Exploring unseen 3D scenarios of physics variables using machine learning-based synthetic data: an application to wave energy converters

César Quilodrán-Casas[a,b], Qian Li[c], Ningbo Zhang[c], Sibo Cheng[a], Shiqiang Yan[c], Qingwei Ma[c] and Rossella Arcucci[a,b]

[a]Data Science Institute, Imperial College London, London, United Kingdom
[b]Department of Earth Science and Engineering, Imperial College London, London, United Kingdom
[c]City University, London, United Kingdom

ARTICLE INFO

Keywords:
Generative models
Synthetic data
Wave energy converters
Model surrogate

ABSTRACT

This work aims to use machine learning to produce synthetic data of wave energy converters from time-expensive 3D simulations based on computational fluid dynamics models. Whilst the potential to study, understand and take advantage of these converters for renewable energy is immense, the simulations to analyse the response of these systems to incoming waves are lengthy and computationally expensive to obtain. Here, we explore the use of a β-VAE and a Principal Components-based adversarial autoencoder for generating new synthetic data. The compression plus the generation of synthetic data introduces an exceptionally fast model surrogate of the original simulation and delivers more samples of either dynamic viscosity and velocity fields, enlarging the design space. The new generated synthetic samples can have a speed up from 5 to 6 orders of magnitude. The newly design space can be used to improve the prediction of dynamic viscosity given the velocity fields. The generative model has the potential to capture the transition and the new physical phenomena under extreme initial conditions.

1. Introduction

Due to the increasing demand for clean energy, various renewable energy resources are being explored, among which wave energy is one of the topics with the greatest potential (Glendenning, 1977). Various forms of oscillating Wave Energy Converter (WEC) devices have been developed to capture wave energy to generate electricity (Antonio, 2010).

WECs are devices that convert the kinetic and potential energy associated with a moving ocean wave into useful mechanical or electrical energy. A point absorber is a floating structure that absorbs energy from all directions through its movements at or near the water's surface. It converts the motion of the buoyant top relative to the base into electrical power. An interesting interaction to predict is how the dynamic viscosity around the point absorber reacts to the stimulus of an incoming wave, and this can be predicted with the velocity fields and the dynamic viscosity from the previous time level (Jin, Patton, and Guo, 2018).

In the process of studying a complete WEC system, it is essential to obtain a general and applicable hydrodynamic description of how the device interacts with the incident waves. This mathematical description is important to suggest the design of the power take-off (PTO), as well as the development of the control system since these WEC subsystems are influenced by the dynamic interaction that the WEC device has with the movement of the waves (Son and Yeung, 2017). However, Computational Fluid Dynamics (CFD) simulations of how these systems behave due to ocean wave perturbations can be computationally expensive to run and time-consuming. Often it can produce a small number of samples to work with, yielding poor predictors for a predictive model of the dynamic viscosity due to the lack of a high number of samples. Therefore, an attractive solution is to generate new simulations, at a considerable speedup, that learn from the original CFD simulations.

*Corresponding author
**Principal corresponding author
c.quilodran@imperial.ac.uk (C. Quilodrán-Casas); qian.li.8@city.ac.uk (Q. Li); ningbo.zhang.2@city.ac.uk (N. Zhang); sibo.cheng@imperial.ac.uk (S. Cheng); shiqiang.yan@city.ac.uk (S. Yan); q.ma@city.ac.uk (Q. Ma); r.arcucci@imperial.ac.uk (R. Arcucci)
ORCID(s):

A classic approach to this difficult problem is Data Augmentation (DA). In DA, the network is trained using additional synthetic data. DA was introduced in object recognition in LeCun, Bottou, Bengio, and Haffner (1998). The advantages of DA are that it increases the size of the training data; eliminates the overfitting problem; and it makes the network more robust to data variations that may exist in any real-world application. The basic idea behind DA is to apply transformations so that the semantics of the labels associated with the data does not change. By training the network with this extra data, one would expect, its performance on unseen data to be enhanced.

Quilodrán-Casas, Arcucci, Mottet, Guo, and Pain (2021) used generative models to create stable rollouts for air pollution. The experimental design space of microfluidics has also been augmented using generative networks (Chagot, Quilodrán-Casas, Kalli, Kovalchuk, Simmons, Matar, Arcucci, and Angeli, 2022). Other successful implementations of generative-based augmentation are Karras, Aila, Laine, and Lehtinen (2017); Berthelot, Schumm, and Metz (2017); Radford, Metz, and Chintala (2015).

The generation of high-quality synthetic data allowed the augmentation of small-sample data sets (Forestier, Petitjean, Dau, Webb, and Keogh, 2017; Hoffmann, Bar-Sinai, Lee, Andrejevic, Mishra, Rubinstein, and Rycroft, 2019). Although the use of the synthetic data needs to be developed and adapted for each case (Chen, Lu, Chen, Williamson, and Mahmood, 2021), it can be a powerful tool to increase the robustness and adaptability of data-driven models (Yoon, Jordon, and Schaar, 2018; Quilodrán-Casas et al., 2021). However, Machine Learning (ML) often requires representative data to be effective (Zhou, Pan, Wang, and Vasilakos, 2017; Li, Zhang, Chen, Shen, and Niu, 2023; Sadeghi, Nguyen, Hsu, and Sorooshian, 2020; Razavi, 2021).

Recent advances in ML have shown strong predictive power to determine complex correlations and find patterns between inputs and outputs (Goodfellow, Bengio, and Courville, 2016). ML has been employed in wave energy studies previously. Rodriguez-Delgado, Bergillos, and Iglesias (2019) have used neural networks to assess the efficiency of WECs. Li, Yuan, and Gao (2018) used Deep Learning (DL) for assessing the energy absorption of a WEC. Sarkar, Contal, Vayatis, and Dias (2016). Sclavounos and Ma (2018) and Mousavi, Ghasemi, Dehghan Manshadi, and Mosavi (2021) used ML for forecasting the time series response of WEC and wave energy conversion rates.

However, generative methods to augment datasets have not been used to augment data and reproduce unseen physics conditions. Here, we use data from a high-fidelity CFD simulating 3D velocity and dynamic viscosity of WECs in some scenarios and we use ML models to develop a surrogate model to reproduce unseen physics conditions for WECs. Due to the problem complexity and the high dimensionality, running this CFD simulation can be computationally expensive and time-consuming. To tackle this bottleneck, in this paper, a β-Variational Autoencoder (VAE) and a Principal Components-based Adversarial Autoencoder (PC-AAE) are used to generate synthetic data to enlarge the experimental dataset, explore unseen scenarios of this Three-dimensional (3D) simulation in a fast manner. The generated data can also be used to train a ML surrogate model for predicting future wave dynamics. To the best of our knowledge, this is the first reported attempt to obtain synthetic data of WECs 3D CFD simulations using generative models.

The contribution of this paper lies in the use of generative networks such as β-VAE and PC-AAE for the generation of unseen synthetic data to expand the relationship between velocity and dynamic viscosity in CFD models, bypassing running a different CFD simulation in a supercomputer to create more samples. These generative models allow increasing the design space and access to sampled data from a matched distribution of the original data.

This paper is organised as follows. Section 2 describes the implementation of the β-VAE and the PC-AAE. Section 3 describes the irregular wave test case of a point absorber WEC. Section 4 shows the results and discussion of the samples of velocity fields and dynamic viscosity generated by these networks. And finally, Section 5 presents a summary, conclusion and future work.

2. Generative models

In this paper, two different generative models are used: β-VAE and PC-AAE. These two methods were chosen as they are well-established and well-documented for producing high-fidelity results.

2.1. β-Variational autoencoder (β-VAE)

Autoencoder (AE)s were developed to reconstruct high-dimensional data using a neural network model composed of an encoder and a decoder. AEs can also reduce the dimensionality of the system with the encoder mapping the input onto a bottleneck layer. Furthermore, a β-VAE instead of mapping onto a fixed vector, maps the input onto an arbitrary distribution (Higgins, Matthey, Pal, Burgess, Glorot, Botvinick, Mohamed, and Lerchner, 2016). The β-VAE is a modification of the VAE with a special emphasis to discover disentangled latent factors. Following the same

incentive in VAE, the probability of generating real data is maximised whilst maintaining the distance between the real and estimated posterior distributions small.

Let $\mathcal{Q}$ and $\mathcal{P}$ denote the encoder and decoder, respectively. Moreover, let $q(\mathbf{z}|\mathbf{x})$ and $p(\tilde{\mathbf{x}}|\mathbf{z})$ denote the encoding and decoding distributions, respectively., where $\mathbf{x}$ is the input vector, $\tilde{\mathbf{x}}$ is the reconstructed input, and $\mathbf{z}$ is the latent space. As suggested by Makhzani, Shlens, Jaitly, Goodfellow, and Frey (2015), a Gaussian posterior can be used assuming that $q(\mathbf{z}|\mathbf{x})$ is a Gaussian distribution, where its mean μ and variance σ are predicted by the encoder $\mathcal{Q}$ by adding two dense layers of means μ and $log\ \sigma$ to the final layer of the encoder $\mathcal{Q}$, and return $\mathbf{z}$ as a vector of samples (Kingma and Welling, 2013). To ensure that $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu, \sigma^2)$, the aggregated posterior, the reparameterisation trick described by Kingma and Welling (2013) was used for backpropagation.

The minimisation of the Kullback-Leibler Divergence Score (KL) loss ($\mathcal{L}^{KL}$) quantifies how much the probability distribution a differs from the probability distribution b as:

$$\mathcal{L}^{KL}(a, b) = -\sum a\ log\left(\frac{b}{a}\right) \tag{1}$$

where, in this case, $a = q(\mathbf{z}|\mathbf{x})$ and $b = g(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$, the arbitrary prior, and $\mathbf{I}$ is the identity matrix. In other words, we expect the latent distribution $g(\mathbf{z})$ to approximate a centred and standard Gaussian distribution. Adam is used as the optimiser (Kingma and Welling, 2013). The total loss $\mathcal{L}^{\theta}$ is then defined as $\mathcal{L}^{\theta} = \lambda\mathcal{L}^{KL} + \mathcal{L}^{mse}$, where $\lambda = 0.001$ acts as a regulariser. The reconstruction error $\mathcal{L}^{mse}$ is the mean squared error defined as:

$$\mathcal{L}^{mse} = ||\mathbf{x} - \tilde{\mathbf{x}}||^2 \tag{2}$$

where $\tilde{\mathbf{x}}$ is the reconstructed input of experimental data, defined as $\tilde{\mathbf{x}} = \mathcal{P}(\mathcal{Q}(\mathbf{x}))$. The inputs were scaled between 0 and 1.

The implementation of the β-VAE is in Python using `pytorch` (Paszke, Gross, Massa, Lerer, Bradbury, Chanan, Killeen, Lin, Gimelshein, Antiga et al., 2019) and the `pytorch-lightning` wrapper (Falcon et al., 2019). The algorithm of the generation of synthetic samples using β-VAE is shown in Algorithm 1.

Algorithm 1: β-VAE

Require : $\theta^{\mathcal{Q}}$ and $\theta^{\mathcal{P}}$ trainable parameters for encoder $\mathcal{Q}$, and decoder $\mathcal{P}$, respectively; batch sizes $m^{\mathcal{Q}}$ for $\mathcal{Q}$; number of epochs k

for $epochs = 0, \dots, k$ **do**

- Match a latent vector of size $m_{\mathcal{Q}}$ to a normal distribution $\mathbf{z} \sim \mathbf{q}(\mathbf{z}|\mathbf{x})$
- Calculate the $\mathcal{L}^{KL}(q(\mathbf{z}|\mathbf{x}), g(\mathbf{z}))$
- Calculate the $\mathcal{L}^{mse}$
- Reconstruction error $\mathcal{L}^{\theta} = \mathcal{L}^{KL} + \mathcal{L}^{mse}$
- Update the β-VAE parameters ($\theta^{\mathcal{Q}}$ and $\theta^{\mathcal{P}}$) via: $\theta^{\beta-VAE} \leftarrow \text{Adam}(\mathcal{L}^{\beta-VAE})$

until convergence detach Decoder $\mathcal{P}$ and use it to generate samples

2.2. PC-based adversarial AE (PC-AAE)

As described by Lever, Krzywinski, and Altman (2017), Principal Component Analysis (PCA) is an unsupervised learning method that simplifies high-dimensional data by transforming it into fewer dimensions. The PCA consists in decomposing $\mathbf{x}$ as $\mathbf{x} = \mathbf{P}\mathbf{\Pi} + \bar{\mathbf{x}}$ where $\mathbf{P} \in \mathfrak{R}^{n \times n}$ are the Principal Components (PC)s of $\mathbf{x}$; $\mathbf{\Pi} \in \mathfrak{R}^{n \times m}$ are the Empirical Orthogonal Functions (EOF)s; and $\bar{\mathbf{x}}$ is the mean vector of the model. The dimension reduction of the system comes from truncating $\mathbf{P}$ at the first τ PCs as $\mathbf{x}_{\tau} = \mathbf{P}_{\tau}\mathbf{\Pi}_{\tau} + \bar{\mathbf{x}}$, with $\mathbf{P}_{\tau} \in \mathfrak{R}^{n \times \tau}$ and $\mathbf{\Pi}_{\tau} \in \mathfrak{R}^{\tau \times m}$. To further reduce the system dimension, a number of recent researches combined PCA with deep learning AEs (see Cheng, Jin, Harrison, Quilodrán-Casas, Prentice, Guo, and Arcucci (2022); Cheng, Chen, Anastasiou, Angeli, Matar, Guo, Pain, and Arcucci (2023); Gong, Cheng, Chen, Li, Quilodrán-Casas, Xiao, and Arcucci (2022)). In this study, the principle components are used to train an Adversarial Autoencoder (AAE) (Makhzani et al., 2015). The functional of our PC-AAE is defined as:

$$f^{PC-AAE} : \mathbf{P}_{t_k} \rightarrow \tilde{\mathbf{P}}_{t_k} \tag{3}$$

where $\mathbf{P}_{\mathbf{t_k}}$ are the scaled PCs time series between -1 and 1 at time-level k. The AE consists of an encoder $\mathcal{Q}$ and a decoder $\mathcal{P}$, both mirrored fully-connected networks, where the scaled reconstructed PCs $\tilde{\mathbf{P}}_{t_k} = \mathcal{P}(\mathcal{Q}(\mathbf{P}_{\mathbf{t_k}}))$. Let $q(\mathbf{z}|\mathbf{P})$ and $p(\tilde{\mathbf{P}}|\mathbf{z})$ be the encoding and decoding distributions, respectively. As suggested by Makhzani et al. (2015), we use a Gaussian posterior and assume that $q(\mathbf{z}|\mathbf{P})$ is a Gaussian distribution, where its mean and variance are predicted by the encoder $\mathcal{Q}$. This is achieved by adding two dense layers of means μ and $log\ \sigma$ to the final layer of the encoder $\mathcal{Q}$, and return $\mathbf{z}$ as a vector of samples. This is achieved similarly to the reparameterisation trick described above for β-VAE.

The adversarial training of PC-AAE includes a discriminator $\mathcal{D}^A$ to distinguish between the real samples, given by an arbitrary prior $g(\mathbf{z})$, and fake samples, given by $q(\mathbf{z}|\mathbf{P})$. Therefore, the adversarial autoencoder is regularised by matching $g(\mathbf{z})$ to $q(\mathbf{z}|\mathbf{P})$. The ground truth PCs $\mathbf{P}$ are fed to the discriminator as real sequences (ground truth). Let, $\mathcal{D}^A(\alpha, \gamma)$ represent the discriminator function with an input α and a target label γ such that, for $\alpha = \mathbf{z} \sim q(\mathbf{z}|\mathbf{P})$, $\gamma = 1$ and for $\alpha = \hat{\mathbf{z}} \sim p(\mathbf{z})$, $\gamma = 0$, where $\hat{\mathbf{z}}$ is the latent space sampled from $g(\mathbf{z})$. The training of $\mathcal{D}^A$ is based on the minimisation of the binary cross-entropy loss ($\mathcal{L}^{bce}$), using the Nesterov Adam optimizer (Nadam) (Dozat, 2016). The adversarial losses $\mathcal{L}^{adv}$ for $\mathcal{D}^A$ and f^{PC-AAE} are then defined as:

$$\mathcal{L}^{adv}_{\mathcal{D}^A}(\mathbf{P}) = \mathcal{L}^{bce}_{\hat{\mathbf{z}}\sim g(\mathbf{z})}(\mathcal{D}^A(\hat{\mathbf{z}}), 1) + \mathcal{L}^{bce}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{P})}(\mathcal{D}^A(\mathbf{z}), 0) \tag{4}$$

$$\mathcal{L}^{adv}_{f^{PC-AAE}}(\mathbf{P}) = \mathcal{L}^{bce}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{P})}(\mathcal{D}^A(\mathbf{z}), 1) + \mathcal{L}^{mse}(\tilde{\mathbf{P}}, \mathbf{P}) \tag{5}$$

where $\mathcal{L}^{mse}$ is the Mean Squared Error (MSE) between $\tilde{\mathbf{P}}$ and $\mathbf{P}$. The algorithm of the generation of synthetic samples using PC-AAE is shown in Algorithm 2.

Algorithm 2: PC-AAE

Require : $\theta^{\mathcal{Q}}$, $\theta^{\mathcal{P}}$, and $\theta^{\mathcal{D}^A}$ trainable parameters for encoder $\mathcal{Q}$, decoder $\mathcal{P}$, and discriminator $\mathcal{D}^A$, respectively; number of discriminator iterations per AE iteration ($n_{\mathcal{D}^A}$); batch sizes $m_{\mathcal{Q}}$ for $\mathcal{Q}$; number of epochs k

for $epochs = 0, \dots, k$ **do**

  Discriminator training: **for** $i = 0, \dots, n_{\mathcal{D}^A}$ **do**

    Sample a latent vector of size $m_{\mathcal{Q}}$ from a normal distribution $\hat{\mathbf{z}} \sim \mathbf{g}(\mathbf{z})$, plus Gaussian noise

    Fake samples $\mathbf{z} \leftarrow \mathcal{Q}_{\theta^{\mathcal{Q}}}(\mathbf{P})$

    Update the discriminator $\mathcal{D}^A$ to differentiate between real and fake sample via: $\theta^{\mathcal{D}^A} \leftarrow \text{NAdam}(\mathcal{L}^{adv}_{\mathcal{D}^A})$

  Update the AE PC-AAE parameters ($\theta^{\mathcal{Q}}$ and $\theta^{\mathcal{P}}$) via: $\theta^{PC-AAE} \leftarrow \text{NAdam}(\mathcal{L}^{adv}_{PC-AAE})$

until convergence Detach Decoder $\mathcal{P}$ and use it to generate samples of $\mathbf{P}$

Then the reconstruction $\tilde{\mathbf{x}}$ to the physical space is given by:

$$\tilde{\mathbf{x}} = \tilde{\mathbf{P}}\mathbf{\Pi} + \bar{\mathbf{x}} \tag{6}$$

The implementation of the PC-AAE is in Python using `tensorflow` (Abadi, Agarwal, Barham, Brevdo, Chen, Citro, Corrado, Davis, Dean, Devin et al., 2016) and the `keras` wrapper (Chollet et al., 2015).

3. Dataset generation methodology

The dataset applied in this study for the ML training is a 3D unstructured grid CFD simulation of a point absorber WEC. Here $\mathbf{x} = [U, \nu_t]$ where $U = [u, v, w]$ is the whole velocity field and u, v and w are the velocity components in the X-axis, Y-axis, and Z-axis, respectively; and ν_t is the dynamic viscosity.

These high-fidelity CFD simulating results are used for data generation and also as reference results for validation. An in-house domain decomposition hybrid solver, `qaleFOAM`, based on the open-source platform `OpenFOAM` (Jasak, Jemcov, Tukovic et al., 2007) is adopted in this study where a two-phase incompressible Navier-Stokes (NS) solver couples with the Quasi Lagrangian-Eulerian Finite Element Method (QALE-FEM) solver based on the fully nonlinear potential theory (Ma and Yan, 2006, 2009; Yan and Ma, 2007, 2010; Yan, Ma, and Cheng, 2011). The main target of this novel model is to boost the computational efficiency while maintaining the efficiency and details about this hybrid model can be found in (Li, Wang, Yan, Gong, and Ma, 2018) and (Yan, Wang, Wang, Ma, and Xie, 2020).

Besides, several investigations using this method in various working scenarios have demonstrated the effectiveness of the proposed model Yan, Li, Wang, Ma, Xie, and Stoesser (2019) and Yan et al. (2020). Even though significant improvement has been seen by applying this hybrid method, the simulation of turbulent effect for the real engineering issue is still very computationally expensive. This contributes to the main motivation of coupling the CFD code with the technology of machine learning.

In the Reynolds-averaged Navier Stokes (RANS) model employed in this study, an ensemble averaging method is applied to the unsteady turbulent flow modelling. This hypothesis introduces the macroscopic representations of the micro-scale fluctuating flow. It offers access to model the overall effects of small vortexes by correlations and meanwhile, resolves the larger eddies through the numerical simulation. where ν is the constant molecular viscosity and $\nu_T(d_i, t)$ is the spatial-temporal dependent turbulent/eddy viscosity, and together they compose the effective viscosity $\nu_{eff}(d_i, t)$:

$$\nu_{eff}\left(d_i, t\right) = \nu + \nu_T\left(d_i, t\right) \tag{7}$$

where d_i and t represent a point in space at time-step t.

As we are more interested in water physics, the kinematic viscosity is supposed to transfer to the dynamic viscosity ν_t before being used in the ML training of generative methods by the following equation:

$$\nu_t = \nu\left[\rho_{air}\left(1-\alpha\right) + \rho_{water}\alpha\right] \tag{8}$$

where water is playing a dominant role and α is the phase fraction in the two-phase flow; ν is the kinematic viscosity and ρ_{water} and ρ_{air} are the water and air density, respectively. Regarding the boundary condition setting, the no-slip boundary condition is applied on the bottom boundary with the total pressure specified on the top boundary in the NS domain. For the subdomain configuration, in the fully nonlinear potential domain, there are wave generation and absorption boundaries employed at the inlet and outlet boundary, respectively. Benefiting from the proposed hybrid model, the turbulent viscosity is only considered in a refined relatively smaller zone which is the NS domain.

In this CFD simulation, a $k-\omega$ Shear Stress Transport (SST) turbulent model that belonged to the RANS equation catalogue is applied in this study. Here, k is the turbulence kinetic energy and ω is the specific rate of dissipation of the turbulence kinetic energy k into internal thermal energy. The WEC surface wall treatment is always one of the biggest challenges raised in turbulent flow simulation, which can be classified into two categories: the Low-Reynolds number (LR) models and High-Reynolds number (HR) models. The LR approach accompanied by a wall function is targeting at the sublayer where exists a local low turbulent Reynolds number. One alternative to wall functions is to adopt a fine-grid configuration that allows the application of a laminar flow boundary condition. To reach the viscous sublayer, the normalised distance (y^+) from the first mesh cell centre to the body surface is supposed to be around 1, where $y^+ = u_* y_w/\nu_{eff}$. In numerical practice, the desired y^+ is usually obtained through consistent trials. However, the HR model can cope with a much larger y^+ (~ 30) which integrates with a log-law to estimate the gradient approaching the body wall. It should be noted that the first computational mesh should be placed either in the log-layer or the viscous sublayer but not in-between (Utyuzhnikov, 2005), since none of the categories can deal with the buffer layer where both viscous and Reynolds stresses are significant. All these factors further contribute to the complicity in the turbulent flow modelling and also can be a source of error since it highly relies on the experiences of the user. Therefore, by cooperating with ML, this study is aimed to deal with these difficulties by employing a surrogate prediction for RANS turbulence eddy viscosity.

3.1. CFD Simulation Configurations

In the numerical simulation, a rectangular computational domain is adopted. A Two-dimensional (2D) side view of the computational domains is given in Figure 1. A series of numerical simulations targeted at the hydrodynamics performances of the three principal motions (surge, heave, and pitch) are conducted.

As the important role played by the CFD dataset, its accuracy is examined before transferring into the ML model. To test the viability of the results generated by the adopted `qaleFOAM` solver, the numerical results have been compared to the experimental measurement carried out by Todalshaug, Ásgeirsson, Hjálmarsson, Maillet, Möller, Pires, Guérinel, and Lopes (2016) for the same wave energy devices under the regular wave conditions, in which good agreements have been achieved in the structure motion response. Figure 2 demonstrates the wave propagation near the WEC and also the pressure profile on the WEC surface. Besides, the mesh configuration of the computational domain is generated by the `SnappyHexMesh` tool with the refined zone around the free surface and structure. The CFD work is highly dependent

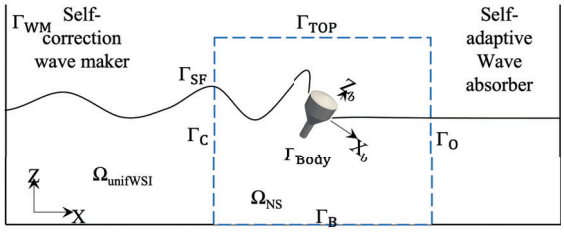


Figure 1: Sketch of the computational domain with boundaries

on the mesh resolution. Therefore, for each wave state, the convergence test against mesh resolution is performed to identify the optimised mesh configuration with a minimal computational cost. In the turbulent model, the initial values of k and ω are set at the inlet boundary with the WEC surface treatment using the low-Reynolds-number approach accompanied by a wall function.

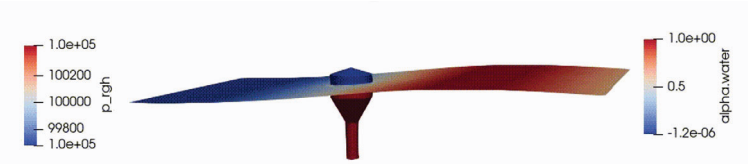
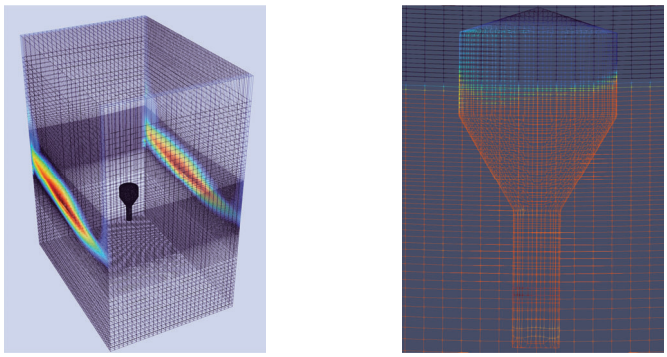


Figure 2: Wave surface profile around the buoy

The working condition considered in this simulation is an irregular wave generated by the JONSWAP spectrum (Hasselmann, Barnett, Bouws, Carlson, Cartwright, Enke, Ewing, Gienapp, Hasselmann, Kruseman et al., 1973). The significant wave height is H=8.8 m and the significant wave period is T=11 s with the numerical wave tank depth d=50 m. Within certain mesh configuration, the time step size Δt is automatically determined by using the fixed Courant number C_0 ($C_0 = \left(u_{max}\Delta d\right)/\Delta t$, where Δd is the mesh size and u_{max} is the largest velocity value at the current time step). This simulation has $n = 60$ time-levels, with $M = 851101$ nodes. Each node contains a scalar feature kinematic viscosity ν_t and a three-dimensional velocity vector U. The 4 fields account for each node containing $m = 3404404$ features. A snapshot of the dataset is shown in Figure 3 showcasing the regular mesh and the irregular mesh around the buoy.



(a) 3D-view of the WEC CFD unstructured mesh

(b) Close-up to the unstructured mesh around the WEC within the global mesh

Figure 3: 3D and 2D view of a snapshot of the WEC CFD simulation

4. Results

Four different experiments were performed. For the PC-AAE we constructed the PC-space for inputs in three different ways:

- PC-AAE (Unut_sep): full velocity field U and ν_t separated
- PC-AAE (uvwnut): 3 components of the velocity field u, v, w and the dynamic viscosity ν_t separated.
- PC-AAE (Unut): Full velocity field U and ν_t together

The rationale behind this design is to show how the PC-space can be constructed and its effects on the training and reconstruction of the physical space. If the velocity field U is considered fully along with the ν_t then the values of U, being larger than the ones of ν_t, will give more weight towards U. However, decomposing the PC-space into U and ν_t; or u, v, w and ν_t, the balance the reduced-space. For the β-VAE, no dimension reduction like PCA was applied and each variable was independently standardised:

- VAE: 3 components of the velocity field u, v, w and the dynamic viscosity ν_t separated.

Therefore, the networks with PC-AAE are small and deal with tens or hundreds of inputs, whilst the β-VAE is large and has ∼3.4M inputs.

For each one of these combinations, the data were standardised by its mean and standard deviation of the field. The training data set is 80% of the original simulation data and the test data is 20%, These datasets were shuffled randomly.

For each one of these experiments we tested their accuracy in the ground truth reconstruction (see Section 4.1), then we used them as a model surrogate and we tested the accuracy and efficiency of seen and unseen scenarios (see Section 4.2)

4.1. Reconstruction of ground truth using generative networks

To assess the effectiveness of the generative networks β-VAE and PC-AAE, the test data of the ground truth is reconstructed via the AEs. The test data is composed of 20% of the time-levels from the original CFD simulation, i.e. 12 time-levels.

Figure 4 shows the Kernel density estimator (KDE) for ground truth (blue), and the reconstructed test data, predicted by different generative methods, of the average values for each node over all samples. There is a good agreement between the reconstruction given by the generative models and the ground truth. Moreover, Figure 5 shows the excellent overlapping per time-level between the ground truth and the generative methods when the test data is reconstructed.

The mutual information (Mutual Information (MI)) between distributions was calculated using:

$$MI(U;V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} \frac{U_i \cap V_j}{N} log \frac{U_i \cap V_j}{\mid U_i \mid\mid V_j \mid} \tag{9}$$

where U and V are the distributions of ground truth values of the test dataset and the predicted values by the generative methods, respectively. Here, a Normalised Mutual Information (NMI) (Strehl and Ghosh, 2002) is used where normalisation of the MI score scales the results between 0 (no mutual information) and 1 (perfect correlation). The NMI is then defined as:

$$NMI(U;V) = \frac{MI(U;V)}{H(U)H(V)} \tag{10}$$

where $H(U)$ and $H(V)$ are the entropies of U and V, respectively.

Another metric to assess the fidelity of the reconstruction of the test dataset is Normalised Root Mean Squared Error (NRMSE). The NRMSE for all points at all time levels is defined by:

$$NRMSE = \frac{\sqrt{||\mathbf{y_{GT}} - \mathbf{y_{pred}}||^2}}{max(\mathbf{y_{GT}}) - min(\mathbf{y_{GT}})} \, . \tag{11}$$

where y_{GT} and y_{pred} are the unravelled values of the ground truth and predicted values by the generative methods, respectively.

A summary of the different metrics is presented in Table 1.

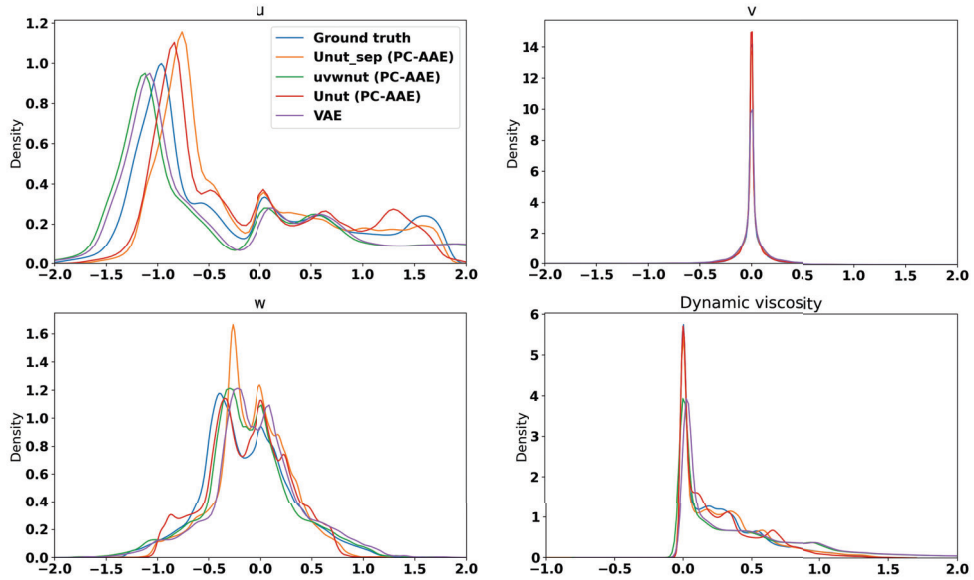


Figure 4: KDE for ground truth (blue), and the reconstructed test data, predicted by different generative methods, of the average values for each node over all samples.

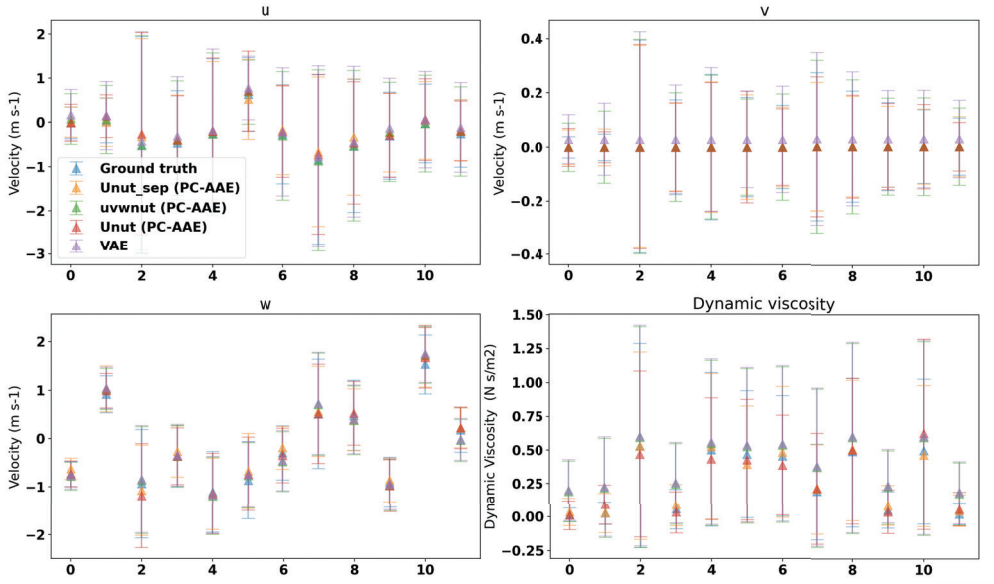


Figure 5: Error bars for reconstructed test data, averaged over all data points.

4.2. Model surrogates of seen and unseen scenarios

High-fidelity synthetic data were generated using the β-VAE (see Section 2.1) and PC-AAE (see Section 2.2). These techniques permit to augment experimental datasets, which can be costly and time-consuming to acquire.

Figure 6 shows the KDE of u, v, w and ν_t comparing ground truth and synthetic data generated by the aforementioned different methods. This is the mean for each node across all samples. For u, PC-AAE (uvwnut) and β-VAE show the best agreement with the ground truth. For v, all experiments exhibit an excellent match with the observed velocity. This can be explained by the fact that v is almost negligible and tends to 0, as there are minimal wave movement in that direction. For w, the movement in that axis is more restricted than in the X-axis which yields

Table 1
NMI between the ground truth values of the test dataset and the predicted values by the generative methods

Experiments	NRMSE				NMI			
	u	v	w	ν_t	u	v	w	ν_t
Unut_sep	0.029	0.017	0.023	0.015	0.942	0.968	0.942	0.945
uvwnut	0.040	0.022	0.028	0.038	0.942	0.968	0.942	0.945
Unut	0.030	0.018	0.021	0.022	0.942	0.968	0.942	0.945
β-VAE	0.022	0.028	0.038	0.170	0.942	0.954	0.942	0.945

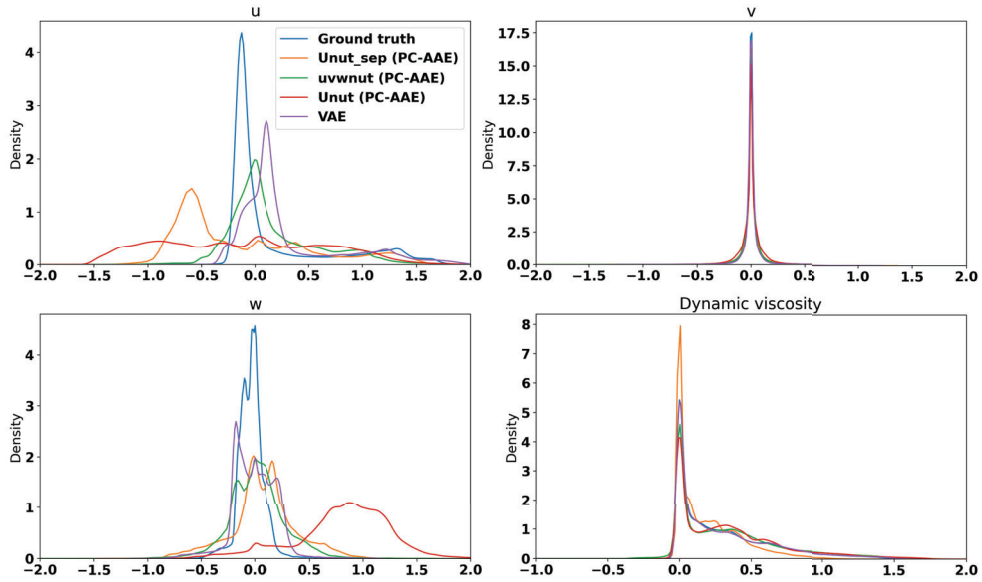


Figure 6: KDE for ground truth (blue), and 100 synthetic data, generated by different generative methods, of the average values for each node over all samples.

smaller velocities, and here is where the different experiments show greater variations with the ground truth in general. Finally, for ν_t, all experiments show an excellent agreement with the ground truth distribution.

In general, the experiments that show the highest fidelity to the ground truth data come from β-VAE and PC-AAE (uvwnut). This can be explained because each field was standardised by its mean and standard deviation. This is not the case for the other experiments where the occasional higher values shown in the velocity fields could weight the PC space towards them.

In particular, the generated data with PC-AAE (Unut) shows the biggest distribution spread which expands the physical scenarios and explores more extreme velocities, due to a covariate shift. This can help understand how the point absorber WEC reacts to larger unseen incoming waves. This characteristic is fundamental for predicting unseen scenarios (Yang, Zhou, Li, and Liu, 2021) and we will expand the analysis of this aspect in later.

Whilst generative networks aim to maximise the probability of generating real data whilst maintaining the distance between the real and estimated posterior distributions small, this might not represent any physical meaning. This is extremely important in test cases like WEC. In Figure 7, the averaged values of velocities u, v and w are plotted against the ν_t to showcase its relationship. It is portrayed that this relationship is preserved in all experiments where the averaged, over the number of data points, samples clusters overlap. Figure 7 also shows the spread over time-level samples from the different experiments. The larger spread is given by PC-AAE (Unut) and the smaller spread of samples is yielded by β-VAE. The latter can be explained due to the large size of the network required to train the β-VAE which has an input and output of 3.4M points making it difficult to train due to memory allocation, rather than the PC-AAE experiments which use tens or hundreds of points for input. However, the PC-AAE-related experiments need to store

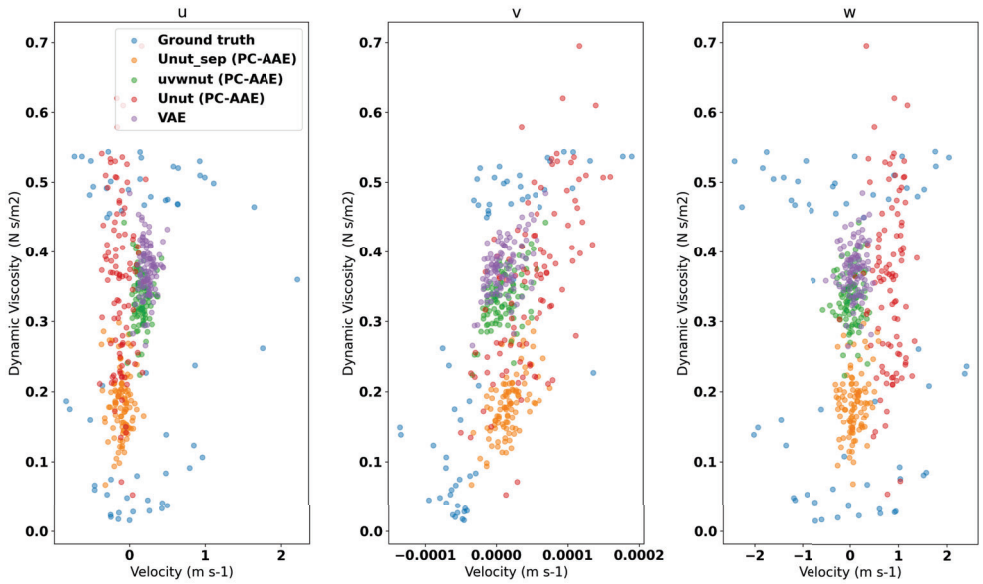


Figure 7: Scatter values of velocities u, v, and w against ν_t of the average values per time samples over all nodes.

the inverse mapping to the physical space which is of a similar storage size to the original simulation used for training data.

To understand how the distributions of the newly generated samples are related to the distribution of the ground truth data, we obtained a t-Student Stochastic Neighbour Embedding (t-SNE) projection (Van der Maaten and Hinton, 2008). The t-SNE projection is depicted in Figure 8. For velocities, it is clear that the ground truth (U_{GT}) clusters together with the synthetic samples for all experiments. Similar behaviour can be observed for ν_t^{GT} (nut_GT) and its synthetic samples. Moreover, the relationship between the velocity fields and the dynamic viscosity observed in the ground truth samples is observed and preserved in the synthetic data. This is another example of how the physical relationship between U and ν_t behaves and how it is preserved in the different generative methods.

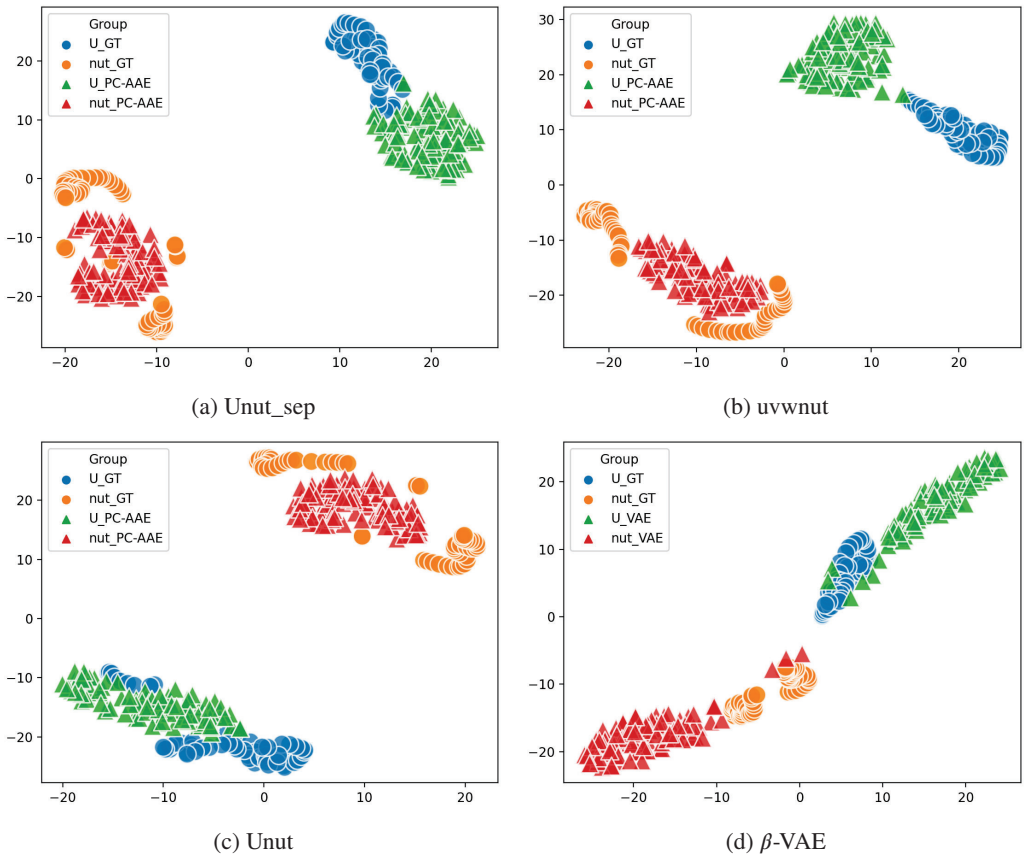
The synthetic data are generated by sampling random Gaussian noise, in these cases with a sample size of 16, before feeding this into the decoder $\mathcal{P}$.

As aforementioned, PC-AAE (Unut) is more able to expand the design space, whilst maintaining the physical relationship between U and ν_t. Therefore, for analysing unseen scenarios, only PC-AAE (Unut) will be discussed.

To further demonstrate the synthetic data's physical meaning, Figure 9 shows the relationship of real and synthetic u, v, and w (X-axis) against ν_t (Y-axis). The scatter plots show $m = 851101$ points and how the synthetic data overlaps with the ground truth data. The PC-AAE overlaps and engulfs the spread shown by the ground truth data. This demonstrates how the design space is expanded by PC-AAE. As aforementioned, the generated data with PC-AAE (Unut) shows the biggest distribution spread which expands the physical scenarios and explores more extreme velocities. The access to this part of the design space comes from how the PC was built. In PC-AAE (Unut) all fields were considered together giving u and w a larger weight than v and ν_t. Thus, PC-AAE can expand the design space with larger incoming waves whilst preserving the physical relationship among variables.

Figures 10a and 10b show 10 newly generated samples of the U and ν_t in the XZ planes, respectively, using PC-AAE. For these Figures, a new set of 100 samples were generated, however, only 10 samples are shown for display purposes.

As shown, for high-fidelity scenarios it is better to separate all variables and preprocess them individually, i.e standardisation, to preserve the statistics without assigning a larger weight to specific fields. However, for expanding the design space, a larger weight can be given to the fields with larger values when constructing the PC space on all fields, specifically, to the fields mostly affected by the variations of irregular incoming waves.



(a) Unut_sep (b) uvwnut

(c) Unut (d) β-VAE

Figure 8: t-SNE projection of the ground truth data (GT) and 100 synthetic data for the Unut_sep, uvwnut, Unut, and β-VAE experiments each experiment using the PC-AAE and β-VAE.

This has tremendous implications for expanding the design space and exploring unseen scenarios of the WEC CFD simulation. The generative model has the potential to capture the transition and the new physical phenomenons under new extreme initial conditions, as shown in the synthetic data generated by PC-AAE (Unut).

4.3. Model architectures

The architectures of both generative models are shown in Table 2 where the networks have been trained with batch normalisation before and a dropout of 0.5 between layers. For the PC-AAE-related architectures, the discriminator updates 10 times before the generator during training, and Nadam parameters with Learning rate $lr = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999$

4.4. Performance

The most notable capability of these approaches is the speed of execution and how fast new synthetic samples can be obtained. The runtimes, averaged over 10 times, to generate 100 synthetic samples with the different methods are 10.46 s, 5.23 s, 10.81 s, and 0.64s for PC-AAE: Unut_sep, uvwnut, Unut, and β-VAE, respectively. For the PC-AAE experiments, it only takes $\sim$ 0.05 s to generate 100 samples in the PC-space before projecting these onto the physical space. These times were obtained using a 2.3 GHz 8-Core Intel Core i9 processor. The speed of these generative models compares to $\sim$ 1 week of simulation using `OpenFOAM` for the original 3D CFD simulation, which only outputs 60 time-levels. This is a speed-up from 5 to 6 orders of magnitude.

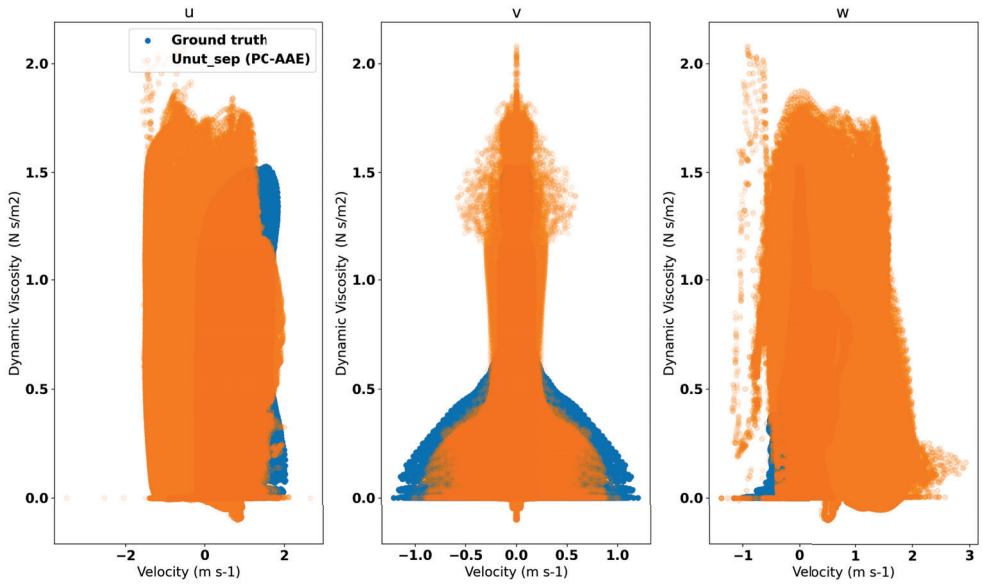
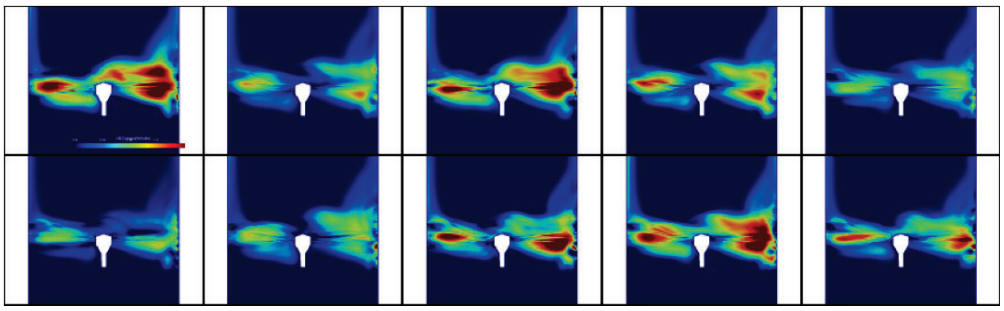
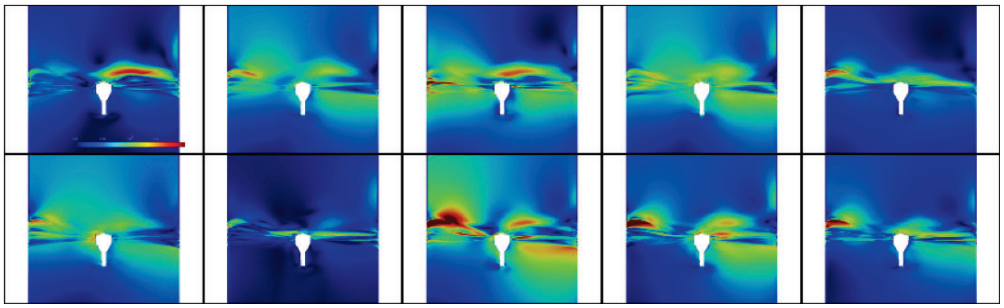


Figure 9: Scatter values of velocities u, v, and w against ν_t of the average values per point for PC-AAE (Unut)



(a) 10 newly generated samples of ν_t using the PC-AAE. Shown in the XZ plane.



(b) 10 newly generated samples of U using the PC-AAE. Shown in the XZ plane.

Figure 10: Example of newly generated samples of U and ν_t using the PC-AAE. Shown in the XZ plane.

5. Conclusion and Future work

We have shown how two different ML-based generative models, β-VAE and PC-AAE, can generate synthetic data of WECs quickly and at reduced computational cost. These models can generate several samples in a fast manner, compared to 1 week time of simulation using `OpenFOAM`. The t-SNE projection of the ground truth and generated

Table 2
Architectures of the different generative networks β-VAE and PC-AAE. The encoder $\mathcal{Q}$, the decoder $\mathcal{P}$ and the discriminator $\mathcal{D}^{\mathcal{A}}$ are fully-connected layers.

Experiments AE	Enc $\mathcal{Q}$	Dec $\mathcal{P}$	Disc $\mathcal{D}^{\mathcal{A}}$
PC-AAE (Unut_sep)	96	16	16
	32	32	
	16	96	
PC-AAE (uvwnut)	192	16	16
	32	32	
	16	192	
PC-AAE (Unut)	48	16	16
	32	32	
	16	48	
	Enc $\mathcal{Q}$	Dec $\mathcal{P}$	
β-VAE	3404404	16	
	64	32	
	32	64	
	16	3404404	

samples are located closely spatially, preserving the relationship between the velocity and dynamic viscosity fields. Furthermore, this is also shown by the relationship of u, v, and w with ν.

The reconstruction of the ground truth using these generative models agrees well with the test dataset. As shown, for high-fidelity scenarios it is better to separate all variables and preprocess them individually to preserve the statistics without assigning a larger weight to specific fields. However, for expanding the design space, a larger weight can be given to the fields with larger values when constructing the PC space on all fields.

This has tremendous implications for expanding the design space and exploring unseen scenarios of the CFD simulation. The generative model has the potential to capture the transition and the new physical phenomenons under extreme initial conditions, as shown in the synthetic data generated by PC-AAE.

Moreover, these generative models can generate snapshots of ∼3.4 M features at a fraction of the cost of the original simulation. Future work will include using these generated samples to improve the prediction of ν_t in future time steps, by augmenting the design space with physically plausible samples. Additionally, other newer generative models like latent diffusion models, due to the number of nodes in these simulations, could be used.

6. Data and Code availability

The code used in this paper is available in `http://github.com/c-quilo/WaveSuite` and the data is available on request.

7. Acknowledgements

The authors would like to acknowledge support from the UK Engineering and Physical Sciences Research Council (EPSRC) Programme Grant PREMIERE (EP/T000414/1), the EPSRC grant EP/T003189/1 Health assessment across biological length scales for personal pollution exposure and its mitigation (INHALE), and the EPSRC grant EP/V040235/1 New Generation Modelling Suite for the Survivability of Wave Energy Convertors in Marine Environments (WavE-Suite).

CRediT authorship contribution statement

César Quilodrán-Casas: Conceptualisation of this study, Methodology, Software, Formal analysis, Investigation, Writing - Original draft preparation, Writing - Review & Editing. **Qian Li:** Data Curation, Writing - Original Draft, Writing - Review & Editing. **Ningbo Zhang:** Writing - Review & Editing. **Sibo Cheng:** Writing - Review & Editing. **Shiqiang Yan:** Supervision, Funding acquisition, Writing - Review & Editing. **Qingwei Ma:** Supervision, Funding acquisition, Writing - Review & Editing. **Rossella Arcucci:** Supervision, Funding acquisition, Writing - Review & Editing.

References

I. Glendenning, Ocean wave power, Applied Energy 3 (1977) 197–222.

F. d. O. Antonio, Wave energy utilization: A review of the technologies, Renewable and sustainable energy reviews 14 (2010) 899–918.

S. Jin, R. J. Patton, B. Guo, Viscosity effect on a point absorber wave energy converter hydrodynamics validated by simulation and experiment, Renewable energy 129 (2018) 500–512.

D. Son, R. W. Yeung, Optimizing ocean-wave energy extraction of a dual coaxial-cylinder wec using nonlinear model predictive control, Applied energy 187 (2017) 746–757.

Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.

C. Quilodrán-Casas, R. Arcucci, L. Mottet, Y. Guo, C. Pain, Adversarial autoencoders and adversarial lstm for improved forecasts of urban air pollution simulations, arXiv preprint arXiv:2104.06297 (2021).

L. Chagot, C. Quilodrán-Casas, M. Kalli, N. M. Kovalchuk, M. J. Simmons, O. K. Matar, R. Arcucci, P. Angeli, Surfactant-laden droplet size prediction in a flow-focusing microchannel: a data-driven approach, Lab on a Chip 22 (2022) 3848–3859.

T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, arXiv preprint arXiv:1710.10196 (2017).

D. Berthelot, T. Schumm, L. Metz, Began: Boundary equilibrium generative adversarial networks, arXiv preprint arXiv:1703.10717 (2017).

A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434 (2015).

G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, E. Keogh, Generating synthetic time series to augment sparse datasets, in: 2017 IEEE international conference on data mining (ICDM), IEEE, 2017, pp. 865–870.

J. Hoffmann, Y. Bar-Sinai, L. M. Lee, J. Andrejevic, S. Mishra, S. M. Rubinstein, C. H. Rycroft, Machine learning in a data-limited regime: Augmenting experiments with synthetic data uncovers order in crumpled sheets, Science advances 5 (2019).

R. J. Chen, M. Y. Lu, T. Y. Chen, D. F. Williamson, F. Mahmood, Synthetic data in machine learning for medicine and healthcare, Nature Biomedical Engineering (2021) 1–5.

J. Yoon, J. Jordon, M. Schaar, Radialgan: Leveraging multiple datasets to improve target-specific predictive models using generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2018, pp. 5699–5707.

L. Zhou, S. Pan, J. Wang, A. V. Vasilakos, Machine learning on big data: Opportunities and challenges, Neurocomputing 237 (2017) 350–361.

H. Li, C. Zhang, M. Chen, D. Shen, Y. Niu, Data-driven surrogate modeling: Introducing spatial lag to consider spatial autocorrelation of flooding within urban drainage systems, Environmental Modelling & Software (2023) 105623.

M. Sadeghi, P. Nguyen, K. Hsu, S. Sorooshian, Improving near real-time precipitation estimation using a u-net convolutional neural network and geographical information, Environmental Modelling & Software 134 (2020) 104856.

S. Razavi, Deep learning, explained: Fundamentals, explainability, and bridgeability to process-based modelling, Environmental Modelling & Software 144 (2021) 105159.

I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

C. Rodriguez-Delgado, R. J. Bergillos, G. Iglesias, An artificial neural network model of coastal erosion mitigation through wave farms, Environmental Modelling & Software 119 (2019) 390–399.

L. Li, Z. Yuan, Y. Gao, Maximization of energy absorption for a wave energy converter using the deep machine learning, Energy 165 (2018) 340–349.

D. Sarkar, E. Contal, N. Vayatis, F. Dias, Prediction and optimization of wave energy converter arrays using a machine learning approach, Renewable Energy 97 (2016) 504–517.

P. D. Sclavounos, Y. Ma, Wave energy conversion using machine learning forecasts and model predictive control, in: 33st International Workshop on Water Waves and Floating Bodies, Brest, France, 2018.

S. M. Mousavi, M. Ghasemi, M. Dehghan Manshadi, A. Mosavi, Deep learning for wave energy converter modeling using long short-term memory, Mathematics 9 (2021) 871.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-vae: Learning basic visual concepts with a constrained variational framework (2016).

A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, arXiv preprint arXiv:1511.05644 (2015).

D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

W. Falcon, et al., Pytorch lightning, GitHub. Note: https://github. com/PyTorchLightning/pytorch-lightning 3 (2019).

J. Lever, M. Krzywinski, N. Altman, Points of significance: Principal component analysis, 2017.

S. Cheng, Y. Jin, S. P. Harrison, C. Quilodrán-Casas, I. C. Prentice, Y.-K. Guo, R. Arcucci, Parameter flexible wildfire prediction using machine learning techniques: Forward and inverse modelling, Remote Sensing 14 (2022) 3228.

S. Cheng, J. Chen, C. Anastasiou, P. Angeli, O. K. Matar, Y.-K. Guo, C. C. Pain, R. Arcucci, Generalised latent assimilation in heterogeneous reduced spaces with machine learning surrogate models, Journal of Scientific Computing 94 (2023) 11.

H. Gong, S. Cheng, Z. Chen, Q. Li, C. Quilodrán-Casas, D. Xiao, R. Arcucci, An efficient digital twin based on machine learning svd autoencoder and generalised latent assimilation for nuclear reactor physics, Annals of Nuclear Energy 179 (2022) 109431.

T. Dozat, Incorporating Nesterov momentum into adam (2016).

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).

F. Chollet, et al., Keras, 2015. URL: https://github.com/fchollet/keras.

H. Jasak, A. Jemcov, Z. Tukovic, et al., Openfoam: A c++ library for complex physics simulations, in: International workshop on coupled methods in numerical dynamics, volume 1000, IUC Dubrovnik Croatia, 2007, pp. 1–20.

Q. Ma, S. Yan, Quasi ale finite element method for nonlinear water waves, Journal of computational physics 212 (2006) 52–72.
Q. Ma, S. Yan, Qale-fem for numerical modelling of non-linear interaction between 3d moored floating bodies and steep waves, International Journal for Numerical Methods in Engineering 78 (2009) 713–756.
S. Yan, Q. Ma, Numerical simulation of fully nonlinear interaction between steep waves and 2d floating bodies using the qale-fem method, Journal of Computational physics 221 (2007) 666–692.
S. Yan, Q. Ma, Qale-fem for modelling 3d overturning waves, International Journal for Numerical Methods in Fluids 63 (2010) 743–768.
S. Yan, Q. Ma, X. Cheng, Fully nonlinear hydrodynamic interaction between two 3d floating structures in close proximity, International Journal of Offshore and Polar Engineering 21 (2011).
Q. Li, J. Wang, S. Yan, J. Gong, Q. Ma, A zonal hybrid approach coupling fnpt with openfoam for modelling wave-structure interactions with action of current, Ocean Systems Engineering 8 (2018) 381–407.
S. Yan, J. Wang, J. Wang, Q. Ma, Z. Xie, Ccp-wsi blind test using qalefoam with an improved passive wave absorber, International Journal of Offshore and Polar Engineering 30 (2020) 43–52.
S. Yan, Q. Li, J. Wang, Q. Ma, Z. Xie, T. Stoesser, Comparative numerical study on focusing wave interaction with fpso-like structure, International Journal of Offshore and Polar Engineering 29 (2019) 149–157.
S. Utyuzhnikov, Generalized wall functions and their application for simulation of turbulent flows, International journal for numerical methods in fluids 47 (2005) 1323–1328.
J. H. Todalshaug, G. S. Ásgeirsson, E. Hjálmarsson, J. Maillet, P. Möller, P. Pires, M. Guérinel, M. Lopes, Tank testing of an inherently phase-controlled wave energy converter, International Journal of Marine Energy 15 (2016) 68–84.
K. Hasselmann, T. P. Barnett, E. Bouws, H. Carlson, D. E. Cartwright, K. Enke, J. Ewing, A. Gienapp, D. Hasselmann, P. Kruseman, et al., Measurements of wind-wave growth and swell decay during the joint north sea wave project (jonswap)., Ergaenzungsheft zur Deutschen Hydrographischen Zeitschrift, Reihe A (1973).
A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of machine learning research 3 (2002) 583–617.
J. Yang, K. Zhou, Y. Li, Z. Liu, Generalized out-of-distribution detection: A survey, arXiv preprint arXiv:2110.11334 (2021).
L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (2008).

Acronyms

2D Two-dimensional

3D Three-dimensional

AE Autoencoder

CFD Computational Fluid Dynamics

VAE Variational Autoencoder

PC Principal Components

PCA Principal Component Analysis

WEC Wave Energy Converter

AAE Adversarial Autoencoder

t-SNE t-Student Stochastic Neighbour Embedding

NS Navier-Stokes

RANS Reynolds-averaged Navier Stokes

KL Kullback-Leibler Divergence Score

KDE Kernel density estimator

MSE Mean Squared Error

PC-AAE Principal Components-based Adversarial Autoencoder

MI Mutual Information

NMI Normalised Mutual Information

LR Low-Reynolds number

HR High-Reynolds number

EOF Empirical Orthogonal Functions

QALE-FEM Quasi Lagrangian-Eulerian Finite Element Method

NRMSE Normalised Root Mean Squared Error

SST Shear Stress Transport

DA Data Augmentation

DL Deep Learning

ML Machine Learning

Conflict of Interest

Declaration of interests

☐The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Rossella Arcucci reports financial support was provided by Engineering and Physical Sciences Research Council. Qingwei Ma reports financial support was provided by Engineering and Physical Sciences Research Council.