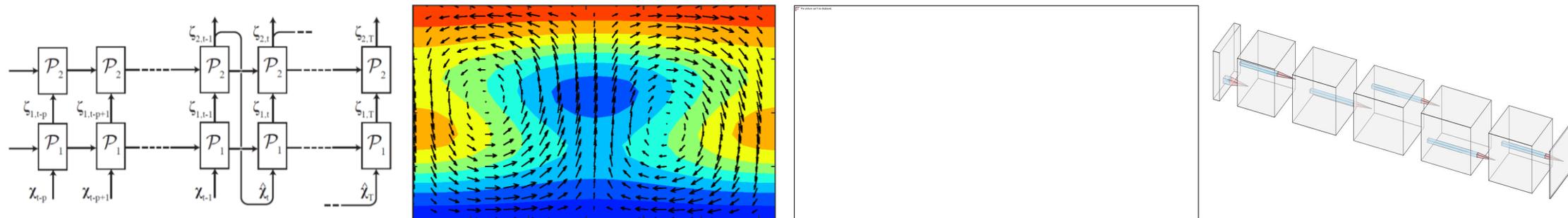


Modeling and controlling turbulent flows through deep learning

Ricardo Vinuesa^{1,2}

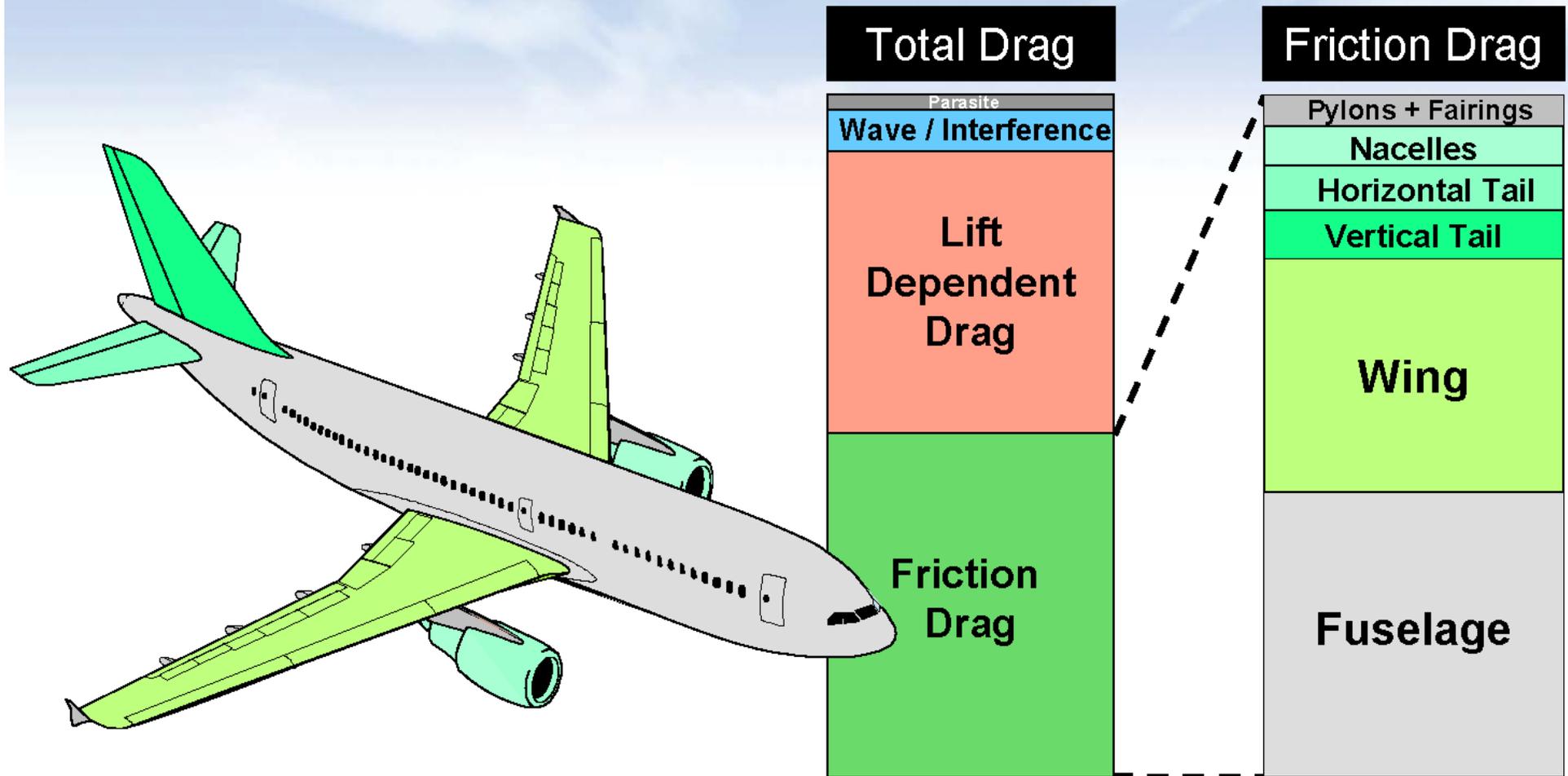


¹FLOW, KTH Engineering Mechanics, Stockholm, Sweden.

²Swedish e-Science Research Centre (SeRC), Stockholm, Sweden.

Motivation

Typical break down of overall aircraft[†] drag by form & component



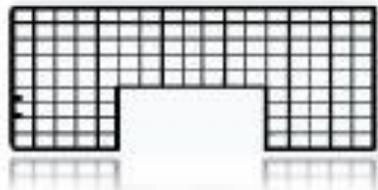
† = Based on a typical A320

Hills. The Airbus Challenge. EADS Engineering Europe, 2008

Numerical code: Nek5000 (Fischer *et al.*, 2008)

- **High-order numerical methods** are required for accurate simulations of turbulent flows due to the significant scale disparity of the flow structures, both in time and space.
- Spectral elements allow to solve **flows in complex geometries**.

Finite element (FE)

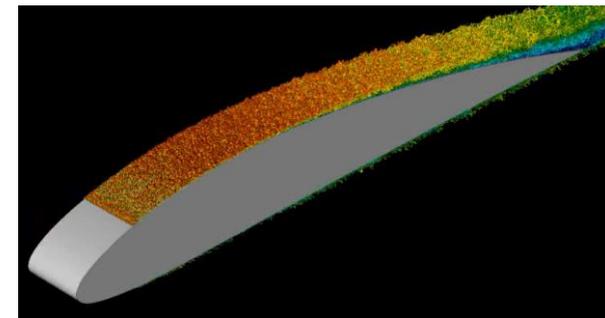
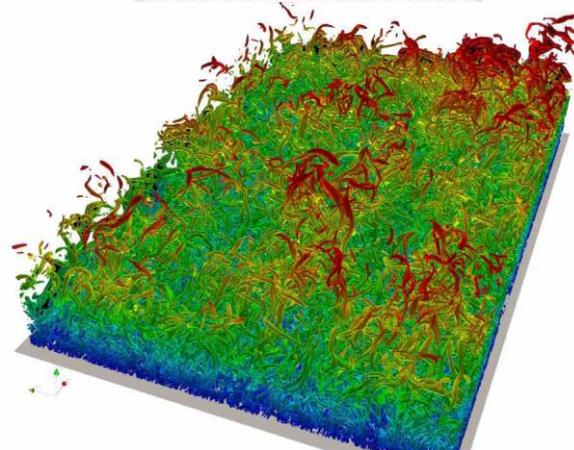
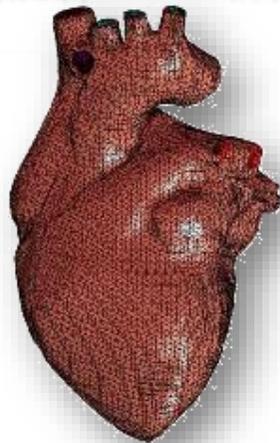
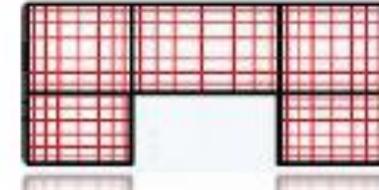


+

Spectral



Spectral element (SE)



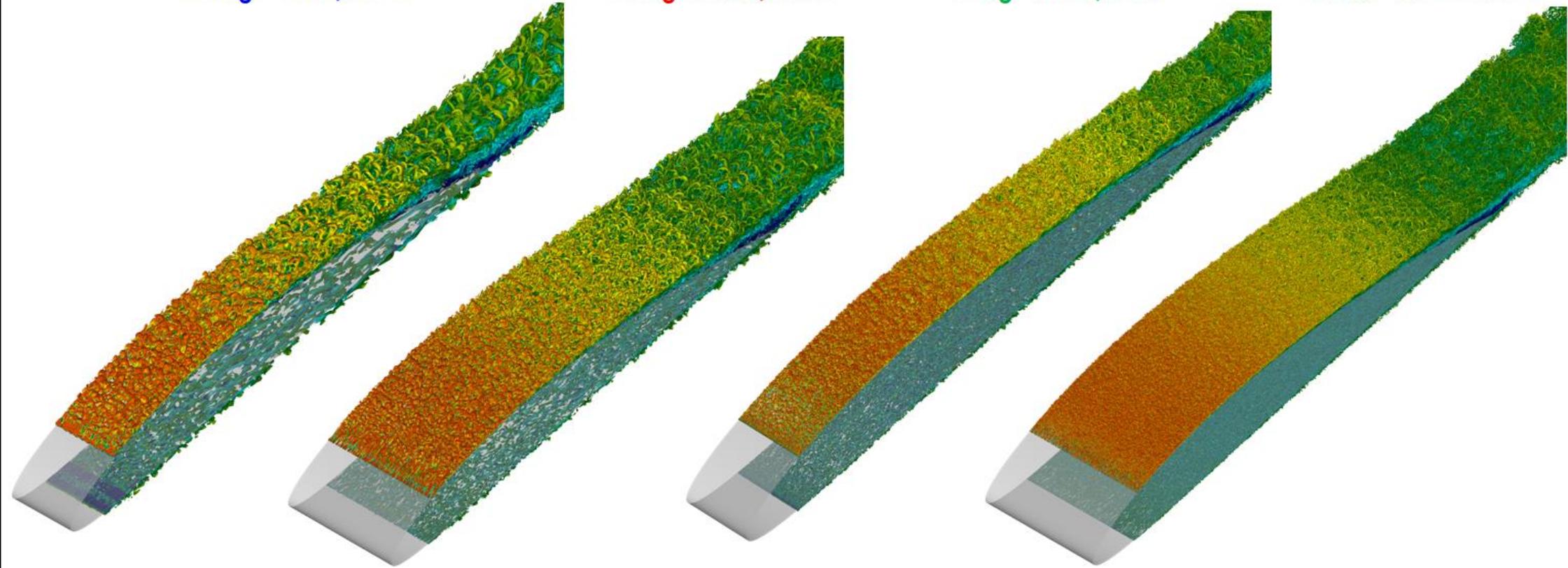
Effect of Reynolds number for a given pressure-gradient history: well-resolved LES

$Re_c=100,000$

$Re_c=200,000$

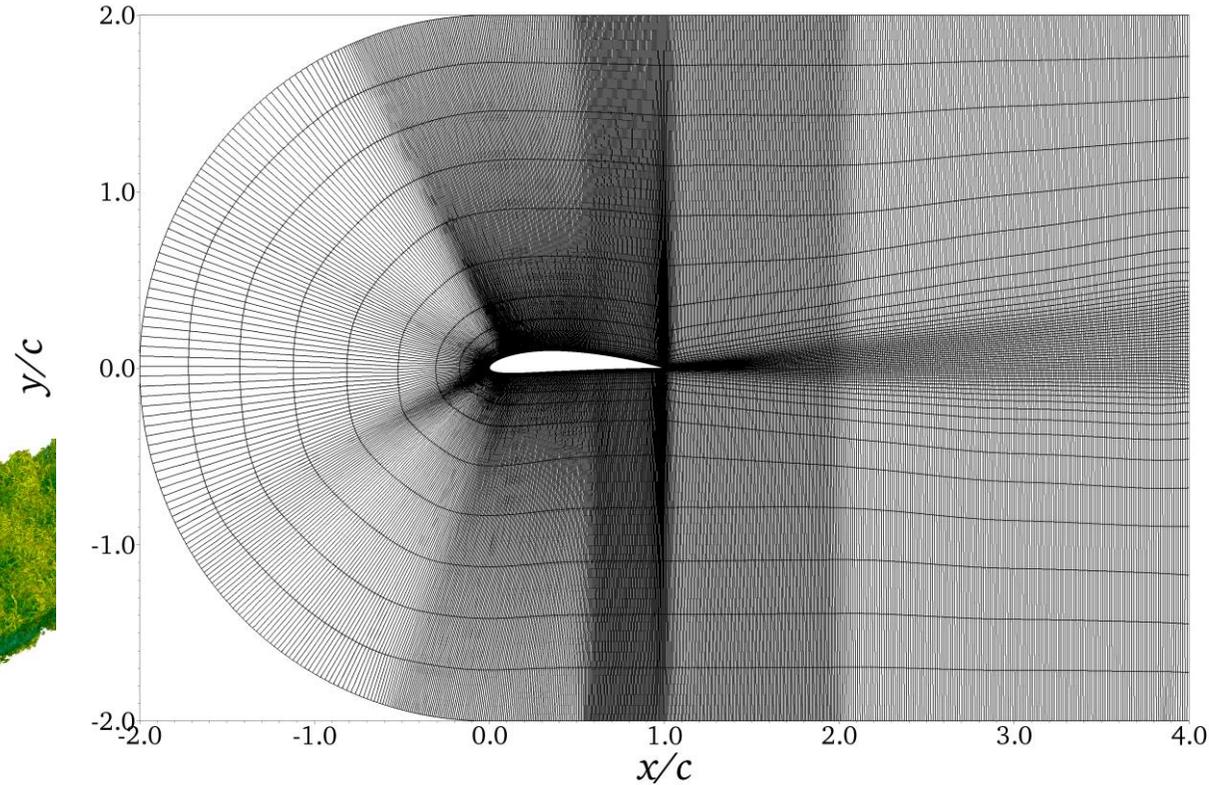
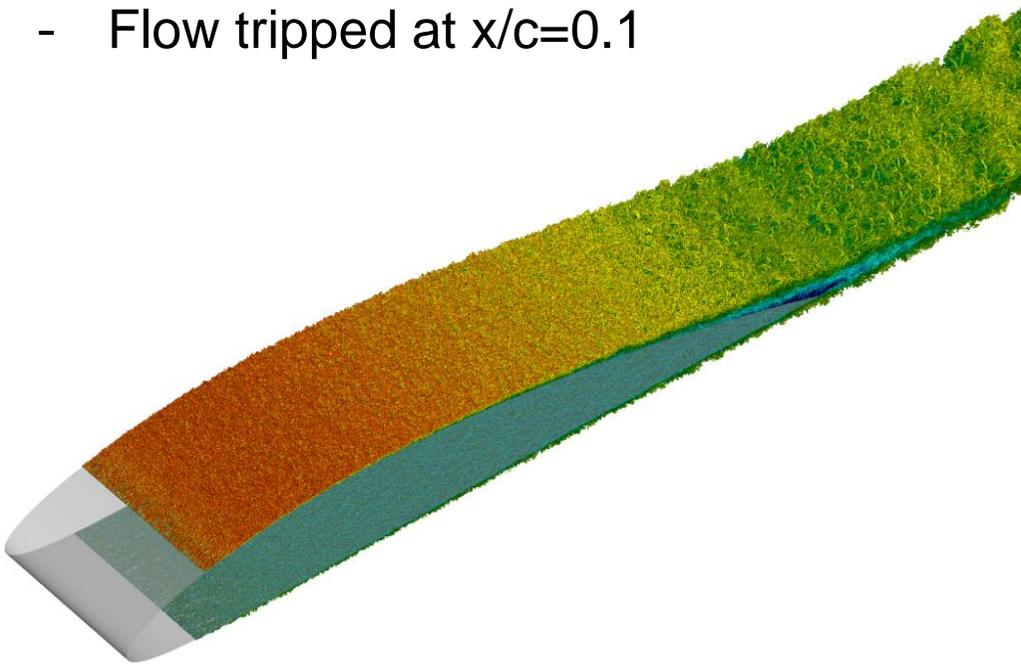
$Re_c=400,000$

$Re_c=1,000,000$



Well-resolved LES of NACA4412 wing section. $Re_c=1,000,000$ and $AoA=5^\circ$

- Relaxation-term LES with Nek5000 (as in Schlatter *et al.*, 2004).
- $(L_x, L_y, L_z) = (6c, 4c, 0.2c)$
- Use of RANS BCs.
- $Re_\theta \approx 6,000$ and $Re_\tau \approx 707$
- Flow tripped at $x/c=0.1$

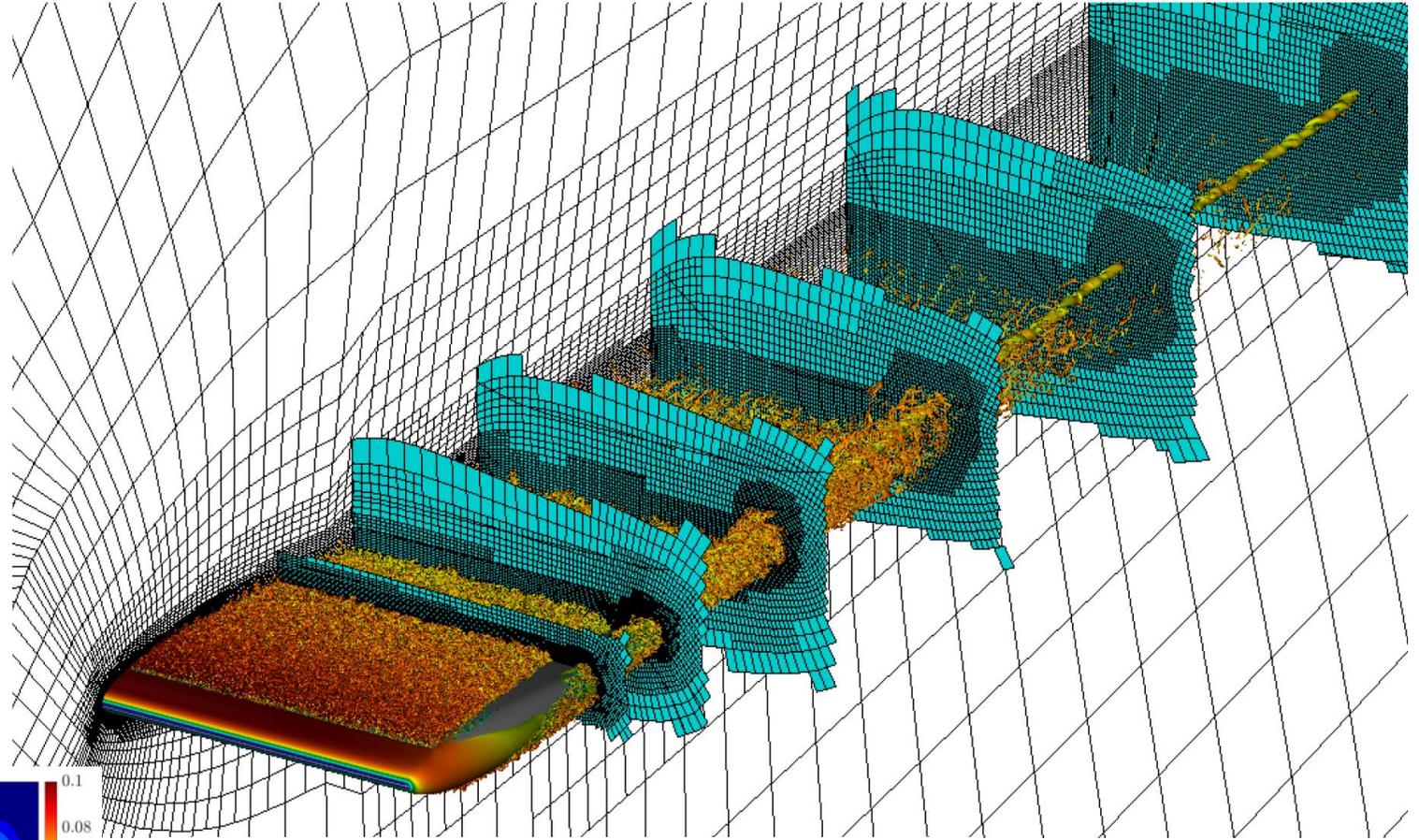


- 4.5 million elements, $N=7$
- **2.3 billion grid points.**
- On-going accumulation of statistics.
- $\Delta x^+ < 27$, $\Delta y^+ < 0.96$, $\Delta z^+ < 13$, $\Delta x < 13\eta$

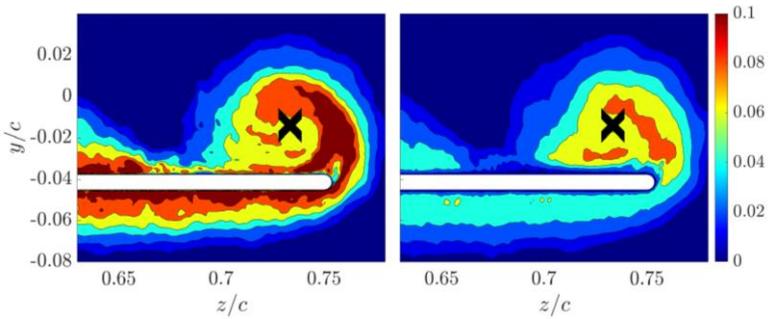
Adaptive simulations of NACA0012 profile with rounded wing tip

Toosi et al., DLES conference (2022)

Statistics, time series, fields and reduced-order models of the wing-tip vortex and the wake.



u_{rms} and v_{rms}



Grid	N_{GLL}	N_{grid}	$(\Delta x_{mean}^+, y_1^+, \Delta z_{mean}^+)$
P-0	376×10^6	249×10^6	(10.3, 0.72, 8.7)
P-5	381×10^6	252×10^6	(10.5, 0.73, 8.5)
P-10	435×10^6	288×10^6	(12, 0.8, 9)
RWT-0	950×10^6	630×10^6	(10.3, 0.7, 5.5)
RWT-5	1.58×10^9	1.05×10^9	(10.5, 0.75, 5.7)
RWT-10	2.16×10^9	1.43×10^9	(12, 0.8, 6)



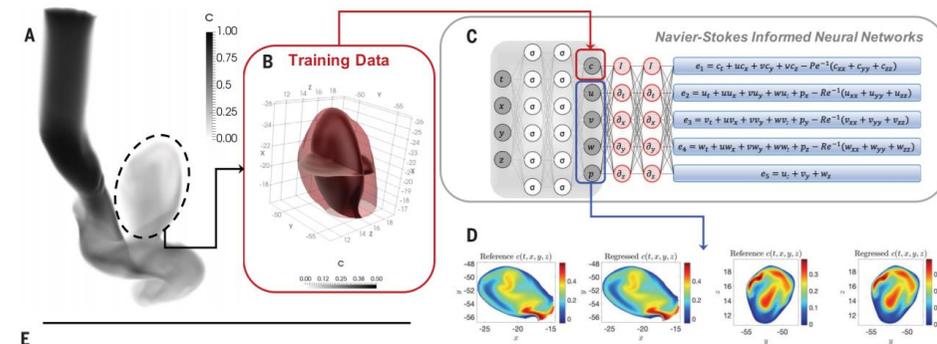
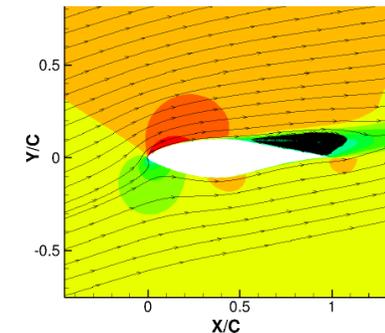
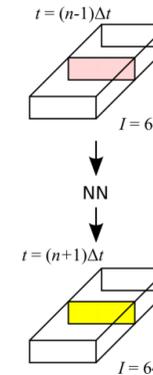
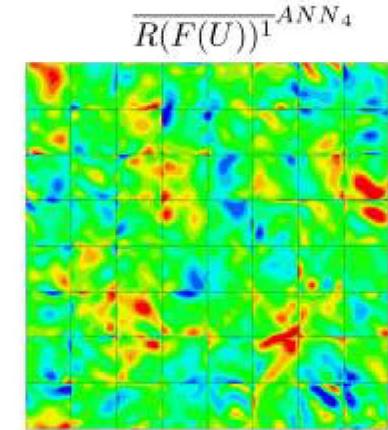
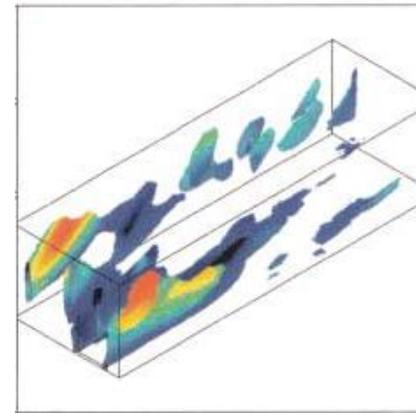
High-fidelity simulation of wing-tip vortex at $Re=200,000$ and 5 degree angle of attack



Toosi et al., DLES conference (2022)

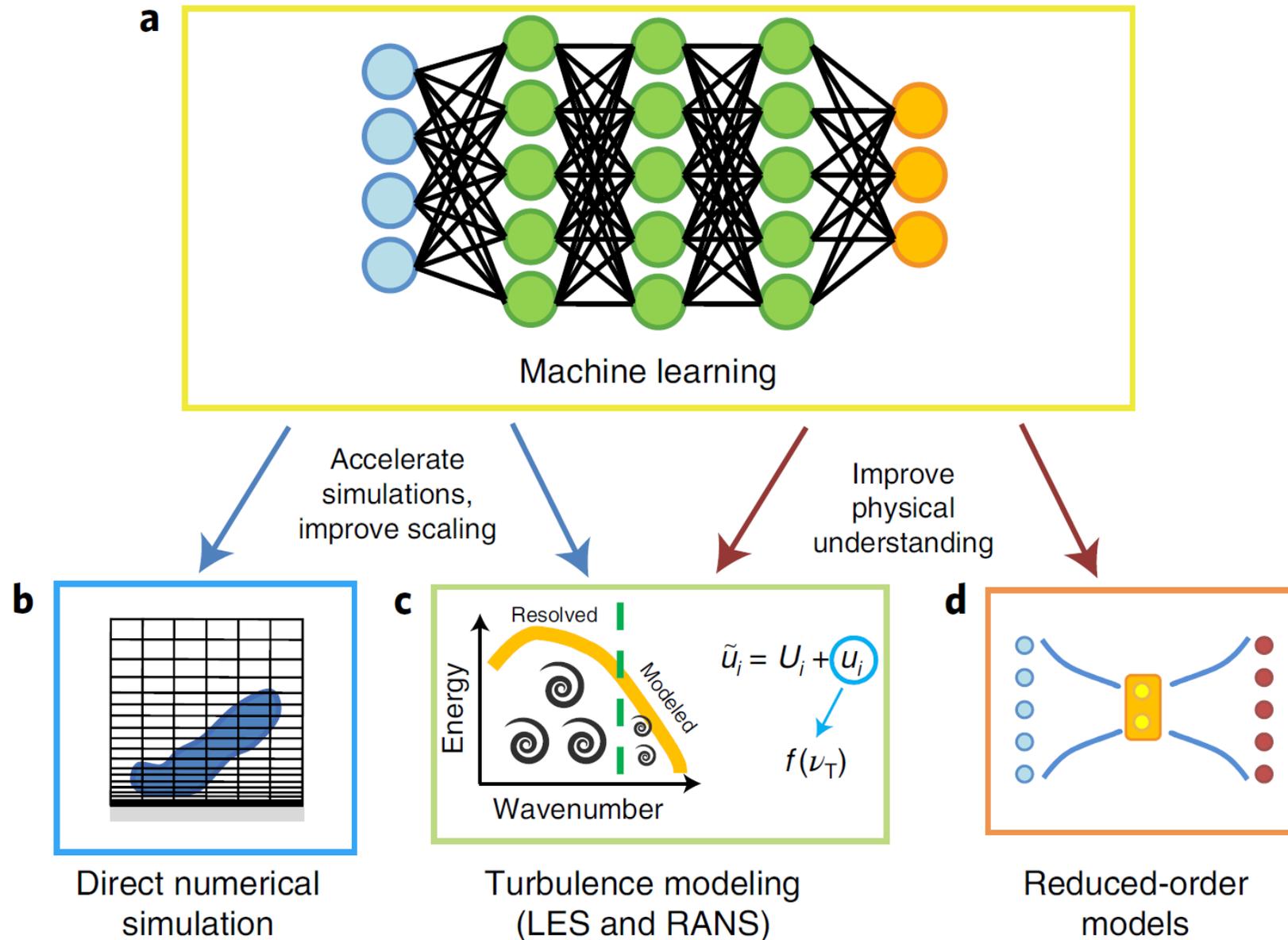
Applications of machine learning to fluid mechanics

- **Milano and Koumoutsakos (2002)** used neural networks to model the **near-wall region of turbulent channel flows**.
- **Beck et al. (2018)** employed neural networks to develop **subgrid-scale (SGS) models for LES**. They used DNS data to train and predict the SGS term.
- **Fukami et al. (2019)** used neural networks to obtain **inflow conditions** and for super-resolution.
- **Duraisamy et al. (2019)** showed the potential of machine learning for developmet of **RANS models**
- **Raissi et al. (2020)** identified the potential of physics-informed neural networks (PINNs) for reconstrcuting flow fields from tracer input.



FLOW - **Brunton et al. (2019)** showed, among others, applications to **flow control**.

Enhanced CFD with machine learning





Outline of machine-learning applications to fluid mechanics

- Non-intrusive sensing in a turbulent channel via CNNs.
- Non-intrusive sensing with coarse measurements via GANs.
- Predictions for wall models via CNNs.
- Reduced-order models in turbulence via AEs.
- Flow control via DRL.



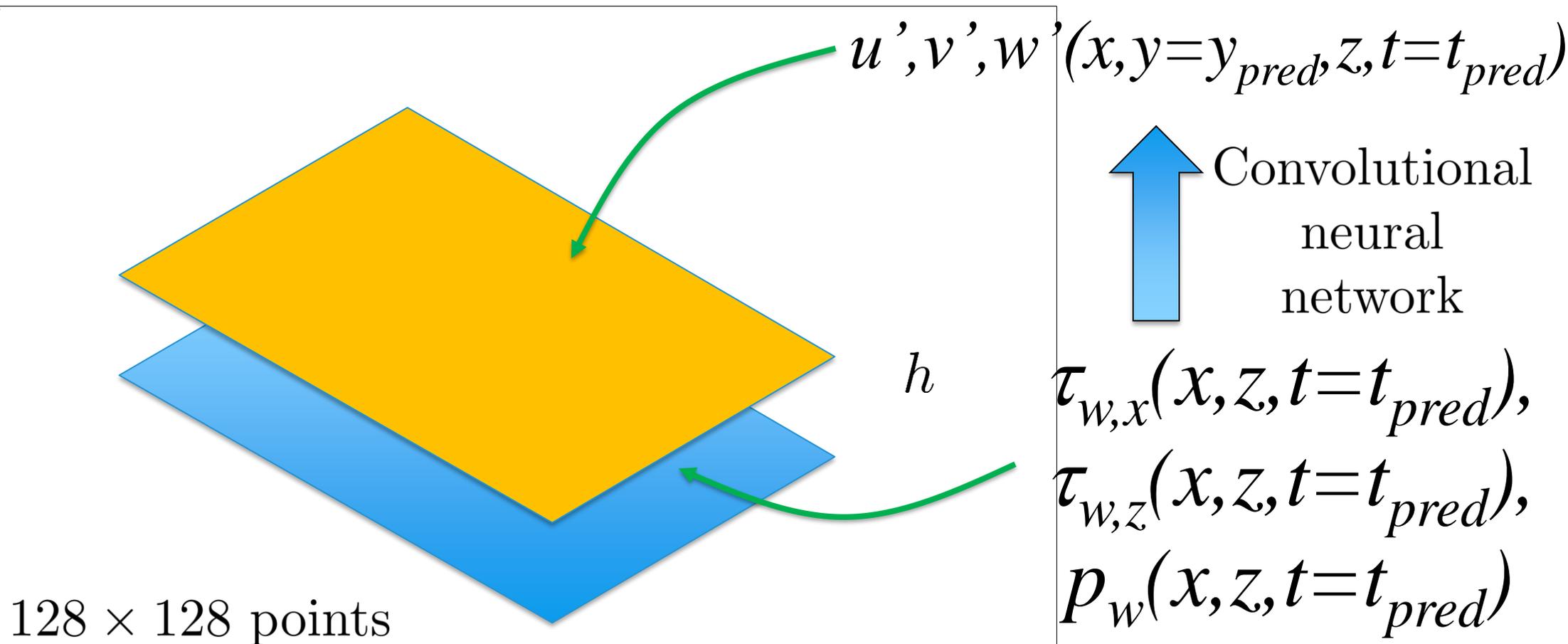
Outline of machine-learning applications to fluid mechanics

- Non-intrusive sensing in a turbulent channel via CNNs.
- Non-intrusive sensing with coarse measurements via GANs.
- Predictions for wall models via CNNs.
- Reduced-order models in turbulence via AEs.
- Flow control via DRL.



Flow reconstruction with a convolutional neural network (CNN)

- Prediction of **off-wall planes** based on information **at the wall**.
- **Additional motivation: closed-loop reactive control.**



FLOW

Guastoni et al., J. Fluid Mech. 928, A27 (2021)



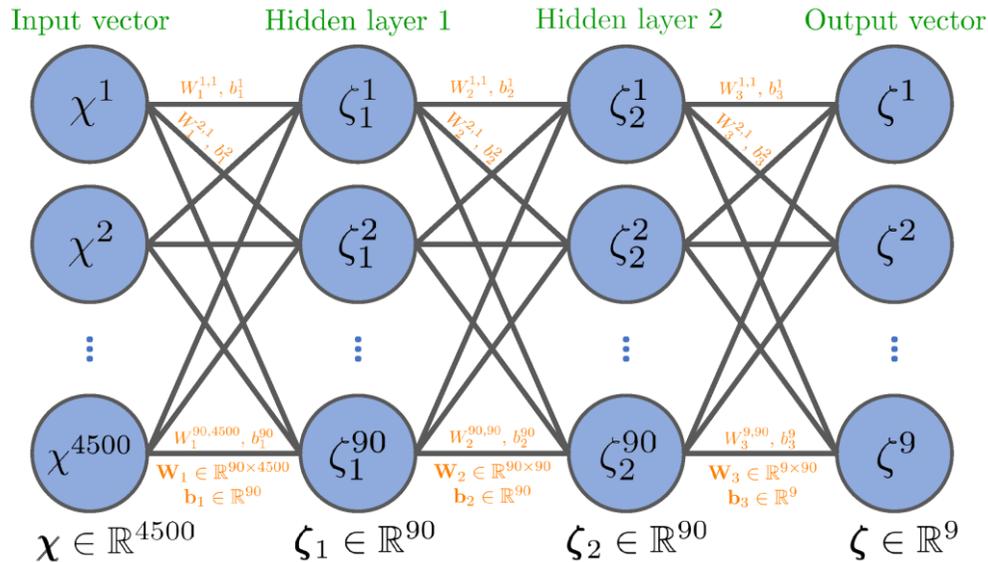
DNS of turbulent open channel flow

- DNS based on the Fourier-Chebyshev code **SIMSON** from KTH (Chevalier et al., 2007).
- Open channel flow (**simplicity**): no large-scale interactions between two walls.

- $Re_\tau = 180$ & 550
- $(L_x, L_y, L_z) = (4\pi h \times h \times 2\pi h)$
- No slip at $y/h=0$
- Symmetry at $y/h=1$

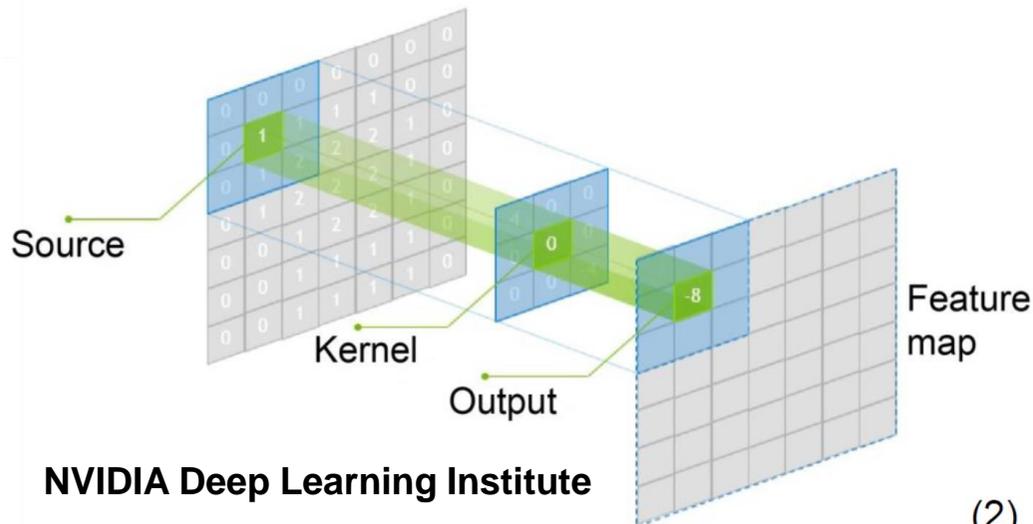
h

Neural-network architecture



Srinivasan et al., Phys. Rev. Fluids (2019)

- **Fully connected layers:** weighted sum of the inputs and non-linear activation function.
- **Very large numbers of parameters:** they are prone to over-fitting the data.



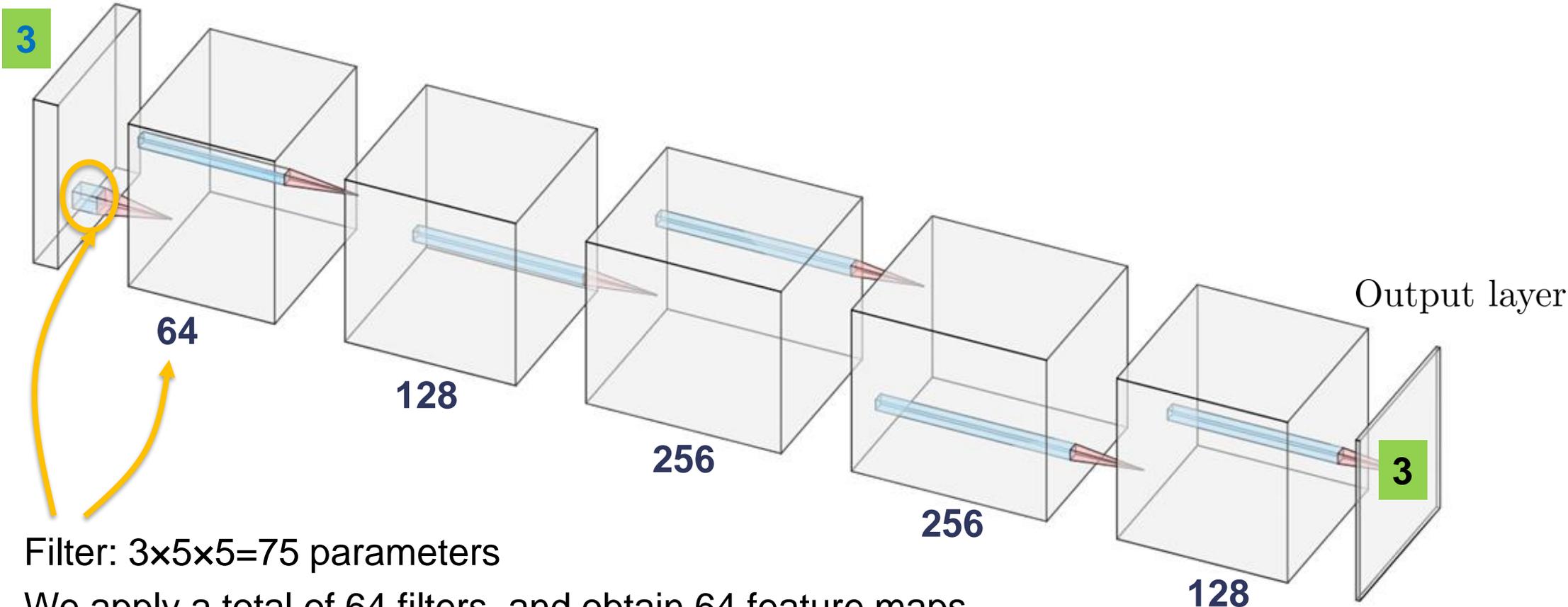
NVIDIA Deep Learning Institute

(2)

- **Convolutional layers:** small set of parameters (kernel of 3x3 or 5x5 sweeps the domain).
- Extensively used in **computer vision** (easy to train and suitable for identification of local features).

- Prediction of **off-wall planes** based on information **at the wall** with CNNs (Li et al., 1984).

Input layer



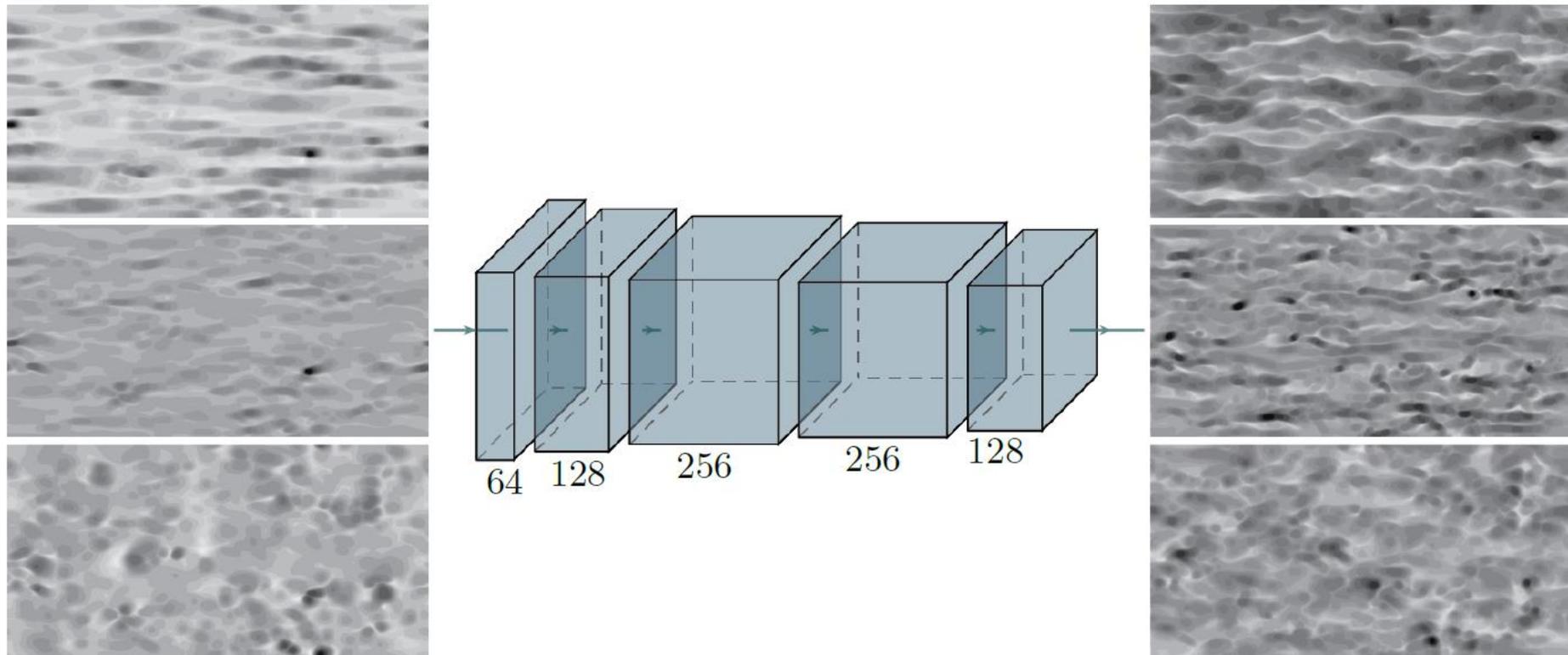
Filter: $3 \times 5 \times 5 = 75$ parameters

We apply a total of 64 filters, and obtain 64 feature maps

Each filter identifies one (or several) feature. More complex towards the output.

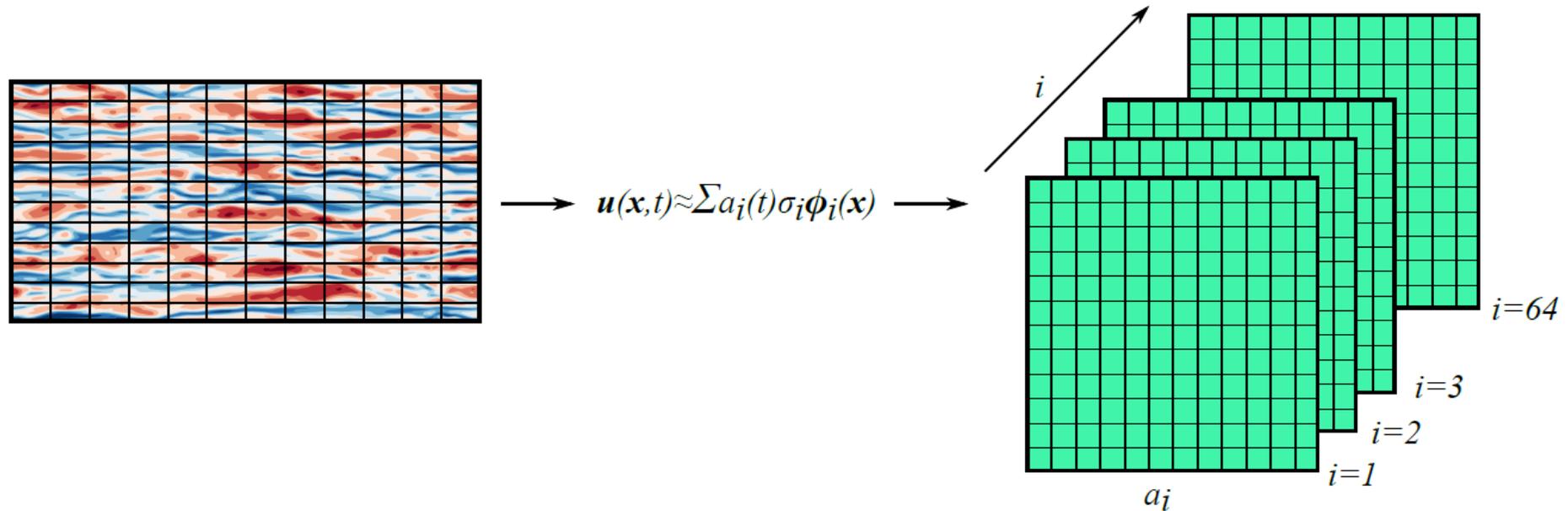
An extension to $Re_\tau=550$

- We consider **3 input fields** (adding the pressure).
- We predict the **3 velocity fluctuations** (scaled to be in the same range).
- Note that we consider **periodic padding** (CNN output is deterministic, local influence).



A new method: FCN-POD

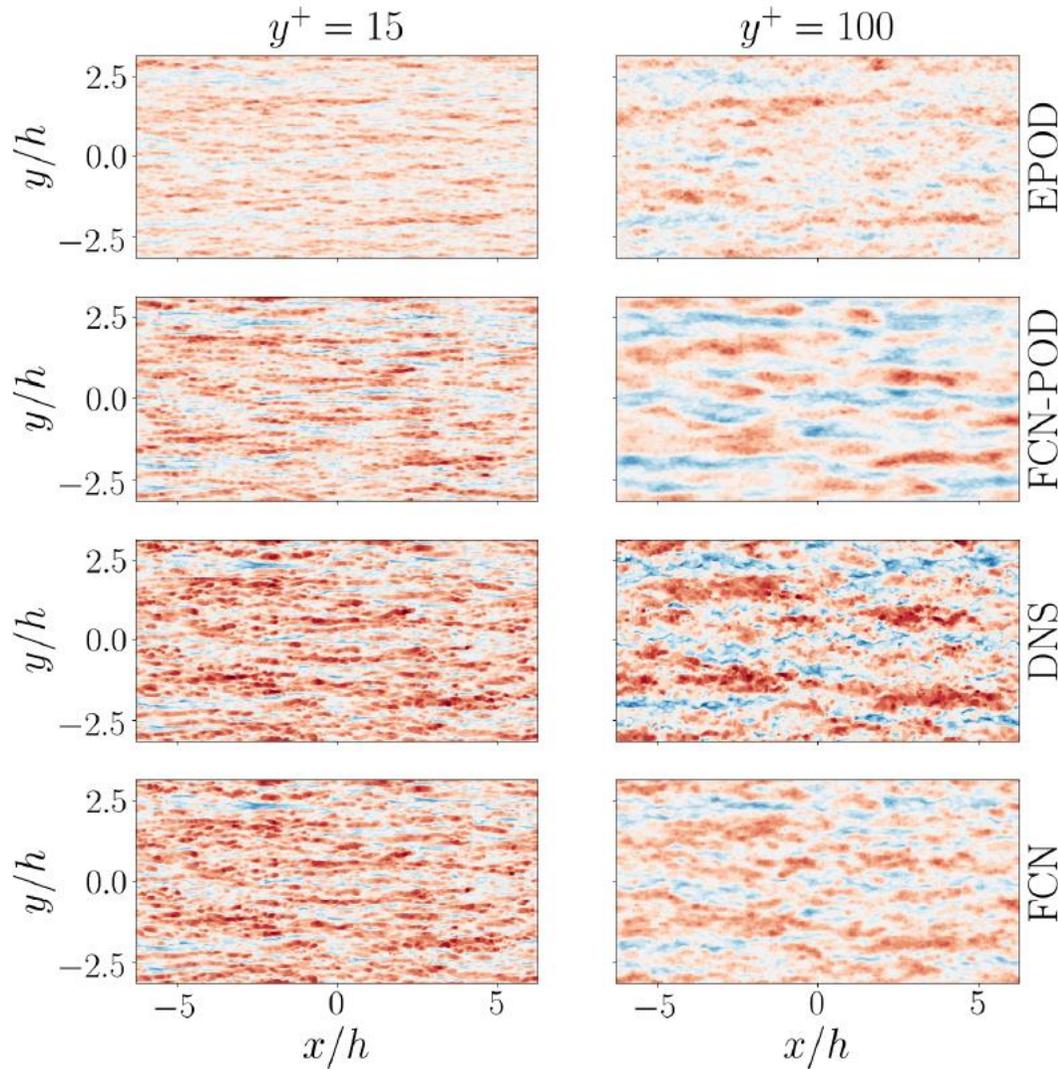
- In addition to the direct field prediction, we consider predicting the **POD coefficients** (embedding known physical information).
- We split the target plane into **subdomains** (reconstruction with fewer POD modes).



Güemes et al., Phys. Fluids (2019)

Guastoni et al., J. Fluid Mech. 928, A27 (2021)

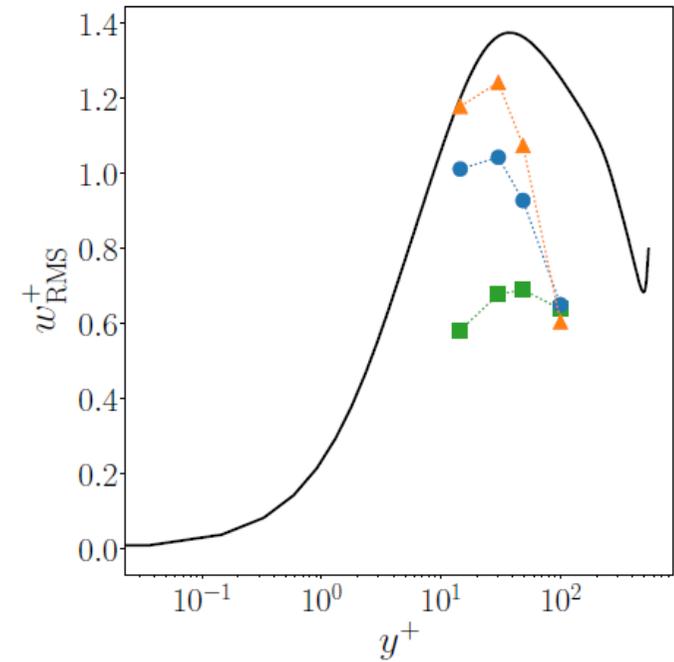
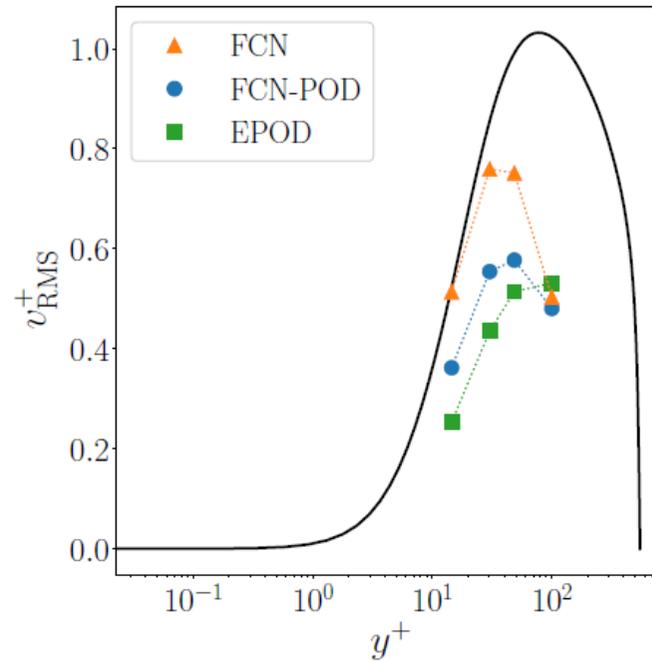
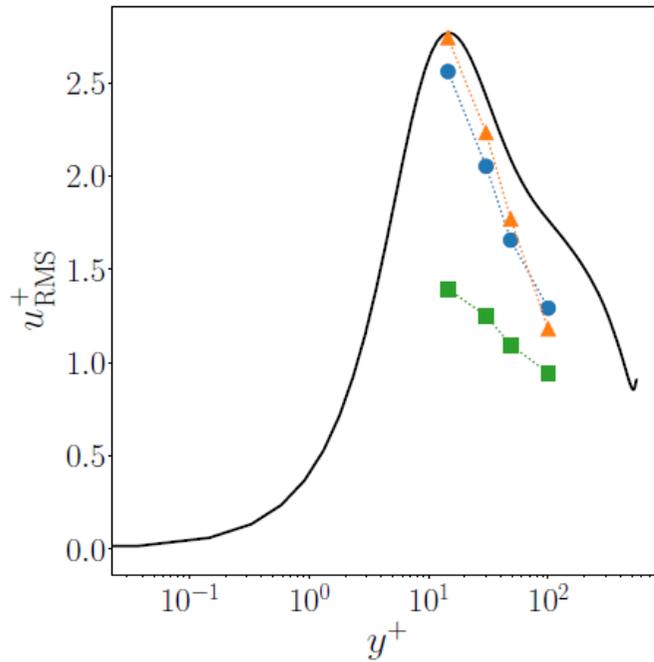
Instantaneous predictions at $Re_\tau=550$



- **EPOD** is formally equivalent to LSE (linear method).
- **Close to the wall FCN is best, farther from it FCN-POD is better** due to the encoded information in the POD modes.

Turbulence statistics at $Re_\tau=550$

- Close to the wall FCN is best, farther from it FCN-POD is better due to the encoded information in the POD modes.



$E_{\text{RMS}}^+(\cdot)$ [%]

$y^+ = 15$

$y^+ = 30$

$y^+ = 50$

$y^+ = 100$

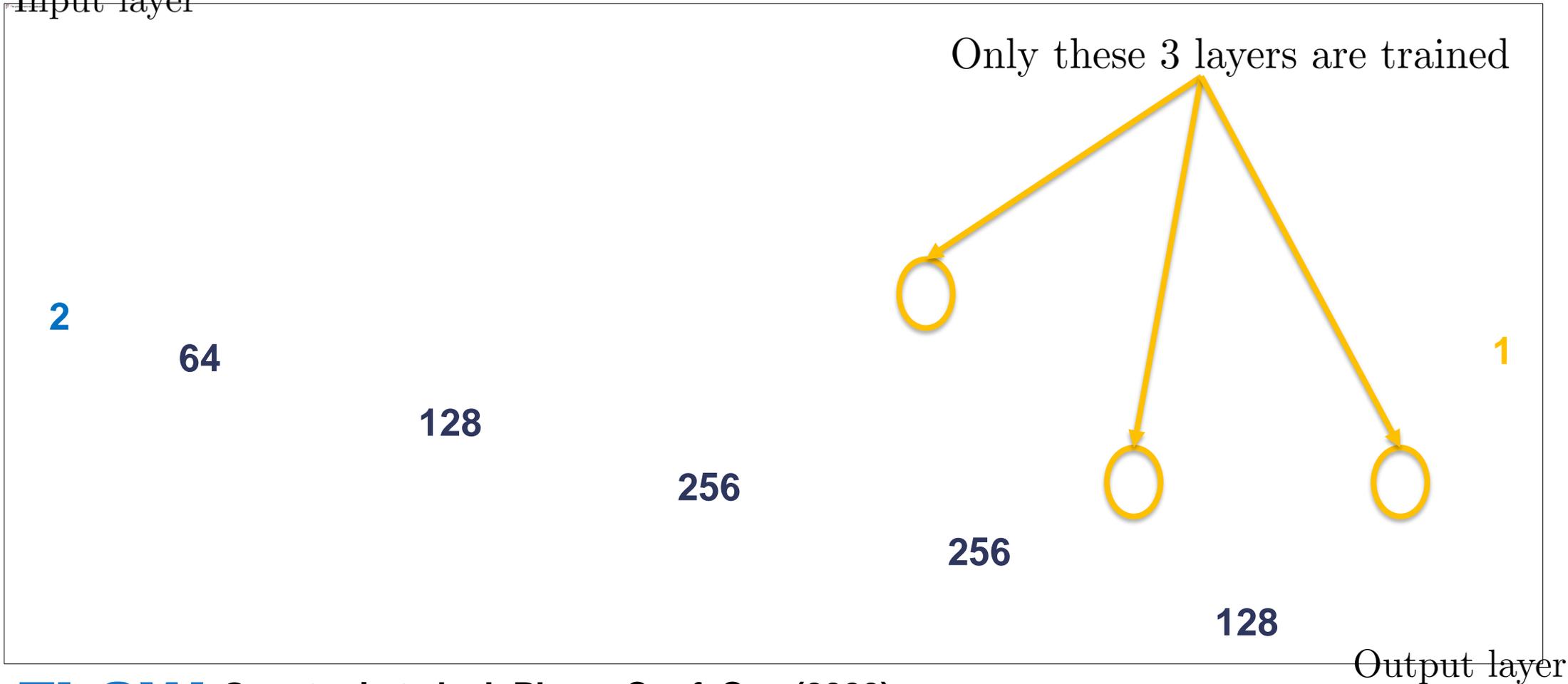
	EPOD	FCN	FCN-POD
u	49.69	0.98 (± 0.66)	7.53 (± 0.27)
v	48.53	8.10 (± 0.62)	15.53 (± 0.39)
w	47.62	15.33 (± 0.22)	20.75 (± 0.85)
	46.53	33.06 (± 0.30)	26.79 (± 0.36)



Improving training performance: Transfer learning at $Re_\tau=180$

- The deeper layers identify progressively **larger features**.
- Fix the first 3 layers (trained at $y^+=15$) and **train the last 3** (to predict at $y^+=50$).

Input layer





Improving training performance: Transfer learning at $Re_\tau=180$

- The deeper layers identify progressively **larger features**.
- Fix the first 3 layers (trained at $y^+=15$) and **train the last 3** (to predict at $y^+=50$).

Input layer

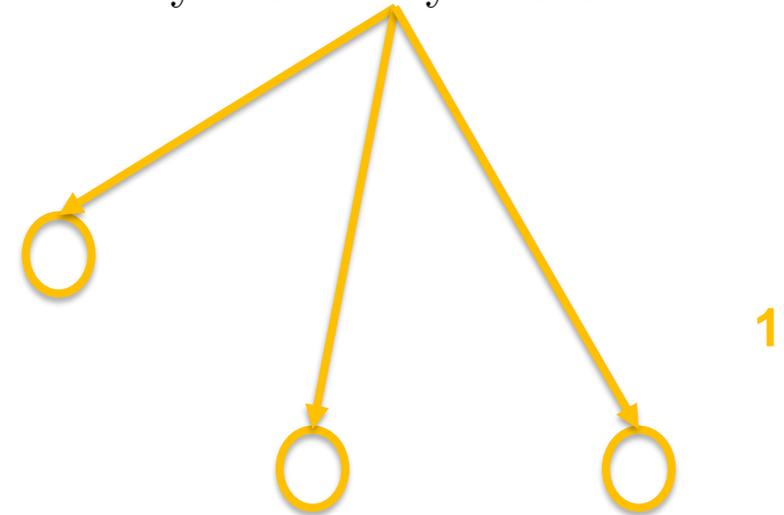
2

64

128

256

Only these 3 layers are trained



256

128

Output layer

	M.S.E. [$\times 10^{-3}$]	$E_{\bar{u}}$ [%]	$E_{u_{RMS}}$ [%]	Time [%]
Fully trainable	2.94	1.35	28.5	100
Transfer learning	3.17	0.5	30.2	23

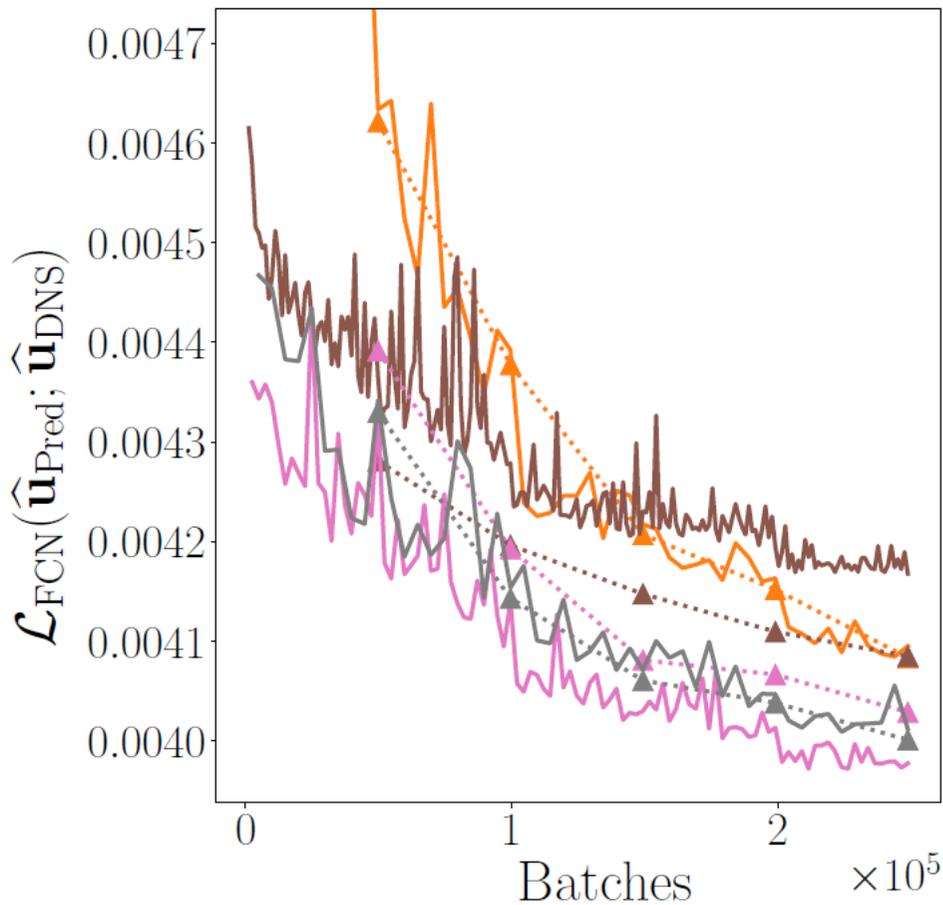
Guastoni et al., J. Phys.: Conf. Ser. (2020)



Transfer learning from $Re_\tau=180$ to 550

Guastoni et al., J. Fluid Mech. 928, A27 (2021)

- We initialize the training for $Re_\tau=550$ with the weights from the $Re_\tau=180$ case.
- The **learning rate** needs to be reduced. — Validation loss; $y^+ = 30$



— Test loss

Random weights, 100% dataset

Transferred weights, 100% dataset

Transferred weights, 50% dataset

Transferred weights, 25% dataset

$E_{RMS}^+(\cdot)$ [%]		$y^+ = 15$	$y^+ = 30$	$y^+ = 50$	$y^+ = 100$
u	Ref.	0.98 (± 0.66)	8.10 (± 0.62)	15.33 (± 0.22)	33.06 (± 0.30)
	100%	1.22	7.15	16.09	32.75
	50%	2.94	7.11	16.33	34.11
	25%	1.15	7.74	14.78	33.78
v	Ref.	1.74 (± 0.11)	11.21 (± 1.41)	24.20 (± 1.38)	50.82 (± 0.26)
	100%	1.86	9.40	24.40	51.59
	50%	2.40	9.46	25.96	50.90
	25%	1.71	11.33	23.15	50.43
w	Ref.	1.86 (± 0.60)	9.03 (± 0.31)	21.21 (± 1.27)	51.83 (± 0.38)
	100%	1.75	8.65	21.05	53.04
	50%	2.61	7.70	21.34	52.60
	25%	1.22	9.67	20.35	49.75

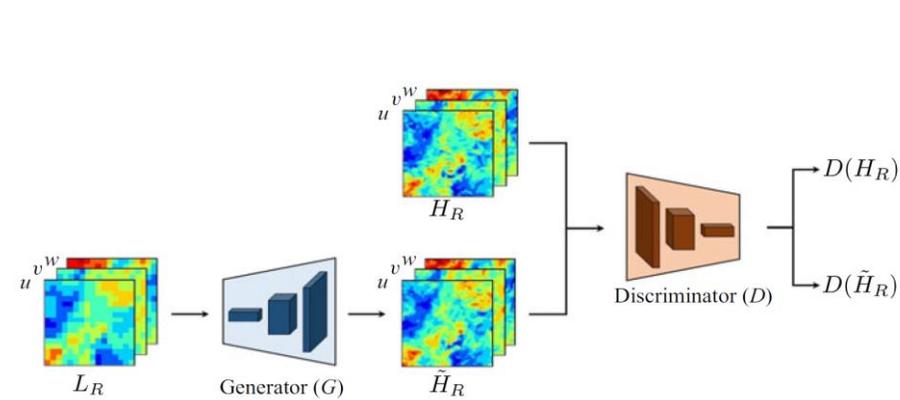
FLOW - Similar error levels with 25% of the training data!



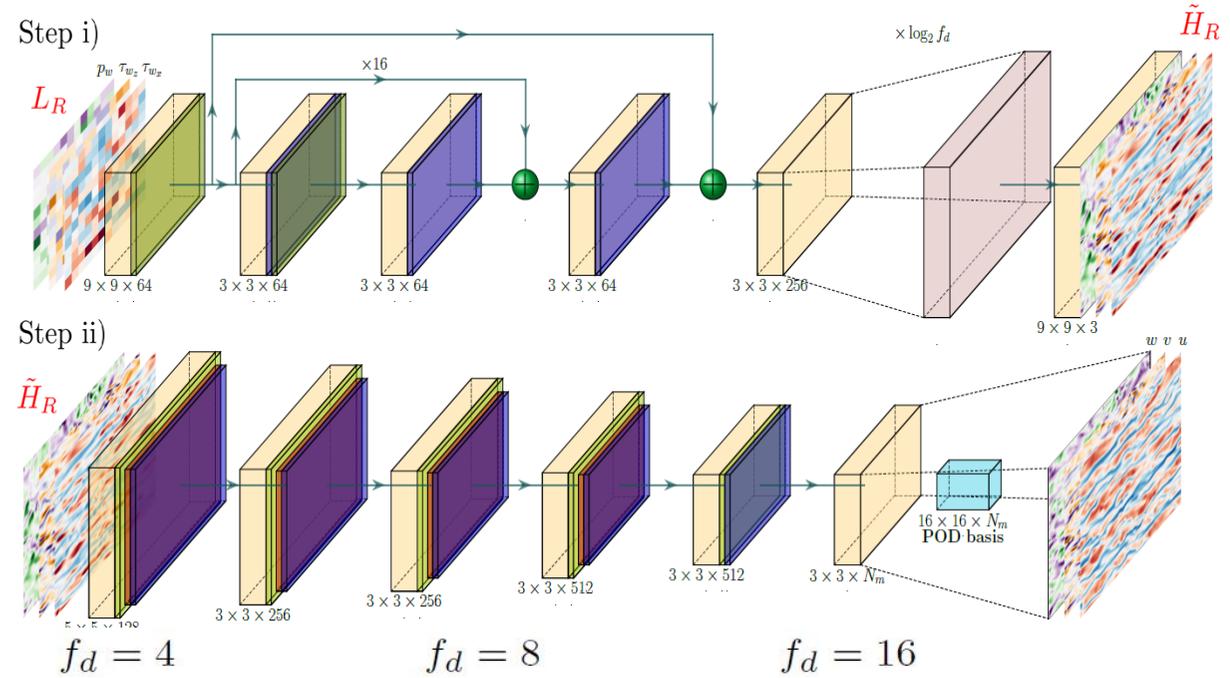
Outline of machine-learning applications to fluid mechanics

- Non-intrusive sensing in a turbulent channel via CNNs.
- **Non-intrusive sensing with coarse measurements via GANs.**
- Predictions for wall models via CNNs.
- Reduced-order models in turbulence via AEs.
- Flow control via DRL.

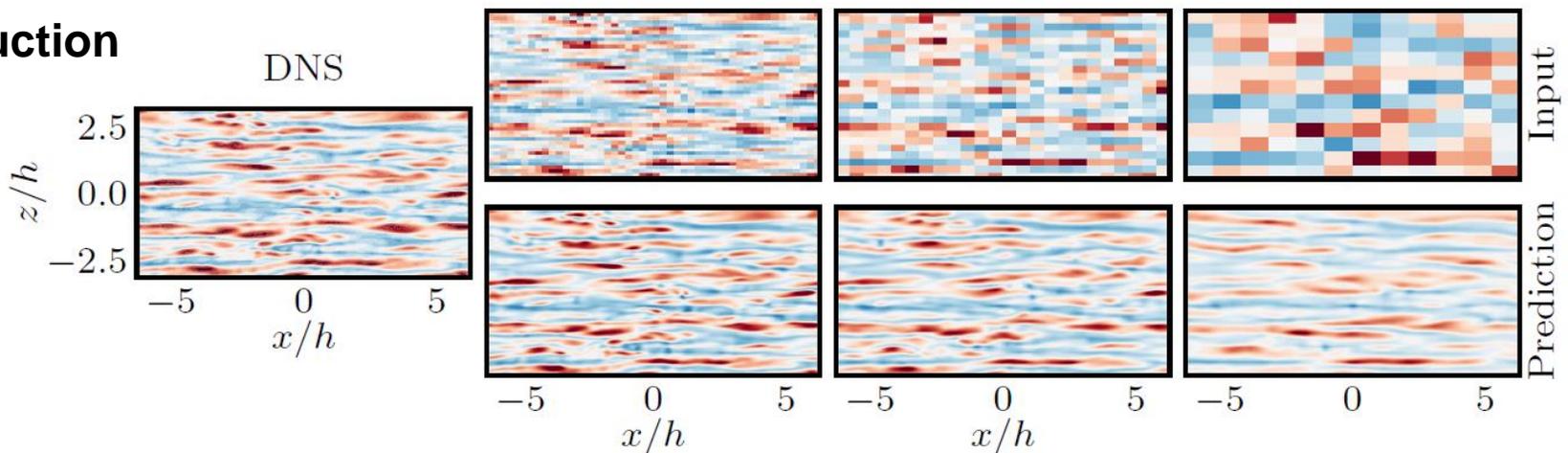
From sparse measurements to high-resolution predictions using GANs



Kim et al., (2021)



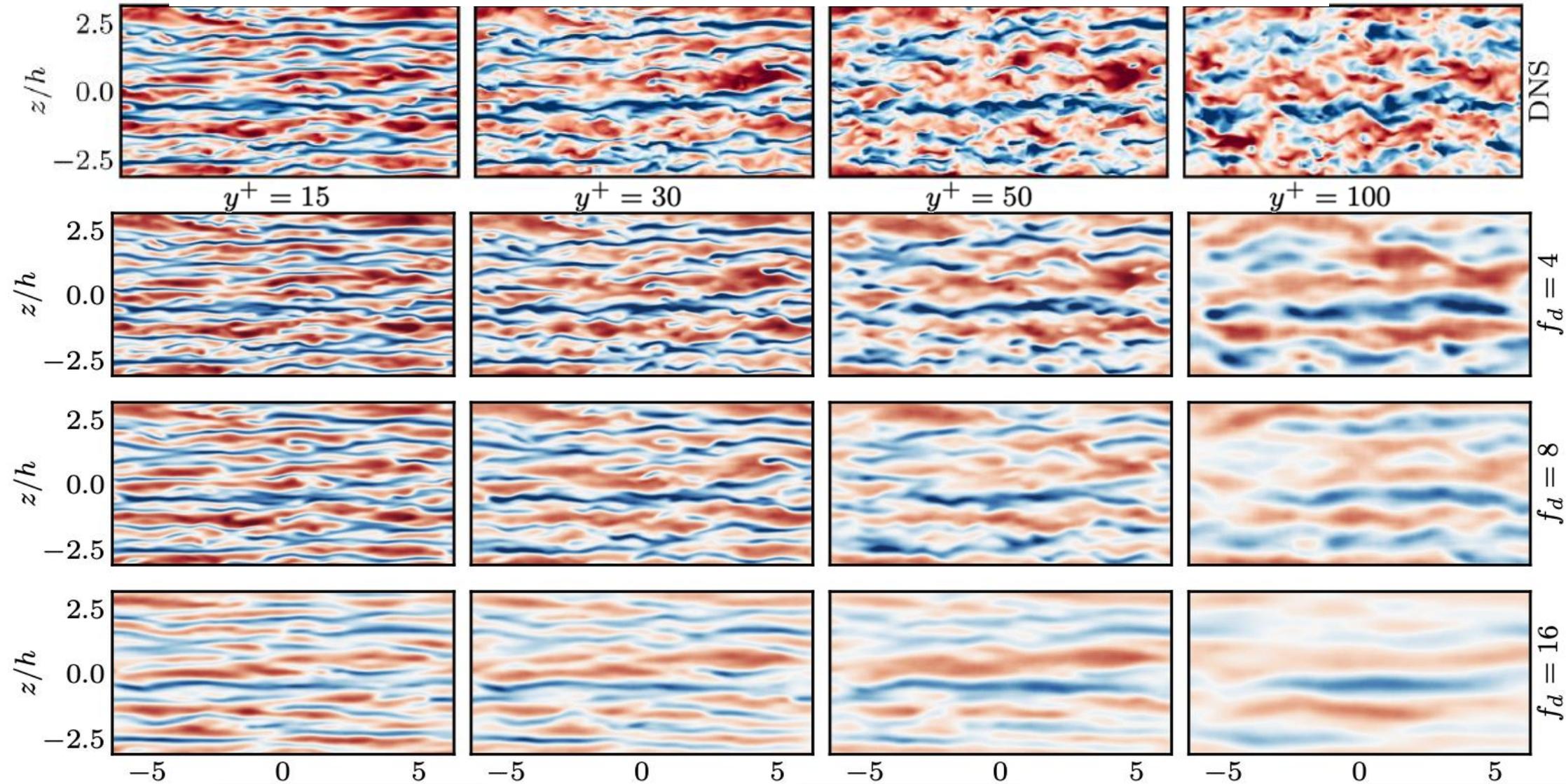
Wall reconstruction



Flow reconstruction from coarse measurements

- 1) Different range of scales in each study.
- 2) Parameter f_d is not an adequate parameter to compare turbulence databases.

$$\tilde{f}_d = f_d \sqrt{\Delta x^{+2} + \Delta z^{+2}} \quad \text{Güemes et al., Phys. Fluids 33, 075121 (2021)}$$





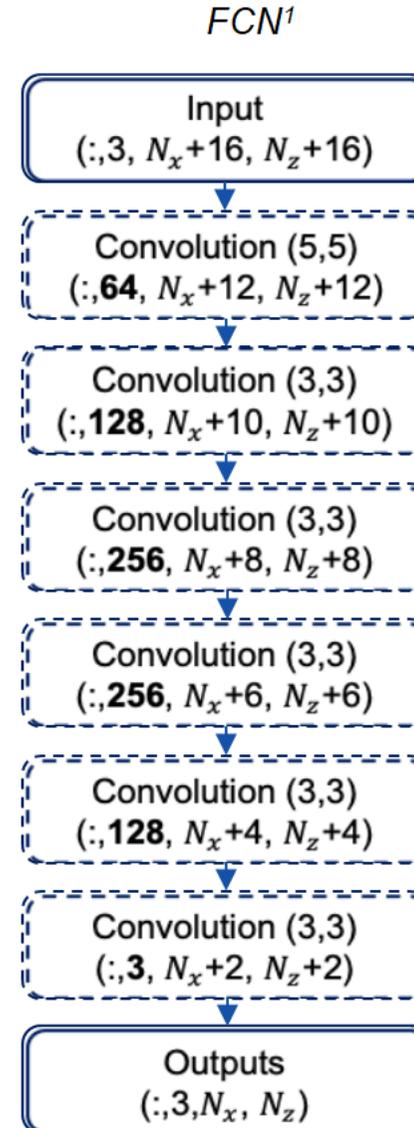
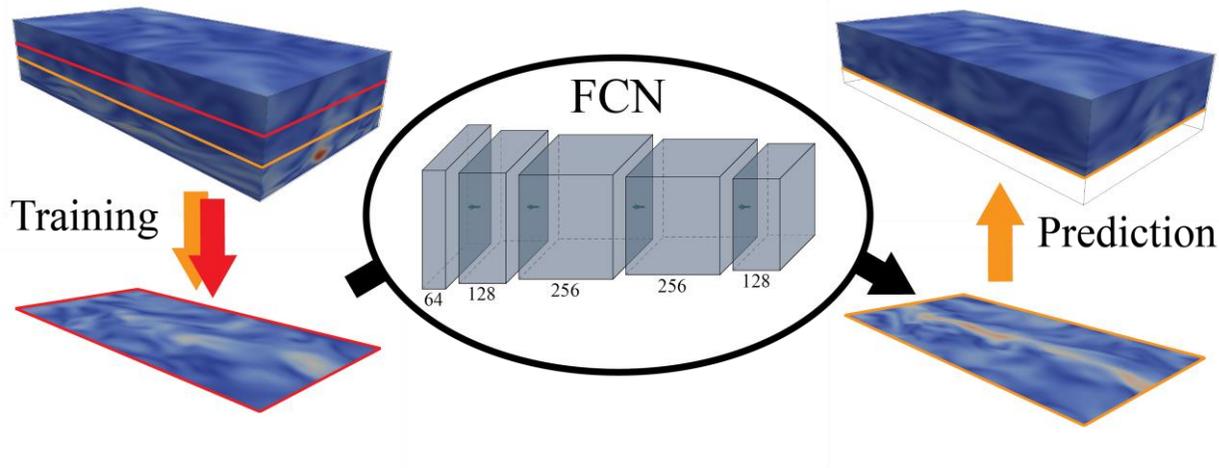
Outline of machine-learning applications to fluid mechanics

- Non-intrusive sensing in a turbulent channel via CNNs.
- Non-intrusive sensing with coarse measurements via GANs.
- **Predictions for wall models via CNNs.**
- Reduced-order models in turbulence via AEs.
- Flow control via DRL.

FCN model for predictions closer to the wall

Balasubramanian et al. (2021),
arXiv:2107.07340

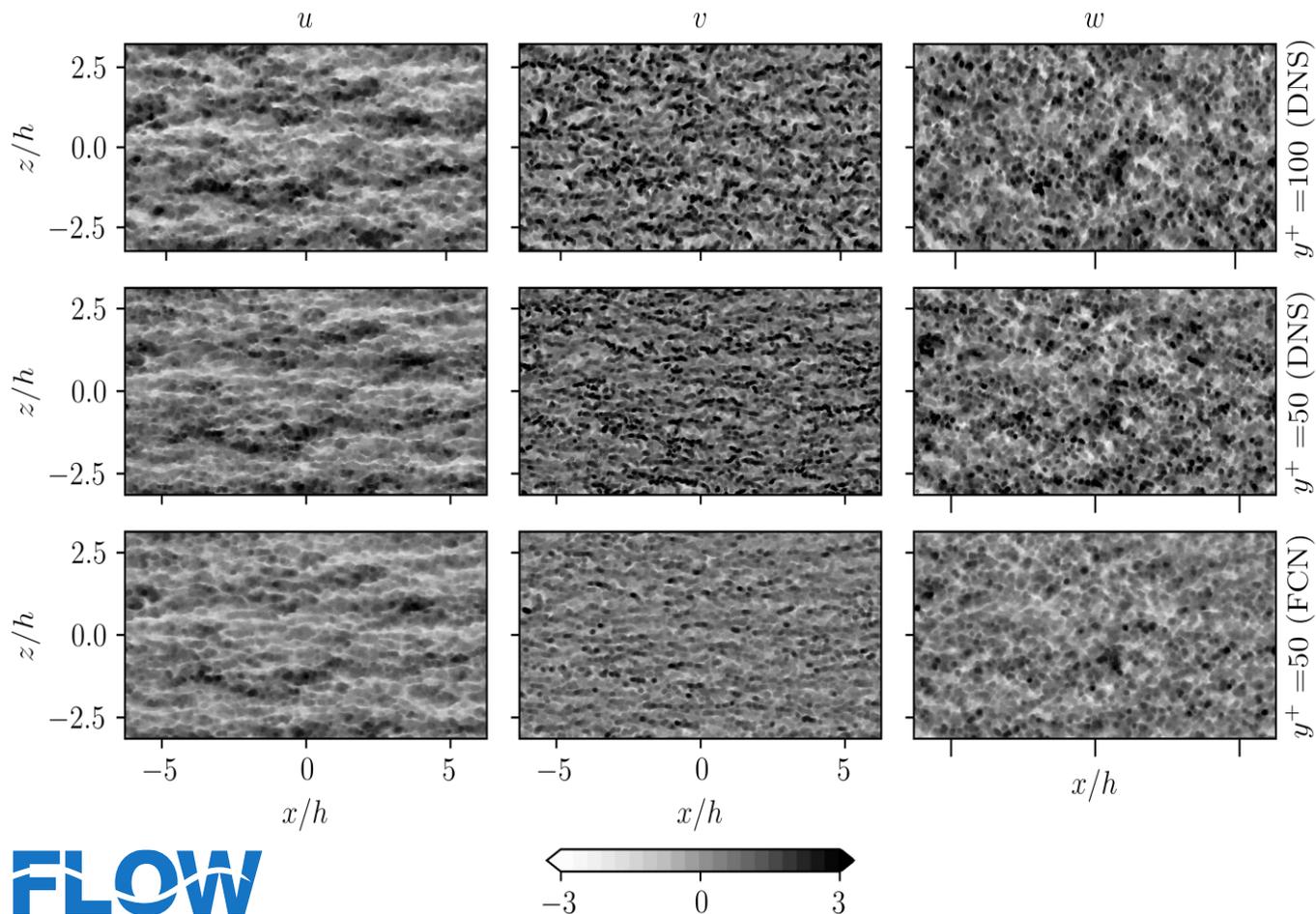
- Is it possible to predict the inner region based on the outer region?
- **Application to off-wall boundary conditions?**



Activation function - ReLU

Self similarity in the overlap region: Off-wall boundary conditions

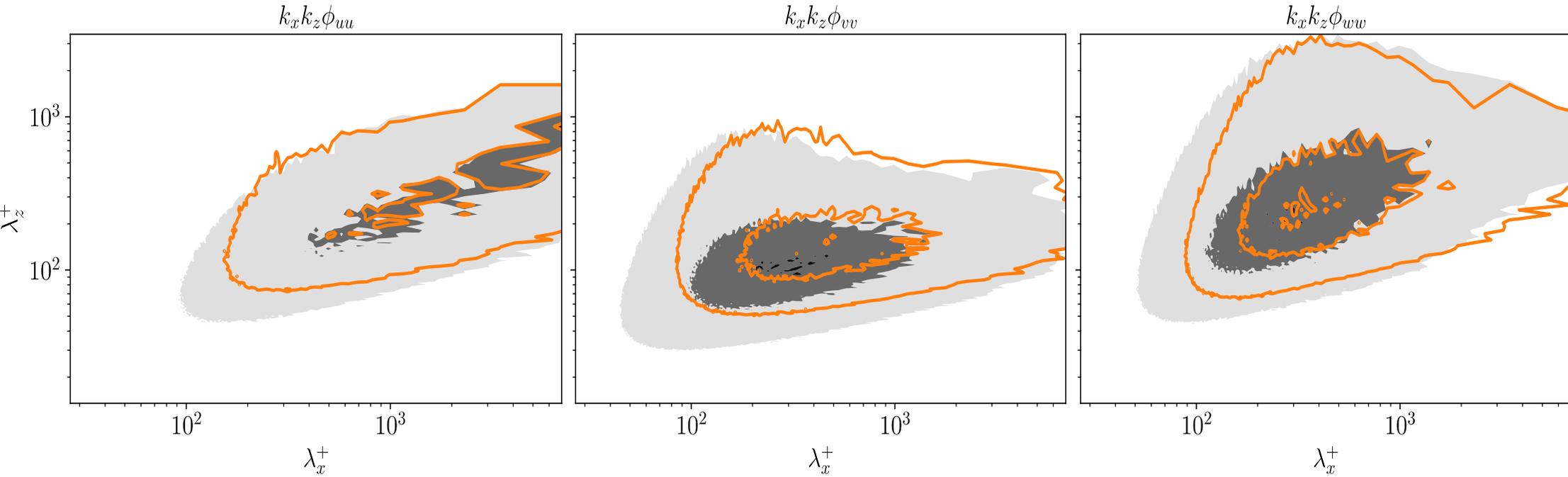
- Exploit the self similarity of scales with y .
- **Mizuno and Jiménez (2013)**: Channel at $Re_\tau=1,000$; from $y/h=0.2$ to 0.1 , rescaling.
- Predictions of $y^+=50$ based on data at 100 with CNNs, $Re_\tau=550$.



- **Small receptive field:** information from large structures cannot be captured.
- **Missing scales:** this information is not present in the large scales.

Balasubramanian
et al. (2021),
arXiv:2107.07340

Spectra and statistics at $Re_\tau=550, y^+=50$



Parameters	i		
	u	v	w
$\mathcal{L}(i_{FCN}; i_{DNS})/i_{RMS}^2$	0.365 ± 0.005	0.548 ± 0.005	0.468 ± 0.006
$E_{RMS}(i)$ [%]	19.04 ± 1.17	31.8 ± 1.25	25.7 ± 1.18
$R_{FCN;DNS}$	0.801 ± 0.003	0.681 ± 0.004	0.733 ± 0.004

- **Small receptive field:** max pooling to better identify features.
- **Missing scales:** Perhaps GANs-based reconstruction (**Güemes et al., Phys. Fluids, 2021**).

Balasubramanian et al. (2021), arXiv:2107.07340



Outline of machine-learning applications to fluid mechanics

- Non-intrusive sensing in a turbulent channel via CNNs.
- Non-intrusive sensing with coarse measurements via GANs.
- Predictions for wall models via CNNs.
- **Reduced-order models in turbulence via AEs.**
- Flow control via DRL.



Non-linear orthogonal modal decomposition in turbulent flows via autoencoders

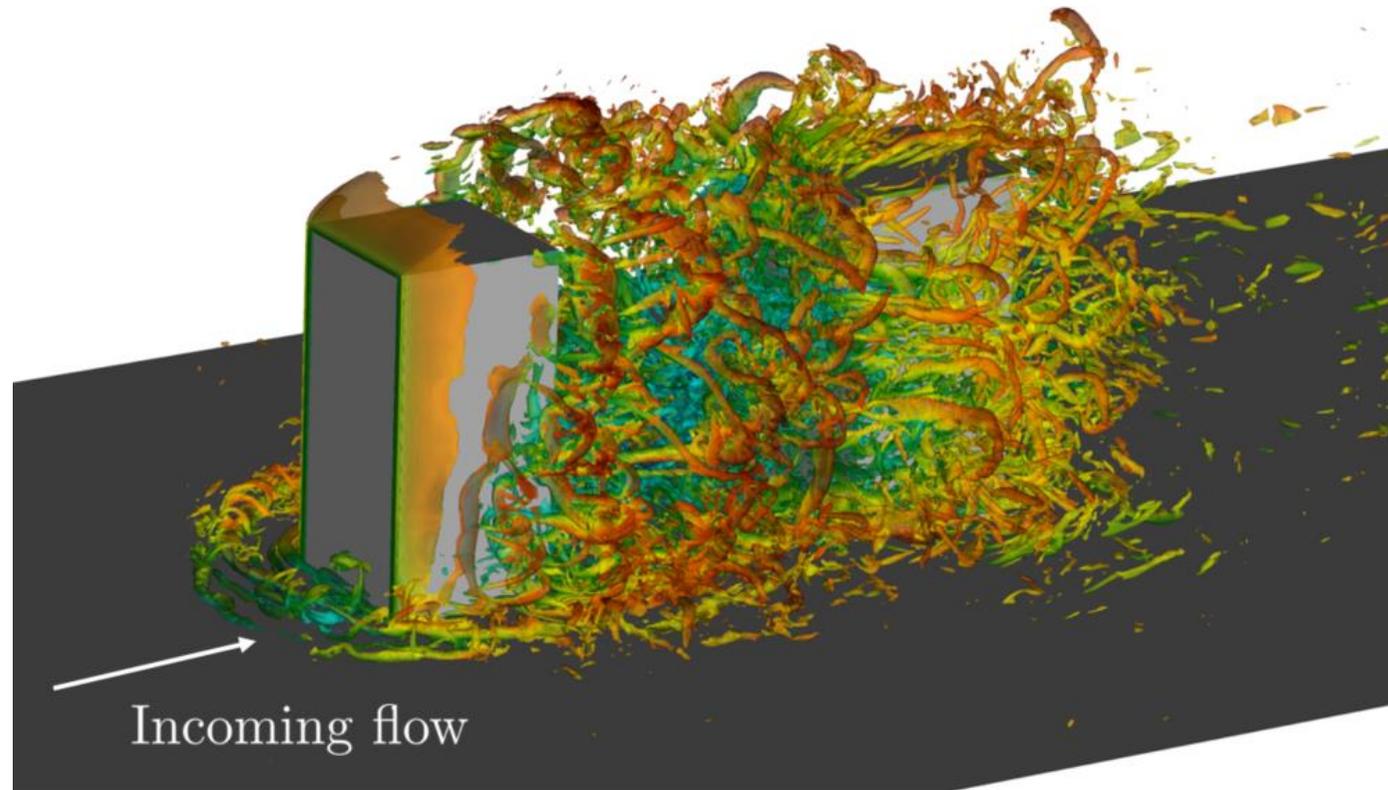
Objectives:

- Learning a compact latent representation of high-dimensional data by accounting for the **non-linearity** in the low-dimensional mapping.
- Ranking modes based on their contribution to the reconstruction (**Optimality**).
- Learning a compact, **near-orthogonal** and parsimonious latent representation of high-dimensional data by minimizing the correlation between the latent variables, as well as penalizing the size of the latent vector.
- Extraction of **interpretable non-linear modes**.

Turbulent flow in a simplified urban environment

We employ a database (Torres et al., 2021; Lazpita et al., 2022) of the flow through a simplified urban environment:

- Well-resolved large-eddy simulation (LES) using Nek5000.
- The x–z cross-sectional velocity fields at $y = 0.5h$ are extracted and used as the input data.
- **1,200 instantaneous fields.**
- Number of grid points in x and z: $(N_x, N_z) = (96, 192)$.



Convolutional-neural-network-based autoencoders (CNN-AEs)

An autoencoder comprises two parts: an encoder that maps the input data to a low-dimensional latent space $\mathbf{x} \mapsto \mathbf{r}$, and a decoder that projects the latent vector \mathbf{r} back to the reference space $\mathbf{r} \mapsto \tilde{\mathbf{x}}$.

$$\mathcal{F} = \mathcal{D} \circ \mathcal{E}, \quad \tilde{\mathbf{x}} = \mathcal{F}(\mathbf{x}; \mathbf{w}), \quad \mathcal{L}_{\text{rec}} = \epsilon(\mathbf{x}, \tilde{\mathbf{x}}), \quad (7)$$

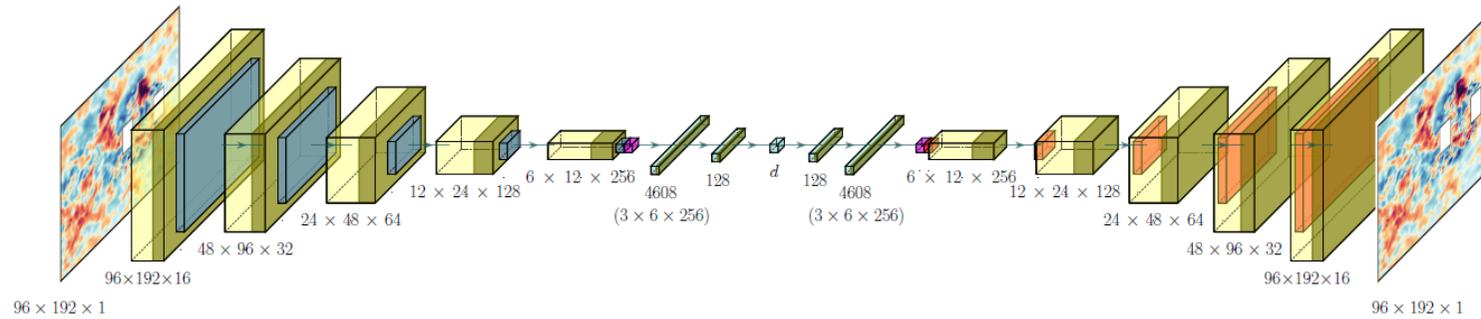


Figure: Schematic view of the CNN-AE architecture. The color coding for each layer is: 2D-convolution (yellow), tanh activation (green), max pooling (blue), reshape (purple), fully-connected layer (cyan), upsampling (orange).

- **Latent variables** contain temporal information.
- **Obtain modes** by using decoder, setting rest of latent variables to zero.

CNN-based hierarchical autoencoders (CNN-HAEs)

Fukami et al. (2020) proposed a CNN-based hierarchical autoencoder for modal decomposition of fluid flows in order to extract the modes ranked in terms of their contribution to the reconstruction:

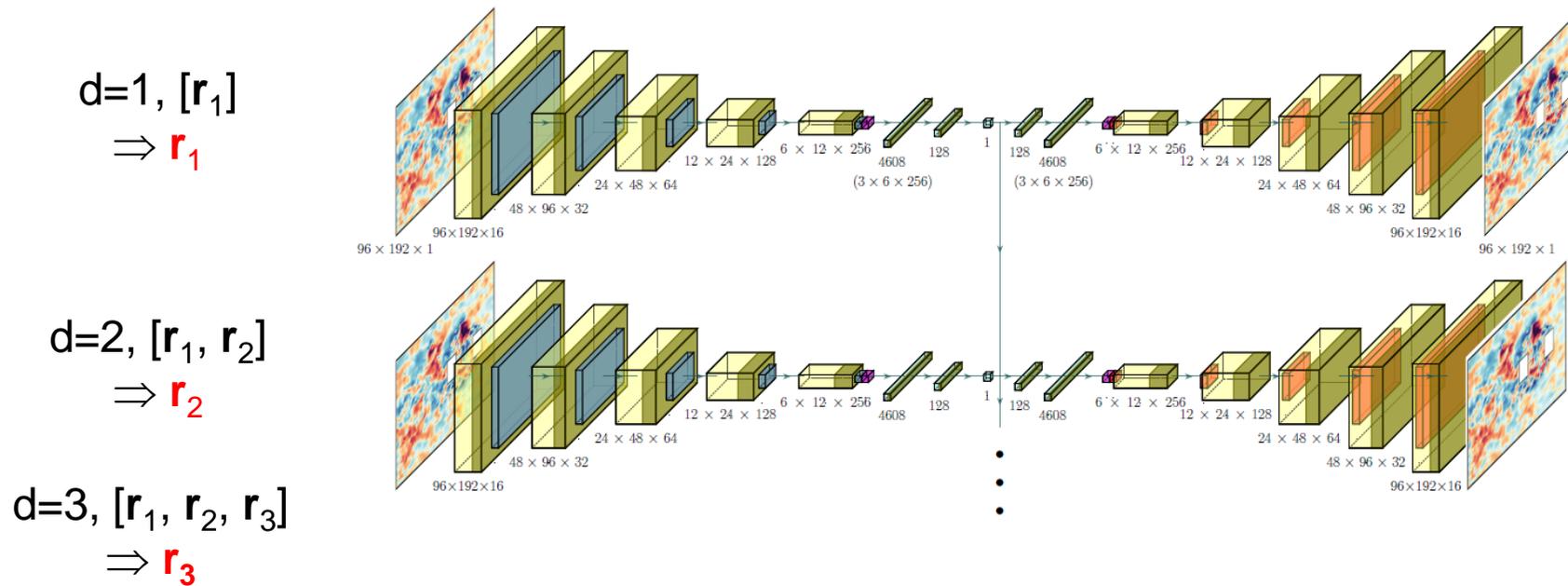


Figure: Schematic view of the CNN-HAE architectures with the same color coding as CNN-AE.



CNN-based β -variational autoencoders (CNN- β VAE)

Introducing stochasticity

Let us consider:

- $p(\mathbf{x})$ is the distribution of the **original data**.
- $p(\mathbf{r})$ is the distribution of the data in the **latent space**.
- We aim to maximize:

$$p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{x}|\mathbf{r})p(\mathbf{r})d\mathbf{r},$$

- $p_{\theta}(\mathbf{x})$ is the **marginal likelihood**: approximation of $p(\mathbf{x})$ with parameters θ .
- In VAEs, $p_{\theta}(\mathbf{x}|\mathbf{r})$ is a **probabilistic decoder** (generative model).
- In VAEs, $q_{\theta}(\mathbf{r}|\mathbf{x})$ is a **probabilistic encoder** (recognition model).



CNN-based β -variational autoencoders (CNN- β VAE)

Introducing stochasticity

We assume that the **encoder** $q_{\theta}(\mathbf{r} \mid \mathbf{x})$ is a Gaussian distribution:

$$\log q_{\phi}(\mathbf{r} \mid \mathbf{x}) = \log \mathcal{N}(\mathbf{r}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathbf{I}),$$

We sample from $q_{\theta}(\mathbf{r} \mid \mathbf{x})$ using an auxiliary normally-distributed random number:

$$\mathbf{r} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon} \text{ where } \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

The β VAE loss function is:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathcal{L}_{\text{rec}} - \frac{\beta}{2} \sum_{i=1}^d (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2)$$

β is a **penalization factor** which promotes learning **statistically-independent latent variables** (minimizing the distance between $p(\mathbf{r})$ and the product of its marginals), and also penalizing the **size of the latent vector \mathbf{d}** . **Disentangled and parsimonious latent space.**



Eivazi et al., Expert Syst. Appl. 202, 117038 (2022)

CNN-based β -variational autoencoders (CNN- β VAE)

Introducing stochasticity

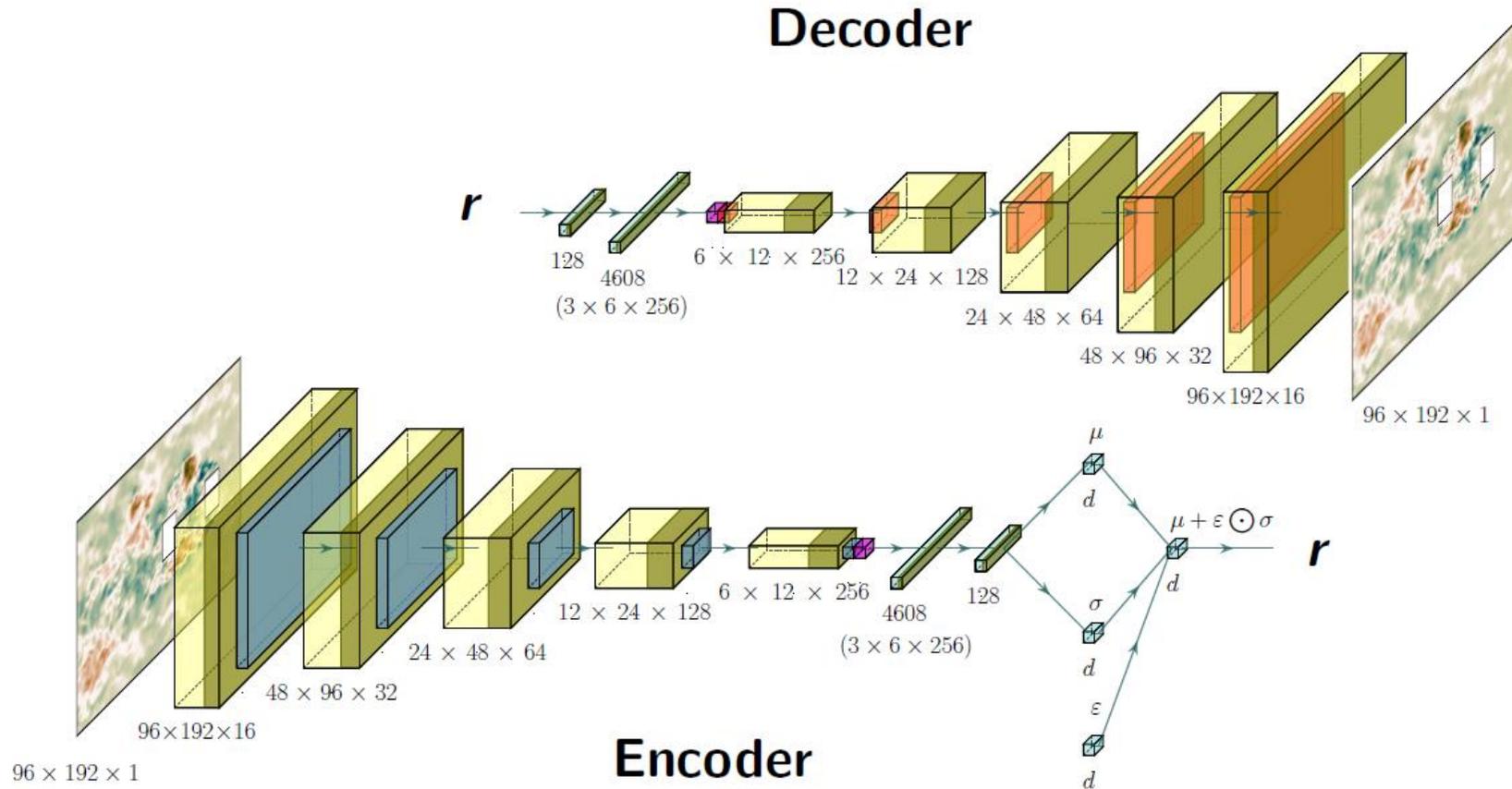


Figure: Schematic view of the CNN- β VAE with the same color coding as CNN-AE.

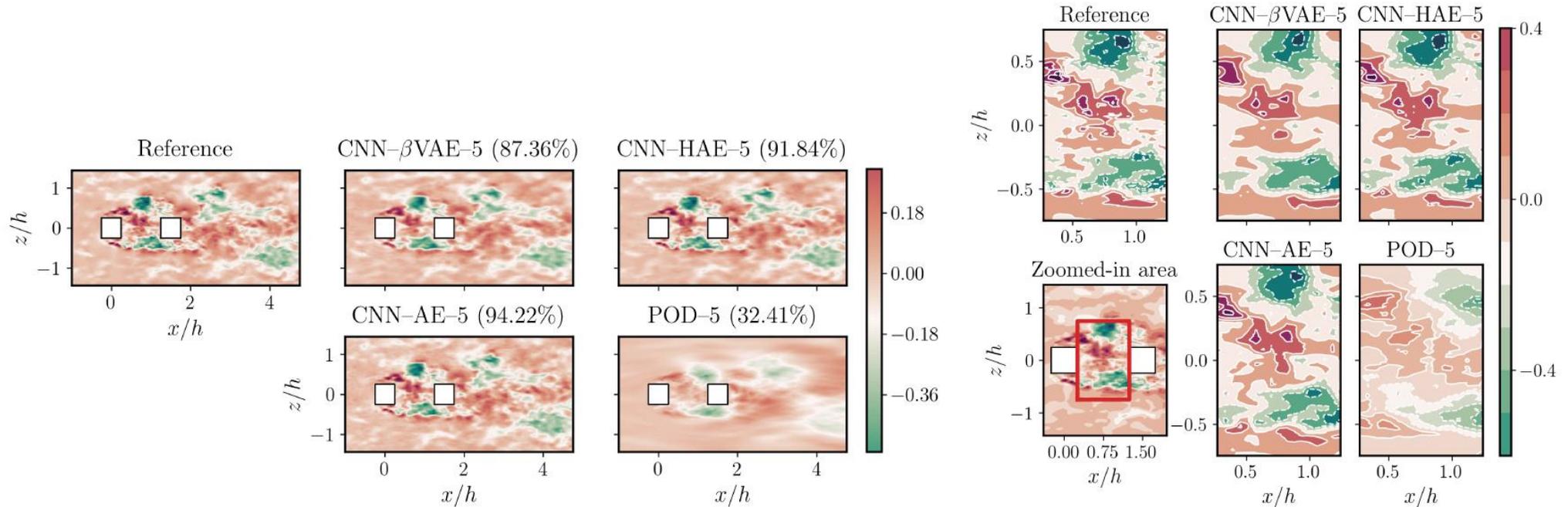


Figure: Reconstruction of the fluctuating component of the streamwise velocity obtained from different methods, as indicated in each panel, in comparison with the reference data. The value in brackets on each panel indicates the obtained E_u .

Orthogonality: determinant of the cross-correlation matrix

Our results show that it is possible to motivate the independence of the latent variables using CNN- β VAEs and obtain near-orthogonal modes, where $\det_{\mathbf{R}}$ is equal to 99.20 for the CNN- β VAE method and 87.59 for the CNN-AE technique.

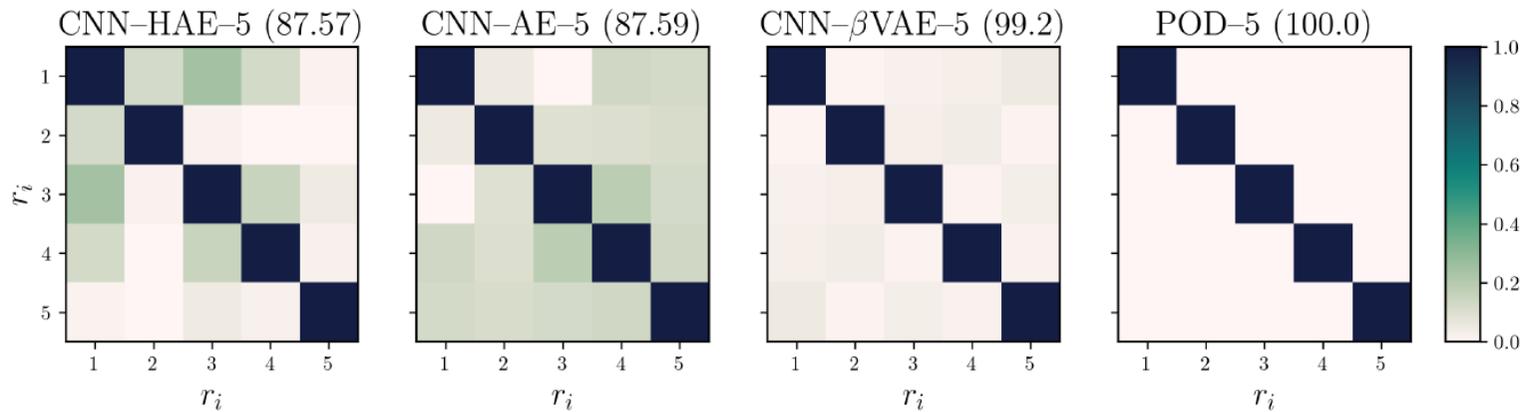


Figure: Correlation matrix \mathbf{R} for the latent variables obtained from different models as indicated on the panels. The value in brackets indicates the corresponding $\det_{\mathbf{R}}$ for each case.

Effect of the penalization factor β

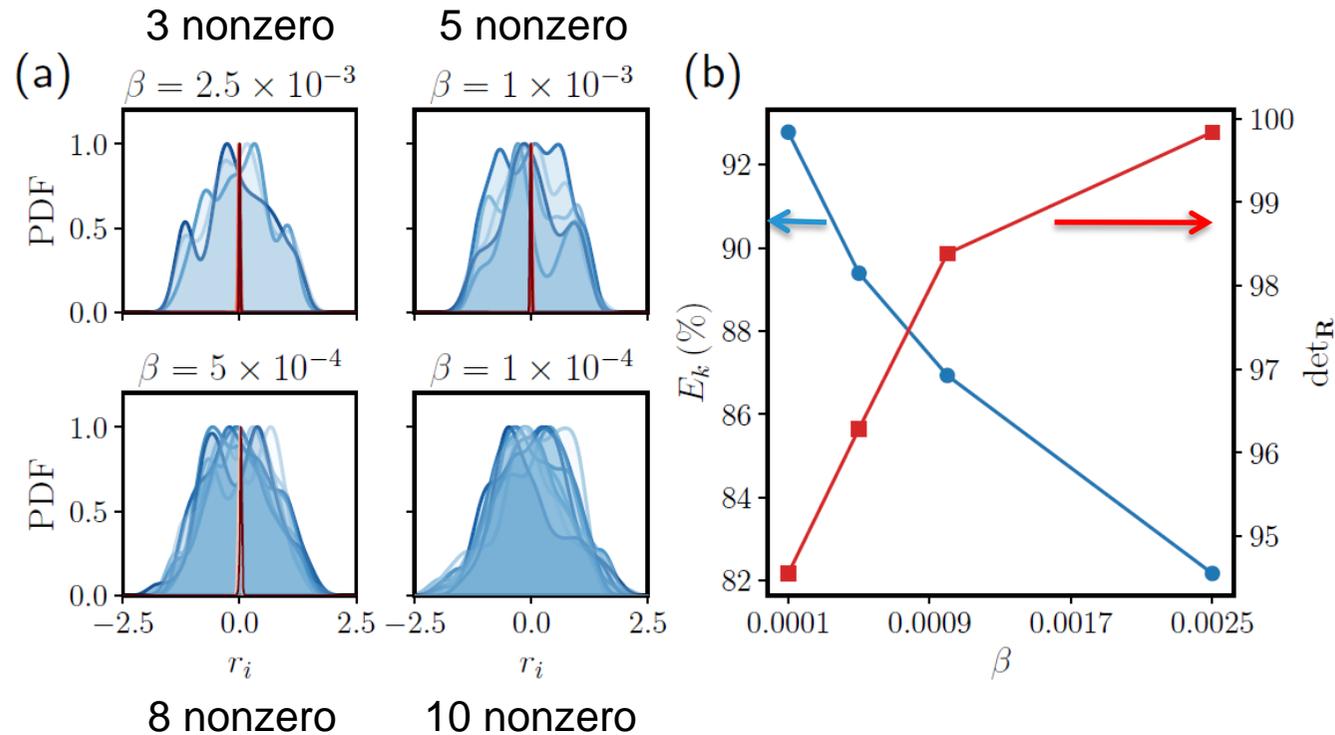


Figure: Effects of the penalization factor β on the performance of the CNN- β VAE models. (a) Normalized probability density function (PDF) of the latent variables obtained from models with different values of β as indicated in each panel. The variables with considerably large values are colored by blue and the variables with very small deviations from zero are depicted by red. (b) Effect of β on E_k (blue) and $\det_{\mathbf{R}}$ (red).



Optimality: ranking CNN- β VAE modes and interpretability

Ranking process:

- Make all latent variables zero except 1, and take the one with highest reconstructed energy.
- The second mode is the one that, together with the first one, has best reconstruction.
- And so on for all the latent variables.

Interpretability: Orthogonal modes related to **large-scale shedding** also found in POD.

Algorithm 2: Ranking the CNN- β VAE modes.

Model trained encoder \mathcal{E} and decoder D of the CNN- β VAE.

Data x

Initialize J vector containing indices of the ranked modes.

$d \leftarrow$ vector containing indices of the modes.

$\mu, \sigma, \epsilon \leftarrow \mathcal{E}(x)$

$r \leftarrow \mu$

for j in d :

 Initialize E

 for i in $d[\sim J]$:

$I \leftarrow [i, J]$ # concatenation

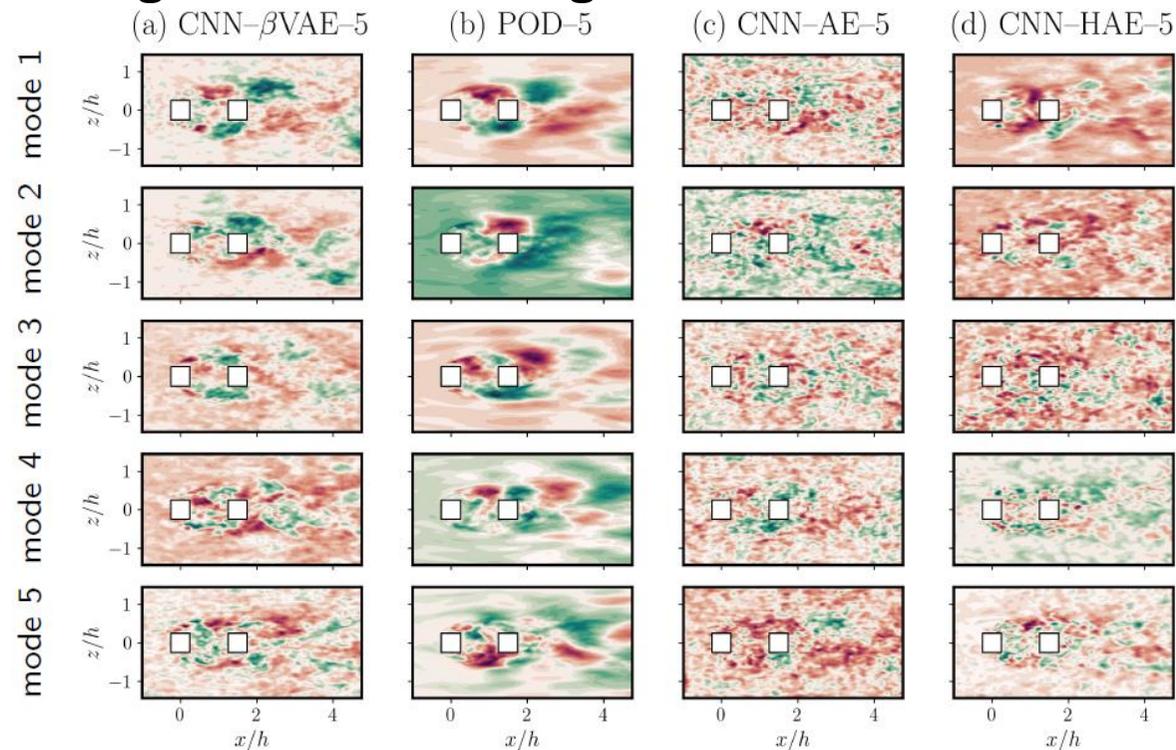
$\hat{r}_i \leftarrow$ zero out all the latent variables except members of I

$\tilde{x}_i \leftarrow D(\hat{r}_i)$

$E \leftarrow E_k(x, \tilde{x}_i)$ # append

$J \leftarrow d[\sim J][\text{argmax}(E)]$ # append

Output: J



Eivazi et al., Expert Syst. Appl. 202, 117038 (2022)



Outline of machine-learning applications to fluid mechanics

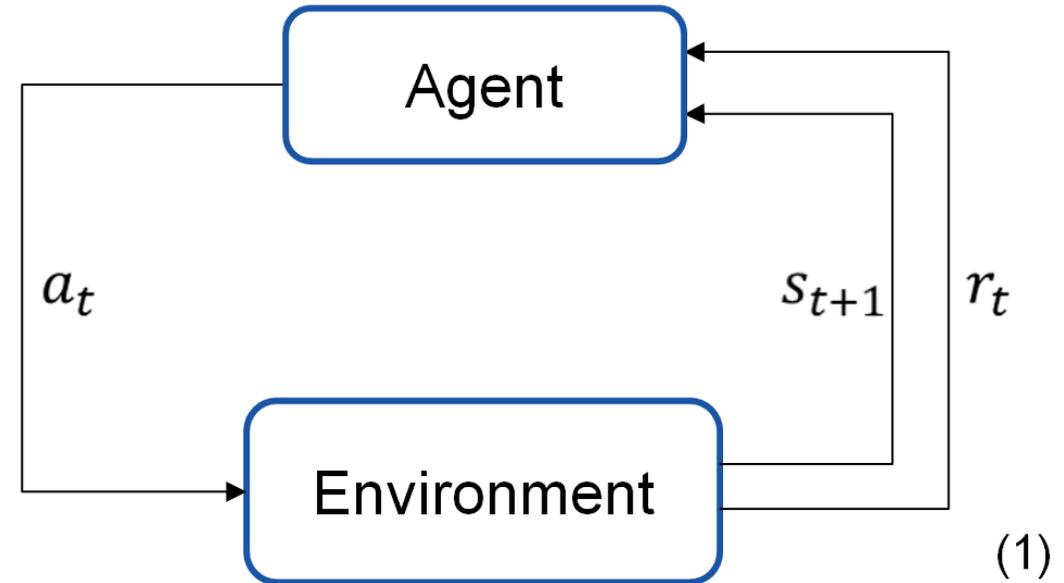
- Non-intrusive sensing in a turbulent channel via CNNs.
- Non-intrusive sensing with coarse measurements via GANs.
- Predictions for wall models via CNNs.
- Reduced-order models in turbulence via AEs.
- **Flow control via DRL.**

Deep reinforcement learning for flow control

Introduction

- Constituting elements:
 - Agent
 - Environment
 - State space $S = \{s_1, s_2, \dots, s_N\}$
 - Action space $A = \{a_1, a_2, \dots, a_M\}$
 - Transition function $\mathbb{P}(s_{t+1}|s_t, a_t)$
 - Reward function $r_t = R(s_t, a_t, s_{t+1})$
- Goal:

“Define a policy $\pi(a_t|s_t)$ that maximizes the reward”



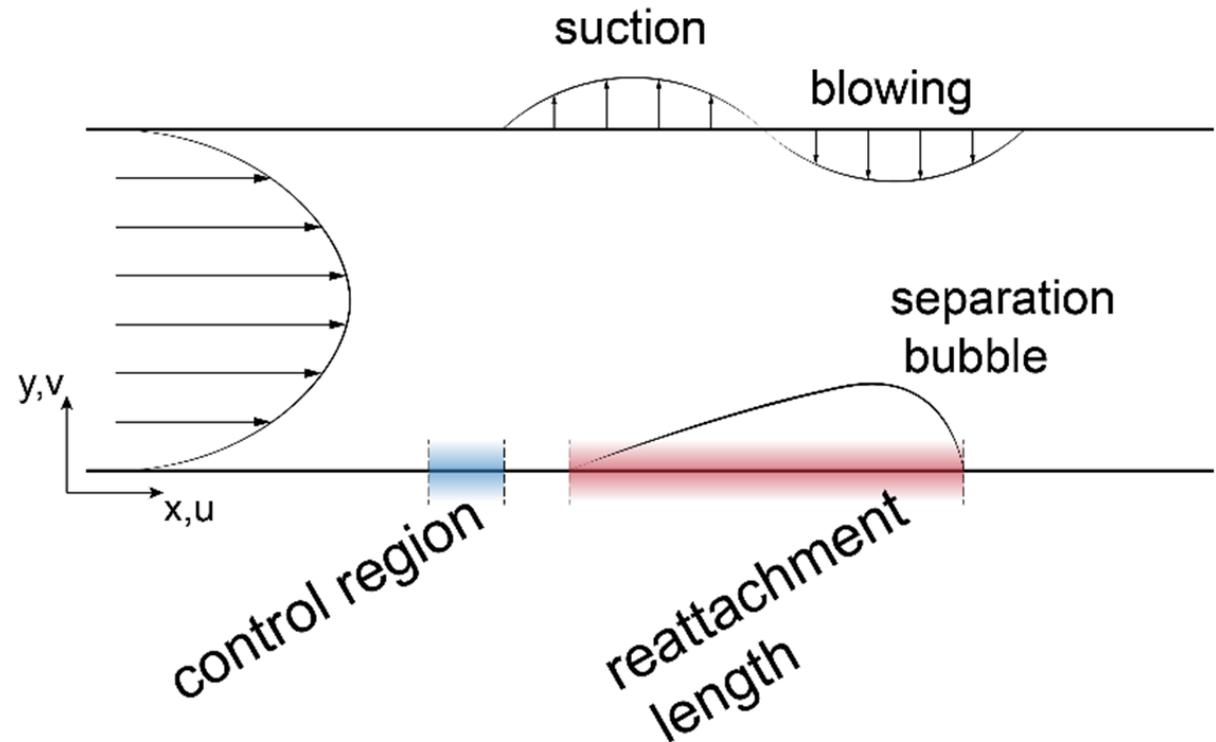
1. Sutton, R.S. Learning to predict by the methods of temporal differences. *Mach Learn* 3, 9–44 (1988)

Control of a 2D separation bubble

Case definition

2D channel flow

- Simple environment
- Harmonic forcing can be used as a baseline control
- Computationally cheap



Control of a 2D separation bubble Implementation in SIMSON

Observed state

- 24 velocity probes (streamwise velocity)

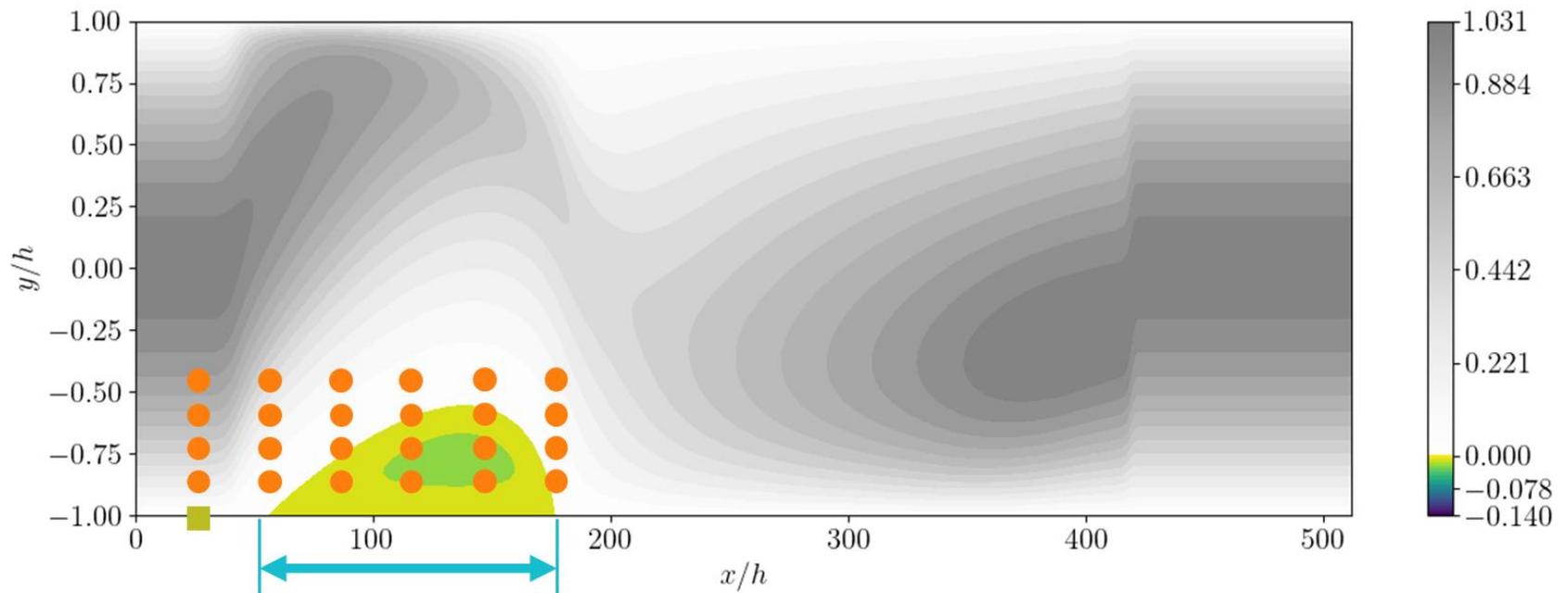
Reward

- Bubble length (lower is better)
- Moving average over time

Control

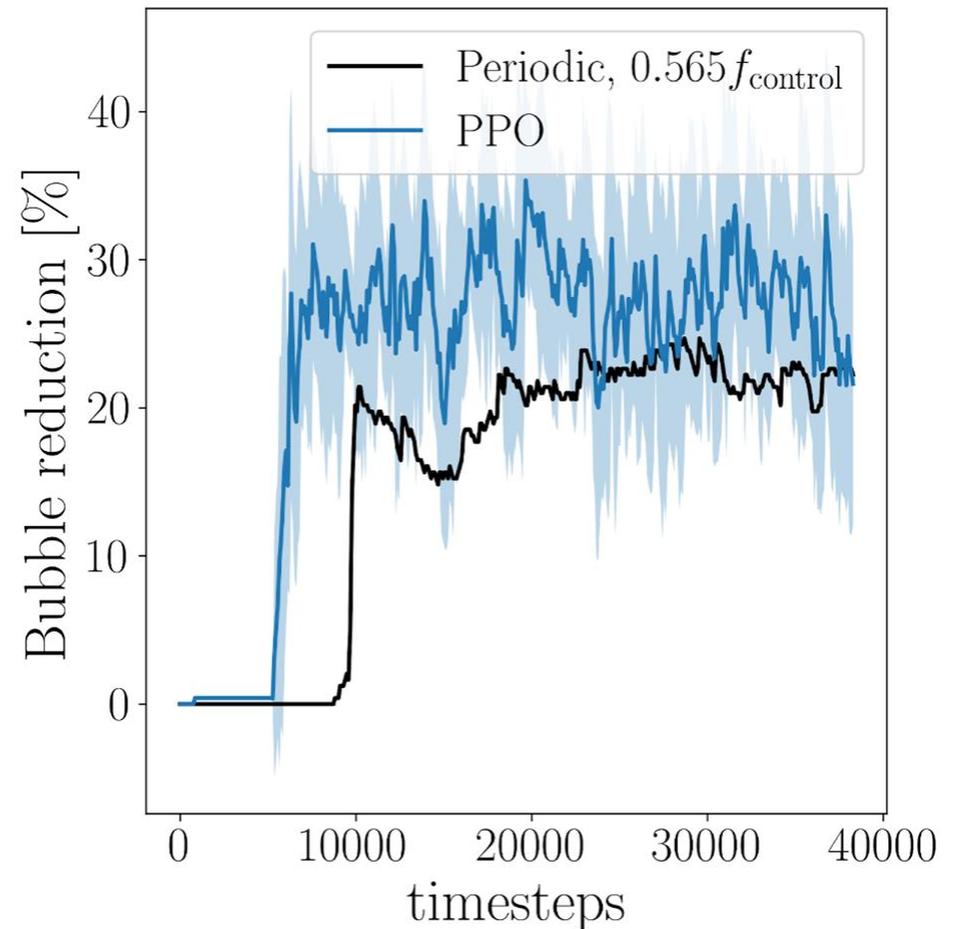
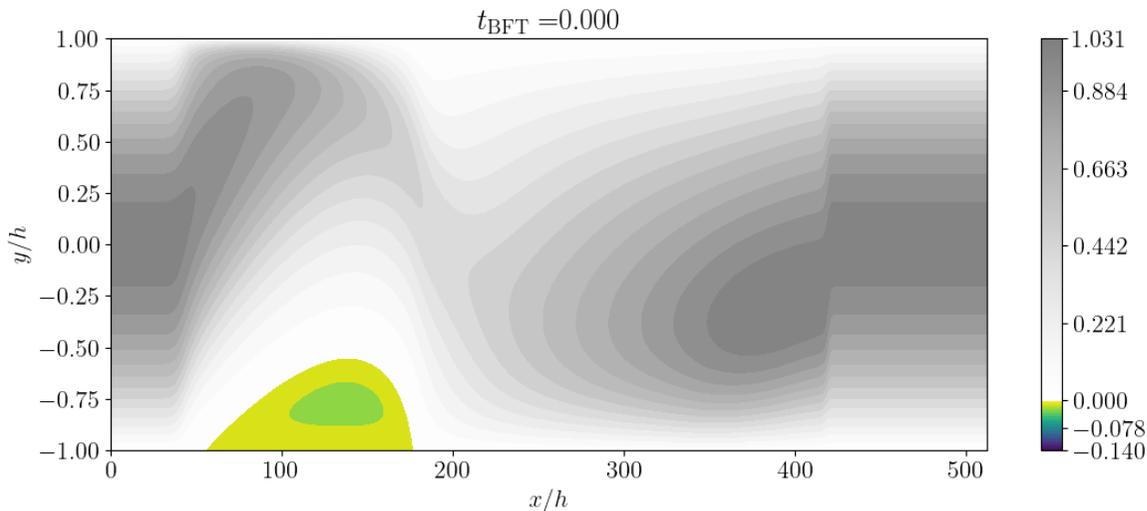
- Agent learns the parameters of a prescribed forcing function

$$f(x, t) = [0, a_y(t)] e^{-(y/y_s)^2} e^{-(x/x_s)^2}$$



Control of a 2D separation bubble Performance

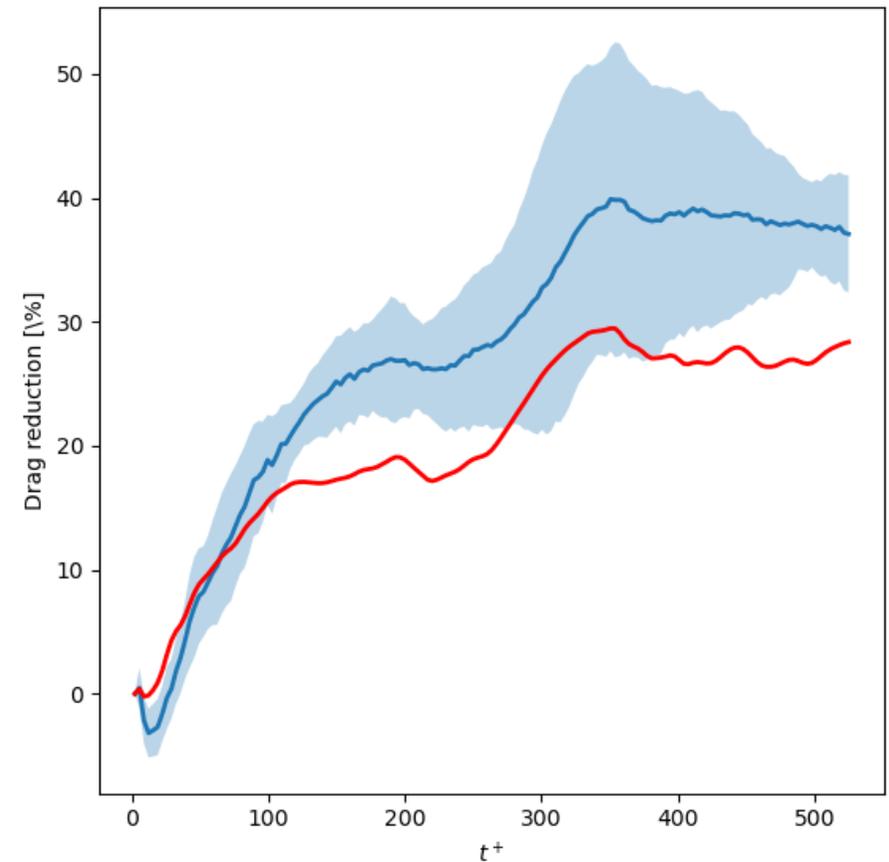
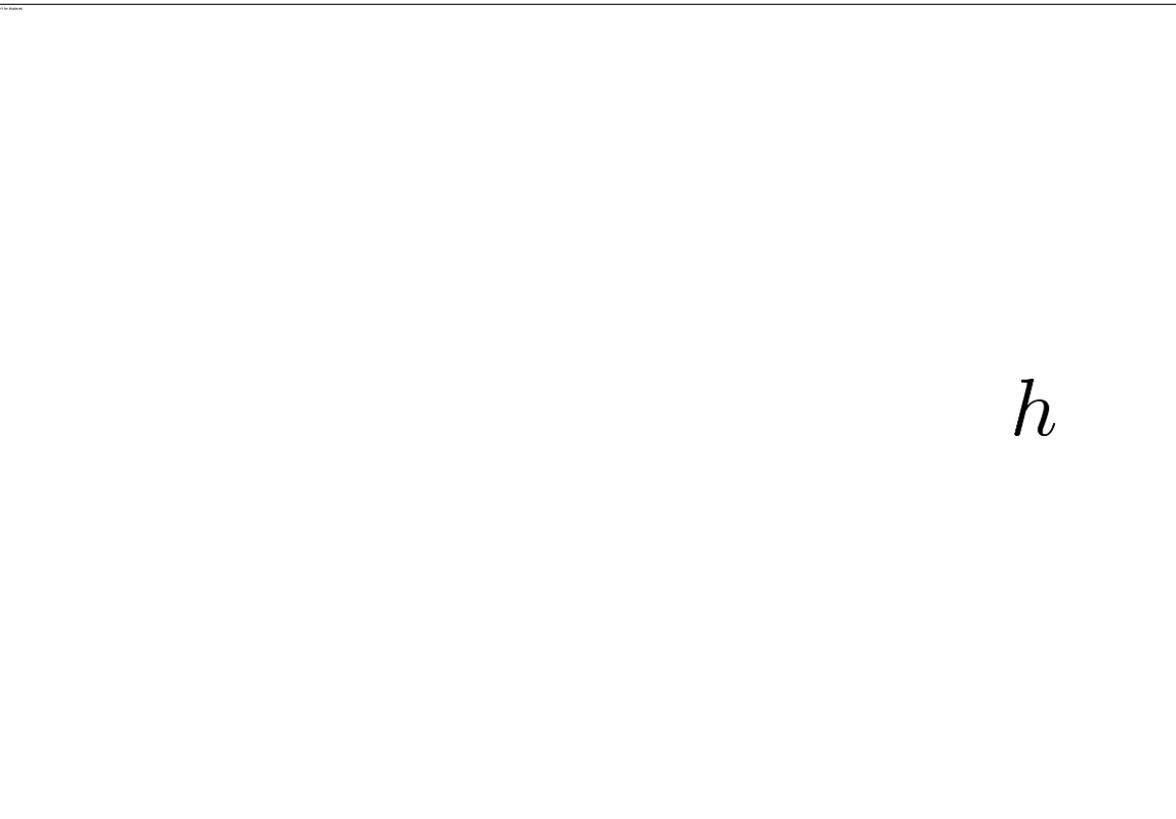
- Larger bubble reduction with **DRL (~30%)** than with **periodic forcing (~20%)**.
- Currently extending to **opposition control** in turbulent channel flow.

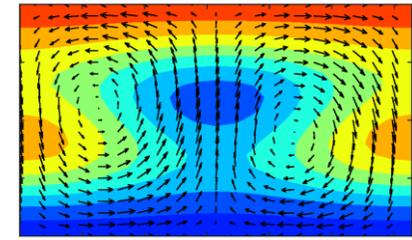




DRL and opposition control in turbulent channel flow: blowing and suction

- Preliminary results in a **minimal channel**: drag reduction from **DRL (~40%)** higher than that of **opposition control (~30%)**.
- Same policy yields similar results in **full channel for $Re_\tau=180$** .





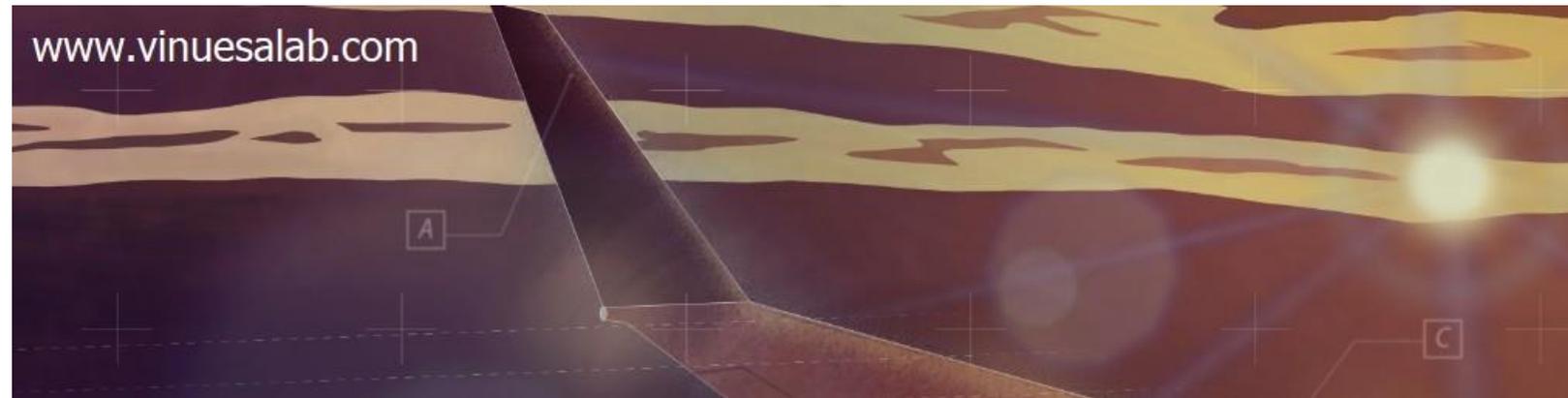
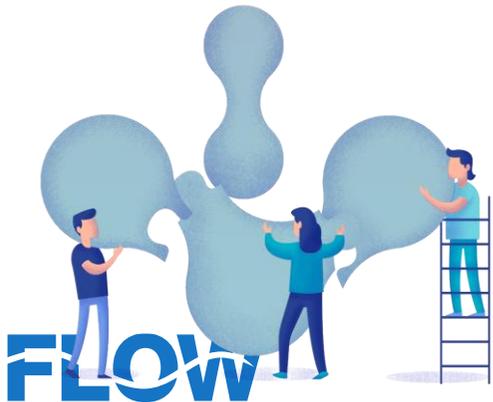
- CNNs are valuable tools for spatial reconstruction of turbulent fields.
- Excellent predictions at $y^+=15$ (<1% error), outperforming linear reconstruction methods.
- Off-wall boundary conditions and GANs for super-resolution.
- Non-linear modal decompositions using autoencoders.
- Deep-reinforcement-learning-based flow control.

#HIRING! 4 PhDs and 3 Postdocs
Check KTH Vacancies Web

 **Ricardo Vinuesa**

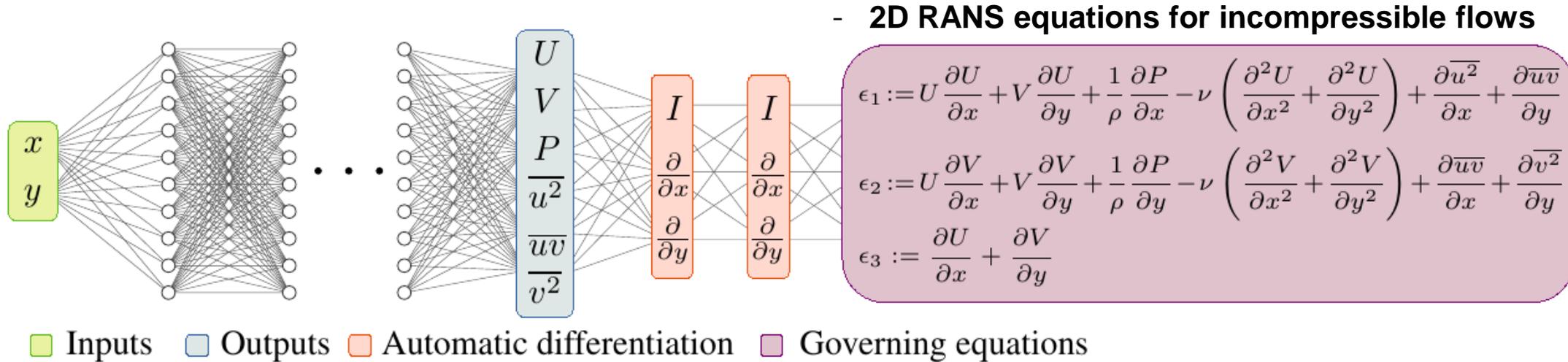
 @ricardovinuesa

Thank you for your attention!



PINNs application to RANS simulations

Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. Eivazi et al., Phys. Fluids 34, 075117 (2022)



DATA + Underdetermined system of Eqs. → Solve RANS equations

PINNs application to RANS simulations

Physics-informed neural networks for solving Reynolds-averaged Navier-Stokes equations. Eivazi et al., Phys. Fluids 34, 075117 (2022)

$$L = L_e + L_b,$$

Total loss

$$L_e = \frac{1}{N_e} \sum_{i=1}^3 \sum_{n=1}^{N_e} |\epsilon_i^n|^2,$$

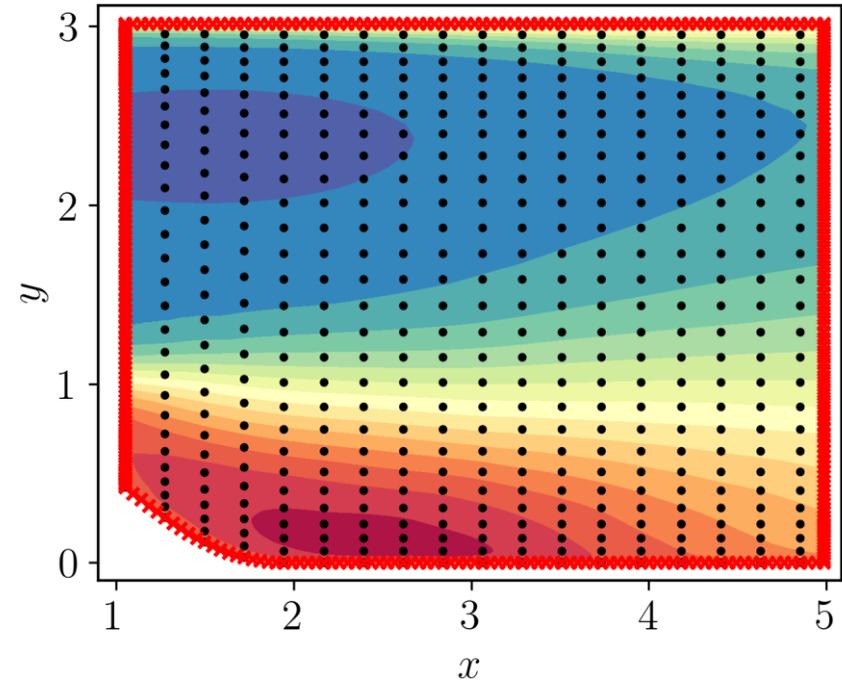
Residual of the RANS equations
(Unsupervised loss)

$$L_b = \frac{1}{N_b} \sum_{n=1}^{N_b} |\mathbf{U}_b^n - \tilde{\mathbf{U}}_b^n|^2, \text{ Loss for the BCs (Supervised loss)}$$

$$\mathbf{U}_b^n = [U_b^n, V_b^n, P_b^n, \overline{u^2}_b^n, \overline{uv}_b^n, \overline{v^2}_b^n]^T$$

N_e : Number of collocation points

N_b : Number of points on the domain boundaries



● Collocation points

✗ Boundary points



PINNs application to RANS simulations

Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. Eivazi et al., Phys. Fluids 34, 075117 (2022)

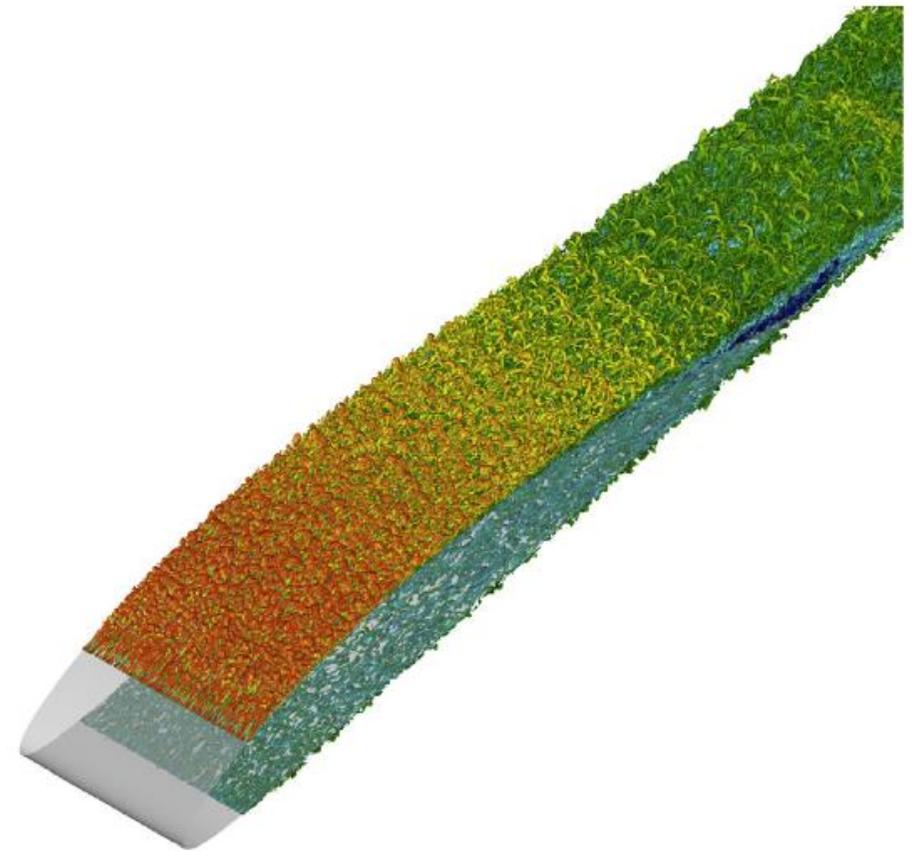
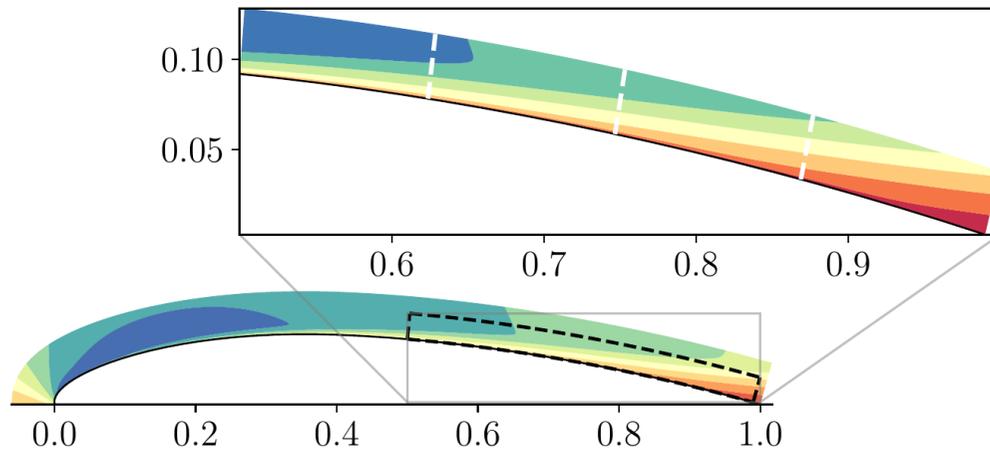
Test case 4: NACA4412 airfoil

DATA from:

Vinuesa et al., Int. J. Heat Fluid Flow 72, 86 (2018).

$$Re_c = U_\infty c / \nu = 200,000$$

$$0.5 < x/c < 1$$

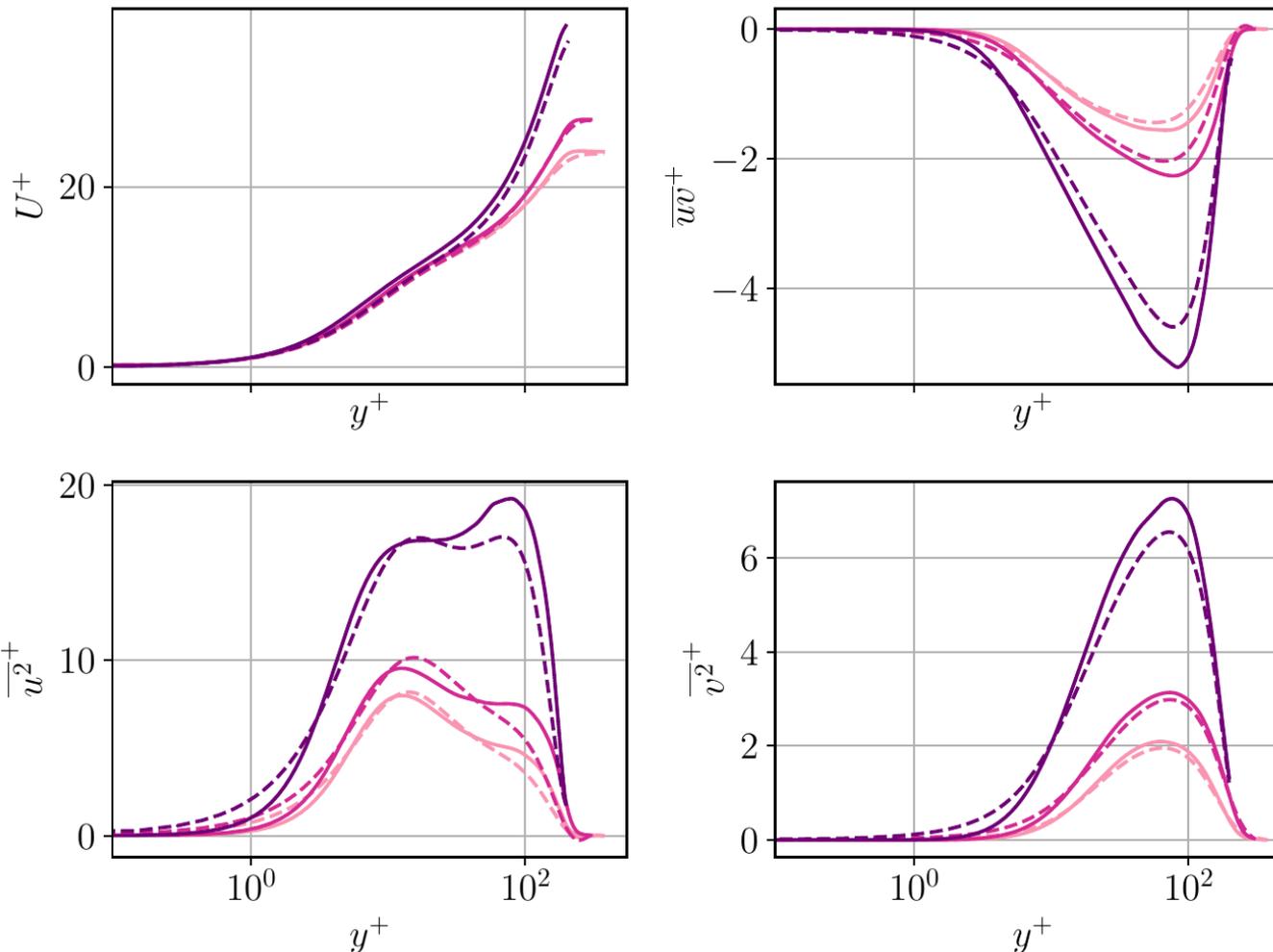




PINNs application to RANS simulations

Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. Eivazi et al., Phys. Fluids 34, 075117 (2022)

Test case 4: NACA4412 airfoil



$$E_U = 1.56\%$$

$$E_V = 2.17\%$$

$$E_P = 7.30\%$$

$$E_{\overline{u^2}} = 9.43\%$$

$$E_{\overline{uv}} = 11.36\%$$

$$E_{\overline{v^2}} = 4.69\%$$

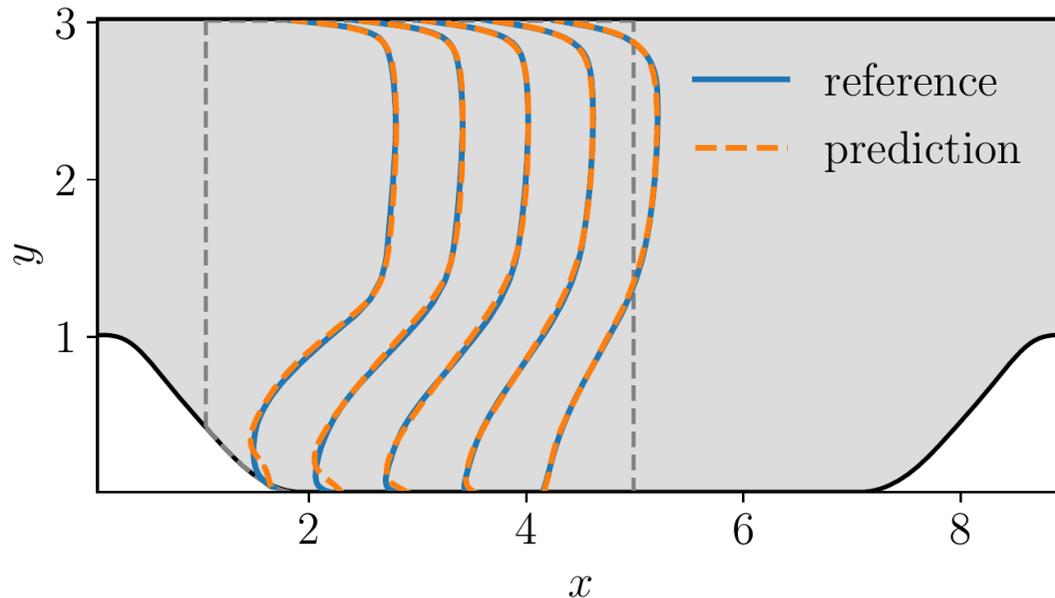
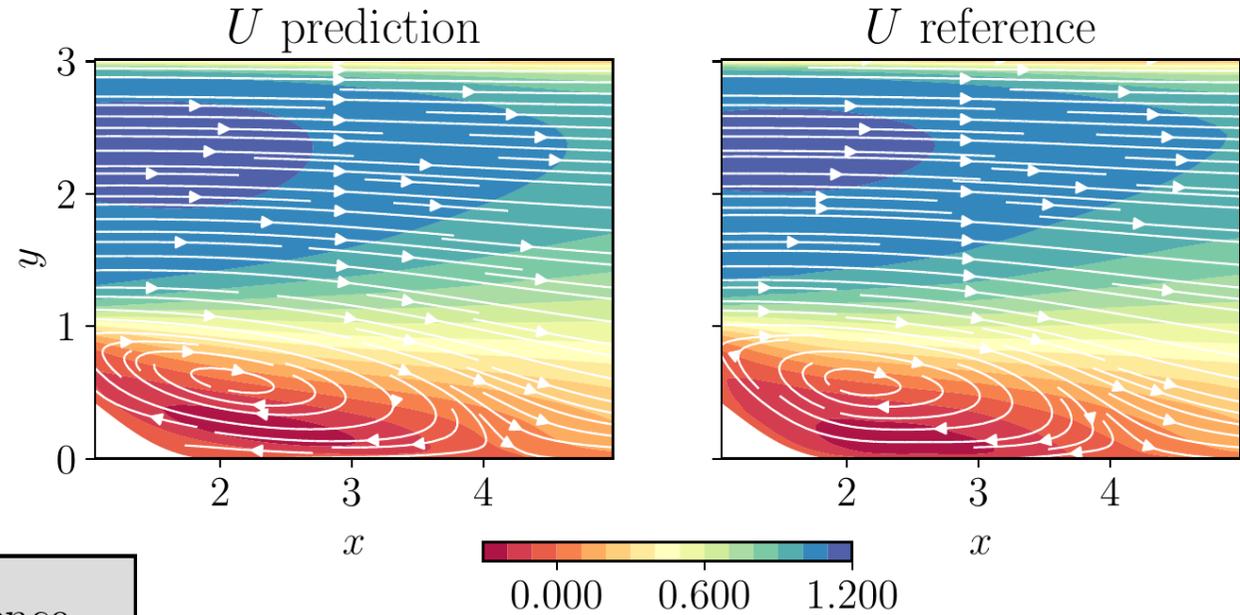
PINNs application to RANS simulations

Physics-informed neural networks for solving Reynolds-averaged Navier-Stokes equations. Eivazi et al., Phys. Fluids 34, 075117 (2022)

Test case 5: Periodic hill

$$Re_b = U_b H / \nu = 2,800$$

$$1 < x/H < 5$$



$$E_U = 2.77\%, \quad E_V = 19.70\%$$

$$E_P = 8.61\%, \quad E_{\overline{u^2}} = 28.18\%$$

$$E_{\overline{uv}} = 16.70\%, \quad E_{\overline{v^2}} = 20.24\%$$



CNN-based β -variational autoencoders (CNN- β VAE)

Let us consider:

- ▶ \mathbf{x} is a data sample in some high-dimensional space \mathcal{X} with the distribution $p(\mathbf{x})$.
- ▶ \mathbf{r} is a vector of latent variables in a low-dimensional space \mathcal{R} with the probability density function (PDF) $p(\mathbf{r})$.
- ▶ $f(\mathbf{r}; \boldsymbol{\theta})$ is a family of deterministic functions, parameterized by a vector $\boldsymbol{\theta}$ in some space Θ in such a way that $f : \mathcal{R} \times \Theta \mapsto \mathcal{X}$.

We aim to maximize:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{r})p(\mathbf{r})d\mathbf{r}, \quad (8)$$

where $p_{\boldsymbol{\theta}}(\mathbf{x})$ is the so-called marginal likelihood, which is the approximation of $p(\mathbf{x})$ with parameters $\boldsymbol{\theta}$.



CNN-based β -variational autoencoders (CNN- β VAE)

VAEs define another probability distribution $q_\phi(\mathbf{r}|\mathbf{x})$, a so-called probabilistic encoder (or recognition model). In a similar vein $p_\theta(\mathbf{x}|\mathbf{r})$ is referred to as a probabilistic decoder (or generative model). The marginal likelihood for each \mathbf{x} can be defined as:

$$\log p_\theta(\mathbf{x}) = D_{\text{KL}}(q_\phi(\mathbf{r}|\mathbf{x})||p_\theta(\mathbf{r}|\mathbf{x})) + \mathcal{C}(\theta, \phi; \mathbf{x}), \quad (9)$$

The KL-divergence is non-negative, which indicates that the second RHS term $\mathcal{C}(\theta, \phi; \mathbf{x})$ is a lower bound on the marginal likelihood, and can be written as:

$$\log p_\theta(\mathbf{x}) \geq \mathcal{C}(\theta, \phi; \mathbf{x}) = -D_{\text{KL}}(q_\phi(\mathbf{r}|\mathbf{x})||p_\theta(\mathbf{r})) + \mathbb{E}_{q_\phi(\mathbf{r}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{r})]. \quad (10)$$

This term is usually called evidence lower bound (ELBO).



CNN-based β -variational autoencoders (CNN- β VAE)

We assume $q_\phi(\mathbf{r}|\mathbf{x})$ to be a Gaussian distribution,

$$\log q_\phi(\mathbf{r}|\mathbf{x}) = \log \mathcal{N}(\mathbf{r}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathbf{I}), \quad (11)$$

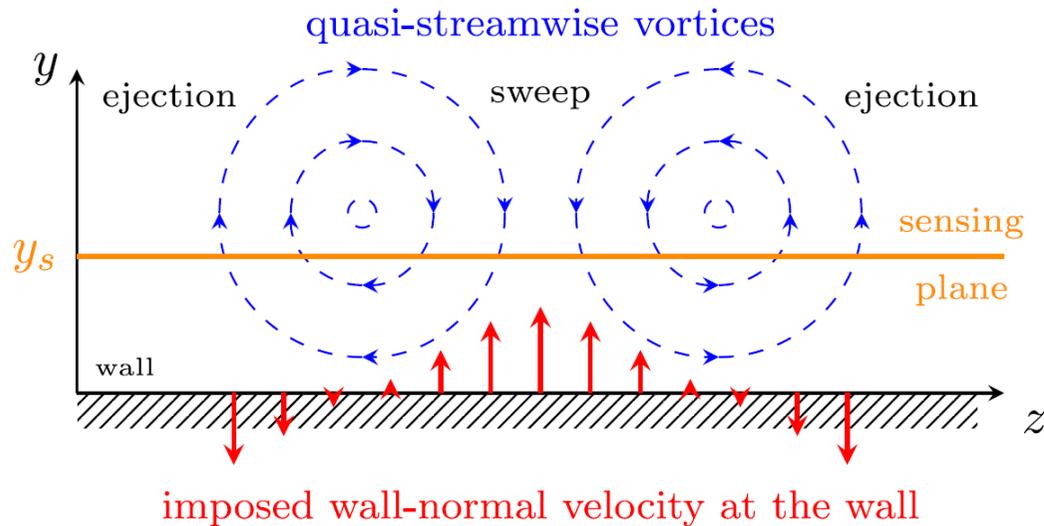
where the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ are outputs of the encoder, and \mathbf{I} is the identity matrix. We sample from $q_\phi(\mathbf{r}|\mathbf{x})$ using $\mathbf{r} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is an auxiliary normally-distributed random number.

β -VAE loss function:

$$\mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}) = \mathcal{L}_{\text{rec}} - \frac{\beta}{2} \sum_{i=1}^d (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2) \quad (12)$$

DRL and opposition control in turbulent channel flow: blowing and suction

- **Minimal channel** has $L_x=2.67h$ ($\sim 400^+$), $L_z=0.8h$ ($\sim 140^+$).
- Results consistent across a number of initial conditions.
- **Full channel** has $L_x=4\pi h$ and $L_z=2\pi h$.



Stroh et al. (2015)

