KSME

Springer

Journal of Mechanical Science and Technology 38 (4) 2024

Original Article
DOI 10.1007/s12206-024-0328-1

Keywords:
· Design optimization
· Reduced-order modeling
· Data-efficient deep learning
· Image processing
· Topology optimization

Correspondence to:
Yiming Zhang
yimingzhang@zju.edu.cn;
Nam-Ho Kim
nkim@ufl.edu

Citation:
Zhang, Y., Jia, C., Zhang, H., Fang, N., Zhang, S., Kim, N.-H. (2024). Improving data-efficiency of deep generative model for fast design synthesis. Journal of Mechanical Science and Technology 38 (4) (2024) 1957~1970.
http://doi.org/10.1007/s12206-024-0328-1

Received June 16th, 2023
Revised October 8th, 2023
Accepted January 11th, 2024

† Recommended by Editor
Hyun-Gyu Kim

Improving data-efficiency of deep generative model for fast design synthesis

Yiming Zhang[1,2], Chen Jia[1], Hongyi Zhang[1], Naiyu Fang[1], Shuyou Zhang[1,2] and Nam-Ho Kim[3]

[1]State Key Laboratory of Fluid Power and Mechatronic System, Zhejiang University, Hangzhou 310058, Zhejiang, China, [2]Engineering Research Center for Design Engineering and Digital Twin of Zhejiang Province, Zhejiang University, Hangzhou 310058, Zhejiang, China, [3]The Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Gainesville 32611, Florida, USA

Abstract The convolutional neural network-based deep generative model (DGM) is a powerful tool for handling image datasets that opens up strategies for fast synthesis of optimum designs at unseen boundary conditions. Existing DGMs for design synthesis are typically based on O (10000) training data, which limits the engineering applications. This paper explores the feasibility of improving DGM data efficiency with O (100) training data through prior constraints. A two-stage data-efficient deep generative model (DE-DGM) is proposed which leverages the first-stage design synthesis from probabilistic proper orthogonal decomposition and the second-stage enhancement from encoder-decoder convolutional neural network. Four topology optimization cases have been adopted, including compliance minimization, heat conduction, airplane bearing bracket design, and three-dimensional machine tool column structure design. The proposed DE-DGMs could be trained with 100-200 data and synthesize the main features of the design at unseen boundary conditions. The overall computation cost of warm-start topology optimization leveraging DE-DGM predictions reduces to 36 %-58 % of the standard cases.

1. Introduction

Topology optimization (TopOpt) computes the shape or material distribution within a predefined feasible domain. Mathematically, TopOpt can be formulated as a high-dimensional constrained optimization problem rooted in finite element analysis [1, 2], where the objective function is to maximize the global performance of the design (e.g., stiffness). Currently, the mainstream approaches for TopOpt can be broadly categorized into five classes: density-based [3-5], grid-based [6, 7], moving boundary-based [8-10], load path-based [11], and optimal partitioning methods [12].

In the past few decades, TopOpt has been widely studied and applied to a broad range of engineering problems, including static/dynamic mechanical structure design [13], eigenvalue problems [14], plastic deformation [15], design-dependent loadings [16], compliant mechanism design [17], and multi-physics problems [18]. With recent advances in additive manufacturing and the associated design schemes, TopOpt has demonstrated promising performance in producing highly flexible and customized designs such as truss structures [19, 20].

Although TopOpt is a powerful design tool, a dedicated effort is needed to derive numerically stable results with affordable computational costs. Deep generative models (DGM) have been emerging to exploit a set of correlated legacy designs and predict near-optimal designs without recourse to simulations [21-28]. The DGMs are expected to enable fast design space exploration without explicitly setting up simulations. In addition, DGM predictions can also serve as an initial guess for advanced TopOpt, such as multi-physics problems or dynamic problems.

However, most of the existing DGMs are data-intensive and easily incur more than 10000 training simulations, which is computationally challenging. Numerous efforts have been made to incorporate the physical insights of topology optimization into DGM for the purpose of improving its accuracy. For example, Cang et al. [29] proposed an intrusive scheme that leverages the simulation optimality criteria and adds simulations iteratively from random selection. Qiu et al. [30] developed a convolutional recurrent neural network and predicted the temporal-spatial evolution of topology optimization. Lew and Buehler [31] combined variational autoencoders with the long short-term memory neural networks to predict the temporal-spatial evolution of topology optimization.

In this work, we concentrate on non-intrusive techniques that are general to handle different types of design synthesis, including the dataset from commercial software, historical designs, and parametric optimizations. This paper addresses the data efficiency challenge by imposing theoretical constraints as prior information to mitigate the dependency on the training datasets. This is achieved by combining proposed probabilistic proper orthogonal decomposition (P-POD) with the convolutional neural network (CNN). P-POD assumes a linear relationship between the image dataset with changing boundary condition variables [32, 33]. The linearity assumption significantly reduces the size of the training dataset and avoids heuristic optimization of model training. The P-POD predictions are deemed as low-resolution due to the linearity assumption. An encoder-decoder convolutional neural network (ED-CNN) is then developed to enhance/transfer P-POD predictions for enhanced fine features.

In this study, topology optimization serves as the data generator for legacy designs at changing boundary condition variables. P-POD generates the optimal linear subspace for a given set of field data. Then DGM adds non-linearity to P-POD predictions and synthesizes design at unseen boundary conditions. The scientific contributions and novelty of this work are summarized as follows:

1) A novel and data-efficient predictive framework for fast design synthesis at given boundary condition variables is proposed. The generality has been validated with compliance minimization, heat conduction, airplane bearing bracket, and three-dimensional (3D) machine tool column structure design.

2) The uncertainty quantification methods for design synthesis, which serves as the foundation for risk management and model improvement, are developed.

3) A two-stage deep generative model to model sparse image datasets is proposed. Effective design synthesis could be produced with even 100-200 training data.

Developing generative models with sparse image datasets is a challenging task, as an initial effort; this work focuses on parametric boundary condition variables and evaluates its feasibility from multiple perspectives. Four design problems with different physics and geometrical characteristics were adopted. The article is organized as follows, Sec. 2 discusses the technical background of DGM and proposes the two-stage DE-DGM, Sec. 3 discusses the numerical experiments of DE-DGM for TopOpt with applications to compliance minimization, heat conduction and airplane bearing bracket design, Sec. 4 discusses a 3D numerical experiment of DE-DGM for TopOpt with application to machine tool column structure design, Sec. 5 concludes the work with a summary and future work.

Table 1. List of acronyms and abbreviations.

ACD	Absolute compliance difference
CNN	Convolutional neural network
DGM	Deep generative model
DE-DGM	Data-efficient deep generative model
ED-CNN	Encoder-decoder convolutional neural network
FEA	Finite element analysis
GAN	Generative adversarial network
GP	Gaussian process
MMA	Method of moving asymptotes
POD	Proper orthogonal decomposition
P-POD	Probabilistic proper orthogonal decomposition
TopOpt	Topology optimization
VAE	Variational autoencoder

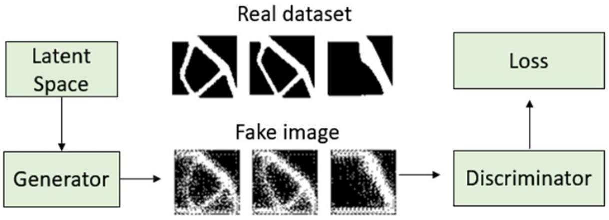


Fig. 1. Schematic diagram of a typical GAN.

2. Methodology

In this section, the fundamental framework of DGM is outlined, and the two-stage DE-DGM proposed in this paper is introduced.

2.1 Deep generative model

In this work, DGM is referred to as an instance of the deep CNN to account for the spatial correlation between a given pixel and its neighboring pixels. In a CNN, a layer is a linear mapping, and non-linearity is introduced thereafter through the activation function. DGMs have shown superior performance in handling images, including vector-to-image, image-to-image, and image-to-vector problems. Typical DGMs include variational autoencoder (VAE) and generative adversarial network (GAN). VAE extends the autoencoder to generate new images from a low-dimensional latent space [34]. In comparison with VAE, GAN has demonstrated superior performance in modeling a variety of fine features and has become the main scheme for design synthesis [21, 22, 35, 36]. There are two neural networks in a GAN as shown in Fig. 1: the generator network and discriminator network, where the generator network is trained to predict/reconstruct the field information, while the discriminator network is trained to classify real and predicted images,

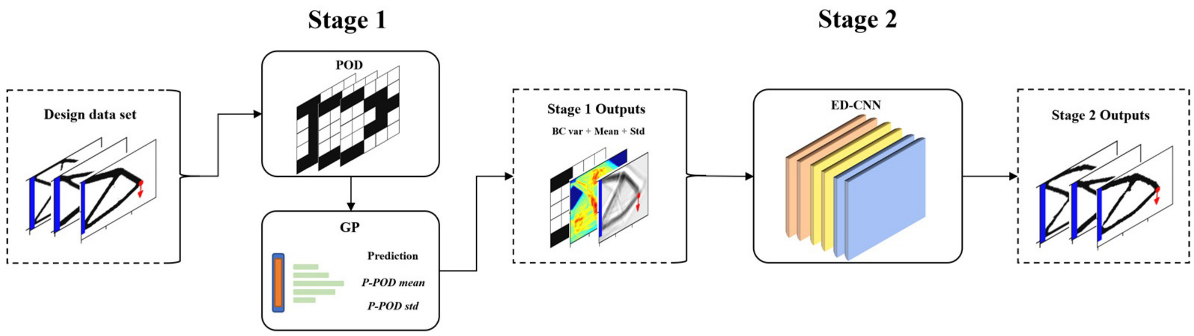


Fig. 2. Flowchart of the proposed DE-DGM.

respectively. For fast topology optimization, DGM is essentially an advanced surrogate that takes boundary condition variables as inputs and outputs the pixel-wise designs.

The generative model maps a random noise distribution to the distribution of latent features observed from the image data. The training of the two networks proceeds alternatively to improve either generator or discriminator. A typical loss function, shown in Eq. (1) [22], comprises a discriminator loss term and a generator loss term, where $\mathbf{x}$ is sampled from the data distribution P_{data}, $\mathbf{z}$ is sampled from the noise distribution $P_{\mathbf{z}}$, and $G(\mathbf{z})$ is the generator distribution. The noise input $\mathbf{z}$ is considered as a latent representation and used for design synthesis. For topology optimization, the latent representation is augmented with boundary condition variables such as loading. The CNN-based generator has synthesized conceptual designs close to TopOpt simulations. A drawback of the CNN-based generator is that it needs a large dataset before getting a meaningful model. Also, the training cost of deep CNN could be significant for a large dataset.

$$\begin{aligned}&\min_G \max_D V(D,G)\\&= E_{\mathbf{x}\sim P_{\text{data}}}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim P_{\mathbf{z}}}[\log(1-D(G(\mathbf{z})))]\,.\end{aligned} \tag{1}$$

2.2 Data-efficient deep generative model

DGM synthesizes a category of designs with parametric boundary conditions. Typical parametric boundary conditions include load locations, load orientation, and length of fixed bound. The DGMs developed by Refs. [22, 35] and Ref. [21] were trained with 10000-100000 topology optimization data. We aim to reduce the number of training data through the introduction of prior constraints. The proposed DE-DGM utilizes P-POD as the first-stage design synthesis. The linear assumption of P-POD enables design synthesis with sparse training datasets. However, it is challenging for P-POD to capture the fine features also due to the linear assumption. The pixel-wise uncertainty of P-POD prediction (i.e., synthesized design) is derived for uncertainty quantification. We formulate an ED-CNN for non-linear mapping of the first-stage synthesized designs with enhanced fine features. The basic flowchart of the proposed DE-DGM is depicted in Fig. 2.

2.2.1 The first-stage design synthesis from probabilistic proper orthogonal decomposition

P-POD decomposes a given set of structural designs into a linear combination of basic modes [37-39]. The basic modes are obtained through an optimizer with constraints to be normalized and orthogonal. The pixel-wise scalar value representing designs (in the format of images) makes up a matrix $P^{(k)} = p^{(k)}_{(i,j)}$, where k denotes the indices for images and (i,j) (i = 1, 2, ..., I; j = 1, 2, 3, ..., J) are the indices of the pixels. Each synthesized design $P^{(k)}_{pred}$ is a linear combination of M basis modes φ_m and the corresponding coefficients $c^{(k)}_m$ as seen in Eq. (2) [39], where m is the mode index.

$$P^{(k)}_{pred} = \sum_{m=1}^{M} c^{(k)}_m \varphi_m\,. \tag{2}$$

The basic modes and mode coefficients $c^{(k)}_m$ are obtained by minimizing the root-mean-square difference between the images and predictions as shown in Eq. (3) [39].

$$\min \sum_{k=1}^{K} \left\| P^{(k)} - \sum_{m=1}^{M} c^{(k)}_m \varphi_m \right\|^2 . \tag{3}$$

The constraints for normalization and orthogonality are defined in Eqs. (4) and (5) [39], respectively.

$$\sum_{i=1}^{I}\sum_{j=1}^{J} \left\| p^{(k)}_{i,j} \right\| = 1 \tag{4}$$

$$\begin{aligned}&\int \varphi_m(x)\varphi_p(x)dx = \delta_{m,p}\,,\\&\delta_{m,p} = \begin{cases} 1 \text{ if } i=j \\ 0 \text{ if } i \neq j\,. \end{cases}\end{aligned} \tag{5}$$

Due to the linear assumption, P-POD can avoid heuristic optimization and obtain the global optimum for mode coefficients. The main features of the design can be reconstructed with even a few training data. The first $10 \times d_{vars}$ modes are adopted for design synthesis, where d_{vars} is the number of changing boundary condition variables.

P-POD mode coefficients at unseen boundary conditions are approximated with Gaussian process (GP). GPs have proven to be flexible and effective surrogate models for expensive non-linear responses [40-42], which makes it highly suitable for sparse data (X,Y) or $(x^{(i)},y^{(i)})$, where $x^{(i)}$ is a vector of n_d-dimensional inputs at the i_{th} data point and $y^{(i)}$ is the associated output (i = 1, 2, . . . , n). The GP is setup with the squared-exponential kernel as in Eq. (6) [41] due to its generality, where m is the m-th dimension of inputs:

$$k(x^{(i)},x^{(j)})=\frac{1}{\lambda_0}\exp\left\{-\sum_{m=1}^{n_d}\beta_m(x_m^{(i)}-x_m^{(j)})^2\right\}. \tag{6}$$

Where $1/\lambda_0$ and β_m represent the process variance and the length-scale. The marginal likelihood of the GP requires inverting the covariance matrix K defined in Eq. (7) [41], where $\delta_{i,j}$ is an identity matrix:

$$K_{i,j}=k\left(x^{(i)},x^{(j)}\right)+\frac{1}{\lambda_1}\delta_{i,j}\ . \tag{7}$$

In GP, the parameters λ_0, λ_1, and β_m need to be learned from the data. Specifically, the parameter β_m controls the length scale in the m-th dimension, while the parameters λ_0, λ_1 control the variance and noise in the GP. The marginal likelihood for GP with a given dataset (X,Y) is shown in Eq. (8) [41]. The model fitting is done with Metropolis-Hastings algorithm for estimating the posterior distribution of the parameters:

$$\log P(Y\mid X)=-\frac{1}{2}Y^TK^{-1}Y-\frac{1}{2}\log|K|-\frac{n}{2}\log(2\pi)\,. \tag{8}$$

The uncertainty of first-stage design synthesis is propagated from P-POD coefficients. The pixel-wise P-POD predictions (synthesized designs) are essentially a d_{pix}-variates Gaussian distribution $N\left(\mu(x^*),\sum(x^*)\right)$, where d_{pix} is the number of image pixels, and x^* denotes the boundary condition variables. The P-POD mean prediction $\mu(x^*)$ is shown in Eq. (9) [42], where $\hat{c}_m(x^*)$ is the predicted mode coefficients from GPs.

$$\mu(x^*)=\sum_{m=1}^{M}\hat{c}_m(x^*)\varphi_m\ . \tag{9}$$

The covariance $\sum(x^*)$ is a diagonal matrix while assuming independent POD modes, and the diagonal elements $Var\left(\mu(x^*)\right)$ of $\sum(x^*)$ are the pixel-wise uncertainty of synthesized designs. $Var\left(\mu(x^*)\right)$ are calculated following Eq. (10) [42]:

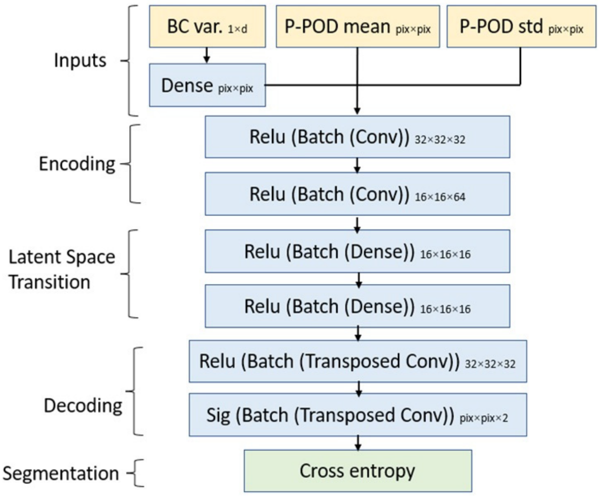

Inputs
BC var. 1×d
P-POD mean pix×pix
P-POD std pix×pix
Dense pix×pix
Encoding
Relu (Batch (Conv)) 32×32×32
Relu (Batch (Conv)) 16×16×64
Latent Space Transition
Relu (Batch (Dense)) 16×16×16
Relu (Batch (Dense)) 16×16×16
Decoding
Relu (Batch (Transposed Conv)) 32×32×32
Sig (Batch (Transposed Conv)) pix×pix×2
Segmentation
Cross entropy

Fig. 3. Architecture of the proposed encoder-decoder convolutional neural network.

$$Var\left(\mu(x^*)\right)=\sum_{m=1}^{M}Var\left(\hat{c}_m(x^*)\right)\varphi_m^2\ . \tag{10}$$

2.2.2 The second-stage design synthesis from encoder-decoder convolutional neural network

CNN-based generators like GANs have demonstrated promising performance in design synthesis. However, GANs are notoriously unstable to train, and they often suffer from mode collapse, where the generator converges to generate samples from one category of the data distribution but misses many others. Besides, O (10000) training points are typically needed, which makes GANs less feasible for practical engineering applications.

To address this issue, an ED-CNN is developed, as shown in Fig. 3. Encoder-decoder architecture has been shown to be superior with image reconstruction by gradually capturing the key features at different scales. The ED-CNN inputs include three channels from loading conditions, P-POD mean predictions, and P-POD prediction uncertainty. The images are shaped with 64×64 pixels. For encoding layers, latent space transitions, decoding layers, and two blocks are utilized, respectively. ReLU layers are used for non-linear mapping; the sigmoid layer is used to output the pixel-wise segmentation probability. The DGM is formulated as a binary image segmentation problem to indicate with/without material for each pixel. The Softmax cross-entropy loss is calculated according to Eq. (11) [21].

$$\begin{aligned}&\text{CELoss}(y_{i,j,k},\hat{y}_{i,j,k})\\&=\frac{-1}{N_t d_{pix}}\sum_{i=1}^{N_t}\sum_{j=1}^{d_{pix}}\sum_{k=1}^{2}y_{i,j,k}\log(\hat{y}_{i,j,k})\,.\end{aligned} \tag{11}$$

$y_{i,j,k}$ is the pixel-wise one-hot encoding label at i_{th} training data, j_{th} pixel, k is the index of one-hot encoding, $\hat{y}_{i,j,k}$ is the Softmax prediction of ED-CNN. Compared with other image reconstruction techniques based on root-mean-square error, our formulation using segmentation eliminates the existence of gray regions and produces a better characterization of the design boundary. Training is obtained with Adam optimizer, which is an advanced stochastic gradient descent method incorporating various enhancing concepts and heuristics, like momentum, adapting learning rate rules, subgradient principles, and Nesterov steps.

3. Numerical experiments

This section evaluates the performance of the proposed DE-DGM for modeling sparse images. Three topology optimization cases are examined with distinctive geometrical characteristics.

3.1 Topology optimization

Topology optimization simulations are adopted as the design generator with changing boundary conditions. Density-based methods are the most widely used algorithm for topology optimization [1, 2]. These methods discretize a specified domain into finite elements and usually treat each element as having variable density values. An objective function is defined based on the response of the structure under certain loads as shown in Eq. (12) [2]:

$$\begin{aligned} &\min f(\rho, U) \\ &s.\,t.\ \mathbf{K}(\rho)U = F(\rho) \\ &\qquad g_i(\rho, U) \le 0 \\ &\qquad\quad 0 \le \rho \le 1 . \end{aligned} \tag{12}$$

In this equation, $f(\rho,U)$ is the objective function, ρ is the vector of density design variables, U is the displacement vector, $\mathbf{K}$ is the stiffness matrix, F is the force vector, and g_i are constraints. For a standard compliance minimization problem $f = U^T\mathbf{K}U$, $g = V / V_0 - V_f \le 0$, V is structure material volume, V_0 is design domain volume, V_f is allowable volume fraction.

Each finite element is assigned a design variable, namely density, where 0 means hole, 1 means solid. Density variables are penalized by a power law and multiplied with physical quantities such as material stiffness, cost, or conductivity as in Eq. (13) [2]:

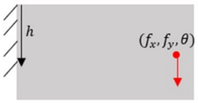
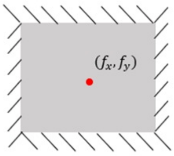
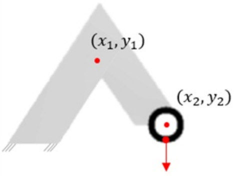
$$E = \eta(\rho)E_0 . \tag{13}$$

For compliance minimization, E is Young's modulus of a finite element, E_0 is Young's modulus of the material, $\eta(\rho)$ is the penalization function ρ^3 , the density values of finite elements within a predefined radius are averaged/filtered to improve the numerical stability of topology optimization. Eq. (12) can be solved using optimizers such as optimality criteria methods. Please refer to Refs. [1, 2] to get more details on heat conduction and compliant mechanism. Three popular TopOpt problems are implemented to produce image dataset.

3.2 Experimental setup

The specific topology optimization boundary conditions pertaining to each case are reported in Table 2. The boundary condition variables of the compliance minimization case include horizontal and vertical locations of the load (f_x, f_y) , load orientation θ , length of the fixed bound h . For the heat conduction problem, the boundary condition variables are the locations of the heat sink. The airplane bearing bracket is con-

Table 2. Topology optimization domain and boundary condition variables of the 3 validation cases.

Case	Domains	Boundary conditions
Compliance minimization		- 150 × 100 finite elements - Location of the load $f_x \in [120, 150]$, $f_y \in [40, 60]$ - Orientation of the load $\theta \in [60°, 120°]$ - Length of fixed bound $h \in [60,100]$
Heat conduction		- 80 × 60 finite elements - Distributed heating all over the plate - The design domain is isolated at all edges except for the heat sink. - Location of the heat sink $x \in [1, 80]$, $y \in [15, 45]$
Airplane bearing bracket		- 200 × 200 finite elements - Two left corner points are fixed as bolts - The center of beam intersection $x_1 \in [80, 100]$, $y_1 \in [120, 130]$ - The center of bearing $x_2 \in [140, 150]$, $y_2 \in [40, 90]$ - A downward point load on the bearing bottom surface

verted from a 3D application [43]. Three boundary condition variables are adopted to characterize the complex design envelope.

Regarding the design of experiments, $50\times d_{vars}$ samples are selected from the Latin hypercube sampling, where d_{vars} = 4 for compliance minimization and airplane bearing bracket, d_{vars} = 2 for heat conduction. Topology optimization simulations stop when the variation of compliance is less than 0.1 %. The DE-DGM has 57574 trainable parameters and is trained with NVIDIA GeForce GTX 1050 Ti. Adam optimizer is used at a learning rate to be 0.0001. The training stops after 300 epochs.

A comparative evaluation was undertaken between DE-DGM and StyleGAN. GAN and its variants have been widely used for image reconstruction. Rich boundary conditions can be modeled through proper training with a large dataset [22]. To the authors' best knowledge, no explicit effort has been made on GAN for handling sparse image datasets (i.e., O (100) data). In this paper, we have adopted StyleGAN [44, 45] as a reference model to compare with. StyleGAN is a popular variation of GAN that attracts significant attention since its development in 2019. StyleGAN remaps the latent code into an intermediate space for the disentangled representation of features. Differing from inputting the following decoder directly in other conditional GANs, StyleGAN controls the style and local details of the synthesized image by adaptive instance normalization. The dimensionality of the latent space and output channels has been adjusted to accommodate our specific cases. Pytorch implementation of the StyleGAN is adopted from https://github. com/rosinality/style-based-gan-pytorch.

3.3 Model validation for compliance minimization and heat conduction

For compliance minimization, P-POD prediction (i.e., synthesized design) at one test run is shown in Fig. 4. The P-POD mean predictions capture the main features. As expected, the fine features are missing and grey regions are observed at most pixels. The DE-DGM predictions eliminate grey regions by nature. The predicted standard deviation indicates higher uncertainty at the center region without running TopOpt simulations. Fig. 5 compares the performance of StyleGAN, DE-DGM, and the truth (i.e., simulations) for compliance minimization. DE-DGM was developed upon the P-POD synthesized designs and synthesizes the major geometrical features. Sophisticated

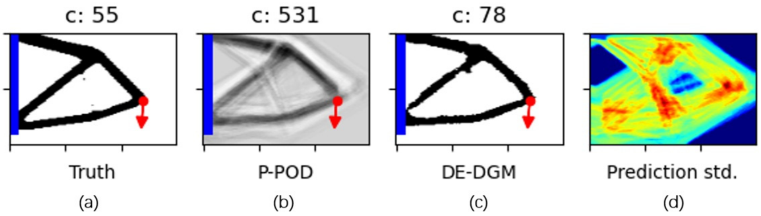


Fig. 4. Design synthesis from P-POD and DE-DGM for compliance minimization: (a) ground truth for compliance minimization; (b) design synthesis from P-POD mean prediction; (c) design synthesis from DE-DGM; (d) design synthesis from P-POD prediction uncertainty. c in (a), (b) and (c) denotes the compliance objection function value at the corresponding convergence time for each design. The blue line in (a), (b) and (c) on the left indicates the length of the fixed bound. The red arrow in (a), (b) and (c) indicates the location and orientation of the point load.

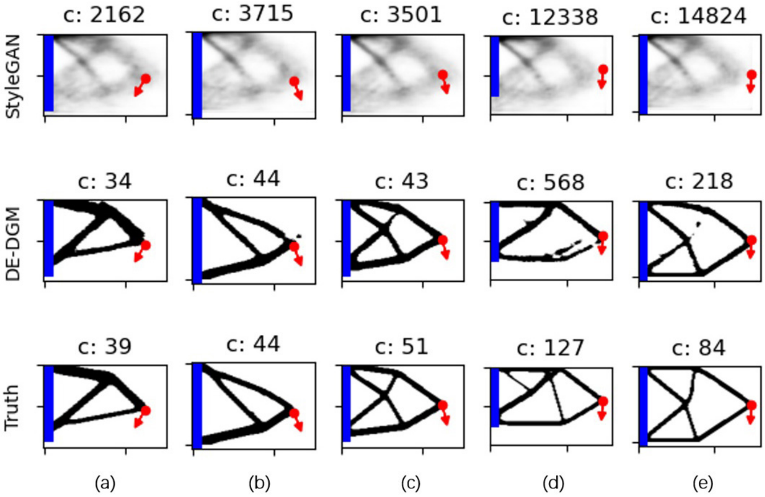
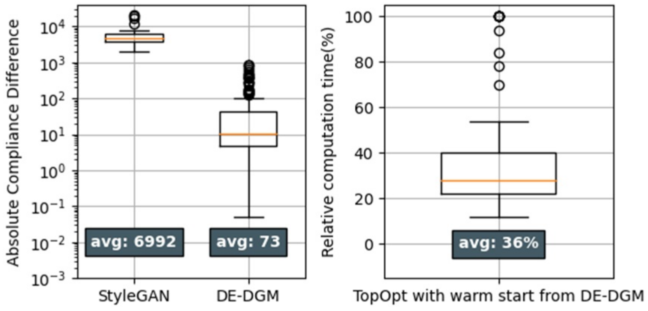


Fig. 5. Comparison of StyleGAN, DE-DGM, and ground truth for compliance minimization; (a)-(e) predictions from StyleGAN and DE-DGM of cases 1-5 as well as ground truths. c in (a), (b), (c), (d) and (e) denotes the compliance objection function value at the corresponding convergence time for each design. The blue line in (a), (b), (c), (d) and (e) on the left indicates the length of the fixed bound. The red arrow in (a), (b), (c), (d) and (e) indicates the location and orientation of the point load.

features might get lost as shown in the 5_{th} run in Fig. 5(e); the compliance of DE-DGM prediction is more than twice that of ground truth. For the five demonstrated runs, DE-DGM captures the overall design qualitatively.

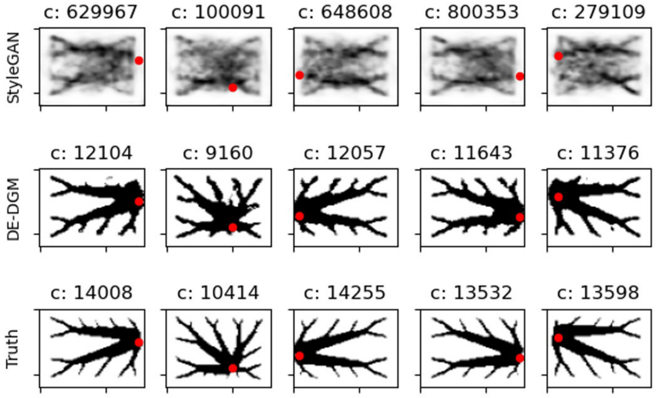
Regarding the StyleGAN predictions, various engineering setups of StyleGAN, including loss functions, learning rates, the number of epochs, the number of neurons, and the number of layers, were explored. StyleGAN predictions hardly capture the design variations and converge to the mean designs over all the training datasets as shown in Fig. 5. The original version of condition GAN is based on a latent space at a dimensionality of around 100000 [46]. The original StyleGAN has a latent space at the dimensionality 512 [44]. Effective characterization of such large latent spaces would require an enormous dataset. In this work, the latent space dimensions have been engineered to range from two to four dimensions. However, the main architecture of StyleGAN was originally designed and evaluated on a large dataset. A specialized effort is needed to make StyleGAN work on sparse datasets. For a further quantitative evaluation, we adopted 100 test runs from Latin hypercube sampling. The absolute compliance difference (ACD) between ground truth and synthesized designs is shown in Fig. 6(a). The average ACD for StyleGAN and DE-DGM is 6992 and 73, respectively. Although DE-DGM is substantially more accurate than StyleGAN, large ACD are observed (i.e., a few hundred) in some test runs. The synthesized designs are expected to assist engineers for fast conceptual designs and design prototyping. Post-processing of the predicted designs is required through engineering judgment to perform finite element analysis (FEA), especially at the boundary condition regions. Or the DE-DGM predictions could serve as the initial guess of high-fidelity topology optimization. Topology optimization simulations are conducted for 100 test runs, utilizing the initial density guesses generated from the DE-DGM predictions. The required computation cost (i.e., number of finite element analyses) of the warm-start topology optimization for reaching a superior compliance value (i.e., smaller than the compliance of standard cases) is recorded. Fig. 6(b) shows the computation cost of warm-start topology optimization relative to that of the standard case. The computation cost of warm-start topology optimization reduces to 36 % of the standard case on average and almost all the test runs get accelerated.

Subsequently, the heat conduction problem is examined, as illustrated in Fig. 7. The designs between heat conduction and compliance minimization have quite distinctive geometrical characteristics. The heat conduction problem grows tree struc-



(a) (b)

Fig. 6. The performance of fast design synthesis with DE-DGM for compliance minimization: (a) absolute prediction errors of compliance for StyleGAN and DE-DGM; (b) relative calculation time of warm-start topology optimization from DE-DGM prediction. 100 test runs are adopted.



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

Fig. 7. Comparison of StyleGAN, DE-DGM, and ground truth for heat conduction: (a)-(e) predictions from StyleGAN and DE-DGM of cases 1-5 as well as ground truths. c in (a), (b), (c), (d) and (e) denotes the objection function value for each design. The red dot in (a), (b), (c), (d) and (e) indicates the location of the heat sink.

tures with different numbers of branches, and these branches have different locations. The structure scale decreases gradually when getting away from the heat sink. The DE-DGM predictions are fairly close to the ground truth and have smaller compliance values. The volume fractions of DE-DGM predictions may be larger than the ground truth and lead to smaller compliance values. We do observe some fine features missing as a few end-branches got lost for the heat conduction problem. The ACD of StyleGAN and DE-DGM for the 100 test runs are shown in Fig. 8(a). The average ACD of DE-DGM is smaller than that of StyleGAN by an order of magnitude. TopOpt has been conducted with the initial density guess derived from DE-DGM predictions. The computation costs of warm-start topology optimization reduce to 38 % of the standard case on average as in Fig. 8(b).

3.4 Fast design of airplane bearing bracket

DE-DGM is a data-driven scheme in nature and its accuracy depends on the geometrical complexities of the design concerned and the dimension of the boundary condition variables. In this section, we examine the fast design of an airplane bearing bracket with four boundary condition variables to illustrate the application of DE-DGM with complex design domain envelope [43].

The bearing bracket is a common component on control surfaces of various aircraft. As a dynamic component that interfaces with moving parts, it must conform to a certain geometrical envelope. DE-DGM is expected to enable fast design of the bracket that adapts to a variety of geometrical envelopes. For 3D topology optimization of the bracket with CalculiX finite element software [47], it takes 36 hours and 7.7 GB RAM for the second-order mesh with 166000 elements. Thus, we adopt the 2D version for preliminary validation.

The DE-DGM is developed with 200 training data. As shown in Fig. 9, the DE-DGM predictions capture the main features fairly well and it is adaptive to the changing design domains/envelopes. DE-DGM misses the lower support near the bearing in the third design, which increases the compliance value noticeably. The DE-DGM predictions may have smaller compliance values due to the increased volume fraction. Quantitative evaluations of StyleGAN and DE-DGM for 100 test runs

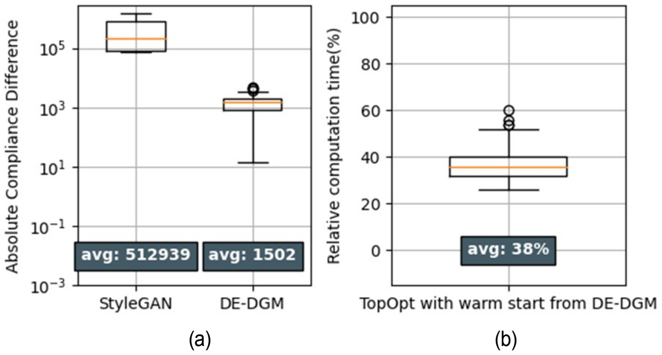


Fig. 8. The performance of fast design synthesis with DE-DGM for heat conduction: (a) absolute prediction errors of compliance for StyleGAN and DE-DGM; (b) relative calculation time of warm-start topology optimization from DE-DGM prediction. 100 test runs are adopted.

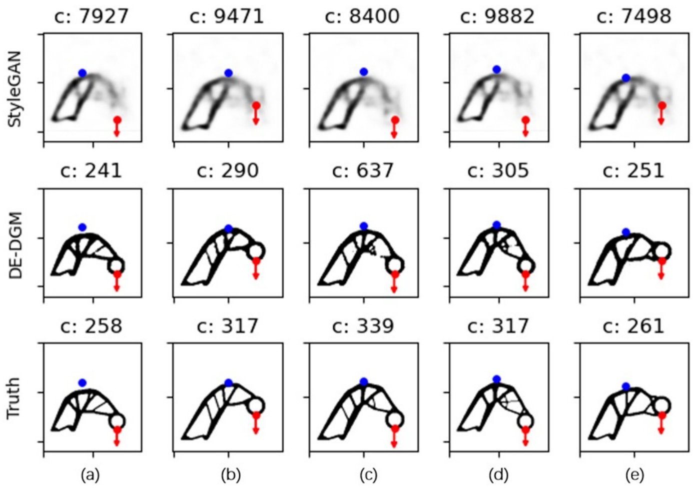


Fig. 9. Comparison of StyleGAN, DE-DGM, and ground truth for airplane bearing bracket design: (a)-(e) predictions from StyleGAN and DE-DGM of cases 1-5 as well as ground truths. c in (a), (b), (c), (d) and (e) denotes the compliance value for each design. The blue dot in (a), (b), (c), (d) and (e) indicates the center of the beam intersection area. The red arrow indicates the location of the point load.

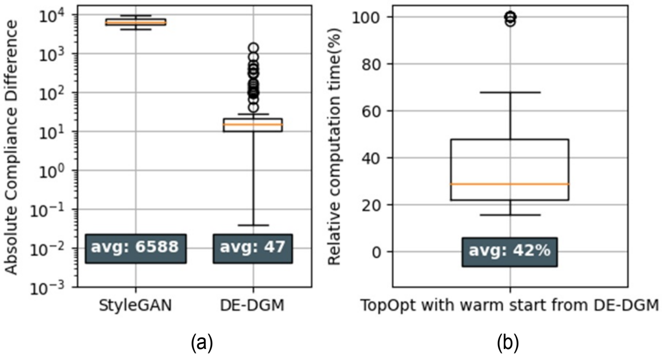


Fig. 10. The performance of fast design synthesis with DE-DGM for airplane bearing bracket design: (a) absolute prediction errors of compliance for StyleGAN and DE-DGM; (b) relative calculation time of warm-start topology optimization from DE-DGM prediction. 100 test runs are adopted.

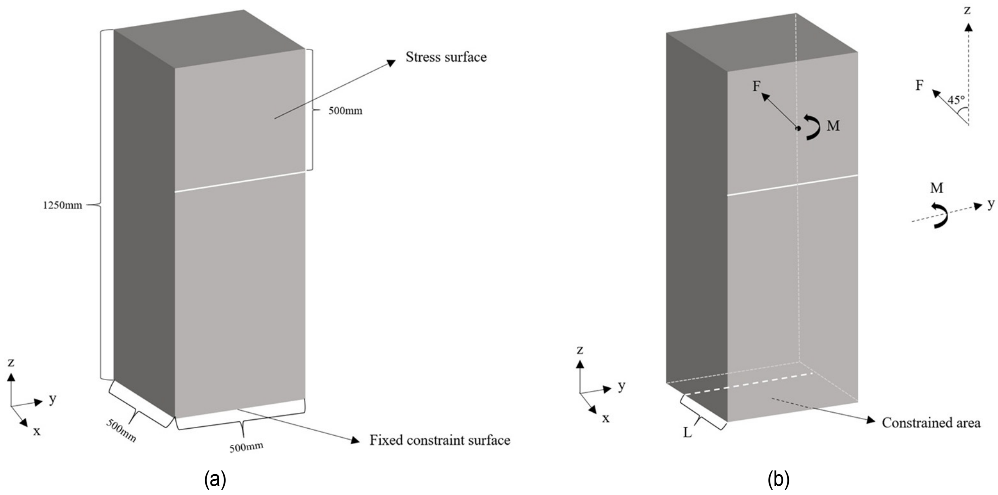


Fig. 11. 3D column model: (a) selection of dimensions, stress surface, and constraint surface of the 3D column model; (b) selection of variable loading conditions of the model.

are shown in Fig. 10(a). The average of ACD for DE-DGM is as small as 47. Large ACD are observed at some test runs (i.e., a few hundred). The computation cost of warm-start topology optimization for the 100 test runs is shown in Fig. 10(b). With 42 % computation cost of the standard case on average, the warm-start topology optimization can produce a design with superior compliance value.

The proposed DE-DGM can synthesize the main features of compliance minimization, heat conduction and airplane bearing bracket design with $50\times d_{vars}$ training data. In order to get a quantitatively accurate design, remodeling and post-processing of the synthesized designs are needed. The DE-DGM predictions can be adopted as the initial guess for high-fidelity topology optimization. Compared with the standard topology optimization, the warm-start cases have been shown to reach a superior compliance with reduced computation cost, which is 36 %-42 % for the three numerical cases.

4. 3D numerical experiment for a machine tool column

To further verify the effectiveness of the proposed DE-DGM, a three-dimensional numerical case needs to be incorporated into the model validation process.

4.1 Case introduction

A machine tool consists of several major components, including the bed, column, spindle box, spindle, and worktable. For the design of machine tool components, many designers still rely on empirical design and structural analogy. Topology optimization can purposefully improve the performance of structures, providing effective references for structural design of parts. This section focuses on the structural design of a machine tool column. The selected column model to be studied is as shown in Fig. 11(a). The model has dimensions of 500 mm in length, 500 mm in width, and 1250 mm in height, representing the initial design domain of the column. The installation of the spindle box corresponds to the loading position, and the connecting parts are simplified to the force-bearing surface in the loading condition.

The material of the solid structure model of the column was set to cast iron. The density of cast iron (ρ) is 7.2×10^3 kg/m^3, Poisson's ratio (μ) is 0.25, and the elastic modulus (E) was 140

GPa. The loading conditions of the column were set with fixed constraints on the bottom surface and applied forces and moments on the upper stress surface. The spindle box was installed on the stress surface of the column. Assuming the end of the spindle is oriented vertically downward, the cutting tool experiences a force in the upward direction at a 45-degree angle to the vertical direction. The force and moment applied to the stress surface of the column are defined as equivalent loads. The equivalent force is simplified as a concentrated load applied at the center of the stress surface, and the equivalent moment is applied across the entire stress surface.

In this case, three variable conditions (d_{vars} = 3) were considered. The first variable condition involves the size of the constrained area on the bottom surface. The width of the constrained area remains constant, while the length (L) of the constrained area is changed to vary the constrained area size. The range of values for the length of the constrained area is [300, 500] mm. The second variable condition pertains to the magnitude of the concentrated load (F), which can take values within the range of [10, 60] kN. The third variable condition focuses on the magnitude of the moment (M), which can vary within the range of [10, 60] kN·m. The distributions of the respective variables are illustrated in Fig. 11(b).

4.2 Model modification

Considering the applicability of the DE-DGM to 3D cases, some modifications are necessary to the model presented in Sec. 3. The first stage with the P-POD component remains unchanged. The second stage, which uses the ED-CNN module, is extended to predict three-dimensional cases. Similar to the previous structure, the ED-CNN inputs include three channels from loading conditions, P-POD mean predictions and P-POD prediction uncertainty. The initial input shape for training data is reset to voxels. For encoding layers, latent space transitions, decoding layers, and two blocks are utilized, respectively. ReLU layers are used for non-linear mapping; the sigmoid layer is used to output the voxel-wise segmentation probability. The DGM is formulated as a binary segmentation problem to indicate with/without material for each voxel. The basic model framework is shown in Fig. 12.

4.3 Model validation

To validate the effectiveness of the proposed DE-DGM, experiments were conducted on the 3D column case. Within the range of variable conditions, four groups of cases with structural representativeness were selected for experimentation. The validation process followed the following steps:

1) 500 sets of variable combinations were obtained through Latin hypercube sampling, each comprising integer values for L, F, and M.

2) The obtained 500 sets of data were used to build models and perform topology optimization computations using the method of moving asymptotes (MMA) algorithm. The results of the topology optimization were exported and matched with the corresponding variable conditions.

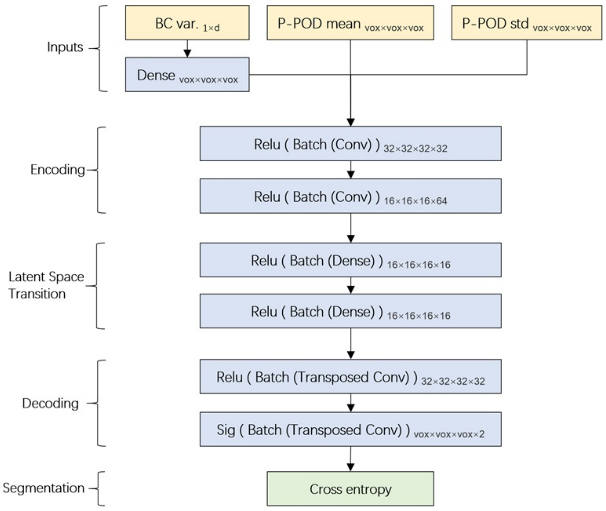

Inputs
BC var. 1×d
P-POD mean vox×vox×vox
P-POD std vox×vox×vox
Dense vox×vox×vox
Encoding
Relu (Batch (Conv)) 32×32×32×32
Relu (Batch (Conv)) 16×16×16×64
Latent Space Transition
Relu (Batch (Dense)) 16×16×16×16
Relu (Batch (Dense)) 16×16×16×16
Decoding
Relu (Batch (Transposed Conv)) 32×32×32×32
Sig (Batch (Transposed Conv)) vox×vox×vox×2
Segmentation
Cross entropy

Fig. 12. Architecture of ED-CNN for 3D cases.

3) After preprocessing the entire dataset, 200 sets of data were randomly selected as the training set, and 100 sets of data were used as the validation set. These datasets were input into the DE-DGM for training. Training stopped after 500 epochs.

4) With the trained model, the remaining 200 sets of data were used for prediction. The inputs consisted of variable conditions, and the outputs were the corresponding predicted topology optimization structures. These predicted structures were used as a warm-start for a second round of topology optimization using the MMA algorithm. Results and convergence speed were compared between the warm-start method and the non-warm-start method.

In the test set, four sets of data were selected with significant differences in the topology optimization structures for comparison. The predicted results by DE-DGM and their corresponding real structural data are shown in Fig. 13. By comparing the results of the upper and lower rows separately, it is evident that DE-DGM is capable of predicting the corresponding structures based on the given variable conditions. These predictions are in close alignment with the results obtained from the MMA applied to the test dataset.

To ensure stability, the material area in the output from DE-DGM was enlarged, and MMA was used for topology optimization again in the remaining area. The comparison between the topology optimization structures based on DE-DGM predictions and the results obtained solely using the MMA algorithm is illustrated in Fig. 14. It can be observed that in these four cases, the overall trends remain consistent before and after incorporating DE-DGM predictions. This suggests that the predictions made by DE-DGM have a certain level of guiding significance, and in most cases, they do not lead the structural optimization into local optima.

Due to the substantial computational complexity associated

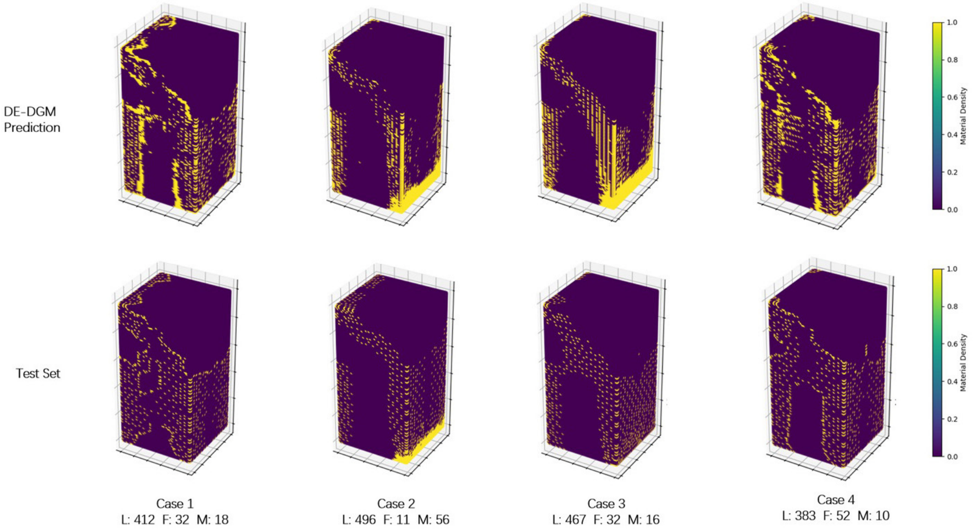


Fig. 13. Comparison of DE-DGM prediction results with test set results.

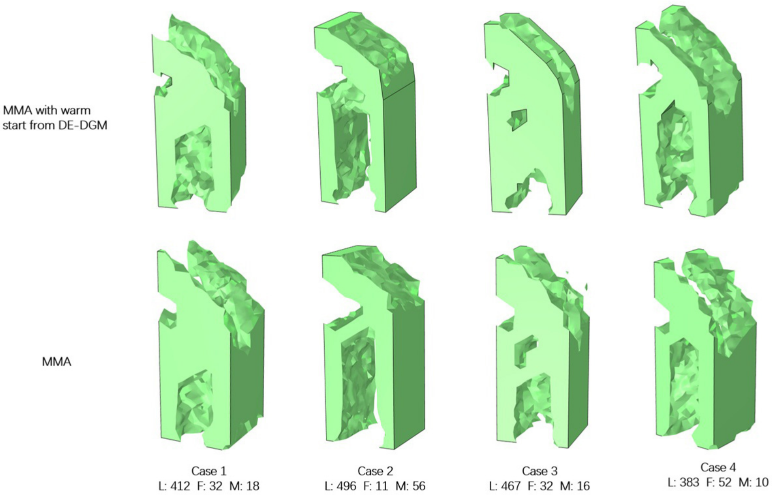
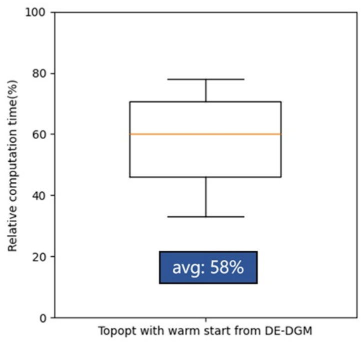


Fig. 14. Comparison of topology optimization results with warm-start and no-warm-start.

No.	L (mm)	F (kN)	M (kN·m)	Obj	Iter for MMA	Iter for MMA with warm start
1	467	32	16	1.976	100	33
2	412	32	18	2.475	100	38
3	496	11	56	4.753	100	43
4	497	14	54	4.824	100	55
5	477	24	48	5.203	100	57
6	483	26	60	7.395	100	63
7	313	17	49	7.013	100	70
8	491	52	52	9.809	100	71
9	332	37	45	9.208	100	76
10	408	39	44	7.426	100	78

(a)



(b)

Fig. 15. The performance of fast synthesis with DE-DGM for 3D column case: (a) selected ten sets of test data; (b) relative calculation time of warm-start topology optimization from DE-DGM prediction.

with 3D topology optimization, the current analysis focuses on a random selection of ten data sets to compare computational speeds, as shown in Fig. 15(a). The objective function values obtained after 100 iterations of the MMA algorithm when used independently were recorded. Using these values, the iteration counts at which they were achieved in the MMA algorithm's convergence curves after warm start were determined. Based on this information, the computational speed improvement was calculated, and an average reduction in computation time to 58 % of the original time was obtained, as shown in Fig. 15(b). Due to the relatively large number of grids and design variables associated with 3D cases, the computational workload was substantial. Therefore, the introduction of DE-DGM predictions did not yield as pronounced an effect on speeding up computations as observed in the previous section for two-dimensional cases. Additionally, an insufficient number of training epochs for the DE-DGM will result in reduced accuracy. A conservative approach to utilizing prediction results will also increase computation time in subsequent steps following the warm start.

5. Conclusion and future work

This paper proposes a two-stage data-efficient deep generative model for fast design synthesis/prototyping. Existing GAN-based DGMs are usually trained with O (10000) data, which limits engineering applications. The proposed DE-DGM demonstrated the feasibility of fast design synthesis based on 100-200 training data for 2-4 changing boundary condition variables. This was achieved through the introduction of prior constraints from P-POD. P-POD model was developed for first-stage low-fidelity design synthesis. The uncertainty of P-POD mode coefficients was modeled and propagated to the synthesized designs. Pixel-wise uncertainty of the synthesized designs could serve as the foundation for risk management and model improvement. Then these predictions (synthesized designs) are transferred to an encoder-decoder convolutional neural network for non-linear enhancement of fine features. The ED-CNN is formulated as image segmentation to categorize each pixel/voxel to zero/one (with/without material). The developed DE-DGM has demonstrated reasonable performance in synthesizing topology optimization with applications to compliance minimization, heat conduction, airplane bearing bracket design, and 3D machine tool column structure design. The DE-DGM predictions could guide design prototyping, or serve as an initial guess for high-fidelity topology optimization. The computation cost of warm-start topology optimization with initial density guess from DE-DGM predictions is 36 %-58 % of the standard cases.

It is encouraging to see the possibility of developing DE-DGM with O (100) data. Future research is needed to evaluate the technique with real designs, which could be more expensive 3D designs and more sophisticated boundary condition variables. A two-stage training process is developed in this paper. An end-to-end framework with simultaneous training might produce better results. To achieve this goal, more effort is needed to couple DE-DGM training tightly with the first-stage prior constraints. This work focuses on non-intrusive techniques which are general to handle different types of design synthesis, including the dataset from commercial software, historical designs, and parametric optimizations. It is possible to incorporate physical insights with DGM for improved accuracy. One potential direction is to couple compliance calculation with DGM training and explicitly tackle the compliance gap between simulations and DGM predictions. Latin hypercube sampling is adopted in this paper for the design of experiments. Adaptive sampling/active learning might be developed to reduce the size of the training dataset. An initial effort has been made to ascertain the uncertainty associated with synthesized designs in this issue. More research is needed for adaptive sampling of topology optimization designs.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant No. 52305288, the National Key R&D Program of China under Grant No. 2022 YFE0196400, the Zhejiang Provincial Natural Science Foundation of China under Grant No. LDT23E05013E05.

References

[1] J. D. Deaton and R. V. Grandhi, A survey of structural and multidisciplinary continuum topology optimization: post 2000, *Structural and Multidisciplinary Optimization*, 49 (1) (2014) 1-38.

[2] M. P. Bendsoe and O. Sigmund, *Topology Optimization: Theory, Methods, and Applications*, Springer, Heidelberg (2013).

[3] B. S. Lazarov, F. Wang and O. Sigmund, Length scale and manufacturability in density-based topology optimization, *Archive of Applied Mechanics*, 86 (1-2) (2016) 189-218.

[4] X. Qian, Topology optimization in b-spline space, *Computer Methods in Applied Mechanics and Engineering*, 265 (2013) 15-35.

[5] M. Zhou and R. Fleury, Fail-safe topology optimization, *Structural and Multidisciplinary Optimization*, 54 (5) (2016) 1225-1243.

[6] J. Liu, G. Parks and P. Clarkson, Metamorphic development: a new topology optimization method for continuum structures, *Structural and Multidisciplinary Optimization*, 20 (4) (2000) 288-300.

[7] A. V. Kumar and A. Parthasarathy, Topology optimization using b-spline finite elements, *Structural and Multidisciplinary Optimization*, 44 (4) (2011) 471.

[8] P. D. Dunning and H. Alicia Kim, A new hole insertion method for level set based structural topology optimization, *International J. for Numerical Methods in Engineering*, 93 (1) (2013) 118-134.

[9] S. Wang, K. M. Lim, B. C. Khoo and M. Y. Wang, An extended level set method for shape and topology optimization, *J. of Computational Physics*, 221 (1) (2007) 395-421.

[10] R. Picelli, S. Townsend, C. Brampton, J. Norato and H. A. Kim, Stress-based shape and topology optimization with the level set method, *Computer Methods in Applied Mechanics and Engineering*, 329 (2018) 1-23.

[11] D. Kelly, C. Reidsema, A. Bassandeh, G. Pearce and M. Lee, On interpreting load paths and identifying a load bearing topology from finite element analysis, *Finite Elements in Analysis and Design*, 47 (8) (2011) 867-876.

[12] R. Lanfear, B. Calcott, D. Kainer, C. Mayer and A. Stamatakis, Selecting optimal partitioning schemes for phylogenomic datasets, *BMC Evolutionary Biology*, 14 (1) (2014) 1-14.

[13] G. H. Yoon, Topology optimization for nonlinear dynamic problem with multiple materials and material-dependent boundary condition, *Finite Elements in Analysis and Design*, 47 (7) (2011) 753-763.

[14] T. S. Kim, J. E. Kim and Y. Y. Kim, Parallelized structural topology optimization for eigenvalue problems, *International J. of Solids and Structures*, 41 (9-10) (2004) 2623-2641.

[15] D. Jung and H. C. Gea, Topology optimization of nonlinear structures, *Finite Elements in Analysis and Design*, 40 (11) (2004) 1417-1427.

[16] E. Lee, K. A. James and J. R. Martins, Stress-constrained topology optimization with design-dependent loading, *Structural and Multidisciplinary Optimization*, 46 (5) (2012) 647-661.

[17] X. Tong, W. Ge, Y. Zhang and Z. Zhao, Topology design and analysis of compliant mechanisms with composite laminated plates, *J. of Mech. Sci. and Tech.*, 33 (2019) 613-620.

[18] R. C. Carbonari, E. C. Silva and G. H. Paulino, Multi-actuated functionally graded piezoelectric micro-tools design: a mutiphysics topology optimization approach, *International J. for Numerical Methods in Engineering*, 77 (3) (2009) 301-336.

[19] J. Zhu, H. Zhou, C. Wang, L. Zhou, S. Yuan and W. Zhang, A review of topology optimization for additive manufacturing: Status and challenges, *Chinese J. of Aeronautics*, 34 (1) (2020) 91-110.

[20] J. Wu, N. Aage, R. Westermann and O. Sigmund, Infill optimization for additive manufacturing—approaching bone-like porous structures, *IEEE Transactions on Visualization and Computer Graphics*, 24 (2) (2017) 1127-1140.

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

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

[23] J. Shin and C. Kim, Bi-directional evolutionary 3D topology optimization with a deep neural network, *J. of Mech. Sci. and Tech.*, 36 (7) (2022) 3509-3519.

[24] T. Guo, D. J. Lohan, R. Cang, M. Y. Ren and J. T. Allison, An indirect design representation for topology optimization using variational autoencoder and style transfer, *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conferencei*, Kissimmee, Florida, USA (2018) 0804.

[25] S. Rawat and M. H. Shen, Application of adversarial networks for 3d structural topology optimization, *SAE Technical Paper* (2019) 2019-01-0829.

[26] E. Ulu, R. Zhang and L. B. Kara, A data-driven investigation and estimation of optimal topologies under variable loading configurations, *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 4 (2) (2016) 61-72.

[27] A. Raina, C. McComb and J. Cagan, Learning to design from humans: Imitating human designers through deep learning, *J. of Mechanical Design*, 141 (11) (2019) 111102.

[28] S. Banga, H. Gehani, S. Bhilare, S. Patel and L. Kara, 3D topology optimization using convolutional neural networks, *arXiv:1808.07440* (2018).

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

[30] C. Qiu, S. Du and J. Yang, A deep learning approach for efficient topology optimization based on the element removal strategy, *Materials & Design*, 212 (2021) 110179.

[31] A. J. Lew and M. J. Buehler, Encoding and exploring latent design space of optimal material structures via a VAE-LSTM model, *Forces in Mechanics*, 5 (2021) 100054.

[32] T. Bui-Thanh, M. Damodaran and K. Willcox, Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics, *21st AIAA Applied Aerodynamics Conference*, Orlando, Florida, USA (2003) 4213.

[33] C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, *International J. of Bifurcation and Chaos*, 15 (3) (2005) 997-1013.

[34] Y. Yu, T. Hur and J. Jung, Deep learning for topology optimization design, *arXiv:1801.05463* (2018).

[35] W. Chen and F. Ahmed, Padgan: learning to generate high-quality novel designs, *J. of Mechanical Design*, 143 (3) (2021) 031703.

[36] Z. Yang, X. Li, L. Catherine Brinson, A. N. Choudhary, W. Chen and A. Agrawal, Microstructural materials design via deep adversarial learning methodology, *J. of Mechanical Design*, 140 (11) (2018) 111416.

[37] Y. Qiu and J. Bai, Stationary flow fields prediction of variable physical domain based on proper orthogonal decomposition and kriging surrogate model, *Chinese J. of Aeronautics*, 28 (1) (2015) 44-56.

[38] J. Weiss, A tutorial on the proper orthogonal decomposition, *AIAA Aviation 2019 Forum*, Dallas, USA (2019) 3333.

[39] K. Lu, Y. Jin, Y. Chen, Y. Yang, L. Hou, Z. Zhang, Z. Li and C. Fu, Review for order reduction based on proper orthogonal decomposition and outlooks of applications in mechanical systems, *Mechanical Systems and Signal Processing*, 123 (2019) 264-297.

[40] S. Ghosh, P. Pandita, S. Atkinson, W. Subber, Y. Zhang, N. C. Kumar, S. Chakrabarti and L. Wang, Advances in bayesian probabilistic modeling for industrial applications, *ASCE-ASME J. Risk and Uncert. in Engrg. Sys. Part B: Mech. Engrg.*, 6 (3) (2020) 030904.

[41] Y. Zhang, N. H. Kim and R. T. Haftka, General-surrogate adaptive sampling using interquartile range for design space exploration, *J. of Mechanical Design*, 142 (5) (2020).

[42] Y. Zhang, J. Kristensen, S. Ghosh, T. Vandeputte, J. Tallman

and L. Wang, Finding maximum expected improvement for high-dimensional design optimization, *AIAA Aviation 2019 Forum*, Dallas, Texas, USA (2019) 2985.

[43] GrabCAD, *Airplane Bearing Bracket Challenge*, GrabCAD (2016).

[44] T. Karras, S. Laine and T. Aila, A style-based generator architecture for generative adversarial networks, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Ling Beach, CA, USA (2019) 4396-4405.

[45] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila, Analyzing and improving the image quality of styleGAN, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA (2020) 8107-8116.

[46] M. Mirza and S. Osindero, Conditional generative adversarial nets, *arXiv:1411.1784* (2014).

[47] F. Löffelmann, Failure index based topology optimization for multiple properties, *Proceedings of the 23rd International Conference on Engineering Mechanics*, Svratka, Czech Republic (2017).

Yiming Zhang is a Professor of Mechanical Engineering, Zhejiang Uni-versity, China. He received his Ph.D. from the University of Florida, US. His research focuses on design optimization, probabilistic machine learning, optimization and decision making under uncertainty.