Check for updates

ASME Journal of Mechanical Design
Online journal at:
https://asmedigitalcollection.asme.org/mechanicaldesign



Topology Optimization Using Neural Networks With Conditioning Field Initialization for Improved Efficiency

Hongrui Chen
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: hongruic@andrew.cmu.edu

Aditya Joglekar
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: aajoglek@andrew.cmu.edu

Levent Burak Kara[1]
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: lkara@cmu.edu

We propose conditioning field initialization for neural network-based topology optimization. In this work, we focus on (1) improving upon existing neural network-based topology optimization and (2) demonstrating that using a prior initial field on the unoptimized domain, the efficiency of neural network-based topology optimization can be further improved. Our approach consists of a topology neural network that is trained on a case by case basis to represent the geometry for a single topology optimization problem. It takes in domain coordinates as input to represent the density at each coordinate where the topology is represented by a continuous density field. The displacement is solved through a finite element solver. We employ the strain energy field calculated on the initial design domain as an additional conditioning field input to the neural network throughout the optimization. Running the same number of iterations, our method converges to a lower compliance. To reach the same compliance, our method takes fewer iterations. The addition of the strain energy field input improves the convergence speed compared to standalone neural network-based topology optimization. [DOI: 10.1115/1.4064131]

Keywords: computational geometry, computer-aided design, data-driven design, machine learning, topology optimization

1 Introduction

There has been a recent increase in machine learning-driven topology optimization approaches, particularly using neural networks for performing topology optimization. Both data-driven and direct approaches have been explored. Data-driven approaches require large training database generation and a long training time. They perform instant optimal topology generation during inference time. Direct approaches use the neural network to represent the density field of a single to a small subset of designs for better parameterization.

The direct approaches require similar or more time compared to conventional topology optimization approaches like solid isotropic material with penalization (SIMP) [1,2]. We find that the results of the direct approaches, particularly the convergence speed, can be improved through insights derived from the mechanical aspects of the problem.

Machine learning-driven topology optimization approaches offer the advantage of accommodating additional insights in the form of pre-computed fields as part of the input. The usage of these fields has been explored in data-driven approaches such as TopologyGAN [3], which use physical fields such as von Mises stress and strain energy density for achieving better results. However, there has been no work incorporating these physical fields in the direct topology optimization setting. In this work, we further improve upon TOuNN (topology optimization using neural networks), an direct approach proposed by Chandrasekhar and Suresh [4], by adding a strain energy field in addition to the domain coordinates as a conditioning input to the neural network. We show that this improves the convergence speed and can give a better compliance. With the additional strain energy field as a conditioning input, the neural network not only learns a mapping function between the domain coordinates and the density field output but also between the strain energy field and the density field output. Ideally, if the conditioning field (CF) is the same as the converged topology, then the neural network only needs to learn a constant function which is the identity function. However, the converged topology is not known at the beginning of the optimization. Thus, the strain energy field is used as a good alternative since it can be computed through a single function call of finite element analysis (FEA) prior to the direct training of the neural network. We verify the performance increase obtained with this additional conditioning input across parametric experiments with varying boundary conditions and volume fractions.

The code for running the experiments in this paper can be found in the footnote.[2]

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received May 27, 2023; final manuscript received November 15, 2023; published online December 12, 2023. Assoc. Editor: Julián Norato.

[2]https://github.com/HongRayChen/Hybrid-TopOpt

 Copyright © 2023 by ASME

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024

2 Related Work

2.1 Conventional Topology Optimization. Bense and Kikuchi [5] introduced the homogenization approach for topology optimization. The SIMP method [1,2] considers the relative material density in each element of the finite element (FE) mesh as design variables, allowing for a simpler interpretation and optimized designs with more clearly defined features. Other common approaches to topology optimization include the level-set method [6,7] and evolutionary algorithms [8].

All these methods use an iterative process to create a complex mapping from problem characteristics (supports, loads, and objective function) to an optimized structure, where each iteration has an expensive FEA calculation involved. A more accurate and detailed solution can be obtained with greater number of elements in the FE mesh, however this increases the computational cost. Therefore, current developments within the field are strongly motivated by the desire to either limit the number of iterations needed to obtain an optimized structure or the computational cost of completing an iteration [9]. Recent advances in deep learning, particularly for image analysis tasks, have showed potential for removing the expensive FEA iterations required until the convergence of the topology in the conventional topology optimization approaches. Hence, various topology optimization approaches that utilize neural networks have been proposed. Woldseth et al. [9] provide an extensive overview on this topic.

2.2 Data-Driven Topology Optimization. We refer to data-driven topology optimization methods as those that aim to learn a neural network model from a database of topology optimization results for instant prediction of the optimal topology. Many methods rely on convolutional neural networks (CNNs) for their capabilities to learn from a large set of image data. Banga et al. [10] used a 3D encoder–decoder CNN to generate 3D topology results and show that interpolating the final output using the 3D CNN from the initial iterations obtained from the "TopOpt" [11] solver offers a 40% reduction in time over the conventional approach of using the solver alone. Yu et al. [12] use a conditional generative adversarial network (cGAN) in addition to CNN-based encoder–decoder network. However, the results indicate that sometimes there may be disconnections present in the predicted topology which may drastically affect the compliance values. Nakamura and Suzuki [13] improve on the results with their direct design network and with a larger dataset; however, disconnections are still observed in some solutions. Behzadi and Ilieş [14] used deep transfer learning with CNN. Zheng et al. [15] used U-net CNN for 3D topology synthesis. Nie at al. [3] used various physical fields computed on the original, unoptimized material domain as inputs to the generator of a cGAN and achieved three times reduction in mean square error as compared to a baseline cGAN. Mazé and Ahmed [16] show that diffusion models can outperform generative adversarial networks (GANs) for this task. They use regressor and classifier guidance to ensure that the generated structures are manufacturable and mechanical compliance has been minimized.

All these data-driven approaches aim to reduce optimal topology prediction time but face difficulties in generalization. Though over the years there have been improvements on the generalization capability, suitable training dataset generation is not trivial, especially for the 3D domain, and satisfactory and reliable results have not been achieved yet for direct use in real-world problems.

2.3 Direct Topology Optimization With Neural Network. We refer to direct topology optimization methods as those which do not use any prior data, rather train a neural network in an self-supervised manner for learning the optimal density distribution/topology. Chandrasekhar and Suresh [4] explored a direct approach where the density field is parameterized using a neural network.

Fourier projection-based neural network for length scale control [17] and application for multi-material topology optimization [18] has also been explored . Deng and To [19] propose topology optimization with deep representation learning, with a similar concept of re-parametrization, and demonstrate the effectiveness of the proposed method on minimum compliance and stress-constrained problems. Deng and To [20] also propose a neural network-based method for level-set topology optimization, where the implicit function of level-set is described by a fully connected deep neural network. Zehnder et al. [21] effectively leverage neural representations in the context of mesh-free topology optimization and use multilayer perceptrons to parameterize both density and displacement fields. It enables self-supervised learning of continuous solution spaces for topology optimization problems. Mai et al. [22] develop a similar approach for the optimum design of truss structures. Hoyer et al. [23] use CNNs for density parametrization and directly enforce the constraints in each iteration, reducing the loss function to compliance only. They observe that the CNN solutions are qualitatively different from the baselines and often involve simpler and more effective structures. Zhang et al. [24] adopt a similar strategy and show solutions for different optimization problems including stress-constrained problems and compliant mechanism design.

Generalization is not an issue with all these direct topology optimization methods. However, the computational time and cost are similar to traditional topology optimization approaches. An advantage offered is that the density representation is independent of the FE mesh and because of the analytical density field representation, sharper structural boundaries can be obtained [4]. We show that by adding an initial condition field as an extra input, we can improve the convergence speed and get better results. We illustrate the difference between SIMP, direct topology optimization with neural network, and data-driven topology optimization in Fig. 1.

3 Proposed Method

In our proposed method, the density distribution of the geometry is directly represented by the topology neural network. The strain energy field and the compliance used for backpropagation is calculated from an FE solver. The program is implemented in python and backpropagation of the loss function into each module is handled by the machine learning package TensorFlow [25].

3.1 Neural Network. The topology network $T_\rho(\mathbf{X})$ (Fig. 2) learns a density field in a different manner as compared to typical topology optimization which represents the density field as a finite element mesh. The topology neural network takes in domain coordinates x, y, as well as the strain energy value e at coordinates x, y. The strain energy value gets concatenated with the domain coordinates to form the input to the topology network, $\mathbf{X} = [x,\ y,\ e]$. The domain coordinates are normalized between -0.5 and 0.5 for the longest edge. It outputs the density value ρ at each coordinate point. The domain coordinates represent the center of each element in the design domain. During topology optimization, a batch of domain coordinates that correspond to the mesh grid and the corresponding strain energy field is fed into the topology network. The output is then sent to the FEA solver. The solver outputs the compliance which is combined with the volume fraction violation as a loss. The loss is then backpropagated to learn the weights of the topology network.

For the topology network design, we employed a simple architecture that resembles the function expression of $f(x) = \mathbf{w}\sin(\mathbf{k}x + \mathbf{b})$. Similar neural network architectures have been used to control the length scale of geometry in topology optimization [17]. The conditioned domain coordinates are multiplied with a kernel $\mathbf{K}$. The kernel $\mathbf{K}$ regulates the frequency of the sine function. We add a constant value of 1 to break the sine function's rotation symmetry around the origin. We use a Sigmoid function to guarantee the

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024

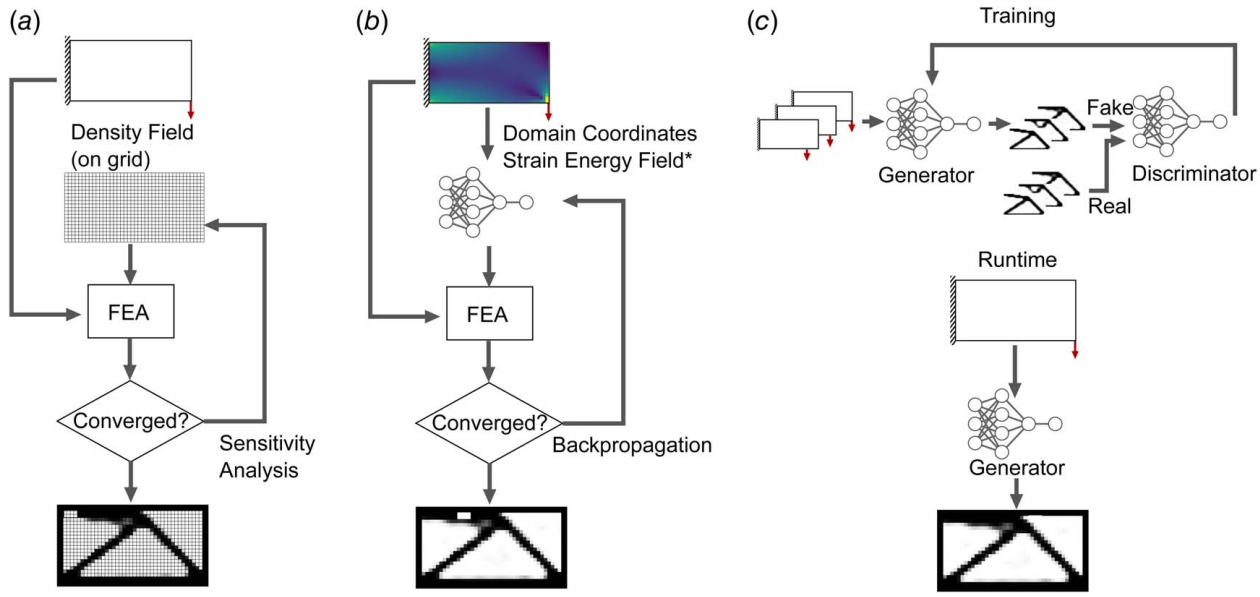


Fig. 1 This figure illustrates the difference between (*a*) SIMP-based topology optimization, (*b*) neural network-based direct topology optimization, and (c) data-driven topology optimization with GAN. Compared to previous work on direct topology optimization with neural networks, we added the strain energy field (*) in addition to the domain coordinates.

output between 0 and 1. The topology network can be formulated as follows:

$$T_\rho(\mathbf{X}) = \sigma(\mathbf{W}\sin(\mathbf{K}\mathbf{X} + 1)) \tag{1}$$

where T_ρ is the density field; $\mathbf{X}$ is the domain coordinate input, $\mathbf{X} = (x, y, e)$; σ is the sigmoid activation function; $\mathbf{K}$ is the trainable frequency kernels, initialized in $[-25, 25]$; and $\mathbf{W}$ is the trainable weights, initialized to 0.

We can upsample the 3D coordinate input or only sample specific regions of the density field to manipulate the resolution of the discretized visualization. Due to the strain energy conditioning field computed from the finite element mesh grid, interpolation needs to be used to calculate the intermediate values when upsampling the domain coordinates.

3.2 Strain Energy Conditioning Field. The strain energy conditioning field is used to augment the domain coordinate input. We calculate the conditioning field from the initial homogeneous density domain. In topology optimization, for a 2D problem with n elements of four nodes each, the strain energy field $\mathbf{E}$ can be calculated as follows:

$$\mathbf{E} = \sum (\mathbf{U}_e \times \mathbf{S}_e) \circ \mathbf{U}_e \tag{2}$$

where $\mathbf{U}_e$ is the displacement matrix, $n \times 8$; $\mathbf{S}_e$ is the element stiffness matrix, 8×8.

In most topology optimization implementations, the compliance is then calculated by summation of the above strain energy for all elements.

The strain energy field can vary greatly in range depending on the problem domain size, boundary condition, and geometry constraints. Therefore, normalization needs to be done to regulate the value range of the strain energy field. Otherwise, the range of the strain energy field will deviate from the normalized range of the domain coordinates. Furthermore, a simple normalization will not suffice as the high max value of the strain energy field reduces the amplitude of other relevant features and patterns (Fig. 3(*a*)). We explore gamma and logarithmic filtering to normalize the strain energy field. For the gamma filtering, we clip the strain

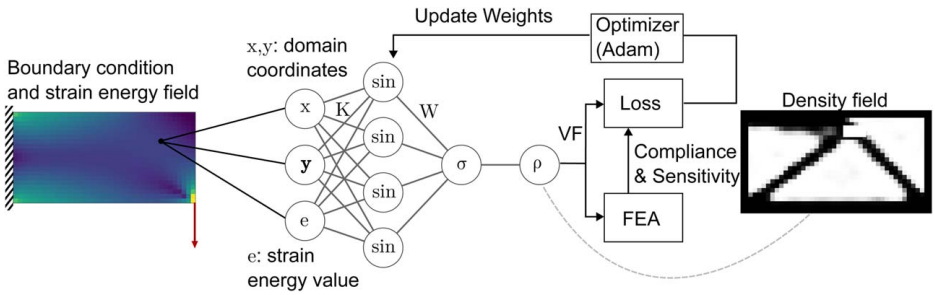


Fig. 2 The strain energy field is calculated at the beginning of the optimization based on the boundary condition. The strain energy conditioning field is fixed throughout the training. Domain coordinates and the strain energy value at each coordinate point are used as the input to the neural network. The neural network outputs density ρ at each coordinate point. By sampling coordinate point across the design domain, we obtain the density field. From the density field, we calculate the current volume fraction and the compliance from a FEA solver. The compliance and volume fraction are then formulated as a loss function which is used in back propagation of the training process until convergence.

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024

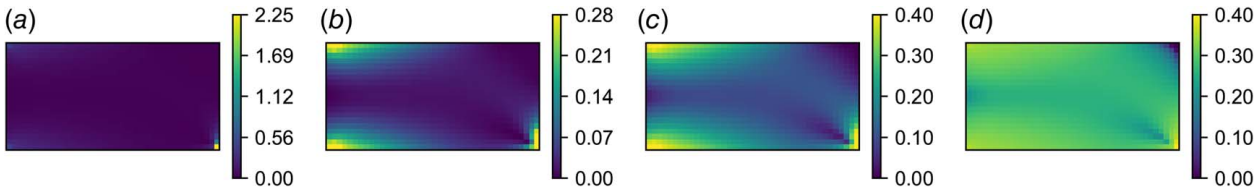


Fig. 3 (*a*) Raw strain energy field output. (*b*) Clipping value above 99 percentile. (c) Conditioning field with gamma filter. (*d*) Conditioning field with log filter. Two methods are evaluated in terms of processing of the strain energy conditioning field. We used a gamma filter in (*a*)–(*c*) and a log filter in (*d*).

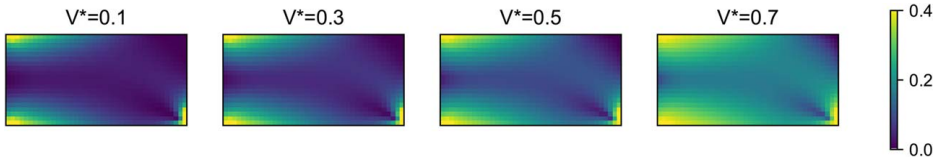


Fig. 4 For the gamma filtering of the conditioning field, we adjust the gamma based on the volume fraction target of the optimization

energy field using the 99th percentile, P_{99}. After clipping, more details of the field $\mathbf{E}_c$ can be seen (Fig. 3(*b*)). We also further adjust the feature of the strain energy field using gamma correction. The gamma value is set to be the complement of the target volume fraction V^* for the optimization ($\gamma = 1 - V^*$). The effect of the gamma correction based on the volume fraction is illustrated in Fig. 4. As the volume fraction increases, the edge feature in the strain energy field is more and more pronounced. Finally, after the gamma correction step, the strain energy field is normalized between 0 and 0.4 to obtain the processed field $\mathbf{E}_p$. The processing step on the gamma filtering of strain energy field can be summarized in the following equation:

$$\mathbf{E}_c = \min(\mathbf{E}, P_{99}) \tag{3}$$

$$\mathbf{E}_\gamma = 0.4\left\{\frac{\mathbf{E}_c - \min(\mathbf{E}_c)}{\max(\mathbf{E}_c) - \min(\mathbf{E}_c)}\right\}^\gamma \tag{4}$$

For the logarithmic filtering, we do not clip the value; instead, the log filter is directly applied to the strain energy field and then normalized between 0 and 0.4. We determine this range empirically to give the best results.

$$\mathbf{E}_{\log} = 0.4\frac{\log \mathbf{E} - \min(\log \mathbf{E})}{\max(\log \mathbf{E}) - \min(\log \mathbf{E})} \tag{5}$$

We perform a zero-normalized cross-correlation (ZNCC) test between the final converged topology of neural network-based topology optimization and the input strain energy field using logarithmic and gamma filtering with the parametric study performed in the next section. Across the 200 examples in the parametric study, the average ZNCC scores for the gamma and log filtering are 0.4585 and 0.3996, respectively. This result suggests that the strain energy field and the final converged topology are weakly correlated. However, the correlation is not strong enough to obviate the need for the boundary conditions or the neural network.

3.3 Direct Topology Optimization With Neural Network. During optimization, the topology network outputs the density value at the center for each element. These density values are then sent to the finite element solver to calculate compliance based on the SIMP interpolation.

The finite element solver is treated as a black box within the neural network. It takes in the density of each element and outputs the compliance and the sensitivity for each element with respect to the compliance. Variables that are being optimized are the weights $\mathbf{W}$ and kernels $\mathbf{K}$ of the neural network. Adam [26] is used to train the neural network. The constrained optimization problem needs to be transferred into unconstrained minimization problem for neural network. We adopt the loss function formulated by Chandrasekhar and Suresh [4] of compliance minimization and volume fraction constraint. The combined loss function is

$$L = \frac{c}{c_0} + \alpha\left(\frac{\bar{\rho}}{V^*} - 1\right)^2 \tag{6}$$

In the optimization, the target volume fraction V^* is an equality constraint and $\bar{\rho}$ is the volume fraction of the current design. When α increased to infinity, the equality constraint is satisfied. We assign a maximum value of 100 for α with initial value of 1 and gradually increase α every iteration. c is the current compliance and c_0 is the initial compliance calculated on the design domain with the uniform volume fraction V^*.

4 Results and Discussions

The possible combinations of boundary conditions, problem size, and configurations is enormous. It is impossible for us to cover all. To demonstrate the effectiveness of our proposed approach, we explore both a beam problem and a parametric study in 2D. In the beam problem, we showcase the convergence of the network's output and the convergence history. In the parametric study, problems across different boundary conditions and volume fractions are explored. We report the compliance value where subscript TONN represents FE compliance solver with neural network as topology representation, and TONNCF as neural topology optimization with strain energy CF. The neural network architecture for TONN and TONNCF is similar but slightly different than TOuNN [4,17] to accommodate the additional conditioning field input. For these two experiments, the problem size is 40×20 elements. All experiments are run on a PC with i7-12700K as processor, 32 GB of RAM, and Nvidia RTX3080 GPU.

4.1 Beam Example. Our first experiment is the beam example. The left side of the domain is fixed and a downward point load is on the center-right side. The boundary condition illustration and the strain energy conditioning field are shown in Fig. 5. The target volume fraction is 0.3. We run the direct topology optimization for a total of 1000 epochs.

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024

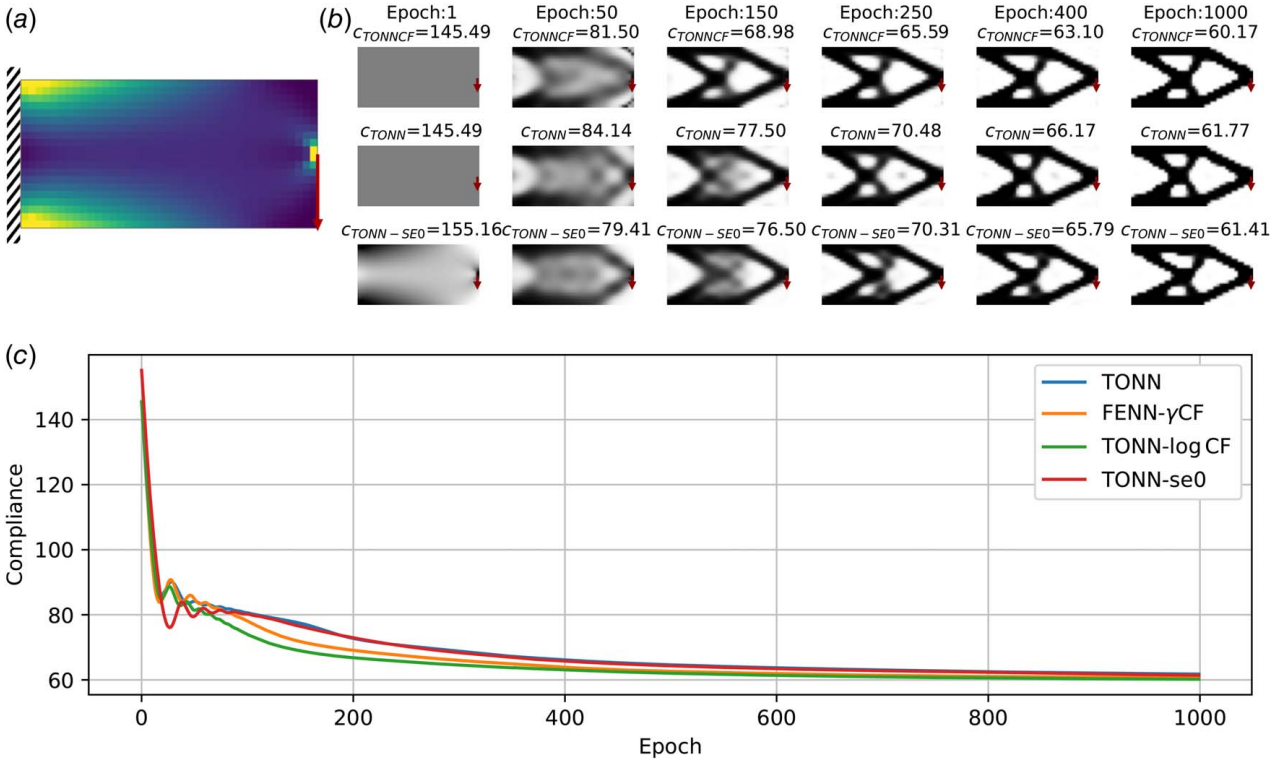


Fig. 5 Comparing the convergence history for a beam example for with and without strain energy conditioning field. The result presented is using the gamma filtering. TONN-logCF took 22.5 s while TONN took 22.1 s. (*a*) Beam boundary condition, (*b*) density field snapshot at 1, 50, 150, 250, 400, and 1000 epochs. For each column, the upper image is the density field from TONNCF, the middle image is TONN, and the bottom image is TONN with strain energy field initialization. (c) Convergence history.

The convergence history plot is illustrated in Fig. 5(*c*). We observe that by epoch 50, with the strain energy conditioning field, the network's compliance takes over the lead and maintains lower compliance all the way to the end of the training epochs. We also show the density field snapshot and the corresponding compliance during training in Fig. 5(*b*). Analyzing the geometry of the neural network with the conditioning field, we observe that there is a subtle difference compared to without the conditioning field. The neural network with conditioning field shares greater similarities to the strain energy field where the top and bottom edges are shorter. We can also observe that most of the geometry convergence happens between 0 and 400 epochs. Between 400 and 1000 epochs, the geometry remained relatively unchanged. The only change being a darker gray, showing the density values get pushed closer toward 1. In both of the examples, the final volume fraction is within 1 % error of the given target volume fraction. Therefore, we do not include the volume fraction convergence plot.

4.2 Strain Energy Field Conditioning Versus Initialization. In the aforementioned conditioning with the strain energy field, the strain energy field is used as an additional input to the neural network. Another possible method of incorporating the strain energy field is to use it as an initialization field. Initialization refers to the neural network outputs of the strain energy field at the very first iteration. This can be achieved by adding an offset before the sigmoid activation function. For consistency, we always initialize the neural network with all weights to zero. This means that by adding the logit of the strain energy field, the output of the density network after the first iteration can be made to be the strain energy field:

$$T_\rho(\mathbf{X}) = \sigma(\mathbf{W}\sin(\mathbf{K}\mathbf{X}+1) + \text{logit}(\mathbf{E})) \tag{7}$$

where $\mathbf{E}$ can be either $\mathbf{E}_{\log}$ or $\mathbf{E}_\gamma$. A small scaling of $\mathbf{E}$ is applied to prevent division by zero in the logit function.

We label the runs with strain energy field initialization as TONN-SE0 in Figs. 5 and 7. We observe that in a total of 200 test cases, there are 29 cases where initialization is able to achieve better compliance than conditioning, all of which occur at $V^* = 0.2$. Overall in 85.5% of the cases, the conditioning field outperforms the initialization. However, initialization can be useful when the target volume fraction V^* is low.

4.3 Parametric Study. We setup a parametric study to analyze the effectiveness of the gamma and log filter of the conditioning field. The boundary condition setup is illustrated in Fig. 6(*a*). The bottom right loading point is varied across the region highlighted in bottom right which accounts for 50 load conditions. We also vary the target volume fraction between 0.2 and 0.5 with an increment of 0.1. In total, this sums up to 200 total combinations. In the previous beam example, we observe that geometries do not change significantly after 400 epochs, therefore we limit the total epochs for the parametric study to 400 epochs.

The parametric study result is summarized in Fig. 7. In Fig. 7(*a*), we sort with respect to the compliance of topology optimization without conditioning field and show the compliance from both methods. We observe that the overall conditioning field converged at lower compliance. The improvement of the conditioning field is more significant when the compliance is higher. The higher compliance occurs when the volume fraction is low. To visualize the increase of convergence speed, Fig. 7 shows the percentage improvement with the conditioning field. The percentage improvement is calculated by identifying the epoch at which the conditioning field reaches a lower compliance compared to the final compliance of the optimization without the conditioning field. The average performance increase with gamma filter is 37.6% and with log filter is 44.7%. With both filters, the performance increase is more pronounced with lower volume fraction examples. The log filter has a better overall performance increase across all solutions compared to gamma filter.

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024

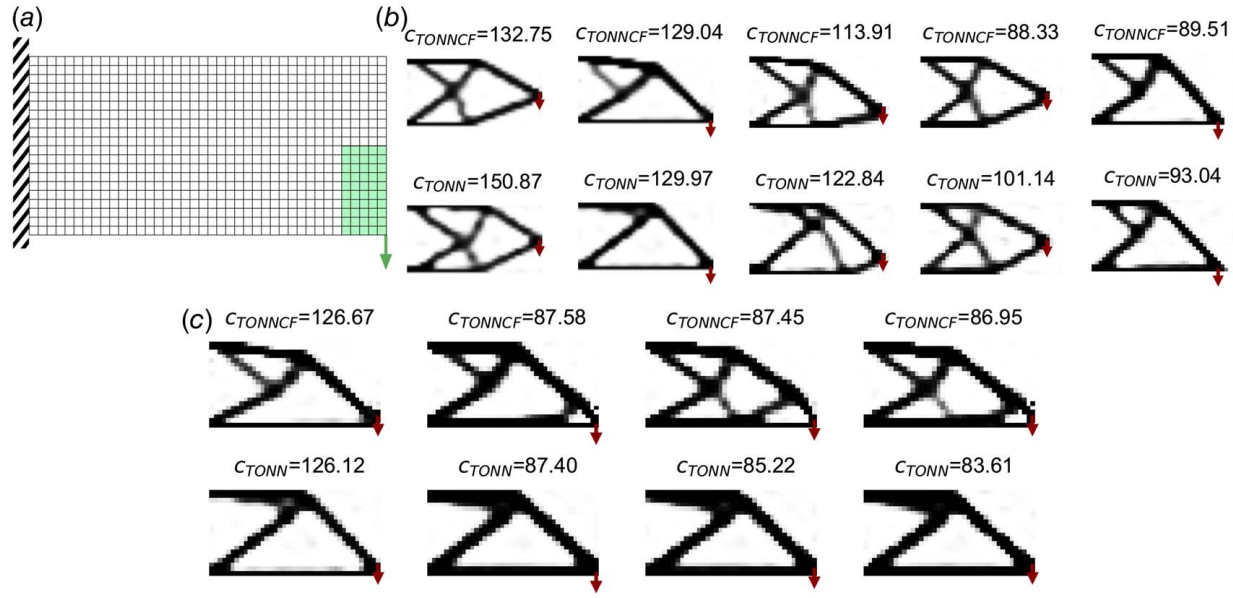


Fig. 6 Boundary conditions used in the parametric study highlighted in bottom right regions in (*a*). Some sample topology optimization results with 0.3 volume fraction within the parametric study examples are shown in (*b*). We also visualized the four cases where TONNCF did not manage to improve the convergence speed compared to TONN. (*a*) Parametric study boundary conditions, (*b*) sample topology optimization with 0.3 volume fraction, (*c*) for four cases in total of 200 examples using the gamma filter, the performance of TONNCF is worse than TONN.

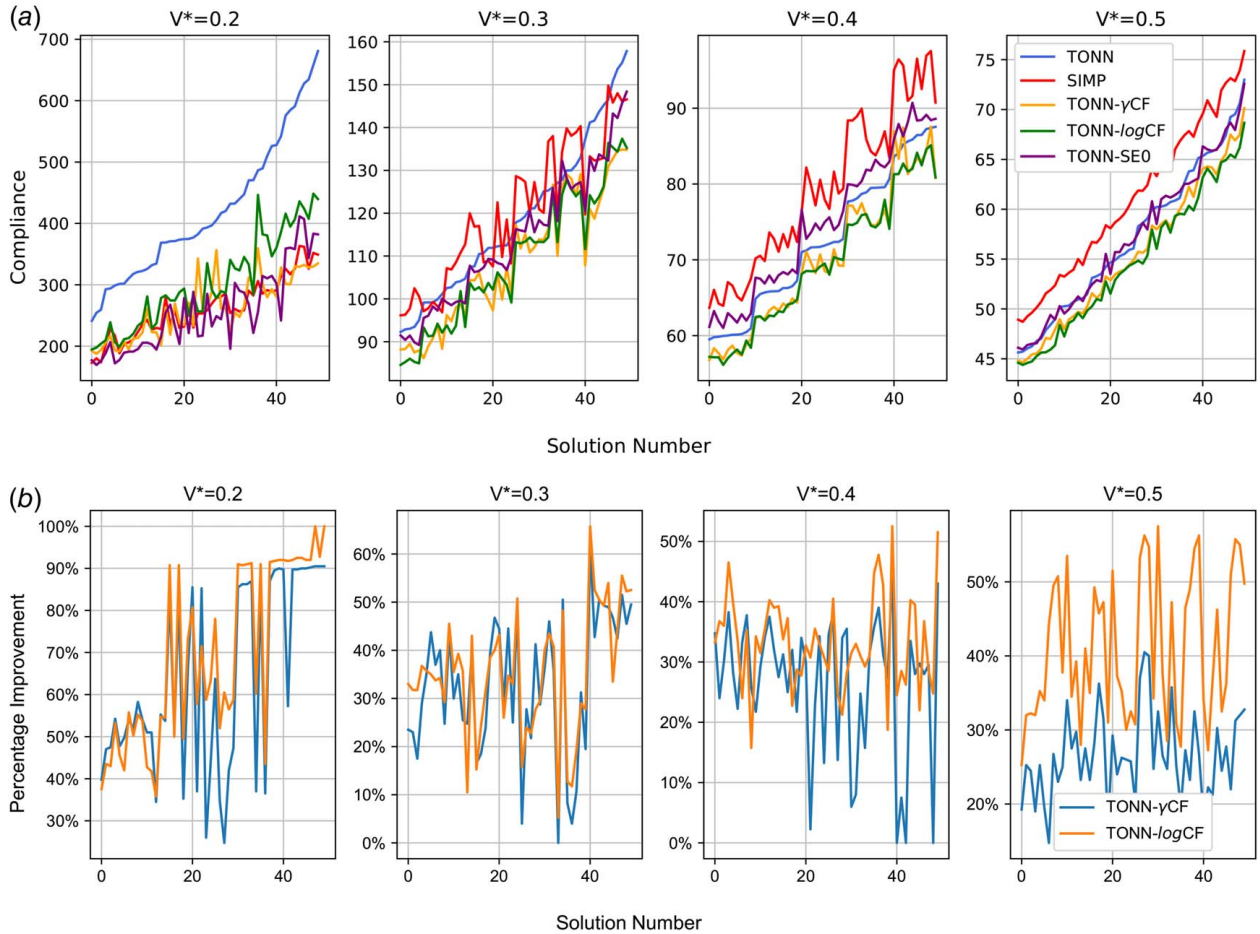


Fig. 7 Comparing the final compliance and the speed of convergence for parametric study examples for with gamma and log filter and without the conditioning field. We also run the same problem configuration with "88-lines" by Andreassen et al. [27] denoted by the legend "SIMP" in the figure: (*a*) compliance comparison of with and without strain energy field and (*b*) percentage convergence speed improvement.

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024

We compare our result against the result of "88-lines" by Andreassen et al. [27] with a filtering radius of 1.5 to accommodate the problem size. We observe that when the compliance is low, TONN performed slightly better than SIMP. For problems with relatively higher compliance, we observe that TONN with conditioning field can in some cases converge to a lower compliance than "88-lines." We note that in general, the Matlab code [27] takes around 0.2–1.5 s to run whereas TONN and TONN with either conditioning field takes around 10 s. However, a definite time comparison is difficult to establish as "88-lines" runs on Matlab whereas TONN runs on python. In "88-lines," the optimizer is optimality criteria whereas TONN rely on Adam with a learning rate of 0.002.

We also observe that within the 200 examples with gamma filter, there are four cases where the conditioning field does not improve convergence speed. When plotting out example results in Fig. 6, the examples with the load on the right bottom edge have lower performance increase with the conditioning field. On the other hand, the

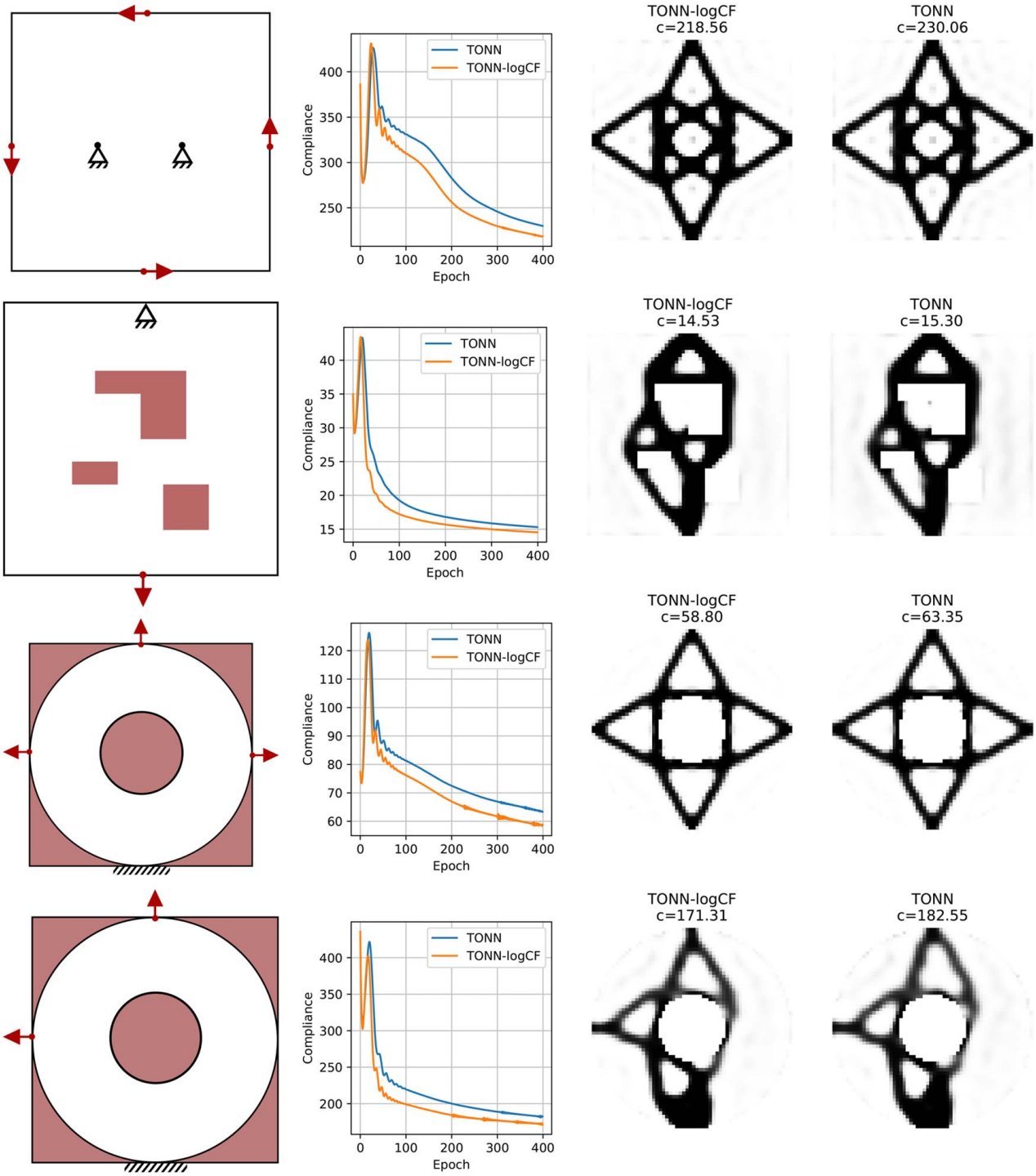


Fig. 8 Four additional test cases across varying boundary conditions and passive elements, all using 0.25 target volume fraction. Each example is 60 × 60 in resolution and takes around 30 s to run with no significant difference between with and without conditioning field. Log filtered conditioning field demonstrates good convergence speed increase.

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024

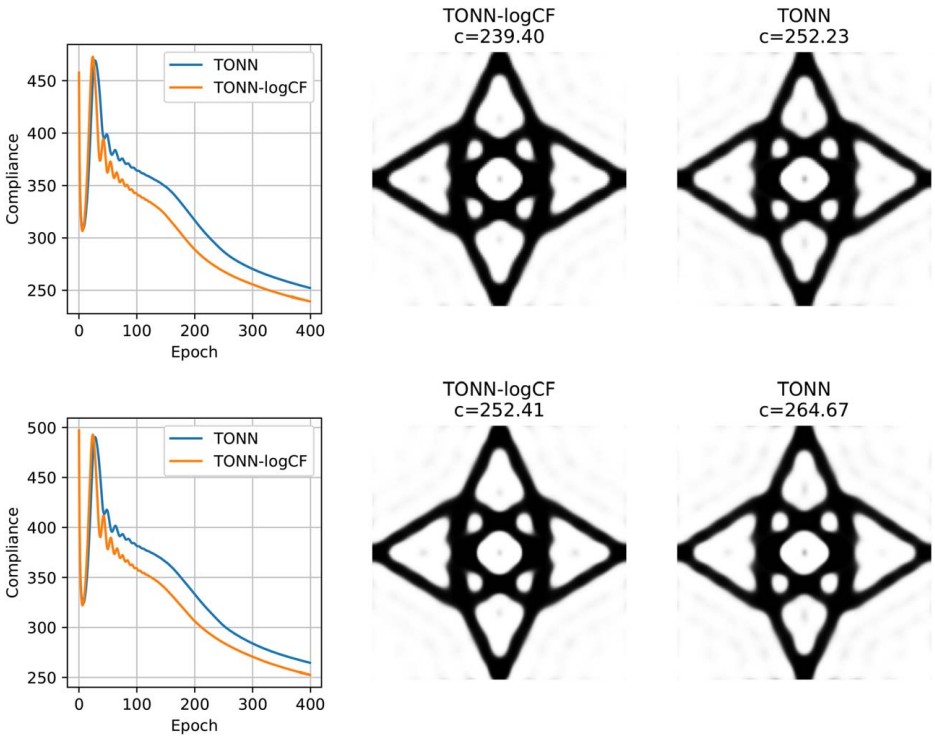


Fig. 9 We run the same boundary condition for case 1 with two and three times the resolution. The runtime for 120 × 120 is 3 min and for 180 × 180 is 10 min, case 1 with 120 × 120 resolution and case 1 with 180 × 180 resolution

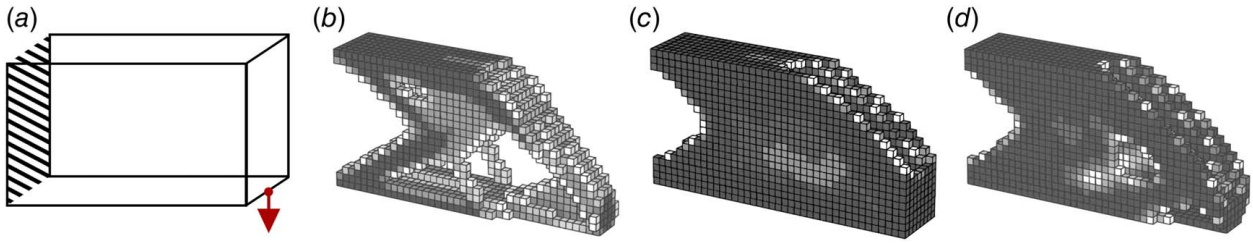


Fig. 10 Comparing the results for a 3D cantilever beam example. All examples are run for 200 epochs. (*b*) Top3d [28] (standard 3D topology optimization code using SIMP). (*c*) Using a neural network for density parametrization. (*d*) Using a neural network for density parametrization and additional initial strain energy input with log filtering. We observe that TONN and TONN-logCF choose to create a shell around both sides which gives an illusion that the volume fraction is higher. However, the volume fraction is also very close to the target volume fraction of 0.3 (both converged to 0.3003 specifically). (*a*) Boundary conditions, (*b*) Top3d ($c = 20.47$), (*c*) TONN ($c = 17.92$), and (*d*) TONN-logCF ($c = 16.96$).

examples with the load close to the center have a greater performance increase and a bigger gap in compliance. Our hypothesis is that the conditioning field approach performs best when the topology is complex. The complexity in geometry can occur based on the volume fraction constraint or the configuration of the boundary conditions. As the volume fraction decreases, thinner members are required which increase the complexity of the structure. Whereas the geometries in Fig. 6(*b*) showed that for the same volume fraction, the length scale of the part is also dependent on the boundary condition.

4.4 Additional Examples. In Fig. 8, we demonstrate the improvements resulting from the conditioning field on four complex boundary conditions in 2D. Cases 2, 3, and 4 in Fig. 8 have obstacle regions (passive elements). Furthermore, in Fig. 9, we analyze the impact of increasing the problem resolution (i.e., the FE mesh size) for the boundary conditions of case 1 in Fig. 8 and observe similar improvements. We also show the improvements seen for a 3D problem in Fig. 10.

5 Limitations and Future Work

We exploit the ability of neural networks as a universal function approximator to learn the additional mapping from the strain energy conditioning field to the density field output. Currently, the improvement with the conditioning field is not stable across all possible boundary condition configurations. More tuning and testing is required. Another aspect is that the current conditioning field remains fixed during optimization. This is due to the neural network's inability to encode temporal features. The strain energy field changes throughout the optimization, without the ability to capture the temporal feature of the changing strain energy field. As such, the neural network has difficulty providing stable optimization results.

This work also demonstrates promising results using a conditioning field for direct neural topology optimization. The strain energy field may not be the best conditioning field out there and future work may focus on trying out different combinations of conditioning fields similar to TopologyGAN [3]. This conditioning field approach may demonstrate great synergy with the existing

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024

data-driven approach. Using the output of data-driven topology optimization as the conditioning field, direct optimization can exploit a conditioning field that is much closer to the final solution. This reduces the complexity of the mapping function for which the neural network needs to learn. Since most data-driven approaches lack the guarantee of compliance minimization, direct optimization can serve as the final post-processing step to connect disconnected edges and truly minimize the compliance.

In this work, we also compare our result against SIMP using "88-lines" [27]. However, it may not be possible to determine which one is definitively better or worse. As each program is tuned for different platforms, the possible combinations of problem configuration are endless. Covering all possible problem configurations to reach a conclusion may not be possible. There are exciting possibilities with neural network-based topology optimization, for example, since the design density field is represented by a continuous function, one can infinitely upsample the result to obtain very crisp boundaries [4]. We can also use the same neural network architecture with physics-informed neural networks to conduct mesh-free topology optimization without a FE solver [29] to name a few.

6 Conclusions

We have proposed a novel approach for improving neural network-based topology optimization using a conditioning field. Our method involves using a topology neural network that is trained on a case-by-case basis to represent the geometry for a single topology optimization problem. By incorporating the strain energy field calculated on the initial design domain as an additional conditioning field input to the neural network, we have demonstrated that faster convergence speed can be achieved. Our results suggest that the efficacy of neural network-based topology optimization can be further improved using a prior initial field on the unoptimized domain. We believe that our proposed conditioning field initialization approach could have broad applications in the field of topology optimization, particularly for problems that involve complex geometries.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The data and information that support the findings of this article are freely available.[3]

References

[1] Bendsøe, M. P., 1989, "Optimal Shape Design as a Material Distribution Problem," Struct. Optim., **1**, pp. 193–202.

[2] Zhou, M., and Rozvany, G., 1991, "The COC Algorithm, Part II: Topological, Geometrical and Generalized Shape Optimization," Comput. Methods Appl. Mech. Eng., **89**(1–3), pp. 309–336.

[3] Nie, Z., Lin, T., Jiang, H., and Kara, L. B., 2021, "Topologygan: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain," ASME J. Mech. Des., **143**(3), p. 031715.

[4] Chandrasekhar, A., and Suresh, K., 2021, "TOuNN: Topology Optimization Using Neural Networks," Struct. Multidiscipl. Optim., **63**, pp. 1135–1149.

[5] Bense, N., and Kikuchi, M., 1988, "Generating Optimal Topologies in Structural Design Using a Homogenization Method, Comp," Methodss Appl. Mech. Eng., **71**(5), pp. 197–224.

[6] Allaire, G., Jouve, F., and Toader, A.-M., 2002, "A Level-Set Method for Shape Optimization," Compt. Rendus Math., **334**(12), pp. 1125–1130.

[7] Wang, M. Y., Wang, X., and Guo, D., 2003, "Level Set Method for Structural Topology Optimizations," Comput. Methods Appl. Mech. Eng., **192**(1/2), pp. 227–246.

[8] Xie, Y. M., Steven, G. P., Xie, Y., and Steven, G., 1997, *Basic Evolutionary Structural Optimization*, Springer.

[9] Woldseth, R. V., Aage, N., Bærentzen, J. A., and Sigmund, O., 2022, "On the Use of Artificial Neural Networks in Topology Optimisation," Struct. Multidiscipl. Optim., **65**(10), p. 294.

[10] Banga, S., Gehani, H., Bhilare, S., Patel, S., and Kara, L., 2018, "3D Topology Optimization Using Convolutional Neural Networks," preprint.

[11] Aage, N., Andreassen, E., and Lazarov, B. S., 2015, "Topology Optimization Using PETSc: An Easy-to-Use, Fully Parallel, Open Source Topology Optimization Framework," Struct. Multidiscipl. Optim., **51**, pp. 565–572.

[12] Yu, Y., Hur, T., Jung, J., and Jang, I. G., 2019, "Deep Learning for Determining a Near-Optimal Topological Design Without Any Iteration," Struct. Multidiscipl. Optim., **59**(3), pp. 787–799.

[13] Nakamura, K., and Suzuki, Y., 2020, "Deep Learning-Based Topological Optimization for Representing a User-Specified Design Area," preprint.

[14] Behzadi, M. M., and Ilieş, H. T., 2021, "Real-Time Topology Optimization in 3d Via Deep Transfer Learning," Comput.-Aided Des., **135**, p. 103014.

[15] Zheng, S., He, Z., and Liu, H., 2021, "Generating Three-Dimensional Structural Topologies Via a U-Net Convolutional Neural Network," Thin-Walled Struct., **159**, p. 107263.

[16] Mazé, F., and Ahmed, F., 2023, "Diffusion Models Beat GANs on Topology Optimization," Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Washington, DC, Feb 7–14, pp. 9108–9116.

[17] Chandrasekhar, A., and Suresh, K., 2021, "Length Scale Control in Topology Optimization Using Fourier Enhanced Neural Networks," CoRR, abs/2109.01861.

[18] Chandrasekhar, A., and Suresh, K., 2021, "Multi-Material Topology Optimization Using Neural Networks," CAD Comput. Aided Des., **136**.

[19] Deng, H., and To, A. C., 2020, "Topology Optimization Based on Deep Representation Learning (DRL) for Compliance and Stress-Constrained Design," Comput. Mech., **66**(2), pp. 449–469.

[20] Deng, H., and To, A. C., 2021, "A Parametric Level Set Method for Topology Optimization Based on Deep Neural Network," ASME J. Mech. Des., **143**(9), p. 091702.

[21] Zehnder, J., Li, Y., Coros, S., and Thomaszewski, B., 2021, "Ntopo: Mesh-Free Topology Optimization Using Implicit Neural Representations," Adv. Neural Inform. Process. Syst., **34**, pp. 10368–10381.

[22] Mai, H. T., Mai, D. D., Kang, J., Lee, J., and Lee, J., 2023, "Physics-Informed Neural Energy-Force Network: A Unified Solver-Free Numerical Simulation for Structural Optimization," Eng. Comput., 1–24.

[23] Hoyer, S., Sohl-Dickstein, J., and Greydanus, S., 2019, "Neural Reparameterization Improves Structural Optimization," preprint.

[24] Zhang, Z., Li, Y., Zhou, W., Chen, X., Yao, W., and Zhao, Y., 2021, "TONR: An Exploration for a Novel Way Combining Neural Network With Topology Optimization," Comput. Methods Appl. Mech. Eng., **386**, p. 114083.

[25] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., et al., 2015, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Software available from tensorflow.org.

[26] Kingma, D. P., and Ba, J., 2014, "Adam: A Method for Stochastic Optimization," preprint.

[27] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., 2011, "Efficient Topology Optimization in Matlab Using 88 Lines of Code," Struct. Multidiscipl. Optim., **43**, pp. 1–16.

[28] Liu, K., and Tovar, A., 2014, "An Efficient 3d Topology Optimization Code Written in Matlab," Struct. Multidiscipl. Optim., **50**, pp. 1175–1196.

[29] Joglekar, A., Chen, H., and Kara, L. B., 2023, "DMF-TONN: Direct Mesh-Free Topology Optimization Using Neural Networks," prepint.

[3]See Note 2.

Downloaded from http://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/146/6/061702/7220216/md_146_6_061702.pdf by Purdue University at West Lafayette user on 19 April 2024