Haoliang Jiang
School of Computer Science,
College of Computing,
Georgia Institute of Technology,
Atlanta, GA 30332
e-mail: hjiang321@gatech.edu

Zhenguo Nie[1]
The State Key Laboratory of Tribology,
Department of Mechanical Engineering,
Tsinghua University,
Beijing 100084, China
e-mail: zhenguonie@tsinghua.edu.cn

Roselyn Yeo
School of Mechanical and
Aerospace Engineering,
Nanyang Technological University,
Singapore 639798
e-mail: RYEO005@e.ntu.edu.sg

Amir Barati Farimani
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: barati@cmu.edu

Levent Burak Kara
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: lkara@cmu.edu

StressGAN: A Generative Deep Learning Model for Two-Dimensional Stress Distribution Prediction

Using deep learning to analyze mechanical stress distributions is gaining interest with the demand for fast stress analysis. Deep learning approaches have achieved excellent outcomes when utilized to speed up stress computation and learn the physical nature without prior knowledge of underlying equations. However, most studies restrict the variation of geometry or boundary conditions, making it difficult to generalize the methods to unseen configurations. We propose a conditional generative adversarial network (cGAN) model called StressGAN for predicting 2D von Mises stress distributions in solid structures. The StressGAN model learns to generate stress distributions conditioned by geometries, loads, and boundary conditions through a two-player minimax game between two neural networks with no prior knowledge. By evaluating the generative network on two stress distribution datasets under multiple metrics, we demonstrate that our model can predict more accurate stress distributions than a baseline convolutional neural-network model, given various and complex cases of geometries, loads, and boundary conditions.

[DOI: 10.1115/1.4049805]

Keywords: StressGAN, stress, conditional generative adversarial network, deep learning

1 Introduction

Structural stress analysis is a critically important foundational tool in many disciplines, including mechanical engineering, material science, civil engineering, etc. It is used for computing the stress distribution and the possibility of structural failure when the structure is subject to the applied load and boundary conditions [1–3]. Finite element analysis (FEA) is commonly used to solve the governing partial differential equations on the discretized domain [4–8]. Traditional methods provide high fidelity solutions but require the solution of large linear systems, which can be computationally prohibitive. With the demand for fast and accurate structural analysis in generative design, topology optimization technologies, and online manufacturing monitoring, increasing the computational speed for stress analysis becomes a focus of interest.

Deep learning techniques start to be applied to solve computational engineering problems to achieve fast mechanical analysis [9]. Several approaches of accelerating mechanical stress analysis by deep learning methods have been carried out and achieved excellent outcomes in terms of computational speed and accuracy [6,10–15]. These studies utilize deep learning models to predict residual stress, shear stress, maximum von Mises stress, or distributions of the stress tensor. Once trained on large datasets, these approaches are able to generate accurate stress predictions. However, most previous work restricts the geometry or the boundary conditions that are applied, making the models difficult to be generalized to new problems.

In this work, we propose a conditional generative adversarial network (cGAN) called StressGAN for stress distribution prediction. StressGAN takes as input arbitrary geometries load and boundary conditions in the form of different input channels and predicts the von Mises stress distribution in an end-to-end fashion. A distinguishing feature of our approach is that we utilize an adversarial learning strategy, which has shown the advantages of the strategy in generating high-quality samples of different modalities [16–19].

We evaluate StressGAN on two datasets: a fine-mesh multiple structure dataset introduced by this work for the first time and a coarse-mesh cantilever-beam dataset used in our previous work [6]. The coarse mesh dataset is used to compare the performances of our baseline neural network and our proposed network. The fine-mesh dataset is used to further study the performances of StressGAN in solving more complex and practical stress distribution problems. The fine-mesh dataset contains 38,400 problem samples modeled as 128×128 meshes. Unlike the cantilever-beam dataset with identical shape, boundary and load conditions, the fine-mesh dataset includes ten patterns of load positions and eight patterns of boundary conditions. A surrogate model which can predict fine-mesh stress distributions fast and accurate will be more critical in practice. Compared to coarse-mesh modeling, fine-mesh modeling helps FEA converge better and generate more accurate stress distributions. Meanwhile, a larger mesh size requires more computational power and time.

To investigate the performance of StressGAN under different scenarios, we design two types of experiments: training and evaluating the network on the entire dataset and training and evaluating the network on datasets with conditions of different categories (generalization experiments). As a result, StressGAN outperforms a selected baseline model, StressNet (SRN), proposed in Ref. [6], on the datasets. Furthermore, StressGAN has a more robust performance than the baseline model for most test cases in the

[1] Corresponding author.

Contributed by the Applied Mechanics Division of ASME for publication in the Journal of Applied Mechanics. Manuscript received October 12, 2020; final manuscript received January 10, 2021; published online February 11, 2021. Assoc. Editor: Caglar Oskay.

generalization experiments with a sparse training dataset. We have shared our full implementation with the dataset and trained models.[2]

2 Background and Related Work

We focus our review on FEA, then on studies of deep learning methods with emphasis on their applications in stress estimation and generative adversarial networks with emphasis on their applications in computational engineering.

2.1 Finite Element Analysis for Stress Computation. Typical linear FEA for stress calculations involves

$$KQ = F \tag{1}$$

where K denotes a global stiffness matrix, F denotes a vector describing the applied load at each node, and Q denotes the displacement. K is assembled from elemental stiffness matrices k_e:

$$k_e = A_e B^T DB \tag{2}$$

where B is the strain/displacement matrix, D is the stress/strain matrix, and A_e is the area of the element. B and D are determined by material properties and mesh geometry. Then the local stiffness matrix k_e will be assembled into the global stiffness matrix. The displacement boundary conditions are encoded into the global stiffness matrix K by operating on the corresponding rows and columns. Various direct factorization based or iterative solvers exist for the solution of Q.

After computing the global displacement using Eq. (1), the nodal displacement q of each element followed by the stress tensor of each element is

$$\sigma = DBq \tag{3}$$

where σ denotes the stress tensor of an element. Then, the von Mises stress of each element is computed using the 2D von Mises stress form:

$$\sigma_{vm} = \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x\sigma_y + 3\tau_{xy}^2} \tag{4}$$

where σ_{vm} is von Mises stress and σ_x, σ_y, τ_{xy} are the normal and shear stress components.

2.2 Deep Learning on Mechanical Stress Analysis. Most of the early attempts to use deep learning in speeding up mechanical stress analysis focus on integrating the models in a FEA software. These models are to solve some auxiliary tasks including updating FEA model [20,21], checking plausibility of a finite element simulation [22], modeling the constitutive relation of a material [23], and optimizing the numerical quadrature in the element stiffness matrix on a per element basis [24]. Liu et al. [25] combine a neural network and a mechanical system which provides real-time data for the neural network to lean nonlinear constitutive models. These works alleviate the complexity of FEA softwares to some extent. However, our approach can be used as a surrogate model to a FEA software. It avoids the computation bottlenecks in a FEA software and its computation cost could be controlled by modifying the architecture.

Deep learning methods are proposed as surrogate models to approximate residual stress in girth welded pipes [13], shear stress in circular channels [14], or stress in 3D trusses [11]. These methods use manually assigned features to represent a fixed geometry or a part of the geometry. The deep learning models will estimate a stress value based on the input parameters. In our work, the deep learning method learns to filter useful features and generates a representation for each combination of the geometry, external load, and boundary condition. A decoder follows the data representation and predicts a stress distribution on the geometry.

Liang et al. [10] have developed an image-to-image deep learning framework as an alternative to predicting aortic wall stress distributions by expanding aortic walls into a topologically equivalent rectangle. Feng and Prabhakar [15] propose a difference-based neural network for predicting the stress distributions in heterogeneous media. The network learns a mapping between the difference of input geometry and reference geometry and the difference of stress distribution and the reference stress distribution. Khadilkar et al. [12] propose a two-stream deep learning framework to predict stress fields in each of the 3D printing process. The network encodes 2D shapes of each layer and the point clouds of 3D models based on a convolutional neural network (CNN) architecture and a PointNet [26]. Madani et al. [27] propose a transfer learning model to predict the value and position of the maximum von Mises stress on arterial walls in atherosclerosis. Our model also uses an image-to-image translation model to estimate the stress distribution. Yang et al. [28] utilize an artificial neural network to learn the yielding function of representative volume elements which are used to solve boundary value problems. Wang et al. [29] design a neural network based on CNN and long short term memory network (LSTM) to capture the sequence of maximum internal stress given fracture propagation and initial stress data. We utilize image-based data representation on both the geometry and the input conditions. Thus, our model can be used to analyze arbitrary 2D stress distribution cases after proper training.

Most related to our work, Nie et al. [6] propose an end-to-end convolutional neural network called StressNet to predict 2D stress distributions given multi-channel data representations of geometry, load, and boundary conditions of cantilever beams. The network contains three downsampling convolutional layers, five squeeze-and-excitation ResNet (SE-RES) blocks [30,31], and three upsampling convolutional layers. Each SE-RES block is composed of two convolutional layers and a SE block which utilizes a global pooling and two fully connected layers to learn extra weights for each channel. Skip connections are used in each block. 9 × 9 kernels are used in the first and last convolutional layer and 3 × 3 kernels are used in all remaining convolutional layers. A dataset composed of 120,960 cases of cantilever beams modeled using 32 × 32 meshes is generated by a linear FEA software to train and evaluate the network. In our work, we aim at analyzing high-resolution cases and use an adversarial learning scenario additionally to capture features in stress distributions. More importantly, all previous work of deep learning methods in stress prediction focuses on specific application cases lacking variety in geometry, external load, and boundary conditions. Moreover, through testing our model by geometries or conditions from the unseen domain, we show the potential of our deep learning model as a transfer learning model for stress field predictions.

2.3 Generative Adversarial Networks. Generative adversarial networks (GANs) are an example of generative models. They model the training of a generative network as a two-player minimax game where a generator G is trained to learn a distribution f with a discriminator D [32]. Both of them represent a differentiable function controlled by the learned parameters. In a conventional GAN, the generator G learns to map a vector sampled from a latent space $z \sim p_z(z)$ to the space of ground truth samples. In the meantime, the discriminator D learns to map a sample to a probability that predicts if the presented sample is real or fake. The Nash equilibrium in training is that the generator forms the same distribution as the training data and the discriminator output 0.5 for all input data [33].

cGAN is built upon the learning algorithm of GAN and has been widely used to date [16–19]. cGAN develops a method to control the mapping from input to output by conditioning the standard generator G and discriminator D on extra information. Figure 1 demonstrates the framework of cGAN. Furthermore, Isola et al. [16] propose a similar network for image-based task such as image-to-image translation. In the comparisons against networks plainly use mean absolute error (MAE) as a loss function, it

[2] https://github.com/zhenguonie/2020_StressGAN.

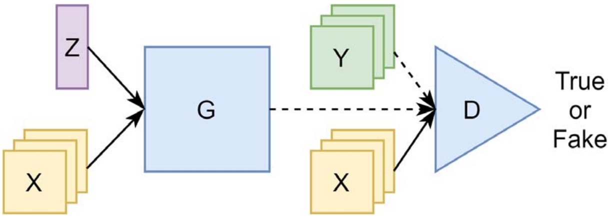

Fig. 1 Framework of cGAN. The generator *G* and discriminator *D* are conditioned on information *X*. A latent vector *Z* and *X* compose the input to *G*. *D* learns to tell whether its input regarding *X* is from real samples *Y* or output of *G*.

shows the superiority of using cGAN framework in image-based tasks. Radford et al. [34] reinforce GAN's training stability using all convolutional net [35], ReLU and LeakyReLU activations, and batch normalization layers.

Nie et al. [36] proposed a cGAN-based model called TopologyGAN for topology optimization based on intermediate representations of the configurations such as stress field and strain field. Farimani et al. [37,38] propose a cGAN architecture based on the network proposed by Isola et al. [16] to learn models of steady-state heat conduction, incompressible fluid flow, and phase segregation. Lee and You use GAN in the prediction of unsteady flow over a circular cylinder with various Reynolds numbers [39]. Paganini et al. [40] use a revised deep convolutional generative adversarial network (DCGAN) which is developed to model electromagnetic showers in a longitudinally segmented calorimeter. The deep learning method speeds up the calculations by more than 100 times. Enomoto et al. utilize a DCGAN architecture for cloud removal in climate images [41]. In the field of astronomy, GANs are used to generate images of galaxies [42,43] and 2D mass distributions [44]. In our work, we use the architecture and learning algorithm introduced by Radford et al. [34] and Isola et al. [16] to build our neural network for stress field predictions cross varying input geometries and boundary conditions.

3 Method

3.1 Neural-Network Architecture. The architecture of StressGAN is shown in Fig. 2. We design the generator as an encoder–decoder network that generates a feature vector with a size of 512 in the bottleneck. The input of the generator is a case of conditions and geometry modeled by $m\times m$ meshes. Three $m\times m$ resolution images are used to represent geometry, boundary conditions, and the applied load. To increase data intensity, we represent geometry and boundary conditions on one image. We use numbers 0, 1, 2, 3, 4 in geometries to represent various boundary conditions, where 0 is void, 1 means free solid node, 2 means node affixed horizontally, 3 means node affixed vertically, and 4 means node affixed in both directions. The remaining two images record magnitudes of vertical or horizontal loads in the corresponding pixel. The output of the generator is a $m\times m$ mesh describing the von Mises stress distributions. The encoder is composed of $\log_2(m)$ downsampling blocks with a convolutional layer, a batch normalization layer, and a LeakyReLU layer. Similarly, the decoder is composed of $\log_2(m)$ upsampling blocks with a deconvolutional layer, a batch normalization layer, and a ReLU layer. When the network is trained and tested using the coarse-mesh dataset, we remove four blocks close to the bottleneck to keep the bottleneck representation of input conditions as a 512 feature vector. We remove the ReLU layer in the last upsampling block with the consideration that mechanics analysis results other than von Mises Stress might contain negative values. The convolutional layers and deconvolutional layers both have kernel sizes as 5×5 and stride size as 2.

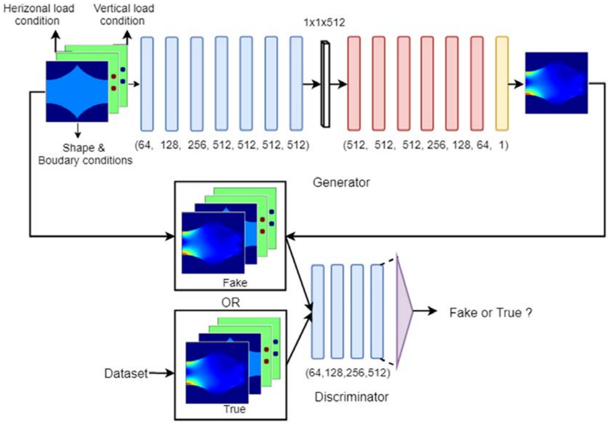

Fig. 2 Architecture of StressGAN. The generator (top) and the discriminator (bottom) are constructed with downsampling blocks and upsampling blocks. For the last upsampling block of the generator, we remove the ReLU activation. The numbers indicate channel dimensions of the output of each blocks. The purple triangle means a reshape layer followed by a linear layer and a Sigmoid activation.

For the discriminator, we adopt a downsampling structure. The input is a stress distribution case described by four $m\times m$ images including the fake or ground truth sample stress distribution and its conditions. The architecture of the discriminator is fixed when experimented on different datasets. It outputs a probability value which describes whether the input analysis result is true regarding the conditions and geometry. Four downsampling blocks are followed by a reshape layer, a fully connected layer, and a Sigmoid activation.

3.2 Loss Function and Metrics. *Loss function*: Our loss function consists of an L2 distance loss and a cGAN objective function:

$$L_{\mathrm{L2}}(G) = E_{x,y}[\|y - G(x)\|_2] \tag{5}$$

where y is ground truth stress distributions, x stands for conditions and geometries, and G denotes the generator. Previous work has shown that utilizing L2 distance which can be fomulated as mean squred error (MSE) to train networks for predicting stress distributions works well. Thus, we use L2 distance as a loss in StressGAN's loss function.

The loss function of cGAN used in our model can be expressed as

$$\min_G \max_D V(G, D) = E_{x,y}[\log(D(x, y))] + E_x[\log(1 - D(x, \mathrm{G}(x)))] \tag{6}$$

cGAN loss function shows the adversarial relationship between the generator G and the discriminator D. Note that in our cases where the network should output a particular analysis result given the conditions and a geometry, we eliminate the Gaussian noise vector z which is usually an input of the generator to add more variation to the output.

The final loss is

$$\min_G \max_D V(G, D) + \lambda L_{\mathrm{L2}}(G) \tag{7}$$

where a hyperparameter λ is to balance the loss function following Refs. [16,45].

Metrics: In addition to MSE, four metrics are introduced to assess the performance of StressGAN: MAE, mean absolute percentage error (MAPE), peak stress absolute error (PAE), and peak stress absolute percentage error (PAPE). These four metrics, whether related to MSE or not, are not an explicit goal of minimizing MSE. Using these four metrics, we can provide an assessment of stress prediction qualities.

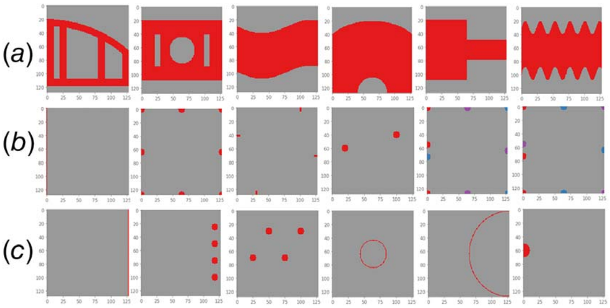

Fig. 3 Data samples in fine-mesh dataset. (*a*) Geometries. (*b*) Boundary conditions. (*c*) Load positions.

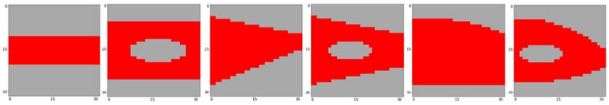

Fig. 4 Categories of geometry in coarse-mesh dataset. From left to right: rectangular beam, rectangular beam with a cellular opening, trapezoidal beam, trapezoidal beam with a cellular opening, beam with parabola contours, and beam with parabola contours and a cellular opening.

Using MAE and a normalized version of MAE, MAPE helps evaluate the overall quality of a predicted stress distribution. MAE is defined as

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}_j| \tag{8}$$

where y_j is the value at element j in a ground truth sample, $\hat{y}_j$ is the estimated value at element j, and n denotes the number of elements of samples. MAPE is defined as

$$\text{MAPE} = \frac{\text{MAE}}{\max\{Y\} - \min\{Y\}} \times 100\% \tag{9}$$

where Y denotes a set of all ground truth stress values in a case, $\max\{Y\}$ is the maximum value in a set of ground truth stress values Y, and $\min\{Y\}$ is the minimum value in a set of ground truth stress values Y.

PAE and PAPE measure the accuracy of the most considerable stress value in a predicted stress distribution which is the most critical local value of stress distributions in engineering applications. PAE is defined as

$$\text{PAE} \ = |\max\{Y\} - \max\{\hat{Y}\}| \tag{10}$$

PAPE is defined as

$$\text{PAPE} \ = \frac{\text{PAE}}{\max\{Y\}} \times 100\% \tag{11}$$

where Y denotes a set of all ground truth stress values in a case, $\max\{Y\}$ is the maximum value in a set of ground truth stress values Y, $\hat{Y}$ is the set of all predicted stress values in a case, and $\max\{\hat{Y}\}$ is the maximum value in a set of all predicted stress values $\hat{Y}$.

4 Experiments

4.1 Dataset and Implementation. *Fine-mesh multiple structure dataset*: To provide a broad evaluation of our network's performance, we introduce a stress distribution dataset composed of multiple structures, each modeled as a 128×128 elements. The dataset is generated using a 2D FEM software SOLIDSPY [46]. All elements in the domain are a four-node quadrilateral with a size of 1×1 (mm). Void regions are modeled using a Young's modulus of infinitesimal value. The dataset contains 60 geometries, ten patterns of load conditions, and eight patterns of boundary conditions, in total, 38,400 instances. The shapes, load conditions, and boundary conditions are not limited to cantilever beams which are affixed on one end and bearing loads on the other end. Samples of geometry, load, and boundary conditions are demonstrated in Fig. 3. Also, for each load pattern, the orientations of the loads can vary from 0 deg to 315 deg with a step of 45 deg. We normalize the load magnitudes in the dataset to reduce the input space since the linear characteristic of homogeneous and isotropic elastic material results in a linear relationship between the loads and the stresses.

Table 1 Quantitative evaluation of StressGAN and StressNet with fine-mesh dataset

Metric	MSE	MAE	MAPE	PAE	PAPE
StressGAN	**77.31**	**1.83**	**0.21%**	**20.29**	**1.47%**
StressNet	1120	10.88	1.20%	132.6	19.80%

Note: The best result under each metric is shown in bold (units: mm–MPa–N).

Coarse-mesh cantilever-beam dataset: The course-mesh stress distribution dataset is proposed by our previous work [6]. The dataset consists of six categories of geometries, in total, 80 geometries. Examples of categories of geometry are shown in Fig. 4. Load is applied on the right end of the beam. The left end of the beam is fixed. For each geometry, load orientation changes from 0 deg to 355 deg, in 5-deg increments. For each orientation, the load magnitudes vary from 0 N to 100 N by a step of 5 N. In total, the dataset includes 120,960 instances with various shapes, load orientations, and load magnitudes.

Implementation detail: We train StressGAN using a learning rate of 0.001 by the Adam optimizer [47] with a batch size of 64. We use 10 or 100 as the value of λ in StressGAN's loss in the experiments with the fine-mesh dataset and coarse-mesh dataset, respectively. The learning rate, batch size, and λ are decided by the performances of the network on a small portion of the training dataset which serves as an evaluation dataset in a grid search on potential values. In each training epoch, we train the discriminator once and the generator twice to keep the training stable. In all experiments, we use an NVIDIA GeForce GTX 1080Ti GPU. Under our experiment setting, each case in fine-mesh dataset take approximately 0.003 s to analyze.

4.2 Experimental Design. *Entire dataset training and evaluation*: In this experiment, we randomly divide both the fine-mesh dataset and the coarse-mesh dataset into train/test sets of 80–20% split, respectively. We then train and evaluate StressGAN on the datasets to demonstrate its effectiveness. Additionally, we train and evaluate our baseline model StressNet under the same scenario to compare their performances. In this experiment setting, we would like to investigate the general predictions capability of StressGAN.

Generalization training and evaluation: To further study StressGAN's performances in general engineering scenarios such as unseen geometries or unseen applied loads, we set three sub-experiments where training and test sets belong to different categories of geometry or load orientation. The whole experiment is set based on coarse-mesh dataset since it is easier to separate geometries into semantic categories. In the first and second sub-experiments, we train and evaluate the networks using samples from different geometry categories. In the first sub-experiment, we train the networks with samples in the categories of rectangular beams, trapezoidal beams, rectangular beams with cellular openings, and trapezoidal beams with cellular openings, and we evaluate the networks with beams with a parabola contour. In the second

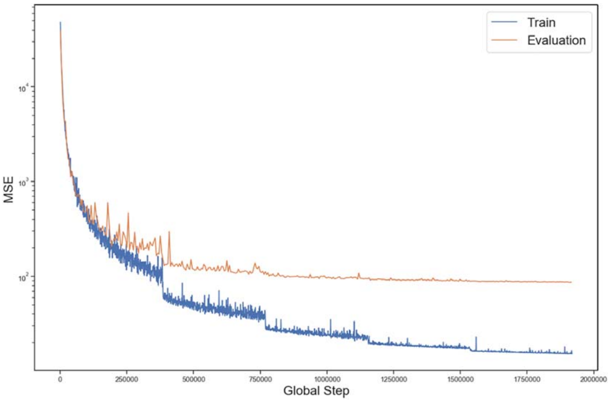

Fig. 5 The training curve of StressGAN trained on fine-mesh dataset

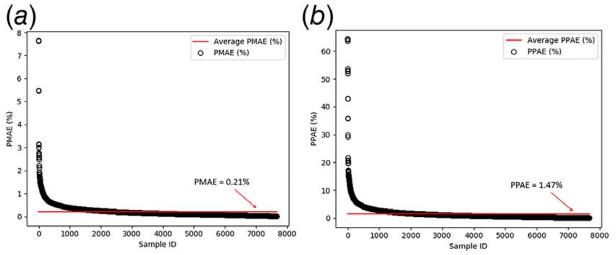

Fig. 6 Statistical accuracy of StressGAN on fine-mesh dataset. (*a*) MAPE of each sample and average MAPE on the test dataset. (*b*) PAPE of each sample and average PAPE on the test dataset.

experiment, we train the networks using beams without holes and evaluate the networks using beams with cellular openings. The third sub-experiment is to study how the network performs when trained and evaluated by cases with different load orientations. The load orientations are split up by quadrants. We randomly select loads in three quadrants for training and use loads in the remaining quadrant for testing. We normalize the load magnitudes in all training and test datasets, which reduces the size of all training datasets to less than 5000 samples.

5 Results and Discussions

5.1 Entire Dataset Evaluation. We train and test our model using the fine-mesh dataset with a split of 80–20%. Meanwhile, we train and test StressNet on the same training and testing dataset. The evaluation results of the three networks are shown in Table 1. The best performance under each metric is shown in bold. StressGAN attains a MAPE of 0.21% and a PAPE of 1.47%, which indicates that StressGAN can produce accurate fine-mesh stress distribution given complex cases. The training process of StressGAN is plotted in Fig. 5. The statistical accuracy of StressGAN on the test dataset is shown in Fig. 6. The most inaccurate predictions are shown in Fig. 7. Even with the highest MAPE, these predictions still provide useful information. Table 1 shows that StressGAN outperforms StressNet with a significant margin in all metrics. Figure 8 shows comparisons of the evaluation results of StressGAN and StressNet. As shown in the visualizations, the predicted stress distributions of StressGAN are sharper than the predictions of StressNet, especially around the edges of the void versus material boundaries. Additionally, StressGAN's predictions of the critical areas are comparatively more accurate.

We visualize the activation layers of a random sample and test the discrimination of the discriminator. Figure 9 shows the output activation layers of the convolutional layers in the generator. From the figures in the upper row, it can be clearly observed that the positions of boundary conditions and external forces are highlighted by the filters, which demonstrates the ability of the network to capture and transfer input conditions. The figures in the bottom row provide insights into how the network computes the stress distribution based on the encoded information. Ground truths and predictions of test dataset are fed into the discriminator. The average output of the discriminator given the ground truths and predictions is 0.899 and 0.002, respectively. This shows that the discriminator has learned the implicit features of stress distributions. Even with test data, it is able to distinguish the ground truth distribution from the predicted ones.

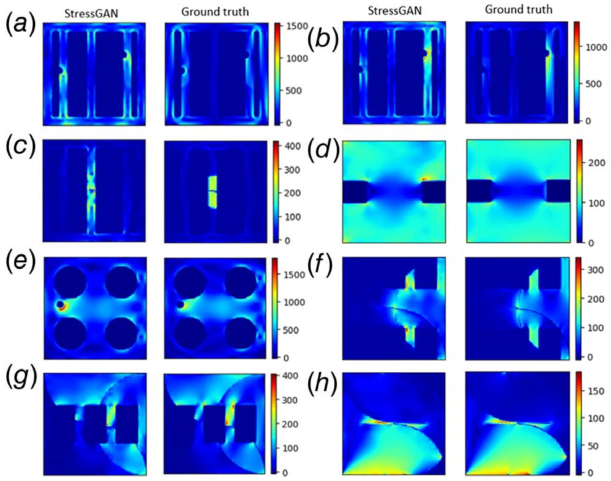

Fig. 7 The predictions of StressGAN on fine-mesh dataset with largest MAPE values. The MAPE value of each case: (*a*) MAPE = 7.65%, (*b*) MAPE = 5.48%, (c) MAPE = 3.15%, (*d*) MAPE = 2.99%, (*e*) MAPE = 2.76%, (*f*) MAPE = 2.72%, (*g*) MAPE = 2.66%, and (*h*) MAPE = 2.56%.

We also train and test our model using the coarse-mesh dataset with a split of 80–20% of training and test dataset. The evaluation results of StressGAN and StressNet are shown in Table 2. With four layers removed, StressGAN still outperforms StressNet under all

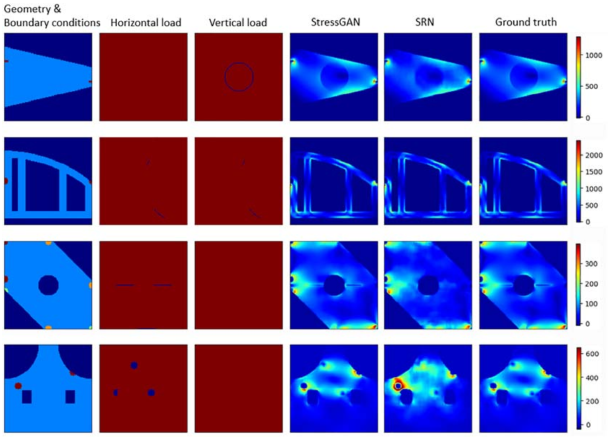

Fig. 8 Evaluation of StressGAN and StressNet on fine-mesh dataset. Four evaluation cases are shown by each row. From left to right: (1) geometry and boundary conditions; (2) horizontal load positions; (3) vertical load positions; (4) predictions of StressGAN; (5) predictions of StressNet; and (6) ground truths. The load magnitudes of each case: (1) $q(x) = 0.0$ N/mm^2, $q(y) = -88.4$ N/mm^2; (2) $q(x) = 125.0$ N/mm^2, $q(y) = 125.0$ N/mm^2; (3) $q(x) = 100.0$ N/mm^2, $q(y) = 0.0$ N/mm^2; (4) $q(x) = -68.8$ N/mm^2, $q(y) = 0.0$ N/mm^2 (units: mm–MPa–N).

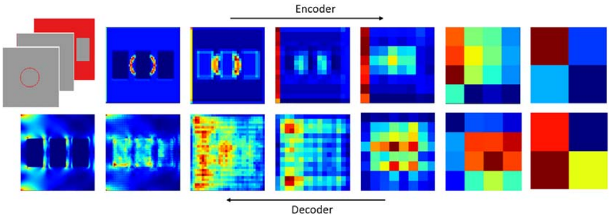

Fig. 9 Output activation layers of the generator. The output activation layers of convolutional layers are shown to display the encoding and decoding processes.

Table 2 Quantitative evaluation of StressGAN and StressNet with coarse-mesh dataset

Metric	MSE	MAE	MAPE	PAE	PAPE
StressGAN	**0.13**	**0.19**	**0.14%**	**0.48**	**0.33%**
StressNet	0.15	0.20	0.15%	0.50	0.37%

Note: The best performance under each metric is shown in bold (units: mm–MPa–N).

metrics. In the qualitative evaluation, StressGAN generates high-quality stress distributions, as shown in Fig. 10.

5.2 Generalization Evaluation. We conduct generalization experiments to explore our method's performance in situations where the training dataset is sparse and testing data contain unseen cases. We include StressNet and StressGAN into this experiment to compare their performances and demonstrate the characteristics of each network. The parametric results of the three experiments are shown in Tables 3–5. The best performance under each metric is shown in bold. The selected samples of prediction results are shown in Figs. 11–13, respectively. In general, StressGAN gives a better performance concerning the average prediction accuracy. The best MAPEs in three experiments are 8.23%, 6.80%, and 1.49%, respectively, which are all obtained by StressGAN.

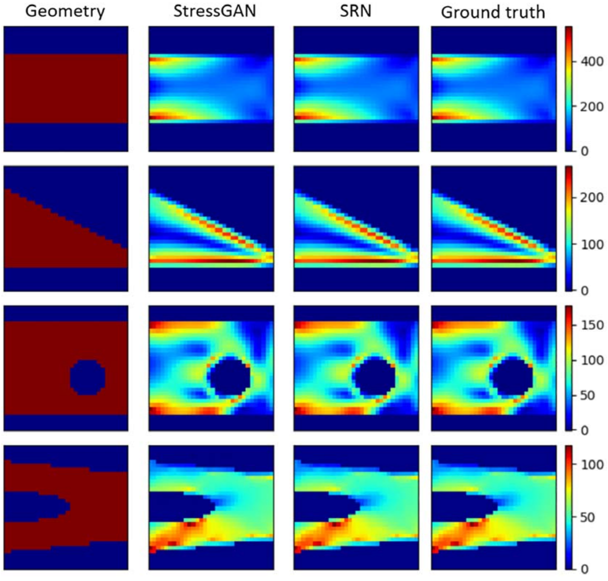

Fig. 10 Evaluation of StressGAN and StressNet on cantilever-beam dataset. Four evaluation cases are shown by each row. The visualizations of results of StressGAN and StressNet are identical to the ground truth stress distributions. From left to right: (1) geometry, (2) predictions of StressGAN, (3) predictions of StressNet, and (4) ground truthstress distributions. The load magnitudes of each case: (1) $q(x) = 27.5$ N/mm^2, $q(y) = -47.6$ N/mm^2; (2) $q(x) = -43.0$ N/mm^2, $q(y) = 61.4$ N/mm^2; (3) $q(x) = -3.5$ N/mm^2, $q(y) = 19.7$ N/mm^2; (4) $q(x) = -54.8$ N/mm^2, $q(y) = -4.8$ N/mm^2 (units: mm–MPa–N).

Table 3 Quantitative evaluation of StressGAN and StressNet with training data of rectangular beams and trapezoidal beams and testing data of beams with a parabolar contour

Metric	MSE	MAE	MAPE	PAE	PAPE
StressGAN	**28.91**	**2.80**	**7.50%**	**6.85**	**18.10%**
StressNet	43.14	3.28	9.30%	12.55	38.39%

Note: The best performance under each metric is shown in bold (units: mm–MPa–N).

Table 4 Quantitative evaluation of StressGAN and StressNet in the second sub-experiment with training data of beams without openings and testing data of beams with cellular openings

Metric	MSE	MAE	MAPE	PAE	PAPE
StressGAN	**77.20**	**4.40**	**6.80%**	16.96	24.10%
StressNet	95.36	4.59	7.54%	**14.09**	**23.62%**

Note: The best performance under each metric is shown in bold (units: mm–MPa–N).

Table 5 Quantitative evaluation of StressGAN and StressNet in the experiment of cross-load direction training and evaluation

Metric	MSE	MAE	MAPE	PAE	PAPE
StressGAN	**3.71**	**0.84**	**1.49%**	**3.15**	**4.86%**
StressNet	6.86	1.29	2.58%	6.72	11.66%

Note: The best performance under each metric is shown in bold (units: mm–MPa–N).

In the two cross-geometry experiments, we can study the characteristics of StressGAN and StressNet including their advantages and disadvantages. Figure 11 shows the visualizations of the ground truth stress distributions and prediction stress distributions in the first cross-geometry experiment. Although the contour information of the input geometries is hard for StressGAN to capture, StressGAN outputs stress distributions closer to the samples in the dataset, especially in the regions of high stresses. Additionally, it generates a sharper (less blurred) prediction. Figure 12 shows similar trends for the second cross-geometry experiment. On the one hand, StressGAN failed to predict stresses around the openings correctly. On the other hand, StressGAN generates more reasonable stress distributions which are more similar to the ground truth samples. Additionally, StressNet could recognize the openings and predict zero stresses in void areas in some test cases. Since cellular openings have a considerable influence on stress concentrations and the networks have no explicit training on this phenomenon, large errors occur when we evaluate the predicted largest stress values as shown in Table 4.

The results of the cross-orientation experiment are shown in Fig. 13. The output stress distributions from StressGAN are quite similar to the ground truths. From Table 5, it can be seen that among the three generalization experiments, the cross-orientation experiment attains the best evaluation results. Since we use two images to express the load positions and magnitudes along with the horizontal and vertical directions, respectively, the deep learning method has a potential to learn the influence of the horizontal and vertical loads from the training dataset separately (essentially,

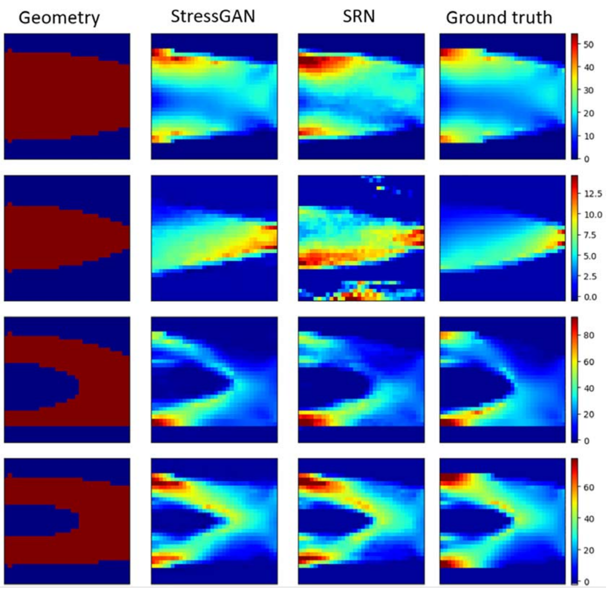

Fig. 11 Evaluation of StressGAN and StressNet on cases of different contours. Four evaluation cases are shown by each row. From left to right: (1) geometry, (2) predictions of StressGAN, (3) predictions of StressNet, and (4) ground truth stress distributions. The load magnitudes of each case: (1) $q(x) = -5.7$ N/mm^2, $q(y) = -8.2$ N/mm^2; (2) $q(x) = 10.0$ N/mm^2, $q(y) = -0.9$ N/mm^2; (3) $q(x) = -7.7$ N/mm^2, $q(y) = 6.4$ N/mm^2; (4) $q(x) = 5.0$ N/mm^2, $q(y) = 8.7$ N/mm^2 (units: mm–MPa–N).

the principle of superposition by exploiting the linear nature of FEA) and synthesize reasonable results when tested on unseen load orientations. This is especially useful in compressing the size of the training dataset for data efficiency without significantly increasing the error rate.

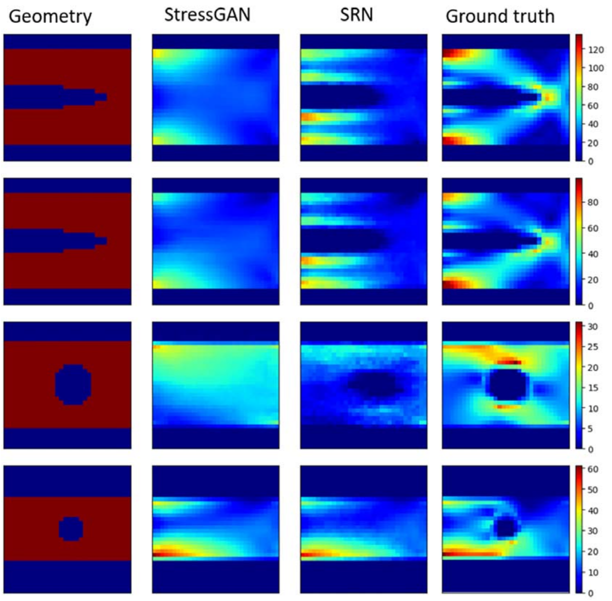

Fig. 12 Evaluation of StressGAN and StressNet on cases of cantilever beams with cellular openings. Models are trained with cantilever beams with openings and tested with cantilever beams without openings. Four evaluation cases are shown by each row. From left to right: (1) geometry, (2) predictions of StressGAN, (3) predictions of StressNet, and (4) ground truth stress distributions. The load magnitudes of each case: (1) $q(x) = 1.7$ N/mm^2, $q(y) = 9.8$ N/mm^2; (2) $q(x) = -7.7$ N/mm^2, $q(y) = 6.4$ N/mm^2; (3) $q(x) = 10.0$ N/mm^2, $q(y) = 0.9$ N/mm^2; (4) $q(x) = -9.1$ N/mm^2, $q(y) = 4.2$ N/mm^2 (units: mm–MPa–N).

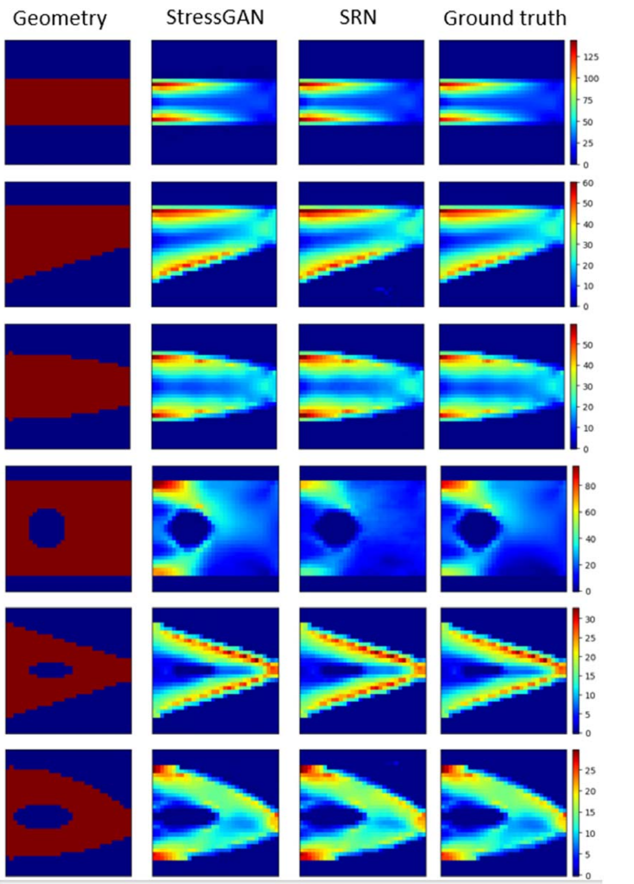

Fig. 13 Evaluation of StressGAN and StressNet on cases of different load orientations. This figure shows six evaluation cases of StressGAN and StressNet when trained and tested with load conditions in different quadrants. From left to right: (1) geometry, (2) predictions of StressGAN, (3) predictions of StressNet, and (4) ground truth stress distributions. The load magnitudes of each case: (1) $q(x) = -4.2$ N/mm^2, $q(y) = -9.1$ N/mm^2; (2) $q(x) = -4.2$ N/mm^2, $q(y) = -9.1$ N/mm^2; (3) $q(x) = -3.4$ N/mm^2, $q(y) = -9.4$ N/mm^2; (4) $q(x) = -7.7$ N/mm^2, $q(y) = -6.4$ N/mm^2; (5) $q(x) = -3.4$ N/mm^2, $q(y) = -9.4$ N/mm^2; (6) $q(x) = -4.2$ N/mm^2, $q(y) = -9.1$ N/mm^2 (units: mm–MPa–N).

6 Conclusion

In this work, we develop a cGAN called StressGAN for von Mises stress distribution prediction. StressGAN learns to predict the stress distribution given the geometries, load, and boundary conditions through a two-player minimax game between its generator and discriminator. A fine-mesh stress distribution dataset composed of 38,400 cases of various geometries, load, and boundary conditions is proposed for evaluating the network's performance on complex stress prediction cases. In addition, a coarse-mesh dataset, which is proposed in our previous work, is used to fairly compare StressGAN with the baseline model.

StressGAN achieves high accuracy in both experiments under multiple metrics, in evaluations of the two datasets. StressGAN outperforms the baseline model in predicting the stress distributions under different scenarios. It achieves an average error rate less than 0.21% on all stress values and 1.47% on the maximum stress value when evaluated on complicated configurations and an average error rate of less than 0.14% on all stress values and 0.33% on the maximum stress value when evaluated on cantilever beams.

Moreover, StressGAN's performance under general scenarios is studied. StressGAN generates stress distributions more similar to samples in the dataset which shows that it is a more effective learner in capturing the underlying knowledge of ground truth stress distributions. Furthermore, StressGAN is more efficient when facing unseen conditions. Although some cases that lead to stress concentration such as holes in geometries result in inaccurate predictions from StressGAN, the computed stress distributions still embody useful information such as the location of the highly stressed regions. The stress distributions are more similar to ground truths compared to the baseline method regardless of the conditions. In contrast, our baseline model StressNet is better at correctly estimating zero stresses in void areas but produces overall less accurate stress distributions under the same problem inputs. Furthermore, both StressGAN and StressNet perform well given unseen load orientations compared to the cases where unseen geometries are involved.

In this work, the potential of generalizing the stress prediction ability to different categories is shown in generalization experiments. These findings constitute a step toward generating data-driven analysis approaches that can generalize well to previously unseen problem configurations.

7 Limitations and Future Work

The generalization experiments show that solving new configurations is a difficult problem for StressGAN. A potential reason is that our dataset is not sufficient, as the configurations and stress distributions are in high dimensional space, which requires a large number of samples to represent the whole distribution. To address the bottleneck, one solution is to sample more configuration-stress fields pairs and improve the capacity of our neural network. However, sampling of high dimensional space can be costly. A complement is to tap into learning methods without full supervision such as semi-supervised learning, weakly supervised learning, and unsupervised learning. How to use these methods in training a surrogate model for mechanics problems is a key point. Our future work will also involve building surrogate models for more practical mechanical problems such as nonlinear problems and 3D model analysis. The current model focuses on stress distribution problems of homogeneous and isotropic elastic materials. Stress analysis of complex cases is still a challenge to existing physics-based models in terms of computational cost and accuracy while for an end-to-end neural-network-based model, the time complexity of approximation remains constant. Nevertheless, the complexity of the problem makes it necessary to come up with efficient data sampling methods in simulation and wise utilization of real experimental data.

Conflict of Interest

There are no conflict of interest.

Data Availability Statement

The authors attest that all data for this study are included in the paper. No data, models, or code were generated or used for this paper.

References

[1] Yang, R., and Chen, C., 1996, "Stress-Based Topology Optimization," Struct. Optim., **12**(2–3), pp. 98–105.

[2] Wang, D., Lee, J., Holland, K., Bibby, T., Beaudoin, S., and Cale, T., 1997, "Von Mises Stress in Chemical–Mechanical Polishing Processes," J. Electrochem. Soc., **144**(3), pp. 1121–1127.

[3] Chen, J., Akyuz, U., Xu, L., and Pidaparti, R., 1998, "Stress Analysis of the Human Temporomandibular Joint," Med. Eng. Phys., **20**(8), pp. 565–572.

[4] Segerlind, L. J., 1976, *Applied Finite Element Analysis*, Vol. 316, Wiley, New York.

[5] Cook, R. D., 2007, *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, New York.

[6] Nie, Z., Jiang, H., and Kara, L. B., 2019, "Stress Field Prediction in Cantilevered Structures Using Convolutional Neural Networks," ASME J. Comput. Inf. Sci. Eng., **20**(1), p. 011002.

[7] Mises, R. v., 1913, "Mechanik der festen körper im plastisch- deformablen zustand," Nachr. von der Gesellschaft der Wissenschaften zu Göttingen, Math. Phys. Klasse, **1913**, pp. 582–592.

[8] Sinclair, G. B., Beisheim, J. R., and Kardak, A. A., 2018, "On the Detection of Stress Singularities in Finite Element Analysis," ASME J. Appl. Mech., **86**(2), p. 021005.

[9] LeCun, Y., Bengio, Y., and Hinton, G., 2015, "Deep Learning," Nature, **521**(7553), p. 436.

[10] Liang, L., Liu, M., Martin, C., and Sun, W., 2018, "A Deep Learning Approach to Estimate Stress Distribution: A Fast and Accurate Surrogate of Finite-Element Analysis," J. R. Soc. Interface, **15**(138), p. 20170844.

[11] Nourbakhsh, M., Irizarry, J., and Haymaker, J., 2018, "Generalizable Surrogate Model Features to Approximate Stress in 3d Trusses," Eng. Appl. Artif. Intell., **71**(1), pp. 15–27.

[12] Khadilkar, A., Wang, J., and Rai, R., 2019, "Deep Learning-Based Stress Prediction for Bottom-Up sla 3d Printing Process," Int. J. Adv. Manuf. Technol., **102**(5), pp. 1–15.

[13] Mathew, J., Moat, R., Paddea, S., Fitzpatrick, M. E., and Bouchard, P., 2017, "Prediction of Residual Stresses in Girth Welded Pipes Using an Artificial Neural Network Approach," Int. J. Pressure Vessels Piping, **150**(1), pp. 89–95.

[14] Khozani, Z. S., Bonakdari, H., and Zaji, A. H., 2017, "Estimating the Shear Stress Distribution in Circular Channels Based on the Randomized Neural Network Technique," Appl. Soft Comput., **58**(1), pp. 441–448.

[15] Feng, H., and Prabhakar, P., 2020, "Difference-Based Deep Learning Framework for Stress Predictions in Heterogeneous Media," preprint arXiv:2007.04898.

[16] Isola, P., Zhu, J., Zhou, T., and Efros, A. A., 2016, "Image-to-Image Translation With Conditional Adversarial Networks," CoRR, abs/1611.07004.

[17] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W., 2017, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 21–26.

[18] Liu, M.-Y., Breuel, T., and Kautz, J., "Unsupervised Image-to-Image Translation Networks," the 31st International Conference on Neural Information Processing Systems (NIPS'17), Red Hook, NY, Dec. 4–9.

[19] Chrysos, G. G., Kossaifi, J., and Zafeiriou, S., 2020, "RoCGAN: Robust Conditional GAN," International Journal of Computer Vision, **128**(10), pp. 1573–1405.

[20] Levin, R., and Lieven, N., 1998, "Dynamic Finite Element Model Updating Using Neural Networks," J. Sound Vib., **210**(5), pp. 593–607.

[21] Atalla, M., and Inman, D., 1998, "On Model Updating Using Neural Networks," Mech. Syst. Signal Process., **12**(1), pp. 135–161.

[22] Spruegel, T., Schröppel, T. and Wartzack, S., 2017, "Generic Approach to Plausibility Checks for Structural Mechanics With Deep Learning," DS 87-1 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 1: Resource Sensitive Design, Design Research Applications and Case Studies, Vancouver, Canada, Aug. 21–25, 2017, pp. 299–308.

[23] Javadi, A. A., Tan, T. P., and Zhang, M., 2003, "Neural Network for Constitutive Modelling in FInite Element Analysis," Comput. Assist. Mech. Eng. Sci., **10**(4), pp. 523–529.

[24] Oishi, A., and Yagawa, G., 2017, "Computational Mechanics Enhanced by Deep Learning," Comput. Methods Appl. Mech. Eng., **327**(1), pp. 327–351.

[25] Liu, X., Tao, F., Du, H., Yu, W., and Xu, K., 2020, "Learning Nonlinear Constitutive Laws Using Neural Network Models Based on Indirectly Measurable Data," ASME J. Appl. Mech., **87**(8), p. 081003.

[26] Charles, R. Q., Su, H., Kaichun, M., and Guibas, L. J., 2017, "Pointnet Deep Learning on Point Sets for 3d Classification and Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 21–26.

[27] Madani, A., Bakhaty, A., Kim, J., Mubarak, Y., and Mofrad, M., 2019, "Bridging Finite Element and Machine Learning Modeling: Stress Prediction of Arterial Walls in Atherosclerosis," ASME J. Biomech. Eng., **141**(8), p. 084502.

[28] Yang, H., Qiu, H., Xiang, Q., Tang, S., and Guo, X., 2020, "Exploring Elastoplastic Constitutive Law of Microstructured Materials Through Artificial Neural Network: A Mechanistic-Based Data-Driven Approach," ASME J. Appl. Mech., **87**(9), p. 091005.

[29] Wang, Y., Oyen, D., Mehta, A., Scott, C. B., Panda, N., Fernández-Godino, M. G., Srinivasan, G., and Yue, X., 2020, "Stressnet: Deep Learning to Predict Stress With Fracture Propagation in Brittle Materials," preprint arXiv:2011.10227.

[30] He, K., Zhang, X., Ren, S., and Sun, J., 2016, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 27–30, pp. 770–778.

[31] Hu, J., Shen, L., and Sun, G., 2018, "Squeeze-and-Excitation Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 18–23, pp. 7132–7141.

[32] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014, "Generative Adversarial Nets," the 27th International Conference on Neural Information Processing Systems (NIPS'14), Montreal, Canada, Dec. 8–13, pp. 2672–2680.

[33] Goodfellow, I., 2016, "Nips 2016 Tutorial: Generative Adversarial Networks," preprint arXiv:1701.00160.

[34] Radford, A., Metz, L., and Chintala, S., 2015, "Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks," CoRR, abs/1511.06434.

[35] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A., 2014, "Striving for Simplicity: The All Convolutional Net," CoRR, abs/1412.6806.

[36] Nie, Z., Lin, T., Jiang, H., and Kara, L. B., 2020, “Topologygan: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain,” the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 46th Design Automation Conference (DAC)., Virtual, Online, Aug. 17–19.

[37] Farimani, A. B., Gomes, J., Sharma, R., Lee, F. L., and Pande, V. S., 2018, “Deep Learning Phase Segregation,” CoRR, abs/1803.08993.

[38] Farimani, A. B., Gomes, J., and Pande, V. S., 2017, “Deep Learning the Physics of Transport Phenomena,” CoRR, abs/1709.02432.

[39] Lee, S., and You, D., 2018, “Data-Driven Prediction of Unsteady Flow Fields Over a Circular Cylinder Using Deep Learning,” Journal of Fluid Mechanics, **879**(1), pp. 217–254.

[40] Paganini, M., de Oliveira, L., and Nachman, B., 2018, “Calogan: Simulating 3d High Energy Particle Showers in Multilayer Electromagnetic Calorimeters With Generative Adversarial Networks,” Phys. Rev. D, **97**(1), p. 014021.

[41] Enomoto, K., Sakurada, K., Wang, W., Fukui, H., Matsuoka, M., Nakamura, R., and Kawaguchi, N., 2017, “Filmy Cloud Removal on Satellite Imagery With Multispectral Conditional Generative Adversarial Nets,” CoRR, abs/1710.04835.

[42] Schawinski, K., Zhang, C., Zhang, H., Fowler, L., and Santhanam, G. K., 2017, “Generative Adversarial Networks Recover Features in Astrophysical Images of Galaxies Beyond the Deconvolution Limit,” Month. Notices R. Astron. Soc.: Lett., **467**(1), pp. L110–L114.

[43] Ravanbakhsh, S., Lanusse, F., Mandelbaum, R., Schneider, J., and Poczos, B., 2017, “Enabling Dark Energy Science With Deep Generative Models of Galaxy Images,” the Thirty-First AAAI Conference on Artificial Intelligence (AAAI’17), San Francisco, CA, Feb. 4–9.

[44] Mustafa, M., Bard, D., Bhimji, W., Lukić, Z., Al-Rfou, R., and Kratochvil, J., 2019, “CosmoGAN: Creating High-Fidelity Weak Lensing Convergence Maps Using Generative Adversarial Networks,” Computational Astrophysics and Cosmology, **6**(1), pp. 1–13.

[45] Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A. P., Tejani, A., Totz, J., Wang, Z., and Shi, W., 2016, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” CoRR, abs/1609.04802.

[46] Gómez, J., and Guarín-Zapata, N., 2018, SolidsPy: 2D-Finite Element Analysis with Python.

[47] Kingma, D. P., and Ba, J., 2014, “Adam: A Method for Stochastic Optimization,” CoRR, abs/1412.6980.