Noname manuscript No.
(will be inserted by the editor)

Length Scale Control in Topology Optimization using Fourier Enhanced Neural Networks

Aaditya Chandrasekhar · Krishnan Suresh

the date of receipt and acceptance should be inserted later

Abstract Length scale control is often imposed in topology optimization (TO) to make the design amenable to manufacturing and other functional requirements. While several length scale control strategies have been proposed, practical challenges often arise due to the coupling between the mesh size and length scales. In particular, when a maximum length scale is imposed to obtain thin members, extraction of the boundary is often impaired by the mesh resolution.

In this paper we propose a mesh-independent length scale control strategy, by extending a recently proposed SIMP-based TO formulation using neural networks (TOuNN). Specifically, we enhance TOuNN with a Fourier space projection, to control the minimum and/or maximum length scales. The proposed method does not involve additional constraints, while the sensitivity computations are automated through the neural network's backpropagation, making the method easy to implement. It is illustrated through several numerical experiments in 2D and 3D.

1 Introduction

Topology optimization (TO) [4] is a design method which distributes material within a domain to minimize an objective function, subject to constraints. Various constraints are often imposed to make the design amenable to manufacturing and other functional requirements. One such important set of constraints is to limit the maximum and minimum length scale of structural members. Imposing a maximum length l_{max} will lead to beam-like structures, while imposing a minimum length l_{min} avoids extremely thin non-manufacturable members.

While several methods have been proposed to impose length scale control in TO (see Section 2), practical challenges often arise due to the coupling between the mesh size and length scales. In particular, when l_{max} is imposed to obtain thin members, extraction of the topology is often limited by the mesh size.

In this paper we propose length scale control by extending a mesh-independent SIMP-based TO formulation using neural networks (TOuNN) proposed in [5]. Specifically, in Section 3, we enhance TOuNN with a Fourier space projection, leading to a control on both the minimum and maximum length scales. The proposed method is

Aaditya Chandrasekhar
Department of Mechanical Engineering
University of Wisconsin-Madison
E-mail: achandrasek3@wisc.edu

Krishnan Suresh
Department of Mechanical Engineering
University of Wisconsin-Madison
E-mail: ksuresh@wisc.edu

illustrated through numerous experiments in Section 5. Open research challenges, opportunities and conclusions are summarized in Section 6.

2 Literature Review

Popular TO methods today include Solid Isotropic Material with Penalization (SIMP) ([4], [35]), level-set methods [32], topological sensitivity methods ([**? ?**]), etc. For a comprehensive review, see [29, 7]. Among these, SIMP is perhaps the most popular, and it serves as a basis for this paper.

2.1 Topology Optimization using Mesh-Based SIMP

Consider the generic TO problem illustrated in Figure 1. In SIMP, one defines a pseudo-density $\rho(\boldsymbol{x}) \in (0, 1]$ over the domain, where ρ field is typically represented using an underlying mesh. Then, by relying on standard SIMP penalization technique of the form $E = E_0\rho^p$, the field is optimized using optimality criteria [4] [16] or MMA [36], resulting in the desired topology, as illustrated in Figure 1.

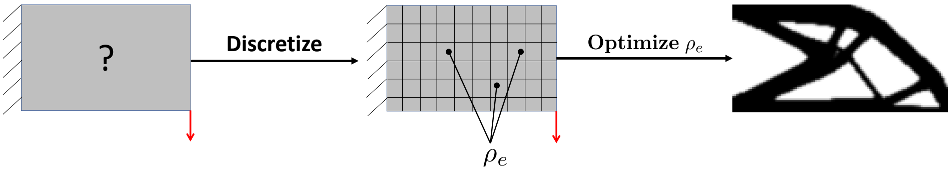


Fig. 1: Classic TO using mesh-based SIMP.

Formally, the TO problem in SIMP is posed as

$$\underset{\rho}{\text{minimize}} \quad \boldsymbol{u}^{\mathsf{T}}\boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{u} \tag{1a}$$

$$\text{subject to} \quad \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{u} = \boldsymbol{f} \tag{1b}$$

$$\sum_e \rho_e v_e = V^* \tag{1c}$$

where $\boldsymbol{u}$ is the displacement field, $\boldsymbol{K}$ is the finite element stiffness matrix, $\boldsymbol{f}$ is the applied force, ρ_e is the density associated with element e, v_e is the volume of the element, and V^* is the prescribed volume.

2.2 Length Scale Control in Topology Optimization

There has been considerable progress in imposing length scale control in TO; see [18] for a review. Broadly, there are two types of length-scale control: exact and approximate. Exact length control implies that all features must lie precisely within the range specified, and is much harder to achieve. The skeletal based minimum and maximum length scale control presented in [42] belongs to this category. Such controls can be computationally demanding; further, practical challenges in designing thin structures were reported [42]. Most other length scale control techniques discussed below are approximate, and are often sufficient in practice.

The popular sensitivity filter [34] to eliminate mesh dependency indirectly controls minimum length scale. A projection filter with tunable support was suggested by [13] to control the minimum length scale. However, these filters result in topologies with diffused boundaries. A scheme for imposing maximum length scale was developed in

[12]. In this approach, the radius of a circular test region determines the maximum allowable member size. However, the large number of nonlinear constraints can pose challenges. Therefore, the constraints were aggregated using a p-norm form in [38, 39], within the context of infill optimization. However, due to the nature of the constraints, the resulting volume fraction was less than the desired value. The authors of [8] recently introduced constraints for minimum and maximum member size, minimum cavity size, and minimum separation distance. Apart from requiring a large number of elements to obtain crisp boundaries, the computation and storage of these filters can also be expensive.

Imposing approximate length scale control in the frequency domain has been suggested in [17]. A band pass filter was constructed to impose both maximum and minimum length scale controls. While this avoids additional constraints, the proposed method was restricted to rectangular domain, with non-rectangular domains requiring padding. Further, the proposed technique produced designs with large number of gray elements requiring post-processing.

While most length scale control formulations are SIMP based, it has also been proposed for level-set methods. For example, an implicit strain energy based approach to impose the minimum length scale constraint in [43]. The authors of [6] introduced a quadratic energy functional into the formulation for length control.

The length scale control method proposed in this paper uses the frequency projection concept (analogous to [17]), but uses a mesh-independent representation of the density field for easy extraction of crisp topologies, and applies to non-rectangular domains as well. The proposed method relies on the TOuNN framework proposed in [5], reviewed next.

2.3 Topology Optimization using Neural Networks

Consider once again a generic TO problem in Fig 2. As in mesh based SIMP, we define a pseudo-density field ρ over the entire domain. However, instead of representing ρ using the mesh, it is represented using a fully-connected neural network (NN). In other words, ρ is spanned by NN's activation functions and controlled by the weights w within the NN. Then, by relying on SIMP penalization, the density is optimized using the NN's built-in optimizer, to result in the desired topology; see Figure 2.

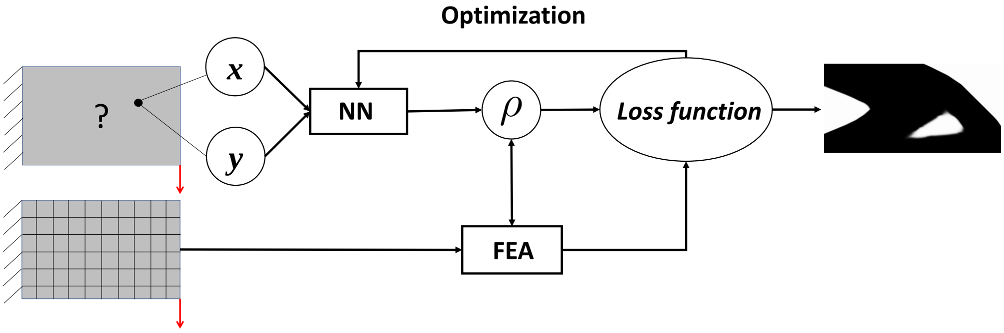


Fig. 2: TOuNN framework [5].

As illustrated in Figure 3, the NN typically consists of several hidden layers (depth); each layer may consist of several activation functions such as LeakyReLU [11], [20] coupled with batch normalization [14]. By varying the height and depth, one can increase the representational capacity of the NN. The final layer is a softMax function that scales the output to values between 0 and 1.

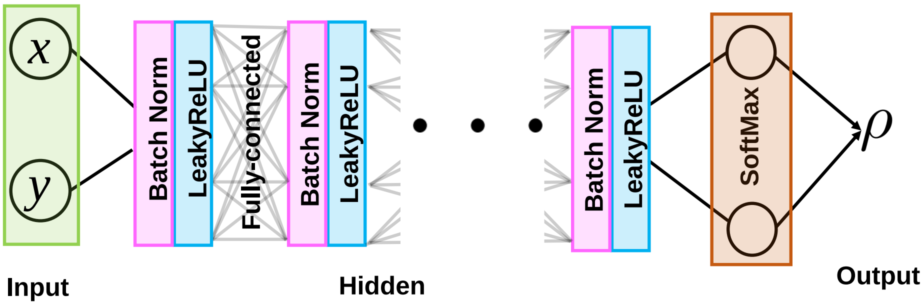


Fig. 3: The architecture of the neural net in the simple TOuNN framework [5].

Formally, in TOuNN, the topology optimization problem is posed as:

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \boldsymbol{u}^{\mathsf{T}}\boldsymbol{K}(\boldsymbol{w})\boldsymbol{u} \tag{2a}$$

$$\text{subject to} \quad \boldsymbol{K}(\boldsymbol{w})\boldsymbol{u} = \boldsymbol{f} \tag{2b}$$

$$\sum_e \rho_e(\boldsymbol{w})v_e = V^* \tag{2c}$$

where $\boldsymbol{w}$ are the weights associated with the NN. Further, TOuNN relies standard on SIMP penalization of the form $E = E_0\rho^p$. Thus, as reported in [5], the significant differences compared to mesh-based SIMP are : 1. The design variables are the NN weights $\boldsymbol{w}$. 2. The boundary of the topology is crisp and differentiable since the field $\rho(\boldsymbol{x})$ is analytically defined over the entire domain. 3. The sensitivities are computed analytically using NN's backpropagation. 4. Optimization is carried out using NN's built-in optimizer.

3 Proposed Method

In theory, one can capture *any* density field in TOuNN by simply increasing the number of weights and tuning them appropriately. However, for reasons discussed in [27], standard NNs fail to efficiently capture high frequencies. This has been reported, for example, in capturing volumetric density [23], occupancy [21], signed distances[25]. Analysis shows that the eigenvalue spectrum of these networks decay rapidly as a function of frequency [37]. To alleviate this problem, projecting the input spatial coordinates to a Fourier space was proposed in [37].

With this as motivation, the proposed architecture of the Fourier enhanced NN for TO is illustrated in Figure 4. The critical features of the proposed Fourier-TOuNN framework are:

1. **Input Layer**: As before, the input to the Fourier-NN is any point $\boldsymbol{x} \in \mathbf{R}^d$ in the domain.
2. **Fourier Projection Layer**: This layer is the focus of the current work. The d dimensional Euclidean input is transformed to a $2n_f$ dimensional Fourier space (see Section 3.1), where n_f is the chosen number of frequencies. The range of frequencies is determined by the length scale controls:

$$f_i \in [\frac{h}{l_{\max}}, \frac{h}{l_{\min}}] \tag{3}$$

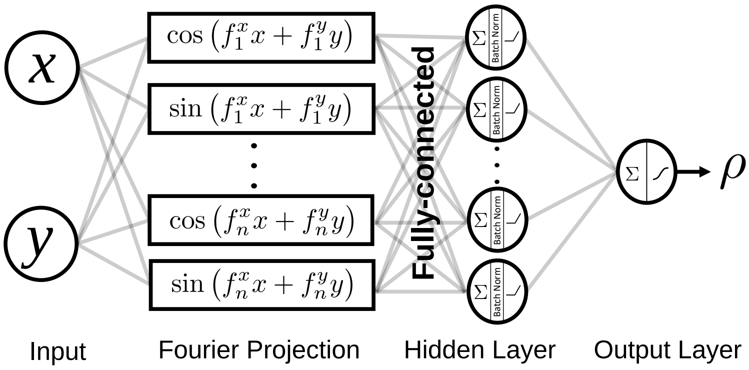


Fig. 4: The architecture of the proposed neural net for 2D problems.

where h is the mesh edge length. Further, in this paper, the frequencies are uniformly sampled within the specified interval.

3. **Hidden Layers**: The output from the projected Fourier space is piped into a single layer of neurons (compared to multiple layers in Figure 3). This hidden layer is important in that it introduces an essential non-linearity, as demonstrated later in Section 5. As in TOuNN, each of the neurons is associated with an activation function such as leaky-rectified linear unit (LeakyReLU) ($LR^{\epsilon}(x) = max(0, x) + \epsilon min(0, x)$). Batch normalization is used for regularization.
4. **Output Layer**: The final layer consists of one neuron with a Sigmoid activation function, ensuring that the output density lies between 0 and 1.

In summary the output density (ignoring batch normalization) can be expressed as:

$$\rho = \frac{1}{1 + exp\Big(- \sum_j w_j LR^{\epsilon}_j \big(\sum_{i=1}^{n_f} w_{ij} \cos(f_i^x x + f_i^y y) + \sum_{i=n_f}^{2n_f} w_{ij} \sin(f_i^x x + f_i^y y) \ \big) \Big)} \tag{4}$$

where LR^{ϵ}_j represents the LeakyReLU operator.

3.1 A Simplified Scenario

To comprehend the transformations within the Fourier-TOuNN netowrk, consider a simplified version in Figure 5, where we limit the input dimension to 1, the frequency terms are limited to 2, and only one LeakyReLU neuron is used. The output is a single sigmoid activated function as before. Then, the closed form expression for density simplifies from 4 to:

$$\rho(x) = \frac{1}{1 + exp(-w_3 LR^{\epsilon}(w_1 \cos(f_1^x x) + w_2 \cos(f_2^x x)))} \tag{5}$$

Consider an instance where $w_1 = 8$, $w_2 = 10$, $w_3 = 4$, $f_1^x = 1$ and $f_2^x = 6$. Figure 6 illustrates the different stages of the transformation. The plot on the top left is a simple linear combination of the two frequencies, while the plot on the top right is the output from the LeakyReLU function, which is essentially a linear transformation in the positive region, but that thresholds the negative valyes to $-\epsilon$. The Sigmoid (bottom right) further thresholds the output to

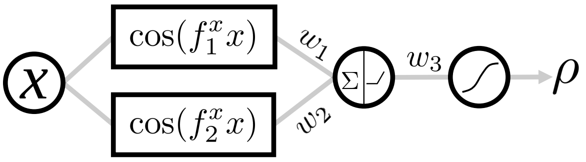


Fig. 5: Simplified representation of Neural Network with Fourier Projection.

$(0, 1]$. The plot on the bottom left is the Fourier decomposition of the density field. As expected, the peaks are at the specified frequencies of 1 and 6, but spread out. This spreading of the frequencies is discussed, for example, in [37]. One can expect similar behavior in higher dimensions, and for a larger set of frequencies.

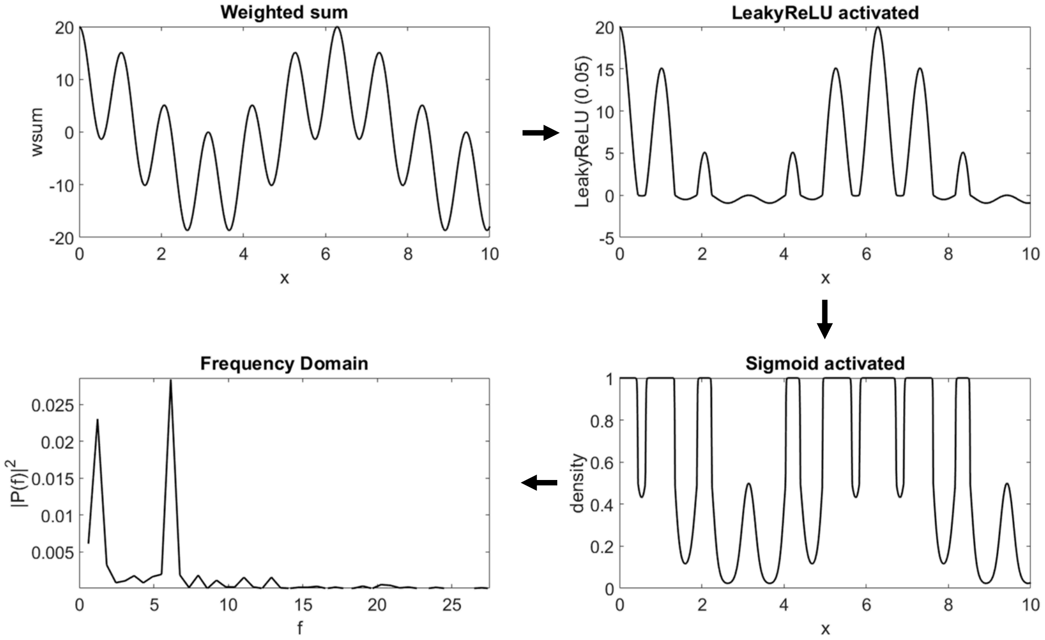


Fig. 6: Different stages of transformation in a Fourier-TOuNN.

3.2 Finite Element Analysis

A finite element mesh is still needed to capture the underlying physics. Here, we use regular node quad elements, and fast Cholesky factorization based on the CVXOPT library [1], for both thermal and structural analysis. During each iteration, the density at the center of each element is computed by the NN, and is provided to the FE solver. The FE solver computes the stiffness matrix for each element, and the assembled global stiffness matrix is then used to determine the field $\boldsymbol{u}$ and the unscaled element compliances:

$$J_e = \{u_e\}^T [K]_0 \{u_e\} \tag{6}$$

The total compliance is given by:

$$J = \sum_e \rho_e^p J_e \tag{7}$$

where p is the usual SIMP penalty parameter.

3.3 Loss Function and Sensitivity

NNs are designed to minimize an unconstrained loss function using built-in procedures such as Adam optimization [15]. We therefore convert the constrained minimization problem in Equation 2 into an unconstrained loss L using a simple penalty formulation [24]:

$$L(\boldsymbol{w}) = \frac{\sum\limits_{e} \rho_e^p J_e}{J_0} + \alpha \left(\frac{\sum\limits_{e} \rho_e v_e}{V^*} - 1 \right)^2 \tag{8}$$

where J_0 is the initial compliance (for scaling), p is the SIMP penalization parameter and α is the constraint penalty. As described in [24], starting from a small positive value for the penalty parameter α, a gradient driven step is taken to minimize the loss function. Then, the penalty parameter is increased and the process is repeated. Observe that, in the limit $\alpha \to \infty$, when the loss function is minimized, the equality constraint is satisfied and the objective is thereby minimized.

We rely on Adam optimization [15] to perform the optimization. The optimizer being a first-order method requires gradient information to optimize. NNs rely on backpropagation [31, 30, 3] to analytically compute the sensitivity of loss functions with respect to the weights and bias. This is possible since the activation functions are analytically defined, and the output can be expressed as a composition of such functions. Here, the sensitivities are computed automatically using automatic differentiation of PyTorch [26]. We refer the readers to [5] for a detailed derivation on sensitivity analysis corresponding to the loss function in Equation 8.

4 Algorithm

The optimization procedure is summarized in Algorithm 1. We will assume that the NN has been constructed with a desired number of layers, nodes per layer and activation functions. The NN is initialized using a Glorot initializer [10]. We discretize the domain into a mesh (Ω^h). The coordinates of the element centers are used as inputs to the NN during the training process. The density at the center of element e is denoted by ρ_e. The loss function is calculated from these discrete values. Here, we use the continuation scheme where the parameter p is incremented to avoid local minima [28], [33]. The process is then repeated until termination. The algorithm is set to terminate if the percentage of gray elements (elements with densities between 0.05 and .95) $\epsilon_g = N_{grey}/N_{total}$ is less than a prescribed value.

5 Numerical Experiments

In this section, we conduct several numerical experiments to illustrate the proposed algorithm. The implementation is in Python.The default parameters are listed in Table 1.

All experiments were conducted on an Intel i7 - 6700 CPU @ 2.6 Ghz, equipped with 16 GB of RAM. Through the experiments, we investigate the following:

1. *Benchmark Studies*: First we consider several TO benchmark problems, and compare the topologies obtained through mesh-based SIMP, TOuNN, and proposed Fourier-TOuNN.
2. *Comparison Studies*: Next, we compare the computed topologies using Fourier-TOuNN against those obtained using other length scale strategies.

Algorithm 1 Fourier-TOuNN

```
procedure TopOpt(NN, $\Omega^h$, $V^*$, $l_{min}$, $l_{max}$)    ▷ NN, discretized domain, desired vol, length scales
    $\boldsymbol{x} = \{x_e, y_e\}_{e \in \Omega^h}$ or $\{x_e, y_e, z_e\}_{e \in \Omega^h}$  ; $\boldsymbol{x} \in \mathbf{R}^{n_e \times d}$    ▷ center of elements in FE mesh; optimization input
    epoch = 0; $\alpha = \alpha_0$; $p = p_0$    ▷ Penalty factor initialization
    $J_0 \leftarrow FEA(\boldsymbol{\rho} = v_f^*, \Omega^h)$    ▷ FEA with uniform gray
    $f \sim \mathcal{U}(\frac{h}{l_{max}}, \frac{h}{l_{min}})$  ; $f \in \mathbf{R}^{d \times n_f}$    ▷ Uniformly sampled frequencies
    repeat    ▷ Optimization (Training)
        $\hat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{x}; f)$  ; $\hat{\boldsymbol{x}} \in \mathbf{R}^{n_e \times 2n_f}$    ▷ Project from Euclidean to Fourier
        $\boldsymbol{\rho} = NN(\hat{\boldsymbol{x}})$    ▷ Call NN to compute $\boldsymbol{\rho}$ at $\boldsymbol{p}$; (Forward propagation)
        $J_e \leftarrow FEA(\boldsymbol{\rho}, \Omega^h)$    ▷ Solve FEA
        L = $\frac{\sum_e \rho_e^p J_e}{J_0} + \alpha \left( \frac{\sum_e \rho_e v_e}{V^*} - 1 \right)^2$    ▷ Loss function
        Compute $\nabla L$    ▷ Sensitivity(Backward Propagation)
        $\boldsymbol{w} \leftarrow \boldsymbol{w} + \Delta \boldsymbol{w}(\nabla L)$    ▷ Adam optimizer step
        $\alpha \leftarrow \min(\alpha_{max}, \alpha + \Delta\alpha)$    ▷ Increment $\alpha$
        $p \leftarrow \min(p_{max}, p + \Delta p)$    ▷ Continuation
        epoch ← epoch + 1
    until $\epsilon_g < \epsilon_g^*$    ▷ Check for convergence
```

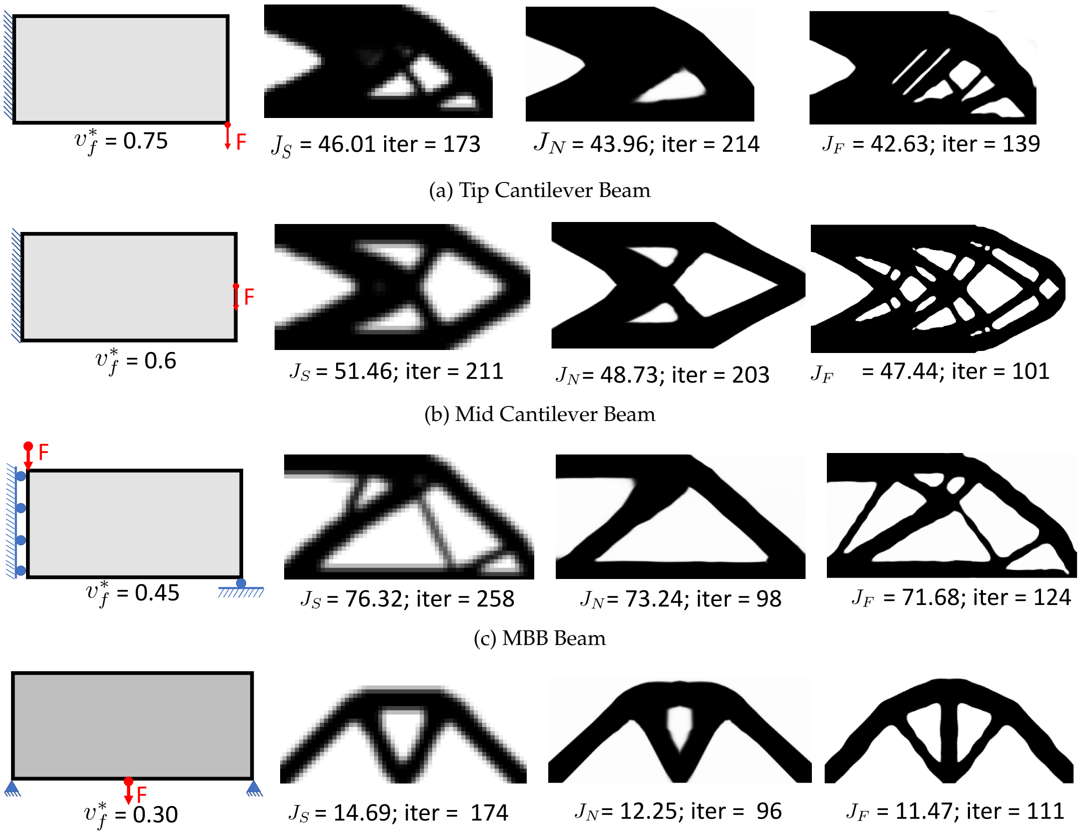
Parameter	Description and default value
E, ν	Isotropic material constants Young's Modulus (E = 1) and Poisson's ratio ($\nu = 0.3$)
NN	Fully connected feed-forward network with one layer (ReLU) with 20 Neurons.
n_f	Number of frequencies in projection space = 250
α	Constraint penalty updated every iteration ($\alpha_0 = \Delta\alpha = 0.05$)
p	SIMP penalty updated every iteration ($p_0 = 1$; $\Delta p = 0.02$; $p_{max} = 8$)
lr	Adam Optimizer learning rate 0.01
nelx, nely	Number of mesh elements of (60, 30) along x and y respectively
ϵ_g^*	Convergence criteria requiring fraction of gray elements be less than $\epsilon_g^* = 0.015$

Table 1: Default simulation parameters.

3. *Convergence Study*: The impact of the Fourier projection on the convergence of a distributed load problem is investigated.
4. *Minimum Length-Scale Control*: Here, we vary the minimum length scale (l_{min}) and study its effect on the topology, while keeping the maximum length scale (l_{max}) a constant.
5. *Maximum Length-Scale Control*: Similarly, we will vary the maximum length scale (l_{max}) and study its effect on the topology, while keeping the minimum length scale (l_{min}) a constant.
6. *Single Length-Scale Control*: Next, we vary both l_{min} and l_{max}, but keeping them equal, and study the impact on the topology.
7. *Number of Frequenct Samples*: Next, we vary the number of frequency terms n_f in the projected frequency space to understand its effects.
8. *Depth of Neural Net*: We vary the depth of NN to understand the nature of frequency spreading.
9. *Boundary Extraction*: We demonstrate the extraction of high resolution boundaries in the proposed framework.
10. *3D Example*: Finally, we illustrate the proposed method in 3D for the optimization of a thermal problem.

5.1 Benchmark Studies

First we consider four benchmark topology optimization problems illustrated in the first column of Figure 7, together with the desired volume fraction. The default mesh size is 60 × 30. The second column illustrates the topologies obtained using the popular 88-line implementation of SIMP [2], with a filtering radius of 2.0 (without length scale control). The final compliance values (after density projection), and the number of finite element iterations are also listed. The third column illustrates the topologies obtained using TOuNN [5], using a neural net size of 4 × 20 on the same mesh, once again without length scale control. The fourth column are the topologies obtained through the proposed Fourier-TOuNN algorithm, with $l_{min} = 4$ and $l_{max} = 40$. As one can observe, the topologies computed through Fourier-TOuNN exhibit thin features, and the compliance values are lower (better) than both SIMP and TOuNN.


v_f^* = 0.75
F
J_S = 46.01 iter = 173
J_N = 43.96; iter = 214
J_F = 42.63; iter = 139
F
v_f^* = 0.6
J_S = 51.46; iter = 211
J_N= 48.73; iter = 203
J_F = 47.44; iter = 101
F
v_f^* = 0.45
J_S = 76.32; iter = 258
J_N= 73.24; iter = 98
J_F = 71.68; iter = 124
F
v_f^* = 0.30
J_S = 14.69; iter = 174
J_N= 12.25; iter = 96
J_F = 11.47; iter = 111

(a) Tip Cantilever Beam

(b) Mid Cantilever Beam

(c) MBB Beam

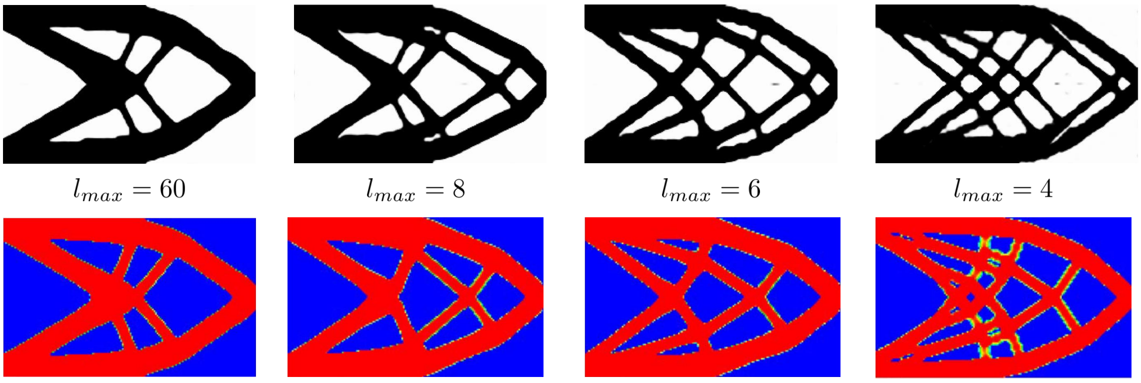
(d) Michell Beam

Fig. 7: Comparison of topologies for various benchmark problems using SIMP [35], TOuNN [5] and proposed Fourier-TOuNN.

5.2 Comparison Studies

Next, we consider the length scale results presented in [12] for a mid-cantilever beam with a mesh of 160 × 100. With a target volume fraction of 0.5, the minimum length scale was kept constant at $l_{min} = 2$ while l_{max} was varied. Figure 8 illustrates the computed topologies using the proposed method (top row) and those reported in [12] (bottom row). In both methods, as l_{max} is decreased, one can observe more structural members beginning to appear. The compliance values are not compared here since it was not reported in [12].

Next we consider the results presented [41] for an MBB beam with a mesh of 300 × 100 elements, and a desired volume fraction of $V^* = 0.5$. The topology obtained without length scale control from [41] is illustrated in Figure 9a. Figure 9b illustrates the topology obtained through Fourier-TOuNN with a relaxed length scale control of $l_{min} = 3$ and $l_{max} = 300$. Figure 9 c is the reported topology[41] with a tight and exact length control of $l_{min} = 5$ and $l_{max} = 8$,

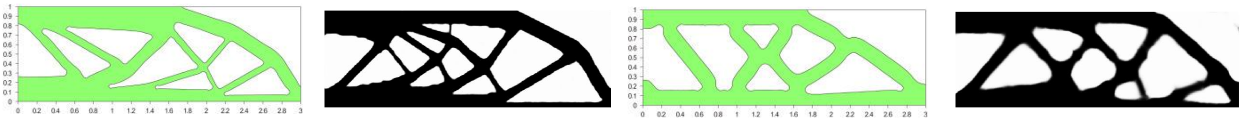


$l_{max} = 60$ $l_{max} = 8$ $l_{max} = 6$ $l_{max} = 4$

$l_{max} = NA$ $l_{max} = 8$ $l_{max} = 6$ $l_{max} = 4$

Fig. 8: Benchmark comparison of designs from [12] and proposed method. The maximum length scale is decreased (left to right) with $l_{min} = 2$ (NA = Not Applicable)

while Figure 9d illustrates the Fourier-TOuNN topology with the same length control of $l_{min} = 5$ and $l_{max} = 8$. We observe that length scale control increases the compliance as expected. and more so, for exact length scale control.



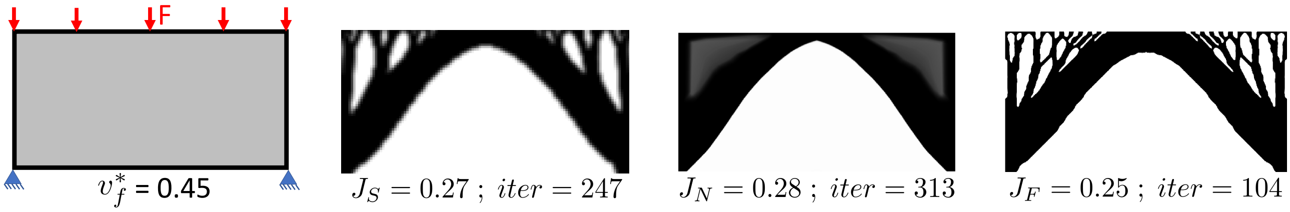
$J^* = 180$ $J = 173$ $J^* = 232$ $J = 216$

(a) (b) (c) (d)

Fig. 9: (a) No length scale control [41]. (b) Relaxed length scale control in Fourier-TOuNN with $l_{min} = 3$ and $l_{max} = 100$. (c) Tight length scale control [41] with $l_{min} = 5$ and $l_{max} = 8$. (d) Tight length scale control in Fourier-TOuNN with $l_{min} = 5$ and $l_{max} = 8$.

5.3 Convergence

The distributed load problem in Figure 10 is considered next since it necessary entails thin features [22] near the loaded edge. While SIMP [35] converges to a topology with thin features, TOuNN [5] does not converge to a 0/1 design. On the other hand, Fourier-TOuNN rapidly converges to a topology exhibiting thin features, with a compliance that is lower than both SIMP and TOuNN.



$J_S = 0.27$; $iter = 247$ $J_N = 0.28$; $iter = 313$ $J_F = 0.25$; $iter = 104$

Fig. 10: Distributed load problem.

The difference in convergence between TOuNN and Fourier-TOuNN is illustrated in Figure 11, with the fraction of gray elements ($\epsilon_g = N_{gray}/N_{total}$) on the y axis. The rapid convergence can be attributed to Fourier-TOuNN's ability to capture a range of spatial frequencies easily.

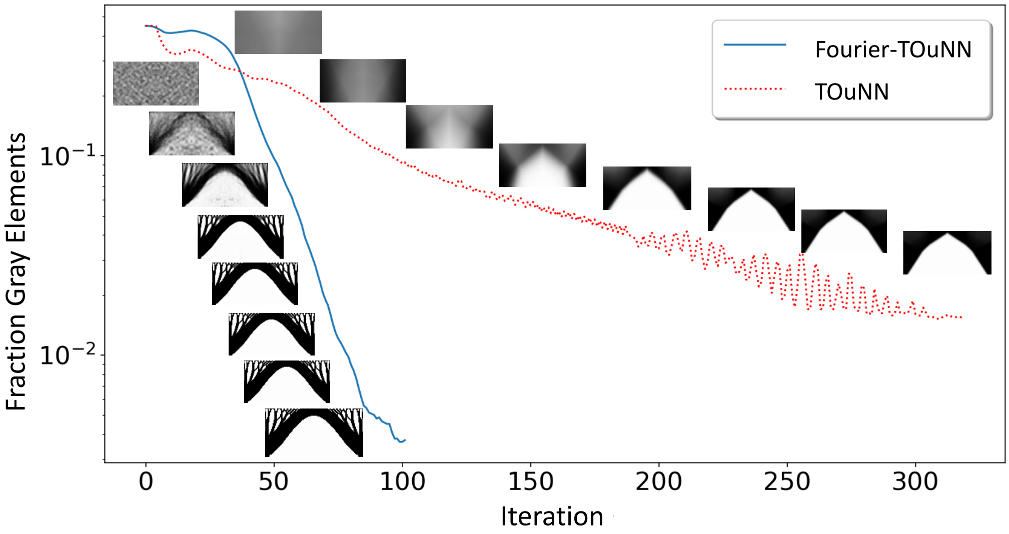


Fig. 11: Convergence graphs for distributed beam at $V_f = 0.45$ using TOuNN [5] and proposed Fourier enhanced TOuNN.

5.4 Minimum Length-Scale Control

Here, we consider again the mid-cantilever beam in Figure 7b, and vary l_{min} between 3 and 55, while keeping l_{max} constant at $l_{max} = 60$. The topologies obtained using Fourier-TOuNN for various volume fractions are illustrated in Figure 12. Observe that for a given volume fraction, i.e. for each row, as l_{min} is increased, the topology exhibits thicker features. Further the compliance increases (gets worse) as l_{min} approaches l_{max}, i.e., as the solution space shrinks.

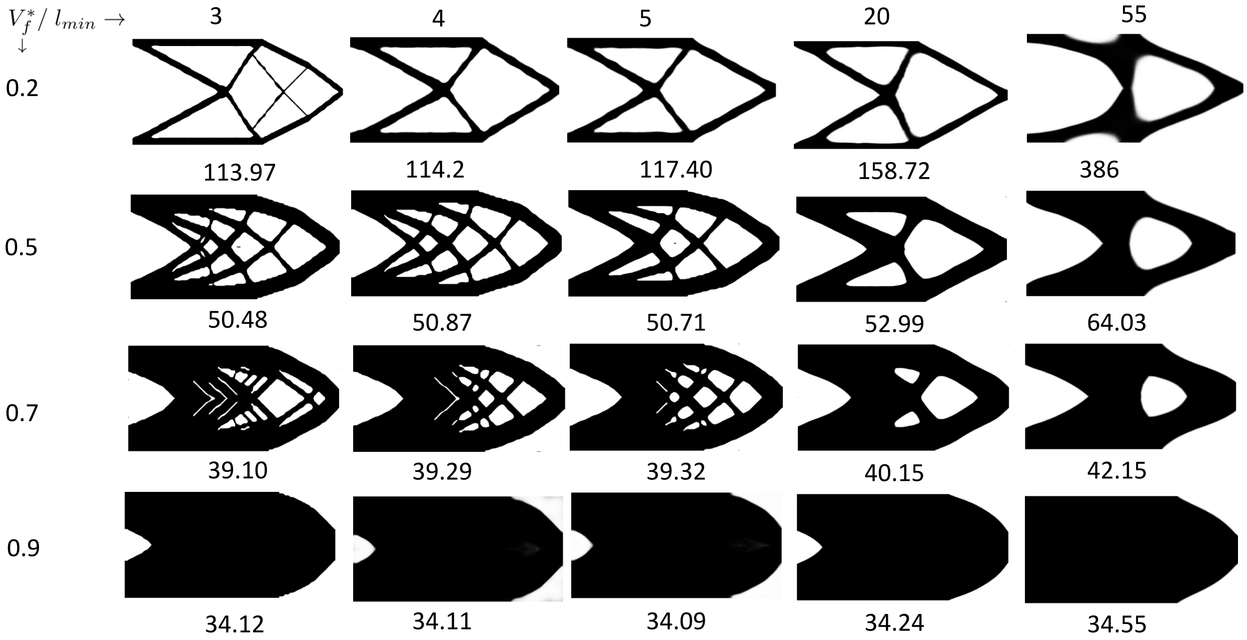


Fig. 12: Varying minimum length scale with $l_{max} = 60$.

5.5 Maximum Length-Scale Control

Next, once again considering the mid-cantilever beam, we vary l_{max} between 4 and 30, while keeping l_{min} constant at $l_{min} = 4$. The topologies obtained using Fourier-TOuNN for the mid-cantilever beam are illustrated are illustrated in Figure 13 for various volume fractions. Similar to the previous experiment, for a given volume fraction, i.e., for each row, as l_{max} is increased, the topologies exhibit thicker features. Further, the compliance decreases (improves) as l_{max} moves away from l_{min}, i.e., as the solution space expands.

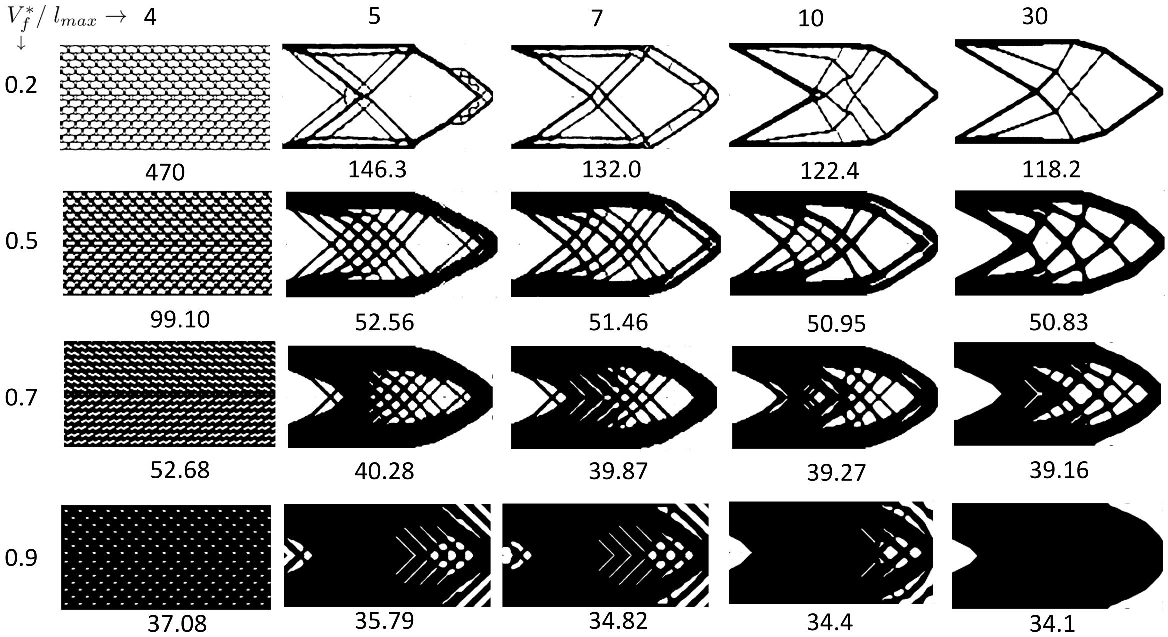
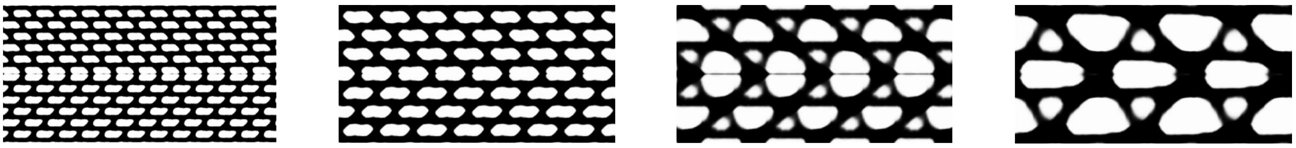


Fig. 13: Varying maximum length scale with $l_{min} = 4$

5.6 Single Length-Scale Control

In the previous experiment, one can observe that, in the first column of Figure 13, when $l_{min} = l_{max} = 4$, the topologies exhibit a repeated pattern. Indeed, one can obtain repeated patterns of various length scales by setting $l_{min} = l_{max} = l$. This is illustrated in Figure 14. Such patterns are often used as fillers in additive manufacturing [9, 19], and have been obtained by other researchers using various constraint techniques [38], [39], [40].



$_{nin} = l_{max} = 5;\ J = 90.69$ $l_{min} = l_{max} = 8;\ J = 92.49$ $l_{min} = l_{max} = 14;\ J = 99.79$ $l_{min} = l_{max} = 20;\ J = 96.36$

Fig. 14: Repeated patterns for a mid-cantilever beam ($V_f = 0.45$) when $l_{min} = l_{max}$.

5.7 Number of Frequency Samples

Thus far we have used a default of 250 frequency terms (n_f) within the range defined by l_{min} and l_{max}. In this experiment, we will vary n_f and study its effect on the topology. In particular, we consider the L-bracket problem illustrated in Figure 15. The length scales were kept constant at $l_{min} = 4$ and $l_{max} = 60$.

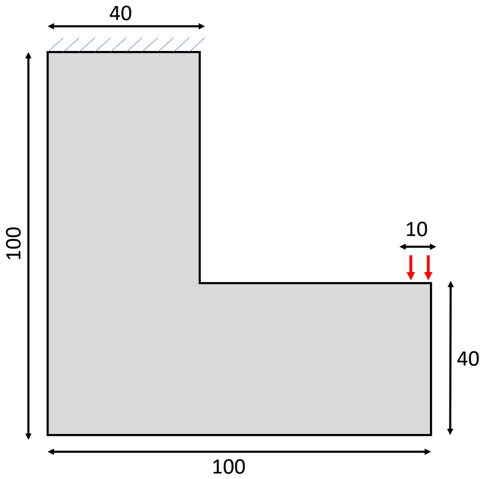


Fig. 15: L-Bracket with load.

The topologies obtained by varying n_f are illustrated in Figure 16. Using very few frequencies leads to disconnected topology. However, about 100 frequency terms are sufficient for convergence in compliance. This is typical of what was observed in other experiments. This experiment also demonstrates that the proposed method is applicable to non-rectangular domains as well.

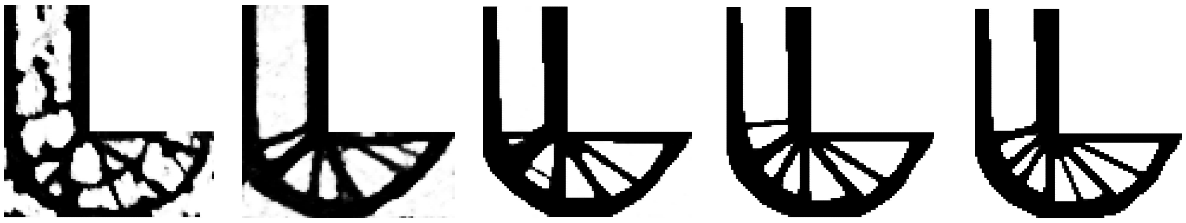
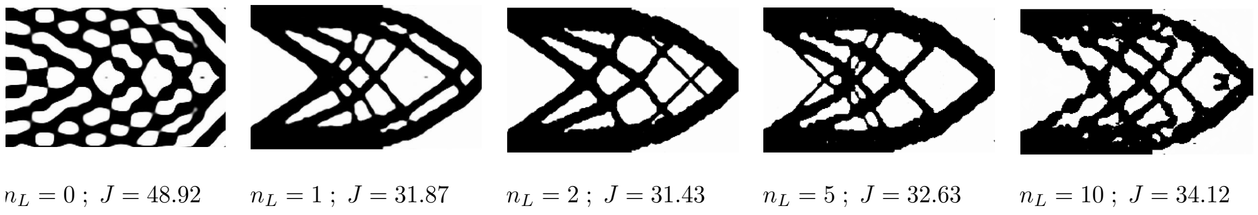

$n = 10$; $J = 170.63$ $n = 20$; $J = 147.86$ $n = 50$; $J = 135.40$ $n = 100$; $J = 133.63$ $n = 250$; $J = 133.96$

Fig. 16: Effect of number of frequency terms n_f on the topology.

5.8 Depth of Neural Net

In all the experiments, we have used a NN with one layer. Here, we briefly consider the effect of the number of layers n_L on the topologies for the mid-cantilever beam. As can be observed in Figure 17, $n_L = 0$ results in a poor topology. In other words, the non-linearity introduced by the LeakyReLU functions is essential. Further, for $n_L \geq 5$, once again the topology begins to degrade. In practice, we observed that $n_L = 1$, or $n_L = 2$ leads to robust convergence.



$n_L = 0\ ;\ J = 48.92$ $n_L = 1\ ;\ J = 31.87$ $n_L = 2\ ;\ J = 31.43$ $n_L = 5\ ;\ J = 32.63$ $n_L = 10\ ;\ J = 34.12$

Fig. 17: Effect of the NN's depth on the topology.

5.9 High Resolution Boundary Extraction

One of the advantages of TOuNN [5] is the ability to recover high resolution topology. This characteristic carries over to Fourier-TOuNN, and is illustrated using consider a thermal TO problem in Figure 18, where $Q = 0.001$, the Dirichlet value at the heat sink is 0, and the desired volume fraction is 0.4.

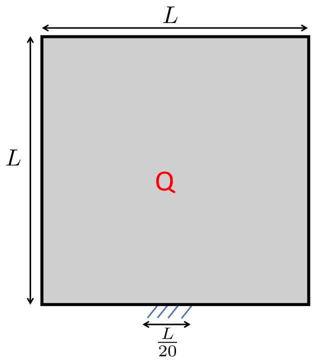


Fig. 18: Square plate with distributed heat load and heat sink.

The computed topology using Fourier-TOuNN on a mesh size of 100×100 is illustrated in the first column of Figure 19). After optimization, the density field is re-sampled using the optimized weights on a 500×500 mesh, yielding the topology in the second column. Further, observing that the density are calculated through forward propagation of analytic functions, one could back-propagate to obtain the derivative of the density field thereby obtaining a well defined analytic boundary as illustrated in the third column. This example highlights one of the main advantages of the Fourier-TOUNN framework: *a simple length scale control strategy, while enabling recovery of a high-resolution boundary.*

5.10 3D Example

The proposed framework can be easily extended to 3D. The number of input neurons is extended to 3, and a 3D finite element solver is used for analysis. For illustration, we consider a cubic block in 3D with a heat sink and a distributed internal load as shown in Figure 20. Once again, $Q = 0.001$, the Dirichlet value at the heat sink is 0, and the desired volume fraction is 0.4. We perform the experiment with varying l_{min}, while keeping a fixed l_{max} (similar to Section 5.4).

The results are illustrated in Figure 21. Once again, we observed an increase in compliance as the range of length scale is decreased.

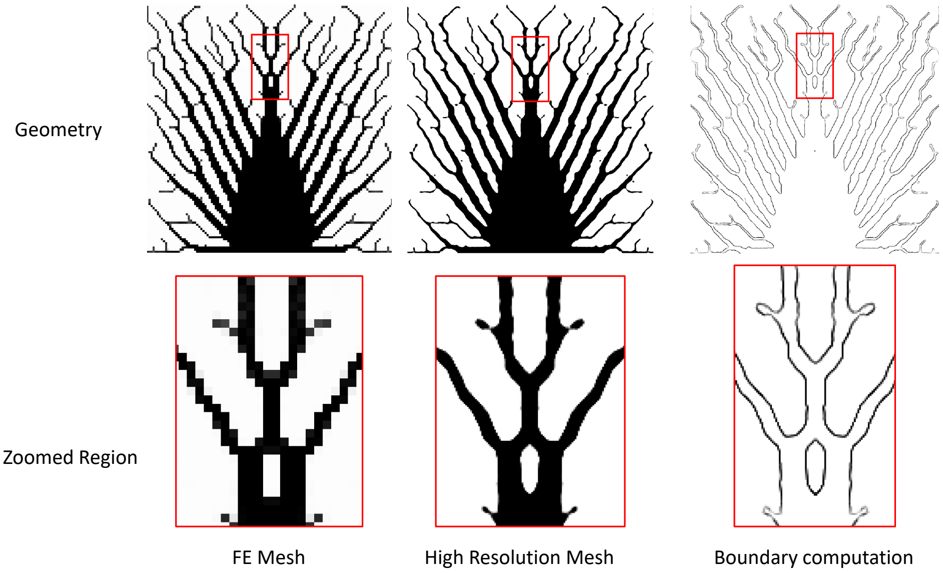


Fig. 19: High resolution interpretation of geometry and boundary extraction.

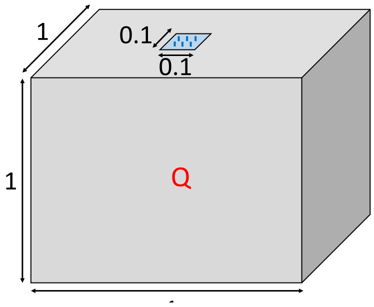
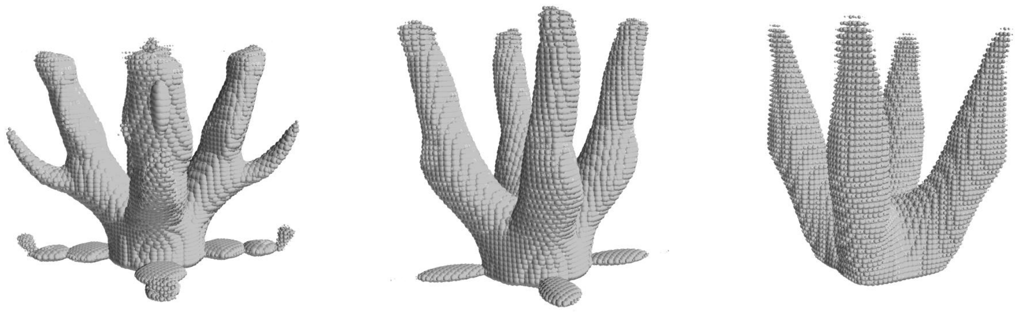


Fig. 20: Domain for thermal optimization in 3D.



$l_{min} = 6$; $l_{max} = 50$; $J = 21.78$ $l_{min} = 8$; $l_{max} = 50$; $J = 23.55$ $l_{min} = 12$; $l_{max} = 50$; $J = 26.32$

Fig. 21: Thermal optimization with varying l_{min}.

6 Conclusion

In this paper, we presented a simple length scale control strategy by extending the recently proposed TOuNN method [5]. The extension relied on a Fourier projection; no additional constraints were required. The mesh independent density representation permitted extraction of high resolution boundaries of topologies with thin features. Further, since the sensitivity computations can be automated through the neural network's backpropagation, the method is easy to implement. Several numerical experiments were presented in 2D and 3D to characterize the proposed method. We noted that the proposed length control strategy is approximate. Future work will explore exact length scale control within this framework.

7 Replication of results

The Python code pertinent to this paper is available at www.ersl.wisc.edu/software/FourierTOuNN.zip.

Acknowledgments

The authors would like to thank the support of National Science Foundation through grant CMMI 1561899. Prof. Krishnan Suresh is a consulting Chief Scientific Officer of SciArt, Corp.

Compliance with ethical standards

The authors declare that they have no conflict of interest.

References

1. Martin S Andersen, Joachim Dahl, and Lieven Vandenberghe. Cvxopt: A python package for convex optimization. *abel. ee. ucla. edu/cvxopt*, 2013.
2. Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S. Lazarov, and Ole Sigmund. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, jan 2011.
3. Atılım Günes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(1):5595–5637, 2017.
4. M P Bendsoe and Ole. Sigmund. *Topology optimization: theory, methods, and applications*. Springer Berlin Heidelberg, 2 edition, 2003.
5. Aaditya Chandrasekhar and Krishnan Suresh. TOuNN: Topology optimization using neural networks. *Structural and Multidisciplinary Optimization*, October 2020.
6. Shikui Chen, Michael Yu Wang, and Ai Qun Liu. Shape feature control in structural topology optimization. *Computer-Aided Design*, 40(9):951–962, 2008.
7. Joshua D Deaton and Ramana V Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014.
8. Eduardo Fernández, Kai ke Yang, Stijn Koppen, Pablo Alarcón, Simon Bauduin, and Pierre Duysinx. Imposing minimum and maximum member size, minimum cavity size, and minimum separation distance between solid members in topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 368:113157, aug 2020.
9. Wei Gao, Yunbo Zhang, Devarajan Ramanujan, Karthik Ramani, Yong Chen, Christopher B. Williams, Charlie C.L. Wang, Yung C. Shin, Song Zhang, and Pablo D. Zavattieri. The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design*, 69:65–89, 2015.
10. Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.
11. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
12. James K. Guest. Imposing maximum length scale in topology optimization. *Structural and Multidisciplinary Optimization*, 37(5):463–473, 2009.

13. James K Guest, Jean H Prévost, and Ted Belytschko. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International journal for numerical methods in engineering*, 61(2):238–254, 2004.

14. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd International Conference on Machine Learning, ICML 2015*, volume 1, pages 448–456. International Machine Learning Society (IMLS), feb 2015.

15. Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, dec 2015.

16. Tej Kumar and Krishnan Suresh. Direct lagrange multiplier updates in topology optimization revisited. *Structural and Multidisciplinary Optimization*, pages 1–16, oct 2020.

17. Boyan S. Lazarov and Fengwen Wang. Maximum length scale in density based topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 318:826–844, may 2017.

18. Boyan S. Lazarov, Fengwen Wang, and Ole Sigmund. Length scale and manufacturability in density-based topology optimization. *Archive of Applied Mechanics*, 86(1-2):189–218, jan 2016.

19. Jikai Liu, Andrew T Gaynor, Shikui Chen, Zhan Kang, Krishnan Suresh, Akihiro Takezawa, Lei Li, Junji Kato, Jinyuan Tang, Charlie CL Wang, et al. Current and future trends in topology optimization for additive manufacturing. *Structural and Multidisciplinary Optimization*, pages 1–27, 2018.

20. Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019.

21. Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

22. Francesco Mezzadri, Vladimir Bouriakov, and Xiaoping Qian. Topology optimization of self-supporting support structures for additive manufacturing. *Additive Manufacturing*, 21:666–682, may 2018.

23. Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*, 2020.

24. Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

25. Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

26. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

27. Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxlcr, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *36th International Conference on Machine Learning,*

ICML 2019, volume 2019-June, pages 9230–9239, jun 2019.

28. Susana Rojas-Labanda and Mathias Stolpe. Automatic penalty continuation in structural topology optimization. *Structural and Multidisciplinary Optimization*, 52(6):1205–1221, dec 2015.
29. G. I.N. Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 37(3):217–237, jan 2009.
30. Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
31. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
32. J. A. Sethian and Andreas Wiegmann. Structural Boundary Design via Level Set and Immersed Interface Methods. *Journal of Computational Physics*, 163(2):489–528, Sep 2000.
33. O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(1):68–75, 1998.
34. Ole Sigmund. On the design of compliant mechanisms using topology optimization. *Journal of Structural Mechanics*, 25(4):493–524, 1997.
35. Ole Sigmund. A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, 21(2):120–127, 2001.
36. Krister Svanberg. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.
37. Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.
38. Jun Wu, Niels Aage, Ruediger Westermann, and Ole Sigmund. Infill Optimization for Additive Manufacturing – Approaching Bone-like Porous Structures. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1127–1140, 2018.
39. Jun Wu, Anders Clausen, and Ole Sigmund. Minimum compliance topology optimization of shell–infill composites for additive manufacturing. *Computer Methods in Applied Mechanics and Engineering*, 326:358–375, Nov 2017.
40. Jun Wu, Charlie CL Wang, Xiaoting Zhang, and Rüdiger Westermann. Self-supporting rhombic infill structures for additive manufacturing. *Computer-Aided Design*, 80:32–42, 2016.
41. Qi Xia and Tielin Shi. Constraints of distance from boundary to skeleton: For the control of length scale in level set based structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 295:525–542, oct 2015.
42. Weisheng Zhang, Wenliang Zhong, and Xu Guo. An explicit length scale control approach in SIMP-based topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 282:71–86, dec 2014.
43. Benliang Zhu and Xianmin Zhang. A new level set method for topology optimization of distributed compliant mechanisms. *International journal for numerical methods in engineering*, 91(8):843–871, 2012.