International Journal of Mechanical Design and Materials Science

Volume 1, Issue 3 2025

In-depth Generative Design Combining Generative Models with Topology Optimization

Hongwu ,Zhang wang. Department of Mechanics,Russia

Research Article

Date of Submission: 10-03-2025 Date of Acceptance: 19-04-2025 Date of Publication: 1-05-2025

Abstract

Recently, deep learning has been used in a number of design optimization study domains. The need and efficacy of using deep learning in the field of generative design (or design exploration) research are discussed in this article. This paper suggests a deep generative design framework based on artificial intelligence (AI) that may provide a wide range of aesthetically pleasing and engineering performance-optimized design solutions. In order to investigate novel design possibilities, the suggested framework iteratively combines topology optimization with generative models (such as generative adversarial networks, or GANs), producing a huge number of designs from a little amount of prior design data. Anomaly detection may also assess the originality of created designs, assisting designers in selecting from a variety of design possibilities. The suggested methodology is validated using the 2D wheel design issue as a case study. Compared to earlier generative design techniques, the framework produces designs that are more resilient, diverse, and aesthetically pleasing.

Keywords: Deep learning, generative models, generative adversarial networks, generative design, design exploration, topology optimization, and design automation

Introduction

Artificial intelligence (AI) encompasses all technologies that aim to mimic human behavior in robots. A branch of artificial intelligence called machine learning uses statistical techniques to try to extract meaningful patterns from unprocessed data. Using a hierarchical neural network topology with several layers, deep learning aims to improve learning capacity [1,2]. Deep learning has recently been used in several technical fields in addition to computer science. In engineering domains, a data-driven method may effectively replace a physics-based approach. It has been extensively used in biomedical engineering, autonomous driving, robot control, prognostics and health management (PHM), and design optimization in mechanical engineering. Topology optimization [3–7], shape parameterization [8,9], computer-aided engineering (CAE) simulation and meta modeling [10–12], material design [13–15], and design preference estimation [16,17] are the five categories of deep learning research related to design optimization. Each study is thoroughly introduced in Section 2. This study began with the premise that deep learning is essential for generative design (i.e., design exploration), and the deep learning research listed above can undoubtedly be incorporated via the suggested framework in a comprehensive way.

Generative Design Using Topology Optimization

One technique for design exploration is generative design, which usually involves changing the design geometry parametrically and evaluating the resultant designs' performance [18, 19]. Instead of using design parameterization, recent generative design research uses topology optimization as a design generator and develops techniques to produce several designs concurrently using cloud computing [20]. Diverse boundary requirements for topology optimization are supplied by a designer, resulting in various optimal designs under various boundary circumstances. "Generative design varies the parameters of the problem definition while parametric design varies parameters of the geometry directly," according to Matejka et al. [21].

While traditional topology optimization looks for an ideal design, generative design aims to investigate design choices that fulfill structural performance and choose appropriate designs for

different designers' demands. The idea of generative design has been quickly expanded, included into commercial software [22], and used to construct a variety of structures, including cars, buildings, and airplanes. The four steps of the generative design process are as follows [20]: • Step 1: Establish the objectives and design parameters for topology optimization. • Step 2: Create designs while optimizing topology under various conditions. • Step 3: Examine your ideas, make many iterations, and decide which design works best. • Step 4: Use 3D printing to produce the design. The advent of 3D printing technology, in particular, has made it possible to produce intricate geometric shapes, which has accelerated the practical use of generative design. However, it was shown that contemporary generative design has a number of shortcomings [23]. First, even though topology optimization may be broadly classified as AI, it does not use cutting-edge AI technology (such as deep learning). Secondly, it cannot produce beautiful designs. Engineering performance is the exclusive objective of topology optimization. As a consequence, from an aesthetic perspective, the outcomes seem paradoxical. Customers, however, place a high value on aesthetics, which should be balanced with technical performance [24]. Third, there is little variation in optimum designs, which may lead to a new design in terms of pixel density or intensity. But in terms of how people perceive them, these designs could be comparable.

Generative Models for Generative Design

Research on generative design may be improved by using generative models, one of the exciting fields of deep learning. An approach for building a generator that learns the probability distribution of training data and uses that knowledge to create new data is known as a generative model. Popular generative models used in design optimization, where high-dimensional design variables are encoded in low-dimensional design space, include variational autoencoder (VAE) and generative adversarial network (GAN) [13,14]. These models are also used in form parameterization and design exploration [8,9].

There is little direct usage of generative models to create engineering designs [23]. This research, however, asserts that integrating with topology optimization may get over the restrictions. First, the generative model needs a lot of training data, but the industry's collected training data for different designs is private and hard to get. It is anticipated that many designs derived via topology optimization would be used as training data. Secondly, viable engineering cannot be guaranteed by the generative model. In this instance, topology optimization may be used to assess engineering performance. Third, one of the primary issues with the generative model is mode collapse, which results in just certain outcomes and significant output quality variation. However, topology optimization may be used in post-processing to enhance low-quality designs.

Research Purpose

This study proposes the new framework for generative design by integrating topology optimization and generative models. The proposed framework can provide a number of meaningful design options accounting for engineering performance and aesthetics and allows evaluation and visualization of the new design options according to the design attributes (e.g., novelty, compliance, and cost).

The proposed framework consists of iterative design exploration and design evaluation parts. Iterative design exploration involves generating a large number of new designs iteratively by using a small number of previous designs. Design evaluation involves quantifying the novelty of generated designs in comparison with the previous designs and visualizing design options with other design attributes. The proposed framework is applied to 2D wheel design of an automobile for demonstration.

The rest of this study is structured as follows. Section 2 reviews previous design optimization studies that employ deep learning. Section 3 proposes a deep generative design framework. Sections 4 and 5 present topology optimization and generative models, respectively, which are the main methodologies used in our study. Sections 6 and 7 present and discuss case study results, respectively. Finally, Section 8 summarizes the conclusions and limitation and introduces future work.

Literature Review: Deep Learning in Design Optimization Research

Here is an explanation of the deep learning-based design optimization study. First, since topology optimization distributes the materials in the design domain while taking objective and constraint functions into consideration, it may be seen as deep learning from the standpoint of pixel-wise picture labeling [3]. Due to the iterative finite element analysis (FEA) of a structure, topology optimization has a disadvantage of high computing demand. Therefore, Yu et al. [4] present a framework that uses a convolutional neural network (CNN)-based encoder and decoder built on 2D topology optimal designs to first generate a low-resolution structure, which is then upscaled to a high-resolution structure using conditional GAN (cGAN) without the need for iterative FEA in topology optimization. By using a 3D CNN-based encoder and decoder, Banga et al. [5] are able to acquire the final optimal structural output from intermediate structural inputs. Guo et al. [6] use style transfer and VAE to optimize the topology of latent variables in reduced dimensional space. Active learning was used by Cang et al. [7] to limit a neural network's training such that the network produces nearly optimum topologies. Second, shape parameterization (i.e., design representation) deep learning applications have been created. Shape optimization requires the parameterization to determine the geometry. However, it is very challenging to define the variables in complex geometry, and there is too much interaction between the variables, which might make parameterization in a mathematical method problematic. Using VAE, Burnap et al. [8] demonstrate the parameterization potential of an automobile's 2D form. Additionally, Umetani [9] demonstrates how to use an autoencoder to parameterize 3D car models. Third, metamodeling and simulation-based optimization have been used with deep learning. Due of the high computational cost of CFD modeling, several studies have attempted to apply deep learning to CFD. Tompson et al. [11] accelerate Eulerian fluid simulation by using CNN to approximate linear systems, whereas Guo et al. [10] propose the CNN model to anticipate the responses of CFD simulation for the 2D form of a car. Additionally, Farimani et al. [12] suggest integrating cGAN to address the steady-state issue of incompressible fluid flow and heat conduction.

Fourth, the direct link between the density of elements in material structures and the pixels of pictures has led to the development of deep learning applications on material design, making the domain transfer from material structures to images simple. GAN is used by Yang et al. [14] to map the microstructure into low-dimensional latent variables in order to find the best microstructure using a Bayesian optimization framework. Cang et al. [13] provide a feature extraction technique that uses a convolutional deep belief network to transform the microstructure into low-dimensional design space. Cang et al. [15] propose VAE under morphological constraint and demonstrate that a modest quantity of training data may generate an indefinite amount of microstructure.

Deep learning has finally been used to the estimate of design preferences. By learning a restricted Boltzmann machine (RBM) using the original design variables as input and extracting the future, Burnap et al. [16] increase the prediction accuracy of the customer's preference model. Using Siamese neural network architecture with cGAN, Pan et al. [17] suggest learning preference on aesthetic attractiveness.

Furthermore, many engineering designs make advantage of deep learning. For example, Dering and Tucker [25] have successfully mapped the design's shape and function using a 3D convolution neural network, while Dering and Tucker [26] suggest a picture generating approach that combines big data and deep learning.

The aforementioned research's classification is not separate from one another, which permits integration and ultimately improves the traditional design optimization procedure. The authors specifically assert that generative design is situated at the nexus of all these fields of study and that it would be a very promising field of study inside an AI-based design automation system.

Deep Generative Design Framework

A deep generative design framework was proposed, which integrates topology optimization and generative models. Fig. 1 shows the entire process which consists of two main parts (i.e., iterative design exploration and design evaluation) and nine stages.

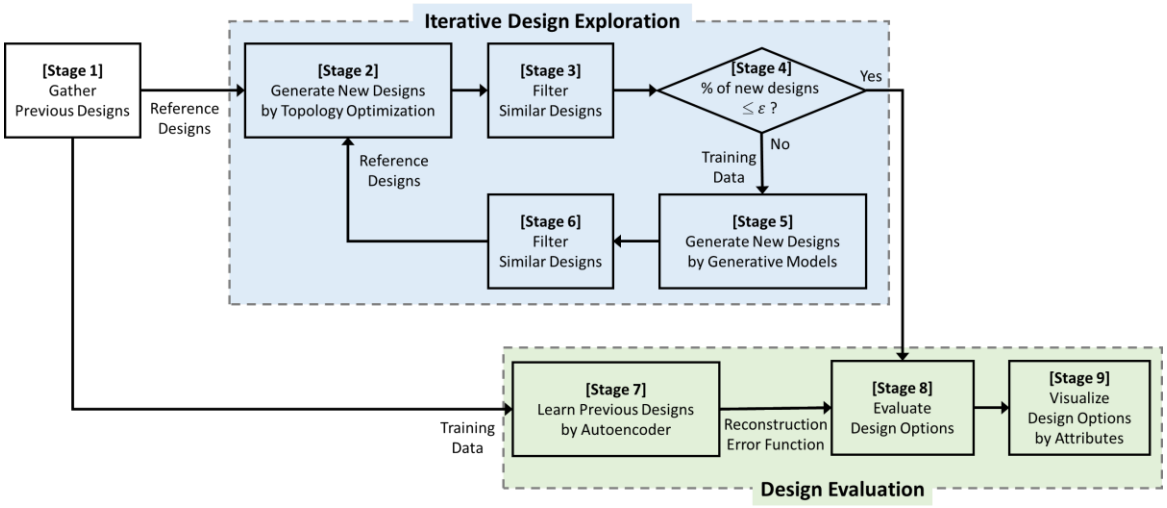


Fig. 1. Deep generative design framework

Design assessment measures and assesses the uniqueness and key characteristics of new designs, whereas iterative design exploration integrates topology optimization and generative models to create new ideas. The following is an explanation of each step:

The market and industry's prior designs are gathered in Stage 1 to serve as reference designs for Stage 2. The reference design in this research is characterized as a benchmark design used to develop new topology optimization designs. Topology optimization is used in Stage 2 to produce new designs based on reference designs. The multi-objective function of topology optimization in this stage is (1) compliance minimization, which reflects technical performance, and (2) difference (i.e., pixel-wise L1 distance) minimization from the reference design, which seeks to enhance variety and aesthetics. Because there is a tradeoff between two purposes, different designs may be achieved by changing the relative weights of each objective function. Since earlier designs were made by human designers, we presume that they are more aesthetically pleasing than traditional topology optimization outcomes. It is anticipated that the ultimate optimization result would be more aesthetically pleasing if topology optimization can benchmark the form of earlier designs. In terms of variety, the outcomes of topology optimization would be more varied if a wider range of reference designs were used as input. Section 4.2 provides a detailed explanation of this level. In Stage 3, the similarity criteria selected for the user-specified threshold filters away comparable designs collected from topology optimization in order to minimize computing costs caused by irrelevant designs. This research evaluates a comparable design using the pixel-wise L1 distance, which is set at 103. The number of designs that must be produced will decrease if this value is set too strictly, but the distinction between designs will be enhanced. To exclude the designs produced by the generative models, this procedure is repeated in Stage 6. It's crucial to remember that the L1-norm separates two identical designs with different rotations and regards them as distinct. This L1-norm constraint may be overcome by using generative models to map the design variables into the latent space and compute the L1-norm. However, there weren't many of these examples in our case study, therefore we solely employed L1-norm in the design space. The number of new designs in the current iteration divided by the total number of designs in the previous iteration is computed in Stage 4. Go to Stage 8 and stop the iterative design exploration if it is less than the user-specified threshold; if not, go on to Stage 5. The threshold for termination criterion in our investigation is fixed at 0.3. Depending on how

many different designs the user want to produce, this variable may be changed. After learning aggregated designs from the current iteration, generative models produce new designs in Stage 5, which are then utilized as reference designs in Stage 2 after Stage 6 filters out similar ideas. The structure and parameters of Boundary Equilibrium GAN (BEGAN), which we employed, are described in Section 5.2. From Stage 2 to Stage 6, iterative design exploration should be carried out continually until the termination requirement is met, or until a significant number of designs are produced. By beginning with a limited number of earlier designs in Stage 1, this iterative approach aims to produce a huge number of different designs. Stages 7 through 9 include the design review portion. In Stage 7, an autoencoder trained by earlier Stage 1 designs is used to produce a loss function, or reconstruction error function. Design uniqueness may be assessed using this function by comparing it to earlier designs. Section 5.2 introduces the autoencoder model's specifics. Stage 8 requires the evaluation of design possibilities derived from iterative design exploration based on a variety of design criteria that are crucial to the designers. It may assess physical parameters like volume and design conformity in addition to the originality of created designs. Lastly, the trade-off between characteristics is shown by charting designs along each design attribute axis. The appropriate designs may then be selected based on the relative significance of each attribute. An automobile's 2D wheel serves as a case study for the suggested framework. The two primary approaches of our suggested framework—topology optimization and generative models—are thoroughly described in Sections 4 and 5 that follow.

Topology Optimization

Section 4.1 introduces the basic theory of topology optimization which our study stems from, and Section 4.2 presents the proposed topology optimization method for the wheel design case study.

Basic Theory

Density-based Approach

Topology optimization is commonly referred to the material distribution method developed and spread to a wide range of disciplines. The basic concept is how to distribute materials in a given design domain without any preconceived design [27-29]. In this study, the compliance minimization related to the stiffness of a structure has been carried out to redesign existing wheels. Many approaches such as homogenization and level-set methods can be applied, but we choose the density-based approach where material distribution is parameterized by the density of elements. Especially, solid isotropic material with penalization (SIMP) is a procedure that implicitly penalizes intermediate density values to lead to the black-and-white design. The basic formulation of SIMP in compliance minimization can be written as [29,30]

$$\begin{aligned} \min \quad & c(\mathbf{x}) = \mathbf{U}^T \mathbf{K}\mathbf{U} = \sum_{e=1}^{N_e} \mathbf{u}_e^T \left(E_e(x_e)\mathbf{k}_0\right)\mathbf{u}_e \\ \text{s.t} \quad & V(\mathrm{x}) / V_0 = f \\ & \mathrm{KU} = \mathrm{F} \qquad (1) \\ & 0 \le x_e \le 1, \; e = 1,\ldots,N_e \end{aligned}$$

where U is a displacement vector; K is a global stiffness matrix; $c(\mathrm{x})$ is the compliance; u_e is an element displacement vector; k_0 is an element stiffness matrix; f is the volume fraction; N_e is the number of elements; x_e is the design variable (i.e., density) of element e; and $V(\mathrm{x})$ and V_0 are the material volume and the volume of design domain, respectively. In modified SIMP, the density that is directly associated with Young's

modulus can be expressed as [31]

$$E_e(x_e) = E_{min} + x_e^p(E_0 - E_{min}) \quad (2)$$

where p is a penalization factor to ensure the black-and-white design, and $E_{\min}$ instability when the density of elements become zero.s introduced to avoid numerical
Many studies have been done to enhance the performance of topology optimization such as filtering techniques. In this study, we develop the code based on 99- and 88-line MATLAB codes, which are the simplest and most efficient two-dimensional topology optimization codes written in MATLAB [30,32]. Thus, we will briefly explain algorithms such as sensitivity analyses and filtering techniques used in this study.

4.1.2 Sensitivity Analysis and Filtering Techniques

In gradient-based optimization, the sensitivity analysis of objective and constraint function with respect to each design variable is required to provide accurate search direction to the optimizer. Therefore, the sensitivity analysis with respect to the density of elements can be given by

$$\frac{\partial c}{\partial x_e} = -px^{p-1}(E_0 - E_{\min})u^T k_e u \quad (3)$$

and

$$\frac{\partial V}{\partial x_e} = \frac{\partial}{\partial x_e}\left(\sum_{e=1}^{N_e} x_e v_e\right) = 1 \quad (4)$$

under the assumption that all elements have a unit volume. On the other hand, the optimality criteria (OC) method, one of the classical approaches to structural optimization problems, is employed in this paper. The OC method updates the design domain as

$$x_e^{new} = \begin{cases} \max(0, x_e - m) & \text{if } x_e B_e^{\eta} \le \max(0, x_e - m) \\ \min(1, x_e + m) & \text{if } x_e B_e^{\eta} \ge \min(1, x_e + m) \\ x_e B_e^{\eta} & \text{otherwise} \end{cases} \quad (5)$$

where m is a positive move-limit and η is a numerical damping coefficient, and

$$B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}} \quad (6)$$

The Lagrange multiplier related to volume fraction constraint can be obtained from a bisection algorithm that is one of the popular root-finding algorithms. The termination criteria for the convergence can be written as

$$\left\| x_{new} - x \right\|_{\infty} \le \varepsilon \quad (7)$$

where ε is the tolerance usually set as a relatively small value such as 0.01.

For the assurance of the existence of well-posed and mesh-independent solutions, several strategies to avoid a checkerboard pattern and gray-scale issues are introduced. In this study, we apply so-called three-field SIMP, which has a projection scheme. Three-field means the original density, filtered density, and projected density. Detailed descriptions can be seen in the literature [33].

The basic filters applied to topology optimization are sensitivity and density filters, which are used in one-field and two-field SIMP, respectively. The main idea of both techniques is to modify sensitivity or physical element density to be a weighted average of the neighborhood. The neighborhood is defined on the basis of the distance from the center of the element, and the maximum distance to include in the neighborhood is a user-specified parameter referred to the mesh-independent radius. The sensitivity

$$\frac{\partial c}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^{N} H_f} \sum_{f=1}^{N} H_f x_f \frac{\partial c}{\partial x} \quad (8)$$

where the convolution operator can be written as

$$H_f = r_{\min} - \mathrm{dist}(e, f), \quad (9)$$

where subscript f means one of the elements that the center-to-center distance expressed as $\mathrm{dist}(e, f)$ between elements is smaller than $r_{\min}$.
The density filter defines the physical density with weighted averaging. The weighted average concept is the
same in the sensitivity filter as Eq. (8), but the density is filtered instead of sensitivity expressed as [35,36]

$$\tilde{x} = \frac{\sum_{f=1}^{N} H_f x_f}{} \quad (10)$$

$\sum_{f=1}^{N} H_{f} \sum_{e}$

Therefore, original and filtered densities can be referred to as a design variable and physical density, respectively. The sensitivity analysis with respect to design variables should be modified by introducing the physical density using a chain rule. A detailed description can be seen in the literature [37].
The weighted average is used in both filtering methods to avoid the checker-board pattern in an optimum design. However, the density filter can induce gray transitions between solid and void regions. Thus, the third field of density, or the so-called projection filter, is introduced. It mitigates the gray transition problem by projecting to solid and void usually using a smoothed Heaviside projection [31,37,38]. In this study, we use the Heaviside projection filter on the filtered density obtained from Eq. (10). The projection filter can be written as

$$\bar{\tilde{x}}_e = 1 - e^{-\beta \tilde{x}_e} + \tilde{x}_e e^{-\beta} \tag{11}$$

where β is a parameter related to slope of projection and can be updated through the optimization. In three-field SIMP with projection filter, the sensitivity analysis is modified compared with Eq. (8) because the finite element analysis is performed based on the physical density obtained from Eq. (11). The sensitivity analysis with respect
to design variables can be easily derived using the chain rule.

4.2. Proposed Topology Optimization
In the deep generative design framework of Fig. 1, Stage 2 generates new engineering designs through topology optimization reflecting the shape of the reference designs that can be either previous wheel designs from Stage 1 or generated designs from generative models (Stages 5 and 6). A number of engineering performances has to be considered when designing the wheel of vehicle, but compliance obtained from the static analysis has been generally employed in this research for the sake of simplicity. Fig. 2 sketches the design domain and boundary conditions for 2D wheel design. The original design domain is 128 by 128 elements, and the reference designs also have 128 by 128 pixels. The outer ring of the wheel is set to the non-design region to maintain the shape of the rim, and the inner region is set to fixed boundary condition for connecting parts. Therefore, spoke is the main component in the design domain.
The element stiffness matrix is based on 4-node bilinear square elements in the 88-line MATLAB code [32]. Normal and shear forces are uniformly exerted along the surface, which are common load conditions in the 2D wheel optimization. The normal force and the shear force represent uniform tire pressure and tangential traction, respectively. The ground reaction induced by vehicle weight is disregarded because it requires an additional symmetric condition.
The ratio between normal and shear force is a user-specified parameter that can significantly change the optimized wheel design. The force ratio is defined as the magnitude of normal force divided by shear force.

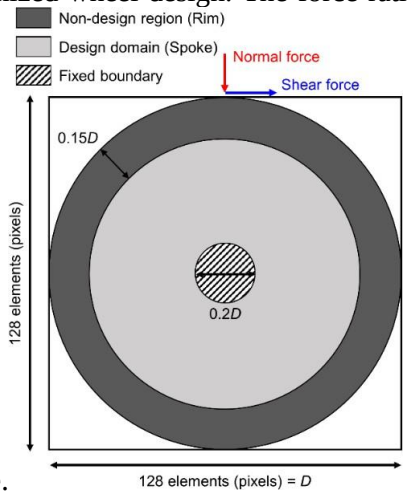


Fig. 2. Design domain and boundary conditions of a 2D wheel design

However, varying boundary conditions only can limit the production of meaningful and diverse designs. Thus, a new objective function in topology optimization was introduced so that the design generator can produce engineering designs while maintaining the shape of various reference designs. The modified objective function can be formulated as

$$f_1(x) = U^T K(x) U + \lambda \left\| x^* - x \right\| \quad (12)$$

where $U^T K(x)U$ is compliance, λ is a user-specified similarity parameter, and x^* is elements of the reference design. Therefore, the L1-norm between generated and reference design represents the similarity. L1-norm is more preferred than L2-norm because it can alleviate the blurring of design, and shows better design quality. In addition,
cross entropy loss can also be an alternative to L1-norm depending on how the problem is defined. However, since
cross entropy is more commonly used in discrete problems, the pixel-wise L1-norm is more appropriate in our problem which is continuous.
All other processes are identical with conventional three-field SIMP explained in Section 4.1. Reference design represented as x^* is a binary matrix with entries from the Boolean domain since it is black-and-white design. Hence, sensitivity analysis for additional similarity term can be expressed as

$$\frac{\partial}{\partial x} \lambda \left\| x^* - x \right\| \quad \left(\left\| \cong \right\|_1 \right) -\lambda x^* \quad (13)$$

The above expression means that if a specific element in reference design is solid, then the sensitivity is set to
$-\lambda$, and 0 otherwise to avoid providing the positive sensitivity to the OC optimizer. In other words, the purpose of Eq. (13) is to give additional weights of sensitivity on the solid elements of reference design, so that the optimized design can be affected by the shape of reference design.
Consequently, five discrete levels of similarity parameter and force ratio are configured to generate new designs from topology optimization as listed in Table 1. For instance, if 100 reference designs are available, then 2500 designs (100 reference designs × 5 similarity parameter levels × 5 force ratio levels) can be obtained from topology optimization. The type of condition and the number of levels can be determined by the designers such as voxel sizes, solver parameters, and the number of iterations in topology optimization [21].

Table 1. Condition parameter levels

Condition	Levels
Similarity	5 levels: 0.0005/0.005/0.05/0.5/5
Force ratio	5 levels: 0/0.1/0.2/0.3/0.4

Generative Models
Section 5.1 introduces popular generative models briefly, and explains the BEGAN model which is mainly used

in our proposed framework. Section 5.2 shows how to utilize BEGAN for generating and evaluating wheel designs.

Basic Theory

Generative Adversarial Networks (GANs)

GANs are designed to infer the data generating a distribution $p_{\mathbf{data}}(x)$ by making the model distribution generated by generator $p_{\mathbf{g}}(G)$ to be close to the real data distribution where x is the variable of real data and G is the generator's differentiable function with parameters $\theta_{\mathbf{g}}$. Function G has an input noise variable z and tries to
map it to the real data space by adjusting $\theta_{\mathbf{g}}$, thus represented as $G(z;\theta_{\mathbf{g}})$. Similarly, the discriminator's differentiable function is derived as $D(x;\theta_{\mathbf{d}})$, which attempts to predict the probability that the input is from the real dataset. The zero-sum game of maximizing the discriminator and generator is equivalent to maximizing $\log D(x)$ and minimizing $\log(1- D(G(z)))$ [39,40] as

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1- D(G(z))] \quad (14)$$

With this standard GAN structure, various GANs have been developed by modifying the generator, discriminator, or objective function.
To overcome the notorious difficulty in training GANs, deep convolutional GANs (DCGANs) provide a stable training model, which works on various datasets by constructing a convolutional neural network in the generator and discriminator. In addition, DCGANs suggest certain techniques such as removing a fully-connected layer on top, applying batch normalization, and using the leaky rectified linear unit activation function [41].
Adversarially learned inference (ALI) and bidirectional GANs (BiGANs) adopt the encoding–decoding model to the generator to improve the quality of generated samples in an efficient way. Moreover, BiGANs emphasize taking advantages of learned features [42,43]. For high-resolution images with stable convergence and scalability, energy-based GANs (EBGANs) have proven to produce realistic 128 × 128 images. EBGANs consider the discriminator as an energy function and the energy as the reconstruction error. From this point of view, an autoencoder architecture is used for the discriminator [44]. The autoencoder consists of encoder and decoder functions. The input value is transformed through the encoder and is restored to its original form again through the decoder [2]. Wasserstein GANs (WGANs) approach the way to obtain good image quality by changing the distance measure of two probability distributions. WGANs show that the earth-mover distance, which is also called Wasserstein-1, provides a differentiable function, and, thus, produces meaningful gradients, whereas Kullback-Leibler and Jensen-Shannon divergence in previous research do not when two probability distributions are disjointed [40]. BEGANs also use the Wasserstein distance as a measure of convergence. BEGANs present an equilibrium concept, balancing the discriminator and generator in training and the numerical way of global convergence [45].
Aside from enhancing the image quality, the way to control the mode of generated outputs is presented by cGANs [46] and InfoGAN (Information maximizing Generative Adversarial Nets) [47]. cGANs provide additional input values to the generator and discriminator for categorical image generation. Furthermore, InfoGAN lets the generator produce uncategorical images by adding a latent code that can be categorical and continuous. It is useful for finding hidden representations from large amounts of data. However, intentionally creating a specific image is still difficult.
Hitherto, many studies on GANs contribute to good image quality in terms of convergence and stability. However, GANs are still weak in utilizing from the design engineering point of view, such as uneven image quality from the same saving point of the model, especially when relatively small amounts of training data are given and images have insufficient engineering features.

Boundary Equilibrium GAN (BEGAN)

This paper employs BEGAN among GANs for the proposed framework because it provides a robust visual quality in a fast and stable manner. The autoencoder architecture as the discriminator used in EBGANs is also introduced by BEGANs. Similar to WGANs, BEGANs use the Wasserstein distance as a measure of convergence. With these techniques, BEGANs achieve reliable gradients that are difficult for high-resolution 128 × 128 images.

Rather than trying to match the probability distribution of real data, BEGANs focus on matching autoencoder loss distribution. It measures the loss, which is the difference between the sample and its output that passed through the autoencoder. Subsequently, a lower bound of the Wasserstein distance between the autoencoder loss

distribution of real and that of generated samples is derived. The autoencoder loss function, $\mathcal{L}: \mathbb{R}^{Nx} \rightarrow \mathbb{R}^{+}$, is defined as

$$\mathcal{L}(v) = |v - A(v)|^{\eta} \quad (15)$$

where $A: \mathbb{R}^{N_x} \rightarrow \mathbb{R}^{N_x}$ is an autoencoder function

$\eta \in \{1,2\}$ is a targe norm

$v \in \mathbb{R}^{N_x}$ is a sample of dimension N_z

Applying Jensens' inequality, the lower bound of the Wasserstein distance is derived as

$$m_1 - m_2 \quad (16)$$

where $m_i \in \mathbb{R}$ is the mean of autoencoder loss distribution. For the maximization of Eq. (16) for the discriminator with $m_1 \rightarrow 0$ and $m_2 \rightarrow \infty$, BEGANs' objective function is described as minimizing the discriminator's
autoencoder loss function $\mathcal{L}_D$ and generator's one $\mathcal{L}_G$ as the following where θ_D and θ_G are the parameters of the discriminator and generator, $G: \mathbb{R}^{Nz} \rightarrow \mathbb{R}^{Nz}$ is the generator function, $z \in [-1,1]^{Nz}$ are uniform random
samples of dimension N_z, and z_D and z_G are samples from z. The objective functions are defined as

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t \mathcal{L}(G(z_D)) & \text{for } \theta_D \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\ k_{t+1} = k_t + \lambda_k(\gamma \mathcal{L}(x) - \mathcal{L}(G(z_G))) & \text{for each training step} \end{cases} \quad (17)$$

where $k_t \in [0, 1]$ is a control factor to determine how much $\mathcal{L}(G(z_D))$ is reflected during gradient descent; λ_k is a proportional gain for k such as the learning rate in machine learning terms; and $\gamma \in [0, 1]$ is a diversity ratio that results in high image diversity as it increases. Given that k_t is changed in every training step to maintain
$\mathbb{E}[\mathcal{L}(G(z))] = \gamma \mathbb{E}[\mathcal{L}(x)]$ for the equilibrium, global measure of convergence is regarded as the closest
reconstruction $\mathcal{L}(x)$ with the minimum absolute value of proportional control algorithm error $|\gamma \mathcal{L}(x) - \mathcal{L}(G(z_G))|$ [31] as

$$M_{global} = \mathcal{L}(x) + |\gamma \mathcal{L}(x) - \mathcal{L}(G(z_G))| \quad (18)$$

Proposed Generative Models

Although designers can utilize various state-of-the-art generative models in parallel for Stage 5 in the proposed framework (see Fig. 1), we chose and modified BEGAN architecture as illustrated in Fig. 3. The encoder of discriminator is a network consisting of five 3 × 3 convolutional layers, four 2 × 2 sub-sampling layers with stride 2, and one fully connected (FC) layer. For the dimension of each layer, w × h × n represents the width, height, and the number of kernels, respectively. The exponential linear unit (ELU) is used for the activation function. Generator and decoder of discriminator use a similar structure as this but by replacing sub-sampling to up-sampling. The model was trained with 16, 32, 64, and 128-dimensional latent variable **z**, and all the results were utilized. Adam optimizer was used with a learning rate of 0.00008, initial value of k_t as 0, λ_k as 0.001, γ as 0.7, and minibatch size of 16 (see Section 5.1.2). The learning rate parameter is set with reference to the settings of previous papers that studied image quality by using BEGAN [48,49].

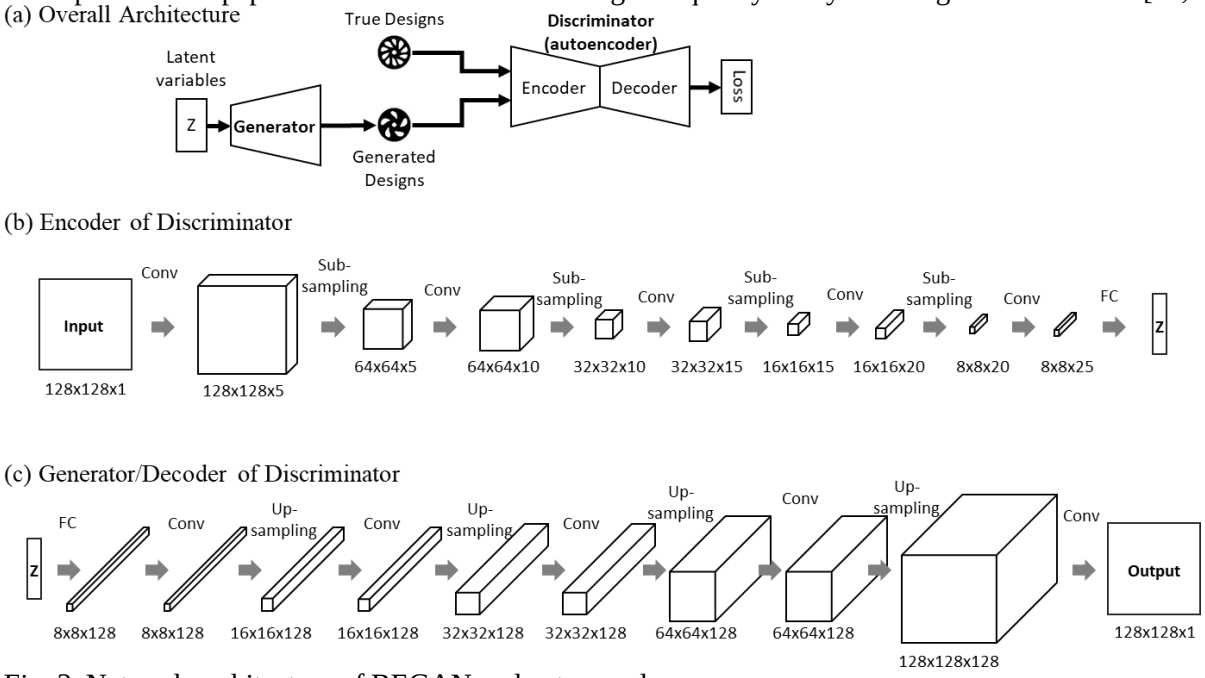


Fig. 3. Network architecture of BEGAN and autoencoder

In Stage 7, autoencoder is used to evaluate design novelty. Reconstruction errors, which are the loss functions of autoencoder, are widely used to detect anomaly [50]. The idea is that autoencoder can effectively reconstruct similar data to training data but not dissimilar data. This study assumes that design novelty can be measured the same way as anomaly detection. Previous designs in Stage 1 used training data for autoencoder, and then trained autoencoder can calculate reconstruction error of new designs generated by iterative design exploration. The new design which has the higher value of reconstructing error is regarded as the design which has the more novelty. This study employs the autoencoder structure which is used in BEGAN as the discriminator in Fig. 3, with the same hyper-parameter settings used. The equation of reconstruction errors is the same as Eq. (15).

In addition, a regression model can be used as an alternative to autoencoder when previous design data are insufficient. Topology design results (in Stage 2) at the first iteration have the similarity parameter (λ), so that regression model can be built where output y is set to the similarity parameter. VGG16 [51], which is one popular CNN for regression model, was tested. Results show that CNN-based regression can also predict similarity which is contrary to novelty.

Results

This section shows the results of a case study applying the proposed framework to 2D wheel design. In Stage 1, frontal wheel designs were collected in the market by web crawling and converted it to binary images. A total of 658 binary images of the wheel are collected through post-processing as reference

designs for the first iteration of iterative design exploration.

Iterative Design Exploration

6.1.1 Design Exploration by Topology Optimization (Stages 2 and 3)
Topology optimization is performed in parallel according to similarity and force ratio parameters listed in Table 1. Fig. 4 shows an example of optimized designs according to five levels of force ratio when the similarity is 0.0005. A large shear force is observed to make many thin spokes of whirlwind shape. On the other hand, large normal force makes thick and less curved spokes.

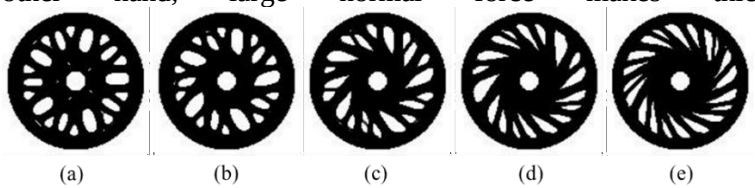


Fig. 4. Topology optimization of wheel design when the force ratio is set to (a) 0, (b) 0.1, (c) 0.2, (d) 0.3, (e) 0.4

Fig. 5 shows the optimal designs under five levels of similarity (i.e., (b) to (f)). (g) shows the reference designs, and (a) shows the result when reference design is unused. The ratio between normal and shear force is set to 0.1 in case of reference design A and 0.2 in case of reference design B, and the volume fraction is identical to reference designs. Table 2 lists the similarity and compliance of each optimal design. The optimal designs from topology optimization evidently indicate the tradeoff between engineering performance and similarity to the reference design.

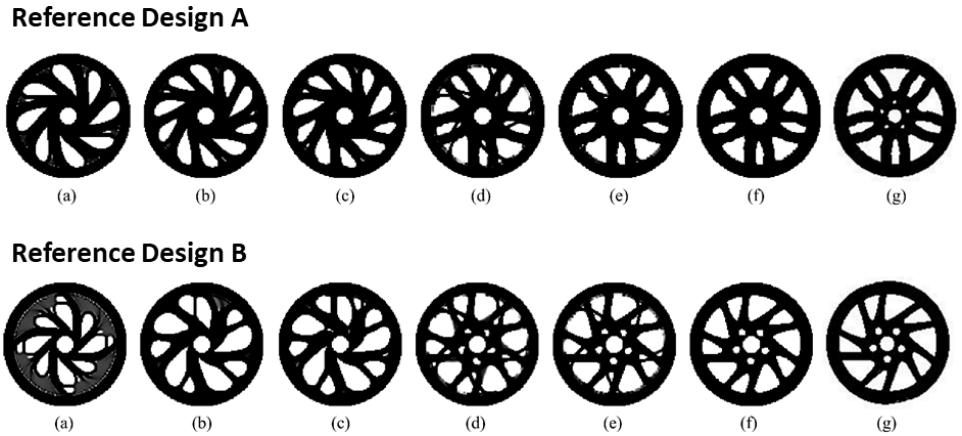


Fig. 5. Topology optimization results when the similarity is set to (a) 0, (b) 0.0005, (c) 0.005, (d) 0.05, (e) 0.5,
(f) 5, and (g) reference design

Table 2. Similarity and compliance of each reference design in Fig. 5

Similarity		0	0.0005	0.005	0.05	0.5	5
Compliance	Reference design A	5.28	5.23	6.17	7.14	8.87	9.28
	Reference design B	8.88	8.90	8.94	9.76	10.71	13.02

Fig. 6 demonstrates the effectiveness of the proposed objective function in Eq. (12), which states that the
proposed method can yield different designs reflecting the shape of reference design when other boundary conditions are the same for all optimized design such as force ratio. The described results suggest that the proper range of similarity that optimal designs are continuously changed varies depending on the reference design. Therefore, an experimental investigation is encouraged in advance.

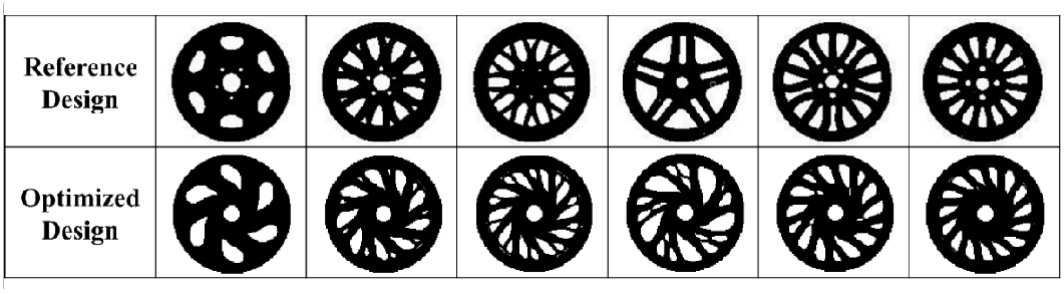


Fig. 6. Optimized designs under the same boundary conditions and different reference design

Consequently, 1619 new designs have been created after filtering at Stage 3. Stage 4 is passed automatically in the first iteration of the iterative design exploration, and the ratio of new designs is calculated as a criterion from the second iteration.

6.1.2. Design Exploration by Generative Model (Stages 5 and 6)
A total of 2277 designs were identified after Stage 4, where 658 previous designs were identified at Stage 1 and 1619 designs were obtained through Stage 4. These 2277 designs are used for training designs of BEGAN at Stage 5. The training takes around three hours on four GTX 1080 GPUs in parallel. Fig. 7 presents examples of 128 × 128 image from the BEGAN generator, and Fig. 8 shows that global convergence $\mathcal{M}_{global}$ is achieved without oscillation. A total of 128 designs were achieved through filtering at Stage 6. These designs were used as reference designs for Stage 2 at the second iteration of iterative design exploration. New 385 topology designs are regenerated after topology optimization (Stages 2 and 3).

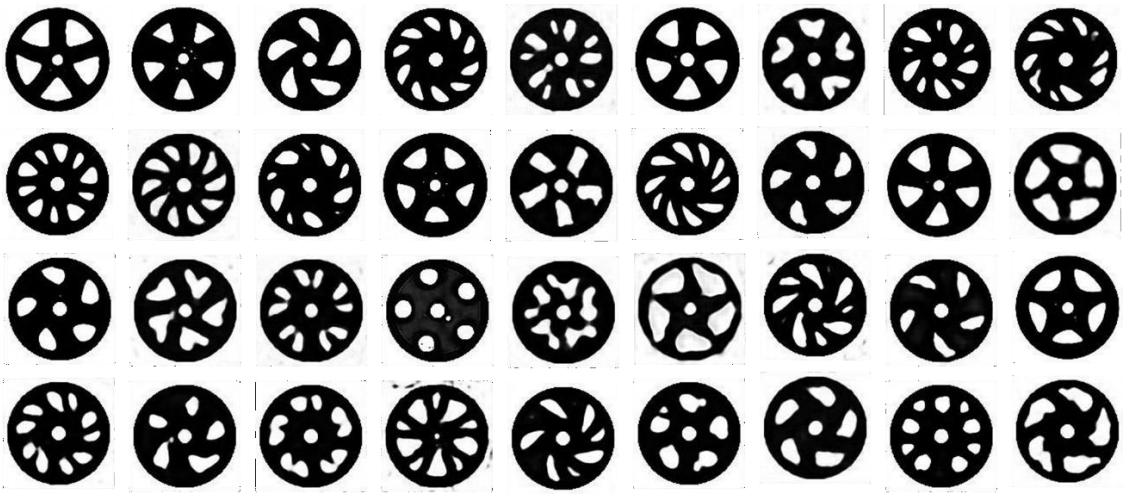


Fig. 7. Generated wheel designs by BEGAN

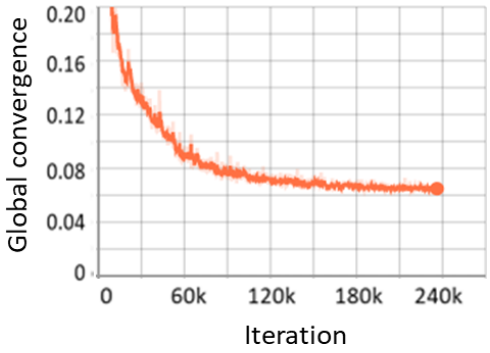


Fig. 8. Convergence results of BEGAN

As shown in Fig. 7, the BEGAN design results are roughly symmetrical, circular, and have holes in the center. Many GAN research studies in computer science use face dataset as benchmark data, and their results also demonstrate GANs capture symmetry features of human faces very well, without being taught [41,45].

Design Evaluation

Novelty Evaluation by Autoencoder (Stage 7)
For the autoencoder, 80% of the previous designs were used as training data, and 20% as test data. Fig. 9 shows examples comparing reconstruction results between test data of previous designs and generated designs. Designs similar to previous ones portray satisfactory reconstruction, while dissimilar designs portray otherwise.

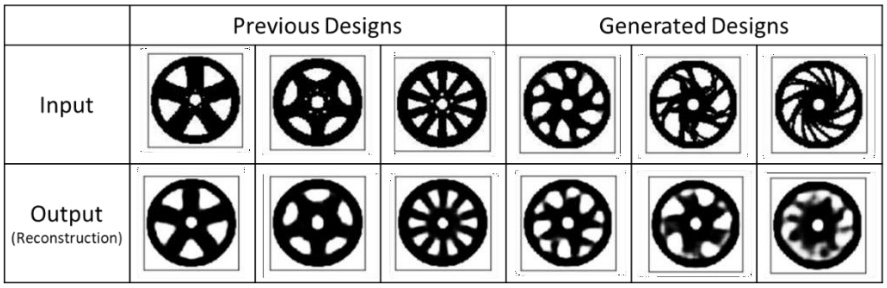


Fig. 9. Comparison between previous designs and generated designs in reconstruction of autoencoder

Evaluation and Visualization (Stages 8 and 9)

Finally, 2004 new designs are generated after two iterations, which are not included in the previous design set. Table 3 summarizes the number of input and output designs used at each stage. The termination criteria calculated at Stage 4 after two iterations is 23.8% (385/1619 = 23.8) which is less than the threshold of 0.3.

Table 3. Number of new generated designs at each stage

Iteration	First	Second

Method	Topology Optimization (Stages 2 & 3)	BEGAN (Stages 5 & 6)	Topology Optimization (Stages 2 & 3)
Input	658	2277 (= 1619 + 658)	128
Output	1619	128	385
New topology designs (accumulated)	1619	-	2004 (= 1619 + 385)

Examples of design options are shown in Fig. 10. A 3D scatter plot for 2004 design options was crafted by using three design attributes (novelty, cost, and compliance) as an axis (Fig. 11(a)). Each attribute value used in the plot is normalized from 0 to 1. Fig. 11(b) shows that trade-offs between compliance and cost make a smooth Pareto curve, because designs are all topologically optimized. On the Pareto curve, two designs are shown as examples, one with the lowest cost and the other with the lowest compliance. Figures 11(c) and 11(d) show trade-offs between novelty and other attributes. Assuming novelty is a positive trait, two designs located on the Pareto curves are shown as examples in the figures, respectively.

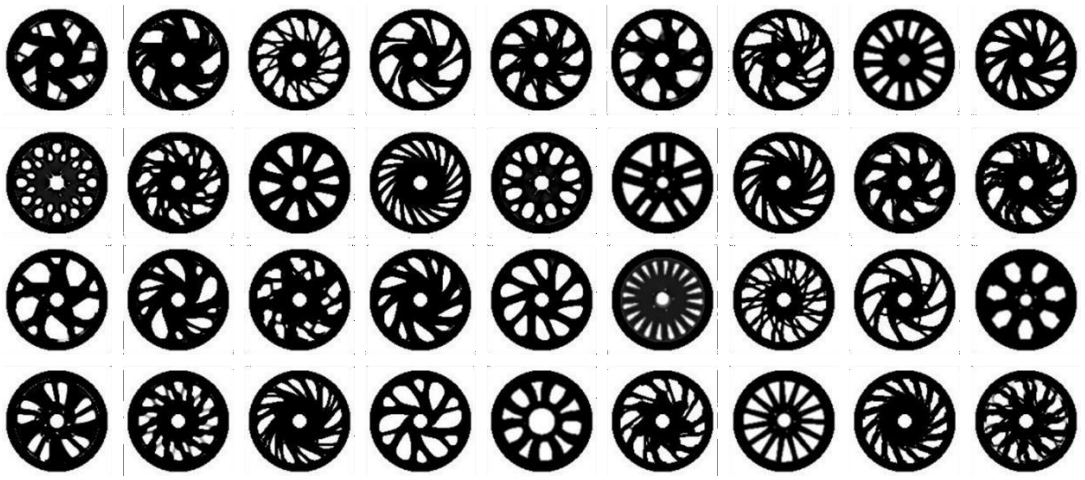


Fig. 10. Generated design options

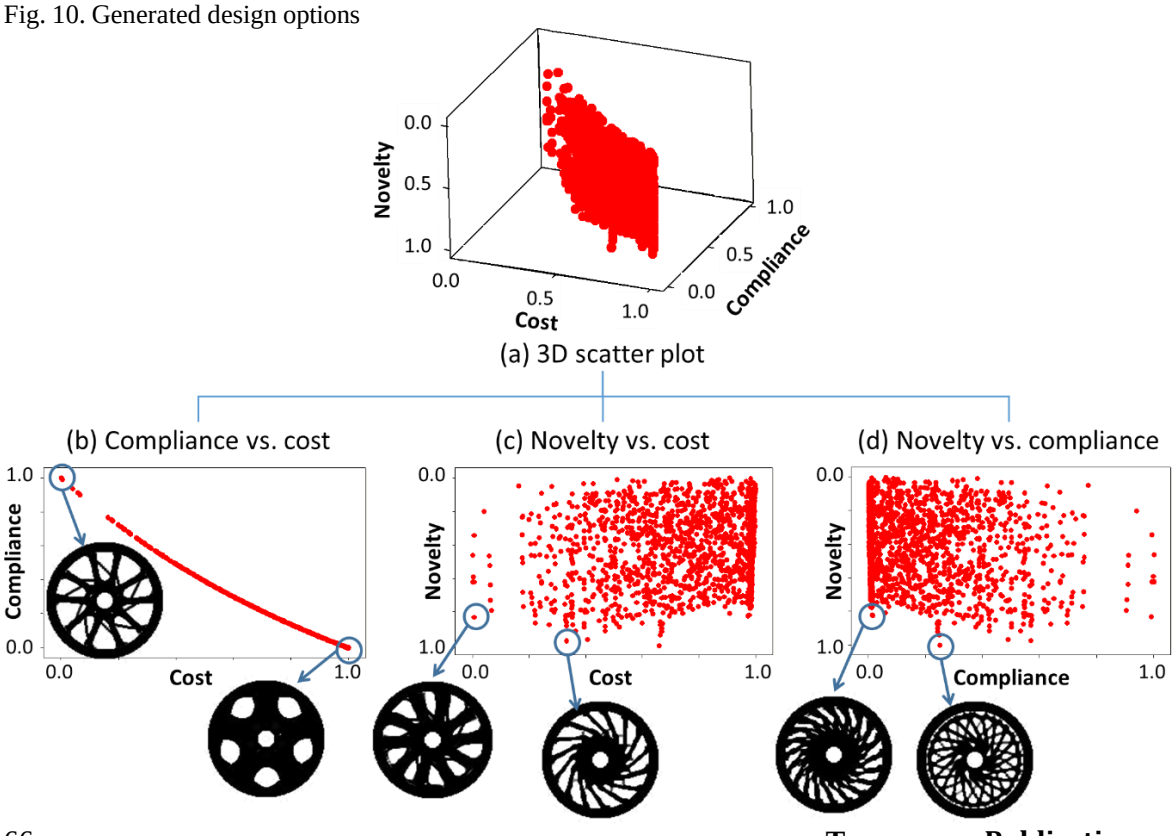


Fig. 11. Visualized design options by three attributes: novelty, cost, and compliance

Designers can trade-off three attributes, and select designs according to their design purpose and preference. Then, they can create a 3D design based on a 2D design and prototype it by 3D printing. Fig. 12 shows an example of a 3D wheel design (i.e., STL file for 3D printing) after selecting a 2D design.



Fig. 12. An example of a 3D wheel design using the selected 2D design

Discussion

This section analyzes and discusses performance and necessity of the main methods used in the proposed framework.

Topology Optimization

As an additional analysis, to check necessity of the reference designs, we conducted topology optimization without reference designs as shown in Fig. 13. Results without the reference designs as the benchmark would converge to an identical optimum if there is no change on boundary conditions. So it cannot yield aesthetical diversity. Also, it sometimes fails to converge, when it starts from uniform density and has no shear force since the displacement caused by the normal force exerted on the surface is almost uniform (see Fig 13(a), (b)). Therefore, the reference designs can enhance the diversity of designs while achieving convergence.



(a) (b) (c) (d) (e)

Fig. 13. Topology optimization without reference designs when the force ratio is set to (a) 0, (b) 0.1, (c) 0.2, (d) 0.3, (e) 0.4

In addition, topology optimization can theoretically generate infinite designs without the help from generative models, when topology optimization results are used as reference designs for topology optimization in the next iteration. We used the topology optimization results of Fig. 4 (b), (c), (d), and (e) as reference designs for the next topology optimization. Fig. 14 shows the selected results which were most different from the reference designs. The results show that iterative topology optimization generates similar designs since their original parent (reference design) is the same. This can be an empirical evidence that reference designs with fundamentally different topologies are needed to obtain a diversity of generated designs, and generative models make this possible. Therefore, we do not use topology optimization results as reference designs in the proposed process.

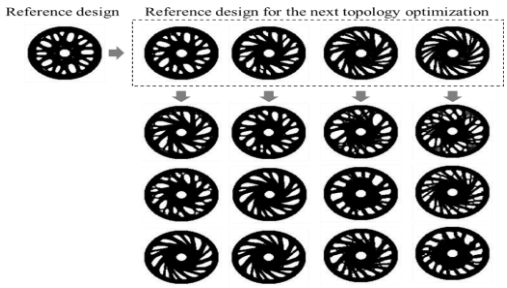
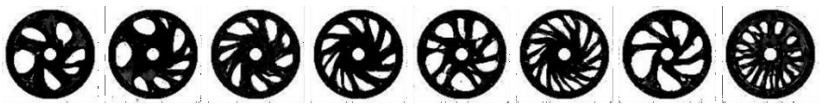


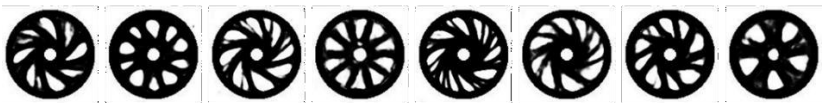
Fig. 14. Iterative design exploration by topology optimization only

BEGAN

One of the main problems of GAN is that there is no standardized method of measuring model performance. The following are some ways. First, one can check in person if generated data looks reasonable. Second, convergence criteria can be checked as shown in Fig. 8. Third, we can compute the difference between real data and generated data in feature space, for example, by using Inception Score (IS) or Fréchet Inception Distance (FID) [52]. In our proposed method, we take the second approach and check the global convergence for model validation. One advantage of our framework is that generated designs by BEGAN are not used as final output but as input for topology optimization. Therefore, even though the performance of BEGAN is slightly low, the framework can still work robustly.
Other than BEGAN, we additionally tested other generative models such as DCGAN and VAE which were introduced in Section 5.1. In our experiments, DCGAN and VAE display similar designs as shown in Fig. 15. They appear to produce more detailed and complex shapes than BEGAN, but they are blurrier and less symmetrical. In addition, the generated designs lack novelty in that the designs are more similar to those in the train data. We also tested these results as reference designs for topology optimization, but the results produced are less diverse.



(a) Wheel Design by DCGAN



(b) Wheel Design by VAE

Fig. 15. Example of generated designs by DCGAN and VAE

In sum, we acquire some empirical insights for utilizing generative models in design exploration. First, BEGAN is a good choice for generating reference designs, because it tends to create topologically novel, yet simple, designs. Detailed shapes of reference designs used in topology optimization fail to generate diverse designs. In fact, CAE softwares also have a feature that simplifies CAD models before conducting topology optimization (e.g., filling up holes). In future research, we plan to train a network that selects only the “good” reference designs from the results of multiple generative models. This is because DCGAN and VAE could also have some reference designs that BEGAN cannot create. Second, designers have to

try different latent space dimensions and epochs. In our study, we varied dimensions for latent variables (i.e., 16, 32, 64, and 128), saved the models at different epochs, and ultimately obtained data from many variations of the model. It is not practical to select only one latent dimension or epochs because each model generates unique designs.

Autoencoder

To validate the performance of autoencoder more quantitatively, we tested how much the model can classify previous designs and generated designs, assuming generated designs have more novelty than previous designs. We select 131 test data set (20% of 658 previous designs) for previous design and generated design, respectively, and create a confusion matrix as shown in Fig. 16. The autoencoder calculates a reconstruction error for 262 test designs, and we sort them by error size. We classify the top 50% of the designs as generated designs, and the bottom 50% as previous designs. In FP and FN cases which represent the designs that were misclassified, we see that it is not easy to distinguish between previous and generated designs even by human eye. As measuring criteria, both precision and recall are 91.6%.

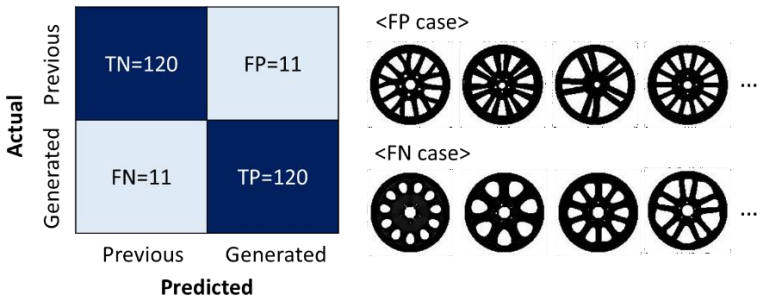


Fig. 16. Confusion matrix for autoencoder

Conclusion

A 2D wheel design case study illustrates the efficacy of this research's proposed design automation system, which produces a variety of designs guaranteeing engineering performance and aesthetics. This study's contribution may be discussed as follows: First, this study takes both engineering performance and aesthetics into account at the same time. As a multi-objective function, the suggested framework may regulate engineering performance and similarity with reference designs. Second, from a limited number of designs, a big number of designs were produced. An iterative procedure is suggested in which the generative models' output designs are used as reference designs for topology optimization once again after topology optimization is carried out to provide training data for the models. Third, compared to the traditional generative design, the suggested framework provided a variety of designs. Furthermore, the utilization of reference designs produced by generative models is responsible for the increasing variety. Fourth, there is an improvement in the resilience of design quality. Conventional generative models have a tendency to cause mode collapse and significant quality variation. However, rather of being directly used (e.g., post-processing), the outcomes of the generative model in the suggested framework are refined via topology optimization. Lastly, it is possible to assess how innovative the developed design is in relation to earlier designs.The index of resemblance to current designs is derived from the autoencoder's reconstruction error. The primary disadvantage of this study is that it uses pixel-wise graphics and a 2D design space. Therefore, several case studies should be examined and a 3D design application utilizing voxel data should be further studied for practical design. Additionally, a recommendation system that predicts designers' and consumers' preferences among design choices will be implemented in order to offer the best designs.

References

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521 (7553), 436.
[2] Deep Learning | The MIT Press, Cambridge, Massachusetts, Goodfellow, I., Bengio, Y., & Courville, A. (2016).
[3] Oseledets, I., and Sosnovik, I. (2017). Neural networks for optimizing topology. arXiv preprint arXiv:1709.09578.
[4] Yu, Y., Hur, T., Jung, J., & Jang, I. G. (2018). Using deep learning, a near-optimal topological design may be found without any iterations. [5] Banga, S., Gehani, H., Bhilare, S., Patel, S., & Kara, L. (2018). 3D Topology Optimization using Convolutional Neural Networks. arXiv preprint arXiv:1808.07440. Structural and Multidisciplinary Optimization, 1–13.
[6] Guo, T., Cang, R., Ren, M. Y., Lohan, D. J., & Allison, J. T. (2018). A style transfer and variational autoencoder-based indirect design representation for topology optimization. AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018 (p. 0804).
[7] Yao, H., Ren, Y., and Cang, R. (2019). Theory-driven machine learning to generate near-optimal topology in a single shot. 109, 12-21; Computer-Aided Design.
[8] Burnap, A., Liu, Y., Pan, Y., Lee, H., Gonzalez, R., & Papalambros, P. Y. (2016). using sophisticated generative models to estimate and investigate the product shape design space. ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pages. V02AT03A013-V02AT03A013). American Society of Mechanical Engineers.
[9] Umetani, N. (2017). investigating autoencoder networks for generating 3D forms. SIGGRAPH Asia 2017 Technical Briefs, page 24. ACM. [10] Guo, X., Li, W., & Iorio, F. (2016). Convolutional neural networks for approximating steady flow. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Proceedings (pp. 481-490).
[11] Tompson, J., Schlachter, K., Sprechmann, P., & Perlin, K. (2017) ACM. Using convolutional networks to speed up Eulerian fluid simulation. 3424–3433 in Proceedings of the 34th International Conference on Machine Learning, Volume 70. JMLR.
[12] Deep learning the physics of transport phenomena, Farimani, A. B., Gomes, J., & Pande, V. S. (2017). arXiv preprint arXiv:1709.02432.
[13] Cang, R., Xu, Y., Chen, S., Liu, Y., Jiao, Y., & Ren, M. Y. (2017). For computational material design, deep belief networks are used to represent and recreate the microstructure of heterogeneous materials. Mechanical Design Journal, 139(7), 071404.
[14] Yang, Z., Li, X., Agrawal, A., Chen, W., Choudhary, A. N., Brinson, L. C., & Agrawal, A. (2018). Deep adversarial learning approach for microstructural materials design. Mechanical Design Journal, 140(11), 111416.
[15] Cang, R., Li, H., Yao, H., Jiao, Y., & Ren, Y. (2018). enhancing the direct physical property prediction of heterogeneous materials from image data using a generative model that takes morphology into account and a convolutional neural network. 150, 212-221; Computational Materials Science.
[16] Burnap, A., Pan, Y., Liu, Y., Ren, Y., Lee, H., Gonzalez, R., & Papalambros, P. Y. (2016). using feature learning to increase the accuracy of design preference prediction. Mechanical Design Journal, 138(7), 071404.
[17] Pan, Y., Burnap, A., Gonzalez, R., Hartley, J., & Papalambros, P. Y. (2017). Deep design: Aesthetics of products for diverse markets. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Proceedings (pp. 1961-1970). ACM. [18] Shea, K., Gourtovaia, M., & Aish, R. (2005). Moving toward integrated generative design tools that are performance-driven. Construction Automation, 14(2), 253-264.
[19] Krish, S. (2011). A useful approach to generative design. 88–100 in Computer-Aided

Design, 43(1).
[20] M. McKnight (2017). What is generative design? How is it put to use? Why it's revolutionary. KnE Engineering, 2(2), 176-181.
[21] Matejka, J., Glueck, M., Bradner, E., Hashemi, A., Grossman, T., & Fitzmaurice, G. (2018). Dream Lens: Investigating and Displaying Extensive Generative Design Datasets. The 2018 CHI Conference on Human Factors in Computing Systems Proceedings (p. 369). ACM. [22] Autodesk (2019), Generative Design, https://www.autodesk.com/solutions/generative-design
[23] Oh, S., Jung, Y., Lee, I., & Kang, N. (2018). Design Automation using Topology Optimization and Generative Adversarial Network Integration. ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pages. V02AT03A008-V02AT03A008). American Society of Mechanical Engineers.
[24] Kang, N. (2014). Multidomain Demand Modeling in Market System Design. University of Michigan PhD thesis.
[25] Tucker, C. S., and Dering, M. L. (2017). A Convolutional Neural Network Model for Predicting a Product's Function, Given Its Form. Mechanical Design Journal, 139(11), 111408.
[26] Tucker, C. S., and Dering, M. L. (2017). Using generative adversarial networks, huge data may be more reliable. [27] Bendsøe, M. P., & Kikuchi, N. (1988) 2017 IEEE International Conference on Big Data (Big Data) (pp. 2595-2602). IEEE. using a homogenization technique to create ideal topologies for structural design. Computer techniques in engineering and applied mechanics, 71(2), 197-224.
[28] M. P. Bendsøe (1989). Designing the ideal form as a material distribution issue. [29] Bendsoe, M. P., & Sigmund, O. (2013) Structural optimization, 1(4), 193-202. Theory, techniques, and applications of topology optimization: Springer Science & Business Media.
[30] Sigmund, O. (2001). A Matlab topology optimization algorithm of 99 lines. Multidisciplinary and structural optimization, 21(2), 120-127.
[31] Sigmund, O. (2007). Black and white filters for topology optimization based on morphology. 33(4-5), 401-424; Structural and Multidisciplinary Optimization.
[32] Lazarov, B. S., Sigmund, O., Clausen, A., Schevenels, M., and Andreassen, E. (2011). 88 lines of code are needed for effective topology optimization in MATLAB. 43(1), 1–16, Structural and Multidisciplinary Optimization.
[33] Topology optimization techniques, Sigmund, O., & Maute, K. (2013). 48(6), 1031-1055, Structural and Multidisciplinary Optimization.
[34] Sigmund, O. (1997). on employing topology optimization to create compliant mechanisms. Structural Mechanics Journal, 25(4), 493-524.
[35] Tortorelli, D. A., and Bruns, T. E. (2001). Topology optimization of compliant mechanisms and non-linear elastic structures. Computer techniques in applied engineering and mechanics, 190(26-27), 3443-3459.
[36] Bourdin, B. (2001). Topology optimization filters. International Journal of Engineering Numerical Methods, 50(9), 2143-2158.
[37] Belytschko, T., Prévost, J. H., and Guest, J. K. (2004). using projection functions and nodal design variables to achieve the minimal length scale in topology optimization. International Journal of Engineering Numerical Methods, 61(2), 238-254.
[38] Xu, S., Cheng, G., and Cai, Y. (2010). Heaviside-based nonlinear density filter that preserves volume. 41(4), 495-505; Structural and Multidisciplinary Optimization.
[39] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). adversarial networks that generate. Neural Information Processing Systems Advances (pp. 2672-2680).
[40] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. arXiv preprint arXiv:1701.07875.
[41] Radford, A., Chintala, S., and Metz, L. (2015). Deep convolutional generative adversarial

networks for unsupervised representation learning. arXiv preprint arXiv:1511.06434.
[42] Adversarially acquired inference by Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., & Courville, A. (2016). Preprint arXiv:1606.00704.
[43] Adversarial feature learning by Donahue, J., Krähenbühl, P., and Darrell, T. (2016). Preprint arXiv:1605.09782.
[44] Zhao, J., LeCun, Y., and Mathieu, M. (2016). generative adversarial network based on energy. arXiv preprint arXiv:1609.03126.
[45] Berthelot, D., Metz, L., and Schumm, T. (2017). Boundary equilibrium generative adversarial networks were the first. The preprint arXiv is arXiv:1703.10717.