**Science Arts & Métiers (SAM)**

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/23384

**To cite this version :**

Waad ALMASRI, Dimitri BETTEBGHOR, Faouzi ADJED, Florence DANGLADE, Fakhreddine ABABSA - Geometrically-driven generation of mechanical designs through deep convolutional GANs - Engineering Optimization p.1-18 - 2022

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

Arts et Métiers Institute of Technology

# Geometrically-driven generation of mechanical designs through deep convolutional GANs

Waad Almasri [a,b], Dimitri Bettebghor [a], Faouzi Adjed [a], Florence Danglade[b] and Fakhreddine Ababsa [b]

[a]Expleo, Montigny-le-Bretonneux, France; [b]Laboratoire d'Ingénierie des Systèmes Physiques et Numériques (LISPEN), Arts et Métiers, Cluny, France

**ABSTRACT**

Despite the freedom Additive Manufacturing (AM) offers when manufacturing organic shapes, it still requires some geometrical criteria to avoid a part's collapse during printing. The most synergetic design approach to AM is Topology Optimization (TO), which finds an optimal free-form given mechanical constraints. However, it is hard for TO to integrate these layout geometry-related constraints and it seldom proposes printable shapes. Therefore, this work leverages the Deep Learning (DL) capability to handle spatial correlations within the mechanical design process by integrating the layout and mechanical constraints at the conceptual level. It proposes a DL-layout-driven solution (DL-TO) trained via a triple-discriminator Generative Adversarial Network (GAN) framework. The DL-TO's performance is demonstrated by generating mechanically valid 2D designs conforming with layout constraints in a fraction of a second. DL-TO's creativity is illustrated by its capability to generate designs with unseen input constraints (passive/active elements) and to propose several shapes for the same input mechanical constraints, a task that is hard for a traditional TO to achieve.

**KEYWORDS**

Topology optimization; Deep learning; Generative adversarial networks

## 1. Introduction

The growing need for fast, organic, versatile and cost- and material-efficient products in the industrial world drives research to develop design approaches accounting for all these criteria. Topology Optimization (TO) falls in the category of these design approaches. It allows the generation of shapes with curvatures and fine details, given a set of parameters such as loads, boundary conditions and volume fraction.

Complementary to the above comes Additive Manufacturing (AM), which allows the printing of any form. However, this synergy is not as idealistic as it looks. Indeed, AM requires the design's shape to comply with some geometrical criteria that are hard to integrate into TO's formulation (Adam and Zimmer 2014). Thus, designing optimal printable parts requires the intervention of experts to interpret the shapes proposed by TO. Nonetheless, this reinterpretation phase can compromise the initial design's optimality, is time-consuming and depends on the engineer's expertise.

Consequently, with the flourishing role of AM in industry, it is further interesting to find a method that simultaneously considers mechanical constraints (boundary conditions, loads and volume fraction) and geometrical constraints (overhangs, complexity, lengths and thicknesses) at the conceptual

**CONTACT** Waad Almasri waad.almasri@ensam.eu

https://doi.org/10.1080/0305215X.2022.2144847

level. A recent survey conducted by Subedi, Verma, and Suresh (2020) has shown that half of TO practitioners regret the absence of layout- and manufacturing-related plugins in TO's software.

Thus, research is recently focusing on integrating both these constraints into TO's formulation. Z. Zhang and Zhou (2018), Han *et al.* (2019) and Bi, Tran, and Xie (2020) adapted TO approaches to account for overhang limitations and to deliver self-supporting designs. Xu *et al.* (2021) integrated AM support structure and thin feature constraints into bidirectional evolutionary structural optimization to obtain AM-friendly designs. Li *et al.* (2020), Cunfu Wang and Quian (2020) and Matos, Rocha, and Costa (2021) focused on including the build orientation into TO, especially since the AM part's accuracy and cost are affected mainly by its build orientation.

Nonetheless, TO is an iterative, Finite-Element-based (FE-based) method, hence computationally expensive. While it converges for simple input constraints in a few seconds, its efficiency drops remarkably when the input design space (*e.g.* the mesh size) increases and input conditions complexify. Moreover, integrating geometric/manufacturing constraints into its formulation, when possible, adds computational costs and often compromises its convergence (Ranjan, Samant, and Anand 2017).

For that reason, other research has turned to machine and Deep Learning (DL) techniques to accelerate the TO process. DL architectures have proven efficient and robust in learning complex spatial correlations and extracting high-level spatial shape-related features from real-world images (Aloysius and Geetha 2017).

The overall DL-based TO approaches found in the literature can be divided into two parts.

On the one hand, the first part partially replaced FE by substituting sensitivity analysis with Neural Networks (NNs) (S. Bi, J. Zhang, and G. Zhang 2020; Chandrasekhar and Suresh 2021) by using super-resolution NNs to enhance the resolution of intermediate FE-TO's outputs (Sosnovik and Oseledets 2019, Kallioras, Kazakis, and Lagaros 2020, Chunpeng Wang *et al.* 2021) by creating a deep NN-based TO that is trained and penalized using a quality function based on FE (Halle, Campanile, and Hasse 2020), *etc.*

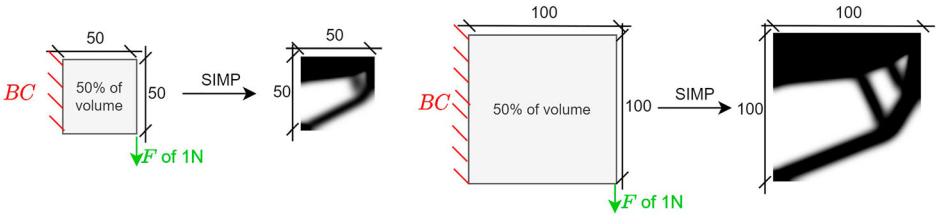
On the other hand, the second part opted to eliminate FE from TO using a Prinicipal Component Analysis (PCA) followed by a shallow NN (Ulu, R. Zhang, and Kara 2016) or directly using deep NNs (Abueidda, Koric, and Sobh 2020; Rade *et al.* 2020) and Generative Adversarial Network (GAN) frameworks (Behzadi and Ilies 2021; Yu *et al.* 2019; Sharpe and Seepersad 2019; Rawat and Shen 2019; Malvia 2020).

Nevertheless, none of these DL-based methods included any layout-geometry-related constraints except that of Chandrasekhar and Suresh (2021). On the contrary, they were left to the reinterpretation phase, knowing that significant modifications might be made in the shape during this phase. In other words, the TO phase was accelerated, and not the whole Design for AM (DfAM) process.

This work's main contribution is not only accelerating the TO phase but also identifying a way to tailor the design's layout to accelerate the design phase in the DfAM process. Thus, the authors benefit from DL's ability to learn spatial correlations to elaborate a DL approach that simultaneously integrates mechanical and layout conditions at the conceptual level and generates 2D designs validating the input conditions.

The layout is the material arrangement in a design space, which induces a modification in the geometry when changed, in other words, a relaxed geometry condition. The layout condition considered in this work is the total number of components in a 2D design ($Nbr_{\text{bars}}$). As a matter of fact, the training dataset consists of 2D truss-like structures, each being a collection of three types of beam (Figure 5). The outer shape of a structure is delimited by the clamped and loaded bars, which makes the inner bars a target variable that an engineer can alter, simplify or complexify so as to make the design the design comply with the input constraint. Furthermore, the $Nbr_{\text{bars}}$ is an example of layout constraints that are scarcely controllable using FE-TO where the design's geometry is unknown beforehand; hence creating an analytical formula to extract and count beams is hardly feasible.

The DL-layout-driven solution (DL-TO) paves the way to handling and tailoring different shapes during design generation. DL-TO always suggests designs, and even when the proposition is suboptimal, it still offers the engineer a good starting point.




(a) A mesh size of 50 × 50. (b) A mesh size of 100 × 100.

**Figure 1.** An example of the mesh-dependency setback of TO. SIMP outputs two different layouts for two different mesh sizes for the same input mechanical conditions.

This work improves on the work of Almasri *et al.* (2021), in which the authors trained DL-TO via a dual-discriminator GAN and reported that the generated designs showed poorer mechanical performance (higher compliance values) than their FE based counterparts. Therefore, a third discriminator, a DL-based compliance predictor, is added to the GAN framework penalizing the generator over the compliance during the training.

It is worth mentioning that conditional GANs have recently captured the interest of researchers. Its applications are spread amongst all fields, in computer vision (image synthesis, image-to-image, text-to-image) (Z. Wang, She, and Ward 2021), video generation (Aldausari *et al.* 2022), medical (Jeong *et al.* 2022), agriculture (Bird *et al.* 2022), cybersecurity (Yinka-Banjo and Ugot 2020) and engineering (Kusiak 2020), particularly inverse design applications (Achour *et al.* 2020; Yilmaz and German 2020; Nobari, Chen, and Ahmed 2021; Jing *et al.* 2022).

The use of the GAN training framework has the advantage over other generative approaches in its flexibility. Additional knowledge is incorporated into the generator via new discriminators appended during training; here, the compliance predictor. In the future, thermal distortion and buckling predictors can be added as discriminators to generate designs accounting for these constraints. Additionally, these discriminators can also be used separately (to predict the design's thermal distortion or buckling).

This article's significant contributions can be summarized as follows.

(1) The integration of mechanical and non-mechanical layout constraints simultaneously at the conceptual level via a data-driven TO.
(2) The ability to tailor a design's shape easily by a simple change of the input layout condition ($Nbr_{\text{bars}}$ in this case), a challenging task to formulate analytically.
(3) The creation of a DL compliance predictor.

The rest of the article is organized as follows: Section 2 provides a theoretical overview of TO and GANs, respectively, formulates the problem and outlines the solution. Section 3 details the consolidation of the training and test datasets used to train and evaluate DL-TO. Section 4 depicts the counter-discriminator and compliance-predictor's performance and evaluates the generated designs. Finally, Section 5 summarizes the methodology and its outcomes, and presents future perspectives.

## 2. Materials

### 2.1. Topology optimization

Structural Optimization (SO) is defined as the process of finding the optimal material distribution within a physical volume domain to support the applied loading conditions and other constraints, *e.g.* increasing stiffness, reducing stress, reducing displacement, *etc.* There are three major SO approaches: (i) size; (ii) shape; and (iii) TO.

TO is the most general form of SO. It simultaneously addresses topology, shape and sizing problems, and ensures efficient material use.

The increase in raw material cost, the environmental challenges (like energy consumption), the rise of computational power and the development of advanced programming methods have made TO a prevalent discipline in industrial design (Querin *et al.* 2017; Sigmund and Maute 2013). The topmost common commercial approach is Solid Isotropic Material with Penalization (SIMP), also called the power-law approach (Bendsøe 1989; Sigmund and Petersson 1998). SIMP is a density-gradient-based iterative method that uses the penalization scheme of the intermediate non-binary values of density material $x$ to converge to an optimal binary design. SIMP represents a design as a distribution of discretized square material elements $e$ (material properties are assumed constant within each $e$). They are modelled as the relative material density raised to some power times the material properties of the solid material. In this work, the objective function to minimize is the energy of deformation or compliance $C(x)$ in Joules ($J$):

$$\min_{x} \quad := U^{\mathrm{T}}KU = \sum_{e=1}^{N} x_e^p u_e^{\mathrm{T}} k_e u_e \tag{1}$$

subject to :

$$\begin{aligned} &\frac{V(x)}{V0} \leq f \\ &KU = F \\ &0(\text{absence of material}) < x_{\min} \leq x \leq 1 \text{ (presence of material)}, \end{aligned} \tag{2}$$

where $U$ and $u_e$ are the global and element-wise displacements, $F$ the forces vector, $K$ and $k_e$ are the global and element-wise stiffness matrices, $N$ the number of elements $e$, $x_{\min}$ the minimum density $x$ (non-zero to avoid singularity), and $p$ the penalization power (typically 3). $V_0$ and $V(x)$ are the design domain and material volumes, respectively, and $f$ the volume fraction.

To solve the problem stated above efficiently, Sigmund and Petersson adopted the optimality criterion approach. A mesh-independency filter is applied to the element sensitivities to avoid checkerboard patterns.

In this study, a modified version of the numerical solution of SIMP written by Sigmund (2001) is used to generate a sufficient database of 2D designs.

### 2.2. Problem definition and solution outline

Despite the previously listed advantages of TO, it suffers from one principal drawback. The general shape is usually identified in the very early iterations, making it difficult for TO to modify it to account for other changes imposed by other constraints; hence, the output shape ends up a local minimum (Allaire, Dapogny, and Jouve 2021). Furthermore, for density-based TO, geometry cannot be defined clearly as it depends on the mesh size (Figure 1). Hence, any layout-related constraint controlling the design's geometry cannot be incorporated easily within TO's formulation. Last but not least, TO suffers from a convergence problem despite the penalization scheme; the grey regions in the output designs are hard to interpret physically (Figure 2).

To compensate for these limitations, the authors propose a data-driven approach to explore the DL capability, particularly Convolutional NN (CNN), to integrate a first layout constraint, the $Nbr_{\text{bars}}$, concurrently at the same level as the mechanical constraints. CNNs have demonstrated their potential in learning spatial correlations, and DL outputs converged to crisp black-and-white designs if trained on converged training designs.

Since the problem is to generate designs, a data-driven generative method, in particular GAN, is chosen for its flexible framework in incorporating additional knowledge into the generator.

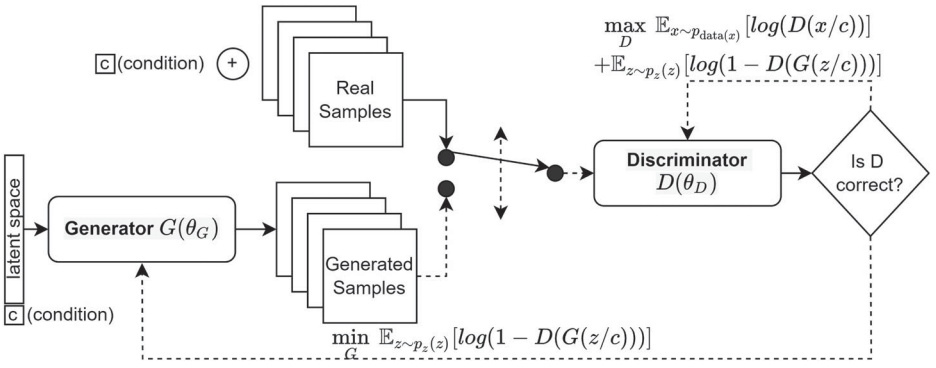



**Figure 2.** Diagram of a Conditional GAN.

As mentioned previously, this work improves on the work of Almasri *et al.* 2021 and further explores and interprets the results.

## 2.3. Convolutional conditional GAN (convolutional cGAN)

The approach adopted is based on the convolutional conditional GAN framework detailed in the following subsections.

### 2.3.1. Generative adversarial networks

GAN is a method that learns to mimic any input data distribution. Its advantage is that it can generate new samples following the same input data distribution. It consists of two neural networks, each working against the other: the generator $G(z, \theta_g)$, with $\theta_g$ its parameters and $z$ a latent random vector following a noise prior distribution $p_z$; and the discriminator $D(x, \theta_d)$, with $\theta_d$ its parameters and $x$ a real sample. $G(z, \theta_g)$ tends to output samples with a distribution $p_g$ similar to the real data distribution $p_{\text{data}}$. However, $D(x, \theta_d)$ tries to discriminate real samples (*i.e.* from $p_{\text{data}}$) from synthesized ones (*i.e.* from $G(z, \theta_g)$). Both networks are trained in a minimax framework to improve the same loss function: the cross-entropy loss $L(G, D)$. This loss, as defined in the original article (Goodfellow *et al.* 2014) is

$$L(G,D) = \min_G \max_D \mathbb{E}_{x\sim p_{\text{data}(x)}}[\log(D(x,\theta_d))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z,\theta_g)))]. \quad (3)$$

The solution of this function is $p_g = p_{\text{data}}$, *i.e.* when the generator starts to output data samples following the same distribution as the real ones.

### 2.3.2. Conditional Generative Adversarial Networks (cGANs)

On the other hand, a cGAN (Mirza and Osindero 2014) is a GAN's extension enabling the generation to be oriented by a specific input condition $c$. In this framework (Figure 3), the basics of cGAN become the conditional generator as $G(z|c, \theta_g)$, the conditional discriminator as $D(x|c, \theta_d)$ and the loss function as

$$L(G,D) = \min_G \max_D \mathbb{E}_{x\sim p_{\text{data}(x)}}[\log(D(x|c,\theta_d))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z|c,\theta_g)))]. \quad (4)$$

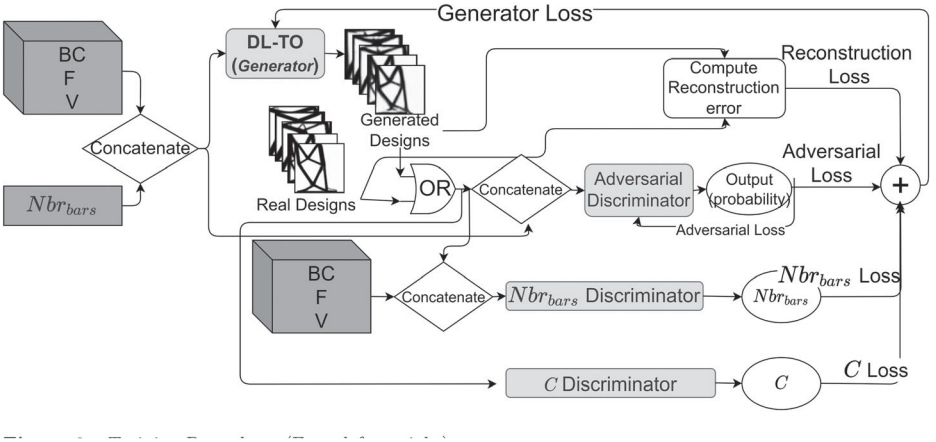



**Figure 3.** Training Procedure. (From left to right). 1) The mechanical constraints (Boundary conditions BC, loads configuration F and volume fraction V) are concatenated with the layout condition ($Nbr_{bars}$) and then input into the generator (DL-TO), which outputs the Generated Designs. The latter are concatenated with BC, F, V and $Nbr_{bars}$ and fed to the traditional discriminator, which predicts a score (the adversarial loss). Then they are concatenated with only the mechanical conditions and fed to the counter discriminator, which predicts their $Nbr_{bars}$. The counting loss is the MSE between the input and predicted $Nbr_{bars}$. Afterwards, the generated designs' compliance ($C$) is predicted by the $C$ discriminator. Finally, the generated design's quality is evaluated (the Reconstruction error). All four losses are summed (a weighted sum), and the final score is fed-back to DL-TO to update its weights. 2) The Traditional discriminator is trained at two levels: first with the real designs and the second with the generated designs. The adversarial loss is used to update its weights. 3) The counter and compliance discriminators are pre-trained ahead with the real designs and used as evaluators during the training.

### 2.3.3. Convolutional Neural Networks (CNNs)

CNNs differ from traditional NNs in the way neurons between successive hidden layers are connected. In a hidden layer, a neuron is only connected to a subset of neurons in the previous layer. This sparse connectivity allows CNNs to learn feature maps implicitly and reduce the network's complexity.

CNNs have proven robustness in learning spatial correlations in images (Aloysius and Geetha 2017), making a CNN-based architecture the adequate generator's architecture.

To sum up, a convolutional cGAN is a cGAN with $G(z|c,\theta_g)$'s and $D(x|c,\theta_d)$'s architectures being based on CNNs.

## 2.4. The methodology

This work aims to create a DL-TO method that simultaneously integrates mechanical and layout constraints. In other words, the training must ensure that DL-TO's network is penalized on both input constraints. The input constraints are formulated as images (Figure 4). Therefore, a conditional convolutional triple-discriminator GAN (Mirza and Osindero 2014) is chosen to train DL-TO. Before the training, the counter and compliance predictors are pre-trained with real data. These discriminators are only used as evaluators during the training of DL-TO. The training procedure is outlined in Figure 3.

### 2.4.1. Architecture of the generator (DL-TO)

The generator (DL-TO) is a deep Res-U-net network (Z. Zhang, Liu, and Wang 2018). It is an encoder–decoder CNN-based architecture with residual (Res) and skip-connections (U-net). DL-TO's encoder encodes the six input conditions formulated as a six-channel-image (the boundary conditions and loads along the $x$- and $y$-axes, the volume fraction and the $Nbr_{\text{bars}}$) and decodes the 2D design (Figure 4). The architecture is justified and detailed in the online supplemental data, which can be accessed at https://doi.org/10.1080/0305215X.2022.2144847.

### 2.4.2. Architecture of discriminators

The first one, the traditional discriminator, takes the design, along with all its conditions, and outputs the probability that it comes from the real data distribution to ensure the generator learns it. The second one, the DL-counter, takes the design, and only its mechanical conditions, and outputs its $Nbr_{\text{bars}}$ to ensure the generator respects the input layout condition. Finally, the DL-compliance predictor takes the design and predicts its compliance to boost the generated designs' mechanical performance.

***2.4.2.1. Architecture of the traditional discriminator.*** The traditional discriminator's network consists of DL-TO's encoder followed by a dropout, then a fully connected layer. It outputs a probability $p$ regarding the design being real ($p \approx 1$) or fake ($p \approx 0$). It helps DL-TO learn the mapping from constraints to designs (quality) and capture various constraints (diversity).

***2.4.2.2. Architecture of the counter discriminator.*** The counter architecture is detailed in the online supplemental data.

In this work, the counter is used at three levels: (1) to augment the training dataset; (2) to train; and (3) to evaluate DL-TO.

It is pre-trained on manually labelled SIMP-designs (4347 samples) before the GAN's full training; the $Nbr_{\text{bars}}$ present in each design are counted manually (Figure 5).

This pre-trained counter is used to predict the $Nbr_{\text{bars}}$ on unlabelled training designs (to augment the labelled training dataset). This pre-training hack improves DL-TO's convergence.

At each training iteration, this discriminator is used to predict the generated designs' $Nbr_{\text{bars}}$ and penalize DL-TO.

At inference time, it is used to predict the $Nbr_{\text{bars}}$ on generated designs to evaluate DL-TO's conformity to the layout constraint (Section 4).

The counter's performance is detailed in Section 4.1.

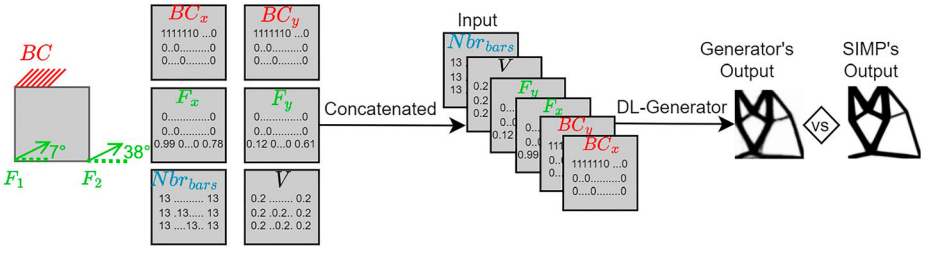



**Figure 4.** Input of the DL-TO. The boundary conditions along the *x*- and *y*-axes ($BC_x$, $BC_y$), load configurations ($F_x$, $F_y$), volume fraction V and layout constraint $Nbr_{\text{bars}}$ are formulated as a six-channel image.

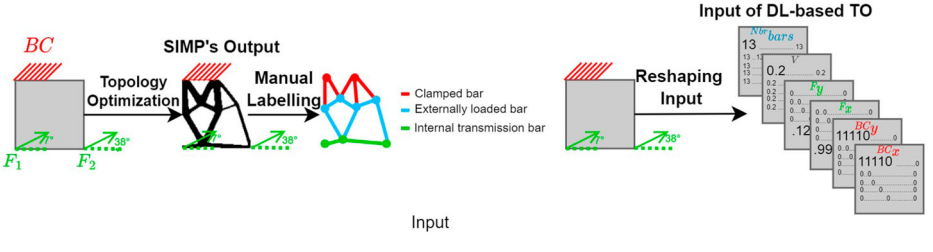



**Figure 5.** Types of bars in a design. The design shown here is clamped from the upper-left edge and loaded with two punctual external forces located in the bottom corners and tilted by 7 and 38 degrees. Indeed, It has 5 clamped (red) bars, 2 externally loaded (green) bars outgoing from two nodes corresponding to the forces locations and 6 internal-transmission (blue) bars, thus, a total of 13 bars.

*2.4.2.3. Architecture of the compliance discriminator.* The compliance predictor's architecture is detailed in the supplemental data. It takes an image-like 2D design as input and outputs an estimation of the compliance value. It is pre-trained on real 2D designs (output by SIMP). Its performance is detailed in Section 4.2. It is interesting to highlight that an FE-based compliance evaluator could conduct the training instead of a DL-based one. However, this is disadvantageous in terms of training time. An FE-compliance computation is thousands of times slower than its DL-counterpart (Section 4.2).

### 2.4.3. Loss function

This triple-discriminator GAN strives to train DL-TO to generate 2D designs of good quality and compliant with the input mechanical and layout conditions. Thus, the original adversarial loss function ($L_{\text{adv}}$) used to train the generator (Equation 4) is adjusted by adding the reconstruction ($L_r$), counting ($L_{\text{count}}$) and compliance ($L_C$) losses. The modified generator loss function $L_G$ adapted in training is the following:

$$L_G = \lambda_1 L_r + \lambda_2 L_{\text{adv}} + \lambda_3 Acc_{\text{count}} L_{\text{count}} + \lambda_4 L_C, \tag{5}$$

where $L_r = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$, $L_{\text{count}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$, $L_C = \frac{1}{n}\sum_{i=1}^{n}|C_i - \hat{C}_i|$ with $x_i$, $\hat{x}_i$ being the true and predicted 2D design, $y_i$ the input $Nbr_{\text{bars}}$, $\hat{y}_i$ the predicted $Nbr_{\text{bars}}$ of the generated design, $C_i$ the compliance of the real design, $\hat{C}_i$ the compliance predicted over the generated one and $n$ the batch size. The accuracy of the counter discriminator $Acc_{\text{count}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{t}_i == y_i)$ with $y_i$, $\hat{t}_i$ the true and predicted $Nbr_{\text{bars}}$ in the real designs and $N$ the total number of training samples.

The adversarial loss encourages creativity and diversity in the generated 2D designs. The reconstruction loss boosts their aesthetics and conformity with the volume fraction constraint; the design's volume fraction is the average of the pixel values. The counting loss assures the generator respects the input layout constraint. Finally, the compliance loss compensates for the weak mechanical performance exhibited in the work of Almasri *et al.* (2021).

It is worth noting that the relative error is used to compare the generated designs' compliance versus the SIMP's in the work of Almasri *et al.* (2021), which is the $L1$-norm between the compliances of the DL and SIMP designs divided by the SIMP designs' compliance. Thus, using the $L1$-norm in the loss function penalizing DL-TO's training is the best choice, especially since the objective is to generate designs with lower compliance. However, the $L2$-norm was also tested. With $L2$, the DL designs always showed higher compliance than their SIMP counterparts. Consequently, the $L1$-norm is chosen for the $L_C$.

The stable-GAN-training challenges and hacks and regularization terms ($\lambda_i; i \in [1, 4]$) used are justified in the supplemental data.

## 3. Data generation

To train the model, 21,538 2D designs were generated following the SIMP method (Section 2.1) via a Python version of the academic open-source TO code written by Sigmund (2001).[1] The layout constraint ($Nbr_{\text{bars}}$) is added by manual labelling over 4347 samples (which are used to pre-train the counter discriminator), and then the $Nbr_{\text{bars}}$ of the remaining samples are predicted using the counter discriminator.

A 2D design of size $n_x \times n_y$ is discretized into a mesh of $(n_x + 1) \times (n_y + 1)$ nodes and is subject to two major constraints: the boundary conditions (BC), *i.e.* the clamped nodes, and the load configurations (F), *i.e.* the loaded nodes (Figure 5). Thus, input conditions ($BC$, $F$, $V$ and $Nbr_{\text{bars}}$) are concatenated as a six-channel image *i.e.* of dimension $6 \times n_x \times n_y$ (Figure 4).

The dataset was separated into a training (80%) and a test (20%). The test set is used to evaluate the generator's performance.

The data generation strategy is further detailed in the supplemental data.

## 4. Results and discussion

An optimal design is a trade-off between mechanical performance and layout constraints. Thus, this section conducts a quantitative mechanical evaluation of the generated designs' mechanical performance via the relative error of compliance and volume fraction and a layout one to validate the designs' conformity with the input layout constraint $Nbr_{\text{bars}}$.

Most state-of-the-art GANs evaluations are subjective, based on the creativity and aesthetics of the generated samples. In this study, generated designs are assessed qualitatively and quantitatively.

The relative error $e_{X_{\%}} = (X_g - X_o)/X_o \times 100$ is the metric used for the volume fraction $V$ and compliance $C$, where $g$ and $o$ refer to generated and original, respectively, and $\{X \in \{\text{V}, C\}\}$, such that V is the mean of density-pixels and $C$ is computed via an FE formula. For the $Nbr_{\text{bars}}$, the metric is the bar-count difference $\Delta Nbr_{\text{bars}} = Nbr_{bars_g} - Nbr_{bars_o}$, such that $Nbr_{\text{bars}}$ is predicted via the counter discriminator.

### 4.1. Counter-discriminator's performance

As previously mentioned, the counter discriminator is pre-trained on manually labelled SIMP designs (4347 samples) before the GAN's full training.

The 2D designs in the dataset consist of structures with 3 to 31 components, which is a wide range for the $Nbr_{\text{bars}}$ variable. An admissible prediction falls within an interval $|\Delta Nbr_{\text{bars}}| \leq 2$. The counter's performance on the training (3885 designs) and test (462 designs) sets:

(1) on the training set: 99.8% of the predictions fall within $|\Delta Nbr_{\text{bars}}| \leq$ 1 bar;
(2) on the test set: 85.4% of the predictions fall within $|\Delta Nbr_{\text{bars}}| \leq$ 1 bar while 94.9% of them within $|\Delta Nbr_{\text{bars}}| \leq$ 2 bars.

In conclusion, the counter-discriminator is validated.

### 4.2. Compliance-predictor's performance

The DL compliance predictor's performance is evaluated by two metrics, the computation time (Figure 6(a)) and the prediction error $e_{\%}$, which is the relative error between the true and predicted compliance values on the real designs (Figure 6(b)). Eighty-seven percent of the test predictions fall into a 5% error margin, and 93% of them are made within a 10% error margin. Moreover, it computes the compliance of one design 16 times faster (on the CPU) and 56 times faster (on the GPU) than the FE-based calculator, and 400 (on the CPU) to 5000 (on the GPU) times faster for a batch of designs (Figure 6(a)). Thus, the DL compliance predictor's advantage over its FE counterpart is its prediction speed, especially in batch prediction, allowing faster GAN training within an acceptable level of precision.

### 4.3. DL-TO's performance

This section summarizes the overall performance of DL-TO and compares the results obtained versus those reported in the work of Almasri *et al.* (2021). The generated designs are thresholded to damp any intermediate density values.

Table 1 clearly shows that $C$ is improved by 21.3% with the integration of the compliance predictor as a third discriminator without any loss of generality. Moreover, this improvement increases remarkably, by 45.4%, if the error is restrained to 5% (15.4% versus 60%).

For a better visualization, Figure 7 plots the $e_{V_{\%}}$, $e_{C_{\%}}$ and $\Delta Nbr_{\text{bars}}$ distributions of the test set using DL-TO's previous and improved version. The improvement over $C$ is clear, the median $e_{C_{\%}}$

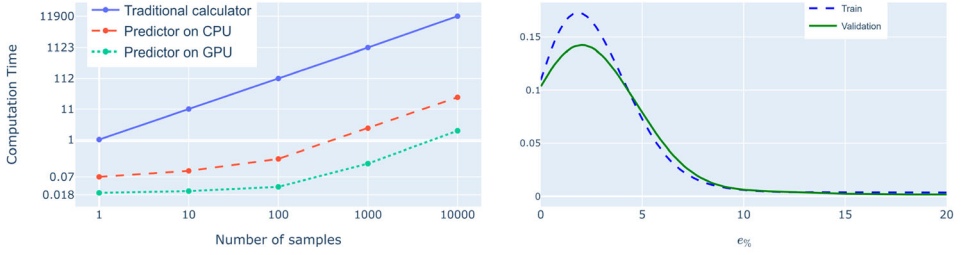



**Figure 6.** This figure shows the relative prediction error of the DL-compliance-predictor and compares its computational efficiency versus the traditional FE-based-compliance-calculator. (a) Computation time in seconds (*s*) and (b) Relative prediction error, $e_{\%}$.

**Table 1.** Comparison of DL-TOs trained without/with a compliance predictor as a third discriminator. MSE is the average mean squared error; $e_{V_{\%}} \leq 5\%$ is the percentage of DL-designs showing a volume fraction less than 1.05 of the SIMP-designs' one; $|e_{C_{\%}}| \leq 10\%$ and $|e_{C_{\%}}| \leq 5\%$ are the percentages of DL-designs within $\pm 1.1$; and $\pm 1.05$ the compliance achieved by SIMP, respectively. $|\Delta Nbr_{\text{bars}}| \leq 2$ is the percentage of DL-designs conforming with the $Nbr_{\text{bars}}$ constraint within an error margin of $\pm$ 2 bars. Adding a compliance predictor improves the generated designs' compliance by **21.3**% and **45.4**% for an error margin of 10% and 5%, respectively.

| | Aesthetics | Mechanical performance (%) | | | lLayout performance (%) |
|---|---|---|---|---|---|
| Training method | MSE | $e_{V_{\%}} \leq 5\%$ | $\lvert e_{C_{\%}}\rvert \leq 10\%$ | $\lvert e_{C_{\%}}\rvert \leq 5\%$ | $\lvert \Delta Nbr_{\text{bars}}\rvert \leq 2$ |
| Without compliance (Almasri *et al.*) | 0.063 | 97.7 | 52.4 | 15.4 | 82.4 |
| With compliance (the proposed approach) | 0.065 | 93.6 | **73.7** | **60** | 73.7 |

is 2.52% versus 9.3% from the work of Almasri *et al.* The distributions of $e_{V_{\%}}$ and $\Delta Nbr_{\text{bars}}$ remain similar.

In other words, the addition of the third discriminator enhanced the compliance without deterioration of the initial performance.

Finally, the computational performance of DL-TO and FE-TO (SIMP) are compared. DL-TO generates a design 3500 times faster on the CPU and 47,000 times faster on the GPU (see Table 2 in the supplemental data). Furthermore, DL-TO has the advantage over SIMP in terms of layout control at the conceptual level; it takes into consideration not only mechanical but also layout constraints, for which SIMP needs post-processing. This aspect is demonstrated in Section 4.4.

A sample of the designs generated by the DL-TO trained in this work compared to those generated by Almasri *et al.* is illustrated in Figures 8(a) and 8(b), respectively; a global threshold is applied over the generated designs. As expected, those generated by the new DL-TO version show lower deformation energy ($C$). It is essential to underline that applying a threshold is critical to a design's mechanical performance and its layout; the lower the threshold, the higher the V; the higher the threshold, the lower the $Nbr_{\text{bars}}$ with possibly the appearance of discontinuities in the design. Hence, the optimal approach is to privilege a local threshold.

To sum up, the overall performance of DL-TO is promising. It generates mechanically valid designs while respecting layout constraints, indistinguishable to the naked eye from those generated by SIMP while being thousands of times faster.

Thus, it offers the designer an alternative way to explore designs faster and easily adjust their layout (here defined by $Nbr_{\text{bars}}$). Tuning the mechanical and layout conditions is accomplished fast and effortlessly. Furthermore, the choice of GANs is justified. The addition of discriminators is advantageous, and when the right balance between their losses is found, transferring new knowledge to the generator is possible and is demonstrated here.

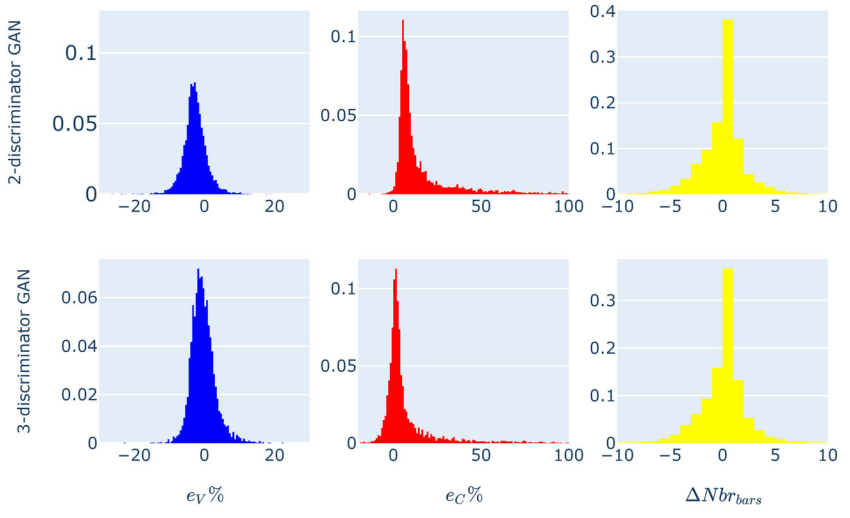



**Figure 7.** The distributions of the relative errors of volume fraction and compliance ($e_{V_\%}$ and $e_{C_\%}$) and the difference between the number of design components ($\Delta Nbr_{\text{bars}}$) computed over the test set for the DL-TO trained with two (Almasri *et al.* 2021) versus three discriminators.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Design with *Threshold*** | **Original Design** *(Threshold =0.35)* | | | | | | | | |
| | **Generated Design** *(Threshold =0.35)* | | | | | | | | |
| **Metrics With Threshold** | $e_{V_\%}$ | -5.93 | 10.2 | -2.63 | 5.31 | 4.2 | 3.63 | 0.76 | 7.7 |
| | $e_{C_\%}$ | **5** | **-6.1** | 0.96 | **-0.066** | -3.3 | -2.2 | **2.64** | **-1.73** |
| | $\Delta C_x$ | 0 | 0 | 0 | +2 | 0 | 0 | 0 | 0 |

(a)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Design with *Threshold*** | **Original Design** *(Threshold =0.4)* | | | | | | | | |
| | **Generated Design** *(Threshold =0.4)* | | | | | | | | |
| **Metrics With Threshold** | $e_{V_\%}$ | -1.36 | -0.16 | -3.72 | -7.16 | -1.34 | 3.45 | 0.17 | -3 |
| | $e_{C_\%}$ | **3.75** | **-2** | 2.83 | **31.75** | -2.53 | 0.48 | **4.31** | **21.1** |
| | $\Delta C_x$ | 0 | 0 | 0 | +1 | 0 | 0 | 0 | -1 |

(b)

**Figure 8.** Comparison between the generated designs from two DL-TO models. The designs in Figure 8(a) are generated by a DL-TO trained with an additional DL compliance predictor. The generated designs' mechanical performance (the compliance) is improved after integrating the DL compliance predictor as a third discriminator. (a) Generation by the DL-TO trained with an additional compliance predictor and (b) Generation by the DL-TO of Almasri *et al.* (2021).

### 4.4. Tailoring the design's layout via DL-TO

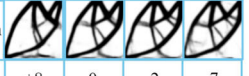
To validate DL-TO's understanding of layout, the mechanical conditions ($BC, F$ and V) were prepended, only $Nbr_{\text{bars}}$ is varied (Figure 9(a)). Indeed, the $Nbr_{\text{bars}}$ in the design increases with the input variable.

However, the supplementary bars are blurry, and DL-TO struggles to conform with the input $Nbr_{\text{bars}}$, especially for the lower and upper extreme values (7 and 30).

Moreover, V and $Nbr_{\text{bars}}$ of a design are correlated. Thus, a better approach would be to vary them together (Figure 9(b)).

Modifying V and $Nbr_{\text{bars}}$ simultaneously improves DL-TO's conformity with the input layout and V constraints ($\Delta Nbr_{\text{bars}}$ and $e_{V_{\%}}$ decreased in Figure 9(b)), and most importantly demonstrates DL-TO's creativity in adding/removing internal bars to/from the design. Nevertheless, while $C$ improved for $Nbr_{\text{bars}} = 20$ and 30, it exploded for $Nbr_{\text{bars}} = 7$. This phenomenon implies a minimum $Nbr_{\text{bars}}$ for every set of mechanical conditions to guarantee the design's integrity and resilience.

To identify this minimum value, a set of mechanical constraints are fixed, and $Nbr_{\text{bars}}$ is varied between 3 and 30 bars. Then, the designs are generated and evaluated (Figure 10). $C$ and $Nbr_{\text{bars}}$ are inversely correlated, and the minimum admissible $Nbr_{\text{bars}}$ of the considered example is 15, after which $C$ does not show any remarkable improvement. This information implies that the designer can choose any design with 15 to 30 components and guarantee that its mechanical performance will not deteriorate, knowing that this process takes a fraction of seconds, thanks to DL.

| Input | Fixed BC, F and $V = 0.33$ | | | | |
|---|---|---|---|---|---|
| | $^{Nbr}bars$ | 7 | 15 | 20 | 30 |
| Output | Generated Design |  | | | |
| | $\Delta Nbr_{bars}$ | +8 | 0 | -2 | -7 |
| | $e_{V_{\%}}$ | -1.01 | 2.06 | 1.36 | -1.64 |
| | $C$ in $J$ | 237.3 | 153.7 | 153 | 169.11 |

(a)

| Input | Fixed BC, F and variable $V = 0.33 + \delta v$ | | | | |
|---|---|---|---|---|---|
| | $^{Nbr}bars$ | 7 | 15 | 20 | 30 |
| | $\delta v$ | -0.25 | 0 | 0.05 | 0.1 |
| Output | Generated Design | | | | |
| | $\Delta Nbr_{bars}$ | +1 | 0 | -1 | 0 |
| | $e_{V_{\%}}$ | -6.7 | 2.06 | -0.055 | -1.9 |
| | $C$ in $J$ | **8.55e6** | 153.7 | 128.3 | 120.7 |

(b)

**Figure 9.** Tailoring the layout of a design: (a) the mechanical constraints (BC, F and V) are fixed and the $Nbr_{\text{bars}}$ condition is variable; (b) BC and F are fixed while $Nbr_{\text{bars}}$ and V are variable. (a) Varying $Nbr_{\text{bars}}$ and (b) Varying $Nbr_{\text{bars}}$ and V proportionally.

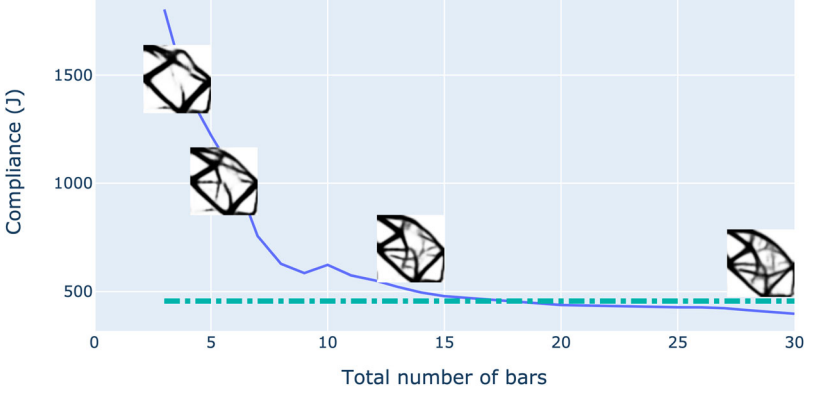



**Figure 10.** Compliance versus $Nbr_{\text{bars}}$ variation for a set of fixed mechanical constraints. The maximum admissible compliance is 450 J (the dashed line). The minimum admissible $Nbr_{\text{bars}}$ here is 15 bars, and the lowest compliance is achieved at $Nbr_{\text{bars}} = 30$.

### 4.5. Generating designs with a new unseen constraint

Some spatial constraints can be enforced on the design area's boundaries. These boundaries can be defined by the V matrix.

This section examines DL-TO's potential to propose new designs accounting for this constraint. It should be noted that it was trained using only input V without any design area boundaries.

Since the input V is formulated as a matrix of $(n_x + 1) \times (n_y + 1)$ elements, the existence/absence of material in particular locations of the design space is obtained by increasing/decreasing the values in these locations.

This formulation allows many layout constraints to be integrated into the design, like passive elements (*e.g.* a hole for a pipe) or active elements (*e.g.* a filled shape for an external pillar). It also allows the addition of boundary and load conditions on these elements by modifying the BC and F matrices. This aspect is not explored in this research.

Figure 11 illustrates a sample of constraints without/with different design area's boundaries and the corresponding generated threshold designs. In all three cases, DL-TO filled/emptied the locations with/from material where extra/no material is forced; it reshaped the internal truss-like shape creatively to maintain the design's integrity while preserving the original design's outer shape. In other words, it respected the input BC and F configurations. As for compliance, it decreased in 4 out of 15 cases, its increase was modest ($e_{C_{\%}} \leq 10\%$) in 5 cases, and its increase was more than 10% in the remaining 6 cases (Figures 11 and 12).

Finally, this experiment was replicated using SIMP (Figure 12). As can be seen clearly, SIMP does not always comply with the constraint. In Figure 12(a), the additional material is on the left side of the domain space; SIMP responded with thicker-bars designs instead of adding extra material only to the left side; in other words, the volume fraction constraint is considered globally and not locally. It is worth mentioning that these designs, while not conforming with the constraint, benefited from lower compliance. In Figure 12(b) where a hole was enforced, SIMP failed to converge to a solution.

To sum up, DL-TO shows an encouraging result in creatively conforming to layout constraints, and its convergence is not easily compromised, especially when trained on converged designs.

To the best of the authors' knowledge, DL-TO is the first strategy that allows for the natural handling of active and passive elements while not explicitly being trained on such samples, demonstrating its generalization capability.

### 4.6. Discussion

Data-driven TO requires three challenging and time-consuming steps: data preparation, model architecture and training. Thus, the acceleration advantage of DL-TO over FE-SIMP can seem moderate in the short term but will definitely pay off in the long run.

This article addresses the problem of handling spatial correlations via a cGAN architecture. cGAN allows the incorporation of a layout constraint, $Nbr_{\text{bars}}$, into the mechanical constraints without the need for explicit formulas for bars. It is essential to highlight that this work does not handle any manufacturing constraint yet, only a simple geometry-related constraint. Further work will try to generalize over more concrete manufacturing constraints. The $Nbr_{\text{bars}}$ condition is not a major manufacturing constraint. However, it is used to validate the approach, which is that DL can compensate for the difficulties faced by TO when dealing with mechanical and layout-related constraints concurrently.

Arguably, the level-set approach resolves the principal drawback of density TO mentioned in Section 2.2. Nevertheless, density-TO, SIMP mainly, is the most implemented in industrial-commercial software, which justifies developing a DL approach trained on SIMP designs. Indeed, the asset of DL-TO is that, unlike FE-TO, the initial guess can be easily tailored to account for a layout constraint. Moreover, it increases the domain space of potential designs, which can be leveraged

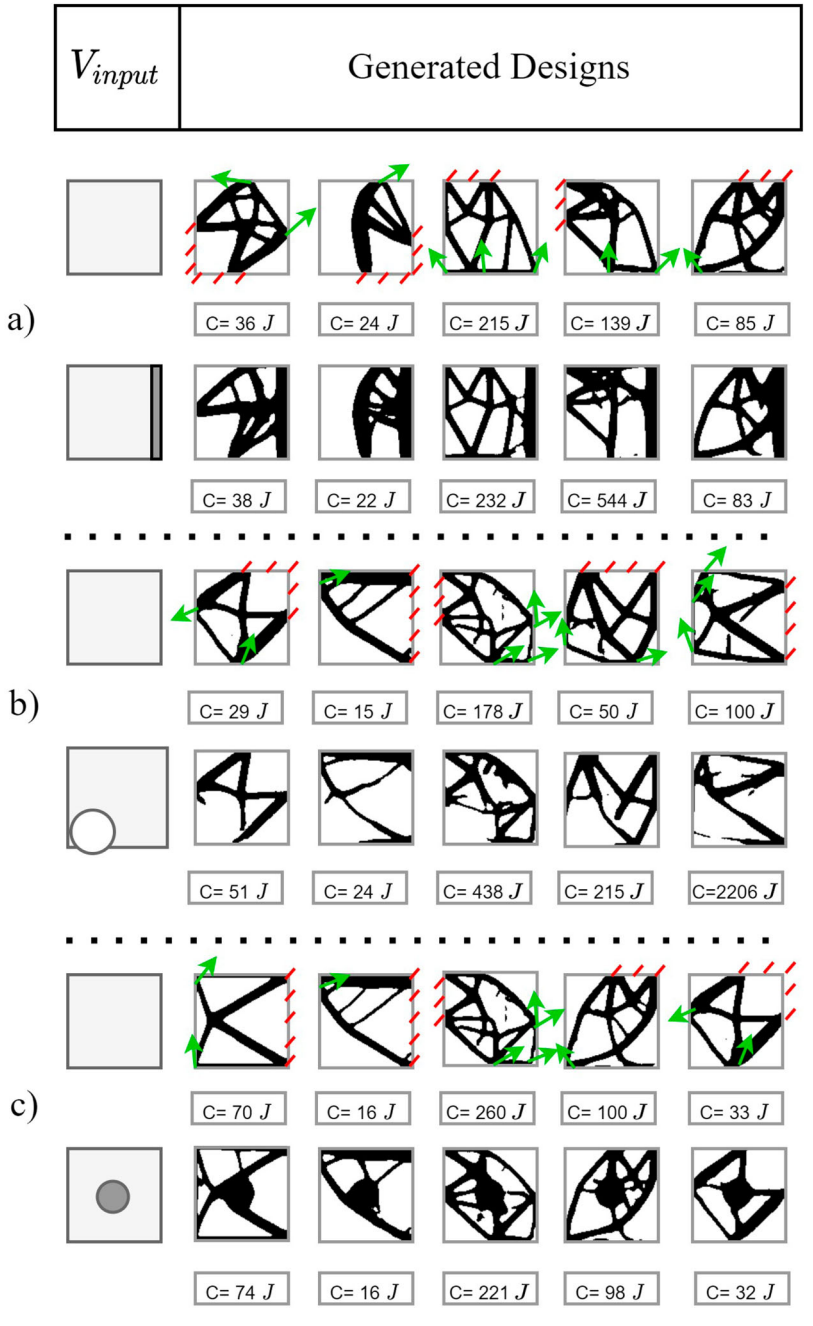



**Figure 11.** Sample of DL-TO designs generated without versus with constraints on the design area's boundary as input ($V_{input}$). All designs are threshold: (a) additional material is added on the right edge; (b) a hole (passive element) of radius $r = 30$ is enforced at the bottom left of the design space; (c) an active element of radius $r = 15$ is enforced at the centre of the design space. The BC (dashes in red) and F (arrows in green) conditions are only shown on the first row. The computed compliance *C* in Joules (*J*) is shown under the design.

by a shape/size optimization afterwards and provides creativity and rapidity to the DfAM. Finally, the FE-TO mesh-dependency problem can be overcome by a pre-trained DL-based super-resolution model.

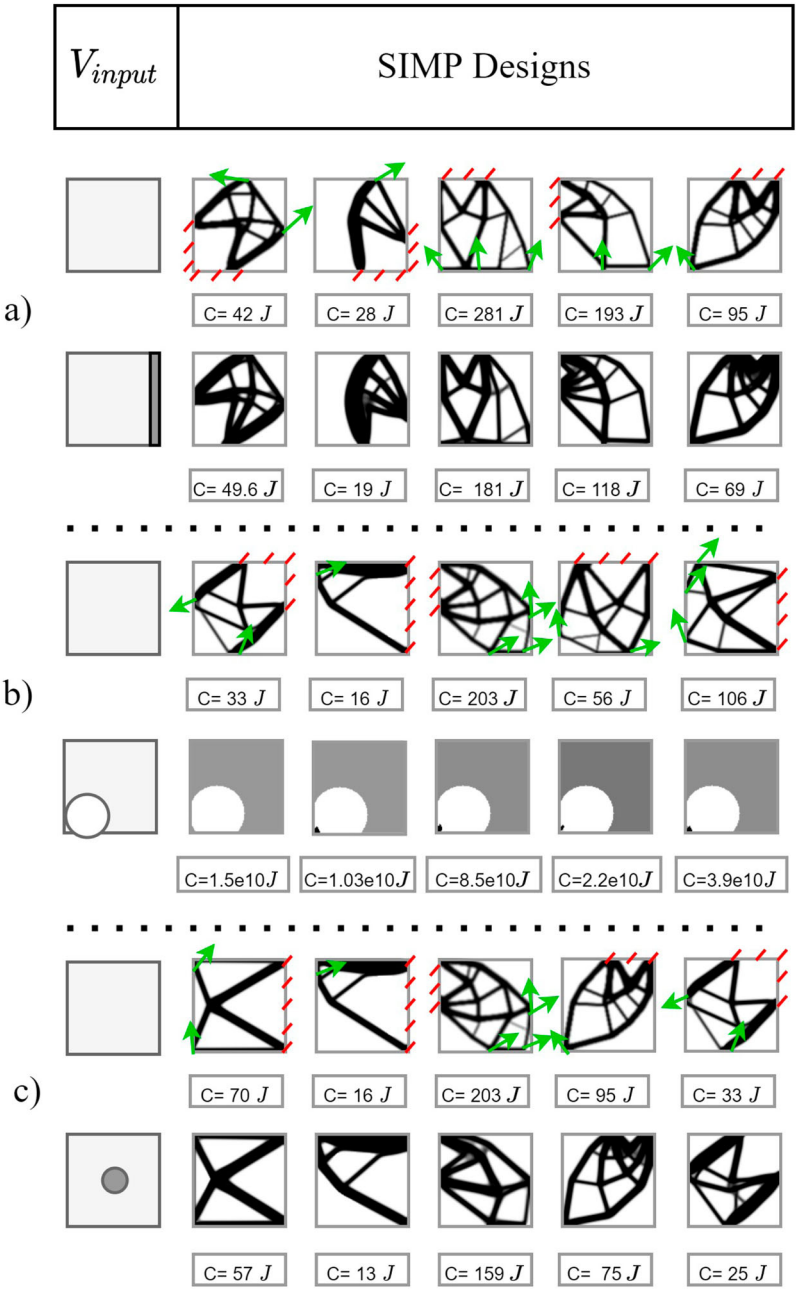



**Figure 12.** Sample of SIMP designs generated without versus with constraints on the design area's boundary as input ($V_{input}$). All designs are threshold: (a) additional material is added on the right edge; (b) a hole (passive element) of radius $r = 30$ is enforced at the bottom left of the design space; and (c) an active element of radius $r = 15$ is enforced at the centre of the design space. The BC (dashes in red) and F (arrows in green) conditions are only shown on the first row. The computed compliance *C* in Joules (*J*) is shown under the design.

Moreover, SIMP is chosen to create the dataset in this work. SIMP is the simplest and most easily implemented TO algorithm. Thus, the designs driving the training are created to validate the proposed methodology. Indeed, any new data coming from other TO algorithms can be used to train the proposed model.

Finally, it is essential to highlight that the layout constraint used in this work ($Nbr_{\text{bars}}$) might not be the most demonstrative example because $Nbr_{\text{bars}}$ is more precisely controllable by the primitives/supershapes methods (Smith and Norato 2021, 2019; Norato 2018). However, it is proof that a layout constraint requiring seeing the design as an image, typically an aesthetic constraint (filling the design space) can be integrated via DL.

## 5. Conclusion

This article improves the DL-TO model proposed in Almasri *et al.* (2021). DL-TO generates topologically optimized designs, similar to those generated by SIMP and a thousand times faster. Moreover, it tailors the design's layout by increasing/decreasing the input number of components. A structure can easily be refined towards a simpler or more sophisticated structure, as the first step towards accelerating the design phase in the DfAM process. Furthermore, DL-TO generates creative designs with previously unseen input conditions, in particular the boundary of the design area (passive and active elements).

This work shows that DL architectures can learn the correlation between mechanical and non-mechanical constraints, such as $Nbr_{\text{bars}}$, a layout-related constraint. Its main objective is not to replace robust TO methods but to compensate for their difficulties in accounting for layout-related constraints. With DL-TO, the mechanics and layout-related are managed concurrently at the conceptual level and not sequentially, increasing the potential designs' space and accelerating the DfAM. Moreover, DL-TO's training strategy is flexible. The adaptability and effectiveness of upgrading the training framework from two to three discriminators to incorporate additional knowledge (here, the compliance constraint) to DL-TO without compromising its initial performance are demonstrated.

It is worth noting that the current method does not alleviate the engineering work necessary but does accelerate the search for the optimal print-ready design.

In the future, this strategy paves the way to facilitate the integration of more complex geometrical/manufacturing constraints via new discriminators (geometry-evaluators) appended in the GAN framework (to account for overhangs, maximum/minimum length/thickness, bar curvatures, *etc.*). Another future perspective is to adapt this approach to 3D designs. Finally, it is interesting to mention that DL-assisted TO algorithms will be good candidates as lighter and faster modules to be implemented in CAD software.

1. This code is available on the GitHub repository: https://github.com/dbetteb/TOP_OPT.git.

## References


Abueidda, Diab W., Seid Koric, and Nahil A. Sobh. 2020. "Topology Optimization of 2D Structures with Nonlinearities Using Deep Learning." *Computers & Structures* 237: Article ID 106283.

Achour, Gabriel, Woong Je Sung, Olivia J. Pinon-Fischer, and Dimitri N. Mavris. 2020. "Development of a Conditional Generative Adversarial Network for Airfoil Shape Optimization." In *Proceedings of the AIAA Scitech 2020 Forum. Session: Topology and Shape Optimization II*, Paper No. AIAA 2020-2261, 2261. Reston, VA: American Institute of Aeronautics and Astronautics. doi:10.2514/6.2020-2261.

Adam, Guido A. O., and Detmar Zimmer. 2014. "Design for Additive Manufacturing—Element Transitions and Aggregated Structures." *CIRP Journal of Manufacturing Science and Technology* 7 (1): 20–28.

Aldausari, Nuha, Arcot Sowmya, Nadine Marcus, and Gelareh Mohammadi. 2022. "Video Generative Adversarial Networks: A Review." *ACM Computing Surveys (CSUR)* 55 (2): 1–25.

Allaire, Grégoire, Charles Dapogny, and François Jouve. 2021. "Shape and Topology Optimization." In *Handbook of Numerical Analysis*, Vol. 22, 1–132. Amsterdam: Elsevier. doi:10.1016/bs.hna.2020.10.004.

Almasri, Waad, Dimitri Bettebghor, Fakhreddine Ababsa, Florence Danglade, and Faouzi Adjed. 2021. "Deep Learning Architecture for Topological Optimized Mechanical Design Generation with Complex Shape Criterion." In *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 222–234. Cham, Switzerland: Springer.

Aloysius, Neena, and M. Geetha. 2017. "A Review on Deep Convolutional Neural Networks." In *Proceedings of the 2017 International Conference on Communication and Signal Processing (ICCSP)*, 0588–0592. Piscataway, NJ: IEEE.

Behzadi, Mohammad Mahdi, and Horea T. Ilieş. 2021. "GANTL: Towards Practical and Real-Time Topology Optimization with Conditional GANs and Transfer Learning." *Journal of Mechanical Design*144 (2): Article ID 021711, 32pp. doi:10.1115/1.4052757.

Bendsøe, Martin P. 1989. "Optimal Shape Design As a Material Distribution Problem." *Structural Optimization* 1 (4): 193–202.

Bi, Minghao, Phuong Tran, and Yi Min Xie. 2020. "Topology Optimization of 3D Continuum Structures Under Geometric Self-Supporting Constraint." *Additive Manufacturing* 36: Article ID 101422.

Bi, Sirui, Jiaxin Zhang, and Guannan Zhang. 2020. "Scalable Deep-Learning-Accelerated Topology Optimization for Additively Manufactured Materials." arXiv:2011.14177.

Bird, Jordan J., Chloe M. Barnes, Luis J. Manso, Anikó Ekárt, and Diego R. Faria. 2022. "Fruit Quality and Defect Image Classification with Conditional Gan Data Augmentation." *Scientia Horticulturae* 293: Article ID 110684.

Chandrasekhar, Aaditya, and Krishnan Suresh. 2021. "Length Scale Control in Topology Optimization Using Fourier Enhanced Neural Networks." arXiv:2109.01861.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative Adversarial Nets." In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*, Vol. 2, 2672–2680. Cambridge, MA: MIT Press.

Halle, Alex, L. Flavio Campanile, and Alexander Hasse. 2020. "An AI-Assisted Design Method for Topology Optimization without Pre-Optimized Training Data." arXiv:2012.06384.

Han, Yong Sheng, Bin Xu, Lei Zhao, and Yi Min Xie. 2019. "Topology Optimization of Continuum Structures Under Hybrid Additive–Subtractive Manufacturing Constraints." *Structural and Multidisciplinary Optimization* 60 (6): 2571–2595.

Jeong, Jiwoong J., Amara Tariq, Tobiloba Adejumo, Hari Trivedi, Judy W. Gichoya, and Imon Banerjee. 2022. "Systematic Review of Generative Adversarial Networks (GANs) for Medical Image Classification and Segmentation." *Journal of Digital Imaging* 35: 137–152. doi:10.1007/s10278-021-00556-w.

Jing, W. A. N. G., L. I. Runze, H. E. Cheng, C. H. E. N. Haixin, Ran Cheng, Z. H. A. I. Chen, and Miao Zhang. 2022. "An Inverse Design Method for Supercritical Airfoil Based on Conditional Generative Models." *Chinese Journal of Aeronautics* 35 (3): 62–74.

Kallioras, Nikos Ath, Georgios Kazakis, and Nikos D. Lagaros. 2020. "Accelerated Topology Optimization by Means of Deep Learning." *Structural and Multidisciplinary Optimization* 62 (3): 1185–1212.

Kusiak, Andrew. 2020. "Convolutional and Generative Adversarial Neural Networks in Manufacturing." *International Journal of Production Research* 58 (5): 1594–1604.

Li, Shaoying, Shangqin Yuan, Jihong Zhu, Chuang Wang, Jiang Li, and Weihong Zhang. 2020. "Additive Manufacturing-Driven Design Optimization: Building Direction and Structural Topology." *Additive Manufacturing* 36: Article ID 101406.

Malviya, Manoj, Akash Agrawal, Pranjali Yadav, Yu-Hao Chang, and Christopher McComb. 2020. "A Systematic Study of Deep Generative Models for Rapid Topology Optimization." engrXiv (Engineering Archive). doi:10.31224/osf.io/9gvqs.

Matos, Marina A., Ana Maria A. C. Rocha, and Lino A. Costa. 2021. "Many-Objective Optimization of Build Part Orientation in Additive Manufacturing." *The International Journal of Advanced Manufacturing Technology* 112 (3): 747–762. doi:10.1007/s00170-020-06369-5.

Mirza, Mehdi, and Simon Osindero. 2014. "Conditional Generative Adversarial Nets." arXiv:1411.1784.

Nobari, Amin Heyrani, Wei Chen, and Faez Ahmed. 2021. "PcDGAN: A Continuous Conditional Diverse Generative Adversarial Network for Inverse Design." arXiv:2106.03620.

Norato, Julián A. 2018. "Topology Optimization with Supershapes." *Structural and Multidisciplinary Optimization* 58 (2): 415–434.

Querin, Osvaldo M., Mariano Victoria, Cristina Alonso, Rubén Ansola Loyola, and Pascual Martí Montrull. 2017. *Topology Design Methods for Structural Optimization*. Cambridge, MA: Academic Press. doi:10.1016/B978-0-08-100916-1.00008-8.

Rade, Jaydeep, Aditya Balu, Ethan Herron, Jay Pathak, Rishikesh Ranade, Soumik Sarkar, and Adarsh Krishnamurthy. 2020. "Physics-Consistent Deep Learning for Structural Topology Optimization." arXiv:2012.05359.

Ranjan, Rajit, Rutuja Samant, and Sam Anand. 2017. "Integration of Design for Manufacturing Methods with Topology Optimization in Additive Manufacturing." *Journal of Manufacturing Science and Engineering* 139 (6): Article ID 061007. doi:10.1115/1.4035216.

Rawat, Sharad, and M.-H. Herman Shen. 2019. "A Novel Topology Optimization Approach Using Conditional Deep Learning." arXiv:1901.04859.

Sharpe, Conner, and Carolyn Conner Seepersad. 2019. "Topology Design with Conditional Generative Adversarial Networks." In *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 59186, p. V02AT03A062. New York: American Society of Mechanical Engineers.

Sigmund, Ole. 2001. "A 99 Line Topology Optimization Code Written in Matlab." *Structural and Multidisciplinary Optimization* 21 (2): 120–127.

Sigmund, Ole, and Kurt Maute. 2013. "Topology Optimization Approaches." *Structural and Multidisciplinary Optimization* 48 (6): 1031–1055.

Sigmund, Ole, and Joakim Petersson. 1998. "Numerical Instabilities in Topology Optimization: A Survey on Procedures Dealing with Checkerboards, Mesh-Dependencies and Local Minima." *Structural Optimization* 16 (1): 68–75.

Smith, Hollis A., and Julián A. Norato. 2019. "Geometric Constraints for the Topology Optimization of Structures Made of Primitives." In *Proceedings of the SAMPE Conference*, Vol. 19. Charlotte, NC: SAMPE. doi:10.33599/NASAMPE/S.19.1518.

Smith, Hollis A., and Julián A. Norato. 2021. "Topology Optimization with Discrete Geometric Components Made of Composite Materials." *Computer Methods in Applied Mechanics and Engineering*376: Article ID 113582.

Sosnovik, Ivan, and Ivan Oseledets. 2019. "Neural Networks for Topology Optimization." *Russian Journal of Numerical Analysis and Mathematical Modelling* 34 (4): 215–223.

Subedi, Subodh C., Chaman Singh Verma, and Krishnan Suresh. 2020. "A Review of Methods for the Geometric Post-Processing of Topology Optimized Models." *Journal of Computing and Information Science in Engineering* 20 (6): Article ID 060801.

Ulu, Erva, Rusheng Zhang, and Levent Burak Kara. 2016. "A Data-Driven Investigation and Estimation of Optimal Topologies Under Variable Loading Configurations." *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 4 (2): 61–72.

Wang, Cunfu, and Xiaoping Qian. 2020. "Simultaneous Optimization of Build Orientation and Topology for Additive Manufacturing." *Additive Manufacturing* 34: Article ID 101246.

Wang, Zhengwei, Qi She, and Tomas E. Ward. 2021. "Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy." *ACM Computing Surveys (CSUR)* 54 (2): 1–38.

Wang, Chunpeng, Song Yao, Zhangjun Wang, and Jie Hu. 2021. "Deep Super-Resolution Neural Network for Structural Topology Optimization." *Engineering Optimization* 53 (12): 2108–2121.

Xu, Bin, Yongsheng Han, Lei Zhao, and Yi Min Xie. 2021. "Topological Optimization of Continuum Structures for Additive Manufacturing Considering Thin Feature and Support Structure Constraints." *Engineering Optimization* 53 (12): 2122–2143.

Yilmaz, Emre, and Brian German. 2020. "Conditional Generative Adversarial Network Framework for Airfoil Inverse Design." In *Proceedings of the AIAA Aviation 2020 Forum. Session: Physics Informed Machine Learning, MetaModeling, and Reduced Order Modeling II*, Paper No. AIAA 2020-3185, 3185. Reston, VA: American Institute of Aeronautics and Astronautics. doi:10.2514/6.2020-3185.

Yinka-Banjo, Chika, and Ogban-Asuquo Ugot. 2020. "A Review of Generative Adversarial Networks and Its Application in Cybersecurity." *Artificial Intelligence Review* 53 (3): 1721–1736.

Yu, Yonggyun, Taeil Hur, Jaeho Jung, and In Gwun Jang. 2019. "Deep Learning for Determining a Near-Optimal Topological Design Without Any Iteration." *Structural and Multidisciplinary Optimization* 59 (3): 787–799.

Zhang, Zhengxin, Qingjie Liu, and Yunhong Wang. 2018. "Road Extraction by Deep Residual U-Net." *IEEE Geoscience and Remote Sensing Letters* 15 (5): 749–753.

Zhang, Weihong, and Lu Zhou. 2018. "Topology Optimization of Self-Supporting Structures with Polygon Features for Additive Manufacturing." *Computer Methods in Applied Mechanics and Engineering* 334: 56–78.