

Machine Learning-Enabled Parameterization Scheme for Aerodynamic Shape Optimization of Wind-Sensitive Structures: A-Proof-of-Concept Study

Shaopeng Li¹, Brian M. Phillips^{*2} and Zhaoshuo Jiang³

¹University of Louisiana at Lafayette, Lafayette, LA 70504, USA

²University of Florida, Gainesville, FL 32611, USA (corresponding author: brian.phillips@essie.ufl.edu)

³San Francisco State University, San Francisco, CA 94132, USA

(Received keep as blank, Revised keep as blank, Accepted keep as blank)

Abstract. Aerodynamic shape optimization is very useful for enhancing the performance of wind-sensitive structures. However, shape parameterization, as the first step in the pipeline of aerodynamic shape optimization, still heavily depends on empirical judgment. If not done properly, the resulting small design space may fail to cover many promising shapes, and hence hinder realizing the full potential of aerodynamic shape optimization. To this end, developing a novel shape parameterization scheme that can reflect real-world complexities while being simple enough for the subsequent optimization process is important. This study proposes a machine learning-based scheme that can automatically learn a low-dimensional latent representation of complex aerodynamic shapes for bluff-body wind-sensitive structures. The resulting latent representation (as design variables for aerodynamic shape optimization) is composed of both discrete and continuous variables, which are embedded in a hierarchy structure. In addition to being intuitive and interpretable, the mixed discrete and continuous variables with the hierarchy structure allow stakeholders to narrow the search space selectively based on their interests. As a proof-of-concept study, shape parameterization examples of tall building cross sections are used to demonstrate the promising features of the proposed scheme and guide future investigations on data-driven parameterization for aerodynamic shape optimization of wind-sensitive structures.

Keywords: aerodynamic shape optimization; wind-sensitive structures; machine learning; autoencoder; parameterization scheme

1. Introduction

Recent decades have witnessed a rapid increase in the height of buildings and the span of bridges, making them very sensitive to wind loads. Numerous efforts have been made to mitigate the wind effects of these slender structures so that the safety and serviceability requirements can be satisfied. In addition to tuning structural properties and implementing structural control, one promising approach for wind mitigation of slender structures is to modify the external shapes directly. Conventionally, aerodynamic shapes of bluff-body civil structures are usually modified by trial and error from a limited number of candidates. For example, high-rise buildings can leverage corner modification of the cross sections and may further adopt modifications along the height, such as helical twisting (Davenport, 1971; Tanaka et al., 2012). Long-span bridges often resort to edge fairing or central slots (Nagao et al., 1993; Yang et al., 2015). Although this intuition-based cut-and-try is routinely used by the wind engineering community, a mathematically optimal (or near optimal) aerodynamic shape is not necessarily guaranteed (Dulikravich, 1992). To this end, there is a need for an automated process to facilitate the comprehensive search of shape design space, which can be realized by coupling optimization-guided search with the efficient evaluation of

aerodynamic performance (Topping, 1983; Ding and Kareem, 2018). As shown in Figure 1, a typical aerodynamic shape optimization process usually encompasses three components, namely, shape parameterization, performance evaluation, and search optimization. Numerous efforts have been made in the wind engineering community to advance aerodynamic shape optimization, which usually focuses on performance evaluation and search optimization. For example, Ding and Kareem (2018) constructed a multi-fidelity surrogate model from CFD simulation, on which the optimization algorithm is performed. Li et al. (2021) proposed a deep reinforcement learning-based shape optimization strategy to reduce the time-consuming CFD-based performance evaluations. Whiteman et al. (2022) developed a mechatronic building model for the efficient evaluation of different aerodynamic configurations in the wind tunnel, which accelerated the shape optimization process. Despite recent advances in performance evaluation and search optimization, shape parameterization, as the first step of the aerodynamic shape optimization pipeline, has not been well studied. In fact, existing studies still heavily depend on empirical judgment, which uses only a small number of parameters to define the geometry of bluff-body civil structures (e.g., Ding and Kareem, 2018; Montoya et al., 2018). The resulting small design space may fail to cover a wide range of promising shapes, and hence could hinder the realization of the full potential in aerodynamic shape optimization.

Determining a proper aerodynamic shape parameterization scheme moves beyond simple selections

^{*} Corresponding author: Brian M. Phillips
Email: brian.phillips@essie.ufl.edu

of a few discrete choices. It is a complex process involving intensive iterations between structural engineers and architects, where a transparent procedure is currently unavailable. To the authors' best knowledge, existing studies on aerodynamic shape optimization of wind-sensitive structures have not explicitly discussed the justification of their adopted shape parameterization schemes. Considering this knowledge gap, it may be desirable to resort to a closely related field of airfoil shape optimization and leverage its recent advances to inform aerodynamic shape parameterization of wind-sensitive structures. For airfoils, the streamlined shapes are usually described using some smooth base functions, such as B-spline and Bézier curves (e.g., Lepine et al., 2001; Venkataraman, 1995), and the parameters controlling the base functions are used as the design variables for the aerodynamic shape optimization. Due to the manual selection of base functions, the design space dimensionality is usually higher than the underlying dimensionality that represents sufficient shape variability (Chen et al., 2017; Chen et al., 2020). The high dimensionality of design variables compromises the search efficiency of the optimal shape. To reduce the number of design variables for shape optimization, researchers have used linear models for dimension reduction, such as principal component analysis (e.g., Cinquegrana and Iuliano, 2017) and single value decomposition (e.g., Allen et al., 2018). However, these models with linear nature may fail to extract the compact representation for the nonlinear cases of real-world airfoil design. To this end, it is promising to utilize deep neural network-based nonlinear dimension reduction techniques, such as autoencoders (Wang et al., 2016) and generative adversarial networks (Hallaji et al., 2021). As a salient example of airfoil shape parameterization, Chen et al. (2020) leveraged generative adversarial networks to project a wide range of airfoil shapes in the existing database (UIUC Airfoil Coordinates Database) to a low-dimensional latent space characterized by only a few continuous variables, which resulted in an accelerated shape optimization process.

Although the abovementioned studies in aerospace engineering could shed light on aerodynamic shape parameterization of wind-sensitive structures, some unique features of bluff-body civil structures need to be highlighted and require additional considerations. Unlike streamlined airfoils that can be conveniently characterized by some smooth base functions, it may be difficult to analytically describe the shape of bluff-body civil structures with sharp edges. Noting the challenges of using analytical approaches, it is tempting to use data-driven approaches to learn the parameterization schemes directly from the possible shapes of wind-sensitive structures. While the aerospace engineering community has built well-documented databases (e.g., UIUC Airfoil Coordinates Database) to cover a wide range of possible airfoil shapes, the same cannot be said for bluff-body civil structures, which requires a systematic collection of possible shapes that are both physically realizable and aesthetically pleasing. It is noted that the construction of this database not only needs

to include the configurations of real-world civil structures, but also may require to encompass a wide range of conceptual designs to enrich the database, which is not a trivial task. Before initializing the time-consuming process of constructing the database involving collaborations between structural engineers and architects, it is desirable to first ensure that there exist suitable data-driven parameterization schemes satisfying the unique demands arising from bluff-body shapes, which is the focus of this study.

Considering that the shape data is usually in the form of coordinates of the points along the edges or the pixels of the raw shape images, it is necessary to reduce the high-dimensional shape data into a low-dimensional latent representation for efficient parameterization. It is noted that the state-of-the-art dimension reduction schemes used for airfoil shape parameterization (e.g., Chen et al., 2020) cannot be effectively applied to the case of bluff-body shapes. Unlike airfoil shapes that can be fully represented by a set of continuous variables, discrete variables are also needed to characterize distinct variations across different categories of the bluff-body shapes. For example, tall building cross sections with recessed corners and chamfered corners should be distinguished by a discrete variable. In addition to the importance of including discrete variables in the latent space, it should also be noted that hierarchy structures may exist in the discrete and continuous variables. For example, the discrete variables could be considered as high-level variables to classify tall building cross sections into different categories (e.g., recessed corners and chamfered corners) while the continuous variables are the low-level variables to capture the variations within each category (e.g., the depth of recessed corner and the slope of chamfer edge). Capturing the hierarchy structures makes it possible to consider conditional relations among the latent variables (i.e., certain variables are only active when some other variables are active). In fact, introducing the mixed discrete-continuous variables with hierarchy structures makes the learned parameterization more intuitive and interpretable, considering it is closer to the underlying generation mechanism (Ross et al., 2021; Ross and Doshi-Velez, 2021). Parameterization of bluff-body shapes using mixed latent variables with hierarchy structures will also benefit subsequent aerodynamic shape optimization. First, it gives stakeholders the flexibility to focus on certain shape types instead of exhausting the whole design space. Another benefit is that varying the variables under their parent category will not result in a drastic change in the shape and hence the aerodynamic performance. This satisfies many optimization schemes' underlying assumption of the smooth objective function.

To this end, this study leverages recent advances in learning hierarchical representations (Ross and Doshi-Velez, 2021) and proposes a novel parameterization scheme for aerodynamic shape optimization of wind-sensitive structures. This machine learning-enabled scheme first learns the hierarchy structure of the mixed discrete and continuous latent variables. Then it constructs a shape

generator conditioned on the obtained hierarchy structure, which could be effectively used as the shape parameterization scheme for optimization purposes. This study is organized in the following manner. First, the existing shape parameterization schemes for typical bluff-body wind-sensitive structures (i.e., tall buildings and long-span bridges) are systematically reviewed. Then, the proposed machine learning-enabled parameterization scheme is presented. After that, proof-of-concept examples are conducted on tall building cross sections to demonstrate the performance of the proposed scheme. The existing gaps between proof-of-concept examples and real-world applications are also discussed subsequently. The concluding remarks and future directions are provided at the end. It is worthwhile to mention that this study, by no means, attempts to solve the problem of parameterization of bluff-body wind-sensitive structures completely. Instead, it aims to provide a proof of concept with preliminary results to guide future investigations toward a data-driven paradigm of shape parameterization for wind-sensitive structures.

between upper and lower surfaces so that the coherent vortices are disrupted (e.g., Yang et al., 2015; Tang et al., 2017). Furthermore, decks with appendages such as guide vanes for box-shape decks and baffle plates for H-shape decks have been introduced to alter the fluid-structure interaction for improved performance (e.g., Larsen et al., 2000; Sakai et al., 1993). Despite a wide range of potential schemes to modify deck shapes for improved aerodynamic performance, existing studies on aerodynamic shape optimizations usually explore the shape variation of box-shape decks using a limited number of variables. Montoya et al. (2018) used two design variables to consider the height and width variation of deck shape with respect to the reference design, which has been adopted in other studies (with slight changes in, for example, the reference design) (e.g., Tinmitondé et al., 2022; Abbas et al., 2022). This two-variable parameterization scheme is further extended to consider the possibility of venting decks by introducing an additional design variable of the slot width (see Fig. 3a) (Nieto et al., 2020). Jaouadi et al. (2020) proposed to use a more flexible approach to allow the change of location of

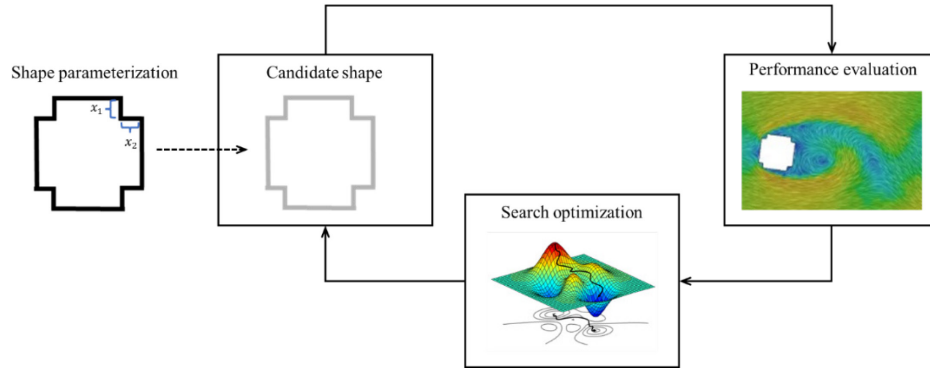


Fig 1. Schematic of a typical aerodynamic shape optimization process

2. Literature review on shape parameterization schemes for wind-sensitive structures

To provide a clear overview of the state of practice, this section presents existing parameterization schemes for aerodynamic shape optimization of two types of wind-sensitive structures: tall buildings and long-span bridges. It is noted that many studies use the same shape parameterization schemes with different research focuses (e.g., different optimization objectives or algorithms). Only the representative study is selected here as the reference for the sake of being concise. For long-span bridges, numerous efforts have been made to modify the shape of basic box-shape and H-shape bridge decks to improve the aerodynamic performance (Birhane et al., 2017), which is shown in Figure 2. For example, partial streamlining of the box-shape deck to tackle the flow separation of leading edges (e.g., Larsen, 1993) has been widely used in major long-span bridges. Similarly, fairings have also been added to the edges of the H-shape deck for aerodynamic retrofit of long-span bridges to resemble the streamlined box decks (e.g., Barelli et al., 2006). In addition, partial openings of bridge decks (slots and gratings for box-shape and H-shape decks) have been utilized to allow the mixing of airflow

between upper and lower surfaces so that the coherent vortices are disrupted (e.g., Yang et al., 2015; Tang et al., 2017). Furthermore, decks with appendages such as guide vanes for box-shape decks and baffle plates for H-shape decks have been introduced to alter the fluid-structure interaction for improved performance (e.g., Larsen et al., 2000; Sakai et al., 1993). Despite a wide range of potential schemes to modify deck shapes for improved aerodynamic performance, existing studies on aerodynamic shape optimizations usually explore the shape variation of box-shape decks using a limited number of variables. Montoya et al. (2018) used two design variables to consider the height and width variation of deck shape with respect to the reference design, which has been adopted in other studies (with slight changes in, for example, the reference design) (e.g., Tinmitondé et al., 2022; Abbas et al., 2022). This two-variable parameterization scheme is further extended to consider the possibility of venting decks by introducing an additional design variable of the slot width (see Fig. 3a) (Nieto et al., 2020). Jaouadi et al. (2020) proposed to use a more flexible approach to allow the change of location of

six corners (see Fig. 3b), which results in five design variables (considering the symmetry and fixed lane width). Compared to bridge deck shapes that always require a flat upper surface to support driving vehicles, the shape of tall buildings may exhibit higher complexities due to the flexibility in cross-section geometry as well as its potential variation along the height (Tanaka et al., 2012; Sharma et al., 2018; Jafari and Alipour, 2021). It is noted that this literature review does not consider irregular buildings (e.g., T-shape and star-shape buildings as well as buildings with novel facades) and instead only focuses on buildings with square/rectangular-like cross-sections. Starting from the basic cylinder configuration of a tall building, shape mitigation for improved aerodynamic performance is generally classified into minor and major modifications. Minor modifications (shown in Fig. 4a) usually change the geometry of cross sections through the corners (using, e.g., chamfered, slotted, rounded, recessed, and finned corners) to alter the flow separation characteristics (Mooneghi and Kargarmoakhar, 2016). It is noted that some of the minor modifications could be further extended to more complex geometries with, for example, multiple corner recessions and cuts (e.g., Wang et al., 2022; Tang et al., 2013). On the other hand, major modifications control the shape variation along the height through, for example, tapering, twisting, set-backing, and opening (Elshaer and Bitsuamlak, 2018) to

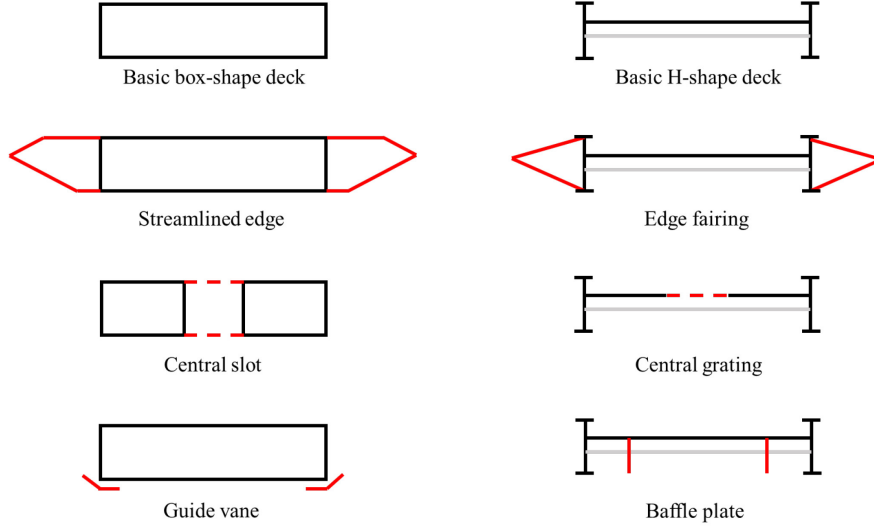


Fig 2. Representative aerodynamic modifications schemes for bridge decks

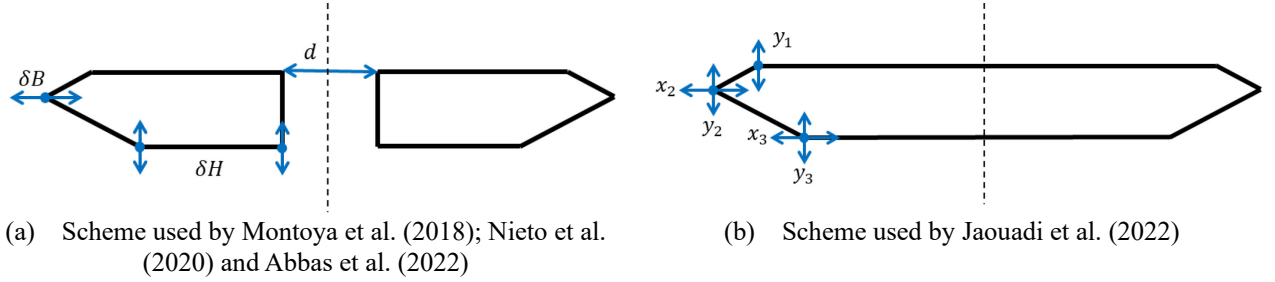
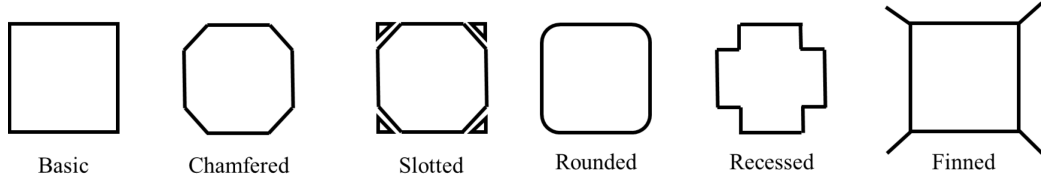


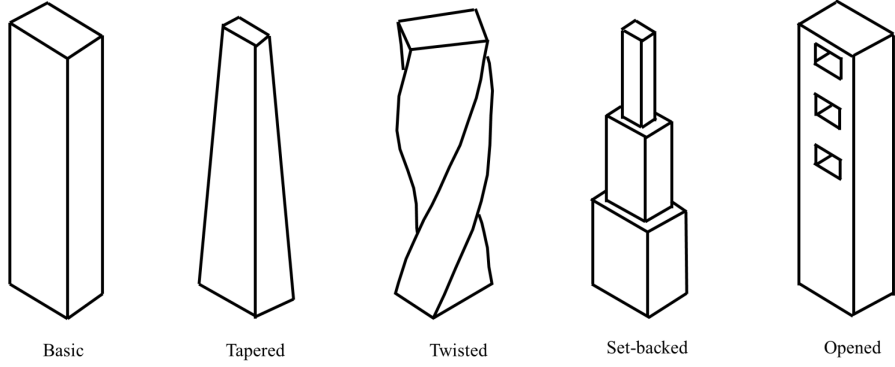
Fig 3. Existing shape parameterization schemes for bridge decks

weaken the coherent vortex shedding, which could result in more complicated 3D building designs (see Fig. 4b). Existing shape parameterization schemes for optimization purposes have involved both minor and major modifications. For minor modifications, Bernardini et al. (2014) used two design variables to control the cross-section shapes of tall buildings (see Fig. 5a), where variations of the two control points with respect to a reference shape (i.e., a square with rounded corners) are used for determining the corner shape via interpolation with a cubic spline. Elshaer et al. (2017) utilized two-variable parameterization schemes to consider potential recession and protrusion of cross section corners (see Fig. 5b). Wang et al. (2022) adopted multiple corner recession with three parameters controlling the recession ratio, recession number and recession angle (see Fig. 5c). Alkhatib et al. (2022) used two design variables (radius of rounded corner and edge angle) to characterize the shape of the cross section (see Fig. 5d). Shirzadi and Tominaga (2021) parameterized a generic trapezoid with cut corners with four design variables (see Fig. 5e). Instead of directly varying the control points in one typical shape, Nieto et al. (2022) proposed to use the weighted sum of three empirically selected baseline shapes (in terms of polar coordinate) to characterize the cross section, where the weights are taken as the design variables (see Fig. 5f). Regarding major modifications, Elshaer and Bitsuamlak (2018)

parameterized the openings in tall buildings using the aspect ratio of the openings as well as the spacing between each opening (see Fig. 5g). Efforts have also been made to integrate minor and major modifications. For example, Elshaer et al. (2016) used three design variables to consider both corner modifications (two design variables for the two control points) and twisting (one design variable for the twisting angle) (see Fig. 5h). He et al. (2022) adopted three parameters to respectively control the aspect ratio of oval cross section, the twisting angle and tapering ratio (see Fig. 5i).

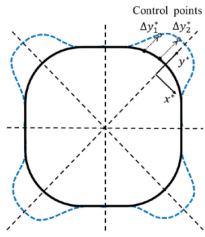


(a) Minor modifications

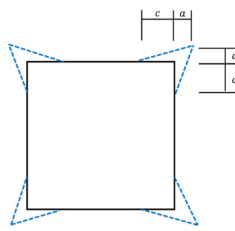


(b) Major modifications

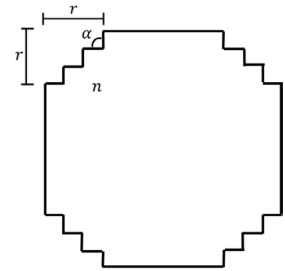
Fig. 4 Representative aerodynamic modification schemes for tall buildings



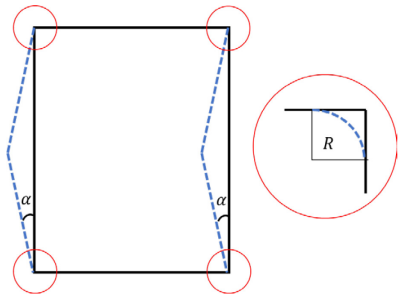
(a) Scheme used by Bernardini et al. (2014)



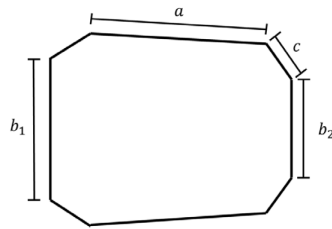
(b) Scheme used by Elshaer et al. (2017)



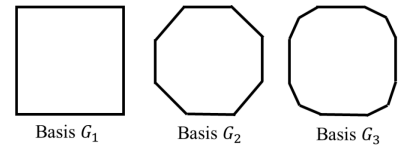
(c) Scheme used by Wang et al. (2022)



(d) Scheme used by Alkhatib et al. (2022)

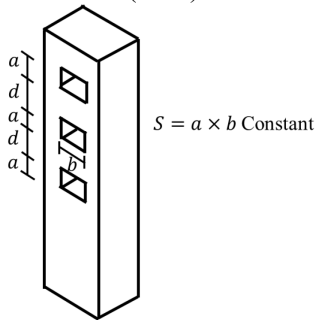


(e) Scheme used by Shirzadi and Tominaga (2021)

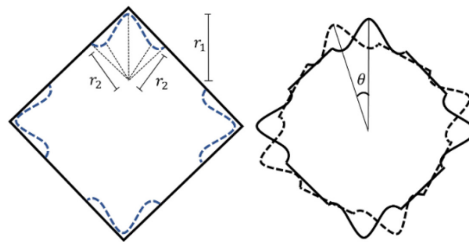


$$\text{Candidate shape } G = w_1 G_1 + w_2 G_2 + w_3 G_3$$

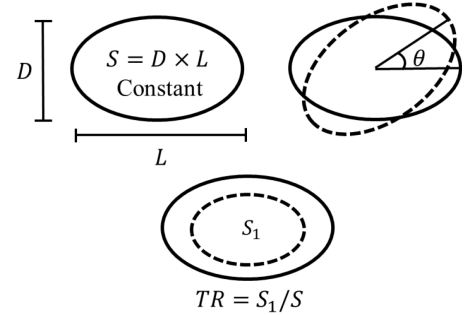
(f) Scheme used by Nieto et al. (2022)



(g) Scheme used by Elshaer and Bitsuamlak (2018)



(h) Scheme used by Elshaer et al. (2016)



(i) Scheme used by He et al. (2022)

Fig 5. Existing shape parameterization schemes for tall buildings

3. Machine learning-based shape parameterization scheme

The proposed machine learning-enabled shape parameterization scheme is presented in this section. For the sake of completeness, the basics of a standard autoencoder are first introduced to present the general idea of parameterization with latent variables. Noting the limitation of using only continuous latent variables for shape parameterization, the importance of having mixed discrete-continuous variables with hierarchy structures is highlighted. After that, the proposed machine learning-enabled shape parameterization scheme is described, which is composed of (1) the “skeleton” to learn the hierarchy structure of latent variables and (2) the “skin” to parameterize the shape based on the learned hierarchy structure.

Autoencoders have been widely used for dimension reduction in many different fields (Hinton and Salakhutdinov, 2006; Wang et al., 2016). A standard autoencoder is a deep neural network with a “bottleneck” architecture (see Fig. 6), which is composed of an encoder and a decoder. The encoder maps from the input layer (the high-dimensional raw data $\mathbf{x}$) to the bottleneck layer (the low-dimensional latent variables $\mathbf{z}$):

$$\mathbf{z} = f_E(\mathbf{x}; \boldsymbol{\theta}_E) \quad (1)$$

where $\boldsymbol{\theta}_E$ is the weight of the encoder. On the other hand, the decoder aims to reconstruct the high-dimensional data using the latent variables as input:

$$\mathbf{x}' = f_D(\mathbf{z}; \boldsymbol{\theta}_D) \quad (2)$$

where $\mathbf{x}'$ is the reconstructed data in the output layer; $\boldsymbol{\theta}_D$ is the weight of the decoder. The learning objective of a standard autoencoder is to minimize the difference between the original data and the reconstructed data by adjusting the weights of the autoencoder:

$$\min_{\boldsymbol{\theta}_E, \boldsymbol{\theta}_D} L_d(\mathbf{x}, \mathbf{x}') \quad (3)$$

where $L_d(\mathbf{x}, \mathbf{x}')$ is the data reconstruction error. After training the autoencoder, the obtained decoder could be utilized to generate high-dimensional data through low-dimensional latent variables. For the application to aerodynamic shape optimization of wind-sensitive structures, the trained encoder can be effectively used as a parameterization scheme, where the latent variables serve as the design variables to characterize the shape.

It is noted that standard autoencoders embed the high-dimensional data in continuous space (i.e., the components in $\mathbf{z}$ are all continuous variables). The continuous variable-based representation, however, may not be suitable for parameterization of the shape of wind-sensitive structures considering the distinct variations across different categories of the bluff-body shapes (e.g., tall building cross sections with recessed corners and chamfered corners). This characteristic calls for additional discrete variables in the latent space. It should also be noted that hierarchy structures may exist in the discrete and continuous variables considering the conditional relationships among the latent variables (i.e., certain variables are only active when some other variables are active). For example, the continuous variables (e.g., the depth of recessed corner and the slope of

chamfer edge) can be considered low-level variables, and they are active only when the high-level discrete variables point to their corresponding category. In addition to being more intuitive and interpretable (Ross et al., 2021; Ross and Doshi-Velez, 2021), parameterization of bluff-body shapes using latent variables with hierarchy structure will also benefit subsequent aerodynamic shape optimization in terms of (1) the flexibility to focus on certain shape types instead of exhausting the whole design space and (2) avoiding a drastic change in the shape (and hence the aerodynamic performance) to facilitate optimization convergence.

To this end, the study aims to learn a mixed discrete-continuous latent space with hierarchy structures for shape parameterization of bluff-body wind-sensitive structures. Specifically, the latent variables $\mathbf{z}$ to learn are composed of both discrete variables $\mathbf{z}_d$ and continuous variables $\mathbf{z}_c$:

$$\mathbf{z} = [\mathbf{z}_d, \mathbf{z}_c] \quad (4)$$

In addition, the hierarchy structure to learn, specifying the conditional relations between the discrete and continuous variables, can be generally represented by a mapping function $f_h(\cdot)$:

$$\mathbf{m} = f_h(\mathbf{z}_d) \quad (5)$$

where $\mathbf{m}$ is the mask (ideally $\mathbf{m}$ is a binary vector with the same dimension as the continuous variable $\mathbf{z}_c$) representing the activation status of the continuous variables based on the value of discrete variables $\mathbf{z}_d$. The active continuous variables $\mathbf{z}_c^a$ can then be calculated by “masking” the original continuous variables $\mathbf{z}_c$ with $\mathbf{m}$:

$$\mathbf{z}_c^a = \mathbf{m} \circ \mathbf{z}_c \quad (6)$$

where $\circ$ denotes the element-wise product.

The learning algorithm adopted in this study is based on recent advances in learning hierarchical representations (Ross and Doshi-Velez, 2021), which is composed of (1) the “skeleton” to learn the hierarchy structure of latent variables and (2) the “skin” to parameterize the shape based on the learned hierarchy structure. The overview of the algorithms is schematically presented in Fig. 7 while the detailed descriptions of the “skeleton” and the “skin” are provided subsequently.

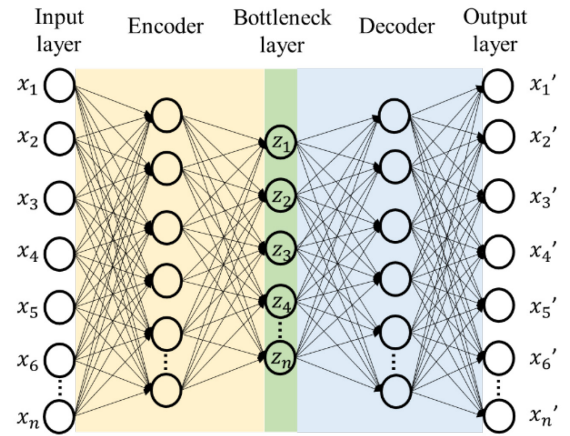


Fig. 6 Schematic of a standard autoencoder

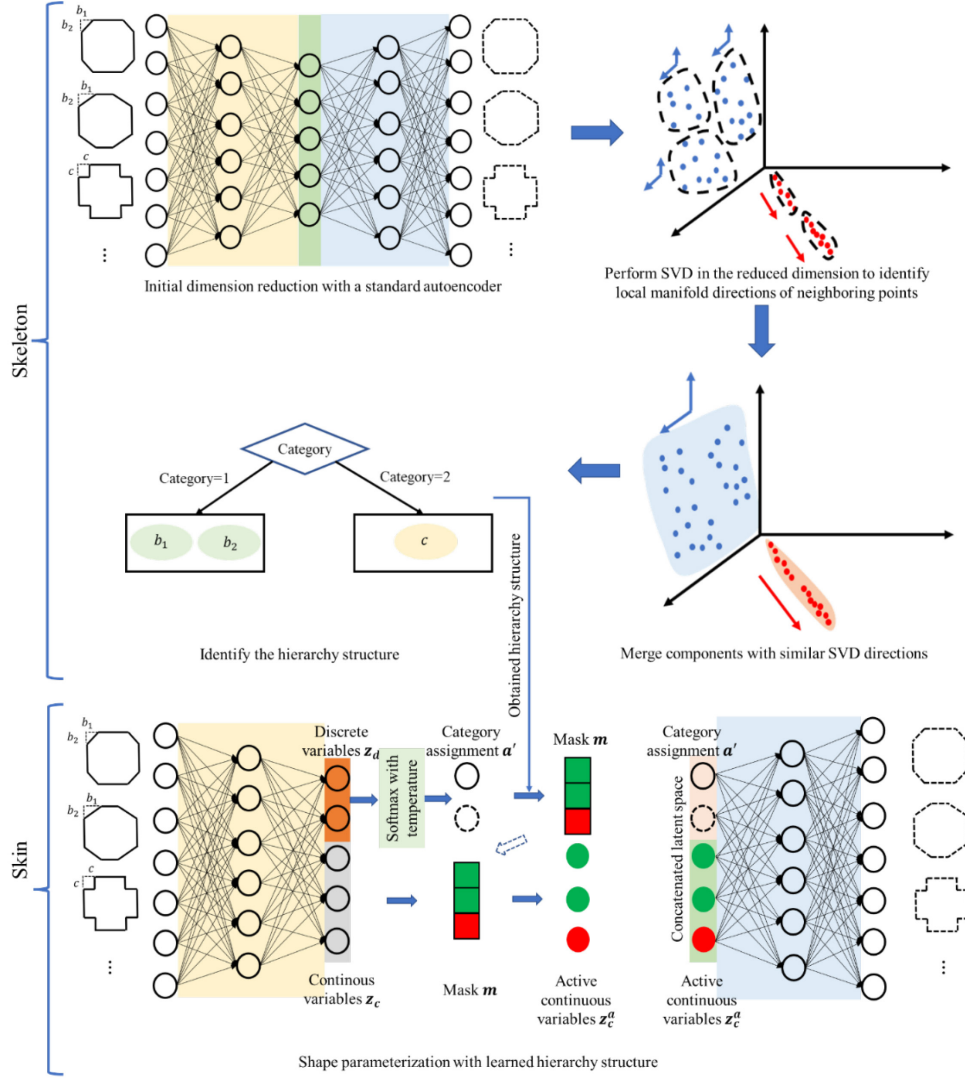


Fig. 7 Proposed machine learning-enabled shape parameterization scheme

3.1 Skeleton: Learning the hierarchy structure

First, the initial dimensional reduction is conducted on the raw data using a standard autoencoder, where the number of latent variables should exceed the intrinsic dimensionality. The intrinsic dimensionality is the smallest number of continuous variables that can faithfully reconstruct the original data, which is related to the underlying generation mechanisms of the data. For example, if the data has two categories, and each category is controlled by one continuous variable, then intrinsic dimensionality of the data is three (one for differentiating the category, similarly as a step function; two for controlling the variation within each category). The intrinsic dimensionality could be determined by trial and error: if the number of latent variables is smaller than the intrinsic dimensionality, the reconstruction error will be very large. This initial dimension reduction is helpful to enhance the learning efficiency considering that the subsequent computation will be conducted in the latent space with a lower dimension. In the reduced dimension, singular value decomposition (SVD) is performed for each point and its

neighbors to obtain the local manifold directions, where a ball tree (Omohundro, 1989) is used to construct a neighborhood graph to identify the neighboring points quickly. Points with similar local manifold directions are then merged into components, which is realized by first merging the neighboring points and then combining similar components over longer distances. With the merged components, the hierarchy structure can be inferred based on which components enclose others. In addition, the assignments of each data point to the corresponding category a can be determined accordingly. The structured description of the “skeleton” model is shown in Algorithm 1 while the details can be found in (Ross and Doshi-Velez, 2021).

Algorithm 1. Skeleton: Learning the hierarchy structure

Use a standard autoencoder to encode the data in a lower dimension.

Construct a ball tree-based neighborhood graph for efficient identification of neighbors.

Perform SVD on each point and its neighbors to identify local manifold directions.

Merge points into components:

Successively merge neighboring points with similar local

manifold directions.

Combine similar components over longer distances.

Identify the hierarchy structure based on the merged components.

Return the hierarchy structure and assignment of each point.

3.2 Skin: Parameterization using the learned hierarchy structure

The abovementioned “skeleton” model can output the hierarchy structure and categorical assignment for each data point, which allows the “skin” model for subsequent parameterization. A new hierarchical encoder is first built $\mathbf{z} = f_{HE}(\mathbf{x}; \boldsymbol{\theta}_{HE})$, where $\boldsymbol{\theta}_{HE}$ is the weight of the hierarchical encoder and the size of latent dimension $\mathbf{z}$ equals to the number of continuous variables plus the number of categorical options in the learned hierarchy structure. The latent variables $\mathbf{z}$ can be then partitioned into two components, i.e., $\mathbf{z}_d$ for the discrete dimension and $\mathbf{z}_c$ for the continuous dimension. The values corresponding to discrete dimensions $\mathbf{z}_d$ are first passed through a softmax function with temperature (temperature τ is a hyperparameter) to convert to the assignment probability for each categorical option, denoted as $\mathbf{a}' = \text{ST}(\mathbf{z}_d)$. The obtained assignment $\mathbf{a}'$ can be used, along with the learned hierarchy structure in the “skeleton” model, to obtain the mask to determine the activation status of the continuous variables $\mathbf{m} = g_h(\mathbf{a}') = g_h[\text{ST}(\mathbf{z}_d)]$ [essentially equivalent to $\mathbf{m} = f_h(\mathbf{z}_d)$ in Eq. (5)]. With the mask $\mathbf{m}$, active continuous variables $\mathbf{z}_c^a$ can be determined as $\mathbf{z}_c^a = \mathbf{m} \circ \mathbf{z}_c$ as shown in Eq. (6). The hierarchical decoder reconstructs the data $\mathbf{x}'$ from the concatenated latent space (including the predicted assignment $\mathbf{a}'$ and the active continuous variables $\mathbf{z}_c^a$), $\mathbf{x}' = f_{HD}(\mathbf{a}', \mathbf{z}_c^a; \boldsymbol{\theta}_{HD})$, where $\boldsymbol{\theta}_{HD}$ is the weight of hierarchical decoder. The weights of the hierarchical encoder and decoder can be learned by:

$$\min_{\boldsymbol{\theta}_{HE}, \boldsymbol{\theta}_{HD}} [L_d(\mathbf{x}, \mathbf{x}') + \lambda_a L_a(\mathbf{a}, \mathbf{a}') + \lambda_r L_r(\boldsymbol{\theta}_{HE}, \boldsymbol{\theta}_{HD})] \quad (7)$$

where, in addition to the data reconstruction error $L_d(\mathbf{x}, \mathbf{x}')$, the assignment error $L_a(\mathbf{a}, \mathbf{a}')$ (representing the discrepancies between the predicted assignment by hierarchical decoder $\mathbf{a}'$ and “ground-truth” assignment given by the “skeleton” model $\mathbf{a}$) also needs to be minimized; λ_a is the weight for the assignment error; the regularization term $L_r(\boldsymbol{\theta}_{HE}, \boldsymbol{\theta}_{HD})$ with the weight of λ_r is used here to remove correlation of the continuous variables (Kim and Mnih, 2018). The trained hierarchical decoder can then be effectively used as the shape parameterization scheme. The structured description of the “skin” model is shown in Algorithm 2, while the details can be found in (Ross and Doshi-Velez, 2021).

Algorithm 2. Skin: Parameterization using the learned hierarchy structure

Obtain the latent variables from the hierarchical encoder $\mathbf{z}_d, \mathbf{z}_c = f_{HE}(\mathbf{x}; \boldsymbol{\theta}_{HE})$.

Calculate the assignment probability $\mathbf{a}'$ from the $\mathbf{z}_d$ using a softmax function with temperature: $\mathbf{a}' = \text{ST}(\mathbf{z}_d)$

Get the mask $\mathbf{m}$ from the predicted assignment $\mathbf{a}'$ using the learned hierarchy structure $\mathbf{m} = g_h(\mathbf{a}')$.

Use the mask $\mathbf{m}$ to determine the active continuous variables

$$\mathbf{z}_c^a = \mathbf{m} \circ \mathbf{z}_c.$$

Reconstruct the data $\mathbf{x}'$ from concatenated variables using the hierarchical decoder $\mathbf{x}' = f_{HD}(\mathbf{a}', \mathbf{z}_c^a; \boldsymbol{\theta}_{HD})$.

Compute the loss as $L = L_d(\mathbf{x}, \mathbf{x}') + \lambda_a L_a(\mathbf{a}, \mathbf{a}') +$

$$\lambda_r L_r(\boldsymbol{\theta}_{HE}, \boldsymbol{\theta}_{HD}).$$

Conduct gradient descent to update the hierarchy autoencoder weights $\boldsymbol{\theta}_{HE}$ and $\boldsymbol{\theta}_{HD}$ by minimizing the loss L .

Repeat until convergence.

4. Proof-of-concept examples

Two numerical examples, as a proof of concept, are presented in this section to demonstrate the performance of the proposed machine learning-enabled shape parameterization scheme. The first case deals with simple geometries represented by the edge coordinates using fully connected neural network-based autoencoders, which can help to determine the critical hyperparameters in the learning algorithm. The second case extends to more general applications with pixel-based representation using convolutional neural network-based autoencoders, which is promising for future applications with real-world raw image data. It should be emphasized that these proof-of-concept examples, by no means, attempt to solve the shape parameterization problem completely. Instead, they aim to serve as pilot studies to investigate the feasibility of data-driven shape parameterization schemes using machine learning tools. In addition, since the ground-truth shape parameterization is predetermined in the synthetic dataset, there is no need in proof-of-concept examples to show a whole optimization process using the learned shape parameterization scheme. Hence, the following examples only focus on demonstrating the performance of the proposed scheme to accurately learn the predetermined shape parameterization in the synthetic dataset.

4.1 Case 1: Coordinate-based shape representation using fully connected neural networks

A synthetic dataset for shapes with simple geometries is built in this example, which includes two types of aerodynamic shape modifications for tall buildings (i.e., recessed and chamfered corners). As shown in Fig. 8, the double-recessed corners have two design variables l_1 and l_2 for the recession depths (both are randomly generated following a uniform distribution in $[0, 0.25]$), while the chamfered corners have only one design variable α for the corner slope (α is randomly generated following a uniform distribution in $[0^\circ, 45^\circ]$). The shapes are symmetric with respect to both vertical and horizontal axis, and hence they can be conveniently represented by the coordinates of one corner. Specifically, the vertical coordinates of 64 points with uniform horizontal distance are used to represent the shape, and fully connected neural networks are used for the autoencoders (the network architecture is shown in Table 1). The size of the dataset is 10,000 (5,000 for each category). The critical hyperparameters (obtained by trial and error) are shown in Table 1. Other parameters that are not shown here adopt the default values specified in (Ross and Doshi-Velez, 2021). It is noted that the network architecture of the

hierarchical autoencoder in the “skeleton” is set as the same as that of the standard autoencoder for initial dimension reduction in the “skin” except for the additional “masking” operations. Adam optimizer is used for gradient descent of both the standard and hierarchical autoencoder (with a batch size of 256 and an epoch number of 50), where a learning rate starts at 0.001 and decays by 1/10 halfway and three-quarters of the way through training. Regarding the hyperparameters used by only the hierarchical autoencoder, the softmax temperature τ , the weight of assignment loss λ_a , and the weight of regularization loss λ_r are 1, 100, and 1, respectively.

After training, the “skeleton” model can successfully merge points with similar local manifold directions and identify the underlying hierarchy structure. As shown in Fig. 9, the obtained hierarchy structure has one discrete variable with two categorical options. The first category can be characterized by one continuous variable, while the second one needs two continuous variables. Note that only three of the five latent dimensions from initial dimensional reduction are plotted in Fig. 9 for the sake of clear illustration. With the obtained hierarchy structure, the hierarchical autoencoder is trained using the “skin” model. The trained hierarchical decoder can then be effectively utilized as the shape parameterization scheme. Specifically, the category assignment $\mathbf{a}'$ is essentially a two-dimensional vector with one-hot embedding (i.e., $\mathbf{a}' = [1, 0]$ denotes category 1 and $\mathbf{a}' = [0, 1]$ denotes category 2). The mask $\mathbf{m}$ can then be determined according to the learned hierarchy structure ($\mathbf{m} = [1, 0, 0]$ for $\mathbf{a}' = [1, 0]$ while $\mathbf{m} = [0, 1, 1]$ for $\mathbf{a}' = [0, 1]$) so that the active status of the three continuous variables $\mathbf{z}_c = [z_{c_1}, z_{c_2}, z_{c_3}]$ can be obtained ($\mathbf{z}_c^a = [z_{c_1}, 0, 0]$ for category 1 and $\mathbf{z}_c^a = [0, z_{c_2}, z_{c_3}]$ for category 2). Based on the projection in the latent space of the training data, it is found that the continuous variable z_{c_1} for category 1 lies in $[-22, 3]$ while the continuous variables z_{c_2} and z_{c_3} for category 2 are in $[-1, 38]$ and $[0, 17]$ respectively.

To show the performance of the shape parameterization, the output of the hierarchical decoder with varying input $[\mathbf{a}', \mathbf{z}_c^a]$ is presented in Fig. 10. Fig. 10(a)-(c) indicate that the learned continuous variable z_{c_1} can successfully capture the variation in the slope of the chamfered corners. In addition to the interpolation results shown in Fig. 10(a)-(c), it is also interesting to observe from Fig. 10(d) that the obtained parameterization scheme can even extrapolate beyond the limit of training data (although the result is not as good as that of interpolation due to the lack of data in the extrapolating region). This extrapolation ability indicates the trained decoder can “understand” the underlying pattern instead of “memorizing” it, which also shows the potential to expand the search space for aerodynamic shape optimization. Similarly, Fig. 10(e)-(j) and 10(i)-(k) show that the learned continuous variables z_{c_2} and z_{c_3} are responsible for controlling the variations in the two recession depths. Fig. 10(h) and (l) also demonstrate the extrapolation ability of the learned decoder in these two dimensions. Furthermore, it is tempting to test if novel designs can be generated by the trained decoder. Intuitively,

the assignment variables $\mathbf{a}'$ can manually be set as $[0.5, 0.5]$ to consider a hybrid of category 1 and 2, and accordingly all the three continuous variables in $\mathbf{z}_c^a$ can be active. Fig. 10(m)-(t) shows the decoder output with eight arbitrary sets of design variables. The obtained shapes (although with zigzag) exhibit the features from both recessed and chamfered corners, which is unseen from the training data. These interesting results demonstrate the potential to manipulate the latent variables to generate novel designs (Li et al., 2020) for aerodynamic shape optimization, which is a promising direction for future work. In addition, it is not essential that the reconstructed data should be exactly the same as the training data or has a perfect quality for the case of extrapolation, considering that the reconstructed data will need to be postprocessed before passing it to the optimization process. For example, a filter based on empirical knowledge can be designed to remove the small errors (e.g., the small bumps in a straight line).

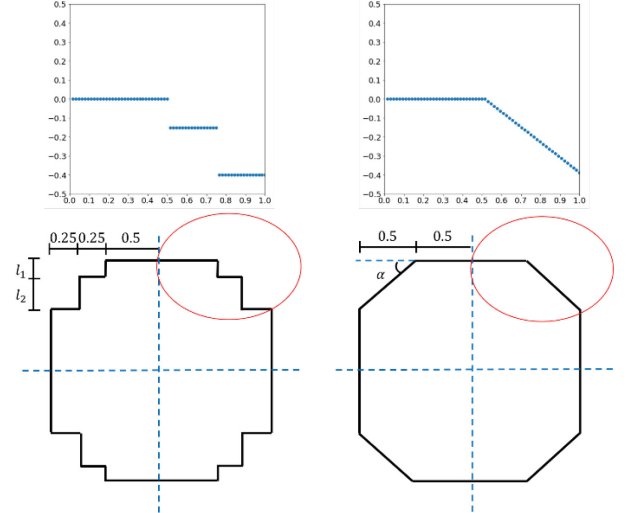
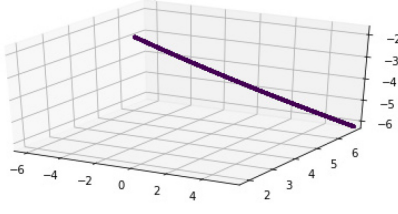


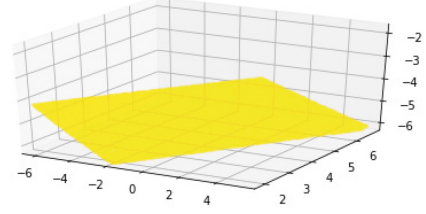
Fig. 8 Synthetic dataset for Case 1 (The two axes in the upper two figures are coordinates for the two-dimensional shape)

Table 1. Hyperparameters for Case 1

Hyperparameters	Values
Number of latent variables in initial dimension reduction	5
Number of hidden layers in the standard/hierarchical autoencoder	2
Number of neurons in standard/hierarchical autoencoder	200, 200
Activation function in the standard/hierarchical autoencoder	Softplus
Learning rate for the standard/hierarchical autoencoder	0.001 start, 0.0001 halfway, 0.00001 3/4 way
Batch size for the standard/hierarchical autoencoder	256
Number of epochs for the standard/hierarchical autoencoder	50
Softmax temperature τ for the hierarchical autoencoder	1
Weight of assignment loss λ_a for the hierarchical autoencoder	100
Weight of regularization loss λ_r for the hierarchical autoencoder	1



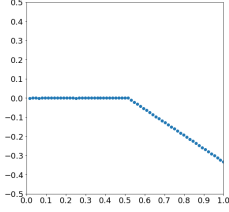
(a) Category 1 (1-dimensional)



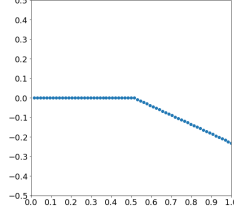
(b) Category 2 (2-dimensional)

Fig. 9 Identified hierarchy structure of Case 1 using the “skeleton” model (projection of data in the latent space)

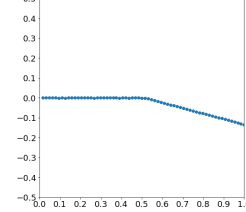
4.2 Case 2: Pixel-based shape representation using



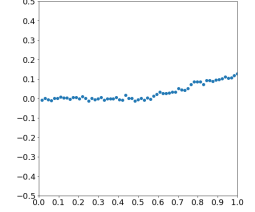
(a) Decoder output with $[a', z_c^a] = [1, 0, -15, 0, 0]$



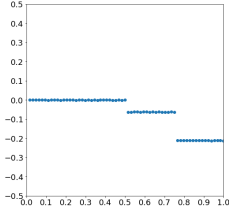
(b) Decoder output with $[a', z_c^a] = [1, 0, -10, 0, 0]$



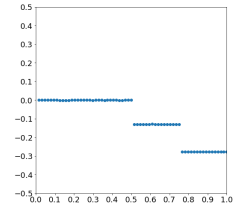
(c) Decoder output with $[a', z_c^a] = [1, 0, -5, 0, 0]$



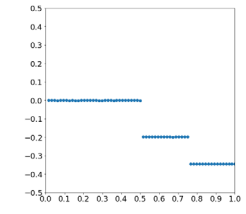
(d) Decoder output with $[a', z_c^a] = [1, 0, 10, 0, 0]$ (extrapolation)



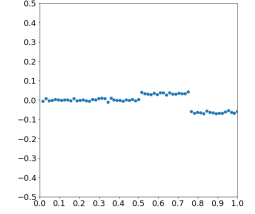
(e) Decoder output with $[a', z_c^a] = [0, 1, 0, 10, 10]$



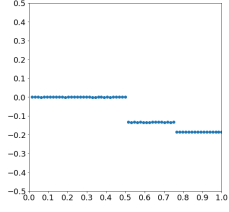
(f) Decoder output with $[a', z_c^a] = [0, 1, 0, 20, 10]$



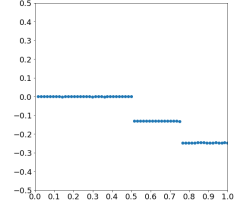
(g) Decoder output with $[a', z_c^a] = [0, 1, 0, 30, 10]$



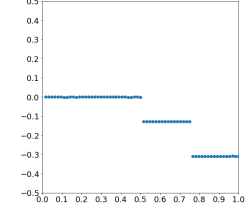
(h) Decoder output with $[a', z_c^a] = [0, 1, 0, -10, 10]$ (extrapolation)



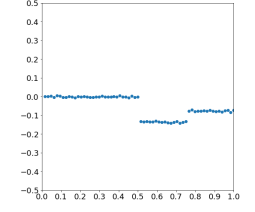
(i) Decoder output with $[a', z_c^a] = [0, 1, 0, 20, 4]$



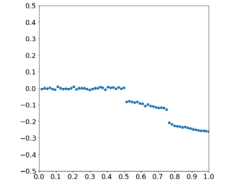
(j) Decoder output with $[a', z_c^a] = [0, 1, 0, 20, 8]$



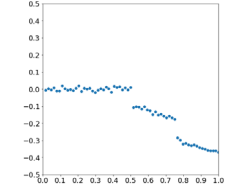
(k) Decoder output with $[a', z_c^a] = [0, 1, 0, 20, 12]$



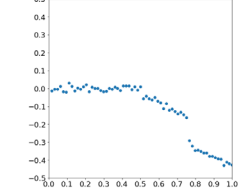
(l) Decoder output with $[a', z_c^a] = [0, 1, 0, 20, -4]$ (extrapolation)



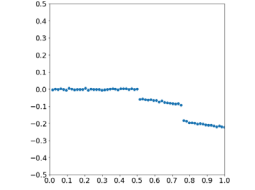
(m) Decoder output with $[a', z_c^a] = [0.5, 0.5, -5, 10, 5]$ (novel design)



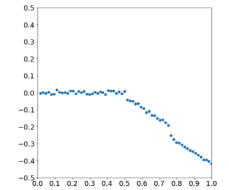
(n) Decoder output with $[a', z_c^a] = [0.5, 0.5, -8, 12, 6]$ (novel design)



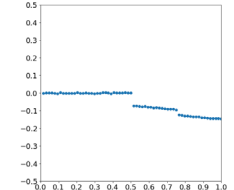
(o) Decoder output with $[a', z_c^a] = [0.5, 0.5, -10, 6, 8]$ (novel design)



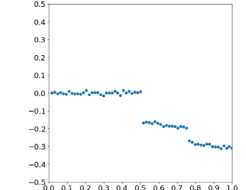
(p) Decoder output with $[a', z_c^a] = [0.5, 0.5, -3, 8, 6]$ (novel design)



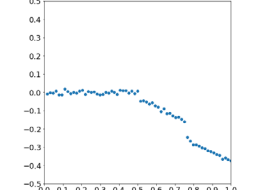
(q) Decoder output with $[a', z_c^a] = [0.5, 0.5, -15, 4, 3]$ (novel design)



(r) Decoder output with $[a', z_c^a] = [0.5, 0.5, -2, 10, 2]$ (novel design)



(s) Decoder output with $[a', z_c^a] = [0.5, 0.5, -5, 20, 4]$ (novel design)



(t) Decoder output with $[a', z_c^a] = [0.5, 0.5, -11, 5, 5]$ (novel design)

Fig. 10 Shape parameterization result of Case 1 using the “skin” model

convolutional neural networks

Instead of using the edge coordinates, Case 2 adopts the pixel-based representation to show the promising potential for future applications with real-world raw image data. Specifically, Case 2 considers two categories of shapes, as shown in Fig. 11. In the first category, the shape considers a special type of chamfered corner, where the square cross-section is cut with an arc curve instead of a straight line (to test the algorithm's capability to tackle curves). The diameter of the cutting circle is equal to the dimension of the square. The diagonal translation of the cutting circle (i.e., $\sqrt{2}l$) is the continuous design variable, while the original square shape is fixed. The shape in the second category has recessed corners, where the horizontal and vertical translation of the cutting square are the two continuous design variables (i.e., d_1 and d_2). Specifically, 64×64 pixel-based representations are used in Case 2 (down sampled from 256×256 pixel space to make images "blurry", considering that real-world image data may need to be preprocessed to have a lower resolution for efficient training), where interpolations are employed in the pixel value for continuous translation of the shapes in discrete pixels. The limit of design variable l is $[4, 16]$ pixel while the limit of design variable d_1 and d_2 is $[8, 24]$ pixel. The size of the dataset is 100,000 (50,000 for each category). Convolutional neural networks (CNN) are used in the autoencoders due to the high ability to capture the spatial features in image data. The structure of CNN is schematically shown in the Appendix. The generated images from the decoder can then be converted to CAD files for manufacturing models of wind tunnel tests or geometric drawings for CFD simulation. The hyperparameters are shown the Table 2.

After training, the "skeleton" model can successfully merge points with similar local manifold directions and identify the underlying hierarchy structure with two categorical options. As shown in Fig. 12, the first category can be characterized by one continuous variable, while the second one needs two continuous variables. With the obtained hierarchy structure, the hierarchical autoencoder is trained using the "skin" model. Like Case 1, the category assignment $\mathbf{a}' = [1, 0]$ denotes category 1 and $\mathbf{a}' = [0, 1]$ denotes category 2. The mask $\mathbf{m}$ can then be determined as $\mathbf{m} = [1, 0, 0]$ for $\mathbf{a}' = [1, 0]$ and $\mathbf{m} = [0, 1, 1]$ for $\mathbf{a}' = [0, 1]$. Accordingly, the active status of the three continuous variables can be obtained as $\mathbf{z}_c^a = [z_{c_1}, 0, 0]$ for category 1 and $\mathbf{z}_c^a = [0, z_{c_2}, z_{c_3}]$ for category 2. Based on the projection in the latent space of the training data, it is found that the continuous variable z_{c_1} for category 1 lies in $[-115, -5]$ while the continuous variables z_{c_2} and z_{c_3} for category 2 are in $[-65, -3]$ and $[-68, -1]$ respectively.

To show the performance of the shape parameterization, the output of the hierarchical decoder with varying input $[\mathbf{a}', \mathbf{z}_c^a]$ is presented in Fig. 13. Fig. 13(a)-(c) indicate that the learned continuous variable z_{c_1} can successfully capture the diagonal translation of the cutting circle. Similarly, Fig. 13(e)-(j) and 13(i)-(k) show that the learned continuous variables z_{c_2} and z_{c_3} are responsible for

controlling the horizontal and vertical translations of the cutting square. Note that they are not perfectly disentangled due to numerical errors [e.g., there are still slight horizontal translations in Fig. 13(i)-(k)]. It should also be noted that Fig. 13(d), (h) and (l) show extrapolation abilities to extend the original data distributions. To test if novel designs can be generated by the trained decoder, eight arbitrary sets of the latent variables are input to the decoder, and the decoder output is shown in Fig. 13(m)-(t). The obtained shapes exhibit features of the rounded cutting (although some of them are not as clear as others), which can be considered as a combination of square and circle. This demonstrates the potential to generate novel designs by manipulating the latent variables (Li et al., 2020), which, however, needs further investigations to make it more systematic and interpretable.

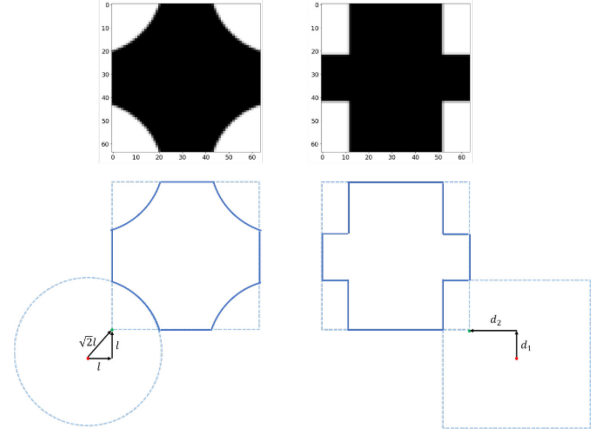


Fig. 11 Synthetic dataset for Case 2

Table 2. Hyperparameters for Case 2

Hyperparameters	Values
Number of latent variables in initial dimension reduction	5
Architecture of convolutional layers in standard/hierarchical autoencoder	4 convolutional layers (each with 32 channels, 4×4 kernels, and a stride of 2)
Architecture of fully connect layers in standard/hierarchical autoencoder	2 fully connected layers (each of 256 neurons)
Activation function in standard/hierarchical autoencoder	ReLU
Learning rate for the standard/hierarchical autoencoder	5×10^{-4} start, 2.5×10^{-4} halfway, 1.25×10^{-4} 3/4 way
Batch size for the standard/hierarchical autoencoder	256
Number of epochs for the standard/hierarchical autoencoder	10
Softmax temperature τ for the hierarchical autoencoder	1
Weight of assignment loss λ_a for the hierarchical autoencoder	1000
Weight of regularization loss λ_r for the hierarchical autoencoder	10

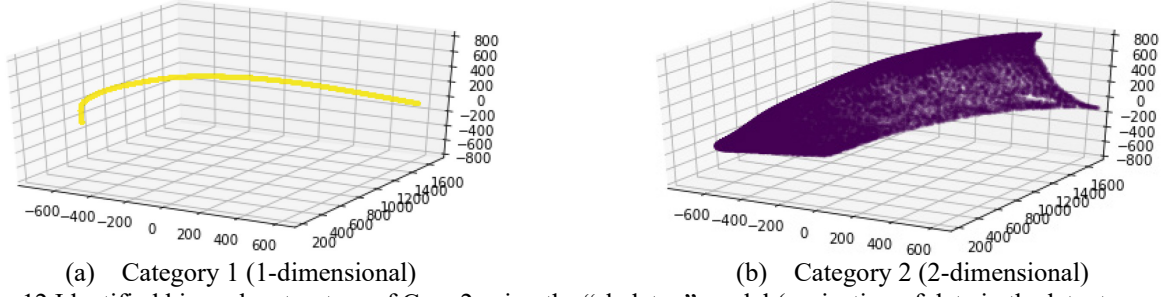


Fig. 12 Identified hierarchy structure of Case 2 using the “skeleton” model (projection of data in the latent space)

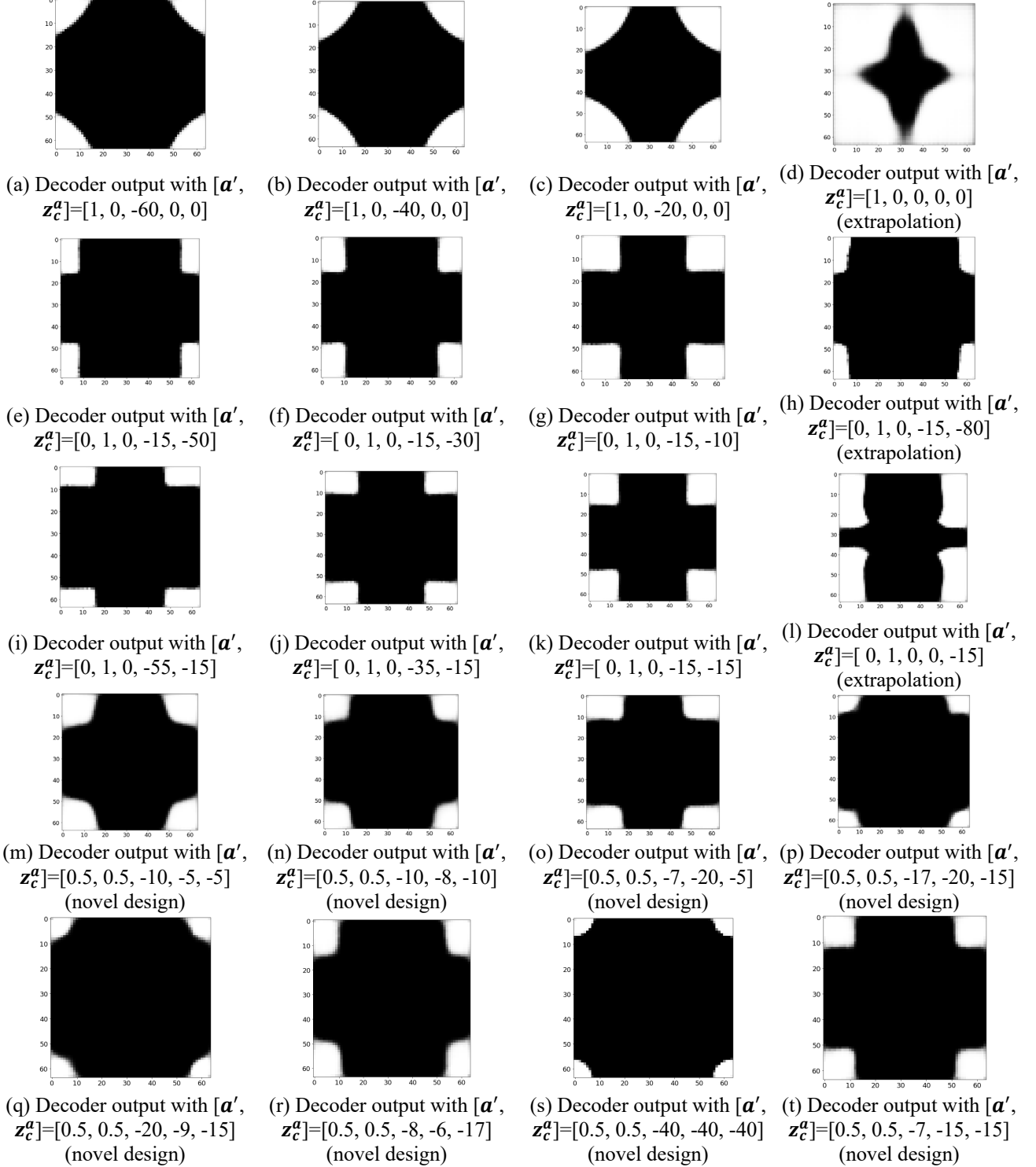


Fig. 13 Shape parameterization result of Case 2 using the “skin” model

5. Moving from proof-of-concept examples to real-world applications

Although previous discussions successfully demonstrate the promising features of the proposed scheme in parameterizing simple shapes for tall building cross sections, there still exist gaps between the proof-of-concept examples and real-world applications. In this section, some considerations on constructing real-world datasets and tuning the learning algorithms for complex scenarios are presented to guide future investigations.

Considering the data-driven nature of the proposed scheme, construction of a suitable dataset is the first and perhaps the most important step in the machine learning-based shape parameterization. There are several potential directions to construct the dataset for real-world applications. For example, Google Earth can be used to directly extract raw images of tall buildings (Google Earth). There are also publicly available datasets containing 3D coordinates of buildings in some cities (e.g., RealCity3D). It should be noted that using the dataset covering only existing structures may fail to generate real novel designs for the new structures to build, and hence it is desirable to also include conceptual designs from architects and engineers. In fact, the potential design space of building shapes is usually determined by practical constraints, which can be explicit (e.g., floor area) or implicit (e.g., aesthetic considerations). One possible approach to generate building shapes that satisfy the practical constraints is to first use computer to randomly generate the building shapes that meet the explicit constraints, and then rely on architects and engineers to filter out the undesirable ones (i.e., to meet the implicit constraints). This human-machine interaction also resembles the concept of generative adversarial networks composed of a generator and a discriminator (Goodfellow et al., 2014), except that in this case the generator is based on computer programming while the discriminator is based on human judgement (Fujii et al., 2020). Furthermore, it should be noted that data from different sources may have different resolutions and/or formats, which need to be carefully processed for uniformity.

Once the dataset of aerodynamic shapes is built based on the abovementioned considerations, the proposed machine learning scheme in this study can be straightforwardly used to automatically categorize these shapes and parameterize them with a limited number of design variables. The obtained parameterization can be subsequently used for aerodynamic shape optimization. In addition, the novel designs generated outside the training data, as demonstrated previously in the proof-of-concept examples, can lead to discovery of promising aerodynamic shapes for wind mitigation.

The increased complexity of real-world dataset also requires tuning the learning algorithms to accommodate the attending challenges. One potential challenge may come from the fact the constructed dataset may have uneven distributions across different categories (note that the two proof-of-concept examples simply assume same amount of data instances for each category). The data from underrepresented shape categories may be considered as

noises and hence could be neglected in the machine learning schemes, which calls for further efforts to make the learning algorithms robust for uneven data distributions. Furthermore, the hierarchy structure for real-world shape datasets may be deep and complex (e.g., subcategories existing within one category), which can be challenging for the learning algorithm. In addition, there may be more than one plausible hierarchy structure that fits the complex dataset. Hence, it is necessary to develop a mechanism to ensure the identified hierarchy structure is most interpretable to human users (Ross and Doshi-Velez, 2021).

5. Concluding remarks and future directions

To move beyond current empirical approach to a data-driven paradigm, a machine learning-enabled parameterization scheme is developed in this study for aerodynamic shape optimization of wind-sensitive structures. Specifically, autoencoders are used to encode the high-dimensional shape data in latent space with mixed discrete-continuous variables. In addition, the hierarchy structure in the latent space is identified to obtain the conditional relationship between the discrete and continuous variables. Proof-of-concept examples on shape parameterization of tall buildings are conducted to demonstrate the performance, where the proposed scheme works well for both coordinate-based shape representation using fully connected neural networks and pixel-based shape representation using convolutional neural networks. To extend from the current proof-of-concept study to practical implementations, future work is needed in the following directions. Firstly, a comprehensive dataset reflecting real-world complexities of possible shapes needs to be constructed for wind-sensitive structures. Secondly, the current parameterization algorithms may need further improvement on the robustness under noisy and/or insufficient real-world data. Thirdly, the potential of manipulating latent variables to generate novel designs should be investigated.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under Grants No. 2028762 & 2028647. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

Appendix: Structure of CNN-based autoencoder in Case 2

For the proof-of-concept example in Case 2, the encoder consists of 4 convolutional layers (each with 32 channels, 4×4 kernels, and a stride of 2), which are followed by 2 fully connected layers, each of 256 neurons. ReLU activation functions are used in these layers. The decoder architecture is simply the transpose of the encoder, and the final output is processed by a sigmoid layer to restrict the value in $[0, 1]$. The structure of the CNN-based autoencoder

is schematically shown in Fig. A1.

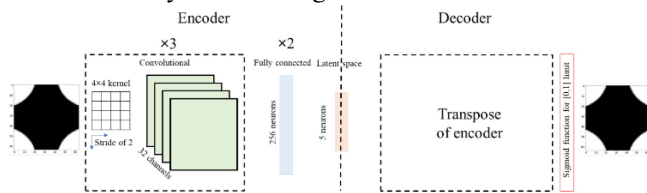


Fig. A1 Structure of CNN-based autoencoder

References

- Abbas, T., Kavrakov, I., Morgenthal, G. and Lahmer, T., (2022). Framework for a simulation-based aerodynamic shape optimization of bridge decks for different limit state phenomena. *arXiv preprint arXiv:2203.14414*.
- Alkhatib, F., Kasim, N., Goh, W.I., Shafiq, N., Amran, M., Kotov, E.V. and Albaom, M.A., (2022). Computational aerodynamic optimization of wind-sensitive irregular tall buildings. *Buildings*, **12**(7), 939. <https://doi.org/10.3390/buildings12070939>
- Allen, C.B., Poole, D.J. and Rendall, T., (2018). Wing aerodynamic optimization using efficient mathematically-extracted modal design variables. *Optimization and Engineering*, **19**(2), 453-477. <https://doi.org/10.1007/s11081-018-9376-7>
- Barelli, M., White, J. and Billington, D.P., (2006). History and aesthetics of the Bronx-Whitestone Bridge. *Journal of Bridge Engineering*, **11**(2), 230-240. [https://doi.org/10.1061/\(ASCE\)1084-0702\(2006\)11:2\(230](https://doi.org/10.1061/(ASCE)1084-0702(2006)11:2(230)
- Bernardini, E., Spence, S.M., Wei, D. and Kareem, A., (2015). Aerodynamic shape optimization of civil structures: A CFD-enabled Kriging-based approach. *Journal of Wind Engineering and Industrial Aerodynamics*, **144**, 154-164. <https://doi.org/10.1016/j.jweia.2015.03.011>
- Birhane, T.H., Bitsuamlak, G.T. and King, J.P., (2017). A Computational Framework for the Aerodynamic Shape Optimization of Long-Span Bridge Decks. In *Structures Congress 2017*, 223-239.
- Chen, W., Fuge, M. and Chazan, J., (2017). Design manifolds capture the intrinsic complexity and dimension of design spaces. *Journal of Mechanical Design*, **139**(5). <https://doi.org/10.1115/1.4036134>
- Chen, W., Chiu, K. and Fuge, M.D., (2020). Airfoil design parameterization and optimization using bézier generative adversarial networks. *AIAA journal*, **58**(11), 4723-4735. <https://doi.org/10.2514/1.J059317>
- Cinquegrana, D. and Iuliano, E., (2017). Efficient global optimization of a transonic wing with geometric data reduction. In *35th AIAA Applied Aerodynamics Conference*.
- Davenport, A.G., (1971). The response of six building shapes to turbulent wind. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, **269**(1199), 385-394. <https://doi.org/10.1098/rsta.1971.0039>
- Ding, F. and Kareem, A., (2018). A multi-fidelity shape optimization via surrogate modeling for civil structures. *Journal of Wind Engineering and Industrial Aerodynamics*, **178**, 49-56. <https://doi.org/10.1016/j.jweia.2018.04.022>
- Dulikravich, G.S., (1992). Aerodynamic shape design and optimization-status and trends. *Journal of aircraft*, **29**(6), 1020-1026. <https://doi.org/10.2514/3.46279>
- Elshaer, A., Bitsuamlak, G. and El Damatty, A., (2016), June. Aerodynamic shape optimization of tall buildings using twisting and corner modifications. In *8th International Colloquium on Bluff Body Aerodynamics and Applications*.
- Elshaer, A., Bitsuamlak, G. and El Damatty, A., (2017). Enhancing wind performance of tall buildings using corner aerodynamic optimization. *Engineering Structures*, **136**, 133-148. <https://doi.org/10.1016/j.engstruct.2017.01.019>
- Elshaer, A. and Bitsuamlak, G., (2018). Multiobjective aerodynamic optimization of tall building openings for wind-induced load reduction. *Journal of Structural Engineering*, **144**(10), 04018198. [https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0002199](https://doi.org/10.1061/(ASCE)ST.1943-541X.0002199)
- Fujii, K., Saito, Y., Takamichi, S., Baba, Y. and Saruwatari, H., (2020), May. HumanGAN: generative adversarial network with human-based discriminator and its evaluation in speech perception modeling. In *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE.
- Google Earth, <https://earth.google.com/>, accessed on July 24, 2023.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Hallaji, E., Farajzadeh-Zanjani, M., Razavi-Far, R., Palade, V. and Saif, M., (2021). Constrained generative adversarial learning for dimensionality reduction. *IEEE Transactions on Knowledge and Data Engineering*, doi: 10.1109/TKDE.2021.3126642.
- He, Z., Liang, T., Lai, X., Gao, M., Tu, X. and Lu, Y., (2022). Vibration acceleration-integrated parameterized aerodynamic shape optimization of super high-rise buildings with spiral configurations. *Structural and Multidisciplinary Optimization*, **65**(10), 1-21. <https://doi.org/10.1007/s00158-022-03387-7>
- Hinton, G.E. and Salakhutdinov, R.R., (2006). Reducing the dimensionality of data with neural networks. *Science*, **313**(5786), 504-507. DOI: 10.1126/science.1127647
- Jafari, M. and Alipour, A., (2021). Review of approaches, opportunities, and future directions for improving aerodynamics of tall buildings with smart facades. *Sustainable Cities and Society*, **72**, 102979. <https://doi.org/10.1016/j.scs.2021.102979>
- Jaouadi, Z., Abbas, T., Morgenthal, G. and Lahmer, T., (2020). Single and multi-objective shape optimization of streamlined bridge decks. *Structural and Multidisciplinary Optimization*, **61**(4), 1495-1514. <https://doi.org/10.1007/s00158-019-02431-3>
- Kim, H. and Mnih, A., (2018), July. Disentangling by factorising. In *International Conference on Machine Learning*. PMLR.
- Larsen, A., (1993). Aerodynamic aspects of the final design of the 1624 m suspension bridge across the Great Belt. *Journal of Wind Engineering and Industrial Aerodynamics*, **48**(2-3), 261-285. [https://doi.org/10.1016/0167-6105\(93\)90141-A](https://doi.org/10.1016/0167-6105(93)90141-A)
- Larsen, A., Esdahl, S., Andersen, J.E. and Vejrum, T., (2000). Storebælt suspension bridge-vortex shedding excitation and mitigation by guide vanes. *Journal of Wind Engineering and Industrial Aerodynamics*, **88**(2-3), 283-296. [https://doi.org/10.1016/S0167-6105\(00\)00054-4](https://doi.org/10.1016/S0167-6105(00)00054-4)
- Lepine, J., Guibault, F., Trepanier, J.Y. and Pepin, F., (2001). Optimized nonuniform rational B-spline geometrical representation for aerodynamic design of wings. *AIAA journal*, **39**(11), 2033-2041.
- Li, X., Lin, C., Li, R., Wang, C. and Guerin, F., (2020), November. Latent space factorisation and manipulation via matrix subspace projection. In *Proceedings of 37th International Conference on Machine Learning*.
- Li, S., Snaiki, R. and Wu, T., (2021). A knowledge-enhanced deep reinforcement learning-based shape optimizer for aerodynamic mitigation of wind-sensitive structures. *Computer-Aided Civil and Infrastructure Engineering*, **36**(6), 733-746. <https://doi.org/10.1111/mice.12655>

- Montoya, M.C., Hernández, S. and Nieto, F., (2018). Shape optimization of streamlined decks of cable-stayed bridges considering aeroelastic and structural constraints. *Journal of Wind Engineering and Industrial Aerodynamics*, 177, 429-455. <https://doi.org/10.1016/j.jweia.2017.12.018>
- Mooneghi, M.A. and Kargarmoakhar, R., (2016). Aerodynamic mitigation and shape optimization of buildings. *Journal of building engineering*, 6, 225-235. <https://doi.org/10.1016/j.jobe.2016.01.009>
- Nagao, F., Utsunomiya, H., Oryu, T. and Manabe, S., (1993). Aerodynamic efficiency of triangular fairing on box girder bridge. *Journal of Wind Engineering and Industrial Aerodynamics*, 49(1-3), 565-574. [https://doi.org/10.1016/0167-6105\(93\)90050-X](https://doi.org/10.1016/0167-6105(93)90050-X)
- Nieto, F., Montoya, M.C., Hernández, S., Kusano, I., Casteleiro, A., Alvarez, A.J., Jurado, J.Á. and Fontán, A., (2020). Aerodynamic and aeroelastic responses of short gap twin-box decks: Box geometry and gap distance dependent surrogate based design. *Journal of Wind Engineering and Industrial Aerodynamics*, 201, 104147. <https://doi.org/10.1016/j.jweia.2020.104147>
- Nieto, F., Cid Montoya, M. and Hernández, S., (2022). Shape optimization of tall buildings cross-section: Balancing profit and aeroelastic performance. *The Structural Design of Tall and Special Buildings*, e1982. <https://doi.org/10.1002/tal.1982>
- Omohundro, S.M., 1989. *Five balltree construction algorithms*. International Computer Science Institute, Berkeley.
- RealCity3D: A Large-scale Georeferenced 3D Shape Dataset of Real-world Cities, <https://ai4ce.github.io/RealCity3D/>, accessed on July 24, 2023.
- Ross, A., Chen, N., Hang, E. Z., Glassman, E. L., and Doshi-Velez, F., (2021). Evaluating the interpretability of generative models by interactive reconstruction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.
- Ross, A. and Doshi-Velez, F., (2021). Benchmarks, algorithms, and metrics for hierarchical disentanglement. In *Proceedings of the 38th International Conference on Machine Learning*, PMLR, 139, 2021.
- Sakai, Y., Ogawa, K., Shimodoi, H. and Saitoh, T., (1993). An experimental study on aerodynamic improvements for edge girder bridges. *Journal of Wind Engineering and Industrial Aerodynamics*, 49(1-3), 459-466. [https://doi.org/10.1016/0167-6105\(93\)90040-U](https://doi.org/10.1016/0167-6105(93)90040-U)
- Sharma, A., Mittal, H. and Gairola, A., (2018). Mitigation of wind load on tall buildings through aerodynamic modifications. *Journal of Building Engineering*, 18, 180-194. <https://doi.org/10.1016/j.jobe.2018.03.005>
- Shirzadi, M. and Tominaga, Y., (2021). Multi-fidelity shape optimization methodology for pedestrian-level wind environment. *Building and Environment*, 204, 108076. <https://doi.org/10.1016/j.buildenv.2021.108076>
- Tanaka, H., Tamura, Y., Ohtake, K., Nakai, M. and Kim, Y.C., (2012). Experimental investigation of aerodynamic forces and wind pressures acting on tall buildings with various unconventional configurations. *Journal of Wind Engineering and Industrial Aerodynamics*, 107, 179-191. <https://doi.org/10.1016/j.jweia.2012.04.014>
- Tang, J.W., Xie, Y.M., Felicetti, P., Tu, J.Y. and Li, J.D., (2013). Numerical simulations of wind drags on straight and twisted polygonal buildings. *The Structural Design of Tall and Special Buildings*, 22(1), 62-73. <https://doi.org/10.1002/tal.657>
- Tang, H., Li, Y., Wang, Y. and Tao, Q., (2017). Aerodynamic optimization for flutter performance of steel truss stiffening girder at large angles of attack. *Journal of Wind Engineering and Industrial Aerodynamics*, 168, 260-270. <https://doi.org/10.1016/j.jweia.2017.06.013>
- Tinmitondé, S., He, X. and Yan, L., (2022). Single-objective aerodynamic optimization of a streamlined bridge deck subjected to shape modification using a polynomial emulator and genetic algorithm. *Structural and Multidisciplinary Optimization*, 65(12), 1-21. <https://doi.org/10.1007/s00158-022-03459-8>
- Topping, B.H.V., (1983). Shape optimization of skeletal structures: a review. *Journal of Structural Engineering*, 109(8), 1933-1951. [https://doi.org/10.1061/\(ASCE\)0733-9445\(1983\)109:8\(1933\)](https://doi.org/10.1061/(ASCE)0733-9445(1983)109:8(1933))
- UIUC Airfoil Coordinates Database, https://m-selig.ae.illinois.edu/ads/coord_database.html
- Venkataraman, P., 1995. A new procedure for airfoil definition. In *13th Applied Aerodynamics Conference*.
- Whiteman, M.L., Fernández-Cabán, P.L., Phillips, B.M., Masters, F.J., Davis, J.R. and Bridge, J.A., (2022). Cyber-physical aerodynamic shape optimization of a tall building in a wind tunnel using an active fin system. *Journal of Wind Engineering and Industrial Aerodynamics*, 220, 104835. <https://doi.org/10.1016/j.jweia.2021.104835>
- Wang, Y., Yao, H. and Zhao, S., (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, 184, 232-242. <https://doi.org/10.1016/j.neucom.2015.08.104>
- Wang, Z., Zheng, C., Mulyanto, J.A. and Wu, Y., (2022). Aerodynamic Shape Optimization of a Square Cylinder with Multi-Parameter Corner Recession Modifications. *Atmosphere*, 13(11), 1782. <https://doi.org/10.3390/atmos13111782>
- Yang, Y., Wu, T., Ge, Y. and Kareem, A., (2015). Aerodynamic stabilization mechanism of a twin box girder with various slot widths. *Journal of Bridge Engineering*, 20(3), 04014067. [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0000645](https://doi.org/10.1061/(ASCE)BE.1943-5592.0000645)