Oral presentation | Industrial applications

Industrial applications-I

Mon. Jul 15, 2024 4:30 PM - 6:30 PM Room B

[3-B-02] Leveraging Graph Neural Networks for the CFD Simulations of Bioreactors with Variable Geometries

*Umut Kaya[1,2], Srikanth Gopireddy[2], Nora Urbanetz[2], Ingmar Nopens[1], Jan Verwaeren[1] （1. Ghent University, 2. Daiichi Sankyo Europe GmbH）

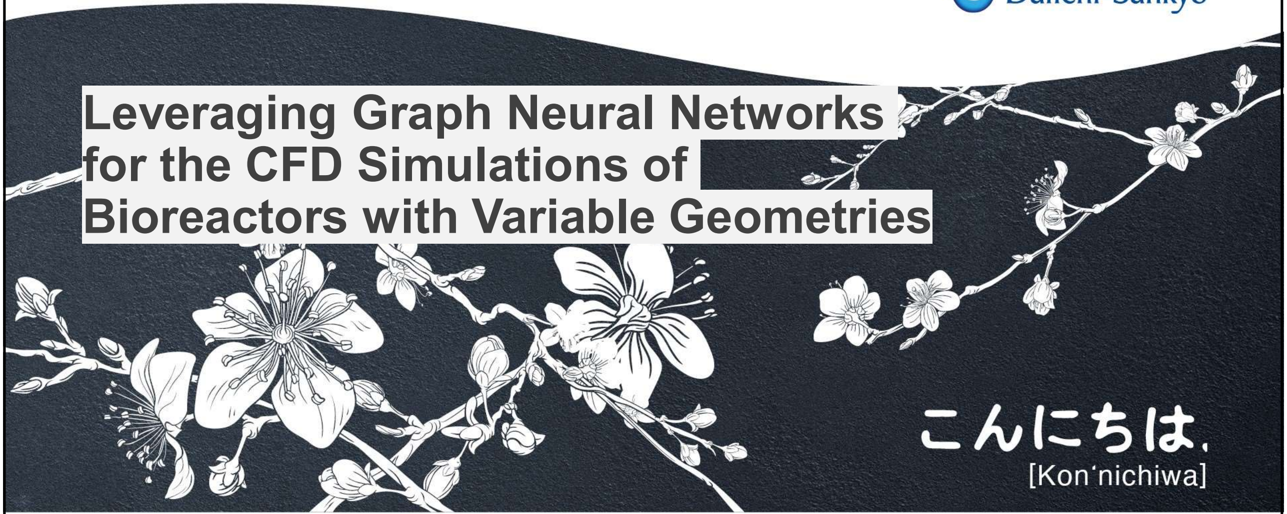
Keywords: Graph Neural Networks (GNNs), CFD, Bioreactor, Surrogate Modeling, Machine Learning

ICCFD12

Umut Kaya

Daiichi-Sankyo

Leveraging Graph Neural Networks for the CFD Simulations of Bioreactors with Variable Geometries



Passion for Innovation.
Compassion for Patients.™

UNIVERSITEIT GENT

Daiichi-Sankyo

Leveraging Graph Neural Networks for the CFD Simulations of Bioreactors with Variable Geometries

Umut Kaya*†, Srikanth Gopireddy*, Nora Urbanetz*, Ingmar Nopens†, Jan Verwaeren†

* Daiichi Sankyo Europe GmbH
umut.kaya@daiichi-sankyo.eu, srikanth.gopireddy@daiichi-sankyo.eu, nora.urbanetz@daiichi-sankyo.eu
† Ghent University
ingmar.nopens@ugent.be, jan.verwaeren@ugent.be

15.07.2024

Umut Kaya

This presentation is about

What?

Surrogate models for
CFD simulations of stirrer tanks

So What?

GNNs offer a natural approach
for regression on unstructured datasets

Now What?

Some insights on the future of these models

Pharma Background

Biopharmaceuticals

Biologics

- Large Molecules (~20.000)
- Complex production methods based on bioprocesses

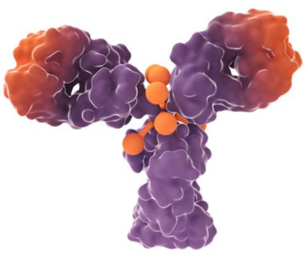


Image Source
enhertuhcp.com/en/mechanism-of-action

Traditional Drugs

- Small Molecules (~100 atoms)
- Complex production methods

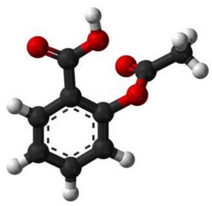


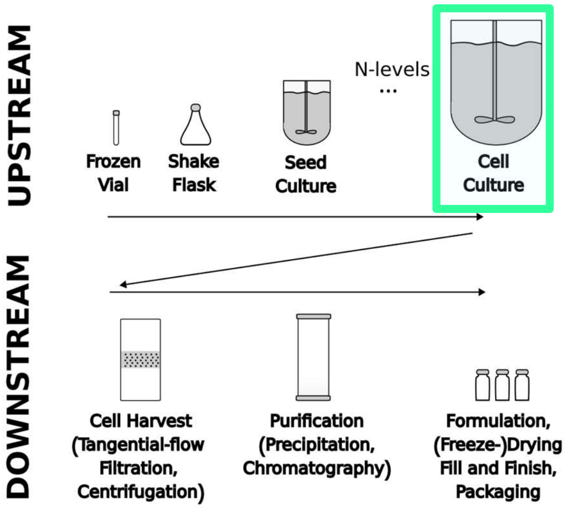
Image Source
en.wikipedia.org/wiki/Aspirin

Pharma Background
Biopharmaceuticals

Production of biopharmaceuticals
is a **complex process**

- **Upstream processes**
  produce the drug substance using living organisms, such as bacteria, yeast, or mammalian cells.

- **Downstream processes**
  extract the drug substance and package as the final drug product



Pharma Background
Design of Experiments

Process development
- ensures best process outcomes
- by finding optimal process inputs

Input

Process Parameters

Monitoring

Inline / Online Measurements

Output

Product Concentration
Quality Attributes

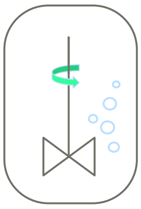
Image Source
merckmillipore.com/DE/de/product/Mobius-3L-Single-use-Bioreactor,MM_NF-C84539

Pharma Background
Design of Experiments

Process development
- ensures best process outcomes
- by finding optimal process inputs

Input

Process Parameters



Monitoring

Inline / Online Measurements

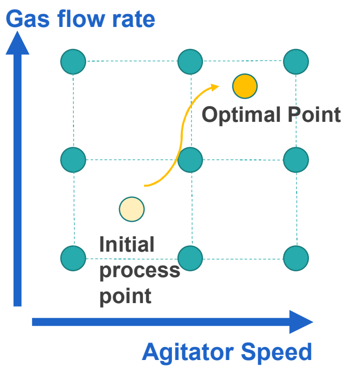
Output

Product Concentration
Quality Attributes

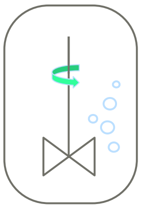
Pharma Background
Design of Experiments

Process development
- ensures best process outcomes
- by finding optimal process inputs

Experiments
- Wet-lab
- Dry-lab



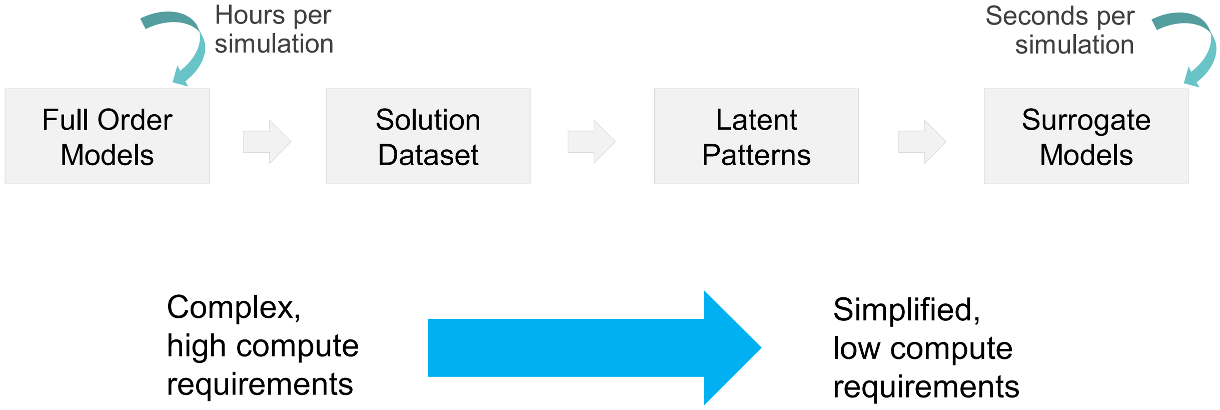
Input
- Gas flow rate
- Agitator speed
- Water level



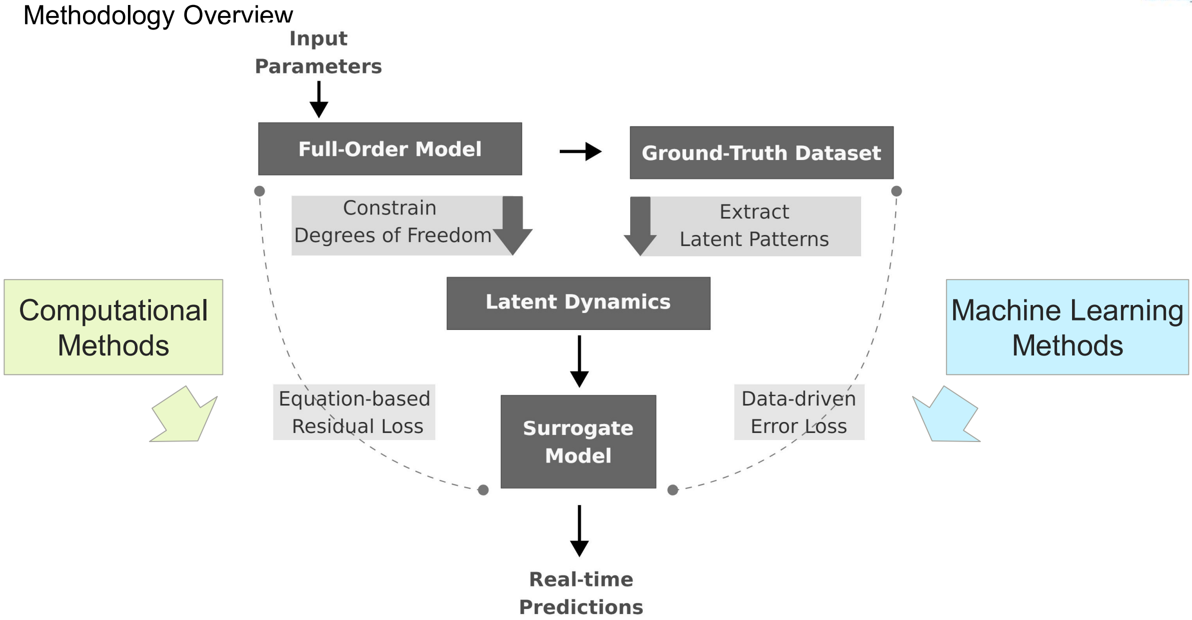
Monitoring
- O_2, CO_2, pH
- Cell density
- Hydrodynamic stress

Output
- Product titer
- Concentration of various substances

Surrogate Modeling
Methodology Overview



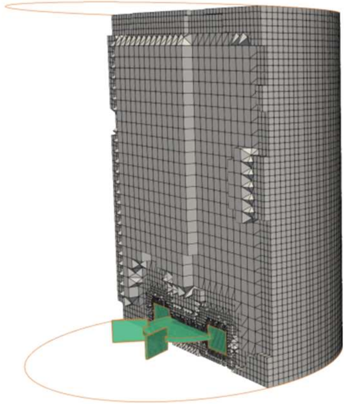
Surrogate Modeling
Methodology Overview



CFD Model
Governing Equations

- Steady-state NS
- k-ϵ turbulence model
- Multiple reference frame (MRF) for rotational boundary conditions

$$\nabla \cdot \mathbf{u} = 0$$
$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u}$$

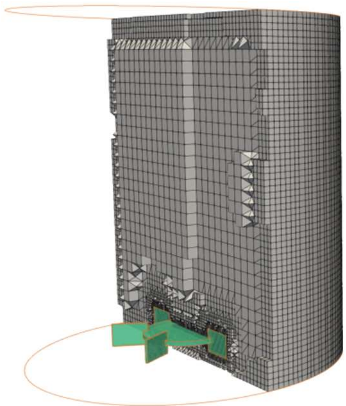


CFD Model
Dataset Preparation

The CFD simulation dataset of
121 simulations:
- 11 impeller speed: 100-200 rpm
- 11 impeller height: 2-3 cm

Test data (9%):
- 150rpm is used as the test data for all impeller heights, **11 test points**

$$\nabla \cdot \mathbf{u} = 0$$
$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u}$$

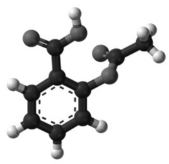


Graph Neural Networks

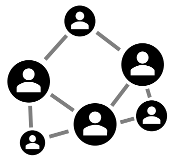
Graph Datasets

Graph structure is common to many datasets:

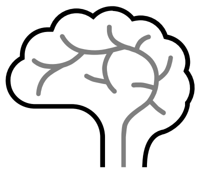
Maps



Molecules



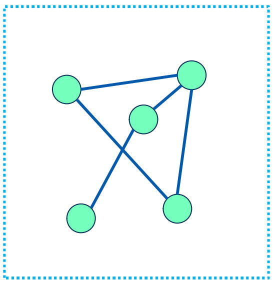
Social Networks



Brain Connectome

Graph Neural Networks

Graphs

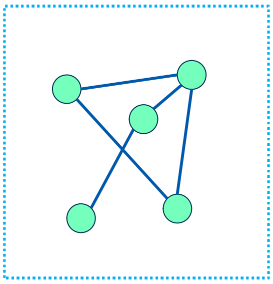


Nodes (Vertices)

Edges (Links)

Graph

Graph Neural Networks
Graphs



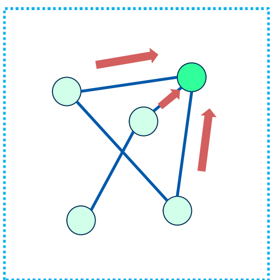
Node Features

Edge Features

Global Features

Graph Neural Networks
Feature Learning

Message Passing



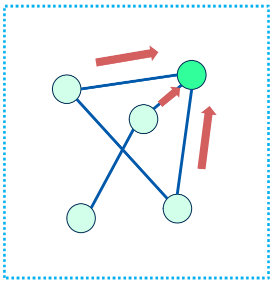
Node Features

Edge Features

Global Features

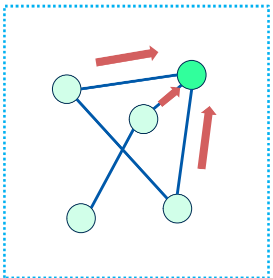
Graph Neural Networks
Feature Learning

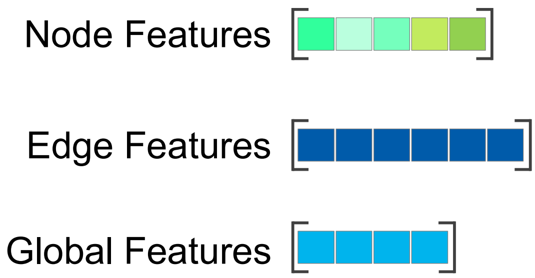
Message Passing



Graph Neural Networks
Feature Learning

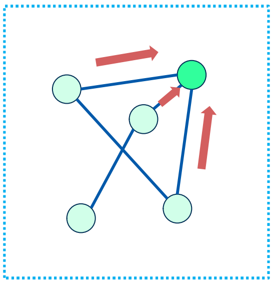
Message Passing





Graph Neural Networks
Feature Learning

Message Passing



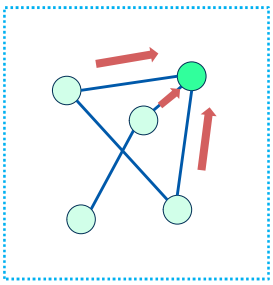
Node Features

Edge Features

Global Features

Graph Neural Networks
Learning Tasks

Message Passing



Node Features
→ Node regression
→ Node classification

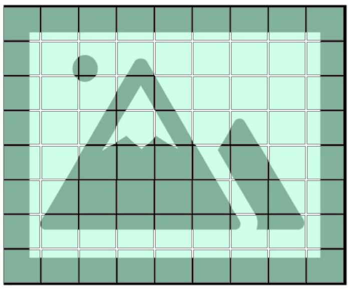
Edge Features
→ Edge Prediction

Global Features
→ Graph classification

Graph Neural Networks
Convolution

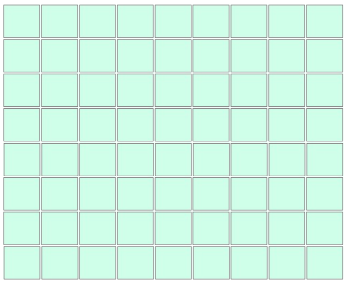
- Image pixels

Graph Neural Networks
Convolution



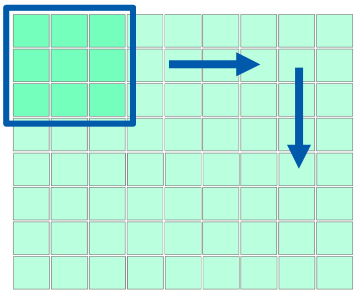
- Image pixels

Graph Neural Networks
Convolution



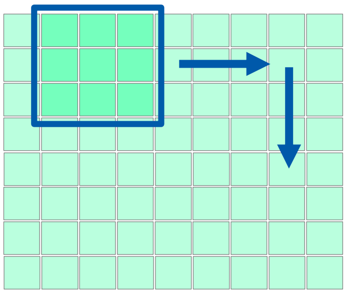
- Image pixels

Graph Neural Networks
Convolution



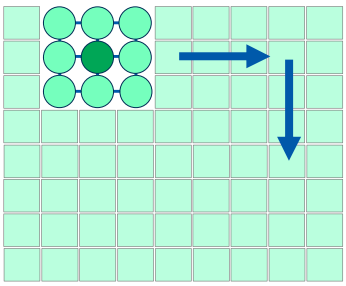
- Image pixels
- Convolutional filters

Graph Neural Networks
Convolution



- Image pixels
- Convolutional filters

Graph Neural Networks
Convolution

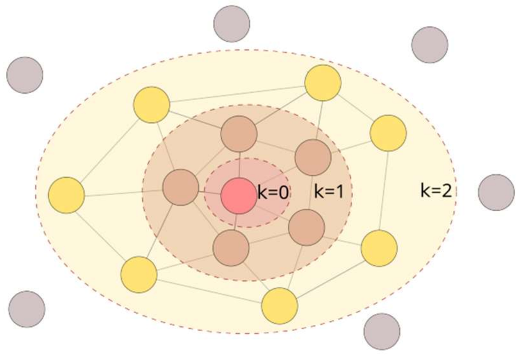


- Image pixels
- Convolutional filters
- Can be extended to graphs

Graph Neural Networks
Convolution

Graph Convolutional Networks (GCN)

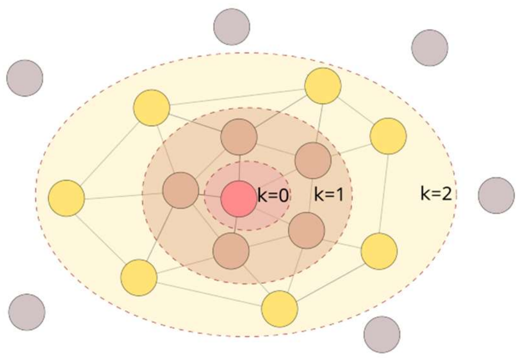
- Each convolutional layer process message from k-neighbourhood



Graph Neural Networks
Convolution

GNN layer

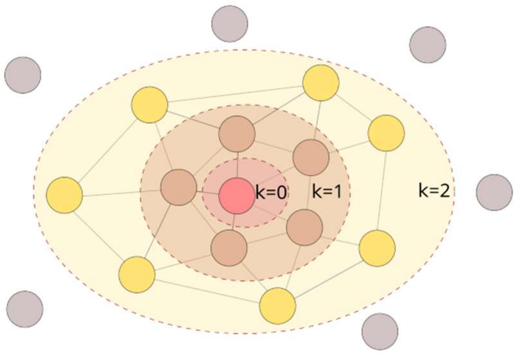
$$x^{(l+1)} = f(x^{(l)}, A)$$



Graph Neural Networks
Convolution

GCN layer

$$x'_i = \sigma\left(\sum_{j\in\mathcal{N}_i\cup\{i\}} \frac{e_{i,j}}{\sqrt{d_i d_j}} W^{(l)} x_j\right)$$



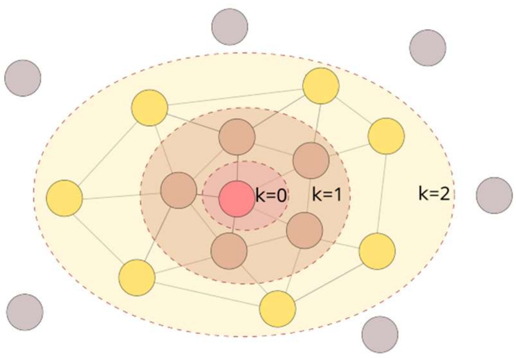
Graph Neural Networks
Convolution

GCN layer

$$x'_i = \sigma\left(\sum_{j\in\mathcal{N}_i\cup\{i\}} \frac{e_{i,j}}{\sqrt{d_i d_j}} W^{(l)} x_j\right)$$

Edges

Nodes



Graph Neural Networks
Convolution

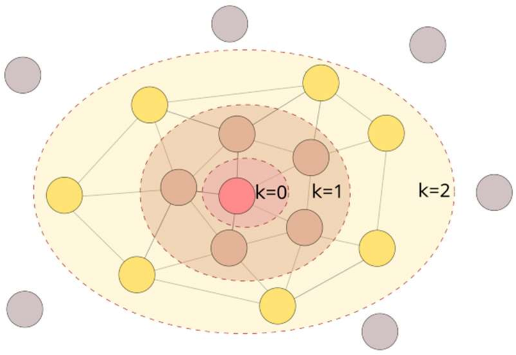
GCN layer

$$x_i' = \sigma\left(\sum_{j\in\mathcal{N}_i\cup\{i\}} \frac{e_{i,j}}{\sqrt{d_i d_j}} W x_j\right)$$

Edges

Learnable Weights

Nodes

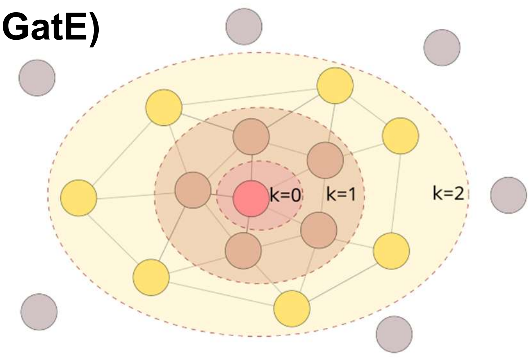


Graph Neural Networks
Convolution

GraphSAGE (SAmple and aggreGatE)

$$x_i' = W_1 x_i + W_2 \cdot \mathrm{mean}_{j\in\mathcal{N}(i)}(x_j)$$

& Neighborhood sampling

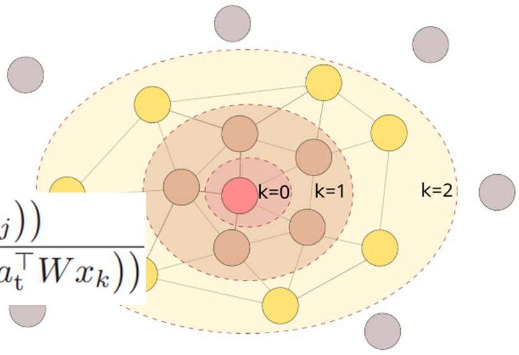


Graph Neural Networks
Convolution

Graph Attention Network (GAT)

$$x_i' = \alpha_{i,i} W x_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} W x_j$$

$$\alpha_{i,j} = \frac{\exp\left(\sigma\left(a_s^\top W x_i + a_t^\top W x_j\right)\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(\sigma\left(a_s^\top W x_i + a_t^\top W x_k\right)\right)}$$

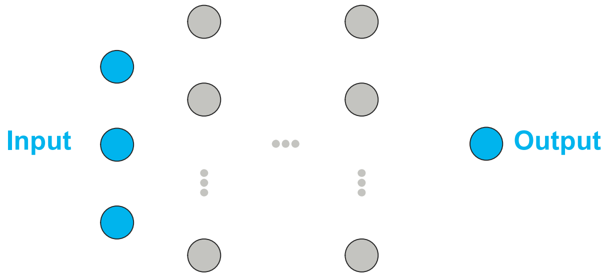


- allow weights to depend on the inputs

Graph Neural Networks
Convolution

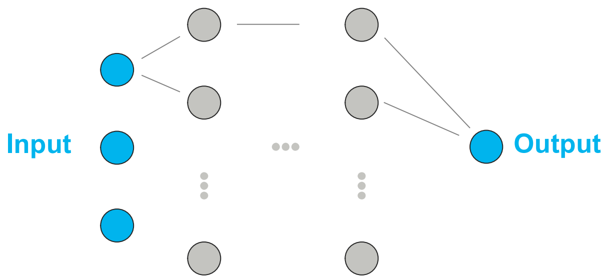
Multilayer perceptron (MLP)

- Node-to-node regression
- Works as a baseline, as well as final layer in our approach



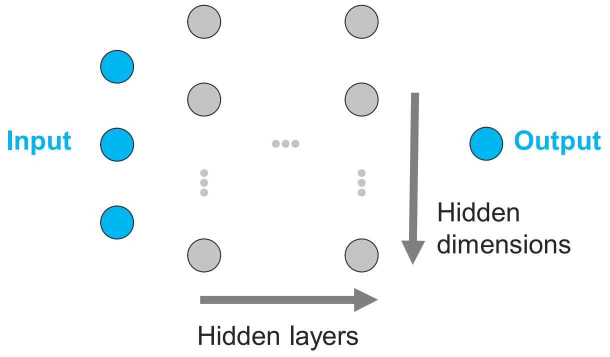
Graph Neural Networks
Convolution

Multilayer perceptron (MLP)
- Node-to-node regression
- Works as a baseline, as well as final layer in our approach

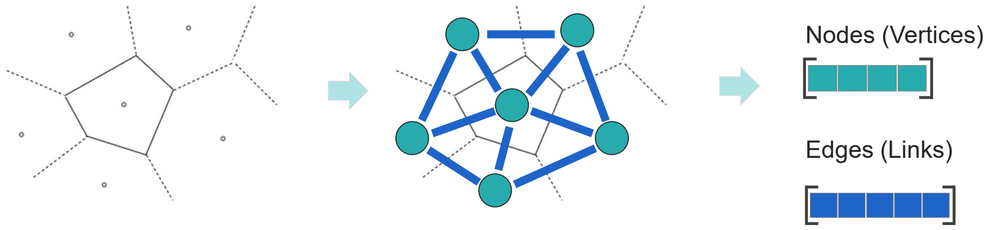


Graph Neural Networks
Convolution

Multilayer perceptron (MLP)
- Node-to-node regression
- Works as a baseline, as well as final layer in our approach

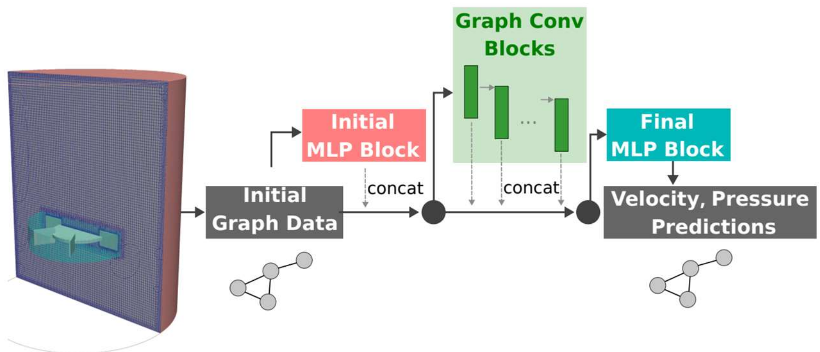


CFD Dataset as a Graph



GNN Model Architecture

- Outputs of GCN layer blocks are **concatenated**
- Each output acts as a **new feature**
- A final MLP layer used for **regression**



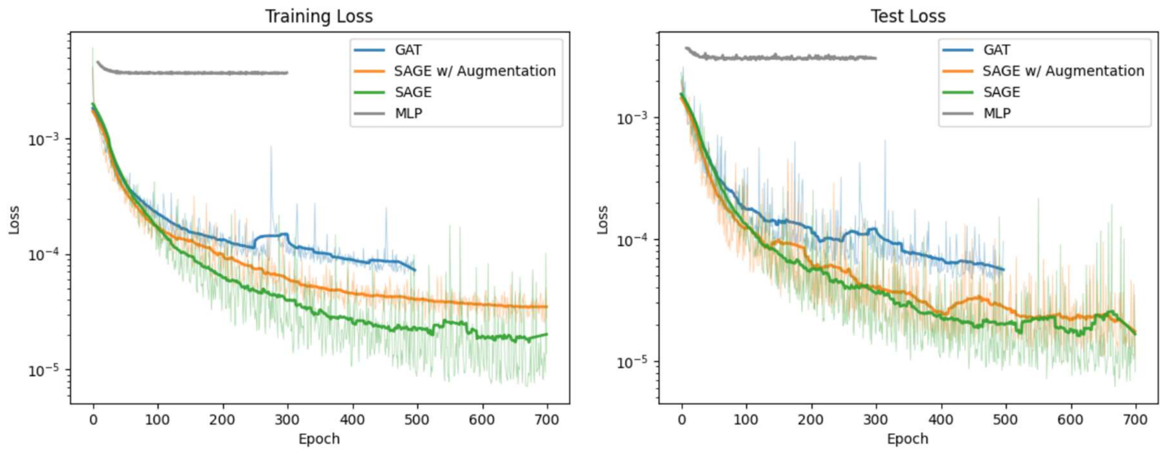
GNN Model Architecture

Hypertuning and size

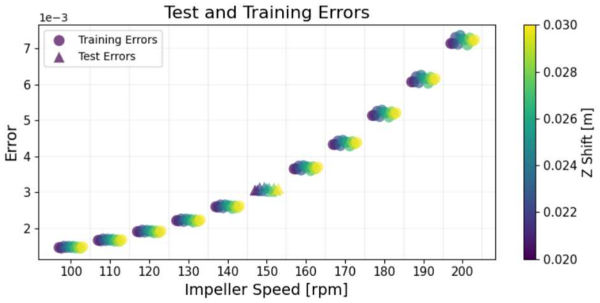
- Long training iterations ~1day → **limited grid search**

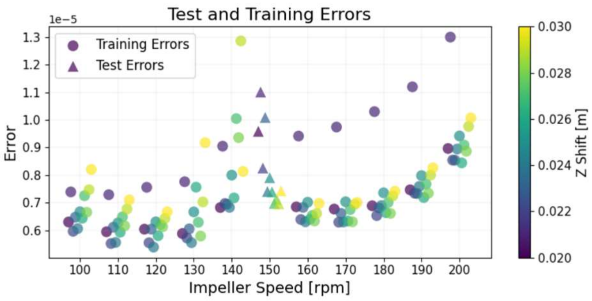
MLP layer: 2 hidden layers with 32 hidden dimension	1.4K parameters 0.15 hours for 300 epochs
GraphSAGE: 20 GCN blocks with 1 layer, 512 hidden dims. Final MLP, 2 hidden layers with 256 hidden dims.	4.8M parameters, 20 hours for 700 epochs, 46GB memory consumption
GAT: 6 GCN blocks with 1 layer, 32 hidden dims, 8 attention heads. Final MLP, 2 hidden layers with 64 hidden dims.	441K parameters, 36 hours for 500 epochs, 46GB memory consumption

Results: Training and Test Loss Curves

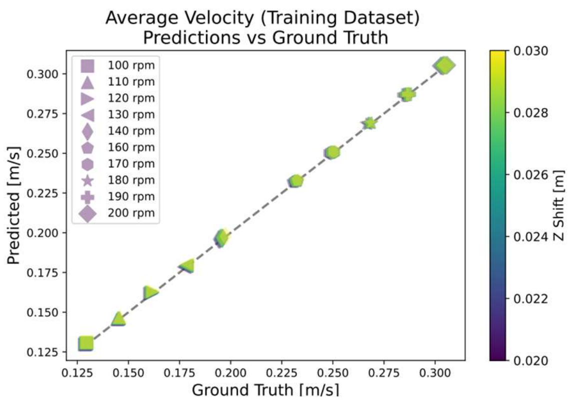


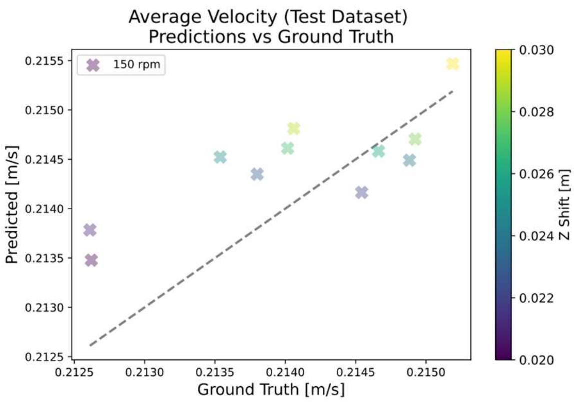
Results: Training and Test Loss





Results: Prediction Accuracy

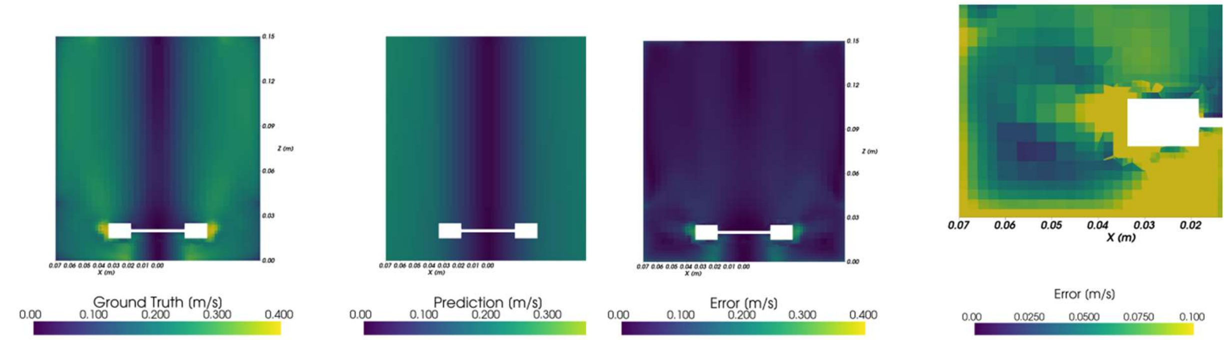




Visual Comparison of Velocity Fields

Test Velocity Predictions

MLP-only

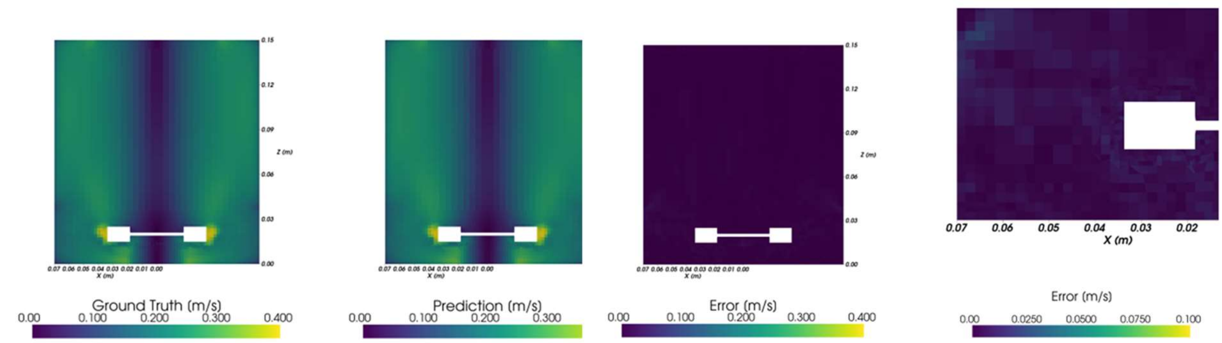


(a) Ground Truth (b) Prediction (c) Error (d) Zoom error with smaller color scale

Visual Comparison of Velocity Fields

Test Velocity Predictions

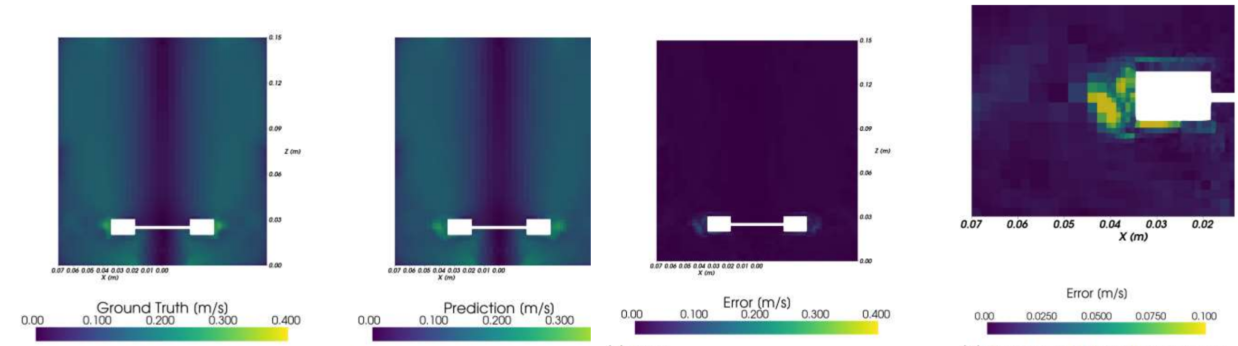
GraphSAGE



(a) Ground Truth (b) Prediction (c) Error (d) Zoom error with smaller color scale

Visual Comparison of Velocity Fields
Extrapolated Test Cases

Large-impeller (10%)



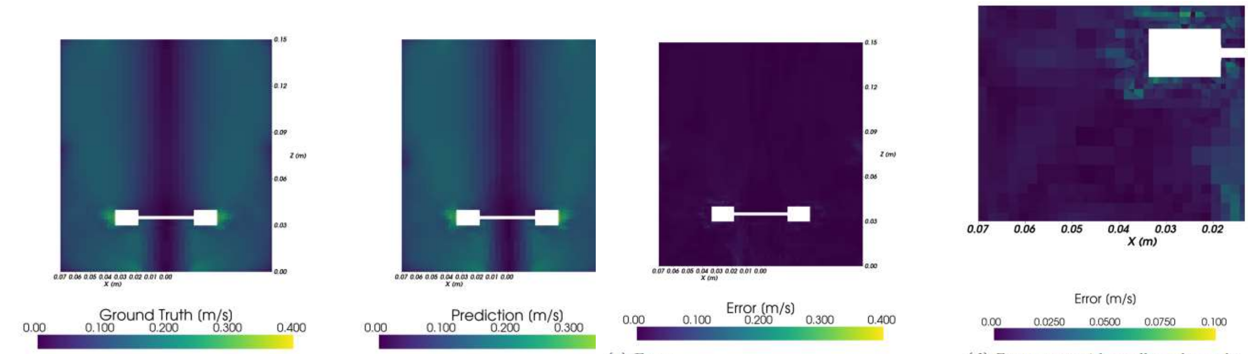
(a) Ground Truth (b) Prediction (c) Error (d) Zoom error with smaller color scale

ICCFD12 GNN for Bioreactors | CONFIDENTIAL

45

Visual Comparison of Velocity Fields
Extrapolated Test Cases

Z-shift extrapolation 3.5cm



(a) Ground Truth (b) Prediction (c) Error (d) Zoom error with smaller color scale

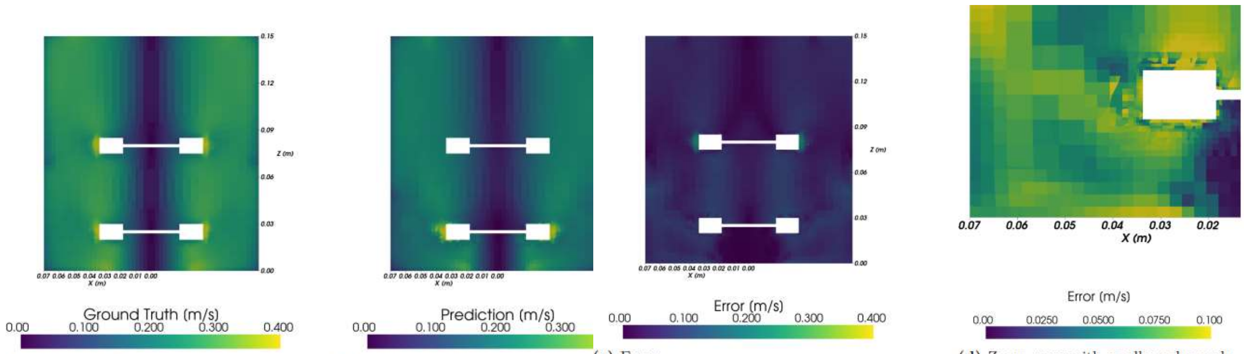
ICCFD12 GNN for Bioreactors | CONFIDENTIAL

46

Visual Comparison of Velocity Fields

Extrapolated Test Cases

Double Impeller



(a) Ground Truth (b) Prediction (c) Error (d) Zoom error with smaller color scale

Conclusions and Outlook

GNNs

- are capable of **regression** tasks on physical-fields
- can work with **variable geometry**
    - **Interpolation** showed good results
    - **Extrapolation** is challenging, especially when the physical patterns change considerably
- further combination with physics-informed neural networks holds a great promise ensuring **physical accuracy**

Thank you for listening!

Q&A