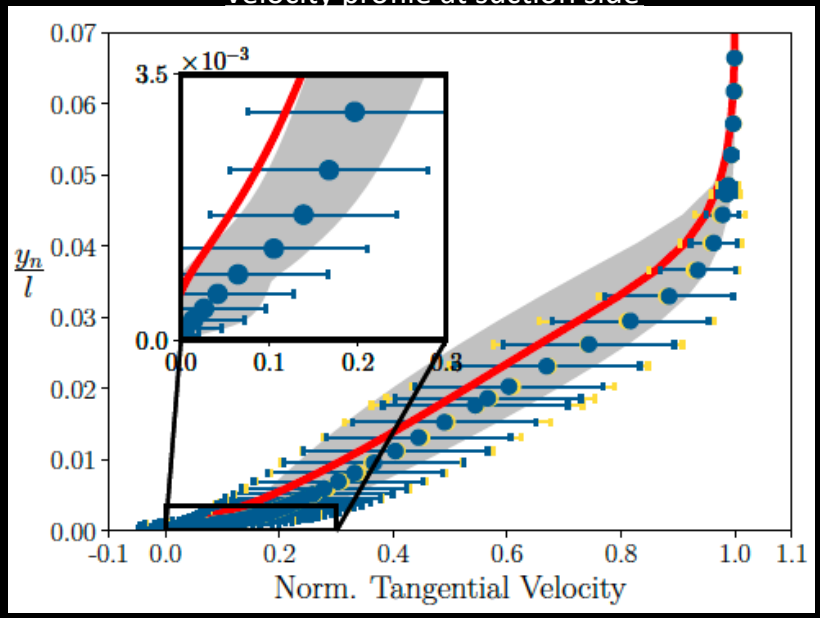
Prediction of selected QoIs for an unseen scenario

- Expectancy and standard deviation computed from local probability mass distribution (model weights)

Total Pressure loss in the wake



Velocity profile at suction side



X-MA consistently favors the best model at each prediction point

Conclusions

- Bayesian framework suitable for
  - updating coefficients associated to turbulence models
  - learning probabilistic models
  - averaging concurrent models according to local flow features
  - providing estimates of confidence intervals
- Encouraging proof-of-concept obtained for data-driven turbulence modeling
  - application to the NASA turbulence modeling testing challenge
- Future work:
  - Further investigate SBL and other Bayesian machine learning algorithms
  - Application to surrogate modeling/reduced-order modeling
  - Optimal choice of features?
  - How to enforce physical constraints in priors/likelihood/weighted reconstruction?
  - How to perform efficient model-in-the-loop learning?

Acknowledgements

- Bayesian learning



Soufiane Cherroud
DynFluid, ENSAM



Maximilien De Zordo-Banliat
DynFluid, ENSAM



Cécile Roques
D'Alembert, SU



Xavier Merle
DynFluid, ENSAM



Grégory Dergham
Safran Tech

- High-fidelity simulations



Xavier Gloerfelt
DynFluid, ENSAM



Aurélien Bienner
DynFluid, ENSAM



Donatella Passiatore
CTR Stanford



Luca Sciacovelli
DynFluid, ENSAM

Questions?