Deltares

AIWAM

AI models for North Sea Storm Surges



Martin Verlaan[1 2] (project leader),
Jing Zhao[1 2],
Willem Tromp[1]

[1]Department of Hydrodynamics and Forecasting, Deltares
[2]Delft Institute of Applied Mathematics, TU Delft

2026-11-19 Digishape

Outline

- Motivation
- Reanalysis dataset for water levels in the North Sea
- Time-series models
  - Concept, software and pre-operational tests
- Graph Neural Network (GNN)
  - MeshGraphNet
  - Experiments
- Student projects

Motivation – Traditional numerical models & challenges

- Deltares has developed multiple accurate unstructured grid models:
  - Global Tide and Surge Model (GTSM)
  - Dutch Continental Shelf Model – Flexible Mesh (DCSM-FM)
- Delft3D Flexible Mesh software
- Accurate and versatile, but …

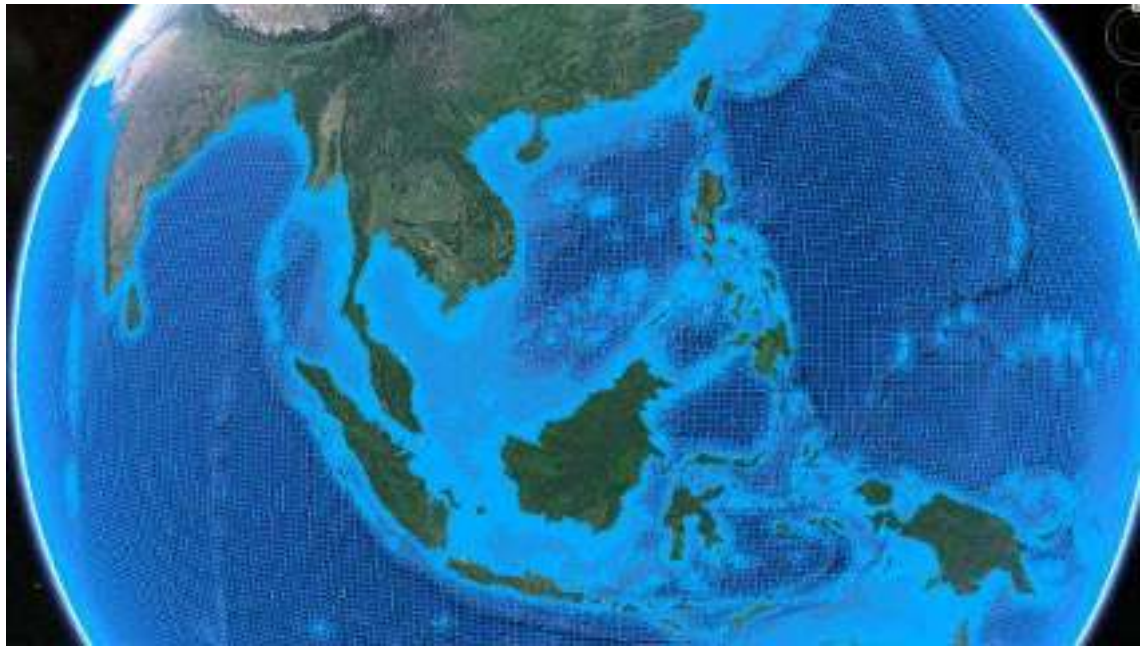


Figure: Grid of the Global Tide and Surge Model for Southeast Asia, from https://www.deltares.nl/en/expertise/projects/global-modelling-of-tides-and-storm-surges

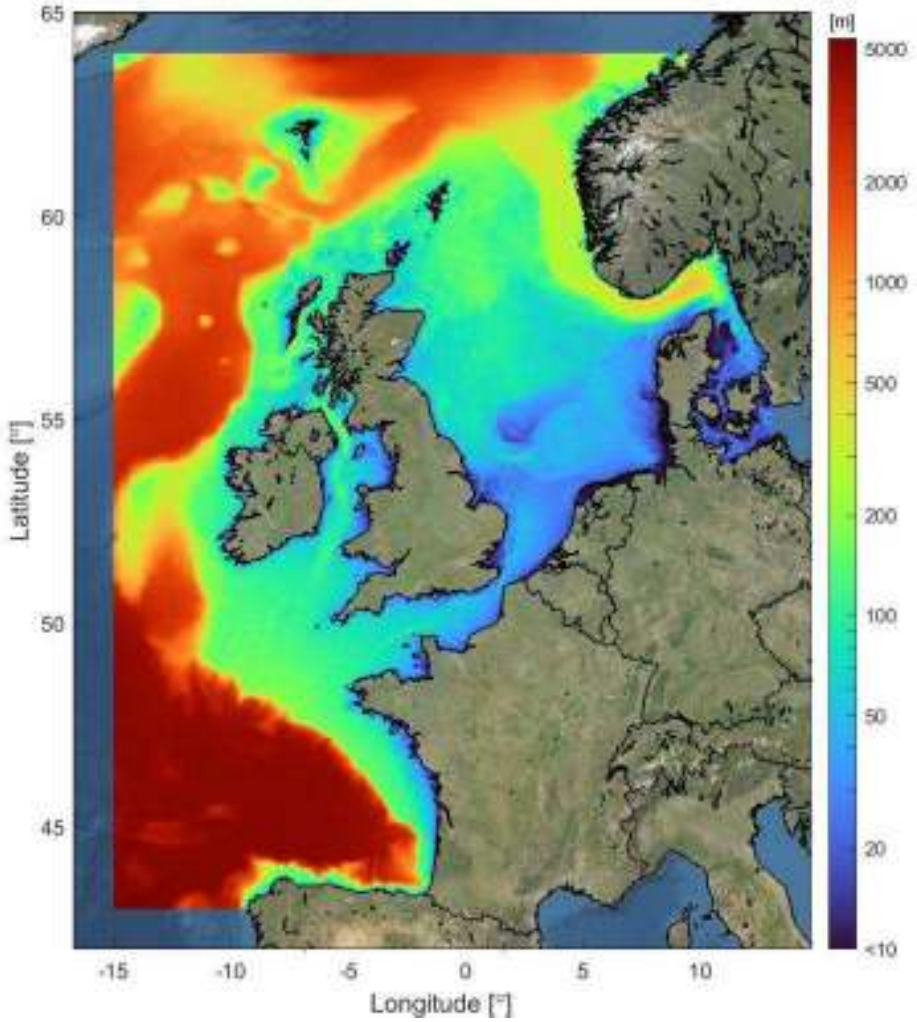
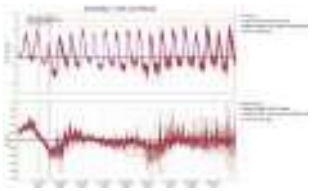


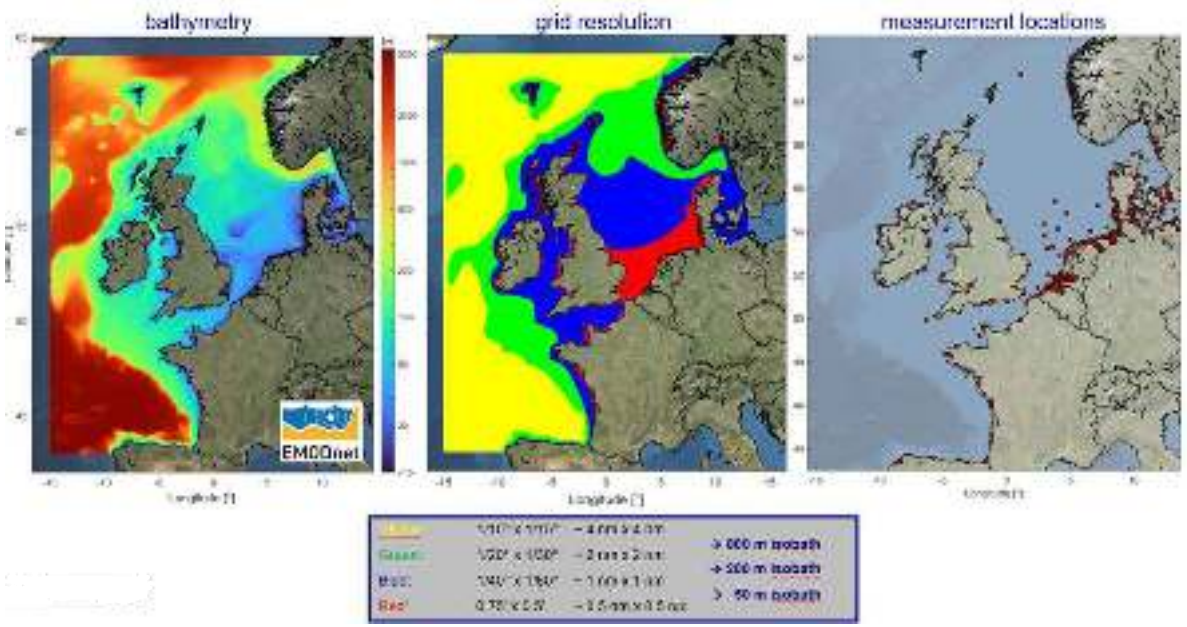
Figure: Overview of the DCSM-FM model bathymetry, from Zijl, F., et al. *3D DCSM FM: a sixth-generation model for the NW European Shelf*. Tech. rep., Deltares, available online at: https://publications.deltares.nl/11208054_004_ 0003. pdf

Ensemble Prediction



December 1st 2025 14:00h

Towards a reanalysis for water levels and currents in the North Sea



Measurements (Rijkswaterstaat)

Data assimilation

Wind & Air-pressure (ERA5/Harmonie)

DCSM Storm-surges

Selection & Compression

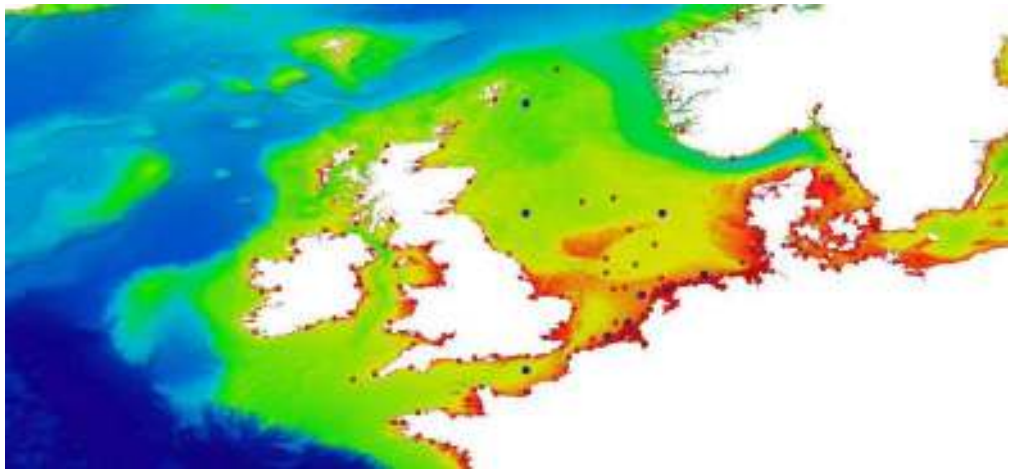
Compressed dataset

Zarr

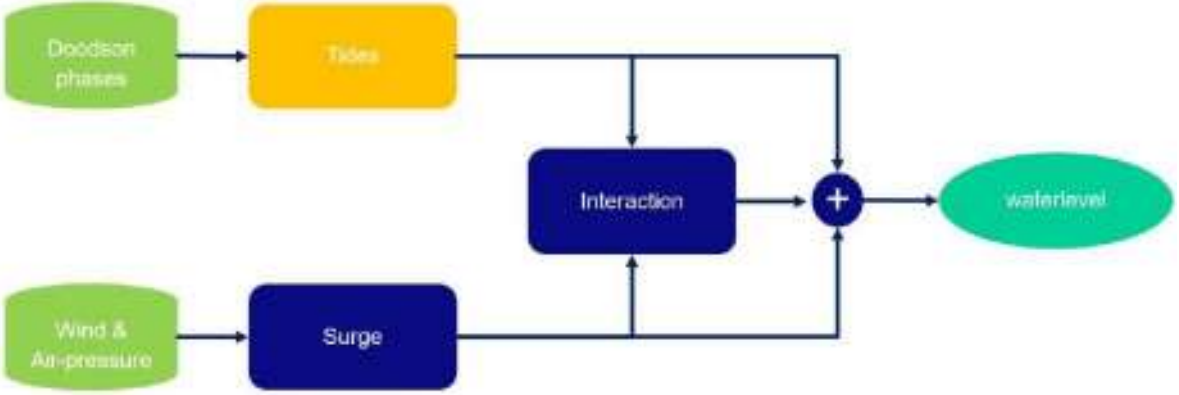
MINIO

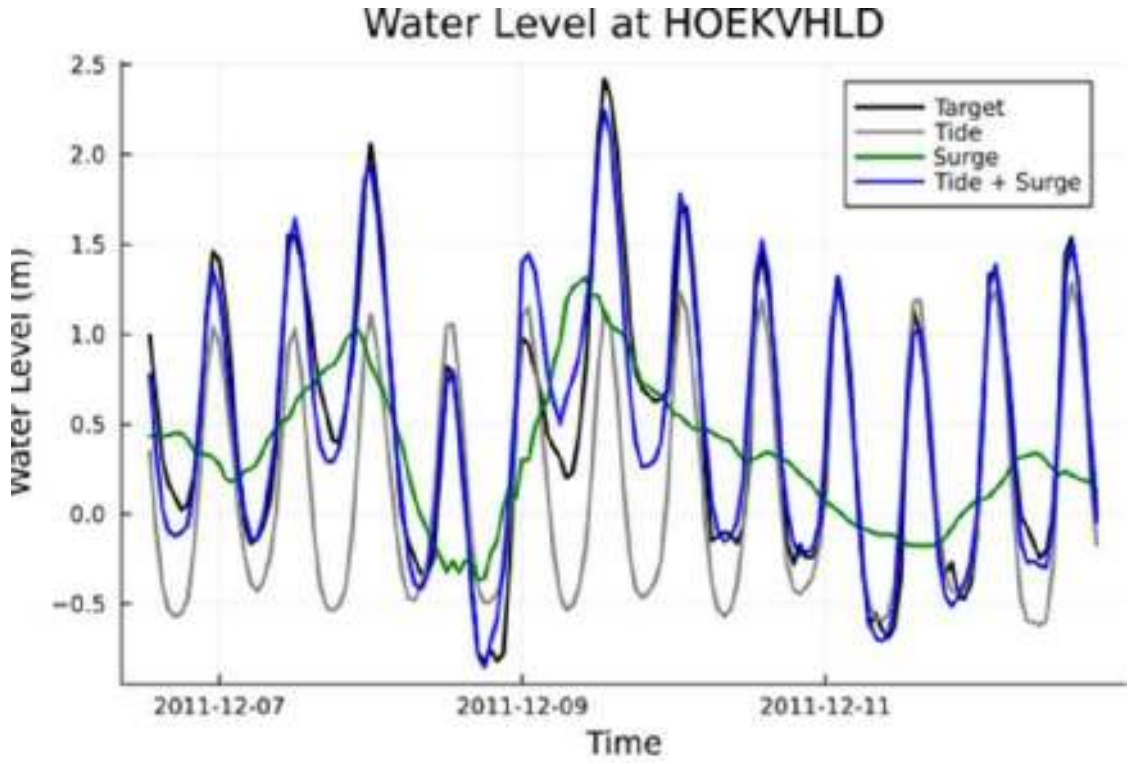
Deltares

North Sea: Timeseries model



Physics inspired components

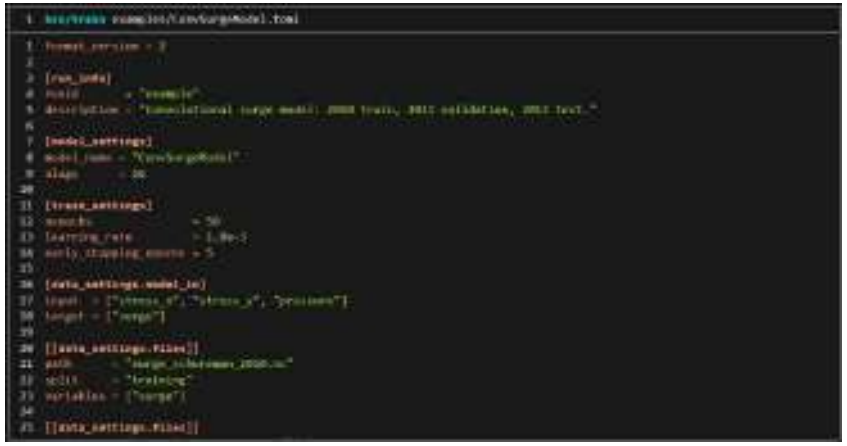




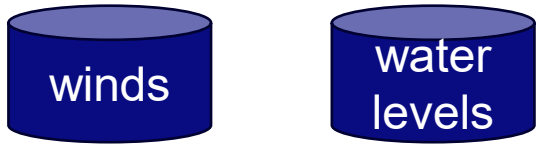
Deltares

Towards production software for a Timeseries model

Configuration



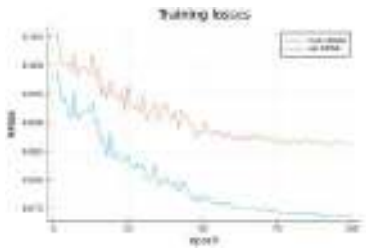
Input files with time-series

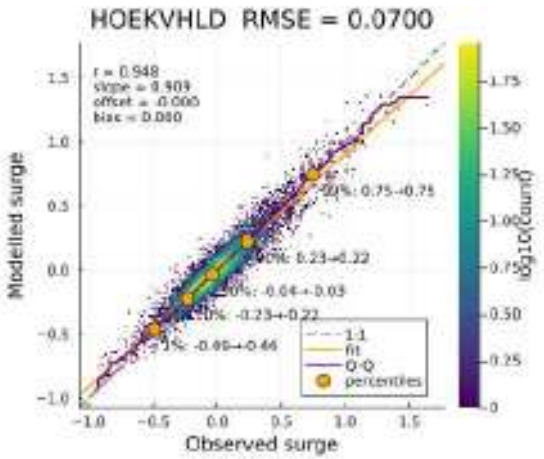


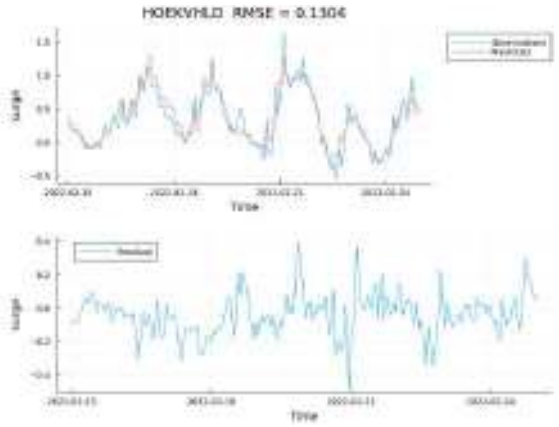
train script

trained model

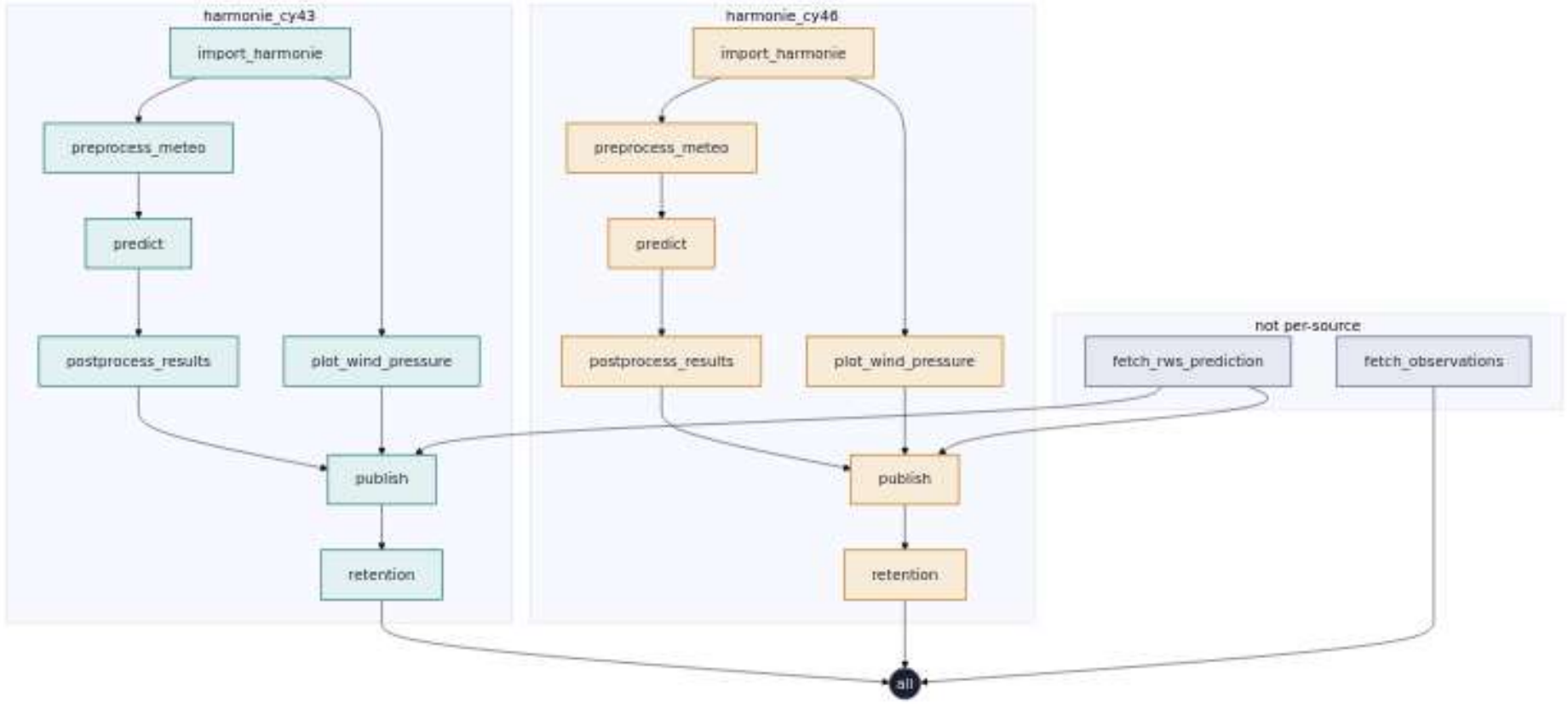
Requested output files

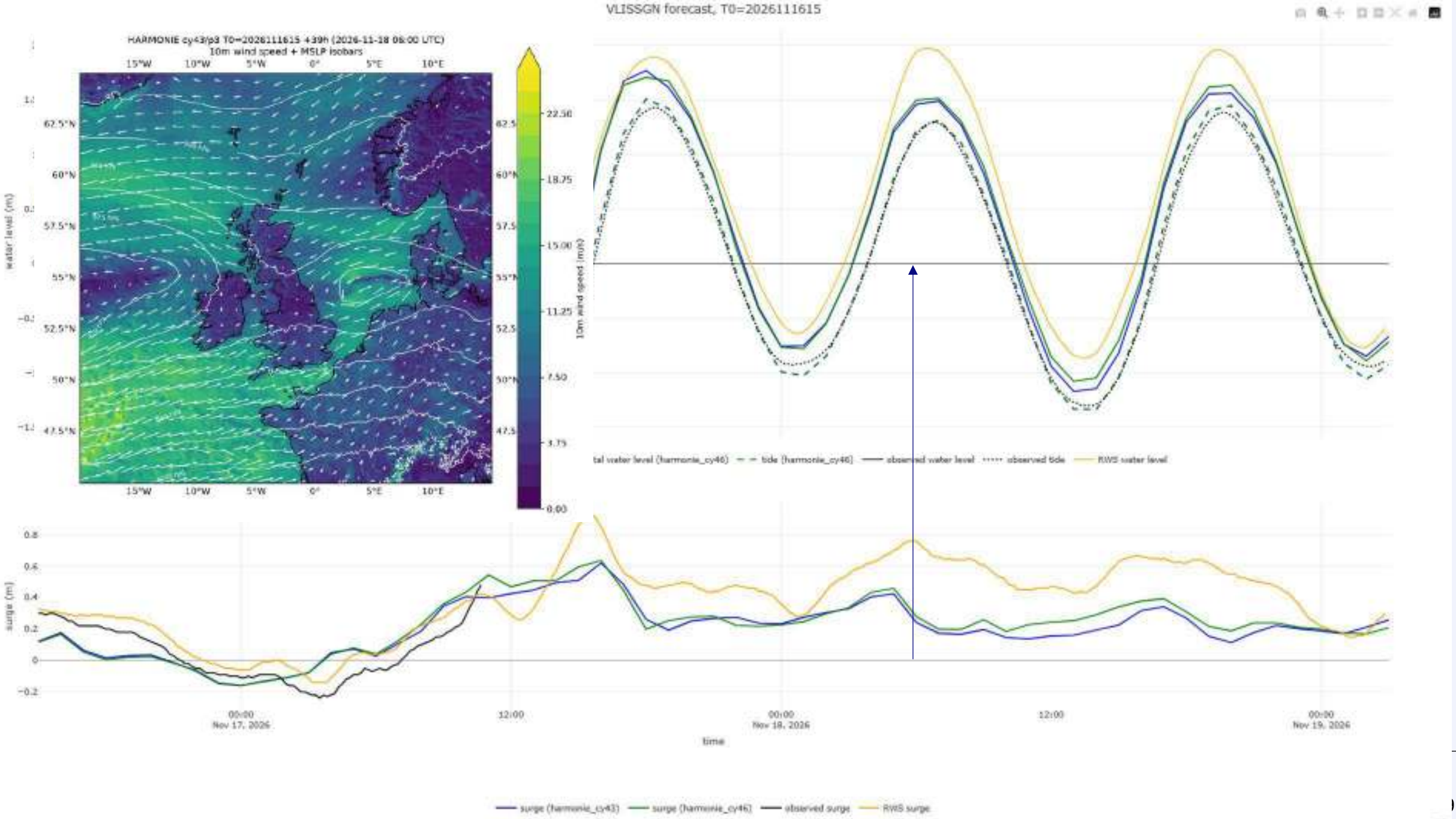






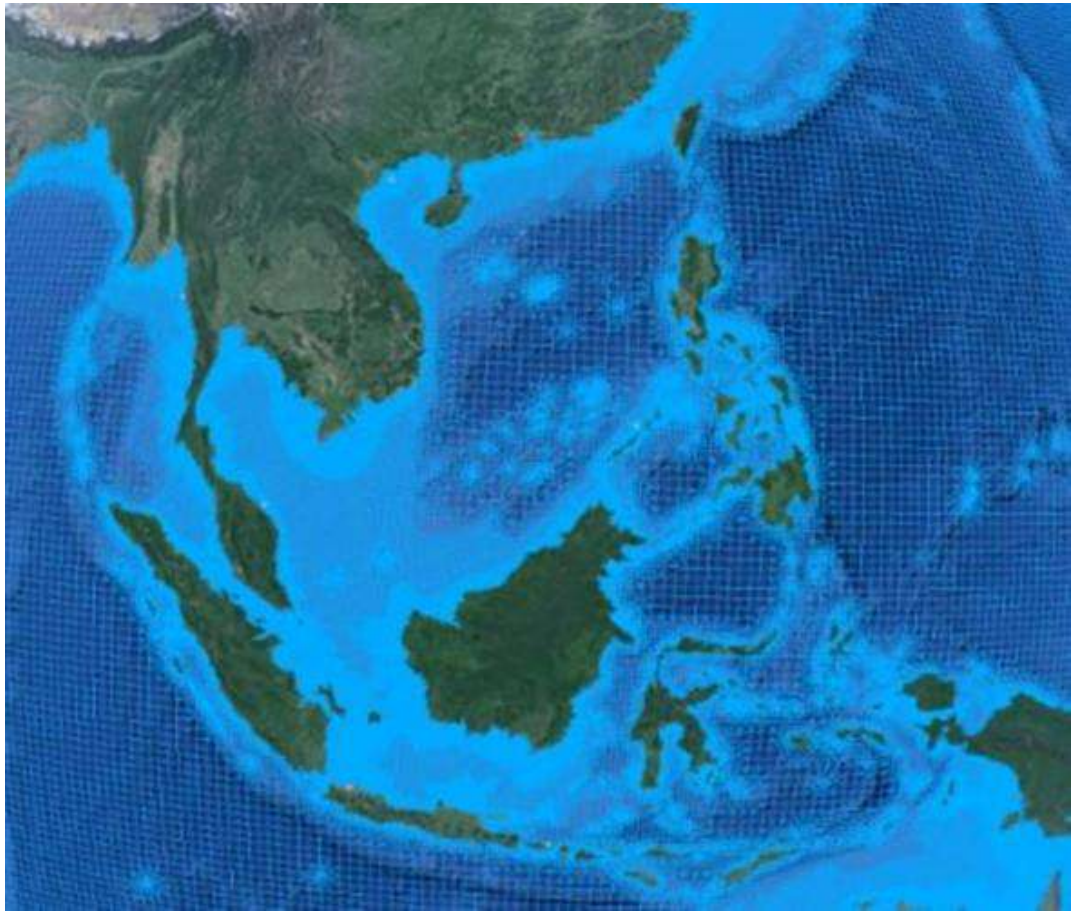
Towards real-time forecasting



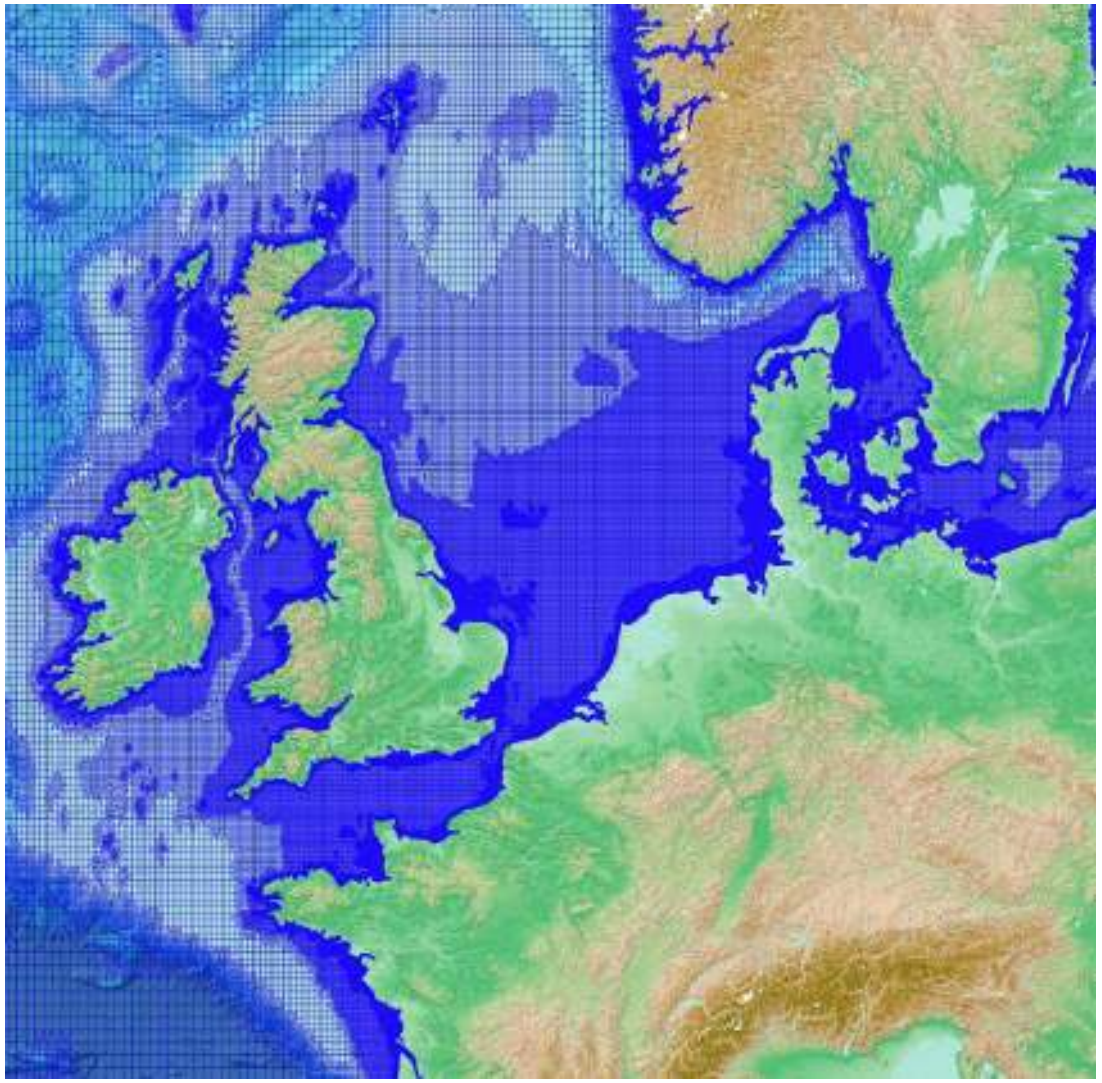

VLISSGN forecast, T0=2026111615
HARMONIE cy43/p3 T0=2026111615 +39h (2026-11-18 06:00 UTC)
10m wind speed + MSLP isobars
15°W
10°W
5°W
0°
5°E
10°E
62.5°N
60°N
57.5°N
55°N
52.5°N
50°N
47.5°N
22.50
18.75
15.00
11.25
7.50
3.75
0.00
water level (m)
10m wind speed (m/s)
tal water level (harmonie_cy46)
tide (harmonie_cy46)
observed water level
observed tide
RWS water level
surge (m)
0.8
0.6
0.4
0.2
0
−0.2
00:00
Nov 17, 2026
12:00
00:00
Nov 18, 2026
12:00
00:00
Nov 19, 2026
time
surge (harmonie_cy43)
surge (harmonie_cy46)
observed surge
RWS surge

FUTURA

- FUTURA EU project
  - Climate scenarios of storm surge by tropical cyclones for CMIP7
  - Based on same time series model
  - Trained with synthetic cyclones and GTSM



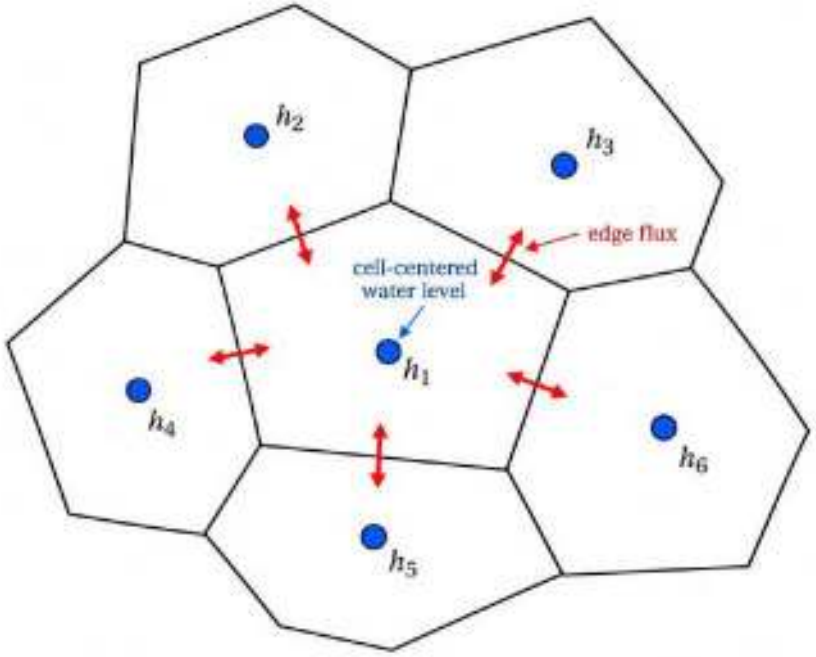
Delft3D Flexible Mesh unstructured grid



$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}\Big((H+h)u\Big) + \frac{\partial}{\partial y}\Big((H+h)v\Big) = 0,$$

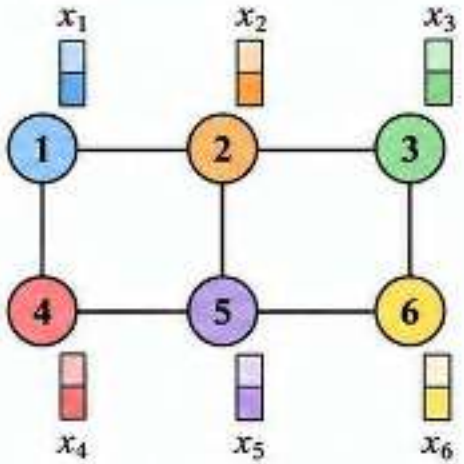
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv = -g\frac{\partial h}{\partial x} - ku + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -g\frac{\partial h}{\partial y} - kv + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$



1 Graph as input

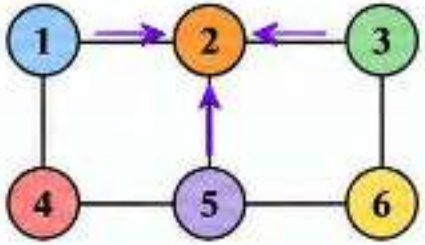
A graph consists of nodes and edges. Each node has an initial feature vector.



Nodes with initial features x_i

2 Message passing

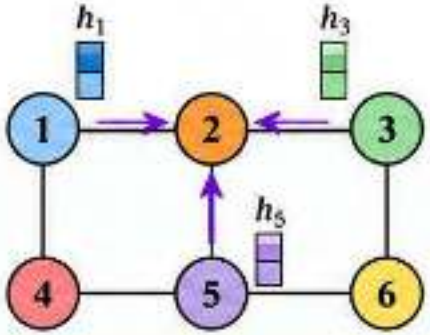
Each node collects messages from its neighbors based on their features (and optionally edge features).



Node 2 receives messages from nodes 1, 3, and 5.

3 Aggregation

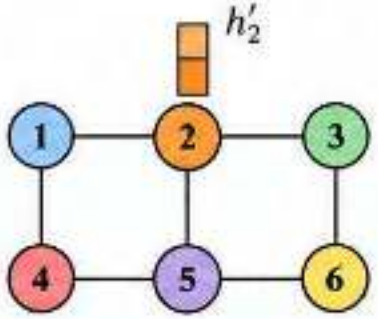
The messages from neighboring nodes are aggregated, e.g., by summing, averaging, or attention-weighted sum.



$m_2 = \mathrm{AGG}(h_1, h_3, h_5)$
(e.g., sum, mean, or attention)

4 Node update

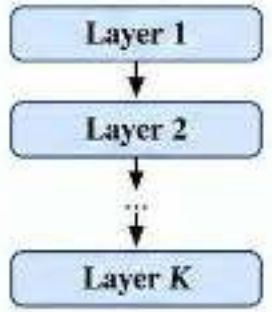
The node updates its feature using its previous feature and the aggregated message (via a neural network).



$h_2' = \mathrm{UPDATE}(h_2, m_2)$
(e.g., MLP, GRU, etc.)

5 Stack layers

Repeat message passing and node update for multiple layers ($k = 1, \ldots, K$). Each layer uses the updated node features.

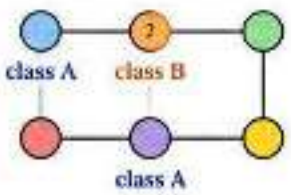


After K layers, each node has a representation that captures information from its K-hop neighborhood.

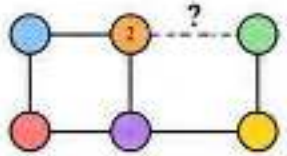
6 Output

The final node representations can be used for different tasks:

Node-level task
(e.g., classification)

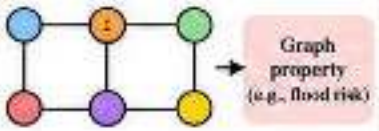


Edge-level task
(e.g., link prediction)

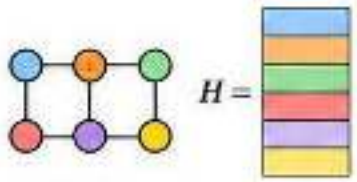


Predict if an edge exists between two nodes.

Graph-level task
(e.g., regression)



Node embeddings



Final node feature matrix
(size: $N \times d$)

Typical GNN layer (mathematical form)

For node i at layer k:

Aggregate messages from neighbors $\mathcal{N}(i)$ | Combine with own features | Neural network (e.g., MLP)

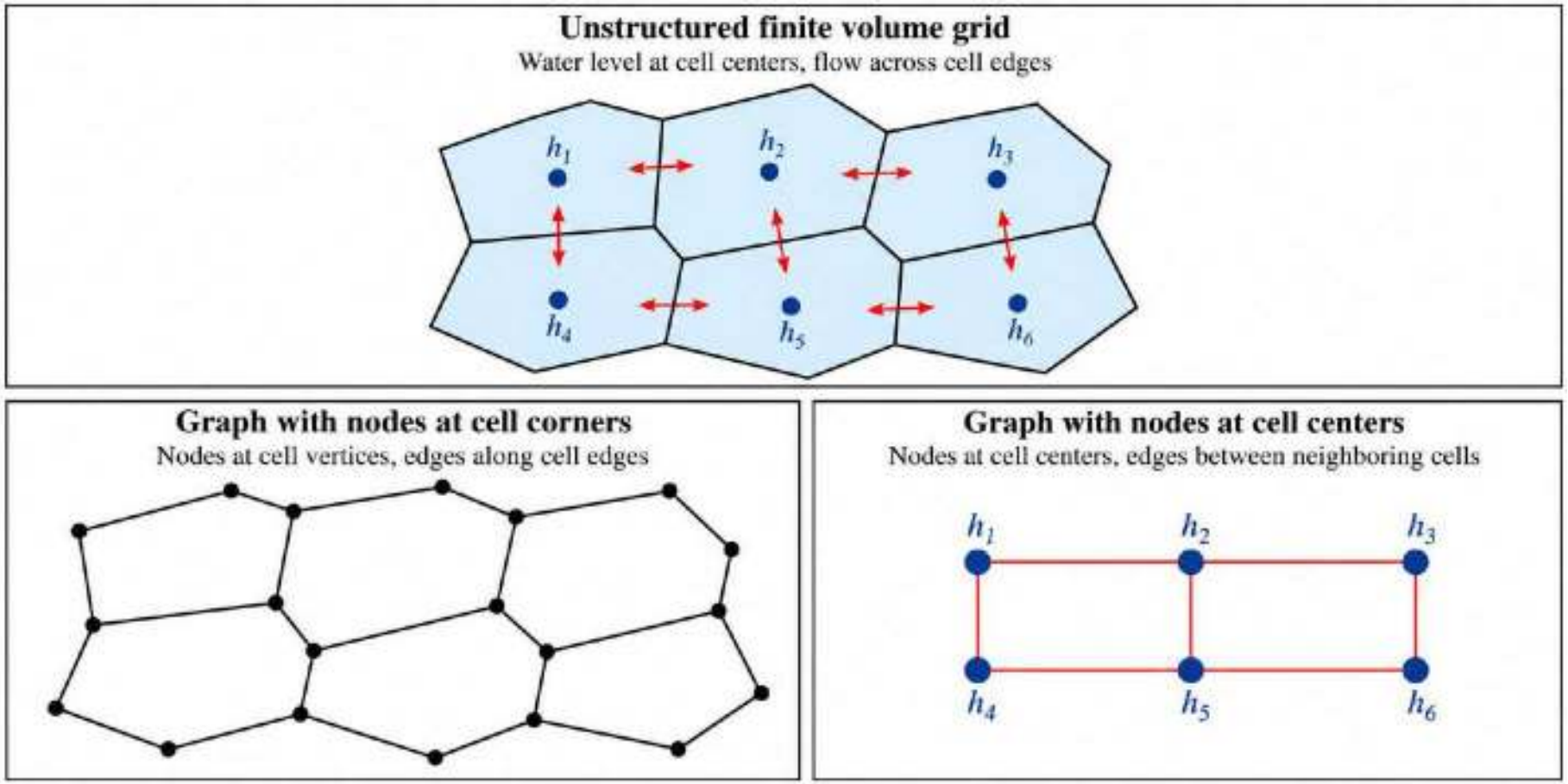
$$h_i^{(k+1)} = \phi\left(h_i^{(k)}, \mathrm{AGG}\left(\left\{h_j^{(k)} : j \in \mathcal{N}(i)\right\}\right)\right)$$

$h_i^{(k)}$: feature vector of node i at layer k
$\mathcal{N}(i)$: set of neighboring nodes of node i
AGG : aggregation function (sum, mean, max, attention, …)
ϕ : update function, typically a neural network (e.g., MLP + nonlinearity)

Deltares

Generated with chatgpt

Unstructured mesh to graph



Generated with chatgpt

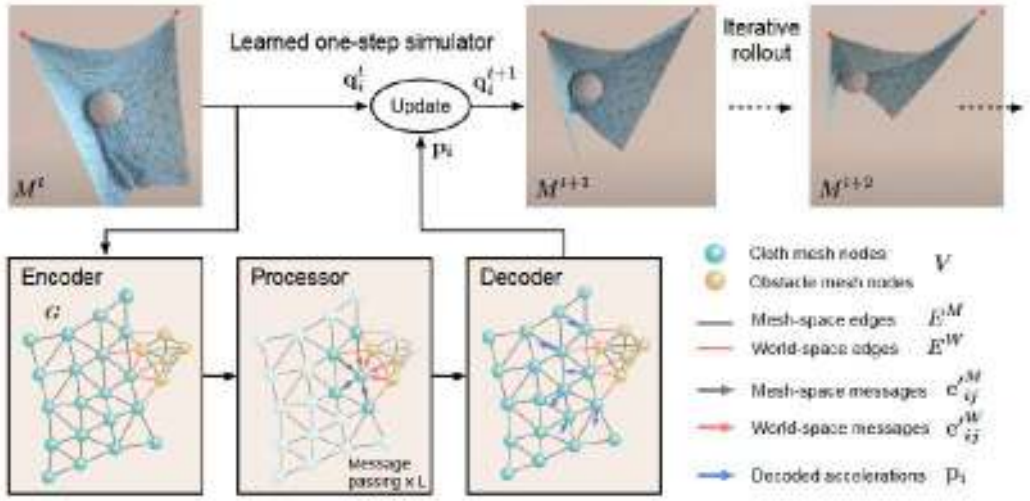
MeshGraphNet

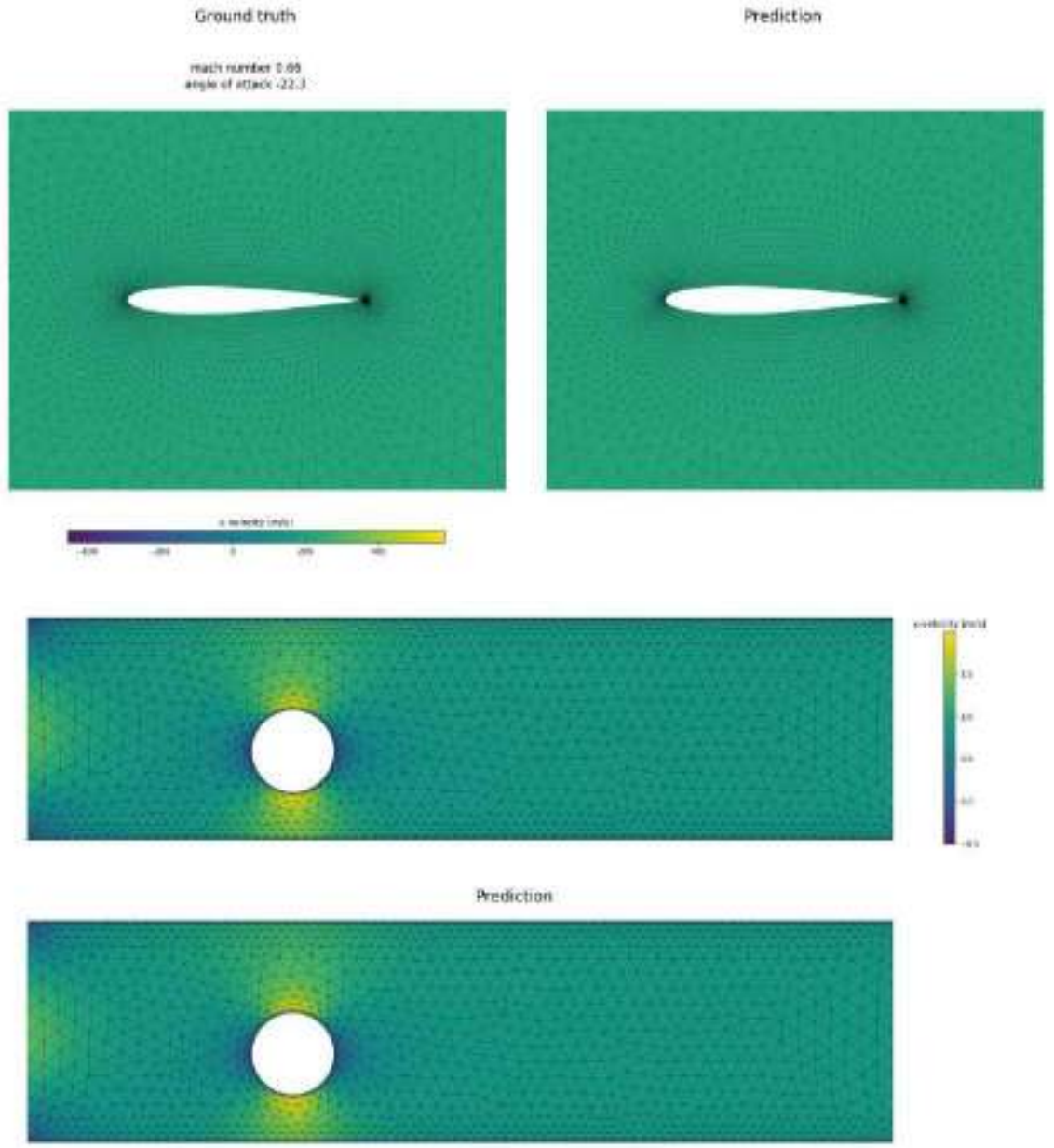
https://sites.google.com/view/meshgraphnets

Published as a conference paper at ICLR 2021

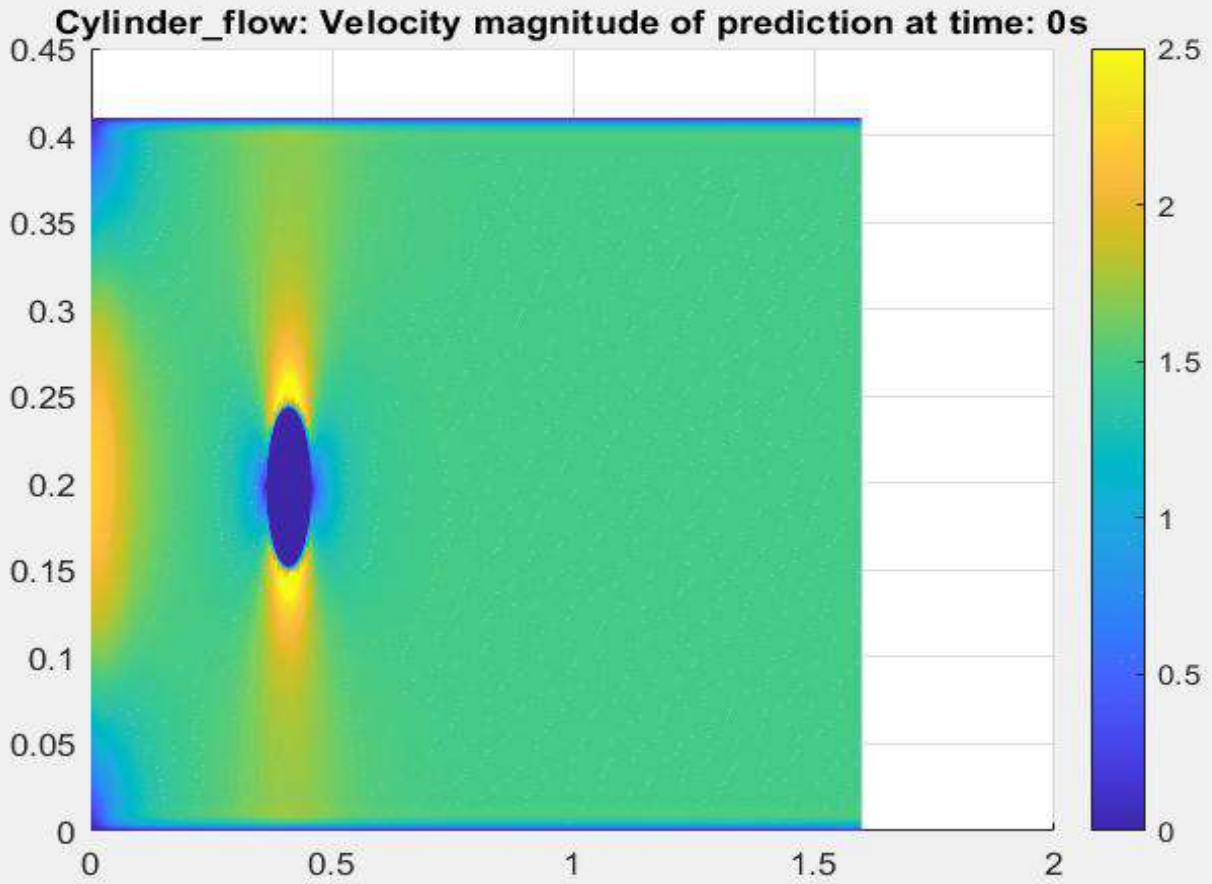
LEARNING MESH-BASED SIMULATION WITH GRAPH NETWORKS

Tobias Pfaff*, Meire Fortunato*, Alvaro Sanchez-Gonzalez, Peter W. Battaglia
Deepmind, London, UK
{tpfaff,meirefortunato,alvarosg,peterbattaglia}@google.com

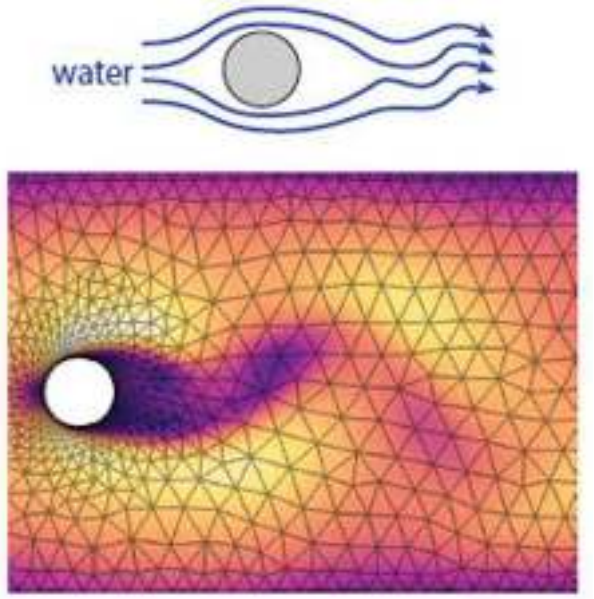




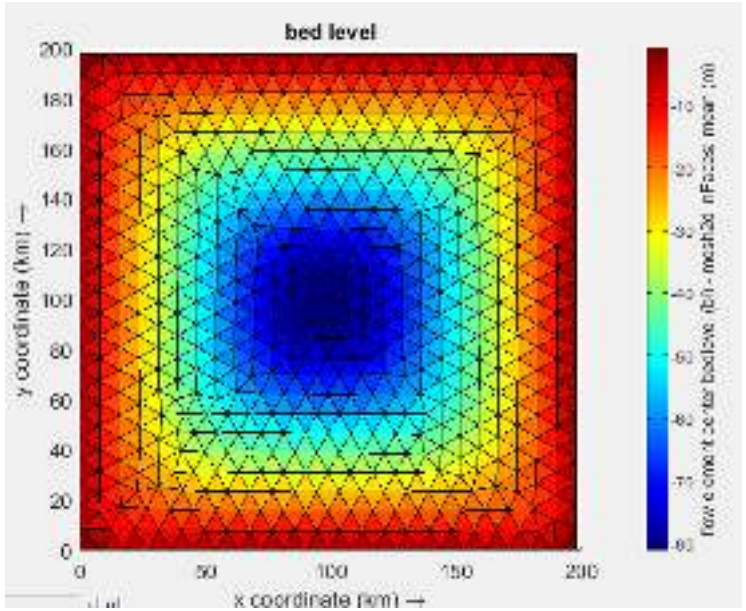
MeshGraphNets.jl

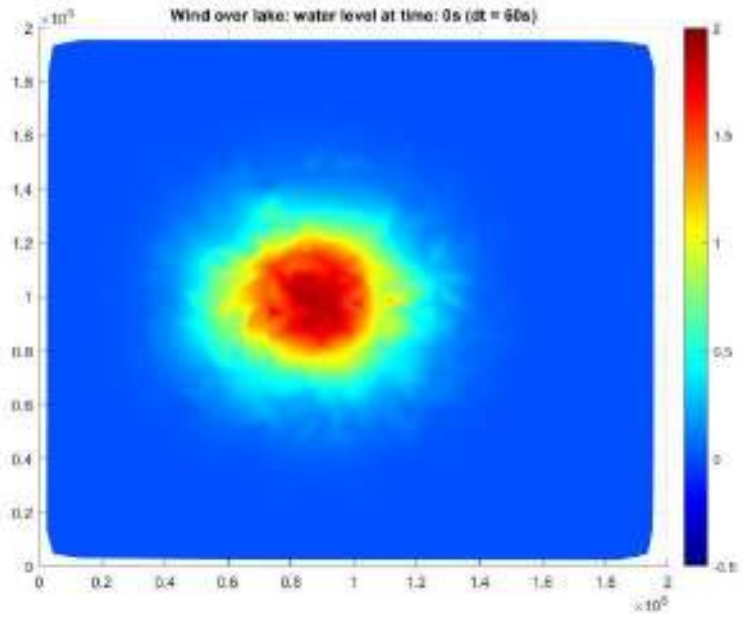


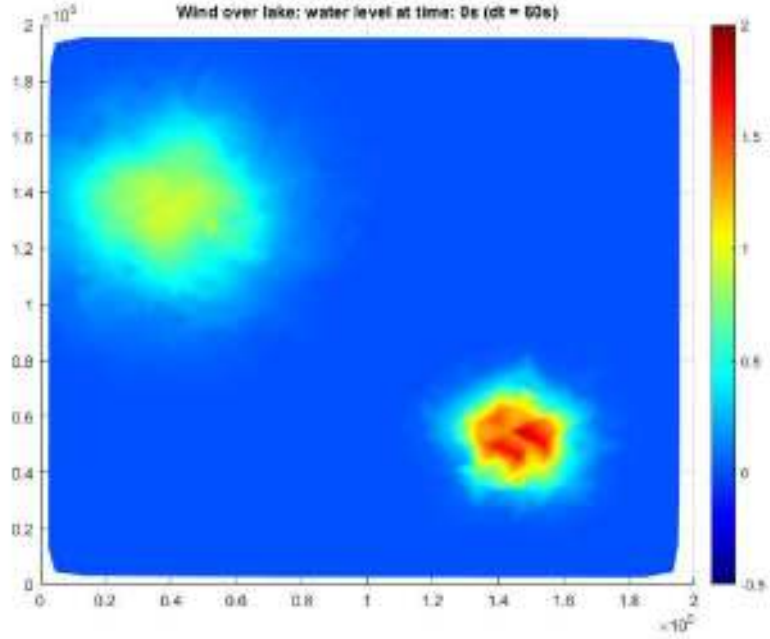
First steps to reproduce the original cylinder flow case.



Test case: Lake







Training data:

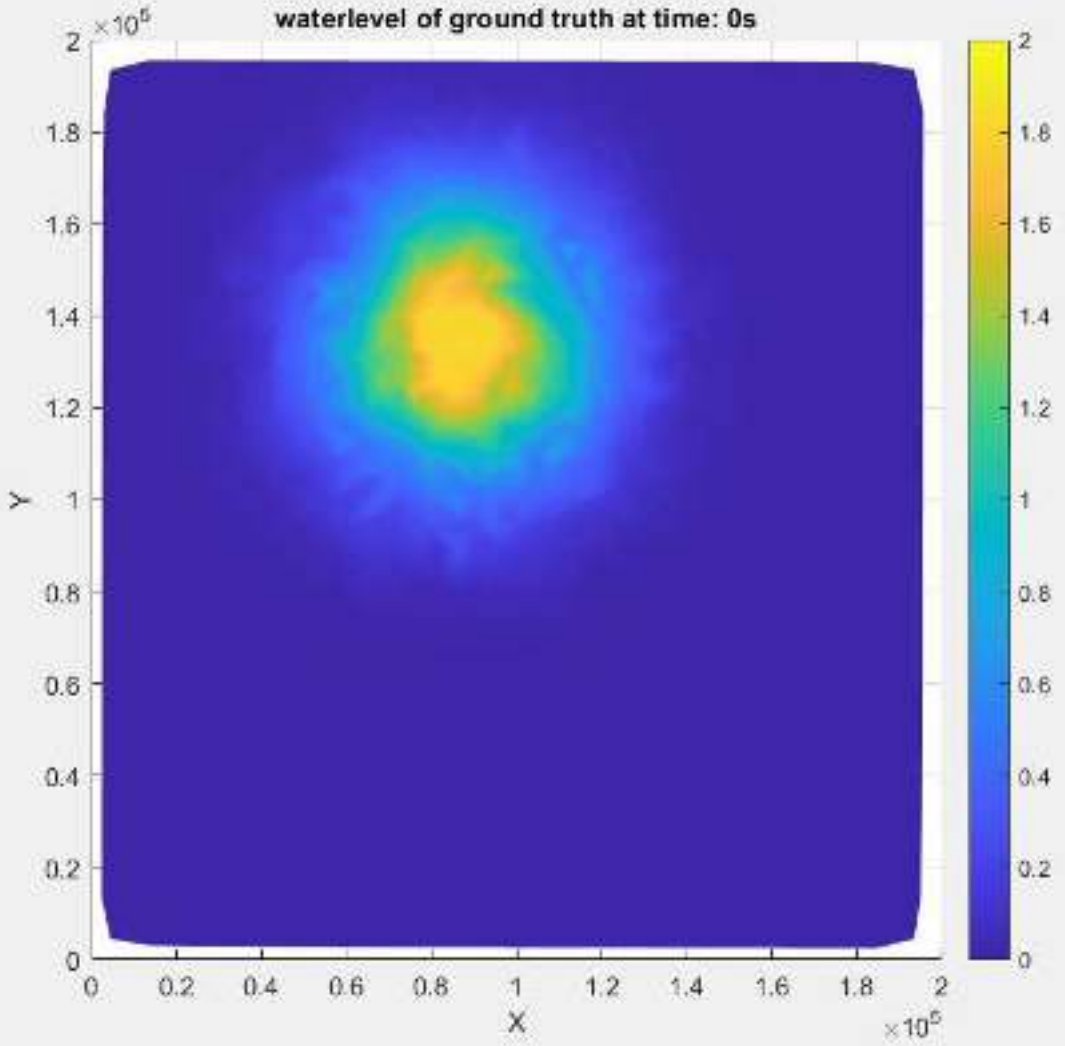
- 200 simulations with different initial water levels:
  - 1 – 100: 1 bump,
  - 101-200: 2 bumps:
- Each simulation:
  200min, dt = 1min

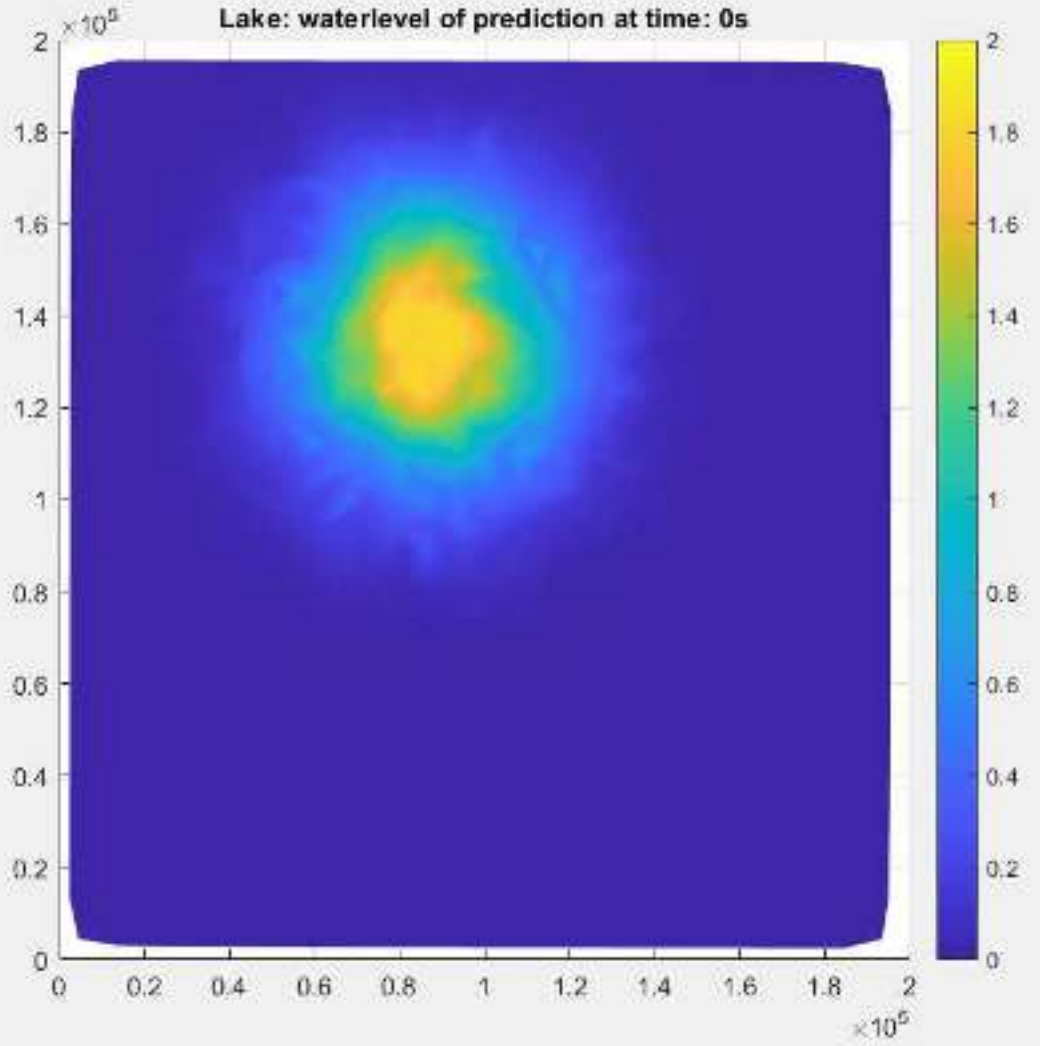
Data files	train	valid	test
Number of simulations	140	40	20

GNN model

- Message passing = 7
- Layer_size = 128
- hidden_layer = 2
- Batch = 1
- Noise: Gauss distributed, with standard deviation 0,02
- Adam optimizer
- Learning_rate: from 0.005 to 0,0001

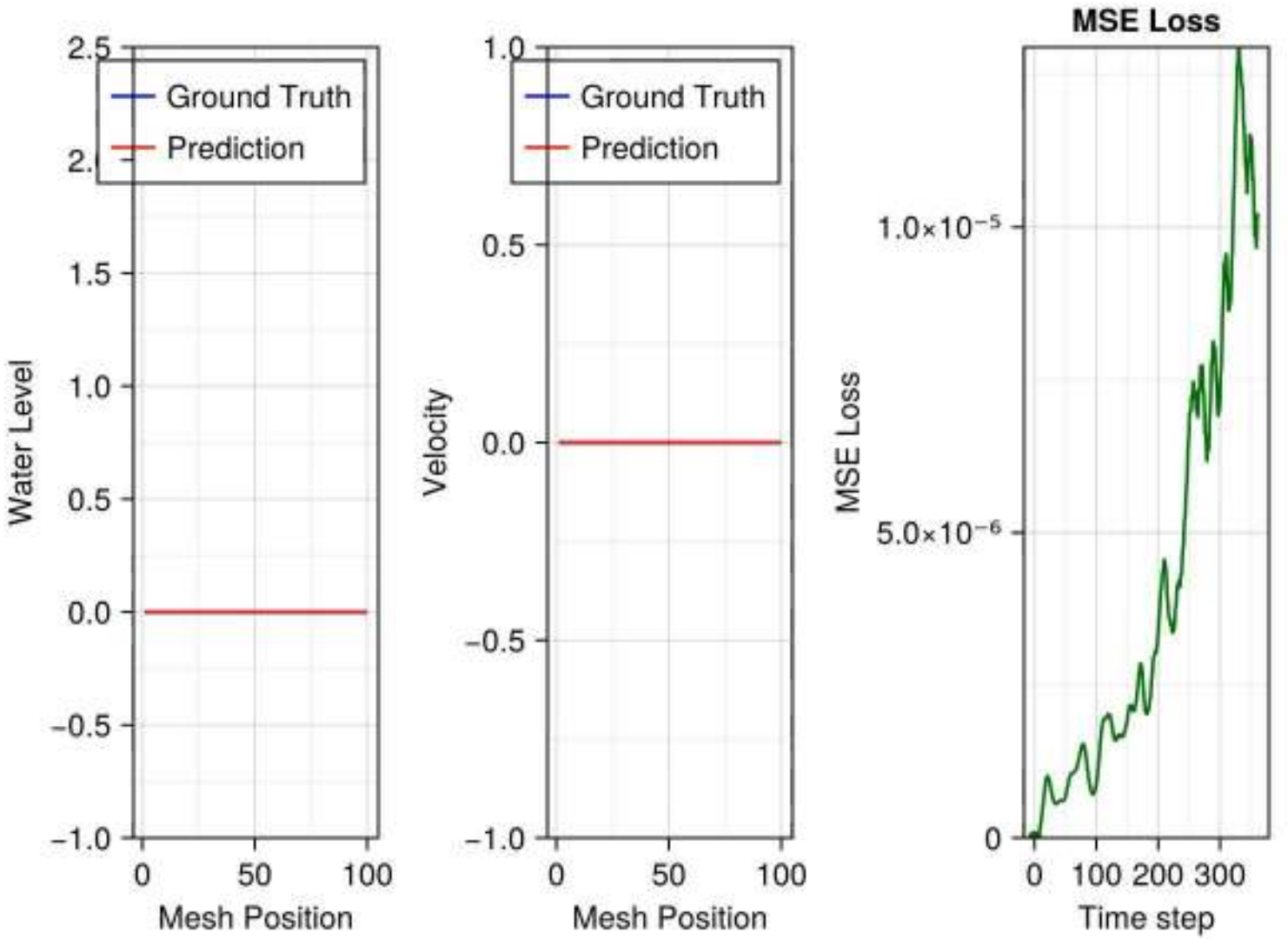
Test case: results (1)





Results for 1D case – wind forcing

- Finite volume type implementation of GNN based emulator



Student Projects

Student	Topic
Yilin Tang (TUD, Applied Maths)	Stability of emulator models
Mats Lenes (TUD, Applied Physics)	Including physics in emulator design
Sara Velez Fuente (TUD, Applied Maths)	Preprocessing ML models for (wind) bias correction
Davide Collella (TUD, Applied Maths)	ML model for controlled (lake) system
Augusto Brito Lima (TUD, Applied Maths)	Including observation data in emulator training
Job Kayhan (VU, AI)	GNN timeseries model for storm surges (FUTURA)
Charlotte Kauppinen (TUD, Applied Maths)	Equivariant variable scaling as an inductive bias
Jasmijn Dankelman (TUD, Applied Maths)	Stability bounds for AI surrogate models

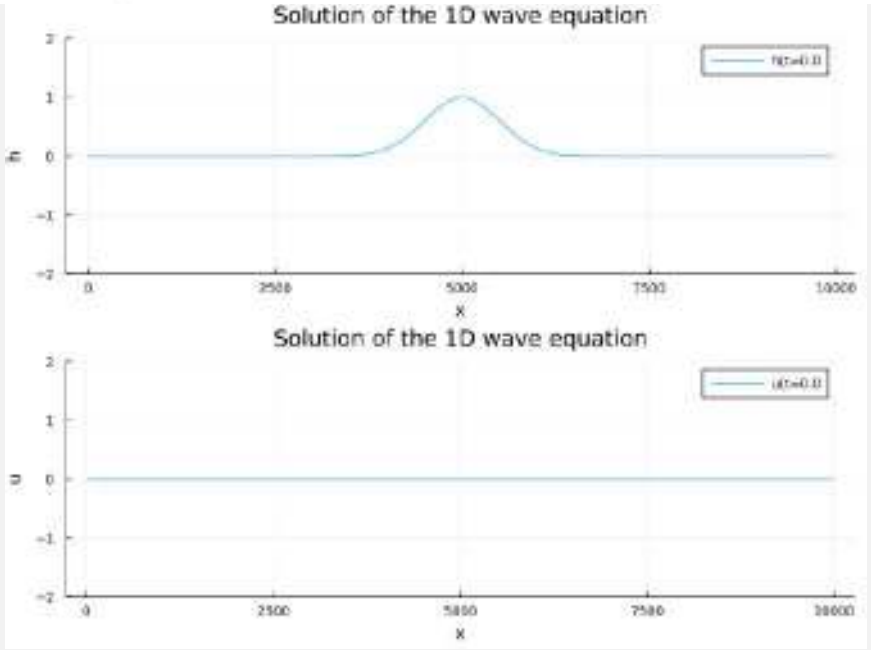
CFL condition

$$\frac{\partial h}{\partial t} + D\frac{\partial u}{\partial x} = 0$$
$$\frac{\partial u}{\partial t} + g\frac{\partial h}{\partial x} = 0$$

$$x \in [0, 10000], D = g = 10.$$

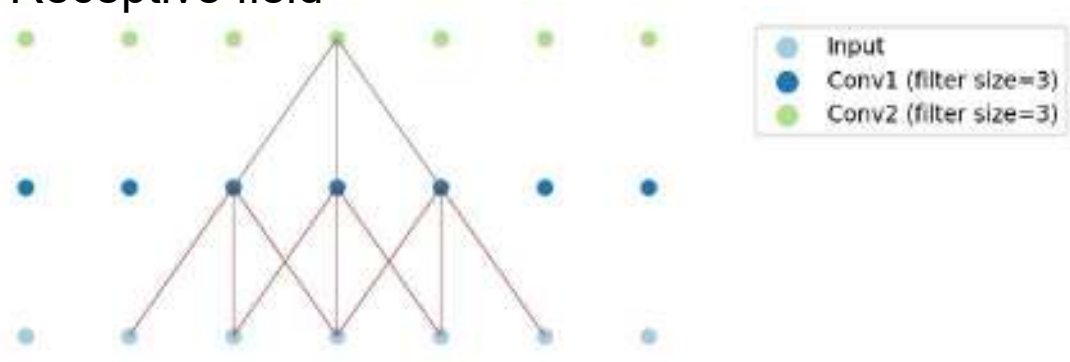
$$t \in [0, 2000], \Delta x = 50, \Delta t = 5.0$$

$$h(x, t = 0) = \exp(-\frac{(x - 0.5)^2}{2w^2}), \quad w = 0.03 : 0.01 : 0.2$$
$$u(x, t = 0) = 0$$



Wave equation

Receptive field



Stability experiments

n_layers\CNN time step	5	10	15	20	25	30
3			×	×	×	×
4				×	×	×
5				×	×	×
6					×	×

n_layers\CNN time step	5	10	15	20	25	30	35	40	45	50
3						×	×	×	×	×
4							×	×	×	×
5	×								×	×
6	×	×								×

Conclusion: CNN is also bound by CFL condition

Thanks for your attention.
Questions?



Deltares

Made with Midjourney