

# Diffusion Models Beat GANs on Topology Optimization

François Mazé<sup>1</sup>, Faez Ahmed<sup>1</sup>

<sup>1</sup>Massachusetts Institute of Technology  
francois.maze@etu.minesparis.psl.eu, {fmaze,faez}@mit.edu

## Abstract

Structural topology optimization, which aims to find the optimal physical structure that maximizes mechanical performance, is vital in engineering design applications in aerospace, mechanical, and civil engineering. Generative adversarial networks (GANs) have recently emerged as a popular alternative to traditional iterative topology optimization methods. However, these models are often difficult to train, have limited generalizability, and due to their goal of mimicking optimal structures, neglect manufacturability and performance objectives like mechanical compliance. We propose TopoDiff — a conditional diffusion-model-based architecture to perform performance-aware and manufacturability-aware topology optimization that overcomes these issues. Our model introduces a surrogate model-based guidance strategy that actively favors structures with low compliance and good manufacturability. Our method significantly outperforms a state-of-art conditional GAN by reducing the average error on physical performance by a factor of eight and by producing eleven times fewer infeasible samples. By introducing diffusion models to topology optimization, we show that conditional diffusion models have the ability to outperform GANs in engineering design synthesis applications too. Our work also suggests a general framework for engineering optimization problems using diffusion models and external performance with constraint-aware guidance. We publicly share the data, code, and trained models here: <https://decode.mit.edu/projects/topodiff/>.

## 1 Introduction

Structural topology optimization (TO) of solid structures involves generating the optimal shape of a material by minimizing an objective function, for instance, mechanical compliance, within a given domain and under a given set of constraints (volume fraction, boundary conditions, and loads). TO is therefore becoming an essential design tool and is now included in most professional design software, such as Autodesk’s Fusion 360 and Solidworks. It is the driving force behind Autodesk’s generative design toolset, where designers input design goals into the software, along with parameters such as performance or spatial requirements, materials, manufacturing methods, and cost constraints. The software quickly generates design alternatives. Most methods to solve TO rely on gradient-based approaches, the most common method being the Solid Isotropic Material with Penalization method (Bendsøe and Kikuchi 1988; Rozvany, Zhou,

and Birker 1992). Despite their wide adoption, these traditional methods have two major pitfalls: their iterative nature makes them computationally expensive and they may generate non-optimal designs, for example, when penalization and filtering augmentations are used to avoid grayscale pixels in SIMP (Sigmund and Maute 2013).

Several deep learning methods have been developed in recent years to improve and speed up the TO process (Yu et al. 2018; Sharpe and Seepersad 2019; Nie et al. 2021; Behzadi and Ilieş 2021) by learning from large datasets of optimized structures. The latest and most promising results were obtained with deep generative models (DGMs) and notably with conditional generative adversarial networks (cGANs) trained for image synthesis, which take as input the boundary conditions and directly generate images of optimized structures. Although popular, most of these models optimize a loss function which does not align with the primary goals of topology optimization — getting high-performance and feasible structures. They often train for loss functions related to image reconstruction to achieve visual similarity and ignore modeling the physical performance of the generated structures. Most of them produce disconnected, floating material that seriously affects the manufacturability of the generated design. They also suffer from limited generalizability, especially for out-of-distribution boundary conditions.

We hypothesize that the absence of explicit methods to generate designs with low compliance and good manufacturability causes these issues. We further hypothesize that the reliance of the optimization objective on the sole cGAN prompts the model to only mimic pixel-wise the ground truth produced by traditional TO methods. As a result, two images with comparable pixel-wise similarity may still have significantly different performance values. The absence of explicit external guidance is even more problematic since the ground truth data is not guaranteed to be optimal, as explained above.

This paper introduces TopoDiff, a conditional diffusion-model-based method for TO. Dhariwal and Nichol (2021) have shown that diffusion models can outperform GANs for image generation, are easier to train, and are thus more readily adaptable to other tasks. We show that by introducing performance and constraints to diffusion models, they outperform GANs on topology optimization problems too. In addition, the sequential nature of diffusion models makes

them compatible with external guidance strategies that assist with performance and feasibility goals. By creating surrogate models to estimate performance, we thus introduce external guidance strategies to minimize mechanical compliance and improve manufacturability in diffusion models.

Our main contributions include proposing: (1) TopoDiff — a diffusion model based end-to-end Topology Optimization framework that achieves an eight-times reduction in average physical performance errors and an eleven-times reduction in infeasibility compared to a state-of-art conditional GAN, (2) a new guidance strategy for diffusion models to perform physical performance optimization with enhanced manufacturability constraint satisfaction, and (3) a generalized framework to solve inverse problems in engineering using diffusion models, when sample feasibility and performance are a high priority.

## 2 Background and Related Work

### 2.1 Topology Optimization

Topology Optimization (TO) finds an optimal subset of material  $\Omega_{opt}$  included in the full design domain  $\Omega$  under a set of displacement boundary conditions and loads applied on the nodes of the domain and a volume fraction condition. A structure is optimal when it minimizes an objective function, such as mechanical compliance, subject to constraints. Fig. 1 summarizes the principle of TO.

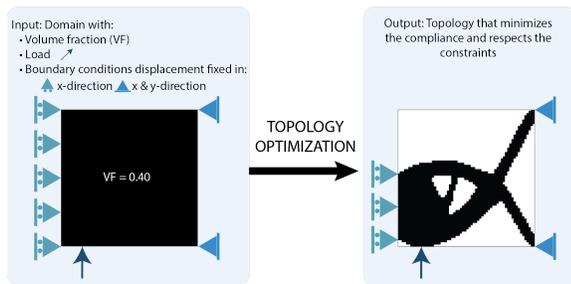


Figure 1: Topology Optimization aims to find the optimal structure that minimizes objectives such as compliance for a given set of load, boundary conditions, and volume fraction.

Traditional TO methods rely on Finite Elements Analysis (FEA) using gradient-based (Bendsøe and Kikuchi 1988) or gradient-free methods (Ahmed, Bhattacharya, and Deb 2013). One popular gradient-based method is the Solid Isotropic Material with Penalization (SIMP) method (Rozvany, Zhou, and Birker 1992).

SIMP associates every element of the mesh with a *continuous* density to perform gradient-based methods (Sigmund 2001). However, because intermediary densities make no physical sense, SIMP uses a penalization factor to encourage binary densities.

This penalization strategy (with  $p > 1$ ) is efficient but introduces non-convexity in the objective function, as stated by Sigmund and Maute (2013). As a result, SIMP is likely to converge towards local optima. Other techniques to encourage binary densities include filters, but they also introduce non-convexity (Sigmund and Maute 2013).

### 2.2 Deep Learning for Topology Optimization

Traditional TO methods are often slow due to the iterative FEA steps they include (Amir and Sigmund 2011). Many deep learning methods (Regenwetter, Nobari, and Ahmed 2022; Guo et al. 2018; Lin et al. 2018; Sosnovik and Osleedets 2019) have recently been developed to improve the speed and quality of topology generation or address issues such as non-convexity.

A group of deep learning approaches, to which our work belongs, consists in proposing an end-to-end TO framework from the constraints to the optimal topology (Oh et al. 2019; Sharpe and Seepersad 2019; Chandrasekhar and Suresh 2021; Parrott, Abueidda, and James 2022). Below, we review a few representative works. Yu et al. (2018) propose an iteration-free method, which predicts a low-resolution solution with a CNN encoder-decoder, which is then passed into a GAN to increase the resolution. In line with the work by Rawat and Shen (2019), Li et al. (2019) use two GANs to solve the TO problem and then predict the refined structure at high resolution. Sharpe and Seepersad (2019) introduce conditional GANs as a means of generating a compact latent representation of structures resulting from TO. Their work is improved by Nie et al. (2021), who greatly extend the generalizability in their model named TopologyGAN, which is trained on more diverse conditions and uses physical fields as input to represent loads and boundary conditions. In parallel, Wang et al. (2021) develop a U-Net to perform TO for improved generalization. These promising models nevertheless show a limited generalization ability, notably regarding boundary conditions outside the training distribution. They are also both prone to the problem of disconnected material. To solve this issue, Behzadi and Ilieş (2021) propose a conditional GAN architecture that includes a topological measure of connectivity in its loss function. Their results seem to improve generalizability and connectivity; however, they set the volume fraction to a constant value, limiting the problem’s scope.

It is crucial to note that none of these methods explicitly include a process to minimize compliance, which is the goal of TO. The minimization of compliance is expected indirectly through the GAN training, which is challenging to control. As incorporating some measure of predicted structural performance inside a conditional model seems necessary, we propose explicit guidance methods in diffusion models for low-compliance and good-feasibility structures.

### 2.3 Diffusion Models

Diffusion models are a new type of deep generative models (DGMs) introduced by Sohl-Dickstein et al. (2015). They have received much attention recently because Dhariwal and Nichol (2021) showed that diffusion models outperform GANs for image synthesis. Diffusion models are increasingly being applied to various fields: image generation (Nichol and Dhariwal 2021), segmentation (Amit et al. 2021), image editing (Meng et al. 2021), text-to-image (Nichol et al. 2022; Kim and Ye 2021), etc.

The idea behind diffusion models is to train a neural network to reverse a noising process that maps the data distribu-

tion to a white noise distribution. The forward noising process, which is fixed, consists of progressively adding noise to the samples following the Markov chain:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I) \quad (1)$$

where  $(\alpha_t)_{t=1}^T$  is a variance schedule.

To reverse this noising process, we approximate the true posterior with the parametric Gaussian process:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t)). \quad (2)$$

We then generate new data by sampling an image from  $\mathcal{N}(0, I)$  and gradually denoising it using Eq. 2.

Training a diffusion model, therefore, involves training two neural networks,  $\mu_\theta(x_t)$  and  $\Sigma_\theta(x_t)$ , to predict the mean and the variance of the denoising process respectively. Let us note that Ho, Jain, and Abbeel (2020) showed that  $\Sigma_\theta(x_t)$  might be fixed to a constant instead of being learned.

## 2.4 Guidance Methods in Diffusion Models

In many machine learning applications, a model is expected to generate samples conditioned on some input conditions. For example, popular text-to-image models such as DALL-E are conditioned on text input. Researchers have developed a few guidance methods to perform conditional image generation, such as including class labels when the model tries to generate an image corresponding to a given class.

**Including conditioning information inside the denoising networks** A method to condition a diffusion model consists in adding the conditioning information (for example, a class label) as an extra input to the denoising networks  $\mu_\theta$  and  $\Sigma_\theta$ . In practice, the conditioning information can be added as an extra channel to the input image. Similarly, Dhariwal and Nichol (2021) suggest adding conditioning information into an adaptive group normalization layer in every residual block.

**Classifier guidance** Additional methods have been developed to guide the denoising process using classifier output. In line with Sohl-Dickstein et al. (2015) and Song et al. (2020), who have used external classifiers to guide the denoising process, Dhariwal and Nichol (2021) introduce *classifier guidance* to perform class-conditional image generation. In *classifier guidance*, a separate classifier is trained on noisy data (with different levels of noise) to predict the probability  $p_\phi(y|x_t)$  that an image  $x_t$  at noise level  $t$  corresponds to the class  $y$ . Let  $p_\theta(x_t|x_{t+1})$  be an unconditional reverse noising process. Classifier guidance consists of sampling from:

$$p_{\theta,\phi}(x_t|x_{t+1}, y) = Z p_\theta(x_t|x_{t+1}) p_\phi(y|x_t) \quad (3)$$

instead of  $p_\theta(x_t|x_{t+1})$ , where  $Z$  denotes a normalizing constant. Under reasonable assumptions, Dhariwal and Nichol (2021) show that sampling from  $p_{\theta,\phi}(x_t|x_{t+1}, y)$  is equivalent to perturbing the mean with the gradient of the probability predicted by the classifier. Specifically, the perturbed mean is:

$$\hat{\mu}_\theta(x_t) = \mu_\theta(x_t) + s \Sigma_\theta(x_t) \nabla_{x_t} \log p_\phi(y|x_t) \quad (4)$$

where  $s$  is a scale hyperparameter that needs to be tuned. A variant called classifier-free guidance is proposed by Ho and Salimans (2021). This technique is theoretically close to classifier guidance but does not require training a separate classifier on noisy data.

However, none of these methods provide guidance for both continuous values (such as performance obtained from regression models) and discrete values (such as class labels obtained from classification models), which is important for TO to achieve feasible, high-performing samples. To overcome these issues, we propose a regressor and classifier guidance strategy that penalizes low-compliance and infeasible structures at every step.

## 3 Method

### 3.1 Architecture and General Pipeline

TopoDiff’s diffusion architecture consists of a UNet (Ronneberger, Fischer, and Brox 2015)-based denoiser at every step with attention layers. We add conditioning to this architecture by including information on constraints and boundary conditions as additional channels to the input image given to the denoiser, as shown in Figure 2. The UNet model uses these extra channels as additional information to denoise the first channel of the input in a way that respects the constraints and is optimal for the given boundary conditions. Similarly to TopologyGAN (Nie et al. 2021), we use physical fields, namely strain energy density and von Mises stress, to represent constraints and boundary conditions. The physical fields are computed using a finite element method (Guarín-Zapata and Gómez 2020) and help avoid the sparsity problem caused by raw constraints and boundary condition matrices. The final input to our conditional diffusion model has four channels representing the volume fraction, the strain energy density, the von Mises stress, and the loads applied to the domain’s boundary.

### 3.2 Minimizing compliance

Most deep learning models used for TO rely on minimizing the pixel-wise error between the output topology and the ground truth obtained with traditional methods. For instance, the reference model TopologyGAN (Nie et al. 2021) attempts to mimic the ground truth topology and is encouraged to do so by the L2 loss function of its generator. GANs for TO are often evaluated using mean absolute error (MAE) between the ground truth topology and the topology predicted by their model. However, we hypothesize that setting the minimization of a pixel-wise error as an objective does not properly address the aim of TO: generate manufacturable structures that minimize mechanical compliance. We pose this hypothesis for two main reasons:

1. The topology used as ground truth may be sub-optimal due to penalization factor and filters (Sec. 2.1);
2. A small pixel-wise error is compatible with a large compliance error if the material is missing at critical places.

Without additional guidance, our conditional diffusion model is prone to the same problem. To solve that issue,

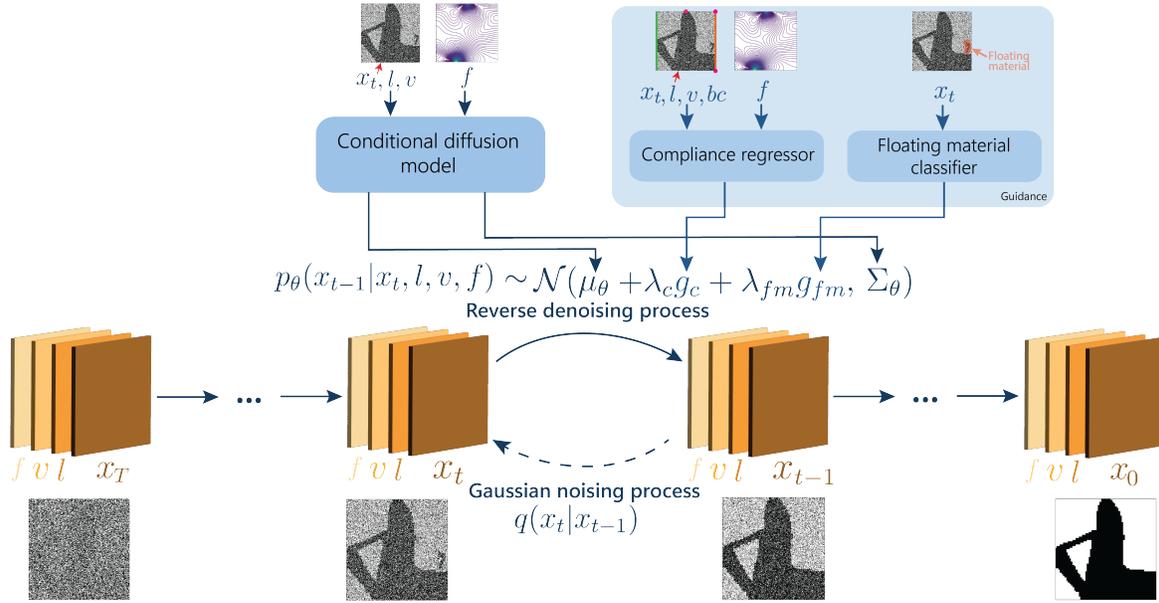


Figure 2: TopoDiff: Proposed constrained guided conditional diffusion model architecture for TO.  $(x_t)_{t=0, \dots, T}$  is the gradually denoised topology;  $g_c$  and  $g_{fm}$  are the guidance gradients;  $l$  is the load applied, represented with an arrow on the topology;  $v$  is the volume fraction;  $f$  are the physical fields, and  $bc$  are the boundary conditions, represented with color lines and dots.

we introduce a new type of guidance called *regressor guidance*, inspired by the *classifier guidance* used by Dhariwal and Nichol (2021).

Consider a conditional diffusion model as presented in Sec. 3.1:  $p_\theta(x_t|x_{t+1}, v, f, l)$ , where  $v$  is the *volume fraction*,  $f$  are the physical fields (strain energy density and von Mises stress) and  $l$  are the loads applied. *Regressor guidance* consists in sampling each transition according to:

$$p_{\theta, \phi}(x_t|x_{t+1}, v, f, l, bc) = \frac{1}{Z} p_\theta(x_t|x_{t+1}, v, f, l) e^{-c_\phi(x_t, v, f, l, bc)} \quad (5)$$

where  $c_\phi$  is a surrogate neural network predicting the compliance of the topology under given constraints,  $bc$  are the boundary conditions applied, and  $Z$  is a normalizing constant. It is worth noting that  $c_\phi$  must be able to predict compliance on noisy images of structures. To perform this task, we use the encoder of a UNet architecture modified for regression values.

We show in Sec. 3.5 that under simple assumptions, adding *regressor guidance* simply amounts to shifting the mean predicted by the diffusion model by  $-\Sigma \nabla_{x_t} c_\phi(x_t, v, f, l, bc)$  where  $\Sigma$  is the variance of the Gaussian distribution representing  $p_\theta(x_t|x_{t+1}, v, f, l)$ . This method thus modifies the distribution according to that which we sample at each step by penalizing structures with high compliance. The resulting algorithm is Alg. 1.

### 3.3 Avoiding Floating Material

Disconnected pixels in predicted structures are a serious problem because this phenomenon leads to floating material and therefore affects the manufacturability of the predicted topology. This problem is generally ignored in deep learning

**Algorithm 1** Regressor guidance for TO, given a conditional diffusion model  $(\mu_\theta(x_t|x_{t+1}, v, f, l), \Sigma_\theta(x_t|x_{t+1}, v, f, l))$  and a regressor  $c_\phi(x_t, v, f, l, bc)$ .

---

**Require:**  $v, l, bc$   $\triangleright$  Volume, loads and boundary conditions  
**Require:**  $f$   $\triangleright$  Physical fields  
**Require:**  $\lambda_c$   $\triangleright$  Regressor gradient scale

$x_T \leftarrow$  sample from  $\mathcal{N}(0, I)$   
**for**  $t$  from  $T$  to 1 **do**  
 $\mu, \Sigma \leftarrow \mu_\theta(x_t|x_{t+1}, v, f, l), \Sigma_\theta(x_t|x_{t+1}, v, f, l)$   
 $x_{t-1} \leftarrow$  sample from  $\mathcal{N}(\mu - \lambda_c \Sigma \nabla_{x_t} c_\phi(x_t, v, f, l, bc)|_{x_t=\mu}, \Sigma)$   
**return**  $x_0$

---

models for TO, and is notably not considered by the pixel-wise error because a small pixel-wise error is compatible with the presence of floating material.

Similar to what has been exposed in Sec. 3.2, we further modify the sampling distribution at each step by penalizing structures that contain floating material. To do so, we train a classifier  $p_\gamma$  that returns the probability that the topology does *not* contains floating material. We then use this classifier to perform *classifier guidance*, as introduced in Sec. 2.4. Eventually, this amounts to shifting the mean predicted by the diffusion model by  $+\Sigma \nabla_{x_t} \log p_\gamma(x_t)$ .

### 3.4 Combining Guidance Strategies

Our model ultimately consists of one conditional diffusion model  $p_\theta(x_t|x_{t+1}, v, f, l)$  and two surrogate models used for guidance when sampling:  $c_\phi(x_t, v, f, l, bc)$  for compliance and  $p_\gamma(x_t)$  for floating material.

One challenge is to find a way to combine these two guidance strategies. To combine them, we sample at every step

according to:

$$p_{\theta, \phi, \gamma}(x_t | x_{t+1}, v, f, l, bc) = Z p_{\theta}(x_t | x_{t+1}, v, f, l) e^{-c_{\phi}(x_t, v, f, l, bc)} p_{\gamma}(x_t). \quad (6)$$

This amounts to shifting the mean predicted by the diffusion model by:

$$-\lambda_c \Sigma \nabla_{x_t} c_{\phi}(x_t, v, f, l, bc) + \lambda_{f_m} \Sigma \nabla_{x_t} \log p_{\gamma}(x_t) \quad (7)$$

where  $\lambda_c$  and  $\lambda_{f_m}$  are gradient scale hyperparameters.

However, as is, this approach has two pitfalls: 1. The gradients are always computed at the same point  $\mu$  (the mean predicted by the diffusion model), even though this mean is shifted by the previous guidance strategy; and 2. The gradients are computed at every denoising step, even if we might want to favor one guidance over the other at a given denoising step. We modify the point at which the second gradient is computed to address these issues by considering the shift induced by the previous gradient. In addition, we determine a maximum level of noise (MLN) beyond which the *classifier/regressor guidance* should not be included for every classifier and regressor. We then introduce *classifier/regressor guidance* only if the image is denoised enough to have a noise level below the MLN of the given classifier or regressor.

The final guidance algorithm resulting from the combination of these guidance strategies is Alg. 2. Fig. 2 also summarizes the overall architecture.

---

**Algorithm 2** Guidance strategy for TO using Conditional Diffusion Model.

---

**Require:**  $v, l, bc$   $\triangleright$  Volume, loads and boundary conditions

**Require:**  $f$   $\triangleright$  Physical fields

**Require:**  $\lambda_c, \lambda_{f_m}$   $\triangleright$  Regressor and classifier gradient scale

**Require:**  $MLN_c, MLN_{f_m}$   $\triangleright$  Maximum levels of noise

$x_T \leftarrow$  sample from  $\mathcal{N}(0, I)$

**for**  $t$  from  $T$  to 1 **do**

$\mu, \Sigma \leftarrow \mu_{\theta}(x_t | x_{t+1}, v, f, l), \Sigma_{\theta}(x_t | x_{t+1}, v, f, l)$

**if**  $t < MLN_{f_m}$  **then**

$\mu \leftarrow \mu + \lambda_{f_m} \Sigma \nabla_{x_t} \log p_{\gamma}(x_t) |_{x_t=\mu}$

**if**  $t < MLN_c$  **then**

$\mu \leftarrow \mu - \lambda_c \Sigma \nabla_{x_t} c_{\phi}(x_t, v, f, l, bc) |_{x_t=\mu}$

$x_{t-1} \leftarrow$  sample from  $\mathcal{N}(\mu, \Sigma)$

**return**  $x_0$

---

It should be noted that we also considered applying *regressor* and *classifier guidance* for other constraints (volume, load position), but the conditional diffusion model already sufficiently respects these constraints and makes guidance unnecessary.

### 3.5 Mathematical Motivations for Regressor Guidance

Similarly to Dhariwal and Nichol (2021) about *classifier guidance*, we show in this section the mathematical motivations behind *regressor guidance*, and in particular, we prove that adding *regressor guidance*, i.e., sampling each transition

according to the modified distribution, amounts to a shift in the mean predicted by the diffusion model.

Let  $p_{\theta}(x_t | x_{t+1}, v, f, l)$  be our conditional diffusion model. *Regressor guidance* consists in sampling according to:

$$p_{\theta, \phi}(x_t | x_{t+1}, v, f, l, bc) = Z p_{\theta}(x_t | x_{t+1}, v, f, l) e^{-c_{\phi}(x_t, v, f, l, bc)} \quad (8)$$

where  $Z$  is a normalizing constant and all variables are the ones defined in Sec. 3.2.

Let  $\mu$  and  $\Sigma$  be the mean and variance of the Gaussian distribution representing  $p_{\theta}(x_t | x_{t+1}, v, f, l)$ .

$$\log p_{\theta}(x_t | x_{t+1}, v, f, l) = -\frac{1}{2}(x_t - \mu)^T \Sigma^{-1}(x_t - \mu) + C \quad (9)$$

where  $C$  is a constant.

By doing a Taylor expansion of the regressor predicting the compliance  $c_{\phi}(x_t, v, f, l, bc)$ , we obtain:

$$c_{\phi}(x_t, v, f, l, bc) \approx c_{\phi}(\mu, v, f, l, bc) + (x_t - \mu) \nabla_{x_t} c_{\phi}(x_t, v, f, l, bc) |_{x_t=\mu}. \quad (10)$$

In Eq. 10, we neglect the terms of second order and above because we make the assumption that the curvature of  $c_{\phi}(x_t)$  is low compared to  $\Sigma^{-1}$ , to which it will be summed in Eq. 11. This assumption is reasonable in the limit of infinite diffusion steps, where  $\|\Sigma\| \rightarrow 0$ , as stated by Dhariwal and Nichol (2021).

Hence, Eq. 10 can be rewritten  $c_{\phi}(x_t, v, f, l, bc) \approx (x_t - \mu)g_c + D$ , where  $g_c$  is the gradient of the compliance regressor evaluated in  $\mu$  and  $D$  is a constant.

This gives:

$$\log(p_{\theta}(x_t | x_{t+1}, v, f, l) e^{-c_{\phi}(x_t, v, f, l, bc)}) \approx -\frac{1}{2}(x_t - \mu)^T \Sigma^{-1}(x_t - \mu) - (x_t - \mu)g_c + C + D. \quad (11)$$

Hence:

$$\log(p_{\theta}(x_t | x_{t+1}, v, f, l) e^{-c_{\phi}(x_t, v, f, l, bc)}) \approx -\frac{1}{2}(x_t - \mu + \Sigma g_c)^T \Sigma^{-1}(x_t - \mu + \Sigma g_c) + \frac{1}{2}g_c^T \Sigma g_c + C + D. \quad (12)$$

The last three terms of Eq. 12 are all constant and are encapsulated in the normalizing constant  $Z$  from Eq. 8. Therefore, we have shown that  $p_{\theta, \phi}(x_t | x_{t+1}, v, f, l, bc)$  can be approximated by a Gaussian with a mean shifted by  $-\Sigma g_c$ .

## 4 Empirical Evaluation

We created three datasets to train the proposed models, which we make public to provide a standard benchmark for future research in this area.

## 4.1 Dataset

The main dataset used to train, validate and test TopoDiff consists of 33000 64x64 2D images corresponding to optimal structures for diverse input conditions. Every data sample contains six channels:

1. The first channel is the black and white image representing the optimal topology;
2. The second channel is uniform and includes the prescribed volume fraction;
3. The third channel is the von Mises stress of the full domain under the given load constraints and boundary conditions, defined as  $\sigma_{vm} = \sqrt{\sigma_{11}^2 - \sigma_{11}\sigma_{22} + \sigma_{22}^2 + 3\sigma_{12}^2}$ ;
4. The fourth channel is the strain energy density of the full domain under the given load constraints and boundary conditions, defined as  $W = \frac{1}{2}(\sigma_{11}\epsilon_{11} + \sigma_{22}\epsilon_{22} + 2\sigma_{12}\epsilon_{12})$ ;
5. The fifth channel represents the load constraints in the x-direction. Every node is given the value of the force applied in the x-direction on this load (0 if no force is applied on the load);
6. The sixth channel similarly represents the load constraints in the y-direction;

where  $(\sigma_{11}, \sigma_{22}, \sigma_{12})$  and  $(\epsilon_{11}, \epsilon_{22}, \epsilon_{12})$  are respectively the components of the stress and strain fields.

We randomly selected a combination of conditions (volume fraction, boundary conditions, loads) to generate every structure and then computed the optimal topology using the SIMP-based TO library ToPy (Hunter et al. 2017). We defined the possible conditions in a similar way to what was done in previous works, namely: 1. The volume fraction is chosen in the interval  $[0.3, 0.5]$ , with a step of 0.02; 2. The displacement boundary conditions are chosen among 42 scenarios for training and five additional scenarios only used for testing; 3. The loads are applied on unconstrained nodes randomly selected on the domain’s boundary. The direction is selected in the interval  $[0, \pi]$ , with a step of  $\frac{\pi}{6}$ .

The main dataset is divided into training, validation, and testing as follows:

1. The **training data** consist of 30,000 combinations of constraints containing 42 of the 47 boundary conditions;
2. The **validation data** consist of 200 new combinations of constraints containing the same 42 boundary conditions;
3. The **level 1 test data** consist of 1800 new combinations of constraints containing the same 42 boundary conditions;
4. The **level 2 test data** consist of 1000 new combinations of constraints containing five out-of-distribution boundary conditions.

In all test data, the combination of constraints is unseen. While *level 1* dataset contains boundary conditions that are also in the training data, we introduce more difficult conditions in *level 2* to rigorously compare the TopoDiff model’s generalization ability with existing methods. In addition to the main dataset, two other datasets consisting of 12,000 and 30,000 non-optimal structures are used to train *regressor* and *classifier guidance* models.

## 4.2 Evaluation Metrics

Using relevant evaluation metrics is vital for every scientific study. It is particularly critical for mechanical design generation because most metrics used in DGMs do not correspond to the physical objective one wants a design to achieve. In this work, contrary to most generative models applied to TO in previous works, we do not use pixel-wise error as a final evaluation metric because it does not guarantee low compliance, which is the objective we are trying to achieve.

Hence, we define and use four evaluation metrics that reflect the compliance minimization objective, as well as the constraints that the generated structures have to respect:

1. Compliance error (CE) relative to the ground truth, defined as:  $CE = (C(\hat{y}) - C(y))/C(y)$  where  $C(y)$  and  $C(\hat{y})$  are, respectively, the compliance of the SIMP-generated topology and the topology generated by our diffusion model. It should be noted that a negative compliance error means that our model returns a topology with lower compliance than the ground truth;
2. Volume fraction error (VFE) relative to the input volume fraction, defined as:  $VFE = |VF(\hat{y}) - VF(y)|/VF(y)$  where  $VF(y)$  and  $VF(\hat{y})$  are, respectively, the prescribed volume fraction and the volume fraction of the topology generated by our diffusion model;
3. Load violation (LV), defined as a boolean that is 1 if there is no material at a place where a load is applied and 0 if there is always material where loads are applied;
4. Presence of floating material (FM), defined as a boolean that is 1 if the topology contains floating material and 0 otherwise.

A model which generates samples with high scores on these metrics is expected to yield high-performance manufacturable designs.

## 4.3 Choice of hyperparameters

One of the most crucial hyperparameters is the gradient scales in our guidance strategies. These parameters quantify the relative importance of compliance minimization and floating material avoidance. As explained in Sec. 4.1, a validation dataset of 200 structures was used to perform hyperparameter tuning. We used a grid search method to decide the hyperparameters using compliance error and floating material presence as evaluation metrics. Topology generation and FEA were used to evaluate the results.

# 5 Results and Discussions

## 5.1 Evaluation of the full diffusion model

To evaluate the performance, we use the two test sets described in Sec. 4.1, corresponding to two difficulty levels. We run every test nine times and then compute the results’ average. We compare the performance of our model on all evaluation metrics (Sec. 4.2) to a state-of-art cGAN model, named TopologyGAN (Nie et al. 2021), which performs the same task as our model.

Fig. 3 shows examples of a few structures obtained with the SIMP method (ground truth), with TopologyGAN, and

Model	Level 1 test data			Level 2 test data		
	TopologyGAN	Unguided TopoDiff	Guided TopoDiff	TopologyGAN	Unguided TopoDiff	Guided TopoDiff
Average Compliance Error (%)	48.51 ± 16.38	<b>4.10</b> ± 0.88	4.39 ± 0.94	143.08 ± 38.50	22.13 ± 8.52	<b>18.40</b> ± 5.88
Median Compliance Error (%)	2.06	<b>0.80</b>	0.83	6.82	1.88	<b>1.82</b>
Proportion of Compliance Error >30% (%)	10.11	<b>2.33</b>	2.56	24.10	8.20	<b>8.10</b>
Average Volume Fraction Error (%)	11.87 ± 0.52	1.86 ± 0.03	<b>1.85</b> ± 0.03	14.31 ± 0.75	1.81 ± 0.04	<b>1.80</b> ± 0.04
Proportion of Load Violation (%)	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
Proportion of Floating Material (%)	46.78	6.64	<b>5.54</b>	67.90	7.53	<b>6.21</b>

Table 1: Comparison of performance between TopologyGAN and TopoDiff (guided and not guided) on the two level test sets. Values after  $\pm$  indicate the 95 % confidence interval around averages. The values in bold are the best ones for each level.

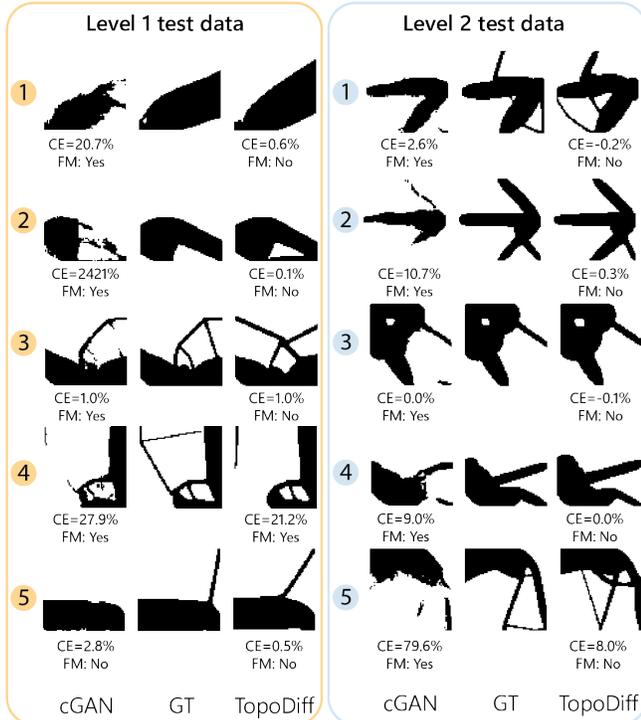


Figure 3: Comparison of generated structures on randomly selected samples from both test datasets. GT stands for ground truth, CE is the compliance error relative to the GT, and FM indicates the presence or not of floating material.

with TopoDiff for randomly selected constraints from *level 1* and *level 2* test sets. Qualitatively, we notice that TopologyGAN tries to mimic pixel-wise the topology obtained from SIMP but neglects both the compliance and the manufacturability of the generated structures, which almost all have some floating material and high compliance error. The ten structures generated by TopoDiff, on the other hand, may visually differ more from the SIMP results but have better physical properties than TopologyGAN. Only one of the TopoDiff-generated structures have no floating material, and all ten outperform the TopologyGAN structures in terms of compliance error. To confirm these qualitative observations, Table 1 summarizes the performance of the structures obtained with all test sets. TopoDiff outperforms TopologyGAN on all the metrics.

On the *level 1* test set, TopoDiff notably reduces the average CE by a factor of eleven and the proportion of FM

by more than a factor of eight. The proportion of non-manufacturable designs thus drops from 46.8% with cGAN to 5.5% with TopoDiff. It also significantly reduces the average VFE from 11.9% to 1.9%.

On the *level 2* test set, TopoDiff achieves demonstrates strong generalizability performance. It performs an eight-times reduction in the average CE, from 143.1% to 18.4%, and a four-times reduction in the median CE. Non-manufacturability drops from 67.9% to 6.2%, while the VFE is reduced by a factor of eight, from 14% to less than 2%. A paired one-tailed t-test confirms a reduction of the average CE and of the average VFE with a p-value of  $9 \cdot 10^{-12}$  and  $5 \cdot 10^{-160}$  respectively. These results show the efficacy of diffusion models in learning to generate high-performing and manufacturable structures for a wide set of testing conditions.

## 5.2 Efficiency of guidance strategy

**Surrogate models** Guidance can only work if the regressors and classifiers can perform well on the challenging task of predicting compliance and floating material for noisy images. Table 2 shows the compliance regressor and floating material classifier performance according to the noise level. These results show that both surrogate models are very reliable on low-noise structures, and as expected, their performance decreases with an increase in noise.

	0-25% noise	25-75% noise	75-100% noise	Global
Regressor R2 (%)	82.4	82.4	61.8	77.3
Classifier accuracy (%)	98.8	76.8	54.6	76.8

Table 2: Performance (R2-score and accuracy) of both surrogate models on validation data with respect to noise level.

**Ablation study** To evaluate the impact of our guidance strategy on the performance of TopoDiff, we also tested it *without* guidance on the two test sets, as shown in Table 1.

With *in-distribution* boundary conditions (*level 1*), our guidance strategy has no significant impact on compliance error (both on average and median). A two-tailed paired t-test does not reject the null hypothesis ( $p = 0.1$ ). We believe that this happens because the diffusion model has implicitly learned to perform well with these boundary conditions and does not need explicit compliance guidance. In contrast, our guidance strategy significantly impacts the proportion of floating material, with decreases from 6.6% to 5.5%.

With *out-of-distribution* boundary conditions (*level 2*), the positive impact of our guidance strategy is evident both on the average compliance error and on the proportion of floating material. A paired one-tailed t-test confirms a reduction

of the average CE with a p-value of 0.05. The average compliance error is reduced by 17% and the average proportion of floating material by 18%. As expected, guidance seems to have no effect on load respect and on volume fraction error. More interestingly, guidance seems to have no significant effect on the median of the compliance error, which suggests that compliance *regressor guidance* primarily reduces the number of structures with very high compliance errors.

### 5.3 Limitations and future work

TopoDiff shows good performance and good generalization to out-of-distribution boundary conditions. The proposed guidance strategy is beneficial to minimizing compliance and ensuring constraints, such as manufacturability. However, several challenges still need to be addressed. The most significant limitation is the computation time, diffusion models being slower than GANs. It takes 0.06 seconds for TopologyGAN to generate one topology, while TopoDiff needs 21.59 seconds. Reducing the computation time of diffusion models has recently seen significant successes (Ma et al. 2022), which will directly improve TO-based diffusion models. Other potential directions for future research include applying TopoDiff to more complex TO problems, notably 3D problems, and scaling it to higher resolutions and more boundary conditions. Reducing dependency on mesh size and large training datasets is also critical. We provide a framework for conditioning a diffusion model with constraints, training it on optimal data, and guiding it with a regressor predicting physical performance and some classifiers predicting the respect of constraints. This is a general method that should allow for solving similar design generation problems involving performance objectives and constraints, such as aerodynamic design (Heyrani, Chen, and Ahmed 2021) and bicycle synthesis (Regenwetter, Curry, and Ahmed 2022). Future work also includes expanding the TopoDiff framework to solve many inverse problems in engineering domains with multi-modal inputs.

## 6 Conclusion

Diffusion Models have been extremely successful in modeling high-dimensional multi-modal distributions with astonishing results in high-fidelity image generation. We propose TopoDiff — a conditional diffusion model to perform end-to-end topology optimization. By introducing conditional diffusion models for topology optimization, we show that diffusion models can outperform GANs in engineering design applications too. Our model is augmented with an explicit guidance strategy to ensure performance maximization and avoidance of non-manufacturable designs. TopoDiff achieves an eight-times reduction in the average compliance error and produces 11-times fewer non-manufacturable designs compared with a state-of-art conditional GAN. It also achieves an eight-times reduction in volume fraction error and generalizes well to out-of-distribution boundary conditions. More generally, we provide a diffusion model-based framework to solve many physical optimization problems in engineering with performance objectives and constraints.

## Acknowledgments

The authors gratefully acknowledge the funding received from Mines Paris Foundation. They would also like to thank MISTI France for supporting this research.

## References

- Ahmed, F.; Bhattacharya, B.; and Deb, K. 2013. Constructive solid geometry based topology optimization using evolutionary algorithm. In *Proceedings of seventh international conference on bio-inspired computing: theories and applications (BIC-TA 2012)*, 227–238. Springer.
- Amir, O.; and Sigmund, O. 2011. On reducing computational effort in topology optimization: how far can we go? *Structural and Multidisciplinary Optimization*, 44(1): 25–29.
- Amit, T.; Nachmani, E.; Shaharabany, T.; and Wolf, L. 2021. SegDiff: Image Segmentation with Diffusion Probabilistic Models. *CoRR*, abs/2112.00390.
- Behzadi, M. M.; and Ilieş, H. T. 2021. GANTL: Toward Practical and Real-Time Topology Optimization With Conditional Generative Adversarial Networks and Transfer Learning. *Journal of Mechanical Design*, 144(2). 021711.
- Bendsøe, M. P.; and Kikuchi, N. 1988. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2): 197–224.
- Chandrasekhar, A.; and Suresh, K. 2021. TOuNN: Topology Optimization Using Neural Networks. *Struct. Multidiscip. Optim.*, 63(3): 1135–1149.
- Dhariwal, P.; and Nichol, A. 2021. Diffusion Models Beat GANs on Image Synthesis. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 8780–8794. Curran Associates, Inc.
- Guarín-Zapata, N.; and Gómez, J. 2020. SolidsPy: 2D-Finite Element Analysis with Python.
- Guo, T.; Lohan, D. J.; Cang, R.; Ren, M. Y.; and Allison, J. T. 2018. An Indirect Design Representation for Topology Optimization Using Variational Autoencoder and Style Transfer. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, AIAA 2018-0804.
- Heyrani, A. N.; Chen, W.; and Ahmed, F. 2021. PcDGAN: A Continuous Conditional Diverse Generative Adversarial Network For Inverse Design. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD-21, KDD '21*. New York, NY, USA: Association for Computing Machinery.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 6840–6851. Curran Associates, Inc.

- Ho, J.; and Salimans, T. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.
- Hunter, W.; et al. 2017. ToPy - Topology optimization with Python. <https://github.com/williamhunter/topy>.
- Kim, G.; and Ye, J. C. 2021. DiffusionCLIP: Text-guided Image Manipulation Using Diffusion Models. *CoRR*, abs/2110.02711.
- Li, B.; Huang, C.; Li, X.; Zheng, S.; and Hong, J. 2019. Non-iterative structural topology optimization using deep learning. *Computer-Aided Design*, 115: 172–180.
- Lin, Q.; Hong, J.; Liu, Z.; Li, B.; and Wang, J. 2018. Investigation into the topology optimization for conductive heat transfer based on deep learning approach. *International Communications in Heat and Mass Transfer*, 97: 103–109.
- Ma, H.; Zhang, L.; Zhu, X.; and Feng, J. 2022. Accelerating Score-Based Generative Models with Preconditioned Diffusion Sampling. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, 1–16. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-20049-6.
- Meng, C.; Song, Y.; Song, J.; Wu, J.; Zhu, J.; and Ermon, S. 2021. SDEdit: Image Synthesis and Editing with Stochastic Differential Equations. *CoRR*, abs/2108.01073.
- Nichol, A. Q.; and Dhariwal, P. 2021. Improved Denoising Diffusion Probabilistic Models. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 8162–8171. PMLR.
- Nichol, A. Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 16784–16804. PMLR.
- Nie, Z.; Lin, T.; Jiang, H.; and Kara, L. B. 2021. Topology-GAN: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain. *Journal of Mechanical Design*, 143(3). 031715.
- Oh, S.; Jung, Y.; Kim, S.; Lee, I.; and Kang, N. 2019. Deep Generative Design: Integration of Topology Optimization and Generative Models. *Journal of Mechanical Design*, 141(11). 111405.
- Parrott, C.; Abueidda, D.; and James, K. A. 2022. Multi-Head Self-Attention GANs for Multiphysics Topology Optimization. In *AIAA AVIATION 2022 Forum*, 3726.
- Rawat, S.; and Shen, M. H. 2019. A Novel Topology Optimization Approach using Conditional Deep Learning. *CoRR*, abs/1901.04859.
- Regenwetter, L.; Curry, B.; and Ahmed, F. 2022. BIKED: A Dataset for Computational Bicycle Design With Machine Learning Benchmarks. *Journal of Mechanical Design*, 144(3).
- Regenwetter, L.; Nobari, A. H.; and Ahmed, F. 2022. Deep Generative Models in Engineering Design: A Review. *Journal of Mechanical Design*, 144(7). 071704.
- Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N.; Hornegger, J.; Wells, W. M.; and Frangi, A. F., eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 234–241. Cham: Springer International Publishing. ISBN 978-3-319-24574-4.
- Rozvany, G. I. N.; Zhou, M.; and Birker, T. 1992. Generalized shape optimization without homogenization. *Structural optimization*, 4: 250–252.
- Sharpe, C.; and Seepersad, C. C. 2019. Topology Design With Conditional Generative Adversarial Networks. Volume 2A: 45th Design Automation Conference. V02AT03A062.
- Sigmund, O. 2001. A 99 Line Topology Optimization Code Written in Matlab. *Struct. Multidiscip. Optim.*, 21(2): 120–127.
- Sigmund, O.; and Maute, K. 2013. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, 48(6): 1031–1055.
- Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 2256–2265. Lille, France: PMLR.
- Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-Based Generative Modeling through Stochastic Differential Equations. *CoRR*, abs/2011.13456.
- Sosnovik, I.; and Oseledets, I. 2019. Neural networks for topology optimization. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 34(4): 215–223.
- Wang, D.; Xiang, C.; Pan, Y.; Chen, A.; Zhou, X.; and Zhang, Y. 2021. A deep convolutional neural network for topology optimization with perceptible generalization ability. *Engineering Optimization*, 54(6): 973–988.
- Yu, Y.; Hur, T.; Jung, J.; and Jang, I. G. 2018. Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization*, 59(3): 787–799.

## A1 Additional datasets used for surrogate models

The main dataset used to train, validate and test the diffusion model was presented in the paper. This section provides more information about the two other datasets that were used to train and validate the two surrogate models, namely the regressor predicting compliance and the classifier predicting the presence of floating material.

For the regressor predicting compliance, we used a dataset of 72000 labeled samples, with the label indicating the compliance of the topology under the given constraints. Every sample is an image containing eight channels (black and white topology, volume fraction, von Mises stress, strain energy density, load in x-direction, load in y-direction, x-displacement boundary condition and y-displacement boundary condition). In order to give our regressor good generalization abilities, we not only used optimal topologies (generated by SIMP), but also negative samples, corresponding to non optimal topologies. This dataset includes:

- 30000 images from the diffusion model main dataset, corresponding to *optimal topologies* (25000 for training, 5000 for validation);
- 12000 images corresponding to *non-optimal topologies*, generated using the *fake-load method*, explained below (10000 for training, 2000 for validation);
- 30000 images corresponding to *non-optimal topologies*, generated by the unguided conditional diffusion model by using the same constraints as in the main dataset (25000 for training, 5000 for validation).

It should be noted that the final dataset contains less than 72000 samples because it was filtered to remove outlier structures (with compliance higher than 50 for the first 42000 and with compliance higher than 25 for the diffusion-generated structures). To generate non-optimal structures, we notably used the *fake-load method*, which consists in adding an extra load to the input constraints given to SIMP for generating the structure. SIMP thus generates a topology that is optimal for the extra-loaded constraints but not for the real ones. This method allows to generate non-optimal data while still respecting basic requirements like the presence of material where loads are applied.

For the classifier predicting the presence of floating material, we used a dataset of 70000 labeled samples (58000 for training and 12000 for validation), with the label being a boolean indicating if floating material is present on the topology or not. Every sample is an image containing only one channel: the black and white topology. This dataset includes:

- the first 15000 images from the diffusion model main dataset, on which floating material was added;
- the first 14000 images from a dataset containing topologies at diverse volume fractions, on which floating material was added;
- the first 6000 images from the non-optimal topologies dataset generated with the *fake-load method*, on which floating material was added;

- the next 15000 images from the diffusion model main dataset, on which floating material was *not* added;
- the next 14000 images from a dataset containing topologies at diverse volume fractions, on which floating material was *not* added;
- the next 6000 images from the non-optimal topologies dataset generated with the *fake-load method*, on which floating material was *not* added.

It should be noted that while some datasets were used to train several different models, we have paid careful attention to avoiding leakage of data from training datasets to validation datasets. All models were validated on data that they had not been trained on. The final TopoDiff model (diffusion model + surrogate models) was tested on data that was never used for training or validating any of its component models.