IASS 2020/21 SURREY 7
International Conference on Spatial Structures 2020/21

Proceedings of the IASS Annual Symposium 2020/21 and the 7th International Conference on Spatial Structures
Inspiring the Next Generation
23 – 27 August 2021, Guilford, UK
S.A. Behnejad, G.A.R. Parke and O.A. Samavati (eds.)

Machine learning for human design: Sketch interface for structural morphology ideation using neural networks

Bryan W. X. ONG*[a], Renaud DANHAIVE*, Caitlin MUELLER*

[a] Massachusetts Institute of Technology
77 Massachusetts Avenue, Room 5-418, Cambridge, MA, USA
* bryano@mit.edu

Abstract

Formal computational approaches in the realm of engineering and architecture, such as parametric modelling and optimization, are increasingly powerful, allowing for systematic and rigorous design processes. However, these methods often bring a steep learning curve, require previous expertise, or are unintuitive and unnatural to human design. On the other hand, analog design methods such as hand sketching are commonly used by architects and engineers alike, and constitute quick, easy, and almost primal modes of generating and transferring design concepts, which in turn facilitates the sharing of ideas and feedback. In the advent of increasing computational power and developments in data analysis, deep learning, and other emerging technologies, there is a potential to bridge the gap between these seemingly divergent processes to develop new hybrid approaches to design. Such methods can provide designers with new opportunities to harness the systematic and data-driven power of computation and performance analysis while maintaining a more creative and intuitive design interface. This paper presents a new method for interpreting human designs in sketch format and predicting their structural performance using recent advances in deep learning. The paper also demonstrates how this new technique can be used in design workflows including performance-based guidance and interpolations between concepts.

Keywords: conceptual design, optimization, deep learning, parametric modelling, machine learning, performance predictions.

1. Introduction

There are several key factors that determine the effectiveness of early-stage design in architecture and engineering: being able to convey information quickly, using a medium which is clearly interpretable by parties involved, having the space for creative exploration, and attaining feedback for design feasibility. The first two factors point towards sketching – an easy and discernable way to share ideas of early-stage designs. What is lacking is the ability to move past what is produced through the sketch while keeping in mind the restrictions placed by design and performance constraints. The latter two factors point towards procedural and parametric modelling – creating accurate generative models capable of producing new designs. However, this is computationally and time intensive. This paper presents a new approach for structural design that links hand sketching and performance data that builds on recent advances in machine learning. By leveraging what each established methodology is designed for, in a medium that all designers are familiar with, the objective is to create a data-driven interface that highlights all the benefits.

1.2. Related work

To address this goal, the fields of computer graphics and algorithmic structural design are both relevant, and advances from each discipline can be applied to the development of performance-driven sketch interfaces. In previous work, sketching in the realm of computer graphics has been predominantly focused on the application of some well-formulated techniques and methodologies in machine learning on sketches confined to structural and architectural representations.

There have been various advancements in machine learning involving predictions based on images. Krizhevsky et al. developed a deep convolutional neural network to effectively classify images from the ImageNet dataset [1]. In sketching, Ha and Eck in Google developed Sketch-RNN based on the QuickDraw! Dataset [2], essentially building a RNN-VAE which allows for a variety of applications such as the completion of user's sketch inputs, generation of new sketches and the interpolation between sketches. Moving further to the third dimension, generating accurate models via generative adversarial networks (Wu et al. [3]) have proved promising. Procedural modelling approaches have also been used to bridge the gap between 2D hand sketches and precise 3D models (Huang et al. [4]). With advances in stochastic optimization (Kingma & Ba [5]) founded on gradient-based optimizers, machine learning methods can presently also be trained more efficiently to meet this objective. These methods, though useful in computer graphics, do not include a performance analysis perspective which limits its usefulness for structural and architectural collaboration.

On the other side of the spectrum, design analysis and exploration have been comprehensively researched. Data-driven surrogate modelling in early-stage design of structures (Tseranidis et al. [6]) and the development of computational strategies for creative structural designs via a combination of parametric modelling and interactive optimization (Danhaive and Mueller [7]) are instances where advancements in technology have been applied to the field of structural and architectural design. However, these analytical and design tools are mostly time expensive and require substantial expertise to master. Many turn to hand drawings and sketches as a mode of communicating and exploring ideas for conceptual and early-stage design as it is both versatile and quick to perform. Murugappan et al. developed a sketch interface which provides quick finite element analysis on 2D sketch inputs [8], whereas Keshavarzi et al. used a parameterized algorithm and design constraints integration to parametrize and optimized designs of floor plans from sketch inputs in real time [9]. While these methods present a way to streamline sketching to performance analysis, they leave little room for design generation or creativity by the user, which is a key attraction of using hand drawings and sketches. These limitations are addressed by the proposed solution presented in this paper.

To develop a machine learning model, a data set containing necessary information is required. In computer vision, there have been many publicly available data sets of designs or drawings, ranging from ImageNet (Deng et al. [10]) to MNIST (LeCun et al. [11]). However, these contain data of a specific type which is not relevant to the problem posed in this paper. Databases for sketches are also available such as Google's Quickdraw! and SketchGraphs (Seff et al. [12]) which provides over 15 million real-world CAD models, containing CAD drawings, sketch augmented drawings and the procedural pipeline used to generate each one of the models. These databases, though extensive, currently have no performance analysis component attributed to them, which makes them better suited for research specific in computer graphics and not beyond. Hence, there will be a need to create a new data generation pipeline, which is presented in the paper.

2. Methodology

2.1. Conceptual overview

This research introduces a method to predict the relative performance of sketch inputs that are structural and architectural in nature to facilitate the design process. There are three functionalities that are specifically enabled by this paper's method:

1. Given a hand sketch of a design, provide real-time prediction of structural performance.
2. Given a hand sketch of a design, suggest better-performing alternatives.

3. Given two design sketches, return a set of new options that visually combine aspects of both (and that perform well).

This is achieved through the application of two concepts from machine learning: variational autoencoders, which project sketch data into compact latent space representations, and surrogate models, which learn relationships between design inputs and performance outputs, shown in Fig. 1. This process is challenging to develop for the complete landscape of open-ended sketches; this paper presents a step towards this goal by focusing on applying this method to specific problem types. From a learning perspective, this means that relevant training data is organized by and developed for specific structural problem types.

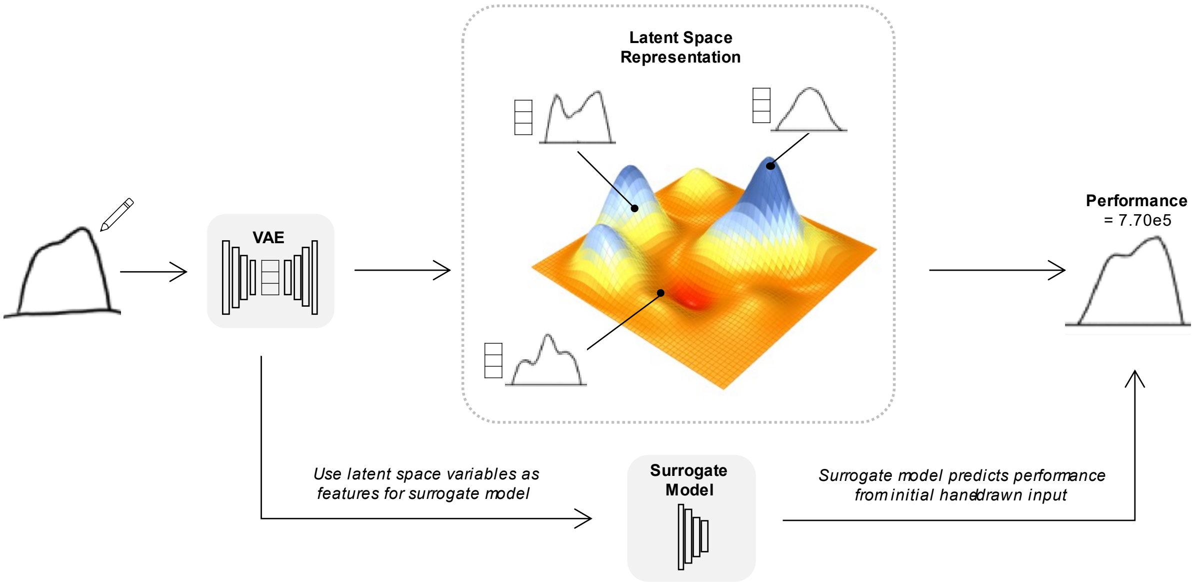


Figure 1: Two-tier surrogate model from hand-drawn input to performance prediction.

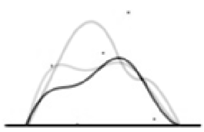
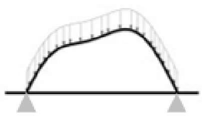
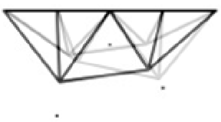
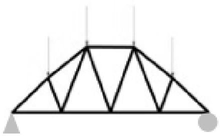
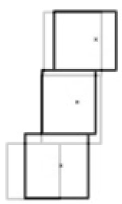
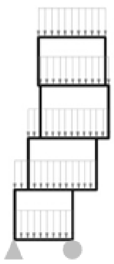
Because natural sketch data from structural problems does not exist at the large scale and in the standardized format needed for machine learning methods, this paper relies on synthetic sketch data generated algorithmically. The synthetic sketch data is designed to mimic human sketches as closely as possible, so that models trained on synthetic data can then be used with real human sketches. Three specific structural model types are used as case studies in this paper: curved frames, spanning trusses, and “stacked box” buildings, summarized in Table 1.

Figure 2: Real examples of chosen structural types: Heydar Aliyev Cultural Center (left, adapted from [13]), Rey Vitacura (middle, adapted from [14]) and the New Art Museum (right, adapted from [15]).

The methodology is explained through the first model type, curved frames, for clarity. In this model, the curved geometry is both the structural frame and the outer boundary of a building, and represents geometrically interesting design options of widely varying structural performance. In the following sections, the process for developing this sketch-based method are given: Data generation (2.2.1), Data processing and augmentation (2.2.2), Two-tier surrogate model (2.2.3). Finally, Section 2.3 outlines how this method can be used to support design tasks.

The presented method is implemented using open-source platforms to allow for easy prototyping and broad, public application in the future. TensorFlow by Google is used as the open-source artificial intelligence library which acts as the basis of the neural network architecture described later in the paper. The model is scripted in Python on Google Colaboratory, a Jupyter notebook and open-source web application environment which supports the TensorFlow library.

Table 1: Test cases with corresponding details.

Case	Variables	Structural Modelling		Performance Metric
Curved frames	 6 variables: vertical locations of control points 1 variable: smoothness		• Pin-pin support conditions set at either ends of frame • Distributed vertical load of 40kN/m (8kN/m x 5m secondary span) applied along frame. • Frame spans a distance of 30m.	Linear elastic FEA, Performance = (strain energy) x (total length of frame)
Trusses	 6 variables: vertical locations of control points 1 variable: number of subdivisions		• Pin-roller support conditions set at either ends of truss • Vertical load of 10kN applied at each node of upper chord • Truss spans a distance of 10 meters	Linear elastic FEA, Performance = summation over structure of (element axial force) x (element length)
Stacked boxes	 5 variables: locations of control points 2 variables: width of boxes 1 variable: number of stacks		• Pin-roller support conditions set at either ends of lowest box • Uniform vertical load of 75kN/m applied on floor of each stacked box • Width of box ranges from 7m to 10m, height of box ranges from 5m to 15m, depth of each box fixed at 20m • Additional structural x-bracing modelled for each stacked box to transfer lateral loads	Linear elastic FEA, Performance = (strain energy) / (total volume of stacked boxes)

2.2. Method description

2.2.1. Generating data sets

An important consideration in working computationally with sketches is the data representation format. In previous work, sketches have been represented as sequences of drawing operations (Ha and Eck [2]) or interpreted into CAD primitives (Murugappan et al. [8]). However, the most common way to work with sketches in a learning method is to represent them as rasterized or bitmap images, which are 2D matrices of pixels. This has the advantage of reasonably high-fidelity visual representation and offers the opportunity to work with the most well-studied machine learning technology of the last several years, convolutional neural networks (CNNs), which operate efficiently on image data. Images also work equally well to represent both real sketch data (scanned from hand drawings or digitally created with a stylus) and synthetic sketch data needed to train models. For these reasons, this paper's research works with image representations of sketches.

The design data needed to train the learning models is generated by sampling parametric models created using Rhino3D and Grasshopper3D. The data generated consists of 2D images of designs of the three model types, curved frames, trusses, and stacked boxes, along with their corresponding performance evaluation metrics. The details of each test case can be seen in Table 1. For each case, a structural model is created automatically and analyzed using Karamba3D, as described in the table. A structural performance metric is associated with each generated design, with a lower value being better.

The open-source Design Space Exploration (DSE) tools suite (Brown et al. [16]), specifically the Sampler and Capture tools, is used to iterate and capture the images and target performance properties of the model.

A dataset of 20,000 bitmap images and their corresponding performance scores are sampled from the proposed data generating pipeline. As seen from Fig. 3, the sampled designs have a wide distribution of performance scores, with a lower performance score corresponding to a better performing design with respect to the abovementioned performance metric. Using this method, a myriad of designs can be generated that approximately encompass the variety of designs that might be drawn by a human user.

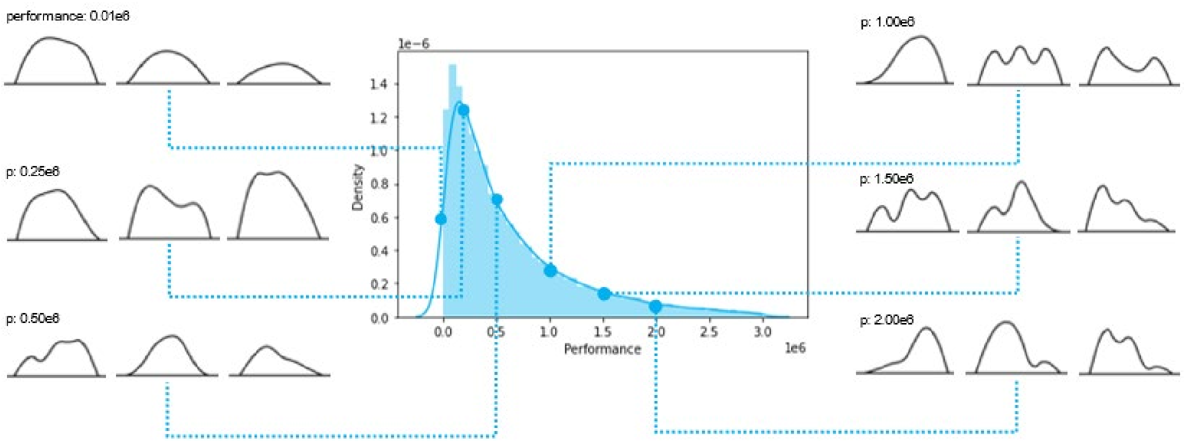


Figure 3: Distribution of structural performance of generated curved frames designs.

2.2.2. Data processing and augmentation

The bitmap images obtained are resized to 72x72 pixels to reduce resolution, which increases the efficiency of the model training while retaining the general aspect ratio of the image. Data augmentation methods are applied to mimic human imperfection found in typical hand drawn sketches. Two such methods that were initially explored were Gaussian white noise and elastic deformation of images: salt and pepper noise was added to recreate blank spaces and discontinuity found in hand sketches, and elastic deformation was explored to impersonate the imperfect non-straight lines of human drawings.

While these methods were promising, they were difficult to replicate effectively for all the data and were more dissimilar to hand sketches than intended. The augmentation technique finally chosen is the application of random, elastic deformations on images using the open source Augmentor Python package (Bloice et al. [17]). Images produced with this method more effectively mimic the nuanced behavior present in the drawings (see Fig. 4).

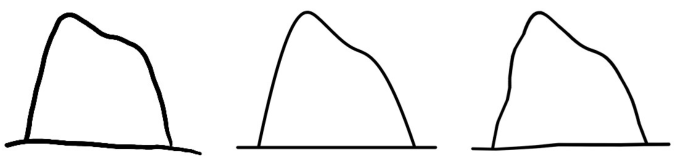


Figure 4: Hand drawn sketch (left), digitally modelled curved frame (middle) and augmented image (right) using Augmentor package [17].

2.2.3. Two-tier surrogate model

A two-tier surrogate model is trained using the augmented data described above. The first tier consists of an encoder and decoder, which reduces the dimensionality of the data into a latent space via training it against reconstruction loss. The second tier utilizes the lower-dimensional latent variables assigned to each of the input to predict its performance by training it against a loss function.

The encoder essentially reduces the features needed to describe the data used via a form of dimensionality reduction. The encoder of a variational autoencoder (VAE) distinguishes itself from other predictive neural networks such as CNNs in the process of returning the distribution of the latent space instead of regressive values. A regularization term expressed as the Kullback–Leibler divergence, which measures differences in probability distributions, regularizes the latent space to obtain one which follows closely to a standard normal distribution. This then allows for smooth sampling of the latent design space required for the proposed design generation technique. The decoder then works as a reverse bottleneck which reconstructs the data from the latent layer into information, which is understandable by the user, which in this case is the pixel-based image of the design. The loss function which is minimized for the VAE is a reconstruction term, and the model is trained and fine-tuned by measuring the accuracy in which it reconstructs the training input data.

The neural network in the second tier of the model is a regression model that predicts a continuous value based on a low-dimensional vectorial representation of a design in the latent space described above. While the performance-predicting surrogate model could also operate directly on image data, early experiments by the authors found that this approach is much less effective. A standard multi-layer perceptron neural network model is used with a mean squared error (MSE) loss function to train the second model.

The synthetic sketch data set is split into three groups: training, validation, and testing. Training data is used for model training and validation data for tuning. This process is repeated using different hyperparameters to obtain models containing the best learning algorithm that produce the lowest error in prediction. Test data is used to evaluate the performance of the resultant model.

Once the two-tier surrogate model is trained, real sketch data can be input into the system, and a numerical performance prediction is instantaneously returned as feedback for the designer.

2.3. Design space generation

In performance-driven design processes, rapid performance predictions are quite useful in their own right (Tseranidis et al. [6]), but they can also be used to support design space exploration (Brown et al. [16]). In this paper, the VAE developed to generate encoded data also allows for exploration of the latent design space, as shown in recent previous work for non-sketches (Danhaive and Mueller [15]). Using the latent features found in this space, new designs can be generated that relate and respond to those input by the user.

2.3.1. Design latent space

As explained previously, the VAE is used to generate encoded data in the form of the latent vector which is then utilized as input for the surrogate model. In generating the latent vector, the design latent space of the chosen structure is also be created. This means that the user can move through the newly defined design space to discover potential new designs.

Through the regularized training of the encoder and decoder of the VAE, a normally distributed latent design space is obtained. This space contains features that can be interpreted into physical designs by the corresponding trained decoder. The dimensionality of these features, or latent space vectors, is subjective and based on the amount needed to achieve a model of high fidelity, and hence dependent on the complexity of the design in question. The extracted latent space vectors provide additional information which, when examined, can be useful to the overall design process. Since the latent space vectors usually contain high-dimensional data, a parallel coordinate plot might be employed to visualize the data of n-dimensions. The combined latent features form the latent vector defining a specific design which can then be interpreted by the decoder. By examining the visualized parallel coordinates plots of the data, it is possible to ascertain specific latent space features that contribute more to the performance of a design, or other features that might be of interest to the designer.

3. Results and case studies

This section applies the methodology described above to the curved frame and other case studies for evaluation and demonstration of its potential use.

3.1. General method evaluation

The method is evaluated in three sections which correspond to the stages in which data is processed and trained against the proposed two-tier surrogate model. Total loss is used as the evaluation metric to train and test the variational autoencoder to recreate input sketches and obtain a lower dimensional latent vector. Model prediction accuracy visualizes the difference in values between the actual performance of a test sketch input and the predicted performance using the trained surrogate model. Lastly, human drawn sketches are used to test the efficacy of the model on real world data.

3.1.1. Total loss of variational autoencoder

The total loss that the variational autoencoder is trained against is the sum of the reconstruction loss and Kullback–Leibler loss. The total loss of the final train model was determined for each of the input data sampled. A density distribution curve is plotted to show how total loss varies with the type of designs present in the data. As seen in Fig. 5, a higher loss corresponds to a fuzzier reconstruction. The complexity and relative realism of designs are largely correlated with the VAE's ability to detect and reconstruct them. Hence, the VAE seems to perform better for more intuitive and realistic designs of structures, but even the designs with poor total loss values are reconstructed quite well. This means that the VAE latent space is a trustworthy design space for exploring variations on sketch inputs.

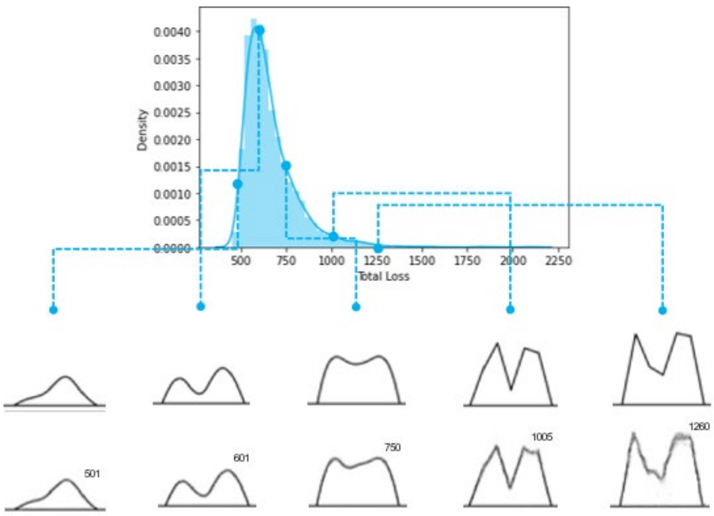


Figure 5: Distribution of total (reconstruction) loss of the trained variational autoencoder across all of the test data (from synthetic sketches). In the called-out designs, the top row shows the original inputs, and the bottom row shows the reconstructions. Even the designs with high reconstruction loss are very recognizable visually compared to the inputs.

3.1.2. Model prediction accuracy

Model prediction accuracy is determined by the absolute difference between actual performance values of designs and predicted values normalized by the median performance score of all designs generated. Designs that are more similar to real life structures or sketches of structures made by humans have a lower error value as compared to more complex and unrealistic designs. Hence, it can be argued that the model presented has a high efficacy for realistic designs of structures.

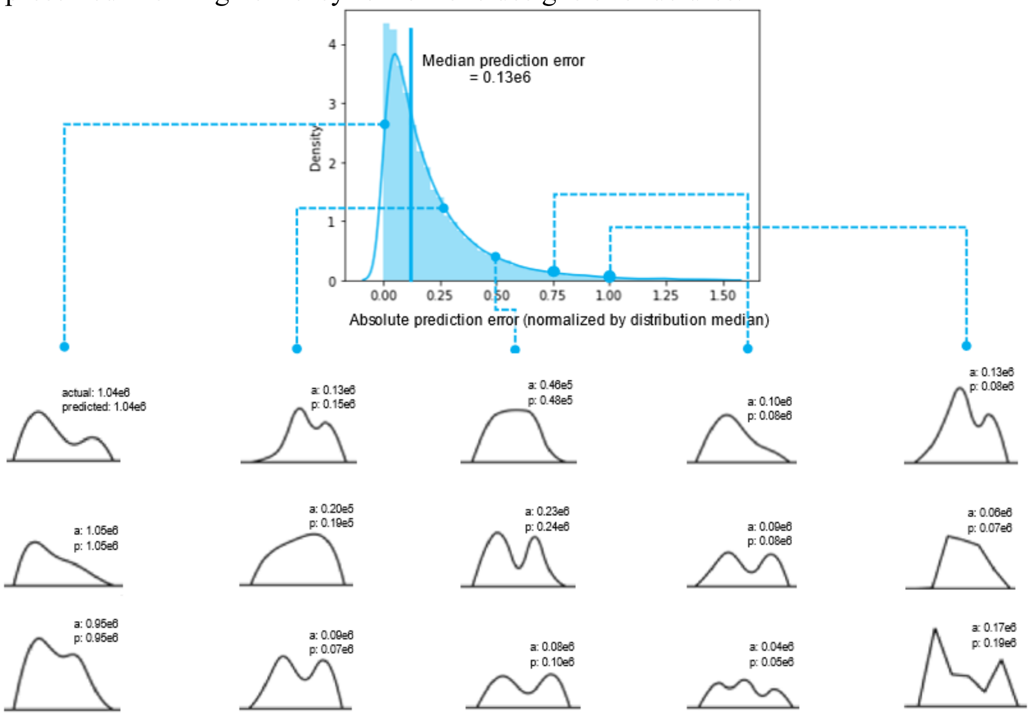


Figure 6: Most designs' absolute prediction error less than 0.25 that of actual value. Distribution median = 0.04e6.

3.1.3. Hand drawings

After fine-tuning the model by the changing of hyperparameters, the trained model is applied to new, unseen input data to predict the structural performance previously set. Hand-drawn input sketches shown in Fig. 4 are presented to the model and designs are reconstructed by the decoder. The predicted performance using the surrogate model for each input is then normalized against the best performing design. Generally, sketches that correspond to designs that are structurally unsound are predicted to have worse (higher) performance scores.

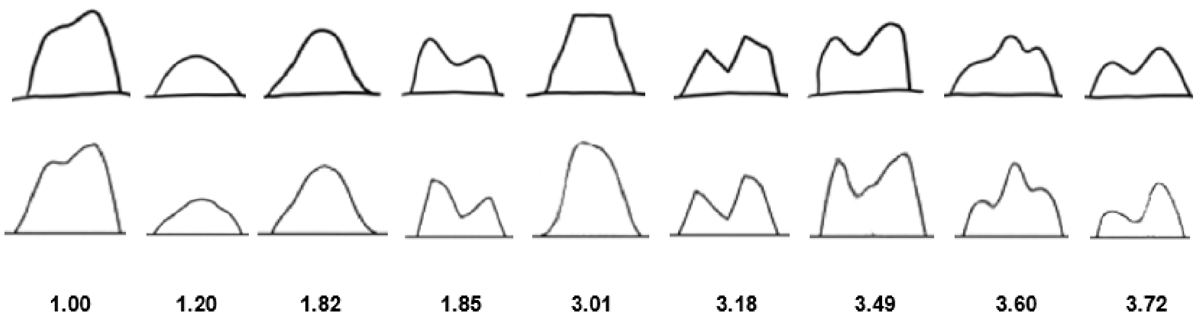


Figure 7: Reconstructions (middle) and normalized predicted performances (bottom) from hand-drawn input designs (top).

3.2. Design suggestions

As shown in Fig. 1, the training of a variational autoencoder in the first tier of the model allows for the generation of a latent space. In doing so, inputs can be encoded into latent vectors which corresponds to points in the latent space. Inputs can then be interpolated to view the transformation between one input design and another, resulting in intermediate designs that contains attributes of both designs. This morphing of designs can be seen in Fig. 8.

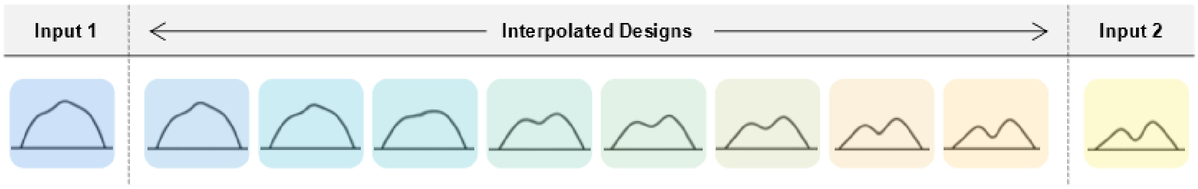


Figure 8: Suggestions via interpolations.

Through the performance prediction of the surrogate model, the performance score can be added as an additional dimension to the latent space created earlier. In addition to generating designs containing similar features to the initial input, a performance condition can be applied to also generate designs that are better performing according to the predetermined performance metric. In Fig. 9, improved designs are filtered from latent space in the vicinity of the initial sketch input design.

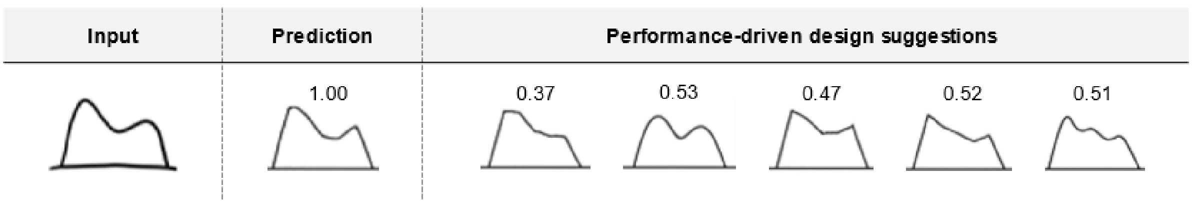


Figure 9: Performance-driven suggestion normalized against predicted performance of hand-drawn input design.

3.3. Other test cases

In addition to the curved frame designs, the same process is applied to the other test cases previously mentioned, specifically for trusses and stacked boxes. The number of latent dimensions and hyperparameters are changed according to the requirements for the separate designs. The models with curved frame and truss required 16 latent dimensions, whereas the model with the stacked box required 10. Similar results are obtained for interpolating between input designs, and performance-based design suggestions, as shown in Fig. 10 and Fig. 11.

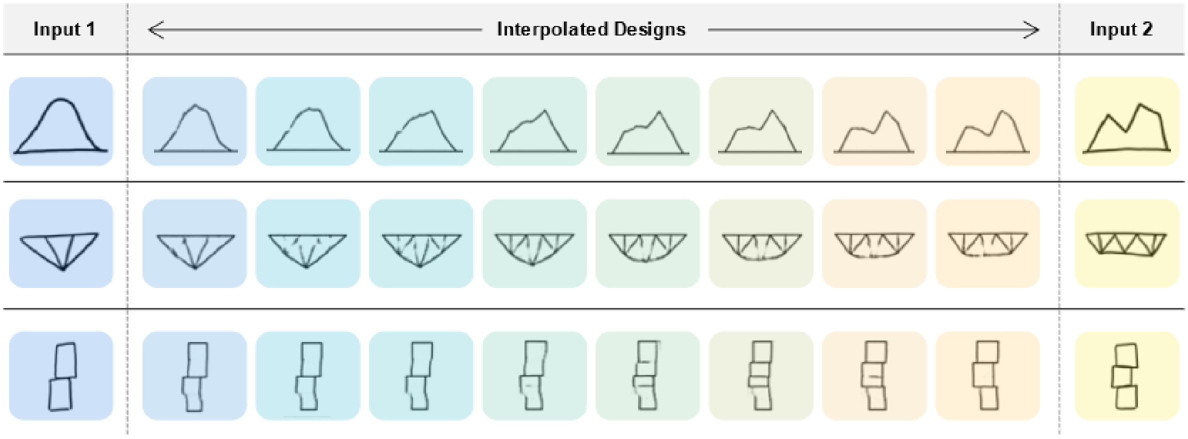


Figure 10: Suggestions via interpolations of 3 selected test cases from hand sketches.

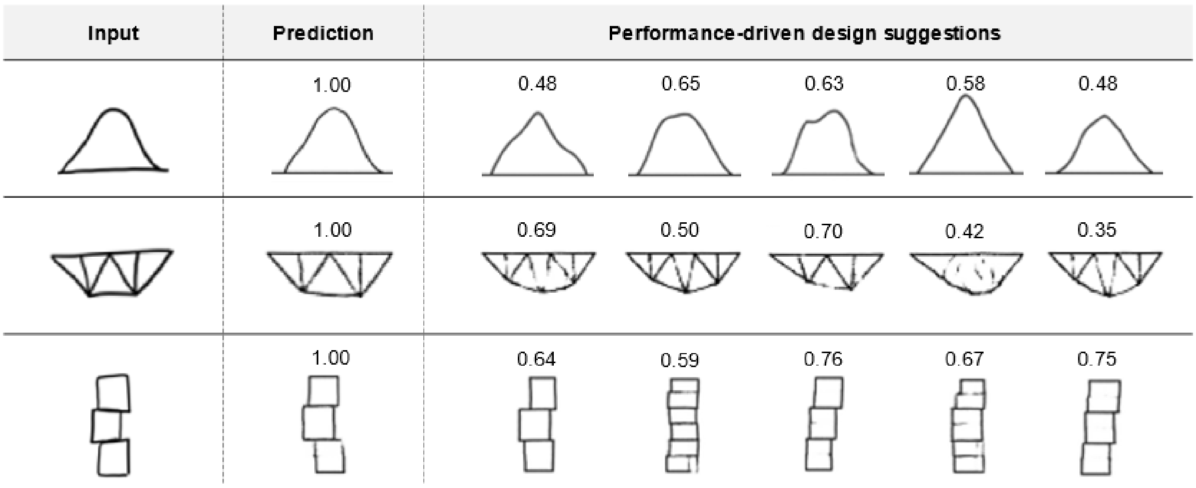


Figure 11: Performance-driven suggestion normalized against predicted performance of hand-drawn input design of three selected test cases.

3.4. Abnormal cases

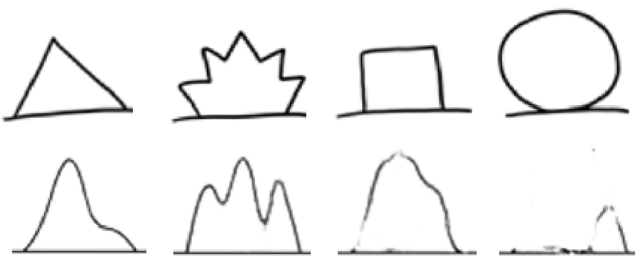


Figure 12: Reconstructions of abnormal designs from hand drawing.

Complex cases dissimilar to those in the training data set are given as input for the model to recreate, as shown in Fig. 9. The reconstructions, and similarly predictions, vary greatly according to the closeness in which the inputs corresponded with the data used. Even so, the model is still able to reconstruct features similar to the initial sketch input for some cases, while failing to detect any similarities for others. This reveals the limits of this approach, and the importance of diverse synthetic sketch data for training the learning models.

4. Conclusion

4.1. Potential impact

This new method of analyzing and generating designs from sketches can have various applications in the fields of engineering, architecture, and design. Being able to analyze and predict relative performance of sketch designs may enable aspects of performance to be included in the early-stage design discussions. This can guide designers towards potentially better performing structures without compromising the initial design intent. Furthermore, the ability to generate new design morphologies based on initial sketch inputs by designers can help facilitate collaboration between different individuals, which in turn fuels the iterative and collaborative nature of the design process. For example, multiple designers such as engineers or architects may have specific designs that they want to put forward. Further designs could be generated using this paper's proposed interpolation method that will not only show a compromise from the initial designs, but also produce new possible varieties for the designers to consider.

4.2. Limitations and future work

As with all machine learning models, the predictions and behavior of this paper's method are highly dependent on the training data. Currently, the methodology proposed is limited by the CAD models used to generate the data for training, which comes with their inherent limitations and biases. One possible goal for the future might be to create a public database to record sketches done by designers, architects, and engineers alike on structures that have been used in the design process. This platform would not only allow individuals to view the variety and complexity of designs that are being created, but also serve as a basis for future research into sketches primarily for the design process. Furthermore, as with the case of computer graphics and machine learning, further methods could be developed to bring performance-informed design generation from sketches to the next stage. Interpolation methods presented here could be expanded to account for performance, compromising between designs input by users and overall design improvement. Another future direction is expanding the number of possible inputs for design interpolations to 3, 4, or more, which aims to further improve the collaboration process between designers. Lastly, as alluded to earlier in the paper, it might be useful for users to directly manipulate the latent variables after starting with an initial design. This can be done so in the form of an interactive parallel-coordinates plot, in which users can view patterns and effects of changes specific latent variables and generate new designs by exploring the latent design space manually.

4.3. Concluding remarks

This paper presents a new method to interpret sketches to predict structural performance, and introduces potential applications of the method for design generation. This work aims to promote and inspire future research and development in the area of design analysis and generation for structural and architectural sketches.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1854833.

This research is made possible by sponsorship from JTC Corporation, Singapore.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," p. 9.

[2] D. Ha and D. Eck, "A neural representation of sketch drawings," *ArXiv Prepr. ArXiv170403477*, 2017.

[3] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Advances in neural information processing systems*, 2016, pp. 82–90.

[4] H. Huang, E. Kalogerakis, E. Yumer, and R. Mech, "Shape synthesis from sketches via procedural models and convolutional networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 8, pp. 2003–2013, 2016.

[5] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv14126980 Cs*, Jan. 2017, Accessed: Apr. 01, 2021. [Online]. Available: http://arxiv.org/abs/1412.6980.

[6] S. Tseranidis, N. C. Brown, and C. T. Mueller, "Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures," *Autom. Constr.*, vol. 72, pp. 279–293, Dec. 2016, doi: 10.1016/j.autcon.2016.02.002.

[7] R. A. Danhaive and C. T. Mueller, "Combining parametric modeling and interactive optimization for high-performance and creative structural design," in *Proceedings of IASS Annual Symposia*, 2015, vol. 2015, no. 20, pp. 1–11.

[8] S. Murugappan and K. Ramani, "Feasy: a sketch-based interface integrating structural analysis in early design," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2009, vol. 48999, pp. 743–752.

[9] M. Keshavarzi, C. Hutson, C.-Y. Cheng, M. Nourbakhsh, M. Bergin, and M. R. Asl, "SketchOpt: Sketch-based Parametric Model Retrieval for Generative Design," *ArXiv Prepr. ArXiv200900261*, 2020.

[10] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[11] Y. LECUN, "THE MNIST DATABASE of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, Accessed: Apr. 01, 2021. [Online]. Available: https://ci.nii.ac.jp/naid/10027939599/.

[12] A. Seff, Y. Ovadia, W. Zhou, and R. P. Adams, "SketchGraphs: A Large-Scale Dataset for Modeling Relational Geometry in Computer-Aided Design," *ArXiv Prepr. ArXiv200708506*, 2020.

[13] "zaha hadid heydar aliyev," *designboom | architecture & design magazine*, Jul. 15, 2013. https://www.designboom.com/architecture/zaha-hadid-heydar-aliyev-cultural-center-shapes-azerbaijan/ (accessed Apr. 14, 2021).

[14] "10 Projects That Feature Striking Steel Trusses," *ArchDaily*, Jul. 27, 2017. https://www.archdaily.com/876423/10-projects-that-feature-striking-steel-trusses (accessed Apr. 14, 2021).

[15] "SANAA / Kazuyo Sejima + Ryue Nishizawa, Dean Kaufman · New Art Museum," *Divisare*. https://divisare.com/projects/289283-sanaa-kazuyo-sejima-ryue-nishizawa-dean-kaufman-new-art-museum (accessed Apr. 14, 2021).

[16] N. C. Brown, V. Jusiega, and C. T. Mueller, "Implementing data-driven parametric building design with a flexible toolbox," *Autom. Constr.*, vol. 118, p. 103252, Oct. 2020, doi: 10.1016/j.autcon.2020.103252.
[17] M. D. Bloice, C. Stocker, and A. Holzinger, "Augmentor: An Image Augmentation Library for Machine Learning," *ArXiv170804680 Cs Stat*, Aug. 2017, Accessed: Apr. 12, 2021. [Online]. Available: http://arxiv.org/abs/1708.04680.