Proceedings of the IASS 2022 Symposium affiliated with APCS 2022 conference
Innovation·Sustainability·Legacy
19 – 22 September 2022, Beijing, China

Enhancing structural form-finding through a text-based AI engine coupled with computational graphic statics

Zifeng GUO*[a], Karla SALDANA OCHOA[a], Pierluigi D'ACUNTO[b]

*[a] University of Florida, College of Design, Construction and Planning, School of Architecture
1480 Inner Rd, Gainesville, FL 32611, USA,
guo.zifeng@ufl.edu
[b] Technical University of Munich, School of Engineering and Design, Professorship of Structural Design

Abstract

This paper introduces *Text2Form3D*, a machine-learning-based design framework to explore the embedded descriptive representation of structural forms. *Text2Form3D* relies on a deep neural network algorithm that joins word embeddings, a natural language processing (NLP) technique, with the Combinatorial Equilibrium Modeling (CEM), a form-finding method based on graphic statics. *Text2Form3D* is trained with a dataset containing structural design options generated via the CEM and labeled with vocabularies acquired from architectural and structural competition reports. For the labeling process, an unsupervised clustering algorithm Self Organizing Maps (SOM) is used to cluster the machine-generated design options by quantitative criteria. The clusters are then labeled by designers using descriptive text. After training, *Text2From3D* can autonomously generate new structural solutions in static equilibrium from a user-defined descriptive query. The generated structural solutions can be further evaluated by various quantitative and qualitative criteria to constrain the design space towards a solution that fits the designer's preferences.

Keywords: machine learning, deep neural networks, natural language processing, combinatorial equilibrium modeling, structural design, form finding.

1. Introduction

1.1 Structural design and artificial intelligence

Artificial intelligence (AI) is challenging the role of the human designer by creating end-to-end generators capable of producing infinite solutions. AI can provide fast predictive answers, handle a large amount of data, and calculate quicker than human designers. Still, its computational power cannot substitute yet the critical thinking that designers engage in the creative process of design. Researchers in computational generative design have long sought methods to elevate a computer from a drafting tool to an intelligent agent capable of creating designs autonomously (e.g., [1][2]). Currently, many research projects focus on developing design machines that automatically generate design solutions without any assistance from human designers (e.g., [3][4][5]). These machines can systematically search within the design space defined by the user's input and produce near-optimal solutions for many well-defined problems with specific design objectives and given constraints, such as minimum use of materials for load path optimization (e.g., [6][7]). The use of massive crowdsourced online datasets has recently pushed this schema further where realistic images can be automatically generated from pure text inputs (e.g., [9][10]). Table 1 shows a summary of the state-of-the-art on AI and generative algorithms.

Table 1. State-of-the-art research in the field of generative architectural and structural design supported by artificial intelligence

Reference	**Field**	**Model**	**Control Features**	**Algorithms**
Nauata *et al.* [3]	floor plan design	graph to image (room segmentation masks)	bubble diagram + layout image	Generative Adversarial Networks (GANs)
Chang *et al.* [4]	building 3D layout design	graph to voxels	bubble diagram + layout 3D image	GANs
Chaillou [5]	floor plan design	image to image	layout contour image + layout image	GANs
Wolf D. Prix & Partner [8]	architectural rendering	image to image	plain color rendering of 3D geometries	GANs
Ramesh *et al.* [9]	language modeling and generative AI	text to image	descriptive text	Variational autoencoders + Transformers
Oppenlaender [10]	language modeling and generative AI	text to image	descriptive text + random noise input	Diffusion model + Contrastive Language-Image Pretraining (CLIP)
Tseranidis *et al.* [11]	truss design	design parameters to multiple engineering performances	input parameters of a truss generator + a fixed topology	Multiple supervised learning algorithms + evolutionary optimization
Fuhrimann *et al.* [12]	shell structure design	form clusters to CEM inputs	node coordinates of structural forms	SOM + CEM
Bertagna *et al.* [13]	building envelope design	solar radiation simulation to CEM inputs	geometric, structural, and sun-shading features	SOM + CEM + Ladybug
Saldana Ochoa *et al.* [14]	structural form design	user preferences to CEM inputs	yes-no label from the user + high-order statistics of structural forms	SOM + CEM
Harding [15]	structural form design	user selection of form clusters	input parameters of form generators	SOM

It can be argued that these approaches are still in the early stage for architectural and structural design applications as the proposed overall workflows are not always able to grasp the complexity of forms inherited from the creative nature of the design process. In fact, several aspects such as user-specific qualitative preferences, and perceptions of spaces and structures, cannot always be adequately formulated as well-defined design objectives. As a result, designers still lack the appropriate tools to tackle architectural and structural design comprehensively.

1.2 Objective

This article focuses on structural design – a discipline at the interface between architecture and structural engineering – as a case study to test the use of AI to assist the human designer within the design process. The present study builds upon the interaction between human and artificial intelligence, with the machine being able to process semantic requests related to structural forms that go beyond the sole quantitative aspects of structural engineering. The study proposes a shift from a conventional, deterministic form-finding process (i.e., generation of forms that are structurally optimized for given boundary conditions) to an open, creative process in which text inputs from the designer are translated to spatial structures in static equilibrium.

The main outcome of this study is the development of *Text2Form3D*, a machine-learning-based design framework to explore the semantic representation of structural design forms. *Text2Form3D* combines a Natural Language Processing (NLP) algorithm [16][17], which translates text into numerical features, with the Combinatorial Equilibrium Modeling (CEM) form-finding algorithm [18][19], which generates structural forms from topological and metric inputs. This connection between NLP and CEM allows *Text2Form3D* to suggest possible design parameters based on the input texts specified by the designer, thus allowing the designer to explore the design space of the CEM method semantically.

2. Method overview

Figure 1 shows the pipeline of *Text2Form3D*. In the first step, the CEM algorithm generates a dataset of various structural forms with randomly initialized design parameters. CEM is an equilibrium-based form-finding algorithm based on vector-based 3D graphic statics [18][19]. The CEM algorithm constructs the equilibrium form of a structure (CEM output) sequentially, based on a given topology of the structure and associated metric properties (CEM input). The dataset contains both the design parameters (CEM input) and the generated forms (CEM outputs).

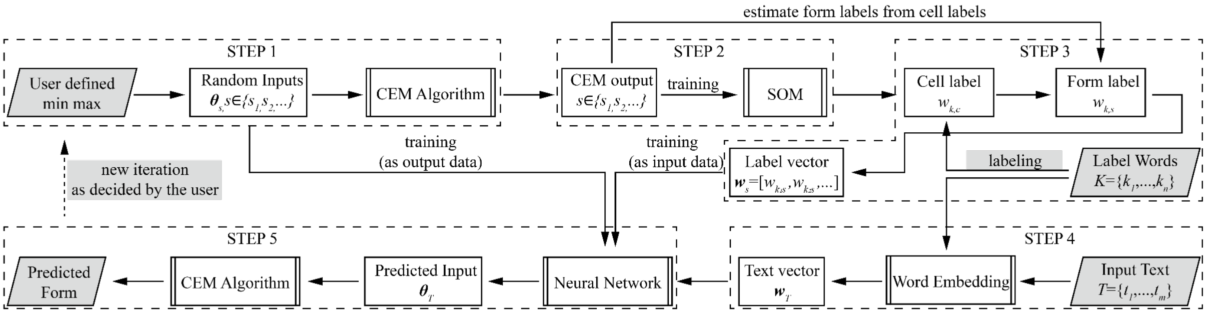

Figure 1. The workflow of *Text2Form3D*. User's operations are colored in light gray to differentiate from the computational process.

The generated forms are clustered in the second step using Self-Organizing Maps (SOM) [20]. The numerical feature vector to train the SOM is based on user-defined quantitative criteria, such as their formal metric characteristics in terms of edge lengths and node positions [12][14].

In the third step, the designer uses the trained SOM to label several thousands of design options using vocabularies acquired from architectural and structural practice competition reports. These labels are additional information manually assigned to each of the generated structural forms to create data to train a machine learning algorithm.

After labeling the forms, in the fourth step, the text labels are processed using word embedding [16][17], a technique that converts the words to numerical vectors of several hundreds of dimensions based on the similarities of their context words. The embedding vectors represent the contextual information of the words and make two words comparable in terms of their semantic meanings. The semantic similarity of words can be obtained by the Euclidean distances of the embedding vectors.

In the fifth step, a deep neural network is trained using the text vectors as the input and the design parameters of CEM as the output. The trained neural network can then generate the CEM input parameters based on new embedding vectors produced from any input texts.

The above design workflow was tested in a design experiment to generate towers.

3. Generation of the structural forms

The generation of the structural forms through the CEM relied on a fixed topology, i.e., all the generated forms have the same number of nodes and edges (Figure 2). In our experiment, the topology has a 4×4 grid on a 10-floor tower, corresponding to 176 nodes and 400 edges. For each generated form, the metric parameters (i.e., the lengths of trail edges and the internal force magnitudes of deviation edges) were randomly initialized within fixed ranges. The position of nodes in the generated form was constrained by several control planes, which kept the nodes of each floor on the same level. The vertical position of the control planes was determined by floor heights randomly sampled from a list of predefined values. Based on this setup, a dataset of 87,728 towers was created. Figure 2 shows the topology diagrams (i.e., node-edge relations) and the form diagrams (i.e., the generated structural forms) of two different combinations of input metric parameters.

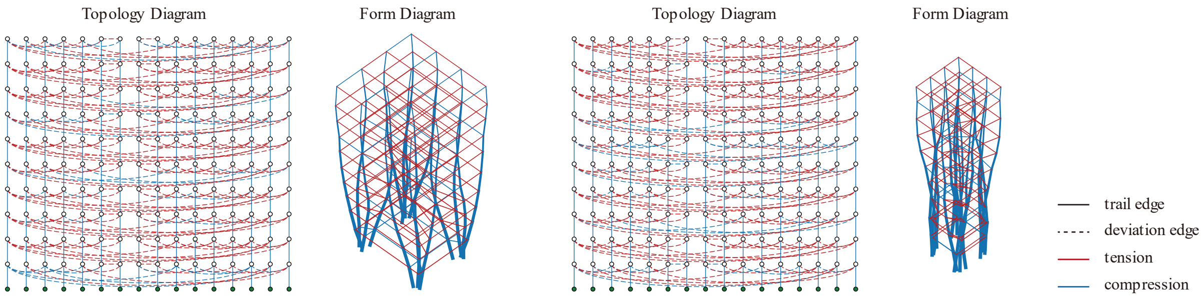

Figure 2. Different combinations of input metric parameters produce structural forms in equilibrium with different geometries.

4. Form clustering and text labeling

To label the generated forms one by one using texts would take an enormous amount of human labor. Therefore, the experiment relied on a clustering labeling strategy to save time and improve the labeling consistency. The assumption is that similar text labels can represent forms within the same cluster.

4.1 Self-Organizing Map

A SOM was trained to cluster the generated forms to support the labeling process based on the geometric characteristic of the forms. The input of the SOM was 192-dimensional vectors computed from the nodes' x-y positions of the structural forms using Discrete Fourier Transform (DFT). The process was implemented by first computing the DFT on the x and y coordinates of each trail path of the structural form and then concatenating the results to a single vector. DFT extracts the spatial patterns of the nodes from their x-y positions, allowing to compare the structural forms based on their geometric similarities. Unlike methods for extracting geometric features such as raw node coordinates [12] and high-order statistics [13][14], DFT finds a compromise between the computational efficiency and the representation of geometric details. Figure 3 shows the trained SOM, consisting of a grid of 20×20 cells.

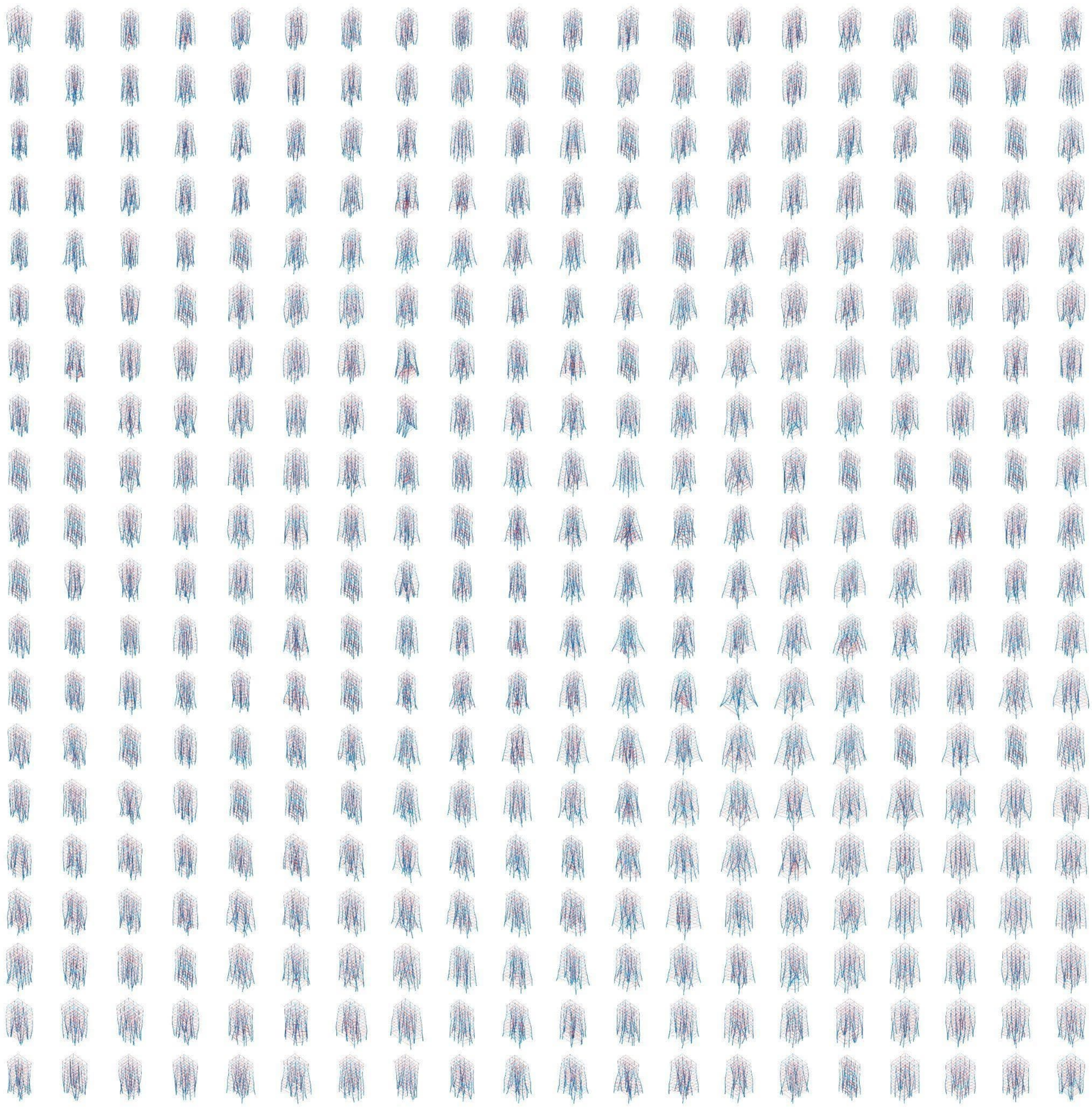

Figure 3. A trained SOM based on a grid of 20×20 cells. Each cell is represented by its best-matching geometry.

4.2 Rapid labeling using the form SOM

To support the labeling process, several common words used to describe architectural projects were identified by crawling jury commentaries for winning projects in architectural competitions [21]. The crawling process specifically focused on architectural projects that included the keyword "tower" to match the database for this experiment, resulting in more than 200 commentaries. These commentaries were pre-processed by 1) lower-casing and de-accenting all the words; 2) removing stop words; 3) selecting all adjectives; and 4) deleting those that appear less than three times, leading to 105 commonly used adjectives. These adjectives were finally manually filtered to erase redundancy, resulting in 68 meaningful adjectives from which the text labels used in this experiment were created. Figure 4 shows a pie chart representing these adjectives and their relative frequency.

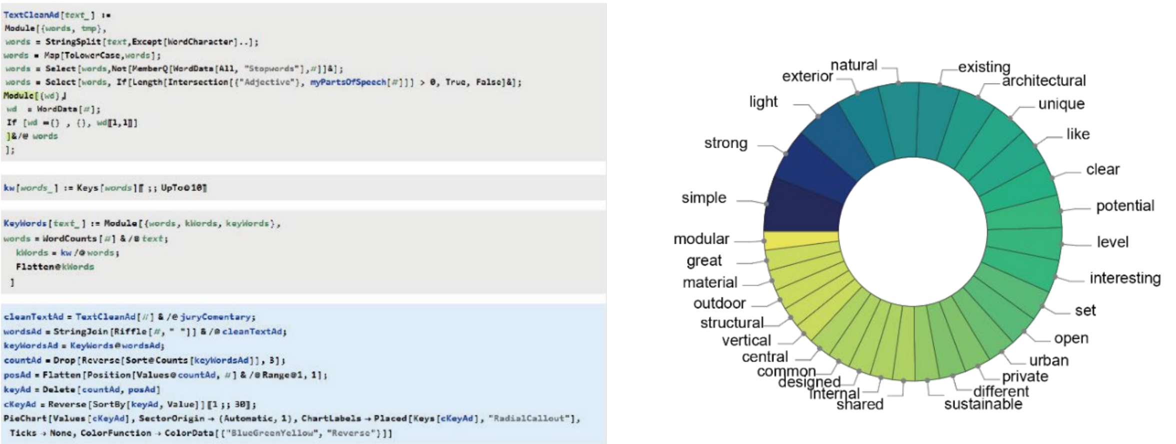

Figure 4. A pie chart of the selected adjectives (right) derived from commentary texts of architectural jury reports using text processing methods (left).

The labeling process was implemented by assigning tags to cells on the SOM grid, with each tag being defined by one of the selected adjectives and a normal distribution (Figure 5). The association of the tags to the grid cells could be controlled by adjusting the parameters (i.e., the sigma and weight values) related to the normal distribution, while the same tag could be assigned to different regions of the SOM. After assigning all necessary tags on the SOM, the labels of each grid cell of the SOM were generated as the association values between the cell and the selected adjective. Let a denote a tag, c a grid cell of the SOM, and k an adjective, the association value $w_{k,c}$ is computed as

$$w_{k,c} = \frac{\sum_{a \in A(k)} G(a)(\|p(a)-p(c)\|)}{\sum_{a \in A} G(a)(\|p(a)-p(c)\|)} \tag{1}$$

where A represents all tags, $A(k)$ is the set of tags that are associated with the same adjective k, p is a function that returns the spatial location of a tag a or a grid cell c, and $G(a)$ is the gaussian function of the normal distribution associated with a.

After labeling the SOM, the text labels of all structural forms were estimated using the Euclidean distances between each form to all the grid cells of the SOM. Let s denote a structural form, v_s and v_c the weight vectors associated with the structural form s and the grid cell c. The text label of a given structural form can be obtained as the association value $w_{k,s}$ between adjective k and structural form s, which is computed as

$$w_{k,s} = \frac{\sum_c \|v_s - v_c\|^{-a} w_{k,c}}{\sum_c \|v_s - v_c\|^{-a}} \tag{2}$$

where a is a user specified scalar that was set to 8 in our experiments.

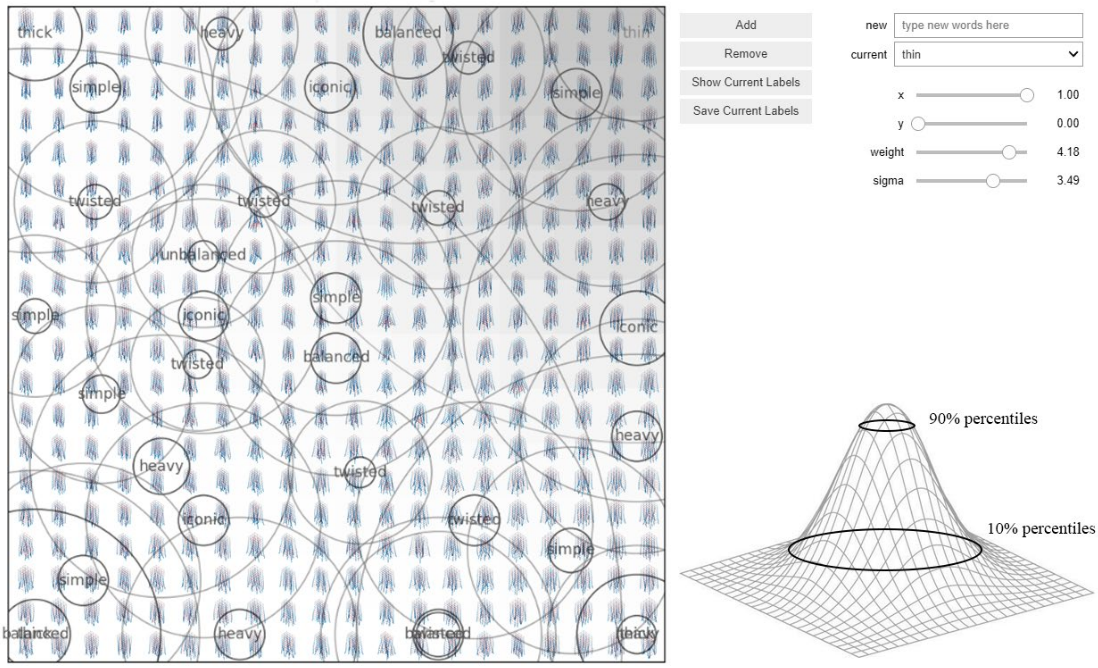

Figure 5. The user interface of the text-based labeling process, where each adjective is associated with a Gaussian distribution shown by the circles representing the 10% and 90% percentiles.

5. Regeneration of the design parameters

The generated labels represent the association values between structure forms and the selected adjectives. Therefore, the text label of each structural form can be represented as a vector that consists of the association values to all adjectives $K=\{k_1, k_2, \ldots, k_n\}$. This vector is denoted as

$$w_s = [w_{k_1 s}, \ldots, w_{k_n s}] \tag{3}$$

As the selection of adjectives varies and solely depends on the designer's personal preferences, we consider the adjectives as the anchor points that convert input texts to association values that are comparable to the generated labels. We used word embedding for this process so that *Text2Form3D* could also process arbitrary words that are not included in the adjectives labeled by the designer.

5.1 Text vector using word embedding

Word embedding converts each word to an embedding vector of a few hundred dimensions that represents the semantic meaning of the words and makes two words comparable. The embedding vectors allow the designer to use any word rather than the limited selected adjectives for the regeneration process. To convert a text input of arbitrary words to a vector comparable to the form-adjective association value $w_{k,s}$, we interpreted $w_{k,s}$ as the frequency of adjective k showing up in the text label corresponding to the structural form s. Let T denote the input text, which consists of $\{t_1, \ldots, t_m\}$ words; the text vector of an arbitrary text input can be computed as

$$w_T = \frac{1}{|T|} \sum_{t \in T} w(t) \tag{4}$$

where $\boldsymbol{w}(t)$ is the association vector of arbitrary word t with all adjectives $K=\{k_1, k_2, \ldots, k_n\}$. This vector is computed as

$$w(t)_i = \frac{\|E(k_i) - E(t)\|^{-a}}{\sum_j \|E(k_j) - E(t)\|^{-a}} \text{ for } i = 1, \ldots, n \tag{5}$$

where a is a user-specified scalar, and E is the word embedding model. In our experiment, we set a=8 and used the *GloVe300* model of *Wolfram Mathematica* [22] for E.

5.2 Regeneration using artificial neural networks

The artificial neural network *Text2Form3D* was used in this step to regenerate structural forms by predicting the CEM parameters (i.e., the design inputs) from the text labels. The input data of the neural network consisted of the text vector $\boldsymbol{w}_s$ and a normally distributed random vector. The output data contain the CEM parameter $\boldsymbol{\theta}_s$ of the structural form, which include the information of node positions, edge lengths of trail edges, and force magnitudes of deviation edges. This network was trained using the generative adversarial network (GAN) method, which uses a discriminator network attached to the generative network to differentiate the predicted CEM parameters from real CEM parameters. The discriminator network substitutes the conventional loss function and fits the network to the data distribution of the structural form dataset. After training, the discriminator network was discarded, and the generative network was used for form regeneration. Figure 6 shows the structure of the generative network of *Text2Form3D*.

For the regeneration of 3D structural forms, the text vector $\boldsymbol{w}_T$ was first obtained from an input text of the designer. The text vector was then fed to the generative network to obtain the corresponding CEM parameters $\boldsymbol{\theta}_T$. As the input data of the generative network consisted of both the text vector and the normally distributed random vector, the designer could generate multiple results by feeding random vectors created from a different random seed. Finally, the produced CEM parameters were provided to the CEM algorithm to obtain the structural forms as the final output.

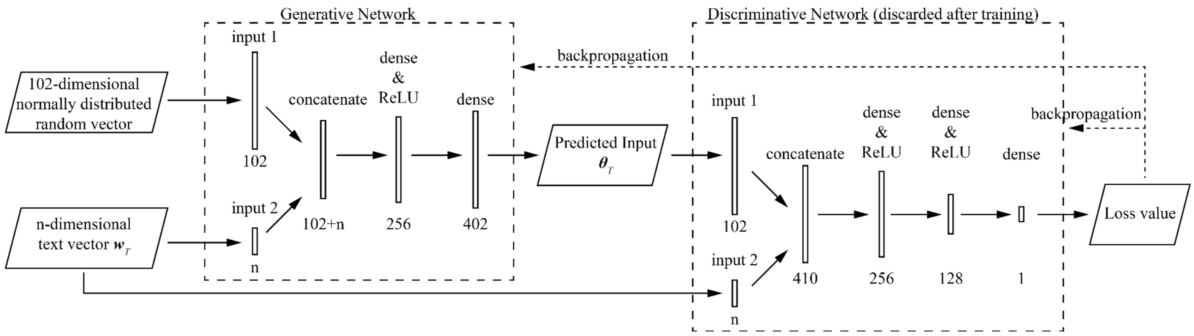

Figure 6. The generative network which predicts the design parameters of CEM method from input texts.

6. Results and discussion

Figure 7 shows four groups of 3D structural forms generated using different input adjectives following the proposed pipeline. Each group clearly shows the geometric characteristics that can be differentiated from other groups, which means that *Test2Form3D* successfully captured the semantic information of the input adjectives and encoded them to the output geometries. For each group, the generated forms follow the same tendency of geometric features while still exhibiting a great variety regarding the geometric details. This result suggests that the normally distributed random vector serves as an effective method to explore the formal variants of the same semantic inputs. Figure 8 shows four samples selected from each group and their corresponding text input.

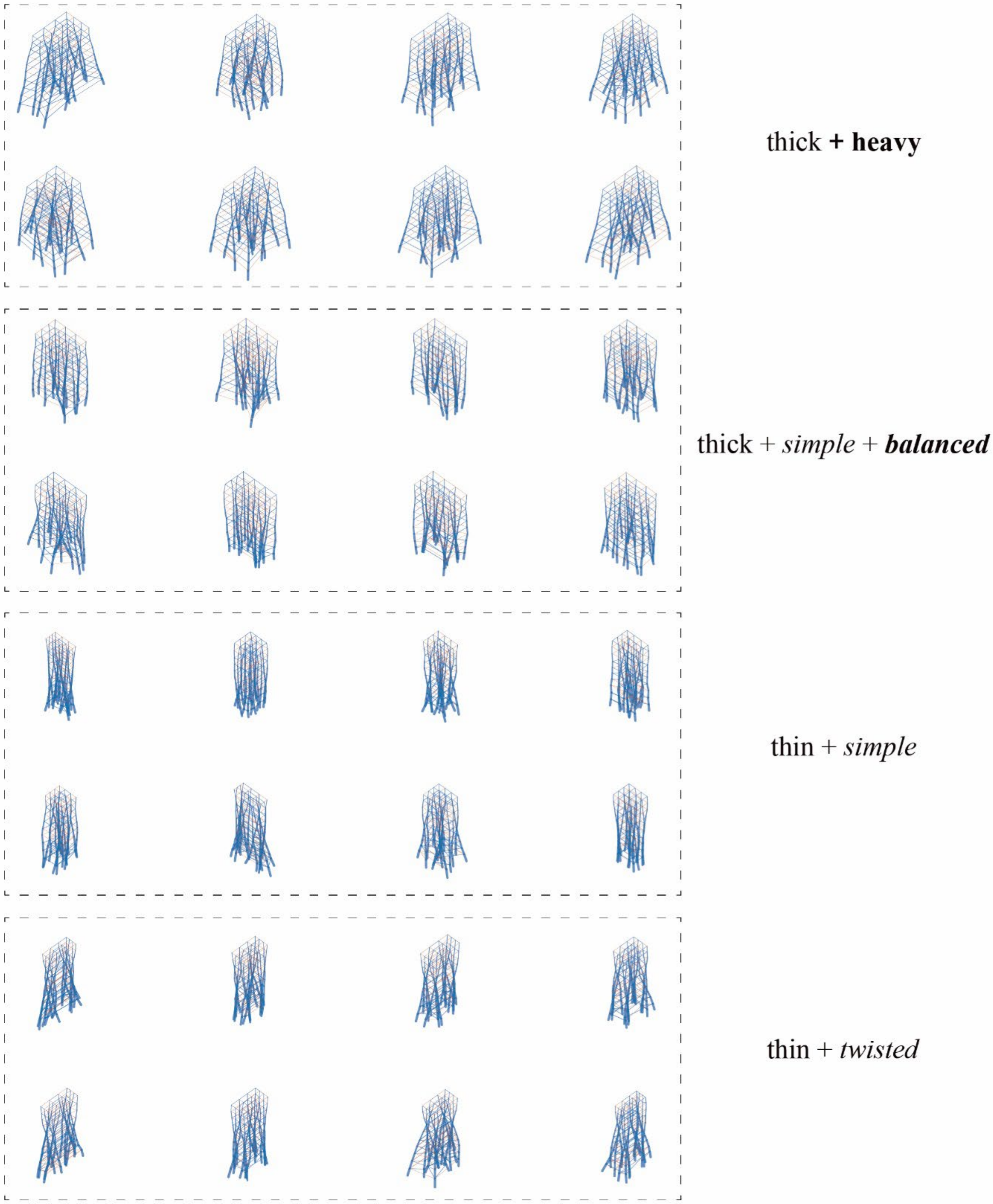

Figure 7. Generated structural forms using the designer-specified input words.

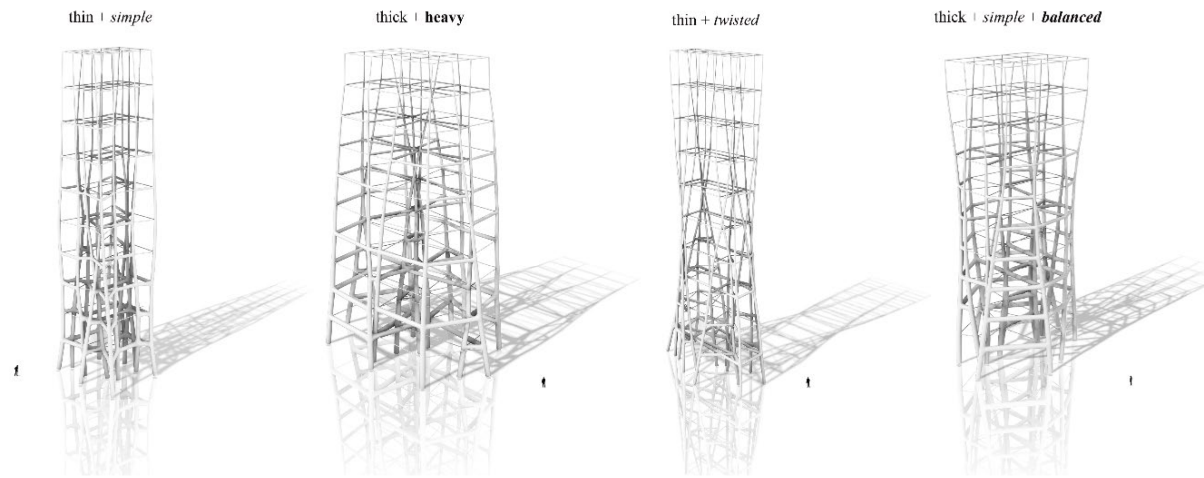

Figure 8. Renders of four samples selected among the generated forms.

The proposed workflow demonstrates the effectiveness of the interplay between human designers and AI in designing structural forms through descriptive text and quantitative parameters as inputs. However, despite the promising results shown above, there are still limitations that require further investigations in future work. First, an immediate next step is to overcome the current restriction of having a predefined topology for structural forms. This would allow applying the proposed approach to other types of structural forms and structural design algorithms. Second, the labeling process shown above relies on the assumption that similar forms have similar semantic interpretations. This assumption needs to be validated by experiments testing different labeling strategies.

7. Conclusions

The presented study focused on exploring and generating 3D structural forms through a text-based AI engine coupled with computational graphic statics. The project proposes a workflow in which the structural forms are generated based on static equilibrium and the designer's semantic interpretation of 3D geometries. Such workflow emphasizes the interplay between designers and AI where the generative algorithm can adapt to the personal preferences of designers and act as a suggestion engine for the semantic interpretation of structural forms. This approach shows the potential of AI not only in solving well-defined problems but also in tackling those weakly-defined tasks that are inherited from the creative nature of design. Finally, an alternative understanding of AI's role within the design process is presented, by proposing a shift from the conventional input-to-output process (i.e., image to image or text to image) to a translation of text into spatial geometries (i.e., text to 3D structural forms).

Acknowledgments

The authors gratefully acknowledge the Nvidia AI Technology Center at UF for its technical support and for granting access to computing resources.

References

[1] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool, "Procedural modeling of buildings", in *ACM SIGGRAPH 2006 Papers*, pp. 614-623, 2006.

[2] P. Merrell, E. Schkufza, and V. Koltun, "Computer-generated residential building layouts", in *ACM SIGGRAPH Asia 2010 Papers*, pp. 1-12, 2010.

[3] N. Nauata, K. H. Chang, C. Y. Cheng, G. Mori, and Y. Furukawa, "House-gan: Relational generative adversarial networks for graph-constrained house layout generation", in *European Conference on Computer Vision*, pp. 162-177, 2020.

[4] K. H. Chang, C. Y. Cheng, J. Luo, S. Murata, M. Nourbakhsh, and Y. Tsuji, "Building-GAN: Graph-Conditioned Architectural Volumetric Design Generation", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11956-11965, 2021.

[5] S. Chaillou, "Archigan: Artificial intelligence x architecture", in *Architectural Intelligence*, pp. 117-127, 2020.

[6] X. Huang, and M. Xie, *Evolutionary topology optimization of continuum structures: methods and applications*, John Wiley & Sons, 2010.

[7] O. Sigmund, "A 99 line topology optimization code written in Matlab", *Structural and Multidisciplinary Optimization*, vol. 21(2), pp. 120-127, 2001.

[8] Wolf D. Prix & Partner, "Deep Himmelblau", https://coop-himmelblau.cn/method/deep-himmelblau/, 2021.

[9] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, ... and I. Sutskever, "Zero-shot text-to-image generation", in *International Conference on Machine Learning*, pp. 8821-8831, 2021.

[10] J. Oppenlaender, "The Creativity of Text-based Generative Art", arXiv:2206.02904, 2022.

[11] S. Tseranidis, N. C. Brown, and C. T. Mueller, "Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures", *Automation in Construction*, vol. 72, pp. 279-293, 2016.

[12] L. Fuhrimann, V. Moosavi, O. P. Ohlbrock and P. D′Acunto, "Data-Driven Design: Exploring new Structural Forms using Machine Learning and Graphic Statics", in *Proceedings of the IASS Symposium 2018 - Creativity in Structural Design*, pp.1-8, 2018.

[13] F. Bertagna, P. D′Acunto, P. O. Ohlbrock and V.Moosavi, "Holistic Design Explorations of Building Envelopes Supported by Machine Learning", *Journal of Facade Design & Engineering*, vol. 9(1), pp. 31-46, 2021.

[14] K. Saldana Ochoa, P. O. Ohlbrock, P. D'Acunto, and V. Moosavi, "Beyond typologies, beyond optimization: Exploring novel structural forms at the interface of human and machine intelligence", *International Journal of Architectural Computing*, vol. 19(3), pp.466-490, 2021.

[15] J. Harding, "Dimensionality reduction for parametric design exploration", in *Proceedings of the conference on advances in architectural geometry*, pp. 9-13, 2016.

[16] T. Mikolov, W.T. Yih, and G, Zweig, "Linguistic regularities in continuous space word representations", in *Proceedings of the 2013 conference of the north American chapter of the association for computational linguistics: Human language technologies*, pp. 746-751, 2013.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality", in *Advances in neural information processing systems 26*, 2013.

[18] P. O. Ohlbrock, and P. D'Acunto, "A Computer-aided Approach to Equilibrium Design based on Graphic Statics and Combinatorial Variations", *Computer-Aided Design*, vol. 21, 102802, 2019.

[19] P. O. Ohlbrock, P. D'Acunto, J. P. Jasienski, and C. Fivet, "Vector-based 3d graphic statics (part iii): designing with combinatorial equilibrium modelling", in *Proceedings of IASS Annual Symposia,* pp. 1-10, 2016.

[20] T. Kohonen, "Exploration of very large databases by self-organizing maps", in *Proceedings of international conference on neural networks* ,1997.

[21] BUILDNER, *Architecture competition results,* https://architecturecompetitions.com/results, 2022.

[22] Wolfram Research, "GloVe 300-Dimensional Word Vectors Trained on Wikipedia and Gigaword 5 Data", *Wolfram neural net repository*, https://resources.wolframcloud.com/NeuralNetRepository/resources/GloVe-300-Dimensional-Word-Vectors-Trained-on-Wikipedia-and-Gigaword-5-Data/, 2022.