Engineering Mechanics Institute Conference and Probabilistic Mechanics & Reliability Conference (EMI/PMC 2024)

Chicago, Illinois | May 28–31, 2024

Physics-informed Graph Neural Network for predicitng Power Generation of Wave Farms

Suraj Khanal, Gaofeng Jia

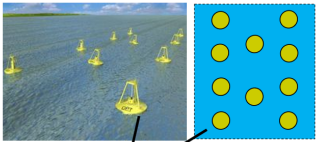
Colorado State University

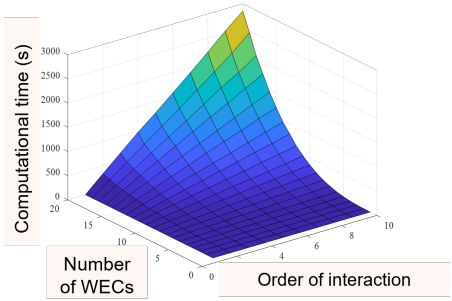
May 28, 2024

Outline

- Background
  - Wave Energy Converters
  - Surrogate Modelling
  - Hydrodynamic Coefficients
- Graph Neural Networks
  - Introduction to Graph
  - GNN: Fundamentals
  - Invariant GNN with Random Node Features
- Implementation Details
- Results
- Conclusion

Wave Energy Converters (WECs)

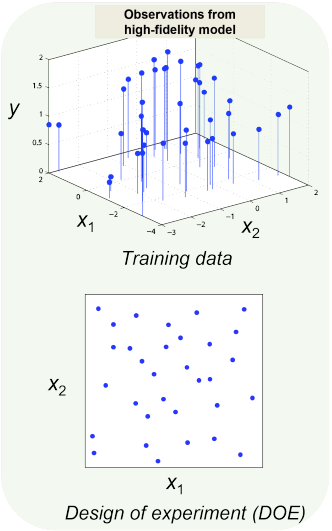


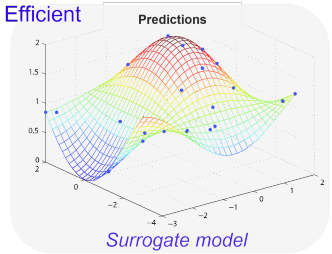


- WECs (buoys) are often deployed in arrays called wave farms
- Depending on layout, different performance, e.g. power

- For high fidelity models and larger arrays, predicting power for different layout is expensive
- Parametric study, design optimization, uncertainty quantification become impossible
- Surrogate Model preferred

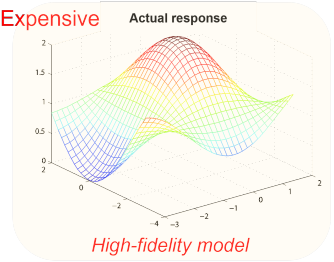
Surrogate Modelling





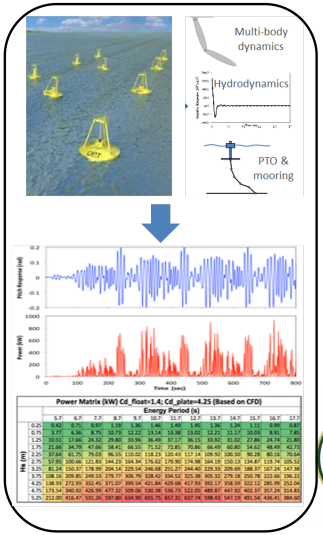
- Polynomial Response Surfaces
- **Gaussian Process(GP)** Model
- **Artificial Neural Networks (ANN)**

Approximation



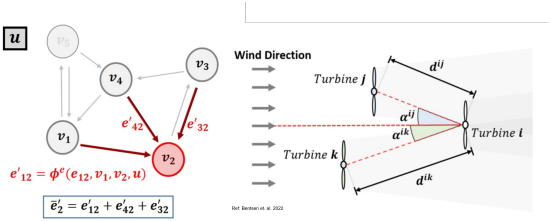
Hydrodynamic Coefficients

- *Li et. al. 2023* used PGP to predict hydrodynamic coefficients on wave farms and achieved good results
- Hydrodynamic Coefficients
  - Added Mass (am): real, $n_{wec} \times n_{wec}$
  - Added Damping (ad): real, $n_{wec} \times n_{wec}$
  - Wave Excitation Force (F): complex, n_{wec}
- Added mass and added damping: effects of radiated waves from WECs
- Wave Excitation Force: effect of incident and diffracted waves
- Power difficult to model, hence work on coefficients



Graph Neural Network (GNN)

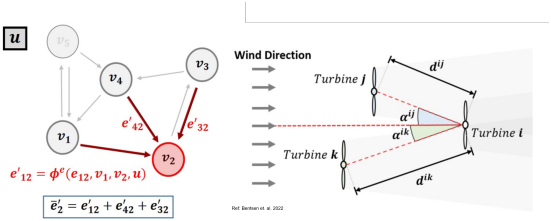
- Good result on Wind Farm, analogous to Wave Farm



GNN on wind farm (*Bentsen 2022*)

Graph Neural Network (GNN)

- Good result on Wind Farm, analogous to Wave Farm



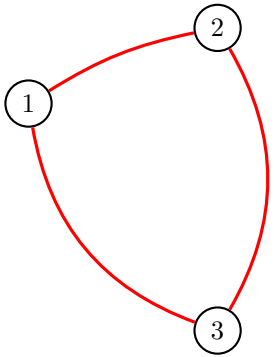
GNN on wind farm (*Bentsen 2022*)

Advantages of GNN:

- Scalability : easy to scale for larger arrays
- Performance: captures intricate relationship with high accuracy
- Efficiency: can operate in parallel

What is Graph?

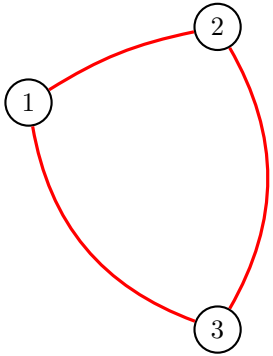
- Graph: nodes/vertices and edges
- $G = (V, E, U)$ where, V: node attributes, E: edge attributes, U: global attributes
- Connectivity of Graph: Adjacency Matrix, $\boldsymbol{a_{ij}}$



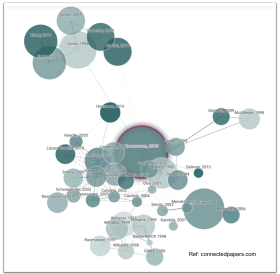
A simple Graph

What is Graph?

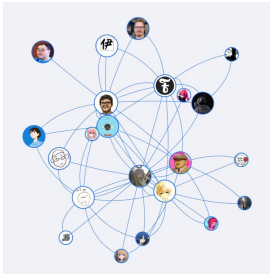
- Graph: nodes/vertices and edges
- $G = (V, E, U)$ where, V: node attributes, E: edge attributes, U: global attributes
- Connectivity of Graph: Adjacency Matrix, $\boldsymbol{a_{ij}}$
- Examples: social network, citation datasets



A simple Graph

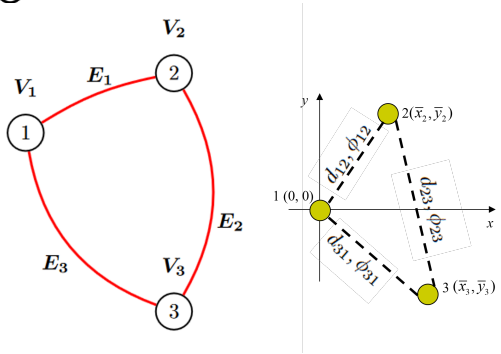


Citation Graph



Social Network Graph

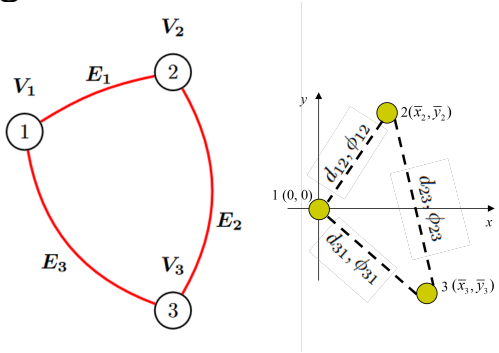
Graph Structure in WECs



3 WEC as a graph

- V : buoys, E: relation/interaction between two buoys

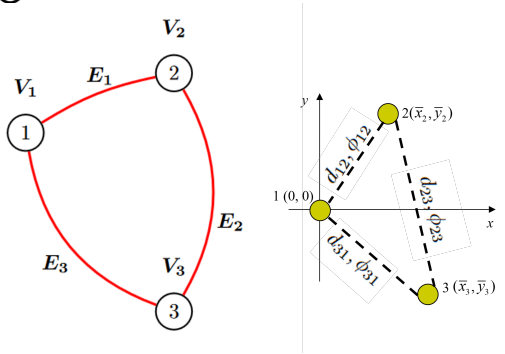
Graph Structure in WECs



3 WEC as a graph

- V : buoys, E: relation/interaction between two buoys
- Each node has features that are independent of other buoys
  $\boldsymbol{V_n} = \{x_i, y_i\}$
- Each edge has features that depend on i^{th} and j^{th} nodes(buoys)
  $\boldsymbol{E_n} = \{d_{ij}, \phi_{ij}\}$
- Connectivity: interaction between buoys. No interaction, no connectivity

Graph Structure in WECs



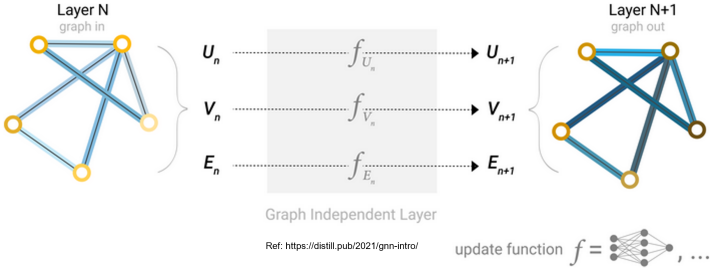
3 WEC as a graph

- V : buoys, E: relation/interaction between two buoys
- Each node has features that are independent of other buoys
  $\boldsymbol{V_n} = \{x_i, y_i\}$
- Each edge has features that depend on i^{th} and j^{th} nodes(buoys)
  $\boldsymbol{E_n} = \{d_{ij}, \phi_{ij}\}$
- Connectivity: interaction between buoys. No interaction, no connectivity

- Given different arrays can be represented with graphs, how do we predict corresponding power for a given array/graph?
  $\Rightarrow$ GNN

Graph Neural Network (GNN)

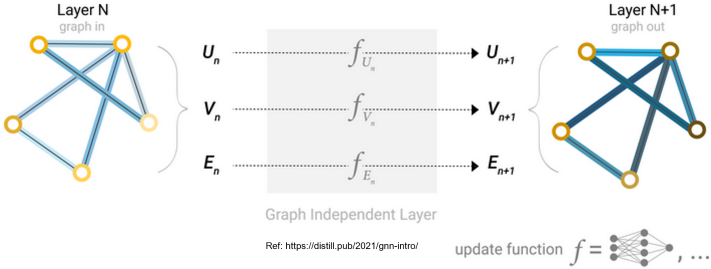
- **GNN** : An **optimizable transformation** that updates various **features** of the graph (node, edge, global) but **preserves** the **connectivity** of the graph



Schematics of a Graph Neural Network

Graph Neural Network (GNN)

- **GNN** : An **optimizable transformation** that updates various **features** of the graph (node, edge, global) but **preserves** the **connectivity** of the graph

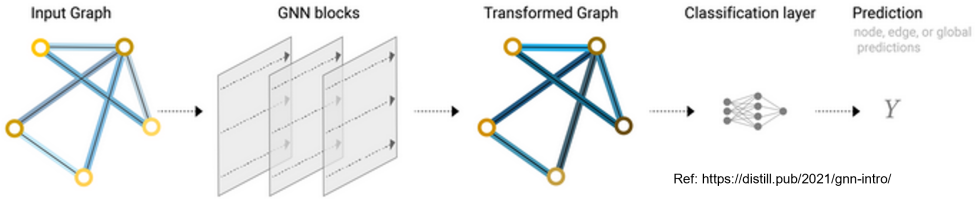


Schematics of a Graph Neural Network

- Given updated features, how to predict?

Predictions from GNN

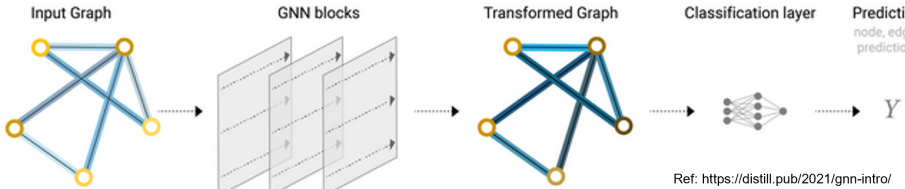
Aggregate features and then use a linear layer for prediction



An end-to-end Prediction Block with a GNN Model

Predictions from GNN

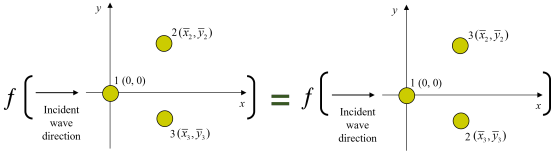
Aggregate features and then use a linear layer for prediction



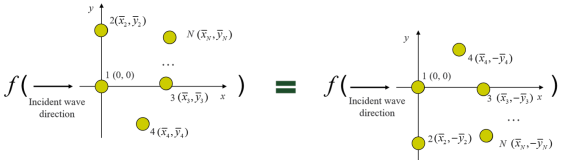
An end-to-end Prediction Block with a GNN Model

- Methods: GCN, GAT, GraphSAGE
- An invariant type GNN with randomly initaited node features used in the study, inspired by the works of Satorras et. al. (2021) and R Sato (2024)

Physics of Wave Farm



Permutation Invariance



X-Axis Reflective Invariance

Invariant GNN with Random Node Features

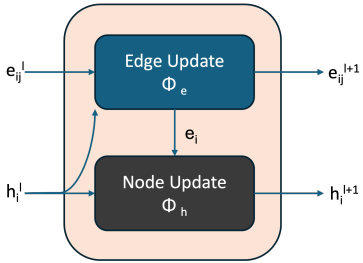
Satorras et. al. (2021) have shown that the following update functions result in invariances of translation and reflection to node and edge features:

$$\mathbf{e}_{ij}^{l+1} = \phi_e(\mathbf{h}_i^l, \mathbf{h}_j^l, ||\mathbf{x}_i^l - \mathbf{x}_j^l||^2, \mathbf{e}_{ij}^l) \qquad \text{Edge Update}$$

$$\mathbf{e}_i^{l+1} = \sum_{j \neq i} \mathbf{e}_{ij}^l \qquad \text{Edge Aggregation to Node}$$

$$\mathbf{h}_i^{l+1} = \phi_h(\mathbf{h}_i^l, \mathbf{e}_i^{l+1}) \qquad \text{Node Update}$$

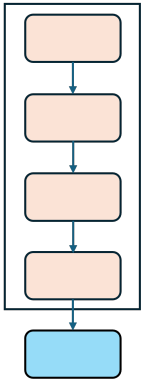
where, ϕ_e and ϕ_h are edge and node update functions respectively, for example: MLP



A GNN Layer

Implementation Details

- GNN layers stacked upon each other and a fully connected layer at the end for required output size
- Edge Features:
  $\mathbf{e}_{ij} = [\mathbf{I}_m(d_{ij}),\ \mathbf{K}_m(d_{ij}),\ cos(\theta_{ij})] \quad m = 1, 2, ...10$
  where $\mathbf{I}_m(\cdot)$ and $\mathbf{K}_m(\cdot)$ are mth order Bessel functions of the first and the second kind
- Node Features: $\mathbf{h}_i$ = Randomly generated from uniform distribution in $[0, 1]$
- Although randomly initialized, the network updates these features through the learning proess



Implementation Details (contd...)

- Implementation is done in Python using PyTorch and PyTorch Geometric
- Edge Update Function (ϕ_e): 4 hidden layers each with 200 units and a linear layer of 20 units at the end
- Node Update Function (ϕ_h): 3 hidden layers each with 100 units, followed by a hidden layer of 50 units and a linear layer of 10 units at the end
- After each hidden layer, ReLU is applied as non-linear layer
- MSE is used as loss function and model trained for 5000 epochs. Train:Val:Test Split = 6400:800:800
- Learning Rate Scheduler and Early Stopping which are standard practices in ML are also implemented
- For learning rate scheduler, PyTorch's 'ReduceLROnPlateau' is used, which reduces learning rate up to a certain limit if the validation loss does not decrease within a threshold for some epochs

Results

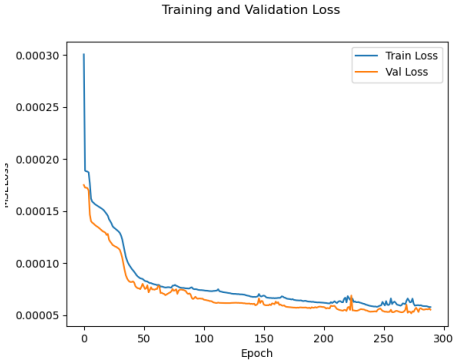
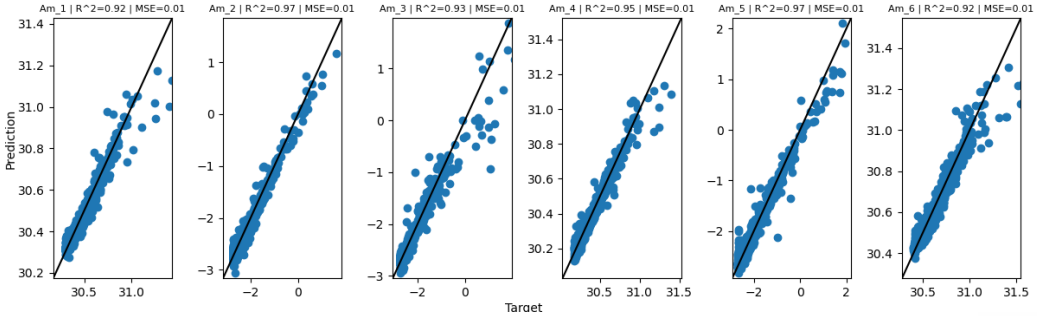
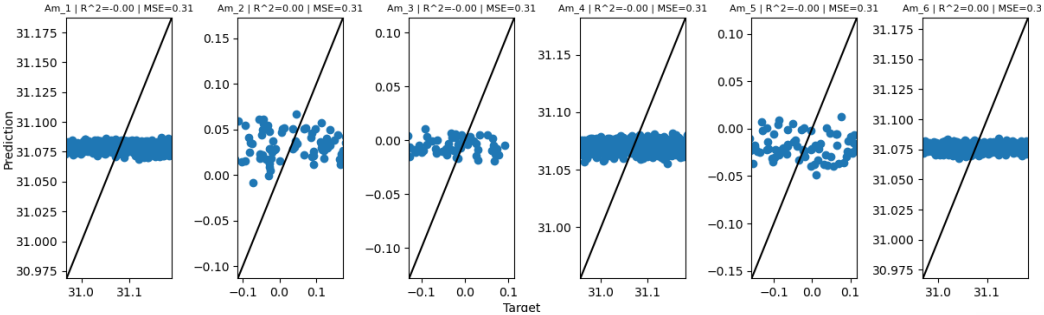


Figure: Loss over Epochs for Added Mass Coefficient Training

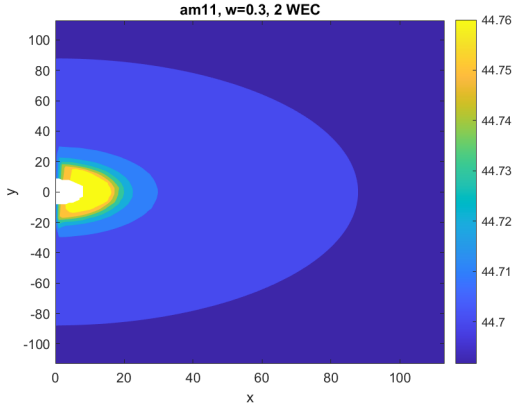
Prediction of Added Mass for $\omega = 0.3$

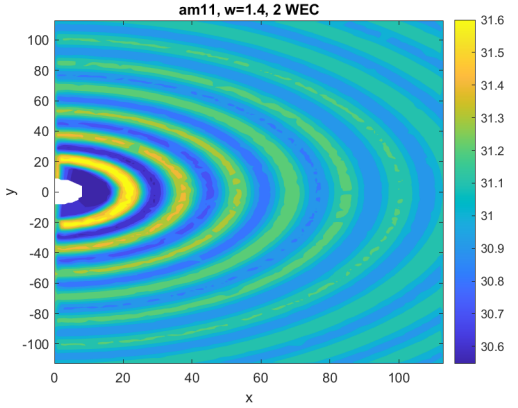


Prediction of Added Mass for $\omega = 1.4$



Variation of Added Mass Coefficient





Conclusion

- An invariant Graph Neural Network with randomly initialized node features was designed to predict hydrodynamic coefficients of wave energy converter arrays
- For lower frequencies of incoming wave, the model predictions are good ($R^2 > 0.90$) but poor performance for higher frequencies
- The underlying variation of coefficients for higher frequencies is too difficult to capture for the given model architecture

Conclusion

- An invariant Graph Neural Network with randomly initialized node features was designed to predict hydrodynamic coefficients of wave energy converter arrays
- For lower frequencies of incoming wave, the model predictions are good ($R^2 > 0.90$) but poor performance for higher frequencies
- The underlying variation of coefficients for higher frequencies is too difficult to capture for the given model architecture

Future Work:

- Analyse the case for higher frequencies and get good predictions
- Work on other hydrodynamic coefficients
- Model power generation after getting good results on coefficients

Acknowledgement

This research is supported in part by the National Science Foundation Engineering Design and System Engineering Program under grant number CMMI-2034040. This support is gratefully acknowledged.