Design for Machinability: Topology Optimization for Subtractive Machining Using Trimmed Neural Fields

HONGRUI CHEN, Carnegie Mellon University, USA
LEVENT BURAK KARA*, Carnegie Mellon University, USA

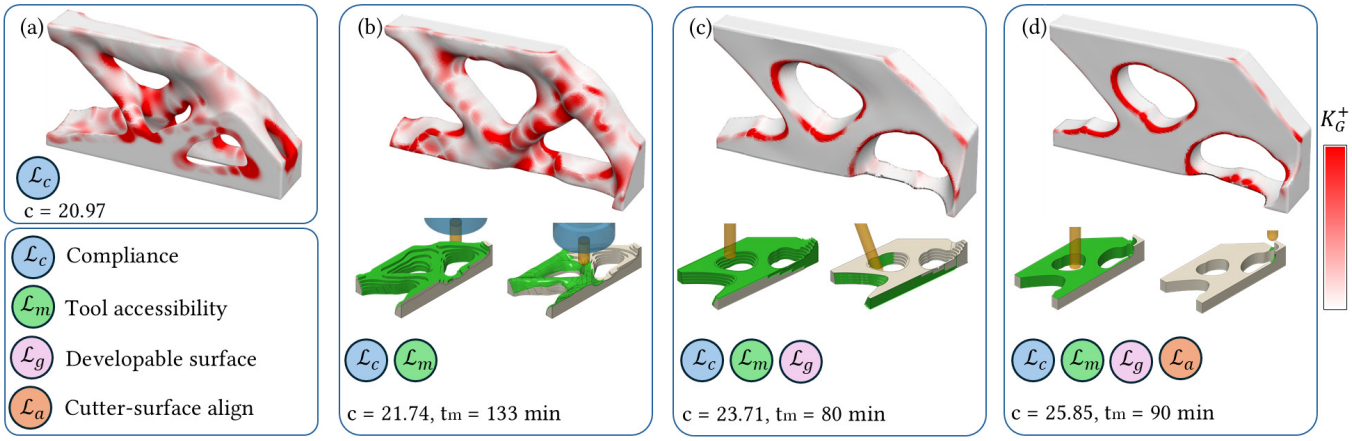


Fig. 1. Our method for topology design for efficient machining using developable surfaces. (a) Topology optimization that minimizes compliance only. (b) Topology optimization with machining tool accessibility taken into account. Although machinable, such designs require time-consuming point milling to obtain an accurate realization of the surfaces. (c) Our method for generating developable surfaces is geared toward efficient flank milling. (d) A special case of our method tailored for 3-axis milling. Each design achieves a final volume fraction of 0.3, satisfying the constraint. The color map represents the absolute Gaussian curvature. Left-to-right: While machining becomes more efficient, geometries become increasingly more restricted (to accommodate machining goals), as reflected in the trade-off observed in the compliance.

Topology optimization has found significant utility in additive manufacturing, but the resulting geometries are often difficult to realize using subtractive methods. The complex, curved surfaces of optimized designs typically require time-consuming point milling operations. In contrast, geometries composed of developable surfaces are well suited to flank milling, enabling substantial improvements in machining efficiency and surface quality. However, globally minimizing Gaussian curvature leads to overly simplistic designs. We address this challenge with a novel method that represents geometry using three implicit neural fields, optimized to produce machinable components. The resulting designs consist of regions of developable surfaces separated by seamlines where curvature is concentrated. Seamlines are efficiently computed as intersections of the implicit fields, while the geometry emerges through their product. By constraining surfaces to align with the cutter's normal or tangential direction, our approach further accommodates practical constraints of 3-axis machining. We demonstrate concurrent optimization of topology and Gaussian curvature, enabling control over seamline feature scale. Compared to methods that enforce subtractive machinability without considering milling efficiency, our approach consistently reduces machining time while maintaining competitive structural performance.

Additional Key Words and Phrases: Topology Optimization, Developable Surface, Subtractive Manufacturing

1 Introduction

Topology optimization has emerged as a transformative tool for engineering design, enabling the creation of lightweight, high-performance structures by optimizing material distribution within a given design space. However, the intricate geometries generated by topology optimization often pose significant challenges for subtractive manufacturing processes such as milling. These geometries typically demand extended machining times and complex toolpaths to achieve acceptable surface finishes, making them poor choices for machinability. A promising alternative lies in leveraging developable surfaces, surfaces with zero Gaussian curvature, which are compatible with the highly efficient flank milling technique. Flank milling sweeps the cutter along the surface, which enables the creation of high-quality surface finishes while significantly reducing machining time [He et al. 2021]. Compared to conventional point milling, which only uses the tip of the cutting tool, flank milling offers a pathway to produce optimized designs that balance complexity and manufacturing efficiency. Despite this potential, no comprehensive method exists for topology design that accounts for the desire to produce developable surfaces.

Current approaches that target topology optimization for subtractive manufacturing can be broadly categorized into those that rely on restrictive geometric constraints, post-processing techniques, or hybrid optimization methods. Methods that restrict designs to 2D extrusions or hierarchical 2.5D topologies, as seen in Autodesk Fusion 360's topology optimization [Autodesk, Inc. 2024], provide manufacturable designs but at the cost of reduced design freedom. Similarly,

*Corresponding author.

post-processing techniques attempt to minimally modify topology-optimized geometries into subtractive-compatible geometries but do not account for structural performance or load requirements during post-processing [Stein et al. 2018]. Other methods aim to incorporate subtractive constraints directly into topology optimization [Langelaar 2019; Mirzendehdel et al. 2020] but often neglect machining efficiency, leading to designs with complex geometries that still require extensive machining time. In this work, we build upon these advances and describe a method that takes into account machining efficiency during topology optimization.

A key to achieving our goal lies in the ability to produce designs with developable surfaces in a computationally tractable way. Directly minimizing the Gaussian curvature everywhere is ineffective, as the resulting shapes would consist of largely elementary building blocks. Moreover, computing accurate curvatures on finite element representations of topology-optimized shapes presents significant numerical challenges, particularly when generating the gradients of the objective functions. Additionally, generating designs whose surfaces align with subtractive manufacturing constraints, such as tool orientation and machinability, further complicates the problem.

We present a new approach based on implicit neural fields to address the challenges of incorporating developable surface constraints into the topology optimization process (Figure 1). Our approach takes as input domain coordinates and outputs a density value at each coordinate, with the final design emerging as the intersection of three implicit fields. This representation enables the computation of analytical Gaussian curvatures. The resulting geometry is characterized by regions of developable surfaces that join across seamlines where high curvature and directional changes are concentrated. By controlling the feature size of the implicit fields, our approach can regulate the spatial distribution of seamlines across the design domain, thereby facilitating the downstream conversion of implicit fields to solid geometries and the subsequent toolpath creation. Together with machinability constraints, our approach can create designs geared toward 5-axis flank milling to efficiently attain high geometric fidelity. Additionally, our method can be tailored to 3-axis milling by incorporating mechanisms to control surface orientation, aligning surface normals with either tangential or normal cutting directions. By integrating these features, our approach ensures machining efficiency while maintaining design flexibility. We evaluate our approach across multiple case studies, comparing curvature, structural performance, and machining times. The results demonstrate that our approach consistently reduces machining times while incurring relatively small trade-offs in structural performance.

Our main contributions are:

- An approach to minimizing Gaussian curvature to generate developable surfaces suitable for efficient 5-axis machining.
- A novel tri-implicit field formulation to represent geometry, enabling precise seamline modeling and curvature concentration for machinable designs.
- An extension of curvature minimization techniques to align geometries with principal directions, enabling practical designs for 3-axis machining.

2 Previous Work

Our review focuses on topology optimization with implicit neural representations, topology optimization for subtractive manufacturing, and developable surfaces.

Implicit neural representations: Implicit neural representations model geometry as an implicit function. Compared to mesh-based representations, neural fields can compactly encode shapes without the need for discretization while being infinitely differentiable. Prior work explored improving the representation capability of neural implicit fields [Atzmon et al. 2019; Deng et al. 2020; Fan and Musialski 2024; Gropp et al. 2020; Marschner et al. 2023; Petrov et al. 2024]. Seamline modeling using implicit fields to represent vortex filaments [Ishida et al. 2022]. Neural implicit representation also facilitates geometric deformations [Božič et al. 2021; Chen et al. 2021]. Developments in implicit fields have also advanced high-quality rendering techniques [Sharp and Jacobson 2022; Takikawa et al. 2021; Wang et al. 2023], leveraging their ability to represent complex geometry compactly and with high precision. Additionally, applications have extended to font rendering, where three implicit fields are utilized for efficient and detailed representation [Chlumskỳ et al. 2018]. The differentiability of implicit fields enables accurate computation of normals and curvature, which is critical for downstream geometric processing tasks [Berzins et al. 2023; Yang et al. 2021].

Topology optimization with machine learning: Topology optimization is a computational design methodology that iteratively distributes material within a given domain to meet specified performance objectives and constraints. Coupling finite element analysis with advanced optimization algorithms enables the creation of lightweight and structurally efficient solutions suitable for a broad range of engineering applications. Classical topology optimization includes SIMP [Bendsøe 1989; Zhou and Rozvany 1991] and the level-set [Allaire et al. 2002] methods. Recent advances have enabled the generation of increasingly complex and intricate structures, with the resulting shapes being naturally compatible with additive manufacturing [Bickel et al. 2010; Liu et al. 2018; Zhu et al. 2017]. Recent approaches to integrating machine learning with topology optimization include data-driven topology optimization and direct topology optimization methods. In data-driven topology optimization, a neural network aims to learn from a database of already optimized designs. The neural network models include convolutional [Aage et al. 2015; Banga et al. 2018; Behzadi and Ilieş 2021; Nakamura and Suzuki 2020; Zheng et al. 2021], generative adversarial [Nie et al. 2021; Yu et al. 2019] and diffusion-based approaches [Mazé and Ahmed 2023]. All these data-driven approaches focus on reducing the time required for optimal topology prediction. Although there have been improvements over the years, producing a widely applicable training set spanning many different boundary conditions is not straightforward, especially for 3D models.

Another approach to neural topology optimization involves direct methods that do not use prior data but rather use a neural network to optimize the material distribution in the design domain [Chandrasekhar and Suresh 2021b, 2022; Deng and To 2020, 2021; Han et al. 2024; Hoyer et al. 2019; Zehnder et al. 2021]. Such direct methods use spatial coordinates as input and produce an implicit field to

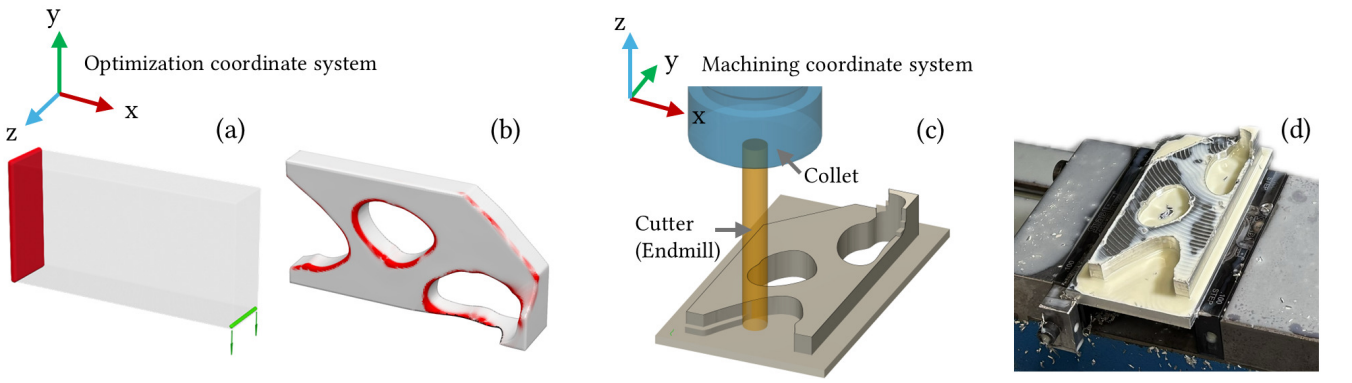


Fig. 2. (a) Original design domain together with zero-displacement (red) and force (green) boundary conditions. (b) The designed part with an imposed 3-axis milling configuration. (c) The cutter relative to the workpiece in a 3-axis milling configuration. Note the coordinate axes relative to the cutter. (d) Part being physically machined.

represent the shape. By manipulating the spatial resolution of the input coordinates, length-scale control is achieved [Chandrasekhar and Suresh 2022]. Through automatic differentiation, the analytical density gradient can be accurately computed and integrated into optimization [Chen et al. 2023]. In this work, we build on these advances and use a neural implicit field for compact Hessian and Gaussian curvature calculations.

Subtractive manufacturing oriented topology optimization: Subtractive machining is a widely used method in manufacturing. Starting from a block of material, it involves a cutting tool that shaves off material until the desired geometry is reached. To date, subtractive manufacturing-oriented topology optimization methods have primarily sought to ensure that the cutting tool (mill) can access all points of the designed part's surface, taking into account criteria such as the mill and collet interference, toolpath developability and other geometric constraints [Chen et al. 2016; Langelaar 2019; Liu and Ma 2015; Liu and To 2017; Mirzendehdel et al. 2020; Morris et al. 2020; Vatanabe et al. 2016]. However, these methods do not consider machining efficiency, where resulting shapes typically require time-consuming point milling to produce the optimized part. Commercial software also demonstrated the stacking of multiple 2D extrusions to approximate designs that are amenable for 3-axis milling [Autodesk, Inc. 2024]. However, these extrusion-based methods further constrain the freedom of design. In this work, we retain the subtractive manufacturing constraints by Langelaar [Langelaar 2019] to ensure accessibility of the tool head while incorporating a developable surface that can be efficiently machined with flank milling.

Developable surfaces: Developable surfaces have zero Gaussian curvature everywhere and can be flattened onto a plane without distortion, reflecting their roots in classical differential geometry [Hilbert and Cohn-Vossen 2021; Yuan et al. 2023] They are generated using straight lines (rulings) [Chu et al. 2008; Wang and Elber 2014] and find utility in origami [Lee et al. 2018; Peraza-Hernandez et al. 2014; Turner et al. 2016], architecture [Nejur and Steinfeld 2016], and industrial design [Barda et al. 2023; Erdine et al. 2019; Kilian et al. 2008; Liu et al. 2011] due to their cost-effective production from sheet materials. If the original shape is developable, prior works on shape optimization [Liu and To 2020; Schulz et al. 2017], and beam-based [Morris et al. 2023; Zhao et al. 2021] can also produce developable surfaces. Developability optimization on shapes aims to facilitate the creation of geometries that can be efficiently manufactured [Rowe et al. 2024; Sellán et al. 2020; Stein et al. 2018]. Our work is inspired by the work of Stein et al. [Stein et al. 2018] that demonstrates how a conventional topology-optimized shape can be post-processed to make it amenable to flank milling. In this work, we extend this insight, where we impose surface developability as a goal during the *design* of the part rather than as a post-processing step. This allows our method to incorporate the knowledge of the boundary conditions and target mass criteria. To this end, we introduce a continuous neural representation of the geometry in the form of implicit fields, which facilitates differentiable geometry queries to drive topology optimization.

To the best of our knowledge, our approach represents the first attempt at combining topology optimization with developable surface generation using implicit neural fields, with the ultimate goal of making subtractive machining more efficient. The next sections detail our approach and demonstrate our results.

3 Methodology

Our approach aims to achieve concurrent topology optimization with developable surfaces to facilitate efficient subtractive machining. We limit our approach to structural compliance minimization under volume fraction constraints [Allaire et al. 2002; Bendsøe and Kikuchi 1988]. We build our approach based on a set of characteristics unique to subtractive machining, and the related objectives pertaining to its efficiency. We use an implicit neural representation to compute the Gaussian curvature, model seamlines through field intersections, and perform topology optimization. The optimization is performed by a neural network that handles gradient computations and backpropagation via automatic differentiation. This section details how these components integrate to optimize structural performance and improve manufacturability.

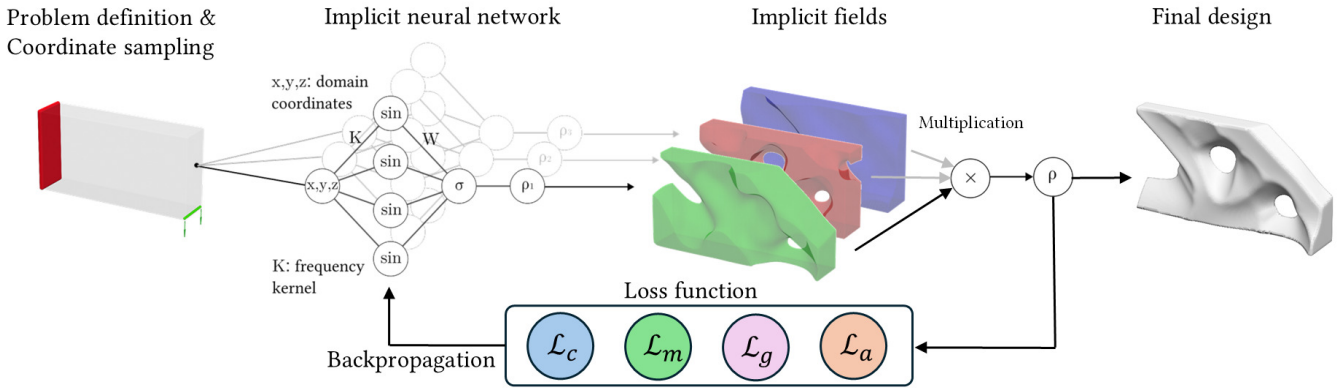


Fig. 3. The density field is represented as the product of three implicit fields. We sample coordinates across the design domain; the coordinates are then fed into three separate neural networks. The forward pass produces an output geometry. Various loss functions are calculated from this geometry and backpropagated to optimize the shape.

3.1 Subtractive Machining

A typical subtractive machining process consists of a roughing stage where the workpiece is cut close to the target geometry, followed by a finishing stage where surfaces and detailed features are machined to precision [Dong et al. 1993; Kim et al. 2013; Lee and Chang 1991; Li et al. 1994]. For roughing, a single machining orientation (fixturing) is commonly used as it is both time and resource-efficient. In particular, most roughing operations are conducted using 3-axis machining due to the ease of programming, collision management, and faster execution times [Jousselin et al. 2019]. Consistent with the conventions in practice, we use $z-$ to denote the cutter's linear axis of motion (Figure 2) and assume the part will be fixtured *only once* throughout the machining steps. Nonetheless, advanced methods that integrate roughing and finishing operations that utilize 5-axis joint roughing and finishing [Dutta et al. 2024; Jousselin et al. 2019; Mahdavi-Amiri et al. 2020] could be adopted for our Computer-Aided Manufacturing (CAM) simulations without loss of generality and without impacting our optimizer. We leave this option for future work. Additionally, in this work, we use Fusion 360 [Autodesk, Inc. 2024] as our CAM engine, which currently does not support 5-axis coordinated roughing.

For finishing, we take into account 5-axis and 3-axis machining, which are supported by our CAM engine. Our proposed developable surface generation permits 5-axis flank milling, which we simulate using finishing operations. This enables the generation of developable surfaces to be cut using 5-axis milling, as shown in Figure 1(c). Similar to roughing, we assume a single fixturing of the part, hence excluding solutions that may be generated through multiple fixturing of the workpiece. Additionally, we do not consider the cutter's diameter on feature reproducibility, or the size of the cutter/collet on potential tool-part interference.

3.2 Implicit neural representation

Multiple studies have explored direct topology optimization with neural networks [Chandrasekhar and Suresh 2021b, 2022]. The neural network replaces the finite element mesh as the geometry representation. During the optimization, the network is iteratively trained to produce a geometry that minimizes the prescribed loss function. In addition to representing geometry, the neural representation has the additional benefit of being an infinitely differentiable function. During each iteration, accurate gradients and Hessians can be computed through automatic differentiation. Furthermore, seamlines can be modeled as the intersections of the surfaces implied by the implicit neural networks.

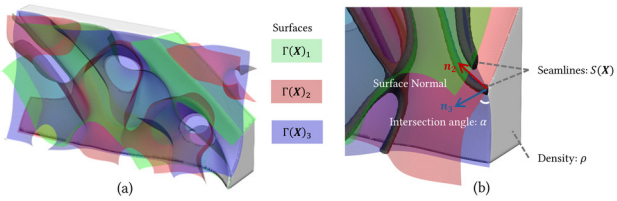


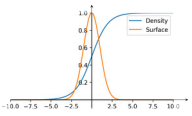
Fig. 4. (a) Surfaces generated by the three networks. (b) Seamlines $S(\mathbf{X})$ are formed at the intersections of surfaces. The seamlines are excluded from Gaussian curvature minimization. At a seamline, the two surfaces intersect at an angle α. Large values of α indicate sharper seams, and thus fewer regions excluded from curvature minimization.

Figure 3 illustrates the workflow. We employ a single-hidden-layer architecture that takes domain coordinates $\mathbf{X} = [x, y, z]$, scaled to [-1, 1] along the longest dimension, and maps them to 512 frequency kernels. These are then linearly combined to produce a single output channel representing the density value ρ. This results in 2048 trainable parameters per field, comprising $512 \times 3 = 1536$ kernel parameters and 512 linear weights. The network capacity can be adjusted by varying the number of parameters.

$$\rho(\mathbf{X}) = F(\mathbf{X}) = \prod_{i=1}^{n} \sigma(\mathbf{W}_i \sin(\mathbf{K}_i \mathbf{X} + 1)) \tag{1}$$

where:
ρ: Density field
$\mathbf{X}$: Domain coordinate input, $\mathbf{X} = (x, y, z)$
n: Number of implicit fields, we use 3
σ: Sigmoid activation function
$\mathbf{K}$: Trainable frequency kernels
$\mathbf{W}$: Trainable weights, initialized to 0

The network relies on the sinusoidal activation function, where similar implicit neural networks have been used for length scale control in topology optimization [Chandrasekhar and Suresh 2022]. The three implicit fields play identical functional roles in the formulation and are not tied to fixed, predefined portions of the geometry. Each field defines a candidate implicit surface, and the final solid is obtained through the product of the three field responses. Accordingly, an individual field contributes by shaping the patch arrangement of the final piecewise developable geometry, rather than by representing a specific named component.



The Sigmoid activation function ensures $0 \leq \rho \leq 1$. With trainable weights initialized to zero, each network outputs a uniform density of 0.5 at initialization, resulting in $\rho(\mathbf{X}) = 0.125$. This value is close to several of the target volume fractions considered in the paper. In total, the neural network consists of one hidden layer. While deeper architectures that produce more complex maps can be utilized, this shallow architecture provides training efficiency.

To extract the surface of the geometry represented by each neural network, one can modify the activation function to obtain:

$$\Gamma(\mathbf{X})_i = e^{(-0.5\mathbf{W}_i \sin(\mathbf{K}_i \mathbf{X}+1))^2} \tag{2}$$

Figure 4 illustrates $\rho(\mathbf{X})$ vs. $\Gamma(\mathbf{X})$. The surface is extracted at $\Gamma = 1$. With the surface $\Gamma(\mathbf{X})$ of the i^{th} network, we can compute the seamlines as the intersection between each surface pair based on the second elementary symmetric polynomial. We further filter the summation with a smooth Heaviside filter to unite the seamlines. The resulting seamlines $S(\mathbf{X})$ are as follows:

$$S(\mathbf{X}) = H\left(\sum_{1 \leq j < k \leq n} \Gamma(\mathbf{X})_j \Gamma(\mathbf{X})_k\right) \tag{3}$$

where we use the Heaviside function whose parameters we empirically determine:

$$H(x) = \frac{1}{1 + e^{-2 \cdot 10 \cdot (x - 0.2)}} \tag{4}$$

Note that $0 \leq S(\mathbf{X}) \leq 1, \forall\ \mathbf{X}$. We use the term seamline with a slight abuse of meaning. Since all calculations are point-wise, it would be more appropriate to use 'seampoint', but we prefer 'seamline' for its colloquial familiarity.

The pairwise surface intersection can exhibit small nonzero values away from genuine intersections, causing false seamlines. To mitigate these, we shift the Heaviside function by 0.2 so only real overlaps activate. This threshold was chosen empirically for the balance between suppressing artifacts and preserving true seamlines.

In this work, we use three implicit fields to generate trimmed surfaces formed by their intersections, which are then divided into distinct patches. This means that the surface patches are n-colorable, where n is the number of neural implicit fields used in the construction (3 for our examples). According to the Heawood Conjecture (a generalization of the Four Color Theorem to higher-genus surfaces [Ringel and Youngs 1968]), some maps on higher-genus surfaces are not n-colorable for *any* fixed n and thus, if such an arrangement of surfaces is desired, more implicit fields will be required. In practice, we have not observed any issues – our optimization controls patch placement and adjacency, which avoids worst-case topologies and enables a three-color scheme that balances clarity and computational efficiency. Furthermore, it is relatively straightforward to increase the number of implicit fields to produce more complex surface patch layouts should the application require it.

3.3 Topology optimization with developable surfaces

Our overall topology design method considers four main objectives. First is topology optimization that performs compliance minimization under a user-prescribed volume fraction target [Allaire et al. 2002; Bendsøe and Kikuchi 1988; Chandrasekhar and Suresh 2021b; Liu and Tovar 2014]. The second goal is to produce topologies that can be subtractively machined based on a projection method for accessibility calculations. Third is Gaussian curvature minimization and the associated concentration of large curvatures along seamlines geared toward multi-axis flank milling. Finally, if specifically 3-axis milling is preferred (which remains very popular due to the ubiquitous availability of the machines in many shops), a fourth goal is to align the surfaces to the normal or tangent direction of the cutter.

Loss #1 - $\mathcal{L}_c$: Compliance for topology optimization. We adopt the advances in neural network-based topology optimization [Chandrasekhar et al. 2021; Chandrasekhar and Suresh 2021a,b; Chen et al. 2024, 2023]. The network takes as input space coordinates and outputs a material pseudo-density value associated with the SIMP method [Bendsøe and Kikuchi 1988]. Traditionally, SIMP-based compliance minimization can be formulated as:

$$\begin{aligned} \text{minimize}_{\ \rho} \quad & c(\rho^{(i)}) = \frac{1}{2}\mathbf{u}^T \mathbf{K}(\rho^{(i)})\mathbf{u}, \\ \text{subject to} \quad & \mathbf{K}(\rho^{(i)})\mathbf{u} = \mathbf{f}, \\ & V(\rho^{(i)}) = \frac{1}{N}\sum_i \rho^{(i)} \leq V^*, \\ & 0 \leq \rho^{(i)} \leq 1, \quad \forall i. \end{aligned} \tag{5}$$

where c is the compliance, $\rho^{(i)}$ is the set of design variables defined on the sampled coordinates. $\mathbf{u}$ is the displacement vector, $\mathbf{f}$ is the external force vector, $\mathbf{K}$ is the stiffness matrix, V is the resulting volume fraction, constrained by a prescribed upper bound V^*.

In this work, we use a linear elastic, hexahedral finite element model defined on a uniform cartesian grid. Our solver receives each element's density $\rho^{(i)}$ (i indexes nodes of the Finite Element

(FE) grid) as input and provides both the compliance and its corresponding sensitivity relative to the input coordinates. Following [Chandrasekhar and Suresh 2021b], we compose a loss function that combines compliance minimization and the volume fraction constraint for topology optimization:

$$\mathcal{L}_c = \frac{c(\rho^{(i)})}{c_0} + \omega_c\left(\frac{V(\rho^{(i)})}{V^*} - 1\right)^2 \tag{6}$$

where c is the current compliance (strain energy), c_0 is a normalization factor obtained from the initial unoptimized design domain. The work in [Chandrasekhar and Suresh 2021b] has shown that this formulation produces results virtually identical to those obtained via SIMP-based compliance minimization algorithms such as the optimality criteria method [Bendsøe 1989].

Loss #2 - $\mathcal{L}_m$: Tool accessibility for subtractive machining. For a part to be subtractively machined, all regions of the part need to be accessible by the milling tool. The accessibility of a point within the design domain is evaluated by projections performed in the direction of machining. For a rectangular prism commonly used as the initial workpiece, the machining orientation is typically aligned with the coordinate axes. As described earlier, we simply adopt $-z$ as the cutter's central axis. With this, we formulate a smooth approximation to the projection via a cumulative summation [Langelaar 2019]:

$$\begin{gathered} a^{(i)}_{[p,q,r]} = \begin{cases} \rho^{(i)}_{[p,q,1]}, & r = 1, \\ a^{(i)}_{[p,q,r-1]} + \rho^{(i)}_{[p,q,r]}, & r > 1. \end{cases} \\ p = 1 \dots n_x, \quad q = 1 \dots n_y, \quad r = 1 \dots n_z. \end{gathered} \tag{7}$$

where $a^{(i)}$ captures the 'accessibility' of node i located at $[p, q, r]$. ρ represents the current density field defined on the same FE grid with the domain size of (n_x, n_y, n_z). We use a smoothed Heaviside function $H(.)$ to normalize the accessibility field and make it congruent with the density field and form a dissimilarity index M that measures the dissimilarity between field a and ρ:

$$M = \sum_i H(a^{(i)})(1 - \rho^{(i)}) \tag{8}$$

Note that $0 \leq M \leq n_x n_y n_z$. During optimization, we employ the augmented Lagrangian method for subtractive loss to ensure the quality of results. The resulting subtractive machining loss is formulated as:

$$\mathcal{L}_m = \lambda M + \frac{1}{2}\gamma M^2 \tag{9}$$

where γ is gradually increased during optimization and λ is initialized to 1 and updated as:

$$\lambda \leftarrow \lambda + \gamma M \tag{10}$$

Loss #3 - $\mathcal{L}_g$: Developable surface. From the neural network's output, we define the surface as the level set $F(\mathbf{X}) = 0.5$ for alignment with the adopted convention of $0 \leq \rho \leq 1$. However, since the Gaussian curvature of the implicit surface only depends on the gradient ∇F, the Hessian $\mathcal{H}(F)$, and the adjoint of the Hessian $\mathcal{H}^*(F)$ [Goldman 2005], the following identities still hold for the neural network's output:

$$\nabla F = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z}\right) = (F_x,\, F_y,\, F_z), \tag{11}$$

$$\mathcal{H}(F) = \begin{pmatrix} \frac{\partial^2 F}{\partial x^2} & \frac{\partial^2 F}{\partial x\,\partial y} & \frac{\partial^2 F}{\partial x\,\partial z} \\ \frac{\partial^2 F}{\partial y\,\partial x} & \frac{\partial^2 F}{\partial y^2} & \frac{\partial^2 F}{\partial y\,\partial z} \\ \frac{\partial^2 F}{\partial z\,\partial x} & \frac{\partial^2 F}{\partial z\,\partial y} & \frac{\partial^2 F}{\partial z^2} \end{pmatrix} = \begin{pmatrix} F_{xx} & F_{xy} & F_{xz} \\ F_{yx} & F_{yy} & F_{yz} \\ F_{zx} & F_{zy} & F_{zz} \end{pmatrix} \tag{12}$$

$$\begin{gathered} \mathcal{H}^*(F) \equiv \operatorname{adj}(\mathcal{H}(F)) = \begin{pmatrix} C_{xx} & C_{xy} & C_{xz} \\ C_{xy} & C_{yy} & C_{yz} \\ C_{xz} & C_{yz} & C_{zz} \end{pmatrix}, \\ C_{xx} = F_{yy}F_{zz} - F_{yz}^2, \qquad C_{xy} = F_{xz}F_{yz} - F_{xy}F_{zz}, \\ C_{xz} = F_{xy}F_{yz} - F_{xz}F_{yy}, \quad C_{yy} = F_{xx}F_{zz} - F_{xz}^2, \\ C_{yz} = F_{xy}F_{xz} - F_{xx}F_{yz}, \quad C_{zz} = F_{xx}F_{yy} - F_{xy}^2. \end{gathered} \tag{13}$$

Goldman then derives the following form of the Gaussian curvature [Goldman 2005]:

$$K_G = \frac{\nabla F * \mathcal{H}^*(F) * \nabla F^T}{|\nabla F|^4} \tag{14}$$

Ultimately, the goal is to minimize $|K_G|$ everywhere. For this, we seek a solid structure consisting of patches of developable surfaces joined along seamlines. These seamlines are where large curvatures are designed to be concentrated, enabling directional changes in the generated surfaces. Such seamlines have to be identified and sharpened within the optimization iterations so that they can be excluded from subsequent Gaussian curvature minimizations. To generate developable surfaces during optimization, we compute the Gaussian curvature and convert it into a loss function by minimizing the absolute value filtered by the seamlines $S(\mathbf{X})$ (see Eqn. 3).

$$\mathcal{L}_g = (1 - S(\mathbf{X}))|K_G| \tag{15}$$

Ensuring seamline quality: As described previously, a seamline $S(\mathbf{X})$ is a scalar-valued function that characterizes the directional difference in the surface normals of two surfaces (Figure 4). The local normal vector to a surface is given by:

$$\mathbf{n} = -\nabla F \tag{16}$$

We quantify the local alignment of two surfaces via their local unit normals:

$$\cos(\alpha) = \tilde{\mathbf{n}}_1 \cdot \tilde{\mathbf{n}}_2 \tag{17}$$

where α is the angle between the two surfaces, $\tilde{\mathbf{n}}_1$ and $\tilde{\mathbf{n}}_2$ are the unit normals.

Ideally, surfaces obtained from our parallel neural implicit fields would intersect at sharp angles, leaving large swaths of surface points to be available for $|K_G|$ minimization. As such, we seek to form sparse but strong seamlines that are characterized by the concentration of large directional changes. For two surfaces that have an intersection angle α greater than the critical intersection angle, the following inequality must hold true:

$$|\cos(\alpha)| = |\tilde{\mathbf{n}}_1 \cdot \tilde{\mathbf{n}}_2| \leq \cos(\alpha_{cr}) \tag{18}$$

where $0 \leq \alpha_{cr} \leq \frac{\pi}{2}$ is a tunable angle that enforces the sharpness of the seam. Large values of α_{cr} help eliminate the formation of overlapping (aligned) surfaces, thereby restricting seam formation to sparse and narrow portions of the identified surfaces. The absolute value $|\cos(\alpha)|$ accounts for intersection angles greater than $\frac{\pi}{2}$.

To produce sharp seams between two surfaces Γ_i and Γ_j in a differentiable way, we formulate a seam quality loss function by modifying Eqn. 18:

$$L_{s,ij} = |H(|\tilde{\mathbf{n}}_i \cdot \tilde{\mathbf{n}}_j| - \cos(\alpha_{cr})) \cdot \mathbf{n}_i \cdot \mathbf{n}_j| \tag{19}$$

where H(.) is the Heaviside function. The unnormalized surface normals allow the loss function to amplify the contributions of $\mathbf{X}$ that form the true surfaces (those that give rise to $\Gamma(\mathbf{X})$) over those that reside off the identified surfaces. For three implicit fields with normals $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$, the combined seam quality loss function becomes:

$$\mathcal{L}_s(\mathbf{X}) = L_{s,12}(\mathbf{X}) + L_{s,13}(\mathbf{X}) + L_{s,23}(\mathbf{X}) \tag{20}$$

Loss #4 - $\mathcal{L}_a$: Cutter-surface alignment. This is an optional loss in case one wants to further restrict the optimization for 3-axis machining. For 3-axis machining, the cutter can remove local material most efficiently when the cutter's axis is normal (end-milling) or tangent (flank-milling) to the surface. We encode this in the form of a normal alignment A_n, and tangent alignment A_t loss. If we denote the cutter's rotational axis as $\mathbf{r}$ (which remains fixed relative to the world coordinates in 3-axis machining), we have:

$$\begin{aligned} A_n &= |H(|\tilde{\mathbf{n}} \cdot \mathbf{r}| - \cos(\beta_{cr})) \cdot \mathbf{n} \cdot \mathbf{r}| \\ A_t &= |H(|\tilde{\mathbf{n}} \times \mathbf{r}| - \cos(\beta_{cr})) \cdot (\mathbf{n} \times \mathbf{r})| \end{aligned} \tag{21}$$

where $\mathbf{n}$ is the surface normal of the current density field, $H(.)$ is the Heaviside function, β_{cr} (we gradually increment to 90°) is a user-defined angle threshold above which misalignments are penalized.

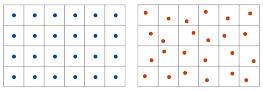
For efficient 3-axis machining, the cutter needs to align either with the normal or the tangent direction of the surface, which can be compactly written as an overall cutter-surface alignment loss:

$$\mathcal{L}_a = A_n A_t \tag{22}$$

It is interesting to note that this loss is complementary to $\mathcal{L}_g$. An open surface that at each point is either normal or tangent to a fixed direction r necessarily has $K_G = 0$ and is therefore developable. In the case of normal alignment where A_n would be minimized, the surface normal remains constant, which forces the surface to be planar. In the case of tangent alignment where A_t would be minimized, the surface is curved at most up to straight lines, all parallel to r, forming a cylindrical surface. Despite this strict alignment already implying developability, for 3-axis machining, we retain the original $\mathcal{L}_g$ to further assist with surface developability.

Combined loss. The topology optimization $\mathcal{L}_c$, subtractive machining $\mathcal{L}_m$, Gaussian curvature with sharp seams $\mathcal{L}_g$, $\mathcal{L}_s$, and alignment loss $\mathcal{L}_a$ are then combined into a single loss function.

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_m + \omega_g \mathcal{L}_g + \mathcal{L}_s + \mathcal{L}_a \tag{23}$$



During optimization, we gradually increase ω_g to minimize $|K_G|$. $\mathcal{L}_a$ is only enabled if optimizing for 3-axis machining. We gradually increase α_{cr} and β_{cr}. For $\mathcal{L}_c$ and $\mathcal{L}_m$, we sample at FE-grid-aligned coordinates (blue points located at the centers of FE cells), as compliance and accessibility are defined on the FE discretization and projection grid. For $\mathcal{L}_g$, $\mathcal{L}_s$, and $\mathcal{L}_a$, we employ quasi-Monte Carlo sampling implemented as jittered FE-grid samples (orange points distributed within the FE grid), with new random samples drawn at each epoch, since these geometric losses are not tied to the FE discretization. In all cases, the number of sampled coordinates per epoch is set equal to the number of FE grid nodes.

Table 1 describes the configurations of various hyperparameters used in our study.

Hyperparameter	Starting epoch	Δ	Final value
ω_c	0	0.5	100
γ	20	0.033	2
α_{cr} [°]	60	0.267	80
β_{cr}[°]	20	0.3	90
ω_g	40	0.033	1.0

Table 1. Our hyperparameter settings. All parameters start at 0. Increments are uniform with a step size of Δ per epoch.

4 Results and Discussion

This section details several 3D optimization studies comparing the following approaches: (A) topology optimization aimed at minimizing compliance, (B) topology optimization incorporating constraints for machining tool accessibility, (C) our proposed method for generating developable surfaces through topology optimization, and (D) a specific implementation of our method optimized for 3-axis milling. We evaluate the structural performance and machining time estimates derived from computer-aided manufacturing (CAM) simulations. Furthermore, we compare CAM-simulated machining times with real, physical machining times to validate the findings.

All experiments were conducted on a PC equipped with an Intel i7-12700K processor, 32 GB of RAM, and an NVIDIA RTX3080 GPU. We use the Adam optimizer [Kingma and Ba 2014] in all experiments, with a learning rate of 0.002, no weight decay, and no gradient clipping. All experiments are run for 400 epochs. The supplementary materials include the code implemented in TensorFlow [Abadi et al. 2016] necessary to replicate these experiments.

4.1 Test Cases, Geometric and Structural Performance Comparisons

Four test cases were utilized to assess the structural and manufacturing efficiency of the proposed method: (1) the MBB beam, a standard benchmark for topology optimization algorithms [Liu and Tovar 2014], (2) a bearing mount, (3) a pedal, and (4) a linkage structure. Figure 5 illustrates the boundary conditions for each case. When comparing compliance-only topology optimization to topology optimization with machining tool accessibility, the latter eliminates

Problem Type	FE Size	V^*	Compliance only ($\mathcal{L}_c$)			Subtractive accessibility ($\mathcal{L}_c$ $\mathcal{L}_m$)			Developable surface ($\mathcal{L}_c$ $\mathcal{L}_m$ $\mathcal{L}_g$)		
			V	c	*runtime(s)*	V	c	*runtime(s)*	V	c	*runtime(s)*
MBB	$40 \times 20 \times 8$	0.3	0.301	20.97	650	0.301	21.74	608	0.301	25.81	608
Mount	$40 \times 20 \times 8$	0.2	0.200	0.84	458	0.201	1.1	453	0.202	1.64	476
Pedal	$50 \times 20 \times 8$	0.2	0.200	31.82	651	0.200	32.65	603	0.201	38.65	626
Link	$40 \times 20 \times 8$	0.25	0.251	11.99	668	0.252	15.63	613	0.252	23.09	645

Table 2. Analytics from the four test cases. We observe that compliance saw a slight increase with the additional loss functions. All runs reached within 1% of the target volume fraction. The runtime slightly varies, but we see no significant worsening in runtime with the additional loss functions.

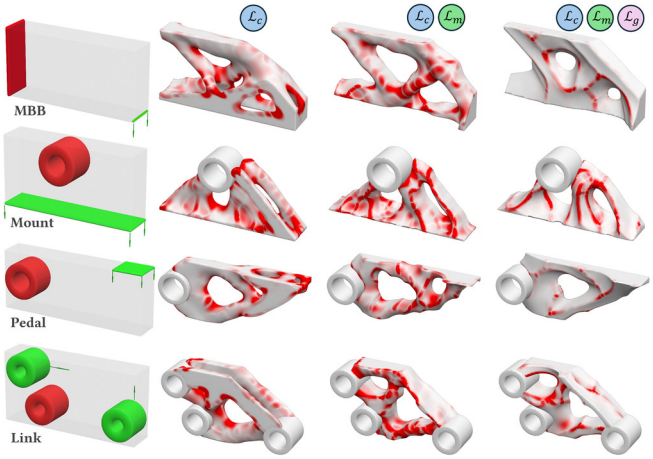


Fig. 5. Rows represent different example problems. The first column illustrates zero-displacement boundary conditions (red) and external forces (green). The subsequent columns display the optimization results for various objective functions, with the final column showcasing designs incorporating the developable surface objective. Regions of high absolute Gaussian curvature K_G are highlighted in red. Topology optimization with tool accessibility constraints (third column) yields complex surfaces characterized by extensive regions of high curvature. In contrast, our method (final column) generates sharper seamlines and larger areas of uniformly low Gaussian curvature, demonstrating improved surface regularity and manufacturability.

overhang regions by incorporating tool accessibility constraints. Similarly, the incorporation of developable surfaces further eliminates overhanging geometries. An analysis of surface curvature reveals that the developable surface optimization concentrates curvature along seamlines, leaving other areas with negligible Gaussian curvature.

Convergence and performance. Compliance was evaluated across the four test problems (Table 2). Results indicate a progressive increase in compliance as more constraints are imposed, from compliance-only optimization to subtractive tool accessibility, and subsequently to developable surfaces. This is consistent with the expected trade-off between structural performance and manufacturability, with additional constraints negatively impacting compliance.

Snapshots captured during optimization (Fig. 6) reveal the gradual concentration of curvature along the seamlines and the corresponding decrease in Gaussian curvature elsewhere.

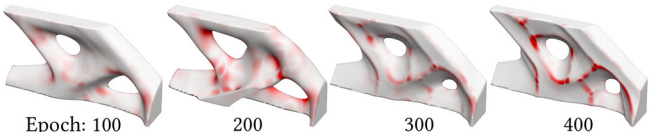


Fig. 6. Convergence history for topology optimization incorporating developable surface objectives. The results demonstrate a progressive concentration of high curvature regions along the seamlines.

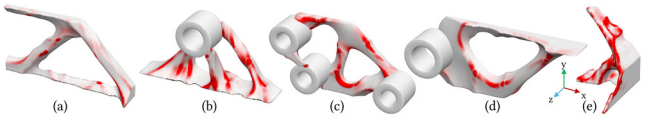


Fig. 7. (a-d) Four test cases, and (e) an elongated z-axis example of the MBB. All problems enforce 0.1 volume fraction. With the lower volume fraction, the results still exhibit prominent seamlines.

Further lightweighting and geometry complexity. Fig. 7 shows the results for 400 epochs, when volume fraction is reduced to 0.1, representing a more aggressive lightweighting. While seamline formation is largely preserved, the MBB (Fig. 7(a)) in particular exhibits a more modest delineation between the flat regions and seamlines. We also configure a new problem (Fig. 7(e)) with the same boundary conditions as the MBB case, with 1.25× stretching in the z direction. With the elongated z-axis, the seamline desirably follows the edges and contours of the geometry. We also evaluate the proposed method under randomized boundary conditions and varying aspect ratios with a low target volume fraction of 0.1 (Figure 8). The machining orientation is set to z- for the left two columns, and to both z+ and z- for the right two columns. Across these runs, the method demonstrates strong robustness.

We further compare against Autodesk Fusion 360's commercial topology optimization solver [Autodesk, Inc. 2024] using identical boundary conditions and a target volume fraction of 0.3 (Figure 9). The left shows standard topology optimization, while the right shows topology optimization with a subtractive manufacturing constraint under a z- orientation. Compared to these results, we do not observe a significant difference in geometric complexity relative to our method. However, this comparison is limited to a single boundary-condition configuration and should not be interpreted as a comprehensive evaluation of Fusion 360. Different boundary conditions, design domains, or manufacturing settings may produce more complex geometries than those observed here. **Comparison**

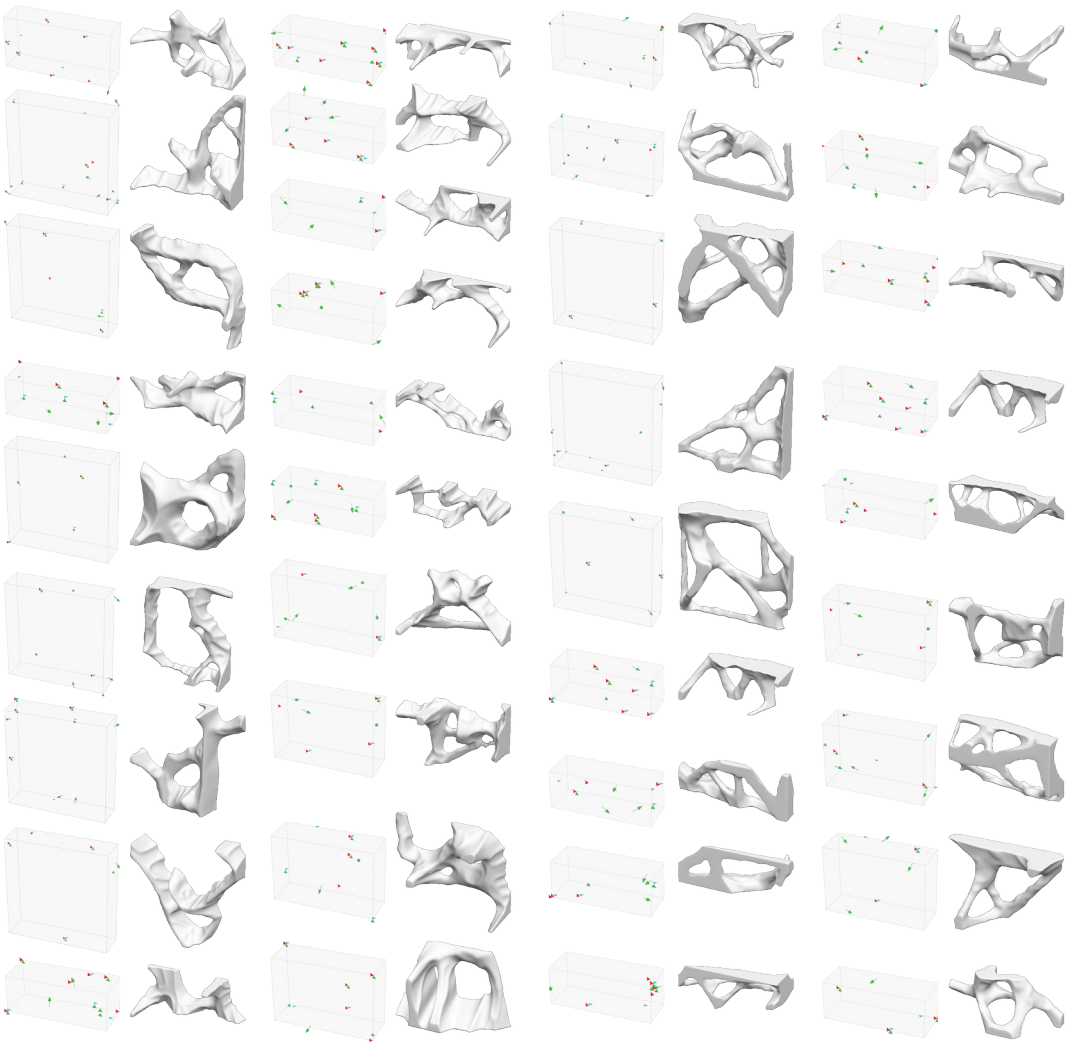


Fig. 8. Our approach is evaluated on varying design domain aspect ratios (gray boxes), randomized fixed displacement boundary conditions (red markers), and load cases (green markers), all with a low volume fraction of 0.1. Best viewed with digital zoom. For the left two columns, we use a single manufacturing orientation (z-), while the right two columns include both z+ and z- orientations. Relative to the compliance-only baseline, subtractive-only optimization and developable-surface optimization increase compliance by 45.82% and 147.3% on average, respectively, across the randomized-boundary-condition cases.

with geometric post-processing. In Fig. 10, we compare our results on the MBB with the surface developability post-processing approach by Stein et al. [Stein et al. 2018]. The post-processed structure by Stein et al. (Fig. 10 middle row) is obtained by applying their approach to the topology optimization with tool accessibility result shown in Figure 5 (first row, third column). The resulting structure (Fig. 10 middle row) exhibits characteristics similar to our developable surface (bottom row), notably bifurcations consisting primarily of three seamlines. However, since Stein et al.'s approach is geared toward transforming an existing mesh into developable pieces, machinability and displacement/force boundary condition management are understandably not the focal concern. As a result, this post-processing tends to smooth out regions critical for machining and an appropriate anchoring of the part for compliance minimization. In this example, post-processing smoothes out the corners at the back end of the part, resulting in regions that are not accessible with the prescribed machining orientation (Figure 10 second column). At the front of the part, the modified corners result in empty voxels at the regions of boundary conditions (third column), resulting in an expected and marked increase in compliance (fourth column).

Number of implicit fields. While we use three implicit fields (Fig. 3) throughout our examples, in this section, we showcase the results using two-, three-, and four-field implicit configurations, as shown in Figure 11. All optimization hyperparameters remain

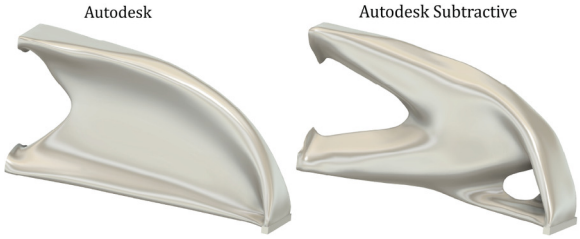


Fig. 9. We compare against Autodesk Fusion 360's generative design for a cantilever beam with identical boundary conditions and a target volume fraction of 0.3. The left shows unconstrained topology optimization in Autodesk Fusion 360, while the right shows generative design with a subtractive cutting constraint under a z- orientation in Autodesk Fusion 360. For this cantilever example, we do not observe significantly more complex features.

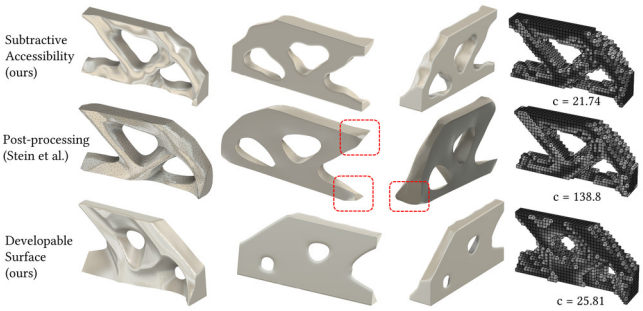


Fig. 10. Comparison between Stein et al.'s post-processing method, the subtractive accessibility and developable surface result for the MBB problem. The post-processed structure exhibits developable features with distinct seamlines. However, post-processing alters the geometry near the boundary conditions: inaccessible regions appear at the rear due to rounded corners. Regions of external forces at the front are also rounded. When all three solutions are identically voxelized for structural evaluation, these alterations negatively impact compliance.

identical across the experiments. With two fields, seamline trifurcation becomes impossible. This, in turn, limits how flat regions are configured and additionally increases the tendency for umbilical points, which are undesirable for flank cutting. However, compliance remains competitively low and has the least number of model parameters. The three-field configuration enables seamline trifurcations, resulting in enhanced geometric freedom, with umbilicals largely eliminated. While it does not always achieve the best compliance, it remains competitive across the cases. Notably, the three-field is qualitatively most similar to that of Stein et al.'s, shown previously in Figure 10. The four-field configuration results in slightly increased seamline complexity. In our experiments, we also observed slight issues with convergence stability in the four-field configurations, possibly due to overparameterization. Ultimately, we view the number of fields to be a hyperparameter where users must balance surface characteristics, geometric complexity, optimization performance, and downstream use of the resulting design in light of their end goals.

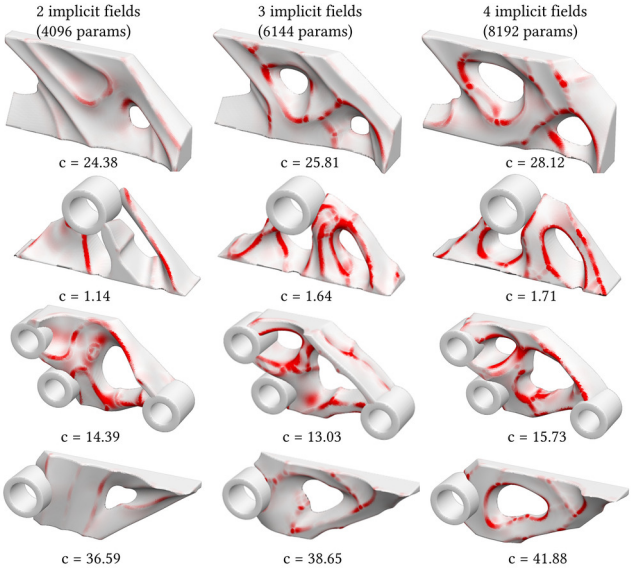


Fig. 11. Results across varying number of implicit fields, together with the varying number of network parameters in each column. Optimization parameters remain identical across the runs. The only differences are the number of implicit neural fields and adjustments to seamline and intersection loss computations to accommodate the additional field.

Feature scale control. The developable surface results highlight the complex orientation and distribution of seamlines (Figure 5). Previous studies on length scale control via implicit neural representations [Chandrasekhar and Suresh 2022] have demonstrated that input coordinate scaling can be used to control the level of detail in the final topology.

To demonstrate the range of feature sizes achievable with our network, we consider compliance minimization on a cantilever beam across multiple resolutions (Fig. 12). For each resolution, we perform 61 topology optimization runs using a *single* neural network, varying the input coordinate scaling. The parameter count is kept fixed for the 1× case and increased by 8× for the 2× and 3× resolutions.

For comparison, we evaluate Top3D [Liu and Tovar 2014] under identical boundary conditions, with its compliance shown as a dashed line. Across resolutions, many runs achieve lower compliance than SIMP. Moreover, increasing the coordinate scaling consistently produces finer feature structures, whereas increasing the resolution in SIMP does not yield a comparable increase in feature complexity. These results highlight the ability of the proposed representation to control feature size through coordinate scaling while maintaining competitive compliance performance.

Next, we analyze the effect of varying input coordinate scaling from 0.5 to 1.0 across the four test cases, which now include the developability requirements. Figure 13 shows that reduced scaling results in fewer, more continuous seamlines, while larger scales produce finer geometric features. As such, coordinate scaling enables feature size control and the exploration of a broader design space. This capability also simplifies solid modeling for practical applications, reducing complexity in CAD workflows.

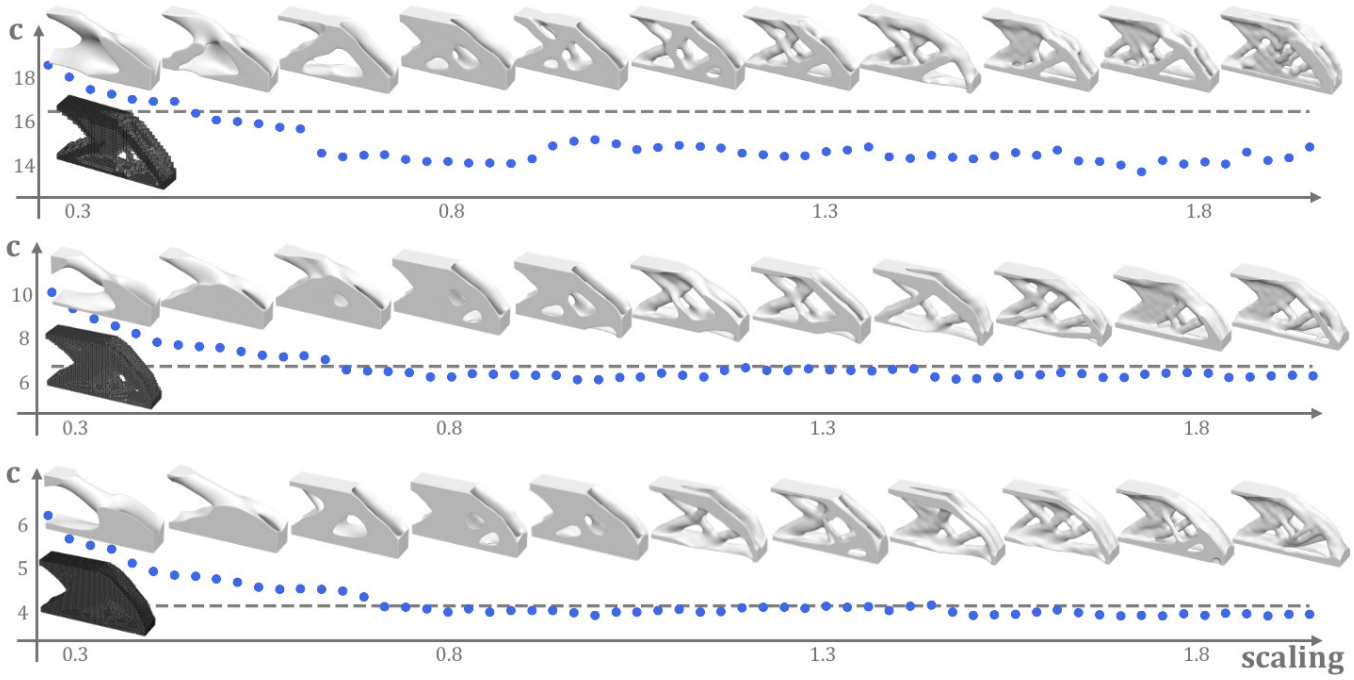


Fig. 12. Feature size control in compliance minimization. We consider the same cantilever beam example at 1× (row 1), 2× (row 2), and 3× (row 3) resolutions. For each resolution, we perform 61 topology optimization runs using a *single* neural network, varying the input coordinate scaling. The number of parameters remains the same for the 1× case, while the 2× and 3× cases use 8× more parameters. The resulting compliance values are shown in blue. For comparison, we also evaluate Top3D [Liu and Tovar 2014] under the same boundary conditions; its compliance is shown as a dashed line, with the corresponding topology inset in the bottom left. Notably, many runs across scaling achieve lower compliance than SIMP. In contrast, increasing the resolution in SIMP does not yield a comparable increase in feature complexity under our experimental settings. As the feature scaling increases, our network produces finer geometric features.

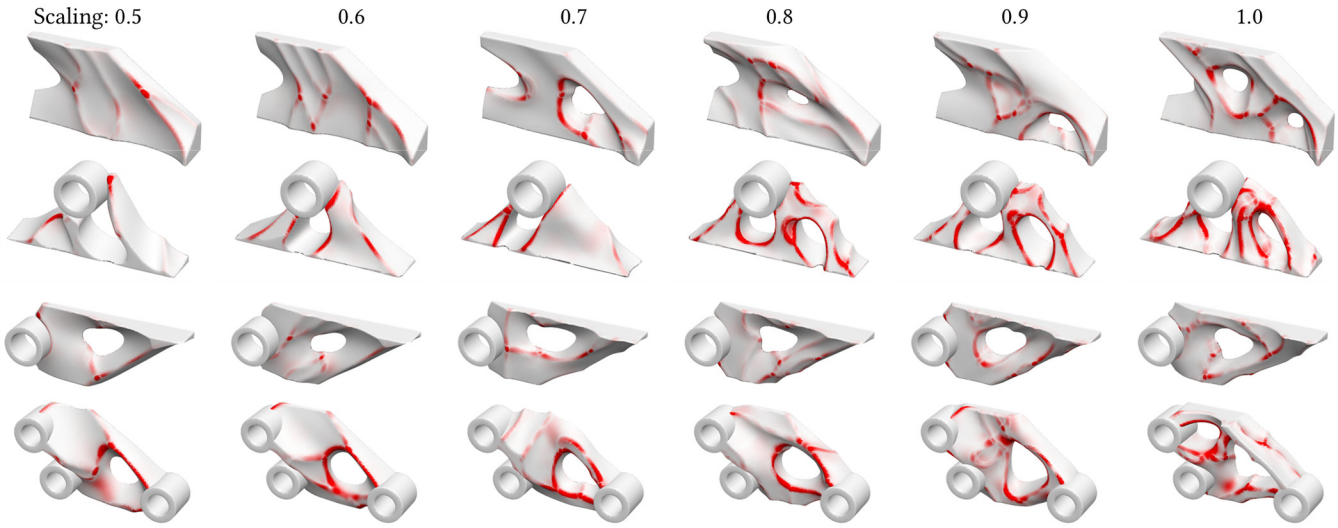


Fig. 13. Feature size control achieved through uniform scaling of the input coordinates. As the scaling factor increases, the number and complexity of seamlines also increase, illustrating a trade-off between geometric intricacy and machinability. Larger scaling factors yield more intricate designs with higher seamline density, while smaller scaling factors result in simpler geometries with fewer and less complex seamlines, enhancing manufacturability.

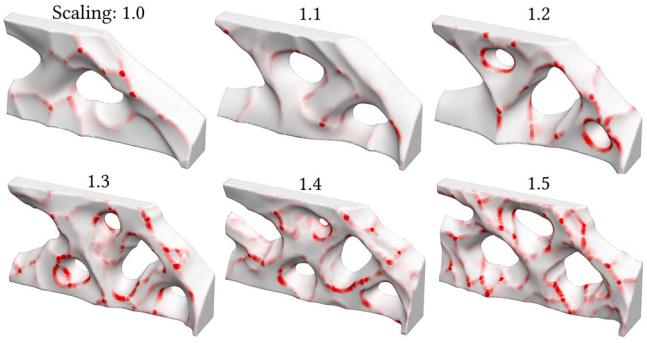


Fig. 14. The finite-element and subtractive-accessibility resolutions are doubled for the MBB problem. At this higher resolution, feature scaling continues to introduce finer geometric details.

From the compliance minimization example (Figure 12), we note that the level of detail cannot be increased indefinitely, as the FE grid resolution eventually becomes the bottleneck. Under the subtractive accessibility constraint, Figure 14 shows the additional level of detail that can be unlocked by doubling the FE resolution, at the expense of increased FE solve times. While the ability to increase geometric complexity through feature scaling and FE resolution is desirable, we note that for subsequent CAM simulations and physical machining, such intricate details can become unnecessarily prohibitive.

We further explore feature size control by leveraging multiple neural networks with independent scaling of z-axis coordinates. As illustrated in Figure 15, reducing z-axis scaling aligns seamlines closer to the xy-plane, further simplifying seamline geometry compared to uniform scaling. This is achieved by scaling the z-axis coordinates of two of the neural networks while keeping the third network's z-axis coordinates unaltered. This approach facilitates solid modeling in CAD software, which often relies on construction planes for defining features.

Passive regions. Passive regions are prescribed subdomains in which material placement is prohibited. More specifically, we consider passive-void regions. In conventional SIMP-based topology optimization, these regions can be enforced by setting the densities of the corresponding finite elements to zero before finite element analysis [Liu and Tovar 2014]. Let $\mathcal{P}$ denote the set of passive elements. The density supplied to the FE solver is then defined as

$$\rho_{\mathrm{FE}}^{(i)} = \begin{cases} 0, & i \in \mathcal{P}, \\ \rho^{(i)}, & i \notin \mathcal{P}, \end{cases} \tag{24}$$

where $\rho^{(i)}$ is the density produced by the neural implicit representation and $\rho_{\mathrm{FE}}^{(i)}$ is the corresponding density used for finite element analysis.

In neural implicit topology optimization, however, masking only the densities supplied to the FE solver does not constrain the underlying neural field. Consequently, $\rho^{(i)}$ may remain nonzero within $\mathcal{P}$, and material may reappear when the final implicit geometry is extracted. We therefore introduce an L_1 penalty on the neural-field output within the passive region:

$$\mathcal{L}_p = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \left| \rho^{(i)} \right|. \tag{25}$$

This term is added to the combined objective with a scalar weight. Since $0 \le \rho^{(i)} \le 1$, minimizing $\mathcal{L}_p$ drives the underlying neural field toward zero throughout the passive region.

We evaluate this formulation using a rectangular design domain whose left face is fully constrained and whose right boundary is subjected to a distributed tensile load. A spherical passive region is placed at the center of the domain, obstructing the otherwise direct structure between the applied load and the fixed boundary. This configuration makes the enforcement of the passive region apparent. Furthermore, a spherical surface has strictly positive Gaussian curvature. Simply subtracting a sphere from an optimized geometry would therefore introduce a nondevelopable surface rather than allow the optimizer to form low-curvature surfaces around it. This example evaluates whether the proposed method can simultaneously respect the prescribed passive region and produce a surrounding topology with low-curvature surfaces. We retain the same $-z$ machining orientation used in the preceding examples. As shown in Figure 16, the optimized topology redirects the structure around the spherical region while maintaining broad regions of low Gaussian curvature.

4.2 CAM Analysis and Machining Time Comparison

To further evaluate manufacturability, we performed CAM analysis using Autodesk Fusion 360 [Autodesk, Inc. 2024]. We set the scaling factor between FE element size and the physical part dimensions to be 0.1 in/element, hence making, for instance, the MBB beam's physical dimensions 4 in×2 in×0.8 in (refer to Table 2). For topology optimization with tool accessibility, we employed the marching cubes algorithm to generate a mesh at the 0.5 level set, which was subsequently imported into Fusion 360 and converted into a solid using the *convert mesh* function. For the optimization of developable surfaces, a test case with a reduced z-axis feature size was manually modeled in Fusion 360 for similarity and ease of manufacturing.

Both examples began with rough machining, a process focused on high material removal rates without prioritizing surface finish. This process involves high material removal rates and higher cutting depths, speeds, and feeds, aiming to shape the workpiece roughly to its near-final form in preparation for finishing operations. The `pocket cleaning` strategy in Fusion 360 was used for rough machining, generating offset contours to rapidly remove bulk material. Finishing operations employed different strategies based on geometry: for tool accessibility cases with complex curved surfaces, a ball-nose endmill with the `Parallel` strategy was utilized, while for developable surfaces, flank milling (or Swarf machining) with a flat-head endmill was employed. Figure 17 provides CAM toolpath visualizations for the MBB beam case study.

For consistency, the cutting feed rate and spindle speed were maintained across rough machining operations. Similar mill speed parameters were targeted for finish machining to ensure a fair comparison. Detailed machining parameters are summarized in Table 3.

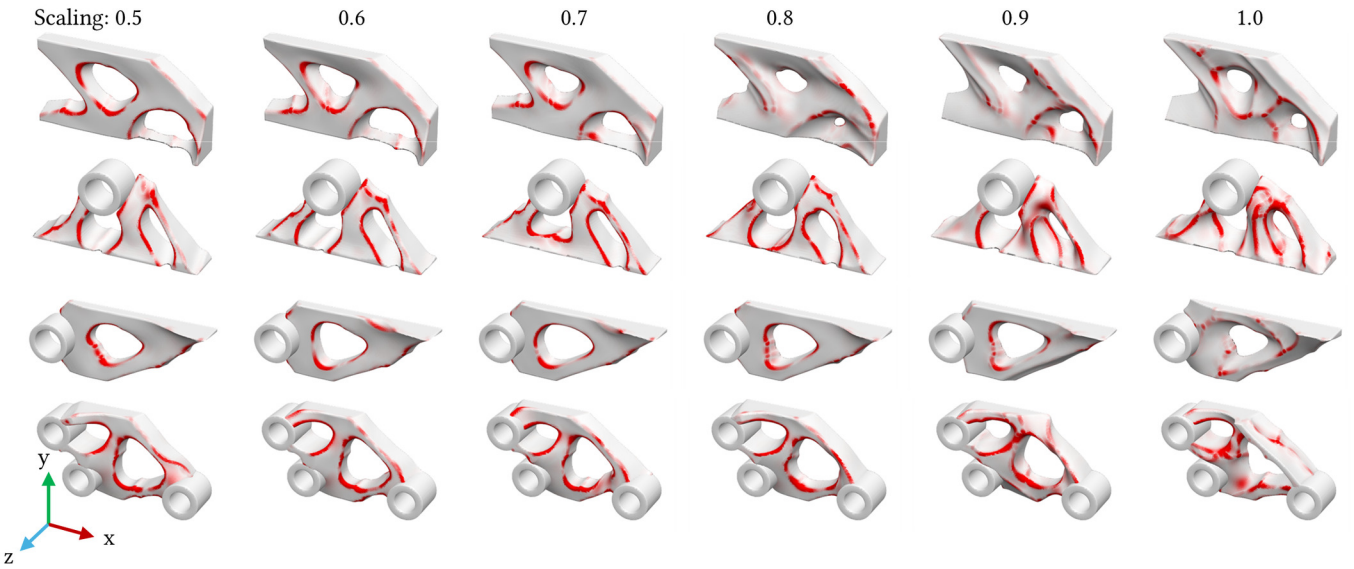


Fig. 15. Variation of z-axis coordinate scaling applied to two implicit neural fields. From right to left, the z-axis scaling factor is progressively reduced. This targeted reduction results in the redistribution of seamlines, aligning them more closely with the xy-plane. The decrease in geometric complexity facilitates simplified modeling and enables more efficient configuration of CAM operations, thereby enhancing manufacturability.

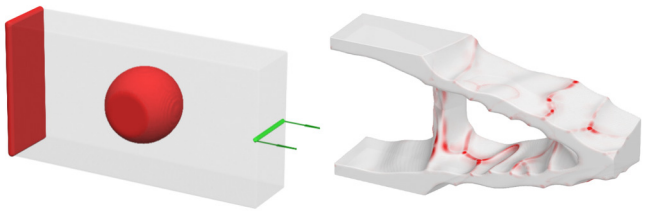


Fig. 16. Topology optimization with a spherical passive region. The left face is fixed and a distributed tensile load is applied on the right boundary, while material placement is prohibited within the central sphere. Under the $-z$ machining orientation, the optimized topology redirects the load path around the passive region while maintaining broad regions of low absolute Gaussian curvature. The geometry is mirrored about the horizontal midplane.

Parameter	Roughing	Parallel	Flank
Tool (3/8 Dia)	Flat Head	Ball Nose	Flat Head
Spindle Speed (RPM)	3000	3000	3000
Feedrate (in/min)	10	10	10
Load/Stepover (in)	0.1	0.02	0.1
Stepdown (in)	0.1	0.1	0.1

Table 3. Configuration of machining strategies for roughing, parallel, and flank operations. The selected parameters are intentionally conservative, reflecting the limitations of the relatively small CNC mill utilized in this study. To ensure consistency and comparability, the same machining parameters are applied when fabricating physical examples.

CAM toolpaths were generated based on specified parameters using Autodesk Fusion 360. Machining time estimates, derived from these toolpaths, are summarized in Table 4 for both optimizations for tool accessibility and developable surfaces across the four test cases. Roughing times were comparable between the two approaches, attributable to the similar volume fractions achieved during topology optimization. This results in largely similar bulk material removal rates across test cases. However, significant differences were observed during finishing operations. Topology optimization incorporating tool accessibility required longer finishing times, primarily due to the increased travel distance of the ball nose endmill necessary to achieve acceptable surface finishes. In contrast, the developable surface approach demonstrates substantial reductions in finishing time. This is because the cutter primarily moves along the tangential direction of the developable surface, requiring fewer passes and shorter travel distances.

4.3 3-axis milling tailored optimization with cutter-surface alignment

Despite the machining efficiencies of developable surfaces, flank milling these geometries necessitates the use of 5-axis machining centers, which are less prevalent than 3-axis configurations [DMG MORI 2022]. As described previously, we introduce an alignment loss function ($\mathcal{L}_a$) to make shape design amenable to 3-axis milling. Observations from prior results showed that reducing z-axis scaling constrained seamlines closer to the xy-plane, aligning surfaces with the cutter's tangential and normal orientations. To validate this approach, we reduced the z-axis scaling to 0.5 and applied the alignment loss across all test cases. The resulting geometries, visualized in Figure 18, exhibit enhanced alignment of surfaces with the 3-axis milling cutter compared to previous developable surface results (Figure 15).

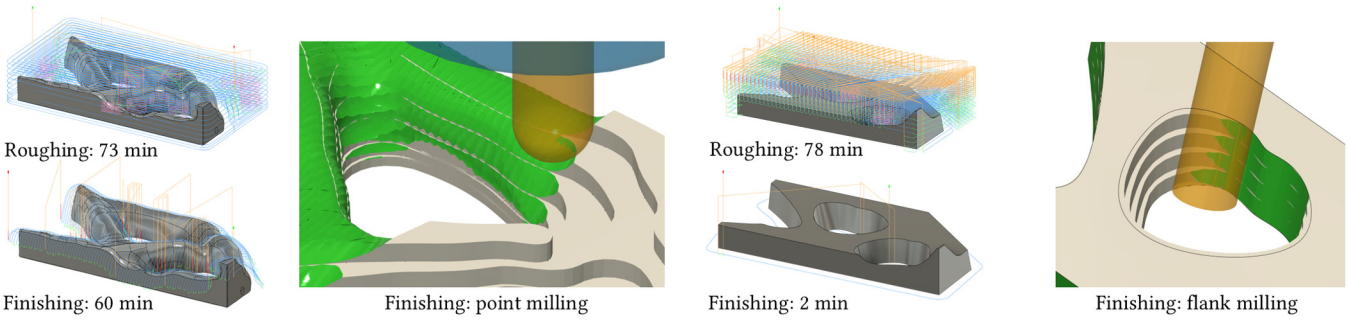


Fig. 17. CAM toolpath visualization for the MBB example. Left: CAM simulation for the tool accessibility case, where the complex surface necessitates the use of a parallel toolpath, a type of point milling. This approach requires multiple traverses to approximate the surface geometry. Right: Optimized for developable surfaces. The cutting direction for flank milling is computed by the CAM software to be tangent to the surface. This design enables the use of flank milling as the finishing toolpath, allowing the entire surface to be finished with significantly fewer passes, thereby improving machining efficiency.

Problem Type	**Subtractive accessibility** $\mathcal{L}_c$ $\mathcal{L}_m$			**Developable surface** $\mathcal{L}_c$ $\mathcal{L}_m$ $\mathcal{L}_g$			**Cutter-surface align** $\mathcal{L}_c$ $\mathcal{L}_m$ $\mathcal{L}_g$ $\mathcal{L}_a$		
	Roughing	*Finishing*	*Total*	*Roughing*	*Finishing*	*Total*	*Roughing*	*Finishing*	*Total*
MBB	73	60	133	78	2	80	89	1	90
Mount	91	44	135	82	3	85	85	3	88
Pedal	111	63	174	99	2	101	112	1	113
Link	76	51	127	65	3	68	74	4	78

Table 4. Machining times for roughing, finishing, and all operations combined for the subtractive-accessibility, developable-surface, and cutter-surface-alignment approaches. All times are in minutes.

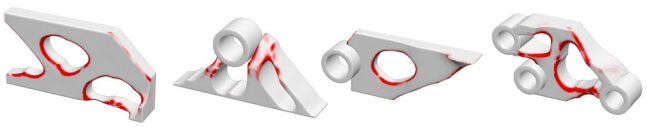


Fig. 18. Four test cases with cutter-surface alignment loss $\mathcal{L}_a$ enabled.

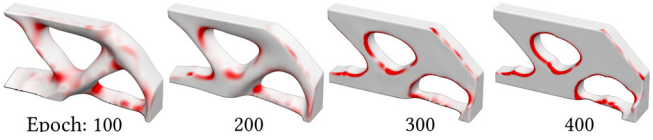


Fig. 19. Convergence history for the mbb beam example with alignment loss enabled.

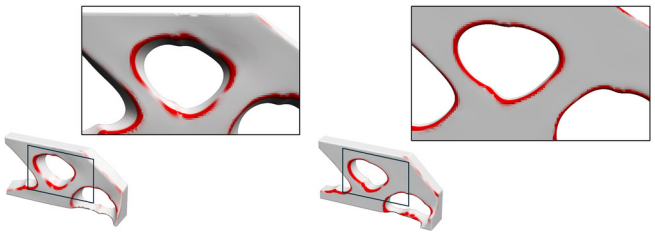


Fig. 20. Close-up comparison between the developable surface (left) and the 3-axis milling-oriented (right) results. While the overall geometric similarities are evident, the 3-axis milling-oriented design exhibits improved surface alignment with the cutter, facilitating enhanced compatibility with 3-axis machining.

Convergence during optimization is shown in Figure 19, illustrating how the surface gradually aligns with the milling tool. A close-up comparison in Figure 20 highlights the improved cutter-surface alignment achieved through this approach.

The 3-axis-optimized geometries were modeled and subjected to CAM analysis. The alignment of surfaces with the xy-plane facilitated simplified solid modeling, as the geometries could be approximated by extruding sketches on reference planes. Additionally, the roughing operation closely approximated the final geometry due to the alignment of most surfaces with the cutter's tangential and normal directions. However, rounded fillets along edges required finishing operations using a ball nose endmill, categorized as finishing operations. Consistent machining configurations, as summarized in Table 3, were used across all cases. The estimated machining times are detailed in Table 4.

To validate the CAM simulations, we machined two MBB beam examples—one optimized for tool accessibility and the other for 3-axis milling—using Aluminum 6061 on a Haas OM-2 machine [Haas Automation, Inc. 2025]. Estimated machining times from Fusion 360 were compared to actual machining times, summarized in Table 5. Results demonstrate a very close agreement between estimated and actual times, reinforcing our trust in the CAM simulations in this presented context.

Notably, the 3-axis-optimized example required slightly more roughing time due to the closer resemblance of the roughing geometry to the final part, which involved marginally higher material removal. For both examples, a uniform base was added to facilitate fixturing during machining and was manually removed afterward using a mill.

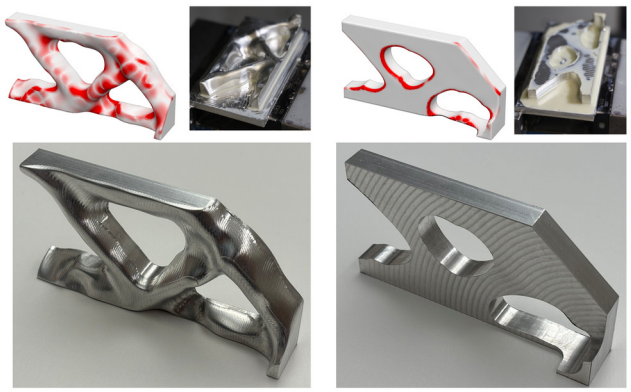


Fig. 21. Physical machining of the two examples was performed on a Haas OM2 using Aluminum 6061. Left: optimized for tool accessibility. Right: optimized for cutter-surface alignment (3-axis milling).

Figure 21 shows the optimized design and the finished part. We can see an overall smoother surface finish on the 3-axis tailored design.

5 Limitations and Future Work

This study presents a potential pathway for optimizing geometries for subtractive manufacturability by leveraging developable surfaces. While promising, several challenges remain in the practical machining of developable surfaces, particularly when using flank milling techniques, as highlighted in prior research [Harik et al. 2013]. The integration of implicit neural representations offers opportunities to compute additional surface properties. For example, these representations could facilitate the inclusion of tool diameter constraints alongside principal curvature constraints, ensuring the cutter's orientation remains within a predefined motion envelope. Furthermore, adopting strategies to prevent umbilical points using formulations derived for implicit surfaces [Che et al. 2007] could further enhance manufacturability.

Currently, the conversion of optimized results into manufacturable geometries depends on manual processes. Prior research has explored automated methods for converting b-rep to implicit neural representations [Guo et al. 2022], as well as mesh [Shen et al. 2025] and point cloud [Li et al. 2023] data into b-rep formats. Future research must build on these advancements to develop algorithms capable of faithfully converting implicitly defined solids with sharp seamlines into b-rep representations. Such algorithms must also preserve the developable nature of the surfaces, as current formulations face limitations in generating convex seams. This constraint, while beneficial in some scenarios such as avoiding the use of very small-diameter endmills, presents an area for improvement in enabling more complex geometric designs.

While compliance minimization remains a widely adopted strategy in topology optimization, more advanced formulations, such as stress-constrained mass minimization, could significantly broaden the applicability of this work. Similarly, incorporating additional physical criteria, including fatigue resistance, thermal-fluidic interactions, or optical performance, would extend the method's utility to more complex engineering applications.

Lastly, tailoring geometry representations to accommodate specific cutter geometries and optimizing motion planning for efficient manufacturing operations [Bartoň et al. 2021] represent promising directions for future research. These efforts would ensure that the optimized designs align with practical machining constraints, further bridging the gap between computational optimization and real-world manufacturability.

6 Conclusion

We introduce a new method that integrates topology optimization with manufacturability constraints to address the challenges associated with the subtractive machining of complex geometries. By leveraging implicit neural fields, we gain precise control over surface developability, enabling the generation of designs characterized by localized seamlines that concentrate curvature while maintaining regions of the geometry that remain developable. These seamlines not only facilitate efficient toolpath generation for subtractive processes such as 5-axis flank milling but also enable the adaptation of designs to meet the constraints of 3-axis machining.

Our findings demonstrate that this approach leads to a reduction in machining time compared to conventional methods, without significantly compromising structural performance. By achieving an optimal balance between design flexibility and machining efficiency, this methodology bridges the gap between the complex geometries generated by topology optimization and the practical requirements of subtractive manufacturing. The tri-implicit field representation offers a versatile framework for regulating curvature distribution and aligning geometries with specific machining constraints. This advancement paves the way for the development of more efficient and manufacturable designs in both additive and subtractive manufacturing contexts.

Acknowledgement

We would like to thank Professor Jim McCann for his valuable feedback and support throughout this work.

References

Niels Aage, Erik Andreassen, and Boyan Stefanov Lazarov. 2015. Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization* 51, 3 (2015), 565–572. doi:10.1007/s00158-014-1157-0

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv:1603.04467 [cs.DC] doi:10.48550/arXiv.1603.04467

Problem Type	Subtractive accessibility			Cutter-surface align		
	Roughing	*Finishing*	*Total*	*Roughing*	*Finishing*	*Total*
mbb (CAM estimated)	73	60	133	89	1	90
mbb (actual)	75	62	137	92	1	93

Table 5. Comparison of estimated machining time with actual machining time on the Haas OM2 mill. The results demonstrate that the Fusion 360 time estimation closely matches the actual machining duration. During the machining process, the feed rate was reduced on a few occasions when the machine indicated a heavy spindle load or when the long ball nose endmill produced chattering, which could potentially affect the quality of the machining process.

Grégoire Allaire, François Jouve, and Anca-Maria Toader. 2002. A level-set method for shape optimization. *Comptes Rendus. Mathématique* 334, 12 (2002), 1125–1130. doi:10.1016/S1631-073X(02)02412-3

Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. 2019. Controlling neural level sets. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc., Red Hook, NY, USA, 2032–2041. https://proceedings.neurips.cc/paper/2019/hash/b20bb95ab626d93fd976af958fbc61ba-Abstract.html

Autodesk, Inc. 2024. Autodesk Fusion 360. https://www.autodesk.com/products/fusion-360/overview

Saurabh Banga, Harsh Gehani, Sanket Bhilare, Sagar Patel, and Levent Kara. 2018. 3D Topology Optimization using Convolutional Neural Networks. arXiv:1808.07440 [cs.LG] doi:10.48550/arXiv.1808.07440

Amir Barda, Guy Tevet, Adriana Schulz, and Amit Haim Bermano. 2023. Generative Design of Sheet Metal Structures. *ACM Transactions on Graphics* 42, 4, Article 116 (2023), 13 pages. doi:10.1145/3592444

Michael Bartoň, Michal Bizzarri, Florian Rist, Oleksii Sliusarenko, and Helmut Pottmann. 2021. Geometry and tool motion planning for curvature adapted CNC machining. *ACM Transactions on Graphics* 40, 4, Article 180 (2021), 16 pages. doi:10.1145/3450626.3459837

Mohammad Mahdi Behzadi and Horea T. Ilieş. 2021. Real-Time Topology Optimization in 3D via Deep Transfer Learning. *Computer-Aided Design* 135 (jun 2021), 103014. doi:10.1016/j.cad.2021.103014

Martin P. Bendsøe. 1989. Optimal shape design as a material distribution problem. *Structural Optimization* 1, 4 (1989), 193–202. doi:10.1007/BF01650949

Martin Philip Bendsøe and Noboru Kikuchi. 1988. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering* 71, 2 (1988), 197–224. doi:10.1016/0045-7825(88)90086-2

Arturs Berzins, Moritz Ibing, and Leif Kobbelt. 2023. Neural implicit shape editing using boundary sensitivity. In *International Conference on Learning Representations*. Kigali, Rwanda. https://openreview.net/forum?id=CMPIBjmhpo

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Transactions on Graphics* 29, 4, Article 63 (2010), 10 pages. doi:10.1145/1778765.1778800

Aljaž Božič, Pablo Palafox, Michael Zollhöfer, Justus Thies, Angela Dai, and Matthias Nießner. 2021. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, USA, 1450–1459. doi:10.1109/CVPR46437.2021.00150

Aaditya Chandrasekhar, Saketh Sridhara, and Krishnan Suresh. 2021. AuTO: A Framework for Automatic Differentiation in Topology Optimization. *Structural and Multidisciplinary Optimization* 64, 6 (aug 2021), 4355–4365. doi:10.1007/s00158-021-03025-8

Aaditya Chandrasekhar and Krishnan Suresh. 2021a. Multi-Material Topology Optimization Using Neural Networks. *Computer-Aided Design* 136 (jul 2021), 103017. doi:10.1016/j.cad.2021.103017

Aaditya Chandrasekhar and Krishnan Suresh. 2021b. TOuNN: Topology Optimization using Neural Networks. *Structural and Multidisciplinary Optimization* 63, 3 (2021), 1135–1149. doi:10.1007/s00158-020-02748-4

Aaditya Chandrasekhar and Krishnan Suresh. 2022. Approximate Length Scale Filter in Topology Optimization using Fourier Enhanced Neural Networks. *Computer-Aided Design* 150 (sep 2022), 103277. doi:10.1016/j.cad.2022.103277

Wujun Che, Jean-Claude Paul, and Xiaopeng Zhang. 2007. Lines of curvature and umbilical points for implicit surfaces. *Computer Aided Geometric Design* 24, 7 (2007), 395–409. doi:10.1016/j.cagd.2007.04.005

Hongrui Chen, Aditya Joglekar, and Levent Burak Kara. 2024. Topology optimization using neural networks with conditioning field initialization for improved efficiency. *Journal of Mechanical Design* 146, 6 (2024), 061702. doi:10.1115/1.4064131

Hongrui Chen, Aditya Joglekar, Kate S Whitefoot, and Levent Burak Kara. 2023. Concurrent build direction, part segmentation, and topology optimization for additive manufacturing using neural networks. *Journal of Mechanical Design* 145, 9 (2023), 091702. doi:10.1115/1.4062663

Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. 2021. SNARF: Differentiable Forward Skinning for Animating Non-Rigid Neural Implicit Shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, Los Alamitos, CA, USA, 11574–11584. doi:10.1109/ICCV48922.2021.01139

Yonghua Chen, Jianan Lu, and Ying Wei. 2016. Topology optimization for manufacturability based on the visibility map. *Computer-Aided Design and Applications* 13, 1 (2016), 86–94. doi:10.1080/16864360.2015.1059199

Viktor Chlumskỳ, Jaroslav Sloup, and I Šimeček. 2018. Improved corners with multi-channel signed distance fields. *Computer Graphics Forum* 37, 1 (2018), 273–287. doi:10.1111/cgf.13265

Chih-Hsing Chu, Charlie CL Wang, and Chi-Rung Tsai. 2008. Computer aided geometric design of strip using developable Bézier patches. *Computers in Industry* 59, 6 (2008), 601–611. doi:10.1016/j.compind.2008.03.001

Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. 2020. NASA: Neural Articulated Shape Approximation. In *Computer Vision – ECCV 2020 (Lecture Notes in Computer Science, Vol. 12352)*. Springer International Publishing, Cham, Switzerland, 612–628. doi:10.1007/978-3-030-58571-6_36

Hao Deng and Albert C To. 2020. Topology optimization based on deep representation learning (DRL) for compliance and stress-constrained design. *Computational Mechanics* 66, 2 (2020), 449–469. doi:10.1007/s00466-020-01859-5

Hao Deng and Albert C To. 2021. A parametric level set method for topology optimization based on deep neural network. *Journal of Mechanical Design* 143, 9 (2021), 091702. doi:10.1115/1.4050105

DMG MORI. 2022. What is 5-Axis Machining? Accessed on January 14, 2025. https://us.dmgmori.com/news-and-media/blog-and-stories/blog/what-is-5-axis-machining

Zuomin Dong, Hui Li, and Geoffrey W. Vickers. 1993. Optimal rough machining of sculptured parts on a CNC milling machine. *Journal of Engineering for Industry* 115, 4 (1993), 424–431. doi:10.1115/1.2901785

Neelotpal Dutta, Tianyu Zhang, Guoxin Fang, Ismail E Yigit, and Charlie CL Wang. 2024. Vector Field-Based Volume Peeling for Multi-Axis Machining. *Journal of Computing and Information Science in Engineering* 24, 5 (2024), 051001. doi:10.1115/1.4063861

Elif Erdine, Antiopi Koronaki, Alican Sungur, Angel Fernando Lara Moreira, Álvaro López Rodríguez, George Jeronimidis, and Michael Weinstock. 2019. Curved-crease folding and robotic incremental sheet forming in design and fabrication. In *Proceedings of the Symposium on Simulation for Architecture and Urban Design*. Society for Modeling and Simulation International, San Diego, CA, USA, 1–8. doi:10.5555/3390098.3390114

Shen Fan and Przemysław Musialski. 2024. Optimizing 3D Geometry Reconstruction from Implicit Neural Representations. In *2024 International Conference on Machine Learning and Applications (ICMLA)*. IEEE, Los Alamitos, CA, USA, 816–821. doi:10.1109/ICMLA61862.2024.00117

Ron Goldman. 2005. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design* 22, 7 (2005), 632–658. doi:10.1016/j.cagd.2005.06.005

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé, III and Aarti Singh (Eds.). PMLR, Virtual Event, 3789–3799. https://proceedings.mlr.press/v119/gropp20a.html

Hao-Xiang Guo, Yang Liu, Hao Pan, and Baining Guo. 2022. Implicit Conversion of Manifold B-Rep Solids by Neural Halfspace Representation. *ACM Transactions on Graphics* 41, 6, Article 276 (2022), 15 pages. doi:10.1145/3550454.3555502

Haas Automation, Inc. 2025. Compact Mills. Accessed on January 19, 2025. https://www.haascnc.com/machines/vertical-mills/compact-mills.html

Xingjian Han, Yu Jiang, Weiming Wang, Guoxin Fang, Simeon Gill, Zhiqiang Zhang, Shengfa Wang, Jun Saito, Deepak Kumar, Zhongxuan Luo, Emily Whiting, and Charlie C. L. Wang. 2024. Motion-Driven Neural Optimizer for Prophylactic Braces Made by Distributed Microstructures. In *SIGGRAPH Asia 2024 Conference Papers*. Association for Computing Machinery, New York, NY, USA, Article 57, 11 pages. doi:10.1145/3680528.3687661

Ramy F Harik, Hu Gong, and Alain Bernard. 2013. 5-axis flank milling: A state-of-the-art review. *Computer-Aided Design* 45, 3 (2013), 796–808. doi:10.1016/j.cad.2012.08.004

Dong He, Zhaoyu Li, Yamin Li, and Kai Tang. 2021. Quasi-developable and signed multi-strip approximation of a freeform surface mesh for efficient flank milling. *Computer-Aided Design* 140 (2021), 103083. doi:10.1016/j.cad.2021.103083

David Hilbert and Stephan Cohn-Vossen. 2021. *Geometry and the Imagination*. AMS Chelsea Publishing, Vol. 87. American Mathematical Society, Providence, RI, USA. Reprint of the 1952 English edition. https://bookstore.ams.org/chel-87/

Stephan Hoyer, Jascha Sohl-Dickstein, and Sam Greydanus. 2019. Neural reparameterization improves structural optimization. arXiv:1909.04240 [cs.LG] doi:10.48550/arXiv.1909.04240

Sadashige Ishida, Chris Wojtan, and Albert Chern. 2022. Hidden degrees of freedom in implicit vortex filaments. *ACM Transactions on Graphics* 41, 6, Article 241 (2022), 14 pages. doi:10.1145/3550454.3555459

Baptiste Jousselin, Yann Quinsat, and Christophe Tournier. 2019. A 5-axis pocket roughing strategy reducing the remaining material volume. *Procedia CIRP* 82 (2019), 368–373. doi:10.1016/j.procir.2019.04.146

Martin Kilian, Simon Flöry, Zhonggui Chen, Niloy J. Mitra, Alla Sheffer, and Helmut Pottmann. 2008. Developable Surfaces with Curved Creases. In *Advances in Architectural Geometry 2008*. Vienna University of Technology, Vienna, Austria, 33–36. https://vecg.cs.ucl.ac.uk/Projects/SmartGeometry/creases/creases_aag_08.html

Dong-Won Kim, MA Suhaimi, Byung-Mun Kim, Min-Ho Cho, and F Frank Chen. 2013. Rough cut machining for impellers with 3-axis and 5-axis NC machines. In *Advances in Sustainable and Competitive Manufacturing Systems: 23rd International Conference on Flexible Automation & Intelligent Manufacturing*. Springer International Publishing, Cham, Switzerland, 609–616. doi:10.1007/978-3-319-00557-7_50

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980 [cs.LG] doi:10.48550/arXiv.1412.6980

Matthijs Langelaar. 2019. Topology optimization for multi-axis machining. *Computer Methods in Applied Mechanics and Engineering* 351 (2019), 226–252. doi:10.1016/j.cma.2019.03.037

Dongchi Lee, Kazuya Saito, Takuya Umedachi, Tung D Ta, and Yoshihiro Kawahara. 2018. Origami robots with flexible printed circuit sheets. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. Association for Computing Machinery, New York, NY, USA, 392–395. doi:10.1145/3267305.3267620

Yuan-Shin Lee and Tien-Chien Chang. 1991. CASCAM—An Automated System for Sculptured Surface Cavity Machining. *Computers in Industry* 16, 4 (1991), 321–342. doi:10.1016/0166-3615(91)90073-I

Hui Li, Zuomin Dong, and Geoffrey W Vickers. 1994. Optimal toolpath pattern identification for single island, sculptured part rough machining using fuzzy pattern analysis. *Computer-Aided Design* 26, 11 (1994), 787–795. doi:10.1016/0010-4485(94)90092-2

Yuanqi Li, Shun Liu, Xinran Yang, Jianwei Guo, Jie Guo, and Yanwen Guo. 2023. Surface and edge detection for primitive fitting of point clouds. In *ACM SIGGRAPH 2023 Conference Proceedings*. Association for Computing Machinery, New York, NY, USA, Article 44, 10 pages. doi:10.1145/3588432.3591522

Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. 2018. Narrow-band topology optimization on a sparsely populated grid. *ACM Transactions on Graphics* 37, 6, Article 251 (2018), 14 pages. doi:10.1145/3272127.3275012

Jikai Liu and Y.-S. Ma. 2015. 3D Level-Set Topology Optimization: A Machining Feature-Based Approach. *Structural and Multidisciplinary Optimization* 52, 3 (2015), 563–582. doi:10.1007/s00158-015-1263-7

Jikai Liu and Albert C To. 2017. Topology optimization for hybrid additive-subtractive manufacturing. *Structural and Multidisciplinary Optimization* 55, 4 (2017), 1281–1299. doi:10.1007/s00158-016-1565-4

Jikai Liu and Albert C To. 2020. Computer-aided design-based topology optimization system with dynamic feature shape and modeling history evolution. *Journal of Mechanical Design* 142, 7 (2020), 071704. doi:10.1115/1.4045301

Kai Liu and Andrés Tovar. 2014. An Efficient 3D Topology Optimization Code Written in MATLAB. *Structural and Multidisciplinary Optimization* 50, 6 (2014), 1175–1196. doi:10.1007/s00158-014-1107-x

Yong-Jin Liu, Kai Tang, Wen-Yong Gong, and Tie-Ru Wu. 2011. Industrial design using interpolatory discrete developable surfaces. *Computer-Aided Design* 43, 9 (2011), 1089–1098. doi:10.1016/j.cad.2011.06.001

Ali Mahdavi-Amiri, Fenggen Yu, Haisen Zhao, Adriana Schulz, and Hao Zhang. 2020. VDAC: Volume Decompose-and-Carve for Subtractive Manufacturing. *ACM Transactions on Graphics* 39, 6, Article 203 (2020), 15 pages. doi:10.1145/3414685.3417772

Zoë Marschner, Silvia Sellán, Hsueh-Ti Derek Liu, and Alec Jacobson. 2023. Constructive solid geometry on neural signed distance fields. In *SIGGRAPH Asia 2023 Conference Papers*. Association for Computing Machinery, New York, NY, USA, Article 121, 12 pages. doi:10.1145/3610548.3618170

François Mazé and Faez Ahmed. 2023. Diffusion Models Beat GANs on Topology Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 8 (2023), 9108–9116. doi:10.1609/aaai.v37i8.26093

Amir M Mirzendehdel, Morad Behandish, and Saigopal Nelaturi. 2020. Topology optimization with accessibility constraint for multi-axis machining. *Computer-Aided Design* 122 (2020), 102825. doi:10.1016/j.cad.2020.102825

Nigel Morris, Adrian Butscher, and Francesco Iorio. 2020. A subtractive manufacturing constraint for level set topology optimization. *Structural and Multidisciplinary Optimization* 61, 4 (2020), 1573–1588. doi:10.1007/s00158-019-02436-y

Nigel JW Morris, Pradeep Kumar Jayaraman, and Adrian Butscher. 2023. BeNTO: Beam Network Topology Optimization. *Computer-Aided Design* 156 (2023), 103439. doi:10.1016/j.cad.2022.103439

Keigo Nakamura and Yoshiro Suzuki. 2020. Deep learning-based topological optimization for representing a user-specified design area. arXiv:2004.05461 [cs.CE] doi:10.48550/arXiv.2004.05461

Andrei Nejur and Kyle Steinfeld. 2016. Ivy: Bringing a Weighted-Mesh Representation to Bear on Generative Architectural Design Applications. In *ACADIA 2016: Posthuman Frontiers: Data, Designers, and Cognitive Machines*. Association for Computer Aided Design in Architecture (ACADIA), Ann Arbor, MI, USA, 140–151. doi:10.52842/conf.acadia.2016.140

Zhenguo Nie, Tong Lin, Haoliang Jiang, and Levent Burak Kara. 2021. TopologyGAN: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain. *Journal of Mechanical Design* 143, 3 (2021), 031715. doi:10.1115/1.4049533

Edwin A. Peraza-Hernandez, Darren J. Hartl, Richard J. Malak, Jr., and Dimitris C. Lagoudas. 2014. Origami-inspired active structures: a synthesis and review. *Smart Materials and Structures* 23, 9 (2014), 094001. doi:10.1088/0964-1726/23/9/094001

Dmitry Petrov, Matheus Gadelha, Radomír Měch, and Evangelos Kalogerakis. 2024. ANISE: Assembly-Based Neural Implicit Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 30, 8 (2024), 4514–4526. doi:10.1109/TVCG.2023.3265306

Gerhard Ringel and John WT Youngs. 1968. Solution of the Heawood map-coloring problem. *Proceedings of the National Academy of Sciences* 60, 2 (1968), 438–445. doi:10.1073/pnas.60.2.438

Dylan Rowe, Alec Jacobson, and Oded Stein. 2024. Sharpening and Sparsifying with Surface Hessians. In *SIGGRAPH Asia 2024 Conference Papers*. Association for Computing Machinery, New York, NY, USA, Article 92, 12 pages. doi:10.1145/3680528.3687666

Adriana Schulz, Jie Xu, Bo Zhu, Changxi Zheng, Eitan Grinspun, and Wojciech Matusik. 2017. Interactive design space exploration and optimization for CAD models. *ACM Transactions on Graphics* 36, 4, Article 157 (2017), 14 pages. doi:10.1145/3072959.3073688

Silvia Sellán, Noam Aigerman, and Alec Jacobson. 2020. Developability of Heightfields via Rank Minimization. *ACM Transactions on Graphics* 39, 4, Article 109 (2020), 15 pages. doi:10.1145/3386569.3392419

Nicholas Sharp and Alec Jacobson. 2022. Spelunking the deep: Guaranteed queries on general neural implicit surfaces via range analysis. *ACM Transactions on Graphics* 41, 4, Article 164 (2022), 19 pages. doi:10.1145/3528223.3530155

Zeyu Shen, Mingyang Zhao, Dong-Ming Yan, and Wencheng Wang. 2025. Mesh2Brep: B-Rep Reconstruction via Robust Primitive Fitting and Intersection-Aware Constraints. *IEEE Transactions on Visualization and Computer Graphics* 31, 10 (2025), 6661–6676. doi:10.1109/TVCG.2025.3525844

Oded Stein, Eitan Grinspun, and Keenan Crane. 2018. Developability of triangle meshes. *ACM Transactions on Graphics* 37, 4, Article 77 (2018), 14 pages. doi:10.1145/3197517.3201303

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. 2021. Neural Geometric Level of Detail: Real-Time Rendering with Implicit 3D Shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, USA, 11353–11362. doi:10.1109/CVPR46437.2021.01120

Nicholas Turner, Bill Goodwine, and Mihir Sen. 2016. A review of origami applications in mechanical engineering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 230, 14 (2016), 2345–2362. doi:10.1177/0954406215597713

Sandro L Vatanabe, Tiago N Lippi, Cícero R de Lima, Glaucio H Paulino, and Emílio CN Silva. 2016. Topology optimization with manufacturing constraints: A unified projection-based approach. *Advances in Engineering Software* 100 (2016), 97–112. doi:10.1016/j.advengsoft.2016.07.002

Charlie CL Wang and Gershon Elber. 2014. Multi-dimensional dynamic programming in ruled surface fitting. *Computer-Aided Design* 51 (2014), 39–49. doi:10.1016/j.cad.2014.02.004

Zian Wang, Tianchang Shen, Merlin Nimier-David, Nicholas Sharp, Jun Gao, Alexander Keller, Sanja Fidler, Thomas Müller, and Zan Gojcic. 2023. Adaptive shells for efficient neural radiance field rendering. *ACM Transactions on Graphics* 42, 6, Article 260 (2023), 15 pages. doi:10.1145/3618390

Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. 2021. Geometry processing with neural fields. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., Red Hook, NY, USA, 22483–22497. https://proceedings.neurips.cc/paper/2021/hash/bd686fd640be98efaae0091fa301e613-Abstract.html

Yonggyun Yu, Taeil Hur, Jaeho Jung, and In Gwun Jang. 2019. Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization* 59, 3 (2019), 787–799. doi:10.1007/s00158-018-2101-5

Chao Yuan, Nan Cao, and Yang Shi. 2023. A Survey of Developable Surfaces: From Shape Modeling to Manufacturing. arXiv:2304.09587 [cs.GR] doi:10.48550/arXiv.2304.09587

Jonas Zehnder, Yue Li, Stelian Coros, and Bernhard Thomaszewski. 2021. NTopo: Mesh-free Topology Optimization Using Implicit Neural Representations. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., Red Hook, NY, USA, 10368–10381. https://proceedings.neurips.cc/paper/2021/hash/55d99a37b2e1badba7c8df4ccd506a88-Abstract.html

Yanfang Zhao, Van-Nam Hoang, Gang-Won Jang, and Wenjie Zuo. 2021. Hollow structural topology optimization to improve manufacturability using three-dimensional moving morphable bars. *Advances in Engineering Software* 152 (2021), 102955. doi:10.1016/j.advengsoft.2020.102955

Shuai Zheng, Zhenzhen He, and Honglei Liu. 2021. Generating Three-Dimensional Structural Topologies via a U-Net Convolutional Neural Network. *Thin-Walled Structures* 159 (feb 2021), 107263. doi:10.1016/j.tws.2020.107263

M Zhou and GIN Rozvany. 1991. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer methods in applied mechanics and engineering* 89, 1–3 (1991), 309–336. doi:10.1016/0045-7825(91)90046-9

Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-scale topology optimization with microstructures. *ACM Transactions on Graphics* 36, 5, Article 164 (2017), 16 pages. doi:10.1145/3095815