Технологии машинного обучения для задач механики жидкости и газа, моделирования турбулентности (обзор)

Machine Learning Techniques for Data-Driven Fluid Mechanics and Turbulence Modeling: a Review

Sergey N. Yakovenko[1,2], Omid Razizadeh[3]

[1]Khristianovich Institute of Theoretical and Applied Mechanics SB RAS

[2]Department of Physics, Novosibirsk State University

[3] Master Program of Big Data Analytics and Artificial Intelligence

Department of Mathematics and Mechanics, Novosibirsk State University

Content

I. Fluid Mechanics: Brief Introduction

- Main terms, fluid flow states, visualization from natural observations & laboratory experiments
- Governing equations for incompressible flows, hierarchy of methods of their numerical solution
- Examples of eddy-resolving model predictions (movies from high-fidelity numerical simulations)
- Predictions for canonical flows from low-fidelity simulations (RANS), model uncertainty sources

II. Why Data Science, Artificial Intelligence, Machine Learning?

- Motivations to use ML methods for development of computational fluid dynamics (CFD) models
- Review of preceding efforts to apply data-driven ML methods in CFD, and general plan of study
- Examples of ML tools and their implementation in fluid dynamics problems from recent papers

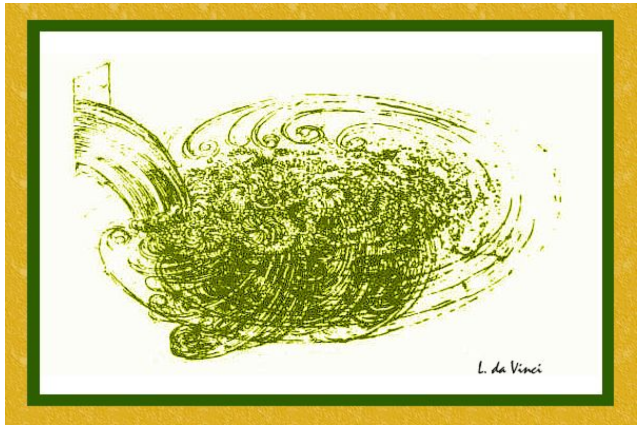
I. Fluid Mechanics: Brief Introduction - 1

- Liquid (water, oil), gas (air, propane, hydrogen), plasma (stars, interstellar clouds)

- Continuous media – particle of fluid is very small, but contains a lot of molecules

- Length scales – from nano-fluids and micro-fluids to astrophysical flow

- Multi-Physics in Fluid Mechanics
  - heat/mass transfer, multiphase flow
  - radiation, electrodynamics
  - molecular physics, astrophysics
  - …



- Methods of fluid flow studies:
  - natural observation
  - laboratory experiment
  - theoretical analysis
  - numerical simulation

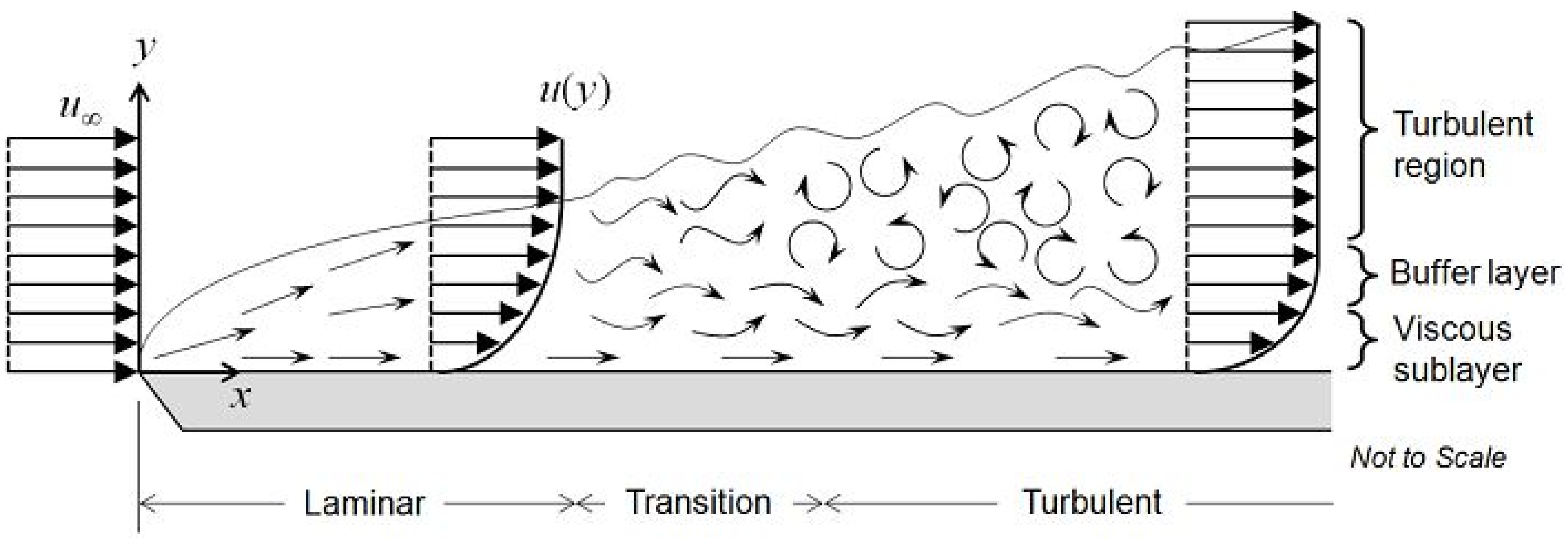
I. Fluid Mechanics: Brief Introduction - 2

- Flow states:

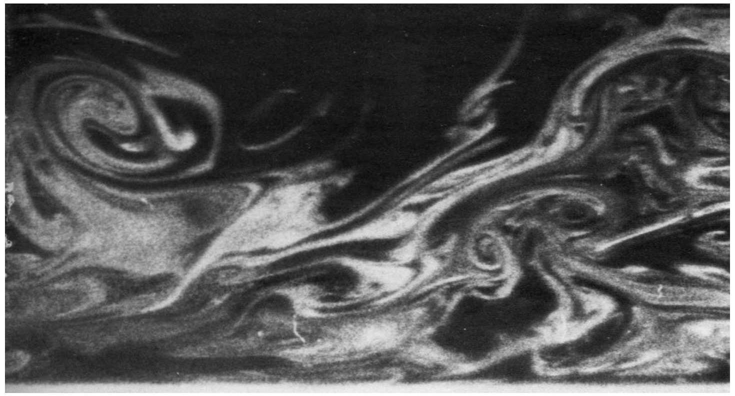
(1) laminar (slow flow, parallel streamlines),

(2) turbulent (chaotic behavior, fluctuations, strong vorticity, cascade of eddies of various scales)

→ boundary layer on a flat plate



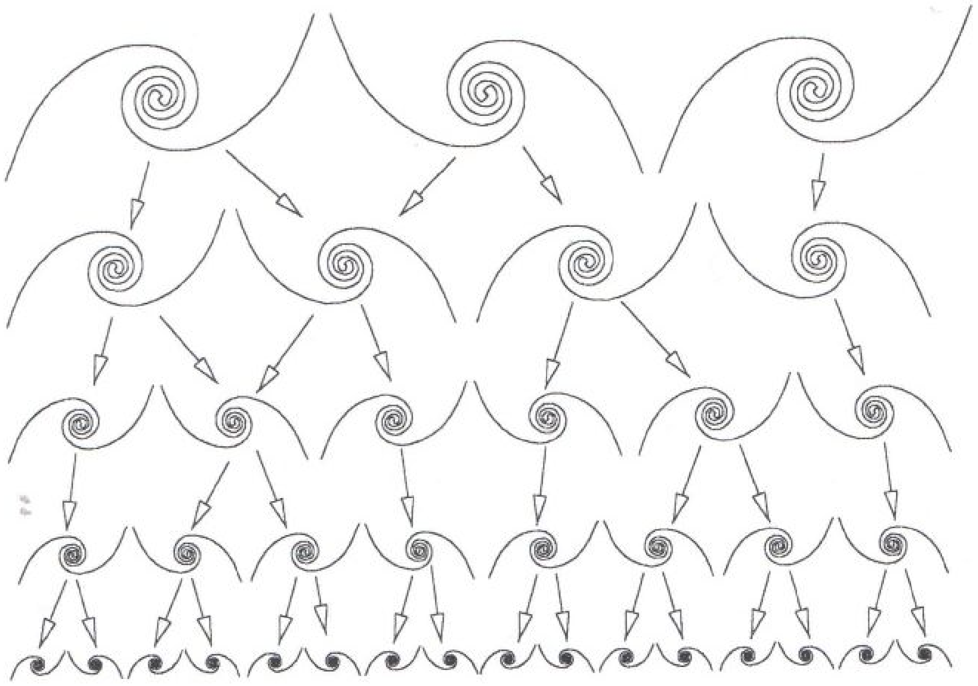
Turbulent eddies: visualization



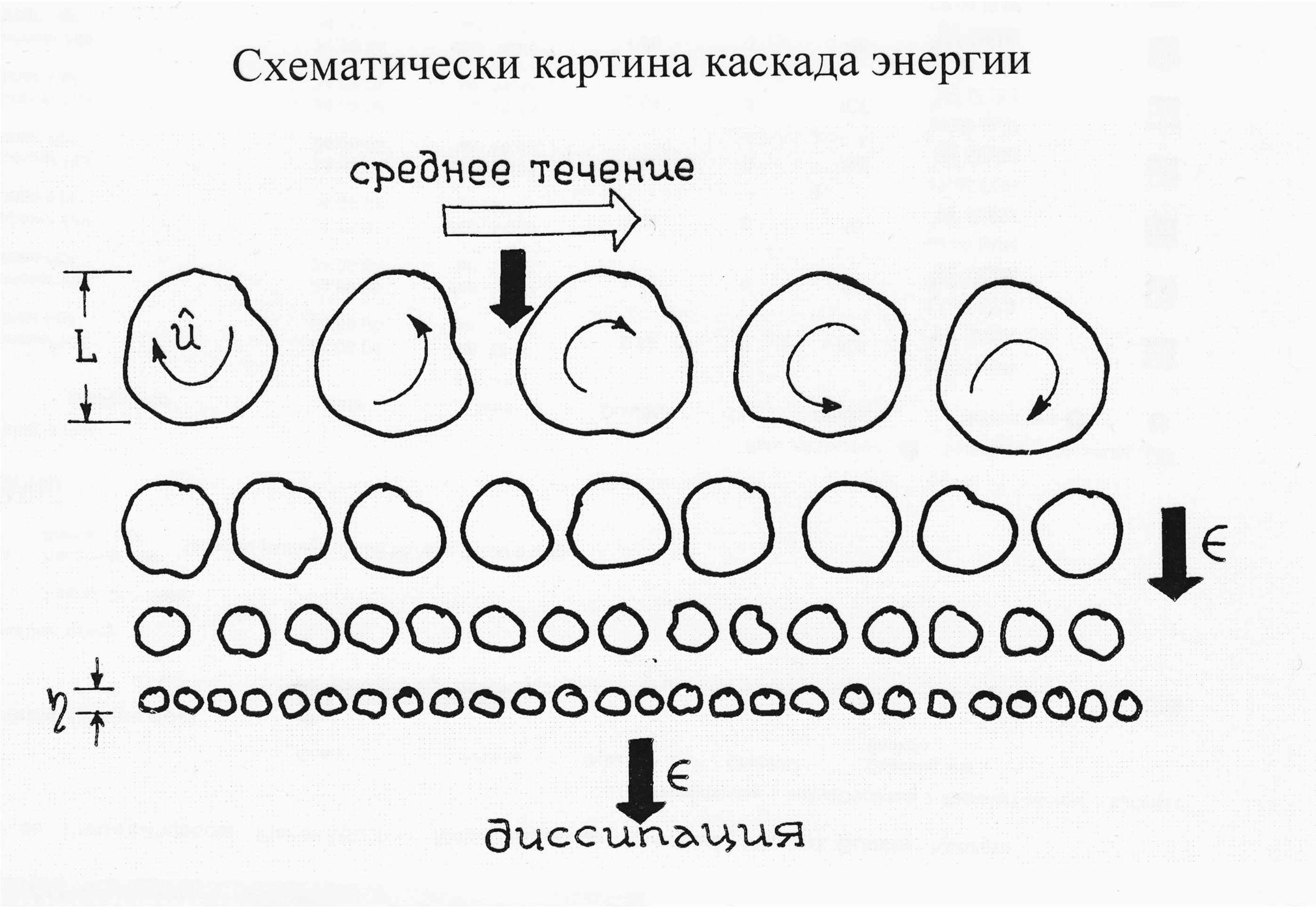
I. Fluid Mechanics: Brief Introduction - 3

- Scheme of cascade of turbulent kinetic energy during interaction of multiple-scale eddies superimposed on each other in a turbulent fluid flow

Поступление энергии



Вязкая диссипация



I. Governing equations for incompressible fluid

- Mass and momentum conservation laws – continuity and Navier–Stokes equations – to obtain the velocity vector $u_i = (u_1, u_2, u_3)$ and pressure p :

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2}$$

$\partial u_i/\partial x_i = 0$

- Reynolds number includes viscosity ν and typical velocity and length scales $Re = \mathcal{U}\mathcal{L}/\nu$

- Direct numerical simulation (DNS) is possible at not very high Re numbers for simple geometry

- Reynolds-averaged Navier–Stokes (RANS) – time- or ensemble-averaged equations

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left[-\bar{p}\delta_{ij} + \nu\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \overline{u_i' u_j'}\right]$$

$$u = \bar{u} + u' \qquad \bar{u} = \frac{1}{T}\int_0^T u(x,t)dt$$

- Large Eddy Simulation (LES) – filtered equations

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial}{\partial x_j}(\overline{u_i}\,\overline{u_j}) = -\frac{\partial \overline{p}}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial^2 \overline{u_i}}{\partial x_k \partial x_k} - \frac{\partial \tau_{ij}}{\partial x_j}$$

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i}\,\overline{u_j} \qquad \overline{f} = \int G(\mathbf{x}')f(\mathbf{x} - \mathbf{x}')d\mathbf{x}'$$

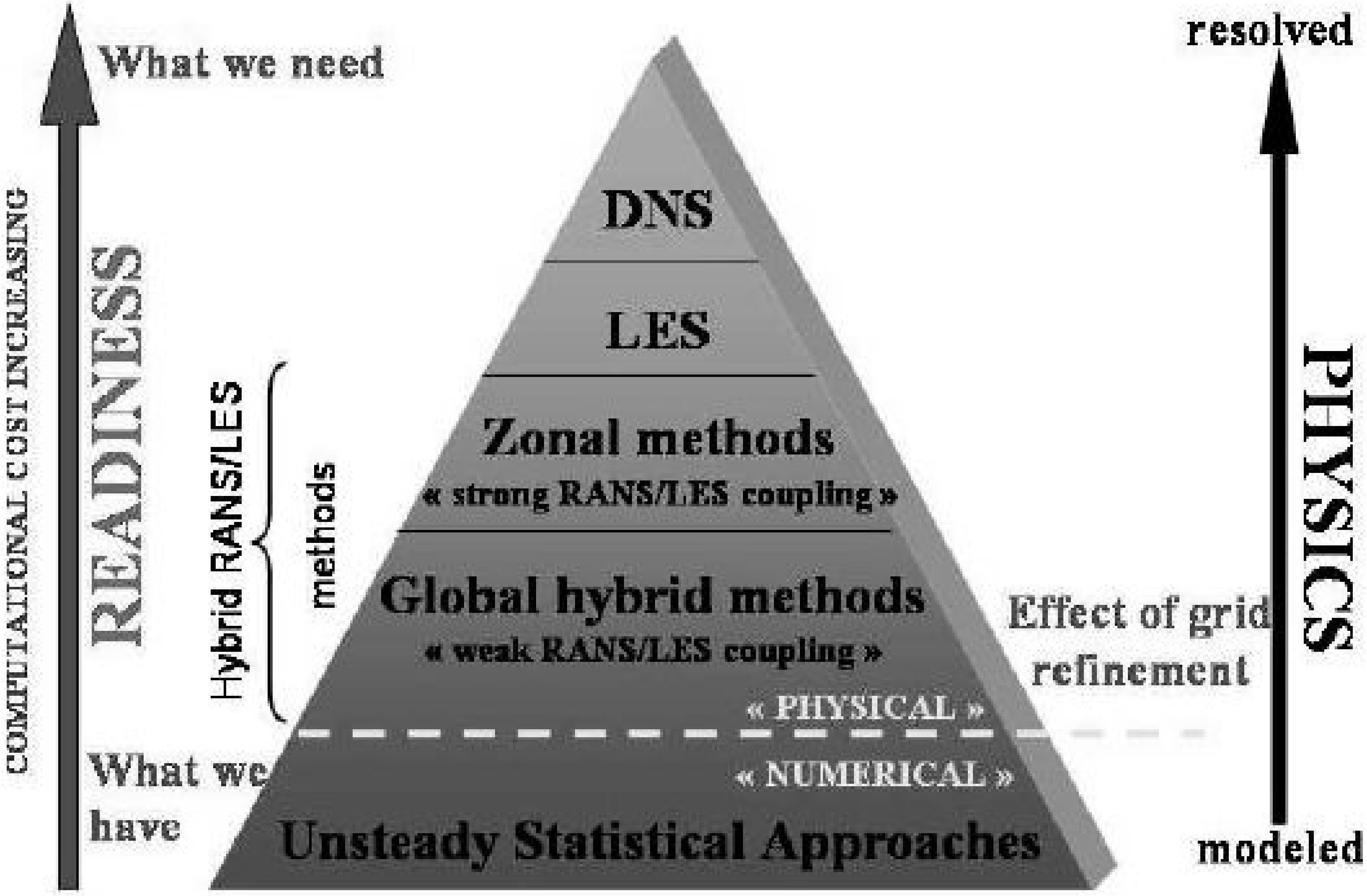
- There are problems to get accurate prediction by turbulence models based on different ways (RANS, LES, hybrid RANS-LES) to close the averaged or filtered Navier–Stokes and continuity equations

Scheme of eddy resolving for turbulence models

Hierarchy of models:

•Eddy-resolving methods (DNS, LES, hybrid RANS-LES) yield **high-fidelity data**, but are more costly

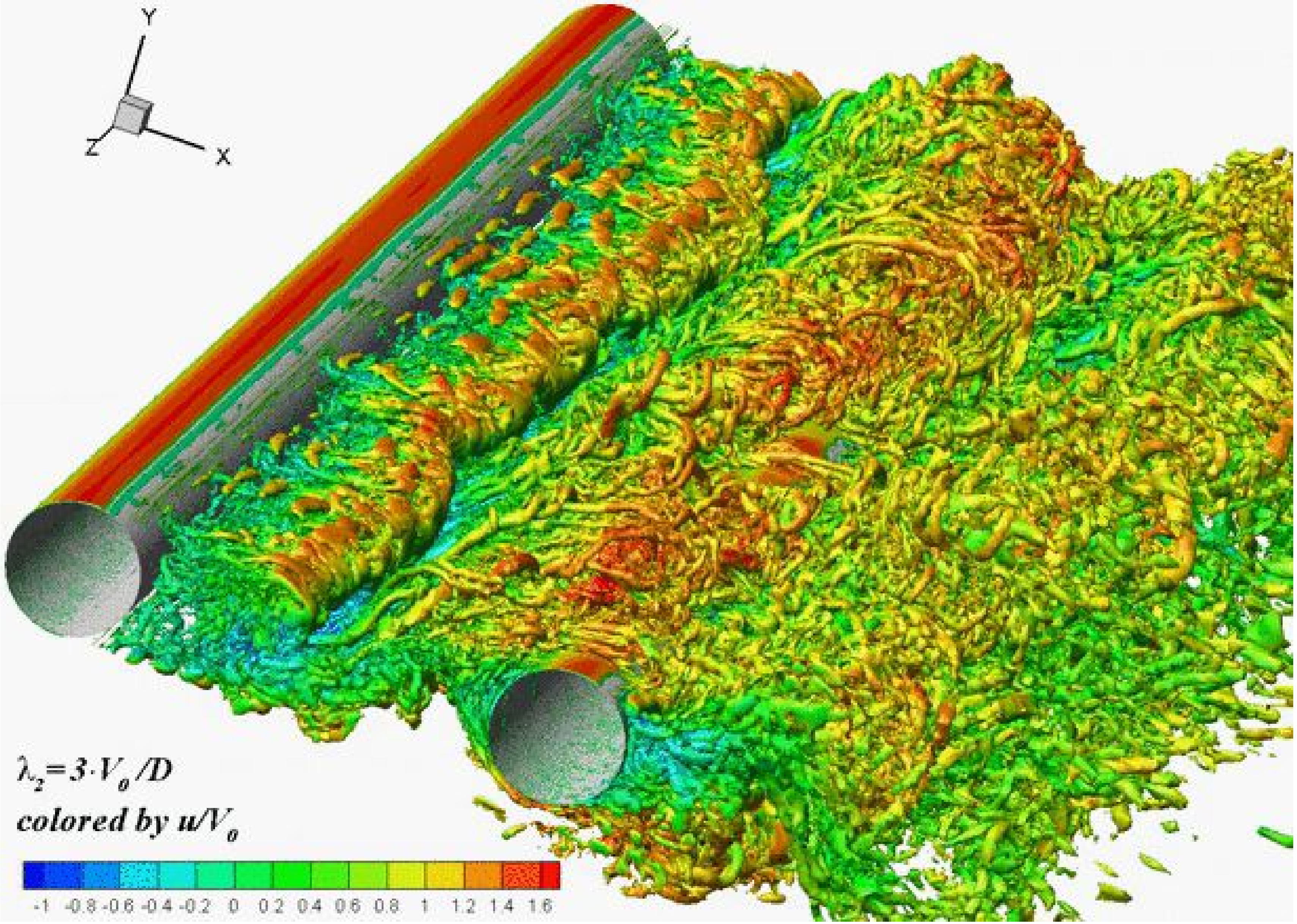
•Computations by steady or unsteady RANS models produce **low-fidelity data**, do not resolve turbulent eddies, however have lower computational costs



Examples of eddy-resolving model predictions

Garbaruk et al. (Computational HydroAeroAcouustics and Turbulence Laboratory, St. Petersburg Polytechnic University):

→ hybrid LES-RANS (DDES) of flow past tandem of cylinders



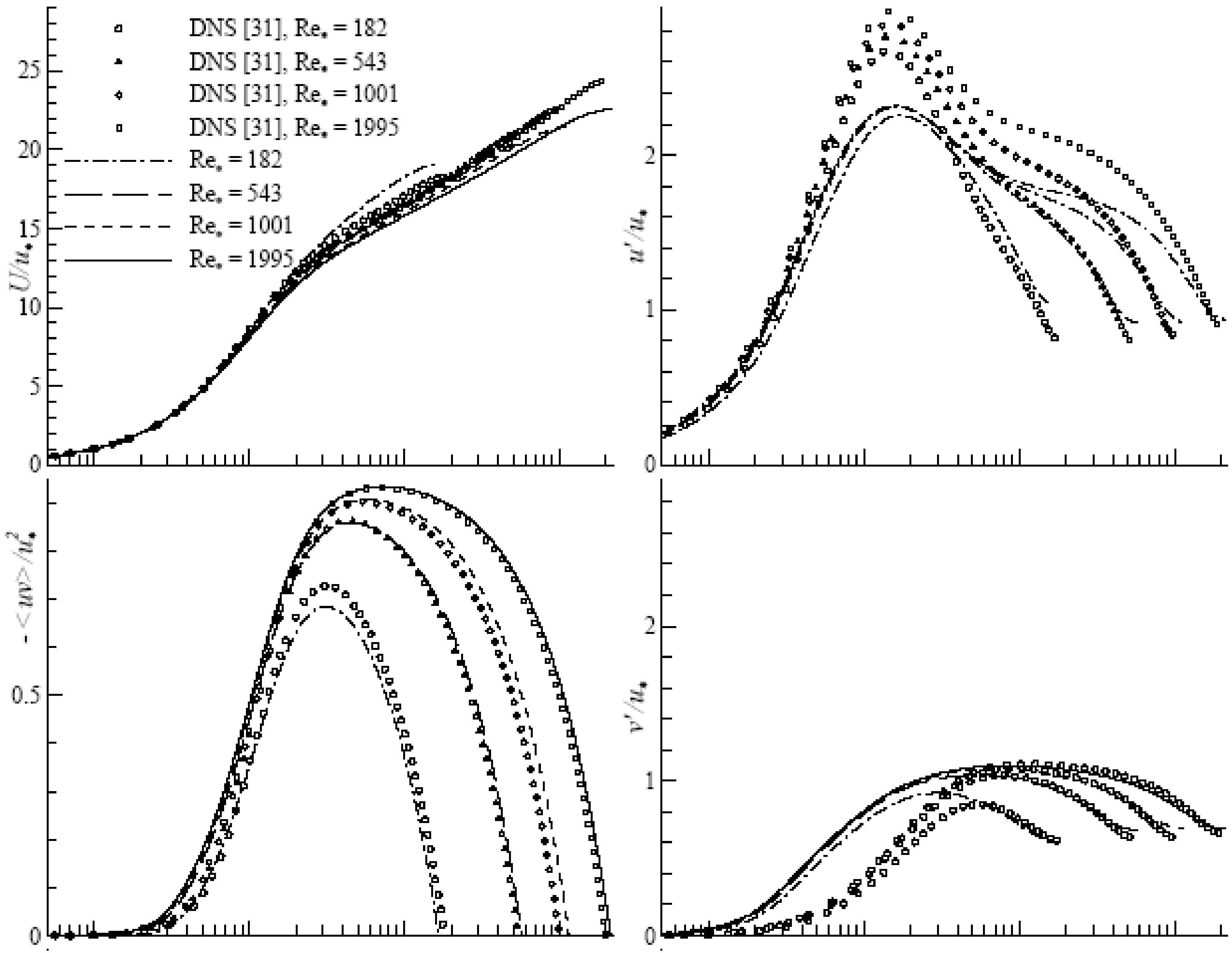
RANS Model Predictions (S.N.Yakovenko, K.C.Chang, AIAA Journal, 2019)

Profiles of velocity field quantities in the fully-developed turbulent flow in a plane channel between two parallel flat walls

at different Reynolds numbers Re_τ (based on friction velocity and channel half-height)

obtained by the advanced RANS model (lines), in comparison with high-fidelity data of DNS (symbols) from the paper

Lee M., Moser R.D. Direct Numerical Simulation of Turbulent Channel Flow up to $Re_\tau \approx 5200$ // J. Fluid Mech. 2015. Vol. 774. P. 395–415



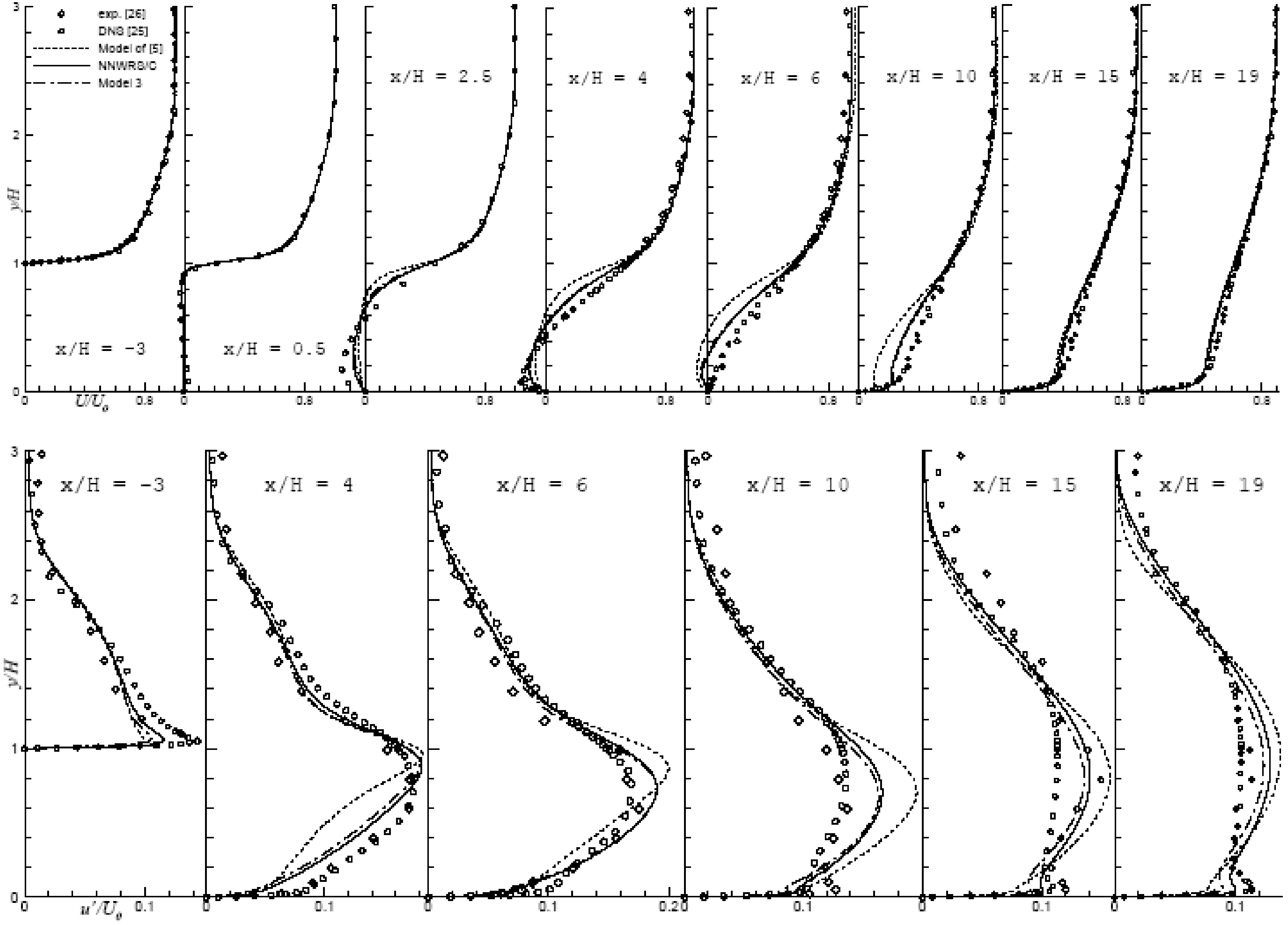
RANS Model Predictions (S.N.Yakovenko, K.C.Chang, AIAA Journal, 2019)

Vertical profiles of velocity field quantities in the turbulent flow in a plane channel with sudden expansion (i.e. with rectangular backward-facing steps located symmetrically on both parallel walls)

at Reynolds number Re = 5100 (based on maximum inlet velocity and backstep height)

obtained by advanced RANS models (lines), versus DNS data (symbols) from

Le H., Moin P., Kim J. Direct Numerical Simulation of Turbulent Flow over a Backward-Facing Step // J. Fluid Mech. 1997. Vol. 330, P. 349–374



TURBULENCE CLOSURE UNCERTAINTIES (XC-2019)

- **Level A:** Dependence on discretization and steps in time and coordinates (mesh properties, numerical schemes, algorithms, solvers) – one needs to get numerics-independent solutions
- **Level B:** Uncertainty associated with initial and boundary conditions – crucial for LES and DNS
- **Level C:** Reynolds-stress (RANS), subgrid-scale-stress (LES) model, including its parameters

- Key review 1 – ***Xiao H., Cinella P. Quantification of model uncertainty in RANS simulations: A review // Progress in Aerospace Sciences. 2019. Vol. 108. P. 1-31.* (XC-2019)**

RANS CLOSURE UNCERTAINTIES (DIX-2019)

- **Level 1:** Application of time- or ensemble-averaging operators <···>, combined with nonlinearity of the Navier–Stokes equations [indicated as $N(\cdot) = 0$], leads to undetermined set of equations, which requires the introduction of modeling assumptions to close the system: $\langle N(\cdot)\rangle \neq N(\langle\cdot\rangle)$
- **Level 2:** To develop closure, a model representation is invoked to relate the macroscopic state to the microscopic state and formally remove the unknowns resulting from the averaging: $\langle N(\cdot)\rangle = N(\langle\cdot\rangle) + M(\cdot)$. Incompressible fluid: $M(\cdot) = \nabla \cdot \tau$ (τ is the Reynolds stress tensor)
- **Level 3:** Once the independent variables are selected, a specific functional form is postulated. Either algebraic or differential equations, denoted as $P(\cdot)$, are typically used to represent physical processes or specific assumptions. Schematically, the model is now $M\,(w;\, P(w)\,)$
- **Level 4:** Finally, given a complete model structure and functional form, a set of coefficients c must be specified to calibrate the relative importance of the various contributions in the closure. Formally the closure is then $M\,(w;\, P(w);\, c)$

→ a RANS prediction of a quantity of interest q is then in general $q = q\,(\,N(\cdot);\, M\,(w;\, P(w);\, c)\,)$

→ Model calibration uses measurement (or high-fidelity simulation) data for q in the same case, and it is assumed that the coefficients c are the source of uncertainty. Therefore, the calibrated model is $\widetilde{\mathcal{M}} = \mathcal{M}(w; \mathcal{P}(w); \tilde{c}_q)$ accuracy is judged by the difference in q

→ This has led to proliferation of model variants and difficulty in assessing predictive capabilities.

II. Why DS, AI, ML ? — Motivations

- Traditional CFD (turbulence) models within RANS and LES frameworks derived *manually* using the physical and mathematical arguments are not universal
- Data from measurement or benchmark solutions of high-fidelity computations in DNS (Direct Numerical Simulation) have historically been used to calibrate engineering CFD models
- RANS models often give large discrepancy versus the data of both measurements and DNS, therefore, improvement of these models is still needed
- Prof. Michael Strelets told (Video-Seminar, 2018) about the opportunity to use the machine learning techniques to obtain automatically new CFD models using the available large data sets and powerful computers, instead of traditional ways of manual development of models
- Novel studies of possibility to use machine learning (ML) techniques to get *automatically* new advanced models using the available large data sets and powerful computers are started in 2013
- Key review – ***Duraisamy K., Iaccarino G, Xiao H. Turbulence Modeling in the Age of Data // Annual Review of Fluid Mechanics. 2019. Vol. 51. P. 357-377.*** **(DIX-2019) →**

→ to study ML methods to systematically inform CFD models with data, quantify/reduce model uncertainties

→ a key perspective → researchers can use data-driven approaches to yield useful predictive models

STATISTICAL INVERSION (DIX-2019)

- Statistical inversion aims to identify parameters ***c*** of model ***M*(*c*)** given data $\boldsymbol{\vartheta}$ with uncertainty $\boldsymbol{\varepsilon_{\vartheta}}$
- Statistical inference is the generalization of the calibration process described above; specifically, uncertainty in the experiments can be directly accounted for, and a potential discrepancy (misfit) between the model prediction $\boldsymbol{\delta}$ and the data is also included.
- The calibration data can include evidence from different sources, while the objective is simply to represent the data. The inference is formulated in a probabilistic setting inspired by the Bayes theorem, and the result is a calibrated, stochastic model:

$$\widetilde{\mathcal{M}} = \mathcal{M}(w; \mathcal{P}(w); \tilde{c}_\theta) + \delta + \epsilon_\theta$$

- Formally, stochasticity is a consequence of uncertainty in the measurements, the prior information on the calibration parameters (for example, the range or the most likely values of ***c***), and the discrepancy function.
- A prior for the discrepancy function is typically left to the intuition of the modeler and is typically represented in a simple mathematical form, for example, by using a Gaussian random field with parameters that are also estimated through the calibration process, i.e., $\boldsymbol{\delta(\vartheta)}$

DATA-DRIVEN MODELING (DIX-2019)

- In the last two decades, the introduction of computationally efficient statistical inference algorithms has led to the possibility of assimilating large amounts of data (e.g., DNS data).
- This has spurred interest in approaches that rely more on the available data than on traditional models; in other words, the emphasis is on $\boldsymbol{\delta}$ rather than on $\boldsymbol{M}$. Different choices for the functional representation of $\boldsymbol{\delta}$ are available, with increasing focus on ML strategies.
- Further work has been devoted toward representing the discrepancy $\boldsymbol{\delta}$ in terms of features $\boldsymbol{\eta}$ selected from a potentially large set of candidates. This enables representation of the resulting model in terms of quantities such as the mean velocity gradients, which are likely to be descriptive in a more general context than the one characterized by the available data.
- Furthermore, constraints such as symmetry properties or Galilean invariance can be enforced in the definition of the candidate features.
- In general, data-driven models can then be expressed as $\widetilde{\mathcal{M}} = \mathcal{M}(w; \mathcal{P}(w); c(\theta); \delta(\theta, \eta); \epsilon_\theta)$

Main Research Groups

- **K. Duraisamy et al., Stanford, CA, Ann Arbor, MI, USA (starting from 2013), RANS + ML**

  Tracey B., Alonso J.J., Duraisamy K. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling // AIAA Paper 2013-0259. 2013.

  Parish E.J., Duraisamy K. A paradigm for data-driven predictive modeling using field inversion and machine learning // J. Comp. Phys. 2016. Vol. 305. P. 758-774.

- **J. Ling et al., Livermore, CA, USA (from 2015), RANS + ML**

  Ling J., Kurzawski A., Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance // J. Fluid Mech. 2016. Vol. 807. P. 155-166.

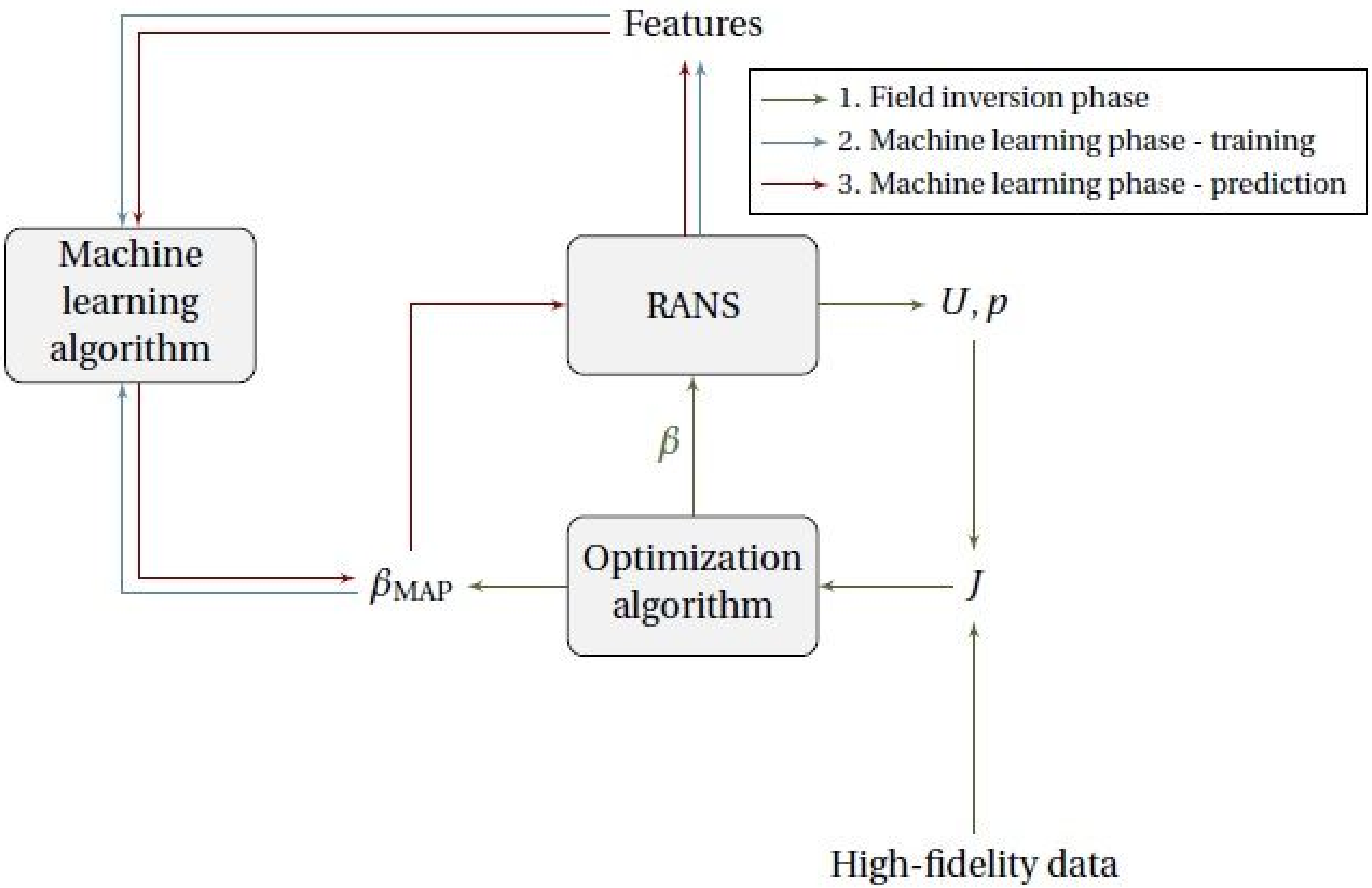
- **R. Sandberg, J. Weatheritt et al., Southampton, UK, Melbourne, Australia (from 2015)**
- **R. Dwight, M. Schmelzer, M. Kaandorp, A. van Korlaar et al., Delft, Netherlands (from 2018)**
- **M. Gawahara, Y. Hattori, Sendai, Japan (from 2017), LES + ML**
- **R. Maulik et al., Stillwater, OK, USA (from 2017), LES + ML**
- **A. Beck et al., Stuttgart, Germany (from 2018), LES + ML**

FIML paradigm flowchart (van Konlaar, 2019)

$\boldsymbol{U}(x,y,t)$, $p(x,y,t)$ – velocity vector and pressure obtained by numerical solution of RANS model equations

$\beta(x,y,t)$ – corrective function in RANS model equations

J – objective function quantifying the discrepancy between low-fidelity data of baseline RANS model and high-fidelity data from dataset



Baseline RANS model versus augmented ML+RANS tool

- Holland (PhD Thesis, 2019): lift coefficient of wind-turbine airfoil S809 (baseline SA model)

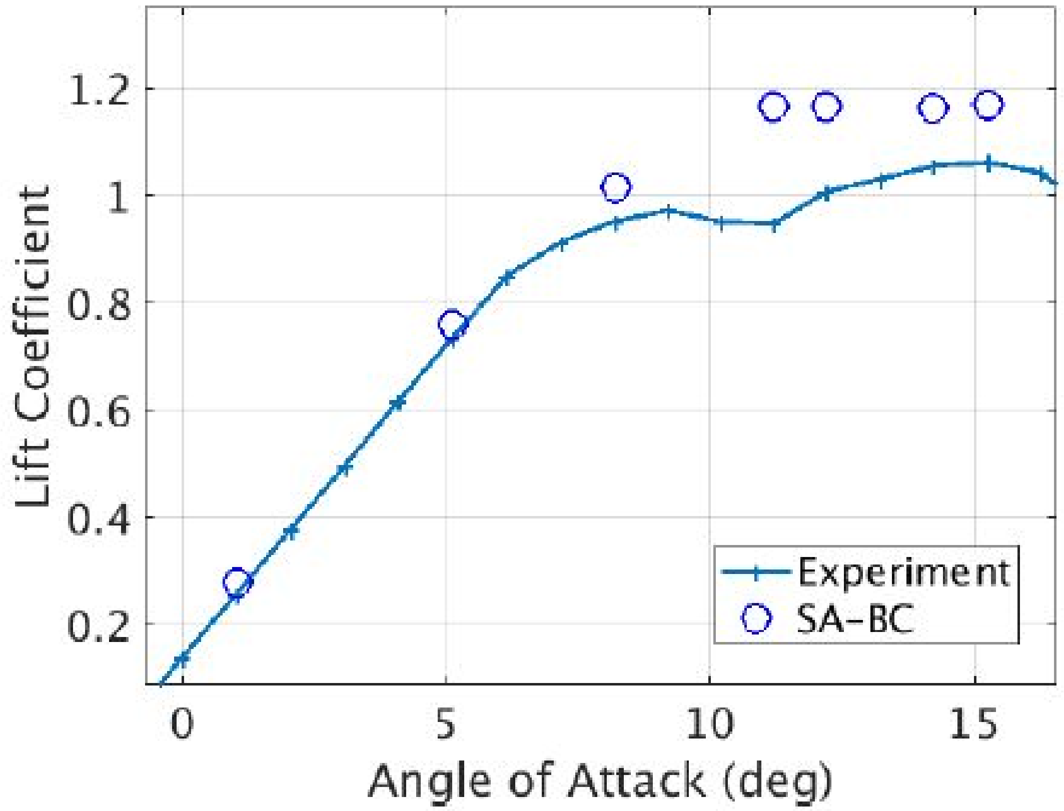


Figure 1.2: Spalart Allmaras Predicted Lift Coefficient for the S809 Airfoil Compared With Wind Tunnel Data.

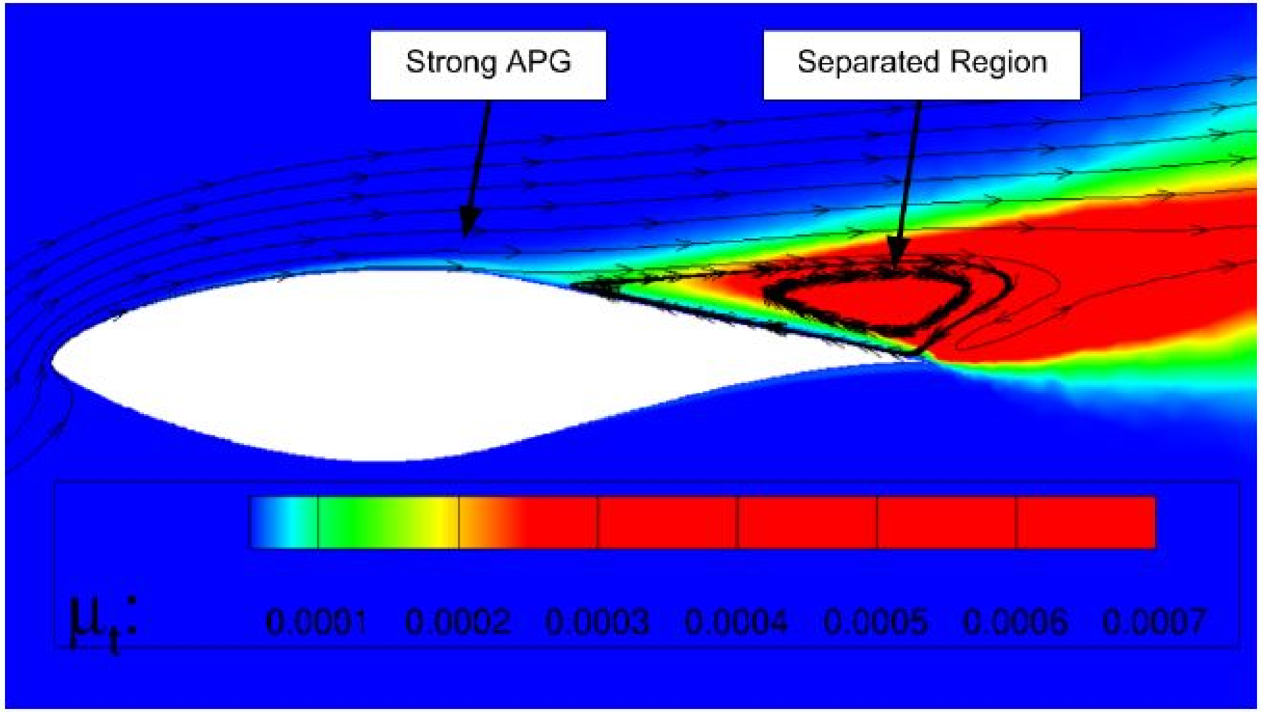


Figure 1.3: Spalart Allmaras Predicted Eddy viscosity for the S809 airfoil at 14.2° angle of attack.

Baseline RANS model versus augmented ML+RANS tool

- Holland (PhD Thesis, 2019): lift coefficient of wind-turbine airfoil S809 (baseline & augmented

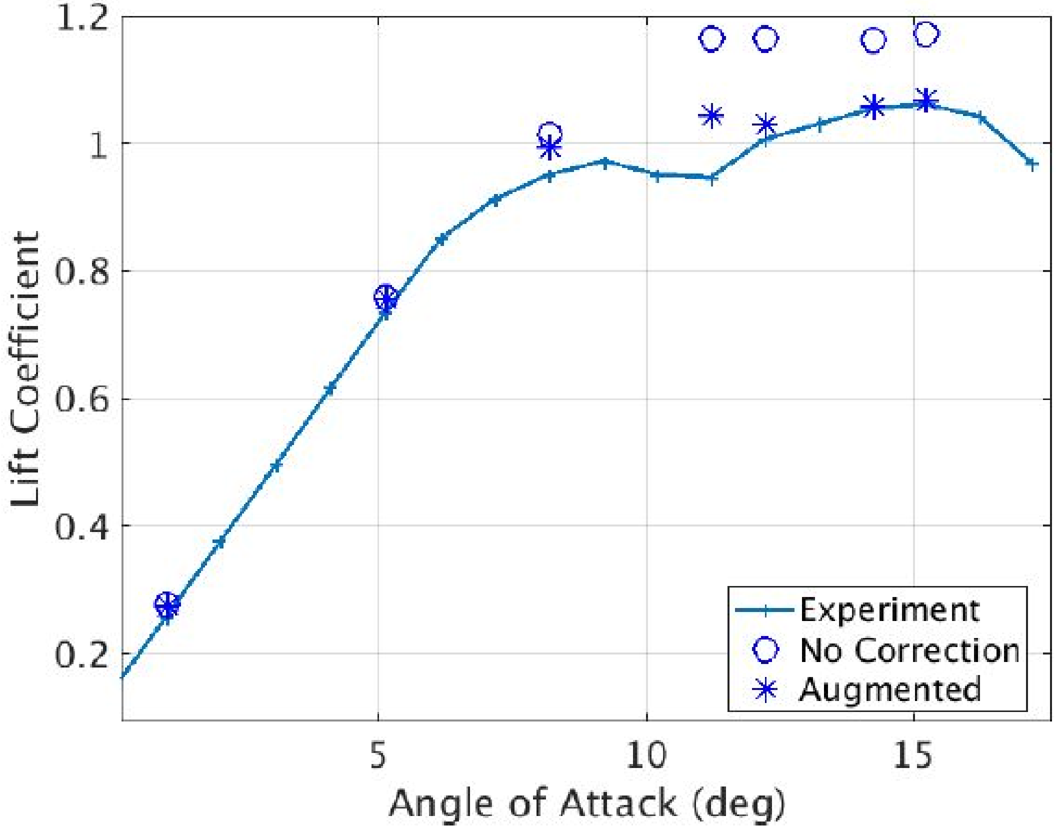


Figure 6.25: S809 augmentation lift performance for training set of all seven angles of attack.

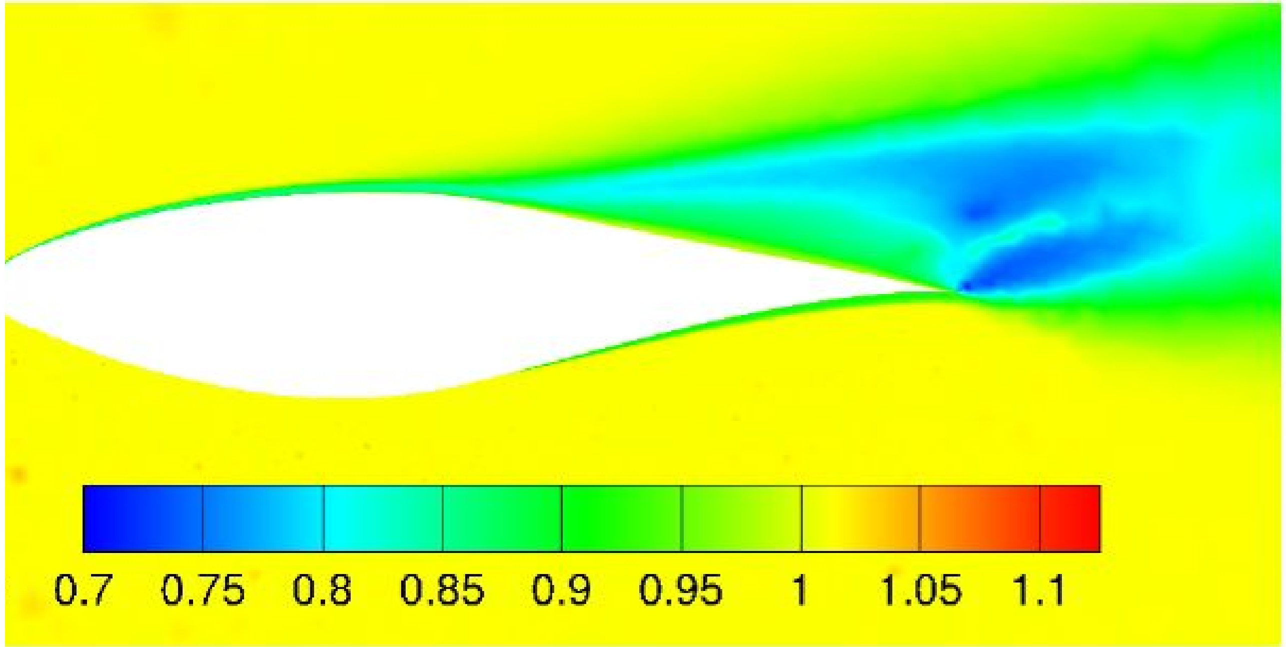


Figure 6.19: Correction field (β) for FIML-Direct with $\lambda = 10^{-5}$.

Baseline RANS model versus augmented ML+RANS tool

- Holland (PhD Thesis, 2019): lift coefficients of airfoils S809 and S814 (baseline & augmented models)

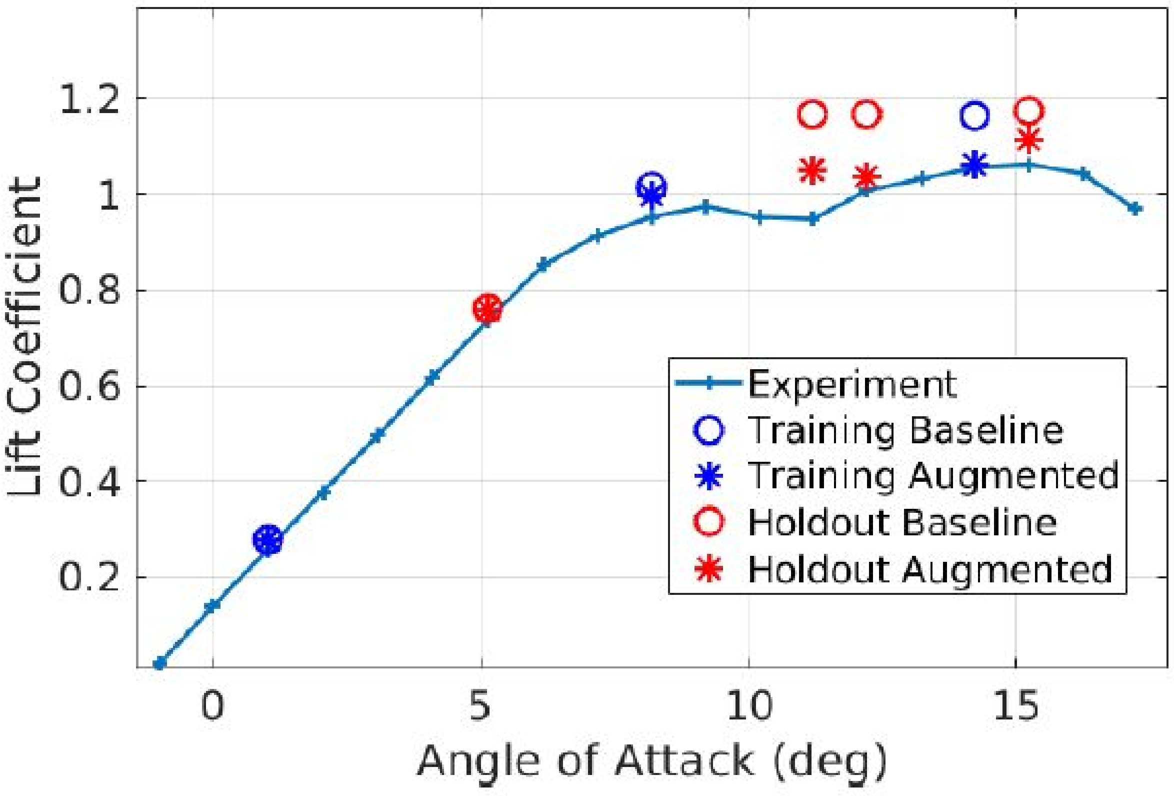


Figure 6.24: Summary of augmentation performance for training and holdout cases for augmentation trained on three angles of attack.

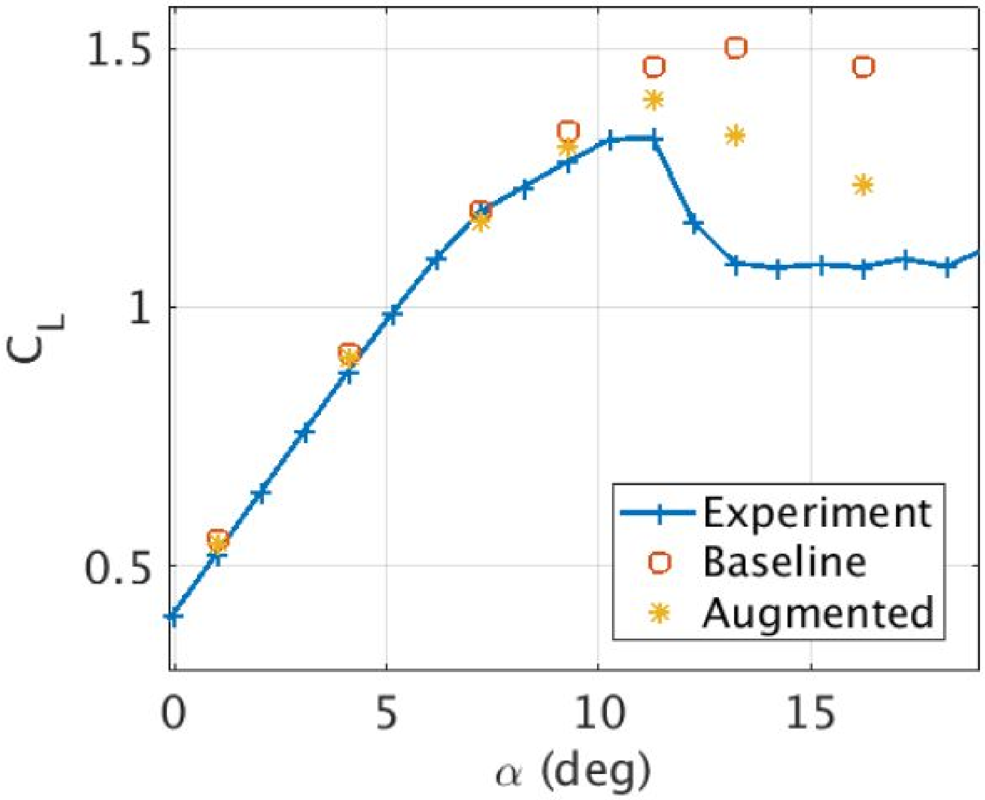


Figure 6.28: S814 C_L resulting from augmentation trained on S809 angles of attack.

the airfoil mesh.

General Plan of ‘ML for CFD’ Studies

- **Stage 1.** Choice of flow test cases, databases and datasets for these cases → high-fidelity data of target solutions (**Y**) for selected test cases

- **Stage 2.** Choice of baseline RANS models, 2D steady-state computations of selected test cases by CFD codes → input features from low-fidelity data of baseline solutions (**X**), grid-independence studies, comparison with measurement data & benchmark solutions

- **Stage 3.** Using the basic and advanced ML methods to reduce deviations (**X – Y**) between low- and high-fidelity data for selected tests, training of models with samples from data sets

- **Stage 4.** Formulation of improved RANS + ML tool, implementation of this tool in CFD codes

- **Stage 5.** Examination of developed RANS + ML tools and baseline models in simulations of new test cases with similar geometry versus high-fidelity data to check predictability of developed tools; extra training of developed RANS + ML tool (if needed)

Stages 1-5 of ‘ML for CFD’ Studies

- **Canonical flow test cases of simple geometries for RANS + ML model training at Stage 1-4:**

→ channel flows with: (a) parallel walls, (b) backward-facing step, (c) periodic hills (used by others)

- **Canonical flow test cases of similar geometry for RANS + ML model verification at Stage 5:**

→ flows with boundary layers, wavy walls, single bumps, cubic obstacles, etc. (used by others)

- **Databases and datasets for these cases, to define target solutions of high-fidelity data (Y):**

→ Johns Hopkins university dataset – DNS of channel flow at $\mathrm{Re}_\tau \leq 5200$ (Lee & Moser, 2015)
Dataset can be installed as a package in Python

→ Database of European Research Community on Flow, Turbulence and Combustion (ERCOFTAC)
Different datasets, including the data of 75 measurements, 13 DNS, 5 highly resolved LES

Stage 2 of 'ML for CFD' Studies

- **CFD codes to produce input features from low-fidelity data (X):**

→ in-house codes (SU2 – Stanford University, Fluidity – Imperial College, …),

→ open source (**OpenFOAM**, Nek 5000, …), ANSYS Fluent/CFX, Star CD/CCM, ...

- **Baseline RANS models to perform 2D steady-state runs of selected test cases for data (X):**

- ***k-ω* SST model** (Tracey et al. 2013, Weatheritt, Sandberg et al. 2015-2019, Schmelzer et al. 2019)

→ **Wilcox *k-ω* model** (Parish & Duraisamy 2016, Kaandorp, 2018, van Konlaar 2019)

→ Spalart – Allmaras one-equation model (Holland et al. 2019)

→ *k-ε* model with linear and quadratic eddy-viscosity models (Ling & Templeton 2015, 2016)

→ ***k-ε* or *k-ω* model with non-linear (cubic) eddy-viscosity models (new?)**

Stages 3-4 of 'ML for CFD' Studies

- **Basic & advanced ML methods to train models with data sets and reduce deviations (X – Y):**

→ Support Vector Machines, Decision Trees, Random Forests (Ling & Templeton 2015)

→ Extended Kernel Regression, Scalar Field Regression, Symbolic Regression, Gaussian Processes (Tracey et al. 2013, Weatheritt & Sandberg 2015, 2016, Parish & Duraisamy 2016)

→ Evolutionary Algorithms, Gene Expression Programming (Weatheritt & Sandberg et al. 2015-2019)

→ Field Inversion and Machine Learning, FIML (Parish & Duraisamy 2016, van Konlaar 2019)

→ FIML-Classic, FIML-Embedded, FIML-Direct (Holland et al. 2019)

→ Fully connected feed forward neural network (NN), multilayer perceptron (MLP), Tensor Basis NN, Convolutional NN, Residual NN (Ling et al. 2016, Gawahara & Hattory 2017, Kaandorp 2018, Beck et al. 2018, Maulik et al. 2019, Holland et al. 2019, …)

Applications of ML Techniques in Fluid Mechanics

Key review 3 – ***Brunton S.L., Noack B.R., Koumoustakos P. Machine Learning for Fluid Mechanics // Annual Review of Fluid Mechanics. 2020. Vol. 52. P. 477-508.*** **(BNK-2020)**

→ ML techniques can extract information from data translated into knowledge about the underlying fluid mechanics

--- extraction of flow features from high-fidelity data (measurements, DNS, LES), post-processing and dimensionality reduction → resulting in reduced-order models, surrogate models (efficiency, real-time work)

→ ML algorithms can augment domain knowledge and automate tasks related to flow control and optimization

→ A powerful information processing framework of ML can augment and transform current lines of fluid mechanics research and industrial applications. The confluence of first principles and data-driven approaches is unique and has the potential to transform both fluid mechanics and ML.

Applications of ML in Fluid Mechanics (BNK-2020)

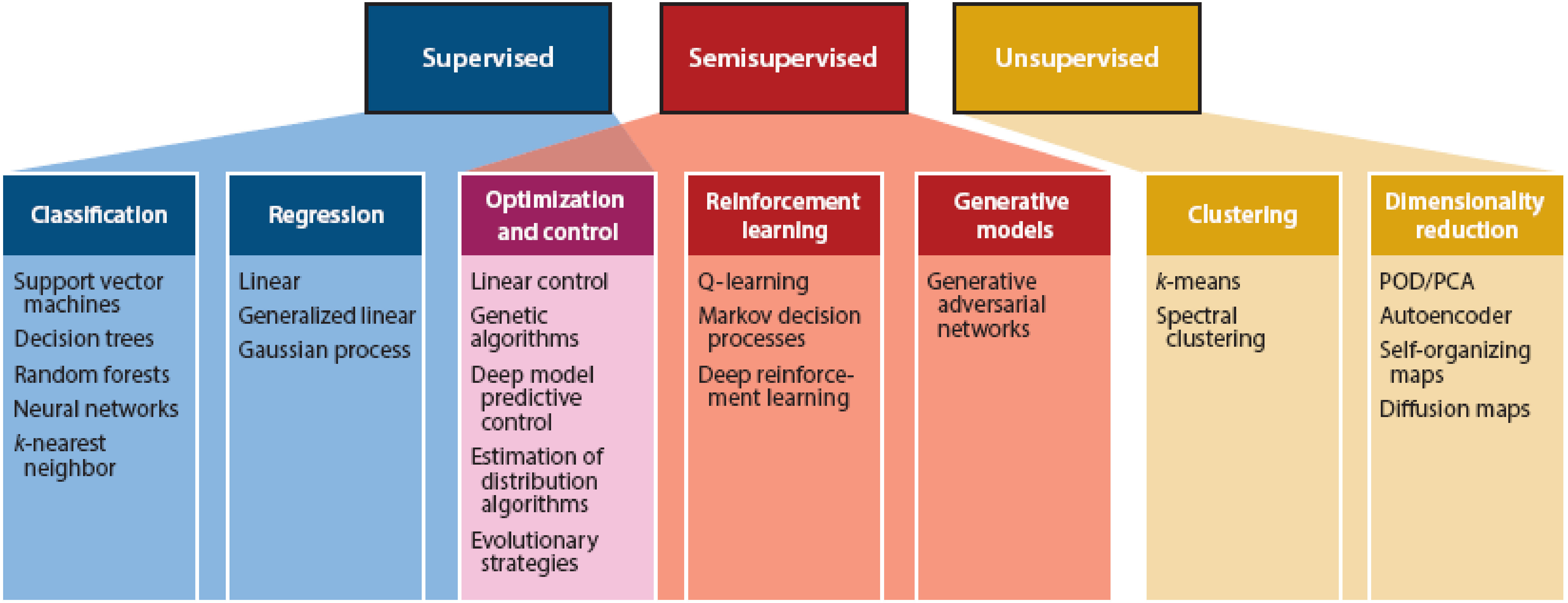


Figure 1

Machine learning algorithms may be categorized into supervised, unsupervised, and semisupervised, depending on the extent and type of information available for the learning process. Abbreviations: PCA, principal component analysis; POD, proper orthogonal decomposition.

What is Machine Learning?

- ML is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead (Wikipedia).
- Машинное обучение — класс методов ИИ, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач.
- The learning problem can be formulated as the process of estimating associations between inputs, outputs, and parameters of a system using a limited number of observations (Cherkassky & Mulier 2007). We distir

- a generator of samples,
- the system in question,
- and LM (Learning Machine),

- Learning process can be summarized

as the minimization of a risk functional:

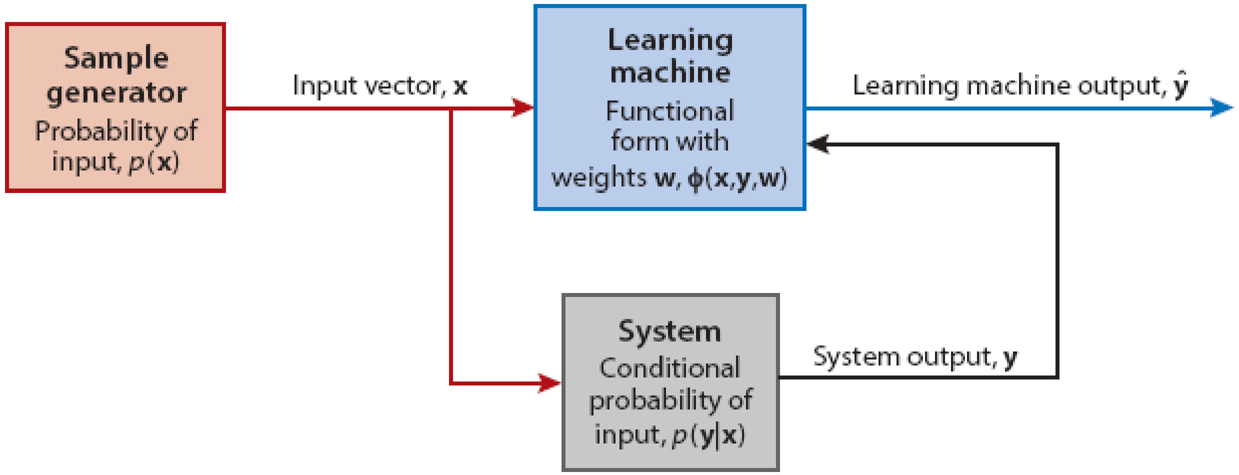


Figure 3

The learning problem. A learning machine uses inputs from a sample generator and observations from a system to generate an approximation of its output. Figure based on an idea from Cherkassky & Mulier (2007).

$$R(\mathbf{w}) = \int L[\mathbf{y}, \phi(\mathbf{x},\mathbf{y},\mathbf{w})]\, p(\mathbf{x},\mathbf{y})\, d\mathbf{x}d\mathbf{y}$$

where the data **x** (inputs) and **y** (output

φ(**x**,**y**,**w**) defines the structure and **w** the parameters of the LM

loss function *L* balances the various learning objectives (e.g., accuracy, simplicity, smoothness, etc.).

Machine learning uses **<u>data</u>** and **<u>answers</u>** to discover the rules behind a problem (Chollet, 2017)

Supervised Learning:

- Both input and desired output data are provided. Input and output data are labeled for classification to provide a learning basis for future data processing.
- The term supervised learning comes from the idea that an algorithm is learning from a training dataset, which can be thought of as the teacher.

- **Classification**

Support vector machines, Decision tree, Random forests, Neural networks, k-nearest neighbor

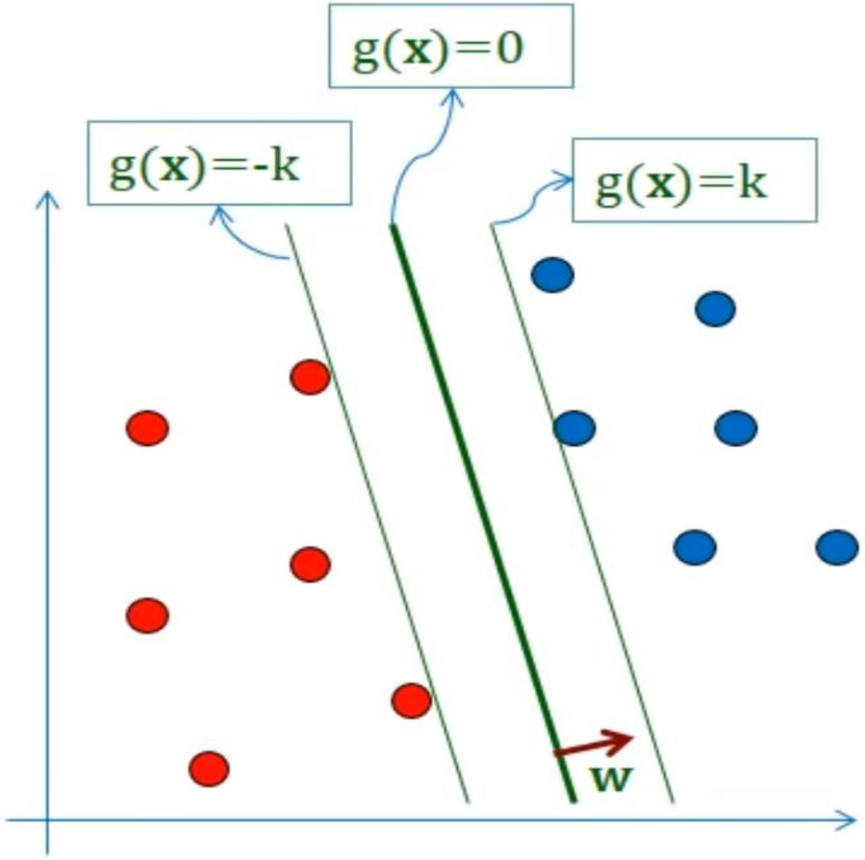
- **Regression**

Linear, Generalized linear, Gaussian Process

- **Optimization and control**

Genetic algorithms, Evolutionary strategies, …

Support Vector Machines:



- for classification or regression problems
- uses a technique called the kernel trick to transform the input data and then based on these transformations it **finds an optimal boundary** between possible outputs

$$g(x) = w^T x + b$$

$$Maximize\ k\ such\ that:$$

$$-\ w^T x + b \geq k\ for\ d_i == 1$$

$$-\ w^T x + b \leq k\ for\ d_i == -1$$

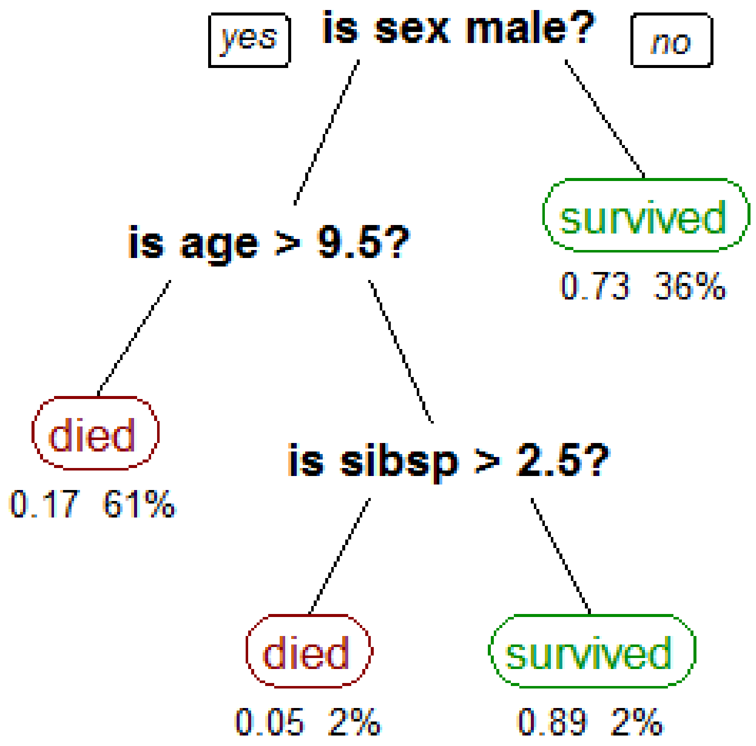
$$Value\ of\ g(x)\ depends\ upon\ ||w||:$$

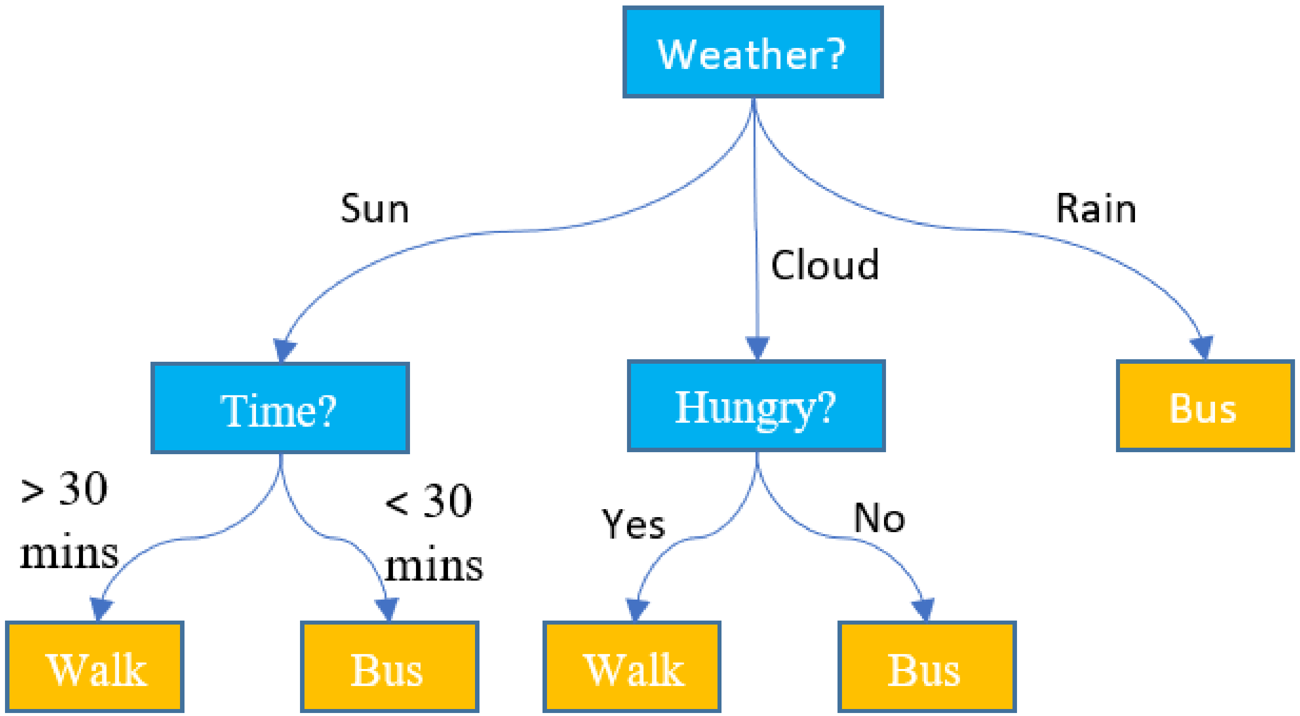
$$1)\ Keep\ ||w|| = 1\ and\ maximize\ g(x)\ or,$$

$$2)\ g(x) \geq 1\ and\ minimize\ ||w||$$

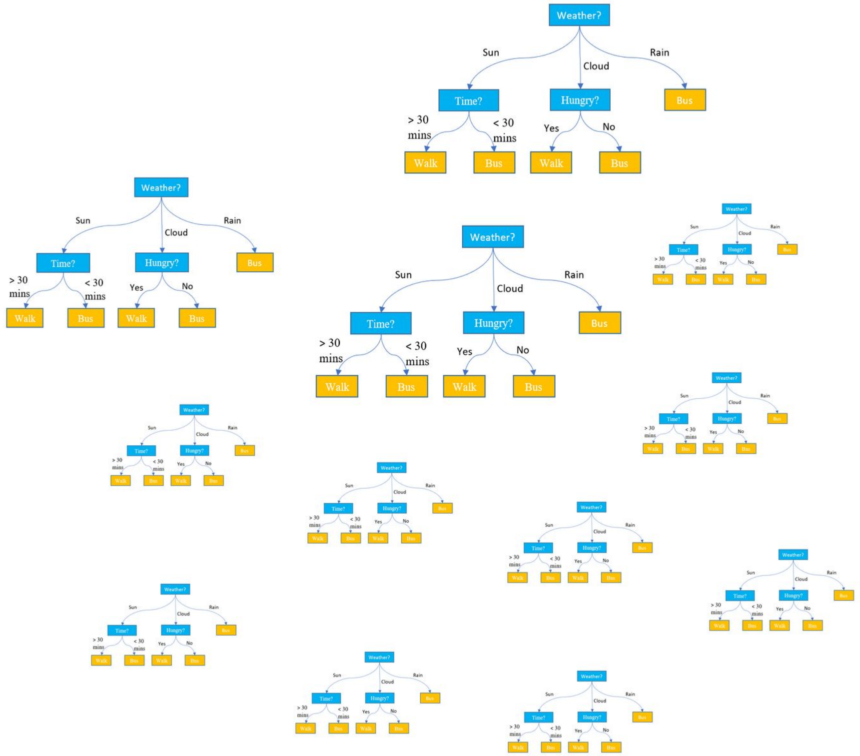
Decision tree:

- The predictive modeling approach
- Decision tree as a predictive model is applied to move from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves)





Random forest:



- collection of decision trees whose results are aggregated into one final result

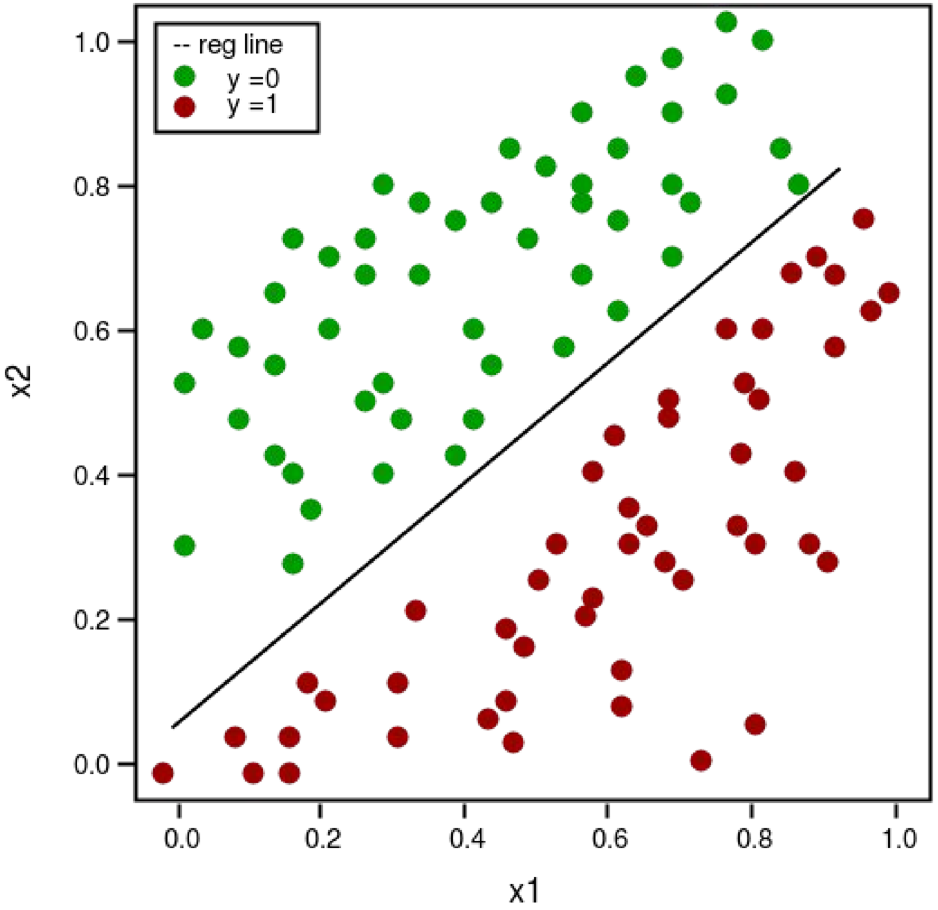
Why random?

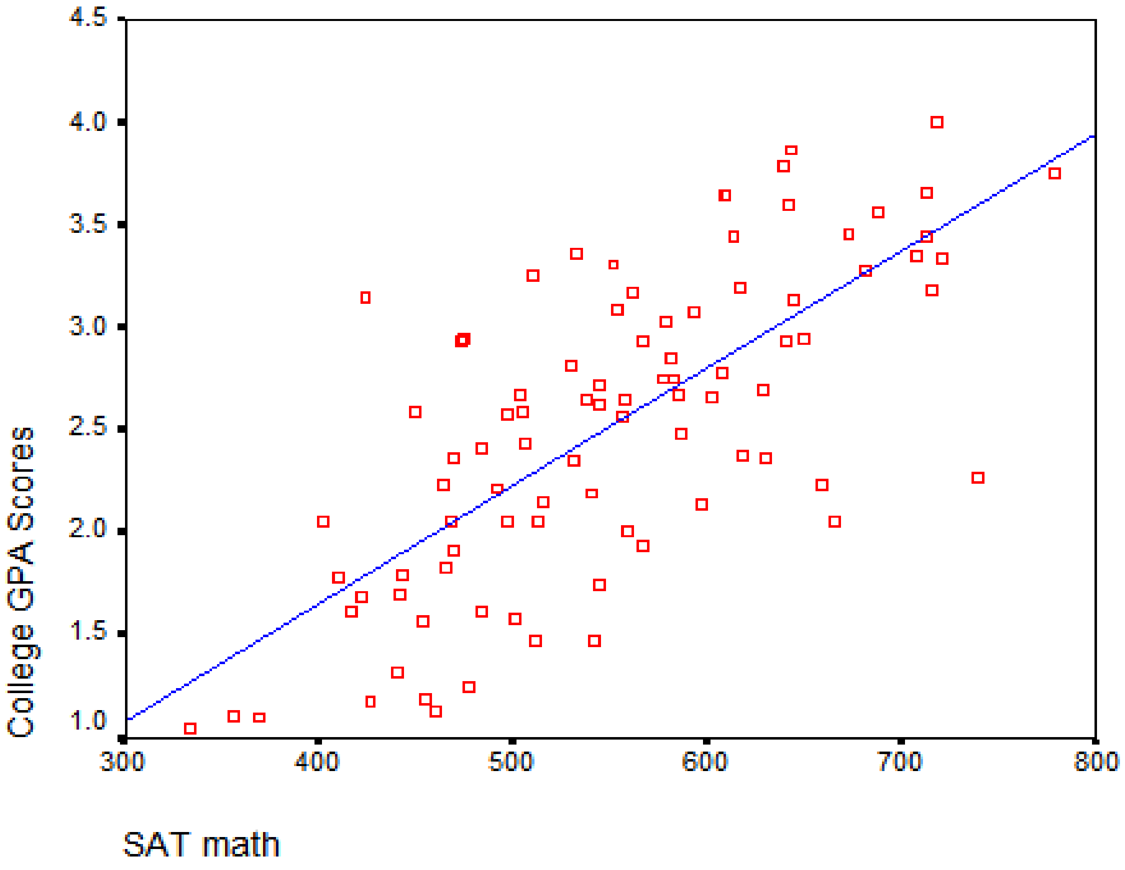
- Each tree is trained on random subsample

- Features for each tree are selected randomly

(Logistic) regression:

- Statistical measurement used in finance, investing, and other disciplines that attempts to determine the strength of the relationship between one dependent variable (usually denoted by Y) and a series of other changing variables (known as independent variables).

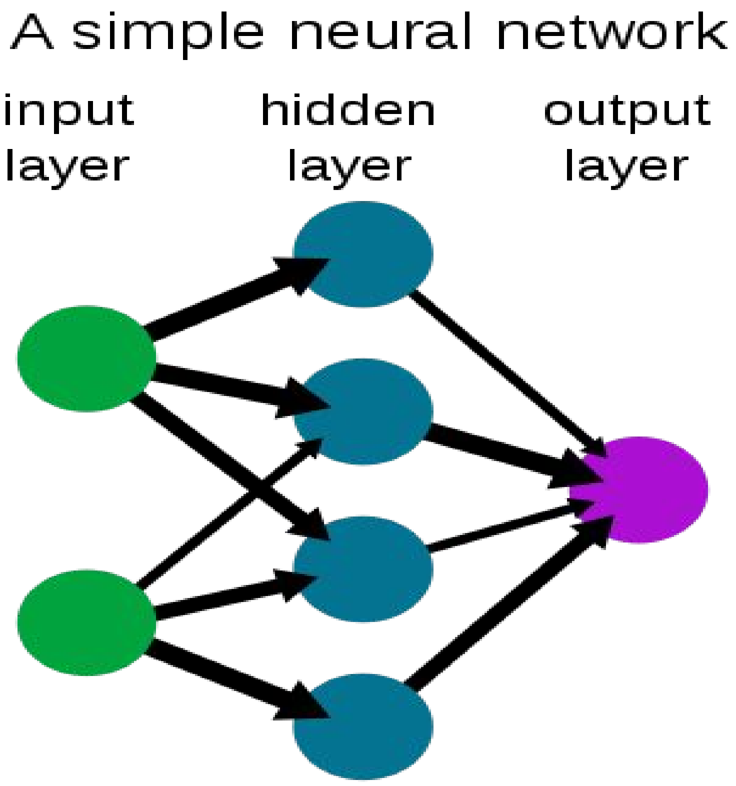




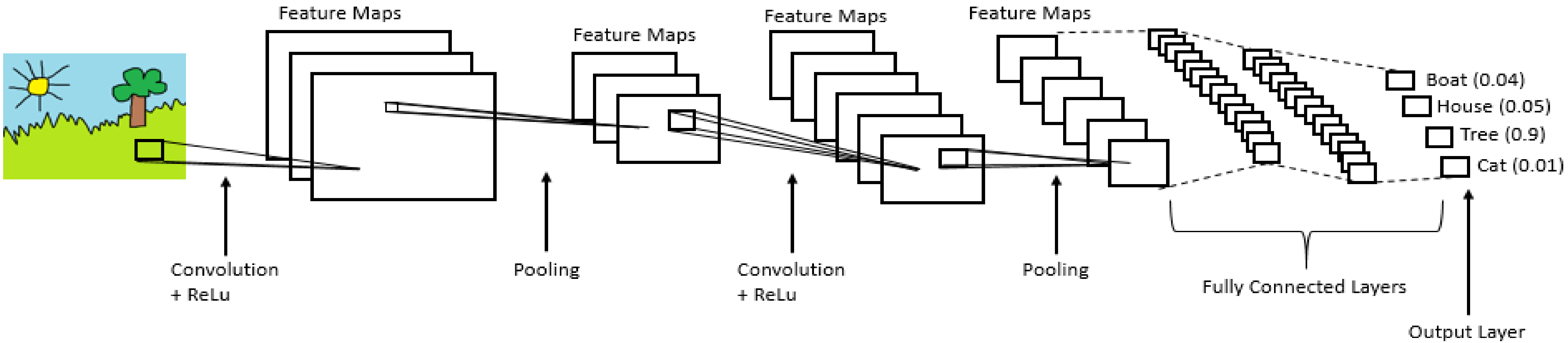
Neural Network (NN):

There are many NN Architectures, for example:

- **Simple (one-layer) Neural Network →**

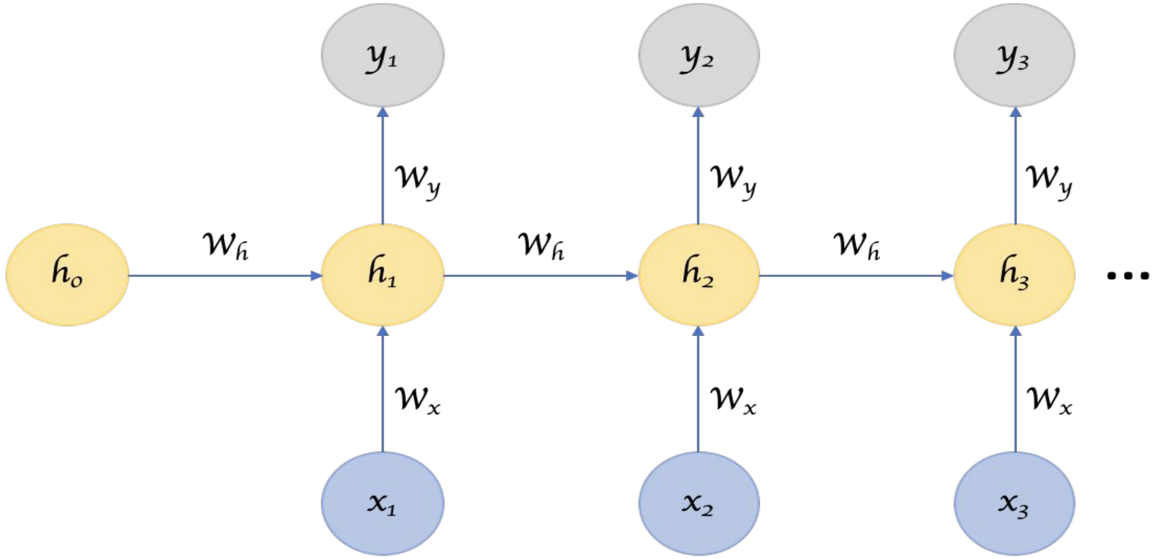


- **Convolutional Neural Networks**



Neural Networks:

- **Recurrent Neural Networks**



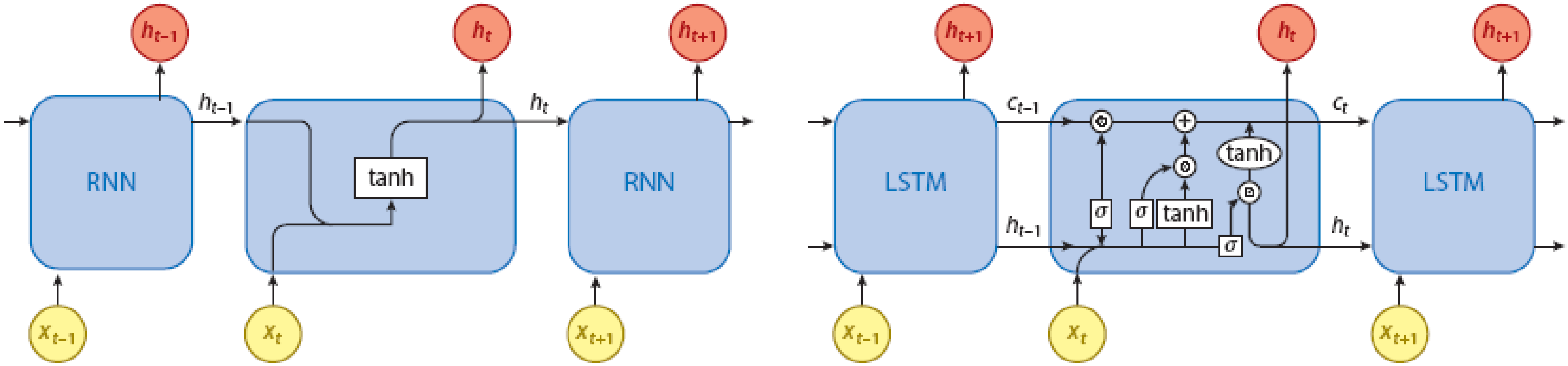


Figure 4

Recurrent neural networks (RNNs) for time series predictions and the long short-term memory (LSTM) regularization. Abbreviations: c_{t-1}, previous cell memory; c_t, current cell memory; h_{t-1}, previous cell output; h_t, current cell output; x_t, input vector; σ, sigmoid. Figure based on an idea from Hochreiter & Schmidhuber (1997).

Tensor-based Neural Network (Ling et al 2016):

- used to find a new closure model for the Reynolds-stress tensor

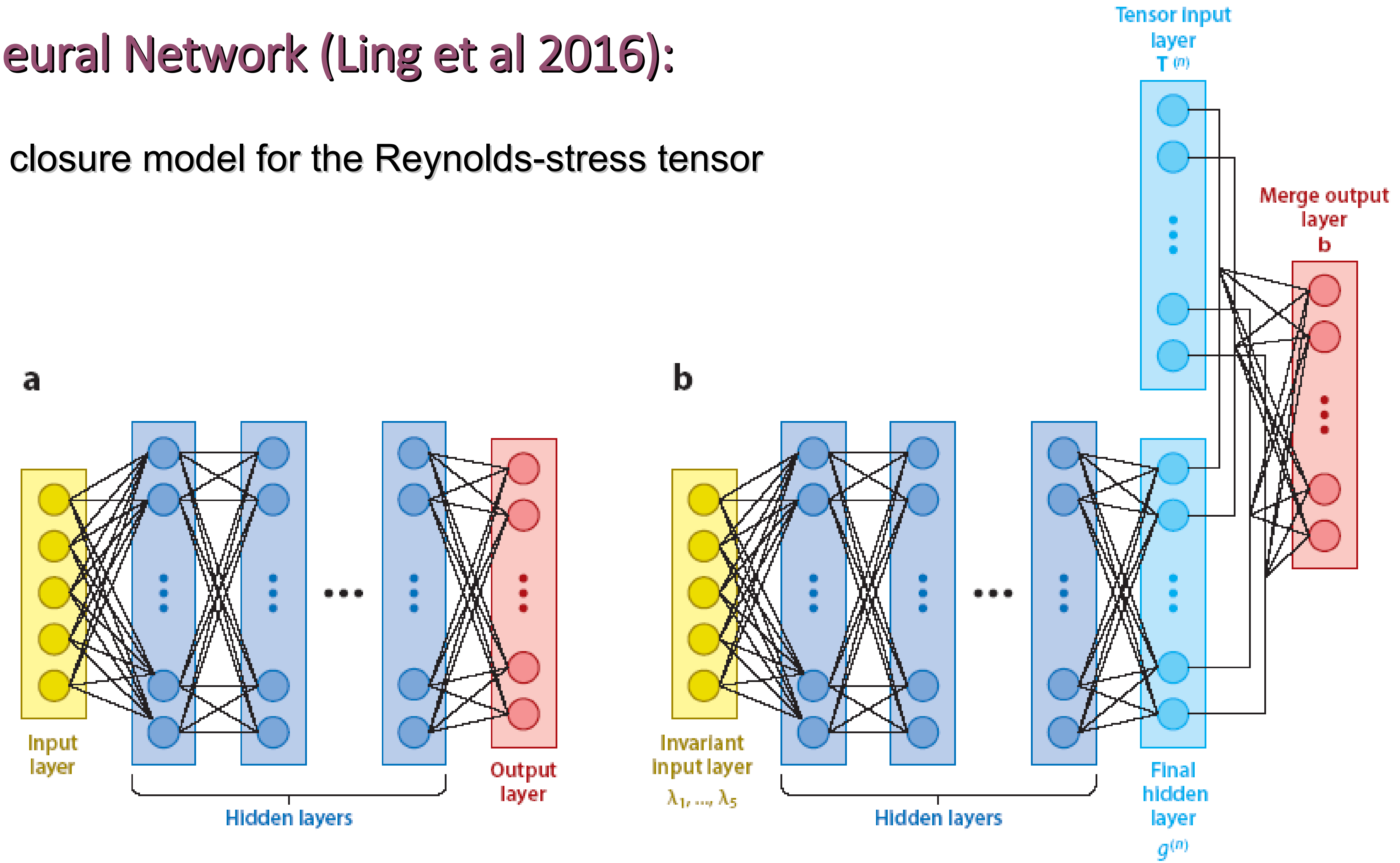


Figure 7

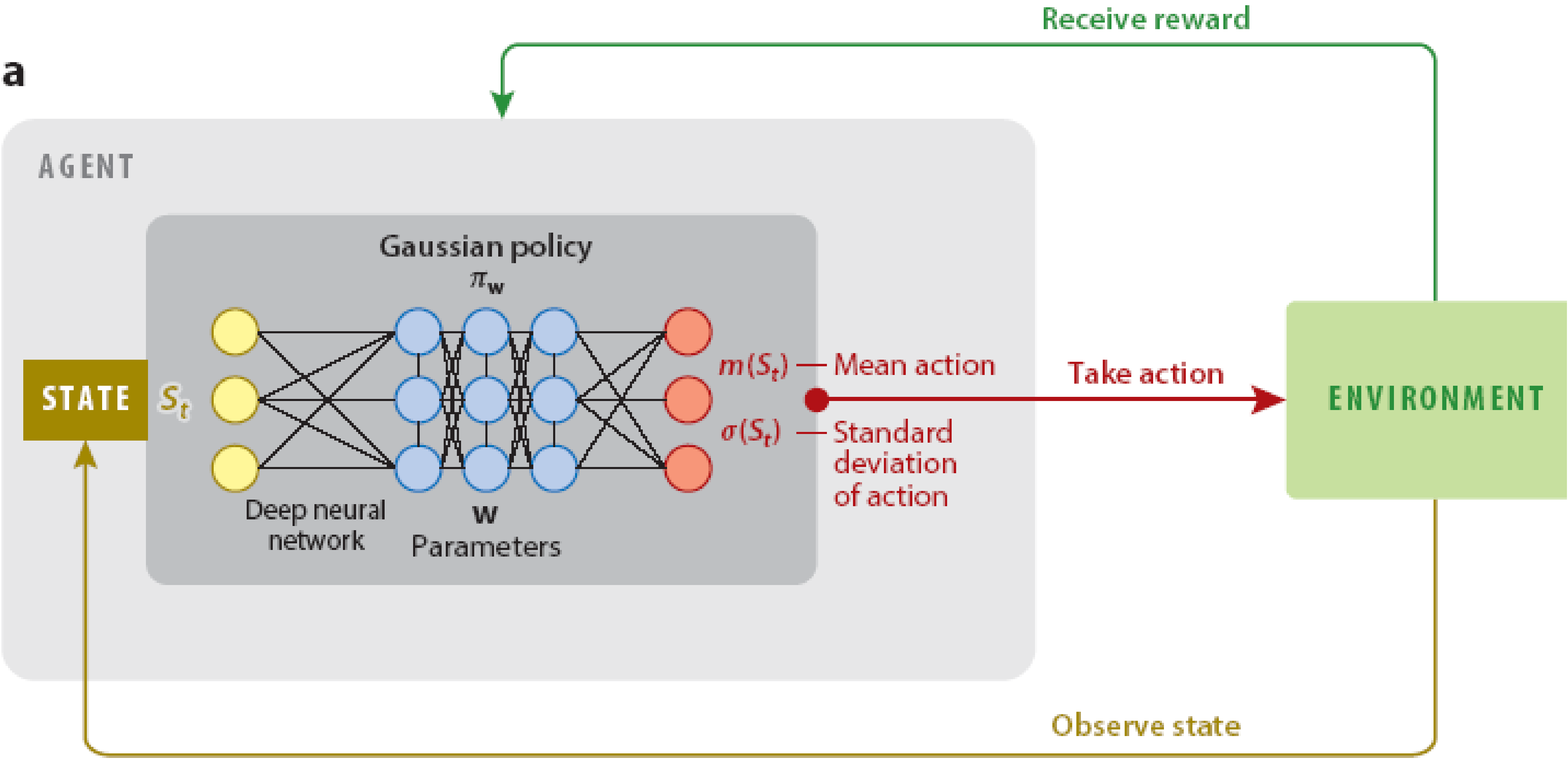
Comparison of standard neural network architecture (*a*) with modified neural network for identifying Galilean invariant Reynolds stress models (*b*). Abbreviations: **b**, anisotropy tensor; $g^{(n)}$, scalar coefficients weighing the basis tensors; $\mathbf{T}^{(n)}$, isotropic basis tensors; $\lambda_1, \ldots, \lambda_5$, five tensor invariants. Figure adapted with permission from Ling et al. (2016b).

Unsupervised Learning:

- Only input data (X) are available without corresponding output variables
- The goal is to model the underlying structure or distribution in the data, in order to learn more about the data
- In contrast to supervised learning, there is no correct answers and no teacher.
- Algorithms are left to their own decisions to discover and extract interested features from data.

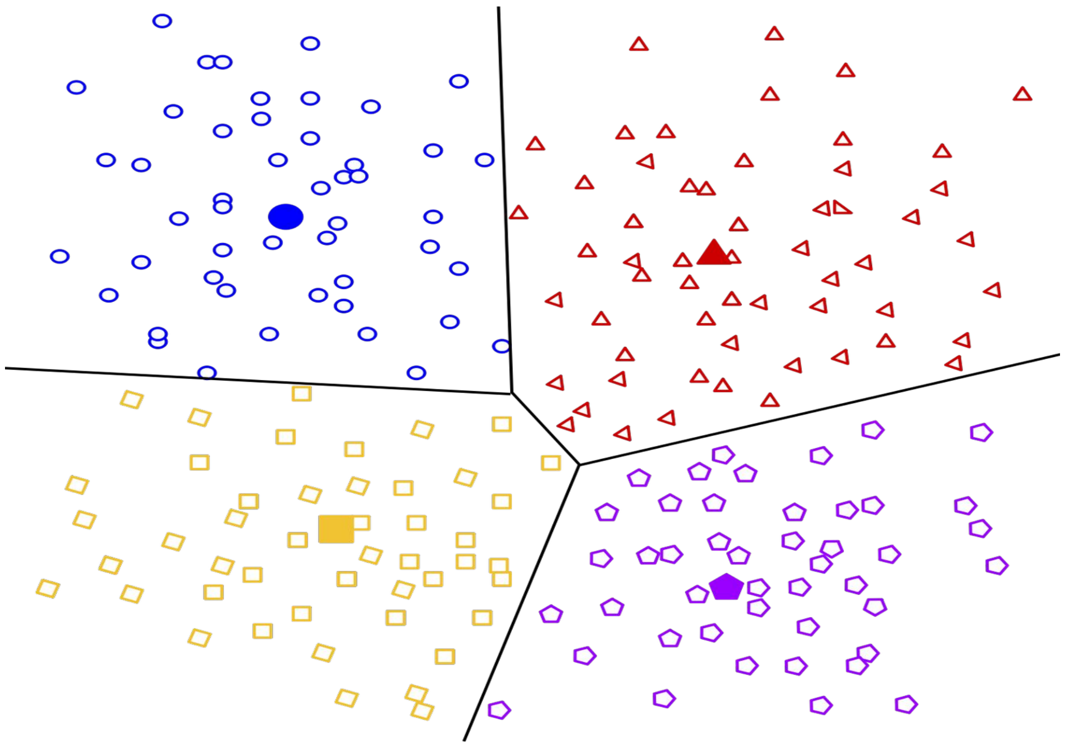
Semi-supervised Learning (Genetic Algorithms, Evolutionary Strategies, Generative Aversarial Network, Reinforcement Learning):

Deep reinforcement learning scheme



Clustering:

- Dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups.
- In simple words, the aim is to segregate groups with similar traits and assign them into clusters.



Dimensionality reduction:

- In statistics, ML, and information theory, dimensionality reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables.
- As a result, the reduced-order models (ROM) or surrogate models are derived, similar to POD/PCA

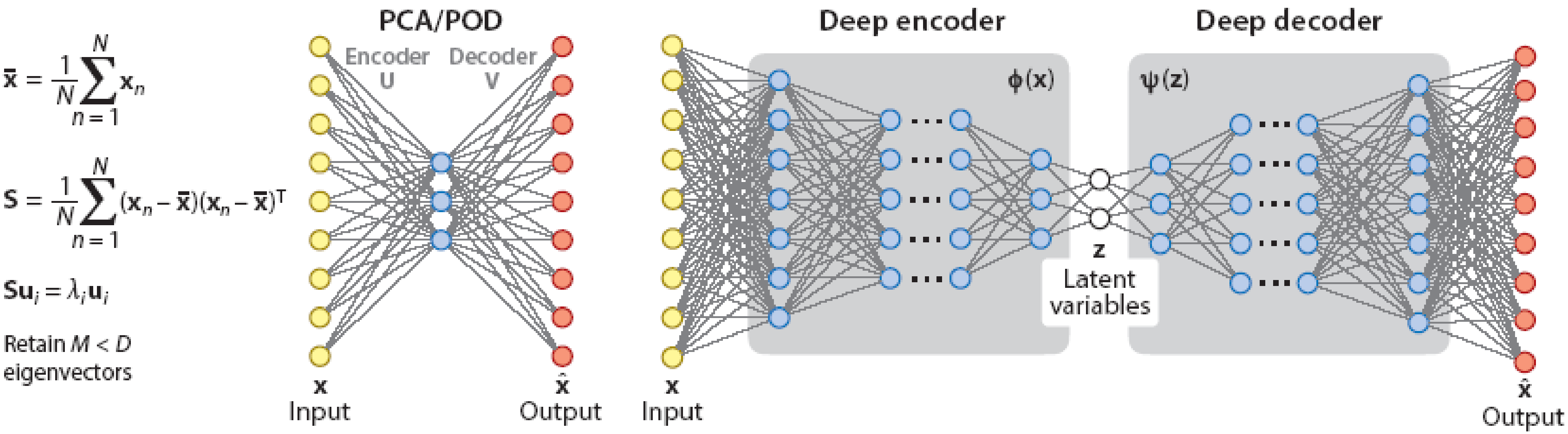


Figure 5

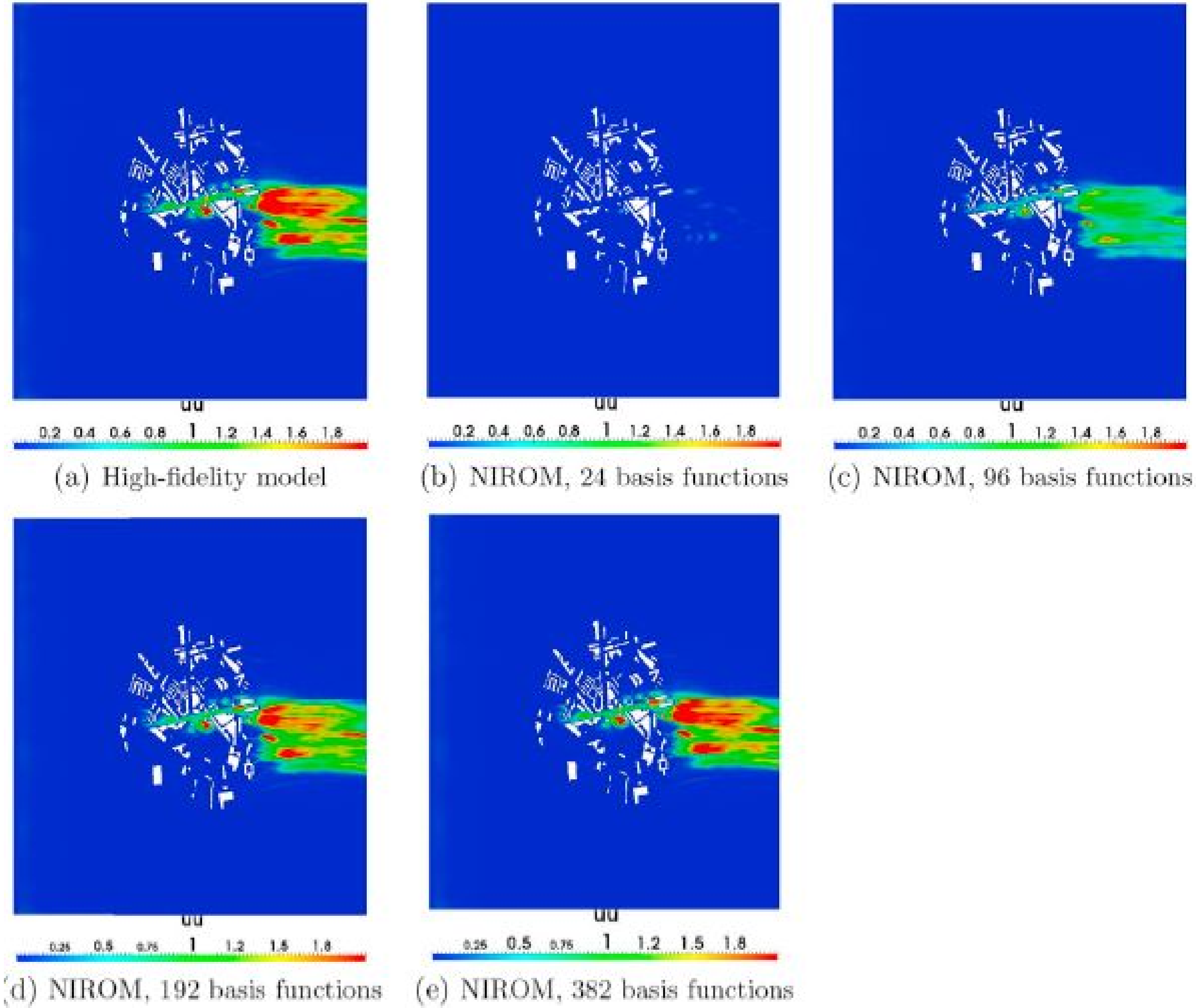
PCA/POD (*left*) versus shallow autoencoders (*center*) and deep autoencoders (*right*). If the node activation functions in the shallow autoencoder are linear, then **U** and **V** are matrices that minimize the loss function, $\|\hat{\mathbf{x}} - \mathbf{VUx}\|$. The node activation functions may be nonlinear, minimizing the loss function, $\|\mathbf{x} - \psi[\phi(\mathbf{x})]\|$. The input $\mathbf{x} \in \mathbb{R}^D$ is reduced to $\mathbf{z} \in \mathbb{R}^M$, with $M \ll D$. Note that PCA/POD requires the solution of a problem-specific eigenvalue equation, while the neuron modules can be extended to nonlinear activation functions and multiple nodes and layers. Abbreviations: PCA, principal component analysis; POD, proper orthogonal decomposition; **S**, covariance matrix of mean-subtracted data; **U**, linear encoder; $\mathbf{u}_i$, eigenvector; **V**, linear decoder; **x**, input vector; $\mathbf{x}_n$, n-th input vector; $\bar{\mathbf{x}}$, mean of input data; $\hat{\mathbf{x}}$, autoencoder reconstruction; **z**, latent variable; λ_i, eigenvalue; $\phi(\mathbf{x})$, deep encoder; $\psi(\mathbf{x})$, deep decoder. Figure based on an idea from Bishop & James (1993).

ML tools for Fluid Dynamics. Example 1. ROM for urban flow

The plots show the Reynolds stresses from the high-fidelity model (Fluidity, LES) and NIROM (predicting) with 24, 96, 192 and 382 basis functions.

These are shown on a horizontal plane at a height of 15m above ground level.

“A reduced order model for turbulent flows in the urban environment using machine learning” (Xiao et al, Buiding and Environment, 2019)



(a) High-fidelity model (b) NIROM, 24 basis functions (c) NIROM, 96 basis functions (d) NIROM, 192 basis functions (e) NIROM, 382 basis functions

ML tools for Fluid Dynamics. Example 2. Flow control

- Deep Reinforcement Learning to perform efficient collective swimming (Verma et al., PNAS, 2018)

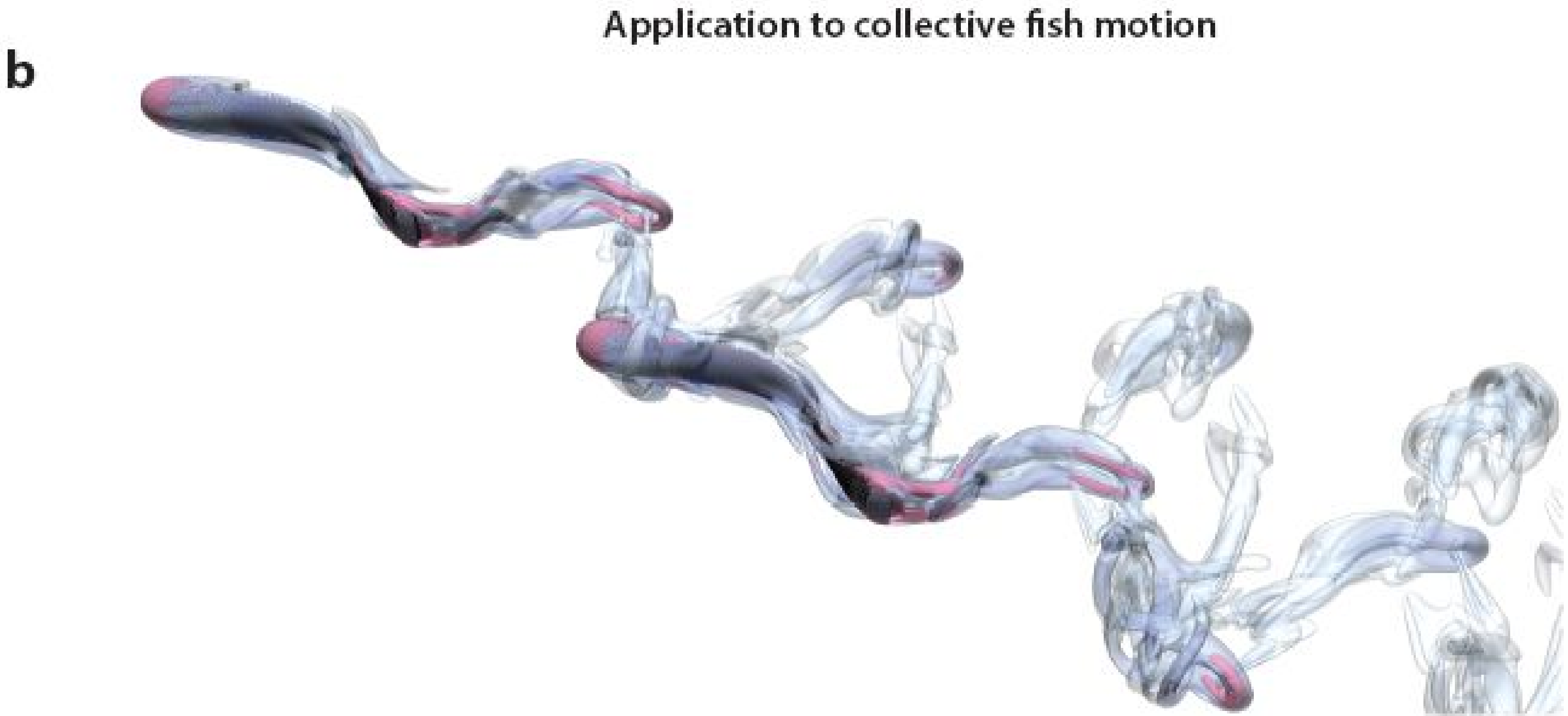
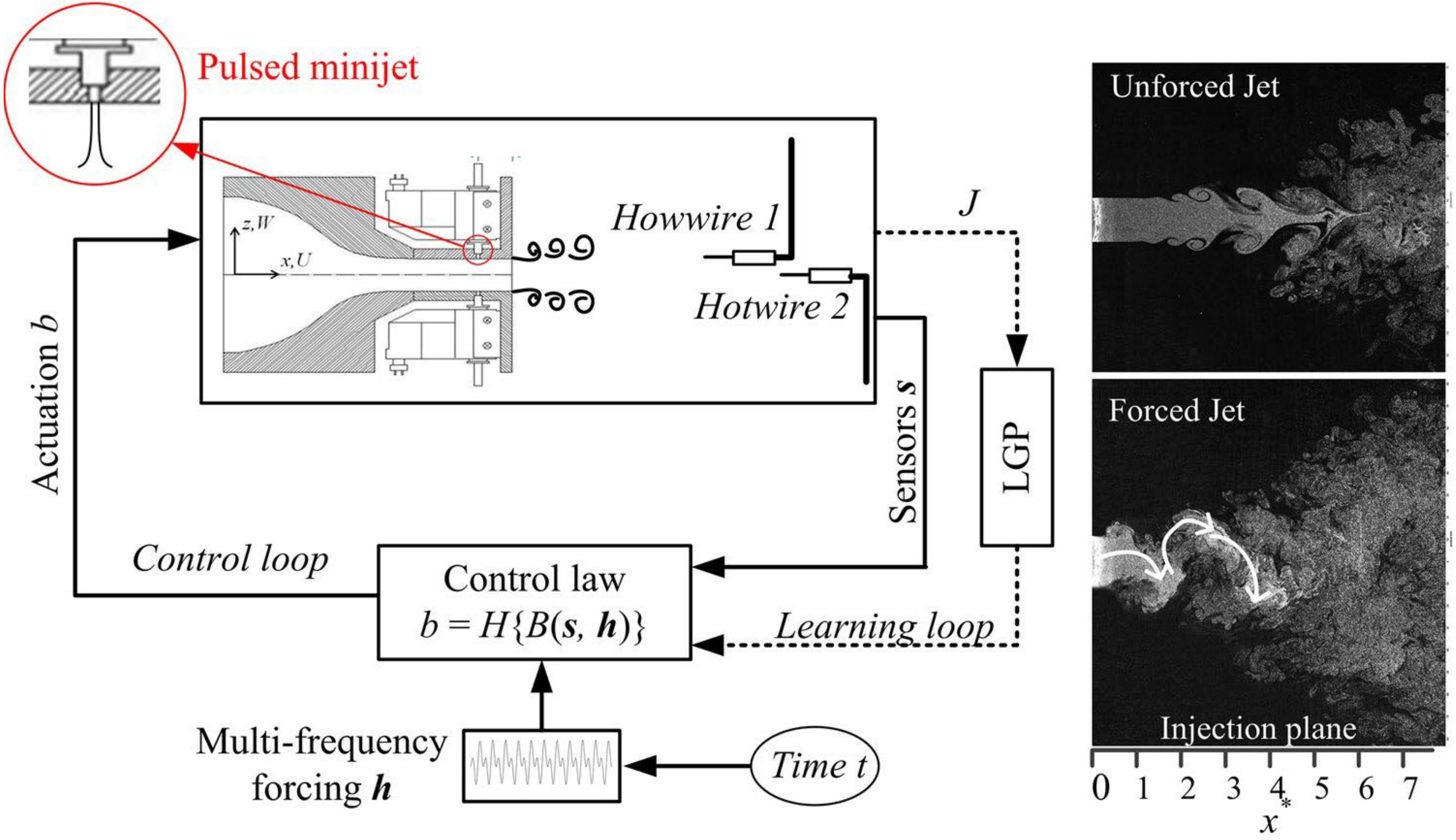


Figure 8

Deep reinforcement learning schematic (*a*) and application to the study of the collective motion of fish via the Navier–Stokes equations (*b*). Panel *b* adapted from Verma et al. (2018).

ML tools for Fluid Dynamics. Example 3. Jet optimization

- Jet mixing optimization using machine learning control (Wu et al., Experiments in Fluids, 2018)

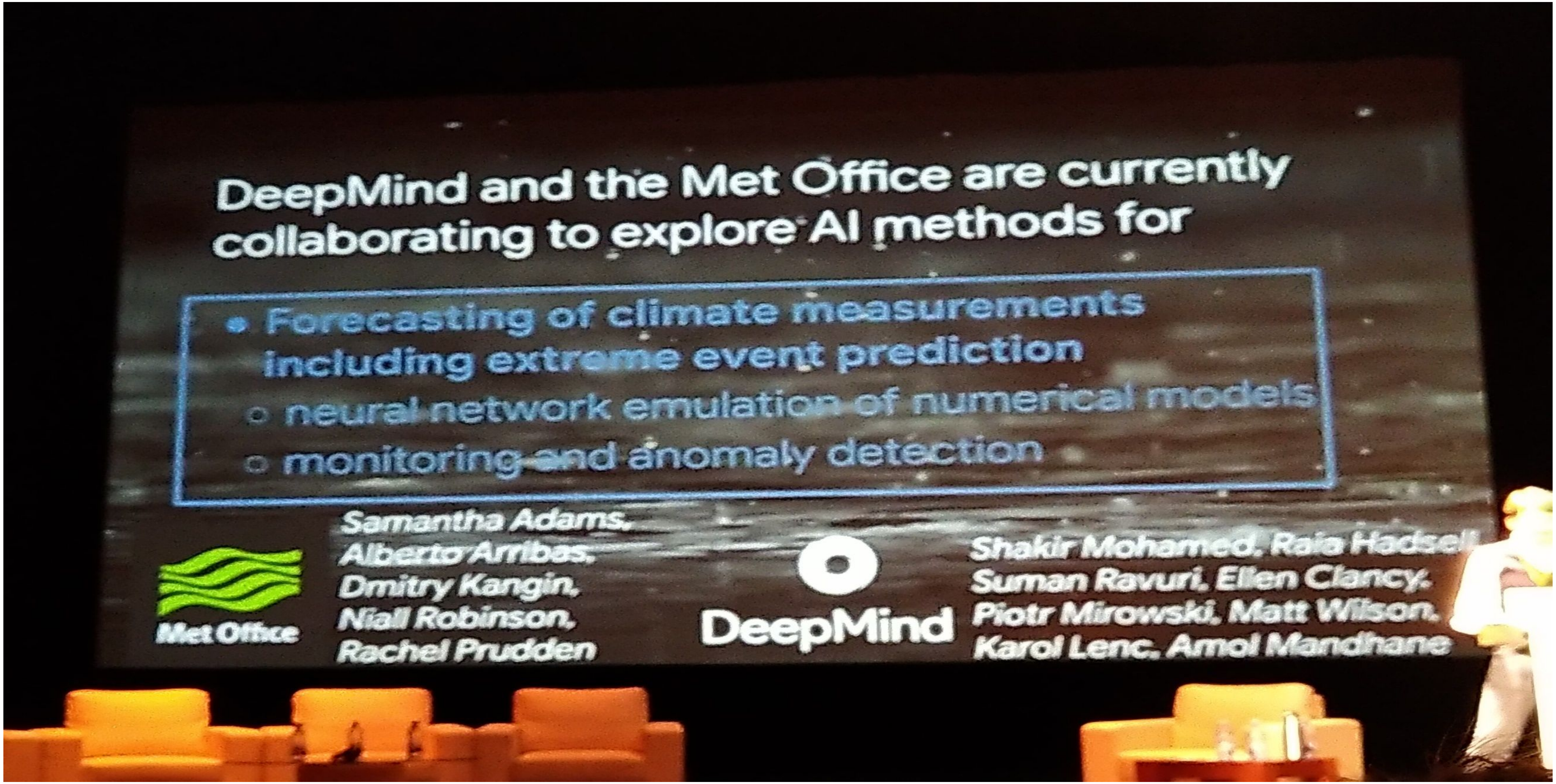


ML tools for Fluid Dynamics. Example 4. Gas & Oil Industry

- Keynote Lecture presentation at Russian Congress on Theoretical and Applied Mechanics (Ufa,

ML tools for Fluid Dynamics. Example 5. Weather Forecast

- Keynote Lecture presentation at East European ML School (Buharest, 2019)



Climate change:

- To facilitate work at the intersection of climate change and machine learning through resource- and knowledge-sharing

- To enable impactful collaborations by connecting machine learning experts and experts working in areas relevant to climate change through physical and digital platforms

- To promote discourse about best practices regarding the use of machine learning in climate change domains

Valuable references to learn Machine Learning and Deep Learning:

- Andrew Ng Course (https://www.coursera.org/learn/machine-learning)
- A Comprehensive Guide to ML (Soroush Nasiriany, Garrett Thomas, William Wang, Alex Yang)
- Practical Deep Learning for Coders (https://course.fast.ai)

1. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с анг. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.
2. Шолле Ф. Глубокое обучение на Python. — СПб.: Питер, 2018. — 400 с.
3. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. — СПб.: Питер, 2018. — 480 с.
4. Zhang A., Lipton Z.C., Li M., Smola A.J. Dive into Deep Learning. Release 0.7.1 (https://d2l.ai/). – Feb 13, 2020. – 904 p.
5. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / пер. с англ. – М.: ДМК Пресс, 2015. 400 с.
6. M. Kubat. An Introduction to Machine Learning. – Springer, 2015. – 291 p.
7. Nils J. Nilsson. Introduction to Machine Learning: AN EARLY DRAFT OF A PROPOSED TEXTBOOK. – Stanford University, 1996 (201 p.), 1998 (179 p.).
8. Хайкин С. Нейронные сети: полный курс, 2-е издание: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.
9. Cherkassky V., Mulier F.M. Learning from Data: Concepts, Theory, and Methods. – Hoboken, NJ: JohnWiley & Sons, 2007. – 538 p.
10. Schölkopf B., Smola A.J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. – Cambridge, MA: MIT Press, 2002. – 626 p.
11. Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction. – Cambridge, MA: MIT Press., 2018. – 526 p.
12. Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. – Boston: MIT Press, 1992. – 813 p.
13. Barber D. Bayesian Reasoning and Machine Learning. Cambridge, UK: Cambridge Univ. Press, 2012, 2015. – 642, 658 p.
14. Theodoridis S. Machine Learning: A Bayesian and Optimization Perspective. San Diego, CA: Academic, 2015. – 1050 p.
15. Loucks D., van Beek E., Stedinger J., Dijkman J., Villars M. Water Resources Systems Planning and Management: An Introduction to Methods, Vol. 2. Cham, Switz.: Springer, 2005. – 680 p.
16. Duriez T., Brunton S.L., Noack B.R. Machine Learning Control: Taming Nonlinear Dynamics and Turbulence. Cham, Switz.: Springer, 2016. – 211 p.

Journals with papers on Machine Learning in Fluid Mechanics:

Papers on ML for fluid mechanics were published in arXiv (free online access) and top-rank journals:

Annu. Rev. Fluid Mech. (14.81), PNAS (9.58), J. Fluid Mech. (3.14), Bioinspiration & Biomimetics (3.13), J. Comput. Phys. (2.85), Phys. Fluids (2.63), Phys. Rev. Fluids (2.44), AIAA J. (1.95)

Nat. Energy (54.00), Nature (43.07), Nat. Commun. (11.88), Phys. Rev. Lett. (9.23), IEEE Trans. Evol. Comput. (8.51), IEEE Trans. Syst. Man Cybern. (7.35), Prog. Aerosp. Sci. (6.81), ACM Trans. Graph. (6.50), Appl. Mech. Rev. (6.14), Comput. Methods Appl. Mech. Eng. (4.82), Int. J. Heat Mass Transf. (4.35), Comput. Chem. Eng. (3.33), Int. J. Uncert. Quant. (3.26), J. Chem. Phys. (3.00), Int. J. Numer. Meth. Eng. (2.75), J. Turbomach. (2.59), Exp. Fluids (2.44), Phys. Rev. E (2.35), SIAM J. Sci. Comput. (2.31), IEEE Trans. Syst. Man Cybern. C (2.19), Theor. Comput. Fluid Dyn. (1.94), J. Aircraft (0.96)

Relevant papers on ML (not for fluid mechanics) were published in:

IEEE Trans. Neural Netw. (11.68), Neural Netw. (7.20), J. Mach. Learn. Res. (4.09), Evol. Comput. (3.47), Mach. Learn. (2.81), Neural Comput. (2.26)

Nature (43.07), Science (41.06), Sci. Adv. (12.80), Proc. IEEE (10.69), SIAM Rev. (7.22), IEEE Control Syst. Mag. (6.23), Control Eng. Pract. (3.23), IEEE Trans. Inf. Theory (3.22), Ann. Stat. (2.90), Proc. R. Soc. A (2.82), PLOS ONE (2.78), Chaos (2.64), J. Phys. A (2.11), J. Guid. Control Dyn. (2.06), SIAM J. Matrix Anal. Appl. (1.91), IEEE Computer Graphics and Applications (1.73), Nat. Comput. (1.33), Biol. Cybernet. (1.31), SIAM J. Math. Data Sci. (started in 2019)