DEVELOPMENT OF A DATA-DRIVEN WALL-MODEL FOR SEPARATED FLOWS

ECCOMAS 2022, Oslo

M. Boxho, M. Rasquin, T. Toulorge, G. Dergham, G. Winckelmans and K. Hillewaert

ULiege, UCLouvain, Cenaero, Safran Tech

Contact: margaux.boxho@cenaero.be

Doc. ref.: 2019030-THESELDP-SAFRAN

PROD-F-015-02

Outline

Outline

PROD-F-015-02

Cenaero

Industrial context

RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

LES reduces the **modeling assumptions but** remains costly at large Re.

Wall-models reduce the computational cost by modeling the near-wall energetic scales.

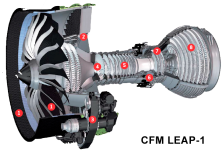
Industrial context

RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

LES reduces the **modeling assumptions but** remains costly at large *Re*.

Wall-models reduce the computational cost by modeling the near-wall energetic scales.

Simulation of turbomachines to compute the **operating points of the engine**.



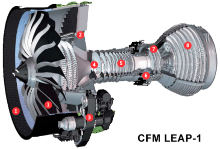
Industrial context

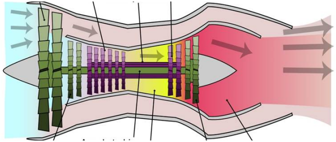
RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

LES reduces the **modeling assumptions but** remains costly at large *Re*.

Wall-models reduce the computational cost by modeling the near-wall energetic scales.

Simulation of turbomachines to compute the **operating points of the engine**.





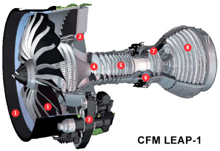
In turbomachines, **transitional** flows are frequently encountered, such as **laminar separation bubbles**.

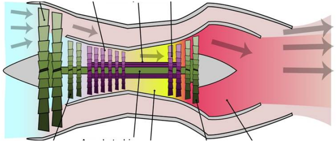
RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

LES reduces the **modeling assumptions but** remains costly at large *Re*.

Wall-models reduce the computational cost by modeling the near-wall energetic scales.

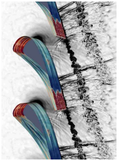
Simulation of turbomachines to compute the **operating points of the engine**.





In turbomachines, **transitional** flows are frequently encountered, such as **laminar separation bubbles**.

Calibration of the wall-model using **Deep Neural Networks** with wrLES on blades.



Industrial context

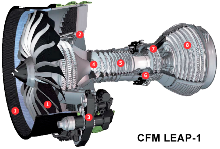
RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

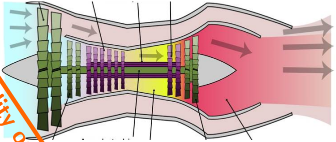
LES reduces the **modeling assumptions but** remains costly at large *Re*.

Wall-models reduce the computational cost by modeling the near-wall energetic scales.

Increasing fidelity on sub-components (wrLES, DNS)

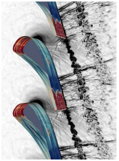
Simulation of turbomachines to compute the **operating points of the engine**.





In turbomachines, **transitional** flows are frequently encountered, such as **laminar separation bubbles**.

Calibration of the wall-model using **Deep Neural Networks** with wrLES on blades.



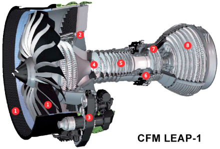
Industrial context

RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

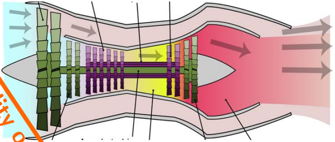
LES reduces the **modeling assumptions but** remains costly at large *Re*.

Wall-models reduce the computational cost by modeling the near-wall energetic scales.

Simulation of turbomachines to compute the **operating points of the engine**.

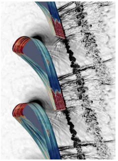


Increasing fidelity on sub-components (wrLES, DNS)



In turbomachines, **transitional** flows are frequently encountered, such as **laminar separation bubbles**.

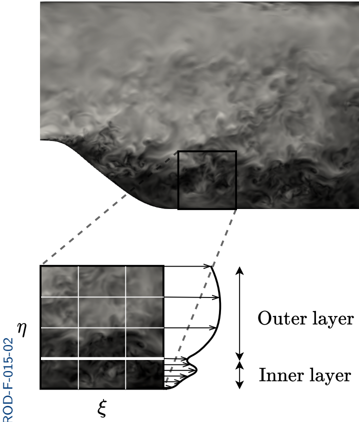
Calibration of the wall-model using **Deep Neural Networks** with wrLES on blades.



Feed lower-fidelity models on larger components (wmLES)

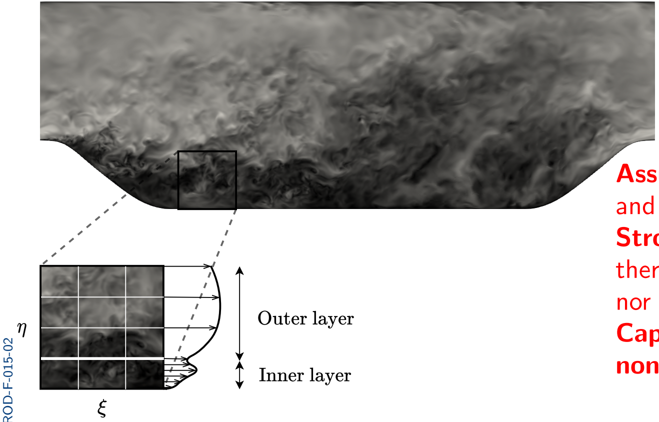
PROD-F-015-02

In the context of Large Eddy Simulations, a wall model (wm) should act as a driver for the wall shear stress boundary condition.



   Cenaero

In the context of Large Eddy Simulations, a wall model (wm) should act as a driver for the wall shear stress boundary condition.



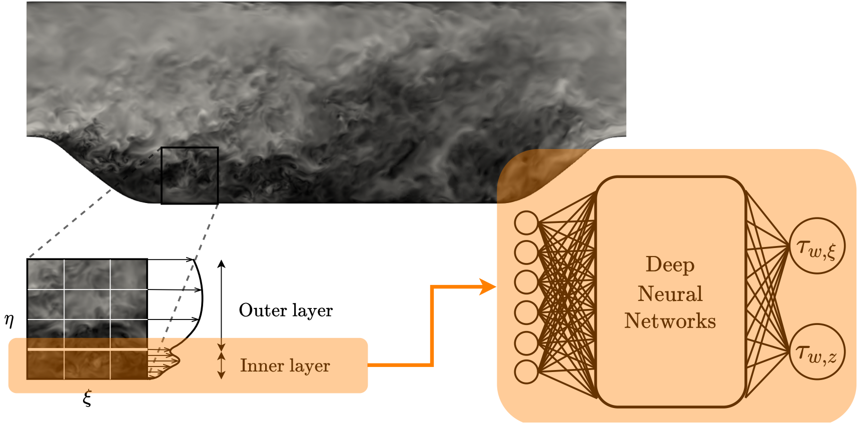
Assumptions: attached, turbulent and at equilibrium (for many wm)

Strong pressure gradient or separation: there is no equilibrium layer nor constant-stress layer

Capability of current non-equilibrium wm: not yet a proven success

   Cenaero

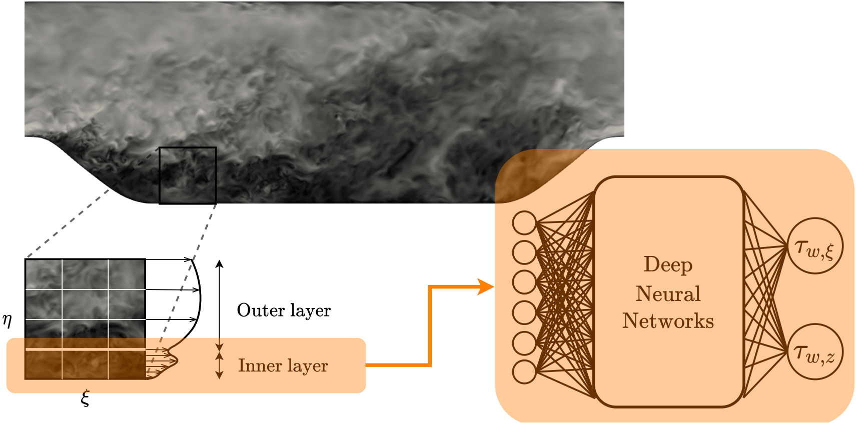
To address this **challenge**, we decide to use the tools provided by deep learning and **deep neural networks**.



Problem definition: finding a complex and dynamic relation between instantaneous flow fields, geometrical parameters and the shear stress

   Cenaero

To address this **challenge**, we decide to use the tools provided by deep learning and **deep neural networks**.



New research subject: need diagnostic tools for pre-processing

Inspired from U. Piomelli. Wall-modeled large-eddy simulations: Present status and prospects. *Springer Netherlands*, 17, 2010.

PROD-F-015-02

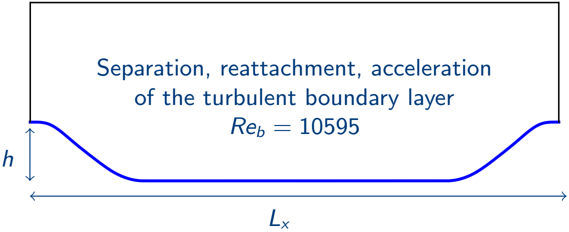
Outline

PROD-F-015-02

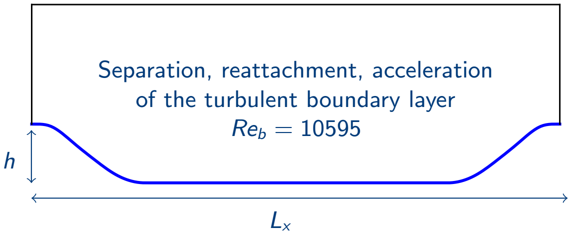
Cenaero

Test cases

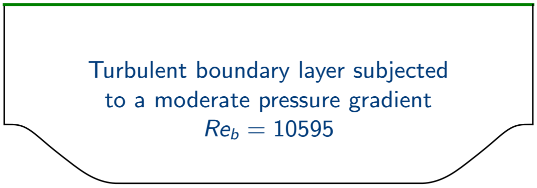
TC1 : Periodic Hill - Lower wall



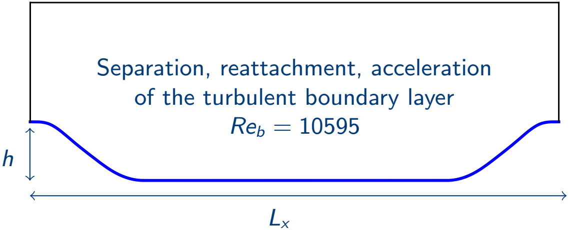
TC1 : Periodic Hill - Lower wall



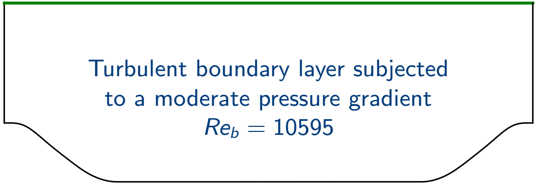
TC2 : Periodic Hill - Upper wall



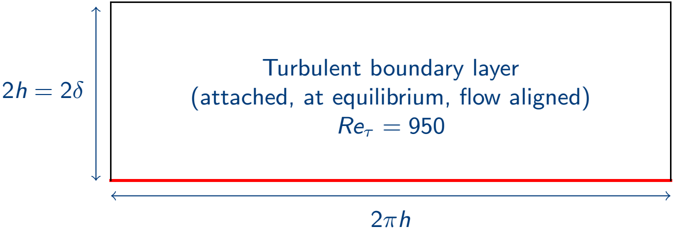
TC1 : Periodic Hill - Lower wall



TC2 : Periodic Hill - Upper wall



TC3 : Channel wall



- **Bi-periodic** flow evolving between two walls featuring a streamwise **constriction**[1]
- **Controlled** pressure gradient to match the bulk Reynolds number ($Re_b = 10595$) combined with a **low bulk Mach** number $M_b = 0.1$

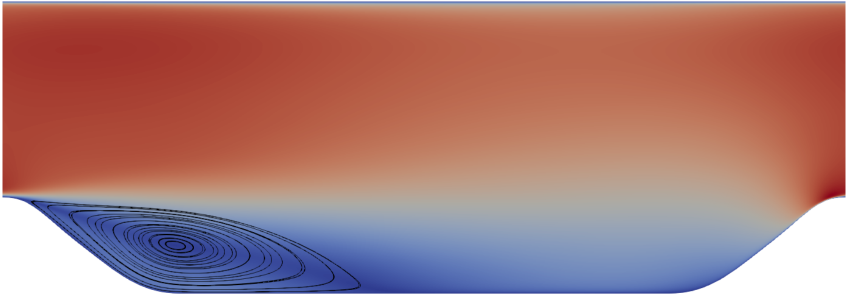


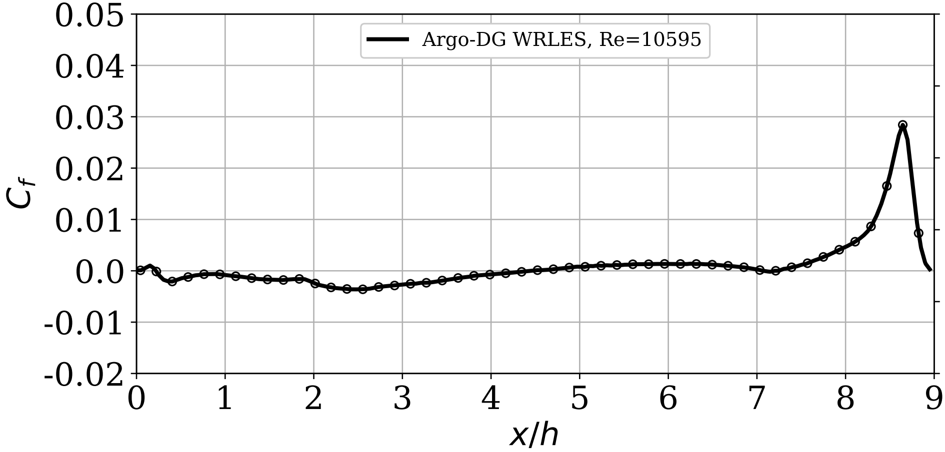
Figure: Streamlines focusing on the separation bubble

[1] https://www.kbwiki.ercoftac.org/w/index.php/Abstr:2D_Periodic_Hill_Flow

PROD-F-015-02

$$C_f = u_\tau^2/(0.5\rho u_b^2)$$

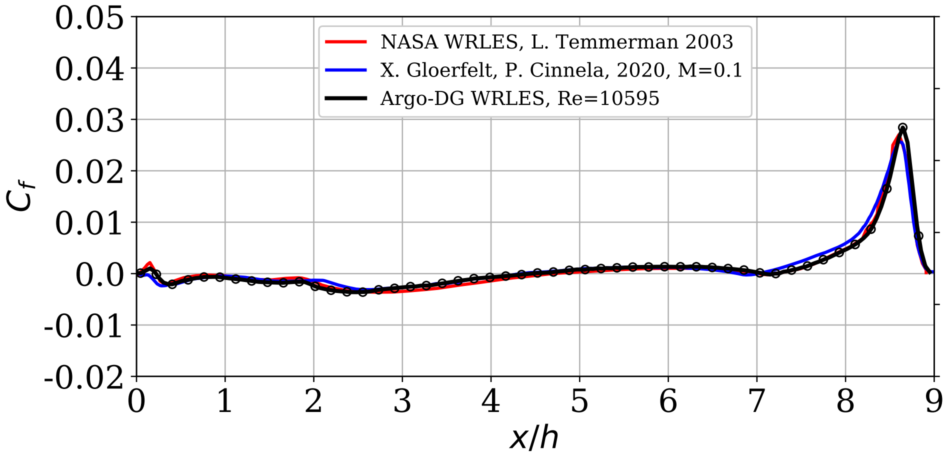
where $u_\tau = \sqrt{\tau_w/\rho_w}$ and $\gamma = \dfrac{\Delta t_{(C_f<0)}}{\Delta t_{tot}}$



Cenaero

$$C_f = u_\tau^2/(0.5\rho u_b^2)$$

where $u_\tau = \sqrt{\tau_w/\rho_w}$ and $\gamma = \dfrac{\Delta t_{(C_f<0)}}{\Delta t_{tot}}$

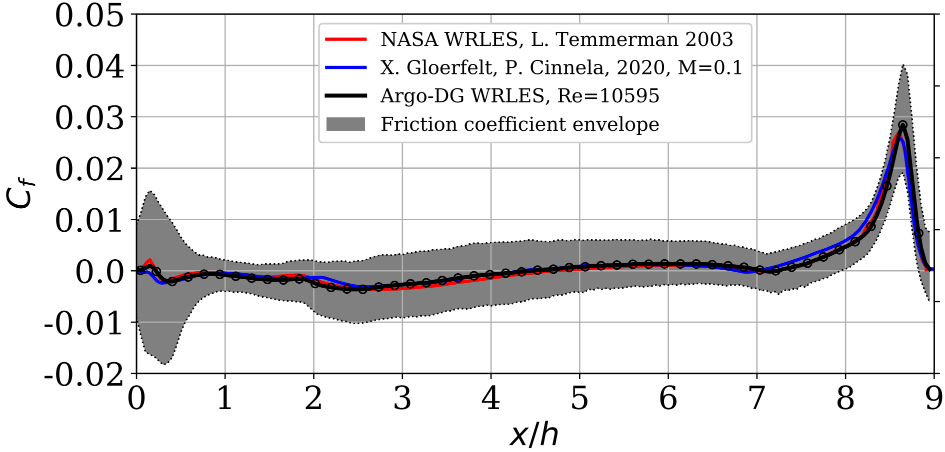


Cenaero

Two-dimensional Periodic Hill

$$C_f = u_\tau^2/(0.5\rho u_b^2)$$

where $u_\tau = \sqrt{\tau_w/\rho_w}$ and $\gamma = \dfrac{\Delta t_{(C_f<0)}}{\Delta t_{tot}}$



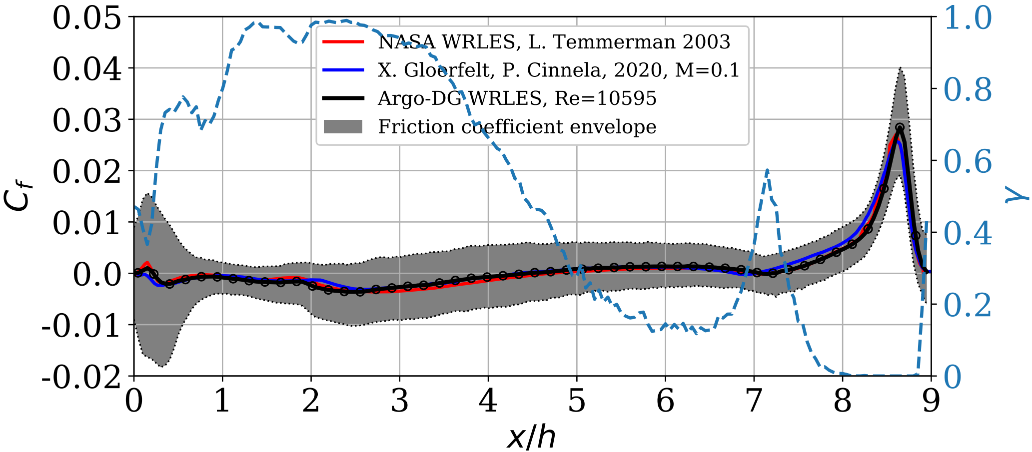
PROD-F-015-02

Cenaero

Two-dimensional Periodic Hill

$$C_f = u_\tau^2/(0.5\rho u_b^2)$$

where $u_\tau = \sqrt{\tau_w/\rho_w}$ and $\gamma = \dfrac{\Delta t_{(C_f<0)}}{\Delta t_{tot}}$



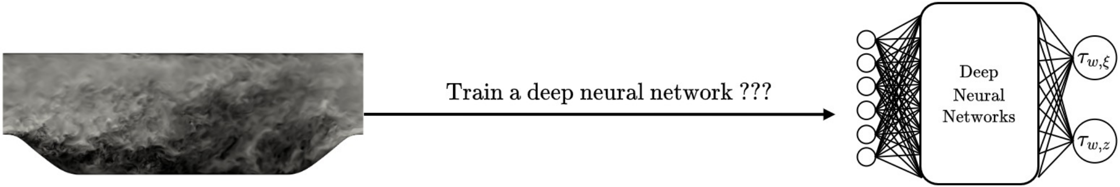
PROD-F-015-02

Cenaero

Outline

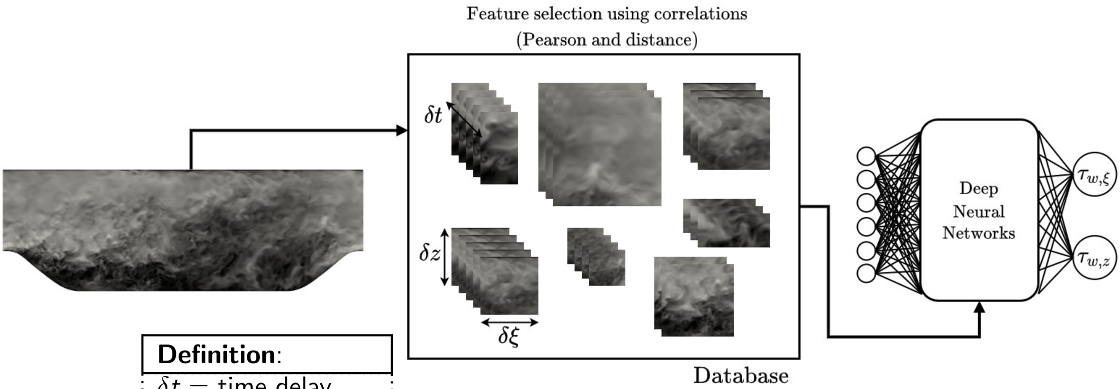
PROD-F-015-02

Cenaero



Reasons:
→ need to select input and output labels that are strongly related
→ feature selection improves model performance and reduces the computational cost of modeling

Space-time correlations



Definition:

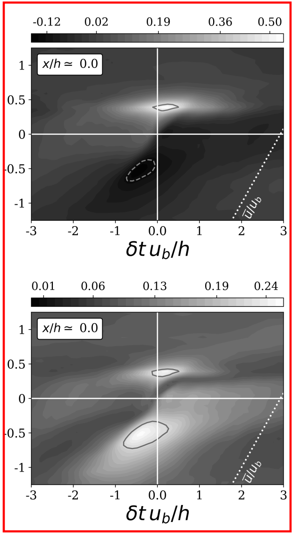
δt = time delay
$\delta\xi, \delta z$ = space shift
are two **quantities**
to be deduced
from correlations

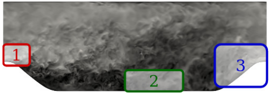
Using$\rightarrow$ Pearson and distance correlations, where distance measures both **linear** and **non-linear** relationships

Space-time correlation in the streamwise direction between $\tau_{w,\xi}$ and u_ξ

Pearson's corr.

Distance corr.





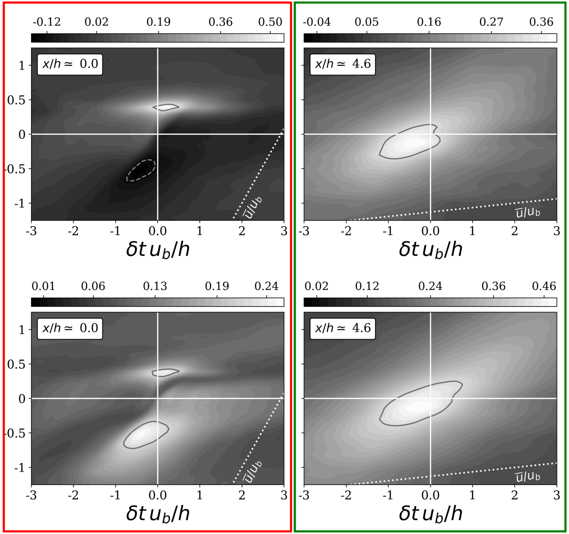
PROD-F-015-02

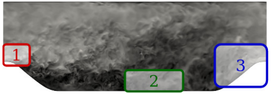
Cenaero

Space-time correlation in the streamwise direction between $\tau_{w,\xi}$ and u_{ξ}

Pearson's corr.

Distance corr.

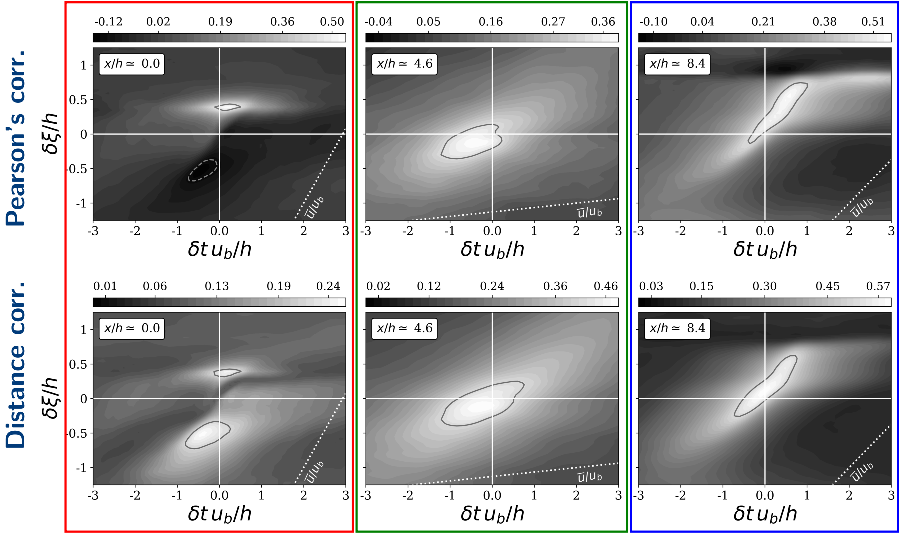


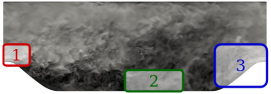


PROD-F-015-02

Cenaero

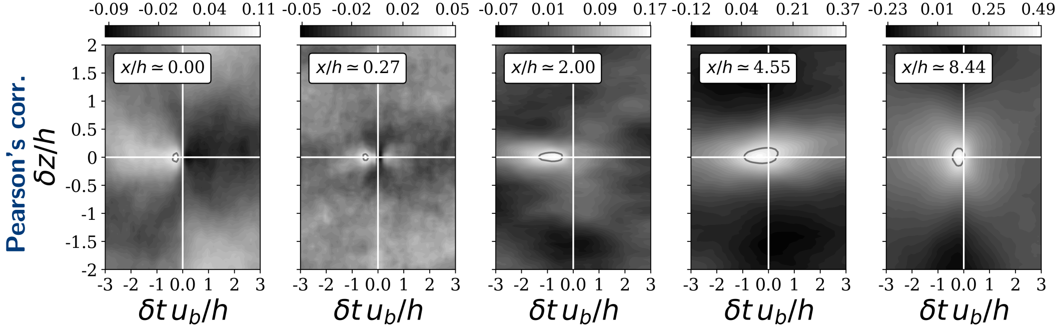
Space-time correlation in the streamwise direction between $\tau_{w,\xi}$ and u_ξ





Instantaneous and local correlation is **negligible** for separated flow :
shifted
in both $\delta\xi > 0$ and $\delta\xi < 0$

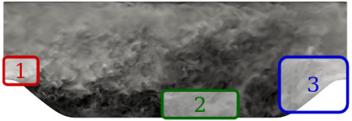
Space-time correlation in the spanwise direction between $\tau_{w,\xi}$ and u_ξ



Symmetric correlation as expected due to:

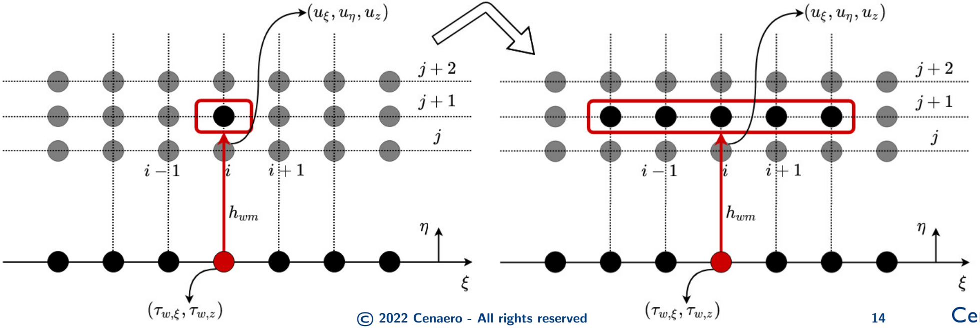
- the homogeneity of z and,
- the absence of convection.

Cenaero



Main Guidelines

- purely **local** and **instantaneous** information is sufficient for **attached** flows
- upstream information can be used **if a convection delay** is considered
- for **separated** flow, information has to be sought **up- and downstream**
- need to **enlarge** the domain of dependence of the wall model in both space ($\delta\xi$) and time (δt)

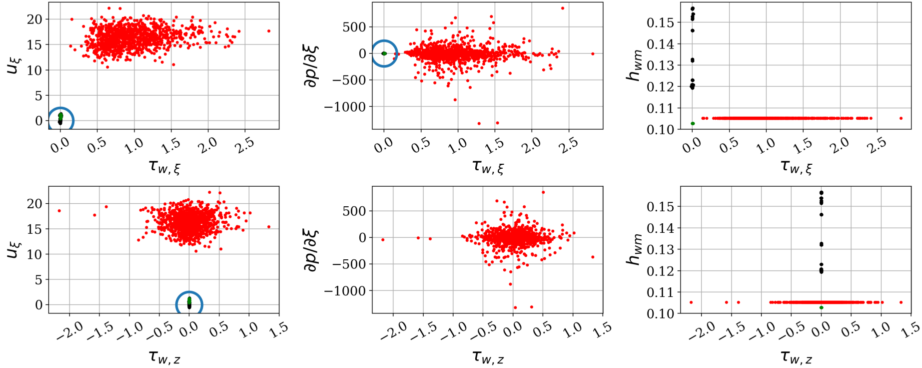


Outline

Cenaero

Which inputs are selected ?

Let us examine the distributions of three inputs (**velocity**, **pressure gradient** and **wall-normal distance**) w.r.t the two components of the wall shear stress for the three configurations (lower wall phill, upper wall phill and channel) :



→ **Not a good behavior for generalization!**

Inputs:

Field	*Description*	*Pre-scaling*
Velocity	$\boldsymbol{u}$	$\boldsymbol{u}^\star$
Pressure Gradients	∇p	$\nabla p^\star$
Length scale	h_{wm}	$\ln\left(h_{wm}/y_{\nu,p}\right)$

for a total of 7 different components.

where $y_{\nu,p}$ is a near-wall scaling compatible with separation[2],

$$y_{\nu,p} = \frac{\nu}{u_{\nu,p}} \quad \text{with,} \; u_{\nu,p} = \sqrt{u_\nu^2 + u_p^2}, \;\; u_\nu = \sqrt{\frac{\nu u_\parallel}{h_{wm}}}, \;\; u_p = \left|\frac{\nu}{\rho}\frac{\partial p}{\partial x}\right|^{1/3}.$$

The velocity is scaled with $u_{\tau,p}$ and the pressure gradient is scaled as the Clauser parameter :

$$\nabla p^\star = \frac{h_{wm}}{\rho u_{\nu,p}^2} \nabla p \, .$$

PROD-F-015-02

[2] C. Duprat et al. A wall-layer model for large-eddy simulations of turbulent flows with/out pressure gradient. Physics of Fluids, 23203(1):015101, 2011.

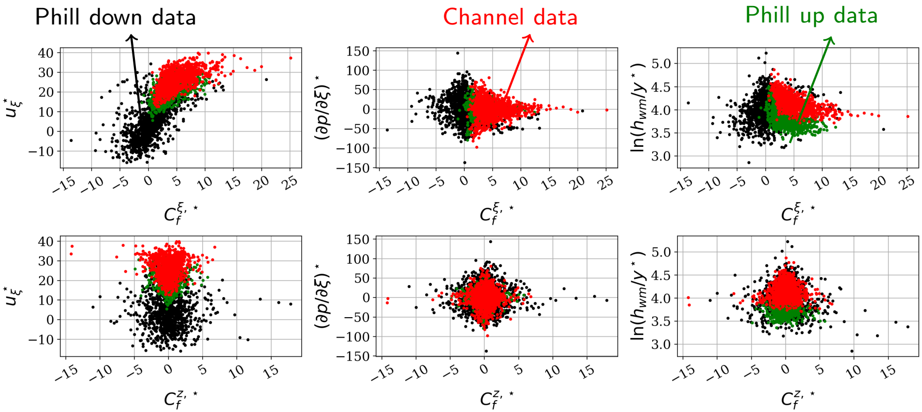
Outputs:

Field	*Description*	*Pre-scaling*
Wall shear stress	$\tau_{w,\xi}$, $\tau_{w,z}$	$\boldsymbol{\tau}_w^\star = \dfrac{\boldsymbol{\tau}_w}{\frac{1}{2}\rho\langle u_{\nu,p}^2\rangle_{\xi,z}}$

which are the components of the wall shear stress in the local wall-aligned reference frame. Note that a **spatial averaging** of the velocity scale is used to avoid spurious oscillations in the predictions.

Which inputs/outputs are selected ?

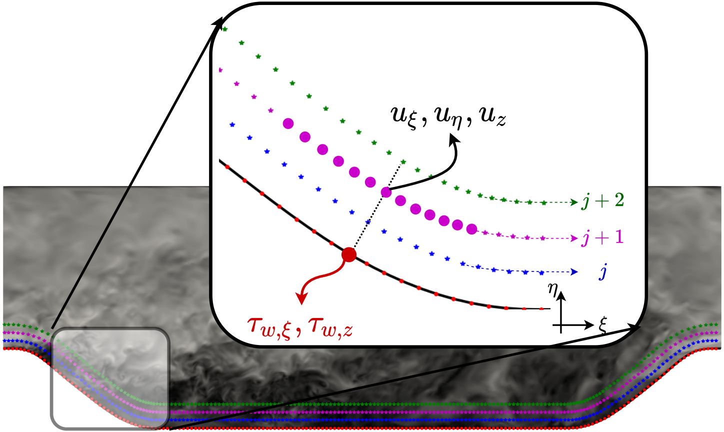
Let us examine the distribution of these normalized fields w.r.t. the wall shear stress :



PROD-F-015-02

→ **Better for generalization!**

Where are the data extracted ?



10 pts. **donwstrean**

$+$

10 pts. **upstream**

$+$

the **current** pt.

$=$

domain of high correlations (at one height)

$=$

union of the space-time correlations.

That's all ?

Two new inputs have been added to improve model performances:

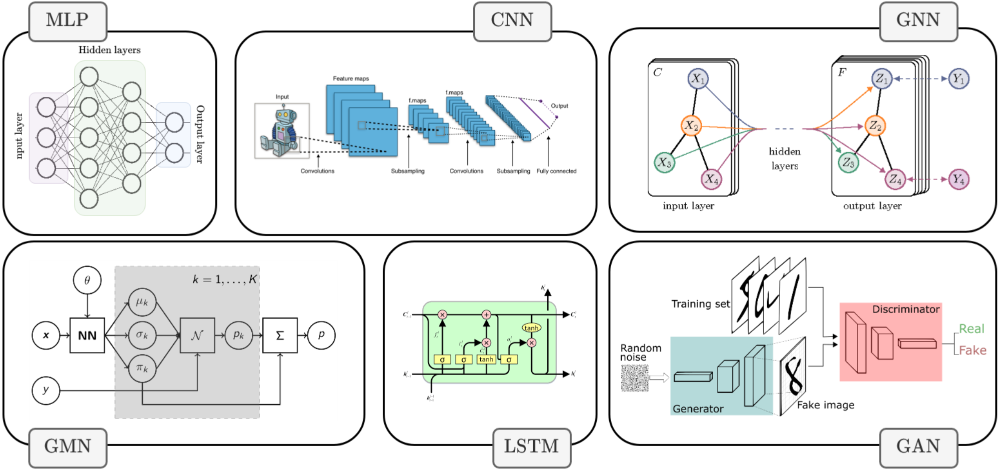
- the **curvature** K defined as,

$$K = \frac{|f''(x)|}{\{1 + (f'(x))^2\}^{(3/2)}},$$

  where $f(x)$ is the function that describes the surface.

- the absolute value of the **relative positions** (information of spacing is hence added), this input has an impact on the self-attention layer.

Neural Network Architecture[3] - Jungle

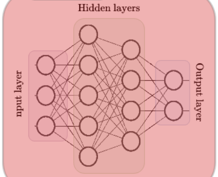


PROD-F-015-02

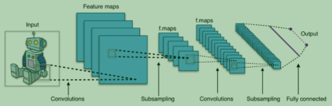
[3] MLP stands for Multi-Layer Perceptron, CNN stands for Convolutional Neural Network, RNN stands for Recurrent Neural Network, GNN stands for Graph Neural Network, LSTM stands for Long-Short Term Memory, GMN stands for Gaussian-Mixture Network, GAN stands for Generative Adversarial Network.

Neural Network Architecture[3] - Jungle

MLP

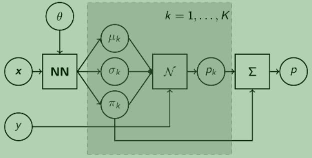


CNN



GNN

unused



GMN

unused because no time history is considered

LSTM

unused

GAN

[3] MLP stands for Multi-Layer Perceptron, CNN stands for Convolutional Neural Network, RNN stands for Recurrent Neural Network, GNN stands for Graph Neural Network, LSTM stands for Long-Short Term Memory, GMN stands for Gaussian-Mixture Network, GAN stands for Generative Adversarial Network.

PROD-F-015-02

Neural Network Architecture - Convolutional Neural Network

PROD-F-015-02

Cenaero

Neural Network Architecture - Convolutional Neural Network

Learnable convolutional kernel

$$\begin{bmatrix} 0 & 0.5 & 0.8 & 0.2 & 0.3 \\ 0.1 & 0 & 0.4 & 0.6 & 0.6 \\ 0.9 & 1 & 0 & 0 & 0.8 \\ 0.9 & 1 & 0.7 & 0.8 & 0.1 \end{bmatrix} = \begin{bmatrix} 2 & & \\ & & \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

PROD-F-015-02

Neural Network Architecture - Convolutional Neural Network

Learnable convolutional kernel

0	0.5	0.8	0.2	0.3
0.1	0	0.4	0.6	0.6
0.9	1	0	0	0.8
0.9	1	0.7	0.8	0.1

0	1	0
1	1	1
0	1	0

=

2	1.8	

Learnable convolutional kernel

0	0.5	0.8	0.2	0.3
0.1	0	0.4	0.6	0.6
0.9	1	0	0	0.8
0.9	1	0.7	0.8	0.1

0	1	0
1	1	1
0	1	0

=

2	1.8	1.8

PROD-F-015-02

Neural Network Architecture - Convolutional Neural Network

Learnable convolutional kernel

$$\begin{array}{|c|c|c|c|c|}\hline 0 & 0.5 & 0.8 & 0.2 & 0.3 \\ \hline 0.1 & 0 & 0.4 & 0.6 & 0.6 \\ \hline 0.9 & 1 & 0 & 0 & 0.8 \\ \hline 0.9 & 1 & 0.7 & 0.8 & 0.1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 2 & 1.8 & 1.8 \\ \hline 2.9 & & \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Neural Network Architecture - Convolutional Neural Network

0	0.5	0.8	0.2	0.3
0.1	0	0.4	0.6	0.6
0.9	1	0	0	0.8
0.9	1	0.7	0.8	0.1

0	1	0
1	1	1
0	1	0

=

2	1.8	1.8
2.9	2.1	

Learnable convolutional kernel

Neural Network Architecture - Convolutional Neural Network

Learnable convolutional kernel

0	0.5	0.8	0.2	0.3
0.1	0	0.4	0.6	0.6
0.9	1	0	0	0.8
0.9	1	0.7	0.8	0.1

0	1	0
1	1	1
0	1	0

=

2	1.8	1.8
2.9	2.1	2.2

PROD-F-015-02

Cenaero

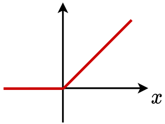
Neural Network Architecture - Convolutional Neural Network

Learnable convolutional kernel

$$\begin{array}{|c|c|c|c|c|}\hline 0 & 0.5 & 0.8 & 0.2 & 0.3 \\ \hline 0.1 & 0 & 0.4 & 0.6 & 0.6 \\ \hline 0.9 & 1 & 0 & 0 & 0.8 \\ \hline 0.9 & 1 & 0.7 & 0.8 & 0.1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 2 & 1.8 & 1.8 \\ \hline 2.9 & 2.1 & 2.2 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Activation function σ

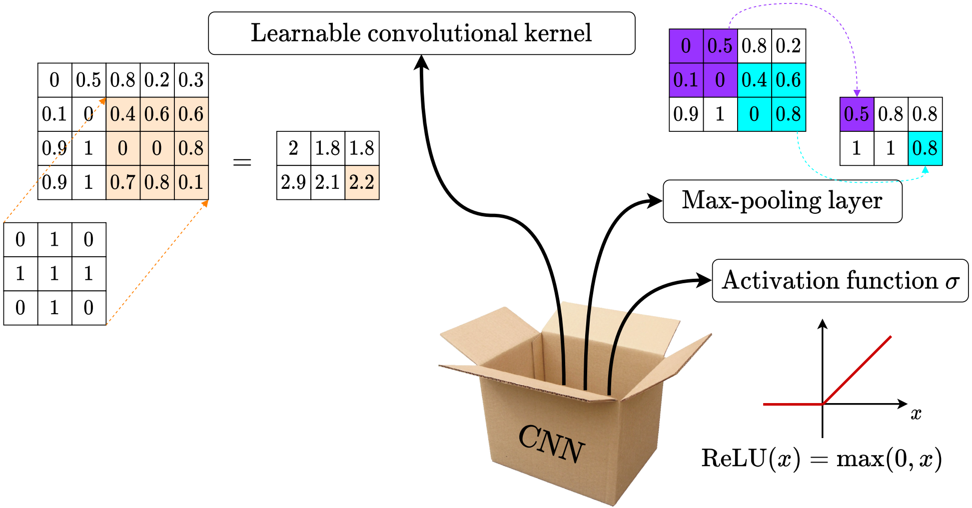


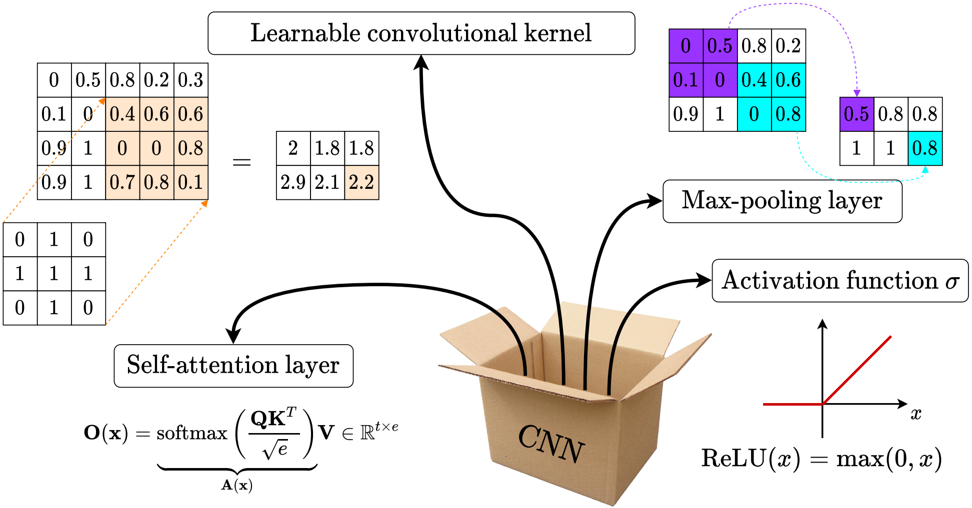
$$\mathrm{ReLU}(x) = \max(0, x)$$

PROD-F-015-02

Cenaero

Neural Network Architecture - Convolutional Neural Network





$$\mathbf{O}(\mathbf{x}) = \underbrace{\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{e}}\right)}_{\mathbf{A}(\mathbf{x})}\mathbf{V} \in \mathbb{R}^{t \times e}$$

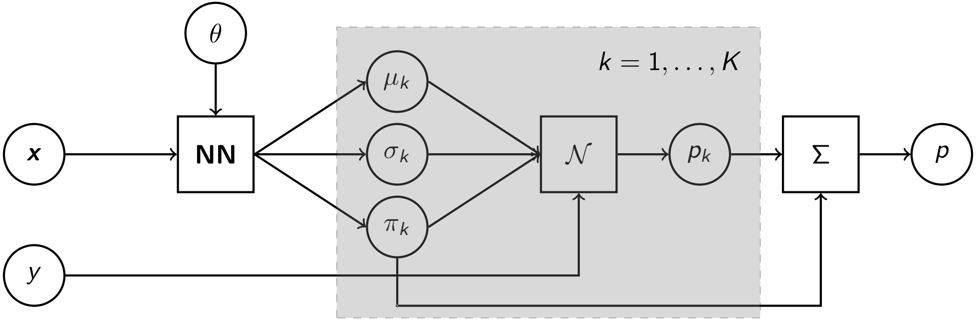
$$\mathrm{ReLU}(x) = \max(0, x)$$

   Cenaero

Neural Network Architecture - Gaussian Mixture heads

Gaussian Mixture Neural Networks (GMN) aim to predict the probability distribution $p(\tau_w)$ of the wall shear stress component as a linear combination of Gaussian distribution :

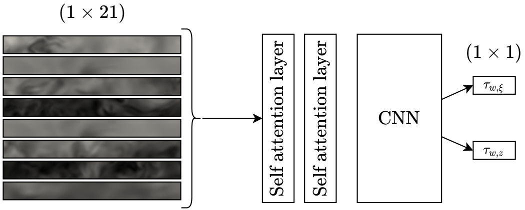
$$p(\tau_w) = \sum_{k=1}^{K} \pi_k p_k = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \sigma_k)$$



PROD-F-015-02

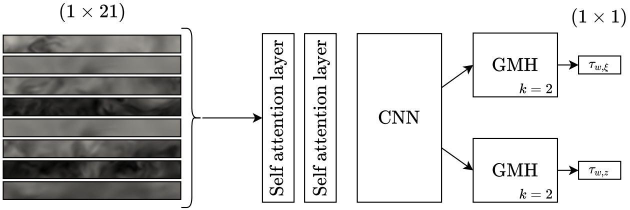
Three neural networks are tested :

- **CNN-1d-SAL** : a one-dimensional convolutional neural network combined with two consecutive self-attention layer



Three neural networks are tested :

- **CNN-1d-SAL** : a one-dimensional convolutional neural network combined with two consecutive self-attention layer
- **GMN-CNN-1d-SAL** : the same architecture as **CNN-1d-SAL** but the last layer is removed for a Gaussian Mixture Head

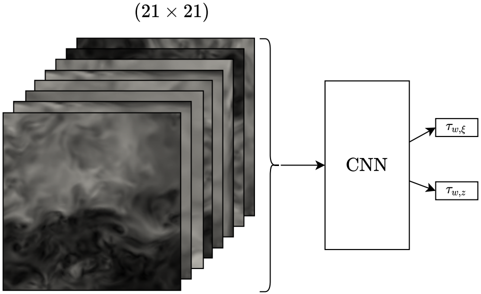


Cenaero

Neural Network Architecture

Three neural networks are tested :

- **CNN-1d-SAL** : a one-dimensional convolutional neural network combined with two consecutive self-attention layer
- **GMN-CNN-1d-SAL** : the same architecture as **CNN-1d-SAL** but the last layer is removed for a Gaussian Mixture Head
- **CNN-2d** : a two-dimensional convolutional neural network that takes as input



PROD-F-015-02

Outline

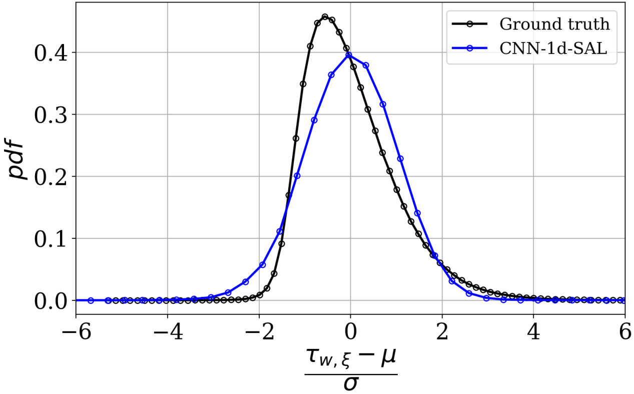
PROD-F-015-02

Cenaero

Results - On the channel at $Re_\tau = 950$ - A priori validation

$n_\xi \times n_\eta \times n_z \times n_t = 192 \times 6 \times 192 \times 123$
Training height : $y^+ = 100$
Training with **181,370 samples** with 10% for test
Training with early stop after 400 epochs
Mean Square Error (MSE) loss

Tips

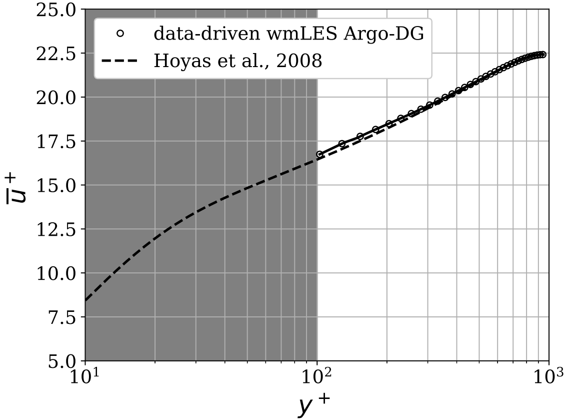


Main statistics about the model prediction distribution and the ground truth (or target) distribution :

	μ	σ	$\mathcal{S}$	$\mathcal{E}_{rel}\%$
Model	1.00798	0.126	-0.051	**0.051**
Target	1.00850	0.423	1.016	-

Results, converged over $51.5t_c$ (where $t_c = L_x/u_b$), are compared with Hoyas *et al.*[4]

Mesh : $n_x \times (n_y/2) \times n_x = 26 \times 13 \times 13$
combined with an order 3 to get
$(\Delta x^+, \Delta y^+, \Delta z^+) \simeq (76, 25, 76)$
for a total of 562,432 dof
No stretching is applied near the wall
$\left(\frac{\text{dof}_{\text{wrLES}}}{\text{dof}_{\text{wmLES}}}\right) \simeq 30$



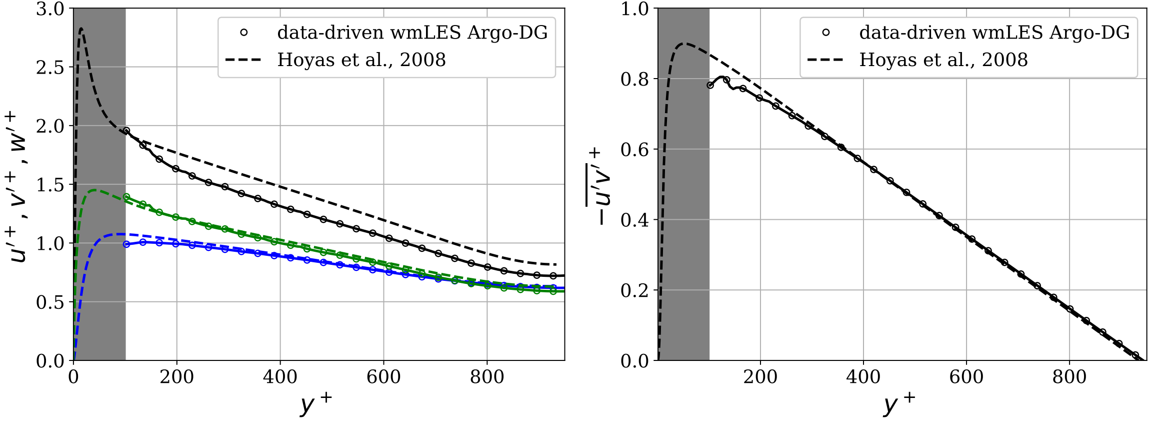
→ small over-estimation of 2% near the wall.

[4] S. Hoyas and J. Jimenez. Reynolds number effects on the Reynolds-stress budgets in turbulent channels. Physics of Fluids, 20:101511, 2008. doi 10.1063/1.3005862.

PROD-F-015-02

Cenaero

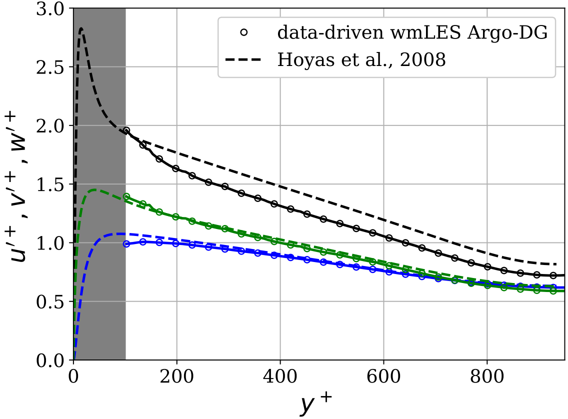
Results, converged over $51.5t_c$ (where $t_c = L_x/u_b$), are compared with Hoyas *et al.*[4]

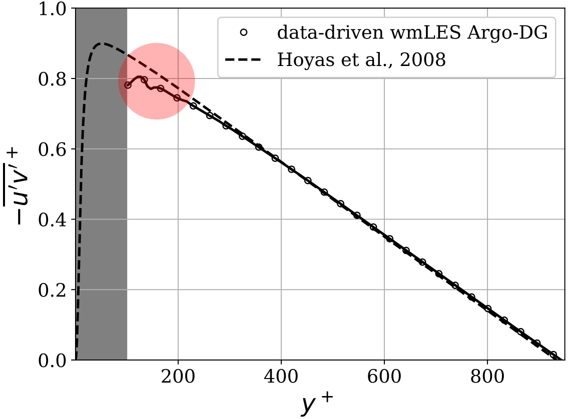


[4] S. Hoyas and J. Jimenez. Reynolds number effects on the Reynolds-stress budgets in turbulent channels. Physics of Fluids, 20:101511, 2008. doi 10.1063/1.3005862.

Results - On the channel - A posteriori validation

Results, converged over $51.5t_c$ (where $t_c = L_x/u_b$), are compared with Hoyas *et al.*[4]





[4] S. Hoyas and J. Jimenez. Reynolds number effects on the Reynolds-stress budgets in turbulent channels. Physics of Fluids, 20:101511, 2008. doi 10.1063/1.3005862.

PROD-F-015-02

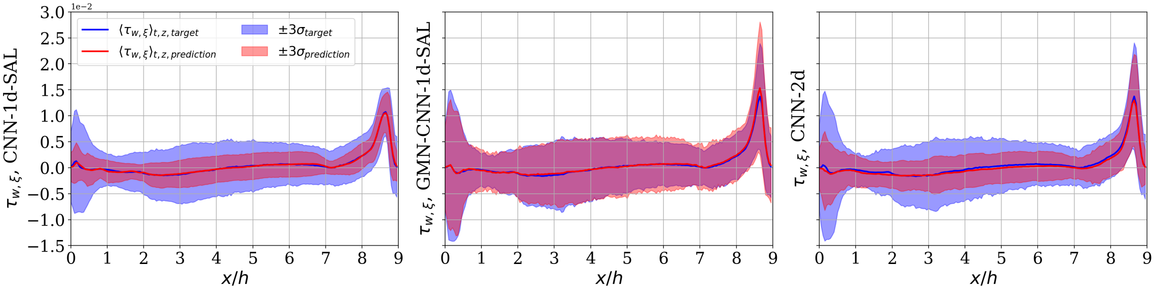
Cenaero

Results - Cross-training - Lower wall

$n_\xi \times n_\eta \times n_z \times n_t = 180 \times 40 \times 100 \times 200$
Training height : $y/h = 0.1$
10% of the database is used for test
Training with early stop

Cross-training : training on the **channel** and the **lower** wall of the periodic hill while extrapolating on the upper wall of the periodic hill.

Averaged **streamwise** wall shear stress $\tau_{w,\xi}$,



PROD-F-015-02

Cenaero

Results - Cross-training - Lower wall

Instantaneous contours of the wall shear stress $\tau_{w,\xi}$:

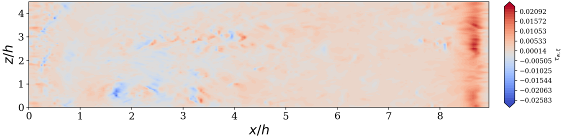


Figure: Reference

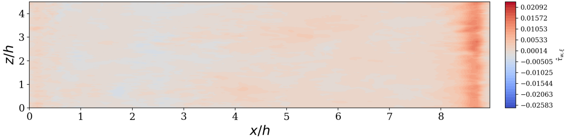


Figure: CNN-1d-SAL

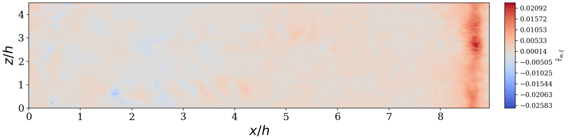


Figure: CNN-2d

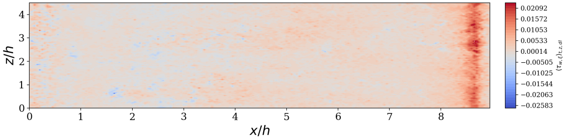
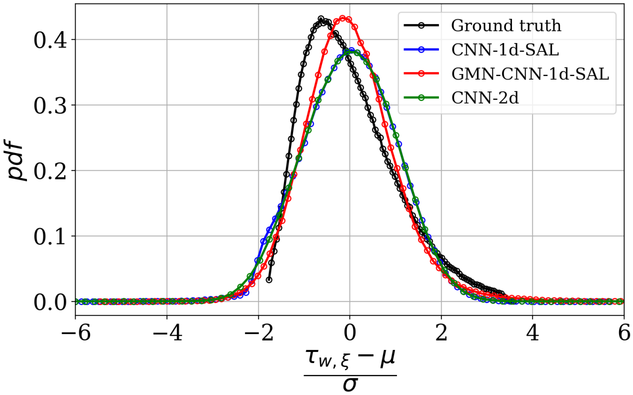


Figure: GMN-CNN-1d-SAL

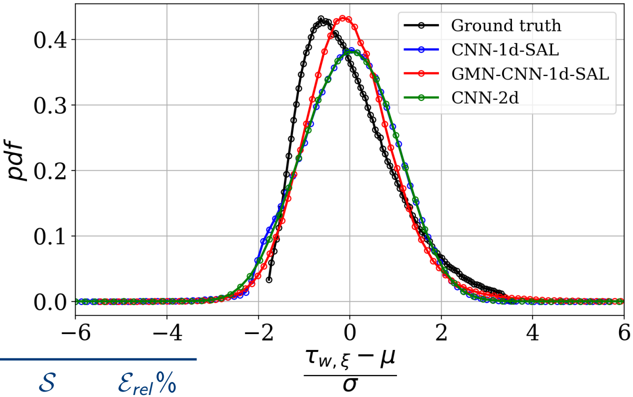
PROD-F-015-02

Results - Cross-training - Channel

Let us look at results on the **channel wall** based on the distribution of the ground truth compared to the predicted ones.



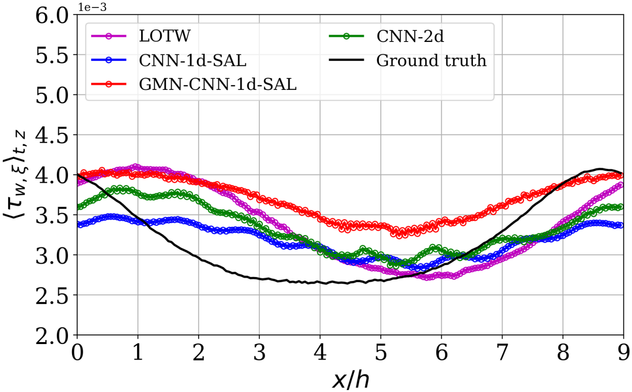
Let us look at results on the **channel wall** based on the distribution of the ground truth compared to the predicted ones.



Model	μ	σ	$\mathcal{S}$	$\mathcal{E}_{rel}\%$
CNN-1d-SAL	0.995	0.089	-0.022	1.367
GMN-CNN-1d-SAL	1.024	**0.427**	**0.498**	1.535
CNN-2d	**0.998**	0.122	-0.039	**1.037**
True distribution :	1.008	0.423	1.016	-

Results - Cross-training - Upper wall

Model	**LOTW**	**CNN-1d-SAL**	**GMN-CNN-1d-SAL**	**CNN-2d**
$\mathcal{E}_{rel}\%$	14.545	11.063	18.972	12.682



PROD-F-015-02

- Training on a **channel** gives great results in the *a priori* and *a posteriori* validation;
- Training on a **channel** wall and on the **lower** wall of the periodic hill gives encouraging results in the *a priori* validation while extrapolating on the upper wall of the periodic hill fails;
- Training on the **upper** and **lower** wall of the periodic hill predicts well on the same configurations and fails at extrapolating on the channel, except if the input "*curvature*" is modified !
  - → the *curvature* may act as a mask in the neural network
  - → the physics of the channel may be better represented by the accelerated flow over the hill than the flow under a moderate pressure gradient over the upper wall

Conclusion

- Developing **wall model** to tackle separation using **deep neural network**
- Generating **databases** from channel and periodic hill flows
- Analyzing **space-time correlations** with $\boldsymbol{\tau}_w$ (i.e., feature selection)
- Training **CNN and GMN** for the prediction of $\boldsymbol{\tau}_w$
- *A posteriori* validation on a channel at $Re_\tau = 950$ gives **encouraging** results

Conclusion

- Developing **wall model** to tackle separation using **deep neural network**
- Generating **databases** from channel and periodic hill flows
- Analyzing **space-time correlations** with τ_w (i.e., feature selection)
- Training **CNN and GMN** for the prediction of τ_w
- *A posteriori* validation on a channel at $Re_\tau = 950$ gives **encouraging** results

- *A posteriori* validation on the channel with a network trained on **multiple configurations**
- *A posteriori* validation on a channel flow at a **higher** Re_τ
- *A posteriori* validation on the **periodic hill** (same Re_b and a higher one)
- Tackle an industrial geometry, a blade (e.g. T106C) featuring a separation

Conclusion

- Developing **wall model** to tackle separation using **deep neural network**
- Generating **databases** from channel and periodic hill flows
- Analyzing **space-time correlations** with τ_w (i.e., feature selection)
- Training **CNN and GMN** for the prediction of τ_w
- *A posteriori* validation on a channel at $Re_\tau = 950$ gives **encouraging** results

- *A posteriori* validation on the channel with a network trained on **multiple configurations**
- *A posteriori* validation on a channel flow at a **higher** Re_τ
- *A posteriori* validation on the **periodic hill** (same Re_b and a higher one)
- Tackle an industrial geometry, a blade (e.g. T106C) featuring a separation

"Data scientists need to think about their models in post-production because only once the model is in production is when it starts generating value." by Alessya Viscnjic, CEO of WhyLabs.

PROD-F-015-02

Acknowledgment

The present research benefited from computational resources made available on the **Tier-1 supercomputer** of the Fédération Wallonie-Bruxelles, infrastructure funded by the Walloon Region under the grant agreement n°1117545. We would also like to gratefully acknowledge **SafranTech**'s funding of Mrs. Boxho's thesis.

PROD-F-015-02