Toward the Rapid Design of Engineered Systems Through Deep Neural Networks

Christopher McComb

The design of a system commits a significant portion of the final cost of that system. Many computational approaches have been developed to assist designers in the analysis (e.g., computational fluid dynamics) and synthesis (e.g., topology optimization) of engineered systems. However, many of these approaches are computationally intensive, taking significant time to complete an analysis and even longer to iteratively synthesize a solution. The current work proposes a methodology for rapidly evaluating and synthesizing engineered systems through the use of deep neural networks. The proposed methodology is applied to the analysis and synthesis of offshore structures such as oil platforms. These structures are constructed in a marine environment and are typically designed to achieve specific dynamics in response to a known spectrum of ocean waves. Results show that deep learning can be used to accurately and rapidly synthesize and analyze offshore structure.

Introduction

A significant amount of the final cost of a system is committed during design. According to the situated function–behavior–structure design framework, design consists of navigating from the requirements for a solution to the documentation of that solution [1, 2]. This process entails negotiating through several ontological categories, including function, expected behavior, derived behavior, and structure. The focus of this paper is on the tasks of analysis (deriving behavior from structure) and synthesis (generating a structure based on desired behavior). Many computational approaches have been developed to assist designers in the analysis and synthesis of engineered systems (e.g., computational fluid dynamics and topology optimization, respectively).

C. McComb (✉)
The Pennsylvania State University, University Park, PA, USA
e-mail: uum209@psu.edu

J. S. Gero (ed.), *Design Computing and Cognition '18*,
https://doi.org/10.1007/978-3-030-05363-5_1

However, these approaches are often computationally intensive, taking significant time to complete an analysis and even longer to iteratively synthesize a solution. The current work proposes a methodology for rapidly evaluating and synthesizing engineered systems through the use of deep neural networks.

The proposed methodology is applied to the analysis and synthesis of offshore structures. Examples of offshore structures include buoys, oil rigs, and cruise ships. The analysis of an offshore structure design often involves a simulation that combines multibody dynamics with computational fluid dynamics. This makes the analysis of solutions computationally intensive, precluding the use of design algorithms which are often stochastic in nature and require thousands of iterations [3–5]. The objective of the proposed work is to alleviate that problem by introducing a methodology for achieving two goals:

1. the rapid performance analysis of an engineered system, and
2. the rapid synthesis of an engineered system given desired performance characteristics.

The proposed approach makes use of deep neural networks to accomplish these objectives. Specifically, variational autoencoders are used to perform to reduce the dimensionality of the input and output data, making it possible to learn analysis and synthesis in a space of reduced complexity. The remainder of the paper is organized as follows. A background section reviews related work in machine learning and the design of offshore structure. The next section lays out the generalizable methodology for achieving both rapid analysis and synthesis of engineered systems. Results of applying this methodology to the design of offshore structure are presented and discussed. This paper concludes with a discussion of future directions for this work, highlighting the possible role of the engineering design community as a driving force in generative machine learning research.

Background

Neural Networks and Deep Learning

Artificial neural networks (referred to in the remainder of this paper simply as neural networks) are computational systems that are analogous to the biological neural networks that make up nervous tissue and animal brains. Neural networks can be trained to accomplish a variety of complex tasks, including regression, classification, and feature extraction. Jain et al. provide a more detailed introduction to neural networks [6]. Deep learning, which is the focus in this work, refers specifically to neural networks that have more than one hidden layer.

Neural networks have shown significant success in two-dimensional image recognition tasks. This success has led researchers to apply similar methodology to three-dimensional recognition tasks [7], facilitated by recent advances in computing that

enable such tasks to be performed at scale. Seminal dataset and classification efforts include ObjectNet3D [8], ShapeNet [9], VoxNet [10], and PointNet [11]. The automated synthesis of three-dimensional objects is still a nascent field in machine learning. Most approaches focus on creating objects with a given form and category (e.g., [12, 13]), rather than attempting to derive a deeper relationship between desired functionality and requisite form.

This work also makes use of autoencoders. These are specially designed neural networks that take an input, map it into a space with reduced dimensionality, and then output a reconstructed version of the input [14]. The two halves of the neural network (the encoder and the decoder, respectively) can then be used for specific and useful functions. The encoder can map an input into a reduced space, essentially performing data compression, while the decoder can take compressed values and reconstruct an output. This work uses *variational* autoencoders which map the input into a space of latent variables so that the training data are normally distributed [15]. This is accomplished by training the network with a loss function that measures reconstruction accuracy as well as how normally distributed the parameters in the maximally compressed layer are (typically Kullback–Leibler divergence). This ensures that the variables in the latent space are rich in information. Variational autoencoders have been used to compress a wide variety of different data, including human faces [16], handwritten numbers [17], and house numbers [18].

Neural networks of many varieties have been utilized in design and engineering to accomplish various tasks. For instance, Tseng, Cagan, and Kotovsky utilized a neural network to learn the preferences of a customer and then utilized that neural network as the objective function for a genetic algorithm [19]. Dering and Tucker utilized convolutional neural networks to predict the function of a product from its form alone [20]. The utilization of deep learning, and specifically autoencoders, also led to the creation of a computational framework that models the curiosity of a given user in order to provide surprising examples [21]. Neural networks have also been utilized to automatically predict quality defects in automotive parts [22] and to support design for additive manufacturing [23–25]. These examples, while not exhaustive, serve to highlight potential utility of neural networks for design and the need for a standardized approach to implementing them. The current research utilizes a generic, voxel-based approach for describing potential design solutions, thus ensuring significant representation flexibility.

Offshore Structures

Offshore structures are comprised of buoys, drilling platforms, and wave energy converters (WECs). WECs are an increasingly common type of offshore structure that are designed to extract energy from ocean waves. WECs may serve an important role in the future of humankind, since it is estimated that approximately 3.7 TW (3.7 trillion Watts) of power can be harvested from the world's oceans [26]. However,

in order to access that power, several challenges in the design of WECs must be overcome [27].

A substantial array of numerical methods have been developed for the simulation of offshore structures, including analytical methods, empirical methods, Navier–Stokes equation methods, and boundary-integral equation methods [28]. Analytical methods offer quick and rough estimates for devices with simple geometry, while most empirical methods attempt to maintain simplicity while making use of experimental values to increase accuracy. Navier–Stokes equation methods (NSEMs) can resolve highly nonlinear phenomena, but generally do not permit closed-form solutions, requiring the use of computational fluid dynamics.

Boundary-integral equation methods (BIEMs) are the focus of this work, as they are the industry standard for design and analysis of offshore structures. BIEMs produce a potential flow solution in the frequency domain [28]. This means that the outputs are given as spectra that indicate how much force, damping, or other quantities are applied to an offshore structure for incoming ocean waves with varying frequencies. Although they are far less computationally expensive than NSEMs, producing a full BIEM solution for a model with a high mesh resolution can still take hours. In addition, pure frequency-domain BIEMs are only weakly nonlinear [28] which makes it impossible to directly implement nonlinear control strategies within the simulation. One way to overcome this limitation is by numerically integrating over several frequencies of the BIEM solution to yield a time-varying series for different fluid phenomena [29]. These series can then be applied in an appropriate 6 degree-of-freedom (6DOF) solver to produce a time-domain simulation, from which important metrics such as average power production can be computed.

The application of the current work focuses on predicting frequency-varying wave force spectra as a function of structure geometry (and vice versa). It should be noted that a similar methodology could be applied directly to other frequency-varying fluid phenomena that are produced by a BIEM solution. The results of the current work could be integrated into software packages that utilize the BIEM + 6DOF approach outlined above [30, 31]. This would enable rapid exploration of conceptual solutions, either by human designers and engineers or by agent-based design algorithms [4, 5, 32].

Methodology

The approach that is proposed in this work for rapidly evaluating and synthesizing engineered systems can be broken into three steps. This process is depicted graphically in Fig. 1. First, a diverse set of examples of a given system type must be generated, and the performance of each example must be analyzed using current methods (finite element analysis, computational fluid dynamics, experimental testing, etc.). The second step entails training two autoencoders: one for the engineered system and one for the performance assessment. The third and final step in the proposed methodology is the recombination of performance and system autoencoders

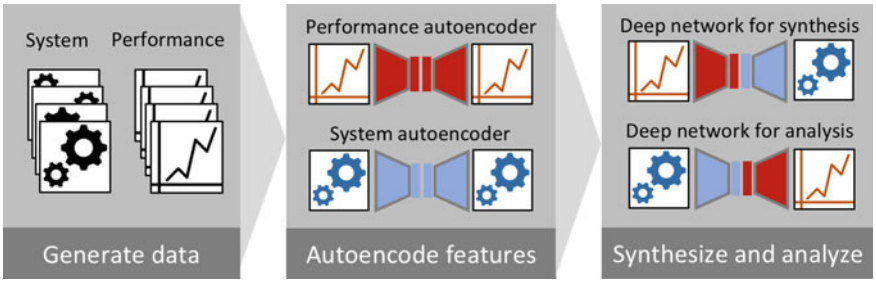

Fig. 1 Primary steps in the proposed methodology

into two new, deep networks that are possible of accomplishing rapid analysis of a system (encoding the system, decoding performance) as well as synthesis of a system according to desired performance (encoding performance, and decoding system). Both of these new networks should have one or more new layers that must be trained between the encoder and the decoder, enabling a mapping between the latent system space and the latent performance space (or vice versa). The use of autoencoders is critical as it permits the learning of synthesis and analysis in the latent space which has fewer dimensions (and thus less complexity) than the input or output.

The current work shares the application of the above methodology to analysis and synthesis of offshore structures. First, NEMOH, a BIEM solver, was used to simulate thousands of different floating body geometries, deriving frequency-varying response forces for each [35]. Next, the data generated in NEMOH was used to train two variational autoencoders, one of which modeled key features of frequency-varying response forces and the other modeled key geometric features of the input geometries. Finally, these autoencoders were used to instantiate two networks: one for predicting the force spectra of known geometries (analysis), and the other for generating geometries for a known force spectrum (synthesis). All neural networks were trained using the Keras neural network API [33] in conjunction with the Theano library [34]. A full implementation of this work, including training data, is available in the Python language under an MIT License.[1]

Data Generation

The dataset used in this work was generated by instantiating 5000 different common shapes for offshore structures. These included wedges, hemispheres, cylinders, rectangular prisms, and cones. All shapes were generated to fit within a 10 m × 10 m × 10 m bounding box. These offshore structure shapes were then analyzed using the NEMOH BIEM solver [35], producing frequency-varying spectra describing the

[1] https://github.com/HSDL/WAnet/releases/tag/v1.0-beta.

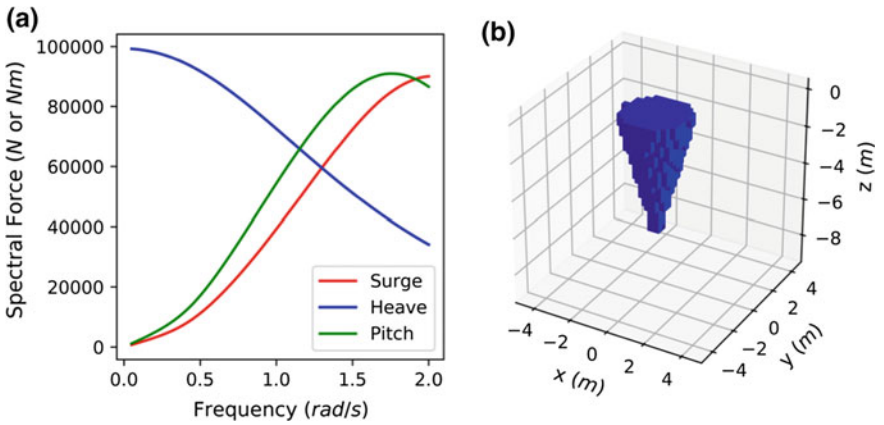

Fig. 2 Example of **a** force response spectra and **b** voxelized geometry

forces applied to the structure both parallel and perpendicular to the direction of the incoming ocean waves (commonly referred to as heave and surge, respectively), as well as a moment about the center of gravity of the body (referred to as pitch). An example of these spectra is provided in Fig. 2a.

The geometry for the offshore structures was originally provided to NEMOH as a mesh. The meshes were converted into a voxel-based format in order to make the geometry data more accessible to the proposed neural network approach. Voxels are a three-dimensional analogue of pixels. Specifically, the bounding box for the offshore structure was discretized into a $32 \times 32 \times 32$ grid, containing 32,768 voxels. The voxel values in this grid were defined as 1 (if the structure occupied part of the voxel) of 0 (if the structure did not occupy part of the voxel).

Thus, the final dataset consists of paired geometry-spectra observations. An example of a paired observation is provided in Fig. 2. Plots of force response spectra and voxelized geometries in the remainder of the paper will omit axis labels and scales in the interest of clarity and concision. This dataset was randomly separated into a training set (80% of the data, 4000 observations) and a testing set (20% of the data, 1000 observations). All accuracies reported in the remainder of this paper correspond to measurements on the testing set.

Training Variational Autoencoders

Two variational autoencoders were trained based on the data generated with NEMOH. The structure of these autoencoders is shown in Figs. 3 and 4. Both variational autoencoders are designed to compress the input data into an N-dimensional latent space, which describes the number of nodes in the smallest layer. The value of N is identified through a parametric search, detailed in the results section of this paper.

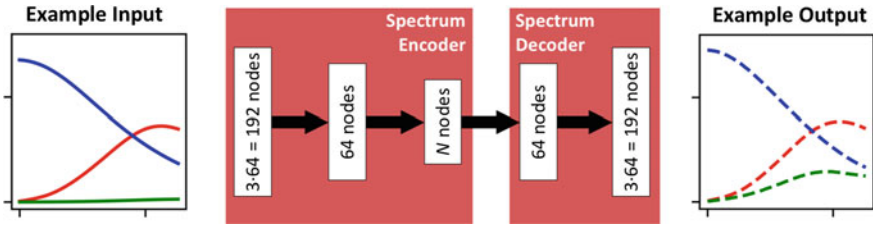

Fig. 3 Architecture for the force spectrum autoencoder

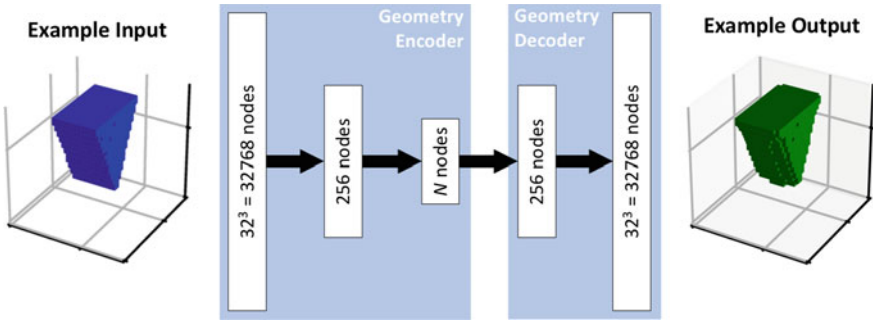

Fig. 4 Architecture for the voxel geometry autoencoder

Both autoencoders were trained using the root mean square propagation (RMSprop) algorithm [36]. The primary term in the loss function of the force spectra autoencoder was based on mean squared error while the primary term for the geometry autoencoder was based on binary cross-entropy. Both training algorithms also included a term for Kullback–Leibler divergence [37] of the values in the latent space (the innermost hidden layer) in the loss function. The computation of the loss functions in this way is standard for variational autoencoders.

It should be noted that the dimensionality of the latent space for both the spectrum autoencoder and the geometry autoencoder is described by a single variable, N, despite the fact that the force spectrum is much simpler than the structure geometry. This is an intentional decision, as equating the dimensionality of the spaces makes it more likely that a one-to-one mapping can be found between them.

Creating Neural Nets for Synthesis and Analysis

The variational autoencoders outlined in the previous section were recombined to instantiate two new deep neural networks, one for synthesizing geometries and the other for evaluating geometries. The structure for these neural networks is provided in Figs. 5 and 6.

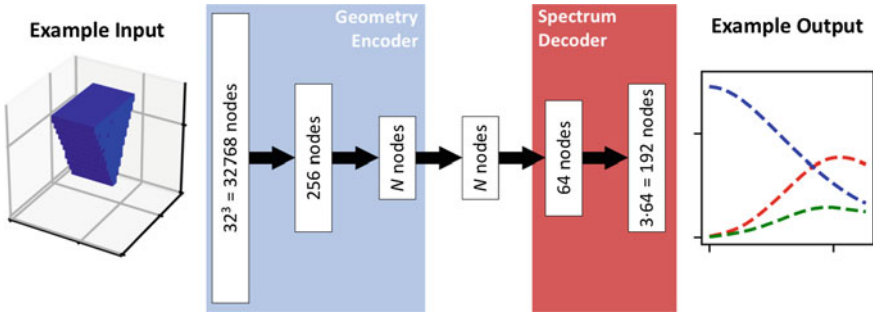

Fig. 5 Architecture for the analysis network

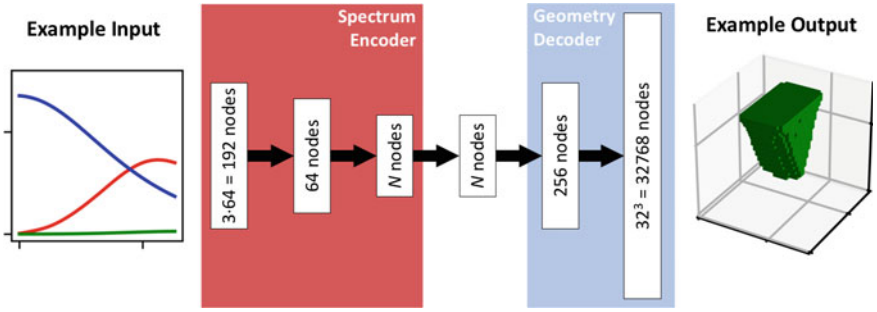

Fig. 6 Architecture for the synthesis network

The network designed for evaluating offshore structures utilizes the geometry encoder and the spectrum decoder (see Fig. 5). A layer of N nodes was included between these two elements, and this interior layer was the only layer that was trained. In essence, this network compresses the geometry of a structure into the N-dimensional geometry latent space (using the geometry encoder), maps the geometry latent space into the spectrum latent space (this is the trainable N-node layer), and then reconstructs the full force spectra, producing the desired output (using the spectra decoder).

The network designed for synthesis of offshore structure utilizes the spectrum encoder and the geometry decoder with a trainable N-node layer in between the two (see Fig. 6). This network compresses the spectra into the N-dimensional spectra latent space (using the spectra encoder), maps that into the N-dimensional geometry latent space (through trainable layer), and then reconstructs the full geometry (using the geometry decoder).

Both analysis and the synthesis networks were trained using the RMSprop algorithm [36]. The primary term in the loss function of the analysis network was based on mean squared error (since the output was a set of real-valued curves) while the primary term for the geometry autoencoder was based on binary cross-entropy (since the output was a set of voxel data with values between 0 and 1).

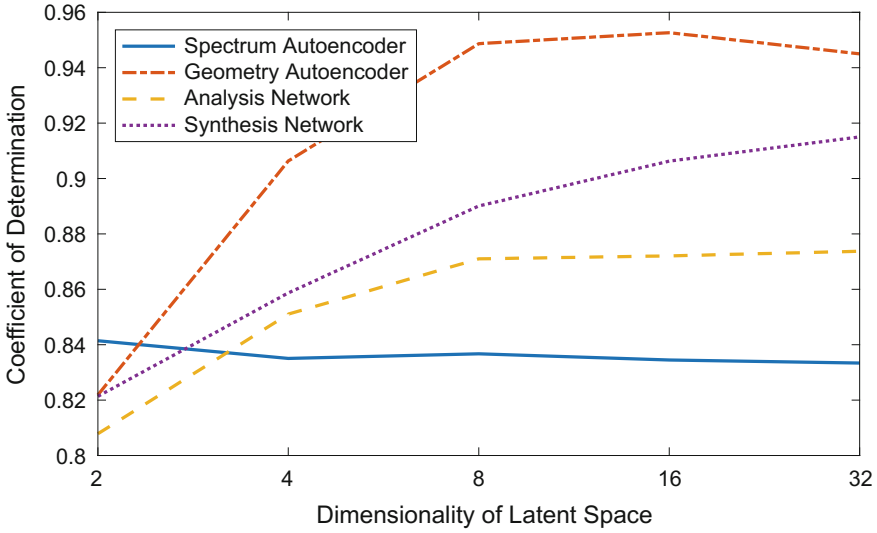

Fig. 7 Study to determine appropriate dimensionality of latent space

Results and Discussion

Results and discussion are provided in three subsections. The first subsection details a parametric study that was used to select the appropriate dimensionality for the latent space. The second reports the results and examples for the autoencoders (both geometry and spectrum). These autoencoders are critical as they permit the tasks of synthesis and analysis to be learned in a space of reduced complexity. The third subsection does the same for the recombined analysis and synthesis networks.

Determination of Latent Space Dimensionality

In order to determine the appropriate dimensionality for the latent space, a parametric study was conducted. All four networks used here (the spectrum autoencoder, the geometry autoencoder, the analysis autoencoder, and the synthesis autoencoder) were trained for increasing values of the dimensionality of the latent space, N. The results of this study are provided in Fig. 7. The horizontal axis shows dimensionality of the latent space and the vertical axis shows network validation accuracy (specifically, the percentage of variance explained by the trained network).

The data for the spectrum autoencoder is relatively flat, indicating that a small number of latent dimensions are sufficient to accurately reconstruct that data. The geometry autoencoder, in contrast, shows consistently increasing accuracy up to approximately 16 dimensions, at which point it begins to decrease. The analysis

and synthesis networks (which make use of portions of the autoencoders) continue to increase. However, training time increases substantially for larger latent spaces. Based on this study, a latent space dimensionality of 16 was selected as all networks are at or near maximum accuracy for this value.

Autoencoders

The force spectra autoencoder was trained for 100 epochs with a batch size of 100. The final mean squared error on the testing dataset was 9.90×10^9. The total variance of the training data was 5.89×10^{10} yielding a coefficient of determination of 0.83. This indicates that this autoencoder can account for approximately 83% of the variance observed in the training data. Several randomly selected examples of original and reconstructed spectra are provided in Fig. 8. Although the curves are not reconstructed exactly, in all cases the reconstructed curves tend to share many similarities with the original curves. These similarities include slope, range, and the location of maxima and minima. However, some distinct differences become apparent for spectra that have low values. For instances, in Fig. 8a, b, the green spectrum (corresponding to pitch) is nearly flat in the original. The reconstructed version, however, shows significantly higher values for that spectrum. A similar overestimation is observed for the blue spectrum in Fig. 8d.

The geometry autoencoder was trained for 40 epochs with a batch size of 100. The final binary cross-entropy on the testing dataset was 0.17. By comparing the final binary cross-entropy of the model to the binary cross-entropy of a mean model (where the value of every voxel is the average over all voxels in the dataset), it is pos-

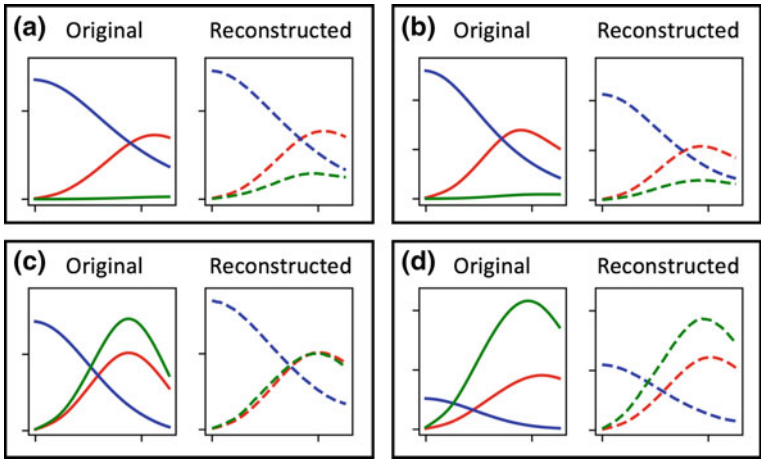

Fig. 8 Example results for force spectrum autoencoder

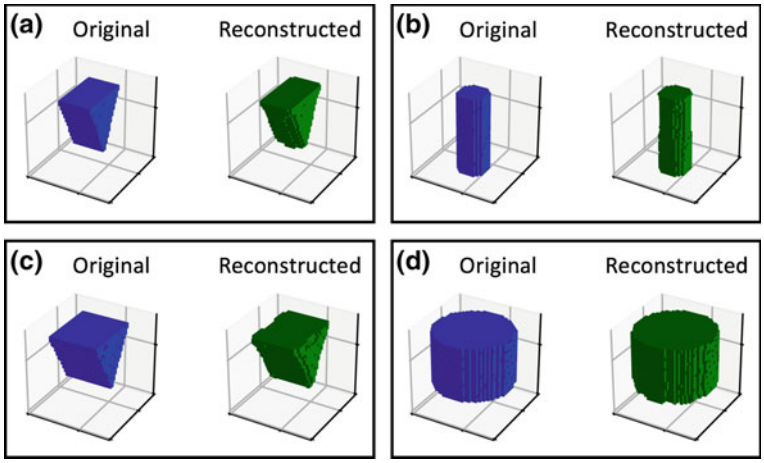

Fig. 9 Example results for offshore structure geometry autoencoder

sible to compute a coefficient of determination. The binary cross-entropy of the mean model is 3.42, yielding a coefficient of determination of 0.95. This value indicates that this autoencoder can reconstruct approximately 95% of the variance observed in the training data. Several randomly selected examples of original and reconstructed geometries are provided in Fig. 9. The original and reconstructed images are practically identical in many cases. The largest differences occur near sharp features, with the reconstructed showing a tendency to round corners and edges. In addition, flat faces in several of the geometries can be observed to bow outwards. It is possible that this could be corrected through the incorporation of convolutional layers in the autoencoder to better learn features that exist across size scales.

Synthesis and Analysis

This section reports the results of the neural networks designed to accomplish analysis and synthesis—these are the ultimate objects of the current work. These networks utilize portions of the autoencoders trained in the previous sections. Specifically, the network trained to perform analysis utilizes the encoder for voxel geometry and the decoder for force spectra. Conversely, the network trained for synthesis uses the encoder for force spectra and the decoder for voxel geometry.

The analysis network was trained for 25 epochs with a batch size of 100. The final mean squared error on the testing dataset was 7.49×10^9, yielding a coefficient of determination of 0.87. Figure 10 shows several examples for the analysis network. From left to right, each example includes the geometry provided to the network as an input, the true spectra (the set of spectra produced by the geometry in NEMOH), and the predicted spectra (the output from the network). The characteristics of the

predicted spectra are similar in some ways to the reconstructed spectra in Fig. 8. The analysis network correctly predicts qualitative aspects of the curves, accurately producing curves with slopes, maxima/minima, and ranges that are similar to the true spectra. However, like the autoencoder, the analysis network tends to overestimate low values. This is evidenced in Fig. 10b, c. In addition, the true spectrum in Fig. 10c shows a very specific cusp feature which does not appear in the predicted spectrum. It is likely that cusp features of this type were rare in the training data, and thus are filtered out by the spectra decoder.

The analysis network was trained for 25 epochs with a batch size of 100. The final binary cross-entropy on the testing dataset was 0.45, yielding a coefficient of determination of 0.90. Figure 11 shows several examples of the synthesis network. From left to right, each example includes the set of spectra that was used as input, the true geometry (the geometry originally used to produce the input spectra in NEMOH), and the predicted geometry (the output from the network). In some of these examples, the synthesized geometry shows distinct departures from the true geometry. Sharp corners tend to be rounded off and flat faces bow outward slightly. This is expected, since similar behavior was observed in the geometry autoencoder.

In addition, it appears that in Fig. 11c a cone-type geometry was synthesized for what should have been a wedge. Similarly, in Fig. 11d, a square geometry was synthesized in place of what should have been a cylinder. At this point, the reason behind such idiosyncrasies is unclear. One possibility is that the departure from expected performance is due to simple errors in the synthesis. On the other hand, the synthesis network may have created a different geometry that provides force spectra that are very similar to what was desired. This will be a subject of future work.

The analysis and synthesis networks potentially provide very real utility for designers of offshore structure. The force spectra that are produced with the analysis network are important for simulation of offshore structures. However, the production of force spectra using BIEM methodology can take minutes for a simple mesh, which precludes the direct use of the approach in many optimization algorithms which might require tens of thousands of iterations. The use of the analysis network as an approximate BIEM makes the direct use of optimization algorithms more feasible.

Regarding the synthesis network, the ocean waves at a given location can be characterized with a power-frequency spectrum similar to the force spectra computed for the structure. If the peaks on the wave and device spectra align, then the device will absorb significant energy from the waves; if the peaks do not align, energy absorption is mitigated. Thus, many designers can estimate a desirable force spectrum for the structure based on known characteristics of the installation location, and use the synthesis network to directly generate a suitable geometry.

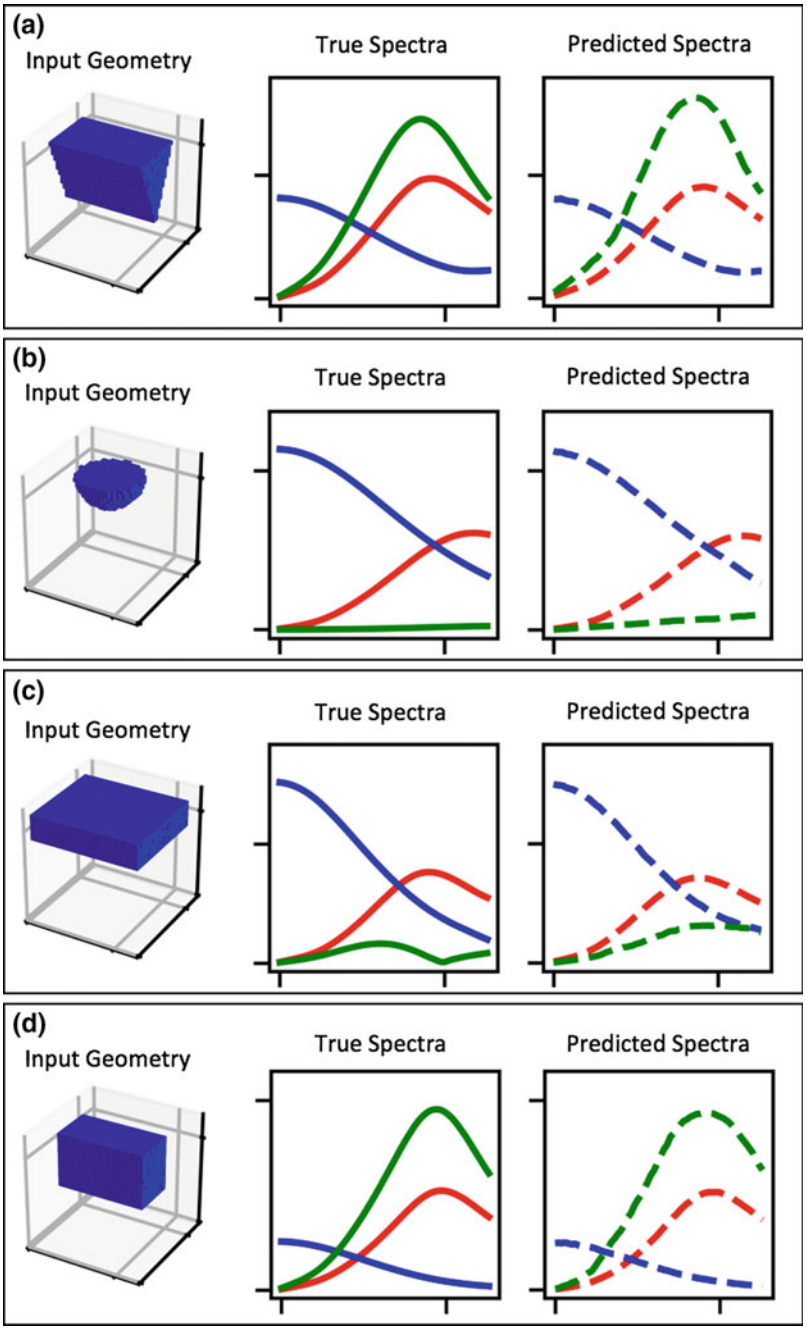

Fig. 10 Example results for analysis network

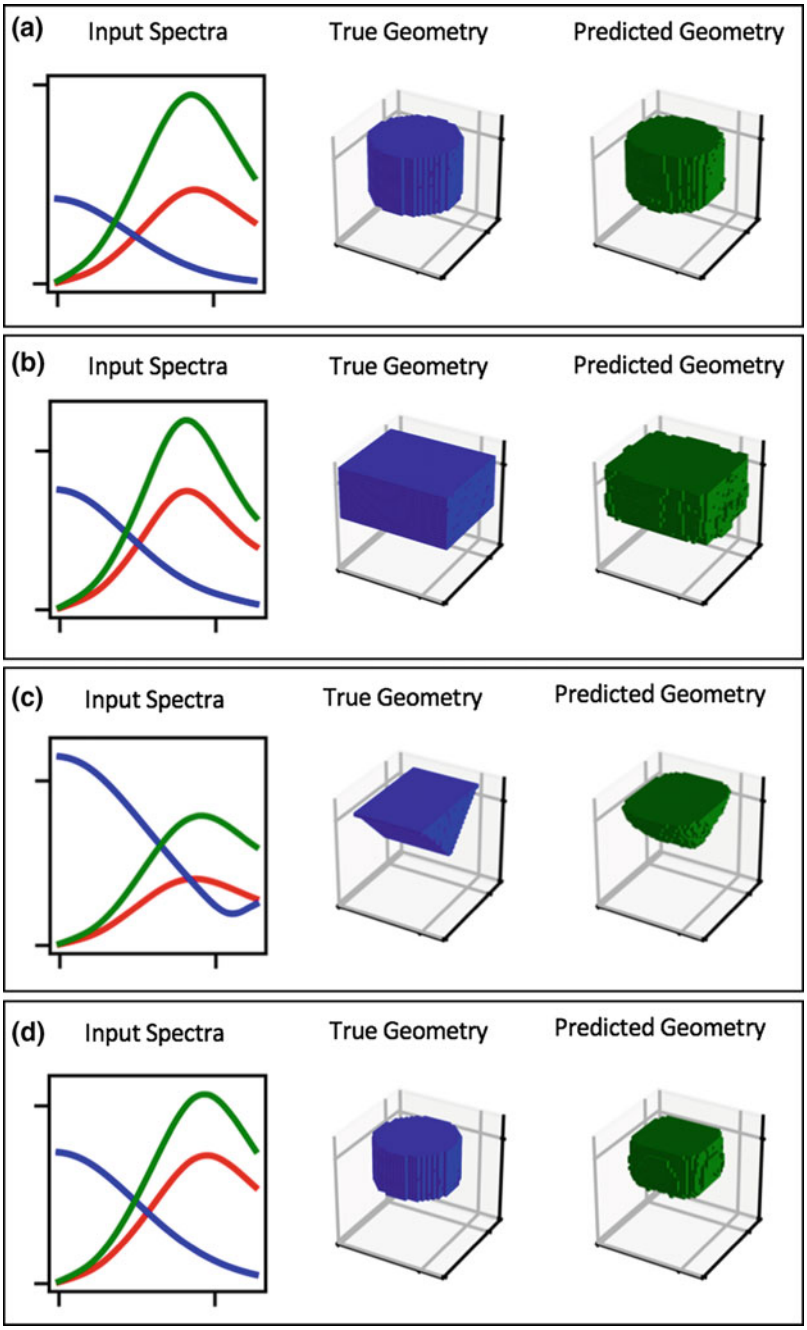

Fig. 11 Example results for synthesis network

Conclusions

The design of modern systems and products typically involves intensive computational analysis. For domains such as the design of offshore structures, these analyses can be particularly time-consuming. Standard methods for evaluating and synthesizing WECs and other offshore structure are too computationally expensive to efficiently implement within modern optimization and design algorithms. This work presented an autoencoder-based methodology for rapidly synthesizing and evaluating engineered systems in a space of reduced complexity, and applied that methodology to the synthesis and analysis of offshore structures.

The first step in the proposed methodology is the generation of data consisting of paired system design and performance information. In the offshore structure application of this paper, this consisted of voxel-based geometry paired with force spectra. The second step is the creation of two autoencoders that can compress and reconstruct both the system design and the performance information. The autoencoder for the force spectra achieved an overall reconstructive accuracy of 0.83, and provided strong qualitative reconstruction of the inputs (matching approximate range and location of maxima). The autoencoder for the voxelized geometry achieved an accuracy of 0.95 showing a strong ability to reconstruct common offshore structures, albeit with a propensity for rounding sharp corners. The third step of the methodology is the construction of networks for synthesis and analysis by reusing portions of the autoencoders. The analysis network (predicting force spectra based on geometry) achieved an accuracy of 0.87 and the synthesis network (predicting geometry based on design spectra) achieved an accuracy of 0.90. These results demonstrate that the proposed deep learning methodology is a promising means for accomplishing the rapid design of engineered systems.

Future work should investigate methods for increasing the accuracy of the autoencoders used here, as they are likely the limiting factor in the final accuracy of the analysis and synthesis networks. It may be possible to increase autoencoder accuracy through the use of convolutional layers [10] or the incorporation of generative adversarial network constructs [38]. In addition, the inclusion of eXplainable Artificial Intelligence (XAI) concepts [39–42] in conjunction with convolutional layers could provide designers with voxelized features that are aligned with high-performance solutions. Furthermore, although the geometries constructed by the synthesis network only differ slightly from the true geometries, the actual performance of the synthesized geometries is unknown. Future work should use NEMOH or another BIEM tool to directly evaluate the actual performance of synthesized geometries. In a similar vein, mapping differences between predicted and actual performance could indicate regions of the space that are particularly high performance.

Extensions of this work should also test the proposed methodology in other domains. As noted in the background section of this paper, machine learning for three-dimensional data is still nascent, particularly for synthesis tasks (typically referred to in machine learning as “generative” algorithms). Engineering design provides a large quantity of structured, three-dimensional data in the form of CAD files and pro-

totyped designs. Particularly, promising sources of training data include GradCAD and Thingiverse, online design communities in which users contribute 3D models. The existence of these, and other, sources of structured data positions the engineering design community as a future driving force in the evolution of generative machine learning methods.

Acknowledgements This material is based upon work supported by the United States Air Force Office of Scientific Research through grants FA9550-16-1-0049 and the Defense Advanced Research Projects Agency through cooperative agreement No. N66001-17-1-4064. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors. We also gratefully acknowledge the support of the NVIDIA Corporation for the donation of the Quadro P5000 GPU used in this work.

References

1. Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Magaz 11(4):26–36
2. Bhatta SR, Goel AK (1994) Discovery of physical principles from design experiences. Int J AI EDAM (AI for Eng Des Anal Manufact) (July 1992):1–22. https://doi.org/10.1017/s0890060400000718
3. Chakrabarti A, Shea K, Stone R, Cagan J, Campbell MI, Hernandez NV, Wood KL (2011) Computer-based design synthesis research: an overview. J Comput Inf Sci Eng 11(2):21003. https://doi.org/10.1115/1.3593409
4. Landry LH, Cagan J (2011) Protocol-based multi-agent systems: examining the effect of diversity, dynamism, and cooperation in heuristic optimization approaches. J Mech Des 133(2):21001. https://doi.org/10.1115/1.4003290
5. McComb C, Cagan J, Kotovsky K (2016) Drawing inspiration from human design teams for better search and optimization: the heterogeneous simulated annealing teams algorithm. J Mech Des 138(4):44501. https://doi.org/10.1115/1.4032810
6. Jain AK, Mao Jianchang, Mohiuddin KM (1996) Artificial neural networks: a tutorial. Computer 29(3):31–44. https://doi.org/10.1109/2.485891
7. Ioannidou A, Chatzilari E, Nikolopoulos S, Kompatsiaris I (2017) Deep learning advances in computer vision with 3D data. ACM Comput Surv 50(2):1–38. https://doi.org/10.1145/3042064
8. Xiang Y, Kim W, Chen W, Ji J, Choy C, Su H, … Savarese S (2016) Objectnet3D: a large scale database for 3D object recognition. Lecture Notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 9912 LNCS: 160–176. https://doi.org/10.1007/978-3-319-46484-8_10
9. Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, … Yu F (2015) ShapeNet: an information-rich 3D model repository. https://doi.org/10.1145/3005274.3005291
10. Maturana D, Scherer S (2015) VoxNet: a 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS. IEEE), pp 922–928. https://doi.org/10.1109/iros.2015.7353481
11. Garcia-Garcia A, Gomez-Donoso F, Garcia-Rodriguez J, Orts-Escolano S, Cazorla M, Azorin-Lopez J (2016) PointNet: a 3D convolutional neural network for real-time object class recognition. In: 2016 International joint conference on neural networks (IJCNN). IEEE, pp 1578–1584. https://doi.org/10.1109/ijcnn.2016.7727386
12. Achlioptas P, Diamanti O, Mitliagkas I, Guibas L (2017) Representation learning and adversarial generation of 3D point clouds, pp 1–20. Retrieved from http://arxiv.org/abs/1707.02392

13. Wu J, Zhang C, Xue T, Freeman WT, Tenenbaum JB (2016) Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. (Nips). Retrieved from http://arxiv.org/abs/1610.07584
14. Hinton GE (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507. https://doi.org/10.1126/science.1127647
15. Kingma DP, Welling M (2013) Auto-encoding variational Bayes, (Ml), pp 1–14. Retrieved from http://arxiv.org/abs/1312.6114
16. Rezende DJ, Mohamed S, Wierstra D (2014) Stochastic backpropagation and approximate inference in deep generative models. Retrieved from http://arxiv.org/abs/1401.4082
17. Salimans T, Kingma DP, Welling M (2014) Markov chain monte carlo and variational inference: bridging the gap. Retrieved from http://arxiv.org/abs/1410.6460
18. Kingma DP, Rezende DJ, Mohamed S, Welling M (2014) Semi-supervised learning with deep generative models. Retrieved from http://arxiv.org/abs/1406.5298
19. Tseng I, Cagan J, Kotovsky K (2012) Concurrent optimization of computationally learned stylistic form and functional goals. J Mech Des 134(11):111006-1–111006-11. https://doi.org/10.1115/1.4007304
20. Dering M, Tucker C (2017) A convolutional neural network model for predicting a product's function, given its form. J Mech Des 139(11):1–14. https://doi.org/10.1115/1.4037309
21. Grace K, Maher M Lou, Wilson D, Najjar N (2017) Personalised specific curiosity for computational design systems. In: Design computing and cognition '16. Springer International Publishing, Cham, pp 593–610. https://doi.org/10.1007/978-3-319-44989-0_32
22. Patel A, Andrews P, Summers JD, Harrison E, Schulte J, Laine Mears M (2017) Evaluating the use of artificial neural networks and graph complexity to predict automotive assembly quality defects. J Comput Inf Sci Eng 17(3):31017. https://doi.org/10.1115/1.4037179
23. Di Angelo L, Di Stefano P (2011) A neural network-based build time estimator for layer manufactured objects. Int J Adv Manuf Technol 57(1–4):215–224. https://doi.org/10.1007/s00170-011-3284-8
24. Xiong J, Zhang G, Hu J, Wu L (2014) Bead geometry prediction for robotic GMAW-based rapid manufacturing through a neural network and a second-order regression analysis. J Intell Manuf 25(1):157–163. https://doi.org/10.1007/s10845-012-0682-1
25. Chowdhury S, Anand S (2017) Artificial neural network based geometric compensation for thermal deformation in additive manufacturing processes, pp 1–10
26. Mørk G, Barstow S, Kabuth A, Pontes MT (2010) Assessing the global wave energy potential. In: 29th international conference on ocean, offshore and arctic engineering, vol 3. ASME, pp 447–454. https://doi.org/10.1115/omae2010-20473
27. Czech B, Bauer P (2012) Wave energy converter concepts: design challenges and classification. IEEE Ind Electron Mag 6(2):4–16. https://doi.org/10.1109/MIE.2012.2193290
28. Li Y, Yu Y-H (2012) A synthesis of numerical methods for modeling wave energy converter-point absorbers. Renew Sustain Energy Rev 16(6):4352–4364. https://doi.org/10.1016/j.rser.2011.11.008
29. Babarit A, Hals J, Muliawan MJ, Kurniawan A, Moan T, Krokstad J (2012) Numerical benchmarking study of a selection of wave energy converters. Renew Energy 41:44–63. https://doi.org/10.1016/j.renene.2011.10.002
30. McComb C, Lawson M, Yu Y-H (2013) Combining multi-body dynamics and potential flow simulation methods to model a wave energy converter. In: 1st marine energy technology symposium. https://doi.org/10.13140/rg.2.1.3817.3285
31. Ruehl K, Michelen C, Kanner S, Lawson M, Yu Y (2014) Preliminary verification and validation of WEC-Sim, an open-source wave energy converter design tool. In: Volume 9B: ocean renewable energy. V09BT09A040ASME. https://doi.org/10.1115/omae2014-24312
32. McComb C, Cagan J, Kotovsky K (2015) Lifting the Veil: drawing insights about design teams from a cognitively-inspired computational model. Des Stud 40:119–142. https://doi.org/10.1016/j.destud.2015.06.005
33. Chollet F (2015) Keras. GitHub

34. Al-Rfou R, Alain G, Almahairi A, Angermueller C, Bahdanau D, Ballas N, … Zhang Y (2016) Theano: a python framework for fast computation of mathematical expressions. arXiv e-prints. abs/1605.0. Retrieved from http://arxiv.org/abs/1605.02688
35. Babarit A, Delhommeau G (2015) Theoretical and numerical aspects of the open source BEM solver NEMOH. In: 11th European wave and tidal energy conference (EWTEC2015), Nantes, France
36. Tieleman T, Hinton G (2012) Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude
37. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86. https://doi.org/10.1214/aoms/1177729694
38. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, … Bengio Y (2014) Generative adversarial networks. Retrieved from http://arxiv.org/abs/1406.2661
39. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhutdinov R, … Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. https://doi.org/10.1109/72.279181
40. Chen K, Wang J, Chen L-C, Gao H, Xu W, Nevatia R (2015) ABC-CNN: an attention based convolutional neural network for visual question answering. Retrieved from http://arxiv.org/abs/1511.05960
41. Ba J, Mnih V, Kavukcuoglu K (2014) Multiple object recognition with visual attention. Retrieved from http://arxiv.org/abs/1412.7755
42. Mnih V, Heess N, Graves A, Kavukcuoglu K (2014) Recurrent models of visual attention. doi: 1406.6247