

Generative Adversarial Networks for Topology Optimization with Orientation

Nathan Hertlein^a, Joseph Kubalak^b

^aMaterials & Manufacturing Directorate, Air Force Research Laboratory, Wright-Patterson AFB, Dayton, Ohio

^bApplied Research Laboratory, Pennsylvania State University, State College, Pennsylvania

Abstract

Topology optimization (TO) techniques leverage the unique design space of additive manufacturing (AM) to help designers find high-performing, readily manufacturable parts. In addition to geometric design, topology optimization with orientation (TOO) simultaneously optimizes for the local orientation of anisotropic mechanical properties inherent to the material extrusion AM process. However, these techniques tend to entail higher computational expense and potential for increased local optimality. Separately, generative models have been leveraged as efficient surrogate models for conventional TO, creating opportunities for computational savings after an initial training period. In this work, we study how a generative adversarial network (GAN) could accelerate AM-specific optimization such as TOO. We compare GAN configurations and loss functions, and quantify their impact on structural performance. Orientation predictions typically show larger error than topology predictions, but overall performance is promising for our problem class. These techniques could fit into ongoing efforts to increase generality in data-driven approaches for TOO.

Keywords: Topology Optimization with Orientation, Machine Learning, Generative Modeling, AI-Enabled Design

1. Introduction

Topology optimization (TO) describes a family of techniques that algorithmically determine the distribution of material that minimizes an objective function (often compliance) under some cost constraint (often a volume fraction). These processes produce highly organic design spaces that synergize well with the geometric flexibility inherent to additive manufacturing (AM) processes. However, common TO algorithms (e.g., solid isotropic material with penalization; SIMP) do not accurately model the mechanical properties produced by many AM processes. For instance, the mechanical performance of parts printed via the material extrusion (MEX) process are highly toolpath-dependent due to the anisotropic behavior of each individual deposition path [1]. To improve TO within the context of this phenomenon, significant research has been performed to explore topology optimization with orientation (TOO), which serves to simultaneously optimize material distribution and orientation.

Early investigations into TOO explicitly coupled material alignment to the shape of the geometry via calculation of either principal stresses [2] or strains [3]. However, this formulation can stymie geometric evolution, especially for complex loading conditions. A much more flexible and generalizable solution is to allow the optimizer direct control of the orientation field through separate design variables [4]. Often, each element is parameterized with an additional Euler angle to describe the desired deposition direction within that element [5]. Jiang et al. used this technique to optimize a planar Messerschmitt-Bölkow-Blohm beam; MEX toolpaths were formed from the results using contour-based toolpaths [6]. The toolpaths using the optimized

material orientations demonstrated a 30% increase in stiffness relative to toolpaths consisting of uni-directional deposition paths. These Euler angle parameterizations of the orientation space have also been extended to optimize 3D (i.e., non-planar) material orientations [7; 8].

While TOO has been shown to generate more performant designs relative to more conventional TO, it is highly susceptible to initial conditions; if unaddressed, material orientations will often converge to local minima such that the prescribed deposition direction is orthogonal to the anticipated load path. To overcome this issue, works have presented naturalization techniques for the Euler angle parameterization in both 2D [4] and 3D [9]. Other mechanisms have been introduced such as free material optimization (which features direct optimization of material properties [10]) and discrete material optimization (which reduces the orientation search to a set of discrete values [11]). While each of these improve final design performance by reducing convergence to local minima, they all add additional design variables that can significantly increase computational cost.

To improve performance of TOO without increasing the number of design variables, we investigate an alternative solution leveraging machine learning. The hypothesis is that, by taking on an up-front training step, the presented method can generate structures and orientation fields that closely match the gradient-descent TOO algorithm, with a dramatic per-problem runtime acceleration. Similar approaches have shown considerable promise in conventional TO. Deep neural networks such as convolutional neural networks (CNN) [12], generative adversarial networks (GAN) [13], and diffusion models [14], among others [15], have produced excellent results in terms of matching optimal topologies ordinarily produced by TO, including additive-focused TO constrained against overhangs [16]. Interestingly, in some cases, models have been observed to generate designs with stronger mechanical performance (lower compliance) than the TO process it was trained on [17]. Although these approaches are often restricted to operating within a single class of TO problem types within a single design domain, recent work has studied how to make these approaches more general. Particular ways of breaking down the problem into sub-modules, for example, may accelerate generalization to some degree [18], and some recent architectures including neural operators have shown strong transfer learning capabilities [19].

In this paper, we investigate whether the benefits of one such model, the GAN, carry into TOO. This entails significant additional complexity, as orientation data presents an entirely new field to predict with different characteristics not present in topology fields seen in previous neural network-based TO. This paper investigates techniques for operating on two complementary fields (topology *and* orientation), methods for helping the model focus only on important regions of the orientation field, and the overall achievable accuracy and computational acceleration relative to TOO.

2. Methods

2.1. Topology Optimization with Orientation

The general TOO problem statement is shown in Equ. 1. Broadly, it is stated to minimize the compliance of the structure relative to the specified loading and boundary conditions. Each element in the design space is assigned a pseudo-density (ρ) and orientation (θ) design variable that describes material placement and orientation, respectively.

$$\begin{aligned}
& \min_{\boldsymbol{\rho}, \boldsymbol{\theta}} : c(\boldsymbol{\rho}, \boldsymbol{\theta}) = \mathbf{U}^T \mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\theta}) \mathbf{U} \\
& \text{subject to : } \frac{V(\boldsymbol{\rho})}{V_0} \leq f \\
& \quad : \mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\theta}) \mathbf{U} = \mathbf{F} \\
& \quad : \mathbf{0} < \boldsymbol{\rho}_{min} \leq \boldsymbol{\rho} \leq 1 \\
& \quad : -\frac{\pi}{2} \leq \boldsymbol{\theta} \leq \frac{\pi}{2}
\end{aligned} \tag{1}$$

\mathbf{U} is the global displacement vector, \mathbf{K} is the global stiffness matrix as defined in Equ. 2. $V(\boldsymbol{\rho})$ is the mass of the current solution, V_0 is the mass of a fully dense design space, and their ratio must be less than or equal to a maximum allowable value f . \mathbf{F} is the forcing vector acting on the design space and $\boldsymbol{\rho}_{min}$ is the minimum allowable element pseudo-density. This is established to be larger than zero to prevent singularities in \mathbf{K} . The bounds on $\boldsymbol{\theta}$ serve to address the rotational symmetries in the orientation design space; fibers are considered undirected (e.g., an orientation of 0 is equivalent to π and vice versa). Therefore, orientations are condensed to the range $[-\pi/2, \pi/2]$ by flipping orientations via the addition or subtraction of π where necessary.

$$\mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\theta}) = \sum_{i=1}^{N_e} \rho_i^\eta \mathbf{L}_i^T \iiint_{\Omega_i} (\mathbf{B}^T \mathbf{T}^T(\theta_i) \mathbf{E}_0 \mathbf{T}(\theta_i) \mathbf{B}) \partial \Omega_i \mathbf{L}_i \tag{2}$$

N_e is the number of elements in the design space, and Ω_i is the volume of element i . \mathbf{L}_i is the locator matrix that places the element stiffness matrix into the global stiffness matrix, and \mathbf{B} is the strain-displacement matrix for the specified element type (in this work, four-node quadrilaterals). The strain transformation matrix $\mathbf{T}(\theta_i)$ is written in terms of the rotation matrix $\mathbf{R}(\theta_i)$ as defined in [20]. \mathbf{E}_0 is the matrix of linear elastic constitutive properties describing the fully dense material, and η is the SIMP penalty factor.

The density field was initialized to the selected volume fraction, and the orientation field was initialized to the principal strains acting on the design space when considering isotropic material properties. To create a mesh independent solution and prevent checkerboard patterning in the density field, the Heaviside projection method [21] was imposed. A similar blur filter was applied to the orientation field to prevent rapid changes in fiber direction [8]. Both filters were applied using a neighborhood radius of $r_{min} = 3$. The SIMP penalty factor used in Equ. 2 was initialized at $\eta = 1$ and increased by 1 every 50 iterations to a maximum value of 3. The Method of Moving Asymptotes [22] was used as the optimizer.

2.2. Visualization

An example TOO solution is included in Fig. 1. The first row shows raw density ($\boldsymbol{\rho}$) and orientation ($\boldsymbol{\theta}$) for the entire rectangular design domain, while the second row only shows solid regions, identified as those elements for which $\rho > 0.5$. As indicated in the example, the majority of the design domain exists at $\rho \approx 0$, representing void regions. However, orientation is highly variable throughout the entire design domain, especially in the void regions, even though orientation is only practically meaningful in solid regions. This will create challenges for machine learning (ML) approaches and surrogate model types originally studied for application to density data.

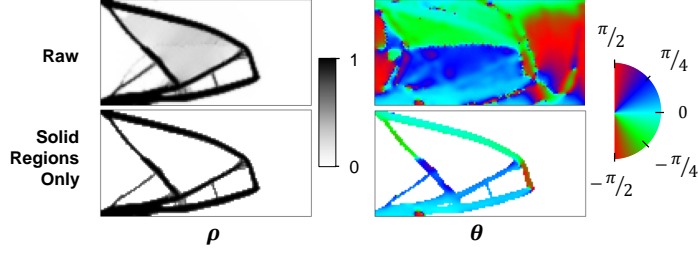


Figure 1: Example TOO solution. Both density ρ and orientation θ fields are shown for one example problem. In the second row, fields are masked to show only solid regions ($\rho > 0.5$).

2.3. Machine Learning Architecture

The objective of the surrogate models explored in this work is to predict ρ and θ that collectively minimize c in Equ. 1. As element values, these terms can be represented as 2D images whose pixel intensities correspond to element densities or orientation angles, respectively. This renders the prediction task a good fit for image-based neural networks, such as GANs. In this work, we consider a conditional GAN shown in Fig. 2, based on the *pix2pix* architecture [23] for image translation. To act as the input, a given TOO problem definition, \mathbf{P} , is represented as a three-channel image, with a channel for each component of an applied load and a channel for the boundary conditions. All pixels are set to 0, except where the load or boundary condition is applied. On x- and y-load channels, the pixel in the same location as the point force is assigned an intensity value equal to the load’s magnitude on the corresponding axis. On the boundary condition channel, pixels corresponding to elements with fixed nodes are assigned a value of 1. An optional fourth input image channel is included in some GAN configurations, as explained further below. The output image is either a one- or two-channel image representing the quantity(ies) of interest (ρ and/or θ).

The inputs are passed into a CNN known as the generator, whose task is to predict the final desired output (ρ and/or θ). An additional network known as the discriminator is alternately shown either an output from the generator or a ground truth output directly from the training dataset, with the goal of predicting which type of image it was shown. Its loss function is the binary cross entropy between its prediction and the correct response. Traditional back-propagation is used to update the learnable parameters within the discriminator, and this can be backpropagated further through the generator to update its learnable parameters. For the

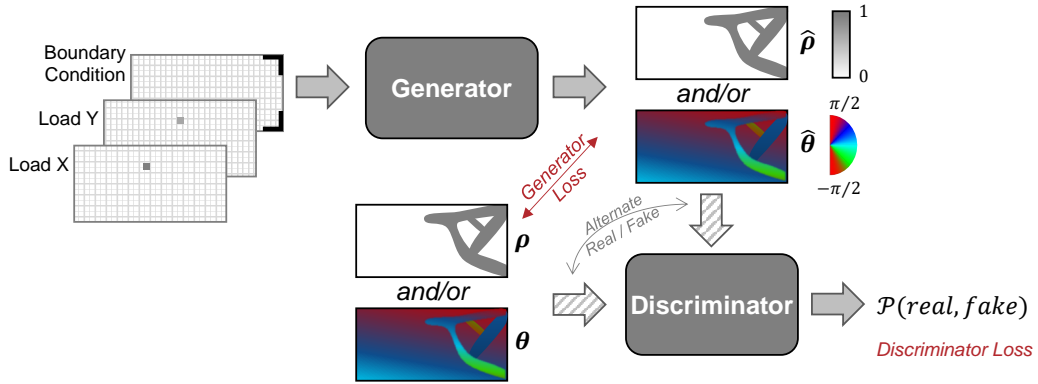


Figure 2: Overview of the general GAN architecture, in the context of TOO. The problem definition \mathbf{P} can be represented as a 3-channel image. This is used as input to an adversarial training routine, where the generator attempts to create realistic ρ and/or θ as solutions to \mathbf{P} , and the discriminator tries to discern real vs. generated solutions, to support the generator’s training.

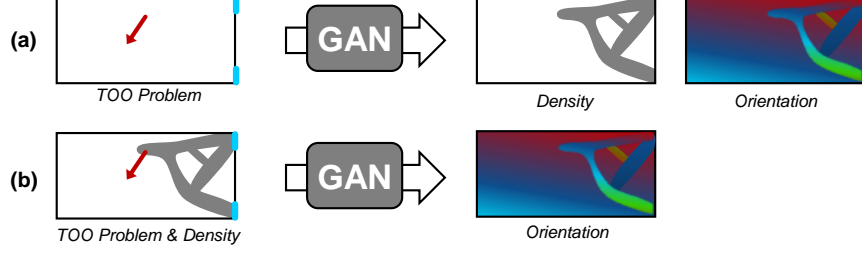


Figure 3: Overview of the GAN configurations considered for TOO. The problem definition \mathbf{P} can be used as the sole input to a model that predicts both ρ and θ (a). Alternatively, ρ could be used as an additional input to predict θ alone (b).

generator, a per-pixel loss function is also calculated and added to this loss with a weighting factor. Multiple per-pixel loss functions were considered, as explained below.

Multiple GAN configurations and loss functions were tested in this work to handle the additional complexities presented by the orientation field, as summarized in Fig. 3. The first configuration consisted of providing only the three TOO problem definition channels \mathbf{P} (boundary condition, horizontal load, and vertical load) as input, and taking both density ρ and orientation θ as output. This represents the most straightforward mapping as it directly mimics the desired inputs and outputs in practice. An additional test case was examined in which the density ρ was moved to the input side, to test the model’s ability to generate accurate orientation fields for a given design. This could be useful, for example, in cases where a design is already determined and the designer wants to prescribe a deposition orientation field.

For each configuration, multiple per-pixel loss functions were tested for the generator, again driven by the added complexity of θ . First, mean absolute error (MAE) was calculated as

$$L_0 = \frac{1}{n} \sum_{i=1}^n |e| \quad (3)$$

and used as the loss function for both ρ and θ channels, to encourage the model to match its predictions to the entire ground truth images as closely as possible. For density, e_ρ could be calculated as the difference between ground truth and predicted density as $\rho_i - \hat{\rho}_i$. For orientation, e_θ was calculated using the angle between the Euler angles in the ground truth and prediction for a given element. In practice, this meant calculating

$$e_\theta = \arccos\left(\frac{\theta_i \cdot \hat{\theta}_i}{\|\theta_i\| \|\hat{\theta}_i\|}\right) \quad (4)$$

This calculation considers equivalence of θ values exactly π radians from each other. For example, when $\theta = -3\pi/4$, a $\hat{\theta} = \pi/4$ would be assessed $e_\theta = 0$, which is consistent with the physical equivalence of such angles during MEX deposition. Note that the term within the inverse cosine was clipped to the range $(-1, 1)$ to prevent numerical difficulties.

Uniform calculation of MAE throughout the design domain for ρ makes intuitive sense, as the density field throughout the entire design domain is necessary for fully describing the design. On the other hand, orientation θ only matters in practice in regions where ρ is greater than the threshold (e.g., 0.5) necessary to trigger printing solid material in the MEX context considered. In other words, while L_0 draws the GAN’s attention to the entire orientation field, orientation only practically matters in solid regions. In practice, during printing, θ is ignored in void regions. To encourage the GAN to attend to θ primarily *in solid regions*, a weighted MAE term is calculated by scaling e_θ proportionally to the density at a given element.

$$L_w = \frac{1}{n} \sum_{i=1}^n \rho_i |e_\theta| \quad (5)$$

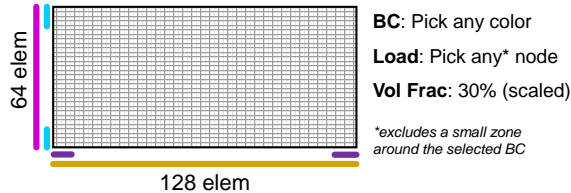


Figure 4: Overview of the space of TOO problems considered. The space of problems included four possible boundary conditions types, and a single point load applied at any angle anywhere in the domain beyond a small radius (10 elements) of the boundary condition. The volume fraction was nominally taken as 0.30, but scaled by the distance between the load and the side with the boundary condition.

The range of ρ is naturally $[0, 1]$, making this a convenient scaling factor without further normalization. For all calculations, θ was linearly scaled from its range in the raw training data of $[-\pi, \pi]$ to a more ML-appropriate range of $[-1, 1]$.

2.4. Problem Domain and Training Data

The TOO problem domain of interest in this work consisted of a rectangular design domain of size 64 by 128 elements, as illustrated in Fig. 4. One of a set of four boundary conditions was selected, which involved fully constraining either the full left hand side or the full bottom side, or constraining 5 nodes on the ends of either of those two sides. A load was applied at any node outside of a small radius of 10 elements of the boundary conditions. The nominal volume fraction f was set at 30%; however, to ensure that typical bracket-like TO geometries resulted rather than large blocks of solid material, f was linearly scaled down to a minimum of 20% based on the distance between the load and the side with the boundary condition relative to the overall domain size.

Training data were collected by randomly selecting problems from this space and performing the TOO process described in Section 2.1. The problem definition (including the loads, boundary condition, and scaled volume fraction) was saved along with the corresponding optimized density and orientation fields. A total of 9,754 samples were collected, with 1,000 reserved for validation and testing. Each sample was optimized using 4 cores of an AMD EPYC 7702, and the average time to generate each sample was 950 ± 317.4 s. The literature frequently cites datasets in the tens of thousands of samples for similar TO problems [13; 16; 24], which are even simpler than the TOO problems considered here. However, this amount was considered sufficient for exploration of GAN configurations and loss functions, and initial demonstration of capabilities.

3. Results

Four GANs were trained: (1) one that predicts ρ and θ using only L_0 , (2) one that predicts ρ and θ using L_w for θ , (3) one that predicts only θ using L_0 , and (4) one that predicts only θ using L_w . Training for each took place on an Nvidia V100 GPU and ran for 100 epochs to give a good picture of convergence trajectory. During the training process, which required under 2 hours per model, training was overall stable. Shown in Fig. 5, the loss curves indicate at least some degree of overfitting. This likely stems from the relatively small training dataset size. Additional data, or augmentation of the existing dataset, could be considered to reduce these effects. The GAN that predicted θ with the weighted loss function L_w is intuitively the simplest mapping, and it showed the least degree of overfitting. After training, designs could be queried from the GAN extremely efficiently depending on hardware. For example, a Core i7 laptop CPU averaged around 0.01 s per GAN design.

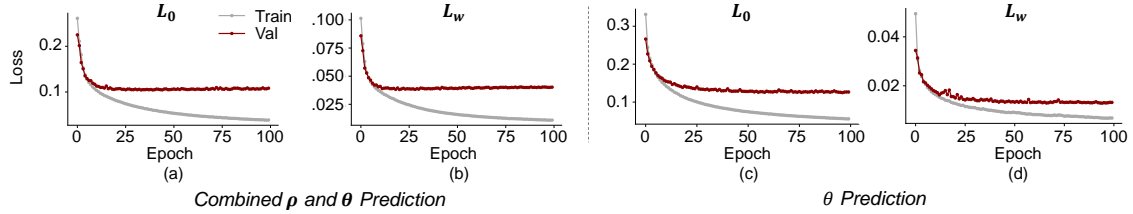


Figure 5: The GAN training curves. Training curves are plotted for each of the 4 GANs, with training loss in gray and validation loss in red. Note that loss functions are raw values, representing a sum of the loss for ρ and θ . The loss type used for θ (unweighted as L_0 , or weighted as L_w) is shown above each plot.

A single validation sample is thoroughly represented in Fig. 6, for the GAN that predicts both ρ and θ trained using L_0 . In this particular example, the density prediction is generally qualitatively similar to the groundtruth from TOO, but it misses some structural pixels in the right hand strut - a known occurrence with image ML-based TO surrogates. The second row shows raw orientation values, as they came from the GAN, after being linearly rescaled from the range that the GAN trained on, $[-1, 1]$, to the range in the training dataset, $[-\pi, \pi]$. Even visually, significant deviation is noticeable here between the groundtruth and the GAN outputs, including perhaps the majority of the entire image. Interesting, the lower half of the GAN prediction includes a checkerboard-like pattern that alternates between positive and negative θ values. In practice, portions of the rescaled GAN-output range of $[-\pi, \pi]$ are redundant: for example, $\theta = \pi/4$ is functionally equivalent in $\theta = -3\pi/4$ as the two orientations imply identical material properties. This fact was considered in the GAN's orientation loss function definition, as described in Sec. 2.3. Both the true and predicted θ fields are included as a third row in Fig. 6, after flipping orientations by addition/subtraction of π where necessary to condense the orientation range to the range of $[-\pi/2, \pi/2]$. The much closer match here indicates that, while the GAN was not tasked with matching the exact numerical values in the training set, it demonstrated learning of the underlying trends in orientation distributions, such that it could perform a reasonable mapping in the physically-meaningful θ space.

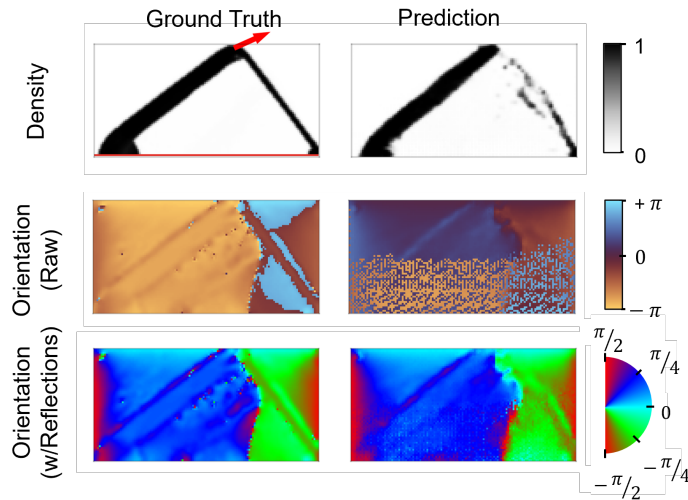


Figure 6: An example output from the GAN that predicts density and orientation (unweighted loss function). The TOO design features a solid, singular main truss member. The GAN qualitatively captures the overall density distribution to some degree, though it does not properly resolve the support on the right-hand side. The raw orientation field is predicted quite differently from the ground truth data, as shown by the raw image in the second row. When orientation values are condensed to the proper range of $[-\pi/2, \pi/2]$ however, the orientation fields are much more similar.

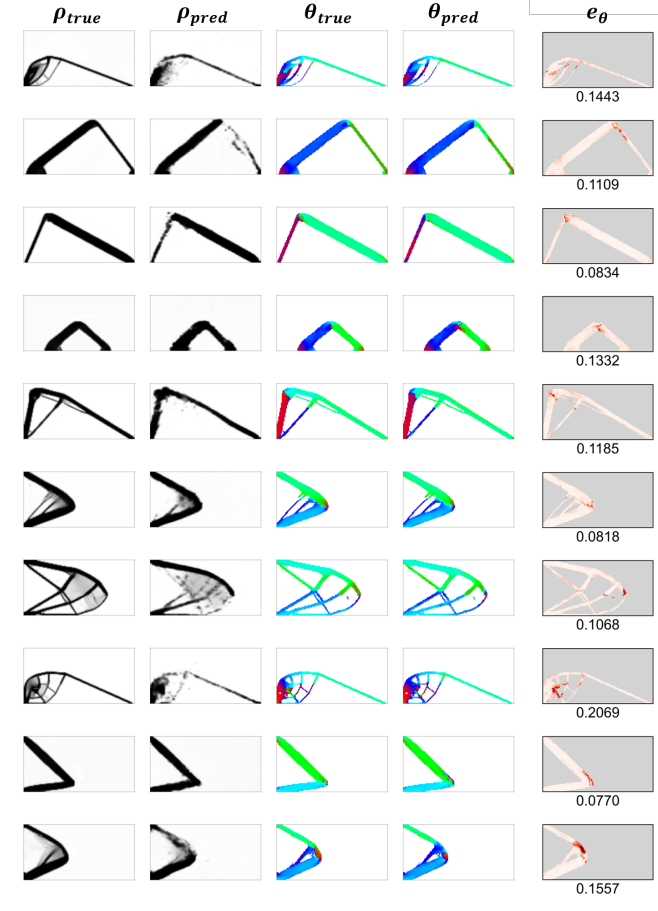


Figure 7: Several outputs from the GAN that predicts density and orientation, along with orientation error heatmaps. Orientation error happens mostly at the transitions, suggesting that the overall orientation maps match the overall shape (on a strut-by-strut basis) well, with error confined to transitions between regions.

A set of randomly-selected validation samples are shown in Fig. 7 for the GAN that predicts density and orientation, trained with the unweighted loss function. The entire ρ fields are shown, with 0 as white. The θ channel is shown only in regions where the ground truth design is solid ($\rho > 0.5$). Considerable error exists for some ρ samples, including especially disconnected struts, likely resulting from the relatively small number of training samples. Orientation predictions tend to capture overall trends in orientation well, with deposition orientation often aligned with the expected load path. A fifth column in the figure illustrates the spatial distribution of error in orientation predictions for solid regions. In most cases, error is most prevalent in areas of transition between two regions of relatively constant orientation. Although the model often struggles to perfectly locate transitions between regions, these examples suggest that it has learned the proper overall trends in orientation fields.

To quantitatively evaluate results, the error in structural compliance, c , was calculated on all 1000 validation predictions for each of the four GANs. Because some predictions contained some ρ values outside of $[0, 1]$, all density values were clipped to this range. A relative comparison of c_{GAN} to c_{true} was made, according to

$$c_e = \frac{c_{GAN} - c_{true}}{c_{true}} \quad (6)$$

Positive values indicate higher (less desirable) compliance in the GAN design than the design

Table 1: Average errors for each GAN. Error metrics are shown for the four GANs which predicted either both ρ and θ or just θ , and either used a unweighted or density-weighted orientation loss function. MAE for ρ and θ are shown in the first two rows, followed by the relative compliance error, \bar{c}_e from Eq. 6.

	ρ & θ	ρ & θ (Weighted Loss)	θ	θ (Weighted Loss)
ρ	.061	.061	-	-
θ	.143	.117	.093	.074
\bar{c}_e	.327	.188	.105	.070

from TOO. A small number of outliers with $c_e > 1$ were identified ($\sim 3\%$ for the GAN predicting both ρ and θ with unweighted loss function, and less than 1% for all others). Table 1 shows the average of c_e for each GAN, along with the MAE on both the ρ (if applicable) and θ output channels. For orientation predictions, MAE values here were calculated on the condensed (properly flipped to fit the range $[-\pi/2, \pi/2]$) θ data. The calculation also only considers regions that are solid ($\rho < 0.5$). The ground truth density was used to determine solid regions for this, so that a direct per-pixel comparison could be made. All values of c_e are also included in the histogram of Fig. 8.

Across both MAE and c_e metrics, GANs predicting only θ gave better performance. Intuitively, this mapping is more straightforward: it involves only a single output channel, and receives an additional input channel in ρ . Rather than generating a density layout, in some sense, it is performing a task more similar to image translation (grayscale density image-to-density colored by orientation), which has shown excellent results in non-TO contexts and was the original motivation behind the *pix2pix* architecture from which the GAN architectures here were developed.

Weighting the loss function on the orientation channel showed a positive effect regardless of GAN configuration. This suggests that the proposed weighting led to GAN parameter updates more focused around important (solid) regions without being bogged down by attempts to match θ in less important (void) regions. Improvements from this loss function were significant, on the order of 30-40% in \bar{c}_e .

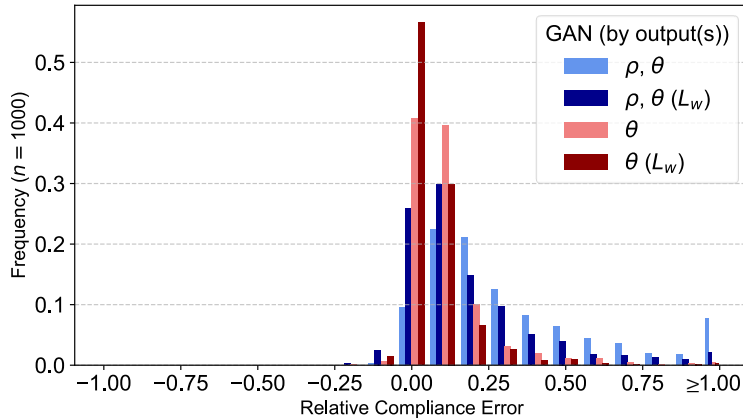


Figure 8: Histogram of relative compliance error, c_e For each of the four GANs considered in this work, c_e is plotted across the 1000 validation samples. GANs predicting only θ show higher peaks near 0 error than those predicting both ρ and θ . Note that c_e is clipped at ± 1 . Densities in GAN-generated designs were clipped to the range $[0, 1]$ before compliance calculation.

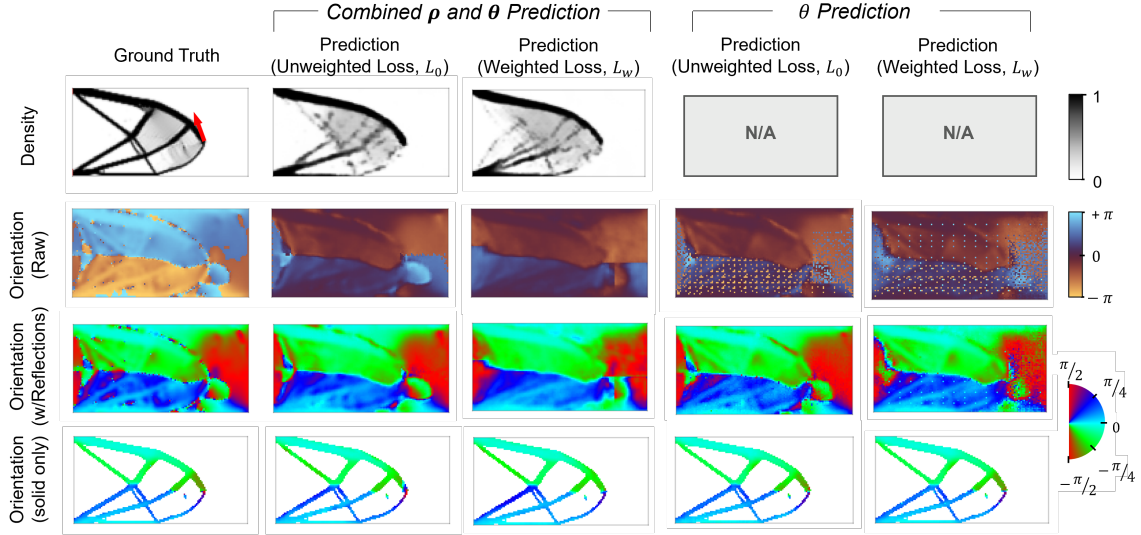


Figure 9: An example output from all the GANs. For each GAN tested, the first row shows the density (if applicable), and the remaining rows show the orientation (raw after rescaling, after flipping all angles to lie in the range $[-\pi/2, \pi/2]$, and the same plot after masking by the ground truth $\rho > 0.5$).

A validation example is shown across all GANs in Fig. 9, for density, raw orientations, reflected orientations, and reflected orientations after hiding void regions. This sample was particularly challenging in terms of density for both relevant GANs. In terms of orientation, however, each GAN properly predicted the overall trends, as evidenced by the bottom two rows. On the raw θ predictions in the second row, in every case, lower-magnitude angle values were generally preferred over the higher magnitude equivalent (e.g., $\pi/6$ instead of $-5\pi/6$), representing a possible general preference by the GAN for lower magnitude values when there are multiple options. Finally, on the θ -only GAN with the weighted loss function, a series of dots of high error are present in a grid pattern. These small dots could likely be removed by image processing techniques.

4. Discussion

As shown in the results, GAN performance is promising when predicting either ρ and θ given the TOO problem \mathbf{P} (Fig. 3(a)), or just predicting θ when given \mathbf{P} and ρ (Fig. 3(b)). Performance was better in the latter configuration, which aligns with intuition that predicting a single output channel from more informative input (i.e., including ρ) is at least a slightly simpler task, reminiscent of image colorization - a known strong point of the ML model type considered. Weighting the loss function to help the model focus more on learning relevant (i.e., solid) regions of θ also boosted results, giving noticeably better compliance errors, \bar{c}_e . This points to a general challenge in surrogate modeling for TOO: orientation data fills the entire design domain but does not necessarily have physical meaning throughout. In this work, weighting the loss function to focus on solid regions was valuable; future work may reveal additional means of confronting this challenge.

To understand the regions of the problem space where the models struggled the most, heatmaps were created to show GAN error as a function of load and boundary condition location. Shown in Fig. 10, these plots indicate that, for ρ , the model struggles most when the load is far from the boundary conditions, while, for θ , the greatest struggle is when the load is near the boundary conditions. Ordinarily, additional training data collection could target errors of high error, but in this case, because the error is spread nearly oppositely for the two channels, future work should take care in sampling further training data.

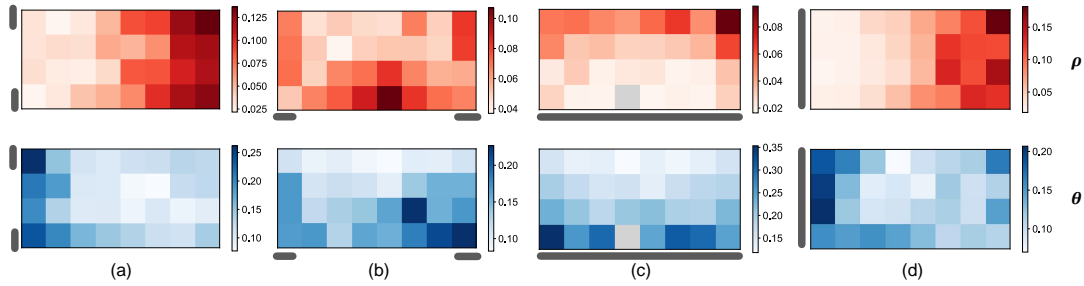


Figure 10: Heatmaps of error as a function of load location. While ρ error tends to be higher farther from the boundary conditions, θ error tends to be higher closer to the boundary conditions. These plots are from the GAN that predicts both ρ and θ , with unweighted loss function L_0 .

With this mix of promise and unique challenges, a strong future use case for ML surrogates could be as tools for warm-starting costly TOO routines. TOO entails significant computational burden beyond that of density-only TO, which usually only involves half the number of design variables. A compelling use case is also suggested by the configuration in this work that predicts orientation only, given the loading condition and the optimal density (Fig. 3(b)). In principle, a TO problem could be solved without consideration for orientation, which could then be rapidly predicted in a secondary step by a ML surrogate. Both approaches warrant further investigation and thorough evaluation on problem domains and datasets beyond the present work.

5. Conclusion

Surrogate models, based on the GAN architecture, were tested for their ability to map to optimal densities and orientations in a TOO routine that calculated both optimal density and optimal AM deposition orientation. Multiple alternative configurations were considered, including prediction channels and loss functions. Even with a limited dataset size due to TOO's relatively high computational expense, model performance was fairly strong across configurations, with improvements from both reducing the output channels and utilizing a custom loss function that pays greatest attention to regions of high material density. The models explored here show promise for use as, for example, warm-starting tools for TOO, or for assigning AM orientation fields on already-optimized density fields. For AM routines that induce anisotropy, this could become a convenient tool for further integration of manufacturing into the design process.

Acknowledgment

NH would like to acknowledge the support of the Air Force Research Laboratory. JK was affiliated with Virginia Tech as a research scientist at the time this research was conducted, and computational resources for generating TOO training data were provided by Virginia Tech Advanced Research Computing.

Data Availability

Data will be made available on reasonable request.

References

- [1] S.-H. Ahn, M. Montero, D. Odell, S. Roundy, P. K. Wright, Anisotropic material properties of fused deposition modeling ABS, *Rapid Prototyping Journal* 8 (4) (2002) 248–257.
- [2] H. C. Cheng, N. Kikuchi, Z. D. Ma, An improved approach for determining the optimal orientation of orthotropic material, *Structural Optimization* 8 (2-3) (1994) 101–112. doi:10.1007/BF01743305.
- [3] P. Pedersen, On optimal orientation of orthotropic materials, *Structural Optimization* 1 (2) (1989) 101–106. doi:10.1007/BF01637666.
- [4] T. Nomura, E. M. Dede, J. Lee, S. Yamasaki, T. Matsumori, A. Kawamoto, N. Kikuch, General topology optimization method with continuous and discrete orientation design using isoparametric projection, *International Journal for Numerical Methods in Engineering* 101 (February) (2015) 1102–1119. arXiv:1010.1724, doi:10.1002/nme. URL <http://onlinelibrary.wiley.com/doi/10.1002/nme.3279/full>
- [5] R. Hoglund, D. E. Smith, Continuous Fiber Angle Topology Optimization for Polymer Fused Filament Fabrication, in: *27th Annual International Solid Freeform Fabrication Symposium*, Vol. 1, 2016, pp. 1078–1090.
- [6] D. Jiang, R. Hoglund, D. E. Smith, Continuous fiber angle topology optimization for polymer composite deposition additive manufacturing applications, *Fibers* 7 (2) (2019). doi:10.3390/FIB7020014.
- [7] N. Boddeti, Z. Ding, S. Kaijima, K. Maute, M. L. Dunn, Simultaneous Digital Design and Additive Manufacture of Structures and Materials, *Scientific Reports* 8 (1) (2018) 1–26. doi:10.1038/s41598-018-33454-3.
- [8] M. P. Schmidt, L. Couret, C. Gout, C. B. Pedersen, Structural topology optimization with smoothly varying fiber orientations, *Structural and Multidisciplinary Optimization* 62 (6) (2020) 3105–3126. doi:10.1007/s00158-020-02657-6.
- [9] J. R. Kubalak, A. L. Wicks, C. B. Williams, Investigation of Parameter Spaces for Topology Optimization with Three-Dimensional Orientation Fields for Multi-Axis Additive Manufacturing, *Journal of Mechanical Design* (2020) 1–18doi:10.1115/1.4048117.
- [10] M. Kocvara, M. Stingl, J. Zowe, Free material optimization: Recent progress, *Optimization* 57 (1) (2008) 79–100. doi:10.1080/02331930701778908.
- [11] C. Wu, Y. Gao, J. Fang, E. Lund, Q. Li, Discrete topology optimization of ply orientation for a carbon fiber reinforced plastic (CFRP) laminate vehicle door, *Materials and Design* 128 (May) (2017) 9–19. doi:10.1016/j.matdes.2017.04.089. URL <http://dx.doi.org/10.1016/j.matdes.2017.04.089>
- [12] Y. Yu, T. Hur, J. Jung, I. G. Jang, Deep learning for determining a near-optimal topological design without any iteration, *Structural and Multidisciplinary Optimization* 59 (3) (2019) 787–799. doi:10.1007/s00158-018-2101-5.
- [13] Z. Nie, T. Lin, H. Jiang, L. B. Kara, TopologyGAN: Topology optimization using generative adversarial networks based on physical fields over the initial domain, *Journal of Mechanical Design*, *Transactions of the ASME* 143 (3) (3 2021). doi:10.1115/1.4049533.
- [14] G. Giannone, F. Ahmed, Diffusing the Optimal Topology: A Generative Optimization Approach, in: *IDETC-CIE*, ASME, Boston, 2023.

- [15] C. Qian, W. Ye, Accelerating gradient-based topology optimization design with dual-model artificial neural networks, *Structural and Multidisciplinary Optimization* 63 (4) (2021) 1687–1707. doi:10.1007/s00158-020-02770-6.
- [16] N. Hertlein, P. R. Buskohl, A. Gillman, K. Vemaganti, S. Anand, Generative Adversarial Network for Early-Stage Design Flexibility in Topology Optimization for Additive Manufacturing, *Journal of Manufacturing Systems* 59 (2021) 675–685. doi:10.1016/j.jmsy.2021.04.007.
- [17] N. Hertlein, A. Gillman, P. R. Buskohl, Generative adversarial design analysis of non-convexity in topology optimization, *IDETC-CIE* (2022). doi:10.1115/DETC2022-89997.
- [18] Y. Joo, Y. Yu, I. G. Jang, Unit Module-Based Convergence Acceleration for Topology Optimization Using the Spatiotemporal Deep Neural Network, *IEEE Access* 9 (2021) 149766–149779. doi:10.1109/ACCESS.2021.3125014.
- [19] K. Liang, D. Zhu, F. Li, A Fourier neural operator-based lightweight machine learning framework for topology optimization, *Applied Mathematical Modelling* 129 (2024) 714–732. doi:10.1016/j.apm.2024.02.011.
- [20] R. D. Cook, D. S. Malkus, M. E. Plesha, R. J. W. Witt, *Concept and Applications of Finite Element Analysis*, John Wiley Sons, Inc., 2002.
- [21] J. K. Guest, A. Asadpoure, S. H. Ha, Eliminating beta-continuation from Heaviside projection and density filter algorithms, *Structural and Multidisciplinary Optimization* 44 (4) (2011) 443–453. doi:10.1007/s00158-011-0676-1.
- [22] K. Svanberg, The method of moving asymptotes - a new method for structural optimization, *International Journal for Numerical Methods in Engineering* 24 (2) (1987) 359–373. arXiv:arXiv:1011.1669v3, doi:10.1002/nme.1620240207. URL <http://onlinelibrary.wiley.com/doi/10.1002/nme.1620240207/abstract>
- [23] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, *CVPR* (11 2016). doi:10.48550/arXiv.1611.07004.
- [24] C. Sharpe, C. C. Seepersad, Topology Design with Conditional Generative Adversarial Networks, in: *IDETC*, Anaheim, CA, 2019.