Machine learning model for gas-liquid interface reconstruction in CFD numerical simulations

ML4CFD - DATAIA 2020

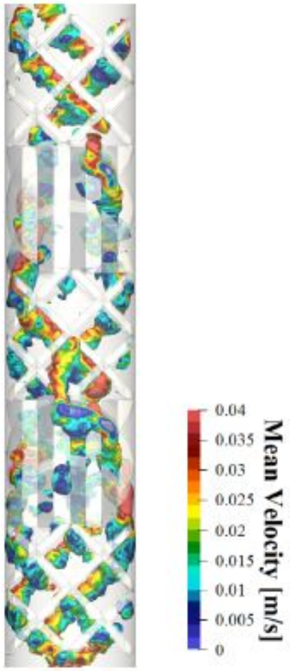
Tamon NAKANO[1], Michele Alessandro BUCCI[1], Jean-Marc GRATIEN[2], Thibault FANEY[2]
Guillaume CHARPIAT[1]

[1] INRIA Saclay, Équipe-projet TAU

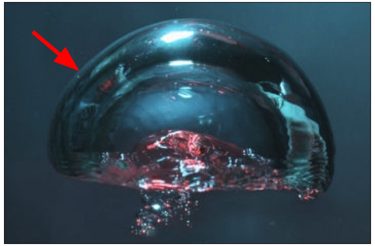
[2] IFPEN, Direction Sciences et Technologie du numérique

Background

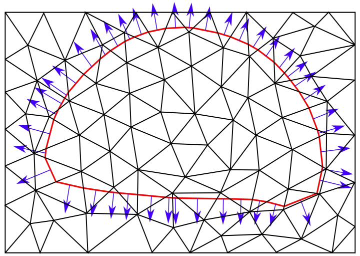
- Immiscible multiphase flow (ex. gas-liquid phase):
  - cavitation, cooling system, mixer
- Numerical method is an effective approach
- The Volume of Fluid (VoF) method:
  - Consider the dynamics of the surface tension force
  - Reconstructs the interface (i.e., normal, location, curvature)



Static mixer*



Free surface

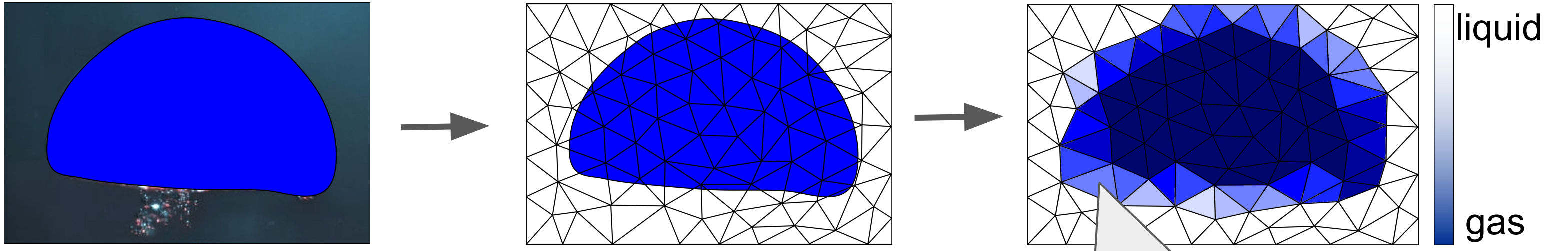


Surface reconstruction

*Picture from Marco Scala's PhD thesis at IFPEN Lyon

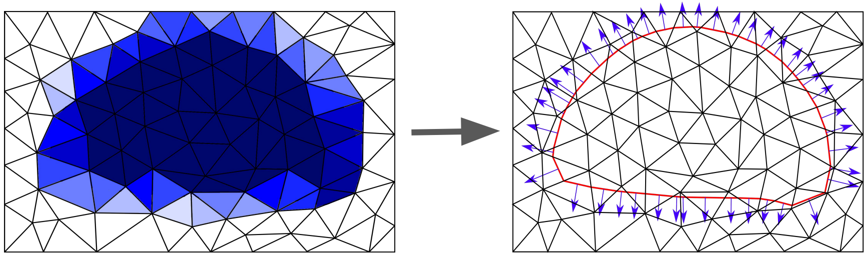
Discretization & Reconstruction

Discretization



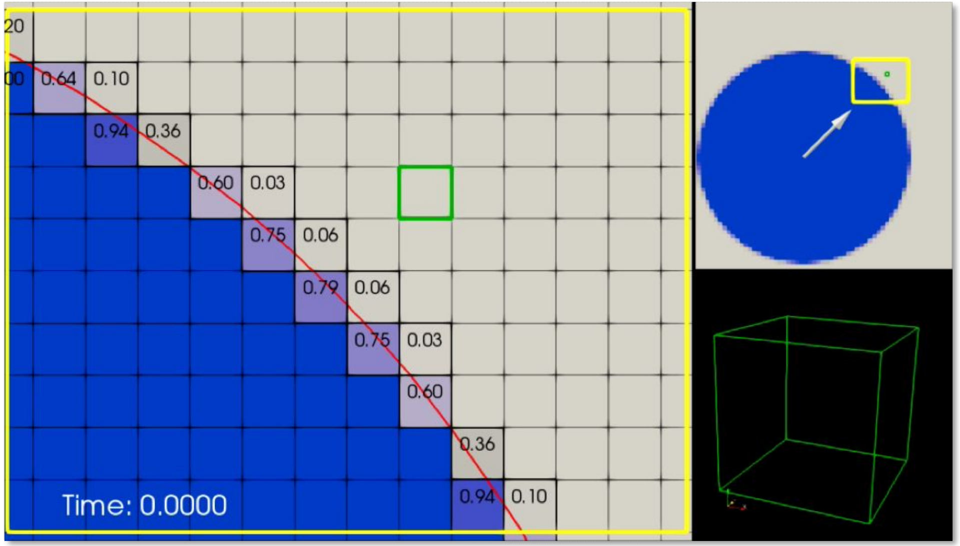
Volume fraction, pressure, velocity are stocked

Reconstruction



- Geometrical information is lost
- In some models geometrical features are required (*i.e.* interface tension force computation)

Reconstruction by the VoF



Source: OpenFOAM
https://www.youtube.com/watch?v=F2WCsF8Sn84

The VoF is **accurate** (good at mass conservation) and performs well on **structured grids**.

Navier - Stokes in multiphase flow

Additional terms in the Navier-Stokes equations to deal with multiphase flows:

Momentum equation

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \underbrace{\nabla \cdot (\rho \mathbf{U}\mathbf{U})}_{\text{Advection}} = \overbrace{-\nabla p}^{\text{Pressure}} + \underbrace{\rho \mathbf{g}}_{\text{Gravity}} + \overbrace{\nabla \cdot (\mu \nabla \mathbf{U})}^{\text{Viscosity}} + \underbrace{\rho \mathbf{S}}_{\text{Extra forces}} + \overbrace{\sigma \kappa \hat{\mathbf{n}}_S \delta(\mathbf{x} - \mathbf{x}_s)}^{\text{Surface tension force}}$$

Continuity equation

$$\underbrace{\frac{\partial \rho}{\partial t}}_{\text{Density}} + \nabla \cdot (\rho \mathbf{U}) = 0$$

$$\underbrace{\rho(\mathbf{x}, t) = \begin{cases} \rho_A, & \text{if } \mathbf{x} \text{ in fluid A} \\ \rho_B, & \text{if } \mathbf{x} \text{ in fluid B} \end{cases}}_{\text{Different densities}}$$

Inria INVENTEURS DU MONDE NUMÉRIQUE

ifp Energies nouvelles

5

Surface tension force

The surface tension depends on **local geometrical features** of the interface:

Note that:

$$\kappa = \nabla \cdot \hat{\mathbf{n}}_S$$

$$F_S = \sigma \kappa \hat{\mathbf{n}}_S \delta(\mathbf{x} - \mathbf{x}_s)$$

Surface tension coefficient
Constant and depends on the liquid/gas at play

Mean curvature
Local curvature of the interface:
$$\frac{1}{2}(\kappa_1 + \kappa_2)$$

Normal
Normal direction to the interface in each point of the interface

kronecker delta
The surface tension is active only on the interface

State of art and limitations

- The bottlenecks of the VoF
  - High computation cost
  - Issue with unstructured grids which leads to residual currents
- Neural network approach to reduce the cost
  - Qi et al.(2018): Curvature estimation on 2D surface with Neural Networks (NNs)
  - Patel et al.(2019): Curvature estimation on 3D surface with NNs. Their approach outperformed the conventional methods

State of art and limitations

- Patel et al.(2019): Multi Layer Perceptron (MLP), emplayable on structured grids with constant discretization.
- Svyetlichnyy et al.(2017): MLP, estimate normal direction on 2D structured grid

	Structured grid	Unstructured grid	Other remarks
VoF	Good	Instability	Expensive
NN	Good	Not available	Less expensive

For industrial cases with complex geometries, unstructured grids are usually employed

Objective

- We propose a machine learning-based method:
  - accurate enough to replace conventional reconstruction methods
  - less computational-demanding

- To that end, we study the use of Graph Neural Network architectures to recover interface properties (e.g., normal, curvature, etc.) from the discretized field.

Graph Neural Networks (GNNs)

Unstructured data are modeled as graph structure

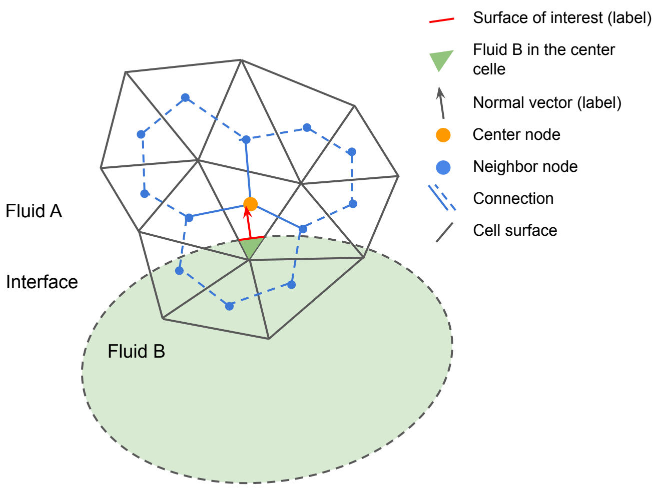
GNNs:

- Geometrically generalized CNNs
- Able to handle a graph-structure (nodes, edges)

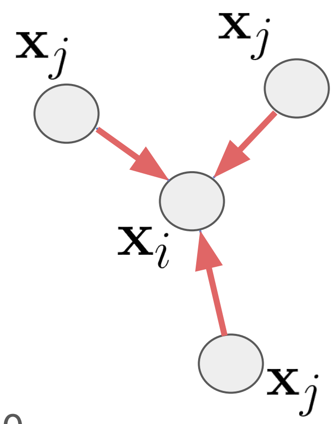
For the prediction, GNNs take:

- Node features: volume fraction α, coordinates of vertices
- Edges: element connections

We employ SAGEConv proposed by Hamilton et al.(2017)*



Message passing in SAGEConv



1) $\mathrm{mean}_{j\in\mathcal{N}(i)}\,\mathbf{x}_j$

2) $\mathbf{W}_2\cdot\mathrm{mean}_{j\in\mathcal{N}(i)}\,\mathbf{x}_j$

3) $\mathbf{x}_i' = \mathbf{W}_1\mathbf{x}_i + \mathbf{W}_2\cdot\mathrm{mean}_{j\in\mathcal{N}(i)}\,\mathbf{x}_j$

4) $\sigma\left(\mathbf{x}_i' = \mathbf{W}_1\mathbf{x}_i + \mathbf{W}_2\cdot\mathrm{mean}_{j\in\mathcal{N}(i)}\,\mathbf{x}_j\right)$

*Inductive Representation Learning on Large Graphs
William L. Hamilton, Rex Ying, Jure Leskovec

Synthetic dataset

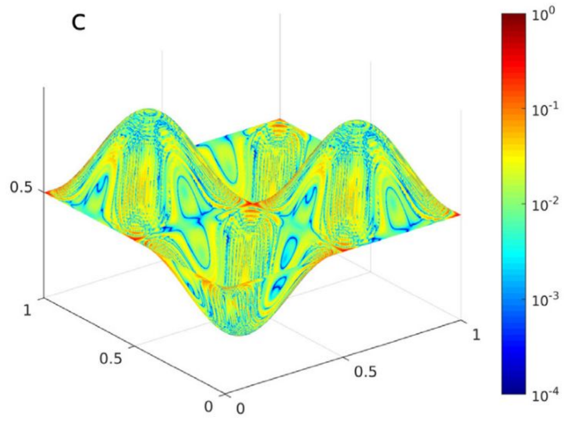
All previous studies used spherical surfaces

- Larger error at saddle points
- **Biased dataset:** only spherical surfaces are considered in the training

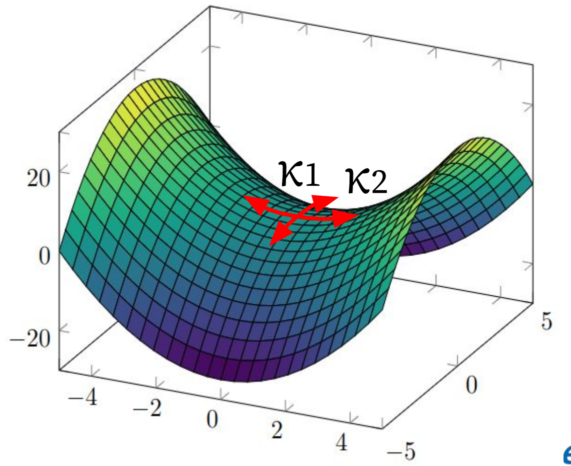
In our study, we introduce a paraboloid-based dataset

$$\frac{x^2\kappa_1}{2} + \frac{y^2\kappa_2}{2} = z$$

κ_1, κ_2 are the curvature on the two parabola at the center which is randomly chosen in space



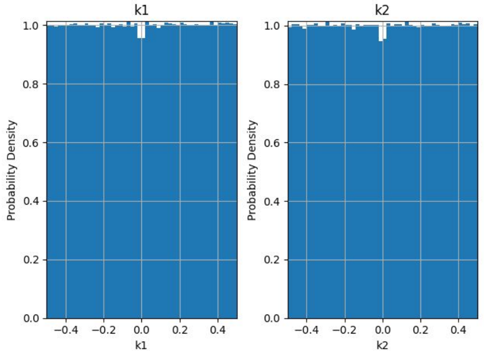
Error on the curvature prediction by ML by Patel et al.(2019)



Paraboloid

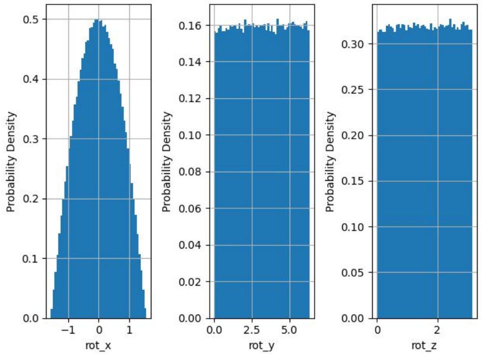
Distribution in the dataset

PDF

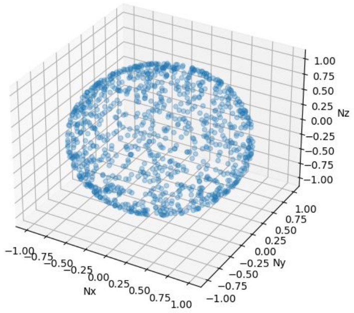


Normalized histogram of κ_1, κ_2

PDF



Normalized histogram of $rot_{x,y,z}$



Normal (1k graphs displayed for the simplicity)

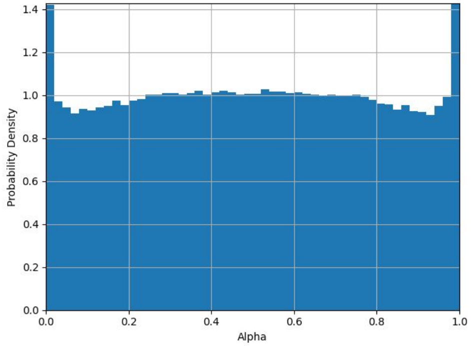
Uniforme distribution:

- Curvature $\kappa 1$, $\kappa 2$
- 3D-rotation
- Volume fraction α
- Center point of the surface

of each paraboloid surface

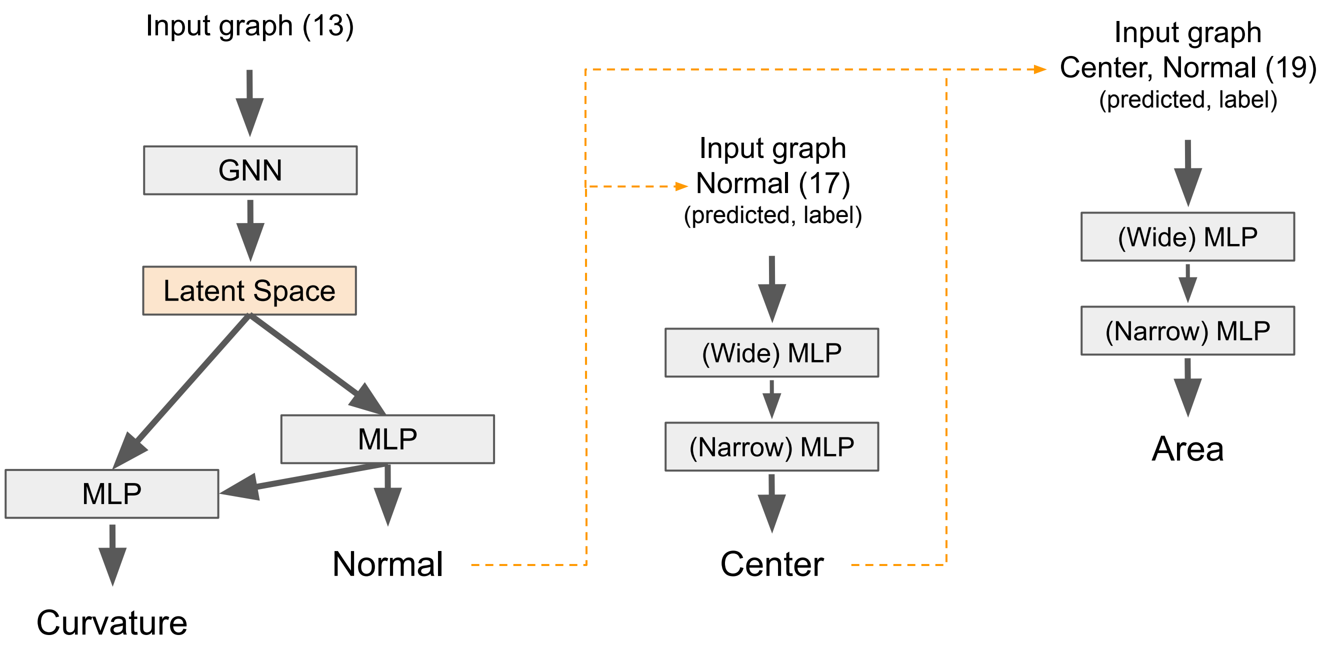
=> **A non-biased dataset has been then generated**

PDF



Alpha

Architecture

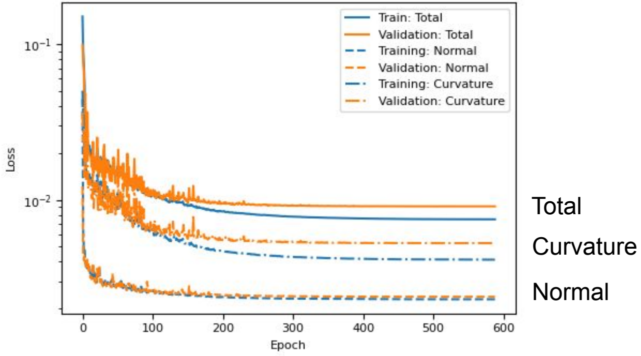


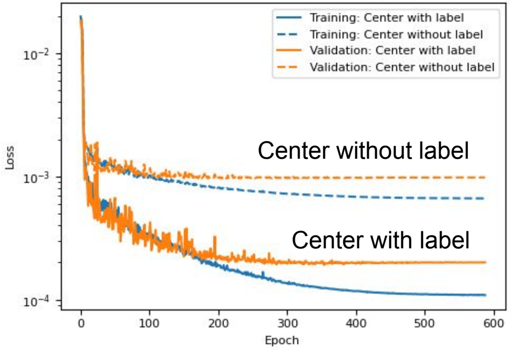
Training

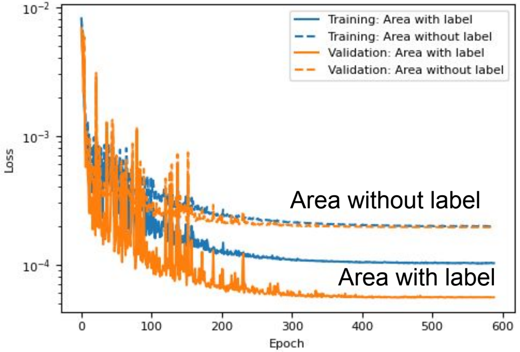
- 500k graphs: 1 graph = 1 normal, 1 curvature
- Train/Validation/Test ratio: 7:2:1
- Variable learning rate (min.1e-6)
- Adam optimizer
- Input: node features, edges
- Predict: normal, curvature, etc.
- Loss: L2
  - Total = Normal + Curvature + Center(w/o label) + Area(w/o label)

- All the variables converge
- No improvement after 508th epoch
- No overfitting

Blue: training
Orange: validation







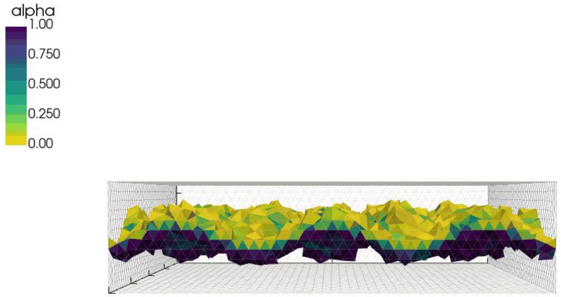
Generalization

Prediction on a **new surface** different from the paraboloid for the training:

$$f(x) = 0.1 \sin 9x \cos 9y$$

$$|H|_{\max} = 0.703$$

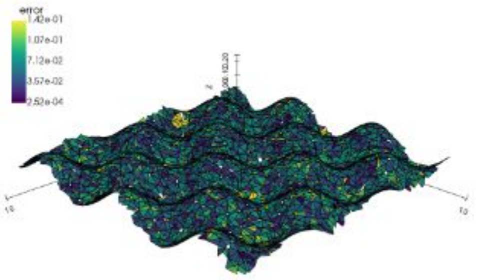
$$H_{\text{training}} \approx [-0.75, 0.75]$$



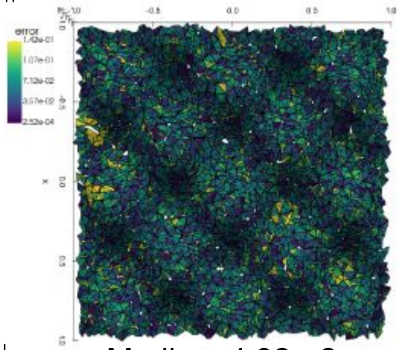
Prediction error

- 0.01< α < 0.99
- Uniform and no dependence on the geometry
- Smooth distribution, no outlier in the histograms

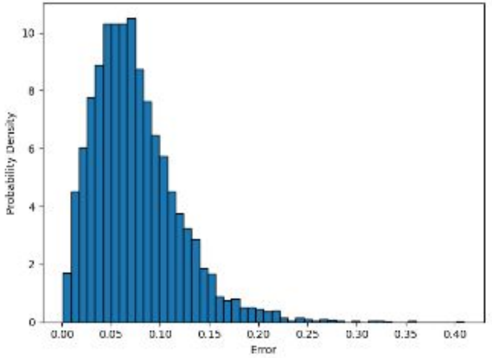
Normal: $N_{error} = \|\mathbf{N}_{label} - \mathbf{N}_{pred}\|$



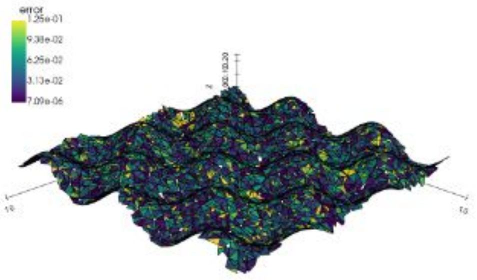
Median: 6.05e-2



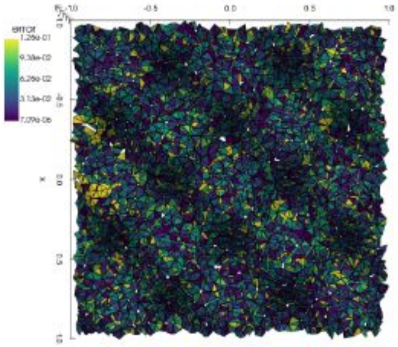
Normalized histogram

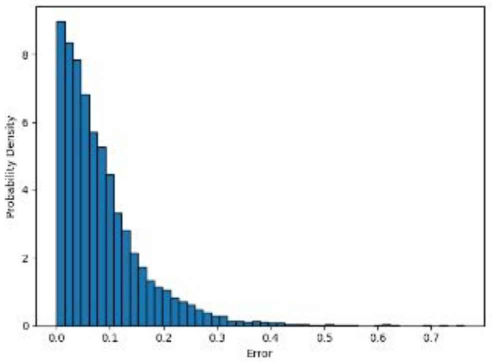


Curvature: $H_{error} = |H_{label} - H_{pred}|$



Median: 4.32e-2





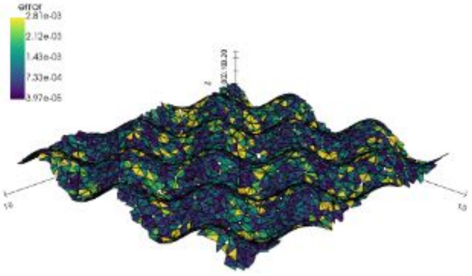
Prediction error

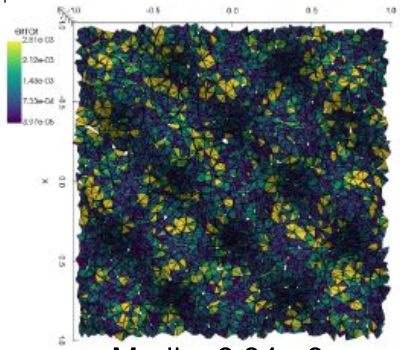
- 0.01< α < 0.99
- Uniform and no dependence on the geometry
- Smooth distribution, no outlier in the histograms

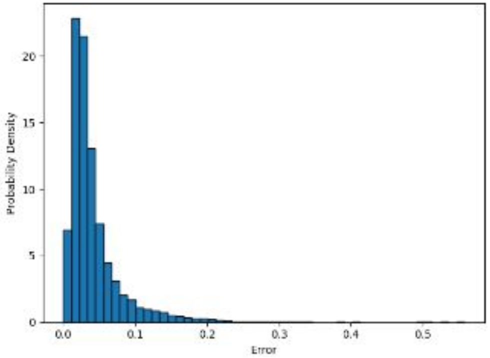
Center: $M_{error} = \|\mathbf{M}_{label} - \mathbf{M}_{pred}\|$

Media: 1.84e-2

Normalized histogram

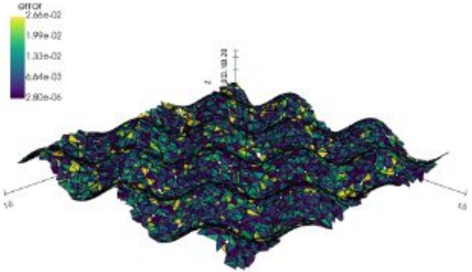


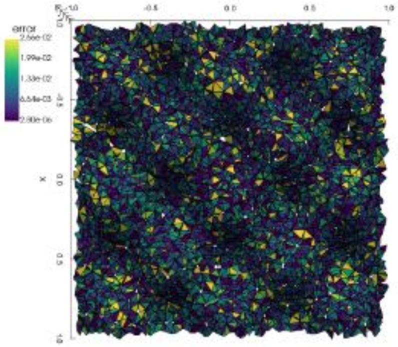


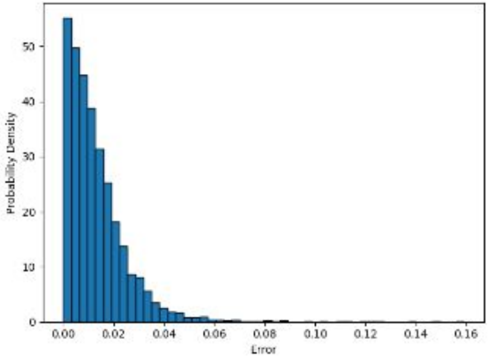


Area: $A_{error} = |A_{label} - A_{pred}|$

Media: 6.84e-3

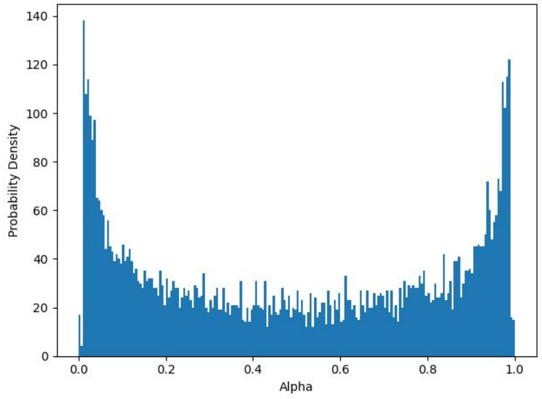




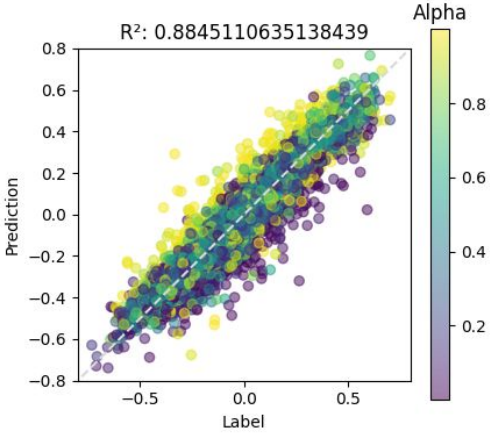


Prediction error

- α has non-uniform distribution
- Curvature prediction, a large error at "marginal α"
  - A large α: the prediction is larger than the label
  - A small α: the prediction is smaller than the label
- This should be taken into consideration when implemented in real simulations



Histogram of α



Curvature (left) prediction vs label

Conclusions & Perspectives

Conclusion

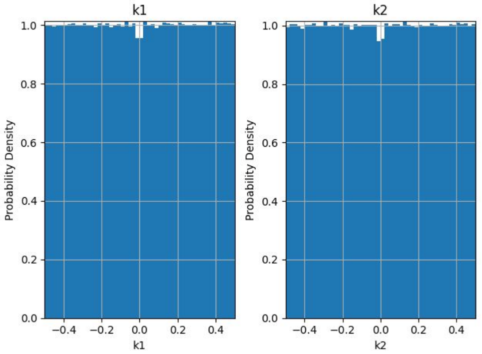
1. **Graph Neural Networks** can be an alternative to the conventional surface reconstruction methods
2. GNNs could be a promising reconstruction approach to unstructured grids

Perspectives

1. Employ the model in a CFD code (i.e. OpenFOAM)
2. Assessment of time performance of the model
3. Treatment of boundary conditions

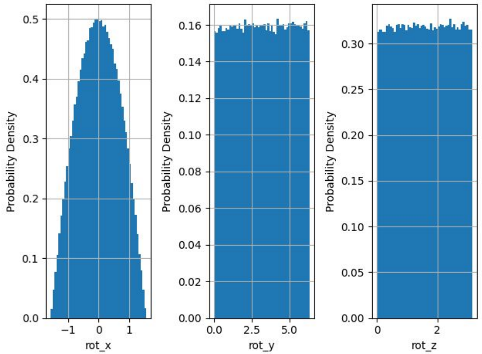
Distribution in the dataset

PDF

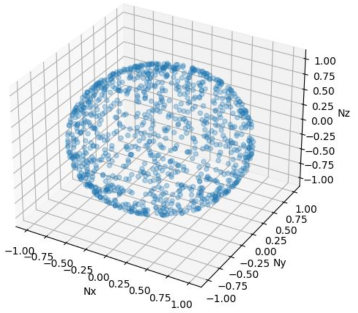


Normalized histogram of κ_1, κ_2

PDF

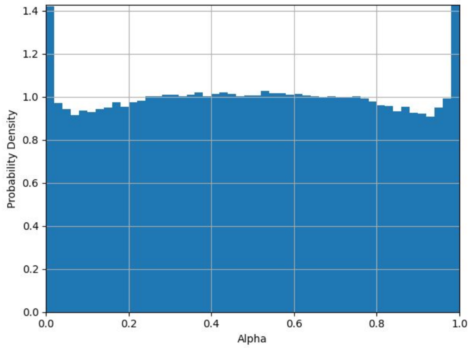


Normalized histogram of $rot_{x,y,z}$



Normal (1k graphs displayed for the simplicity)

PDF



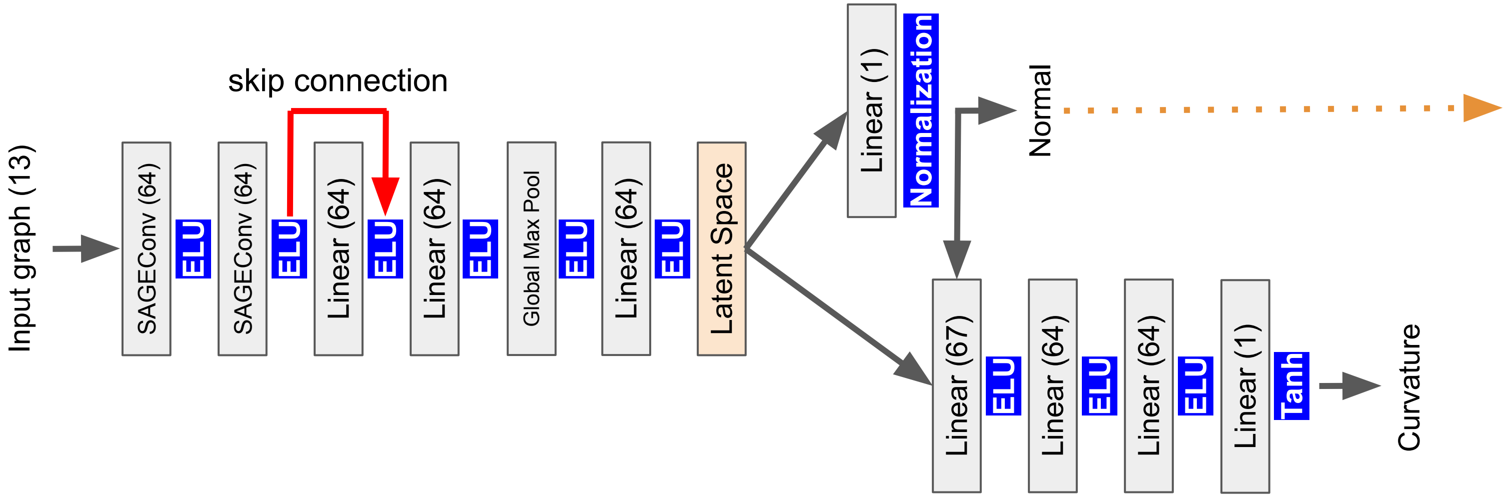
Alpha

- $\kappa_1, \kappa_2 \sim \pm[0.001, 0.5]$ (uniforme distribution)
- $(rot_x, rot_y, rot_z) = \cos^{-1}(2u_2 - 1), \pi u_2, 2\pi u_2$ (uniform rotation in 3D)
- Alpha distribution is uncontrollable in advance => quasi-uniform is generated
- Center of graph randomized

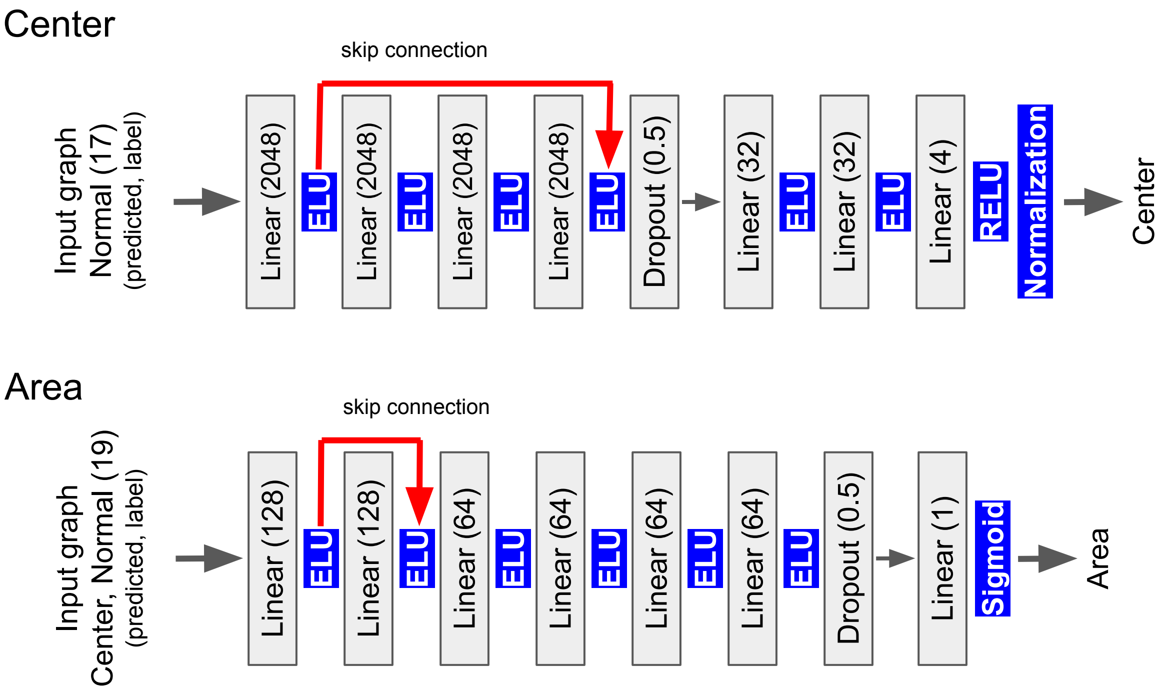
=> **A non-biased dataset has been then generated**

Architecture

Normal, Curvature



Architecture



Loss function

$$\text{Loss}_{\text{total}} = \text{Loss}_{\text{normal}} + \text{Loss}_{\text{curv}} + \text{Loss}_{\text{c:wl}} + \text{Loss}_{\text{a:wl}} + \text{Loss}_{\text{c:wol}} + \text{Loss}_{\text{a:wol}}$$

$$\text{Loss}_{\text{normal}} = \frac{1}{N_b}\sum_{b=1}^{N_b}\sum_{i=1}^{3}\frac{(N_{b,i,label} - N_{b,i,pred})^2}{3}$$

$$\text{Loss}_{\text{c:wl,wol}} = \frac{1}{N_b}\sum_{b=1}^{N_b}\sum_{i=1}^{4}\frac{(p_{b,i,label} - p_{b,i,pred})^2}{4}$$

$$\text{Loss}_{\text{curv,a:wl,wol}} = \frac{1}{N_b}\sum_{b=1}^{N_b}(y_{b,label} - y_{b,pred})^2$$

Model performance

Table 1: Model performance at the 508$^{\text{th}}$ epoch

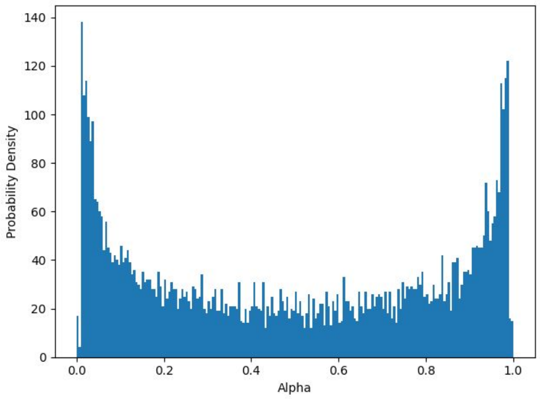
Type of loss	Training	Validation	MSE
$\text{Loss}_{\text{total}}$	7.50e-3	9.07e-3	9.30e-3
$\text{Loss}_{\text{normal}}$	2.30e-3	2.39e-3	2.41e-3
$\text{Loss}_{\text{curv}}$	4.13e-3	5.26e-3	5.40e-3
$\text{Loss}_{\text{c:wl}}$	1.10e-4	2.01e-4	1.84e-4
$\text{Loss}_{\text{c:wol}}$	6.66e-4	9.76e-4	1.05e-3
$\text{Loss}_{\text{a:wl}}$	1.03e-4	5.53e-5	5.51e-5
$\text{Loss}_{\text{a:wol}}$	1.99e-4	1.95e-4	2.03e-4

Table 2: The model performance the test-No.1

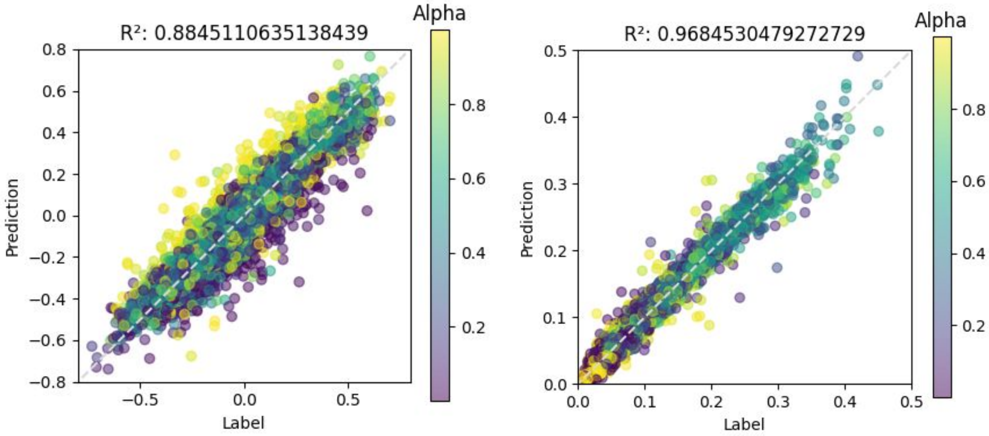
Predicted variable	Median
$N_{error} = \lVert\mathbf{N}_{label} - \mathbf{N}_{pred}\rVert$	5.61e-2
$H_{error} = \lvert H_{label} - H_{pred}\rvert$	4.16e-2
$M_{error} = \lVert\mathbf{M}_{label} - \mathbf{M}_{pred}\rVert$	1.98e-2
$A_{error} = \lvert A_{label} - A_{pred}\rvert$	7.57e-3

INVENTEURS DU MONDE NUMÉRIQUE

Prediction error



Histogram of α



Curvature (left), Area (right)
prediction vs label