

# DeepMill: Neural Accessibility Learning for Subtractive Manufacturing

FANCHAO ZHONG\*, Shandong University, China

YANG WANG\*, Shandong University, China

PENG-SHUAI WANG, Peking University, China

LIN LU, Shandong University, China

HAISEN ZHAO<sup>†</sup>, Shandong University, China

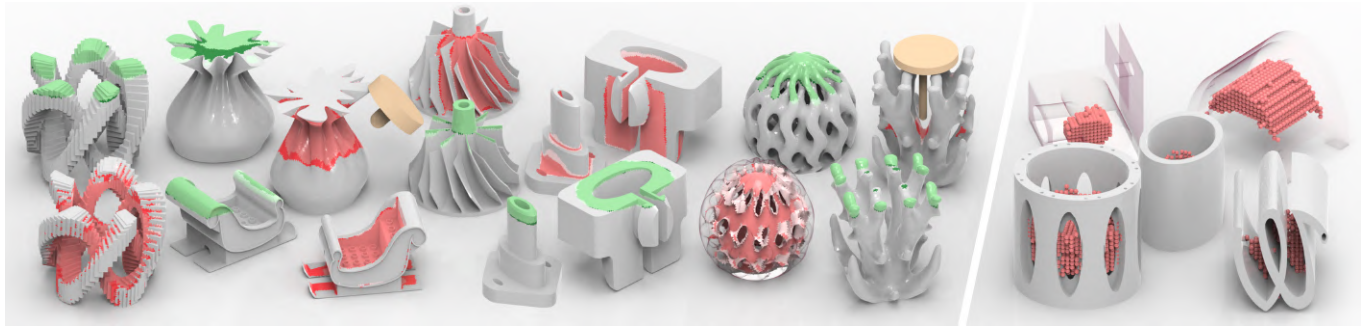


Fig. 1. We propose an octree-based neural network for cutter accessibility and severe occlusion detection on arbitrary meshes. Compared to traditional geometric methods, our network significantly reduces computation time, enabling real-time prediction during shape editing. Across various CAD and freeform model datasets with different mesh resolutions, the accuracy for inaccessible regions and occlusion regions reach 94.7% and 88.7%, respectively. The cutter sizes for each shape in the figure are randomly generated. We display the cutter shapes used for two shapes. On the left, the red and green areas represent inaccessible and severely occlusion regions predicted by the network. Dark red (green) and light red (green) areas indicate under- and over-predictions, respectively, compared to the traditional geometric method. On the right, the results of applying our network for volume accessibility analysis are shown.

Manufacturability is vital for product design and production, with accessibility being a key element, especially in subtractive manufacturing. Traditional methods for geometric accessibility analysis are time-consuming and struggle with scalability, while existing deep learning approaches in manufacturability analysis often neglect geometric challenges in accessibility and are limited to specific model types. In this paper, we introduce DeepMill, the first neural framework designed to accurately and efficiently predict inaccessible and occlusion regions under varying machining tool parameters, applicable to both CAD and freeform models. To address the challenges posed by cutter collisions and the lack of extensive training datasets, we construct a cutter-aware dual-head octree-based convolutional neural network (O-CNN) and generate an inaccessible and occlusion regions analysis dataset with a variety of cutter sizes for network training. Experiments demonstrate that DeepMill achieves 94.7% accuracy in predicting inaccessible regions and 88.7% accuracy in identifying occlusion regions, with an average processing

time of 0.04 seconds for finely-tessellated geometries. Based on the outcomes, DeepMill implicitly captures both local and global geometric features, as well as the complex interactions between cutters and intricate 3D models. Code is publicly available at <https://github.com/fanchao98/DeepMill>.

CCS Concepts: • **Computing methodologies** → **Shape modeling**; *Graphics systems and interfaces*.

Additional Key Words and Phrases: Subtractive manufacturing, Accessibility Analysis, Manufacturability Analysis

## ACM Reference Format:

Fanchao Zhong, Yang Wang, Peng-Shuai Wang, Lin Lu, and Haisen Zhao. 2025. DeepMill: Neural Accessibility Learning for Subtractive Manufacturing. *ACM Trans. Graph.* 1, 1 (June 2025), 11 pages. <https://doi.org/10.1145/3721238.3730657>

## 1 INTRODUCTION

Manufacturability is a fundamental concept in product design and production [Gupta et al. 1997; Shukor and Axinte 2009], referring to the ease and efficiency with which a design can be transformed into a physical product, while accounting for factors such as material constraints, cutter capabilities, and production costs [Hoefer et al. 2017; Joshi and Chang 1988; Li and Frank 2006]. Ensuring manufacturability early in the design process is critical to avoiding costly revisions, delays, and inefficiencies, thereby optimizing production timelines and minimizing resource waste.

Accessibility is a key aspect of manufacturability [Elber 1994; Spyridi and Requicha 1990]. In subtractive manufacturing, accessibility pertains to whether all surfaces and features of a part can be reached by machining tools during production [Zhang et al. 2020]. Accessibility issues arise when certain features are difficult to access,

\*Equal contribution

<sup>†</sup>corresponding author

Authors' addresses: Fanchao Zhong, [fanchaoz98@gmail.com](mailto:fanchaoz98@gmail.com), Shandong University, Qingdao, China; Yang Wang, [1766897491wy@gmail.com](mailto:1766897491wy@gmail.com), Shandong University, Qingdao, China; Peng-Shuai Wang, [wangps@hotmail.com](mailto:wangps@hotmail.com), Wangxuan Institute of Computer Technology, Peking University, Beijing, China; Lin Lu, [lulin.linda@gmail.com](mailto:lulin.linda@gmail.com), Shandong University, Qingdao, China; Haisen Zhao, [haisenzhao@sdu.edu.cn](mailto:haisenzhao@sdu.edu.cn), Shandong University, Qingdao, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0730-0301/2025/6-ART

<https://doi.org/10.1145/3721238.3730657>

such as deep holes, internal cavities, or overhanging geometries. Addressing these challenges through accessibility analysis is essential for identifying potential machining difficulties early, enabling designers to adjust part geometry or select appropriate cutters to optimize the manufacturing process. Beyond subtractive manufacturing, accessibility analysis also plays a critical role in decisions such as setup planning [Zhong et al. 2023], cutter selection [Athawale and Chakraborty 2010], cutter orientation adjustment [Mahdavi-Amiri et al. 2020], and tool path planning [Balasubramaniam et al. 2003], contributing to overall production efficiency and cost-effectiveness.

Traditional methods for geometric accessibility analysis, which emerged in the 1990s, primarily rely on geometric and computational techniques to evaluate cutter accessibility in multi-axis CNC machining. These methods, while foundational, are often time-consuming, especially for geometrically complex parts, with analysis of intricate designs taking hours—unacceptable in fast-paced design environments that require rapid iteration [Dai et al. 2018]. Although early approaches advanced from basic visibility analysis to more precise accessibility evaluations, they struggle with scalability and high computational overhead when applied to high-resolution 3D models. These limitations highlight the need for faster, more scalable methods capable of handling complex geometries and diverse cutter parameters.

In recent years, the advent of deep learning techniques has opened new possibilities for improving computational efficiency in manufacturability analysis. Several studies have explored the use of deep learning models to predict non-manufacturable regions [Chen et al. 2020; Ghadai et al. 2018; Kerbrat et al. 2011], enhancing performance by reducing processing time. However, most of these efforts focus primarily on process-related issues, such as process planning and collision detection [Chen et al. 2020], while neglecting the crucial geometric challenges inherent in accessibility analysis. Moreover, these methods often rely on feature-based CAD models [Balu et al. 2020; Yan and Melkote 2023], which limits their applicability to freeform or highly complex product designs.

This paper presents DeepMill, the first neural framework, to the best of our knowledge, specifically designed for predicting non-manufacturable regions in arbitrary models, including freeform shapes, with high accuracy and efficiency. Unlike previous methods, DeepMill focuses specifically on cutter accessibility, identifying regions where cutter collisions occur due to geometric constraints such as occlusion. We propose to utilize a neural network capable of real-time predictions for both non-manufacturable regions and the occlusion regions causing these issues, providing designers with actionable insights to quickly iterate and refine their designs. DeepMill demonstrates exceptional generalization across various cutter sizes and complex geometries, making it suitable for a wide range of design contexts.

One of the key challenges in accessibility analysis is the complexity of cutter collisions, which can be both local and global in nature. Factors such as cutter rotation and size affect accessibility, requiring methods that efficiently learn and represent these underlying geometric features. Moreover, the scarcity of extensive training datasets for these specific tasks has hindered the creation of robust models.

To overcome these challenges, we propose utilizing octree-based convolutional neural network (O-CNN) to efficiently capture both

local and global geometric features, while embedding cutter modules to capture intricate interactions between cutters and complex 3D surfaces. This approach enables our network to handle both CAD and freeform models, providing a scalable and flexible solution to the manufacturability analysis problem. Additionally, we created the first inaccessible and occlusion regions analysis dataset with diverse cutter parameters for training DeepMill and generated multiple test set categories, addressing the challenges of data scarcity.

In summary, DeepMill offers a significant advancement in both computational efficiency and accuracy. Experiments indicate DeepMill achieves 94.7% and 88.7% accuracy on average in identifying inaccessible and occlusion regions, with an average processing time of only 0.04 seconds for finely-tessellated geometries. Our model is adaptable to a wide range of cutter sizes, ensuring its applicability across diverse design contexts. Additionally, we introduce a new dataset to support further research in this area and facilitate the development of more robust manufacturability analysis cutters.

## 2 RELATED WORK

The rapid advancements in artificial intelligence (AI) have significantly propelled solutions in digital geometric design and manufacturing [Abdelaal 2024]. AI applications in additive manufacturing [Wang et al. 2020b; Zhang et al. 2024] and subtractive manufacturing [Manikanta et al. 2024; Soori et al. 2023] have been extensively reviewed. Cutting-edge research similarly advances computer graphics, encompassing assembly planning [Jones et al. 2021], LLM-centric design and manufacturing [Makatura et al. 2024a,b], 3D printing path optimization [Huang et al. 2024a; Liu et al. 2024], and feedback-based 3D printing control [Piovarci et al. 2022]. These advancements underscore a significant trend towards AI-based automation and optimization in design and manufacturing processes. This paper primarily examines the manufacturability analysis using a learning method, specifically focusing on the accessibility analysis for subtractive manufacturing [Gupta et al. 1997; Hoefer et al. 2017], which is also the main content of this section.

*Traditional Manufacturability Analysis.* Manufacturability of subtractive manufacturing is defined as four characteristics: visibility, reachability, accessibility, and setup complexity [Gupta et al. 1997; Hoefer et al. 2017]. Traditional techniques of manufacturability analysis primarily relies on two approaches: feature-based and feature-less methods [Zhang et al. 2020]. Feature-based methods extract machining features as a prerequisite, using techniques like graph-based analysis [Joshi and Chang 1988], volumetric decomposition [Kailash et al. 2001; Kim 1990], and hint-based approaches [Regli III 1995]. Feature-less methods, on the other hand, analyze surface representations to assess manufacturability, employing techniques such as slicing for machinable range mapping [Li and Frank 2006] and octree decomposition [Kerbrat et al. 2011].

Aligned with these conventional studies, where accessibility is the main evaluation metric for both feature-rich and feature-agnostic methods [Zhang et al. 2020], this paper primarily focuses on assessing accessibility. Unlike traditional approaches, we investigate the use of neural networks for accessibility analysis.

*Learning based Manufacturability Analysis.* Recent advancements leverage deep learning [Xu et al. 2023], such as autoencoder-based generative models for feature matching [Yan and Melkote 2023], 3D-CNNs with orthogonal distance fields (ODF) for manufacturability prediction [Balu et al. 2020], and enhanced B-rep structures with surface normal data for feature recognition [Ghadai et al. 2018]. Hierarchical graph neural networks have also been applied to analyze B-rep topology and UV network geometry for multi-level learning [Huang et al. 2024b]. A key limitation of these studies is the difficulty of applying learning methods to complex tasks like predicting cutter accessibility for freeform shapes.

A recent study explored only one directional accessibility analysis using a neural network [Harabin et al. 2023], and it requires retraining for each shape and cutter, struggling particularly with the high computational cost on finely-tessellated meshes. In contrast, DeepMill does not require retraining for the input shapes, enabling real-time prediction of inaccessible and occlusion areas in all directions while supporting general cutters.

*Geometric Accessibility Analysis.* Determining accessibility in multi-axis CNC machining remains a significant challenge, with numerous methods proposed since the 1990s. Early research focused on geometric and computational approaches, with Spyridi introducing the concepts of global and local accessibility for CMM [Spyridi and Requicha 1990], Saito proposing a graphics-inspired G-buffer method for tool accessibility analysis [Saito and Takahashi 1991], and Woo developing the spherical visibility map, which laid the foundation for 3D surface visibility analysis [Elber 1994; Woo 1994]. Other methods explored NURBS surfaces for interference calculations [Lee and Chang 1995], configuration space mapping for cutter range feasibility [Choi et al. 1997], effective cutter radius evaluations for end mills [Vafaeseefa and ElMaraghy 1998], and curvature-based differential inequality that verifies tool accessibility [Glaeser et al. 1999]. To simplify or accelerate calculations, various methods have been widely adopted, including sampling-based cutter direction analysis [Balasubramaniam et al. 2003; Dhaliwal et al. 2003; Mahdavi-Amiri et al. 2020; Zhao et al. 2018], cutter shape simplification [Glaeser et al. 1999], implicit shape representation using composite adaptively sampled distance fields [Sullivan et al. 2012], plane projections for blisk machining [Chen et al. 2015], boundary-based range construction [Liang et al. 2016], and bounding volume hierarchy construction for meshes [Bartoň et al. 2021]. More advanced approaches include Gaussian spherical mapping [Liu et al. 2020], ray-tracing-based metrics for assessing five-axis milling manufacturability [Chen and Frank 2021], and fast collision-free point removal based on footpoint computation [Chichell et al. 2024].

These approaches reflect the progression from basic visibility analysis to more efficient and precise accessibility evaluations in complex machining scenarios. However, when applied to intricate or high-resolution 3D models, they often face scalability challenges and significant computational overhead due to the time-intensive nature of strictly geometric methods [Dai et al. 2018]. Therefore, developing fast and scalable methods capable of handling complex geometries and diverse cutter parameters is highly meaningful.

*Spatial Analysis Learning.* Early works extended deep learning methods to 3D voxels [Maturana and Scherer 2015a; Wu et al. 2015]

for spatial analysis. However, voxel-based approaches are computationally expensive and memory-intensive, making them unsuitable for high-resolution 3D data. To address these limitations, sparse voxel-based CNNs leverage octrees [Wang et al. 2017a] or hash tables [Choy et al. 2019a; Graham et al. 2018] to confine computation to sparsely occupied voxels, significantly improving efficiency. Point-based neural networks [Li et al. 2018; Qi et al. 2017a,b] eliminate the need for voxelization by directly processing point clouds, offering an alternative solution. Recently, transformers have also been applied to 3D data, demonstrating promising results [Guo et al. 2021; Wang 2023; Zhao et al. 2021]. In this work, we adopt O-CNN [Wang et al. 2017a] for accessibility analysis due to its efficiency and strong performance across a range of 3D tasks.

### 3 METHODOLOGY

This section outlines the problem and goal of the network, followed by an introduction to our octree-based network, DeepMill, designed for predicting inaccessible and occlusion regions on the surface of the input mesh  $M$ .

#### 3.1 Problem formulation

The detection of inaccessibility and occlusion regions, which segments these areas from the surface of  $M$ , is formulated as a 3D geometric segmentation problem. To facilitate the computation, we use points to represent its local region. The goal of inaccessibility detection is to identify inaccessible points on  $M$  (labeled as  $I_I$ ) that are inaccessible to the cutter, meaning the cutter cannot reach them from any direction, even from an infinite distance, without causing a collision (Figure 3c). While occlusion detection aims to locate points causing the most severe occlusion for the inaccessible points. In this paper, we classify the top 10% of points with the highest occlusion severity as "occlusion points" (labeled as  $I_O$ ). Due to the severe imbalance in data distribution, we use the F1-score to evaluate the occlusion points. Unlike standard visibility problems, the shape and size of the cutter  $C$  directly influence the segmentation results. Consequently, this task is formulated as a dual-task binary segmentation problem with a cutter-aware objective, expressed as:

$$\text{Max}(\text{accuracy}(I_I) + F1(I_O) \mid C). \quad (1)$$

#### 3.2 DeepMill

DeepMill's main components include the encoder and the decoder, which are well-suited for segmentation task. Additional cutter modules and prediction head are added to further adapt to our problem. Figure 2 shows the network architecture of DeepMill.

*O-CNN with U-Net architecture.* Unlike more complex network architectures, such as the Hierarchical Graph Neural Network [Huang et al. 2024b], the O-CNN-based U-Net architecture uses a concise representation—point clouds and their normals—as input. The output consists of two predicted labels for each point. The U-Net architecture [Ronneberger et al. 2015] is composed of an encoder and a decoder, both of which are stacked with multiple Octree-based residual blocks [Wang et al. 2017b], with skip connections between the encoder and decoder. The encoder progressively extracts multi-scale features from the input 3D data, while the decoder gradually restores

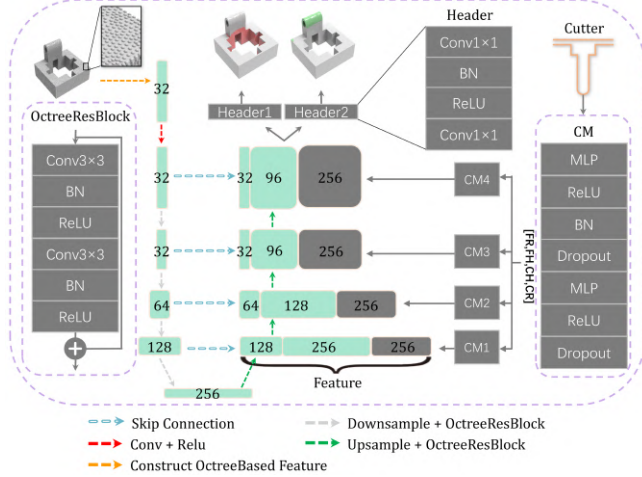


Fig. 2. DeepMill’s network architecture. The number represents the channel count. The input mesh is converted into a point cloud with normals, with each point corresponding to a Voronoi cell’s site. Features are progressively extracted through the encoder, and the decoder, embedded with a cutter module (CM), restores spatial resolution. Both encoder and decoder are stacked with several Octree-based residual blocks. Finally, each site is subjected to dual-task binary segmentation through two header layers. Red and green represent the inaccessible and occlusion regions, respectively.

spatial resolution and reconstructs accurate predictions through the use of skip connections. DeepMill adopts this architecture, and the benefits of this approach will be discussed in subsection 5.4.

**Cutter embedding.** In finishing machining of CNC, a ball-end cutter is commonly employed to finish the surface, modeled as a hemisphere combined with two differently sized cylinders. Figure 3(a) illustrates the cross-section of the cutter, which can be characterized using four parameters: two radii ( $CR$  and  $FR$ ) and two heights ( $CH$  and  $FH$ ). To enable the network to learn the impact of the cutter on inaccessibility and occlusion points, we embed cutter modules into the network. Compared to the encoder, the decoder is closer to the network’s final decision-making region, and concatenating the cutter features at this stage minimizes interference with the network’s early learning of geometric shapes. Furthermore, considering that the cutter causes collisions in both local regions and distant global regions (collide with the shaft space above the cutter) of  $M$ , we embed cutter modules at every layer of the decoder to help the network better learn the collision patterns between the cutter and  $M$  at different scales.

In detail, we encode the four shape parameters of the cutter into a vector  $\mathbf{V} = [v_1, v_2, v_3, v_4]^T$  and pass it through four fully connected cutter modules. As shown on the right side of Figure 2, each cutter module consists of two “Linear-ReLU-BN-Dropout” sub-blocks, where the 4-dimensional vector  $\mathbf{V}$  is transformed into a 256-dimensional cutter feature. These features are then concatenated into each layer of the decoder in the U-Net architecture:

$$f'_i = f_i \oplus f_i^c, \quad i = 1, 2, 3, 4 \quad (2)$$

where  $f_i$  indicates the output feature of the  $i$ -th layers in the decoder and  $f_i^c$  is the output feature of the  $i$ -th cutter module.

**Dual-head segmentation.** To predict inaccessible points and occlusion points separately, we use two fully connected header layers to predict these two types of labels. Since occlusion points are calculated based on inaccessible points, and both labels are computed using the same geometric algorithm during collision detection, there is a strong geometric correlation between the two labels. Therefore, before passing through the header layers, their features are fully shared. The predicted results  $\hat{y}_j^i$  for  $i$ -th point are denoted as:

$$\hat{y}_j^i = \text{header}_j(f'_i), \quad j = 1, 2 \quad (3)$$

**Architecture details.** As shown on the left side of Figure 2, each octree-based residual block consists of two “Convolution + BN + ReLU” sub-blocks, connected by residual connections [He et al. 2016]. Batch normalization (BN) is applied to reduce internal covariate shift [Ioffe 2015], while the ReLU activation function ( $f : x \in \mathbb{R} \mapsto \max(0, x)$ ) is used to activate the output.

In the encoder, the input point cloud undergoes multiple octree-based 3D convolution operations through several octree-based residual blocks, generating feature maps at different levels to capture multi-scale geometric features for hierarchical representation. Unlike traditional 3D-CNN convolutions [Maturana and Scherer 2015b], the octree structure marks non-empty nodes at the current depth, representing regions containing point clouds, and applies convolutions only to these nodes. The depth of the octree gradually decreases, and high-resolution child node features are aggregated into their corresponding parent nodes.

In the decoder, the global feature map is progressively processed through deconvolution for feature upsampling and spatial detail recovery, with cutter feature fusion enhancing the modeling of inaccessibility and occlusion effects. As the depth of octree increases, features are progressively passed down to the high-resolution child nodes. Output-guided skip connections [Wang et al. 2020a] are used to transfer features from the encoder to the decoder, excluding sparse regions. If the octree node corresponding to a feature output from a block is empty, the skip connection is not applied.

**Loss function.** During network optimization, we use cross-entropy loss function to compute the loss for inaccessible and occlusion points separately, denoted as  $\mathcal{L}_I$  and  $\mathcal{L}_O$ . The total loss function is defined as:

$$\mathcal{L} = \mathcal{L}_I(\hat{y}_1, y_1) + \mathcal{L}_O(\hat{y}_2, y_2) \quad (4)$$

where  $y_1$  and  $y_2$  denote the ground truth labels calculated using the geometric method mentioned in section 4.

## 4 GEOMETRIC METHOD FOR DATASET GENERATION

In this section, we introduce a rapid geometric approach to generate datasets with labels for inaccessible and occlusion regions.

### 4.1 Voronoi-based accessibility analysis

We use the subtractive collision detection method from [Zhong et al. 2023] to gather accessibility training data, as it’s efficient. We introduced a slight modification to their method by incorporating a



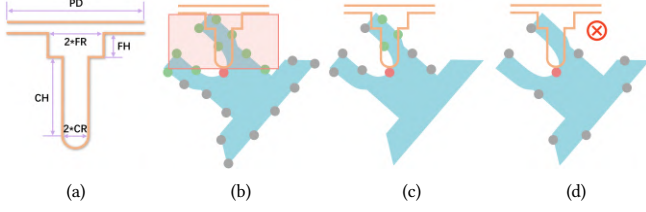


Fig. 3. Illustration of inaccessible point detection. Orange represents the cutter, and the gray points represent sampled Voronoi sites. (a) A ball-end cutter can be simplified using four parameters. Note that above cutter is a non-accessible shaft space, and  $PD$  is set to infinity. (b) For collision detection with red points, the mesh is first rotated, and points are quickly filtered by checking whether they lie within the detection box (red) of radius  $FR + \sigma$ , which eliminates most points far from the cutter.  $\sigma$  is set to 5 in our experiments. (c) A finer collision check is performed for the points inside the box. (d) To prevent the cutter from penetrating the mesh without detection, the spacing between adjacent sites must be smaller than the cutter's ball-end radius ( $CR$ ).

detection box for pre-detection, enabling faster calculation. Details of the method are outlined below.

*Voronoi-based sampling.* The inset figure demonstrates the use of Voronoi relaxation [Meng et al. 2023] for uniform sampling on the surface of  $M$ , with each Voronoi cell represented by its site  $s_i \in S$ , where  $S = \{s_1, s_2, \dots, s_n\}$ . The surface of  $M$  can be simplified as  $M = \bigcup_{i=1}^n s_i$ . We uniformly sample cutter directions  $D = \{d_1, d_2, \dots, d_m\}$  using the Fibonacci Sphere sampling method [Vorobiev 2002] on the upper Gaussian hemisphere. To ensure successful manufacturing, it is essential to guarantee the cutter  $C$  does not collide with  $M$  in any direction:

$$\forall s_i \in S, \forall d_k \in D, s_i \cap C|d_k = \emptyset \quad (5)$$

where  $C|d_k$  represents the cutter in direction  $d_k$ .

*Inaccessible points.* For each  $s_i \in S$ , collision detection is performed with other sites,  $\forall s_j \in S$  ( $j \neq i$ ), using the method from [Zhong et al. 2023]. We first rotate  $M$  contrarily along direction  $d_k \in D$ . To accelerate computation, a cylinder with a radius of  $FR + \sigma$  is added as a detecting box before collision detection, allowing only points within the cylinder to undergo finer detection, as shown in Figure 3(b). Next, as illustrated in Figure 3(c), each  $s_j$  is rapidly evaluated for collision by calculating its horizontal distance from the center of  $C$ . If the Z-coordinate of  $s_j$  exceeds  $CR + CH + FH$ , it is immediately classified as colliding with the infinitely large shaft space (called global collision). After the traversal, if  $s_i$  collides with at least one  $s_j$  in all cutter directions, it is classified as an inaccessible point:

$$s_i \leftarrow I_I \iff \forall d_k \in D, |S \cap C(d_k)| > 0 \quad (6)$$

Compared to the triangle-facet-based approach [Dhaliwal et al. 2003], which involves collision detection between the cutter's cylindrical surface and the triangular mesh, the proposed method is based

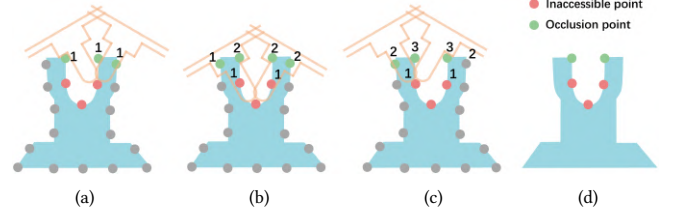


Fig. 4. Illustration of occlusion point calculation. (a)~(c) Perform collision detection for three inaccessible points, recording the points that collide with the cutter in each cutter direction and counting the total number of collisions for each point. (d) The top 10% of points with the highest total collision counts are labeled as occlusion points ( $I_O$ ).

on discrete sampling points (sites of Voronoi cells), significantly improving computational efficiency. Even so, this method still has a worst-case complexity of  $O(mn^2)$ . Besides, it is crucial to ensure that the shortest edge length of the smallest Voronoi cell is greater than  $2 * CR$  to prevent the cutter from passing through the cell without detection, as illustrated in Figure 3(d).

## 4.2 Occlusion analysis

To further assist designers in modifying the "culprit" causing inaccessible points, we compute the "occlusion factor"  $\beta_i$  for each  $s_i \in S$  to quantify the severity of its occlusion for the inaccessible points:

$$\beta_i = \sum_{s_j \in S} \sum_{d_k \in D} \begin{cases} 1, & \text{if } s_j \text{ is inaccessible and } s_i \text{ occludes } s_j \text{ in } d_k \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

The top 10% of  $s_i$  with the highest  $\beta_i$  values are defined as "occlusion points" ( $s_i \leftarrow I_O$ ). As shown in Figure 4, the points in the upper region of the 2D shape are marked as "occlusion points."

## 5 RESULTS AND DISCUSSIONS

### 5.1 Data processing and datasets

*Data processing and cleaning.* We performed the accessibility and occlusion analysis described in section 4 to construct the training and test sets, uniformly sampled 150 cutter directions on the upper hemisphere. We selected various CAD shapes from the ABC dataset [Koch et al. 2019] and freeform shapes from Thingi10K [Zhou and Jacobson 2016]. After cleaning non-manifold, non-watertight, and multi-component assemblies, we retained high-quality shapes. To avoid invalid data from shapes significantly smaller than the cutter, we extracted each shape's bounding box, ensuring its minimum edge length was at least 80 mm. For each shape, the coordinates of Voronoi cell sites, normals, and corresponding inaccessible and occlusion labels were recorded in the training and test sets.

*Datasets.* In the training set, we randomly generated the four cutter parameters for over 5K CAD shapes within specified ranges:  $CR \in [1, 2]$ ,  $FR \in [5, 100]$ ,  $CH \in [0.1, 10.1]$ , and  $FH \in [0.1, 10.1]$ . These parameter ranges make the ratio of inaccessible to accessible points in the training set close. Each shape had an average of approximately 7K mesh vertices. For the test sets, we selected 1K CAD shapes distinct from the training set and created two sets

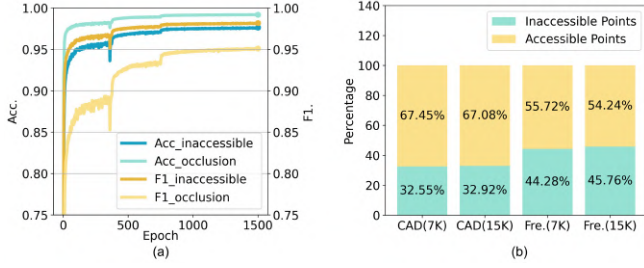


Fig. 5. (a) Training curves for four key metrics, including the accuracy and F1-score of inaccessible and occlusion points. (b) The proportion of inaccessible and accessible points in the datasets.

through remeshing, containing 7K and 15K mesh vertices, respectively. Similarly, we generated two test sets for freeform shapes. To further validate DeepMill’s generalization ability, we curated a finely-tessellated dataset of 500 shapes with over 100K vertices from the ABC dataset. Additionally, we varied cutter size parameters and generated multiple test sets based on CAD shapes to demonstrate DeepMill’s adaptability to different cutter sizes.

## 5.2 Implementation

All experiments were conducted on a desktop computer equipped with an Intel Core i7-11700F CPU running at 16 GB of memory, and an RTX 3090 GPU with 24 GB of memory. The octree depth was set to 5, with encoder channels configured as [32, 32, 64, 128, 256] and decoder channels as [256, 256, 128, 96, 96]. The cutter module channels were set to [256, 256, 256, 256]. We used the stochastic gradient descent (SGD) optimizer for training, starting with an initial learning rate of 1.0, which was adjusted using the Cosine Annealing method. The network was trained for 1500 epochs with a batch size of 128 for both training and testing. All input points were normalized to the unit cube  $[-1, 1]^3$ , and data augmentation methods from [Choy et al. 2019b], including random mirroring and elastic deformations, were applied.

## 5.3 Evaluation

*Efficiency and Generalization.* DeepMill was trained on a dataset of over 5K CAD shapes. The entire training process was conducted on an GeForce RTX 3090 GPU and took a total of 89 hours. Figure 5 shows the convergence curves of accuracy and F1-score during training process (a), as well as the ratio of accessible to inaccessible points in the datasets (b). We evaluated DeepMill’s efficiency on datasets containing diverse shapes, as shown in Table 1. We map the Voronoi sites onto the triangles for better visualization. Figure 1 and Figure 6 illustrate plenty of examples of inaccessible and occlusion regions predicted by DeepMill with various cutter sizes.

In general, DeepMill maintains high prediction accuracy, with most errors occurring near the boundaries of inaccessible and occlusion regions, which minimally affects the overall distribution. Additionally, as the mesh resolution increases (e.g., 7K vs. 15K Freeform shapes), finer geometric details further improve DeepMill’s prediction accuracy, and the higher resolution amplifies the time advantage of DeepMill over traditional geometric methods.

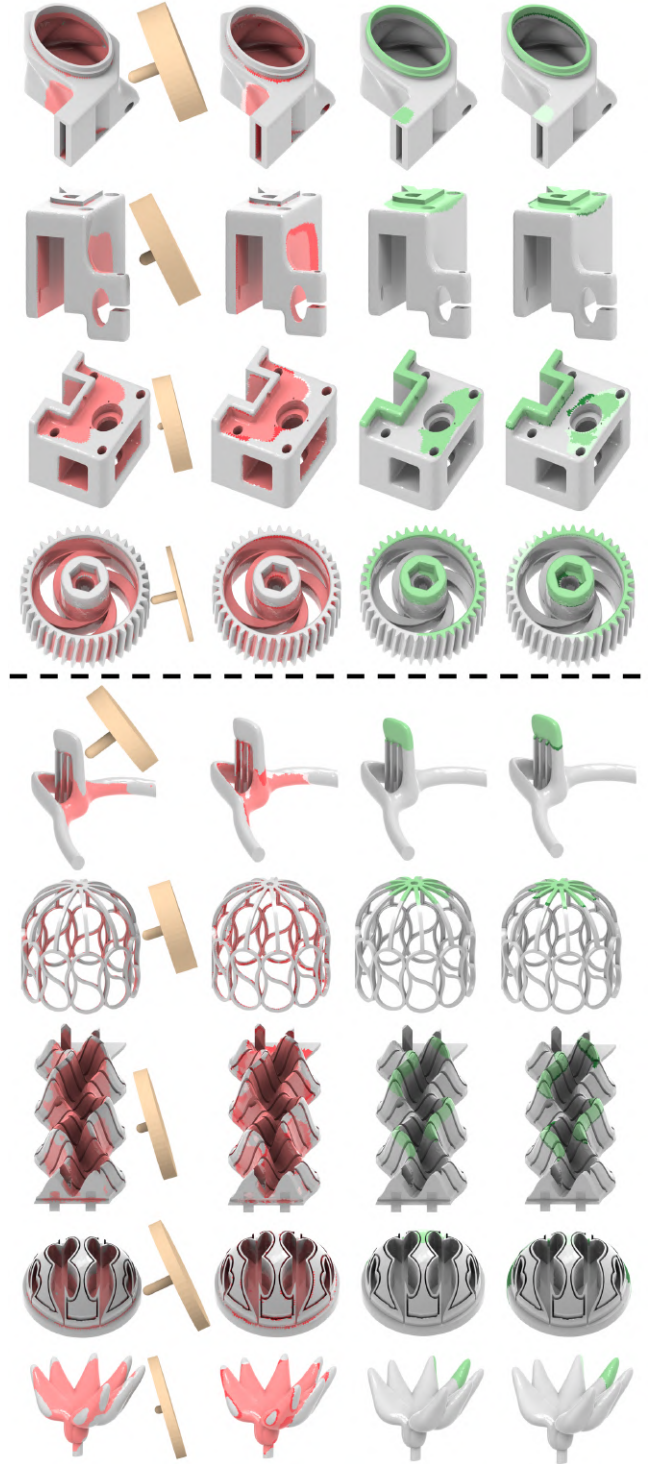


Fig. 6. The gallery of DeepMill prediction results. The top four are CAD shapes, and the bottom five are freeform shapes. The cutter size used for each shape is randomly generated. For each row of shapes, the first and third columns show the inaccessible and occlusion regions predicted by DeepMill. In the second and fourth columns, darker shades represent under-predicted areas, while lighter shades indicate over-predicted areas.

Table 1. Statistics of DeepMill and geometric method (section 4) on various datasets. The values in ( ) indicate the average number of mesh vertices in the dataset.  $Acc_i$  and  $Acc_o$  are the prediction accuracy for inaccessible points and occlusion points, respectively.  $F1_i$  and  $F1_o$  represent their F1-scores.  $T_i$  and  $T_o$  denote the calculation time (second) for inaccessible and occlusion points by geometric method, while  $T$  represents the total time. We do not statistically analyze occlusion points for complex models due to the excessive computation time required by geometric method.

DATASET	DeepMill					Geometric		
	$Acc_i$	$F1_i$	$Acc_o$	$F1_o$	$T$	$T_i$	$T_o$	$T$
CAD(7K)	96.3%	97.2%	98.3%	89.4%	0.01	4.0	25.1	29.1
CAD(15K)	96.3%	97.3%	98.3%	90.0%	0.01	17.3	207.4	224.7
Freeform(7K)	92.8%	93.7%	97.5%	86.5%	0.01	5.4	41.3	46.7
Freeform(15K)	93.2%	93.3%	98.0%	88.7%	0.02	19.8	373.1	392.9
Finely(100K+)	90.5%	90.0%	\	\	0.04	137.0	\	\

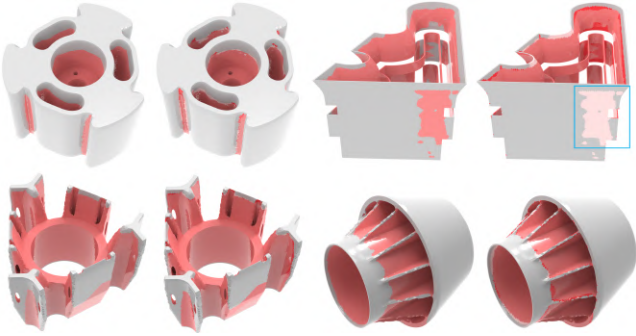


Fig. 7. Testing of DeepMill on finely-tessellated shapes. For each mesh, the left side shows the prediction results from DeepMill, while the right side displays the differences compared to the geometric method.

For different datasets, DeepMill exhibits varying performance: 1) On CAD datasets with geometric styles similar to the training set, DeepMill achieves up to 96.3% accuracy for predicting inaccessible regions. Since occlusion regions are derived from inaccessible regions, they present greater prediction challenges. 2) For freeform shapes, which feature fewer sharp or weak geometric characteristics and differ significantly from CAD shapes, the network maintains high prediction accuracy (92.8%, 86.5%), demonstrating its ability to effectively learn the geometric relationship between shapes and cutters. 3) The finely-tessellated dataset consists of meshes with a large number of triangular facets (over 100K), as shown in Figure 7. The intricate geometries increase prediction difficulty, and for unconventional structures that diverge significantly from the training set, DeepMill show reduced accuracy (e.g., top-right corner of Figure 7).

**Computation time.** DeepMill offers significant advantages over traditional geometric methods, achieving real-time predictions. For inaccessible and occlusion regions, DeepMill requires only **0.004%** of the total time needed by geometric method (section 4) on CAD shapes with 15K mesh vertices. For finely-tessellated shapes, the time is reduced to **0.029%** for inaccessible analysis only. Moreover,

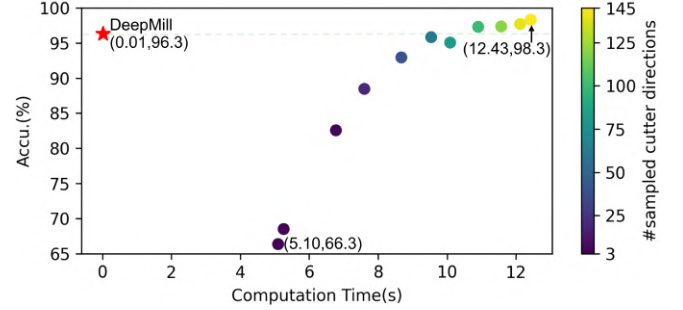


Fig. 8. Comparison between DeepMill (red star) and the geometric method (spot) [Mahdavi-Amiri et al. 2020] with varying numbers of sampled cutter directions. The color bar indicates the number of cutter directions used in the geometric method, and the vertical axis represents the accuracy of the computed inaccessible points.

Table 2. Comparison statistics of the cutter module in DeepMill for test sets with different cutter size. The baseline refers to the control group without the cutter module.  $Avg_i$  and  $Avg_f$  represent the average values of accuracy and F1-score, respectively. In the Uniform test set, the random range of cutter parameters is the same as in the training set. In the Short set, the ranges of  $CH$ ,  $FR$ , and  $FH$  are  $[0.1, 0.2]$ ,  $[80, 100]$ , and  $[0.1, 0.2]$ , respectively. In the Long set, the ranges are  $[10, 10.1]$ ,  $[5, 5.1]$ , and  $[10, 10.1]$ . In the Extreme set, the ranges are  $[20, 20.1]$ ,  $[5, 5.1]$ , and  $[20, 20.1]$ .

Cutter	Decoder	$Acc_i$	$F1_i$	$Acc_o$	$F1_o$	$Avg_{acc}$	$Avg_{f1}$
Uniform	Baseline	0.932	0.949	0.976	0.856	0.954	0.903
	Our	0.963	0.972	0.983	0.894	0.973	0.933
Short	Baseline	0.898	0.916	0.972	0.838	0.935	0.877
	Our	0.962	0.967	0.981	0.896	0.972	0.932
Long	Baseline	0.869	0.909	0.930	0.576	0.900	0.743
	Our	0.940	0.960	0.961	0.763	0.951	0.862
Extreme	Baseline	0.809	0.873	0.901	0.388	0.855	0.631
	Our	0.928	0.937	0.975	0.537	0.952	0.737

if finer cutter direction sampling is used in geometric methods, the time efficiency of DeepMill becomes even more pronounced.

To further demonstrate the advantage of DeepMill, we compared it with a single-threaded sampling-based approach [Mahdavi-Amiri et al. 2020] on the CAD (15K) dataset. This geometric method performs collision detection on triangle meshes and varies the number of sampled cutter directions to adjust the accuracy of inaccessible point estimation. As shown in Figure 8, DeepMill achieves a significant efficiency advantage while maintaining competitive accuracy.

#### 5.4 Ablation and Comparisons

**Ablation study for cutter module.** To demonstrate the effectiveness of the cutter module, we compared it with the baseline model without the cutter module. As shown in Table 2, DeepMill with the cutter module significantly outperforms the baseline across test sets with various cutter parameter ranges. This improvement is especially noticeable when using short and long cutters, as the baseline model tends to predict using an "average-sized cutter," resulting in a larger accuracy gap compared to DeepMill. It is worth noting that some



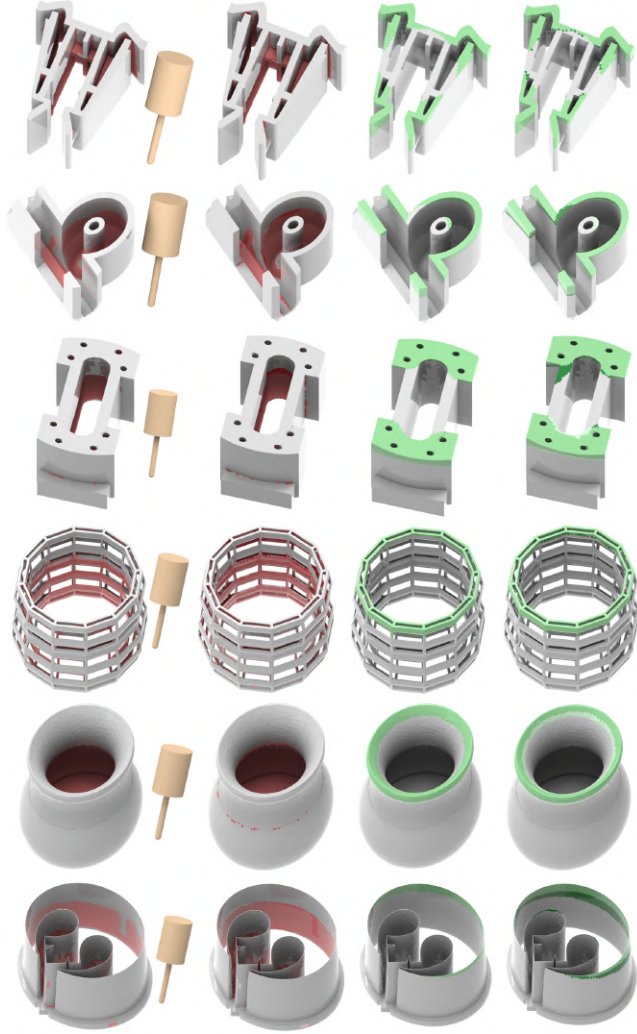


Fig. 9. Demonstration of DeepMill prediction results with extreme size of cutter. After adding the dataset generated with extreme cutters to the training set, DeepMill was able to extrapolate its prediction capability to cases involving extreme cutters.

regions of the shape are inherently "cutter-independent," meaning their accessibility remains unchanged regardless of cutter size. The baseline successfully learns these regions, maintaining reasonable prediction accuracy.

On the other hand, since shorter cutter sizes were used in the training set, the ratio of local collisions to global collisions is relatively small. As a result, DeepMill performs less accurately in predicting occlusion regions caused by extensive local collisions from extremely long cutters. To demonstrate it, we added 3K+ CAD shapes with extreme cutter sizes to the training set for calculating inaccessible and occlusion regions. The accuracy and F1-score for the two regions improved to 95.8% and 87.8%, respectively. Figure 9 shows several examples.

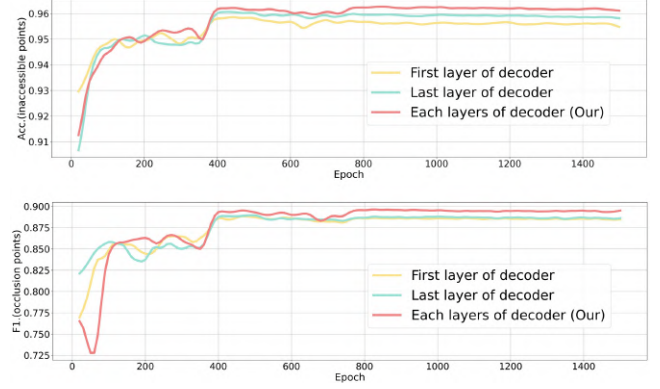


Fig. 10. Comparison of cutter module concatenation methods. The top and down show the prediction accuracy of inaccessible points and the F1 score of occlusion regions for different concatenation methods on the same test set. Our approach performs the best in both measures.



Fig. 11. Illustration of the effect of cutter length on inaccessible regions. Generally, longer cutters lead to fewer inaccessible regions.

*Comparison of cutter module positions.* We compared the impact of adding the cutter module at different positions in the decoder on the prediction results. As shown in Figure 10, adding the module to each layer, rather than only to the first or last layer of the decoder, enables the network to better learn the influence of cutter parameters on both local and global geometry.

*Comparison of different cutter sizes.* Figure 11 illustrates the effect of cutter length on inaccessible regions. Short cutter is more prone to collide with the shaft above cutter, resulting in larger inaccessible regions. On the other hand, longer cutter is less likely to cause collisions, allowing for more accessible regions.

*Comparison with other network.* To demonstrate the simplicity and effectiveness of DeepMill, we constructed GNN-based framework similar to the approach in [Huang et al. 2024b] for comparison. See Figure 12, we used the classic GraphSAGE model [Hamilton et al. 2017]. Compared to GCN [Kipf and Welling 2016] or GAT [Veličković et al. 2017], it is more suitable for high-resolution mesh-based graphs in accessibility analysis. However, significant global collisions exist between the cutter and  $M$ , and occlusion points are often topologically distant from inaccessible points, making it difficult for graph convolutions to efficiently capture this relationship.

In contrast, O-CNN-based architecture efficiently processes sparse 3D data through the octree structure, avoiding redundant computations. In accessibility analysis, CAD shapes often contain numerous holes, grooves, etc., and O-CNN effectively captures these key sparse geometric features. Additionally, the multi-scale convolution operation based on 3D space extracts shape features at different scales and



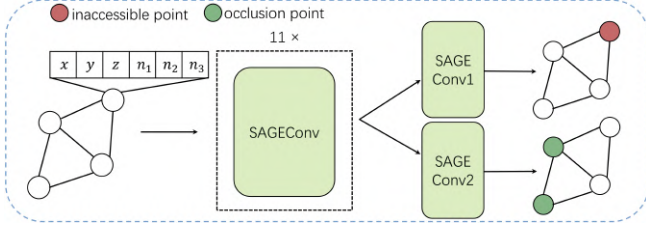


Fig. 12. Comparisons with GraphSAGE.  $M$  is converted into a graph, with nodes representing mesh vertices and edges representing topological connections of them. Similar to O-CNN, initial node features include vertex coordinates and normals. Node features are propagated and updated through successive convolutions on neighboring nodes.

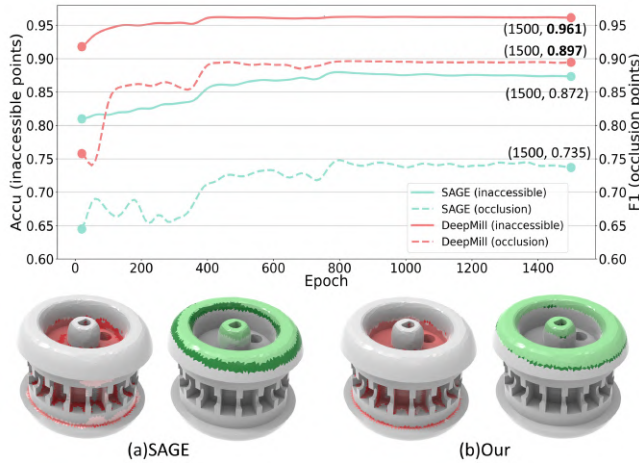


Fig. 13. Comparison with SAGE. DeepMill shows significantly better prediction capabilities for inaccessible and occlusion regions compared to SAGE.

is more effective at capturing collision relationships between distant positions on the mesh. Furthermore, U-Net’s encoder-decoder structure with skip connections enables it to capture both local features and global context, preserving the detailed geometric insights essential for predicting inaccessible points. Since rotating  $M$  alters its accessibility, rotational invariance is not applicable in this problem. Figure 13 presents the comparison results. DeepMill significantly outperforms GraphSAGE, demonstrating its superior ability to capture complex geometric interactions.

### 5.5 Discussion and Extension

**Geometric symmetry.** The geometric method in the dataset uses Fibonacci sphere sampling for evenly distributed cutter directions, as shown in Figure 14 (a). However, the directions lack axial symmetry, causing asymmetric inaccessible distributions for symmetric shapes, as shown in Figure 14 (b). Another sampling method based on spherical coordinates, shown in Figure 14 (c), achieves symmetric direction distribution but suffers from uneven spacing, easily missing directions. Surprisingly, DeepMill combines the strengths of both methods, learning to symmetrically adjust predictions from uniformly distributed directions. It produces more symmetric and

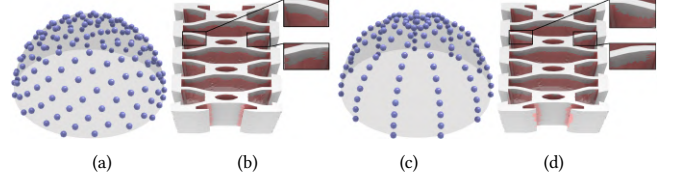


Fig. 14. Geometric symmetry illustration. (a) Non-axisymmetric cutter sampling causes asymmetric inaccessible regions (b). (c) Axisymmetric method has uneven distribution. (d) DeepMill combines both, yielding more symmetrical inaccessible regions.

reasonable inaccessibility regions when predicting geometrically symmetric shapes, as illustrated in Figure 14 (d).

**Volume accessibility analysis.** DeepMill can also be applied in accessibility analysis for other machining methods. For instance, rough machining removes the blank material layer by layer using a mill cutter, which is typically the pre-process for the finishing process. The accessibility analysis focuses on the interior volume of the blank. We use the same collision detection method to detect the interior sampling points of the bounding box of the input mesh to generate datasets. As shown in Figure 15, after training on the new dataset of over 5K CAD shapes, DeepMill can accurately predict the inaccessible regions within the volume. Compared to surface-based accessibility analysis, DeepMill can more easily predict accessibility within the volume, achieving an accuracy of up to 97.9%. Volume sampling uses voxelization, where adjacency resembles pixels in 2D, making it more suitable for 3D convolution operations.

## 6 CONCLUSION AND FUTURE WORK

This paper introduces DeepMill, a deep learning framework that improves cutter accessibility and manufacturability analysis for complex designs. Utilizing octree-based convolutional neural network (O-CNN), DeepMill efficiently predicts inaccessible regions and occlusions across various cutter sizes, overcoming the scalability and computational limitations of traditional methods. Its real-time predictions enable faster design iterations and enhance production efficiency. Additionally, the new dataset introduced supports further research and development of robust manufacturability analysis cutters, making DeepMill a significant advancement in the field. Extensive testing and comparisons have demonstrated DeepMill’s powerful cutter-aware prediction ability.

Based on the challenges and opportunities outlined, several directions for future work are proposed. First, to enhance the prediction performance of our current network, integrating an attention mechanism will be a promising approach. Second, incorporating geometric prior knowledge, such as symmetry, similarity, and topological properties, will allow the network to better capture the underlying structures. Another promising avenue involves incorporating cutters that work with irregular shapes in subtractive manufacturing, which will broaden the applicability of our framework to a wider range of real-world scenarios. Additionally, exploring the downstream applications of our current framework, such as path planning and model correction from non-accessible to accessible models, would be an exciting direction for further development.

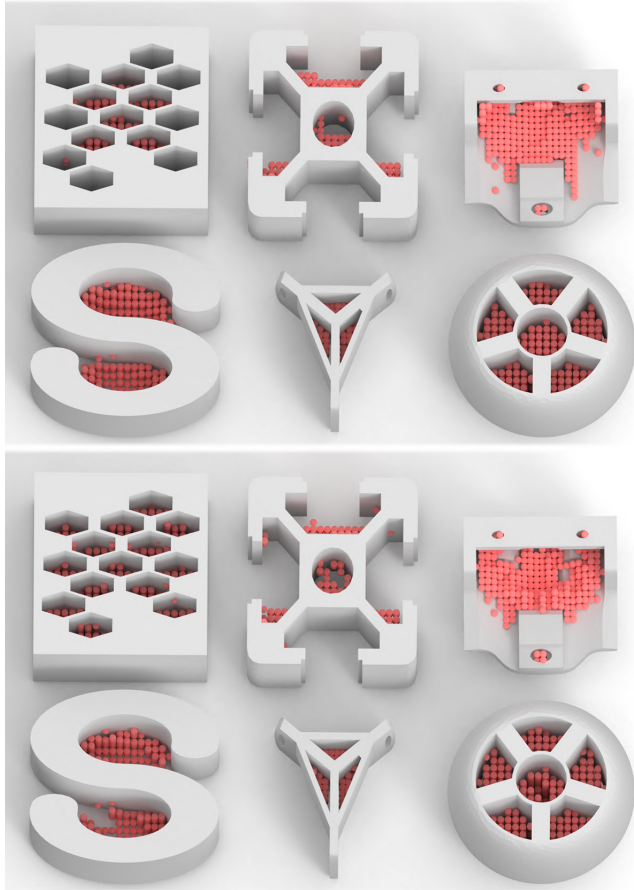


Fig. 15. Illustration of accessibility analysis within the volume. The red points represent inaccessible sampling points. On the top, the results predicted by DeepMill are shown, and on the bottom, the results obtained by the geometric method are displayed.

## ACKNOWLEDGMENTS

We thank all reviewers for their valuable comments and constructive suggestions. We thank Zhihao Zhang for assisting with the video production, as well as Yueze Zhu, Liyuan Wang and Jiaoyang Fulu for helping proofread the paper. Thanks for the models provided by ABC and Thingi10K. This work is supported in part by grants from National Key Research and Development Program of China (Grant No. 2024YFB3309500), National Natural Science Foundation of China (Grant No. U23A20312, 62472257, 62472258) and Beijing Natural Science Foundation (Grant No. 4244081).

## REFERENCES

- Mohamed Abdelaal. 2024. AI in Manufacturing: Market Analysis and Opportunities. *arXiv preprint arXiv:2407.05426* (2024).
- Vijay Manikrao Athawale and Shankar Chakraborty. 2010. A TOPSIS method-based approach to machine tool selection. In *Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management Dhaka, Bangladesh*, Vol. 2. 83–94.
- Mahadevan Balasubramaniam, Sanjay E Sarma, and Krzysztof Marciniak. 2003. Collision-free finishing toolpaths from visibility data. *Computer-Aided Design* 35, 4 (2003), 359–374.
- Aditya Balu, Sambit Ghadai, Soumik Sarkar, and Adarsh Krishnamurthy. 2020. Orthogonal distance fields representation for machine-learning based manufacturability analysis. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 83983. American Society of Mechanical Engineers, V009T09A044.
- Michael Bartoň, Michal Bizzarri, Florian Rist, Oleksii Slusarenko, and Helmut Pottmann. 2021. Geometry and tool motion planning for curvature adapted CNC machining. (2021).
- Li Chen, Tak Yu Lau, and Kai Tang. 2020. Manufacturability analysis and process planning for additive and subtractive hybrid manufacturing of Quasi-rotational parts with columnar features. *Computer-Aided Design* 118 (2020), 102759.
- Li Chen, Ke Xu, and Kai Tang. 2015. Collision-free tool orientation optimization in five-axis machining of bladed disk. *Journal of Computational Design and Engineering* 2, 4 (2015), 197–205.
- Niechen Chen and Matthew C Frank. 2021. Design for manufacturing: Geometric manufacturability evaluation for five-axis milling. *Journal of Manufacturing Science and Engineering* 143, 8 (2021), 081007.
- Juan Zaragoza Chichell, Alena Rečková, Michal Bizzarri, and Michael Bartoň. 2024. Collision-free tool motion planning for 5-axis CNC machining with toroidal cutters. *Computer-Aided Design* 173 (2024), 103725.
- Byoung K Choi, Dae H Kim, and Robert B Jerard. 1997. C-space approach to tool-path generation for die and mould machining. *Computer-Aided Design* 29, 9 (1997), 657–669.
- Christopher Choy, JunYoung Gwak, and Silvio Savarese. 2019a. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.
- Christopher Choy, JunYoung Gwak, and Silvio Savarese. 2019b. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3075–3084.
- Chengkai Dai, Charlie CL Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, and Yong-Jin Liu. 2018. Support-free volume printing by multi-axis motion. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.
- Savinder Dhaliwal, Satyandra K Gupta, Jun Huang, and Alok Priyadarshi. 2003. Algorithms for computing global accessibility cones. *J. Comput. Inf. Sci. Eng.* 3, 3 (2003), 200–209.
- Gershon Elber. 1994. Accessibility in 5-axis milling environment. *Computer-Aided Design* 26, 11 (1994), 796–802.
- Sambit Ghadai, Aditya Balu, Soumik Sarkar, and Adarsh Krishnamurthy. 2018. Learning localized features in 3D CAD models for manufacturability analysis of drilled holes. *Computer Aided Geometric Design* 62 (2018), 263–275.
- Georg Glaeser, Johannes Wallner, and Helmut Pottmann. 1999. Collision-free 3-axis milling and selection of cutting tools. *Computer-Aided Design* 31, 3 (1999), 225–232.
- Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 2018. 3D semantic segmentation with submanifold sparse convolutional networks. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.
- Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. 2021. PCT: Point cloud transformer. *Comput. Vis. Media* 7, 2 (2021).
- Satyandra K Gupta, William C Regli, Diganta Das, and Dana S Nau. 1997. Automated manufacturability analysis: a survey. *Research in Engineering Design* 9 (1997), 168–190.
- Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).
- George Harabin, Amir M Mirzendehtdel, and Morad Behandish. 2023. Deep Neural Implicit Representation of Accessibility for Multi-Axis Manufacturing. *Computer-Aided Design* 163 (2023), 103556.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- Michael Hoefer, Niechen Chen, and Matthew Frank. 2017. Automated manufacturability analysis for conceptual design in new product development. (2017).
- Bo Huang, Rui Huang, Xiuling Li, Hang Zhang, Zhen Wang, Kai He, and Shusheng Zhang. 2024b. Hierarchical Graph Neural Network for Manufacturability Analysis. *SSRN* (2024). <https://doi.org/10.2139/ssrn.5065158>
- Yuming Huang, Yuhu Guo, Renbo Su, Xingjian Han, Junhao Ding, Tianyu Zhang, Tao Liu, Weiming Wang, Guoxin Fang, Xu Song, et al. 2024a. Learning Based Toolpath Planner on Diverse Graphs for 3D Printing. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–16.
- Sergey Ioffe. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
- Benjamin Jones, Dalton Hildreth, Duowen Chen, Ilya Baran, Vladimir G Kim, and Adriana Schulz. 2021. Automate: A dataset and learning approach for automatic mating of cad assemblies. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–18.
- Sanjay Joshi and Tien-Chien Chang. 1988. Graph-based heuristics for recognition of machined features from a 3D solid model. *Computer-aided design* 20, 2 (1988), 58–66.
- SB Kailash, YF Zhang, and Jerry YH Fuh. 2001. A volume decomposition approach to machining feature extraction of casting and forging components. *Computer-Aided*

- Design* 33, 8 (2001), 605–617.
- Olivier Kerbrat, Pascal Mognot, and Jean-Yves Hascoët. 2011. A new DFM approach to combine machining and additive manufacturing. *Computers in Industry* 62, 7 (2011), 684–692.
- Yong Se Kim. 1990. *Convex decomposition and solid geometric modeling*. Stanford University.
- Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. ABC: A Big CAD Model Dataset For Geometric Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yuan-Shin Lee and Tien-Chien Chang. 1995. 2-phase approach to global tool interference avoidance in 5-axis machining. *Computer-Aided Design* 27, 10 (1995), 715–729.
- Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2018. PointCNN: Convolution on X-transformed points. In *Advances in Neural Information Processing Systems*.
- Ye Li and Matthew C Frank. 2006. Machinability analysis for 3-axis flat end milling. (2006).
- Yongshou Liang, Dinghua Zhang, Junxue Ren, Zezhong C Chen, and Yingying Xu. 2016. Accessible regions of tool orientations in multi-axis milling of blisks with a ball-end mill. *The International Journal of Advanced Manufacturing Technology* 85 (2016), 1887–1900.
- Changqing Liu, Yingguang Li, Sen Jiang, Zhongyu Li, and Ke Xu. 2020. A sequence planning method for five-axis hybrid manufacturing of complex structural parts. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 234, 3 (2020), 421–430.
- Tao Liu, Tianyu Zhang, Yongxue Chen, Yuming Huang, and Charlie CL Wang. 2024. Neural slicer for multi-axis 3D printing. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–15.
- 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 (TOG)* 39, 6 (2020), 1–15.
- Liane Makatura, Michael Foshey, Bohan Wang, Felix Hähnlein, Pingchuan Ma, Bolei Deng, Megan Tjandrasuwita, Andrew Spielberg, Crystal Owens, Peter Yichen Chen, et al. 2024a. How can large language models help humans in design and manufacturing? Part 1: Elements of the LLM-enabled computational design and manufacturing pipeline. *Harvard Data Science Review Special Issue* 5 (2024).
- Liane Makatura, Michael Foshey, Bohan Wang, Felix Hähnlein, Pingchuan Ma, Bolei Deng, Megan Tjandrasuwita, Andrew Spielberg, Crystal Owens, Peter Yichen Chen, et al. 2024b. How can large language models help humans in design and manufacturing? Part 2: Synthesizing an end-to-end LLM-enabled design and manufacturing workflow. *Harvard Data Science Review* (2024).
- Javvadi Eswara Manikanta, Nitin Ambhore, Amol Dhumal, Naveen Kumar Gurajala, and Ganesh Narkhede. 2024. Machine Learning and Artificial Intelligence Supported Machining: A Review and Insights for Future Research. *Journal of The Institution of Engineers (India): Series C* (2024), 1–11.
- Daniel Maturana and Sebastian Scherer. 2015a. VoxNet: A 3D convolutional neural network for real-time object recognition. In *IROS*.
- Daniel Maturana and Sebastian Scherer. 2015b. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 922–928.
- Wenlong Meng, Pengbo Bo, Xiaodong Zhang, Jixiang Hong, Shiqing Xin, and Changhe Tu. 2023. An efficient algorithm for approximate Voronoi diagram construction on triangulated surfaces. *Computational Visual Media* 9, 3 (2023), 443–459.
- Michał Piovarci, Michael Foshey, Jie Xu, Timothy Erps, Vahid Babaei, Piotr Didyk, Szymon Rusinkiewicz, Wojciech Matusik, and Bernd Bickel. 2022. Closed-loop control of direct ink writing via reinforcement learning. *arXiv preprint arXiv:2201.11819* (2022).
- Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.
- Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*.
- William Clement Regli III. 1995. *Geometric algorithms for recognition of features from solid models*. University of Maryland, College Park.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18. Springer, 234–241.
- Takafumi Saito and Tokiichiro Takahashi. 1991. NC machining with G-buffer method. *ACM Siggraph Computer Graphics* 25, 4 (1991), 207–216.
- Syaimak Abdul Shukor and DA Axinte. 2009. Manufacturability analysis system: issues and future trends. *International Journal of Production Research* 47, 5 (2009), 1369–1390.
- Mohsen Soori, Behrooz Arezoo, and Roza Dastres. 2023. Machine learning and artificial intelligence in CNC machine tools, a review. *Sustainable Manufacturing and Service Economics* 2 (2023), 100009.
- Antonia J Spyridi and Aristides AG Requicha. 1990. Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines. In *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1284–1289.
- Alan Sullivan, Huseyin Erdim, Ronald N Perry, and Sarah F Frisken. 2012. High accuracy NC milling simulation using composite adaptively sampled distance fields. *Computer-Aided Design* 44, 6 (2012), 522–536.
- Abbas Vafaesef, and Hoda A ElMaraghy. 1998. Accessibility analysis in 5-axis machining of sculptured surfaces. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, Vol. 3. IEEE, 2464–2469.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- Nicolai N Vorobiev. 2002. *Fibonacci numbers*. Springer Science & Business Media.
- Peng-Shuai Wang. 2023. OctFormer: Octree-based Transformers for 3D Point Clouds. *ACM Trans. Graph. (Proc. SIGGRAPH)* 42, 4 (2023).
- Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. 2017a. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph. (Proc. SIGGRAPH)* 36, 4 (2017).
- Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. 2017b. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)* 36, 4 (2017), 1–11.
- Peng-Shuai Wang, Yang Liu, and Xin Tong. 2020a. Deep octree-based CNNs with output-guided skip connections for 3D shape and scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 266–267.
- YuanBin Wang, Pai Zheng, Tao Peng, HuaYong Yang, and Jun Zou. 2020b. Smart additive manufacturing: Current artificial intelligence-enabled methods and future perspectives. *Science China Technological Sciences* 63, 9 (2020), 1600–1611.
- Tony C Woo. 1994. Visibility maps and spherical algorithms. *Computer-Aided Design* 26, 1 (1994), 6–16.
- Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A deep representation for volumetric shape modeling. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.
- Qun-Ce Xu, Tai-Jiang Mu, and Yong-Liang Yang. 2023. A survey of deep learning-based 3D shape generation. *Computational Visual Media* 9, 3 (2023), 407–442.
- Xiaoliang Yan and Shreyes Melkote. 2023. Automated manufacturability analysis and machining process selection using deep generative model and Siamese neural networks. *Journal of Manufacturing Systems* 67 (2023), 57–67.
- Xijun Zhang, Dianming Chu, Xinyue Zhao, Chenyu Gao, Lingxiao Lu, Yan He, and Wenjuan Bai. 2024. Machine learning-driven 3D printing: A review. *Applied Materials Today* 39 (2024), 102306.
- Ying Zhang, Sheng Yang, and Yaoyao Fiona Zhao. 2020. Manufacturability analysis of metal laser-based powder bed fusion additive manufacturing—a survey. *The International Journal of Advanced Manufacturing Technology* 110 (2020), 57–78.
- Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. 2021. Point transformer. In *IEEE International Conference on Computer Vision (ICCV)*.
- Haisen Zhao, Hao Zhang, Shiqing Xin, Yuanmin Deng, Changhe Tu, Wenping Wang, Daniel Cohen-Or, and Baoquan Chen. 2018. DSCarver: decompose-and-spiral-carve for subtractive manufacturing. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.
- Fanchao Zhong, Haisen Zhao, Haochen Li, Xin Yan, Jikai Liu, Baoquan Chen, and Lin Lu. 2023. VASCO: Volume and Surface Co-Decomposition for Hybrid Manufacturing. *ACM Trans. Graph.* 42, 6 (2023), 1–17.
- Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).