Deep Generative Models in Engineering Design:
A Review

Lyle Regenwetter
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: regenwet@mit.edu

Amin Heyrani Nobari
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: ahnobari@mit.edu

Faez Ahmed
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: faez@mit.edu

Automated design synthesis has the potential to revolutionize the modern engineering design process and improve access to highly optimized and customized products across countless industries. Successfully adapting generative Machine Learning to design engineering may enable such automated design synthesis and is a research subject of great importance. We present a review and analysis of Deep Generative Machine Learning models in engineering design. Deep Generative Models (DGMs) typically leverage deep networks to learn from an input dataset and synthesize new designs. Recently, DGMs such as feedforward Neural Networks (NNs), Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and certain Deep Reinforcement Learning (DRL) frameworks have shown promising results in design applications like structural optimization, materials design, and shape synthesis. The prevalence of DGMs in engineering design has skyrocketed since 2016. Anticipating continued growth, we conduct a review of recent advances with the hope of benefitting researchers interested in DGMs for design. We structure our review as an exposition of the algorithms, datasets, representation methods, and applications commonly used in the current literature. In particular, we discuss key works that have introduced new techniques and methods in DGMs, successfully applied DGMs to a design-related domain, or directly supported the development of DGMs through datasets or auxiliary methods. We further identify key challenges and limitations currently seen in DGMs across design fields, such as design creativity, handling constraints and objectives, and modeling both form and functional performance simultaneously. In our discussion, we identify possible solution pathways as key areas on which to target future work.

1 Introduction

The human design process is a ubiquitous element of modern society, playing a critical role in the technologies producing the food we eat, the products we use, and the spaces in which we live. Accelerating the design process through automation can reduce cost and increase industrial productivity, which would be immensely desirable for global productivity and prosperity. Successfully integrating AI into the design process can alleviate a dependence on human experts and can revolutionize user customizability, providing optimal products for individual users without the prohibitive cost of manual design. Driven by the widespread potential to advance global equity and prosperity through design automation, "generative design" methods emerged as computing and automation technologies took root.

Early "generative design" methods leveraged computational tools to pioneer a variety of different approaches for design synthesis. Many of these approaches leveraged human design expertise in grammar and rule-based frameworks [1]. As availability of computational resources increased over the past decade, data-intensive methods like deep learning opened doors to successfully automate complex human tasks such as image processing and natural language processing. In deep learning, data is propagated through sequential layers to learn progressively higher-level meaning, an architecture generally known as an Artificial Neural Network (ANN) or just Neural Network (NN) [2]. Most of the deep learning-based approaches pioneered during the 2010s leveraged extensive quantities of data to avoid explicit functional programming. This trend is mirrored in Engineering Design with algorithms learning data distributions instead of requiring them to be pre-defined. Among these algorithms are Deep Generative Models (DGMs). In this review paper, we specifically define "Deep Generative

Models" as algorithms that generate new samples using deep learning. Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) are two classes of DGMs that have demonstrated compelling synthesis of images, text, and tabular data in numerous domains. Considering that images, text, and tabular data are all common representation methods for design, one might assume that DGMs should be capable of synthesizing full designs as well with relative ease. However, several unique properties of the generative design task pose particular challenges for DGMs. Many of these challenges are so fundamental that the future success of DGMs in engineering design is largely contingent on the ability to overcome these challenges. We list four of these challenges below:

1. Data sparsity: Compared to other research fields like Computer Vision, which have massive publicly available datasets, the availability of large, well-annotated, public datasets in engineering is severely lacking. Furthermore, even when data is available, the distribution of the data often does not cover the design space evenly with much sparsity often observed in the data.

2. The creativity gap: In conventional DGM applications, the overarching goal is to mimic the training data and emulate existing designs. In engineering design, emulation of existing products is often undesirable. Designers typically aim to introduce products with novel features to target new market segments.

3. Modeling design performance: Real world functional performance is critical in many engineering design tasks. Developing performance-aware DGMs capable of synthesizing designs for a given set of target requirements (a process termed as inverse design) is a challenging task that is exacerbated by the computational cost of numerical simulation and the even greater difficulty of real world evaluation.

4. Usability and feasibility: For synthesized designs to be physically fabricated, they must be physically feasible. Furthermore, designs must be encoded in a data representation that contains significant parametric detail and can be converted to a representation usable for fabrication.

Over the past few years, the design community has made substantial progress in using generative machine learning models to create new designs. DGMs have been applied to a broad range of design tasks such as structural optimization, materials design, and shape synthesis. Over the years, researchers have introduced increasingly advanced methods, which have begun to address some of the above challenges. For example, many works have proposed approaches to incorporate design performance and optimization into DGM training. Other works have explored incorporating novelty and creativity into DGMs. Despite these advancements, DGMs for engineering design are still in their infancy and will require further efforts to effectively overcome these fundamental challenges.

Our primary goal in this work is to help build cohesion between the countless active researchers in the design field working with DGMs and furthermore provide a starting point for researchers entering the field. In particular, we seek to provide researchers a reference guide in planning projects in the data-driven generative design space. To this end, we provide an overview of common methods and tools (Sec. 2), a discussion of different data parameterization methods (Sec. 3), a review of potentially relevant research across various design domains (Sec. 5), an overview of relevant datasets (Sec. 6), and an analysis of common challenges in the field (Sec. 7). Figure 5 provides an overview of the standard process to apply DGMs in engineering design.

2 Overview of Deep Generative Models

Deep Generative Machine Learning approaches share the goal of high quality synthesis but significantly vary in methodology. In practice, we identify four common approaches in generating new samples: direct generation using deep neural networks (DNN), adversarial generation with Generative Adversarial Networks (GAN), generation from embedding vectors using Variational Autoencoders (VAE), and sequential generation using Reinforcement learning (RL). In the design community we observe that VAEs, GANs, and RL are used most commonly for design synthesis. While DNNs as well as extensions like recurrent DNNs (RNNs) are frequently used for non-generative tasks, they are occasionally used for direct design synthesis. In this section, we briefly discuss the background and methodology of GANs, VAEs and RL.

2.1 Generative Adversarial Networks

Originally introduced in 2014, Generative Adversarial Networks (GANs) [3] found initial success with convincing image synthesis performance [4, 5, 6, 7]. We provide an introduction of GANs, but refer the reader to [8] for a detailed overview. A generative adversarial network [3] consists of two models — a *generator* and a *discriminator*. The generator G maps an arbitrary noise distribution to the data distribution, in our case the distribution of designs, and can thus generate new data; simultaneously, the discriminator D learns to distinguish between real and generated data. Both models are usually built with deep neural networks. As D improves, G also improves as it learns to generate data that fools D.

GAN Training GANs are notoriously difficult to train, suffering from training instability stemming from a number of sources [9, 10]. One common issue in GAN training occurs when the discriminator overpowers the generator, easily distinguishing generated samples, and causing the gradient to the generator to vanish, effectively halting the generator's training. This issue has been addressed in many follow-up works. For example, the WGAN replaces the discriminator with a critic, modifies the loss function of the GAN to estimate the Wasserstein (Earth Mover's) distance between the distribution of original data and the distribution of generated data, and modifies the training process [11, 12].

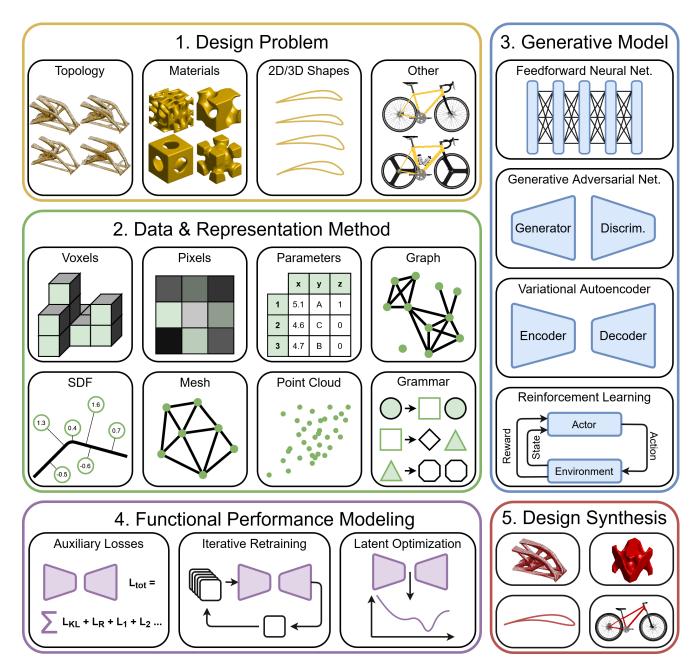Another problem that GANs face is the issue of "mode

Fig. 1: This figure outlines the typical application process for Deep Generative Models. We divide our review (Sec. 5) into papers discussing topology generation, material design, 2D shape synthesis, 3D object generation, and other applications. To set some context before our review, we discuss the various design data representations (Sec. 3) used in DGMs and the various DGMs architectures (Sec. 2) frequently used in current research. After our analysis, we discuss some of the key findings from the review, including a discussion of methods that researchers have attempted to incorporate performance optimization into DGMs (Sec. 7.1).

collapse," where the generator fails to encompass all modes in the data distribution or even generates only a handful of unique samples that are capable of fooling the discriminator. To overcome these issues, researchers have developed novel algorithmic techniques [13, 14, 15].

GAN Conditioning: In the design domain, we often have design constraints and objectives that we may seek to impart on our DGM. For example, we may seek to train a DGM to generate bikes, but depending on our user, we may want to constrain it to generate only roadbikes or mountain bikes without retraining for each generation task. Model conditioning is one method to do this. Several proposed approaches add conditioning to the GAN using a condition vector which is intended to be interpretable. Typically GANs are discretely conditioned by feeding the condition vector into both the generator and discriminator, in a configuration known as a Conditional GAN (cGAN) [16]. Instead of feeding the condition vector into the discriminator, an auxiliary

network and cross entropy loss can instead be used to reconstruct the condition vector from the generated samples in a configuration known as an Information Maximizing GAN (InfoGAN) [17]. Conditioning is also essential in design applications where inverse design is being done on performance metrics, which often exist in continuous spaces (*e.g.*, stiffness, lift coefficient, drag coefficient, density, *etc.*). Researchers have come up with continuous conditioning solutions for GANs such as Regressional GANs [18], continuous conditional GANs (CcGAN) [19] and performance conditioned diverse GANs (PcDGAN) [20].

2.2 Variational Autoencoders

Introduced in 2013, Variational Autoencoders found significant success in many machine learning applications. Autoencoders are unsupervised embedding algorithms consisting of an *encoder* that maps an input design into a (typically) lower dimensional latent space and a *decoder* that reconstructs the design as accurately as possible from the latent space. The encoder and decoder are conventionally implemented using deep neural networks. To generate new samples, latent vectors are sampled from the latent space and fed through the decoder. Typically, the distribution of the real data mapped to the latent space of an autoencoder is sparse, meaning that sampling a realistic latent vector is difficult. This limitation is addressed with the introduction of the Variational Autoencoder (VAE) first proposed by Kingma *et al.* [21]. The Variational Autoencoder adds in a probabilistic sampling in the latent space which regularizes the latent distribution. In practice, the VAEs's encoder outputs n means and n variances, from which n dimensional latent vectors are sampled before decoding. To maintain a predictable latent space distribution, The VAE adds a Kullback-Liebler (KL) divergence [22] loss between the distribution of the latent space and a standard Gaussian. Interested readers are encouraged to refer to the literature [23] for a more detailed overview.

Conditional VAEs Just as we did for GANs, we may also seek to condition VAE training on design constraints or user preferences. The VAE has a natural advantage over the GAN in that its latent space is typically already structured (though this structure may be fairly weak). Despite this, explicitly conditioning VAEs may still be desirable. The Conditional VAE (cVAE) [24] extends on the conventional VAE by adding a conditioning vector as an input to both the encoder and decoder and helps achieve this goal.

2.3 Reinforcement Learning

Reinforcement Learning fundamentally differs from the other DGMs discussed in that it learns without a dataset in an unsupervised fashion through a large set of trial and error interactions between an *actor* and an *environment* [25, 26]. This is typically done through some reward signal being sent to the actor after taking actions, based on the effects of said actions on the environment. In this scenario, the actor's goal is to maximize the rewards it receives by making decisions (*i.e.*, taking actions) such that the total reward is maximized. From this point of view, reinforcement learning can be thought of as an approach similar to optimization, where an objective (maximizing the reward) is being optimized.

One of the first attempts at introducing deep learning to the reinforcement learning approach was done in 2013 by Mnih *et al.*, when they introduced deep learning to a reinforcement learning process known as Q-Learning [27]. Q-Learning refers to learning the state-action value function or Q-function, which is a progressively updated estimate of the expected reward to be received from taking a particular action in a particular state. Mnih *et al.*, attempted to learn the Q-function using convolutional neural networks (CNN). Many Deep RL techniques have been introduced since this first work by Mnih *et al.*. Further exploration of the details of these approaches is left to the reader.

In practice, when applying RL to design applications, the design process is usually broken down into a sequential process of building a design or altering existing designs in steps (*i.e.*, actions taken to alter or expand the current state of a design being generated) and the reward is measured by the quality or performance of the resulting design (*i.e.*, the environment). While RL requires no dataset, this advantage is balanced by a dependence on meaningful and reliable reward signals, which may often require a high-fidelity simulation environment. One major benefit of RL over GANs and VAEs is the fact that the reward function can be set based on any objective which does not need to be differentiable. In contrast, any objective added to the loss function of a GAN or VAE must be differentiable since GANs and VAEs are trained using the gradient-based optimization [15].

3 Overview of Design Representation Methods

In this section, we discuss common design representation methods seen in DGMs for engineering design. We include a definition, general discussion, as well as pros and cons of each method.

3.1 Images

Design data often comes in the form of images (*e.g.* microstructure scans) or can be represented in an image form (*e.g.* topology optimization). An image consists of a rectangular grid of pixels, each of which contains a color parameter. They are commonly represented by third order tensors (*height* × *width* × *channels*). Common color schemes are black-and-white (boolean color channel), grayscale (integer color channel), and color (3-4 integer color channels). **Pros:** The image is an information-rich representation and can capture many details of a design. The use of convolution/convolution-transpose filters in deep learning provide a convenient tool for learning/generation of both high level and low level features as well as upsampling/downsampling. Many cutting edge ML techniques are pioneered in the computer vision domain, and are often directly applicable to images. **Cons:** Representing designs using pixels means that the generated design images can be

infeasible for downstream tasks. Accurately fabricating designs based on images can be difficult or impossible. Even performance evaluation using conventional simulation tools like FEA or CFD can require an intermediate conversion from an image to a 3D model. The poor usability of images is exacerbated by the prevalence of artifacts in many applications (hanging pixels, disconnected geometry, *etc.*). Artifacts are especially common when training on (typically) small datasets in the design domain, since training is often terminated early due to over-fitting concerns. All in all, images can be considered surrogate representations of engineering designs and may lack domain knowledge and information on the physical realization of the design. Therefore, DGMs using images as representations often have a gap between the generated images and the actual design they are representing.

3.2 Voxelizations

Voxels are 3D grid points that are effectively the 3D equivalent of pixels. As such, voxelizations share many characteristics with images. In practice, voxels aren't conducive to 'color' parameterization and are typically represented as booleans (space vs. object). This effectively makes them third order tensors with dimension ($height \times width \times depth$). **Pros:** Voxelizations support 3D convolution which can learn high level and low level features in 3D. **Cons:** Compared to images and other representations, the curse of dimensionality is especially pronounced with voxels, with the number of parameters scaling with the cube of spatial resolution. Voxelizations share the same issues as images. Their usability is limited in downstream tasks and artifacts are very prevalent. Like images, voxelixations serve as surrogate representations (often representing CAD models which originate from parametric representations or 3D shapes which originate from meshes). Like many other representations such as point clouds and Signed Distance Fields, voxelizations often require conversion before they can be used in downstream tasks. For example, they are often converted to Boundary Representation (BRep) or polygonal representations, which are often the native parameterizations of rendering and graphics software, Finite Element Analysis, and Computational Fluid Dynamics simulation.

3.3 Point Clouds

Point Clouds are simple collections of points, often in 3D space, which are defined to be within some object. **Pros:** Point Clouds can represent arbitrarily complex geometry with a finite number of points, though fidelity may vary. Point Clouds are often the native output of 3D scanning software, making them relatively easy to create [28]. **Cons:** Like Voxelizations and Signed Distance Functions, Point Clouds often require conversion to BRep or polygonal representations such as meshes [29] for downstream tasks.

3.4 Meshes

Meshes are a common method to represent objects in 3D space. Triangular meshes are by far the most commonly-used form. Triangular meshes are the native representation used in many computer graphics algorithms and software, as well as many Finite Element tools. **Pros:** Meshes can be directly visualized and simulated in many FEA or CFD tools, enabling easy pipelines for performance evaluation using numerical methods. A mesh can be considered a specialized type of graph and can leverage graph operators like graph convolutional operators. **Cons:** In contrast to other representations like voxelizations and point clouds, meshes are more challenging to directly generate using Machine Learning methods, despite recent advances in algorithms that directly generate meshes [30, 31, 32].

3.5 Signed Distance Functions

The Signed Distance Function/Field (SDF) is a parameterization method which consists of a (typically 3D) functional map from a coordinate point to SDF value. The magnitude of this value indicates the distance to the nearest point on the surface of the object and the sign indicates whether the point is inside or outside the object. SDFs themselves can be parametrized in many ways, for example, as a rasterized representation in which each 'voxel' contains a continuous numerical value denoting the SDF value at that point. **Pros:** SDFs can serve as a convenient intermediate parameterization for many learning tasks. **Cons:** Like point clouds or voxels, SDFs are difficult to use in downstream tasks without first converting to BRep or polygonal representations.

3.6 Parametric

We use the term "parametric" data to represent any design representation consisting of a collection of design parameters where any spatial or temporal significance of parameters is unknown. Most parametric data can be organized in tabular form with each row being a collection of parameters representing a single design and each column describing a design parameter. Tabular design data often consists of a collection of mixed-datatype parameters where relations between these parameters may be unclear or nonexistent. Since parametric data may come in many varieties, the pros and cons discussed may not be applicable to every case. **Pros:** Quality parametric data is typically very information-dense (i.e. requiring fewer parameters to encode the same level of geometric detail). Whereas spatially-organized representations such as pixels or voxels encode designs with uniform information density, parametric data can contain more detail in design-critical areas without the need for upsampling the entire representation. This information density often comes with a lower dimensionality which can make optimization of parametrically represented designs significantly easier. Parametric data may also be more supportive of downstream tasks, especially if design parameters are human-interpretable. For example, a detailed enough design parameterization may allow generated designs to be directly fabricated using conventional (non-additive) manufacturing techniques. Human-interpretable parameterizations can also give human designers a tractable method to interact with generative methods to allow for human-in-loop design. Finally,

design parameters can sometimes be directly linked to the latent space of a generative method, as demonstrated in numerous works [33, 34], creating a pipeline to directly condition design generation on high level design goals. Linking design parameters with latent variables has several potential advantages, such as enabling more effective optimization or inverse design using generative methods. **Cons:** Learning parametric data can be particularly challenging. Parametric data commonly uses mixed datatypes and inherits the training challenges of the constituent components. Multimodal distributions, skewed categories, non-Gaussian distributions, data sparsity, and poor data scaling additionally make the application of DGMs and training very difficult. Since methods that are robust to all of the above challenges are difficult to come by, successfully applying existing methods to the parametric data domain can be hard. Finally, since parametric data may be nontrivial to convert to 3D models, generated parametric designs may be challenging to evaluate using numerical simulations or through qualitative visualization.

3.7 Grammars

Grammars are representation methods consisting of variables, terminal symbols, nonterminal symbols, and a set of rules. Rules describe how non-terminal symbols can expand into other terminal and nonterminal symbols. The most prevalent grammars in engineering design are graph and spatial grammars [1], and they have been applied in a wide variety of applications, such as design of satellites and electromechanical systems. Grammars can be especially useful to dictate feasible assembly hierarchies of design components as in [35]. **Pros:** Grammars can explicitly constrain design spaces to feasible or desirable regions by nature of their construction thereby encoding domain knowledge. **Cons:** Grammars are challenging to implicitly learn and often must be manually defined. In a few cases, grammars can restrict exploration of the entire design space. A survey on grammar-based design synthesis approaches is provided in [1].

3.8 Graphs

Graphs are a highly flexible representation method consisting of nodes and edges, which can be directed or undirected. Graph-based representations have proven successful for many different aspects of design generation and optimization and provided avenues for describing complex systems efficiently. **Pros:** Graphs are highly adaptable and are capable of representing many different kinds of complex systems and designs [36, 37, 38, 39]. Graphs also provide representations for design processes and modeling complex interactions in systems [40, 41] which may enable methods for automating the design of systems or modelling interpart dependencies. Graph neural networks (GNNs) [42, 43, 44, 45, 46] provide an excellent tool for machine learning on graphs. **Cons:** Despite the developments of graph neural networks (GNNs) in the computer science community [47, 48, 49, 50] and their success in molecular graph generation [51, 52], there is less usage of graph-based DGMs in the design community, possibly due to the lack of graph-based design datasets.

4 Literature Review Methodology

Sec. 5 discusses specific works that apply Deep Generative Models to engineering design or make advancements to existing generative ML methods in the context of engineering design. We consider works based on a pre-defined scope, with each work we discuss meeting the following selection criteria. Note that these criteria only apply to the engineering design papers presented and that the works we cite to add context to or substantiate the discussion of fundamentals, applications, and datasets need not adhere to these rules.

1. We limit our consideration specifically to papers involving Deep Generative Models, focusing on Variational Autoencoders, Generative Adversarial Networks, and Reinforcement Learning in particular. Works must utilize deep learning. Works only considering design optimization are excluded.
2. We only consider work specifically relevant to engineering design and not other domains (such as computer science).
3. We consider only work published between Jan. 2014 and Sep. 2021, when we conclude our review, as many pivotal works in deep learning (CNNs, VAEs, GANs) were introduced in this period.

To identify works, we specifically searched for "Generative Adversarial Network," "Variational Autoencoder," and "Reinforcement Learning" in Google Scholar. We initially confined our search to a set of known design venues, specifically the Journal of Mechanical Design, the Proceedings of the International Design Engineering Technical Conferences, Computer-Aided Design Journal, International Conference on Engineering Design, and Artificial Intelligence for Engineering Design, Analysis, and Manufacturing Journal. This helped us identify an initial set of seed papers. From all search results identified through these methods, 41 papers were deemed to be relevant and included. The relevance was decided by independent assessment by two raters, who are also authors on this paper. For papers where there was a disagreement, all authors discussed them and mutually decided on their classification. Next, papers cited in these seed papers were considered and added to this paper. 22 papers from other venues were deemed to be relevant and included as well. Of the 63 papers, 48 are published between Jan. 2019 and Aug. 2021, while a mere 15 are published between Jan. 2016 and Dec. 2018, indicating a strong growth in the field in recent years.

5 Application Domains in Engineering Design

Engineering design encompasses a wide variety of applications, ranging from designing aircraft models to small-scale metamaterials. To structure different types of applications using DGMs, we grouped them into a few categories of application areas, which are discussed in this section and are reported in Table 1.

Copyright © by ASME

Table 1: Characterization of application studies by domain and architecture. Some works use multiple architectures or fit into multiple application domains and are listed more than once. We additionally classify works by datatype used: Image[i], Point Cloud[c], Voxel[v] Mesh[m], Signed Distance Field (SDF)[s], Grammar[g], Graphs[h], and Parametric[p] (other parameterization)

	Topology Optimization	Materials	2D Shape Synthesis	3D shape synthesis	Other Domains
Plain NN	[53]^i [54]^{vi} [55]^{p*} [56]^i	[57]^p [58]^i [59]^i			
GAN	[60]^i [61]^i [62]^i [63]^i	[64] [65]^i [66]^{vi} [67]^v [68]^i [69]^{vi}		[70]^c	
GAN+Conditioning	[71]^i [72]^i [73]^i [74]^i		[75]^p [76]^p [77]^p [34]^{ph}		
New GAN-based	[73]^i		[15]^p	[78]^v	[79]^i
AE/VAE	[80]^i	[81]^i [82]^i [83]^i [33]^i [84]^i		[85]^v	[86]^p, [87]^p, [88]^p
cVAE	[89]^{i†} [90]^i				[86]^p [91]^i
New VAE-based		[92]^i		[93]^s	[94]^{ip}
RL-based			[95]^p [96]^p		[97]^{ip} [35]^g [98]^{ip} [99]^p [100]^p
Other Method	[101]^v [102]^i	[103]^i [104]^i [68]^i [105]^i [106]^i [107]^i	[96]^p		[101]^v [108]^p [35]^g [109]^{ip} [110]^{ip} [111]^g
+Style-Transfer	[80]^i	[65]^i [66]^{vi} [81]^i [58]^i [105]^i			
+Genetic Alg.		[83]^i [106]^i		[93]^s	
+Bayesian Opt.		[65]^i [84]^i	[77]^p		

*Model-based reconstruction using parameters in a lower dimensional space
† Technically not images, but use an image-like parameterization structure

5.1 Deep Generative Models in Topology Optimization

Fig. 2: Sample topologies generated by 3D Topology Optimization

Topology Optimization (TO) is a research field with a long history of research and methods. TO searches a design space to find an ideal spatial distribution of material to optimize some predefined objective. Common areas of application include solid mechanics [112, 113], fluid dynamics [114, 115], additive manufacturing [116, 117], and heat transfer [118, 119]. While topology optimization is considered a method for generative design, standard TO does not leverage deep learning and is not considered a DGM. In recent years, however, several papers have proposed methods to use TO and DGMs together, in many cases to address the computational cost of TO on large amounts of data. Despite differences between architectures, we found that many DGMs for TO share certain characteristics. In particular, due to the predominant use of voxelized or pixelized representations, many hybrid TO-Generative ML models use methods originally developed for computer vision applications, such as convolutional neural networks and super-resolution [60, 54].

Supervised Generation of Optimized Topologies: To avoid the computational cost of Topology Optimization, DGMs have been used to predict the final optimized topologies directly. This approach can also be used as an initialization technique for conventional TO, which allows conventional TO algorithms to rapidly converge, saving computational cost. We consider a baseline for DGMs in TO, in which an existing generative architecture is applied to a dataset of TO-generated designs and the network is trained to generate samples mimicking the training data. A few papers fall within this category of methods: Rawat and Chen [61] train a WGAN on a dataset generated through TO and train an auxiliary network to additionally predict performance metrics. Sharpe and Seepersad [71] expand on this baseline with a cGAN conditioned on volume fraction and load location. Guo *et al.* [80] train a VAE with an additional style-transfer [120, 121] loss on a TO-generated dataset for heat transfer and propose iterative strategies for targeted design optimization using the VAE's latent space. Style-transfer is discussed further in Sec. 5.2.

Iterative DGM training, filtering, and human guidance in TO: Oh *et al.* [63, 62] propose a method that expands on the baseline generative-network-fitting by iteratively synthesizing new designs, optimizing them in TO, then dropping designs that are too similar from the full collection of designs. The authors use a modified BEGAN [122], which extends on the WGAN and begin training on a collection of existing TO-generated topologies. However, unlike the previously discussed generative-network-fitting approaches used in literature [61, 71, 80], the retraining and re-optimization allows the framework to generate, optimize, and explore new areas of the design space. The proposed method is applied to the problem of wheel design, with the key motivation being to find a trade-off between the aesthetics of real designs and the structural performance of designs found through TO. The authors present strong empirical results on this wheel design problem.

The issue of gaps in the design space of topologies was also addressed by Fujita *et al.* [102], who propose an approach leveraging a Variational Deep Embedding (VaDE) [123]. An initial dataset is generated using TO. The proposed method sequentially identifies voids in the design space using the VaDE, decodes designs from the void, optimizes them using TO, then adds them to the training dataset.

Although the above works have proposed methods with

automated retraining steps, human input can also be injected into the training process. For example, Valdez *et al.* [74] propose a cGAN-based human-in-loop topology design generation framework in which the designer iteratively selects design clusters to gradually hone in on preferable designs.

DGMs for TO that utilize super-resolution: Since one of the key limitations of Topology Optimization is computational cost, which scales with resolution, a major focus in DGMs for TO has focused on attaining high-resolution TO-like results. A common approach uses super-resolution, which is a technique used in computer vision to convert low-resolution images to high-resolution ones. Several works expand on the baseline DGM in TO framework by learning from a low-resolution TO-generated dataset, then performing super-resolution on synthesized topologies, such as Yu *et al.* [73], who add a cGAN for upscaling. A similar approach proposed by Li *et al.* [60] uses a Super-Resolution GAN (SRGAN) [124] to generate high-resolution optimal topologies for heat transfer problems after generating low-dimensional topologies on a different GAN. Other researchers have proposed transfer learning for the super-resolution task instead of using GANs or VAEs for super-resolution. For example, Behzadi *et al.* [54], train a feed-forward NN-based model to predict optimized topologies directly without any discriminator and using the MSE loss. Once they have trained their model on the low-resolution data, they lock the model weights and add a few layers to the model to increase the output resolution and only train said layers to obtain high-resolution samples, avoiding the need for a large quantity of high-resolution data or the extra time needed to train a high-resolution model from scratch. Many papers that have applied super-resolution techniques demonstrate that after initial training, their framework consistently generates near-optimal topologies many times faster than classic TO [73, 60].

Improving DGM-based Topology Generation using physical properties: Several papers have succeeded in improving the baseline performance of DGMs for topology generation through the use of physical properties of the design domain. For example, Nie *et al.* [72] propose to adapt the cGAN architecture Pix2Pix [125] to generate synthetic topologies based on spatial fields of various physical parameters (displacement, strain energy density, and Von Mises stress) as input to the generator. Topologies generated by TO are taken as ground truth for training. The authors also propose a new generator architecture combining the Squeeze and Excitation ResNet [126, 127] with U-Net [128]. Cang *et al.* [56] use a Neural Network to generate optimized topologies from loading conditions. They propose an approach to rapidly evaluate the deviation of proposed solutions from the problem's optimality conditions. They then progressively augment their dataset during training by recalculating optimal solutions (using TO) for proposed solutions that most violate these conditions.

Other DGM approaches in Topology Generation: Sosnovik and Oseledets [53] propose to use gradient information from a topology distribution to inform estimates of a final topology through a DNN. Their DNN takes both the density distribution and gradient of this density distribution from some intermediate step of a TO process to generate an estimate of the final topology without waiting for TO convergence. The authors demonstrate binary accuracy of 98% when predicting the final topology after only five iterations of TO, when the full TO process would take 100 iterations to complete.

In a different kind of approach, Keshavarz-zadeh *et al.* [55], address the generalization problem of training for different scales and different domains. They represent shapes in both 2D and 3D in a parametric way using what they call the "Disjunctive Normal Shape Model (DNSM)" [55]. By doing this, they create a platform for resolution-independent shape reconstruction. They then train a NN model to generate optimal topologies given boundary conditions and other problem-specific information in the DNSM space, which then can be reconstructed at any resolution using the DNSM. By doing this, they show that they are capable of applying their methods to different problems and different domains without having to deal with the super-resolution problem.

5.2 Microstructure, Nanostructure, and Metamaterials

Fig. 3: Deep Generative Models are often employed to generate 3D Metamaterial Unit cells

Many design applications not only require the need to design the topology or shape of the artifact, but also the material properties within it. Inverse design of materials is one of the key elements of Computational Materials Science. The most common approach to developing inverse materials design frameworks is the development of Process-Structure-Property (PSP) links, *i.e.* understanding how a particular material processing approach impacts its microstructure and how the corresponding microstructure impacts its physical properties [129]. Designing material microstructures for direct use is difficult, as there must then be some fabrication process to generate a material with the target microstructure. A common goal in computational materials science is microstructure image "reconstruction," in other words, generating microstructure images that exhibit certain characteristics. Bostanabad *et al.* [129] give an overview of the Microstructure Characterization and Reconstruction (MCR) field. Reconstruction can accelerate downstream tasks like augmenting the training data of networks that attempt to model PSP links. Better modeling PSP links can in turn create more accurate generative material design pipelines. Many classes of DGMs have been applied to this reconstruction task, including GANs [65], VAEs [81] and Convolutional Deep Belief

Networks (CDBNs) [130] [104, 103, 107]. Other studies attempt to bridge the gap between microstructure and properties in generative tasks using trained black-box surrogates, such as Tan *et al.*'s work [64]. Most research focuses on 2D microstructure images, though several also consider 3D voxelizations [67, 69].

Since reconstruction often involves mimicking existing microstructures, several papers have applied style-transfer [120, 121] to the problem as part of a DGM architecture. Style-transfer is in essence a loss between two images that attempts to capture the difference in "style" between the images. It is typically calculated by comparing the intermediate layers of an auxiliary style convolutional neural network. Cang *et al.* [81], for example, propose a VAE with style-transfer which is targeted at applications where only a small set of training data is available. Li *et al.* [58] also utilize style-transfer in their proposed transfer learning reconstruction framework. In another approach using style-transfer, this time using a vanilla GAN, Yang *et al.* [65] expand on the reconstruction task by attempting inverse design on the structure-property link. After training their GAN with style-transfer loss and an additional loss to penalize mode collapse, they treat the noise vectors as input variables and optimize them with Bayesian optimization, using the generator to translate between design vectors and microstructure images. The authors have chosen to optimize microstructures for energy absorption which can be evaluated from images using coupled wave analysis [131] for performance, providing them a convenient structure-property link that would be more challenging if optimizing for other objectives.

More recently, Fokina *et al.* [105] have adapted Style-GAN [7] to the microstructure image synthesis domain. Other studies have enforced the "style" of generated microstructure images through enforcement of physical properties. For example, Chen *et al.* [90] propose an approach to generate Random Heterogeneous Material (RHM) microstructure images using a cGAN conditioned on target images. Their cGAN uses an augmented loss based on the matching of perimeter, volume, and Euler characteristics. DGM studies in microstructure design have also used other advanced image translation methods introduced in computer vision such as super-resolution [59], CycleGAN [68, 4], and Pix2Pix [68, 125].

Although 2D images are a widely used means to represent microstructures, Zhang *et al.* [66] expand the microstructure reconstruction task to the 3D domain. Their ScaffoldGAN method generates scaffold materials to mimic real world examples of human bone scaffolds as well as foam metal scaffolds in both 3D and 2D data. They employ the conventional GAN approach with style loss for better visual similarity between generated scaffolds and real ones. However, the authors additionally introduce a novel "structural loss" term to specifically address spatial coherence, a limitation of GANs in emulating scaffolds. This additional loss helps them achieve better results that mimic the features of the data more realistically. Other studies also consider DGMs on 3D microstructures, such as Mosser *et al.* [67] and Liu *et al.* [69].

Photonics and Phononics: Within microstructure design, design of materials with photonic or phononic properties is an an area of interest in numerous industries including sensing, communications, and display technology. The work by Yang *et al.* [65] is an example of a subfield of generative microstructure design targeting the development of microstructures with particular photonic or phononic properties. Molesky *et al.* [132] also provide a review of inverse design in nanophotonics.

In the photonics and phononics subfield, performance evaluations such as the one used in [65] are frequently used to incorporate performance into DGMs. For example, the technique of optimization using learned mappings from the latent space of a trained autoencoder to the property space has been employed in several papers. Li *et al.* [82], Liu *et al.* [83], and Wang *et al.* [106] train an autoencoder, VAE, and Gaussian Mixture VAE [133] respectively on images of microstructures. They map the latent variables to the property space using a MLP, CNN, and GP regressor, respectively. Li *et al.* [82] directly optimize the properties using the MLP, while Liu *et al.* [83], and Wang *et al.* [106] optimize using a Genetic Algorithm.

Though optimizing latent variables is a common approach, several other DGMs have been proposed for photonics/phononics design that optimize performance in other ways. For instance, Malkiel *et al.* [57] uniquely use a parametric representation and a bidirectional DNN to simultaneously learn a bidirectional mapping between nanostructures and the property space. Ma *et al.* [92] propose a novel VAE-based framework consisting of feature extraction, prediction, recognition, and generation networks. The proposed method is capable of both forward prediction of properties based on metamaterial structure as well as the inverse (generative) prediction task based on properties. Furthermore, the framework supports self-supervised learning where the model trains on unlabeled metamaterial pattern images without corresponding property labels. The authors also demonstrate transfer learning to other microstructure shape classes as well as the inverse design of multiple microstructures forming a meta-mirror with desired properties. DGMs have also been applied to nano-scale photonic devices, such as an optic broadband power splitter as in Tang *et al.* [89].

Unit-cell-based Metamaterials: Research in generative design for microstructures has also focused on the development of unit cell structures for metamaterials. For instance, Wang *et al.* [33] fit a VAE to metamaterial unit cells and demonstrate that latent space dimensions can reflect physical properties of the unit cells. They then demonstrate several downstream tasks using their VAE, such as diverse subset selection and generation, targeted generation to match desired stiffness matrix values, and metamaterial family design. The authors also demonstrate several 2D macro-optimization runs to design arrays of unit cells to match macro-level deflection targets in an approach similar to classic TO. Xue *et al.* [84] train a VAE to generate unit cells,

then perform Bayesian Optimization in the latent space of the VAE to attain desired macroscopic elastic properties.

5.3 2D Shape Synthesis

Fig. 4: Airfoil synthesis using DGMs usually entails employing 2D shape generation approaches.

Designing features with critical geometric considerations is a task that appears in several engineering fields such as aerospace and automobile design. DGMs have been widely applied to these problems. Within aerospace, the design of an airfoil, which is the cross-sectional shape of a wing, is of particular interest to many researchers. Airfoils have a wide variety of uses in engineering domains such as propeller, rotor, and turbine blade design. Since airfoil performance parameters are of key interest, most of the research focuses on performance-aware conditional generation. For example, Yilmaz and German [75] apply a cGAN to the airfoil design problem, conditioning their network on various stall parameters. While the overwhelming majority of research in this domain uses DGMs trained to learn shape parameterization based on spline interpolation of Cartesian points, equation-based parameterization is also sometimes used. For example, Li *et al.* [95] propose a performance-aware RL framework where the RL agent learns the optimal equation coefficients using Proximal Policy Optimization (PPO) [134].

Other works generalize their applications to general 2D shape synthesis. For example, Chen and Fuge [76] propose Bézier GAN, a framework that learns shape representations through Bézier curves, featuring infoGAN style conditioning [17]. Chen *et al.* [77] expand on this work with Bayesian Optimization to maximize lift/drag ratio. Classic test data for 2D shape synthesis methods are the UIUC airfoil dataset as well as artificially designed shapes like superformulas [135]. While the emphasis of certain papers is on shapes or airfoils themselves, several papers primarily propose methodology advancements that they choose to demonstrate on the 2D shape synthesis domain. For example, Chen and Fuge [34] address the challenging problem of multi-component design generation using a GAN to synthesize parts using inter-part dependencies. The authors assume inter-part dependencies in a design are known and propose modeling them using directed acyclic graphs. They propose a hierarchical adaptation of the infoGAN using a single discriminator for the entire design and one generator/auxiliary network pair for each part in the design, which they term the hierarchical GAN (HGAN). The method is tested on a medley of synthetic datasets generated using BézierGAN [76] and demonstrated to form meaningful and interpretable latent spaces. Although the paper assumes that part dependency graphs for the object class are well defined, this paper takes a significant step towards multi-component design synthesis and interpretable GANs.

Several other papers use the 2D shape synthesis domain to address the challenges of performance evaluation in DGMs. Dering *et al.* [96], for example, apply an iterative retraining approach to boat sketches using an adaptation of the LSTM-based Sketch-RNN [136] as the generator. To evaluate candidate designs, the performance is scored using a simulated environment in a game engine, within which the "behavior" (motion) of the design is learned.

A recent work by Chen and Ahmed [15] addresses both performance evaluation and design novelty through their proposed Performance Augmented Diverse GAN (PaDGAN) framework. Conventional GANs are trained to mimic the design space they are trained in and as a consequence, are penalized from generating novel designs. PaDGAN expands on the conventional GAN architecture by modeling design performance and design diversity using a Deterministic Point Process (DPP) kernel [137]. By promoting diversity, PaDGAN directly addresses mode collapse in GANs too (see Sec. 2.1). The authors demonstrated that PaDGAN is capable of generating high quality and previously unseen novel designs for the UIUC airfoil database, using the lift to drag ratio as the quality metric.

5.4 3D Shape Synthesis

Fig. 5: DGM models, such as Range-GAN [78], can be used to generate 3D models of aircrafts with constraints.

3D object generation using deep learning is an active field in computer science, with significant research effort being dedicated to generating realistic looking shapes and objects. 3D shapes and objects are typically represented by voxels, point clouds, or meshes. Advancements in 3D shape synthesis in the computer graphics community have leaned heavily on GANs or Autoencoders, as well as other machine learning advancements like Recurrent Neural Networks (RNNs), Transformers, and Graph Neural Networks (GNNs). This research is tangentially relevant to engineering design, but is typically focused more on visual appearance and aesthetics rather than design considerations like functional performance derived from simulations or manufacturability. For this reason, we do not discuss these methods in detail and opt to list a few of the many noteworthy papers in this field from 2016 to 2021 for the reader to further explore at their discretion: [138, 139, 140, 141, 142, 143]. It is important to note that many of these proposed DGM architectures could be adapted to the design domain too. Despite the prominence of 3D Object Generation papers in the computer graphics community, a few have arisen within the engineering design community as well. In an early work, for example, Brock *et al.* [85], implement a 3-Dimensional cVAE to reconstruct and interpolate between voxelized models from the Modelnet-10 Dataset.

The majority of 3D shape synthesis work in engineering design, however, has implemented some kind of design performance consideration. Zhang *et al.* [93] propose the use of a Genetic Algorithm (GA) to optimize latent space design

embeddings of a trained VAE variant called the Variational Shape Learner (VSL) [144], which they demonstrate on the optimization of 3D aircraft models. Shu *et al.* [70] take an iterative retraining approach by retraining a GAN on high performance models evaluated using Computational Fluid Dynamics (CFD) evaluation. The authors apply the proposed method to point-cloud aircraft models from ShapeNet and use minimization of aerodynamic drag as the performance objective of choice. The authors use a standard GAN loss but use a discriminator architecture from [145]. Nobari *et al.* [78] introduce another self-augmentation approach to train on skewed or sparse datasets. They couple it with a "range loss" to encourage designs to comply to design constraints related to bounds and apply it to generate 3D aircraft models.

5.5 Other Applications

In this section, we discuss several application domains for DGMs with only a few relevant papers identified in our search.

Manufacturability: Synthesizing manufacturable designs is an important, albeit less-explored area in DGMs. Greminger [101] proposes approaching the problem of manufacturability using a MSG-GAN architecture [146] adapted to the 3D domain. The author synthetically generates topologies that are manufacturable by a 3-axis mill, trains the GAN to generate similar topologies, then optimizes generated topologies using TO.

Fluid Flow Shapes: DGMs have also been used to generate flow shapes. For example, Lee *et al.* [97] apply a Double Deep Q-Network (DoubleDQN) [147] RL framework to learn the geometry of a microfluidic channel to generate target flow shapes. The authors also propose methods in which a human designer can participate in the design process alongside the RL agent to impart design knowledge or constraints. Other approaches that integrate humans into the design generation process are discussed next.

Integrating Humans Into The Generation Process: One element of design that is often less emphasized in DGMs for design is human interpretation and design input during the generative process. Though datasets are often created using some human input, humans are not directly involved during the training process of most DGMs. Burnap *et al.* [91] note that numerical performance measures often do not correspond to a human's perception of design quality, and study this phenomenon on automobile images generated through a conditional VAE. Supporting the involvement of humans during the training process is a potential approach to find middle ground between the expertise and intuition of humans and the detailed information of datasets, which can enable different tasks. For example, Wang *et al.* [79] look at the problem of generating samples that are visually pleasing for any given human subject. To do this, they train a modified version of the auxiliary classifier GAN (ACGAN) [148]

which classifies the type of image being generated at the same time as learning to generate images. The authors, however, do not opt for a cGAN architecture. Instead, they condition the GAN using Electroencephalography (EEG) signals from human subjects. For this, they train an encoder which takes EEG signals and transforms them into a set of input features for the GAN. By exposing human subjects to the images as they are being generated, the encoder introduces the human response to the design into the generation process. This effectively allows for a human subject to extract more visually pleasing samples from the GAN model. Several other studies such as [74, 86, 97] propose frameworks that incorporate humans into the design process.

Kinematic Synthesis: In machine design, generating a mechanism to generate specific motion curves or transmit power is a common design task. Kinematic synthesis involves selecting the size, type, and configuration of mechanism components to achieve such a goal. Several works have applied DGMs to various kinematic synthesis problems. Deshpande and Purwar [86], for example, propose a VAE-based method for planar linkage synthesis. They deterministically parameterize both coupler paths and linkages, then generate sample coupler paths from a variety of linkages (four-bar, slider-crank, Stephenson-type six bar). In the first of their two studies, the authors train a VAE to reconstruct trajectories. They then propose a method in which humans can interact with the trained VAE to customize linkage paths to their design goals by visualizing the effect of various latent space perturbations and selecting one that matches the their design vision. The authors then expand on their first study with a cVAE architecture, feeding in the linkage as the data samples and the coupler curve as the condition vector. In a similar work, Sharma and Purwar [87] explore spatial linkage synthesis of 5-SS mechanisms using VAEs. In other works, the same authors take a look at paths themselves as guiding mediums for linkage mechanism design by first training models to encode paths, then using said models to search within a dataset of mechanisms to obtain solutions for the specific paths [94].

Other studies have approached the kinematic synthesis task through Reinforcement Learning. Vermeer *et al.* [108] propose using Deep Q-Learning to synthesize planar linkage mechanisms to obtain mechanisms which are capable of producing straight lines, showing that RL can be effective in generating linkage mechanisms using machine learning.

Truss Design: Raina *et al.* [109] propose a RL-based approach to truss design. In their approach, they learn from a dataset of sequential human design decisions [149]. They first train an Autoencoder to map trusses to and from a design embedding. Sequential design embeddings are then used to predict a heatmap of possible next design states using a supervised transition network. Interestingly, the authors choose to convert to and from images from their parametric representation when training the autoencoder and using decoded results. Finally, a rule-based agent selects from possible next design steps. Even without knowledge of design metrics and

performance, the RL agents were found to generate competitive designs compared to humans. Puentes *et al.* [110] expand on this work by learning action-sequence heuristics instead of individual design actions. Raina *et al.* [109] further expand on this work with a goal-based reinforcement learning agent.

Hierarchical Product Design Synthesis: Some works have applied DGMs to multi-component products, typically with a well known hierarchical structure. Stump *et al.* [35], for example, propose a unique approach which simultaneously combines RL, Recurrent Neural Networks (RNNs), grammars, and physics-based simulation. In their approach, the RNN learns to generate modular sailing crafts by sampling discrete selections from a predefined shape grammar. The training of the RNN is detailed in [111]. The control policy of the craft is then optimized using RL in a physics-based simulated sailing environment. Regenwetter *et al.* [88] explore full synthesis of bicycles, a diverse class of products typically consisting of numerous hierarchical components. The authors' demonstrate full generative synthesis of bikes using VAEs for both image and parametric representations. The authors note that their detailed design parameterization allows AI-generated designs to be physically fabricated.

Procedural Content Generation Lopez *et al.*, explore procedural content generation in a specific context for virtual reality. In their work, they introduce a reinforcement learning model that can learn to generate 3D virtual reality (VR) content which users can explore in a VR environment [99]. In their approach they specifically work towards generating manufacturing environments which are physically valid and feasible. In an extension of this work, Cunningham *et al.*, extend the method to generate content for multiple contexts instead of just one at a time [100], which enables the integration of user specific parameters.

6 Datasets

In this section, we present commonly-used datasets that have been or can be used to train algorithms for data-driven design tasks. While the datasets listed are not comprehensive, we aim to note key types of datasets that have been commonly used in engineering design applications. We hope that researchers can create larger well-annotated datasets and make them public for the community to use.

6.1 Topology Optimization Datasets

Most papers on DGMs for Topology Optimization generate their own datasets, which are often not publicly available. The most common method for dataset generation is Solid Isotropic Material with Penalisation (SIMP), which has several publicly available software implementations. A few TO datasets, however, are open source. Sosnovik and Oseledets [53] generate a dataset of 10,000 artificially generated topologies with intermediate topologies from the opti-

mization process[1] which is linked in their work. Each included topology contains one 40x40 image of the topology generated at every stage of 100 steps of Topology Optimization. Hence, a total of one million images are provided. Nie *et al.* [72] provide the dataset used to train their TopologyGAN framework[2]. Unlike the aforementioned, this dataset consists only of final topologies, but contains 49078 generated topologies at a resolution of 64x128, generated from 42 unique boundary conditions. Both datasets discussed use the ToPy [150] implementation of SIMP.

6.2 Microstructure Datasets

Technologies such as Optical microscopy (OM) and scanning electron microscopy (SEM), are often used to visualize material microstructures and are quite well established. As such, numerous datasets of microstructure scan images are publicly available, such as [151], [152], [153], *etc.*. Synthetic microstructure datasets also exist. For example, Yang *et al.* [65] provide a trained GAN model which generates synthetic microstructure images[3]. Numerous datasets are also available for design of composite materials of various types, such as the NanoMine nanopolymer composite database[4], which contains over 20,000 datapoints [154]. Compiled lists of materials science datasets are also available [5].

6.3 Design Geometry Datasets

The UIUC airfoil database[6] has been used as a case study in several generative design research frameworks [77, 15, 76]. The database details nearly 1,600 real-world airfoil designs using coordinates of points on the surface. Since the original data provides inconsistent numbers of coordinates along the top and bottom surfaces, Chen *et al.* [77] propose a method to standardize the data with B-spline interpolation over the airfoil which is also used in other works [34, 15].

6.4 3D Object Datasets

ShapeNet[7] [155] is one of the most commonly-used 3D model datasets, consisting of over 51,300 3D models of 55 object categories. PartNet [156] expands on ShapeNet with fine-grained heirarchical semantic annotations for component parts of ShapeNet objects. Princeton ModelNet[8] [157] is another commonly-used 3D model dataset consisting of 127,915 voxel-based 3D models of 662 object categories. Numerous works discussed in this review use ShapeNet or ModelNet models [70, 78, 93, 85].

---

[1]https://github.com/ISosnovik/top
[2]https://github.com/zhenguonie/2020_TopologyGAN
[3]https://github.com/zyz293/GAN_Materials_Design
[4]https://github.com/tetherless-world/nanomine-ontology
[5]https://github.com/sedaoturak/data-resources-for-materials-science
[6]https://m-selig.ae.illinois.edu/ads/coord_database.html
[7]https://shapenet.org/
[8]https://modelnet.cs.princeton.edu/

Sangpil *et al.* [158] introduce a dataset of 58,696 models of mechanical components from 68 classes called the Mechanical Components Benchmark (MCB)[9]. The MCB contains a hierarchical label tree grouping components into subclasses of different levels, such as Components → Fasteners → Nuts → Wingnuts, for example. Models are represented as point clouds, voxels, and 2D views.

6.5 CAD and CAD-based Datasets

Willis *et al.* [159] introduce two datasets of Autodesk Fusion models[10]. One dataset, intended for reconstruction tasks, contains 8,625 models and the other, intended for segmentation, contains 35,680 models. The reconstruction dataset is particularly interesting for generative tasks, as it contains information governing the sequential CAD operation steps taken to generate a part.

Regenwetter *et al.* [88] introduce a dataset called BIKED[11] consisting of mixed data extracted from 4,512 bicycle CAD models. The dataset includes bicycle assembly images, segmented subcomponent images, as well as "parametric" data. The "parametric" data consists of 2,395 mixed-type design parameters describing both high level and low-level characteristics. BIKED provides an advantage over conventional 3D object datasets for generative tasks in that synthesized designs contain the necessary parametric information to physically fabricate designs. BIKED's parametric data is used in [88] for bicycle synthesis and its image data in [160] for generating novel designs.

6.6 Metamaterials Datasets

Wang *et al.* [33] introduce a dataset of 248,396 2D unit cells represented by 50x50 pixelated matrices. The unit cells have associated stiffness tensor components provided and can also be used for TO research. Chan *et al.* [161] introduce a dataset of 3,000 3D isosurface unit cells sampled from 30 level-set functions, along with corresponding 3D elastic tensor components. Wang *et al.* [162] introduce a dataset of 795 unit cells generated from 10 lattice models. Associated stiffness tensors are provided. The three datasets can be found here[12].

6.7 Sketch Datasets

QuickDraw [163] is a sketch dataset of 50 million doodles from 345 categories[13]. The doodles were collected by Google from user-drawn sketches in an interactive sketching game. QuickDraw data is used in several works discussed [164, 96]. Toh & Miller [165] introduce a dataset of 934 innovative milk frother design sketches with associated text descriptions that can potentially be used to train DGMs, perhaps factoring in Natural Language Processing (NLP)[14].

6.8 Sequential Human Design Datasets

McComb *et al.* [149] provide a tabular truss design dataset taken from a truss design activity executed by sixteen human teams[15]. Sequential design operations such as joint and member placement are recorded using geometric parameters such as joint coordinates, member size, *etc.*. Performance metrics such as safety factors and weights are also included. While this dataset can be used as a truss design dataset, it is primarily intended as a resource to study or mimic the human design process. We note that the Autodesk Fusion reconstruction dataset mentioned previously [159] can also be used for learning sequential design tasks.

7 Discussion, Challenges and Future Work

When applying DGMs to engineering design, the 'standard' objective of mimicking the training data is often insufficient, or even counterproductive. Instead, real designs are governed by specific objectives and constraints, often including novelty or creativity. Thus, different objectives, such as real world performance metrics, novelty, and adherence to constraints may make for better training objectives. We discuss the challenges with performance evaluation, constraint violation, and incorporation of novelty in the following sections, as well as pathways to potential solutions. We further discuss some general challenges for DGMs in engineering design, such as limited availability of data, lack of benchmark problems and metrics, and delayed adoption of cutting edge methods from different research communities. Our discussion is supported by observations from our literature review.

7.1 Design Performance Evaluation

Incorporating performance evaluation into DGMs is one of the key stepping stones toward practical applications of ML in design. Three major challenges in performance evaluation are fidelity, cost, and differentiability.

1. Evaluation methods lacking fidelity may result in DGM-generated designs that do not meet specifications.
2. Computational cost precludes compatibility with methods that heavily sample performance values.
3. Lack of differentiability makes implementation into the training process difficult in any machine learning models which rely on gradient-based optimization.

Physical evaluation, which means building a product and testing its performance in the real-world, typically has the highest fidelity, but is rarely adopted in DGMs due to its prohibitive cost. Qualitative human evaluation of designs has also been investigated in several papers [86, 74, 97]],

---

[9]https://bit.ly/3ne4gwv
[10]https://github.com/AutodeskAILab/Fusion360GalleryDataset
[11]https://decode.mit.edu/projects/biked/
[12]https://ideal.mech.northwestern.edu/research/software/
[13]https://github.com/googlecreativelab/quickdraw-dataset

---

[14]https://sites.psu.edu/creativitymetrics/2018/07/18/milkfrother/
[15]https://www.sciencedirect.com/science/article/pii/S2352340918302014?via%3Dihub

although the evaluation is typically not focused on performance. Medium-fidelity evaluations, such as numerical simulations often deliver satisfactory fidelity, but can be costly and are rarely differentiable. Methods that do incorporate medium fidelity performance evaluation like [70, 63] do not rely on performance score gradients, and instead alternate (re)training and performance evaluation, typically doing this retraining a handful of times.

Low-cost evaluation methods like surrogate models can be worked into the training objective and are by far the most common performance evaluation method seen in DGMs [15, 20, 78, 18, 71, 80, 61]. Unfortunately, surrogate models can be brittle and generalize poorly to designs that differ from the data the surrogate was trained on. This is a particular concern when training data for surrogates is unevenly distributed across the performance space [78] and when novelty and performance are incorporated into training objectives for DGMs. Many different approaches have attempted to improve the performance of low-fidelity surrogate models. For example, self-supervised data augmentation [78] uses the DGM itself to generate samples in sparse regions (through optimization [166] or conditioning [78]) which then can be used to train the low-fidelity models to perform more accurately and in turn improve methods that rely on them for design generation. Multi-fidelity modeling is another promising approach, which involves generating surrogate models that augment a few costly high-fidelity samples with low-fidelity samples to attain higher fidelity surrogates with minimal expense.

Advancements have also been made in using machine learning to improve and accelerate medium-fidelity physics simulations such as FEA [167, 168, 169, 170, 171] and CFD [172, 173, 174, 171]. These works, although not yet robust enough to be easily applied as an alternative to high-fidelity simulations, provide a proof of concept for machine learning-based acceleration and even replacement of higher-fidelity simulations.

Numerous studies have also proposed crowdsourcing methods to evaluate synthesized data from generative machine learning methods [91, 175, 164] instead of physical or physics-based evaluation. Though the fidelity of crowd-sourced evaluation is highly task- and crowd-expertise dependent, the high evaluation cost and lack of differentiability is fairly universal.

In summary, current DGMs are largely constrained in performance evaluation by an inherent fidelity vs. cost trade-off, however several directions show promise in enabling faster and more accurate evaluation methods that may escape this limitation.

7.2 Feasibility, Constraints, and Manufacturability

One key component of design performance is obeying explicit design constraints, as well as implicit constraints such as physical feasibility and manufacturability. DGMs are difficult to rely upon for explicit design constraints since they are generally probabilistic and may generate completely invalid designs. This issue is a concern that many researchers have pointed to [176, 34, 104]. One potential solution is to develop inexpensive and reliable validation methods, however, this is a challenging task that may require significant human input.

Another underlying issues lies in design representation, which is discussed in detail in Sec. 3. Representations such as images often have no clear translation to representations with practical uses. Other representations, such as 3D models of various types can only be realistically fabricated using additive manufacturing, which precludes many design domains. For ML-generated designs to be physically fabricated and used in practical applications, the feasibility across this 'domain gap' must be overcome [104, 88]. Though a few works have investigated manufacturability [101], we observed that modeling it is not considered in most of the papers we reviewed. Other works have attempted to apply DGMs to parameterizations that encode similar parametric design data to what would be used in manufacturing drawings, but identified that parametric data of this sort is challenging to learn and generate [88].

7.3 Creativity and Novelty

Whereas creativity and novelty are essential aspects of the classic design process, DGMs rarely explicitly consider either. Most DGMs learn to mimic the data covering the existing and already explored portions of the design space. While this emulative behavior is helpful for maintaining realism and ensuring sample quality, it incentivizes DGMs against generating creative or novel designs [177]. For DGMs to progress towards more human-like design, advancements must be made in modeling creativity as well as in developing architectures that promote creativity.

Several recent works [166, 15, 160] have proposed methods to encourage creativity and novelty in DGMs. In their framework, CreativeGAN, Nobari *et al.*, focus on identifying novelty and guiding DGMs towards such behaviour by directly introducing novel features into typical designs, thereby expanding the design space and novelty of the DGM's data. Creativity in machine learning has also been explored outside of the design community. For example, Creative Adversarial Networks (CAN) [177] introduce entropy into the training to encourage the generation of surprising images. Readers are directed to Franceschelli *et al.*'s survey on the topic for more detail [178].

7.4 Evaluating Model Performance & Benchmark Problems

Readers may have been noted that many of the works discussed have nearly identical methodology, though they are applied to a specific dataset and only compare their work in this respect with a few other baselines. This may be attributed to a large variety of design applications compared to other domains such as computer vision and the focus of design researchers to find solutions to their specific application, instead of finding generalizable solutions to many applications. However, this makes any measure of 'state-of-the-art' algorithm dependent on specific applications. With-

out a good understanding of state-of-the-art methods that are broadly applicable to engineering applications, practitioners will struggle to select models for practical deployment. While some works in the design community have specifically introduced benchmark problems and datasets [88], there is still a need to have a large number high-quality datasets and benchmarks.

The difficulty of establishing a state-of-the-art lies in the lack of benchmark problems and performance metrics within the DGM field and within design automation as a whole. Design datasets tend to be small, restricted in domain, and sparse in distribution as we discuss in Section 7.5. Furthermore, many methods are developed on proprietary datasets. Finally, the hugely different design representation methods across the field make establishing standardized model performance benchmarks difficult, even if there were good datasets upon which to do so. These problems are noted by many authors in the field as well, who find it difficult to compare their approach with existing ones [61, 176].

7.5 Data Limitations and Quality

Data sparsity is one of the greatest challenges facing the data-driven design community and is a particular concern for researchers developing data-hungry DGMs. Broadly, there are three problems when it comes to the data. The first of these is a general lack of data in many design domains. Although datasets continue to cover more and more design fields, there still are many that lack publicly available data. There are also data representations that are underrepresented in the current datasets, notably graphs as discussed in Sec.3. The second key data-related problem that designers face in applying DGMs is the insufficient size of current datasets. A lot of the latest breakthroughs in deep learning, notably in computer vision and natural language processing, have owed their success to very large models which are notoriously data hungry, requiring millions or billions of training examples, and cannot be properly utilized with datasets of smaller size. Focusing resources on the generation of very large publicly available datasets or effective data augmentation methods would open doors for the design community to leverage larger deep models. The last major limitation in existing data is sparsity and bias. These problems are common in many design datasets. How the data is distributed in the performance space can cause some challenges for many DGMs, especially in inverse design problems [78]. Diversity is important to ensure the data covers the design space as evenly as possible to avoid bias in the data. Keeping diversity and bias avoidance in mind will help researchers generate better datasets.

We highly encourage researchers to develop diverse datasets, publicly release these datasets, and establish benchmark problems for future works. We subsequently encourage researchers to test their methods on other researchers' data, publicly release their benchmarking results, and acknowledge state-of-the-art methods when possible.

7.6 Other Challenges and Directions

So far our discussion has been focused on the VAE, GAN, and RL-based approaches, which have dominated the field of DGMs in engineering design. Several other DGMs developed in the computer science space have not yet been leveraged for generative tasks within design such as sequential DGMs based around the transformer architecture [179] and generative diffusion models [180, 181].

Transformer-based sequential models were initially developed for natural language processing and generation or translation of text, however, researchers have shown that the scalability of these models allows for them to be used for the generation of complex data in a sequential manner [182, 183, 184, 185]. Furthermore, these models have an unprecedented ability to be mixed with language processing. This may enable human language-based conditioning and control over the generation process of data [183, 185], enabling deployment for a much wider user base and a much wider set of tasks. The applications of transformer-based models as well as their scalability may appeal to researchers in the design community. Despite this, it is important to note that these models are particularly large and therefore require very large datasets, further highlighting the need for better datasets.

The other underexplored class of DGMs we identify does not share limitations with dataset size and provides an entirely new way of generating and conditioning generation of images. Diffusion models generate data by learning to denoise noisy images, rather than learning a latent distribution as in GANs and VAEs. Although diffusion models are currently applied primarily to image generation, they may also be of significant interest to design automation researchers.

8 Conclusion

In this review, we discussed the applications of Deep Generative Models (DGMs) across engineering design fields. To give readers a sense of the tools and methods available, we began our review with an overview of DGMs typically used for engineering design problems, emphasizing Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Reinforcement Learning (RL). To help weigh different design representation methods, we then discussed the strengths and weaknesses of common parameterization methods. To inform readers about existing work outside of their subdomain, we then collected and reviewed 63 papers from a variety of different engineering design subdisciplines which directly propose DGMs. To make readers aware of the datasets available on which to develop and test data-driven design methods, we review commonly-used datasets in the field. Finally, to provide inspiration, we discuss key challenges and limitations currently seen across the field and highlight possible solution pathways.

References

[1] Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N. V., and Wood, K. L., 2011.

"Computer-Based Design Synthesis Research: An Overview". *Journal of Computing and Information Science in Engineering,* **11**(2), 06. 021003.

[2] Deng, L., and Yu, D., 2014. "Deep learning: Methods and applications". *Found. Trends Signal Process.,* **7**(3–4), jun, p. 197–387.

[3] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014. "Generative adversarial nets". In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, p. 2672–2680.

[4] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A., 2017. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2242–2251.

[5] Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W., 2020. "Stargan v2: Diverse image synthesis for multiple domains". In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8185–8194.

[6] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T., 2020. "Analyzing and improving the image quality of stylegan". In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8107–8116.

[7] Karras, T., Laine, S., and Aila, T., 2019. "A style-based generator architecture for generative adversarial networks". In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4396–4405.

[8] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A., 2018. "Generative adversarial networks: An overview". *IEEE Signal Processing Magazine,* **35**(1), pp. 53–65.

[9] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., 2016. "Improved techniques for training gans". In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc., p. 2234–2242.

[10] Arjovsky, M., and Bottou, L., 2017. "Towards principled methods for training generative adversarial networks". *arXiv preprint arXiv:1701.04862.*

[11] Arjovsky, M., Chintala, S., and Bottou, L., 2017. "Wasserstein generative adversarial networks". In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, JMLR.org, p. 214–223.

[12] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A., 2017. "Improved training of wasserstein gans". In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., p. 5769–5779.

[13] Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C., 2017. "Veegan: Reducing mode collapse in gans using implicit variational learning". In Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Vol. 30, Curran Associates, Inc., pp. 3308–3318.

[14] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X., 2016. "Improved techniques for training gans". In Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds., Vol. 29, Curran Associates, Inc., pp. 2234–2242.

[15] Chen, W., and Ahmed, F., 2021. "Padgan: Learning to generate high-quality novel designs". *Journal of Mechanical Design,* **143**(3), p. 031703.

[16] Mirza, M., and Osindero, S., 2014. "Conditional generative adversarial nets". *arXiv preprint arXiv:1411.1784.*

[17] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P., 2016. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 2180–2188.

[18] Dong, Y., Li, D., Zhang, C., Wu, C., Wang, H., Xin, M., Cheng, J., and Lin, J., 2020. "Inverse design of two-dimensional graphene/h-bn hybrids by a regressional and conditional gan". *Carbon,* **169**, pp. 9–16.

[19] Ding, X., Wang, Y., Xu, Z., Welch, W. J., and Wang, Z. J., 2021. "Ccgan: Continuous conditional generative adversarial networks for image generation". In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net.

[20] Heyrani Nobari, A., Chen, W., and Ahmed, F., 2021. "Pcdgan: A continuous conditional diverse generative adversarial network for inverse design". *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Aug.

[21] Kingma, D. P., and Welling, M., 2013. "Auto-encoding variational bayes". *arXiv preprint arXiv:1312.6114.*

[22] Kullback, S., and Leibler, R. A., 1951. "On information and sufficiency". *The annals of mathematical statistics,* **22**(1), pp. 79–86.

[23] Kingma, D. P., and Welling, M., 2019. "An introduction to variational autoencoders". *Foundations and Trends in Machine Learning,* **12**(4), pp. 307–392.

[24] Sohn, K., Yan, X., and Lee, H., 2015. "Learning structured output representation using deep conditional generative models". In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15, MIT Press, p. 3483–3491.

[25] Kaelbling, L. P., Littman, M. L., and Moore, A. W., 1996. "Reinforcement learning: A survey". *J. Artif. Int. Res.,* **4**(1), May, p. 237–285.

[26] Sutton, R. S., and Barto, A. G., 2012. *Reinforcement learning: an introduction.* The MIT Press.

[27] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al., 2015. "Human-level control through deep reinforcement learning". *nature,* **518**(7540), pp. 529–533.

[28] Daneshmand, M., Helmi, A., Avots, E., Noroozi, F., Alisinanoglu, F., Arslan, H. S., Gorbova, J., Haamer, R. E., Ozcinar, C., and Anbarjafari, G., 2018. "3d scanning: A comprehensive survey". *arXiv preprint arXiv:1801.08863*.

[29] Remondino, F., 2003. "From point cloud to surface: the modeling and visualization problem". *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* **34**.

[30] Ranjan, A., Bolkart, T., Sanyal, S., and Black, M. J., 2018. "Generating 3d faces using convolutional mesh autoencoders". In Proceedings of the European Conference on Computer Vision (ECCV), pp. 704–720.

[31] Cheng, S., Bronstein, M., Zhou, Y., Kotsia, I., Pantic, M., and Zafeiriou, S., 2019. "Meshgan: Nonlinear 3d morphable models of faces". *arXiv preprint arXiv:1903.10384*.

[32] Zhang, Z., Wang, Y., Jimack, P. K., and Wang, H., 2020. "Meshingnet: a new mesh generation method based on deep learning". In International Conference on Computational Science, Springer, pp. 186–198.

[33] Wang, L., Chan, Y.-C., Ahmed, F., Liu, Z., Zhu, P., and Chen, W., 2020. "Deep generative modeling for mechanistic-based learning and design of metamaterial systems". *Computer Methods in Applied Mechanics and Engineering,* **372**, p. 113377.

[34] Chen, W., and Fuge, M., 2019. "Synthesizing designs with interpart dependencies using hierarchical generative adversarial networks". *Journal of Mechanical Design,* **141**(11), p. 111403.

[35] Stump, G. M., Miller, S. W., Yukish, M. A., Simpson, T. W., and Tucker, C., 2019. "Spatial grammarbased recurrent neural network for design form and behavior optimization". *Journal of Mechanical Design,* **141**(12).

[36] , 2020. Graph Representation of 3D CAD Models for Machining Feature Recognition With Deep Learning, Vol. Volume 11A: 46th Design Automation Conference (DAC) of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. V11AT11A003.

[37] Yang, W., Ding, H., Zi, B., and Zhang, D., 2017. "New Graph Representation for Planetary Gear Trains". *Journal of Mechanical Design,* **140**(1), 11. 012303.

[38] Hsu, C.-H., and Lam, K.-T., 1992. "A New Graph Representation for the Automatic Kinematic Analysis of Planetary Spur-Gear Trains". *Journal of Mechanical Design,* **114**(1), 03, pp. 196–200.

[39] Lee, J. Y., and Kim, K., 1996. "Geometric reasoning for knowledge-based parametric design using graph representation". *Computer-Aided Design,* **28**(10), pp. 831–841.

[40] , 2014. Graph Based Representation and Analyses for Conceptual Stages, Vol. Volume 1A: 34th Computers and Information in Engineering Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. V01AT02A071.

[41] Patalano, S., Vitolo, F., and Lanzotti, A., 2013. "A graph-based software tool for the cad modeling of mechanical assemblies". In GRAPP/IVAPP.

[42] Henaff, M., Bruna, J., and LeCun, Y., 2015. Deep convolutional networks on graph-structured data.

[43] Yun, S., Jeong, M., Kim, R., Kang, J., and Kim, H. J., 2019. "Graph transformer networks". In Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., Vol. 32, Curran Associates, Inc.

[44] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y., 2018. "Graph attention networks". In International Conference on Learning Representations.

[45] Vashishth, S., Sanyal, S., Nitin, V., and Talukdar, P., 2019. "Composition-based multi-relational graph convolutional networks". In International Conference on Learning Representations.

[46] Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R., 2017. Gated graph sequence neural networks.

[47] Liao, R., Li, Y., Song, Y., Wang, S., Hamilton, W., Duvenaud, D. K., Urtasun, R., and Zemel, R., 2019. "Efficient graph generation with graph recurrent attention networks". In Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., Vol. 32, Curran Associates, Inc.

[48] Bojchevski, A., Shchur, O., Zügner, D., and Günnemann, S., 2018. "Netgan: Generating graphs via random walks". In International Conference on Machine Learning, pp. 609–618.

[49] You, J., Ying, R., Ren, X., Hamilton, W., and Leskovec, J., 2018. "Graphrnn: Generating realistic graphs with deep auto-regressive models". In International conference on machine learning, PMLR, pp. 5708–5717.

[50] Li, Y., Vinyals, O., Dyer, C., Pascanu, R., and Battaglia, P., 2018. Learning deep generative models of graphs.

[51] Cao, N. D., and Kipf, T., 2018. Molgan: An implicit generative model for small molecular graphs.

[52] You, J., Liu, B., Ying, R., Pande, V., and Leskovec, J., 2018. "Graph convolutional policy network for goaldirected molecular graph generation". In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 6412–6422.

[53] Sosnovik, I., and Oseledets, I., 2019. "Neural networks for topology optimization". *Russian Journal of Numerical Analysis and Mathematical Modelling,* **34**(4), pp. 215–223.

[54] Behzadi, M. M., and Ilieş, H. T., 2021. "Real-time

topology optimization in 3d via deep transfer learning". *Computer-Aided Design,* **135**, p. 103014.

[55] Keshavarzzadeh, V., Alirezaei, M., Tasdizen, T., and Kirby, R. M., 2021. "Image-based multiresolution topology optimization using deep disjunctive normal shape model". *Computer-Aided Design,* **130**, p. 102947.

[56] Cang, R., Yao, H., and Ren, Y., 2019. "One-shot generation of near-optimal topology through theory-driven machine learning". *Computer-Aided Design,* **109**, pp. 12–21.

[57] Malkiel, I., Mrejen, M., Nagler, A., Arieli, U., Wolf, L., and Suchowski, H., 2018. "Plasmonic nanostructure design and characterization via deep learning". *Light: Science & Applications,* **7**(1), pp. 1–8.

[58] Li, X., Zhang, Y., Zhao, H., Burkhart, C., Brinson, L. C., and Chen, W., 2018. "A transfer learning approach for microstructure reconstruction and structure-property predictions". *Scientific reports,* **8**(1), pp. 1–13.

[59] Jung, J., Na, J., Park, H. K., Park, J. M., Kim, G., Lee, S., and Kim, H. S., 2021. "Super-resolving material microstructure image via deep learning for microstructure characterization and mechanical behavior analysis". *npj Computational Materials,* **7**(1), pp. 1–11.

[60] Li, B., Huang, C., Li, X., Zheng, S., and Hong, J., 2019. "Non-iterative structural topology optimization using deep learning". *Computer-Aided Design,* **115**, pp. 172–180.

[61] Rawat, S., and Shen, M. H., 2019. Application of adversarial networks for 3d structural topology optimization. Tech. rep., SAE Technical Paper.

[62] Oh, S., Jung, Y., Lee, I., and Kang, N., 2018. "Design automation by integrating generative adversarial networks and topology optimization". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 51753, American Society of Mechanical Engineers, p. V02AT03A008.

[63] Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N., 2019. "Deep generative design: Integration of topology optimization and generative models". *Journal of Mechanical Design,* **141**(11).

[64] Tan, R. K., Zhang, N. L., and Ye, W., 2020. "A deep learning–based method for the design of microstructural materials". *Structural and Multidisciplinary Optimization,* **61**(4), pp. 1417–1438.

[65] Yang, Z., Li, X., Catherine Brinson, L., Choudhary, A. N., Chen, W., and Agrawal, A., 2018. "Microstructural materials design via deep adversarial learning methodology". *Journal of Mechanical Design,* **140**(11).

[66] Zhang, H., Yang, L., Li, C., Wu, B., and Wang, W., 2021. "Scaffoldgan: Synthesis of scaffold materials based on generative adversarial networks". *Computer-Aided Design,* **138**, p. 103041.

[67] Mosser, L., Dubrule, O., and Blunt, M. J., 2017. "Reconstruction of three-dimensional porous media using generative adversarial neural networks". *Physical Review E,* **96**(4), p. 043309.

[68] Lee, J.-W., Goo, N. H., Park, W. B., Pyo, M., and Sohn, K.-S., 2021. "Virtual microstructure design for steels using generative adversarial networks". *Engineering Reports,* **3**(1), p. e12274.

[69] Liu, S., Zhong, Z., Takbiri-Borujeni, A., Kazemi, M., Fu, Q., and Yang, Y., 2019. "A case study on homogeneous and heterogeneous reservoir porous media reconstruction by using generative adversarial networks". *Energy Procedia,* **158**, pp. 6164–6169.

[70] Shu, D., Cunningham, J., Stump, G., Miller, S. W., Yukish, M. A., Simpson, T. W., and Tucker, C. S., 2020. "3d design using generative adversarial networks and physics-based validation". *Journal of Mechanical Design,* **142**(7), p. 071701.

[71] Sharpe, C., and Seepersad, C. C., 2019. "Topology design with conditional generative adversarial networks". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 59186, American Society of Mechanical Engineers, p. V02AT03A062.

[72] Nie, Z., Lin, T., Jiang, H., and Kara, L. B., 2021. "Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain". *Journal of Mechanical Design,* **143**(3), p. 031715.

[73] Yu, Y., Hur, T., Jung, J., and Jang, I. G., 2019. "Deep learning for determining a near-optimal topological design without any iteration". *Structural and Multidisciplinary Optimization,* **59**(3), pp. 787–799.

[74] Valdez, S., Seepersad, C., and Kambampati, S., 2021. "A framework for interactive structural design exploration". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21, ASME.

[75] Yilmaz, E., and German, B., 2020. "Conditional generative adversarial network framework for airfoil inverse design". In AIAA aviation 2020 forum, p. 3185.

[76] Chen, W., and Fuge, M., 2018. "Béziergan: Automatic generation of smooth curves from interpretable low-dimensional parameters". *arXiv preprint arXiv:1808.08871*.

[77] Chen, W., Chiu, K., and Fuge, M., 2019. "Aerodynamic design optimization and shape exploration using generative adversarial networks". In AIAA Scitech 2019 Forum, p. 2351.

[78] Heyrani Nobari, A., Chen, W. W., and Ahmed, F., 2021. "RANGE-GAN: Design Synthesis Under Constraints Using Conditional Generative Adversarial Networks". *Journal of Mechanical Design*, 09, pp. 1–16.

[79] Wang, P., Peng, D., Li, L., Chen, L., Wu, C., Wang, X., Childs, P., and Guo, Y., 2019. "Human-in-the-loop design with machine learning". *Proceedings of the Design Society: International Conference on Engineering Design,* **1**(1), p. 2577–2586.

[80] Guo, T., Lohan, D. J., Cang, R., Ren, M. Y., and Allison, J. T., 2018. "An indirect design representation for topology optimization using variational autoencoder and style transfer". In 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, p. 0804.

[81] Cang, R., Li, H., Yao, H., Jiao, Y., and Ren, Y., 2018. "Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model". *Computational Materials Science,* **150**, pp. 212–221.

[82] Li, X., Ning, S., Liu, Z., Yan, Z., Luo, C., and Zhuang, Z., 2020. "Designing phononic crystal with anticipated band gap through a deep learning based data-driven method". *Computer Methods in Applied Mechanics and Engineering,* **361**, p. 112737.

[83] Liu, Z., Raju, L., Zhu, D., and Cai, W., 2020. "A hybrid strategy for the discovery and design of photonic structures". *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* **10**(1), pp. 126–135.

[84] Xue, T., Wallin, T. J., Menguc, Y., Adriaenssens, S., and Chiaramonte, M., 2020. "Machine learning generative models for automatic design of multi-material 3d printed composite solids". *Extreme Mechanics Letters,* **41**, p. 100992.

[85] Brock, A., Lim, T., Ritchie, J. M., and Weston, N., 2016. "Context-aware content generation for virtual environments". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 50084, American Society of Mechanical Engineers, p. V01BT02A045.

[86] Deshpande, S., and Purwar, A., 2019. "Computational creativity via assisted variational synthesis of mechanisms using deep generative models". *Journal of Mechanical Design,* **141**(12).

[87] Sharma, S., and Purwar, A., 2020. "Path synthesis of defect-free spatial 5-ss mechanisms using machine learning". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 83990, American Society of Mechanical Engineers, p. V010T10A034.

[88] Regenwetter, L., Curry, B., and Ahmed, F., 2021. "BIKED: A dataset and machine learning benchmarks for data-driven bicycle design". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21, ASME.

[89] Tang, Y., Kojima, K., Koike-Akino, T., Wang, Y., Wu, P., Tahersima, M., Jha, D., Parsons, K., and Qi, M., 2020. "Generative deep learning model for a multilevel nano-optic broadband power splitter". In 2020 Optical Fiber Communications Conference and Exhibition (OFC), IEEE, pp. 1–3.

[90] Chen, H., and Liu, X., 2021. "Geometry enhanced generative adversarial networks for random heterogeneous material representation". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21, ASME.

[91] Burnap, A., Liu, Y., Pan, Y., Lee, H., Gonzalez, R., and Papalambros, P. Y., 2016. "Estimating and exploring the product form design space using deep generative models". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 50107, American Society of Mechanical Engineers, p. V02AT03A013.

[92] Ma, W., Cheng, F., Xu, Y., Wen, Q., and Liu, Y., 2019. "Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy". *Advanced Materials,* **31**(35), p. 1901111.

[93] Zhang, W., Yang, Z., Jiang, H., Nigam, S., Yamakawa, S., Furuhata, T., Shimada, K., and Kara, L. B., 2019. "3d shape synthesis for conceptual design and optimization using variational autoencoders". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 59186, American Society of Mechanical Engineers, p. V02AT03A017.

[94] Deshpande, S., and Purwar, A., 2020. "An Image-Based Approach to Variational Path Synthesis of Linkages". *Journal of Computing and Information Science in Engineering,* **21**(2), 10. 021005.

[95] Li, R., Zhang, Y., and Chen, H., 2021. "Learning the aerodynamic design of supercritical airfoils through deep reinforcement learning". *AIAA Journal*, pp. 1–14.

[96] Dering, M., Cunningham, J., Desai, R., Yukish, M. A., Simpson, T. W., and Tucker, C. S., 2018. "A physics-based virtual environment for enhancing the quality of deep generative designs". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 51753, American Society of Mechanical Engineers, p. V02AT03A015.

[97] Lee, X. Y., Balu, A., Stoecklein, D., Ganapathysubramanian, B., and Sarkar, S., 2019. "A case study of deep reinforcement learning for engineering design: Application to microfluidic devices for flow sculpting". *Journal of Mechanical Design,* **141**(11), p. 111401.

[98] Raina, A., Puentes, L., Cagan, J., and McComb, C., 2021. "Goal-directed design agents: Integrating visual imitation with one-step lookahead optimization for generative design". *Journal of Mechanical Design,* **143**(12), p. 124501.

[99] , 2019. Reinforcement Learning Content Generation for Virtual Reality Applications, Vol. Volume 1: 39th Computers and Information in Engineering Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. V001T02A009.

[100] , 2020. Multi-Context Generation in Virtual Real-

ity Environments Using Deep Reinforcement Learning, Vol. Volume 9: 40th Computers and Information in Engineering Conference (CIE) of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. V009T09A072.

[101] Greminger, M., 2020. "Generative adversarial networks with synthetic training data for enforcing manufacturing constraints on topology optimization". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 84003, American Society of Mechanical Engineers, p. V11AT11A005.

[102] Fujita, K., Minowa, K., Nomaguchi, Y., Yamasaki, S., and Yaji, K., 2021. "Design concept generation with variational deep embedding over comprehensive optimization". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21, ASME.

[103] Cang, R., Vipradas, A., and Ren, Y., 2017. "Scalable microstructure reconstruction with multi-scale pattern preservation". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 58134, American Society of Mechanical Engineers, p. V02BT03A010.

[104] Cang, R., Xu, Y., Chen, S., Liu, Y., Jiao, Y., and Yi Ren, M., 2017. "Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design". *Journal of Mechanical Design,* **139**(7), p. 071404.

[105] Fokina, D., Muravleva, E., Ovchinnikov, G., and Oseledets, I., 2020. "Microstructure synthesis using style-based generative adversarial networks". *Physical Review E,* **101**(4), p. 043308.

[106] Wang, Z., Xian, W., Baccouche, M. R., Lanzerath, H., Li, Y., and Xu, H., 2021. "A gaussian mixture variational autoencoder-based approach for designing phononic bandgap metamaterials". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21, ASME.

[107] , 2016. Deep Network-Based Feature Extraction and Reconstruction of Complex Material Microstructures, Vol. Volume 2B: 42nd Design Automation Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. V02BT03A008.

[108] , 2018. Kinematic Synthesis Using Reinforcement Learning, Vol. Volume 2A: 44th Design Automation Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. V02AT03A009.

[109] Raina, A., McComb, C., and Cagan, J., 2019. "Learning to design from humans: Imitating human designers through deep learning". *Journal of Mechanical Design,* **141**(11).

[110] Puentes, L., Raina, A., Cagan, J., and McComb, C., 2020. "Modeling a strategic human engineering design process: Human-inspired heuristic guidance through learned visual design agents". In Proceedings of the Design Society: DESIGN Conference, Vol. 1, Cambridge University Press, pp. 355–364.

[111] Yukish, M. A., Stump, G. M., and Miller, S. W., 2020. "Using recurrent neural networks to model spatial grammars for design creation". *Journal of Mechanical Design,* **142**(10), p. 104501.

[112] Zhu, J.-H., Zhang, W.-H., and Xia, L., 2016. "Topology optimization in aircraft and aerospace structures design". *Archives of Computational Methods in Engineering,* **23**(4), pp. 595–622.

[113] Xia, L., and Breitkopf, P., 2017. "Recent advances on topology optimization of multiscale nonlinear structures". *Archives of Computational Methods in Engineering,* **24**(2), pp. 227–249.

[114] Borrvall, T., and Petersson, J., 2003. "Topology optimization of fluids in stokes flow". *International journal for numerical methods in fluids,* **41**(1), pp. 77–107.

[115] Zhou, S., and Li, Q., 2008. "A variational level set method for the topology optimization of steady-state navier–stokes flow". *Journal of Computational Physics,* **227**(24), pp. 10178–10195.

[116] Zegard, T., and Paulino, G. H., 2016. "Bridging topology optimization and additive manufacturing". *Structural and Multidisciplinary Optimization,* **53**(1), pp. 175–192.

[117] Langelaar, M., 2016. "Topology optimization of 3d self-supporting structures for additive manufacturing". *Additive Manufacturing,* **12**, pp. 60–70.

[118] Dbouk, T., 2017. "A review about the engineering design of optimal heat transfer systems using topology optimization". *Applied Thermal Engineering,* **112**, pp. 841–854.

[119] Koga, A. A., Lopes, E. C. C., Nova, H. F. V., De Lima, C. R., and Silva, E. C. N., 2013. "Development of heat sink device by using topology optimization". *International Journal of Heat and Mass Transfer,* **64**, pp. 759–772.

[120] Gatys, L., Ecker, A. S., and Bethge, M., 2015. "Texture synthesis using convolutional neural networks". *Advances in neural information processing systems,* **28**, pp. 262–270.

[121] Gatys, L. A., Ecker, A. S., and Bethge, M., 2016. "Image style transfer using convolutional neural networks". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2414–2423.

[122] Berthelot, D., Schumm, T., and Metz, L., 2017. "Began: Boundary equilibrium generative adversarial networks". *arXiv preprint arXiv:1703.10717*.

[123] Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H., 2017. "Variational deep embedding: an unsupervised and generative approach to clustering". In Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 1965–1972.

[124] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2017. "Photo-realistic single image super-resolution using a generative adversarial network". In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 105–114.

[125] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A., 2017. "Image-to-image translation with conditional adversarial networks". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134.

[126] He, K., Zhang, X., Ren, S., and Sun, J., 2016. "Deep residual learning for image recognition". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[127] Hu, J., Shen, L., and Sun, G., 2018. "Squeeze-and-excitation networks". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141.

[128] Long, J., Shelhamer, E., and Darrell, T., 2015. "Fully convolutional networks for semantic segmentation". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

[129] Bostanabad, R., Zhang, Y., Li, X., Kearney, T., Brinson, L. C., Apley, D. W., Liu, W. K., and Chen, W., 2018. "Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques". *Progress in Materials Science,* **95**, pp. 1–41.

[130] Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y., 2009. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations". In Proceedings of the 26th annual international conference on machine learning, pp. 609–616.

[131] Yu, S., Zhang, Y., Wang, C., Lee, W.-k., Dong, B., Odom, T. W., Sun, C., and Chen, W., 2017. "Characterization and design of functional quasi-random nanostructured materials using spectral density function". *Journal of Mechanical Design,* **139**(7), p. 071401.

[132] Molesky, S., Lin, Z., Piggott, A. Y., Jin, W., Vucković, J., and Rodriguez, A. W., 2018. "Inverse design in nanophotonics". *Nature Photonics,* **12**(11), pp. 659–670.

[133] Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K., and Shanahan, M., 2016. "Deep unsupervised clustering with gaussian mixture variational autoencoders". *arXiv preprint arXiv:1611.02648.*

[134] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., 2017. "Proximal policy optimization algorithms". *arXiv preprint arXiv:1707.06347.*

[135] Gielis, J., 2003. "A generic geometric transformation that unifies a wide range of natural and abstract shapes". *American journal of botany,* **90**(3), pp. 333–338.

[136] Ha, D., and Eck, D., 2018. "A neural representation of sketch drawings". In International Conference on Learning Representations.

[137] Kulesza, A., Taskar, B., et al., 2012. "Determinantal point processes for machine learning". *Foundations and Trends® in Machine Learning,* **5**(2–3), pp. 123–286.

[138] Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., and Guibas, L., 2017. "Grass: Generative recursive autoencoders for shape structures". *ACM Transactions on Graphics (TOG),* **36**(4), pp. 1–14.

[139] Zou, C., Yumer, E., Yang, J., Ceylan, D., and Hoiem, D., 2017. "3d-prnn: Generating shape primitives with recurrent neural networks". In Proceedings of the IEEE International Conference on Computer Vision, pp. 900–909.

[140] Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M., 2018. "A papier-mâché approach to learning 3d surface generation". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 216–224.

[141] Gao, L., Yang, J., Wu, T., Yuan, Y.-J., Fu, H., Lai, Y.-K., and Zhang, H., 2019. "Sdm-net: Deep generative network for structured deformable mesh". *ACM Transactions on Graphics (TOG),* **38**(6), pp. 1–15.

[142] Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N. J., and Guibas, L. J., 2019. "Structurenet: hierarchical graph networks for 3d shape generation". *ACM Transactions on Graphics,* **38**(6), pp. 1–19.

[143] Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S., 2019. "Deepsdf: Learning continuous signed distance functions for shape representation". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 165–174.

[144] Liu, S., Giles, L., and Ororbia, A., 2018. "Learning a hierarchical latent-variable model of 3d shapes". In 2018 International Conference on 3D Vision (3DV), IEEE, pp. 542–551.

[145] Qi, C. R., Su, H., Mo, K., and Guibas, L. J., 2017. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652–660.

[146] Karnewar, A., and Wang, O., 2020. "Msg-gan: Multi-scale gradients for generative adversarial networks". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7799–7808.

[147] van Hasselt, H., Guez, A., and Silver, D., 2016. "Deep reinforcement learning with double q-learning". *Proceedings of the AAAI Conference on Artificial Intelligence,* **30**(1), Mar.

[148] Odena, A., Olah, C., and Shlens, J., 2017. "Conditional image synthesis with auxiliary classifier gans". In International conference on machine learning, PMLR, pp. 2642–2651.

[149] McComb, C., Cagan, J., and Kotovsky, K., 2018. "Data on the design of truss structures by teams of engineering students". *Data in brief,* **18**, pp. 160–163.

[150] Hunter, W., et al., 2017. Topy-topology optimization with python.

[151] Iren, D., Ackermann, M., Gorfer, J., Pujar, G., Wesselmecking, S., Krupp, U., and Bromuri, S., 2021. "Aachen-heerlen annotated steel microstructure dataset". *Scientific Data,* **8**(1), pp. 1–9.

[152] Larmuseau, M., Sluydts, M., Theuwissen, K., Duprez, L., Dhaene, T., and Cottenier, S., 2020. "Compact representations of microstructure images using triplet networks". *npj Computational Materials,* **6**(1), pp. 1–11.

[153] DeCost, B. L., Hecht, M. D., Francis, T., Webler, B. A., Picard, Y. N., and Holm, E. A., 2017. "Uhcsdb: ultrahigh carbon steel micrograph database". *Integrating Materials and Manufacturing Innovation,* **6**(2), pp. 197–205.

[154] Zhao, H., Wang, Y., Lin, A., Hu, B., Yan, R., McCusker, J., Chen, W., McGuinness, D. L., Schadler, L., and Brinson, L. C., 2018. "Nanomine schema: An extensible data representation for polymer nanocomposites". *APL Materials,* **6**(11), p. 111108.

[155] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al., 2015. "Shapenet: An information-rich 3d model repository". *arXiv preprint arXiv:1512.03012.*

[156] Mo, K., Zhu, S., Chang, A. X., Yi, L., Tripathi, S., Guibas, L. J., and Su, H., 2019. "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 909–918.

[157] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J., 2015. "3d shapenets: A deep representation for volumetric shapes". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1912–1920.

[158] Kim, S., Chi, H.-g., Hu, X., Huang, Q., and Ramani, K., 2020. "A large-scale annotated mechanical components benchmark for classification and retrieval tasks with deep neural networks". In Proceedings of 16th European Conference on Computer Vision (ECCV).

[159] Willis, K. D., Pu, Y., Luo, J., Chu, H., Du, T., Lambourne, J. G., Solar-Lezama, A., and Matusik, W., 2021. "Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences". *ACM Transactions on Graphics (TOG),* **40**(4), pp. 1–24.

[160] Nobari, A. H., Rashad, M. F., and Ahmed, F., 2021. "Creativegan: Editing generative adversarial networks for creative design synthesis". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21, ASME.

[161] Chan, Y.-C., Ahmed, F., Wang, L., and Chen, W., 2021. "Metaset: Exploring shape and property spaces for data-driven metamaterials design". *Journal of Mechanical Design,* **143**(3), p. 031707.

[162] Wang, L., van Beek, A., Da, D., Chan, Y.-C., Zhu, P., and Chen, W., 2021. "Data-driven multiscale design of cellular composites with multiclass microstructures for natural frequency maximization". *arXiv preprint arXiv:2106.06478.*

[163] Jongejan, J., Rowley, H., Kawashima, T., Kim, J., and Fox-Gieg, N., 2016. "The quick, draw!-ai experiment". *Mount View, CA, accessed Feb,* **17**(2018), p. 4.

[164] Lopez, C., Miller, S. R., and Tucker, C. S., 2018. "Human validation of computer vs human generated design sketches". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 51845, American Society of Mechanical Engineers, p. V007T06A015.

[165] Toh, C. A., and Miller, S. R., 2013. "Exploring the utility of product dissection for early-phase idea generation". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 55928, American Society of Mechanical Engineers, p. V005T06A034.

[166] Chen, W., and Ahmed, F., 2021. Mo-padgan: Reparameterizing engineering designs for augmented multi-objective optimization.

[167] Liang, L., Liu, M., Martin, C., and Sun, W., 2018. "A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis". *Journal of The Royal Society Interface,* **15**(138), p. 20170844.

[168] , 2020. StressGAN: A Generative Deep Learning Model for 2D Stress Distribution Prediction, Vol. Volume 11B: 46th Design Automation Conference (DAC) of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* V11BT11A023.

[169] , 2019. Stress Field Prediction in Cantilevered Structures Using Convolutional Neural Networks, Vol. Volume 1: 39th Computers and Information in Engineering Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* V001T02A011.

[170] Nie, Z., Jiang, H., and Kara, L. B., 2019. "Stress field prediction in cantilevered structures using convolutional neural networks". *Journal of Computing and Information Science in Engineering,* **20**(1), Sep.

[171] Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P., 2020. "Learning mesh-based simulation with graph networks". In International Conference on Learning Representations.

[172] Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., and Hoyer, S., 2021. "Machine learning–accelerated computational fluid dynamics". *Proceedings of the National Academy of Sciences,* **118**(21).

[173] Duraisamy, K., Iaccarino, G., and Xiao, H., 2019. "Turbulence modeling in the age of data". *Annual Review of Fluid Mechanics,* **51**(1), pp. 357–377.

[174] Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B., 2019. "Deep fluids: A gen-

erative network for parameterized fluid simulations". *Computer Graphics Forum,* **38**(2), May, p. 59–70.

[175] Dering, M. L., and Tucker, C. S., 2017. "Generative adversarial networks for increasing the veracity of big data". In 2017 IEEE International Conference on Big Data (Big Data), IEEE, pp. 2595–2602.

[176] Panchal, J. H., Fuge, M., Liu, Y., Missoum, S., and Tucker, C., 2019. "Special Issue: Machine Learning for Engineering Design". *Journal of Mechanical Design,* **141**(11), 10. 110301.

[177] Elgammal, A., Liu, B., Elhoseiny, M., and Mazzone, M., 2017. "Can: Creative adversarial networks generating "art" by learning about styles and deviating from style norms". In 8th International Conference on Computational Creativity, ICCC 2017, Georgia Institute of Technology.

[178] Franceschelli, G., and Musolesi, M., 2021. Creativity and machine learning: A survey.

[179] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I., 2017. "Attention is all you need". In Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Vol. 30, Curran Associates, Inc.

[180] Ho, J., Jain, A., and Abbeel, P., 2020. Denoising diffusion probabilistic models.

[181] Dhariwal, P., and Nichol, A., 2021. Diffusion models beat gans on image synthesis.

[182] Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I., 2020. "Generative pretraining from pixels". In Proceedings of the 37th International Conference on Machine Learning, H. D. III and A. Singh, eds., Vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 1691–1703.

[183] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I., 2021. Zero-shot text-to-image generation.

[184] Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I., 2020. Jukebox: A generative model for music.

[185] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I., 2021. Learning transferable visual models from natural language supervision.