Data & Physics-Driven Graph Convolutional Networks for Rapid Cavitation Energy Predictions on Marine Propeller Blades

[1]Ian Hubbard*, [1]Themistoklis Melissaris
[1]*Propulsion Technology R&D, Wärtsilä Netherlands B.V.*

Keywords: Graph Neural Networks, Geometric Deep Learning, Cavitation Erosion, Propeller Design

Abstract. Accurately predicting cavitation-induced implosion loads on a propeller blade surface via CFD can be computationally expensive. While RANS turbulence modelling offers faster computations than scale-resolving approaches, the required resolution in space and time to resolve the larger-scale cavitating structures still result in long lead times for routine propeller design cycles. Consequently, such high-fidelity analyses are typically limited to critical design studies or severe erosion cases. The focus of this work is to explore the possibility of significantly accelerating the erosion risk assessment of arbitrary blade geometries and operating conditions. To reduce inference time, we propose a hybrid data-driven and physics-based approach using a graph-based geometrical deep learning (GDL) framework. Specifically, we are developing a graph neural network (GNN) architecture to predict the non-dimensional surface energy, over a propeller blade geometry. We represent the propeller blade as a point cloud, from which we construct a neighbourhood connected graph. The GNN then leverages convolutional layers and incorporates the cavitating pressure coefficient, C_P , at each graph node, obtained via the vortex-lattice code MPUF-3A, as an additional physics-based feature. The proposed model, gASE-PS1, demonstrates promising accuracy in predicting propeller surface energy distributions across different time-steps of a blade revolution, particularly in identifying erosion-prone sites. This model can significantly reduce inference compute costs by more than 800x, making energy-based cavitation erosion risk assessments potentially feasible within the design cycle.

1 Introduction

Predicting cavitation erosion and propeller noise relies upon accurately capturing the vapor volume and its transient behaviour around the propeller blades. A novel erosion model, introduced by Schenke et al. [1] and implemented in a high-fidelity RANS CFD solver, has demonstrated reliable predictions of cavitation impact loads. This methodology has been validated against full-scale propellers and thruster units [2, 3] and further base-lined [4, 5], where it effectively identifies locations on blade surfaces prone to cavitation erosion.

However, the high-fidelity approach is computationally expensive and is generally limited to special cases instead of routine propeller design cycles. In an effort to broaden accessibility, the present work focuses on creating a low-cost computational model through a geometric deep learning (GDL) framework. The intent is to perform cavitation erosion risk assessments far more quickly than the current RANS-based work flow. The project is divided into phases; the study shown here corresponds to a feasibility or proof-of-concept stage.

Specifically, the proposed model attempts to learn a functional mapping between an initialized cavitating-pressure distribution, from the potential flow–based vortex lattice method (VLM), MPUF-3a [6] and the RANS-based cavitation impact loads. By representing the blade geometry as a graph, this type of learn-

ing provides topological consistency and permutation invariance (which standard CNN, convolutional neural networks cannot provide) [7] such that node-level features and their gradients can propagate from one node to another, helping the neural network capture important spatial correlations in the data.

2 Physical Modelling

Rather than only using the graph representation of blade geometry to predict the impact loads, we instead, and additionally pre-compute the potential flow solution using VLM under conditions consistent with the RANS simulation. The VLM code MPUF-3a is a fast and accurate blade cavitation assessment code which provides a computed pressure distribution on the blade surface due to the upstream wakefield, which is referred to here as features $\{u_1, u_2, \ldots, u_n\}$ in our model. To enable the model to generalize across various blade designs and operating conditions, these pressure values are expressed in a dimensionless form as the pressure coefficient C_{P_i}:

$$C_{P_i} = \frac{P_{\text{cav},i}}{0.5\rho n^2 D^2}, \tag{1}$$

where $P_{\text{cav},i}$ is the local cavitating pressure for the i^{th} vertex of the panel mesh, ρ is fluid density, n is the propeller rotation speed, and D is the diameter.

The erosion impact loads, which are the learning target set $\{y_1, y_2, \ldots, y_n\}$ for each node of the model, are computed using a RANS-based framework. The predicted impact loads are derived by projecting the total radiated energy at each computational cell onto the blade surface. By applying a volume integration on all the volume sources, the impact power per unit surface at any location $\mathbf{x}_s$ is computed as,

$$\dot{e_S}(t, \mathbf{x}_s) = \frac{1}{4\pi A_f} \int_V \Omega_d \dot{e}_{rad}(t, \mathbf{x}_{cell}) dV, \qquad where \qquad \Omega_d = \min \frac{(\mathbf{x}_{\text{cell}} - \mathbf{x}_s) \cdot \mathbf{n}}{\|\mathbf{x}_{\text{cell}} - \mathbf{x}_s\|} A_f, 2\pi \tag{2}$$

In the equations given in (2), Ω_d is the discrete solid angle, $\mathbf{n}$ is the surface normal vector, and A_f the surface cell face area. Eventually, by integrating over the whole surface, the total surface accumulated energy can be estimated, as well as the energy distribution over the surface. Furthermore, to remove dependencies on the mesh resolution and global operating parameters, the integrated surface energy is converted into a non-dimensional form given by [4]:

$$ASE_{\text{Non}} = \frac{e_S}{\rho A_f n^2 D^3}, \tag{3}$$

At the initial phase modelling approach, we attempt to avoid ducted propellers for the time being, which puts limitations on the size of our dataset of designs and operating conditions for which cavitation erosion assessments have been performed. To expand the training data, the model is instead trained on the individual time steps between a full blade revolution.

Table 1: An overview of the test case propeller design and corresponding operating condition.

Type	P_B [kW]	D_P [m]	N [RPM]	V_S [kn]	Pitch $\Delta\phi$	Erosion
Open	5500	5.6	134	13.5	-8.0	Yes (PS)

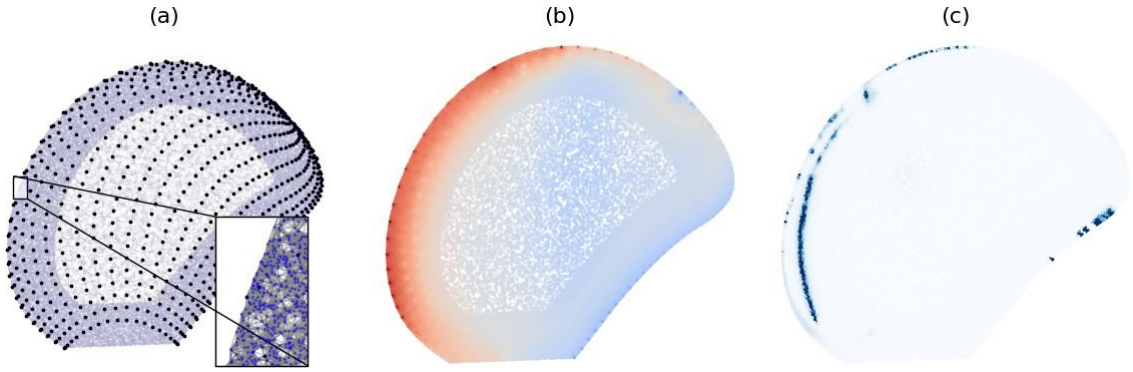


Figure 1: (a) VLM vertex points in black and corresponding graph representation of up-sampled nodes & edges in blue. (b) C_P distribution across the nodes. (c) ASE_{Non} distribution mapped across the nodes.

In MPUF-3a, the smallest available time step is $6^\circ/\delta t$ to obtain instantaneous pressure distributions. Within each time step of a blade revolution for the CFD solution, the total radiated energy is projected from the volume sources onto the blade surface to obtain an instantaneous energy distribution on the surface. This process is repeated every $6^\circ/\delta t$ of the blade rotation, yielding up to 60 snapshots (RANS) that can be paired with the corresponding VLM-based cavitating pressure solutions. Overall, this approach effectively expands the training data set from a single operating point to a temporal sequence of blade positions. For this initial feasibility phase, we employ this strategy for a test case, in self-propelled condition shown in table 1.

3 Graph Representations & Model Architecture

To exploit geometric relations, the blade geometry is transformed into a graph, $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ represented by nodes V and edges E. This blade geometry is represented as a point cloud $\{x_1, x_2, \ldots, x_n\}$ of the VLM panel mesh vertices, and to simplify learning, the 3D blade surface is split into individual suction and face-sides. The original VLM output, comprising $O(10^2)$ mesh vertices, is far sparser than the RANS dataset of $O(10^5)$ per surface mesh. Thus, the VLM nodes are up-sampled to retain the detail of the RANS data, such that new points $p_i \in P_{\text{new}}$ are randomly generated within the blade contour and other defined refinement offsets to capture more local detail. Each up-sampled point p_i inherits VLM features u_j via a k-nearest neighbors (k-NN) interpolation implementation similar to that proposed in PointNet++ [8], such that $p_j \in N_k(p_i)$:

$$u_{\text{new},i} = \sum_{j \in N_k(p_i)} \frac{\tau\, x_j\; u_j}{\sum_{j' \in N_k(p_i)} \tau\; x_{j'}}, \quad \text{where } \tau\; x_j = \frac{1}{d\; p_i, x_j + \epsilon}. \tag{4}$$

Here, ϵ is a small scalar to avoid dividing by zero. A matching k-NN approach also maps the RANS-based ASE_{Non} energy data onto the same P_{new} to form the target set. Finally, a k-NN graph construction step forms an adjacency matrix $\tilde{A}$ (of node edges), linking only node pairs (p_i, p_j) that lie within each others k-nearest neighborhoods. An example of this can be seen in figure 1 (a), with figure 1 (b, c) showing the feature and target inheritance and interpolation onto the graph.

Of the three fundamental types of existing graph neural network (GNN) layers, the convolutional variant has been used here due to its computational efficiency. Attentional layers for instance, may have provided architectural improvements to this model, however they have not been investigated in this phase due to

their significant additional memory overhead [7]. The convolutional layer variant is introduced by Kipf and Welling [9], which aggregates neighbor features for each node i in a normalized manner:

$$h_i^{(l+1)} = \sigma \sum_{j \in \mathrm{N}(i) \cup \{i\}} \frac{1}{\sqrt{d_i d_j}} h_j^{(l)} W^{(l)} + b \quad , \tag{5}$$

where $\mathbf{h}_i^{(l)}$ represents the feature vector at node i in layer l, d_i is the node degree, $W^{(l)}$ is the trainable weight, b a bias term, and $\sigma(\cdot)$ is an activation function (in this work, the well known ReLU is used). In matrix form, this can be written as,

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A}\, \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right) . \tag{6}$$

Here, $\tilde{D}$ is the diagonal degree matrix for the adjacency $\tilde{A}$, $H^{(l)}$ is the stacked node features, and $W^{(l)}$ is the trainable weight matrix in layer l.

The GNN is constructed using *PyTorch Geometric* [10]. An autoencoder-like architecture is adopted to identify lower-dimensional feature embeddings that capture the most relevant information about local geometry and cavitation impact loads, while also helping filter out outlier ASE_{Non} values commonly found in these analyses [5]. Batch normalization is used between layers to stabilize training on batches. In between the layers, skip-connections are either added or linearly projected to enable residual information to by-pass the encoding-decoding bottleneck and helping to preserve finer details potentially lost in the encoder's compression.

From the complete dataset of 60 graphs, two (30° and 312° blade positions) are intentionally excluded as hold-out data for final generalization testing (see in section 4). The final network is trained using a 80/20 train/test split of the remaining graphs, which are shuffled in order to avoid learning temporal dependences. A 10-fold cross-validation is performed, with up to 300 epochs per fold and early stopping based on validation loss to avoid overfitting. The AdamW [11] weight-decay decoupled optimizer is applied, with a scheduled learning rate ranging from 1×10^{-3} down to 1×10^{-6}. A combined loss function is minimized, composing of a mean-squared error (MSE) and a weighted penalty function for node-level predictions $\hat{y}_i < 0$, which are non-physical. This loss is given as,

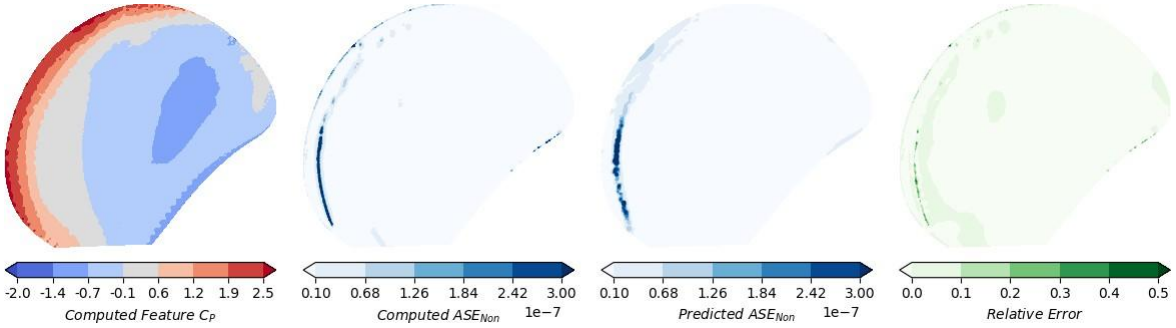
$$\mathrm{L} = \frac{1}{N} \sum_{i=1}^{N} \left[(y_i - \hat{y}_i)^2 + \gamma \max\left(0, -\hat{y}_i\right) \right] \tag{7}$$

Where γ is a scalar weighting. The vanilla final model is here-after refer to it as *gASE-PS1* in the following sections, which is trained on only pressure-side (PS) features and targets as this particular test case only shows risk of pressure side erosion.

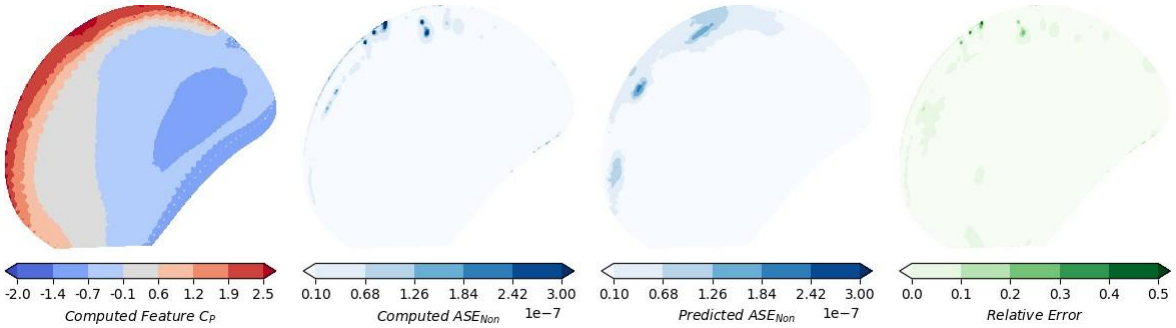
4 Results & Discussion

To evaluate the performance of the model, gASE-PS1, we focus on two aspects: individual time step predictions and total accumulated surface energy for a full rotation.

Figure 2 compares the target results (computed CFD impact loads) with gASE-PS1 node-level predictions for two individual time steps from the hold-out set. These cases test the model's generalizability based on input C_p distributions. In Figure 2(a), the model accurately predicts the dominant erosion site on the lower radii area of the blade, consistent with the computed ASE_{Non}. This demonstrates the



(a) 30° blade position



(b) 312° blade position

Figure 2: Far-left showing the VLM computed input feature C_P , followed by a comparison of the computed and the predicted ASE_{Non} distributions over the blades for two individual time steps (a) and (b) of the hold-out set. Far-right indicates the relative error between computation and prediction.

model's ability to leverage C_p distribution to predict impact sites. Figure 2(b) presents a more challenging case where the pressure side cavitation has moved towards the blade tip. While the model no longer predicts extensive erosion on the lower radii, it also does not quite capture the impact loading at the tip position. This can be expected for two reasons, the first being there is lack of detail in the C_P distribution that could suggest these collapse events, and the second being that this is not a dominant feature in the training set, and details may be left out during the convolution and embedding stages.

Figure 3 presents the accumulated surface energy for a full rotation, a typical approach for such analyses [2, 3]. This is calculated by aggregating ASE_{Non} for all 60 individual time steps/graphs, including training, validation, and hold-out data sets. The model performs well at predicting the general trend of accumulated energy. It accurately captures both the location and magnitude of the major impact zone when compared to the target data. This suggests that despite some limitations in individual time step predictions, the model performs well in estimating overall erosion patterns over a complete rotation.

5 Conclusion & Future Work

The current base-line model is able to leverage the pressure predictions from the low-fidelity, physics-based VLM in order to make a good prediction of the impact load sites, and reasonable prediction of its magnitude when observing unseen timesteps. When accumulating all the energy over a revolution, the prediction remains fairly close to the computed energy and overall indicates promising results for this model's implementation. It should be noted that this level of prediction accuracy would also not have been possible on such a limited dataset without the a-priori VLM computed pressure information.

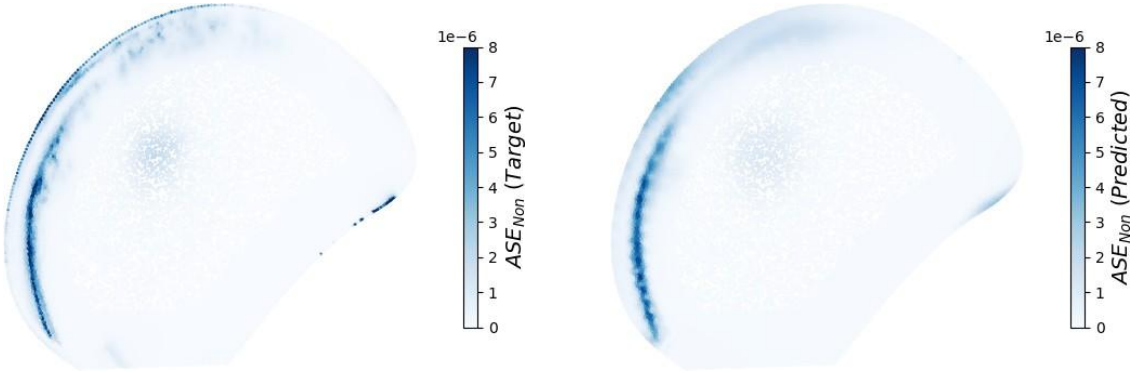


Figure 3: A comparison between the computed, target accumulated surface energy (left) and accumulated energy following the individual time-step predictions (right) for one revolution.

While the current study may indicate that gASE-PS1 may have learned the most meaningful embedding and mapping between C_P and ASE_{Non}, the generalization of this model towards other designs and operating conditions is still questionable due to the limited size and scope of this dataset. The generalization will be a subject of further investigation and the next phase will be to perform the erosion risk assessment calculations again for other designs and conditions previously investigated in [4]. This would require saving the individual time step volumetric energy as done in this test case to expand this dataset.

Once additional data is available, generalization will be further tested and architectural improvements will be investigated. Overall, the inference cost of this model is a *factor of greater than 800x lower* than performing a full CFD computation on our typically allocated amount of CFD compute. This gain has the potential to shift the lead-time of an erosion risk assessment from weeks down to minutes per case/condition, providing readily accessible results within a typical propeller design cycle time-line.

References

[1] S. Schenke, T. Melissaris, T.J.C. van Terwisga 'On the relevance of kinematics for cavitation implosion loads', Phys. Fluids 31, 052102 (2019)

[2] T. Melissaris, S. Schenke, N. Bulten, T.J.C. van Terwisga 'Cavitation erosion risk assessment on a full-scale steerable thruster', Ocean Engineering, 251 (2022)

[3] T. Melissaris, S. Schenke, T.J.C. van Terwisga 'Cavitation erosion risk assessment for a marine propeller behind a RoRo container vessel', Phys. Fluids 35, 013342 (2023)

[4] T. Melissaris, I. Hubbard 'An overview of cavitation erosion risk assessment on propeller blades in full scale', Proceedings of the 12th International Symposium of Cavitation, Chania, Greece, 2024

[5] I. Hubbard, T. Melissaris 'Detecting numerical instabilities in propeller cavitation erosion modelling using unsupervised learning', Proceedings of the 26th Numerical Towing Tank Symposium (NuTTS), Mulheim an der Ruhr, Germany, 2024

[6] S. Kinnas, S. Pyo, 'Cavitating propeller analysis including the effects of wake alignment', SNAME Journal of Ship Research, Vol. 43, No. 1, pp. 38-47 (1999).

[7] M.M. Bronstein, J. Bruna, T. Cohen, P. Velickovic 'Geometric deep learning: grids, groups, graphs, geodesics, and gauges', arXiv preprint, arXiv:2104.13478, (2021)

[8] C. Ruizhongtai, Q. Yi, L. Yi, H. Su, L.J. Guibas, 'PointNet++: Deep hierarchical feature learning on point sets in metric space' arXiv:1706.02413

[9] T.N. Kipf, M. Welling, 'Semi-supervised classification with graph convolutional networks', arXiv preprint, arXiv:1609.02907, (2017)

[10] F. Matthias, J.E. Lenssen, 'Fast graph representation learning with PyTorch Geometric', ICLR Workshop on Representation Learning on Graphs and Manifolds, arXiv:1903.02428, (2019)

[11] I. Loshchilov, F. Hutter, 'Decoupled Weight Decay Regularization', arXiv preprint, arXiv:1711.05101, (2019)