A Point-Cloud Deep Learning Framework for Prediction of Fluid Flow Fields on Irregular Geometries

Ali Kashefi
Davis Rempe
Prof. Leonidas Guibas

Manuscript

A Point-Cloud Deep Learning Framework for Prediction of Fluid Flow Fields on Irregular Geometries

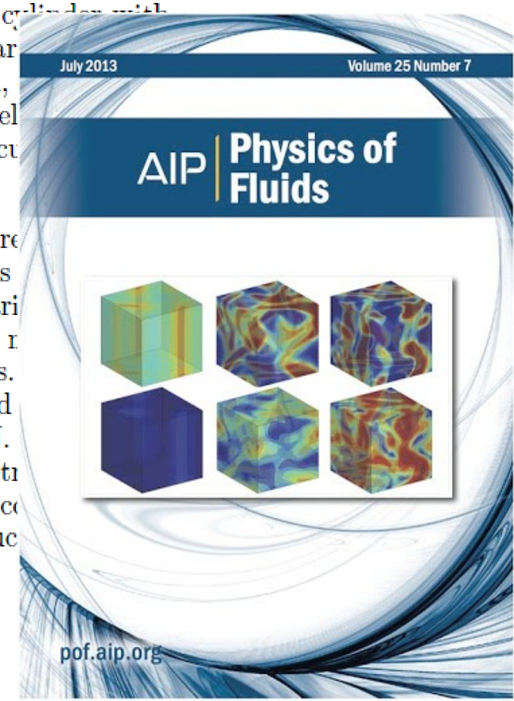
Ali Kashefi,[1,a)] Davis Rempe,[1,b)] and Leonidas J. Guibas[1,c)]
Stanford University, Stanford, CA 94305, USA

We present a novel deep learning framework for flow field predictions in irregular domains when the solution is a function of the geometry of either the domain or objects inside the domain. Grid vertices in a computational fluid dynamics (CFD) domain are viewed as point clouds and used as inputs to a neural network based on the PointNet architecture, which learns an end-to-end mapping between spatial positions and CFD quantities. Using our approach, (i) the network inherits desirable features of unstructured meshes (e.g., fine and coarse point spacing near the object surface and in the far field, respectively), which minimizes network training cost; (ii) object geometry is accurately represented through vertices located on object boundaries, which maintains boundary smoothness and allows the network to detect small changes between geometries; and (iii) no data interpolation is utilized for creating training data; thus accuracy of the CFD data is preserved. None of these features are achievable by extant methods based on projecting scattered CFD data into Cartesian grids and then using regular convolutional neural networks. Incompressible laminar steady flow past a c
various shapes for its cross section is considered. The mass and momentum of predicted fields ar
For the first time, our network generalizes the predictions to multiple objects as well as an airfoil,
only single objects and no airfoils are observed during training. The network predicts the flow fiel
of times faster than our conventional CFD solver, while maintaining excellent to reasonable accu

I. INTRODUCTION AND MOTIVATION

One of the main contributions of machine learning techniques to Computational Fluid Dynamics (CFD) simulations is reducing the computational costs. Even with the presence of high performance computing tools (see e.g., Refs. 1–5) and efficient numerical schemes (see e.g., Refs. 6–11) to accelerate CFD simulations, investigation of design parameters for device optimization remains computationally expensive mainly because a huge

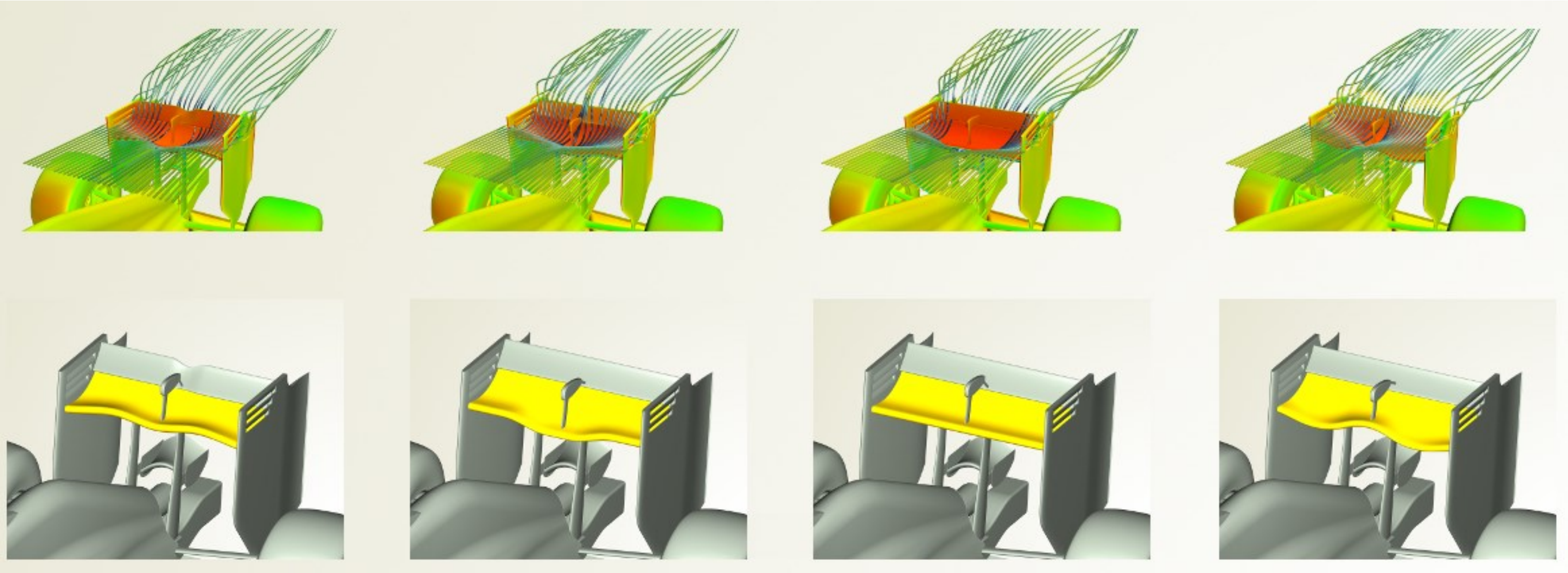
a network, and thus, an effective data repre
crucial. The connection of neural networks
sian grids is straightforward. For this scenari
and three-dimensional CNNs is a popular r
among the CFD community (see e.g., Refs.
this method, each vertex of a Cartesian grid
to a pixel of an image processed by a CNN.
real-world applications with complex geometr
ing unstructured grids is unavoidable. In co
Cartesian grids, the connection of unstruc

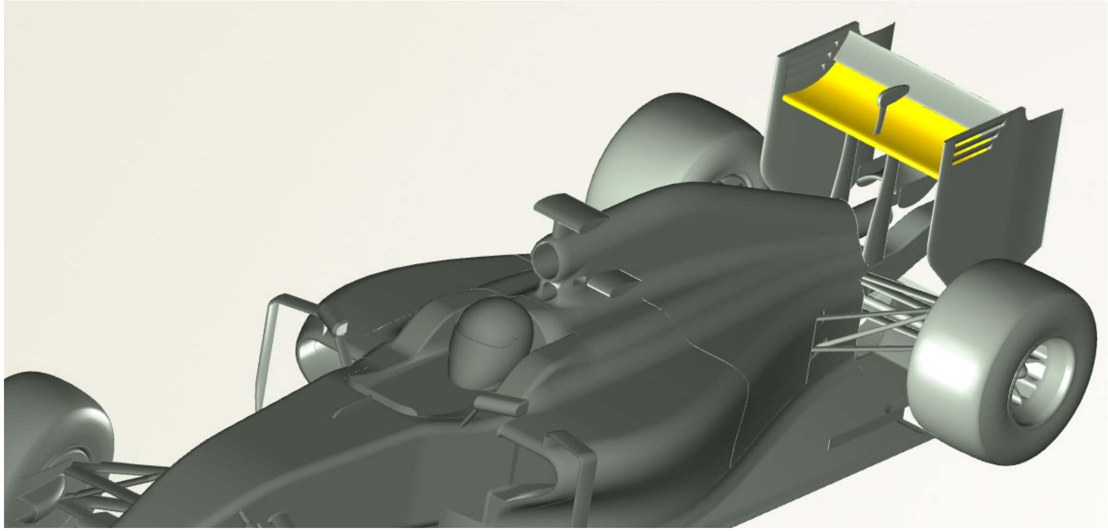


1 Introduction and motivation

1 Introduction and motivation

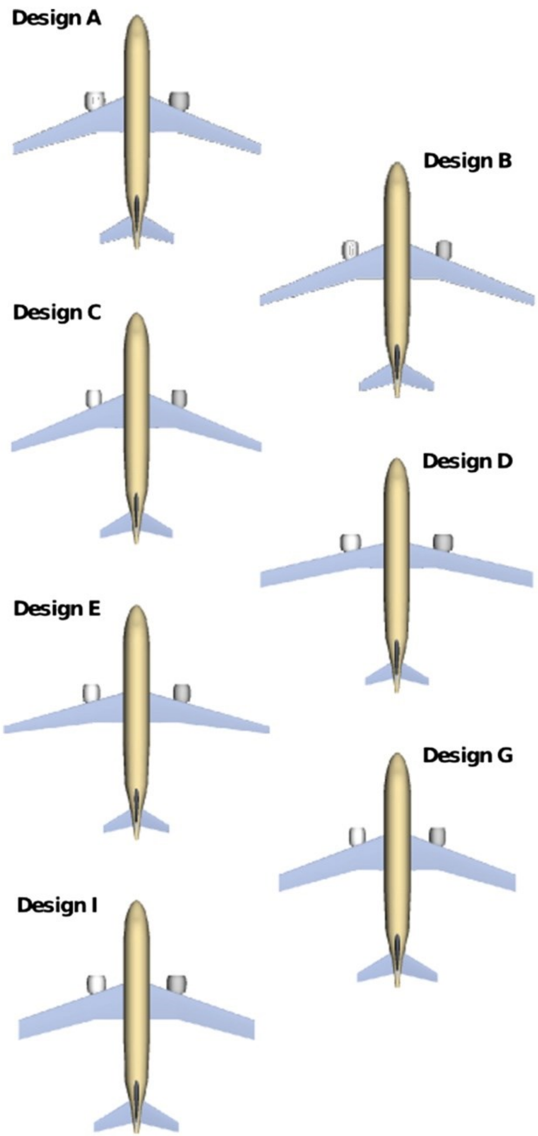
What is the best (optimized) shape?

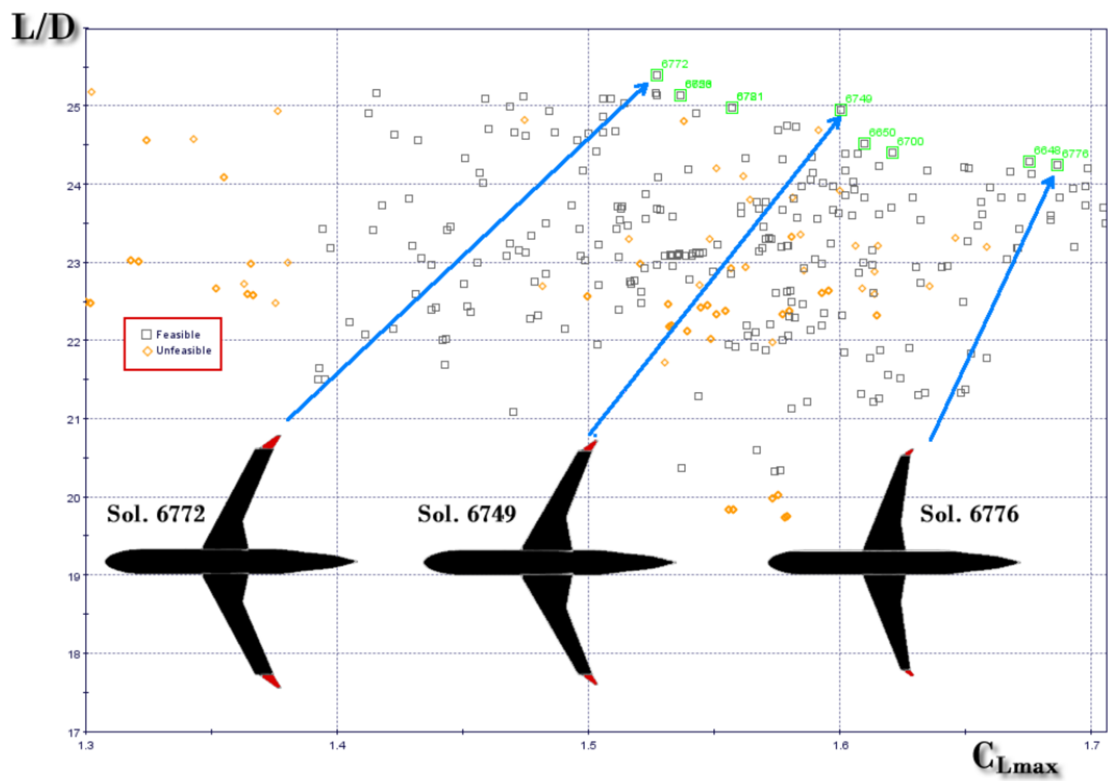




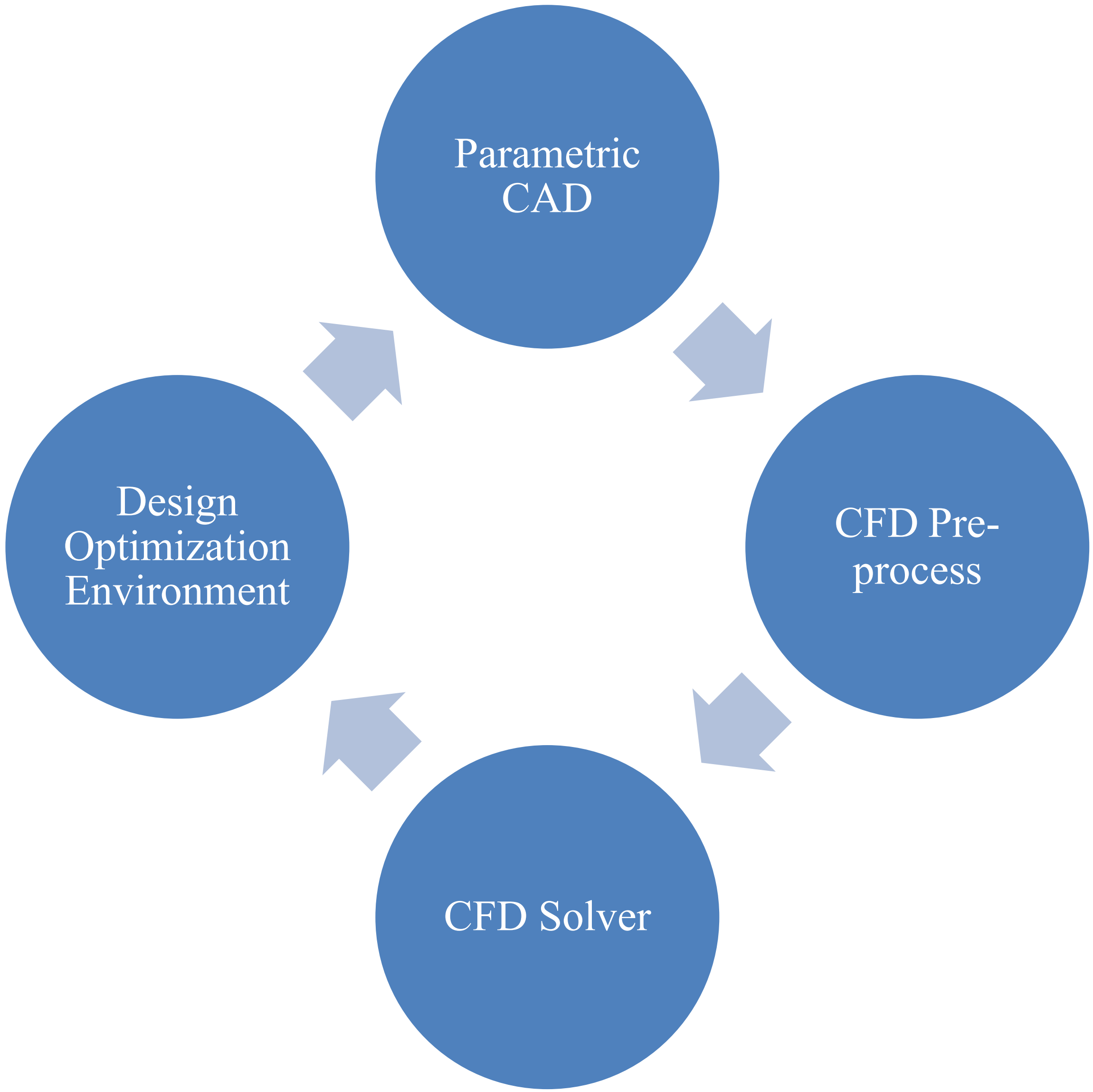
1 Introduction and motivation

What is the best (optimized) shape?



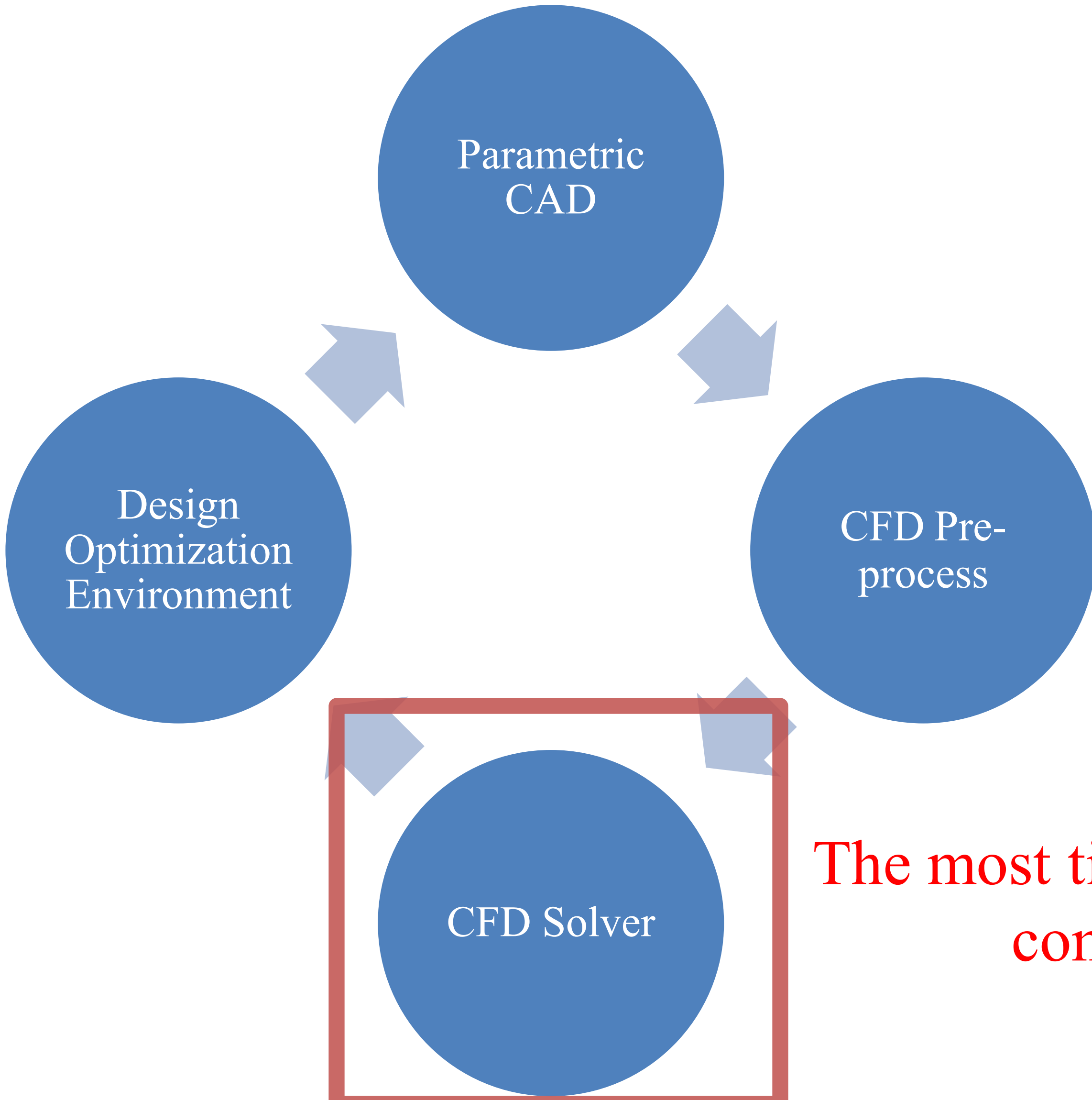


Shape optimization based on geometric parameters



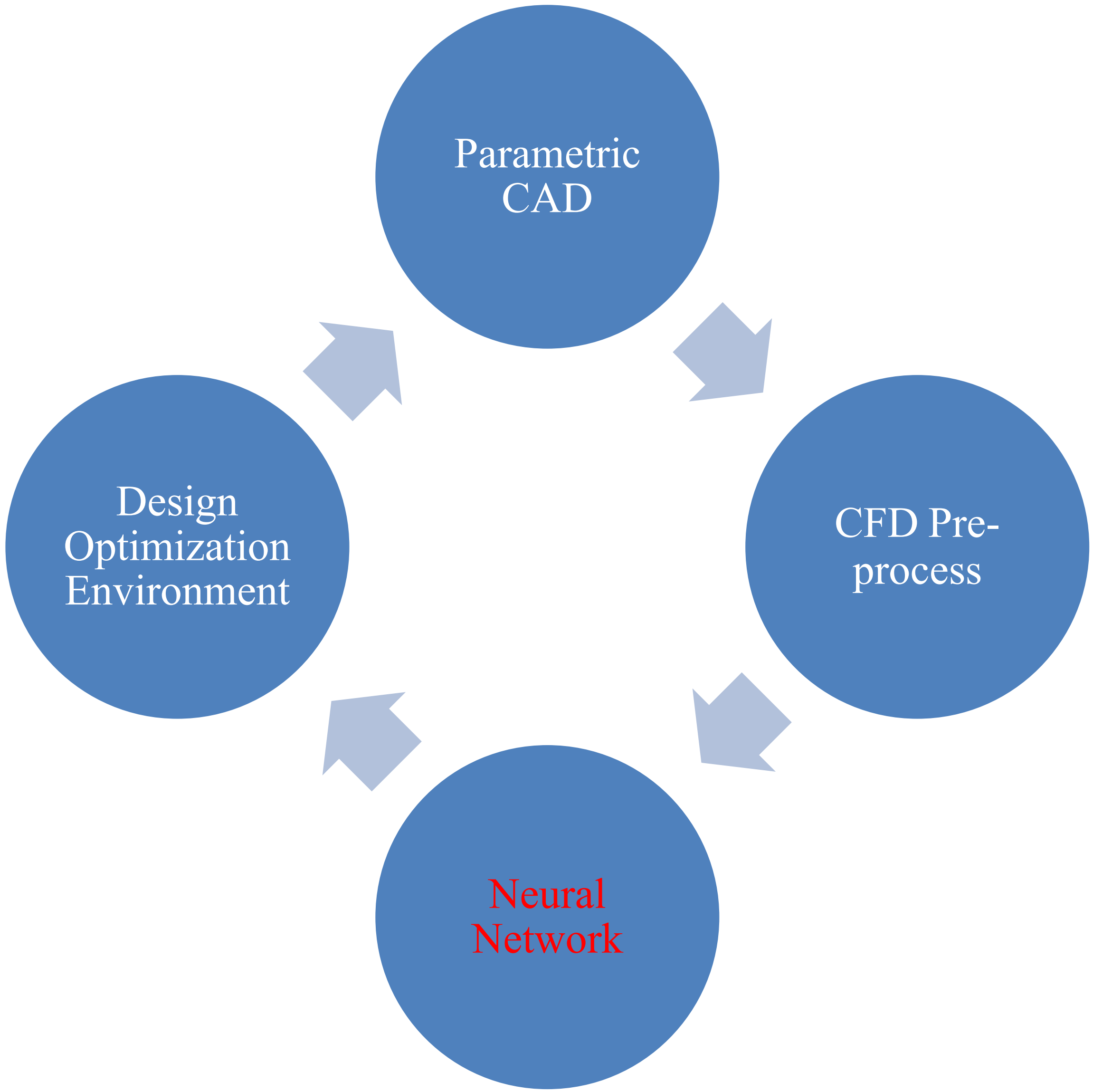
1 Introduction and motivation

Shape optimization based on geometric parameters



The most time consuming component

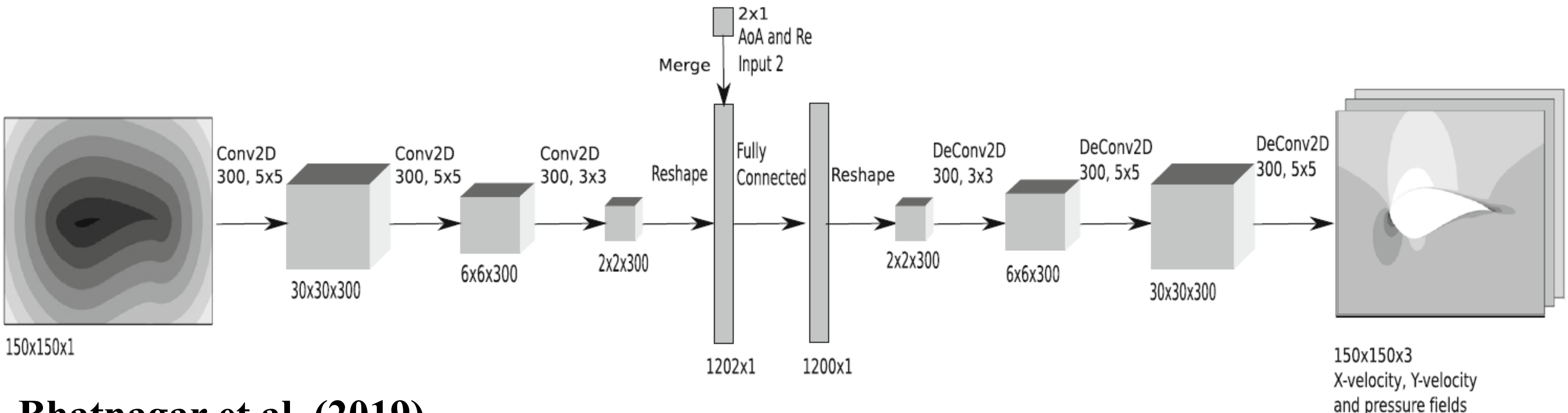
Shape optimization based on geometric parameters



1 Introduction and motivation

How to represent the geometry?

“Pixelation”, a so-far-used strategy for this purpose!



Bhatnagar et al. (2019)

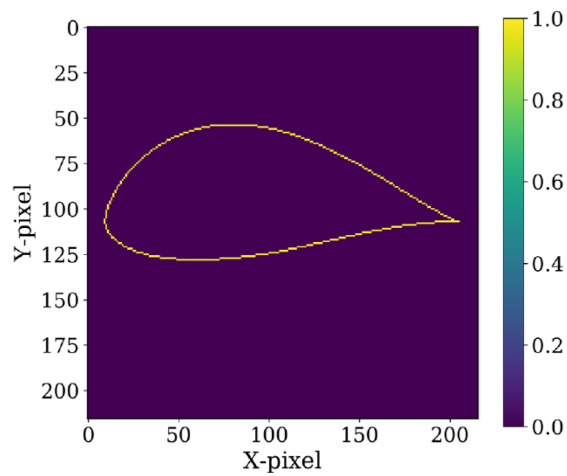
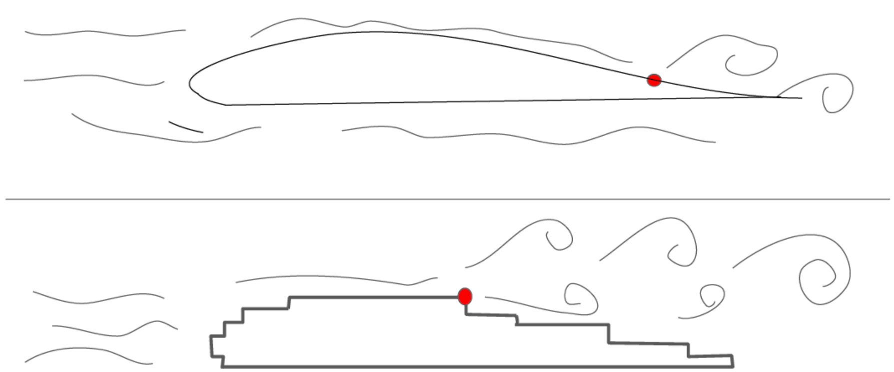


FIG. 5. Airfoil input image for the parameterization CNN.

5 main shortcoming

1# Pixelation leads to coarsening previously-smooth boundaries of a shape and introduces artificial roughness to its surface. This error can dramatically change the flow features such as the location of the detachment point on the surface of an airfoil.



Detachment points for an airfoil and its voxelisation.

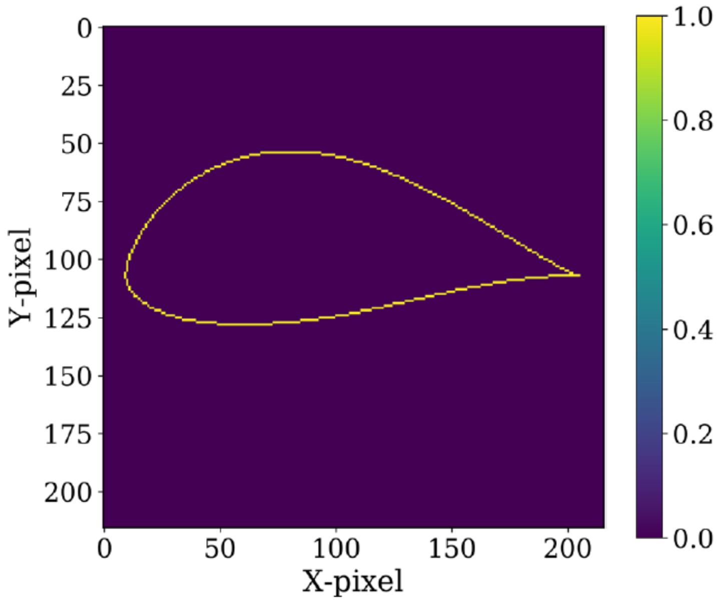
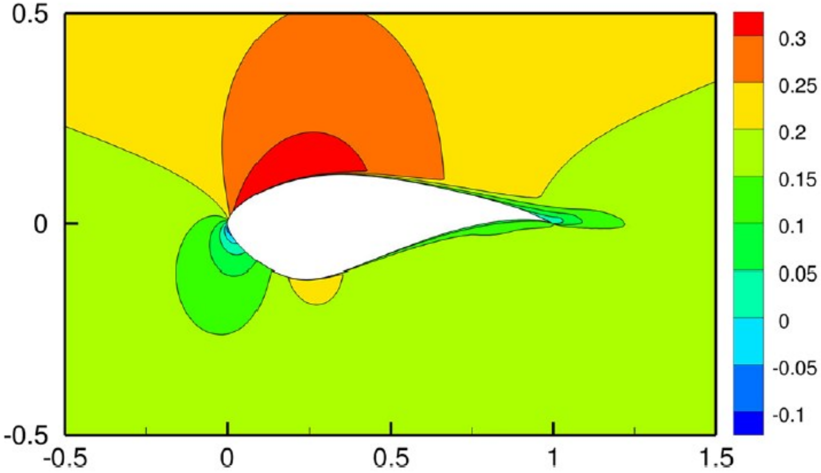


FIG. 5. Airfoil input image for the parameterization CNN.

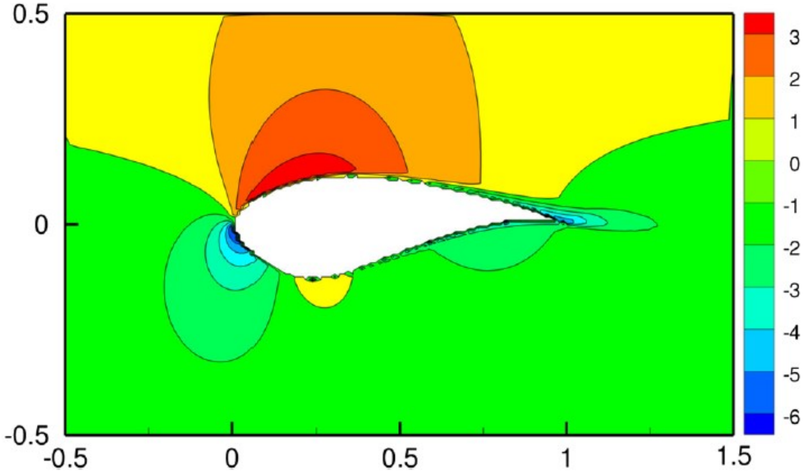
Sekar et al. (2019)

5 main shortcoming

2# Decreasing the order of accuracy of CFD data due to the data interpolation/extrapolation.



Before interpolation



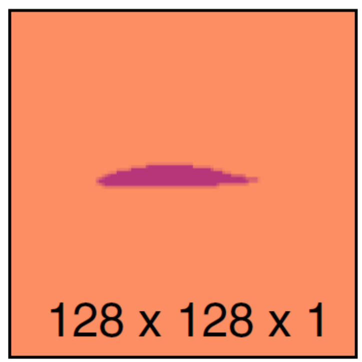
After interpolation

Bhatnagar et al. (2019)

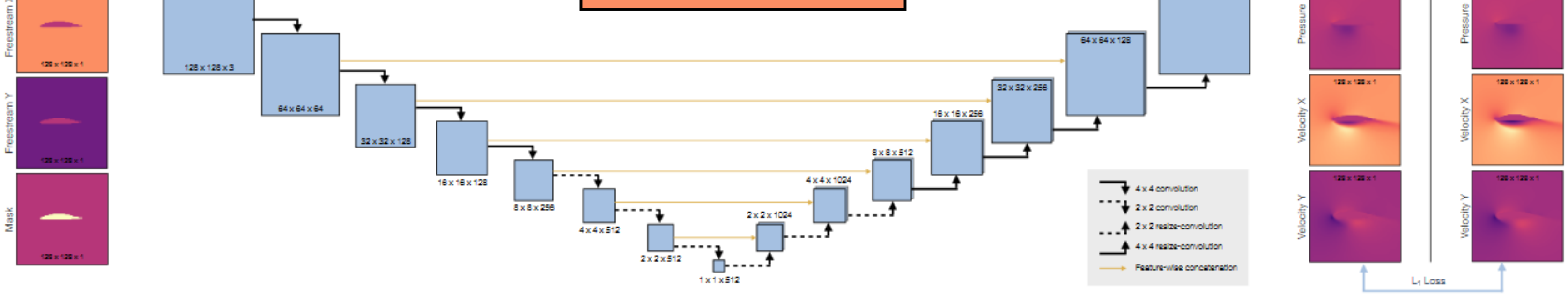
1 Introduction and motivation

5 main shortcoming

3# While the flow field around an object is highly sensitive to small changes (e.g., rotation or length increment of the object), the pixelation method cannot capture these changes unless a CNN with super resolution input is used, which by itself imposes high computational cost to the system.

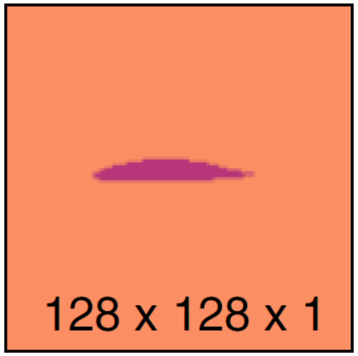


Thuerey et al. (2020)

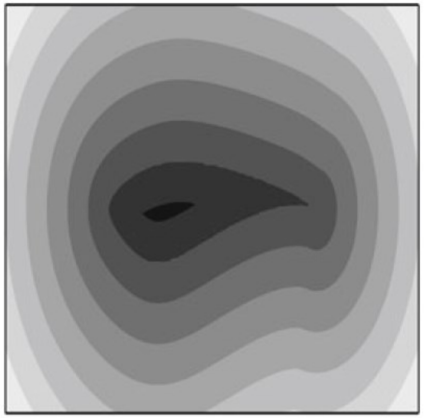


5 main shortcoming

4# It is common to mask the interior points of objects However, this procedure leads to ignoring the corresponding pixels of a CNN and, in fact, some portions of its computational potential.



Thuerey et al. (2020)



150x150x1

Bhatnagar et al. (2019)

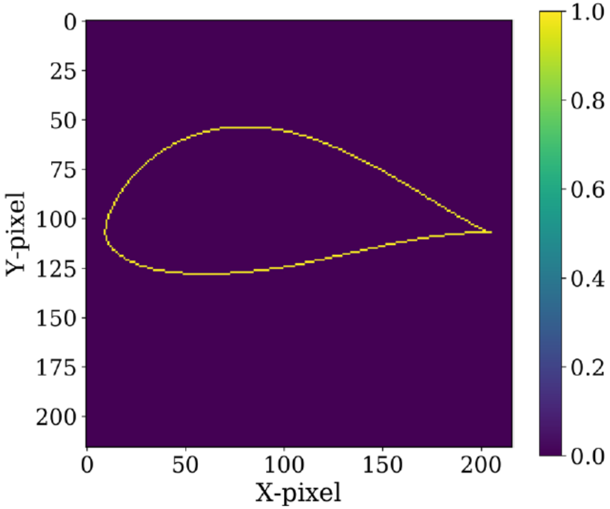
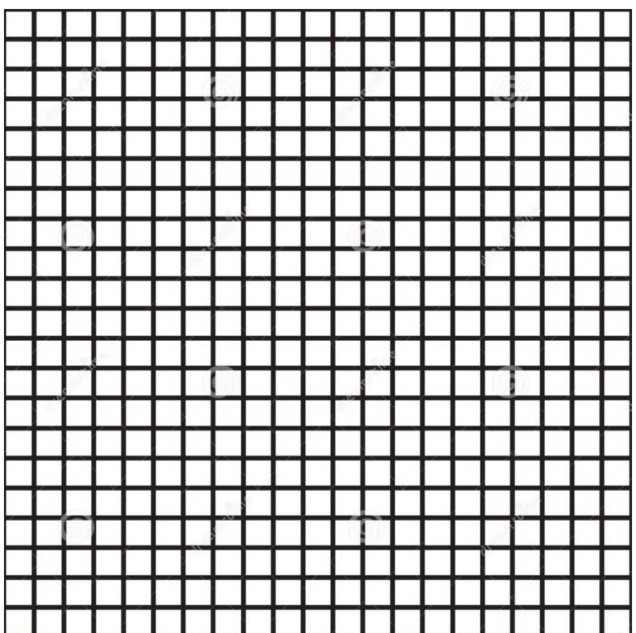


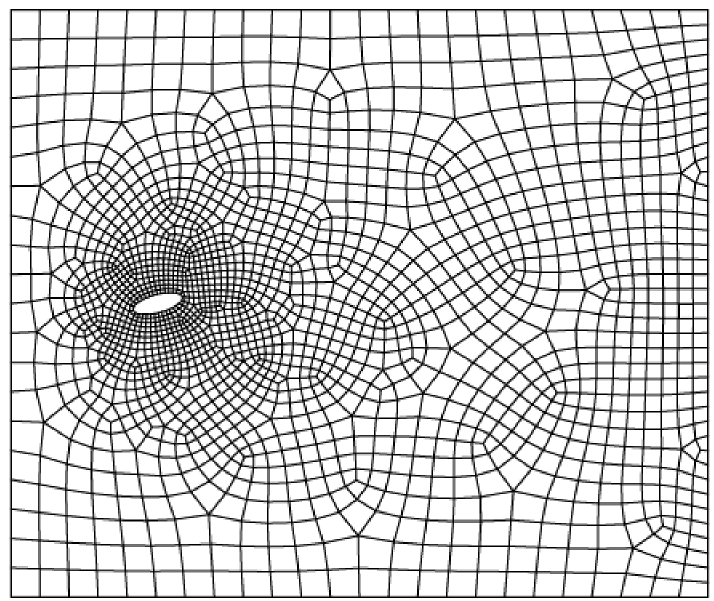
FIG. 5. Airfoil input image for the parameterization CNN.

Sekar et al. (2019)

5 main shortcoming

5# The importance of information in a CFD domain is not equal. For instance, the velocity and pressure fields near the surface of an airfoil and in its wake region are more important than other areas. Nonetheless, using pixelation and consequently Cartesian grids, the distribution of CNN pixels is uniform everywhere in the domain!





2 Problem formulation

2. 1 Governing equations of fluid dynamics

Navier-Stokes & continuity for incompressible flows

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right] - \mu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{f} \text{ in } V, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \text{ in } V, \tag{2}$$

$$\boldsymbol{u} = \boldsymbol{u}_{\Gamma_D} \text{ on } \Gamma_D, \tag{3}$$

$$-p\boldsymbol{n} + \mu\nabla\boldsymbol{u}\cdot\boldsymbol{n} = \boldsymbol{t}_{\Gamma_N} \text{ on } \Gamma_N, \tag{4}$$

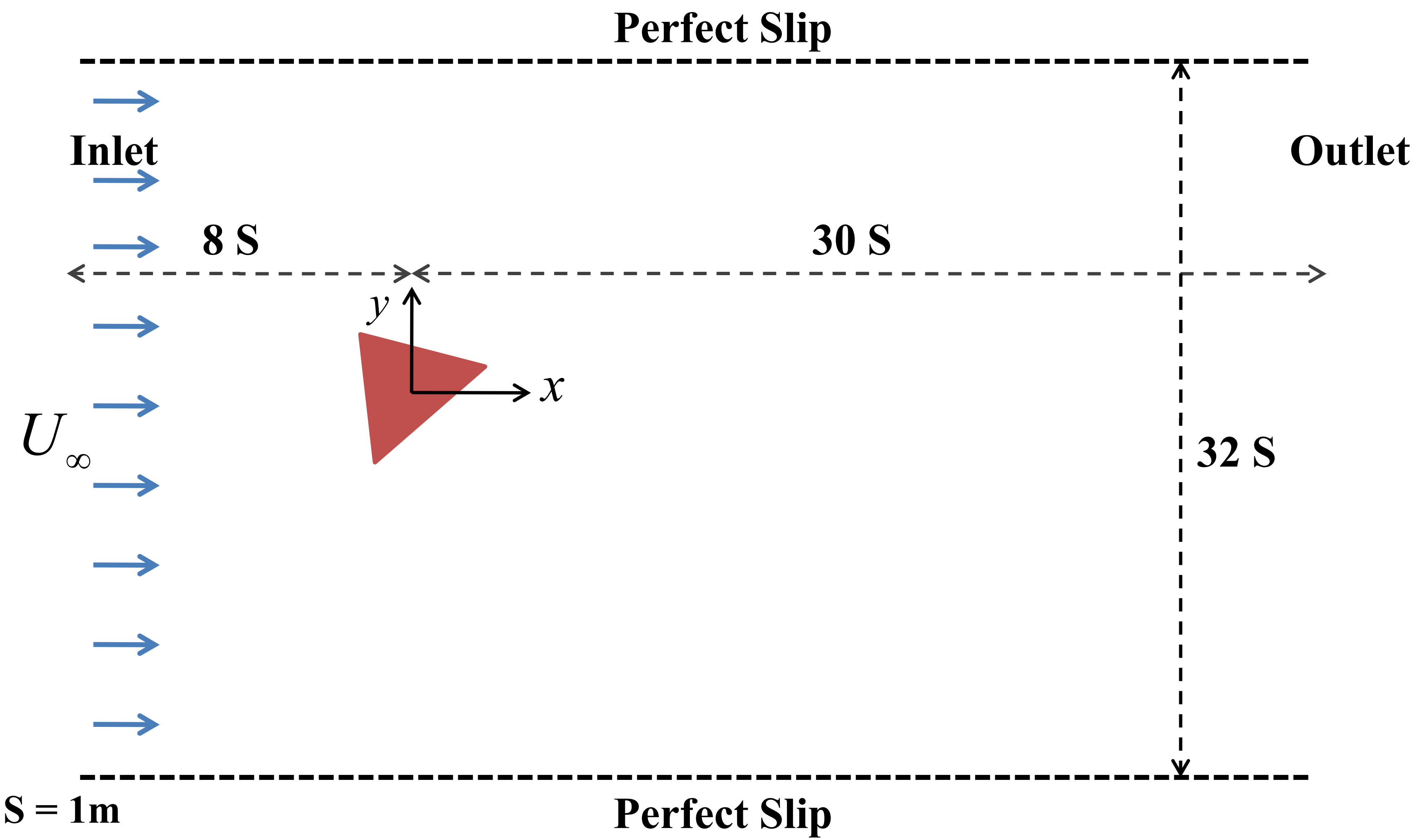
Velocity vector: $\boldsymbol{u} = (u, v, w)$

Pressure: p

Fluid density: ρ

Fluid viscosity: μ

Geometry and boundary conditions



2. 1 Governing equations of fluid dynamics

Mesh Generator: **Gmsh** (open source)

gmsh

http://gmsh.info/

CFD Solver: **OpenFoam** (open source)

OpenFOAM

https://openfoam.org/

2. 1 Governing equations of fluid dynamics

(a)

(b)

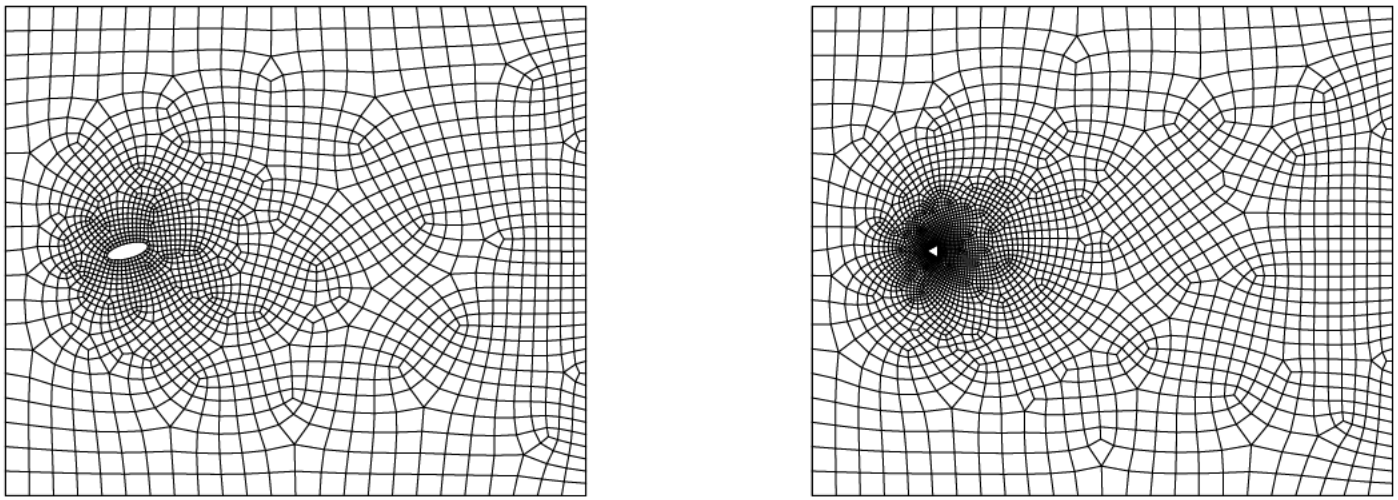


Figure 1: Representation of the finite volume meshes used for solving the continuity and Navier-Stokes equations in the simulation of flow over a cylinder; **a** An elliptical cross section, 2672 vertices; **b** A triangular cross section, 2775 vertices

2. 1 Governing equations of fluid dynamics

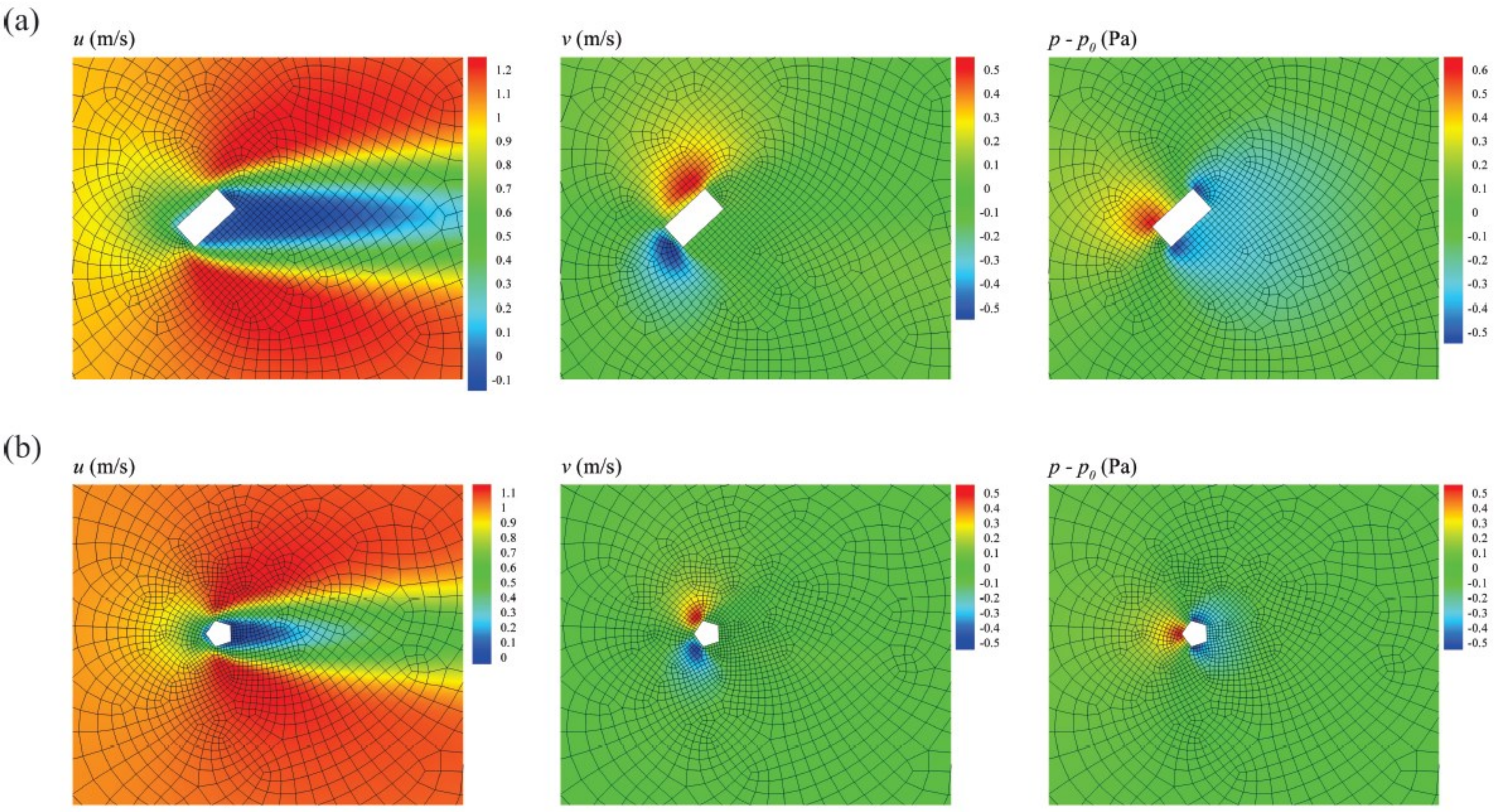
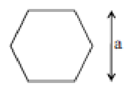
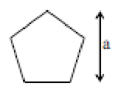
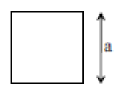
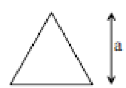
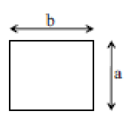
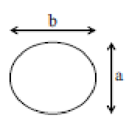
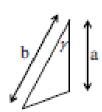


Figure 2: Velocity and pressure fields for the steady-state flow over a cylinder with **a** rectangular cross section and **b** pentagonal cross section; p_0 is the atmospheric pressure

2. 2 Data generation

TABLE I. Description of the generated data

Shape	Schematic figure	Variation in orientation	Variation in length scale	Number of data
Circle		-	$a = 1$ m	1
Equilateral hexagon		$3°, 6°, \dots, 60°$	$a = 1$ m	20
Equilateral pentagon		$3°, 6°, \dots, 72°$	$a = 1$ m	24
Square		$3°, 6°, \dots, 90°$	$a = 1$ m	30
Equilateral triangle		$3°, 6°, \dots, 180°$	$a = 1$ m	60
Rectangle		$3°, 6°, \dots, 180°$	$a = 1$ m; $b/a = 1.2, 1.4, \dots, 3.6$	780
Ellipse		$3°, 6°, \dots, 180°$	$a = 1$ m; $b/a = 1.2, 1.4, \dots, 4.2$	960
Triangle		$3°, 6°, \dots, 360°$	$a = 1$ m; $b/a = 1.5, 1.75$ $\gamma = 40°, 60°, 80°$	720

2. 2 Data generation

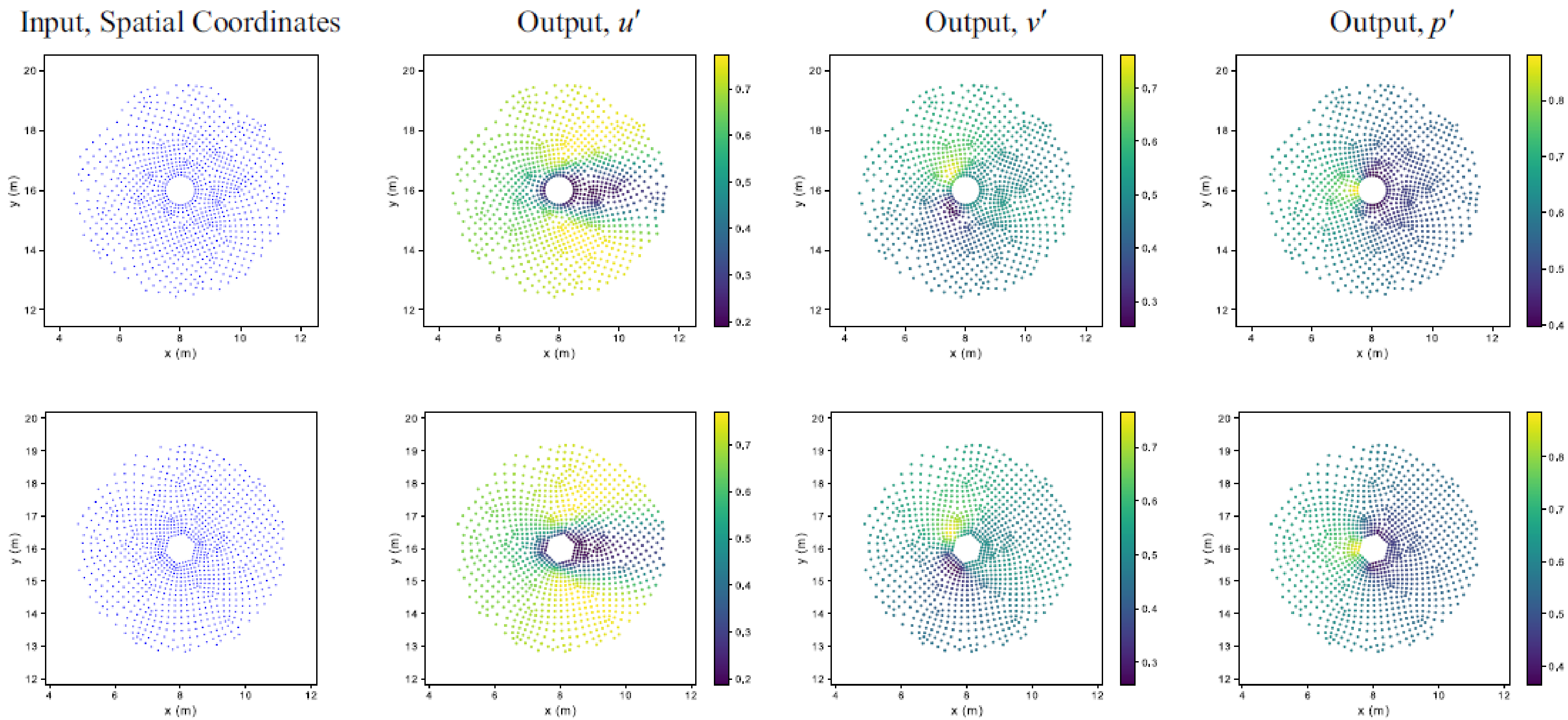


Figure 4: Examples of input and output data

2. 3 Data generation

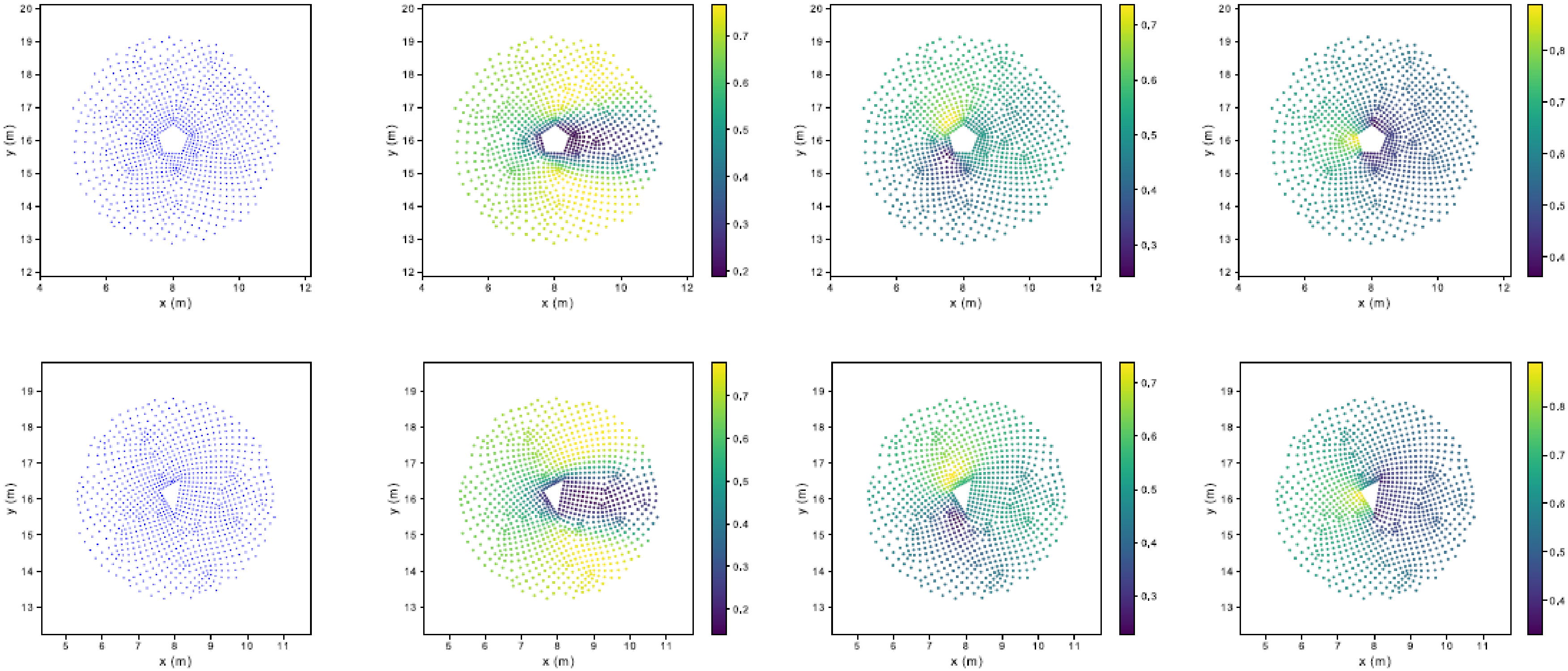


Figure 4: Examples of input and output data

2. 3 Data generation

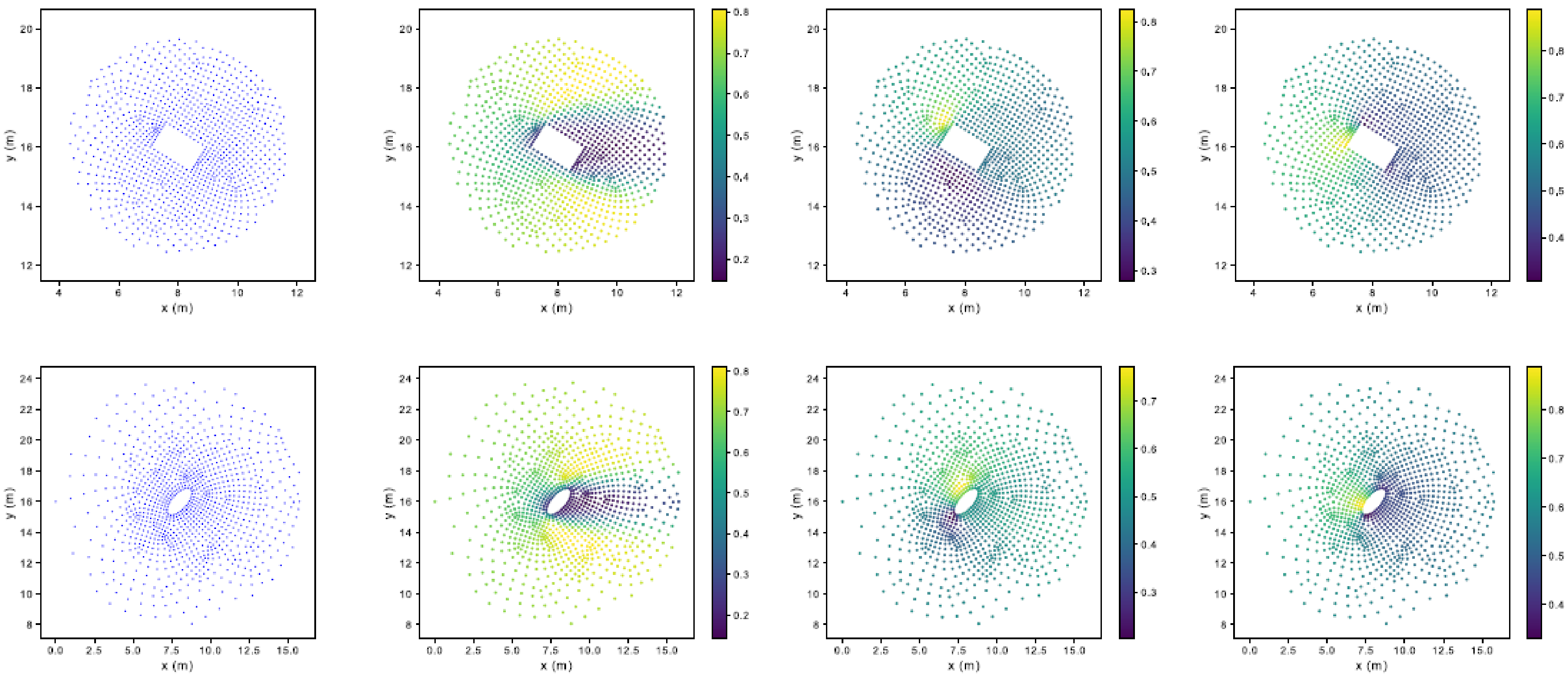


Figure 4: Examples of input and output data

2. 3 Neural network architecture

Segmentation component of PointNet

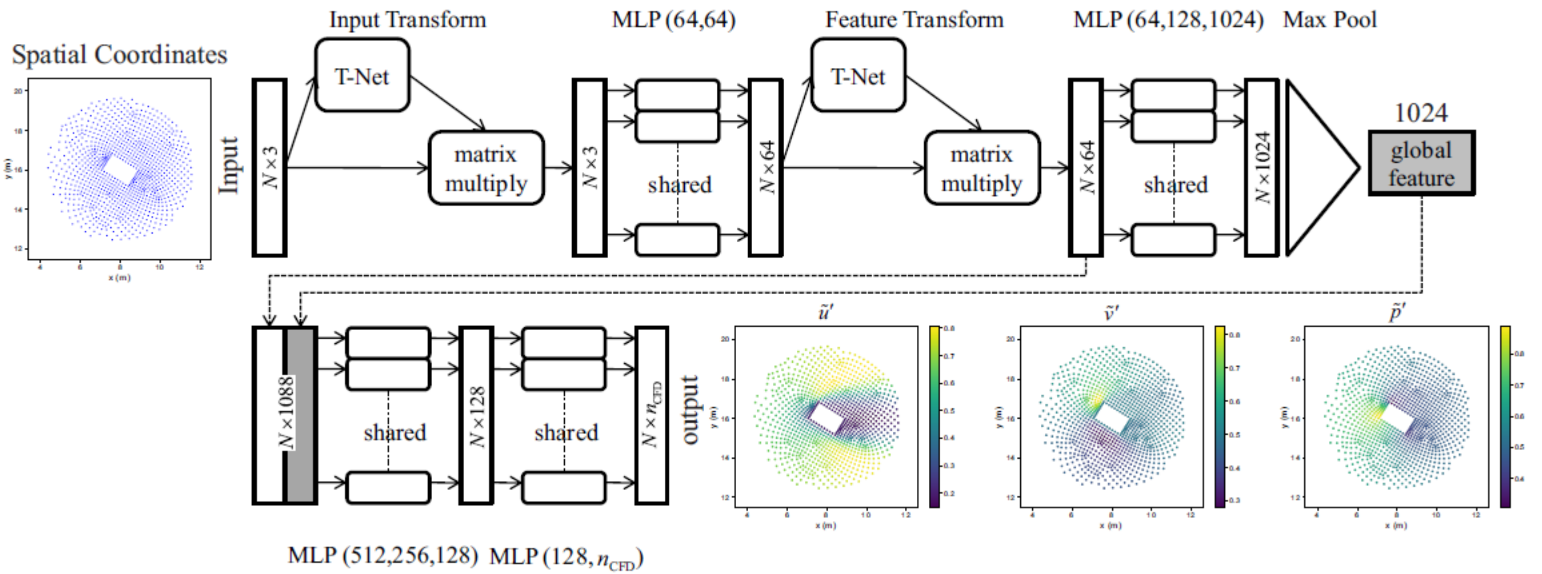


FIG. 5. Structure of our neural network; Labels in the format of (A, B) demonstrate the size of the first layer, A, and the second layer, B, of the MLP. Labels in the form of (A, B, C) are similarly classified for three layers. n_{CFD} indicates the number of CFD variables; in this study, $n_{\mathrm{CFD}} = 3$. The figure shows the structure for handling three-dimensional problems; though we consider two-dimensional problems in this study. Some parts of this figure are reproduced from Ref. 37.

2. 4 Training

$$\mathcal{L} = \frac{1}{3 \times N} \left(\sum_{i=1}^{N} \left[(u_i^{'} - \tilde{u}_i^{'})^2 + (v_i^{'} - \tilde{v}_i^{'})^2 + (p_i^{'} - \tilde{p}_i^{'})^2 \right] \right)$$

- Adam optimizer
- Batch size of 256
- 3552588 parameters
- Three sets of training (80%), validation (10%), and test (10%) through a random process
- Train the neural network over 2076 data; 260 data for the validation and the remaining 259 for evaluation
- NVIDIA TITAN Xp graphics card with the memory clock rate of 1.582 GHz and 12 Gigabytes of RAM
- Stop after 4000 epochs
- Approximately takes 10 hours

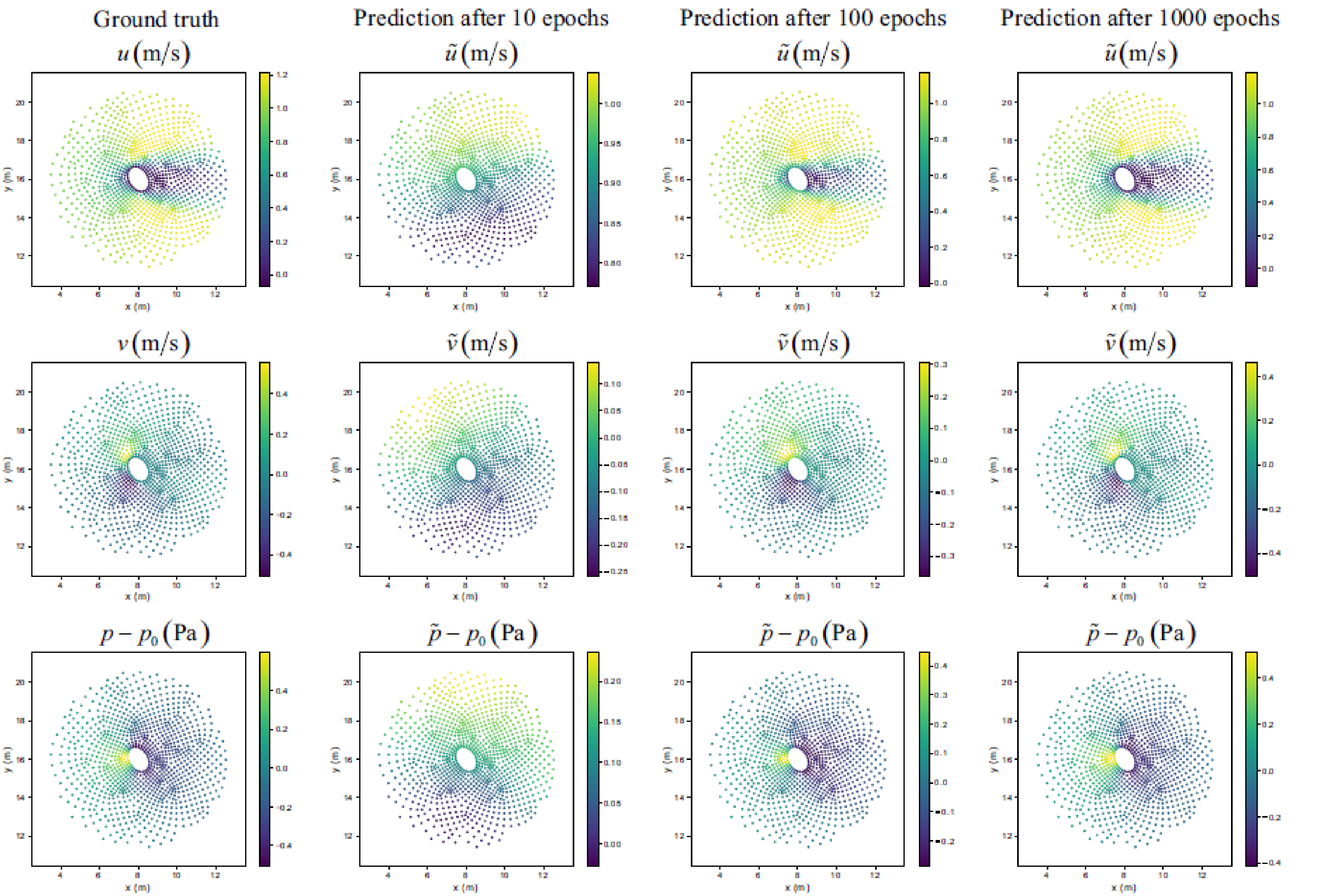


FIG. 6. A comparison between the ground truth and prediction of the network for the velocity and pressure fields after 10, 100, and 1000 epochs

3 Results and discussion

3.1 General analysis

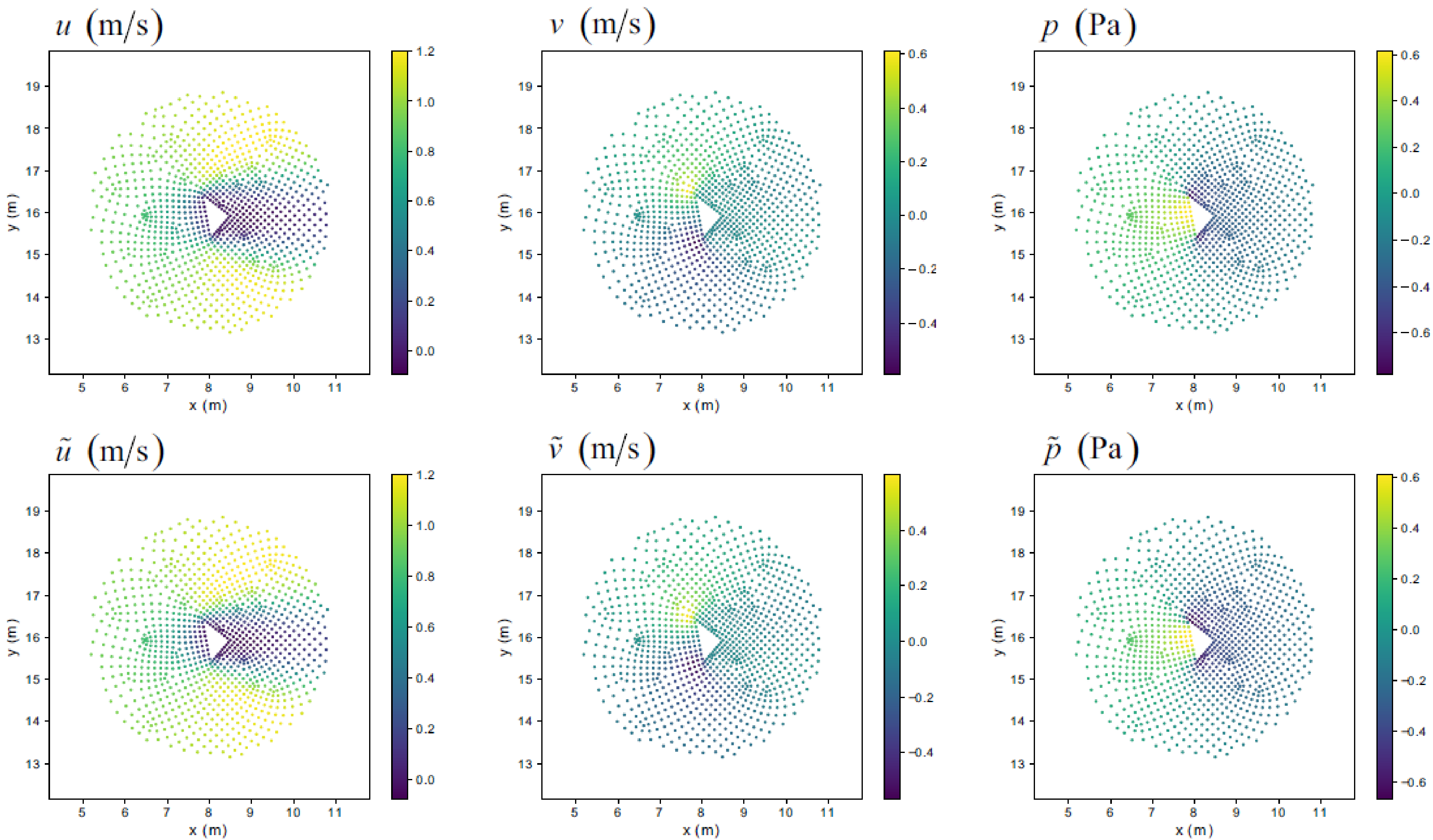


Figure 6: A comparison between the ground truth and our network prediction for the velocity and pressure fields for two different cross sections; the first set of examples

3.1 General analysis

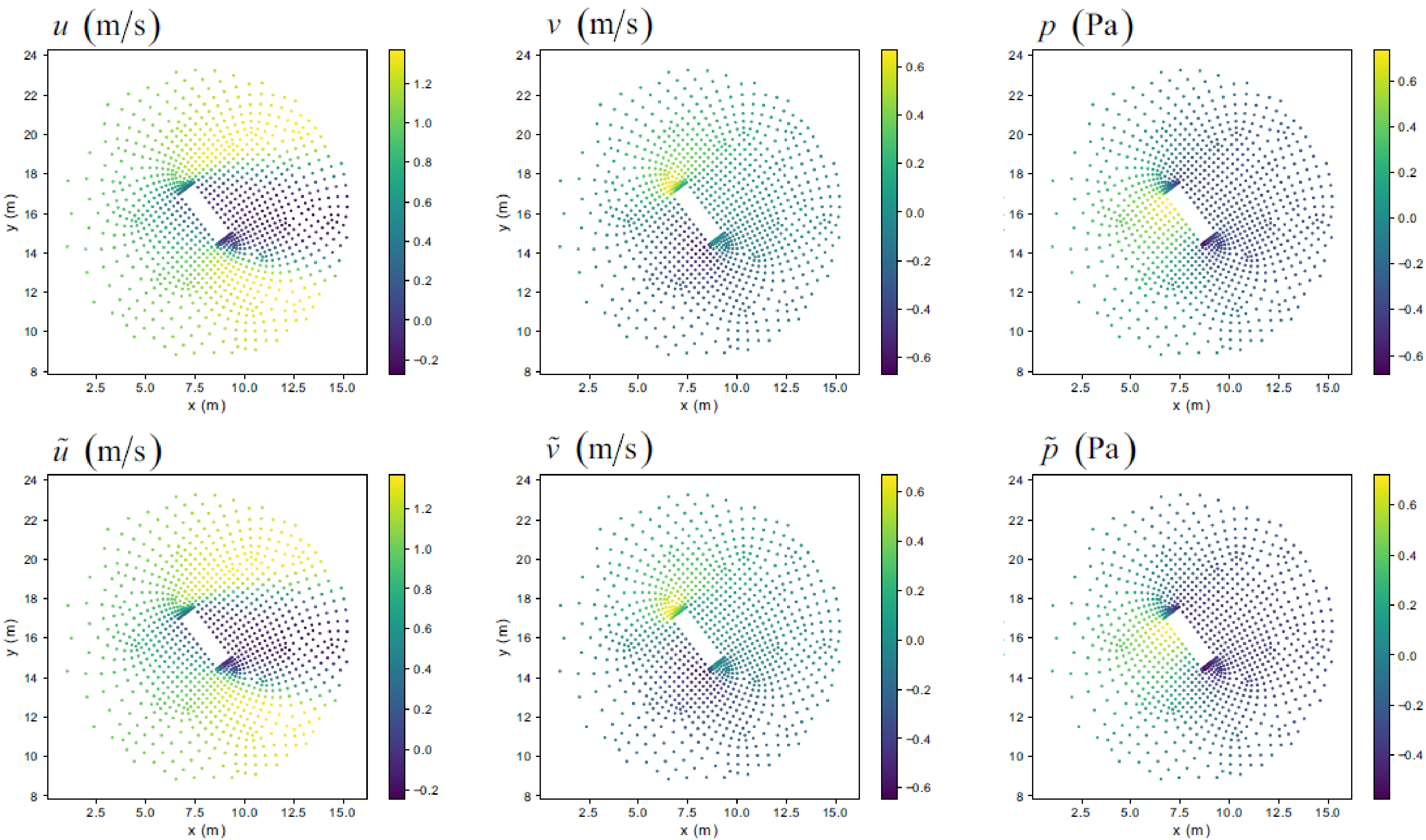


Figure 6: A comparison between the ground truth and our network prediction for the velocity and pressure fields for two different cross sections; the first set of examples

3.1 General analysis

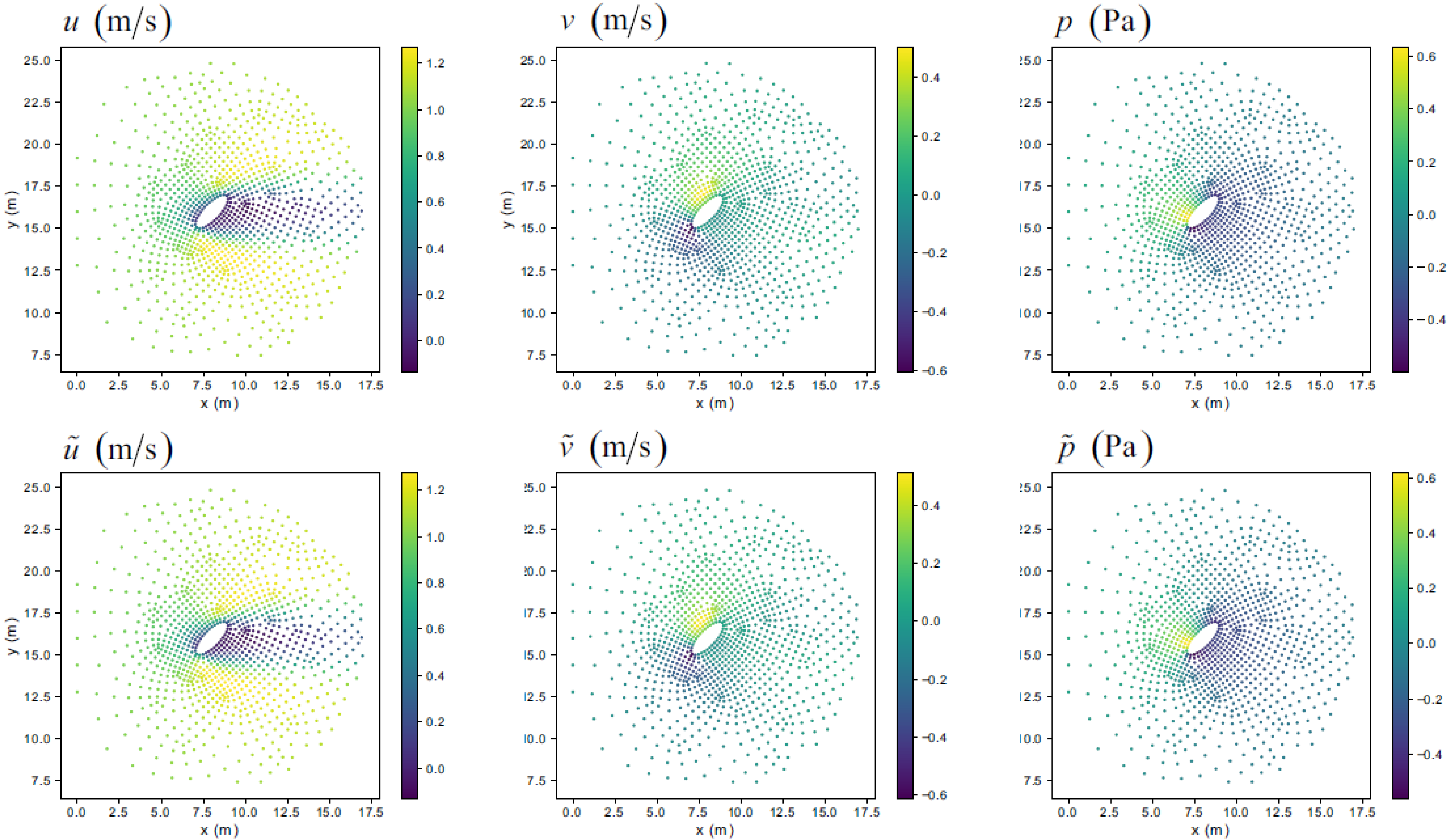


Figure 7: A comparison between the ground truth and our network prediction for the velocity and pressure fields for two different cross sections, the second set of examples

3.1 General analysis

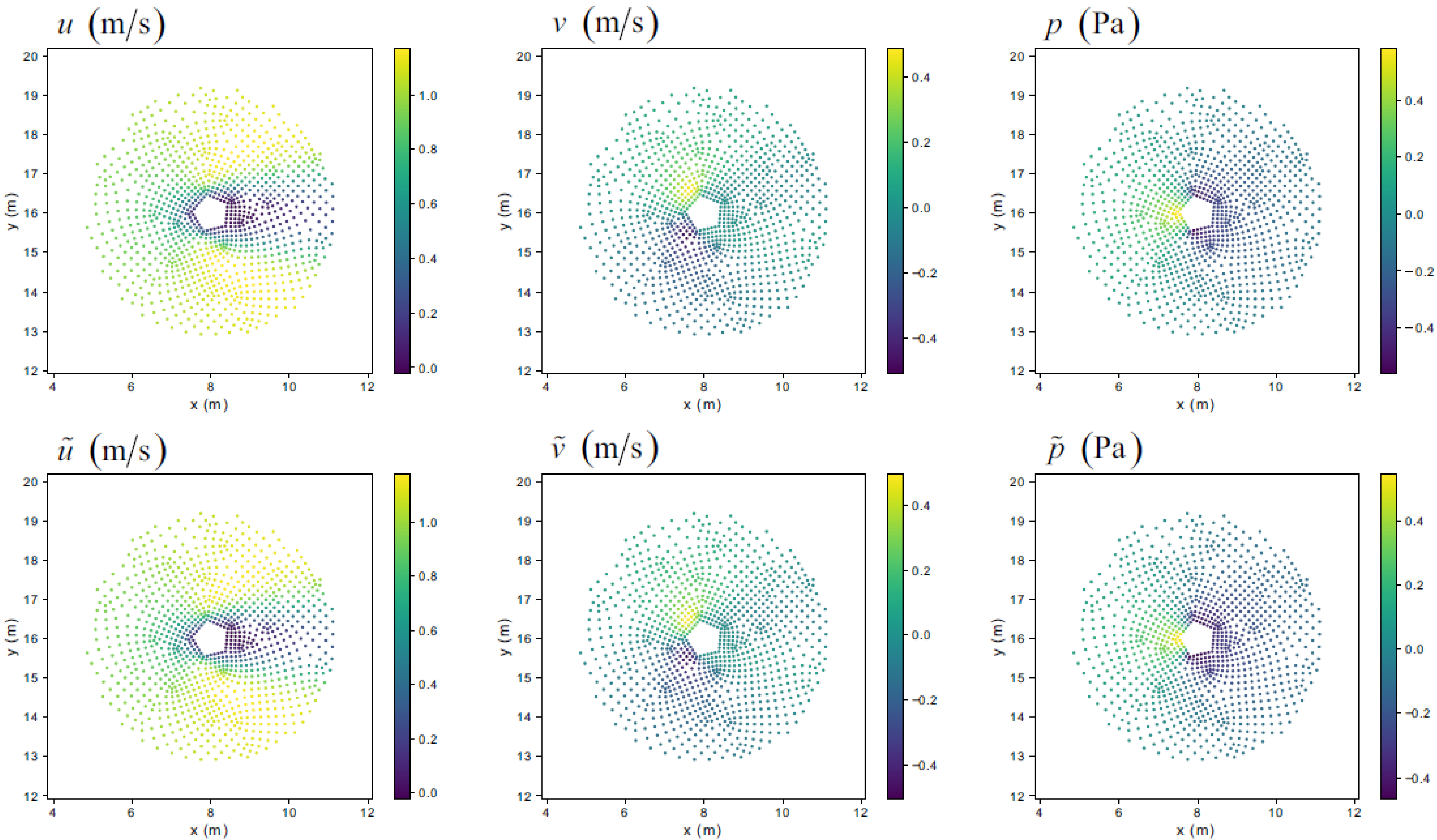


Figure 7: A comparison between the ground truth and our network prediction for the velocity and pressure fields for two different cross sections, the second set of examples

3.1 General analysis

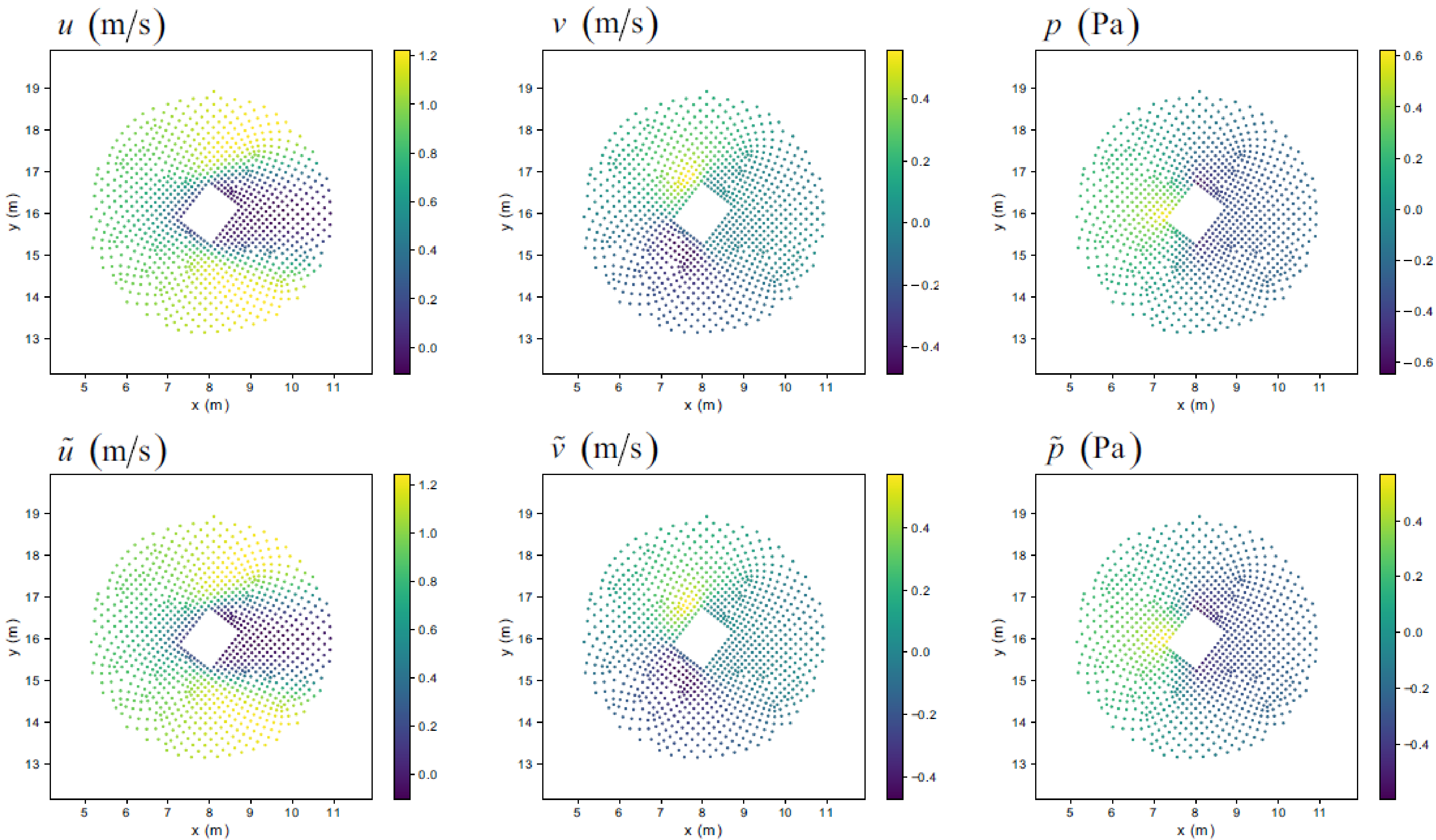


Figure 8: A comparison between the ground truth and our network prediction for the velocity and pressure fields for two different cross sections, the third set of examples

3.1 General analysis

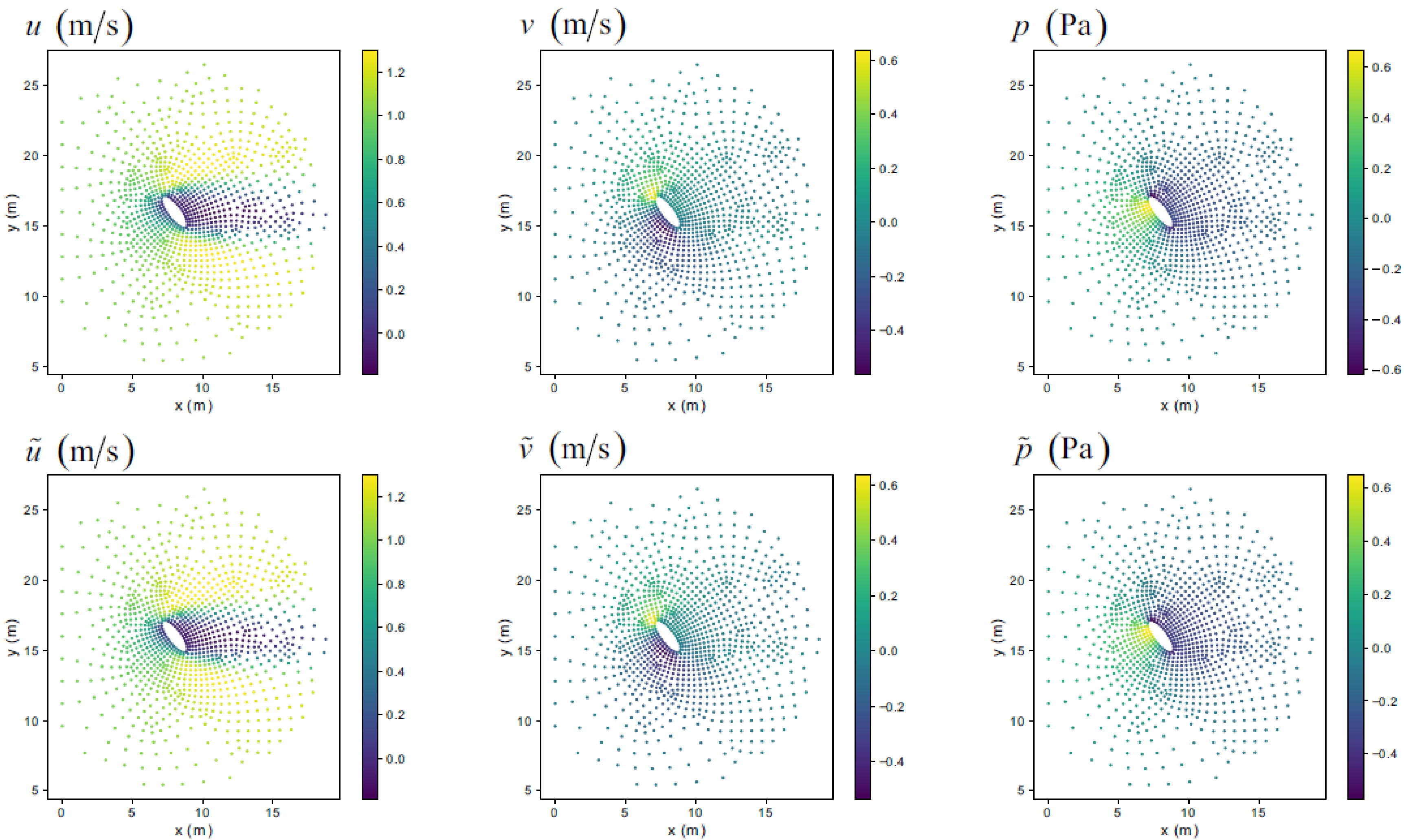


Figure 8: A comparison between the ground truth and our network prediction for the velocity and pressure fields for two different cross sections, the third set of examples

3.1 General analysis

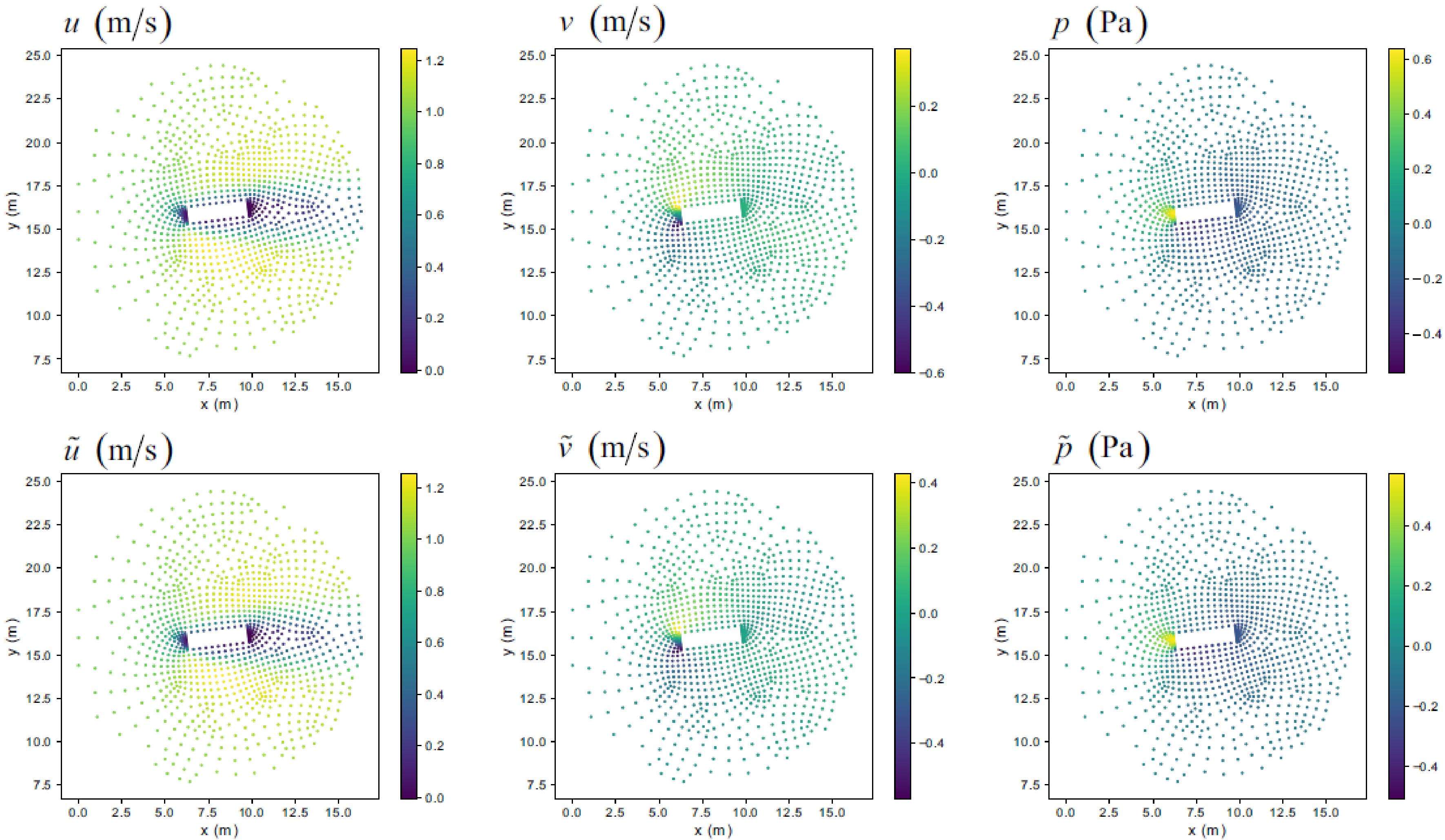


Figure 9: A comparison between the ground truth and our network prediction for the velocity and pressure fields for two different cross sections, the fourth set of examples

3.1 General analysis

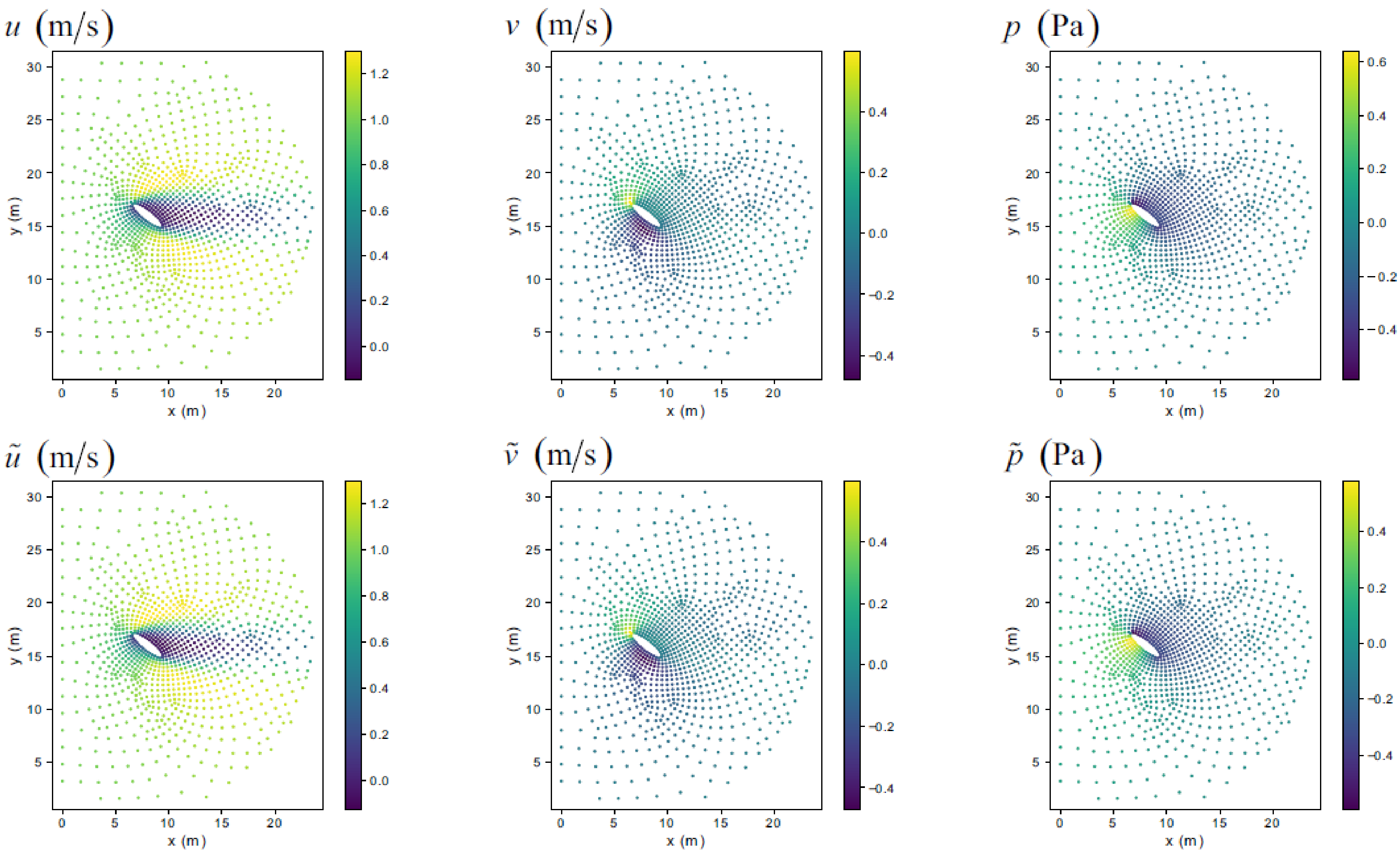


Figure 9: A comparison between the ground truth and our network prediction for the velocity and pressure fields for two different cross sections, the fourth set of examples

3.1 General analysis

TABLE II. Error analysis of the velocity and pressure fields predicted by our neural network for 259 unseen data; $||\ldots||$ indicates the L^2 norm.

	$\|\|u - \tilde{u}\|\|$	$\|\|v - \tilde{v}\|\|$	$\|\|p - \tilde{p}\|\|$
Average	4.49666E−2	3.70540E−2	2.71661E−2
Maximum	2.49088E−1	2.34281E−1	1.16901E−1
Minimum	1.10453E−2	9.20977E−3	7.58447E−3

3.1 General analysis

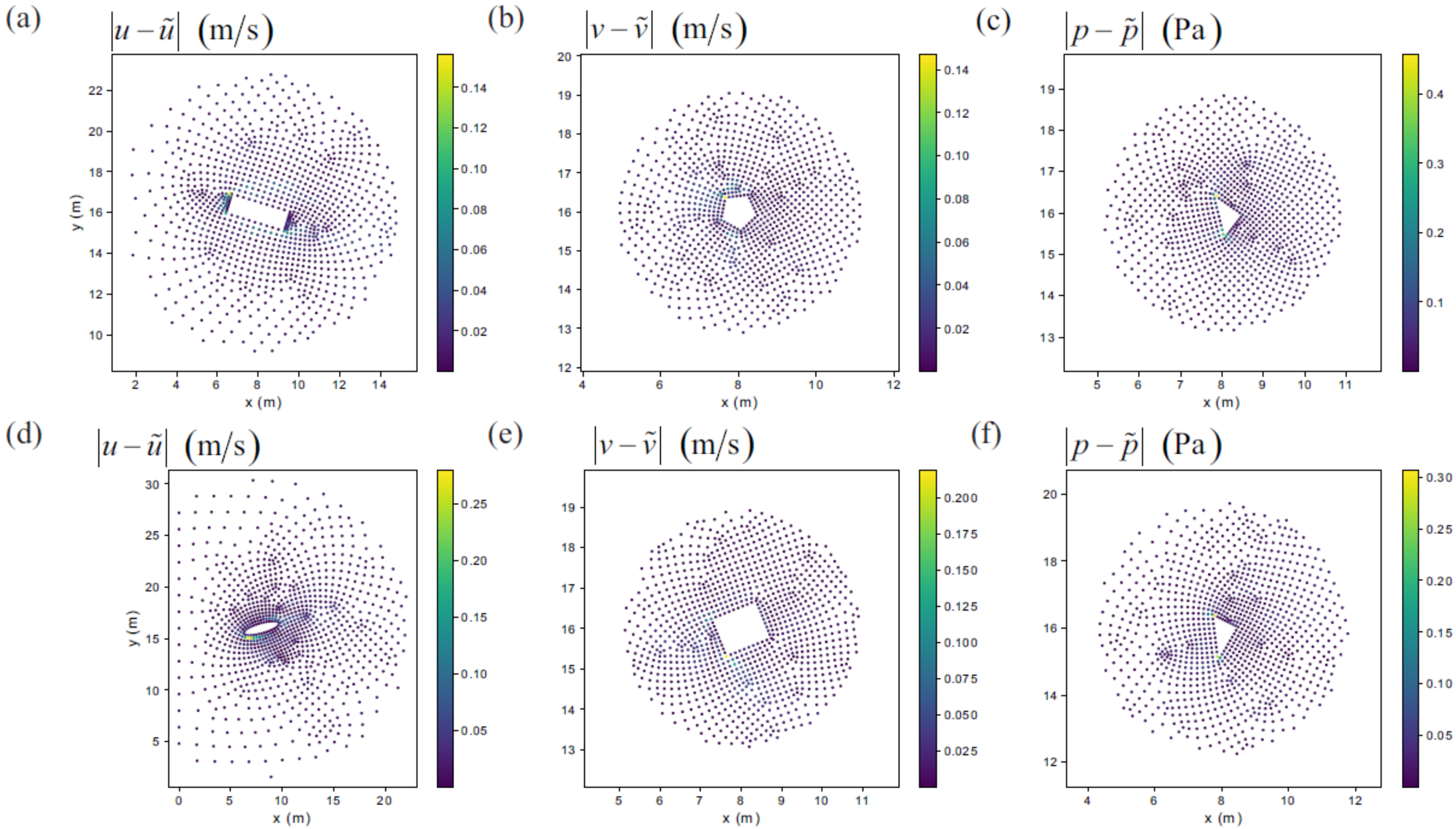


FIG. 11. Distribution of absolute pointwise error when the mean square error becomes **a** maximum for $\tilde{u}$, **b** maximum for $\tilde{v}$, **c** maximum for $\tilde{p}$, **d** minimum for $\tilde{u}$, **e** minimum for $\tilde{v}$, and **f** minimum for $\tilde{p}$.

3.1 General analysis

- Speed-up:

Wall time for prediction of the 259 unseen data: **6 seconds**

Simulation of the flow fields for these 259 geometries using the CFD software: **11071 seconds** (about 3 hours)

Average achieved speedup: **1846**

This is not an absolute number and depends on the framework of our available computational facilities!

3.2 Investigation of conservation of mass and momentum

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right] - \mu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{f} \text{ in } V, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \text{ in } V, \tag{2}$$

$$\boldsymbol{u} = \boldsymbol{u}_{\Gamma_D} \text{ on } \Gamma_D, \tag{3}$$

$$-p\boldsymbol{n} + \mu\nabla\boldsymbol{u}\cdot\boldsymbol{n} = \boldsymbol{t}_{\Gamma_N} \text{ on } \Gamma_N, \tag{4}$$

Residuals:

$$r_{momentum_x} = \left|\int_{V_{NN}} \left(\rho\left(\tilde{u}\frac{\partial\tilde{u}}{\partial x} + \tilde{v}\frac{\partial\tilde{u}}{\partial y}\right) + \frac{\partial\tilde{p}}{\partial x} - \mu\left(\frac{\partial^2\tilde{u}}{\partial x^2} + \frac{\partial^2\tilde{u}}{\partial y^2}\right)\right) dV\right|, \tag{16}$$

$$r_{momentum_y} = \left|\int_{V_{NN}} \left(\rho\left(\tilde{u}\frac{\partial\tilde{v}}{\partial x} + \tilde{v}\frac{\partial\tilde{v}}{\partial y}\right) + \frac{\partial\tilde{p}}{\partial y} - \mu\left(\frac{\partial^2\tilde{v}}{\partial x^2} + \frac{\partial^2\tilde{v}}{\partial y^2}\right)\right) dV\right|, \tag{17}$$

$$r_{continuity} = \left|\int_{V_{NN}} \left(\frac{\partial\tilde{u}}{\partial x} + \frac{\partial\tilde{v}}{\partial y}\right) dV\right|, \tag{18}$$

3.2 Investigation of conservation of mass and momentum

TABLE III. Investigation of conservation of mass and momentum of the flow fields predicted by our neural network for 259 unseen data. All values are reported in the International Unit System.

	$r_{momentum_x}$	$r_{momentum_y}$	$r_{continuity}$
Average	4.14958E−3	2.46155E−3	2.99411E−3
Maximum	3.38842E−1	3.59399E−2	8.74008E−2
Minimum	5.69245E−6	9.03372E−7	3.58928E−6

What does it mean?!

In computer graphics:

For instance, Qi et al. tested PointNet for the semantic segmentation of unseen categories such as “face”, “house”, “rabbit”, and “teapot”, while these objects did not exist in their data set.

In computational mechanics:

???

3.3 Neural network generalizability

What does it mean?!

In computer graphics:

For instance, Qi et al. tested PointNet for the semantic segmentation of unseen categories such as “face”, “house”, “rabbit”, and “teapot”, while these objects did not exist in their data set.

In computational mechanics:

???

3.3 Neural network generalizability

For the first time:

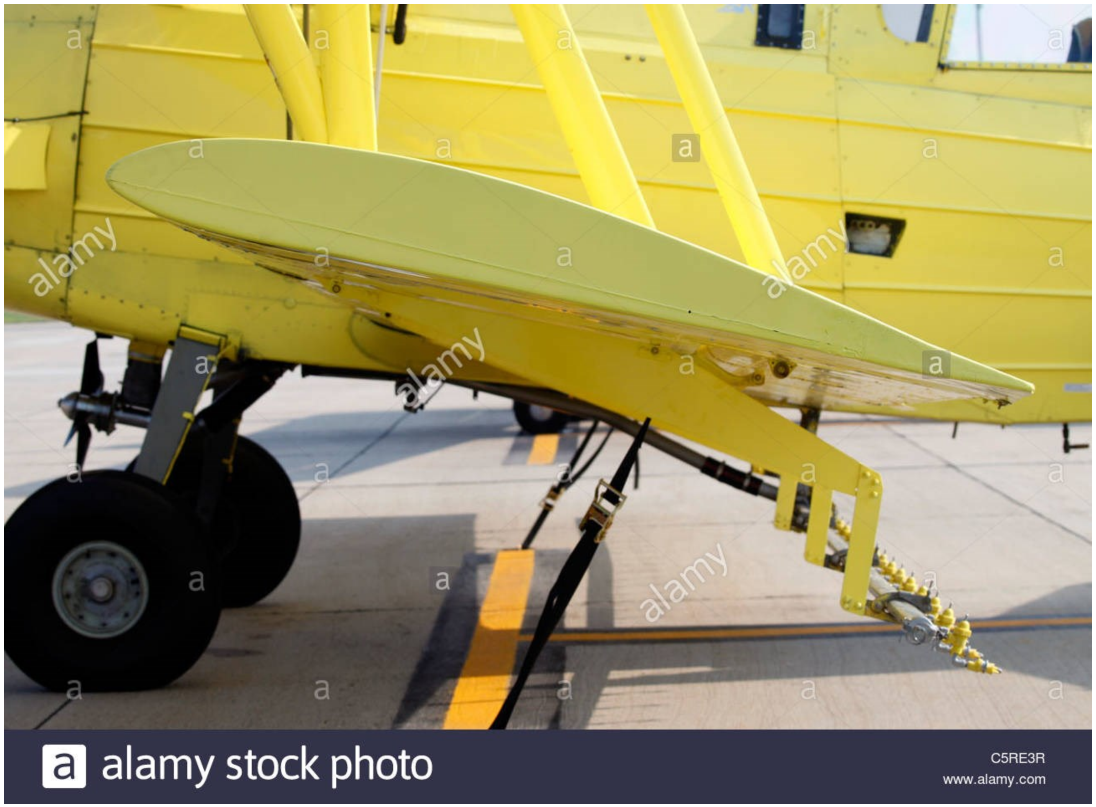
our network generalizes the predictions to **multiple objects** as well as an **airfoil**, even though **only single objects** and **no airfoils** are observed during training.

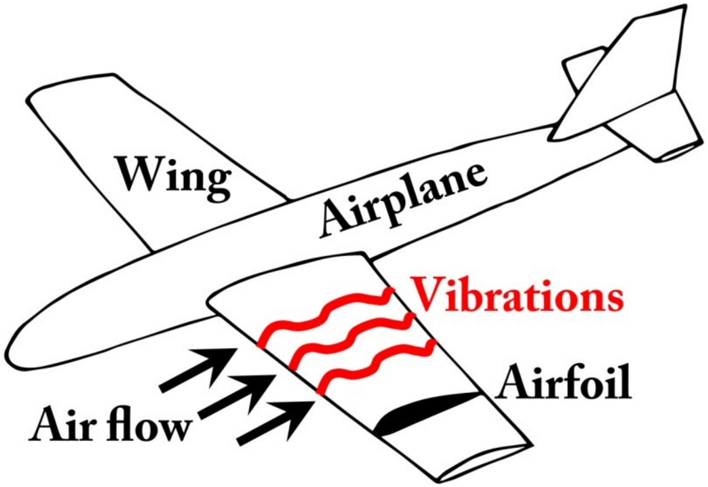
3.3.1 Prediction of flow around an airfoil

NACA 0028

3.3.1 Prediction of flow around an airfoil

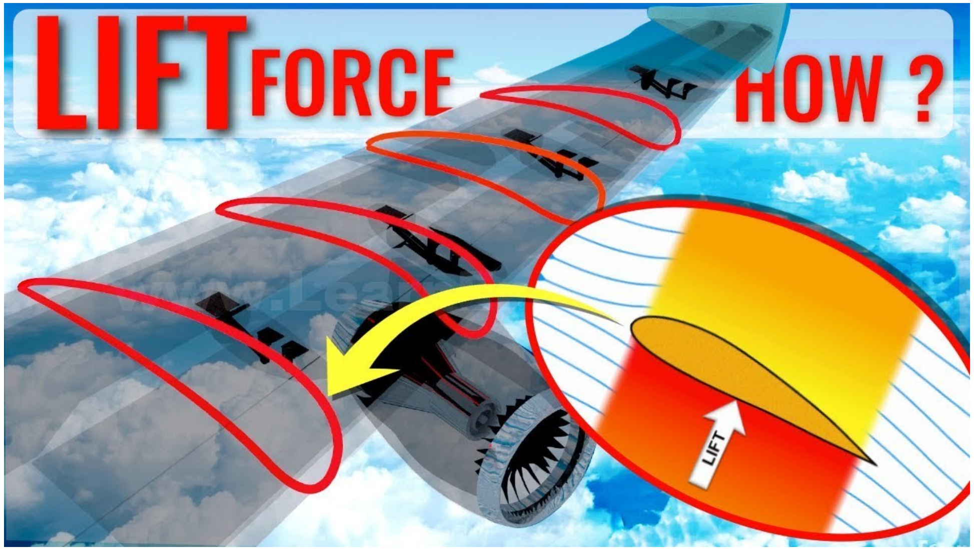
Applications of airfoils in **airplane wings**





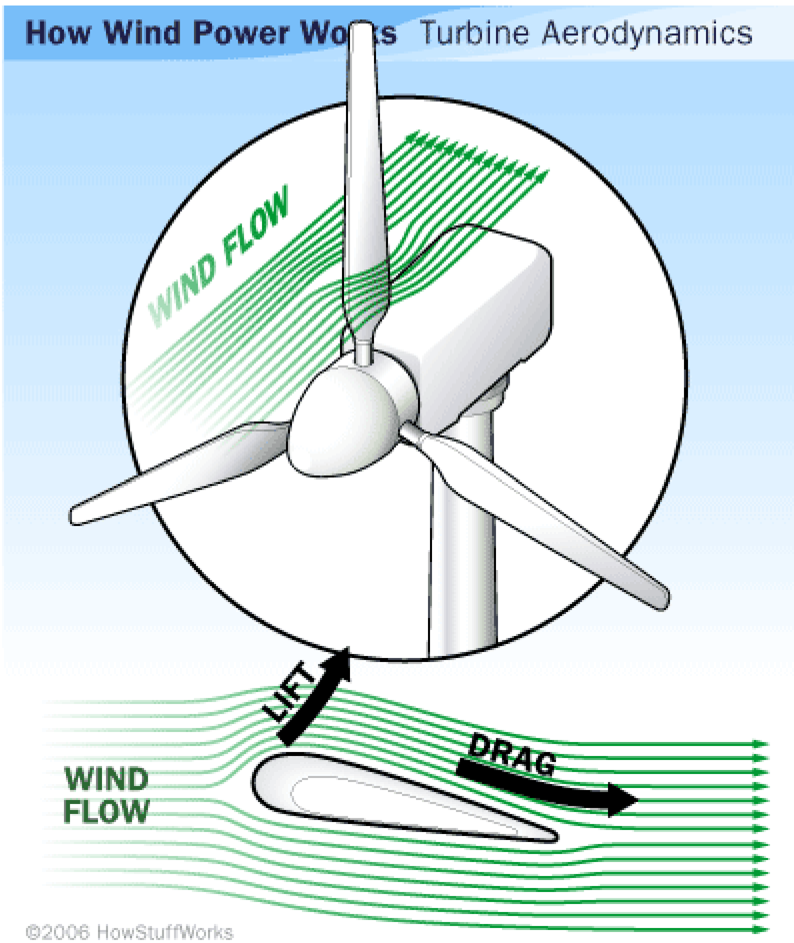
3.3.1 Prediction of flow around an airfoil

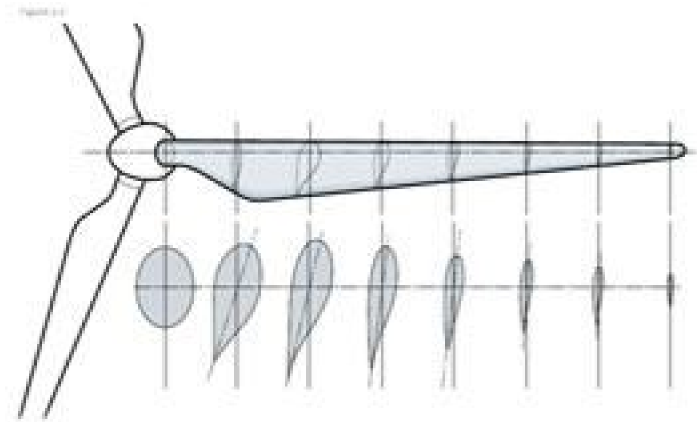
Applications of airfoils in **airplane wings**

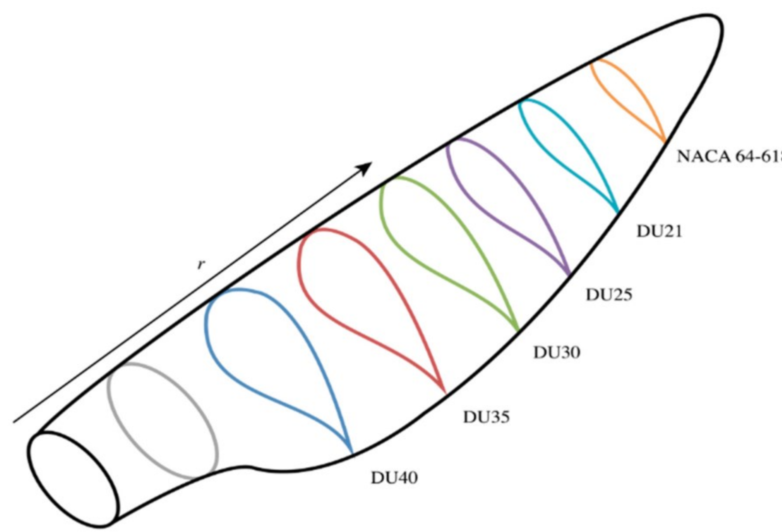


3.3.1 Prediction of flow around an airfoil

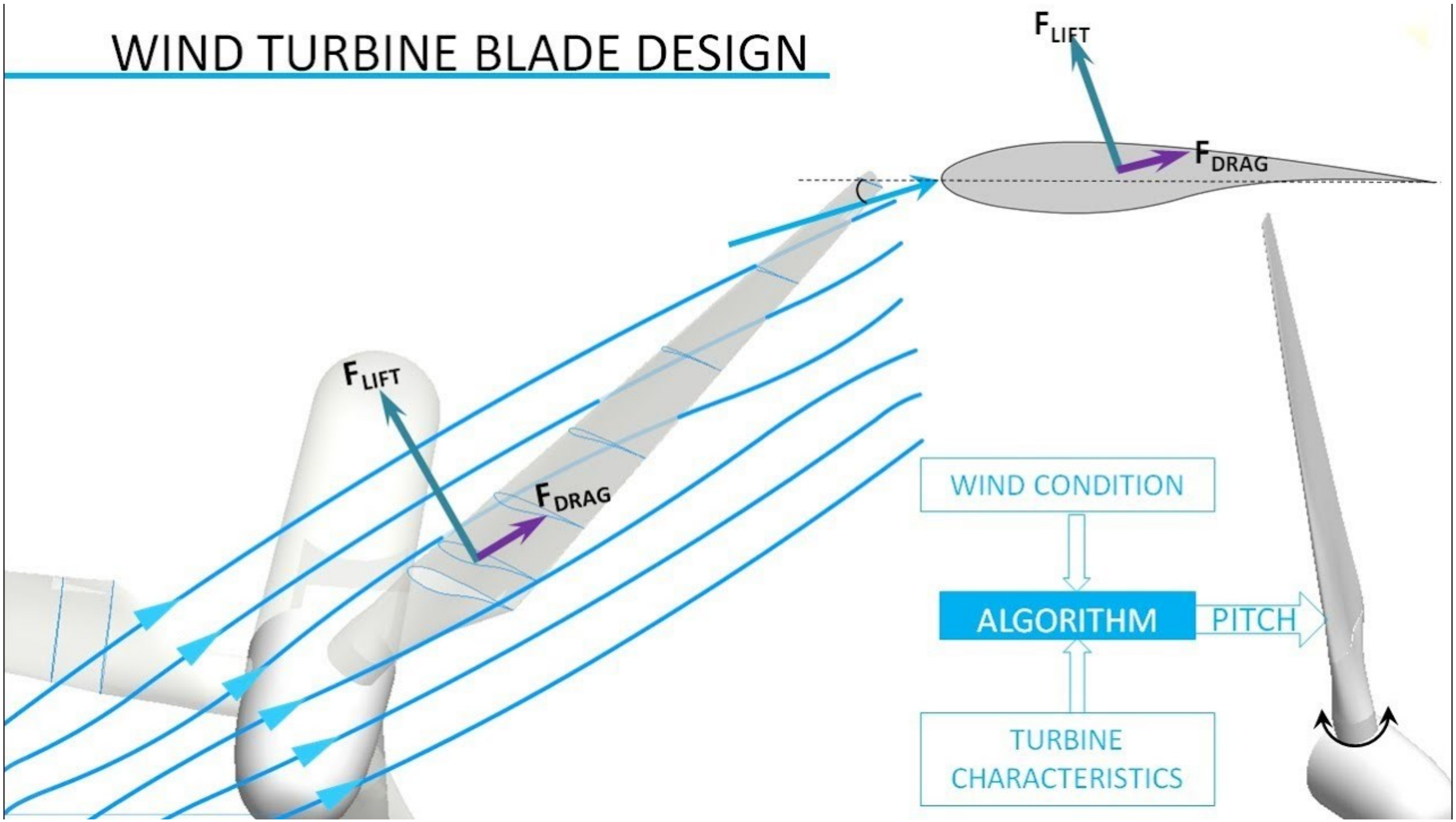
Applications of airfoils in **wind turbines**



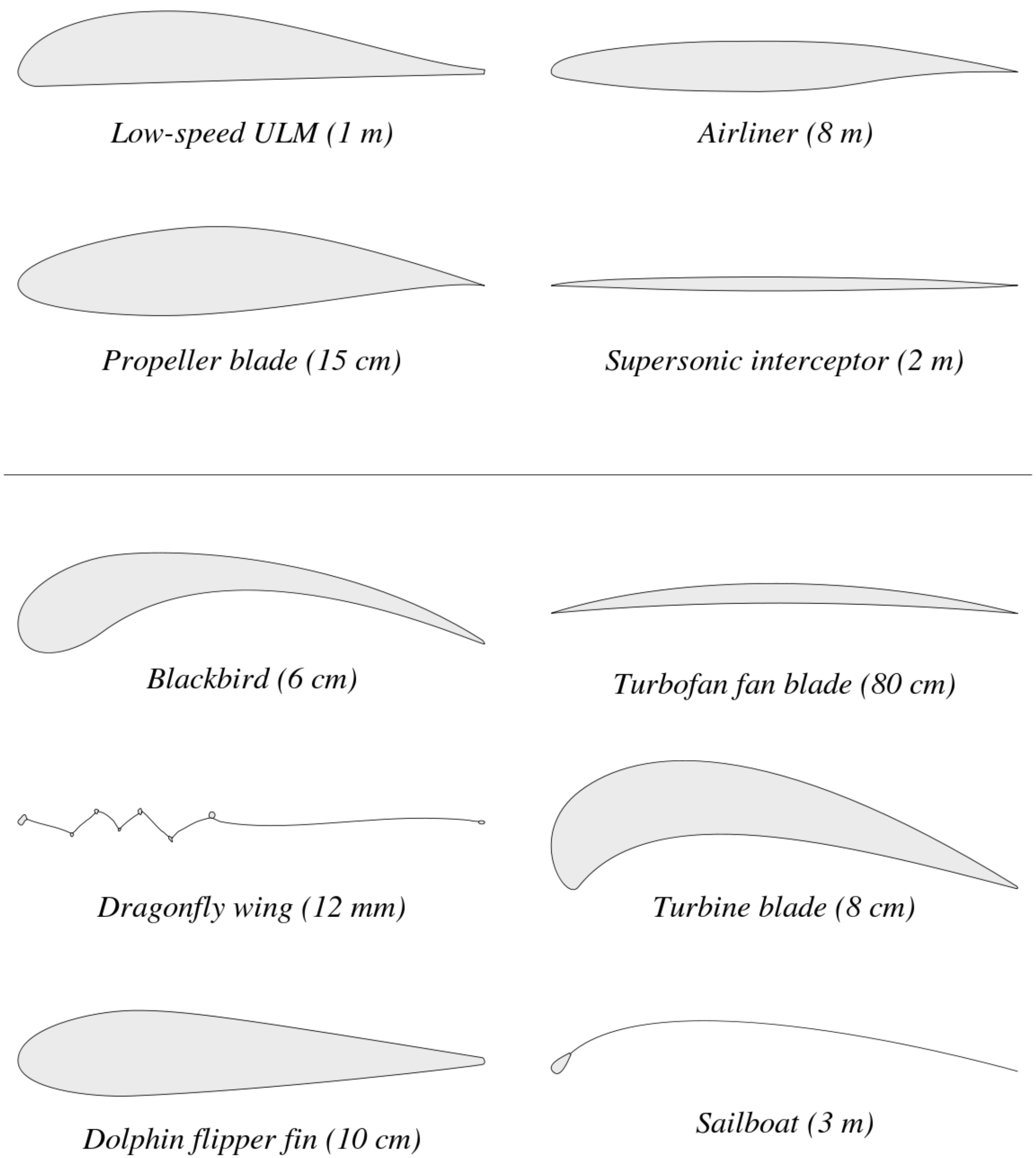




Applications of airfoils in **wind turbines**

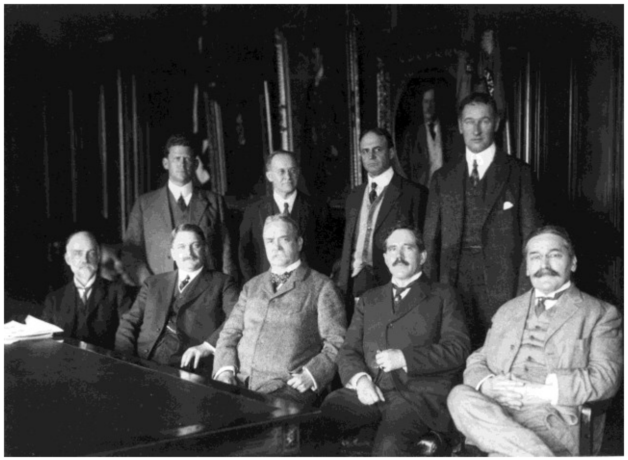


3.3.1 Prediction of flow around an airfoil



3.3 Prediction of flow around an airfoil

National **A**dvisory **C**ommittee for **A**eronautics (NACA)



The first meeting of the NACA in 1915

The NACA Test Force at the High Speed Flight Station in Edwards, California.

Formed: March 3, 1915

Dissolved: October 1, 1958

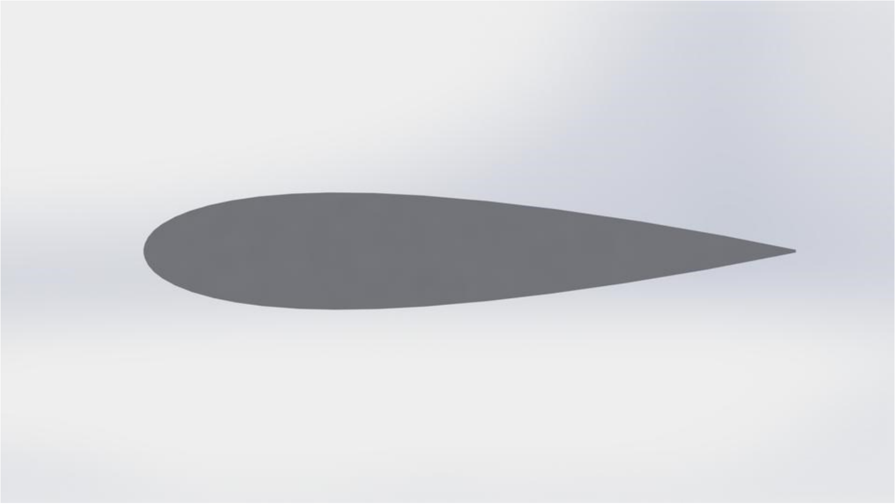
Superseding agency: NASA

3.3 Prediction of flow around an airfoil

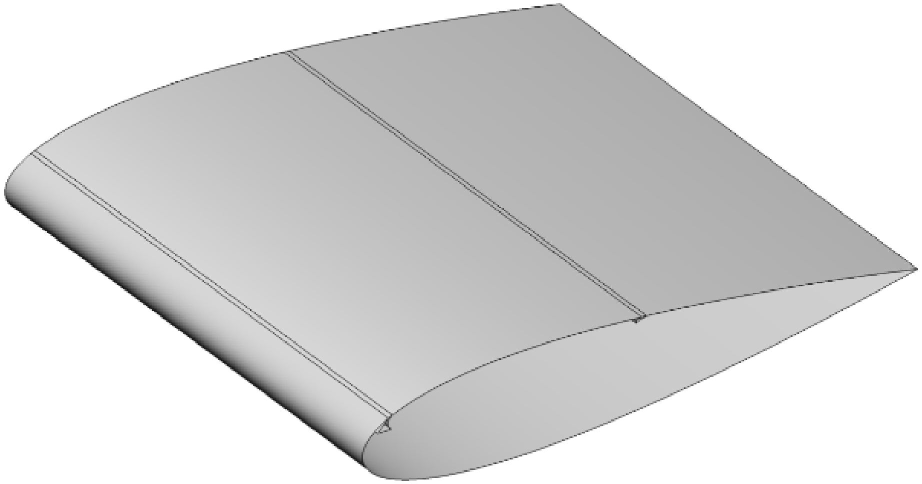
National **A**dvisory **C**ommittee for **A**eronautics (NACA)

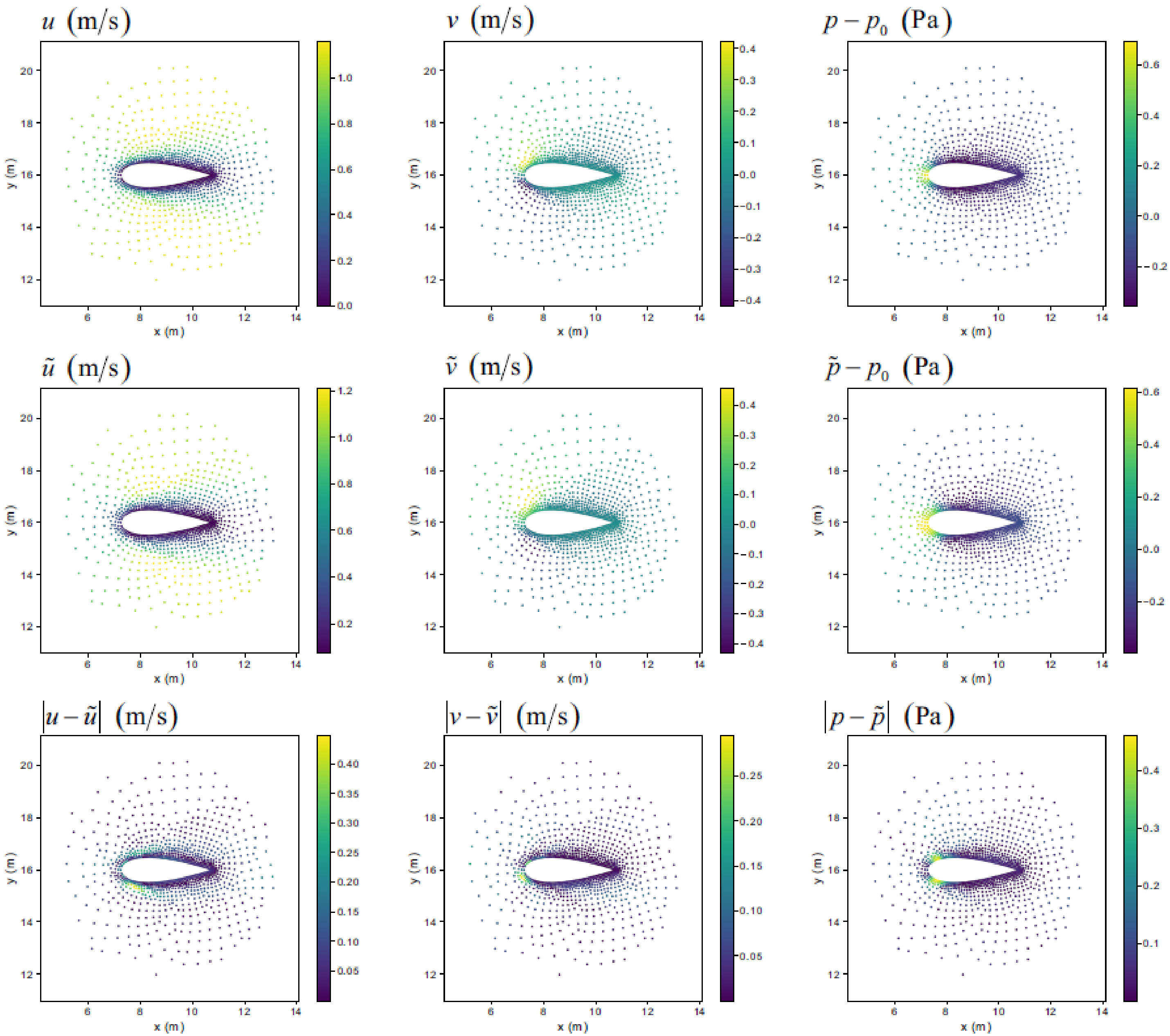
4 Digits NACA airfoils:
https://en.wikipedia.org/wiki/NACA_airfoil

NACA 0028

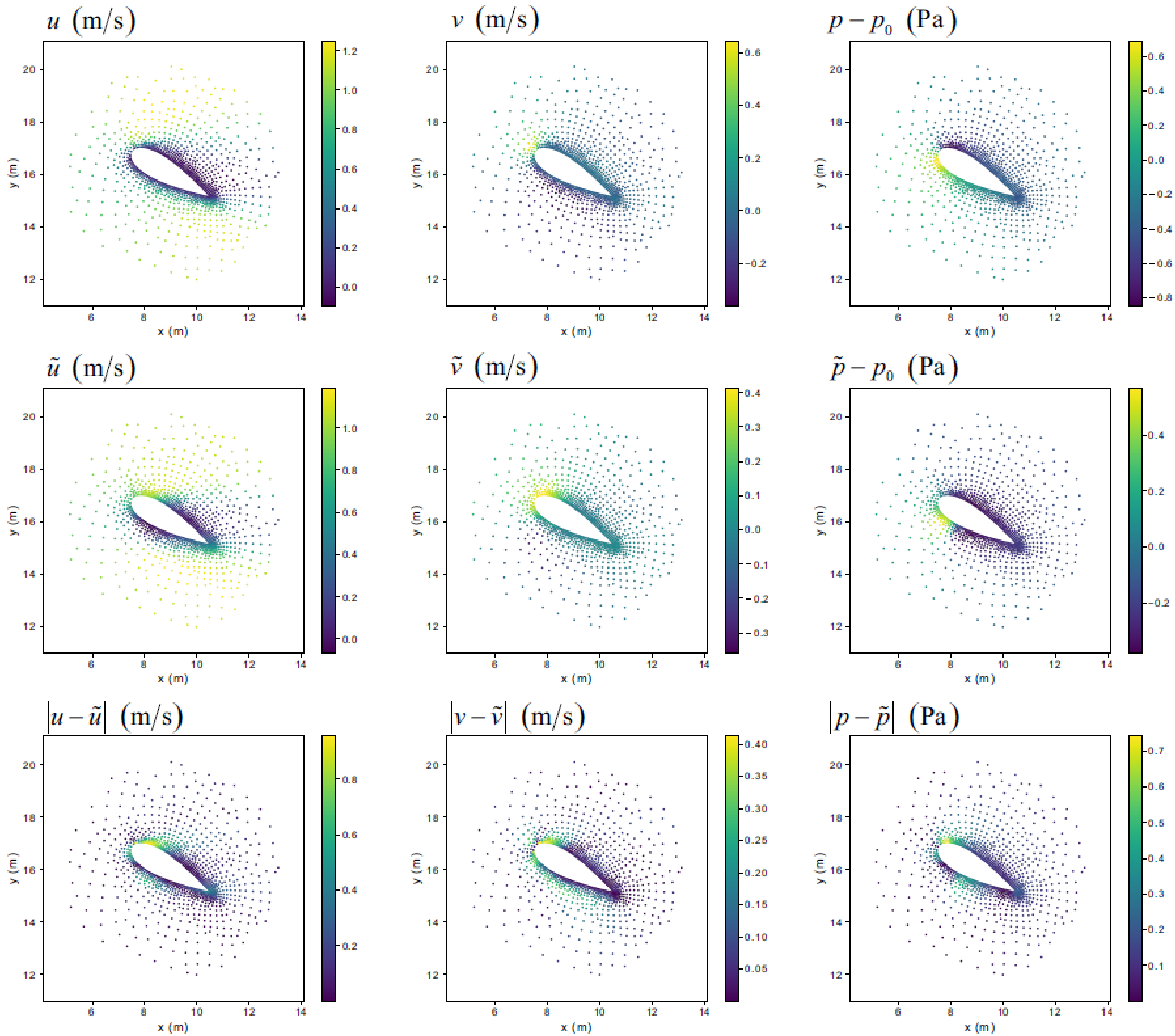


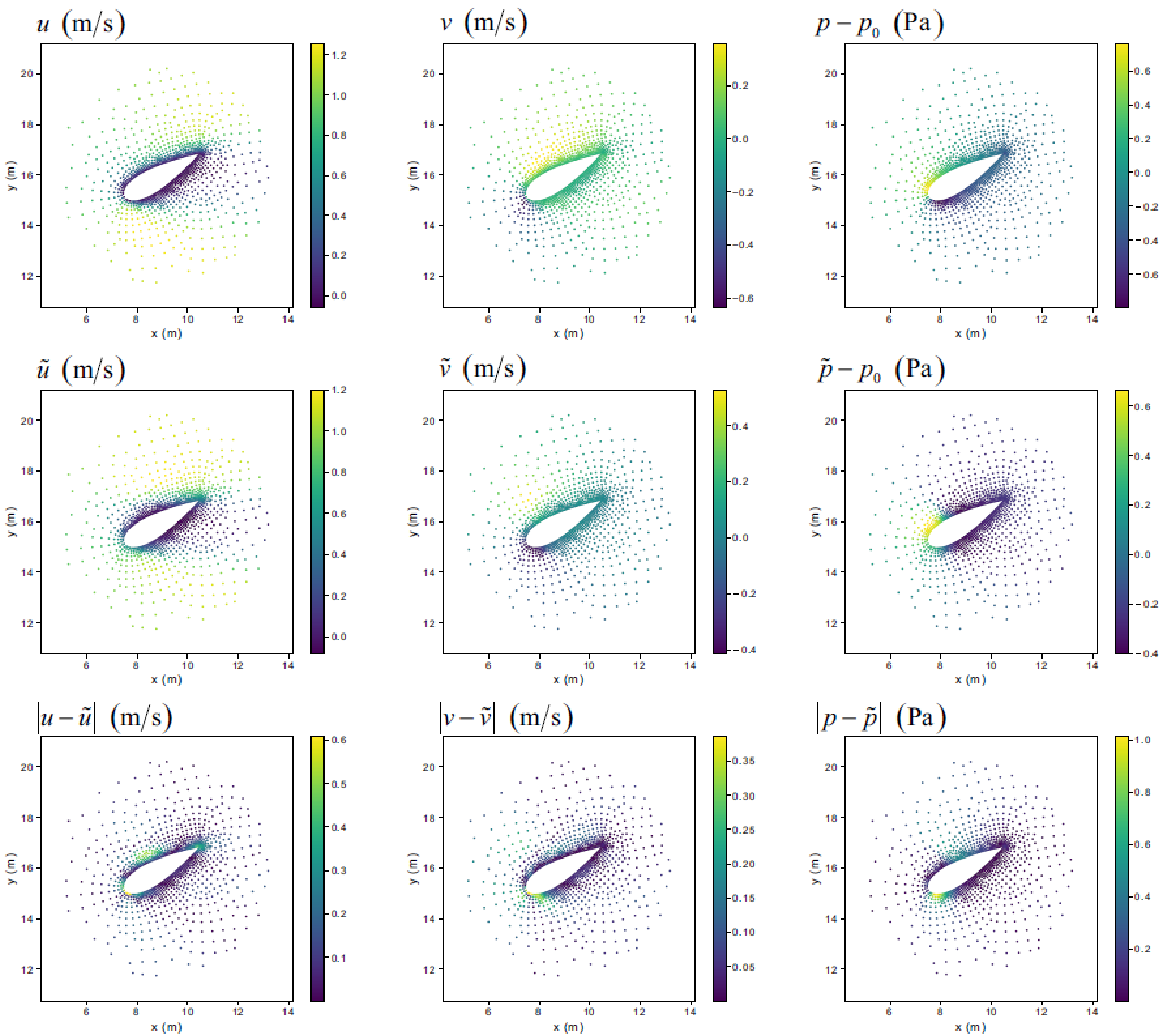
NACA 0028





u (m/s)
v (m/s)
$p-p_0$ (Pa)
$\tilde{u}$ (m/s)
$\tilde{v}$ (m/s)
$\tilde{p}-p_0$ (Pa)
$|u-\tilde{u}|$ (m/s)
$|v-\tilde{v}|$ (m/s)
$|p-\tilde{p}|$ (Pa)
x (m)
y (m)


u (m/s)
v (m/s)
$p - p_0$ (Pa)
$\tilde{u}$ (m/s)
$\tilde{v}$ (m/s)
$\tilde{p} - p_0$ (Pa)
$|u - \tilde{u}|$ (m/s)
$|v - \tilde{v}|$ (m/s)
$|p - \tilde{p}|$ (Pa)
x (m)
y (m)



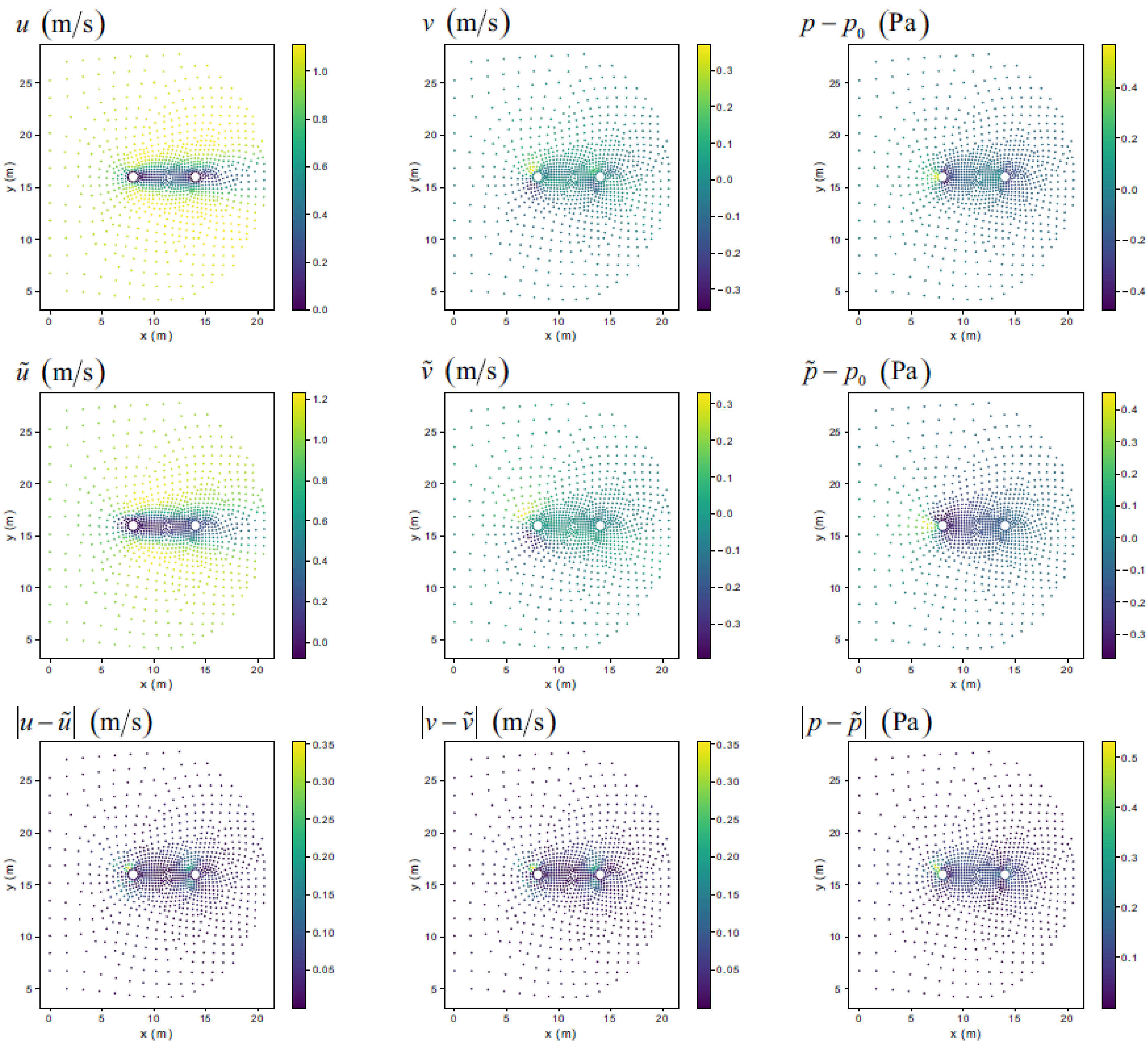
u (m/s)
v (m/s)
$p - p_0$ (Pa)
$\tilde{u}$ (m/s)
$\tilde{v}$ (m/s)
$\tilde{p} - p_0$ (Pa)
$|u - \tilde{u}|$ (m/s)
$|v - \tilde{v}|$ (m/s)
$|p - \tilde{p}|$ (Pa)
x (m)
y (m)

3.3.1 Prediction of flow around an airfoil

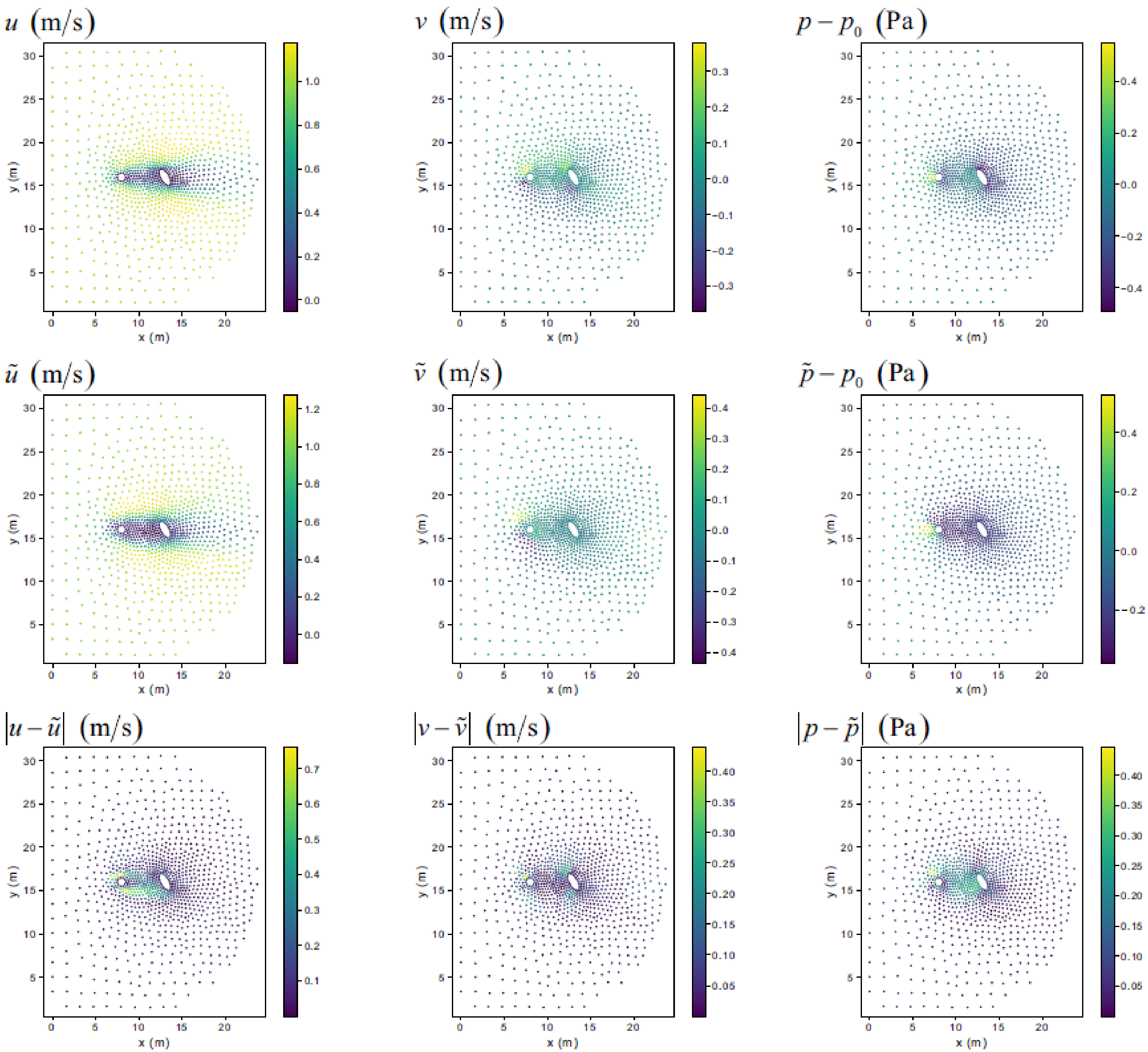
TABLE V. Error analysis of the velocity and pressure fields predicted by our neural network for NACA 0028; $||\ldots||$ indicates the L^2 norm.

Angle of attack	$\|\|u-\tilde{u}\|\|$	$\|\|v-\tilde{v}\|\|$	$\|\|p-\tilde{p}\|\|$
0°	1.05631E−1	5.47357E−2	1.04842E−1
30°	2.83492E−1	1.30977E−1	2.07335E−1
-30°	1.62185E−1	1.00821E−1	2.42353E−1

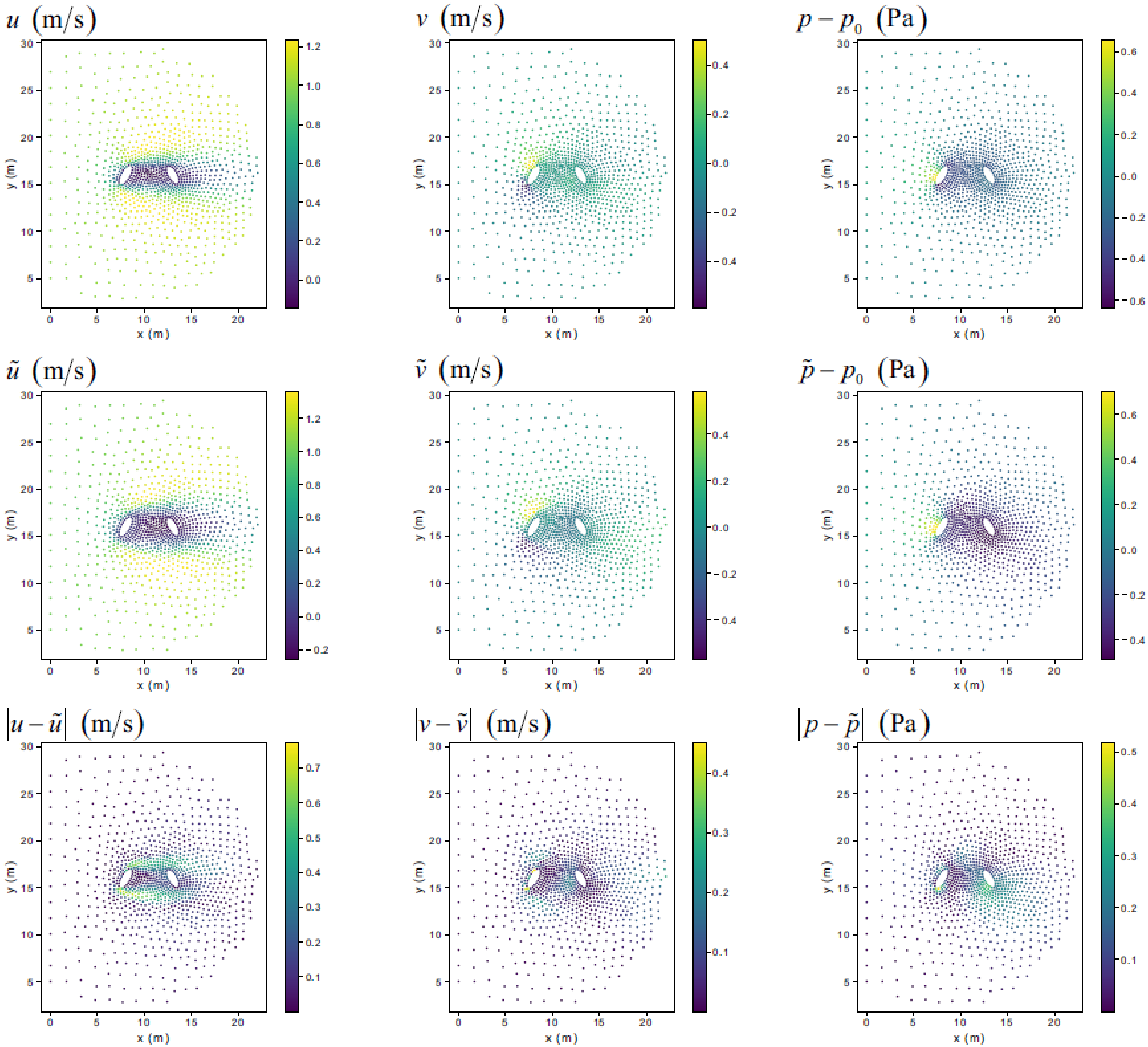
3.3.2 Prediction of flow around multiple objects



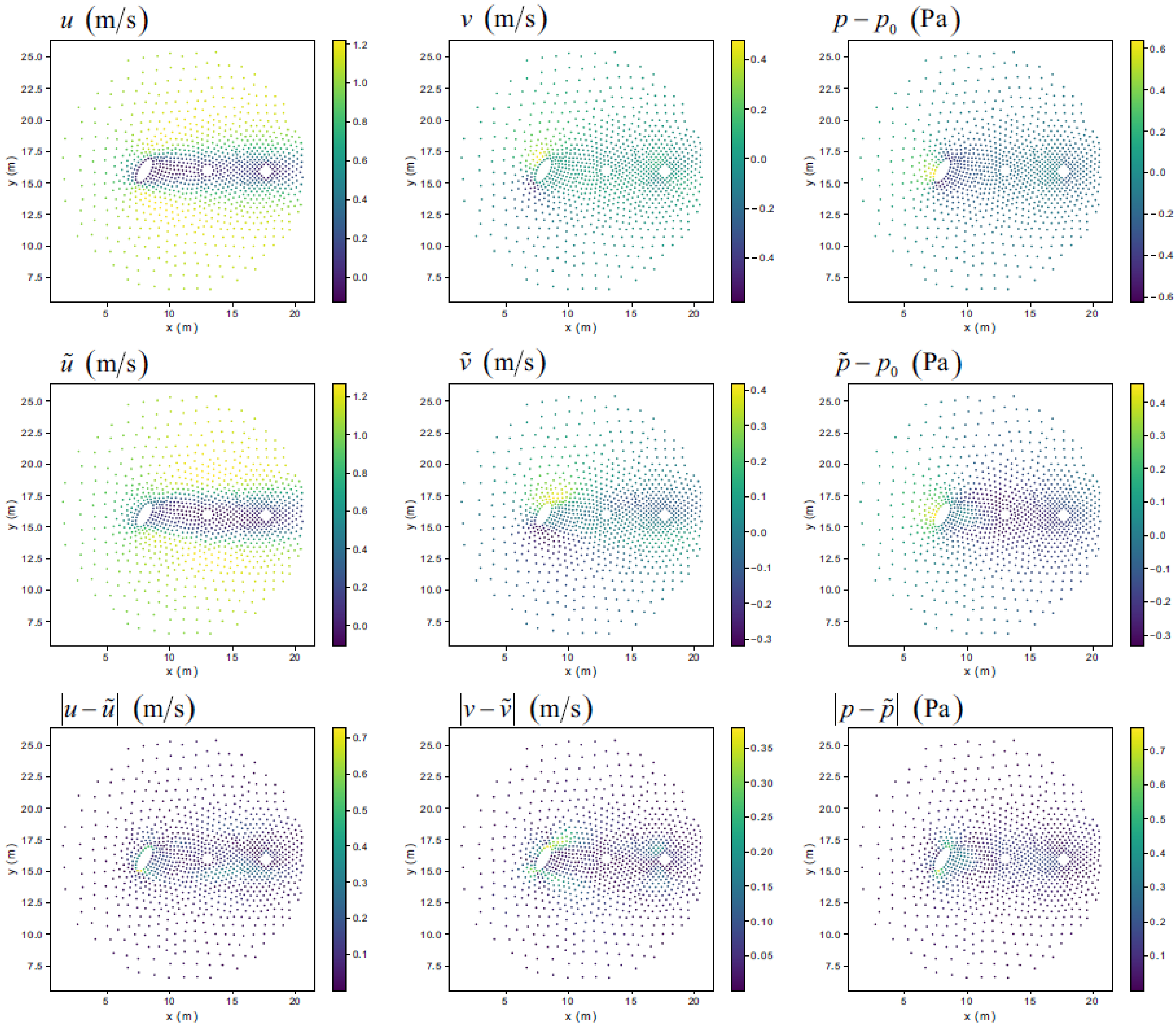
u (m/s)
v (m/s)
$p-p_0$ (Pa)
$\tilde{u}$ (m/s)
$\tilde{v}$ (m/s)
$\tilde{p}-p_0$ (Pa)
$|u-\tilde{u}|$ (m/s)
$|v-\tilde{v}|$ (m/s)
$|p-\tilde{p}|$ (Pa)
x (m)
y (m)



u (m/s)
v (m/s)
p − p0 (Pa)
ũ (m/s)
ṽ (m/s)
p̃ − p0 (Pa)
|u − ũ| (m/s)
|v − ṽ| (m/s)
|p − p̃| (Pa)
x (m)
y (m)



u (m/s)
v (m/s)
p − p0 (Pa)
ũ (m/s)
ṽ (m/s)
p̃ − p0 (Pa)
|u − ũ| (m/s)
|v − ṽ| (m/s)
|p − p̃| (Pa)
x (m)
y (m)


u (m/s)
v (m/s)
$p - p_0$ (Pa)
$\tilde{u}$ (m/s)
$\tilde{v}$ (m/s)
$\tilde{p} - p_0$ (Pa)
$|u - \tilde{u}|$ (m/s)
$|v - \tilde{v}|$ (m/s)
$|p - \tilde{p}|$ (Pa)
x (m)
y (m)

3.4 Prediction of flow around multiple objects

TABLE IV. Error analysis of the velocity and pressure fields predicted by our neural network for multiple bodies; $||\ldots||$ indicates the L^2 norm.

	$\|\|u-\tilde{u}\|\|$	$\|\|v-\tilde{v}\|\|$	$\|\|p-\tilde{p}\|\|$
Two circular cylinders (Fig. 12)	1.51130E−1	5.80172E−2	9.22659E−2
Circular and elliptical cylinders (Fig. 13)	1.63664E−1	7.50613E−2	1.09544E−1
Two elliptical cylinders (Fig. 14)	2.19626E−1	8.93397E−2	1.35396E−1
Elliptical, circular, and rectangular cylinders (Fig. 15)	1.30335E−1	8.59303E−2	1.33239E−1

$\|\|u-\tilde{u}\|\|$	$\|\|v-\tilde{v}\|\|$	$\|\|p-\tilde{p}\|\|$
1.51130E−1	5.80172E−2	9.22659E−2
1.63664E−1	7.50613E−2	1.09544E−1
2.19626E−1	8.93397E−2	1.35396E−1
1.30335E−1	8.59303E−2	1.33239E−1

4 Conclusions and future directions

Future directions:

1) Unsteady CFD problems due to moving objects and mesh deformation

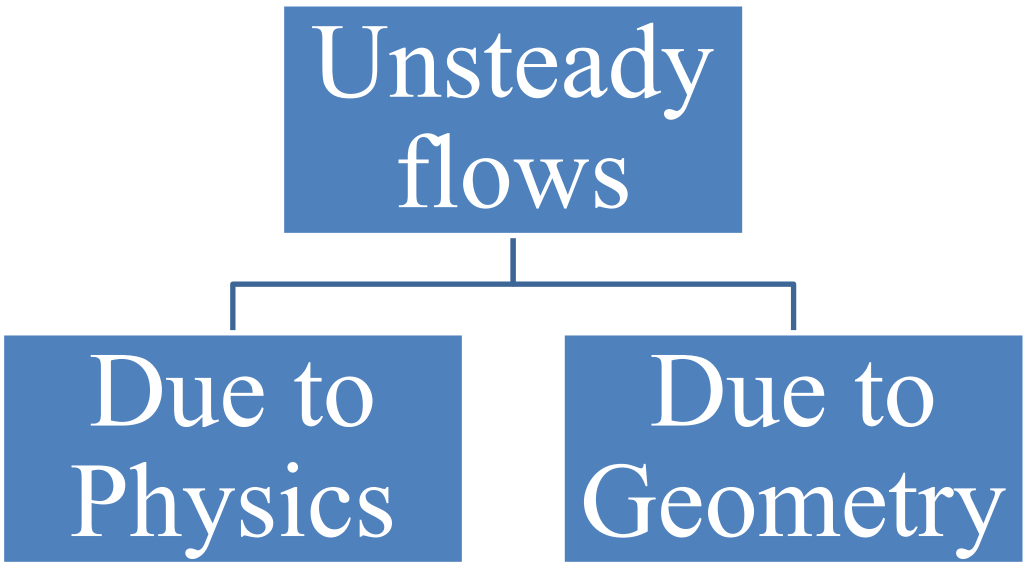
2) Unsupervised learning of CFD problems in variable geometries

3) Multidisciplinary design optimization

Unsteady Flows with Moving Objects and Boundaries

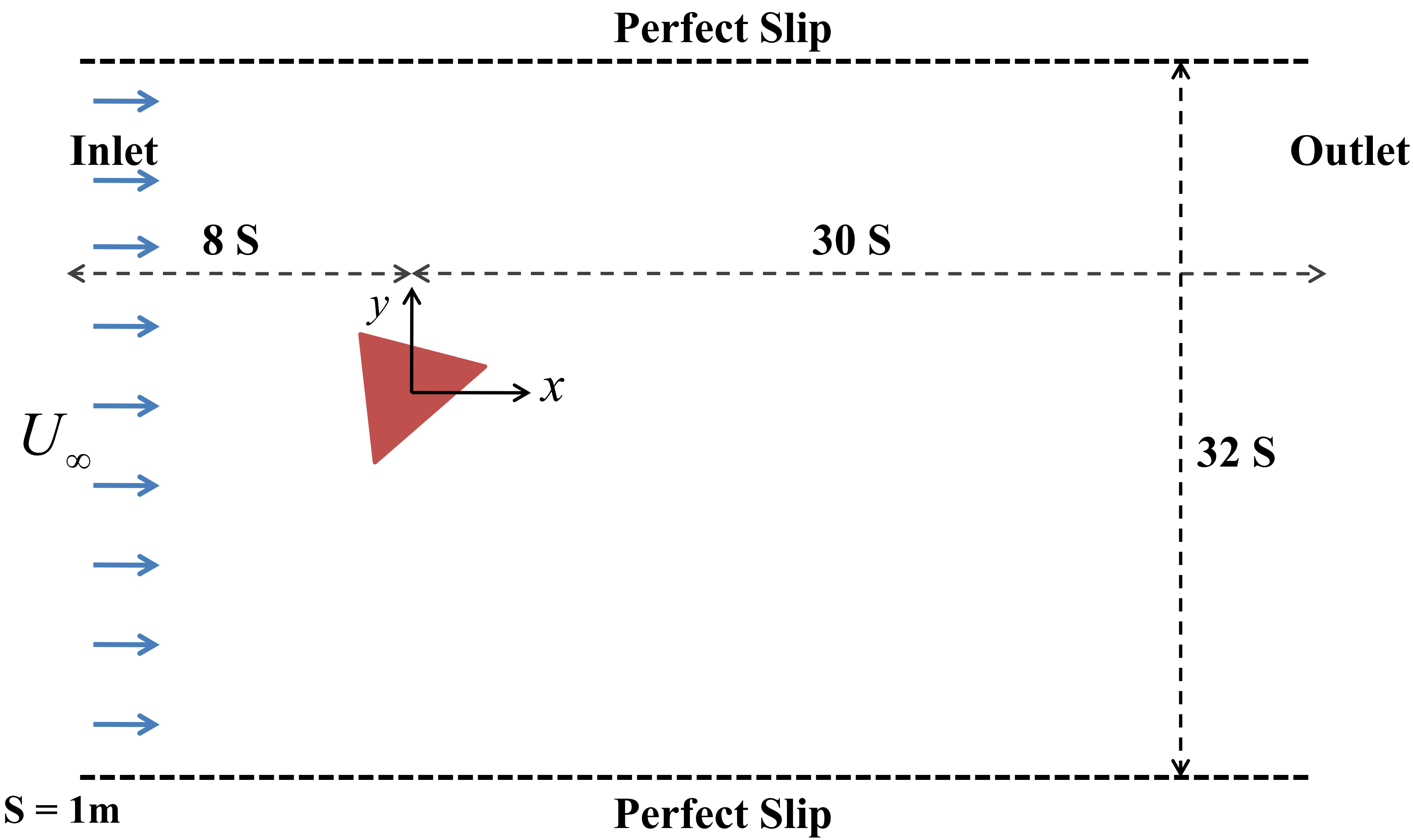
Future directions:

1) Unsteady CFD problems



2. 1 Governing equations of fluid dynamics

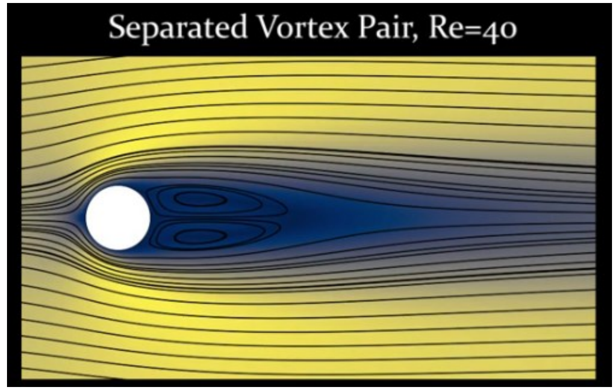
Unsteady flows due to the physics but with a fixed geometry



4 Conclusions and future directions

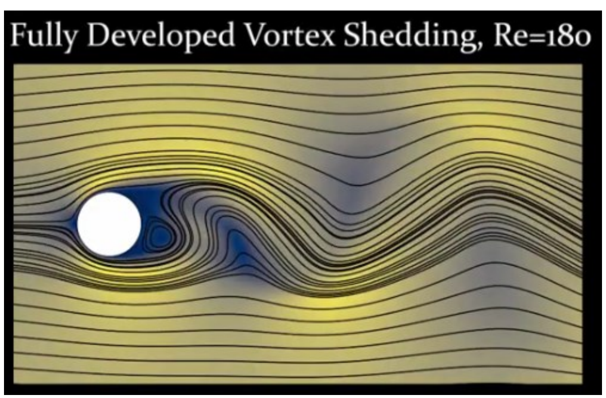
Unsteady flows due to the physics but with a fixed geometry

$$Re = \frac{\rho L u_\infty}{\mu}.$$

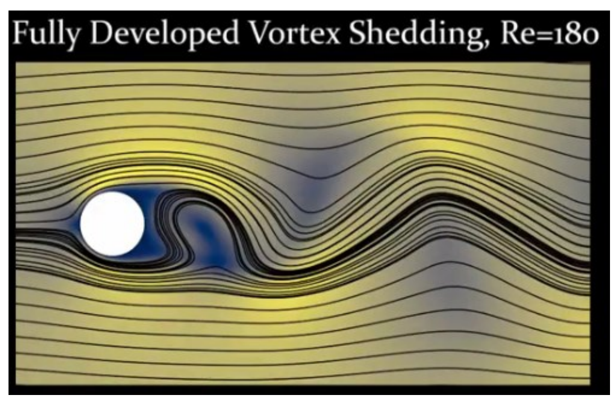


Steady-State

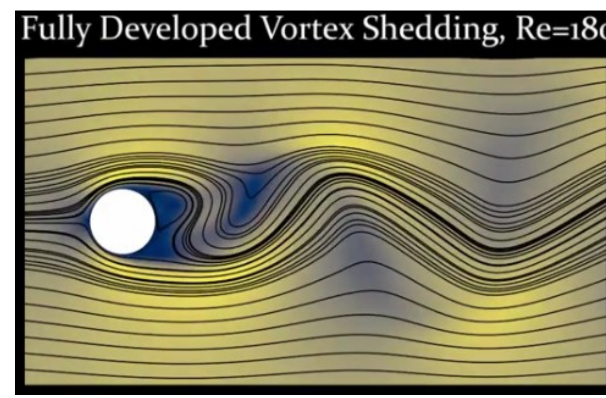
t_1



t_2



t_3

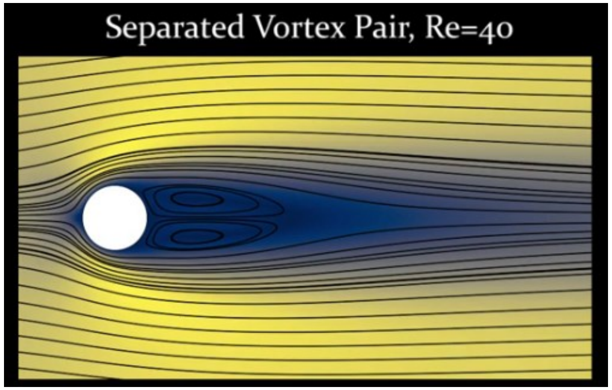


Unsteady

4 Conclusions and future directions

Unsteady flows due to the physics but with a fixed geometry

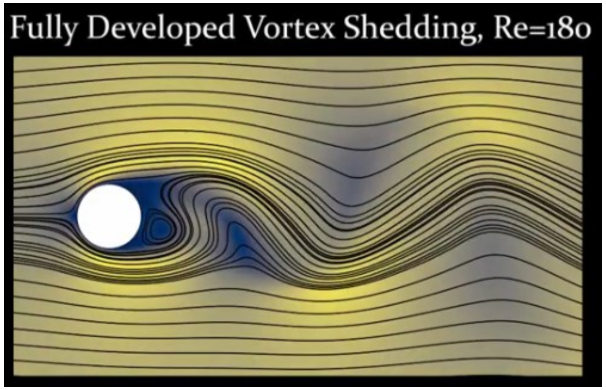
$$Re = \frac{\rho L u_\infty}{\mu}.$$



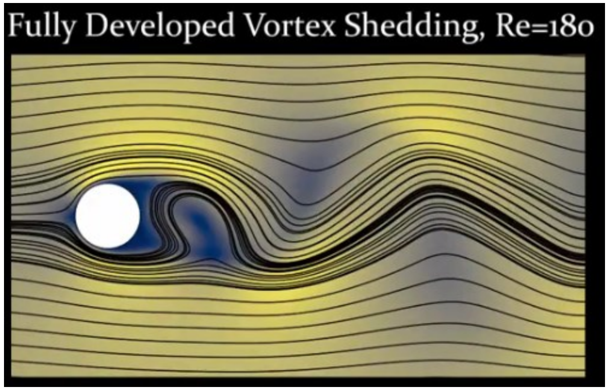
Steady-State

But we are not interested in these types of problems, because the geometry is fixed in time!

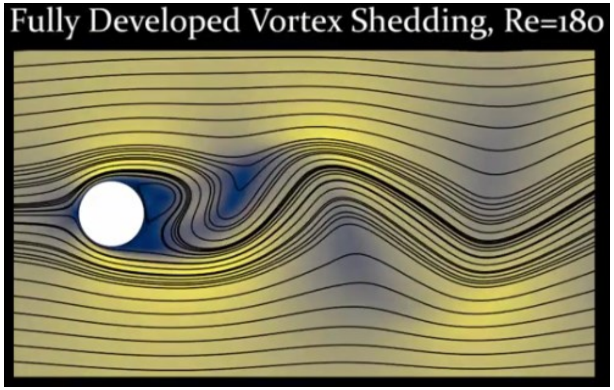
t_1



t_2



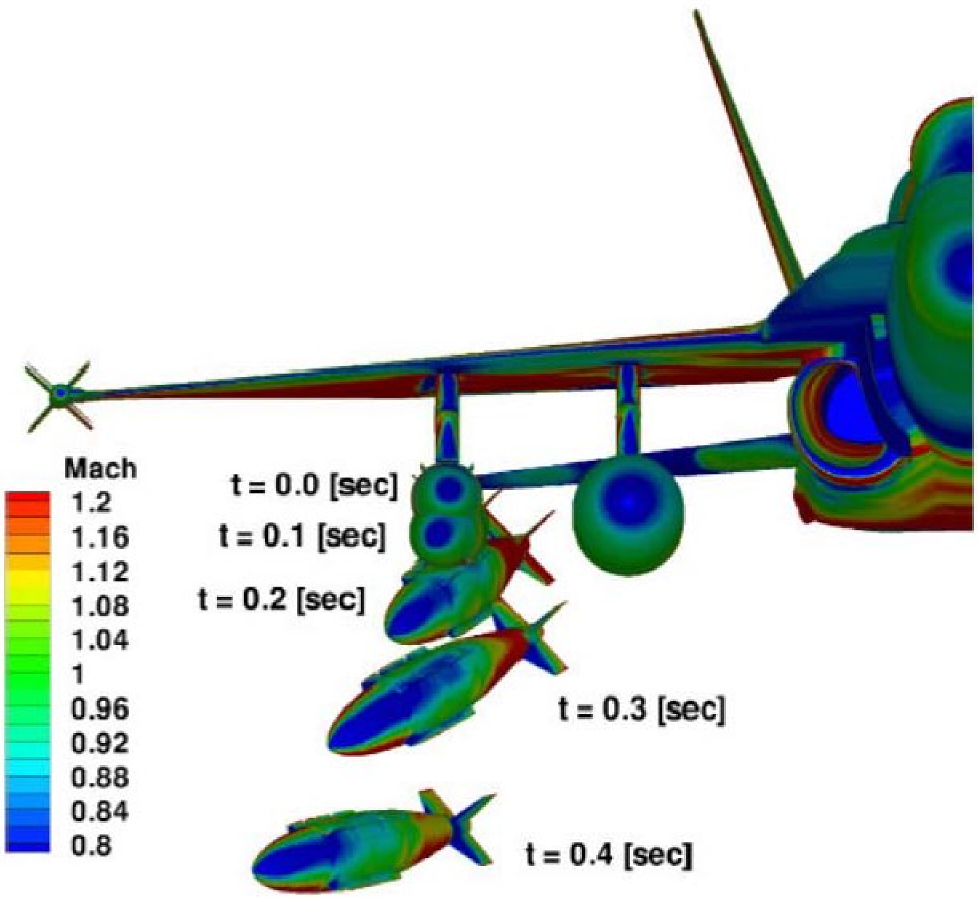
t_3



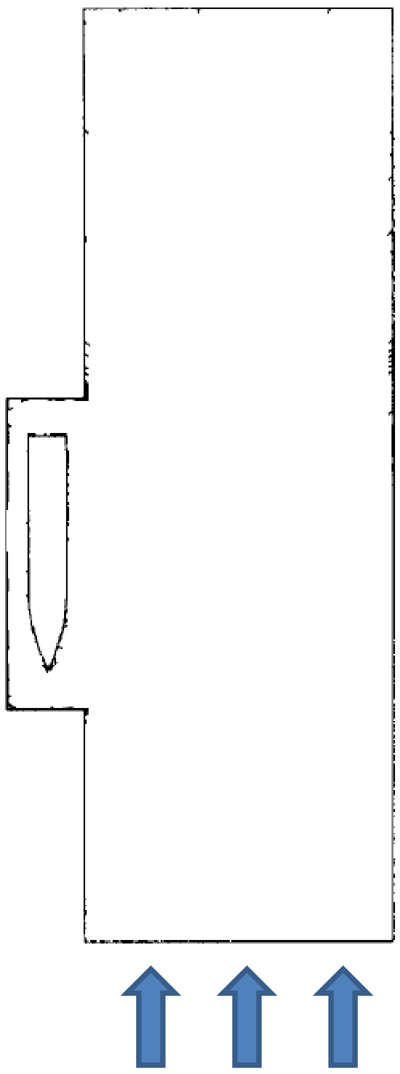
Unsteady

4 Conclusions and future directions

Unsteady flows due to the inconsistent geometry



Unsteady flows due to the inconsistent geometry



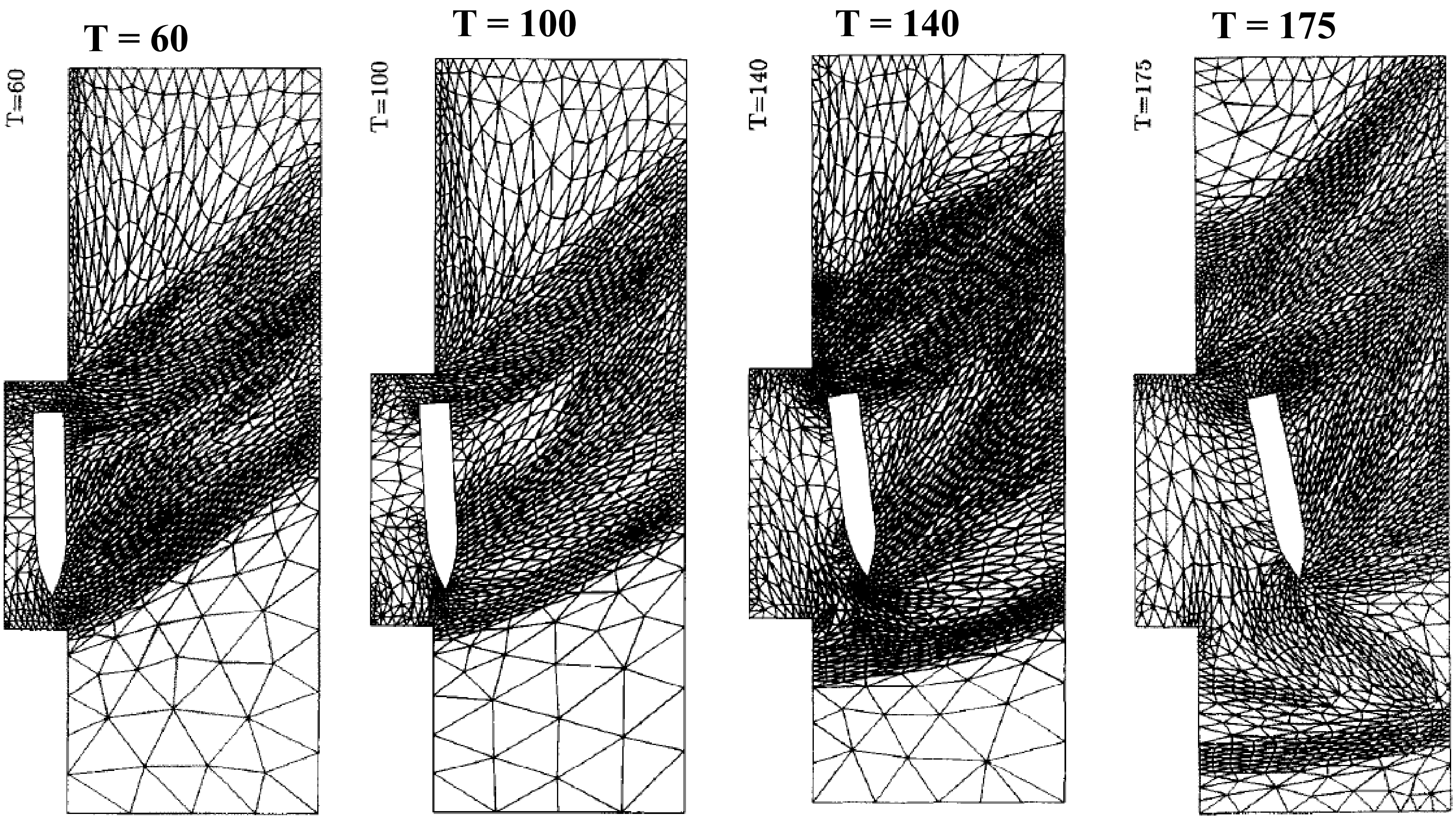
A warplane leaves an object in sky.

What is the position of the object at time T?

And what is the flow field (velocity, density, and pressure) around the object at time T?

Supersonic Flow (compressible)

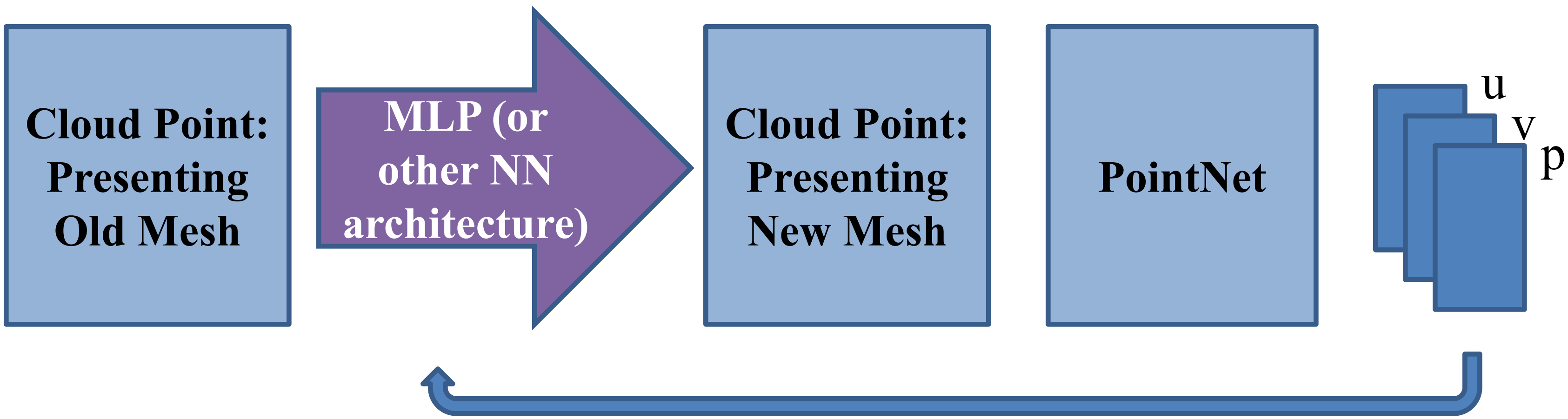
Unsteady flows due to the inconsistent geometry



4 Conclusions and future directions

Unsteady flows due to the inconsistent geometry

Using LSTM (at each time step)

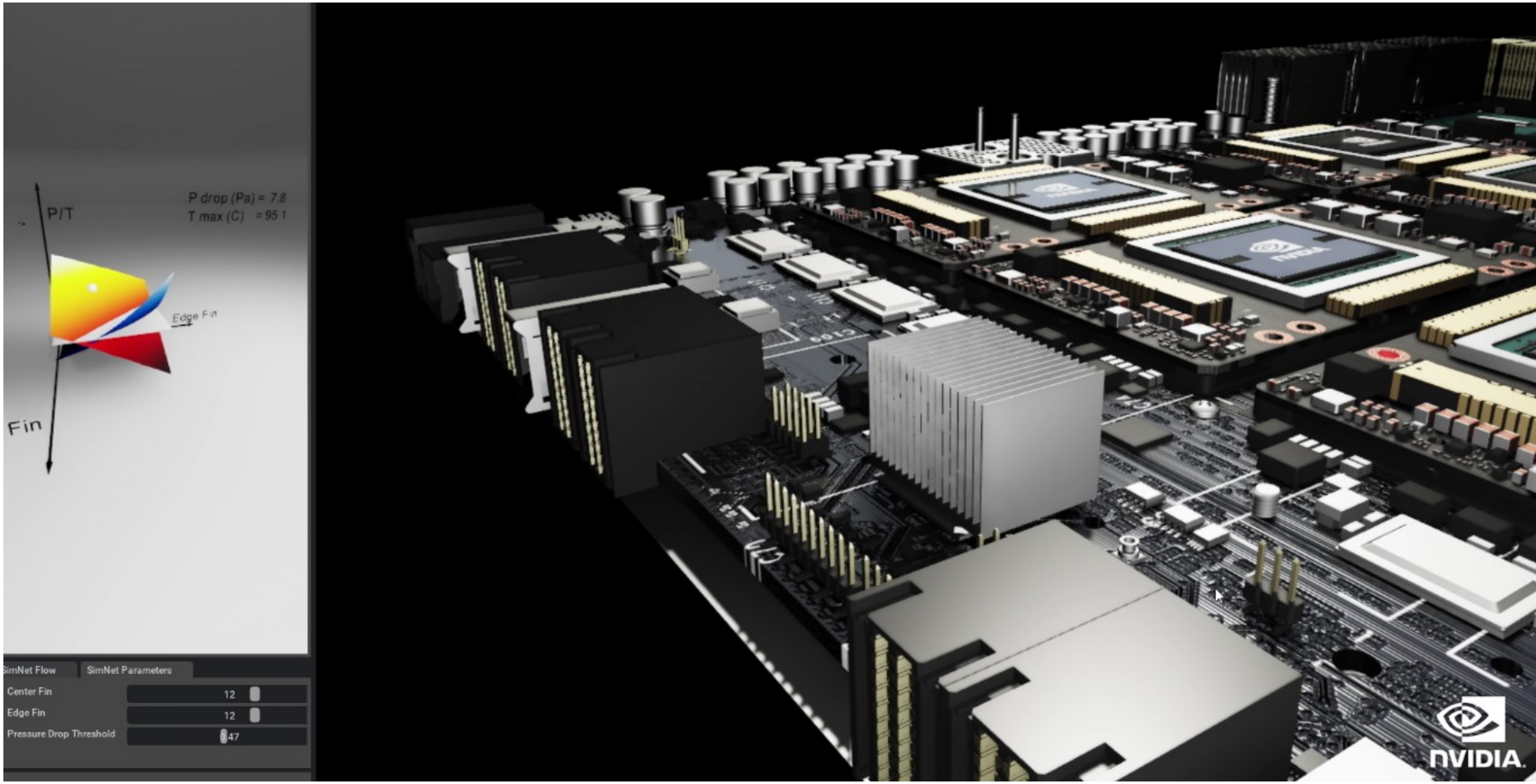


$$\sum \tilde{F} = m\tilde{a}$$

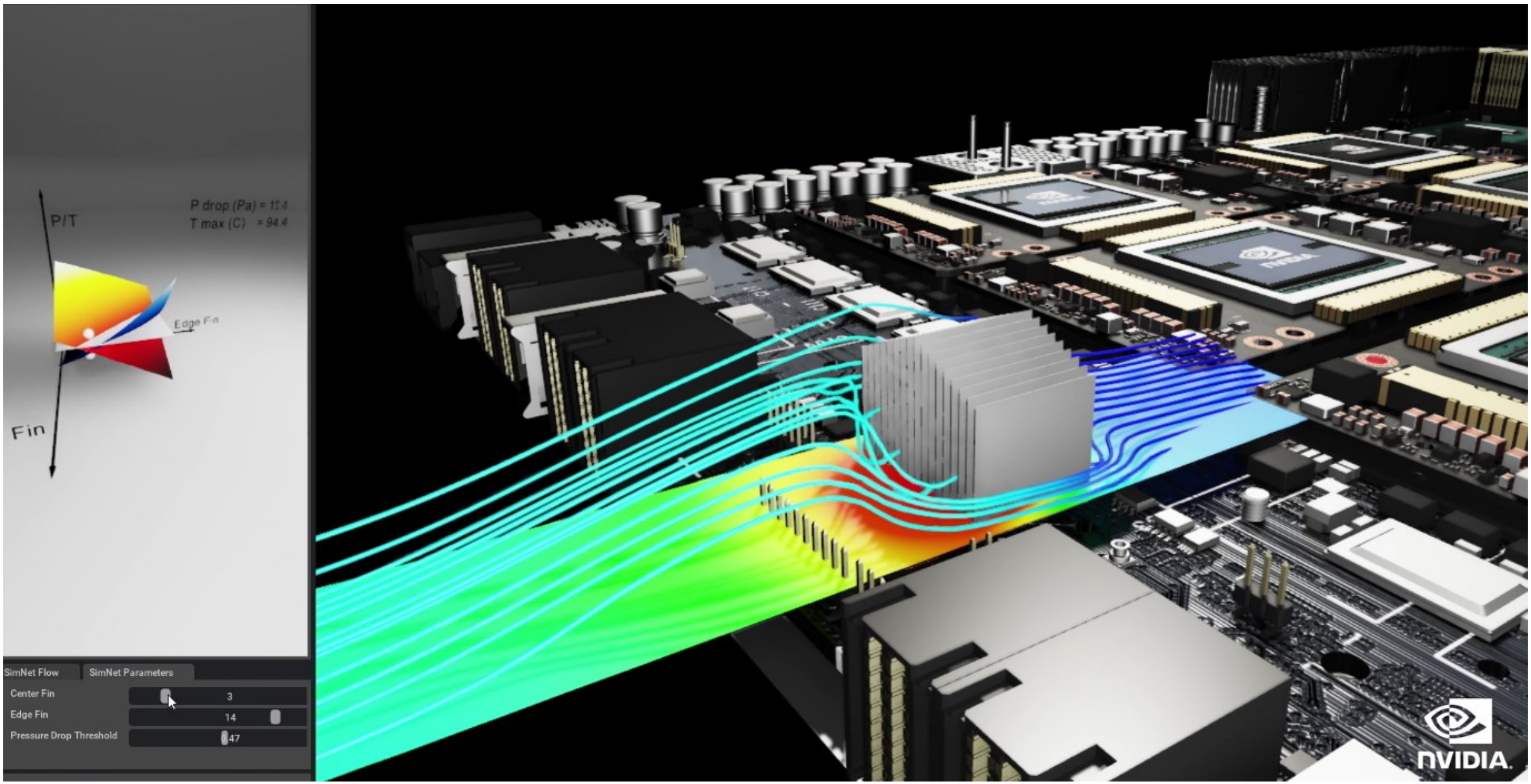
$$\sum \tilde{M} = I\tilde{\alpha}$$

Physics Informed Neural Network: An-unsupervised DL approach

4 Conclusions and future directions



4 Conclusions and future directions



4 Conclusions and future directions

The idea of Physics Informed Neural Network (PINN) first introduced at Brown University and has become popular recently. Here are a few papers discussing this approach:

- **Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations**
https://www.sciencedirect.com/science/article/pii/S0021999118307125

- **NSFnets (Navier-Stokes Flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations**
https://arxiv.org/abs/2003.06496

- **Physics-informed neural networks for high-speed flows**
https://www.sciencedirect.com/science/article/pii/S0045782519306814

4 Conclusions and future directions

To say the idea of PINN at a high level, we enforce the loss function to minimize the residual of governing equations. Specifically, let's see how it works for continuity and Navier-Stokes equations.

Continuity & Navier-Stokes Equations:

$$\left.\begin{aligned} u_t + uu_x + vu_y &= -p_x + Re^{-1}(u_{xx} + u_{yy}), \\ v_t + uv_x + vv_y &= -p_y + Re^{-1}(v_{xx} + v_{yy}) - \eta_{tt}, \\ u_x + v_y &= 0. \end{aligned}\right\}$$

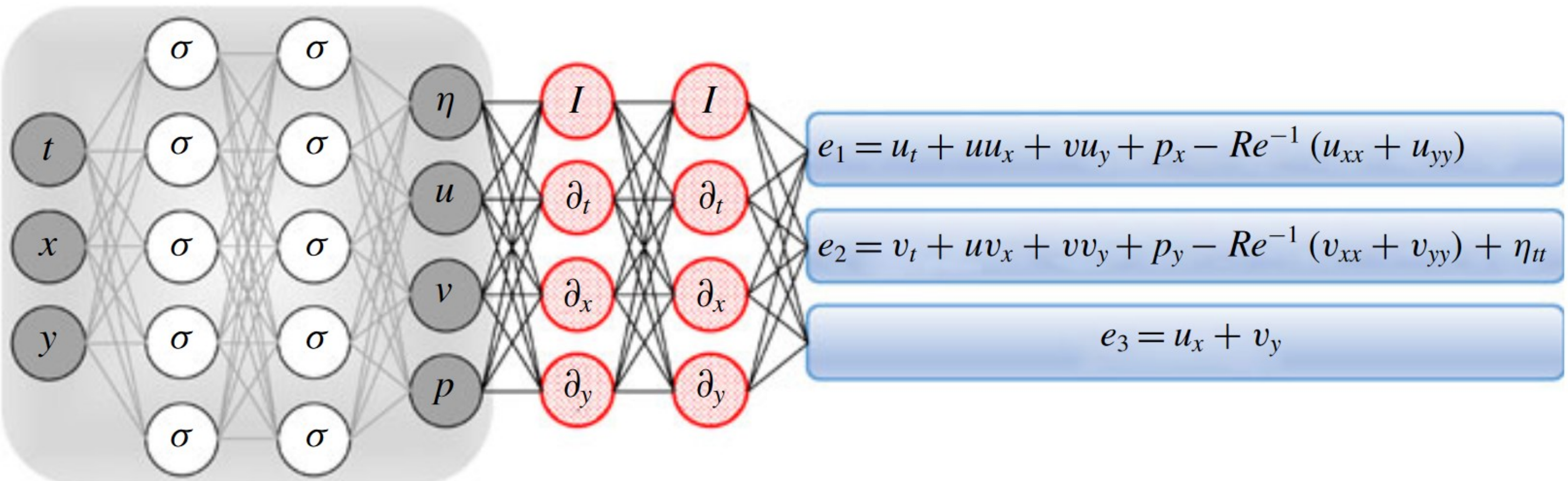
Residuals:

$$\left.\begin{aligned} e_1 &:= u_t + uu_x + vu_y + p_x - Re^{-1}(u_{xx} + u_{yy}), \\ e_2 &:= v_t + uv_x + vv_y + p_y - Re^{-1}(v_{xx} + v_{yy}) + \eta_{tt}, \\ e_3 &:= u_x + v_y. \end{aligned}\right\}$$

4 Conclusions and future directions

Loss function: $$\sum_{i=1}^{3}\sum_{n=1}^{N}(|e_i(t^n, x^n, y^n)|^2)$$

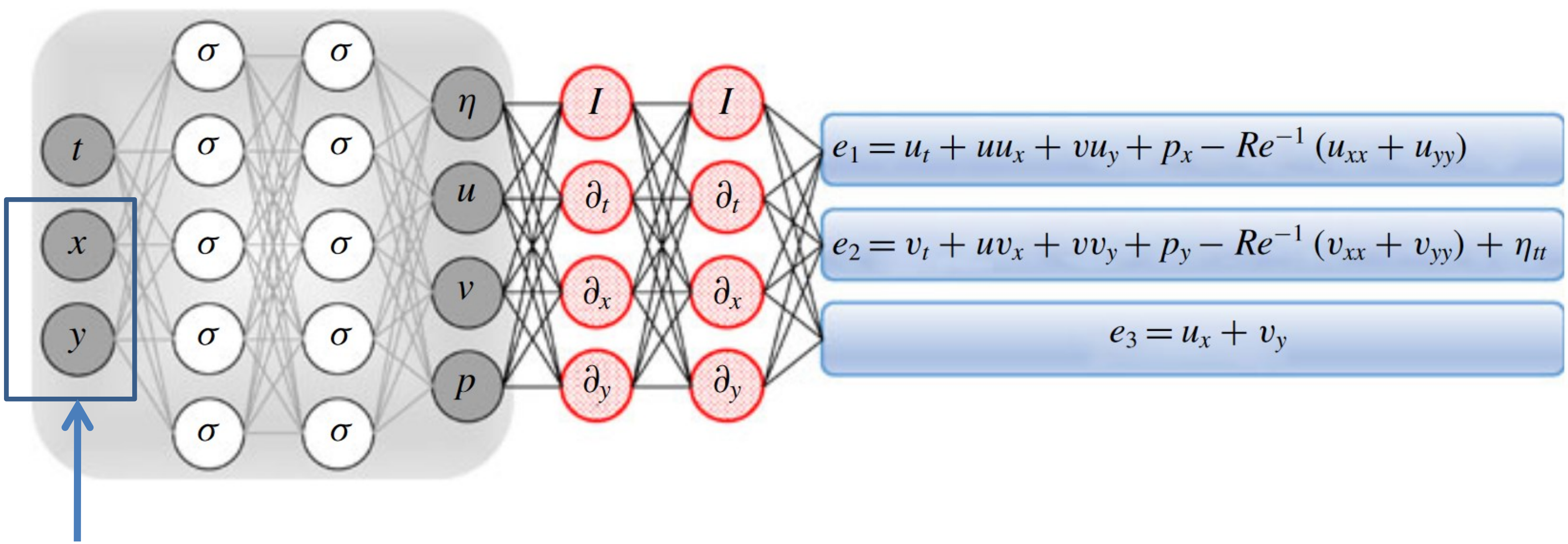
Neural Network for PINN



All figures in this section are taken from Raissi et. al. (2018) and Mao et. al. (2020)

4 Conclusions and future directions

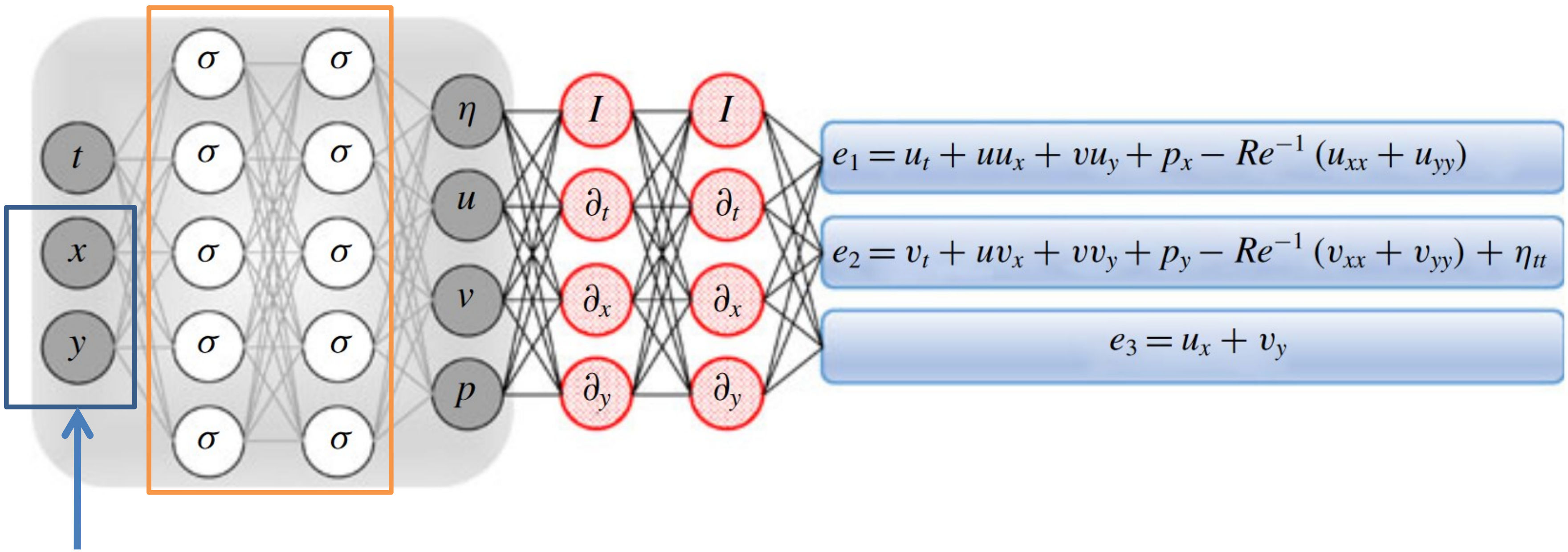
None of the current PINN can capture variable geometries! They have designed for the fixed geometries!



Connection to Point Cloud and PointNet

4 Conclusions and future directions

Replace the MLP by our CFD PointNet

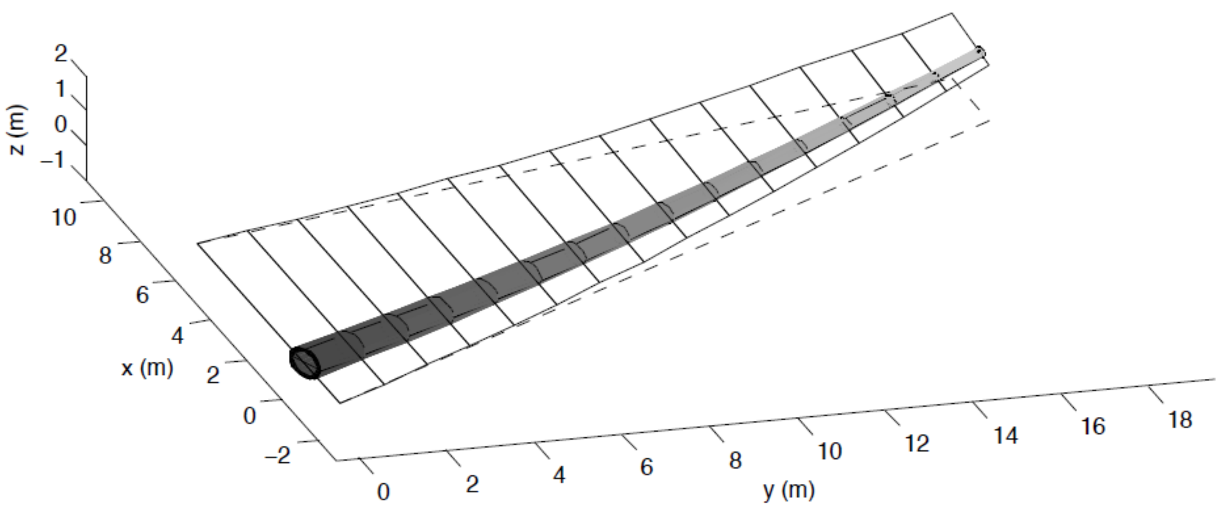


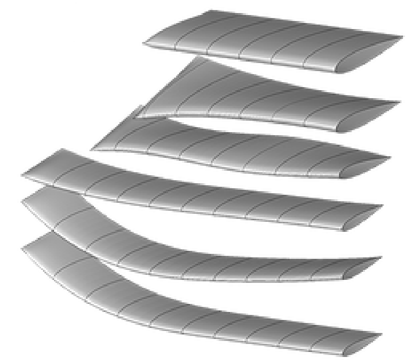
Connection to Point Cloud and PointNet

4 Conclusions and future directions

Future directions:

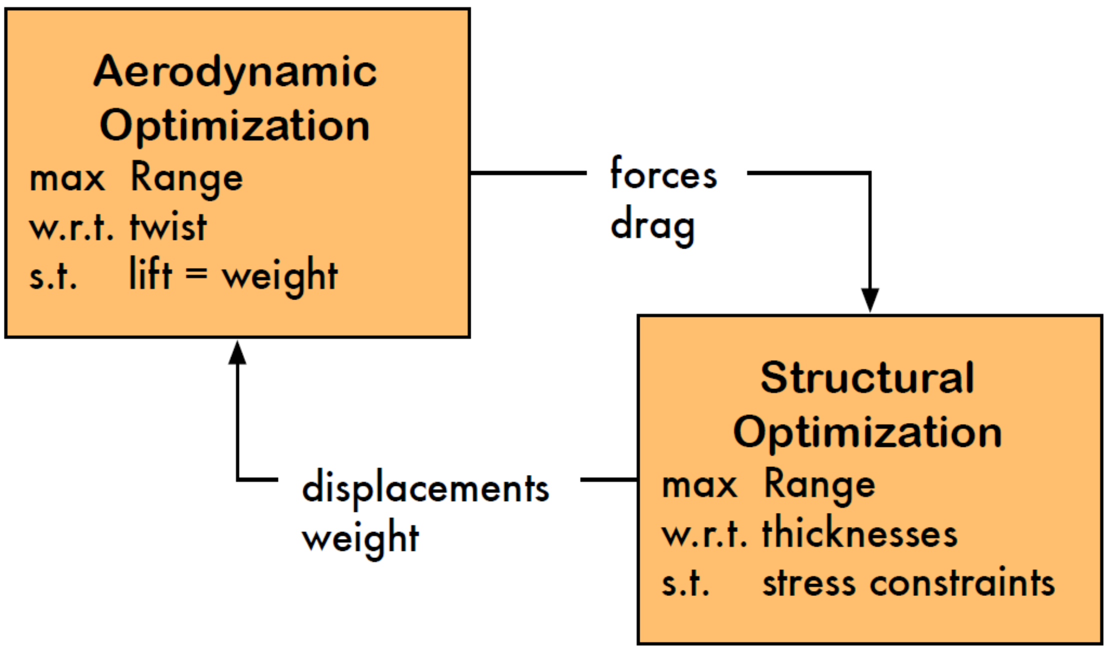
3) Multidisciplinary Design Optimization





4 Conclusions and future directions

1) Multidisciplinary Design Optimization



$$\text{Range} \propto \frac{\text{Lift}}{\text{Drag}}$$

So many directions exist to incorporate neural networks!

Needs literature review to take a “evolutionary” direction!

Thank you!