Deep Learning in CFD @BMW

Data Innovation Lab Final Presentation

Mentor:

Laure Vuaille M.Sc.

Co-Mentor:

Marija Tepegjozova M.Sc.

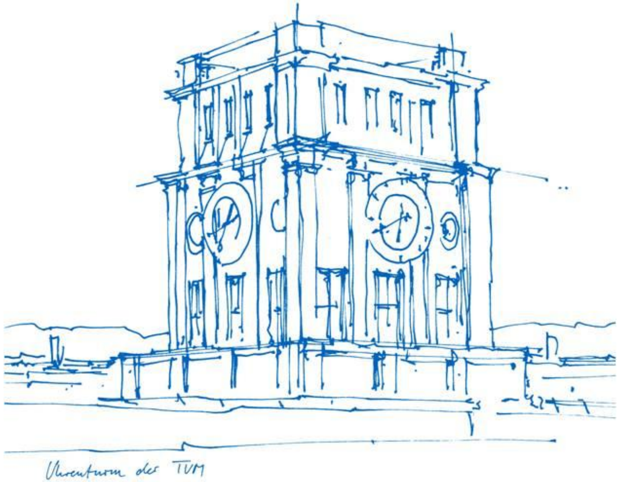
Presenters:

Felix Asanger

Meng Liu

Thana Guetet

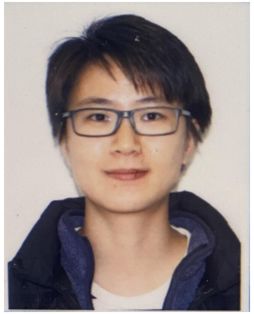
Munich,July 31. 2020



Team Members

Felix Asanger

Robotics, Cognition and Intelligence



Meng Liu

Robotics, Cognition and Intelligence

Thana Guetet

Robotics, Cognition and Intelligence

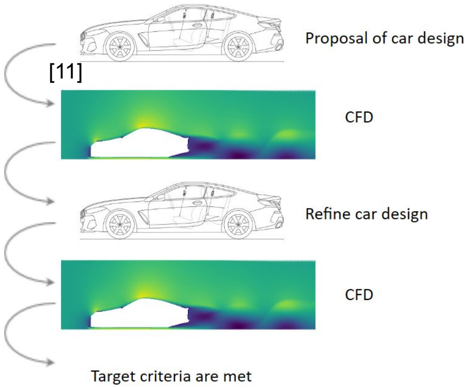
Outline

1. Problem Definition
2. Goal of the Project
3. Workflow
4. Data Management and Resources
5. Data Profile
6. Literature Research
7. Data Analysis
8. Adapted Architecture
9. Results
10. Summary

References

1. Problem Definition

Simplified car design cycle



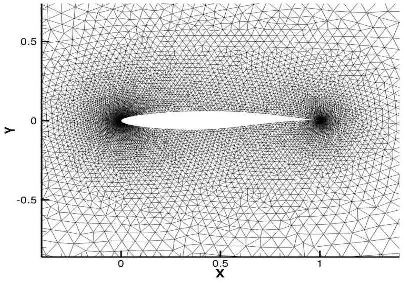
Problem

- Iterative approach
- After every design step -> CFD simulation
- High accuracy trough CFD simulations, but costly and time consuming optimization process

Why are CFD simulations so costly?

- Solves the Navier-Stokes equations for every cell in our domain for a lot of timestamps
- Generates a well fitting mesh + data preprocessing
- Speedup through usage of Reynolds–Averaged Navier-Stokes Equations

BUT the mesh generation and the computation are still consuming a lot of time.



[12]

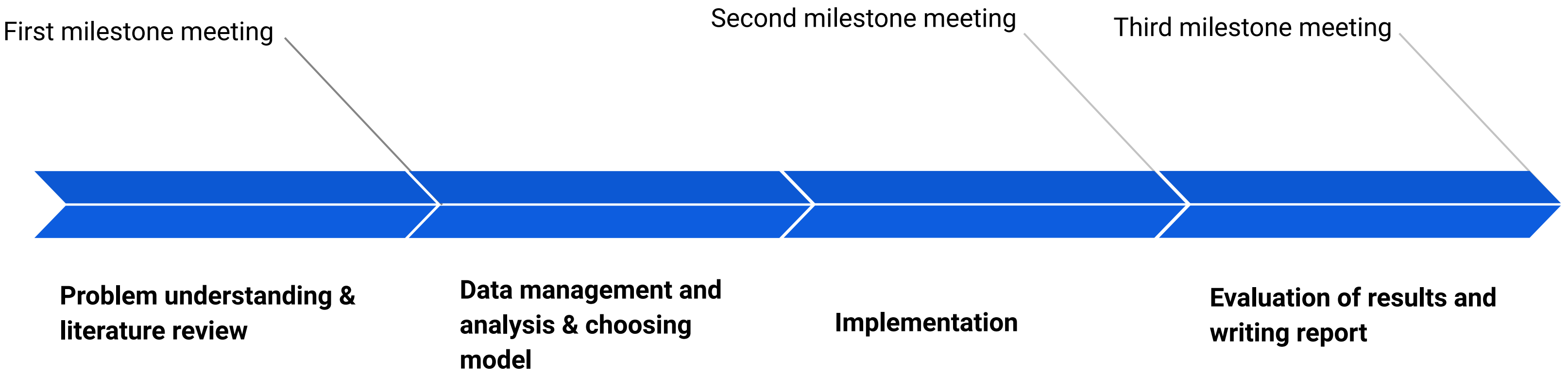
2. Goal of the Project

Goal of the project

- Adapt and evaluate a Deep Learning architecture to predict the 2D airflow characteristics (velocity in x and y directions and pressure) .

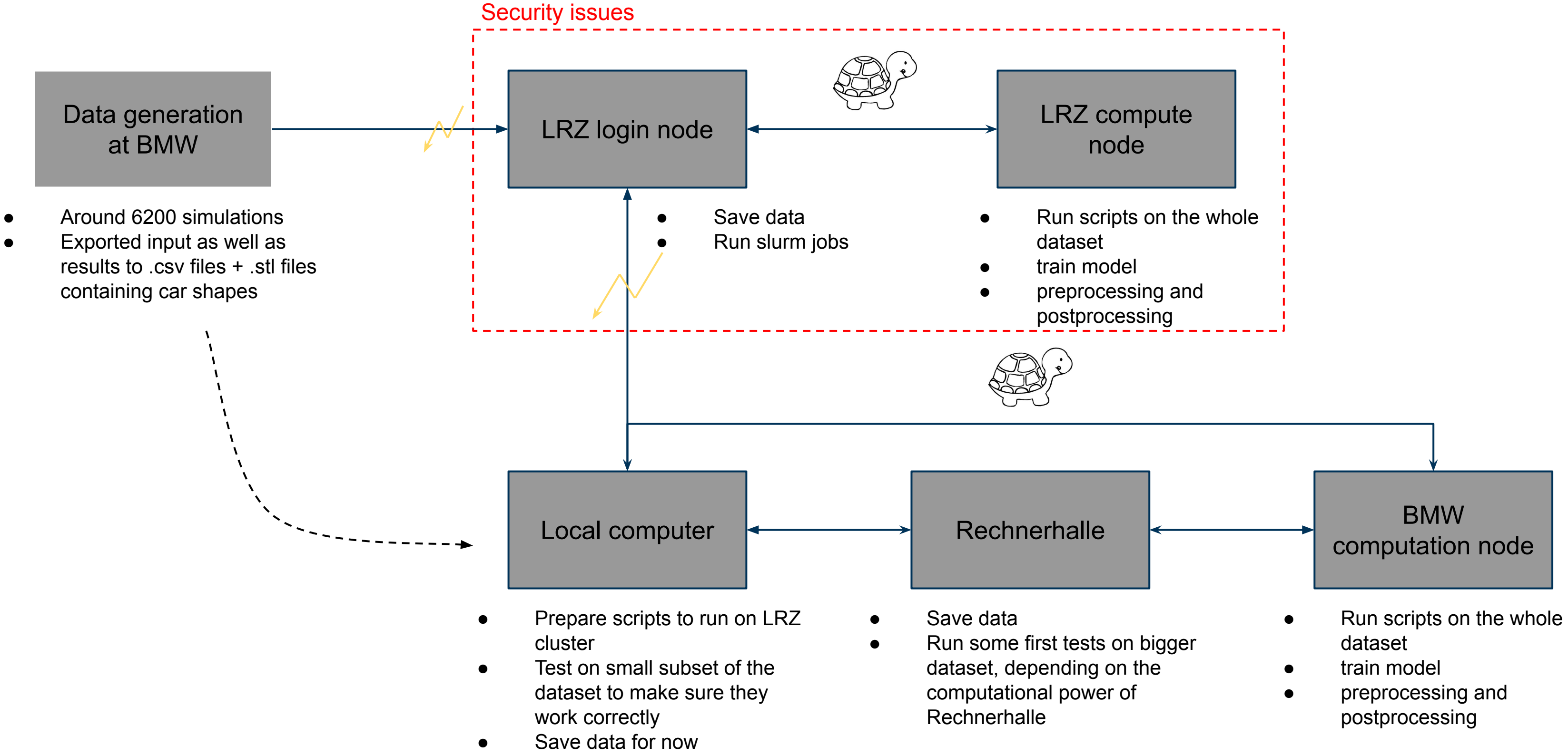
- Assess the performance of the chosen model given the complexity of the problem, namely the varying input velocities and car shapes.

3. Workflow



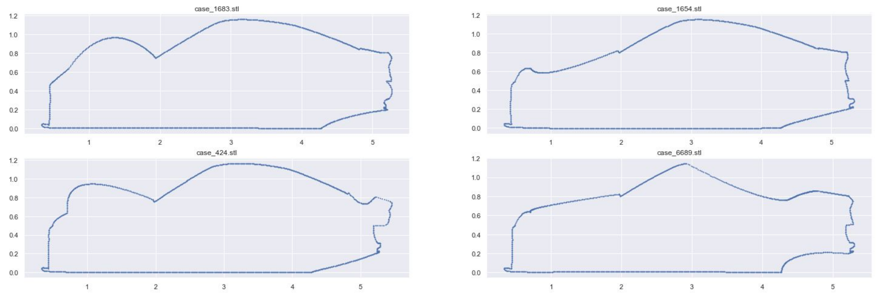
First milestone meeting
Second milestone meeting
Third milestone meeting
Problem understanding & literature review
Data management and analysis & choosing model
Implementation
Evaluation of results and writing report

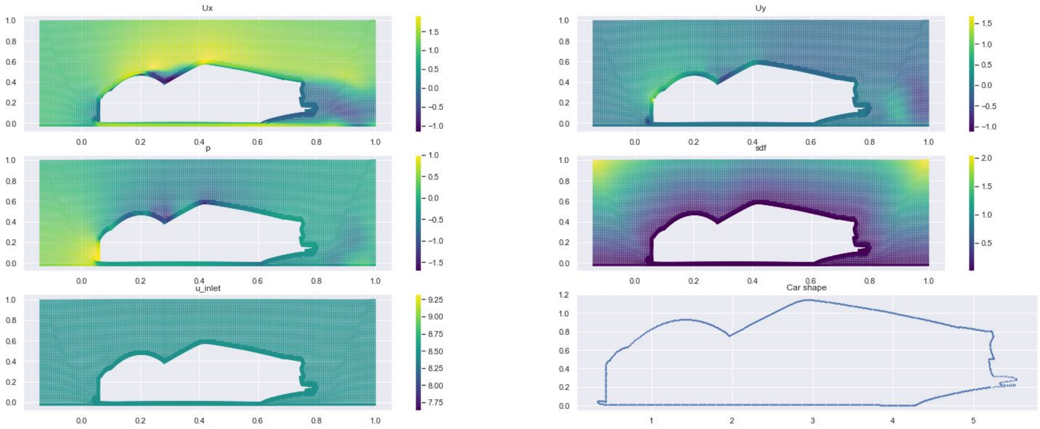
4. Data Management and Resources



5. Data Profile

- 6200 simulations gathered by our mentor at BMW
  - One simulation (.csv) contains values: **Ux, Uy, x, y, pressure**, force, drag force only, lift force only; plus 3D **car mesh** (.stl)
  - **U_inlet**





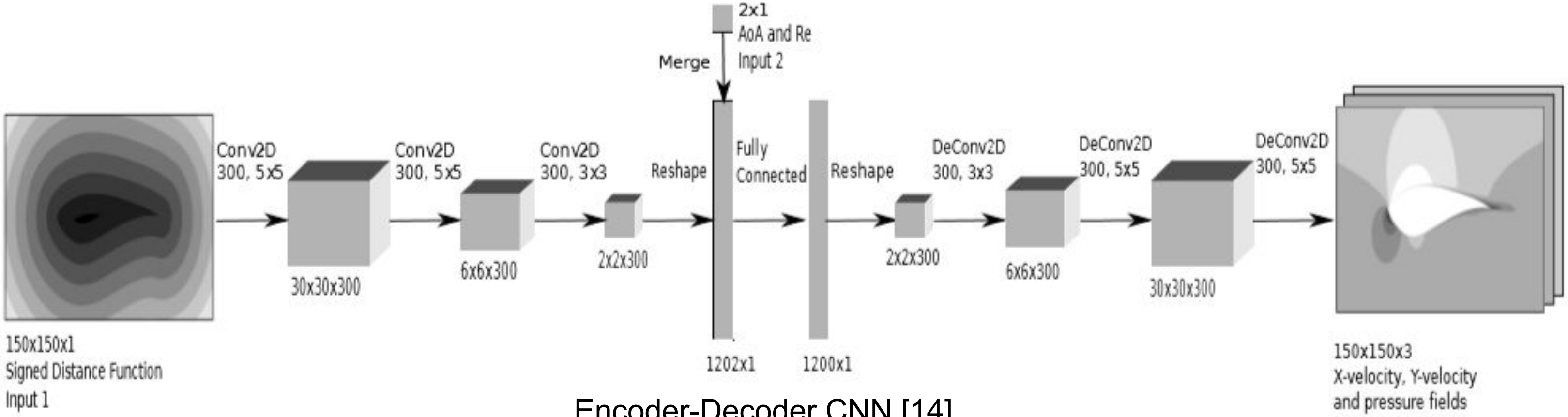
Figures:
Left: Ux, Uy, pressure, Signed Distance Function (SDF), U_inlet, car shape projected on 2D.
Upper: Examples of of different car shapes used for generating the CFD simulation samples

6. Literature Research

1. Train the data from a computer vision point of view -> structured grids (images)
2. Directly work with point clouds
3. Adapt physical constraints to boost the performance (accuracy).

6. Literature Research

Encoder-Decoder CNN [9, 14]



Encoder-Decoder CNN [14]

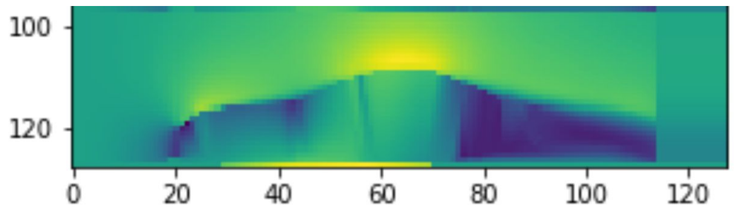
- Setting: Laminar Flow, constant inlet velocity, airfoils or simple shapes, structured grids
- Shape representation: Signed Distance Function (SDF)
- Feature extraction : Convolutions, images reconstruction: Deconvolutions
- Mask on MSE: Depending on the sign of SDF

6. Literature Research

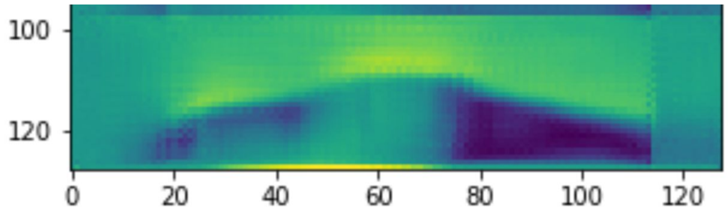
U-Net [2,3]

- Using skip connections by concatenation.
- Variations of U-Net: Stacked U-Net, Parallel U-Net, Coupled U-Net.
- Setting:
    - Varying inlet freestream (Angle of Attack in range +/-22.5),
    - 1505 airfoil shapes,
    - 128^2x3 input tensors, 128^2x3 output tensors.
- Using of physical-related preprocessing.
- Our setting: unstructured point clouds, different shapes of cars and varying inlet freestream.

Overfitting on one sample experiment



Ground Truth - Ux



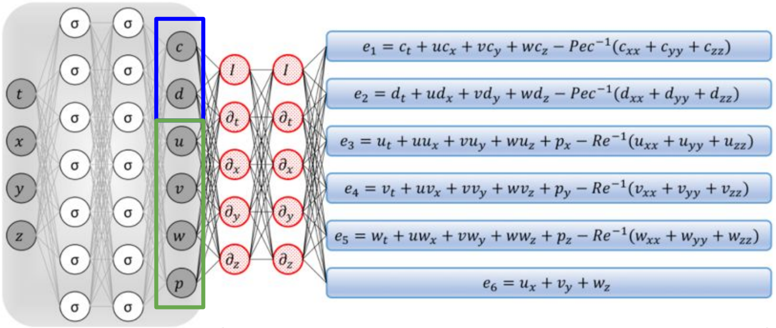
Prediction - Ux

6. Literature Research

Hidden Fluid Mechanics [8]

Key ideas

- Leverage physical laws in model training (loss function)
- Infer the velocity and pressure field from concentration of a passive scalar
- Learn the mapping: (x, y, z, t) -> (c, d, u, v, w, p)



Non-dimensional transport equation for a passive scalar **c** and **d = 1-c**

Non-dimensional Navier-Stokes equations

Continuity equation for incompressible flows

Trainable neural network

NO trainable parameters, but computed deterministically using automatic differentiation

$$
\begin{aligned}
SSE = & \sum_{n=1}^{N} |c(t^n, x^n, y^n, z^n) - c^n|^2 \\
& + \sum_{n=1}^{N} |d(t^n, x^n, y^n, z^n) - d^n|^2 \\
& + \sum_{i=1}^{6} \sum_{n=1}^{N} |e_i(t^n, x^n, y^n, z^n)|^2
\end{aligned}
$$

6. Literature Research

Vector Field Based Neural Network [11]

Analogue to the incompressible fluid mechanics.

$$X'(t) = K(X(t)),\ X(t_0) = X_0$$

$$X_{i+1} = X_i + hK(X_i, \theta) \qquad 0 \leqslant i \leqslant N,$$

- Physical laws preserved
- Scalability is unknown, especially given high-resolution fields; directly modifies the coordinate; accuracy is limited

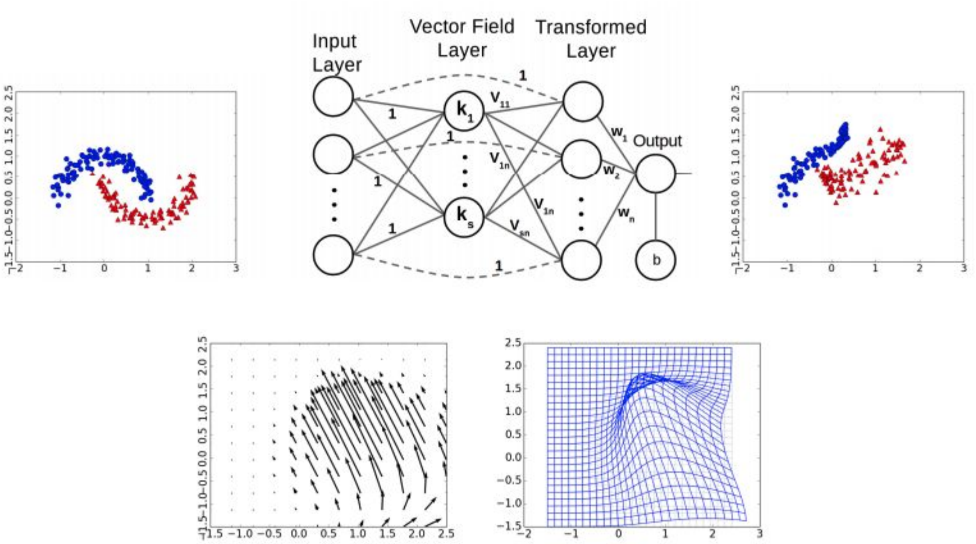
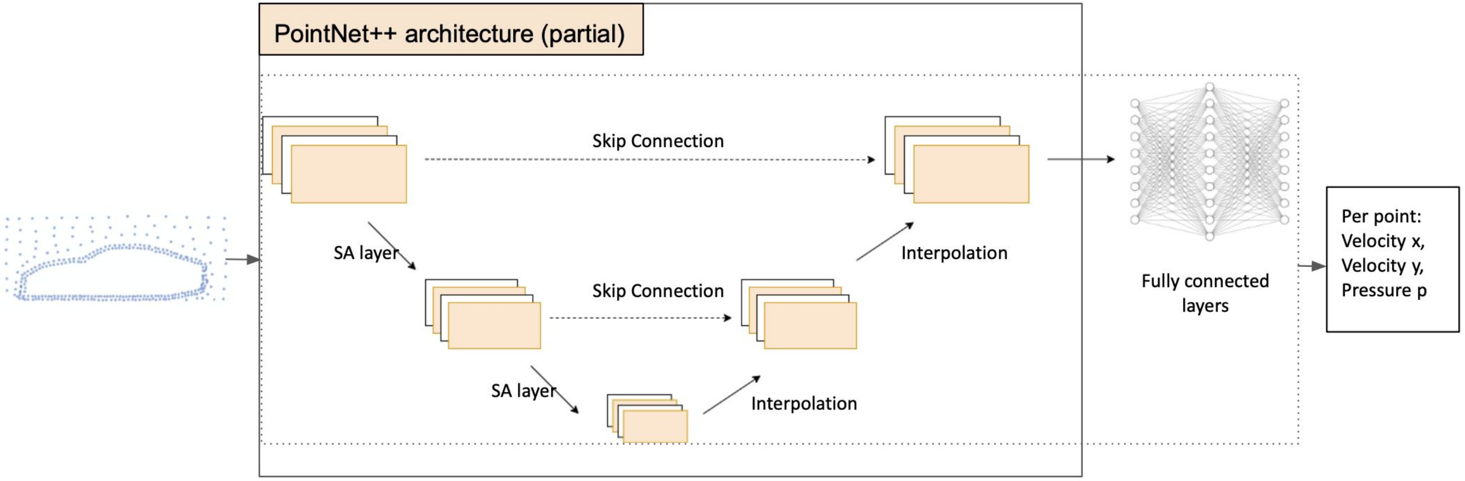


Figure: From left to right, first row presents input data, the architecture, and the transformed data by the vector field layer. Second row presents the vector field and the space distortion.[11]

6. Literature Research

PointNet++ [7]

- Set abstraction layer
  - Sampling & grouping
  - Unit PointNet [6]
- Feature propagation layer
  - Interpolation
  - Feature concatenation
- Designed tasks
  - Classification
  - Segmentation



7. Data Analysis

Pre-Processing Pipeline

outlier detection → cropping the pointcloud → computing SDF → sampling centroids → concatenating old and new features → physical pre-processing → ML pre-processing

outlier detection

outlier if there exits a cell s.t:
|Ux_cell| > 50
or
|Uy_cell|> 50
or
|p_cell| > 10e3

cropping the pointcloud

cut all points with:
- x > 7
- y > 2

computing SDF

Establishes the spatial relation between the point in the point cloud, the surface of the car, and the floor of the control volume.

sampling centroids

sampling 4096 points from the original pointcloud

concatenating old and new features

data consists now of **x**, **y**, **z**, **Ux**, **Uy**, **p**, **sdf**, **u_inlet**

physical pre-processing

```
df['Ux']  /=  uinlet
df['Uy']  /=  uinlet
df['p']   /=  (rho * uinlet ** 2)
df['x']   /=  df['y'].max()
df['y']   /=  df['y'].max()
```

ML pre-processing

Use training set to determine the absolute max of each feature and use it to normalize every feature

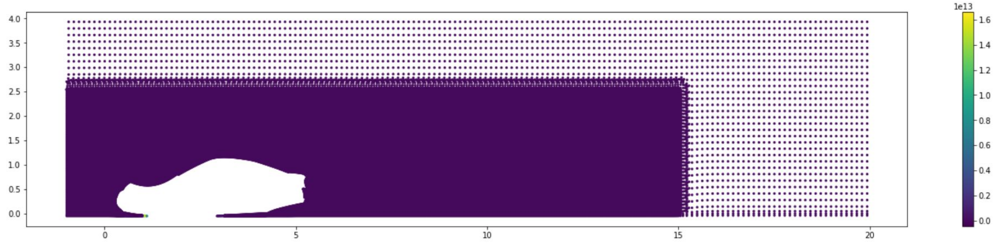
7. Data Analysis

Outlier Detection

- Outliers are samples drawn from a different data distribution
- Outliers don't have physically sane values (very high values whereas inlet velocity range from 2 to 14 m/s)

  → Can disturb the performance of the model

- Detection :
  - z-score,
  - bounds for the mean value of each sample,
  - bounds for the value for each point of the point cloud



The produced Ux flow for an outlier sample (case_6966)

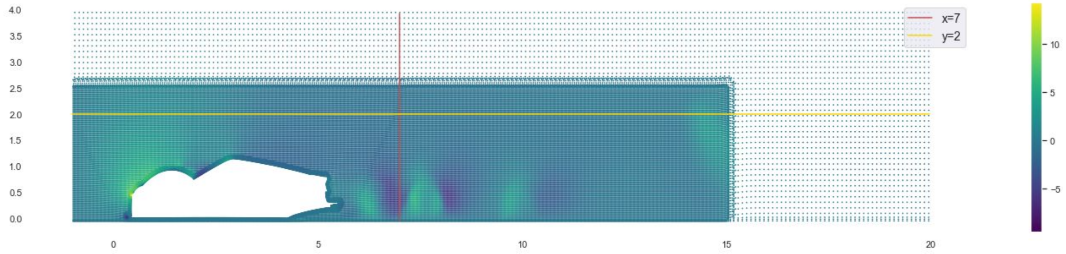
7. Data Analysis

Region Cropping

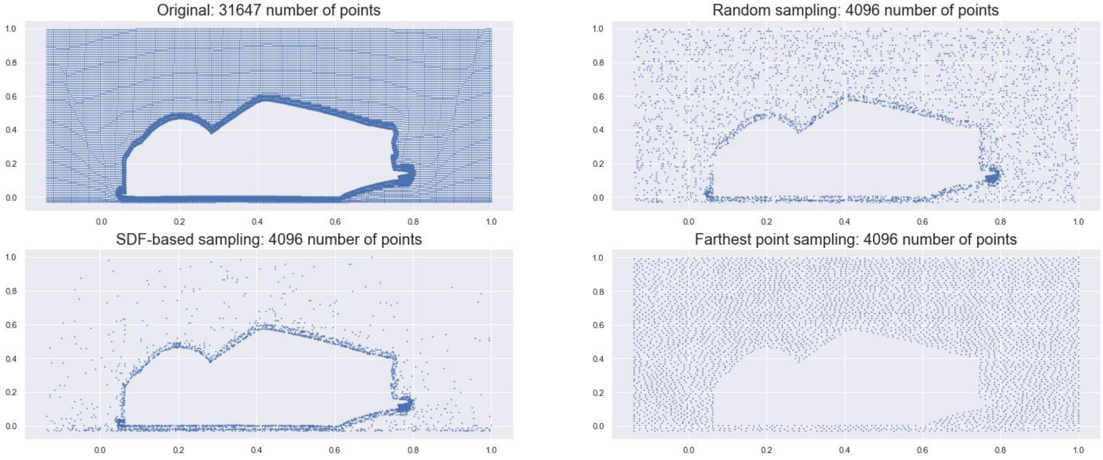
- CFD solver produces large point clouds, which are not needed for a ML problem

Points Down-Sampling

- Random
- SDF-based
- Farthest point



Dimension of an original point cloud and the respective cropped version



Original point cloud and the results applying the different sampling methods

7. Data Analysis

Raw Data

- Velocity field $(\vartheta_{Out}, \vartheta_{in})$
- Pressure field p

Normalization

- $\tilde{p}_{out} = p_{out}/\rho|v_{in}|^2$
- $\tilde{\vartheta} = \vartheta_{Out}/|v_{in}|$

Substract the pressure mean

- $\hat{\mathrm{x}}_{\mathrm{out}} = \tilde{p}_{\mathrm{out}} - p_{\mathrm{mean}}$
- $p_{\mathrm{mean}} = \frac{1}{n}\sum_{i=1}^{n} \tilde{p}_{\mathrm{out}}$

Physics-related Pre-processing [3]

Standardization

- Velocity field values are transformed to a [-1,1] range

ML Pre-processing

8. Adapted Architecture

From Classification/Segmentation to Regression

- After set abstraction and feature propagation layers, the output is of the same shape as the input
    - **Problem**: Both layers only use ReLu activation functions which results in positive predictions only
    - **Solution**: Add an additional fully connected dense layer without activation function to the head of the network to predict negative values as well
- Loss-function from categorical cross-entropy loss to a mean-squared error loss

8. Adapted Architecture

Fluid Mechanics Induced Loss - Reynolds Averaged Navier-Stokes (RANS) equations

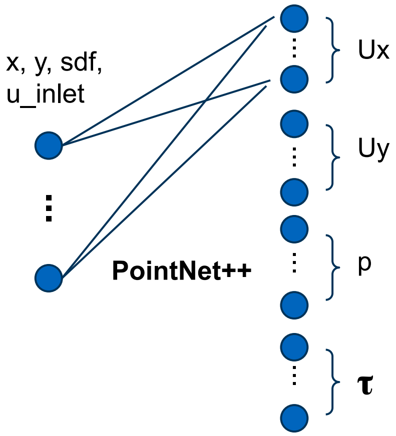
mass conservation: $$\frac{\partial \bar{u}_i}{\partial x_i} = 0$$

momentum conservation: $$\rho \bar{f}_i + \frac{\partial}{\partial x_j}\left[-\bar{p}\delta_{ij} + \mu\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \overbrace{\rho \overline{u_i' u_j'}}\right] - \rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = 0$$

- Overall 4 equations (x- and y-direction for both mass and momentum conservation
- Adaptation for usage with preprocessed data
- RANS equations need physically meaningful values as input
- For example **replace** x_i by $\bar{x}_i \times x_{ml_i} \times x_{max_i}$
- **τ** is called viscosity-induced fluctuation around the mean flow and is typically computed by more complex models

8. Adapted Architecture

Fluid Mechanics Induced Loss - Network Architecture adaption



- Deploying automatic differentiation to calculate the derivatives of the predictions w.r.t. x and y
- First and second derivatives
- **τ** is unsupervised and should be learned by minimizing the physics induced loss terms

Physics induced loss terms:

- use differential equations from previous slide
- differential equations approach zero when fulfilled

Loss = means squared error loss + differential equation loss

8. Adapted Architecture

Fluid Mechanics Induced Loss - Problem

Problem:

- PyTorch doesn’t allow to compute Jacobians but Jacobian Vector Products
- Example: For the derivative of a vector w.r.t. to a vector one gets back a vector (should be a matrix)
- CPU requirement

Possible solution: Compute one backpropagation per predicted value w.r.t. to the input coordinates

- Results in the desired derivatives we need
- Computationally very costly -> not possible in our problem setting
- Example: To compute only the derivative of Ux with respect to x needs 4096 backpropagations

8. Adapted Architecture

Mask on the Loss

- Emphasize higher prediction accuracy on points close to car surface or floor
- Indicate impenetrable boundaries

$$L_{masked}(L_\epsilon(x,y), SDF(x)) = \frac{1}{SDF(x)+c} \times L_\epsilon(x,y)$$

$$L_\epsilon(x,y) = (y-x)^2$$

SDF(x) is the Signed Distance Function at point x; c is a constant to avoid divide-by-zero problem

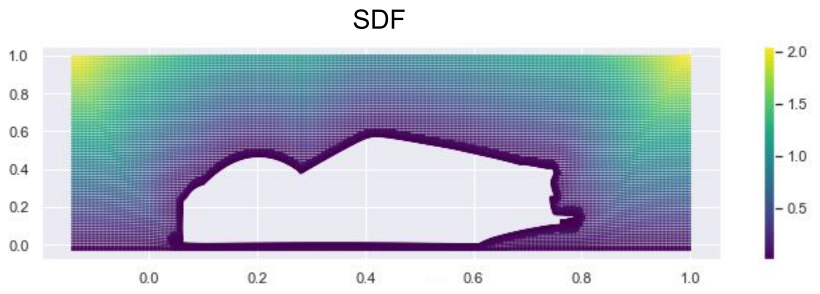


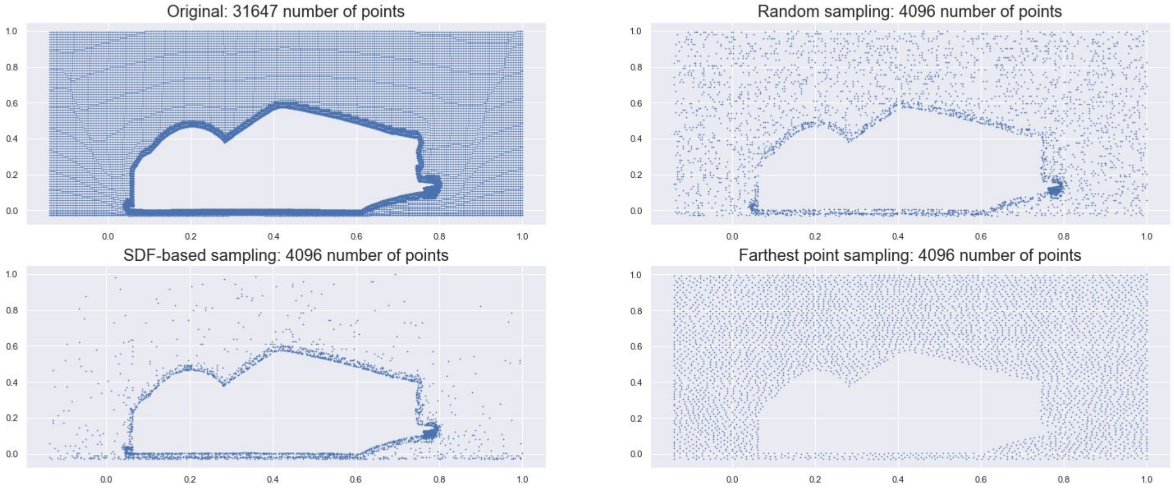
Figure: Signed Distance Function. Darker color implies closer to the car surface or floor, and will be weighted higher in loss computation.

8. Adapted Architecture

Sampling methods - SDF based, random, farthest point

- Implemented SDF-based sampling and random sampling in Pytorch

- Usage in the set abstraction layers of PointNet++ [7]



9. Results

Overfitting

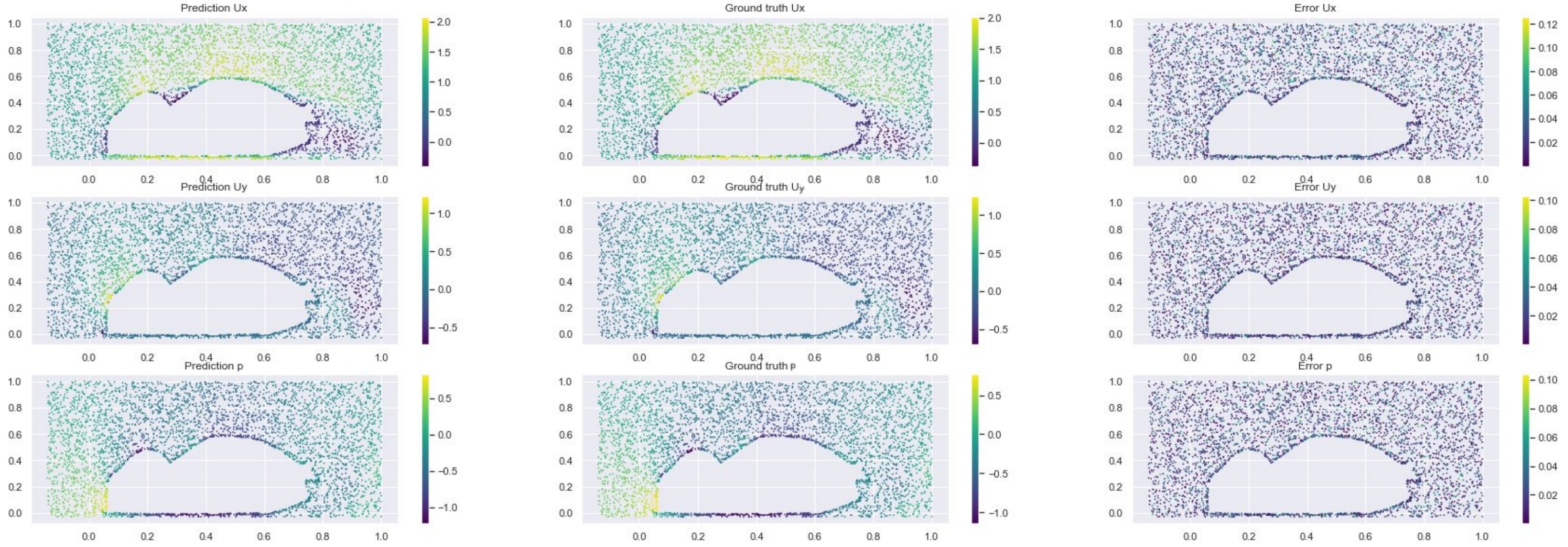


Figure: Predictions (left column), ground truth values (middle column), absolute error (right column). The model is overfitted with one batch (16 samples) from unnormalized U_inlet in range [2, 4]; 1000 iterations; tested with one of training samples.

9. Results

Training with Model Variations

Models No.	Base		Set Abstraction layer			Loss		Mask on Loss		Data		
	Baseline	HFM	Random	SDF	Farthest	MSE	SSE	w/o	SDF	Physical + ML		Physical
										U_{inlet} split	No split	U_{inlet} split
1	✓		✓			✓		✓			✓	
2	✓			✓		✓		✓			✓	
3	✓				✓	✓		✓			✓	
4		✓	✓								✓	
5	✓		✓				✓	✓	✓		✓	
6	✓			✓			✓	✓	✓		✓	
7	✓				✓		✓	✓	✓		✓	
8	✓		✓				✓	✓		✓		
9	✓		✓				✓	✓				✓

Table 1: Variations of the model. More than one ✓in one feature (same color columns) means several sub-models are trained for comparison, and each contains one of the properties of the feature.

9. Results

An example of training results

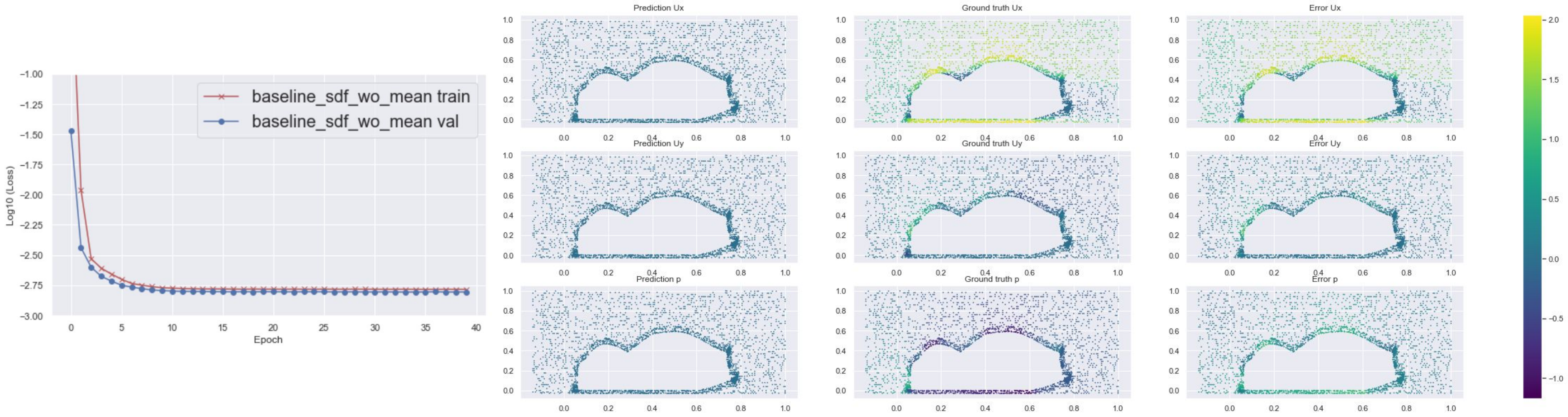


Figure: One example of train/validation loss (left diagram). Prediction of values of interests (left column in right grids), ground truth values (middle column in right grids), absolute error (right column in right grids).

9. Results

Factors Influencing the Model’s Performance

1. **Data**
2. **Problem complexity**
   a. **Flow variation:** Mixed turbulent flows samples with laminar flows samples due to variation of u_inlet.
   b. **Car geometry variation**
3. **Model**: Model capacity might be too small
4. **Downsampling methods**: No significant difference between downsampling methods
5. **Hyperparameter tuning**: learning rate, numbers of SA layers, numbers of centroids in SA layers

10. Summary

- PointNet++, with our current hyperparameter setting, is not able to capture the detailed flow fields.
- In the current state, PointNet++ predicts a nearly uniform distribution of values for all point clouds
- The factors that may influence the performance of the model were studied
- Hypothesis have been formulated and later ruled out with different experiments
- Using a different architecture:
  - Hidden Fluid Mechanics (considering Tensorflow as a platform to solve the gradients computation problem),
  - U-Net (after processing the unstructured point clouds to structured grids).

Thank you!

Deep Learning in CFD @ BMW

Felix Asanger, Meng Liu, Thana Guetet

References

[1] Glenn Research Center and NASA.Navier-Stokes equations.url:https://www.grc.nasa.gov/WWW/K-12/airplane/nseqs.html.

[2] Chen Junfeng, Viquerat Jonathan, and Hachem Elie. “U-net architectures for fast prediction in fluid mechanics”. In: ().url:https://arxiv.org/abs/1910.13532.

[3] Thuerey N. et al. “Deep Learning Methods for Reynolds-Averaged Navier-Stokes Simulations of Airfoil Flows”. In: (2019).url:https://arxiv.org/pdf/1905.13166.pdf.

[4] Thuerey N. et al.Github repository: Deep-Flow-Prediction.url:https://github.com/thunil/Deep-Flow-Prediction.

[5] Pytorch.Pytorch.url:https://pytorch.org/.

[6] Charles R Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In:arXiv preprint arXiv:1612.00593(2016).

[7] Charles R Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In:arXiv preprint arXiv:1706.02413(2017).

References

[8] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. “Hidden Fluid Mechanics: A Navier-Stokes Informed Deep Learning Framework for Assimilating FlowVisualization Data”. In:arXiv preprint arXiv:1808.04327(2018).

[9] Bhatnagar Saakaar et al. “Prediction of Aerodynamic Flow Fields Using Convolutional Neural Networks”. In: (2019).url:https://arxiv.org/pdf/1810.08217.pdf.

[10] Tensorflow.Tensorflow.url:https://www.tensorflow.org

[11] Daniel Vieira and Joao Paixao.Vector Field Neural Networks. 2019. arXiv:1905.07033 [cs.LG].

[12] Wikipedia.Computational fluid dynamics.url:https://en.wikipedia.org/wiki/Computational_fluid_dynamics.

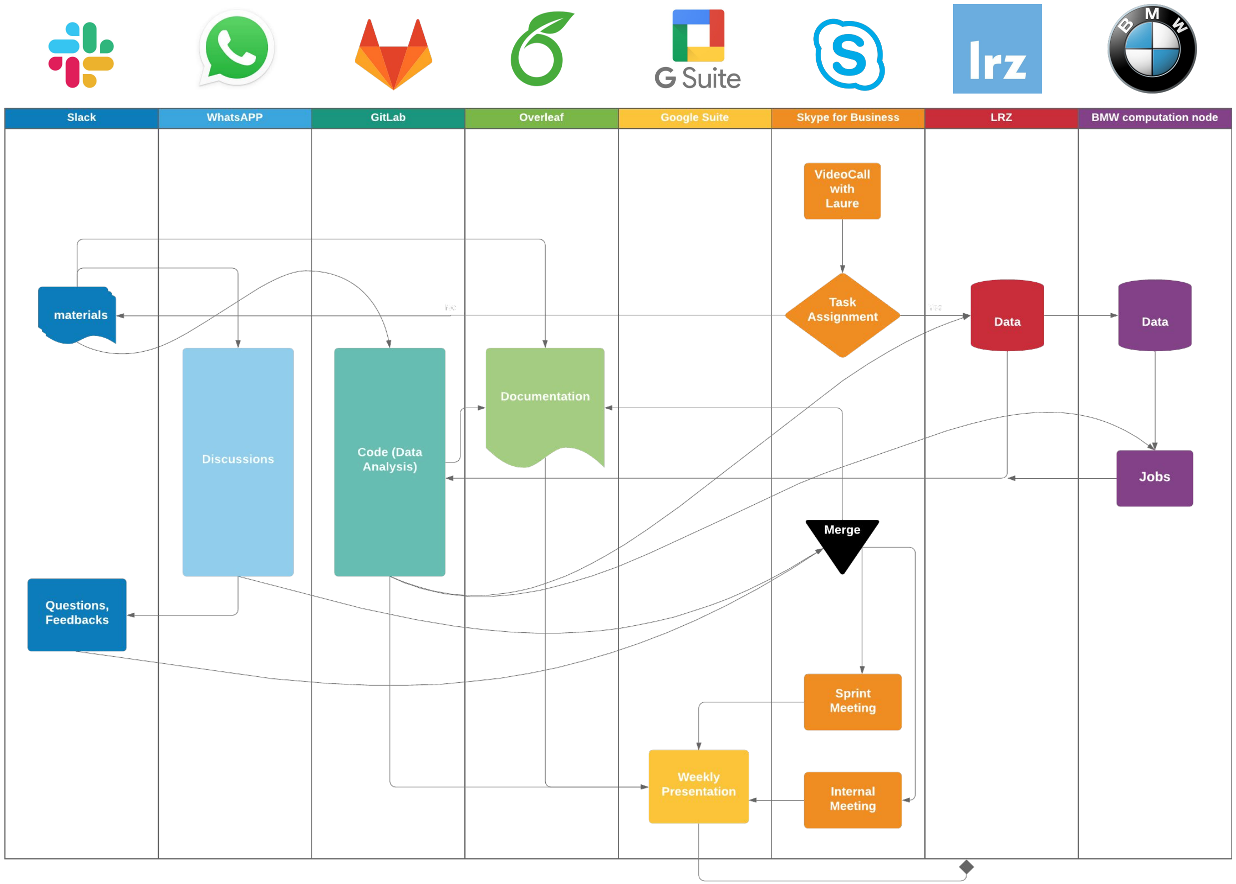
[13] Wikipedia.Turbulence.url:https://en.wikipedia.org/wiki/Turbulence.

[14] Guo X., Li W., and Iorio F. “Convolutional neural networks for steady flow approximation”. In:Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(2016).

Backup Slides

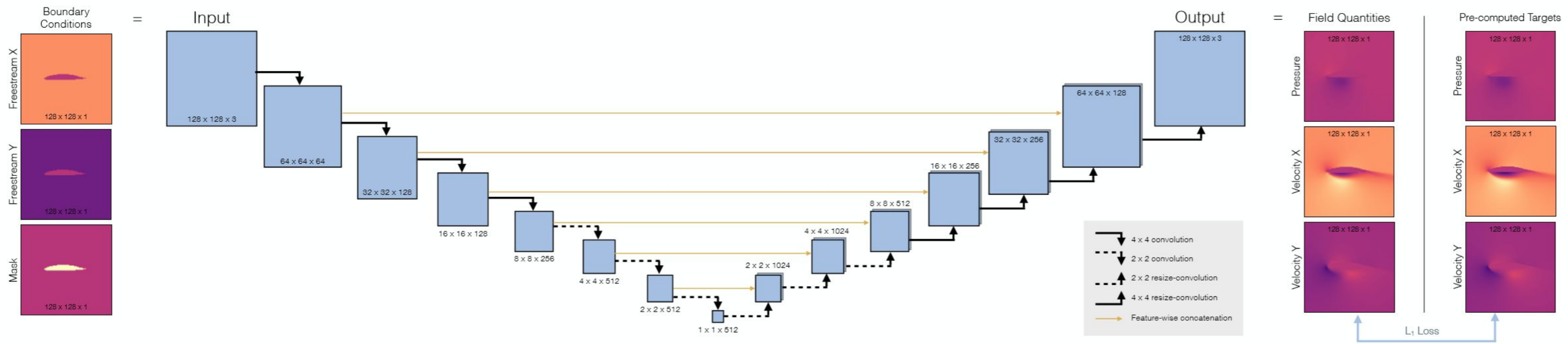
3. Communication and Data Management

- 3 x meetings/week
  - Weekly report with the mentor
  - Sprint meeting with the mentor
  - Internal meeting
- Several platforms to boost cooperation



5. Literature Research

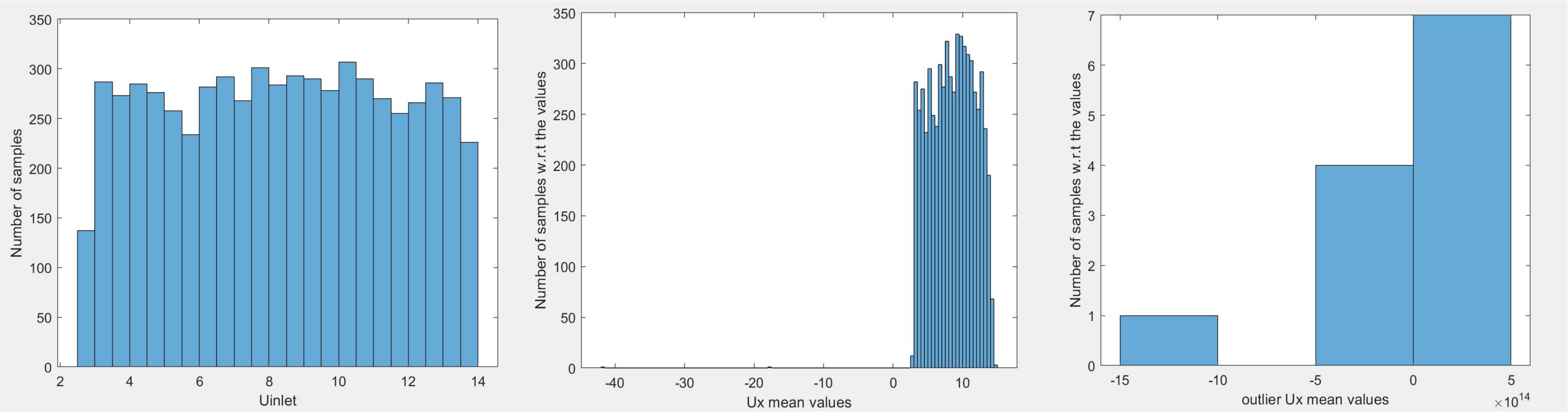
U-Net [2, 3]



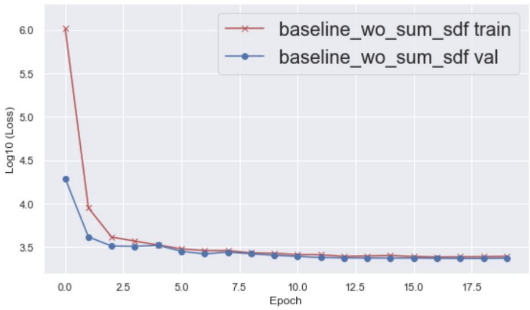
U-Net architecture [3]

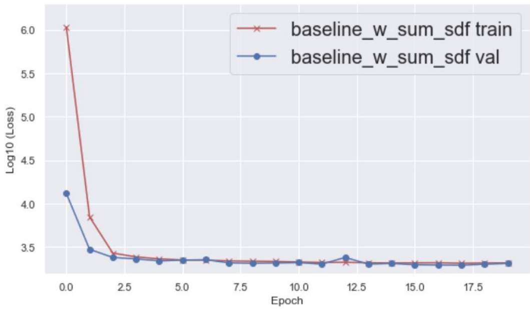
6. Data Analysis

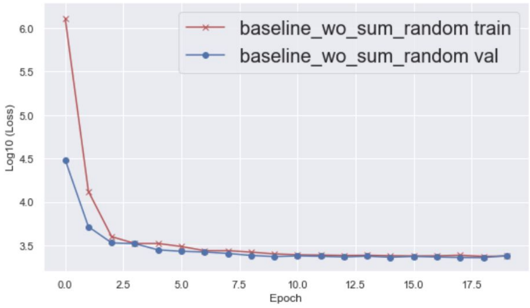
Inlet freestream distribution - Inliers and outliers of Ux

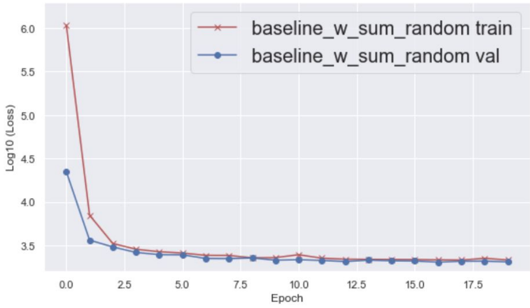


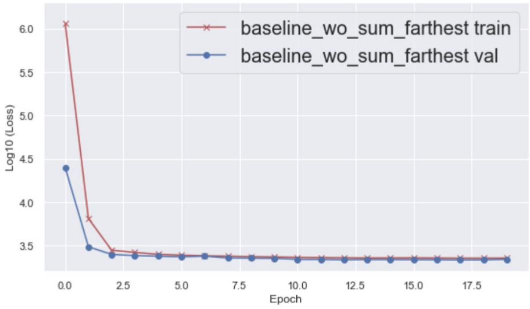
Training results

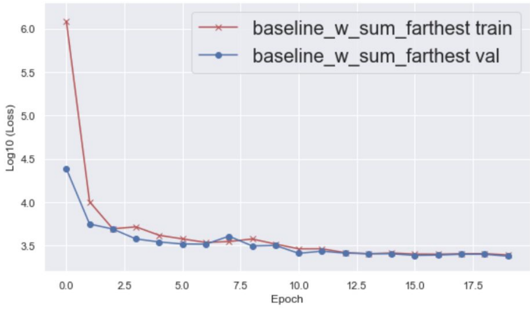












Choose number of centroids