Discover the Tractable Latent Space of Floating Offshore Wind Turbine Based on a Novel GNN-Encoder-Decoder-LSTM Deep Learning Architecture

Kobe Hoi-Yin Yung[1,2(✉)], Qing Xiao[1], Xiuqing Xing[2], Atilla Incecik[1], and Chang Wei Kang[2]

[1] University of Strathclyde Glasgow, Glasgow, UK
kobehy.yung@strath.ac.uk
[2] Institute of High Performance Computing (IHPC) Agency for Science, Technology and Research (A*STAR), Singapore, Singapore

Abstract. Floating Offshore Wind Turbines (FOWT) provided new potential in harvesting wind energy in far offshore deep-sea regions and contributed to the world decarbonization Net-Zero target. Providing structural health monitoring (SHM) is crucial for ensuring the structural integrity of FOWT in lifecycle. However, the SHM is technically challenging with high Operational and Maintenance Expenditure (OPEX). Recently, Digital Twin (DT) and advanced sensor technologies offer alternative solutions to provide effective strategy in SHM remotely. Data-driven DT with deep learning models can formulate highly nonlinear dynamics systems. Yet, these existing models only perform the "black box" prediction without explicitly modeling the spatial-temporal relationship and consider only homogenous loading exerted in contrast to the complicated loading combination of FOWT with wind, wave and sea current.

To address the existing modelling limitations, a new Graph Neural Network (GNN)-Encoder-Decoder-Long Short-Term Memory (LSTM) surrogate model of FOWT is presented in this work, which can perform 50 times faster than the real-time of simulation data set with accurate prediction of wind turbine tower bottom forces in the dominant dynamic modes force-aft and side-side directions. The training data is based on the software QBlade simulation and focuses on the OC4 5MW DeepCwind FOWT structure. A holistic quantitative analysis is carried out to validate the tractable latent space vectors for this complex FOWT system.

Keywords: Floating Offshore Wind Turbine · Artificial Intelligence · Deep Learning · Digital Twin · Tractable Latent Space · Surrogate model

1 Introduction

Offshore wind energy has been demonstrated promising renewable energy over the past decade, which was implemented all over the globe, including China, UK and Germany [1]. Especially, FOWT are considered to have potential for harvesting wind energy from

M. H. Lees et al. (Eds.): ICCS 2025, LNCS 15903, pp. 91–107, 2025.
https://doi.org/10.1007/978-3-031-97626-1_7

far offshore deep-sea regions compared to fix-bottom offshore wind. When implementing construction projects, there are huge technical challenges in providing maintenance of large-scale wind farm in which pose huge OPEX. With the advancement of state-of-the-art sensor technology and cloud technology, it allows the development of DT technologies [2], which is a virtual representation of the physical asset. Finite Element Modeling (FEM) is widely used to model multibody dynamics, but it is computationally expensive to run in real-time for DT. Surrogate models allow DT running instantaneously but keeping the accuracy in highly nonlinear dynamics.

In this work, we presented a novel GNN-Encoder-Decoder-LSTM for solving mentioned above. In deep learning architecture, there are autoencoder and encoder-decoder used for dimension compression, which are the nonlinear reduced order model approaches. However, in literature, there is a lack of organized analysis explaining the behavior of latent space characteristics. Therefore, we provide a systematic analysis for the relationship between the higher dimensions graph embedding and the latent space vector and provide the first application of Modal Assurance Criterion (MAC) [3] to demonstrate the tractability of the latent representation of performing this temporal dynamic prediction task of FOWT.

2 Related Work

Nowadays mid-fidelity engineering tools e.g. OpenFAST [4] and QBlade [5] can simulate complex coupling in between the aerodynamic, hydrodynamic, structural dynamic, servo-dynamic, and provide efficient computation time comparing to high fidelity tool e.g. Computation Fluid Dynamics. However, mid-fidelity engineering tools still have limitations in running real-time in a desktop computer and typically for time series dynamic simulation that require the numerical transient period for the iterating the convergence in between different dynamic modules. For example, QBlade employed Hilber-Hughes-Taylor formulation [6] integrator for solving the Differential-Algebraic Equations. To address the inherent computational time problem, Reduced Order Model (ROM) is commonly adopted to speed up the simulation. In this section, we review recent developments in ROM for structural dynamics.

2.1 Physics-Based and Conventional ROM

A reduced ordered state space model with linearized approach can be used to run the interested parameters simulation of Ordinary Differential Equation. Which requires to linearize the nonlinear function at certain operating points e.g. sea state for FOWT. For instance, [7] formulated the time series ROM with N4SID system identification technique and created a bank of state space models at several sea states for Kalman Filtering and then merged the estimation result with Bayesian Multiple-Model Adaptive Estimation algorithm. This provided the mooring forces estimation for FOWT for unseen sea states with 20 times faster than real-time. There are also effective compression approaches for high dimensional data, such as the method of Proportional Orthogonal Decomposition (POD) [8] and Dynamic Mode Decomposition (DMD) [9] are used to truncate the high order component to provide the low rank matrix for faster computation.

2.2 Deep Learning ROM

Recently, machine learning and neural network provide new alternatives with the concept of "Universal approximation theorem", and deeper layer network "Deep Learning" can provide better prediction. Especially Autoencoder [10] forms the architecture that learns the lower dimension representation by training the neural network with means square error loss. For instance, [11] used AE-LSTM to extract the low-dimension and pass to LSTM. However, this does not explicitly incorporate the physical properties of the structure e.g. mass, stiffness and geometrical space. Furthermore, the existing deep learning applications in FEM only considered simple structures e.g. a single beam element or a rectangular multistorey frame with homogeneous loading input e.g. one horizontal force or one acceleration. [12] used the Variational Autoencoder (VAE) and POD for time series force prediction and homogeneous loading or ground motion acceleration, and the real-time factor was not reported. On the contrary, FOWT experiences a complicated combination of wind, wave and current with different amplitudes and turbulence, which are heterogenous loading properties problem to the structural system. The above-mentioned deep learning methods have not addressed the highly nonlinear problem for FOWT.

Recently, a Physics-Guided Spatial Temporal Graph Neural Network (GNN) [13] was presented that overcome the limitations mentioned above and able to handle heterogeneous loadings input. The GNN encoded the aero-hydrodynamic loadings with geometry and material properties, and combined with LSTM for predicting the tower forces 14 times faster than real-time. With the "baseline" model GATv2LSTM in [13], we presented a new architecture that compose of GNN-Encoder-Decoder-LSTM which can further accelerate the prediction by reducing the LSTM hidden layer parameter size substantially with the additional Encoder-Decoder and present the latent representation.

3 Simulation in QBlade

This study focuses on the FOWT OC4 5MW DeepCwind semisubmersible [14] as shown in Fig. 1. The model file is openly available on QBlade website [5]. The metocean data is based on the West of Barra Scotland from LIFES50 + project [15].

This study refers to the Design Load Case DLC 1.2 in the technical specification IEC 61400-3-2 [16]. The sea state follows Pierson-Moskowitz spectrum with significant wave height 3.5 m and period 10.68 s. The hydrodynamic modeling is Potential Flow with Morison Drag. To capture the higher order hydrodynamics effect, vertical stretching for wave and the second-order full Quadratic Transfer Functions (QTF) are used. For sea current, the near surface current 0.88m/s with reference depth of 30 m and subsurface 0.84 m/s with power 1/7 are adopted.

TurbSim [17] is used for modeling the turbulence wind with a mean wind speed of 13.63 m/s at hub height with turbulence Class IC and Kaimal model. The aerodynamic loading from the rotor is solved by the Unsteady Blade Element Momentum (UBEM) [18]. Øye dynamic stall and the tower shadow models are activated. A tower drag coefficient of 0.5 is used.

Figure 2(a) shows the simulation setting in QBlade and Fig. 2(b) shows the discretized nodes and the critical tower bottom for the focus of this study. Detailed GNN is described

in Sect. 4 with Table 1 and 2. Figure 3 refers to the wind velocity at hub height and Fig. 4 refers to the wave elevation time series that the FOWT system experiences.

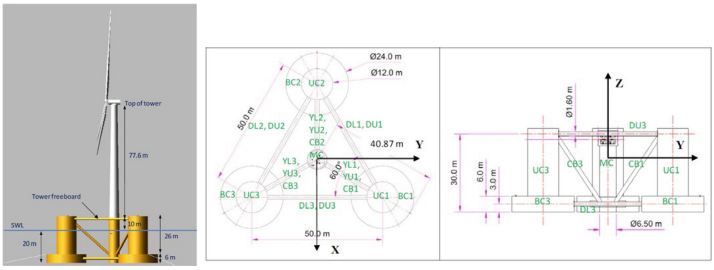

Fig. 1. Geometric specification of OC4 5MW DeepCwind FOWT [14]

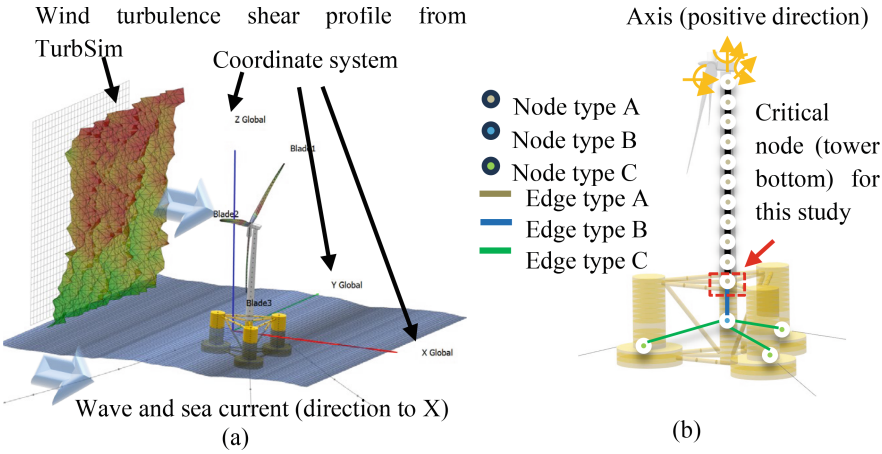

Fig. 2. (a) QBlade simulation and (b) GNN discretized of OC4 5MW DeepCwind FOWT

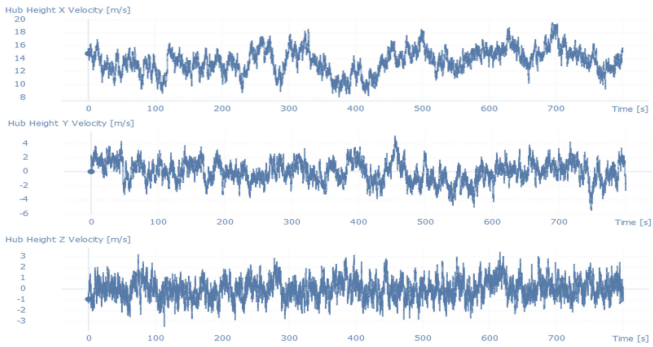

Fig. 3. Wind velocity generated from TurbSim

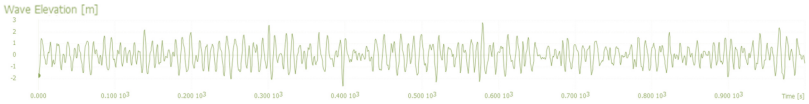

Fig. 4. Wave elevation profile time series data in QBlade

4 Novel GNN-Encoder-Decoder-LSTM

Table 1 and 2 summarize the node features and edge attribute of the GNN respectively as illustrated in Fig. 2(b).

Table 1. Node features inputs and node target outputs of the GNN

	Node features and inputs	Node target outputs
Node type A (11 nodes)	• Nodal mass • Node coordinates • Aerodynamic load	• Internal forces and moment in X, Y and Z directions
Node type B (1 node)	• Lumped mass of hull structure • 6DOF displacement, velocity and acceleration • Wave elevation • Water kinematic velocity	• 6DOF displacement, velocity and acceleration • Wave elevation • Water kinematic velocity
Node type C (3 nodes)	• Node coordinates • Wave elevation • Water kinematic velocity	• Support reaction forces from mooring fairlead

Figure 5 illustrates the whole proposed architecture: firstly the node embedding from GNN is concatenated to the graph embedding of size 240 and compressed to the latent space vector with the encoder; then the latent vector is passed to LSTM and the predicted latent space vector is mapped to the node feature prediction with the decoder and Fully

Table 2. Edge attributes of the GNN

	Edge attributes
Edge type A	• Length • Stiffness (bending, axial and torsion)
Edge type B and C	• Length

Connected (FC) layer. The activation functions of Rectified Linear Unit (ReLU) and Exponential Linear Unit (ELU) are tested with different combinations. Figure 5 refers to the optimum architecture of LSTM192ED128-128relu-elu32res. The procedure of Hyperparameter identification will be elaborated in Sect. 5.

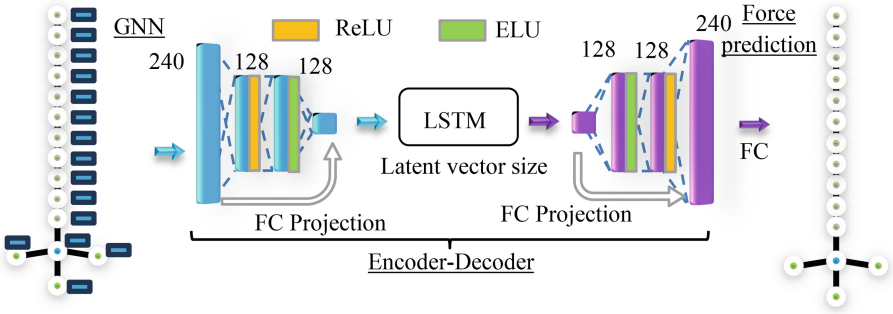

Fig. 5. GNN-Encoder-Decoder-LSTM architecture

The Graph Attention Network GATv2Conv [19, 20] is adopted and the node embeddings are calculated in the following:

$$x_i' = \alpha_{i,i}\Theta_s x_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\Theta_t x_j \tag{1}$$

The attention weight $\alpha_{i,j}$ is calculated as

$$\alpha_{i,j} = \frac{\exp(a^T LeakyReLu(\Theta_s x_i + \Theta_t x_j + \Theta_e e_{i,j})}{\sum_{k\in\mathcal{N}(i)\cup\{i\}}\exp(a^T LeakyReLu(\Theta_s x_i + \Theta_t x_k + \Theta_e e_{i,k})} \tag{2}$$

The node features inputs x_i are defined in matrix [number of nodes, number of features] and edge attribute $e_{i,j}$ in matrix [number of edges, number of features]. The inputs to GAT include node type A feature $x_A \in \mathbb{R}^{11\times 11}$, node type B feature $x_A \in \mathbb{R}^{1\times 20}$, node type C feature $x_C \in \mathbb{R}^{3\times 2}$, edge type A attributes $e_B \in \mathbb{R}^{10\times 4}$, edge type B attributes $e_B \in \mathbb{R}^{1\times 1}$ and edge attributes type C $e_C \in \mathbb{R}^{3\times 1}$ as listed in Table 1 and 2. After the node embeddings are calculated with 1-hop neighbors for each edge type messaging passing, they are concatenated to form the whole graph embedding vector $z \in \mathbb{R}^{240}$.

Long Short-Term Memory (LSTM) model is used for handling temporal dynamics:

$$i_t = \sigma(W_{ii}\tilde{z}_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \tag{3a}$$

$$f_t = \sigma(W_{if}\tilde{z}_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \tag{3b}$$

$$g_t = \tanh(W_{ig}\tilde{z}_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \tag{3c}$$

$$o_t = \tanh(W_{io}\tilde{z}_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \tag{3d}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{3e}$$

$$h_t = o_t \odot \tanh(c_t) \tag{3f}$$

where $\tilde{z}_t$ is the input latent vector at time t, h is the hidden state, W are the weight matrix, b are the bias, i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively. σ is the sigmoid function, $\odot$ is the Hadamard product. A time window size 100s is used for the sliding window.

The FC layer is used to map the output of LSTM to the node target prediction matrix y in the format of [number of nodes, number of target features] for $y_A \epsilon \mathbb{R}^{11\times6}$, $y_B \epsilon \mathbb{R}^{1\times20}$ and $y_C \epsilon \mathbb{R}^{3\times3}$. Additionally, the residual skip connection, FC projection, is used to improve gradient flow for the deeper layers. The deep learning model is implemented in PyTorch [21] and PyTorch Geometric [22] and trained with Backpropagation ADAM optimizer [23], weight decay 0.0001, learning rate 0.001 and epoch 100 and mean squared error (squared L2 norm). 731.5 s of time series data is used for training and another unseen 731.5 s of time series data for prediction and testing. The training data is preprocessed with z-score normalization.

5 Results

5.1 Validation of Simulation Training Data

A pitching decay test is used to benchmark the present QBlade simulation model for preparing Deep Learning model training data is aligned with the reported [24] QBlade model (QB) and the OpenFAST (OF) result as shown in Fig. 4 (Fig. 6).

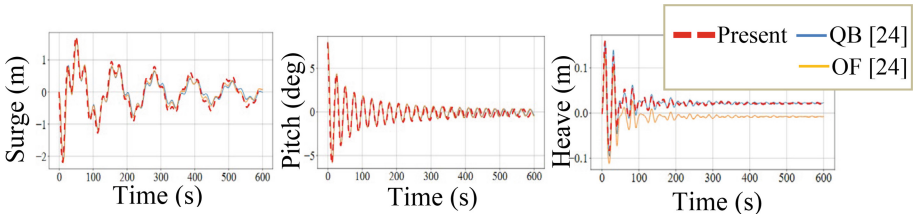

Fig. 6. Pitch decay test for benchmarking

5.2 Deep Learning Model Prediction Results

This section shows the prediction experiments on the neural network Hyperparameters identification procedure e.g. hidden layer sizes and activation function of the proposed GNN-Encoder-Decoder-LSTM and compares with the baseline model GATv2LSTM (without the Encoder-Decoder), against the "truth value" QBlade simulation. This study will only reveal the critical tower bottom forces and moments as stated in Fig. 7, 8, 9, 10, 11, 12, 13, 14 and 15. The time series prediction of forces are shown in detail to illustrate the trend difference. The forces and moments for each architecture results are summarized in Table 3. An example of legend name LSTM192ED128-128relu-elu32res means LSTM192 [hidden size] ED128-128 [Encoder-Decoder structure] relu-elu [activation function in the Encoder-Decoder] 32 [latent vector size] res [residual skip connection].

Study on Latent Space Vector Size. Based on the LSTM hidden size 100, the latent space vector size is varied from 10 to 32 with all fully connected layers attached with ReLU activation function.

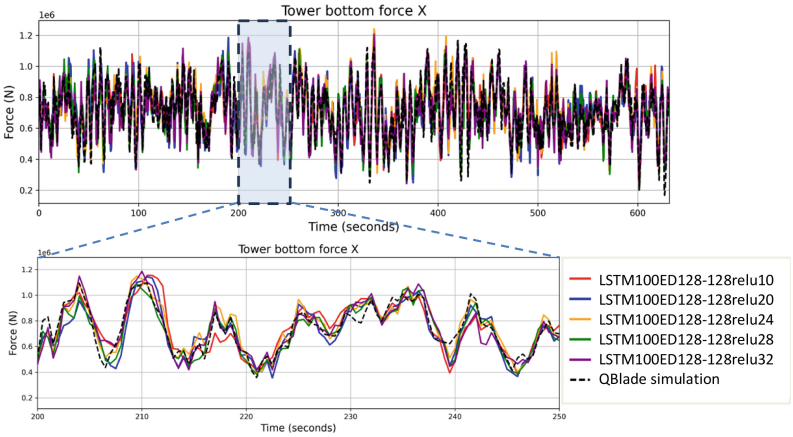

Fig. 7. Tower bottom force X prediction with varying latent vector size.

Higher latent space size can preserve more information from the high dimensional vector. The latent size 32 balances the capacity for model complexity and prevents the noisy details of larger size so it provides better prediction to the QBlade simulation.

Study on LSTM Hidden Size. Based on the optimal value latent space vector 32, the LSTM hidden size is varied from 100 to 192 with all fully connected layers attached with ReLU activation function.

Increasing the LSTM hidden size can capture more nonlinearity and intricate temporal pattern, especially the mixture of low-frequency and high-frequency dynamics. The optimum number of 192, multiple of 8, also fits with the tensor core operation. Hence, it matches better with the QBlade simulation.

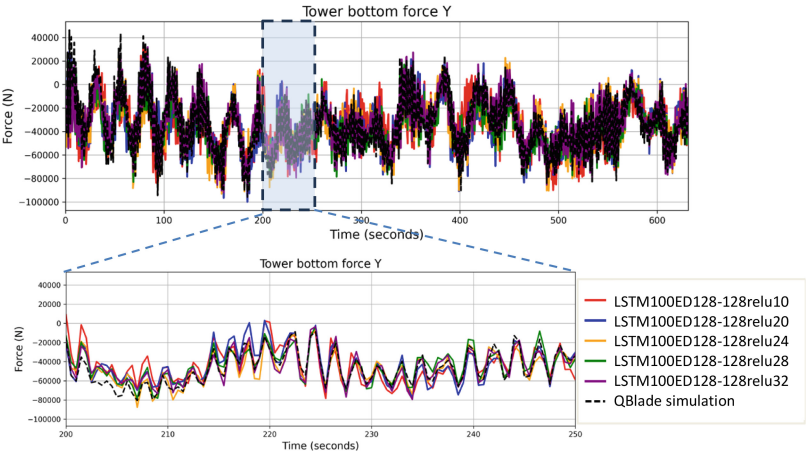

Fig. 8. Tower bottom force Y prediction with varying latent vector size.

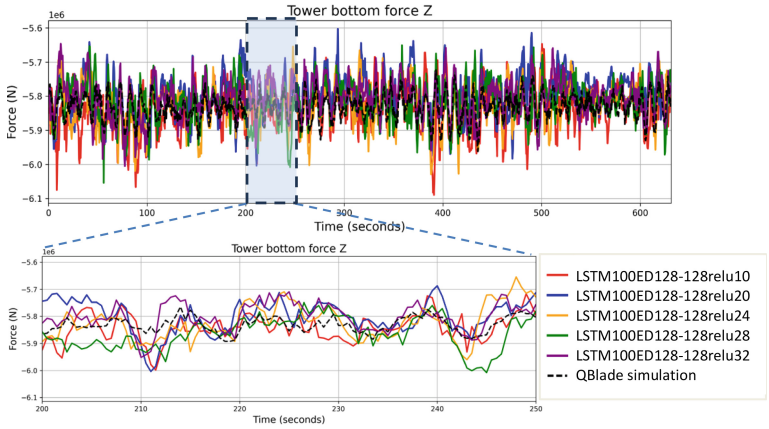

Fig. 9. Tower bottom force Z prediction with varying latent vector size.

Study on Residual Skip Connection. Based on the optimal value latent space vector 32 and LSTM hidden size 192, residual skip connection is applied for testing the effect of gradient flow. ELU consists of a small negative slope which can perform a better gradient flow than ReLU in the inner layer.

Figure 13, 14 and 15 reveal the residual skip connection can improve the prediction in finer detailed variation with better flow of information and gradient, and LSTM192ED128-128relu-elu32 can produce precise result as the baseline GATv2LSTM against the "truth value" QBlade simulation.

The prediction stage runs on the CPU 11th Gen Intel(R) Core(TM) i5-1145G7 @ 2.60 GHz 2.61 GHz. The first 100 s data (time window) is excluded from the calculation of Real-Time factor so 631.5 s is used for the reference value. A common performance

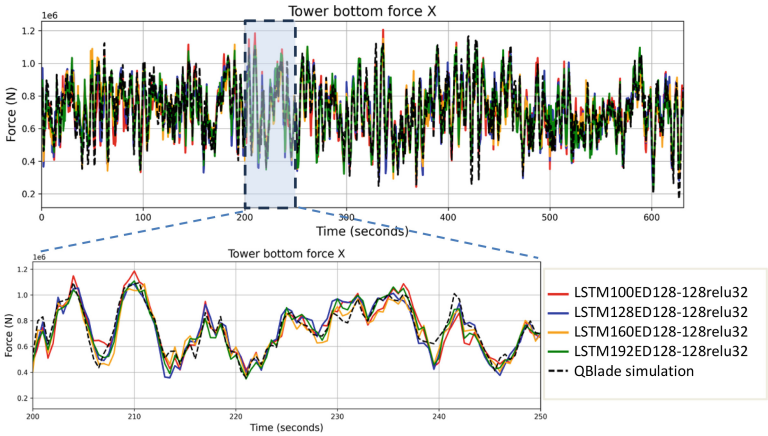

Fig. 10. Tower bottom force X prediction with varying LSTM hidden size.

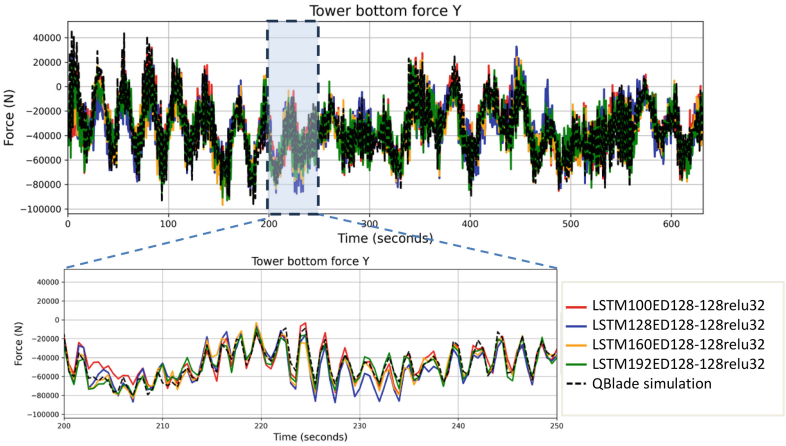

Fig. 11. Tower bottom force Y prediction with varying LSTM hidden size.

metric for machine learning prediction is Percent Bias (PBIAS) defined as

$$PBIAS = 100\% \times \frac{\sum(Q_{pred} - Q_{sim})}{\sum Q_{sim}} \tag{4}$$

where Q_{pred} is the prediction and Q_{sim} is the reference from QBlade simulation.

To conclude, most models shown in Table 3 can provide accurate prediction for the dominant modes force-aft (force X and moment Y) and side-side (force Y and moment X) forces and moments with ±5%. With about 50 times faster than the real-time data. The large differences are evaluated for moment Z (torsion mode) for sorting the suitable candidates for further study, and they are LSTM100ED128-128relu24, LSTM100ED128-128relu32, LSTM192ED128-128relu32, LSTM192ED128-128relu32res and LSTM192ED128-128relu-elu32res.

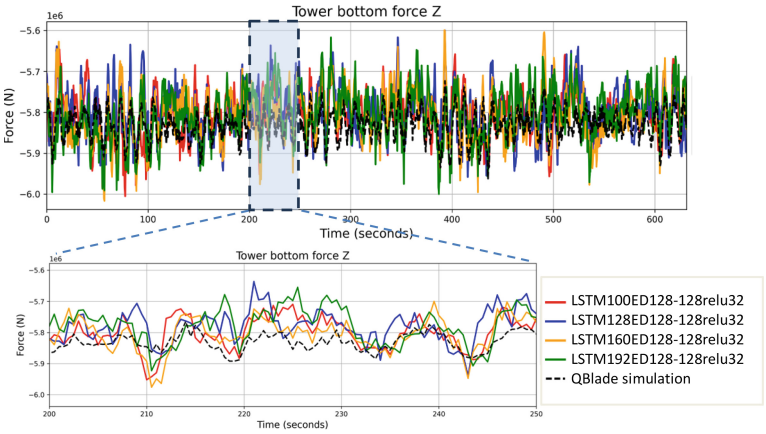

Fig. 12. Tower bottom force Z prediction with varying LSTM hidden size.

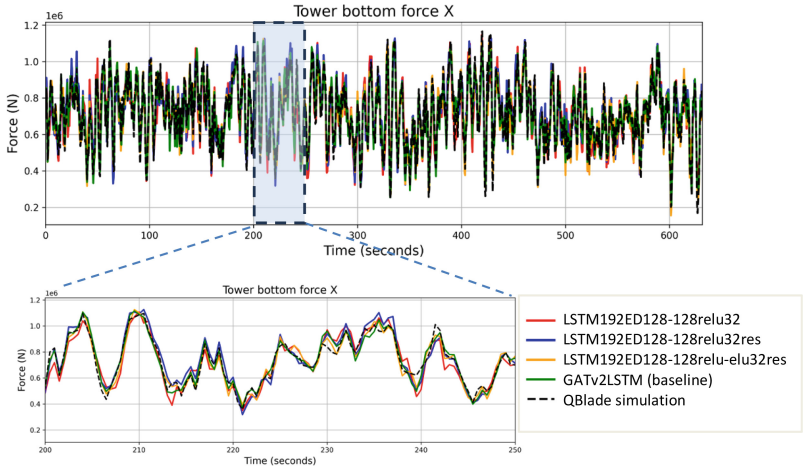

Fig. 13. Tower bottom force X prediction with the effect of residual skip connection.

The following section reveals the tractability of latent space vector comparing to the baseline model in high dimension, the detailed Singular Value Decomposition (SVD) analysis is used to demonstrate how the variance (structural energy) is preserved, and the MAC analysis is used to demonstrate how the mode shape is preserved.

Figure 16 shows the LSTM192ED128-128relu-elu32res can preserve the most singular values up to mode 18 as the baseline model. MAC require two set of data in the same dimension. Therefore, Principal Component Analysis is applied to the baseline model vector to reduce the dimension to the same as the latent vector. As the basis of reduced dimension vector of the baseline model can be different from the latent vector produced from the encoder, Procrustes analysis (function scipy.spatial.procrustes [25]) is used to align both set of vectors without changing the mode shape, and then followed

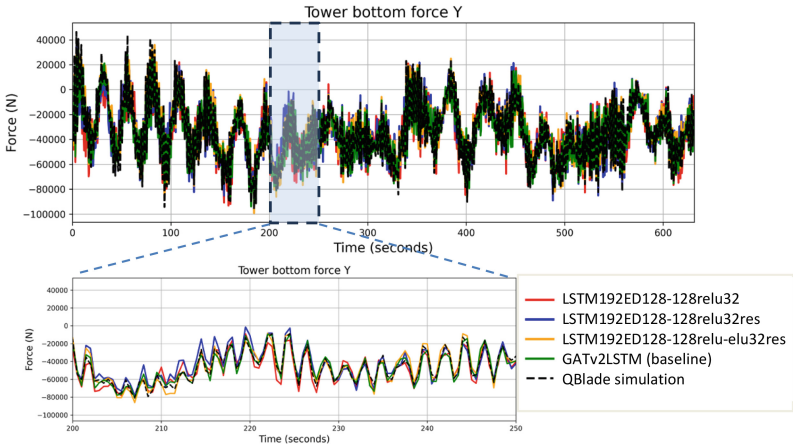

Fig. 14. Tower bottom force Y prediction with the effect of residual skip connection.

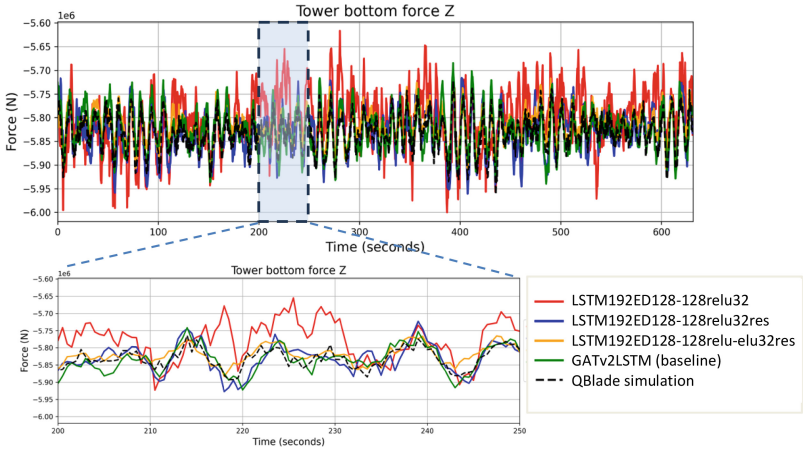

Fig. 15. Tower bottom force Z prediction with the effect of residual skip connection.

by MAC calculated as

$$MAC(\phi, \psi) = \frac{\left(\phi^T \psi\right)^2}{\left(\phi^T \phi\right)\left(\psi^T \psi\right)} \tag{5}$$

where ϕ is the eigenvector of latent space vector and ψ is the eigenvector of baseline model vector after Procrustes analysis.

Figure 17, 18, 19, 20 and 21 show the improvement of capturing mode shapes from model LSTM100ED128-128relu24 to LSTM192ED128-128relu-elu32res with MAC = 0.9 indicate highly correlated results. Particularly Fig. 20 and 21 show the significant improvement of skip residual connection from capturing 2 modes to at least 9 modes.

Table 3. Percent Bias and CPU execution wall time for the testing data set (631.5s)

	Percent Bias (%)							Real-Time Factor
Prediction model	force X (N/N)	force Y (N/N)	force Z (N/N)	moment X (Nm/Nm)	moment Y(Nm/Nm)	moment Z(Nm/Nm)	Wall time (s)	631.5/Wall time
GATv2LSTM (baseline)	0.01	1.03	−0.11	0.27	−0.09	−9.79	51.5	12.3
LSTM100ED128-128relu10	−0.08	2.01	0.15	0.60	−0.62	−53.13	11.6	54.4
LSTM100 ED128-128relu20	−0.20	1.77	−0.57	0.48	−1.11	−66.08	12.3	51.3
LSTM100 ED128-128relu24	1.31	2.30	−0.12	0.64	1.07	6.14	11	57.4
LSTM100 ED 128-128relu32	−1.23	−3.31	−0.42	−1.25	−2.25	−10.86	10.4	60.7
LSTM128ED128-128rel32	−0.33	2.76	−0.55	0.51	−1.20	32.67	14.5	43.6
LSTM160ED128-128relu32	−0.85	4.80	−0.42	1.10	−1.58	−28.33	12.2	51.8
LSTM192ED128-128relu32	−0.22	1.61	−0.46	0.50	−0.85	−26.46	13.3	47.5
LSTM192ED128-128relu32res	0.62	−0.88	−0.03	−0.42	0.54	−26.99	13.2	47.8
LSTM192ED128-128relu−elu32res	−0.49	−3.12	−0.26	−1.15	−1.14	−11.60	12.8	49.3

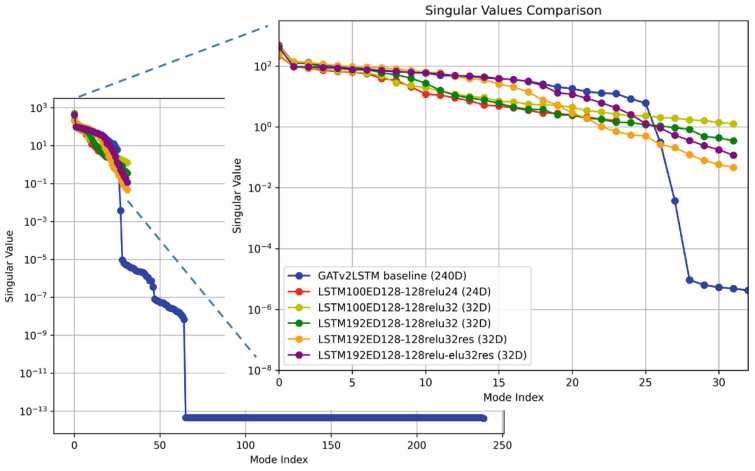

Fig. 16. Singular value comparison

This is because the high dimensional vector supplementing the information to the latent vector.

The disparity value of Procrustes analysis indicates similarity of two sets of mode shape data and smaller disparity means better fit. Figure 21 shows that LSTM192ED128-128relu-elu32res preserves the most system dynamics up to mode 13 with the smallest disparity of 0.112. In summary, LSTM192ED128-128relu-elu32res preserves the most dynamic properties of the high dimension vector of baseline model.

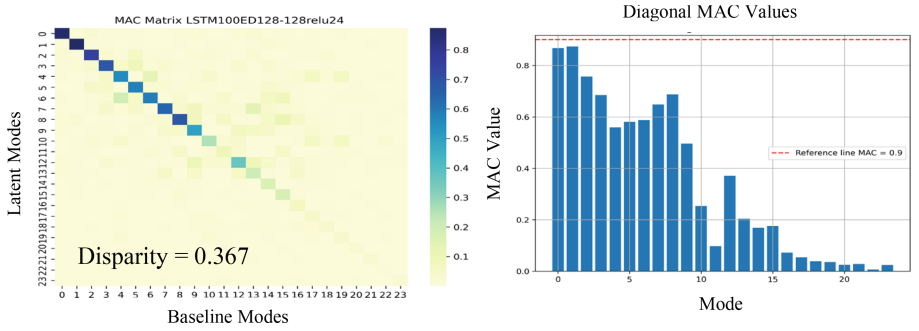

Fig. 17. MAC of LSTM100ED128-128relu24 model

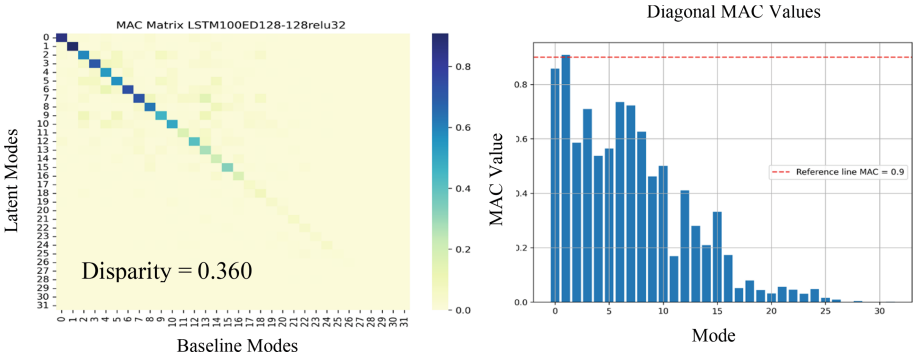

Fig. 18. MAC of LSTM100ED128-128relu32 model

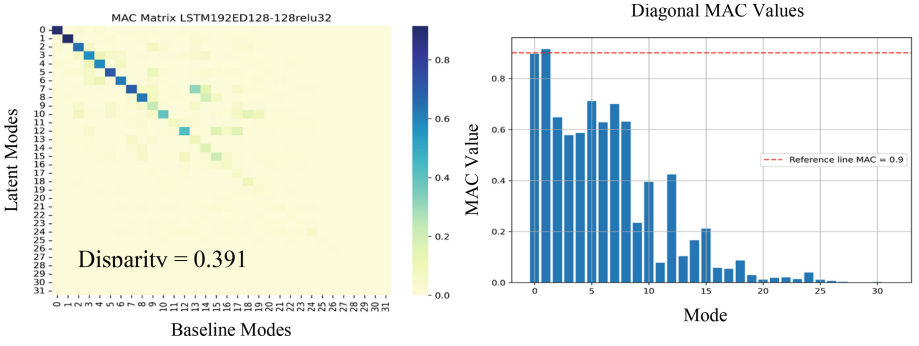

Fig. 19. MAC of LSTM192ED128-128relu32 model

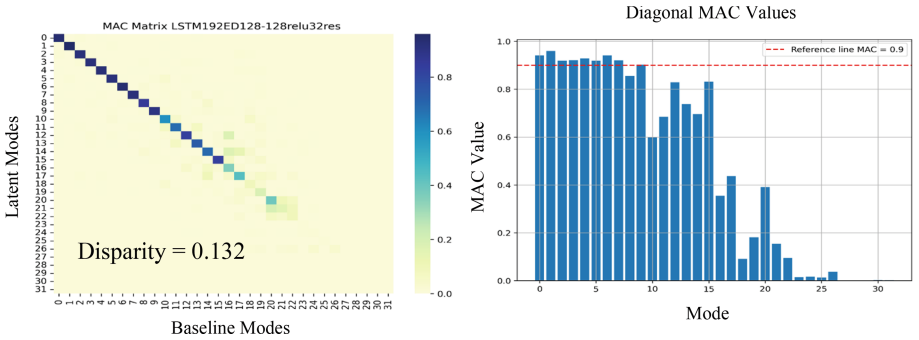

Fig. 20. MAC of LSTM192ED128-128relu32res model

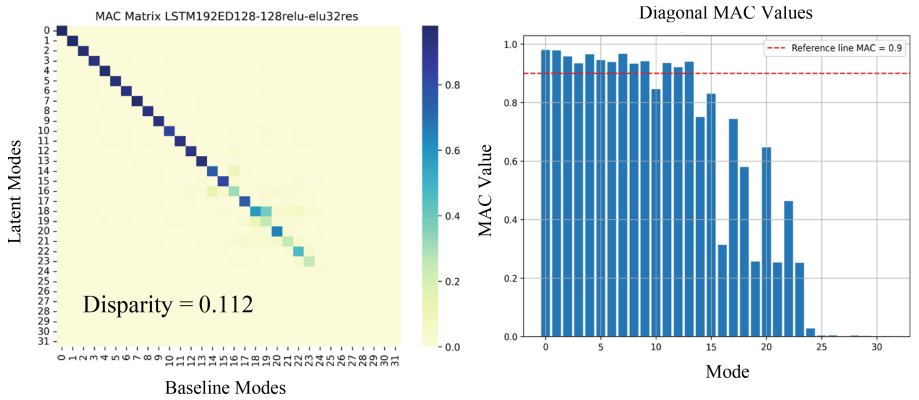

Fig. 21. MAC of LSTM192ED128-128relu-elu32res model

6 Conclusion

In this work, we present a novel GNN-Encoder-Decoder-LSTM for FOWT time series internal force prediction and provide a holistic analysis of the latent space vector characteristic comparing to the high dimensional vector. The addition of Encoder-Decoder can accelerate the prediction time by 4 times compared with the baseline model with only minimal difference in accuracy, and it is also about 50 times faster than real-time data set. Which is excellent for Digital Twin implementation. In terms of the quality of the latent vector from the Encoder-Decoder, the SVD and MAC analysis demonstrate the preservation of variance energy and the mode shapes up to mode 13 with the optimum model LSTM192ED128-128relu-elu32res. This result aligns with the neural network hyperparameter experiment prediction results of time series data. Which means the latent vector is tractable as the high dimensional graph embedding vector.

In future work, more loading combinations and scenarios will be included to extend the generalizability of the prediction model.

Acknowledgments. The first author would like to acknowledge the research grant of Research Attachment Programme from the Agency for Science, Technology and Research (A*STAR) Singapore.

References

1. Global Offshore Wind Report: Global Offshore Wind Report (2024)
2. Glaessgen, E., Stargel, D.: The digital twin paradigm for future NASA and US air force vehicles. In: Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA (2012)
3. Lein, C., Beitelschmidt, M.: Comparative study of model correlation methods with application to model order reduction. In: Proceedings 26th ISMA (International Conference on Noise and Vibration Engineering) (2014)
4. Jonkman, Jason, et al.: OpenFAST v3.0.0. https://github.com/OpenFAST/openfast/releases/tag/v3.0.0
5. Marten, D.: Qblade. https://qblade.org/
6. Tasora, A.: Time integration in CHRONO::engine, project::chrono technical documentation (2018). http://www.projectchrono.org/assets/white_papers/ChronoCore/integrator.pdf
7. Yung, K. H.-Y., Xiao, Q., Incecik, A., Thompson, P.: Mooring force estimation for floating offshore wind turbines with augmented Kalman filter: a step towards digital twin, in ASME 2023 5th International Offshore Wind Technical Conference (2023)
8. Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech. **25**(1), 539–575 (1993)
9. Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. **656**, 5–28 (2010)
10. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
11. Simpson, T., Dervilis, N., Couturier, P., Maljaars, N., Chatzi, E.: Reduced order modeling of non-linear monopile dynamics via an AE-LSTM scheme. Front. Energy Res. **11** (2023)
12. Simpson, T., Vlachas, K., Garland A., Dervilis N., Chatzi, E.: VpROM: a novel variational autoencoder-boosted reduced order model for the treatment of parametric dependencies in nonlinear systems. Sci. Rep. **14**(6091) (2024)
13. Yung, K. H.-Y., Xiao Q., Incecik A., Xing, X., Kang, C. W.: Structural surrogate modelling of a floating offshore wind turbine with physics-guided spatial-temporal graph neural network. In: EERA DeepWind Conference 2025, Trondheim, Norway (2025)
14. Robertson, A., Jonkman, J., Masciola, M., Song, H., Goupee, A., Coulling, A., Luan, C.: Definition of the Semisubmersible Floating System for Phase II of OC4.. National Renewable Energy Laboratory (2014)
15. European Union: Deliverable 1.1 Oceanographic and meteorological conditions for the design. LIFES50+ (2015)
16. International Electrotechnical Commision: Wind energy generation systems–Part 3-2: Design requirements for floating offshore wind turbines (2021)
17. Jonkman, J.: TurbSim User's Guide v2.00.00. National Renewable Energy Laboratory (2014)
18. Madsen, H., Larsen, T., Pirrung, G., Li, A., Zahle, F.: Implementation of the blade element momentum model on a polar grid and its aeroelastic load impact. Wind Energy Sci. **5**(1–27) (2020)
19. Brody, S., Alon, U., Yahav, E.: How attentive are graph attention networks?. In: International Conference on Learning Representations (2022)

20. PyTorch Geometric: conv.GATv2Conv. https://pytorch-geometric.readthedocs.io/en/2.5.1/generated/torch_geometric.nn.conv.GATv2Conv.html
21. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. **32**, 8024–8035 (2019)
22. Fey, M. and Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric, arXiv preprint arXiv:1903.02428
23. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
24. Saverin, J., Perez-Becker, S., Luna, R.B.D., Marten, D., Gilloteaux, J.-C., Kurnia, R.: D1.2. Higher order hydroelastic module (2021)
25. SciPy: scipy.spatial.procrustes. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.procrustes.html