International Journal of Mechanical Design and Materials Science

Volume 1, Issue 4 2025

TOPOLOgyGAN: OPTIMIZATION OF TOPOLOgy USING GENERATIVE ADVANCED NETWORKS BASED ON PHYSICAL FIELDS OVER THE INITIAL DOMAIN

Sajad Mirjavadi **.**Department of Mechanical and Industrial Engineering, Qatar University, P.O. Box 2713, Doha, Qatar

Research Article

Date of Submission: 10-07-2025 Date of Acceptance: 19-08-2025 Date of Publication: 1-09-2025

ABSTRACT

Load and boundary constraints expressed as vectors or sparse matrices in deep learning topology optimization often lack the chance to incorporate a full understanding of the design challenge, resulting in less than optimal generalization outcomes. We present TopologyGAN, a novel data-driven topology optimization approach that uses different physical fields calculated on the initial, unoptimized material domain as inputs to the generator of a conditional generative adversarial network (cGAN). On test issues with previously unknown boundary conditions, TopologyGAN achieves a roughly 3× decrease in the mean squared error and a 2.5× reduction in the mean absolute error when compared to a baseline cGAN. We also provide a hybrid network for the generator dubbed U-SE (Squeeze-and-Excitation)-ResNet, which is based on a number of current network models and significantly improves overall accuracy. We make our trained network and complete implementation available to the public.

Keywords: U-SE-ResNet, TopologyGAN, Deep Learning, Topology Optimization, and Physical Fields

Introduction

Subject to a set of displacement and load boundary constraints, topology optimization of solid structures entails creating optimal forms by minimizing an objective function, such as compliance or mass, within a material domain (Figure 1). Topology optimization is becoming more and more common because to the fast advancements in additive manufacturing and related design tools, which enable the automated creation of optimal structures. Density-based approaches, like the Solid Isotropic Material with Penalization (SIMP) method [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], grid-based approaches [11, 12, 13, 14, 15, 16], moving boundary-based approaches [17, 18, 19, 20, 21, 22, 23, 24, 25], and load-path-based approaches [26, 27, 28]. Topology optimization techniques continue to be computationally intensive and are not easily integrated into other design optimization modules, such as layout or configuration design tools, despite considerable attempts to increase solution efficiency [29]. To speed up the process, new data-driven techniques for topology optimization have been put forward in recent years. In terms of form similarity, deep learning techniques have shown promise in effectively generating near-optimal solutions as

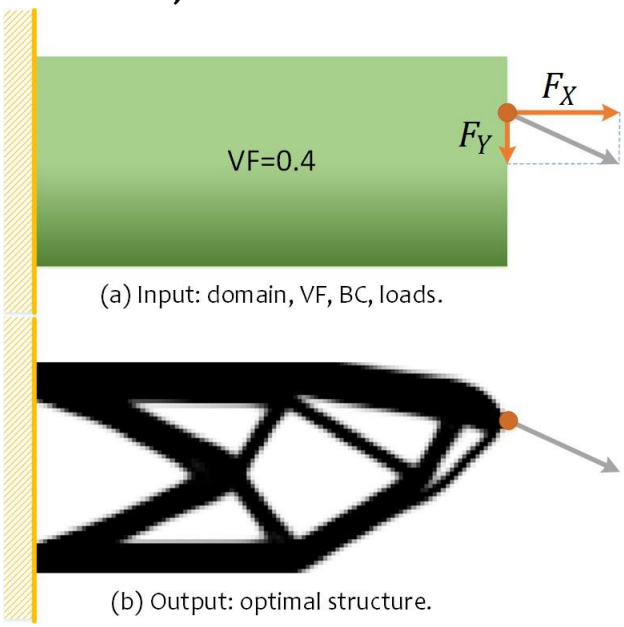


Figure 1: 2D topology optimization.

as well as adherence to minimal run-time expenses [30, 31, 32, 33, 34]. The mapping between the design variables and the external boundary conditions is established using theory-guided machine learning techniques using domain-specific theories [35, 36, 37]. However, developing a precise and broadly applicable mapping from the boundary conditions to the final optimum structure is a major difficulty in topology optimization. Because the findings are hard to apply to new boundary conditions, methods that directly construct this map often have to strongly constrain the displacement and external load configurations.

We suggest a novel deep learning-based generative model for topology optimization termed TopologyGAN as a first step in resolving this problem. Conditional Generative Adversarial Networks (cGAN) are the foundation of TopologyGAN. Our main hypothesis is that different physical fields computed on the initial, unoptimized domain subject to the specified boundary conditions may contain useful information that enables the network to learn more accurate maps, instead of attempting to map the boundary conditions to the resulting optimal shapes directly. Our observation that the input displacement and load boundary conditions are often expressed as sparse vectors or matrices2 in deep learning-based systems served as the impetus for our strategy. The network is unable to take advantage of beneficial spatial variations and physical processes that take place inside the material world because of this sparsity. In contrast, we suggest using these differences in this work.

In order to achieve this, in TopologyGAN (i) the generator's input channels are connected to the physical fields (non-sparse matrices) calculated in the initial, unoptimized domain, such as the von Mises stress fields, strain energy fields, and displacement fields, rather than just the original boundary conditions; and (ii) the discriminator's condition involves both the physical fields and the initial inputs. Although our method may be applied to ground truth data from various topology optimization techniques, we utilize ground truth data produced by the SIMP method.

The von Mises stress field and the strain energy density fields are the most valuable channels of information to supplement the initial displacement and load boundary conditions, according to this formulation and the model selection studies we carry out. Our findings demonstrate that TopologyGAN delivers lower test errors than the baseline method's training error on previously unknown boundary

circumstances when compared to a baseline cGAN model that does not use such fields. These findings are promising since they may be helpful to other researchers who want to investigate the use of such fields in comparable topology optimization situations. Our trained network and complete implementation are available to the public at https://github.com/zhenguonie/TopologyGAN. Upon request, the whole data set used in this study is publicly accessible.Our primary contributions are:
2Unless, of course, they are fixed, in which case changes in these circumstances cannot be taken into account by the network.
• TopologyGAN, a novel approach to topology optimization using deep learning models.
• A new input matrix design that incorporates the original physical fields. The original problem input matrices are enhanced by this input.
• U-SE-ResNet, a hybrid neural network architecture, serves as TopologyGAN's generator.

Related Work

Our review focuses on studies that highlight topology optimization, deep learning for topology optimization, generative adversarial networks (GAN), and two network architectures closely related to our work.

Topology Optimization and SIMP

Topology optimization seeks an optimal subset $\Omega_{mat} \subset \Omega$, where Ω is the design domain. To formulate this problem, an objective function $f(y)$ to be minimized is defined as in Eq.(1), in which y denotes the structural design (material distribution)[3] and $h(y)$ is a resulting physical outcome such as stress, strain or displacement.

vectors, $\mathbf{K}$ is the stiffness matrix, $\mathbf{u}_e$ is the elemental displacement vector, $\mathbf{k}_e$ is the elemental stiffness matrix, and N is the number of total elements.
While the above existing methods can provide optimized solutions, our work aims to accelerate the iterative nature of these solvers using a data-driven approach.

Deep Learning for Topology Optimization

Deep neural networks have been extensively used in a variety of sectors, including autonomous cars, robotics, medical diagnostics, biomedicine, material design, machine health monitoring, mechanical design, and manufacturing, thanks to recent advancements in computer technology. Deep neural networks have shown efficacy in learning intricate mappings between goal design objectives and issue input variables and limitations. Engineering design exploration and optimization, as well as the mapping out of possible portions of the design space, have shown the efficacy of supervised machine learning algorithms [43]. A data-driven design representation is suggested by Guo et al. [30], where an enhanced variational

3In order to maintain consistency with the cGAN output shown in the subsequent sections, we use y for material distribution.
2D topologies are encoded into a lower-dimensional latent space using an autoencoder, which then decodes samples from this space back into 2D topologies. A deep generative adversarial system that can produce a large number of aesthetically pleasing and engineering performance-optimized design alternatives is proposed by Oh et al. [44]. In order to create novel 3D structures utilizing small data samples, Rawat and Shen [32] combine a convolutional neural network (CNN) and Wasserstein GAN into a coupled model for 3D topology optimization. Lin et al. [45] provide a deep learning method using U-Nets to accelerate convergence in SIMP-based topology optimization for heat conduction. Only the first SIMP findings are input into the deep learning network to directly generate the final outputs. CNNs are used by Sosnovik and Oseledets [46] to speed up topology optimization from two midway grayscale pictures to the final binary image produced by SIMP. Nevertheless, these networks do not provide an end-

to-end mapping from the boundary conditions to the optimal topologies; instead, they concentrate on either latent candidate generation or rapid optimization.
Yu et al. [31] provide a CNN-based encoder-decoder for the creation of low-resolution structures, which are then run through a super-resolution GAN to produce the final results, in order to achieve an end-to-end topology optimization from specified boundary conditions. The use of cGANs to provide a compact latent representation of structures arising from topology optimization is investigated by Sharpe and Seepersad [33]. Nevertheless, only a small number of optimization parameters and boundary conditions are taken into account. It is still very difficult to extend data-driven topology optimization to new displacement and external load scenarios.

Generative Adversarial Networks

Generative Adversarial Networks (GAN) [47], is a generative model formulated as a minimax two-player game between two models. It consists of: (1) A generator G whose aim is to learn a generative density function that models the training data and, (2) a discriminator D that aims to discern if an input sample is part of the original training set or a synthetic one generated by G. The structure of GAN is shown in Figure 2. The input to G is random noise z sampled from a distribution $p_z(z)$. The output of G, $y_g = G(z)$, is a fake data. A real y_r or a fake sample y_g is then fed into D to obtain an output $D(y)$, which generates a probability estimate of the true nature of y. In this case, D is trained to maximize the probability of assigning the correct label to both the real samples y_r and fake samples y_g. G is simultaneously trained to minimize $\log(1 - D(G(z)))$. The training loss functions L^{GAN}_{D} and L^{GAN}_{G} are:

$$\max_D L^{GAN}_D = E_{y \sim p_{data}(y)}[\log D(y_r)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (4)$$

$$\min_G L^{GAN}_G = E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (5)$$

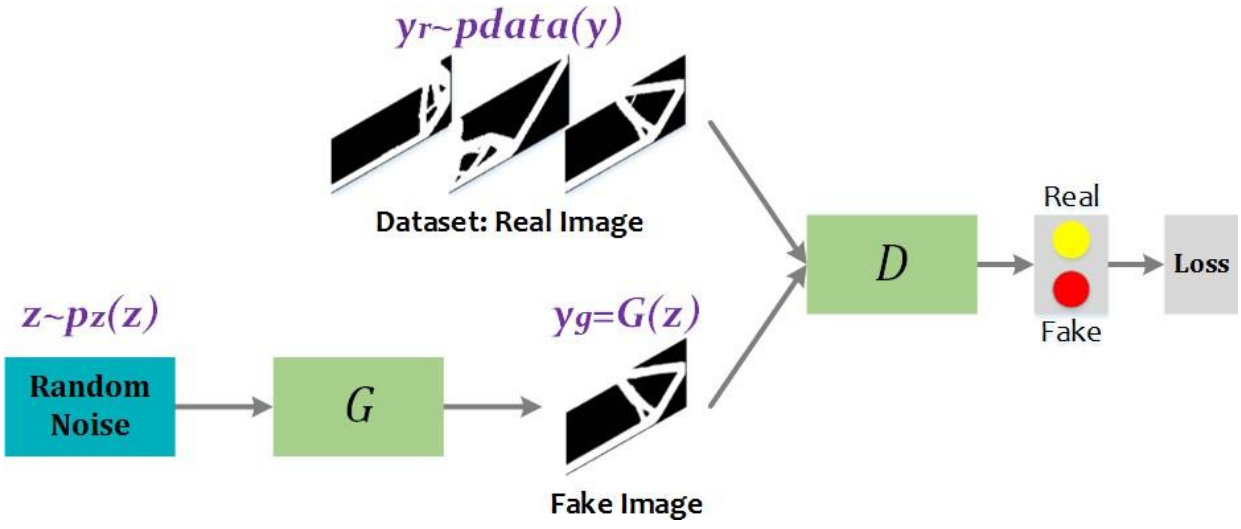


Figure 2: Schematic diagram of GAN.

Conditional GAN Generative adversarial networks can be extended to a conditional model when the generator and the discriminator are conditioned on prescribed information [48]. A conditional GAN

(cGAN) learns a mapping from an input x to an output y, as shown in Figure 3. The cGAN loss ([49]) is:

$$L^{cGAN} = \mathrm{E}_{(x,y)\sim p_{data}}[\log D(x,y)] + \mathrm{E}_{G,D} \quad (6)$$

$E_{x\sim pdata}(x),z\sim pz(z)[\log(1-D(x,G(x,z)))]$

$L_{L1}(G) = E_{x,y,z}[\|y - G(x,z)\|_1]$ (7)

$G^* = \arg\min\max L^{cGAN} + \lambda L_{L1}(G)$ (8)
$G \quad D \quad G,D$
where G^* is the final optimized generative model. In this model, the inputs x are composed of the original (full) domain, the desired volume fraction, the displacement boundary conditions, and the external loads. These are utilized as conditions that the generator has to attune to. The ground truth final optimized structure (real structure) y is computed
through the SIMP method and provided as input to the discriminator, alongside x for training.

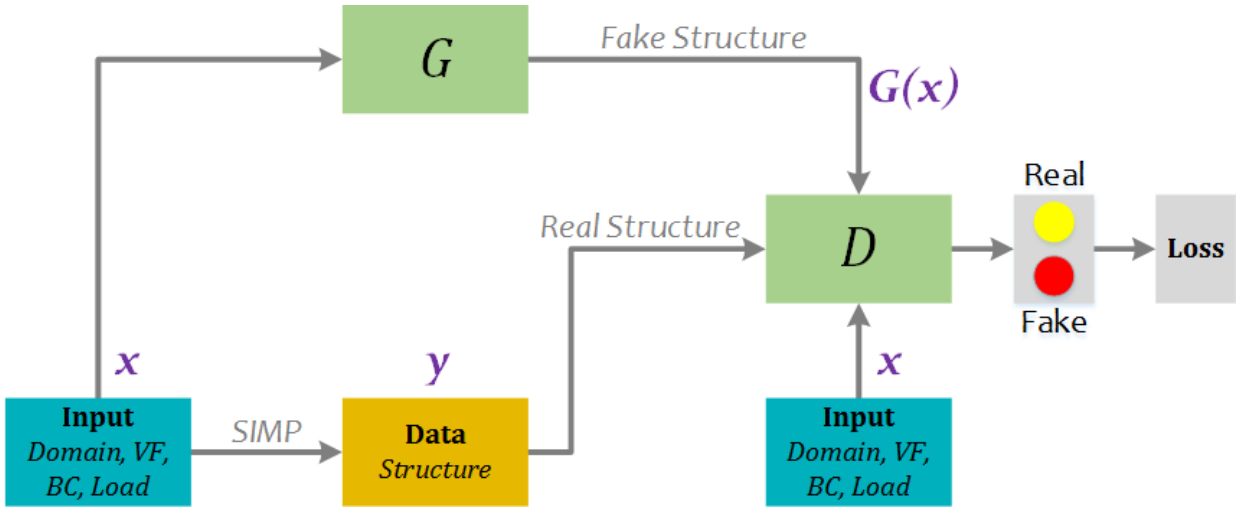


Figure 3: Baseline: cGAN for topology optimization.

U-Net and SE-ResNet
The network may transfer context information from the downsampling levels to the upsampling layers at different resolutions thanks to the U-Net design ([50]) seen in Figure 4. By reflecting the input picture, the missing context is extended to forecast the output around the image's boundary. When the network is applied to huge pictures, this tiling technique is crucial because without it, the GPU capacity would restrict the resolution [51].
As shown in Figure 5, SE-ResNet is a CNN-based ResNet [52] that has been improved using SE-blocks [53]. There are two CNN blocks and one SE block in each SE-ResNet module [54]. The skip connection, which only adds identity mapping to the output of the stacked layers, is what sets ResNet apart. For the intended underlying mapping, ResNet may thus dynamically choose the layer depth. A feedforward neural network with a shortcut link computes the SE-ResNet module's final output: w = v + x. The SE block is used in SE-ResNet to enable dynamic channel-wise feature recalibration, which increases the network's representational capacity. The global average-pooling layer shrinks the input data u ∈ RH×W×C to S (u) ∈ RC. The linear array S (u) is then downsampled and upsampled using two completely linked layers. The excitation output E(u), which has the same dimensions as the original input u, is produced by a reshape operation. Through element-wise matrix multiplication, the SE block's ultimate output is a Hadamard product of E(u) and u: v = E(u) ÷ u.

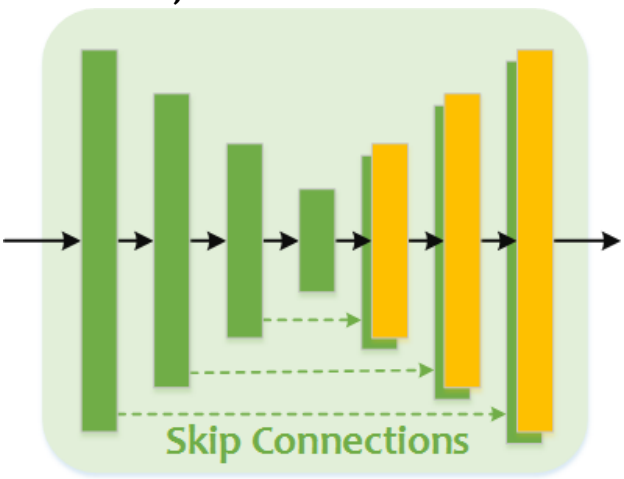


Figure 4: U-Net

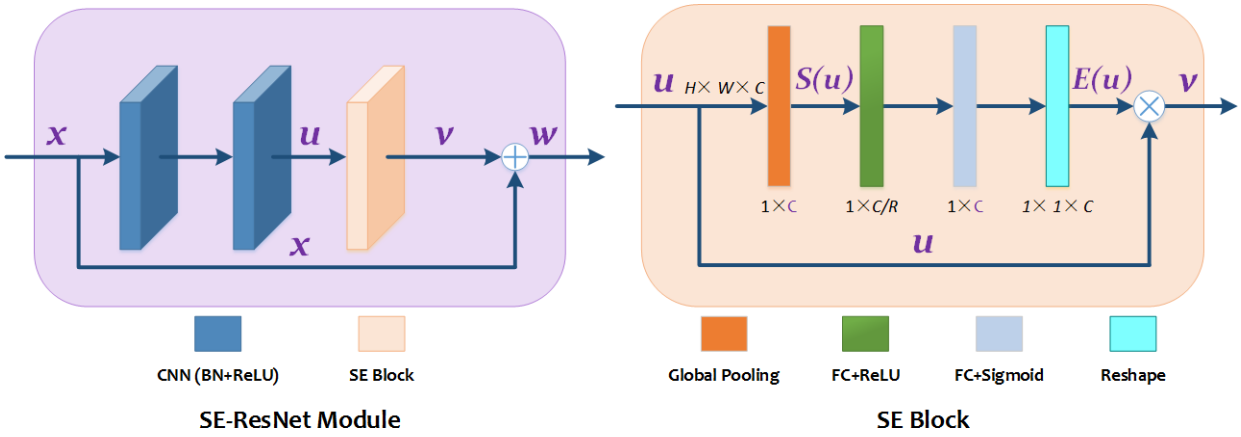


Figure 5: ResNet enhanced by SE block [54]

Technical Approach

The high discreteness of the boundary conditions and the sparsity of the input matrices make it challenging to extend topology optimization using deep learning to previously uncovered boundary circumstances. The mapping function has a significant variance due to the input matrices' high sparsity. We suggest a novel model named TopologyGAN as a first step toward resolving this issue. In order to supplement the baseline cGAN with this additional data, the approach relies on a number of physical fields, including the strain energy field, von Mises stress field, displacement fields, and strain fields calculated on the original (unoptimized) design domain. These are referred to as the beginning fields f in this study. Figure 6 provides an illustrated schematic of TopologyGAN's operation. The issue input matrices that encode the intended VF, displacement BC, and load make up the horizontal axis. The vertical axis represents the final structure, with three structural designs produced by the cGAN as the baseline, the SIMP technique as the ground truth, and TopologyGAN as the suggested model. The inputs are directly mapped to the output structure by the baseline cGAN: x cGAN y. In contrast, our suggested TopologyGAN creates a mapping from the inputs x to the starting fields f, then a mapping from f to the output structure y as follows: x FEM

TopologyGAN f ———————— × y. Keep in mind that the initial fields are only calculated once during

run time.

By using such starting fields, we hypothesize that the network may further leverage the relevant information they give about the physical state of the original domain under the inputs x. For example, Figure 6 illustrates how the von Mises stress maps and the initial strain energy density both provide scalar fields with more information than the original issue input matrices x alone. Specifically, it is expected that the initial fields would provide information f (green solid curve) that corresponds well with the end structure y (yellow dotted curve), making the remaining job of the network less difficult than if x were directly mapped to y. The initial fields do, in fact, enable TopologyGAN achieve far greater training and test accuracies than the baseline cGAN, as will be shown in Section 5.1.

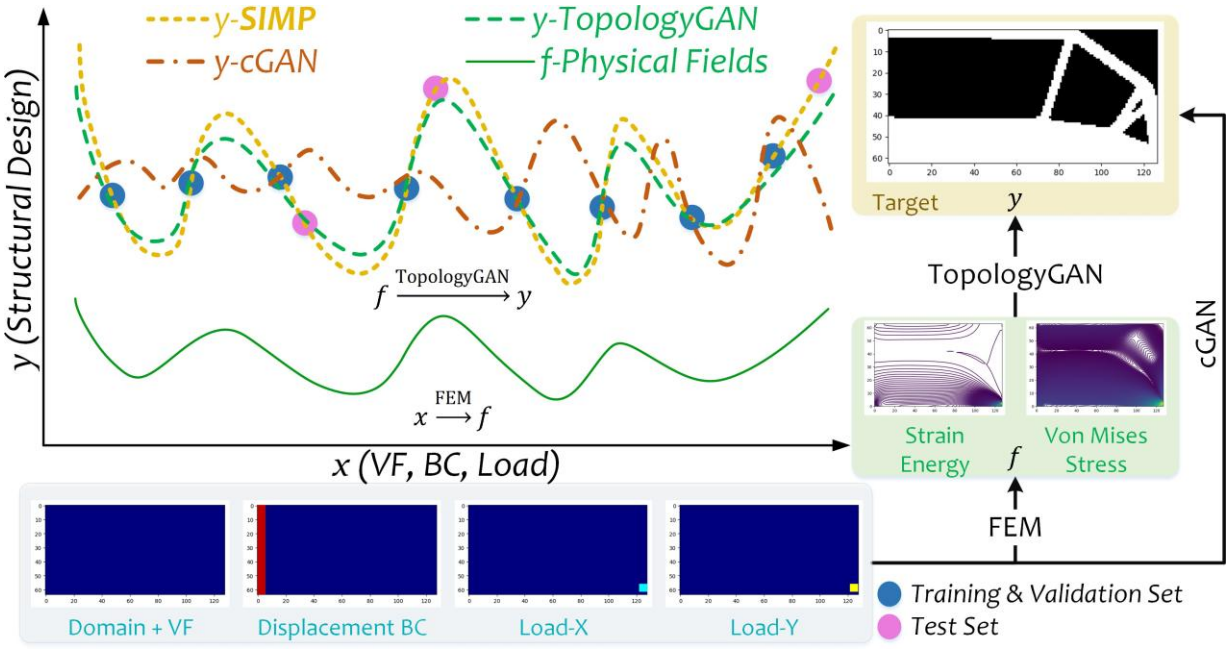


Figure 6: Schematic diagram of TopologyGAN.

Network Architecture of TopologyGAN

TopologyGAN is based on the conditional Pixel2Pixel GAN architecture [49] and incorporates the FEM and the SIMP solver, as shown in Figure 7. The generator of TopologyGAN establishes a mapping from the initial physical fields f to the estimated optimized topology $\hat{y} = G(f(x), \text{VF})$. The ground truth output y is generated by the SIMP method, as was the case with the baseline cGAN shown in Figure 3. In TopoloyGAN, both the problem inputs x and the initial fields $f(x)$ are used as the condition: $r(x) = [x, f(x)]$.

Both the generator and the discriminator utilize CNN blocks as shown in Figure 8. For the generator, we propose a hybrid architecture called U-SE-ResNet. U-SE-ResNet is a U-Net with the SE-ResNet module in a downsampling-upsampling structure. In the upsampling, transposed convolution layers are used instead of pooling layers. According to [49], PatchGAN discriminator is adopted because the problem of blurry images caused by failures at high-frequency features such as edges and textures can be alleviated by restricting the GAN discriminator to only model high frequencies. The Sigmoid function is used in the generator's last layer for outputting a grayscale structure $\hat{y}$.

TopologyGAN Loss Function

As shown in Eq.(9) - Eq.(12), the loss consists of three parts: (1) the loss of TopologyGAN: L^{TGAN}, (2) the L2 loss of the generator: $L_{L2}(G)$, and (3) the absolute error of the volume fraction of the generator: AE^{VF}.

$$L^{TGAN} = E_{(x,y)\sim p_{data}} [\log D(r(x), y)] + E_{x\sim p_{data}(x)} [\log(1 - D(r(x), G(f(x))))] \quad (9)$$

$$L^{TGAN} = E_{x\sim p_{data}(x)} [\log(1 - D(r(x), G(f(x))))] \quad (10)$$

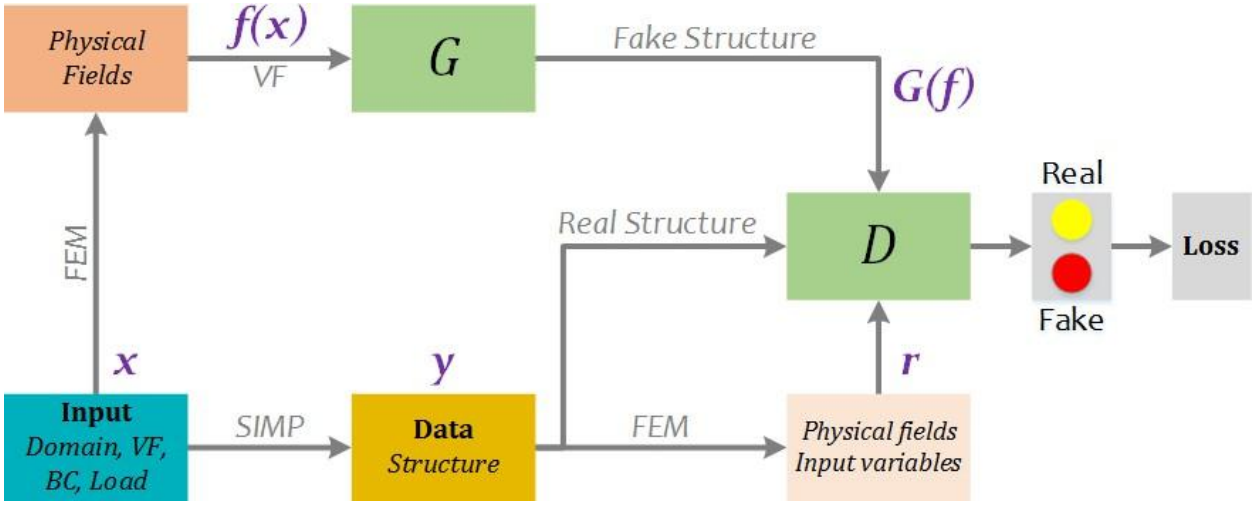


Figure 7: TopologyGAN approach to topology optimization.

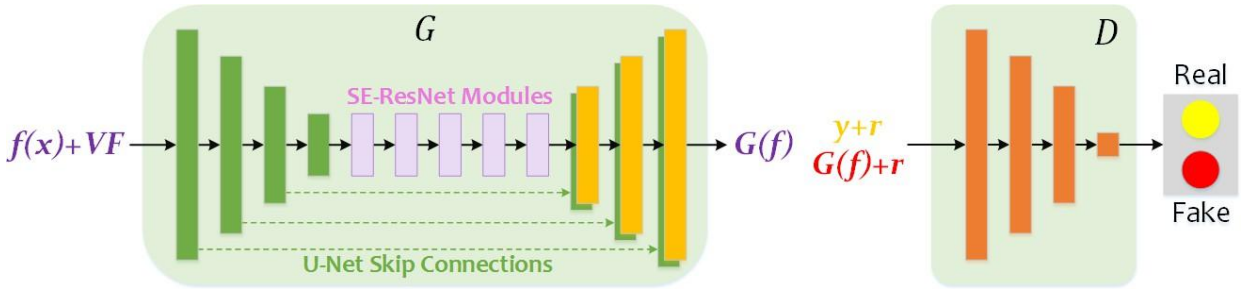


Figure 8: Architectures of the generator *G* and discriminator *D*.

$$L_{L2}(G) = E_{x,y}[\|y - G(f(x))\|_2] \quad (11)$$

$$AE = |VF - \widehat{VF}| = \frac{1}{G}\left|\sum_{e=1}^{N}(y_e - G(f(x)^{VF})_e)\right| \quad (12)$$

$$G^* = \arg\max\min L^{TGAN} + \lambda_1 L_{L2}(G) + \lambda_2 AE^{VF} \quad (13)$$

D *G* *G,D* *G*

where the generator *G* tries to minimize the combined loss $L^{TGAN} + \lambda_1 L_{L2}(G) + \lambda_2 AE^{VF}$. At the same time, the
G *G*
adversarial *D* tries to maximize$_{G,D}$ L^{TGAN} . $r(x)$ is the condition that contains the problem inputs x and the initial physical fields $f(x)$. n is pixel count of the output image. Scalars λ_1 and λ_2 are used to balance the three loss functions. The two scalars are determined empirically using parametric studies. In this paper, λ_1

= 10,000, λ_2 = 1.
We select the cGAN as our baseline model. Its structure is shown in Figure 3. Its loss is shown in Eq.(8).

Experiments
The proposed TopologyGAN and the baseline cGAN are trained on the same data set that is generated by the SIMP method and then are evaluated and compared on their prediction performance.

Experiment Design

Our code is written in TensorFlow and trained on a NVIDIA GeForce GTX 2080Ti GPU. Adam [55] is used for optimization, combining the advantages of AdaGrad [56] and RMSProp [57].

In addition to a comparison of TopologyGAN and the baseline cGAN, we also study the impact of different physical fields as inputs. Additionally, we compare different generator structures, namely U-Net, SE-ResNet, and the proposed U-SE-ResNet.

Data set

A SIMP-based topology optimization framework called ToPy [58] is used to generate the data set. The 2D domain consists of a 64 × 128 grid structure consisting of square elements. A total of 49, 078 optimized structures are generated using ToPy with the randomized conditions as follows:

volume fraction: [0.3 : 0.02 : 0.5]

displacement BCs: 42 scenarios

load position: any point on the boundary of the domain

load direction: ˙0 : π : 2π˙

SIMP penalty: 2

SIMP filter radius: 1.5

The data set is divided into training, validation, and test sets as follows: 4 displacement BCs of the total 42 are randomly selected as the test set. All data samples from the remaining 38 displacement boundary conditions are shuffled and split into training (80%) and validation (20%) sets. Note that the boundary conditions in the test set will not have been seen by the network during training.

Six samples are randomly selected from the entire data set for an illustration of the input and outputs in Figure 9. Images in each row form a sample for TopologyGAN. The first four images in each row are input variables x(VF,BCs) including volume fraction, displacement boundary condition and load boundary conditions along the x-axis and the y-axis. VF is input as a 2D matrix that has the same dimension as the design domain. The value of each element in the matrix is equal to the VF value. Displacement BCs are represented as a 2D matrix by assigning one of the four integers to each element: 0 represents unconstrained, 1 represents $u_x = 0$, 2 represents $u_y = 0$, and 3 represents $u_x = u_y = 0$. Based on the input variables, the initial stress ($\sigma = [\sigma_{11}, \sigma_{22}, \sigma_{12}]$), strain ($\varepsilon = [\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12}]$), and displacement ($u = [u_x, u_y]$) fields are computed by FEM. Strain energy density and von Mises stress are shown as the example physical fields, which are expressed in Eq(14) and Eq(15). The last image is the output structure y from SIMP.

$$\sigma_{vm} = \sqrt{\sigma_{11}^2 - \sigma_{11}\sigma_{22} + \sigma_{22}^2 + 3\sigma_{12}^2} \quad (14)$$

$$W = \frac{1}{2}(\sigma_{11}\varepsilon_{11} + \sigma_{22}\varepsilon_{22} + 2\sigma_{12}\varepsilon_{12}) \quad (15)$$

Evaluation Metrics

To assess the performance of TopologyGAN, we use the following four metrics: mean absolute error (MAE), mean squared error (MSE), relative error of volume fraction (AE^{VF}) and its absolute value ($|AE^{VF}|$), relative error of the compliance (RE^{C}) and its absolute value ($|RE^{C}|$). Denote $\hat{y}$ as the prediction from the generator G:

$$\hat{y} = G(f(x)), \quad (16)$$

and then both the ground truth y and the prediction $\hat{y}$ are reshaped into vectors, with the length of 8,192, from 2D matrices, with the size of 64 × 128.

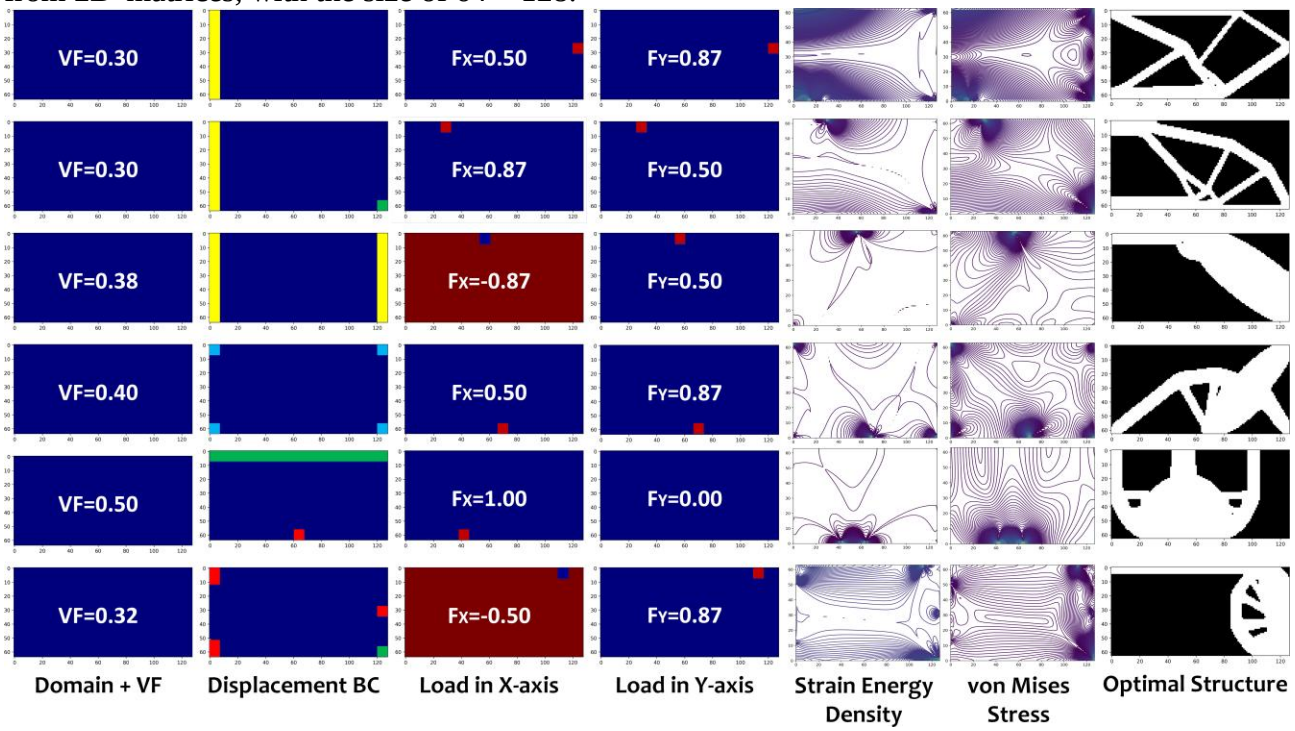


Figure 9: Data samples generated by FEM solver and SIMP solver. Images in each row form one sample. The first four images in each row are input variables x(VF,BC,Load), the last image is output y. Strain energy density and von Mises stress are shown in level sets with contour lines.

MAE shown in Eq.(17) and MSE shown in Eq.(18) are used to evaluate the model. MAE measures the average magnitude of the absolute differences between the prediction values and the ground truth. MSE measures the average squared difference between the estimated values and the ground truth.

$$\text{MAE} = \frac{1}{M}\sum_{i=1}^{M}|y^{(i)} - \hat{y}^{(i)}| = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N}\sum_{e=1}^{N}|y_e^{(i)} - \hat{y}_e^{(i)}| \quad (17)$$

$$\text{MSE} = \frac{1}{M}\sum_{i=1}^{M}(y^{(i)} - \hat{y}^{(i)})^2 = \frac{1}{M}\sum_{e=1}^{N}\frac{1}{N}\sum_{e}(y^{(i)} - \hat{y}^{(i)})^2 \quad (18)$$

where M is the total number of data samples, and $N = 8{,}192$ is the number of grid elements.
In addition to these commonly used metrics, we define two other metrics for evaluation: (1) RE^{VF} is the relative error of the volume fraction between prediction output and ground truth output, and (2) RE^{C} is the ratio between the compliance of the predicted structure and the ground truth structure. These are defined as follows:

$$\text{VF} = \frac{1}{N}\sum_{e=1}^{N} y_e \quad (19)$$

$$\text{RE}^{\text{VF}} = \frac{\widehat{\text{VF}} - \text{VF}}{\text{VF}} = \frac{\sum_{e=1}^{N}(\hat{y}_e - y_e)}{\sum_{e=1}^{N} y_e} \quad (20)$$

$$C(y) = \mathbf{U}^T\mathbf{K}\mathbf{U} = \sum_{e=1}^{N}(y_e)^p\mathbf{u}^T\mathbf{k}_0\mathbf{u}_e \quad (21)$$

$$\text{RE}^{\text{C}} = \frac{C(\hat{y}) - C(y)}{C(y)} \quad (22)$$

where $C(y)$ is the compliance of the predicted structure under the given loads.

Results and Discussions

We find that TopologyGAN outperforms the baseline cGAN in both the training and test. Furthermore, we analyze the physical field selection and the network architecture for the generator. We find that the VF+σ_{vm}+W combination performs significantly better than all other combinations. Moreover, to find the best generator structure, we compare the U-SE-ResNet, U-Net, and SE-ResNet. A comparative experiment is conducted using different generators on the same discriminator and the data set. The results show that the proposed U-SE-ResNet outperforms the other two structures.

Model Evaluation

Our training results of TopologyGAN and the baseline cGAN are summarized in Table 1. It can be seen that TopoloyGAN achieves a 3× lower MSE error and 2.5× lower MAE error than that of the cGAN. More surprisingly, both the MSE and MAE of TopologyGAN on the *test* set are lower than those of cGAN on the *training* set.

Loss functions of TopologyGAN are shown in Figure 10, where $L^{TGAN}_{G,D}$ is the discriminator loss of TopologyGAN,

$L^{TGAN}_{G}$is the generator loss of TopologyGAN, and G^* is the whole objective of the generator. As the training progresses,

it can be seen that L^{TGAN} and G^* decrease gradually, and L^{TGAN} oscillates and tends to balance.

G G,D

Table 1: Comparison of results between TopologyGAN and cGAN. Model MAE MSE

	Training	Validation	Test	Training	Validation	Test
TopologyGAN	0.001808	0.019133	0.070128	0.001340	0.018022	0.059943
Baseline: cGAN	0.088257	0.100565	0.181095	0.085916	0.097966	0.175226

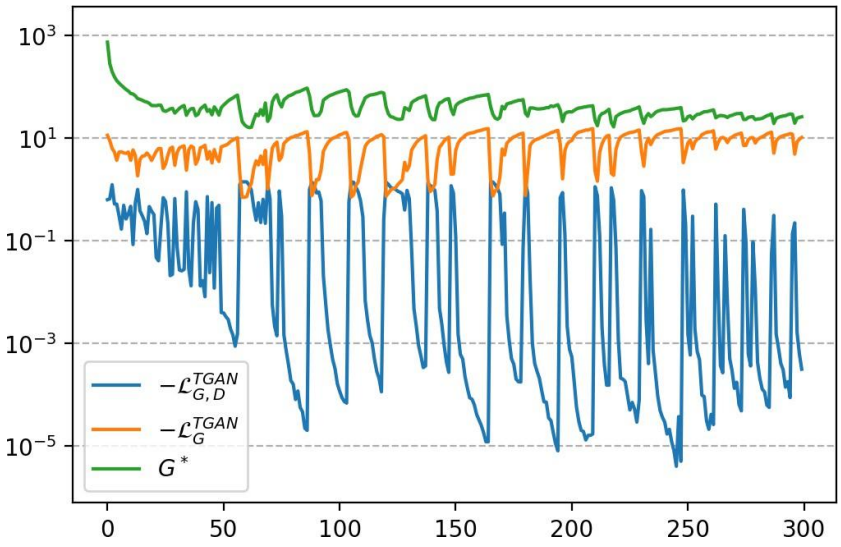


Figure 10: Loss functions of TGAN, Discriminator and Generator in training.

Accuracy and Performance

The accuracy and performance of the trained TopologyGAN are discussed to show the prediction performance. As shown in Figure 12, the generated structures from TopologyGAN become increasingly more defined over the training epochs. Each row denotes a randomly selected sample from the training set. After two-hundred epochs, the predictions become virtually indistinguishable from the ground truth (GT) structures.

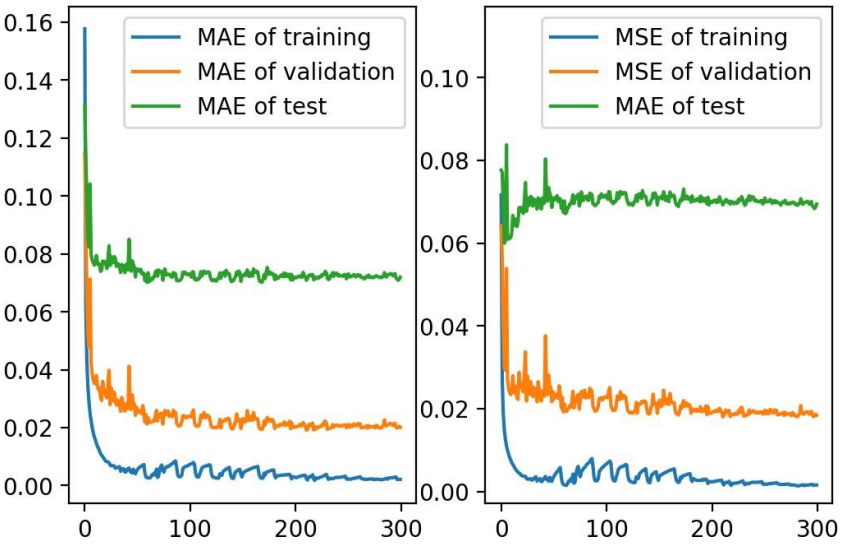


Figure 11: MAE and MSE of TopologyGAN in training.

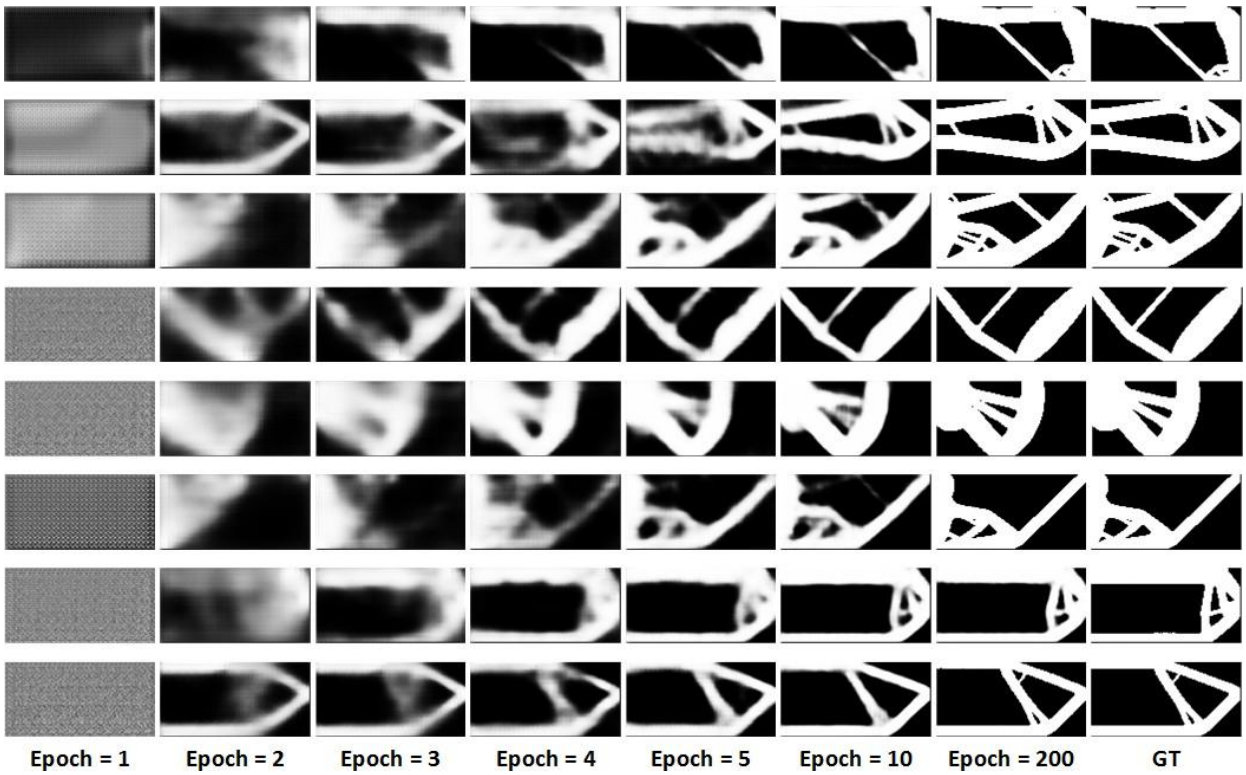


Figure 12: Evolution of TopologyGAN predictions during training.
To visually compare the prediction accuracy of the fully trained TopologyGAN on the training, validation, and test sets, Figure 13 shows the computed results and the corresponding ground truth structures. By comparing each set of images, we find that TopologyGAN performs well on the training and validation sets. The performance of TopologyGAN on the test set is expectedly lower.

To further quantify the performance, we compute the VF and the compliance of the resulting structures and compare them to those of the ground truth structures. Based on the fully trained TopologyGAN, we randomly select 640 samples from each of the training and test sets. For each data sample, this results in two structures: a prediction $\hat{y}$ and a ground truth y. The relative error of the volume fraction RE^{VF} and the relative error of compliance RE^{C} are computed respectively using Eq.(20) and Eq.(22).
The sorted RE^{VF} values for the training and test sets are shown in the first diagram of Figure 14. Histograms of RE^{VF} in the training and the test are also shown in Figure 14. RE^{VF} is close to zero for the majority of the samples. Additionally, the histogram of RE^{VF} on the training set is more concentrated around zero than that on the test set as expected. RE^{C} shows as similar trend as shown in 15.

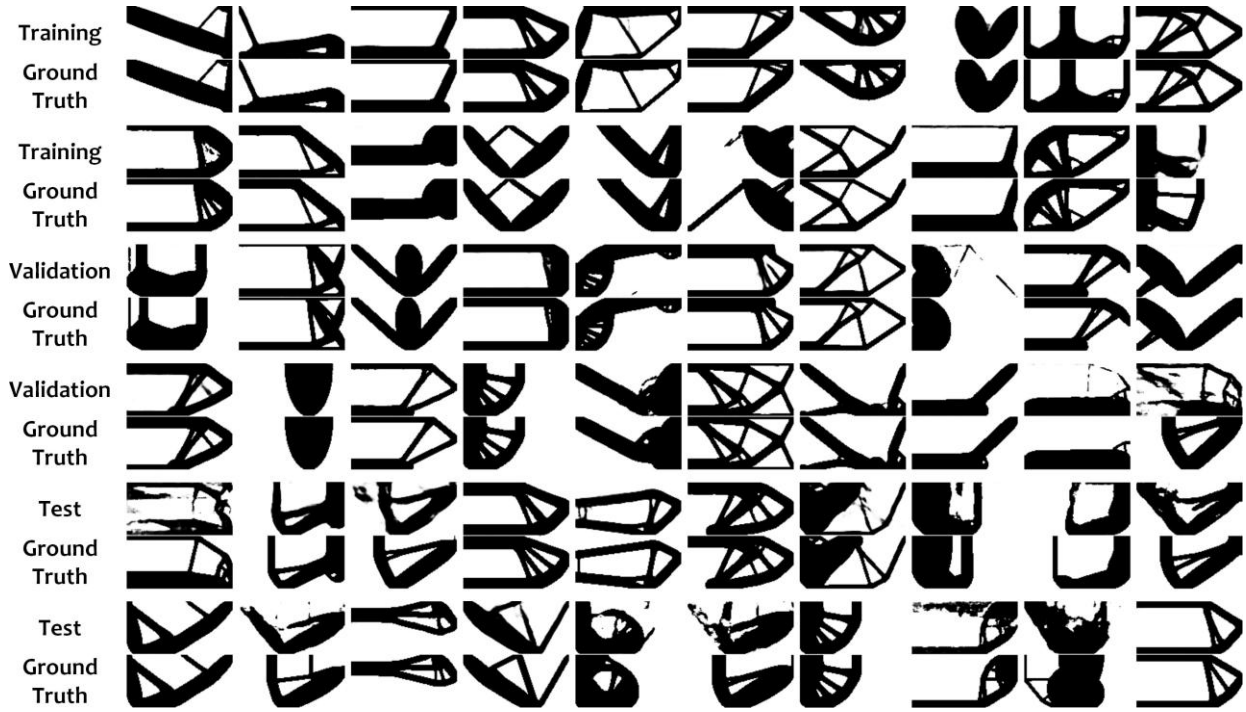


Figure 13: Comparison of the predictions of the fully trained TopoloyGAN on training, validation and test sets.

Comparison and Selection of Physical Fields

In addition to the combination of von Mises stress σ_{vm} and strain energy density W, various physical field combinations are studied as the inputs. The comparison is shown in Table 2. The results indicate that №8 - [VF + U + σ_{vm} + W] has the best performance on the training and validation sets, but №4 - [VF + σ_{vm} + W] shows a better prediction performance on the test set.

Generator Architecture

The most important feature of the U-Net is the skip connections between mirrored downsampling and upsampling layers to transfer local information and to merge features at the various resolution levels. The most distinct feature of SE-ResNet is the shortcut connection which performs identity mapping added to the output of the stacked layers to dynamically select the layer depth for the desired underlying mapping. Served as the generator of TopologyGAN, the proposed architecture U-SE-ResNet in this article combines U-Net and SE-ResNet, as shown in Figure 16. We train the TopologyGAN model with different generator architectures, which are U-Net, SE-ResNet, and U-SE-ResNet. The training results are shown in Table 3. U-SE-ResNet has the best performance of the three. This is due to the fact that the U-SE-ResNet combines the advantages of the U-Net and the SE-ResNet, which improves the model flexibility in the local information transfer and the adjustable network depth.

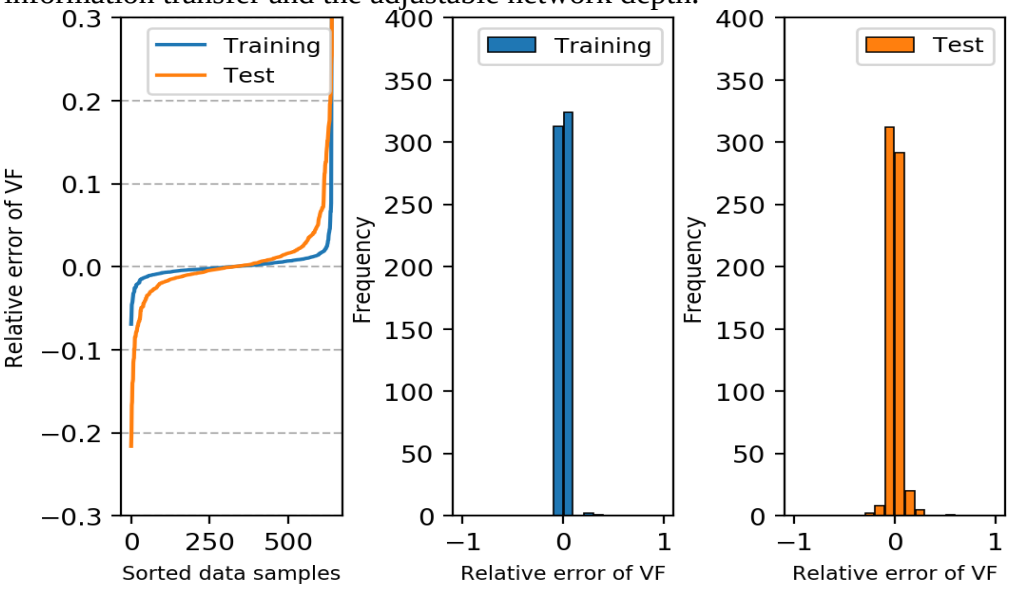


Figure 14: Comparison of volume fraction on training and test sets Table 2: Comparison and selection of physical fields as GAN inputs.

Metrics		MAE			MSE		
№	Physical Fields	Training	Validation	Test	Training	Validation	Test
0	Baseline	0.088257	0.100565	0.181095	0.085916	0.097966	0.175226
1	VF+*U*	0.001941	0.039551	0.105560	0.002391	0.030464	0.099863
2	VF+*W*	0.001781	0.039145	0.100626	0.002343	0.033687	0.094903
3	VF+σ_{vm}	0.001758	0.040411	0.098619	0.002333	0.035467	0.081702
4	VF+σ_{vm}+*W*	0.001808	0.019133	0.070128	0.001340	0.018022	0.059943
5	VF+σ	0.002339	0.037105	0.101626	0.002382	0.031802	0.095526
6	VF+ε	0.001729	0.034620	0.093073	0.002306	0.029235	0.087086
7	VF+σ+ε	0.010823	0.037518	0.122126	0.001976	0.032496	0.092162
8	VF+*U*+σ_{vm}+*W*	0.000942	0.019519	0.088748	0.001099	0.031628	0.079184
9	VF+LP	0.001914	0.033079	0.139781	0.00328	0.037652	0.100326

Note: VF is volume fraction, *F* is external loads, *U* is displacement field, σ_{vm} is von Mises stress, *W* is strain energy density, σ is stress field, ε is strain field, and LP is load path vector. Number marked in yellow is the minimum in each column.

Limitations and Future Work

TopologyGAN exhibits good performance compared to the baseline cGAN and generalizes to previously unseen boundary conditions. However, there are several limitations. First, there is no in-built mechanism that ensures a single connected component (or the avoidance of the checkerboard pattern) outside of the penalization enhanced SIMP-based ground truth training data the network observes. Second, while VF is typically employed as an upper bound for the amount of material that SIMP can utilize, TopologyGAN treats VF as a condition to match as closely as

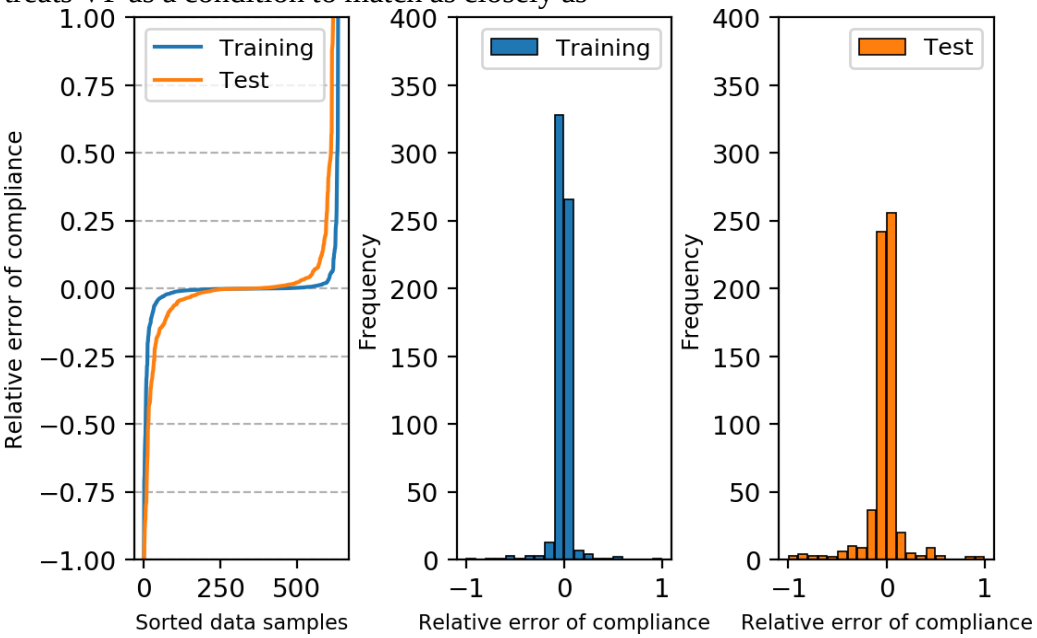


Figure 15: Comparison of compliance on training and test sets.

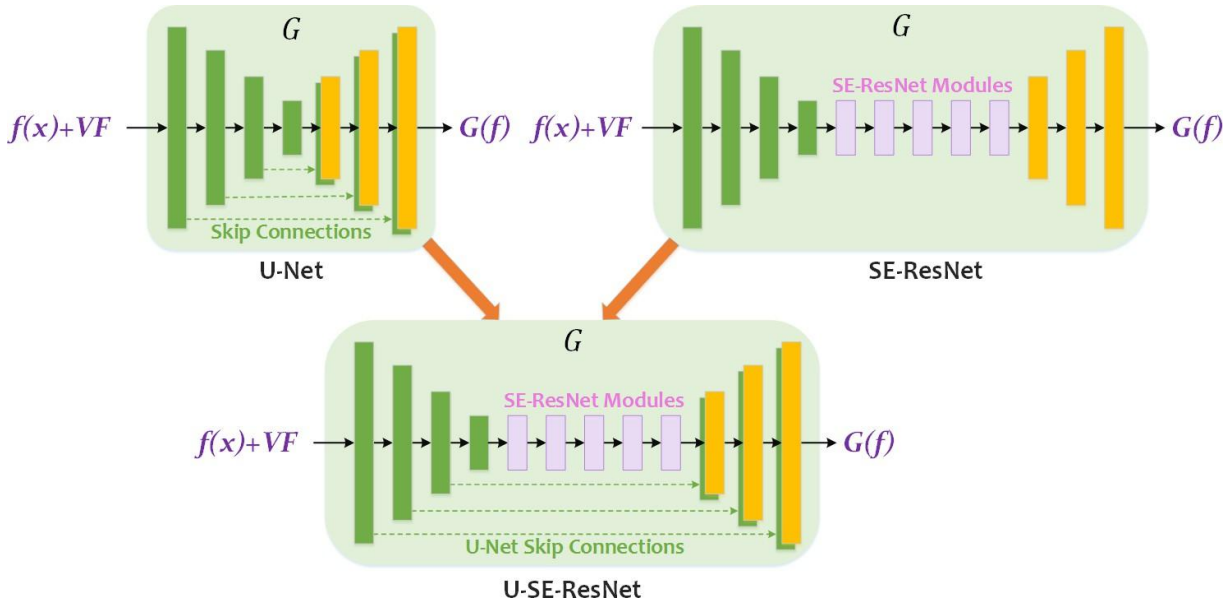


Figure 16: Architectures of U-Net, SE-ResNet, and U-SE-ResNet

possible (minimizing the absolute error of the target and generated VF) rather than a true inequality constraint. Third, TopologyGAN is implemented for 2D topology optimization and, while the approach will be similar in nature, will require modifications to extend to 3D.

Table 3: Training results of the three generator architectures

Architecture	MAE			MSE		
	Training	Validation	Test	Training	Validation	Test
U-Net	0.002908	0.027434	0.101455	0.002471	0.029133	0.098439
SE-ResNet	0.008597	0.100675	0.142915	0.058231	0.089755	0.157362
U-SE-ResNet	0.001808	0.019133	0.070128	0.001340	0.018022	0.059943

TopologyGAN only takes advantage of the fields computed on the original, unoptimized domain. However, theoretically, there is no restriction on which fields can be used to augment the network. In particular, a new network could be devised where the structure generated by TopologyGAN is reassessed to compute the fields of interest and this information can be fed back to the network for improved predictions in several iterative (yet fast) steps. These limitations and observations will be the subject of our future work.

Conclusions

We introduce TopologyGAN, a novel deep learning-based generative model for topology optimization that takes as inputs the goal volume percentage, displacement, and load boundary conditions. TopologyGAN supplements the traditional inputs with dense starting fields calculated across the initial, unoptimized domain. The network is able to significantly outperform a baseline cGAN thanks to these added fields. Furthermore, we suggest U-SE-ResNet, a novel hybrid generator design that blends the characteristics of SE-ResNet and U-Net. TopologyGAN is able to considerably lower the test errors on problems with previously unknown boundary conditions by using the starting fields and the U-SE-ResNet model.

References

[1] Noboru Kikuchi and Martin Philip Bendsøe. use a homogenization approach to create ideal topologies in structural design. Computer techniques in applied engineering and mechanics, 71(2):197–224, 1988.
[2] Noboru Kikuchi and Katsuyuki Suzuki. A homogenization technique for topology and form optimization. Computer techniques in applied engineering and mechanics, 93(3):291-318, 1991.
[3] Bendsøe, Martin P. Topology optimization. Springer, 2009.
[4] MPe Bendsøe and AR Diaz. Shape optimization of structures utilizing a homogenization approach for various loading scenarios. Optimization of structures, 4(1):17–22, 1992.
[5] Bendsøe, Martin P. Optimal form design as a material distribution issue. Optimization of structures, 1(4):193–202, 1989.
[6] Ole Sigmund and Martin P. Bendsøe. Topology optimization techniques for material interpolation. Applied Mechanics Archive, 69(9–10):635–654, 1999.
[7] Sigmund Ole. Black and white filters for topology optimization based on morphology. 33(4-5):401–424, Structural and Multidisciplinary Optimization, 2007.
[8] Albert C. To and Emre Biyikli. A novel non-sensitivity technique for resolving stress-constrained and minimal compliance issues, proportional topology optimization is implemented in Matlab.
[9] Zhenguo Nie, Sangjin Jung, Levent Burak Kara, and Kate Whitefoot, PloS one, 10(12):e0145041, 2015. using additive manufacturing to optimize part consolidation for the lowest possible production costs and time. Mechanical Design Journal, pages 1–16, 2019.

[10] Suresh Krishnan. A 199-line matlab code for pareto-optimal tracing in topology optimization. 42(5):665–679, Structural and Multidisciplinary Optimization, 2010.
[11] Grant P. Steven and Yi M. Xie. A straightforward evolutionary process for structure optimization. 49(5):885–896, Computers & Structures, 1993.
[12] GP Steven, D Nha Chu, YM Xie, and A Hira. covers many facets of evolutionary structural optimization for stiffness-constrained issues. 24(4):197–212, Finite Elements in Analysis and Design, 1997.
[13] YM Xie, GP Steven, and OM Querin. A bidirectional approach for evolutionary structural optimization (eso).5(8):1031–1048, Engineering Computations, 1998.
[14] 3D and multiple load case bi-directional evolutionary structural optimization (beso) Virginia Young, Osvaldo M. Querin, GP Steven, and YM Xie. Optimization of structures, 18(2-3):183–192, 1999.
[15] WC Christie, JW Bull, P Bettess, D Reynolds, and J McConnachie. A novel evolutionary method for structure optimization is reverse adaptivity. International Journal of Engineering Numerical Methods, 45(5):529–552, 1999.
[16] PJ Clarkson, GT Parks, and JS Liu. A novel approach to topology optimization for continuous structures is metamorphic evolution. 20(4):288–300, Structural and Multidisciplinary Optimization, 2000.
[17] Axel Schumacher, Vladimir V. Kobelev, and Hans A. Eschenauer. Bubble technique for optimizing the geometry and topology of buildings. Optimization of structure, 8(1):42–51, 1994.
[18] A Schumacher and HA Eschenauer. techniques for optimizing form and topology utilizing hole location criteria. Topology optimization in structural mechanics, Springer, 1997, pp. 135–196.
[19] Anca-Maria Toader, François Jouve, and Grégoire Allaire. A level-set approach to form optimization. 334(12):1125–1130, Comptes Rendus Mathematique, 2002.
[20] Anca-Maria Toader, François Jouve, and Grégoire Allaire. structural optimization using a level-set approach and sensitivity analysis. Computational Physics Journal, 194(1), 363–393, 2004.
[21] Grégoire Allaire, A-M Toader, François Jouve, and Frédéric De Gournay. Topological and form sensitivity structural optimization utilizing a level set approach. Cybernetics and control, 34(1):59, 2005.
[22] Dongming Guo, Xiaoming Wang, and Michael Yu Wang. An optimization technique for structural topology using level sets.Computer techniques in engineering and applied mechanics, 192(1-2):227-246, 2003.
[23] James K. Guest and Vivien J. Challis. Fluids in Stokes flow are optimized using level set topology. International Journal of Engineering Numerical Methods, 79(10):1284–1308, 2009.
[24] Meisam Takalloozadeh and Krishnan Suresh. A topological level-set method for stress-constrained topology optimization. 48(2):295–309, Structural and Multidisciplinary Optimization, 2013.
[25] Levent Burak Kara, James Mccann, and Erva Ulu. Design of lightweight structures in the face of unpredictable force locations.36(4):1–13, ACM Transactions on Graphics (TOG), 2017.
[26] M. Elsley and DW Kelly. A method for figuring out elastic continuum load routes. 12(5):415–424, Engineering Computations, 1995.
[27] DW Kelly, M Asudullah, and P Hsu. In finite element analysis, load routes and load flow. 18(1/2):304–313, Engineering Computations, 2001.
[28] D Kelly, M Lee, G Pearce, C Reidsema, and A Bassandeh. on using finite element analysis to determine a load-bearing topology and analyze load pathways. Analysis and design of finite elements, 47(8):867-876, 2011.
[29] Kurt Maute and Ole Sigmund. methods for topology optimization. 48(6):1031–1055, Structural and Multidisciplinary Optimization, 2013.
[30] James T. Allison, Ruijin Cang, Max Yi Ren, Danny J. Lohan, and Tinghao Guo. A style transfer and variational autoencoder-based indirect design representation for topology optimization. Page 0804 of the 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference.

[31] In Gwun Jang, Jaeho Jung, Taeil Hur, and Yonggyun Yu. Using deep learning, a near-optimal topological design may be found without any iterations. 59(3):787–799, Structural and Multidisciplinary Optimization, 2019.
[32] MH Herman Shen and Sharad Rawat. Adversarial networks are used in 3D structural topology optimization. SAE Technical Paper, 2019 technical report.
[33] Carolyn Conner Seepersad and Conner Sharpe. Conditional generative adversarial network topology design. Computers and Information in Engineering Conference and ASME 2019 International Design Engineering Technical Conferences. [34] Erva Ulu, Rusheng Zhang, and Levent Burak Kara, American Society of Mechanical Engineers Digital Collection, 2005. Optimal topologies under varied loading configurations are investigated and estimated using data. Computer Techniques in Biomedical Engineering and Biomechanics: Imaging & Visualization, 4(2):61–72, 2016.
[35] Yi Ren, Hope Yao, and Ruijin Cang. Theory-driven machine learning to generate near-optimal topology in a single shot. 2019 Computer-Aided Design, 109:12–21.