

# Curve-Based Slicer for Multi-Axis DLP 3D Printing

CHENGKAI DAI\*, Centre for Perceptual and Interactive Intelligence, Hong Kong, China

TAO LIU\*, The University of Manchester, United Kingdom

DEZHAO GUO, The Chinese University of Hong Kong, Hong Kong, China

BINZHI SUN, The Chinese University of Hong Kong / Centre for Perceptual and Interactive Intelligence, Hong Kong, China

GUOXIN FANG, The Chinese University of Hong Kong / Centre for Perceptual & Interactive Intelligence, Hong Kong, China

YEUNG YAM, The Chinese University of Hong Kong / Centre for Perceptual and Interactive Intelligence, Hong Kong, China

CHARLIE C.L. WANG<sup>†</sup>, The University of Manchester, United Kingdom

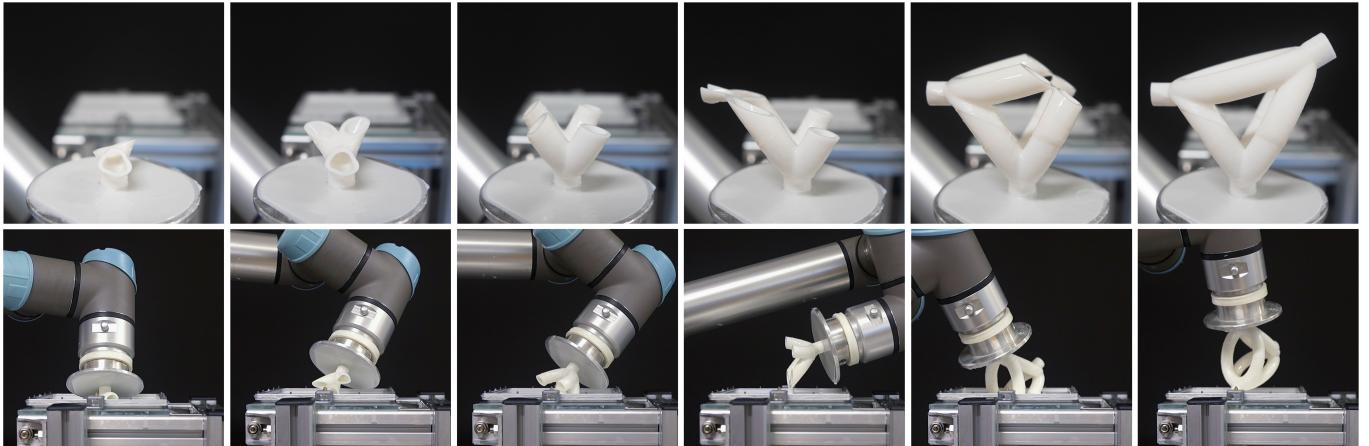


Fig. 1. Our method generates curve-based robotic motion trajectories to enable multi-axis DLP 3D printing. The progressive results illustrate the fabrication process of a Toroidal-Tubes model, showcasing the effectiveness of our approach in handling challenging topologies without support structures.

This paper introduces a novel curve-based slicing method for generating planar layers with dynamically varying orientations in digital light processing (DLP) 3D printing. Our approach effectively addresses key challenges in DLP printing, such as regions with large overhangs and staircase artifacts, while preserving its intrinsic advantages of high resolution and fast printing speeds. We formulate the slicing problem as an optimization task, in which parametric curves are computed to define both the slicing layers and the

model partitioning through their tangent planes. These curves inherently define motion trajectories for the build platform and can be optimized to meet critical manufacturing objectives, including collision-free motion and floating-free deposition. We validate our method through physical experiments on a robotic multi-axis DLP printing setup, demonstrating that the optimized curves can robustly guide smooth, high-quality fabrication of complex geometries.

CCS Concepts: • Computing methodologies → Shape modeling; • Applied computing → Engineering.

Additional Key Words and Phrases: Curve-Based Slicer, Optimization, Digital Light Processing, Multi-axis 3D Printing

## ACM Reference Format:

Chengkai Dai, Tao Liu, Dezhao Guo, Binzhi Sun, Guoxin Fang, Yeung Yam, and Charlie C.L. Wang. 2025. Curve-Based Slicer for Multi-Axis DLP 3D Printing. In *Proceedings of (ACM SIGGRAPH Asia 2025)*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3763352>

\*Joint first authors.

<sup>†</sup>Corresponding author: charlie.wang@manchester.ac.uk (Charlie C.L. Wang).

Authors' Contact Information: Chengkai Dai, ckdai@cpii.hk, Centre for Perceptual and Interactive Intelligence, Hong Kong, China; Tao Liu, tao.liu@manchester.ac.uk, The University of Manchester, Manchester, United Kingdom; Dezhao Guo, dzguo@mae.cuhk.edu.hk, The Chinese University of Hong Kong, Hong Kong, China; Binzhi Sun, bzsun@cpii.hk, The Chinese University of Hong Kong / Centre for Perceptual and Interactive Intelligence, Hong Kong, China; Guoxin Fang, guoxinfang@cuhk.edu.hk, The Chinese University of Hong Kong / Centre for Perceptual & Interactive Intelligence, Hong Kong, China; Yeung Yam, yyam@mae.cuhk.edu.hk, The Chinese University of Hong Kong / Centre for Perceptual and Interactive Intelligence, Hong Kong, China; Charlie C.L. Wang, charlie.wang@manchester.ac.uk, The University of Manchester, Manchester, United Kingdom.

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).

ACM SIGGRAPH Asia 2025, Hong Kong, China

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-x-xxxx-xxxx-x/YYY/MM  
<https://doi.org/10.1145/3763352>

## 1 Introduction

Digital Light Processing (DLP) 3D printing is valued for its high resolution and fast speed, but conventional systems use a single vertical build axis with fixed planar layers. This restriction often requires extensive supports for overhangs, leading to surface damage, increased material waste, and difficult to remove challenges [Zhang et al. 2015]. Incorporating multi-axis motion offers a way to overcome these limitations and improve the manufacturability of complex geometries.

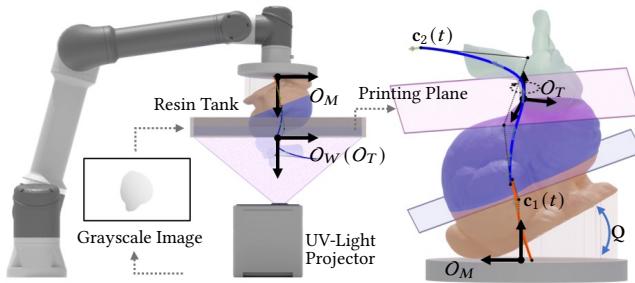


Fig. 2. An illustration for the working principle and the layers of multi-axis DLP 3D printing: the model partition and the multiple trajectories of printing can be formed by multiple curves (displayed in different colors), where material layers are printed at the bottom of resin tank with thickness controlled by the grayscale of the UV-light projection.  $O_M$  represents the frame of MCS and  $O_T$  denotes the frame defined in MCS for each individual printing plane.  $O_W$  is the frame of WCS placed at the center of resin tank for the sake of computation. During printing,  $O_T$  is aligned with  $O_W$  by inverse kinematic computing.

Multi-axis strategies have been explored primarily in fused filament fabrication, where dynamically adjusted tool orientations enable support-free printing [Dai et al. 2018], improved surface quality [Etienne et al. 2019], and enhanced mechanical strength [Fang et al. 2020; Zhang et al. 2022]. Applying this concept to DLP is more difficult: its projection process solidifies an entire resin layer at once, preventing free orientation changes within the layer. Instead, multi-axis DLP adjusts orientations between successive layers (Fig. 1), exploiting grayscale control of UV light-based curing to modulate local thickness [Luongo et al. 2020].

### 1.1 Challenges

The working principle of multi-axis DLP 3D printing can be described as progressively positioning a sequence of oriented planes,  $\{\mathcal{P}_{i=0,1,\dots,n}\}$ , that align with the bottom of the resin tank (as illustrated in Fig. 2). These planes define layers generated by slicing the target model  $\mathcal{M}$  along different orientations while several manufacturing objectives must be satisfied to ensure successful fabrication.

- **Collision-Free:** The pose used to realize each planar layer  $\mathcal{P}_i$  should ensure the already printed portion of the 3D model remains collision-free with environmental obstacles.
- **Connectivity:** Each newly formed layer  $\mathcal{P}_i$  must maintain connection with the previously printed structure to prevent floating regions during fabrication.
- **Surface Quality:** To mitigate steep overhangs and minimize staircase artifacts, the angles between the normal vectors of  $\mathcal{P}_i$  and the surface of  $\mathcal{M}$  should be optimized.
- **Smooth Motion:** The planar layers should minimize abrupt orientation changes to ensure smooth, stable, and fast hardware motion.

These objectives must be satisfied for all planar layers. For each planar layer  $\mathcal{P}_i$ , the already printed part refers to the subset of  $\mathcal{M}$  located ‘below’  $\mathcal{P}_i$  in the model coordinate system (MCS) (i.e., the regions highlighted in blue and brown in Fig. 2).

The primary challenge is to determine feasible layers that are both collision-free and connected, while also optimizing other manufacturing objectives. Although most objectives can be formulated based on layers defined in MCS, ensuring global collision avoidance between the printed model and environmental obstacles requires additional considerations. Specifically, this involves the local frame defined on each layer and the model’s setup orientation within the MCS. The setup orientation of a model  $\mathcal{M}$  must also be optimized, which can be represented by a transformation  $Q \in \text{SE}(3)$  applied to  $\mathcal{M}$  in the MCS. The motion of the build platform is controlled by aligning the local frames of individual layers with the resin tank’s frame positioned at the center of the tank, defined in the world coordinate system (WCS). To address these challenges, we propose a curve-based method that optimizes the planar layers and their corresponding local frames simultaneously.

### 1.2 Our Method and Contribution

The slicing problem of multi-axis DLP 3D printing involves computing an optimized sequence of planar layers and their corresponding frames. This task can be viewed as a trajectory planning problem that must satisfy the aforementioned manufacturing objectives. In this work, we propose to represent the trajectory as a Bézier curve  $\mathbf{c}(t)$  ( $\forall t \in [0, 1]$ ), where its tangent planes define the slicing layers and the corresponding frames are given with the help of the Bishop frames of the curve. The control points  $\{\mathbf{p}_k = (x_k, y_k, z_k, \theta_k) \in \mathbb{R}^4\}$  are treated as variables to optimize the printing trajectory. The first three components of each control point define the shape and frames of the curve in MCS, while the fourth component introduces an additional rotation of the frame in the tangent planes. The manufacturing objectives are encoded as loss functions, defined in terms of these control points and the model’s setup transformation in MCS. These objectives are then minimized using a stochastic gradient descent (SGD) solver to achieve an optimal slicing strategy.

To ensure scalability, we extend the curve-based slicing algorithm to support multiple curves  $\{\mathbf{c}_{j=1,2,\dots,m}(t)\}$ , allowing the trajectory in multiple segments to be jointly optimized with the partitions. Specifically, each curve  $\mathbf{c}_j(t)$  is responsible for generating layers and trajectories for the  $i$ -th region of  $\mathcal{M}$  that is *above* the tangent plane at  $\mathbf{c}_j(0)$  and *below* the tangent plane of the next region at  $\mathbf{c}_{j+1}(0)$ . Note that the second condition is omitted for the last curve (i.e., when  $j = m$ ). This partitioning divides the model  $\mathcal{M}$  into convex regions using the half-spaces defined by the tangent planes at the starting points of the curves (see Fig. 2 for an illustration). Similar space partitioning strategies were employed in [Wu et al. 2020] for multi-directional 3D printing. However, unlike prior methods where the printing direction remains constant within each partition, our approach optimizes the trajectory within each region. The control points of all curves are optimized jointly, allowing the space partitions to evolve dynamically during the optimization process. Moreover, a refinement scheme is introduced to add curves adaptively.

The technical contributions of our approach are as follows.

- We formulate the slicing problem for multi-axis DLP 3D printing as a trajectory optimization problem, with the slicing trajectories represented as parametric curves to be optimized.

- Our formulation enables an efficient solution using gradient-based solvers and smooth realization on a robotic system.
- To ensure printability for points inside a solid, we first define a continuous function that encodes their printing state based on the trajectory curve to be optimized, and then impose constraints requiring that already printed points remain outside environmental obstacles.
  - Our computational framework supports slicing with multiple curves (trajectories). Tangent planes at the starting points of these curves define a partition of space, enabling the partition to be jointly optimized together with individual trajectories.
  - Manufacturing objectives, such as floating-free, cliff-angle, surface quality, and completeness, can be seamlessly formulated and integrated into our slicer framework for multi-axis DLP 3D printing.

To the best of our knowledge, this is the first slicer that optimizes the continuous change of layer orientations to ensure the manufacturability of complex freeform models in multi-axis DLP 3D printing. We have demonstrated the effectiveness of our approach through computational and physical experiments across a variety of models.

## 2 Literature Review

### 2.1 DLP 3D Printing

DLP 3D printing is a projection-based vat photopolymerization technique that delivers exceptional resolution and accuracy by selectively curing photopolymer resin through controlled, layer-by-layer light exposure [Chaudhary et al. 2023]. Recent advances have significantly expanded the range of photopolymerizable materials suitable for DLP printing [Nam and Kim 2024]. In addition, sophisticated material processing techniques have been introduced to enable precise control over layer thickness [Luongo et al. 2020], colors [Peng et al. 2022], and even spatially varying mechanical properties such as elasticity [Kuang et al. 2019; Yue et al. 2023]. These capabilities are achieved by modulating the grayscale intensity of projected images during resin curing, offering fine-grained control over the printed structure's material distribution.

Despite the advantages of DLP 3D printing in speed and quality, it shares with other parallel planar-layer (PPL) techniques the limitation of requiring support structures for large overhangs and floating points, as illustrated by the failed print in Fig. 3. This issue is especially problematic for interior regions, such as the tubular features in the Toroidal-Tubes model (Fig. 1), where supports cannot be removed post-fabrication. Prior strategies include generating slimmed-down supports [Cao et al. 2025; Huang et al. 2014; Vanek et al. 2014], optimizing model orientation to reduce visible artifacts [Liu et al. 2021; Zhang et al. 2015], and deforming input geometry to minimize support demand [Hu et al. 2015]. Another drawback of PPL-based printing is staircase artifacts, commonly mitigated by adaptive layer thickness [Mao et al. 2019], also explored in filament fusion printing [Wang et al. 2015; Zhong et al. 2023a]. Decomposing models into sub-components printed along different fixed directions has also been attempted [Cao et al. 2021; Wang et al. 2016; Zhong et al. 2023b]. However, none of these methods tackles the fundamental cause of both problems – the globally fixed build direction

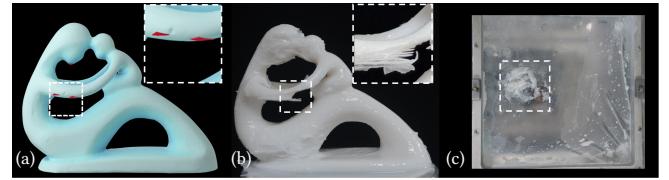


Fig. 3. Failure case of PPL-based DLP printing due to floating points and regions with large overhangs, highlighted in red color in (a). Directly printing these unsupported regions results in (b) print failure and (c) residual material accumulation at the bottom of the resin tank.

of planar layers – which motivates the approach proposed in this paper.

Recently, multiple degrees-of-freedom (DoF) motion in DLP 3D printing has been demonstrated by a robotic system in [Huang et al. 2021], which was further extended with a dynamic slicing algorithm based on skeleton-guided strategies [Huang et al. 2022]. However, their method follows a deterministic approach to handle simple geometries, and the slicing layers are not optimized to improve either manufacturability or surface quality. Separately, various robotic systems have been explored to increase the printable volume in DLP setups [Yi et al. 2018, 2016]. Zhao et al. [Zhao et al. 2013] employed DLP printing for adaptive repair of damaged parts through multi-axis tool motion; however, their approach was based on line-wise consolidation rather than layer-wise fabrication. In contrast, we propose a novel computational pipeline that generates planar layers with dynamically varying orientations to tackle the core challenges in DLP-based 3D printing.

### 2.2 Multi-Axis 3D Printing

Multi-axis 3D printing has emerged as a rapidly expanding research area due to the advantages offered by high DoF in motion, which enable material deposition along wireframes of lightweight structures [Huang et al. 2016; Wu et al. 2016], curved layers within solid volumes [Dai et al. 2018], and shell structures [Mitropoulou et al. 2020]. In general, multi-axis 3D printing can more effectively handle geometries with large overhangs and also enables material deposition along principal stress directions, thereby enhancing mechanical performance [Fang et al. 2020; Tam and Mueller 2017]. These capabilities are often achieved by computing scalar fields whose isosurfaces satisfy both design and manufacturability requirements, and are then extracted as working surfaces for printing.

Such printing layers have also been generated through optimized spatial deformation, forming layers that are either conformal or perpendicular to the model surface to improve surface quality [Etienne et al. 2019]. Building upon this idea, deformation-based optimization has been incorporated into multi-objective frameworks that jointly consider mechanical strength and support-free fabrication [Liu et al. 2024; Zhang et al. 2022]. Most recently, this line of research of multi-axis 3D printing has been extended to the simultaneous optimization of structural topology, manufacturable layers, and path orientations for the 3D printing of fiber-reinforced composites [Liu et al. 2025]. However, the layers produced in these works are generally curved

surfaces, which are not directly compatible with DLP printing due to its reliance on planar, projection-based fabrication.

On the other aspect, skeleton-based model decomposition was used in [Wu et al. 2017] to segment an input model so that each segment could be fabricated using PPL-based 3D printing along distinct but constant directions. This approach was later extended to support 3D printing with both 4-axis and 5-axis motion systems [Wu et al. 2020]. However, similar to the previously discussed DLP slicer [Huang et al. 2022], these methods do not generate dynamically changing printing directions within sub-regions, nor do they optimize the segmentation based on manufacturing objectives. Notably, the model decomposition and associated layer generation method presented in [Wu et al. 2020] can be interpreted as a special – yet unoptimized – case of the solutions produced by our method. In particular, their approach corresponds to the scenario where each sub-region is represented by a straight slicing curve.

### 2.3 Robot-Assisted Manufacturing & Motion Planning

Robot-assisted manufacturing has seen growing adoption across both additive and subtractive processes. Beyond multi-axis 3D printing enabled by robotic hardware, examples include robotic sculpting [Ma et al. 2020] and curved hot-wire cutting of double-curved surfaces [Duenser et al. 2020]. A central challenge across these processes is avoiding collisions between moving tools and environmental obstacles, which complicates motion planning. Since explicit collision detection can be computationally expensive [Huang et al. 2016], surrogate models of the environment have been proposed using implicit representations (e.g., [Chen et al. 2025; Dai et al. 2018]). In DLP-based systems, collision risks are even more complex, as they may occur between the resin tank and the dynamically evolving printed part, directly coupling with the slicing process. To address this, we define a state function that represents the time-varying geometry of the part derived from optimized slicing curves, capturing both layer geometry and printing sequence.

More broadly, collision avoidance relates to swept volume computation (e.g., [Kim et al. 2004; von Dziegielewski et al. 2010]), which models the space occupied by moving objects. Recent work extends this with implicit swept volumes using signed distance functions for continuous collision-free trajectory generation [Wang et al. 2024]. Finally, while our curve-based slicing optimization is conceptually related to curve-based motion planning (e.g., [Marcucci et al. 2023; Pan et al. 2012]), our focus is distinct in targeting manufacturability constraints and print quality optimization.

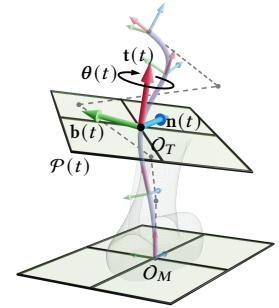
## 3 Overview

### 3.1 Curve Representation, Frames and Variables

We represent the motion trajectory of the build platform by a Bézier curve  $\tilde{c}(t) \in \mathbb{R}^4$  with  $t \in [0, 1]$  due to its compact formulation and the efficient schemes for evaluation and subdivision by the de casteljau algorithm. Specifically, the curve is given by  $\tilde{c}(t) = \sum_{k=0}^n p_k B_k^n(t)$ , where the control points  $p_k = (x_k, y_k, z_k, \theta_k) \in \mathbb{R}^4$  define both the spatial trajectory and a rotational DoF. The control points are blended together by the Bernstein basis functions  $B_k^n(t)$  in order  $n$ . For clarity, we denote the 3D spatial curve as  $c(t)$  using the  $(x, y, z)$ -components of  $\tilde{c}(t)$ , and the angular profile as  $\theta(t)$  using

the  $\theta$ -component.  $c(t) \in \mathbb{R}^3$  is defined in the model coordinate system (MCS).

At each parameter  $t$ , a tangent plane is defined by the point  $c(t)$  and its tangent vector  $t(t) = \dot{c}(t)$ . This plane serves as the dynamically changed printing plane  $\mathcal{P}(t)$ , whose intersection with the model  $\mathcal{M}$  yields a printable layer for DLP fabrication. To define valid local frames along the curve, we adopt the Bishop frame  $(t(t), n(t), b(t))$ , which avoids issues associated with degeneracies in the normal vector [Wang et al. 2008]. We then augment this frame with a rotation of  $\theta(t)$  around the tangent vector  $t(t)$ , defining the full tooling frame  $O_T$  used for printing (see the insert figure above).



For physical fabrication, we compute a rigid body transformation that aligns  $O_T$  with the world coordinate system (WCS), specifically with a reference frame  $O_W$  placed at the center of the resin tank's bottom surface as illustrated in Fig. 2. This alignment is represented by a transformation  $T(t) \in SE(3)$ , which maps any point  $q \in \mathbb{R}^3$  (in homogeneous coordinates) in the model  $\mathcal{M}$  to a point in WCS as  $T(t)q$ .

Both the printing plane  $\mathcal{P}(t)$  and the transformation  $T(t)$  are determined by the control points  $\{p_k\}$ , which serve as the design variables in our trajectory optimization for multi-axis DLP printing. Additionally, we can optimize the initial orientation of the model  $\mathcal{M}$  by introducing a global transformation  $Q \in SE(3)$ , treated as an additional variable. Under this formulation, a model point  $q$  is transformed to  $Qq$  in MCS, and further to  $T(t)Qq$  in WCS.

### 3.2 Optimization for Single and Multiple Curves

We formulate the curve-based slicing problem as an optimization problem, where the goal is to minimize a set of manufacturing objectives. The optimization variables include the control points  $\{p_k\}$  of the Bézier curve and the global model orientation  $Q$ . Formally, it gives

$$\begin{aligned} \min_{\{p_k\}, Q} \quad & \mathcal{L}(\{p_k\}, Q, \mathcal{M}) \\ \text{s.t.} \quad & \Gamma(\{p_k\}, Q, \mathcal{M}) = 0 \end{aligned} \quad (1)$$

where  $\mathcal{L}(\cdot)$  aggregates design goals such as surface quality requirements. Meanwhile,  $\Gamma(\cdot)$  ensures the manufacturability of intermediate shapes, including collision-free, connectivity and completeness. The specific formulation about how to evaluate these terms are detailed in Sec. 4.

In practice, it is *not* always able to find a single curve that makes the constraint  $\Gamma(\cdot) = 0$  being satisfied. Therefore, we extend the formulation to support multiple curves as

$$\begin{aligned} \min_{\{p_k^j\}, Q} \quad & \sum_{j=1}^m \mathcal{L}_j(\{p_k^j\}, Q, \mathcal{M}) \\ \text{s.t.} \quad & \Gamma_j(\{p_k^j\}, Q, \mathcal{M}) = 0 \quad (j = 1, \dots, m) \end{aligned} \quad (2)$$

where  $\{p_k^j\}$  are the control points of the  $j$ -th curve  $c_j(t)$ .  $\mathcal{L}_j(\cdot)$  and  $\Gamma_j(\cdot)$  are piecewise functions that only have non-zero value for the

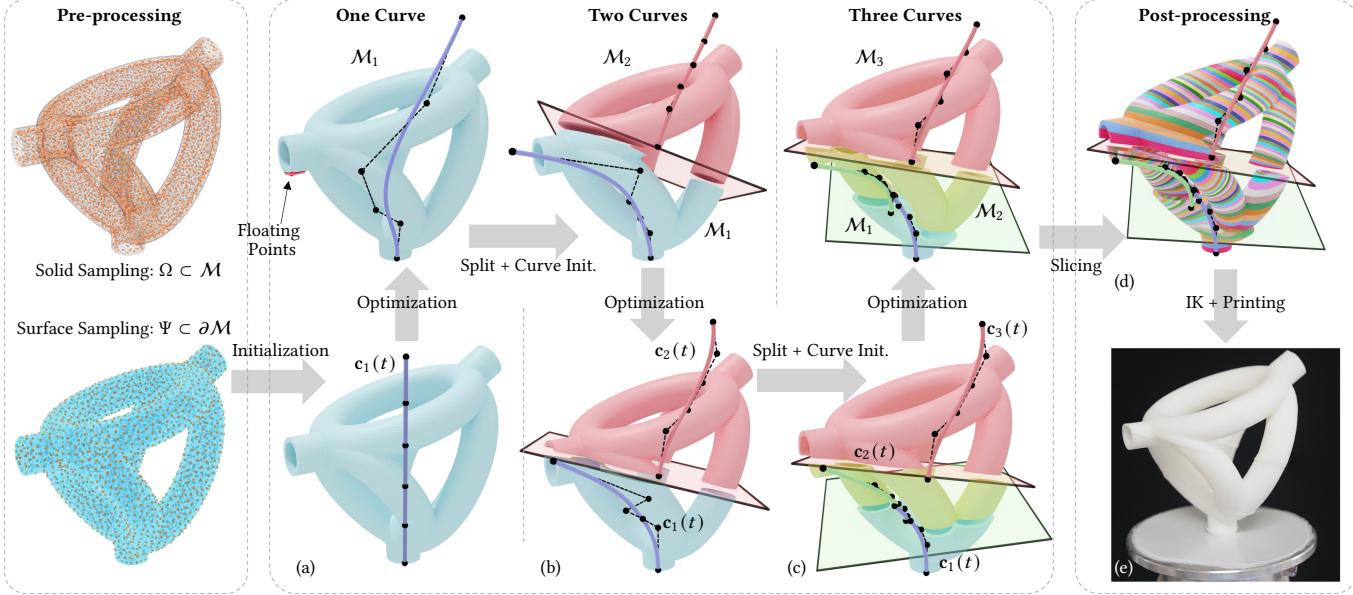


Fig. 4. After preparing the computational domain as a set of sample points  $\Omega$  for the solid  $\mathcal{M}$  and a set of surface sample point  $\Psi$ , our framework is based on the adaptively applied steps of optimization and splitting. (a) Starting from a straight curve as the trajectory of DLP printing, we optimize the trajectory by changing the positions of its control points (black dots). When critical manufacturing requirement such as floating-free (with floating point highlighted by the arrows), the curve is split into two curves to be further optimized (b). Note that the region partition defined by the curves are co-optimized jointly with the curves as trajectories. When hard constraints are still not satisfied in the region covered by  $c_1(t)$ ,  $c_1(t)$  is further adaptively subdivided into two – i.e., (c) there are three curves in total to be optimized to obtain the slicing result (d). The resultant curves define the trajectories of motion, which can be converted into the execution of robotic system by the computation of inverse kinematics to fabricate the final result (e).

$j$ -th region of  $\mathcal{M}$  as

$$\mathcal{M}_j := \{q \in \mathcal{M} \mid \forall k \leq j, (\mathbf{Q}q - c_k(0)) \cdot \dot{c}_k(0) \geq 0 \wedge (\mathbf{Q}q - c_{j+1}(0)) \cdot \dot{c}_{j+1}(0) < 0\}. \quad (3)$$

In other words, the tangent planes of curves at their starting points define ordered half-spaces to divide the input model  $\mathcal{M}$  in partitions  $\{\mathcal{M}_j\}$ . Note that for the last sub-region  $j = m$ , the last condition will not be applied as the curve  $c_{m+1}(\cdot)$  does not exist. Note that, the transformed position  $\mathbf{Q}q$  is employed here for the membership classification.

### 3.3 Computational Pipeline

We develop an adaptive refinement algorithm to generate slicing layers via multiple curves. The input model  $\mathcal{M}$  is processed as follows:

- **Step 1:** Sample the solid model  $\mathcal{M}$  into a set of points  $\Omega$  used to evaluate manufacturing objectives such as collision-free motion, connectivity, and completeness.
- **Step 2:** Sample  $\mathcal{M}$ 's surface boundary  $\partial\mathcal{M}$  as a point set  $\Psi$  for surface quality and support-free objectives.
- **Step 3:** Initialize the curve set  $C = \{c_1(t)\}$  with a single vertical curve  $c_1(t)$  that spans the height of  $\mathcal{M}$ .
- **Step 4:** Optimize all curves in  $C$  by updating the design variables using a gradient-based solver according to Eq. (2).
- **Step 5:** For each sub-region  $\mathcal{M}_j$ , check whether the condition  $\Gamma_j(\cdot) = 0$  is satisfied along the trajectory defined by  $c_j(t)$ . If

not, split the curve into two new curves and update  $C$  by replacing  $c_j(t)$  with the resulting pair.

- **Step 6:** Repeat from Step 4 until all curves satisfy the condition  $\Gamma_j(\cdot) = 0$ .
- **Step 7:** Adaptively slice the model  $\mathcal{M}$  in all sub-regions  $\{\mathcal{M}_j\}$  by the tangent planes of  $c_j(t)$  to generate layers satisfying the manufacturable layer thickness.

Figure 4 illustrates the progressive refinement of the curve set  $C$  when applied to a geometrically complex model, Toroidal-Tubes.

Unlike prior approaches that partition space deterministically (e.g., [Wu et al. 2017, 2020]), our method co-optimizes the space partition and slicing curves. During the gradient-based optimization in Step 4, the first control points of all curves  $\{c_j(t)\}$  with  $j > 1$  are allowed to move, enabling dynamic partition adjustment. In contrast, the first control point  $p_0^1$  of the initial curve  $c_1(t)$  is kept on the plane of the build platform with  $\dot{c}_1(0) = (0, 0, 1)$  to ensure the manufacturability of the first layer. The co-optimization is visualized through the evolving curves and space partitions in Fig. 4.

### 4 Manufacturing Objectives

This section presents the formulation of the manufacturing objectives, which must be differentiable with respect to the optimization variables – namely, the control points  $\{p_k\}$  of a curve and the setup transformation  $\mathbf{Q}$  for adjusting an input model's orientation.

#### 4.1 States of Point in DLP Printing

The following formulas define the state of each point  $\mathbf{q} \in \mathcal{M}$ , including its position in WCS, its *local printing direction* (LPD), and its state about whether already printed.

For a point  $\mathbf{q} \in \mathcal{M}$  defined in MCS, its position in WCS at time  $t$  is given by  $\mathbf{T}(t)\mathbf{Q}\mathbf{q}$ , where  $\mathbf{T}(t)$  is the transformation matrix derived from the Bishop frame  $O_T$ , as described in Sec. 3.1.

To determine whether a point  $\mathbf{q}$  has been printed, we define a state function  $D(\mathbf{q}, t)$  that indicates whether  $\mathbf{q}$  contacts the bottom of the DLP printing tank at time  $t$ . The effective working area<sup>1</sup> at the bottom of resin tank is a rectangle with dimension  $[-R_X, R_X] \times [-R_Y, R_Y]$  (see Fig. 5 for an illustration). Incorporating the factor of effective region, the contact state function  $D(\mathbf{q}, t)$  is defined as follows:

$$D(\mathbf{q}, t) := \delta((\mathbf{q} - \mathbf{c}(t)) \cdot \dot{\mathbf{c}}(t)) R(\mathbf{q}) \quad (4)$$

where  $\delta(\cdot)$  denotes the Dirac delta function, and  $R(\mathbf{q})$  defines the effectiveness according to the working area as

$$R(\mathbf{q}) = H(R_X - |(\mathbf{q} - \mathbf{c}(t)) \cdot \mathbf{n}(t)|) H(R_Y - |(\mathbf{q} - \mathbf{c}(t)) \cdot \mathbf{b}(t)|).$$

$H(\cdot)$  is the Heaviside step function, and  $\mathbf{n}(t)$  and  $\mathbf{b}(t)$  are unit vectors from the Bishop frame. A value of  $D(\mathbf{q}, t) = 1$  indicates that  $\mathbf{q}$  is cured into solid by the effective area at the bottom of tank centered at  $\mathbf{c}(t)$ ; otherwise,  $D(\mathbf{q}, t) = 0$ . Based on this, we define the state function  $s(\mathbf{q}, t)$  to indicate whether  $\mathbf{q}$  has been printed:

$$s(\mathbf{q}, t) := \max_{\tau \in [0, t]} D(\mathbf{q}, \tau), \quad (5)$$

where  $s(\mathbf{q}, t) = 1$  means that the point has been printed at time  $t$ ; otherwise,  $s(\mathbf{q}, t) = 0$ .

We define a differentiable weight function  $w(\mathbf{q}, t) \approx s(\mathbf{q}, t)$  in MCS by applying the following approximations:

- Replace the Heaviside function  $H(x)$  with the sigmoid function  $\sigma_\beta(x) = 1/(1 + \exp(-\beta x))$ ;
- Approximate the Dirac delta function  $\delta(x)$  with the derivative of the sigmoid function  $\dot{\sigma}_\beta(x)$ ;
- Replace the max function  $\max_{\tau \in [0, t]} D(\mathbf{q}, \tau)$  with the log-sum-exp function  $\text{LSE}_\alpha(x)$ .

$\alpha, \beta \rightarrow \infty$  are scale coefficients controlling the sharpness of the respective approximations.

Many manufacturing objectives in multi-axis DLP 3D printing are defined based on the LPD at a model point  $\mathbf{q} \in \mathcal{M}$ . Given the trajectory curve  $\mathbf{c}(t)$  for DLP printing, the LPD at  $\mathbf{q}$  can be obtained by

$$\mathbf{d}(\mathbf{q}) := \int_0^1 \dot{\mathbf{c}}(t) D(\mathbf{q}, t) dt \approx \int_0^1 \dot{\mathbf{c}}(t) \dot{\sigma}_\beta((\mathbf{q} - \mathbf{c}(t)) \cdot \dot{\mathbf{c}}(t)) R_\delta(\mathbf{q}) dt \quad (6)$$

with

$$R_\delta(\mathbf{q}) := \sigma_\beta(R_X - |(\mathbf{q} - \mathbf{c}(t)) \cdot \mathbf{n}(t)|) \sigma_\beta(R_Y - |(\mathbf{q} - \mathbf{c}(t)) \cdot \mathbf{b}(t)|).$$

The same approximation of  $\delta(\cdot)$  and  $H(\cdot)$  as above is applied.

<sup>1</sup>The effective area is determined by the hardware design of a DLP 3D printer, including the power of UV-light projector, the focus length of lens etc.

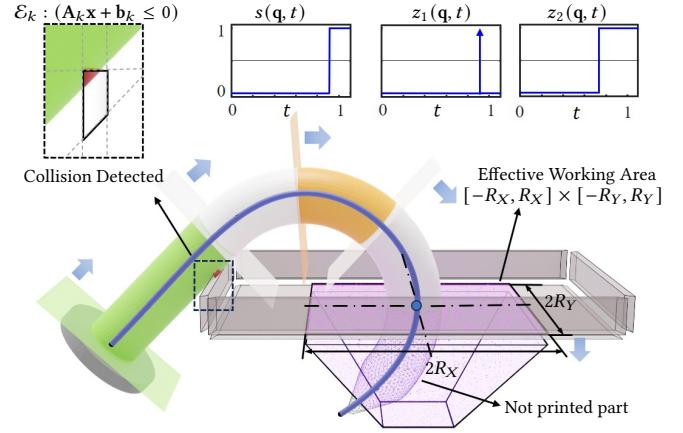


Fig. 5. Illustration of collision detection by representing the environmental obstacles as convex polytopes that can be defined by a set of half-spaces, where the region falling inside the convex polytope  $\mathcal{E}_k$  has been highlighted in red color. The status whether a point has been printed is indicated by a state function  $s(\mathbf{q}, t)$  (Eq.(5)) defined according to the values of  $z_1 = \delta((\mathbf{q} - \mathbf{c}(t)) \cdot \dot{\mathbf{c}}(t))$  and  $z_2 = R(\mathbf{q})$  in the range of  $[0, t]$  – see the example shown in the upper-right corner.

#### 4.2 Collision Avoidance

While optimizing the trajectories for 3D printing, it is essential to prevent collisions between the already printed part and any environmental obstacles. Note that the building plate  $\mathcal{B}$  (illustrated as the gray cylinder in Fig. 5) needs to be considered as an already printed part for collision check.

We formulate the environmental obstacles as a finite set of convex polytopes  $\mathcal{E} = \{\mathcal{E}_1, \dots, \mathcal{E}_N\}$ , and each convex polytope  $\mathcal{E}_k \subset \mathbb{R}^3$  is defined by a set of  $N_k$  half-spaces (see Fig. 5):

$$\mathcal{E}_k = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{A}_k \mathbf{x} + \mathbf{b}_k \leq 0\}, \quad (7)$$

where  $\mathbf{A}_k$  as a tensor in  $\mathbb{R}^{N_k \times 3}$  and  $\mathbf{b}_k$  as a vector in  $\mathbb{R}^{N_k}$  define the oriented boundary planes of this convex polytopes. Here,  $\mathbf{x}$  is defined in WCS. If  $\mathbf{x}$  lies strictly inside  $\mathcal{E}_k$  (i.e., a collision occurs), all components of the vector as  $[\mathbf{A}_k \mathbf{x} + \mathbf{b}_k]_i$  ( $\forall i = 1, \dots, N_k$ ) should be negative, resulting in

$$\max(\mathbf{A}_k \mathbf{x} + \mathbf{b}_k) < 0 \quad (8)$$

where  $\max(\cdot)$  denotes the maximum (non-absolute) component of a vector. Based on this observation, we use the Heaviside function  $H(\cdot)$  to indicate whether any already printed point is in collision with the environment while considering all convex polytopes:

$$\int_{\mathcal{M}} \int_0^1 s(\mathbf{Q}\mathbf{q}, t) \sum_k H(-\max(\mathbf{A}_k \mathbf{T}(t) \mathbf{Q}\mathbf{q} + \mathbf{b}_k)) dt d\mathbf{q}. \quad (9)$$

This integral returns zero when no collision occurs throughout the printing process.

After approximating the maximal function and the Heaviside function with the LSE function and the sigmoid function, we define the collision avoidance loss as

$$\mathcal{L}^{CO} := \sum_{\mathbf{q} \in \Omega} \int_0^1 w(\mathbf{Q}\mathbf{q}, t) \sum_k \sigma_\beta(-\text{LSE}_\alpha(\mathbf{A}_k \mathbf{T}(t) \mathbf{Q}\mathbf{q} + \mathbf{b}_k)) dt. \quad (10)$$

When computing the optimization for multiple curves, we can define  $\mathcal{L}_j^{CO}$  for  $\mathbf{q} \in \Omega_j$  in different regions by using both the transformation  $T(t)$  and the weight function  $w(\cdot, t)$  defined on their corresponding curve  $c_j(t)$ . Here  $\Omega_j$  denote the subset of  $\Omega$  as  $\Omega_j = \Omega \cap \mathcal{M}_j$ .

#### 4.3 Completeness

The completeness condition ensures that every point  $\mathbf{q} \in \mathcal{M}_j$  within a region  $\mathcal{M}_j$  has been effectively swept by the bottom of tank along the trajectory defined by  $c_j(t)$ . This condition is satisfied when  $s(\mathbf{q}, 1) = 1$  for all points at  $t = 1$ . Therefore, we define the completeness loss as

$$\mathcal{L}_j^{CP} := \sum_{\mathbf{q} \in \Omega_j} (1 - s(\mathbf{q}, 1)) \approx \sum_{\mathbf{q} \in \Omega_j} (1 - w(\mathbf{q}, 1)). \quad (11)$$

Again, the weight function  $w(\cdot, t)$  is defined on its corresponding curve  $c_j(t)$  in different regions.

#### 4.4 Support-free: Cliff-angle

One of the primary manufacturing constraints in AM is the requirement for support structures beneath regions with significant overhangs. The necessity for support can be evaluated by examining the angle between the surface normal and the local printing direction at each surface point. Specifically, for a surface point  $\mathbf{q} \in \partial\mathcal{M}$  with surface normal  $\mathbf{n}_q$ , the support-free condition is commonly defined as (ref. [Zhang et al. 2022]):

$$-\mathbf{n}_q \cdot \mathbf{d}(\mathbf{q}) / \|\mathbf{d}(\mathbf{q})\| \leq \sin \eta \quad (12)$$

where  $\eta$  is the material related coefficient.  $\mathbf{d}(\mathbf{q})$  is the LPD at  $\mathbf{q}$  (i.e., Eq.(6)), which is determined by the trajectory curve  $c(t)$ .

We define the support-free loss based on cliff-angle (CA) for multiple curves as

$$\mathcal{L}_j^{CA} := \sum_{\mathbf{q} \in \Psi_j} \rho_q \text{ReLU}\left(-\sin \eta - \mathbf{n}_q \cdot \frac{\mathbf{d}_j(\mathbf{q})}{\|\mathbf{d}_j(\mathbf{q})\|}\right) \quad (13)$$

with  $\Psi_j$  being the subset of surface sample points as  $\Psi_j = \Psi \cap \mathcal{M}_j$  and  $\mathbf{d}_j(\mathbf{q})$  denoting the LPD function as Eq.(6) for the  $j$ -th curve  $c_j(t)$ . The function  $\text{ReLU}(\cdot)$  is used to penalize violations where the expression inside is positive. A point sample density, denoted as  $\rho_q$ , is introduced to account for the non-uniformity of samples. It is defined as the average distance between the point  $\mathbf{q}$  and its  $k$ -nearest neighbors with  $k = 15$ .

#### 4.5 Support-free: Floating

A floating point (also known as point overhang [Vanek et al. 2014]) represents an isolated region that lacks supporting structure from below. This occurs when a sample point lies below all of its neighbors along the LPD, forming a local minimum in the geometry. Formally, for a surface sample point  $\mathbf{q} \in \partial\mathcal{M}$  with its  $k$ -nearest neighbors denoted by  $\mathcal{N}_q$ , it becomes a floating point if

$$(\mathbf{p} - \mathbf{q}) \cdot \mathbf{d}(\mathbf{q}) \geq 0 \quad (\forall \mathbf{p} \in \mathcal{N}_q). \quad (14)$$

Therefore, the objective of no floating can be reformulated as a requirement to  $\mathbf{d}(\mathbf{p})$  as

$$-\max_{\mathbf{p} \in \mathcal{N}_q} ((\mathbf{q} - \mathbf{p}) \cdot \mathbf{d}(\mathbf{p})) < 0. \quad (15)$$

By Eq.(6), we can then define the floating prevention loss as follows.

$$\mathcal{L}_j^{FL} := \sum_{\mathbf{q} \in \Psi_j} \sigma_\beta \left( -\text{LSE}_\alpha((\mathbf{q} - \mathbf{p}) \cdot \mathbf{d}_j(\mathbf{p})) \right) \quad (16)$$

where the  $\max(\cdot)$  function is approximated by the LSE function. Again, this is defined on a subset  $\Psi_i$  to support the optimization with multiple curves.

#### 4.6 Surface Quality

Surface quality must be considered to prevent staircase artifacts on the printed model's surface. As analyzed in [Etienne et al. 2019], maintaining the angle between the surface normal  $\mathbf{n}_q$  at a surface point  $\mathbf{q} \in \partial\mathcal{M}$  and its LPD  $\mathbf{d}(\mathbf{q})$  close to perpendicular is essential.

Using Eq.(6), a surface quality loss can be defined accordingly for each region as:

$$\mathcal{L}_j^{SQ} := \frac{A_s}{A_\Psi} \sum_{\mathbf{q} \in \Psi_j^s} \rho_q \left| \mathbf{n}_q \cdot \frac{\mathbf{d}_j(\mathbf{q})}{\|\mathbf{d}_j(\mathbf{q})\|} \right| \quad (17)$$

The surface quality loss is only applied to user-selected regions  $\Upsilon$  in practice as  $\Psi_j^s = \Psi_j \cap \Upsilon$ , and the loss is weighted by the ratio of the selected region's surface area  $A_s$  w.r.t. the input model's surface area  $A_\Psi$ .

#### 4.7 Total Loss

By incorporating all the loss terms defined above, we formulate the objective term in Eq. (2) as

$$\mathcal{L}_j := \omega_{SQ} \mathcal{L}_j^{SQ} + \omega_{CA} \mathcal{L}_j^{CA} \quad (18)$$

and the constraint term as

$$\Gamma_j := \mathcal{L}_j^{CO} + \mathcal{L}_j^{CP} + \mathcal{L}_j^{FL}. \quad (19)$$

Note that the two support-related losses are handled differently: the floating-prevention loss  $\mathcal{L}_j^{FL}$  is treated as a hard constraint, while the cliff-angle loss  $\mathcal{L}_j^{CA}$  for support-free is only included in the soft objective. This distinction arises from their practical impact on the printing process. Floating leads to complete print failure, making it a critical constraint. In contrast, as entire layers are solidified simultaneously, surfaces with large cliff angles primarily affect surface quality but do not necessarily cause failure in DLP printing. Nonetheless, such regions must still be carefully managed, particularly where the layer thickness is small.

### 5 Implementation Details

The geometry of the target models is sampled with sufficient density while balancing computational efficiency and accuracy. In our implementation, we set the average distance between sample points slightly smaller than the typical layer thickness (i.e., 0.2 mm) to ensure reliable results. We use Bézier curves with 6 control points in our implementation. Fewer control points offer insufficient degrees of freedom for optimization, while a larger number results in high-order Bézier curves that are prone to numerical oscillations. When subdivision is applied, each Bézier curve is split at  $t = 0.5$

using de Casteljau's algorithm, ensuring that the original curve is exactly represented by the two newly generated sub-curves.

### 5.1 Weighting and Optimization Scheme

Solving the trajectory optimization problem for multi-axis DLP 3D printing, as defined in Eq. (2), is particularly challenging due to the hard constraints of manufacturability defined as  $\Gamma_j = 0$  ( $\forall j = 1, \dots, m$ ). To balance these strict constraints with other optimization objectives, we adopt the following strategy:

- We formulate the total loss as a weighted sum:  $\sum_j \mathcal{L}_j + \omega_\Gamma \sum_j \Gamma_j$ , where  $\omega_\Gamma$  is a large penalty weight assigned to the constraint terms. Based on empirical tests taken across all examples in this paper, we set  $\omega_\Gamma = 100$ .
- We employ the Adam optimizer [Kingma and Ba 2015], a widely used stochastic-gradient based machine learning solver, to minimize the total loss.
- The optimization begins with a learning rate of 1.0e-3, which is halved every epoch with 1000 iterations. For each epoch, optimization resumes from the best result found in the preceding epoch, allowing for progressive refinement.

This hybrid approach – combining weighted loss formulation with adaptively reduced learning rate – gives effective optimization of multi-curve trajectories while maintaining strict adherence to manufacturability constraints in our experimental tests.

To enable differentiable optimization, we approximate Heaviside and Dirac delta functions using smoothed variants. The sharpness of these approximations is controlled by tunable parameters. In our implementation, we set  $\alpha = 1e+3$  and  $\beta = 1e+3$ . These values are chosen based on prior literature and validated through empirical tuning. The corresponding weights for different manufacturing objectives are set as  $\omega_{SQ} = 0.1$  and  $\omega_{CA} = 1.0$  to balance surface quality and support-free criteria. Finally, the material-specific parameter for cliff-angle evaluation is set to  $\eta = 0.698$  (corresponding to a 40° cliff-angle) across all experiments. This value is derived from our empirical measurements using physical DLP printing with photopolymer resin.

Our optimization algorithm proceeds iteratively. The iteration terminates when the total loss stops decreasing in the subsequent epoch and all hard constraints are satisfied.

### 5.2 Curve Initialization

The single-curve optimization is usually initialized using a vertical straight line, with the last control point of  $c(t)$  positioned slightly above the target model. This provides a more favorable starting condition for minimizing the completeness loss. For the  $\theta$ -components, we simply assign them as zero – e.g., no additional rotation for initialization.

When splitting a single curve into two (i.e., Step 5 of our algorithm presented in Sec. 3.3), the following heuristics are performed to initialize the curves in preparation for the next round of optimization:

- *Floating Heuristic:* To address a region with floating points that is covered by the curve  $c_j(t)$ , we first move the last control point of  $c_j(t)$  to the closest floating point,  $q_f$ . To improve the likelihood of eliminating the floating point, we

slightly adjust the control point such that the tangent vector  $\dot{c}_j(1)$  becomes nearly perpendicular to the average direction formed by  $q_f$  and its the  $k$ -nearest neighbors. Specifically, we require:

$$\dot{c}_j(1) \perp \frac{1}{|\mathcal{N}_{q_f}|} \sum_{p \in \mathcal{N}_{q_f}} (p - q_f),$$

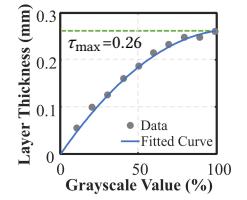
where  $\mathcal{N}_{q_f}$  denotes the  $k$ -nearest neighbors of  $q_f$  in  $\mathcal{M}$ , and  $|\cdot|$  represents the number of elements in the set.

- *Margin Heuristic:* For each curve  $c_j(t)$  ( $\forall j = 2, \dots, m$ ), we move the first control point slightly along the inverse tangent vector  $-\dot{c}_j(0)$  by a small distance, creating a marginal overlap between the regions covered by neighboring curves.

After applying these initialization heuristics, we observed faster convergence in the optimization process during our numerical experiments. An ablation study for this scheme of curve initialization will be given in Fig. 14.

### 5.3 Post-processing

The final layers of multi-axis DLP 3D printing are determined by sampling the trajectory curves  $c_j(t)$  densely and slicing the solid model  $\mathcal{M}_j$  using the tangent planes of neighboring sample points (e.g.,  $c_j(t_c)$  and  $c_j(t_c + \Delta t)$ ) with a very small step size  $\Delta t$ , such as 1.0e-3.



While the layer thickness during consolidation can be controlled by the grayscale values of the pixels in the projected images – see the curve shown in the insert figure for a constant exposure time as 5 seconds per layer, the working principle of DLP printing imposes a maximum manufacturable thickness,  $\tau_{\max}$ . Note that the curve is material-dependent, and can be obtained via calibration. Let  $\mathcal{P}_{\text{current}}$  and  $\mathcal{P}_{\text{next}}$  denote the planes formed by the tangent plane at  $c_j(t_c)$  and  $c_j(t_c + \Delta t)$  respectively. For every point  $q \in \mathcal{P}_{\text{current}} \cap \mathcal{M}_j$ , if the distance from  $q$  to the plane  $\mathcal{P}_{\text{next}}$  exceeds  $\tau_{\max}$ , we iteratively refine the sampling step  $\Delta t$  by halving it until the maximum layer thickness constraint is satisfied.

After determining the slicing result as solid regions on each slice, these regions are converted into images for projection. The motion of the working platform can be computed using inverse kinematics for robotic arms [Dai et al. 2020] or parallel mechanisms with tilting tables [Zhang et al. 2021], to align  $O_T$  with  $O_W$ , which is positioned at the center of the resin tank.

## 6 Results and Discussion

We have implemented the method introduced in this paper by Python and the source code of our pipeline will be released upon the acceptance of this paper. The effectiveness of our approach has been verified by both computational and physical experiments. Details are presented below.

### 6.1 Computational Experiments

All computational experiments are performed on a workstation with an Intel Xeon Sliver 4416+ CPU (20 cores, 3.9 GHz), 512 GB of system memory, and a single NVIDIA RTX 4090 graphics card, operating under Ubuntu 24.04. Gradient computations were performed using

Table 1. Statistics of computational and physical experiments

Model	Fig.	#Curves	Objective	Setup Ort.	# Floating Pnt.	# Pnt. $\Omega + \Psi$	Memory <sup>†</sup> CPU (MB)	Computing Time (sec.)			Fabrication Info.			
					Before	After		Optm. <sup>‡</sup>	Post-proc.	Total	#Layers	Weight	Time	
Toroidal-Tubes	1, 4	3	SF	Fixed	3	0	13,564	1,607	39.3	3.1	42.4	1,495	57g	2.9h
Hook*	7	1	SF	Fixed	0	0	9,268	1,478	4.1	4.2	8.3	1,967	52g	3.8h
Yoga	8	1	SF	Fixed	0	0	9,881	1,496	10.2	2.6	12.8	1,108	53g	2.1h
Woman	9	2	SF + SQ	Fixed	8	0	20,820	1,825	22.3	2.4	24.7	976	122g	1.8h
Bunny	10	2	SF + SQ	Fixed	0	0	33,060	2,192	45.6	3.1	48.7	1,148	173g	2.2h
Fertility	11	1	SF	Co-Optm.	6	0	13,848	1,615	10.8	2.4	13.2	961	84g	1.8h
Armadillo	19(a)	4	SF	Co-Optm.	49	1	41,804	2,432	62.1	-	-	-	-	-

\* Note that there is no floating point when using the medial axis as the initial curve of the hook model; but the computation also converges when using straight line as the initial curve.

<sup>†</sup> The memory used on GPU by the JAX implementation is pre-allocated with 75% of the graphics memory as 18,428MB on our hardware.

<sup>‡</sup> The optimization time for examples with multiple curves includes the total time of all stages in adaptive subdivision.

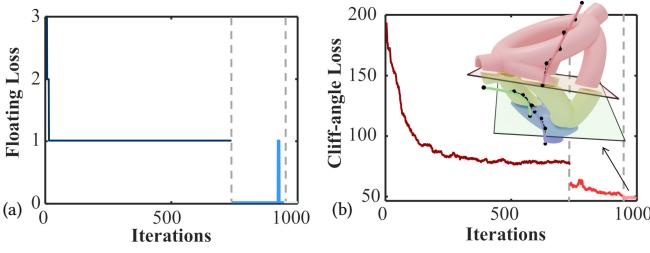


Fig. 6. Evolution of (a) the floating loss  $\mathcal{L}^{FL}$  and (b) the cliff-angle loss  $\mathcal{L}^{CA}$  during the optimization process for the Toroidal-Tubes example using 1 to 3 curves. The optimization phases with different numbers of curves are separated by dashed lines.

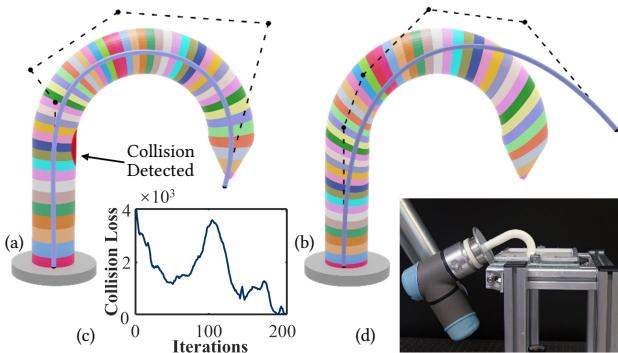


Fig. 7. The intuitive choice of support-free curve slicing for the Hook model is its medial axis, which however can lead to collision (a) – see also the collided configuration shown in Fig. 5. The issue can be well solved by our optimization with the collision-free curve for slicing and printing as shown in (b). We also provide (c) the convergence curve of the collision loss  $\mathcal{L}^{CO}$  and (d) a photo of the collision-free printing process.

the JAX framework, and optimization was carried out by the Adam solver in the Optax library.

**6.1.1 Examples and computing time.** The first example is the Toroidal-Tubes model, which features a complex topology with genus number 4, as shown in Figs. 1 and 4. In conventional planar layer-based DLP printing, the support structures added inside the tubes cannot be removed after fabrication. For this model, the final DLP layers are

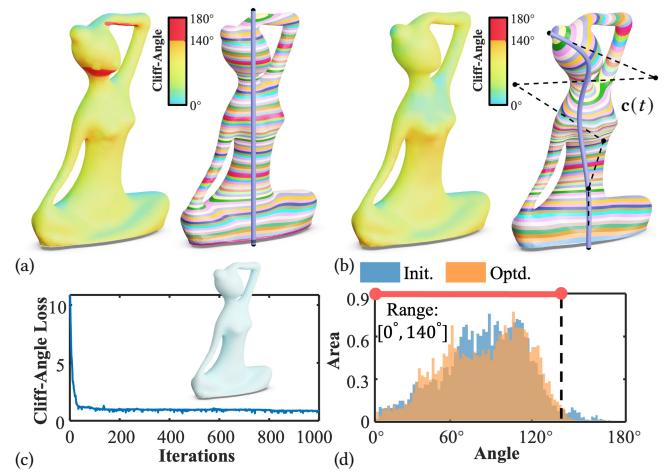


Fig. 8. Regions of large overhang in planar DLP printing (highlighted by red color) as shown in (a) can be effectively eliminated by using our curve-based slicer – see (b) the optimized curve  $c(t)$  and its corresponding control points & layers, (c) the convergence curve of the cliff-angle loss  $\mathcal{L}^{CA}$  and (d) the corresponding histogram of cliff-angles.

generated using three curves – see Fig. 4 for the curves as trajectories and the corresponding layers. The convergence behaviors of the floating point and the cliff-angle, evaluated with different numbers of trajectories, have been plotted in Fig. 6. It can be observed that the number of floating points, which is optimized based on the  $\mathcal{L}^{FL}$  loss, drops to zero at the end of optimization. The value of  $\mathcal{L}^{CA}$  has also been significantly reduced when using three curves. A fixed setup orientation is used throughout this example.

Our method can generate support-free layers using a single curve for a number of models – for example, the Hook model (Fig. 7) and the Yoga model (Fig. 8). In the Hook model example, an intuitive choice of the slicing curve is its medial axis, which however leads to collisions when printing toward the end of the hook (see Fig. 7(a)). However, this issue can be effectively resolved by our optimization-based method as shown in Fig. 7(b). In the Yoga model example, regions with large overhangs are eliminated after generating layers along a curved trajectory. In these three examples, we neglect the surface quality loss by setting  $\omega_{SQ} = 0.0$ .

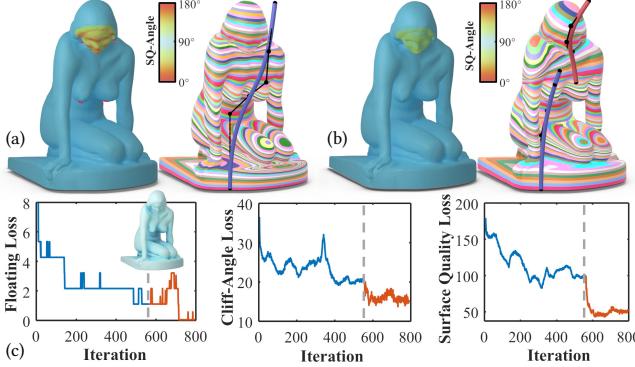


Fig. 9. Given an input woman model – the region for surface quality optimization is defined on the face, the results are generated by our method using (a) a single curve – the floating points exist and highlighted by red dots, and (b) two curves – no floating point anymore. (c) The convergence curves of computation. The angles between surface normals and the layer orientations have been visualized as color maps on the faces.

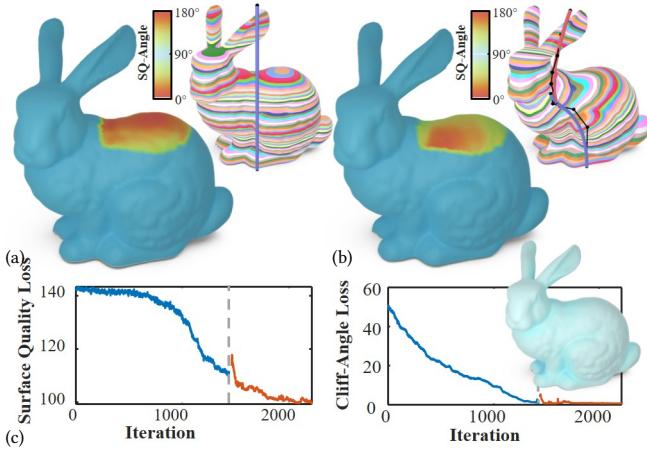


Fig. 10. The surface quality of the Bunny model (a) in the selected regions on the back can be effectively optimized – see (b) the new layers and the improved color maps for the distribution of angles between surface normals and the layer orientations. (c) The convergence curves of  $\mathcal{L}^{SQ}$  (left) and  $\mathcal{L}^{CA}$  (right) are also given.

In the fourth example of a Woman statue model, we incorporate the surface quality objective defined on a user-selected face region (see Fig. 9). A manufacturable solution is achieved by subdividing the initial curve into two curves, since a single curve alone cannot fully eliminate floating points. The angles between surface normals and the layers' normals have been visualized as color maps on the faces. It can be observed that our method can effectively shift the distribution toward orthogonality. Surface quality optimization is also demonstrated on the Bunny model shown in Fig. 10. Again two curves are employed in this example to realize support-free printing while preserving the surface quality at the back of the Bunny model.

The final example is the Fertility model, which poses significant challenges for conventional PPL-based DLP printing. By allowing

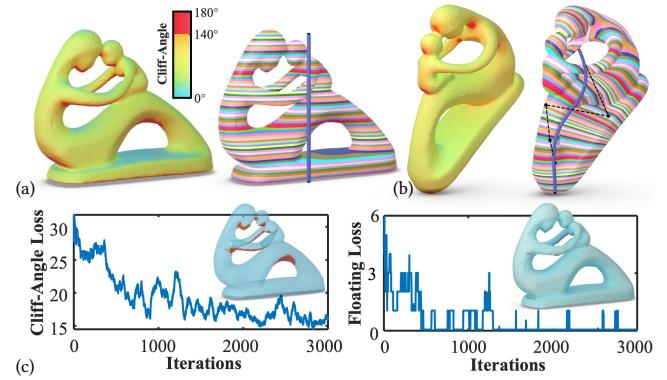


Fig. 11. There are many challenging regions on the Fertility model with large overhangs – the regions with large overhangs are highlighted by red color in (a). Our method can optimize the slicing curve and the setup orientation together to reduce most of these regions and eliminate all floating points – see the resultant curve and its corresponding layers shown in (b). The convergence curves of the cliff-angle loss and the floating loss are given in (c).

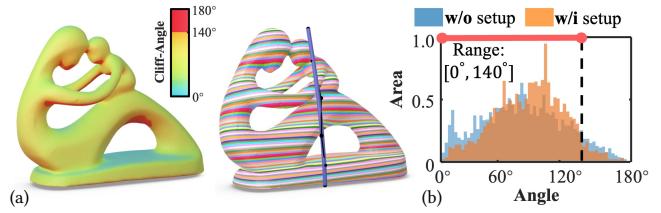


Fig. 12. Disabling the co-optimization of setup orientation results in large regions with significant overhang, highlighted in red in (a). This is further visualized by (b) the histograms comparing the angles between surface normals and local printing directions. This also leads to floating points.

the setup orientation to be co-optimized, we are able to generate layers for multi-axis DLP printing using a single curve (see Fig. 11).

The computational statistics for all examples are summarized in Table 1. Overall, our approach demonstrates efficient performance – all models are completed within one minute. It can be observed that the step of curve optimization takes the majority of computing time while the post-processing step can be completed within few seconds in general.

**6.1.2 Ablation study.** Co-optimizing the setup orientation  $\mathbf{Q}$  along with the curve trajectory for multi-axis DLP printing generally provides the optimizer with a larger solution space, increasing the likelihood of finding a feasible result. For the Fertility model, an ablation study has been taken by turning on / off the co-optimization of setup orientation. From the result shown in Fig. 12, we find that it is difficult to effectively reduce the area of overhangs without the co-optimization.

We further conducted an ablation study to demonstrate the importance of both floating-free losses,  $\mathcal{L}^{FL}$  and  $\mathcal{L}^{CA}$ . On the Woman model, a substantial number of floating points are generated when the  $\mathcal{L}^{FL}$  loss is removed (see Fig. 13(a)). Conversely, when using

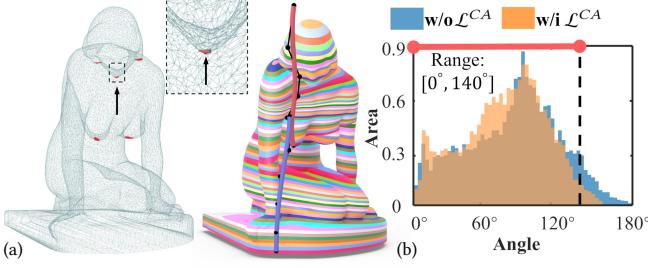


Fig. 13. Ablation study of two different support-free losses on the Woman model: (a) the result by removing the floating loss  $\mathcal{L}^{FL}$  but keeping  $\mathcal{L}^{CA}$ , and (b) the histograms of cliff-angles with vs. without using the  $\mathcal{L}^{CA}$  loss while keeping the floating loss  $\mathcal{L}^{FL}$ .

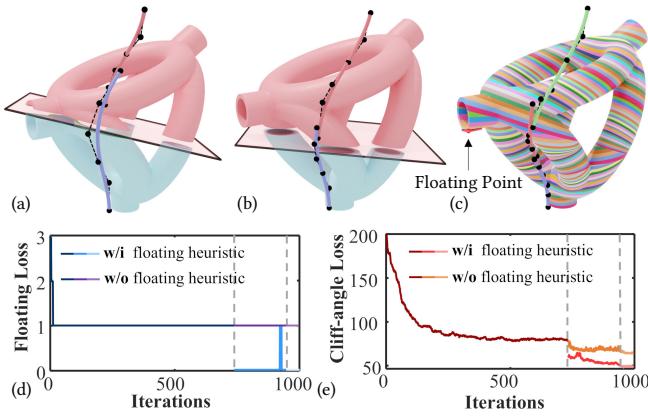


Fig. 14. Ablation study of the floating heuristic employed in the curve initialization scheme: (a) the two curves after initialization without applying the floating heuristic, (b) the optimized curves and the corresponding partitions, and (c) the result of optimization taken on three curves (i.e., floating point cannot be eliminated without the floating heuristic). The computational convergence curves of (d) the floating loss  $\mathcal{L}^{FL}$  and (e) the cliff-angle loss  $\mathcal{L}^{CA}$  further demonstrate the effectiveness of the floating heuristic.

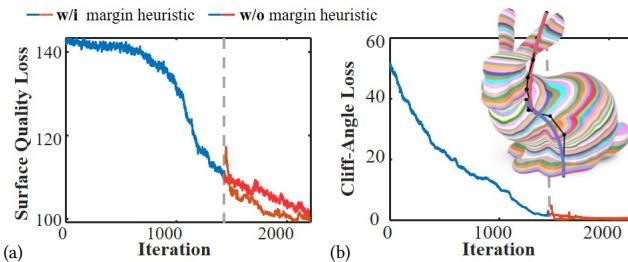


Fig. 15. Ablation study of the margin heuristic in the curve initialization scheme, demonstrating its effectiveness through the convergence behavior of (a) the surface quality loss  $\mathcal{L}^{SQ}$  and (b) the cliff-angle loss  $\mathcal{L}^{CA}$  on the Bunny model.

only  $\mathcal{L}^{FL}$  without considering the cliff-angle loss  $\mathcal{L}^{CA}$ , more regions with large cliff-angles are generated (see the histograms given

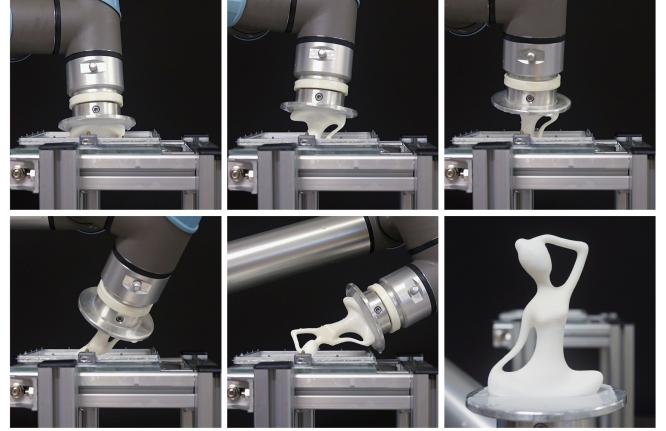


Fig. 16. The progressive results of printing the Yoga model by the robot-assisted multi-axis DLP hardware.

in Fig. 13(b)), highlighting the complementary roles of these two objectives.

The other two ablation studies have been taken to verify the effectiveness of our curve initialization scheme. As shown in Fig. 14, when the floating heuristic described in Sec. 5.2 is not applied, the computation becomes more difficult to converge. Similarly, omitting the margin heuristic leads to slower convergence during optimization as shown in Fig. 15.

## 6.2 Physical Experiments

The slicing results generated by our method were validated on a lab-built multi-axis DLP 3D printing setup, comprising a resin tank, a UV-light projector, and a UR5e robotic arm. The robotic arm is controlled via ROS2 and achieves a motion repeatability of  $\pm 0.03\text{mm}$ . The Sumaopai SMPDE75T4B projector emits UV light at a wavelength of 450nm, covering an exposure area of  $144\text{mm} \times 81\text{mm}$  as the effective area at the bottom of the resin tank (see Fig. 5). The curing process is digitally controlled using grayscale images at a resolution of  $1920 \times 1080$ . For physical fabrication, we used Anycubic Standard Resin V2 as the printing material in our experiments.

All models discussed above were successfully fabricated using our multi-axis DLP 3D printing system. As illustrated in Figs. 1 and 16, complex geometries can be effectively produced with our method. In addition to the computational ablation study presented in Sec. 6.1.2, we conducted a physical ablation study on the Bunny model to evaluate the impact of the surface quality loss term  $\mathcal{L}^{SQ}$ . The results are given in Fig. 17, which demonstrate a significant reduction in staircase artifacts on the back of the Bunny model when  $\mathcal{L}^{SQ}$  is applied. Final fabrication results for other models are shown in Fig. 18, and detailed fabrication statistics are provided in Table 1. In summary, all models were fabricated within 4 hours, with a total number of layers reaching up to 2.0k, demonstrating the efficiency of multi-axis DLP 3D printing.

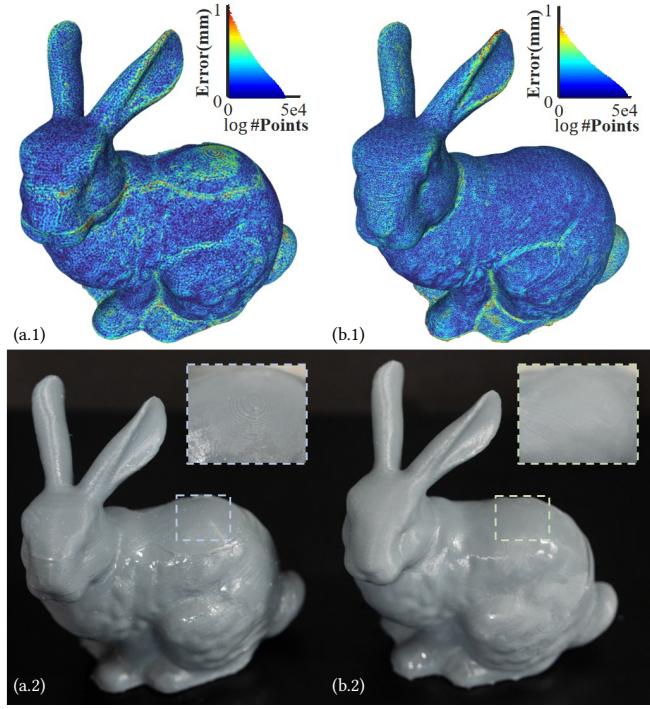


Fig. 17. By turning off the surface quality loss  $\mathcal{L}^{SQ}$ , the back of the Bunny model generate significant staircase artifacts (a.1 & a.2) comparing to the model fabricated from our optimization using the full set of losses (b.1 & b.2). The geometric errors on the scanned models are visualized as the color maps and the corresponding histograms in the top row (a.1 & b.1).



Fig. 18. The group photo of all models fabricated by our multi-axis DLP 3D printing method.

### 6.3 Discussion

While our curve-based slicing algorithm successfully generates motion trajectories for multi-axis DLP 3D printing – effectively overcoming manufacturing challenges such as collision, floating points, large overhangs, and staircase artifacts – there remain several limitations to the current method.

First, the manufacturability of the optimized curve trajectories can still be hindered by environmental obstacles, most notably the resin tank. For example, in the Armadillo model shown in Fig. 19(a), the support-free trajectory for printing the tail results in a collision between the tank and the already-printed part, rendering the

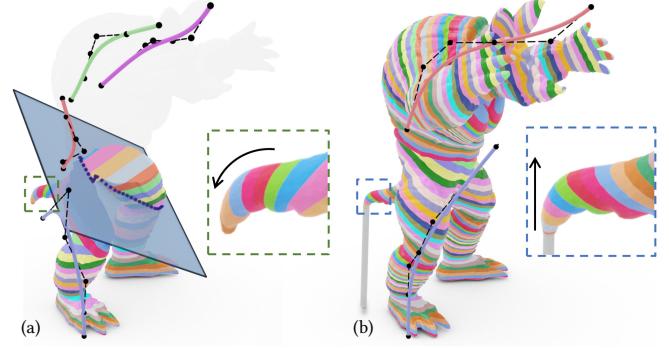


Fig. 19. A failure case caused by the collision between the tank and the already printed part when attempting to generate curves for support-free printing the tail of the Armadillo model (a). However, the model can become printable by simply adding a support to the tail (b).

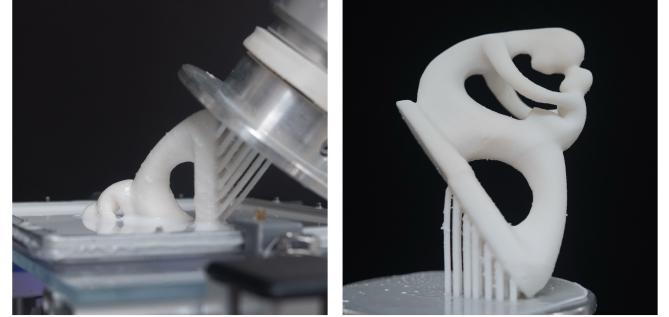


Fig. 20. Additional supports are manually added to the base of the Fertility model to improve the structural strength of the initial layers.

fabrication infeasible. However, this issue can be mitigated by introducing a single support under the tail (see Fig. 19(b)), which enables successful printing while maintaining most of the benefits of support-free fabrication in other regions.

Second, our current collision-avoidance loss function (described in Sec. 4.2) only accounts for potential collisions between the printed structure (including the building plate) and environmental obstacles (e.g., the resin tank). It does not consider collisions involving the robotic arm and its surrounding environment. At present, we rely on the IK solver to discard unrealizable robot poses that would result in collisions. However, this approach does not guarantee the existence of feasible collision-free arm motions for all optimized trajectories.

Lastly, our method does not currently account for the mechanical stability of partially printed structures during the printing process (ref. [Zhou et al. 2013]). For instance, additional supports were manually added to the base of the Fertility model to improve the structural integrity of the initial layers (see Fig. 20). The support structure is generated in two steps:

- (1) A convex region is defined by the building plate's plane and the first slicing plane at  $c(0)$ ;

- (2) Within this convex region, support material is added along the ‘vertical’ direction, which corresponds to the normal vector of the build plate.

This method of support generation ensures collision-free DLP 3D printing during the initial stages of fabrication. Without these reinforcements, the connection between the early printed parts and the building plate would be too weak, resulting in unstable or failed prints. Furthermore, mechanical and thermal deformations occurring during the printing process may introduce misalignments between partitioned regions of the model. All of these limitations will be considered in our future work.

## 7 Conclusion

In this paper, we presented a novel curve-based slicer for jointly generating, 1) planar layers with dynamically varying orientations and 2) model partitions, to enable the fabrication of complex geometries using multi-axis DLP 3D printing. By introducing additional DoF in motion, our approach addresses several key limitations of conventional PPL-based DLP printing, including the need for support structures to handle floating points and large overhangs, as well as the presence of staircase artifacts on printed surfaces. Specifically, we define slicing layers using the tangent planes of one or more parametric curves, which also determine the model partitioning and the motion trajectories of the build platform through associated pose frames. Manufacturing objectives such as surface quality and overhang minimization, along with constraints like collision avoidance and floating-free deposition, are formulated as loss functions and jointly optimized through the control points of these curves. Additionally, the optimization of model’s setup orientation can be integrated into the optimization framework. As a result, our method significantly enhances the capability of DLP 3D printing to handle complex geometries while preserving its intrinsic advantages of high resolution and fast printing speed. The effectiveness of our approach has been demonstrated through both numerical simulations and physical experiments.

## Acknowledgments

The authors gratefully acknowledge Professor A. John Hart and Nicholas S. Diaco of the Massachusetts Institute of Technology for their valuable comments on DLP printing. This research was supported by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government, the Chair Professorship Fund at the University of Manchester and UK Engineering and Physical Sciences Research Council (EPSRC) Fellowship Grant (Ref.#: EP/X032213/1).

## References

- Lingxin Cao, Lihao Tian, Hao Peng, Yu Zhou, and Lin Lu. 2021. Constrained stacking in DLP 3D printing. *Computers & Graphics* 95 (2021), 60–68.
- Lingxin Cao, Xin Yan, Longdu Liu, Jikai Liu, Tengfei Jiang, and Lin Lu. 2025. Auto-support: traction force constrained support structures optimization for bottom-up vat photopolymerization. *Virtual and Physical Prototyping* 20, 1 (2025), e2466189.
- Rajat Chaudhary, Paride Fabbri, Enrico Leoni, Francesca Mazzanti, Raziye Akbari, and Carlo Antonini. 2023. Additive manufacturing by digital light processing: a review. *Progress in Additive Manufacturing* 8, 2 (2023), 331–351.
- Yongxue Chen, Tianyu Zhang, Yuming Huang, Tao Liu, and Charlie C.L. Wang. 2025. Optimization of Tool Orientations, Kinematic Redundancy, and Waypoint Timing for Robot-Assisted Manufacturing. *IEEE Transactions on Automation Science and Engineering* 22 (2025), 12102–12117.
- Chengkai Dai, Sylvain Lefebvre, Kai-Ming Yu, Jo M. P. Geraedts, and Charlie C.L. Wang. 2020. Planning Jerk-Optimized Trajectory with Discrete Time Constraints for Redundant Robots. *IEEE Transactions on Automation Science and Engineering* 17, 4 (2020), 1711–1724.
- Chengkai Dai, Charlie C.L. Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, and Yong-Jin Liu. 2018. Support-free volume printing by multi-axis motion. *ACM Transactions on Graphics* 37, 4 (2018), 1–14.
- Simon Duenser, Roi Poranne, Bernhard Thomaszewski, and Stelian Coros. 2020. Robo-Cut: Hot-wire Cutting with Robot-controlled Flexible Rods. *ACM Transactions on Graphics* 39, 4 (2020), 1–15.
- Jimmy Etienne, Nicolas Ray, Daniele Panozzo, Samuel Hornus, Charlie C.L. Wang, Jonas Martínez, Sara McMains, Marc Alexa, Brian Wyvill, and Sylvain Lefebvre. 2019. Curvislicer: Slightly curved slicing for 3-axis printers. *ACM Transactions on Graphics* 38, 4 (2019), 1–11.
- Guoxin Fang, Tianyu Zhang, Sikai Zhong, Xiangjia Chen, Zichun Zhong, and Charlie C.L. Wang. 2020. Reinforced FDM: Multi-axis filament alignment with controlled anisotropic strength. *ACM Transactions on Graphics* 39, 6 (2020), 1–15.
- Kailun Hu, Shuo Jin, and Charlie C.L. Wang. 2015. Support slimming for single material based additive manufacturing. *Computer-Aided Design* 65 (2015), 1–10.
- Jigang Huang, Qin Qin, Cheng Wen, Zhuoxi Chen, Kunlan Huang, Xia Fang, and Jie Wang. 2022. A dynamic slicing algorithm for conformal additive manufacturing. *Additive Manufacturing* 51 (2022), 102622.
- Jigang Huang, Henry Oliver T. Ware, Rihan Hai, Guangbin Shao, and Cheng Sun. 2021. Conformal Geometry and Multimaterial Additive Manufacturing through Freeform Transformation of Building Layers. *Advanced Materials* 33, 11 (2021), 2005672.
- Pu Huang, Charlie C. L. Wang, and Yong Chen. 2014. Algorithms for Layered Manufacturing in Image Space. In *Advances in Computers and Information in Engineering Research, Volume 1*.
- Yijiang Huang, Juyong Zhang, Xin Hu, Guoxian Song, Zhongyuan Liu, Lei Yu, and Ligang Liu. 2016. FrameFab: robotic fabrication of frame shapes. *ACM Transactions on Graphics* 35, 6 (2016), 1–11.
- Young J. Kim, Gokul Varadhan, Ming C. Lin, and Dinesh Manocha. 2004. Fast swept volume approximation of complex polyhedral models. *Computer-Aided Design* 36, 11, 1013–1027.
- Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015).
- Xiao Kuang, Jiangtao Wu, Kaijuan Chen, Zeang Zhao, Zhen Ding, Fengjingyang Hu, Daining Fang, and H. Jerry Qi. 2019. Grayscale digital light processing 3D printing for highly functionally graded materials. *Science Advances* 5, 5 (2019), 5790.
- Shengjun Liu, Tao Liu, Qiang Zou, Weiming Wang, Eugeni L. Doubrovski, and Charlie C. L. Wang. 2021. Memory-Efficient Modeling and Slicing of Large-Scale Adaptive Lattice Structures. *Journal of Computing and Information Science in Engineering* 21, 6 (05 2021), 061003.
- Tao Liu, Tianyu Zhang, Yongxue Chen, Yuming Huang, and Charlie C.L. Wang. 2024. Neural Slicer for Multi-Axis 3D Printing. *ACM Transactions on Graphics* 43, 4 (2024), 1–15.
- Tao Liu, Tianyu Zhang, Yongxue Chen, Weiming Wang, Yu Jiang, Yuming Huang, and Charlie C.L. Wang. 2025. Neural co-optimization of structural topology, manufacturable layers, and path orientations for fiber-reinforced composites. *ACM Transactions on Graphics* 44, 4 (2025), 1–17.
- Andrea Luongo, Viggo Falster, Mads Brix Does, Macarena Mendez Ribo, Eypó Ruinar Eiríksson, David B Pedersen, and Jeppe Revall Frisvad. 2020. Microstructure Control in 3D Printing with Digital Light Processing. *Computer Graphics Forum* 39, 1 (2020), 347–359.
- Zhao Ma, Simon Duenser, Christian Schumacher, Romana Rust, Moritz Bächer, Fabio Gramazio, Matthias Kohler, and Stelian Coros. 2020. RobotSculptor: Artist-Directed Robotic Sculpting of Clay. *Proceedings - SCF 2020: ACM Symposium on Computational Fabrication*, 1–12.
- Huachao Mao, Tsz-Ho Kwok, Yong Chen, and Charlie C.L. Wang. 2019. Adaptive slicing based on efficient profile analysis. *Computer-Aided Design* 107 (2019), 89–101.
- Tobia Marcucci, Mark Petersen, David von Wrangel, and Russ Tedrake. 2023. Motion planning around obstacles with convex optimization. *Science Robotics* 8, 84 (2023), eadf7843.
- Ioanna Mitropoulou, Mathias Bernhard, and Benjamin Dillenburger. 2020. Print Paths Key-framing: Design for non-planar layered robotic FDM printing. *Proceedings - SCF 2020: ACM Symposium on Computational Fabrication*, 1–10.
- Jisoo Nam and Miso Kim. 2024. Advances in materials and technologies for digital light processing 3D printing. *Nano Convergence* 11, 1 (2024), 45.
- Jia Pan, Liangjiao Zhang, and Dinesh Manocha. 2012. Collision-free and smooth trajectory computation in cluttered environments. *International Journal of Robotics Research* 31, 10 (2012), 1155–1175.
- Xirui Peng, Liang Yue, Sizhuang Liang, Stuart Montgomery, Chunliang Lu, Chieh-Min Cheng, Raheem Beyah, Ruike Renee Zhao, and H. Jerry Qi. 2022. Multi-Color 3D

- Printing via Single-Vat Grayscale Digital Light Processing. *Advanced Functional Materials* 32, 28 (2022), 2112329.
- Kam-Ming Mark Tam and Caitlin T. Mueller. 2017. Additive Manufacturing Along Principal Stress Lines. *3D Printing and Additive Manufacturing* 4, 2 (2017), 63–81.
- J. Vanek, J.A.G. Galicia, and B. Benes. 2014. Clever support: Efficient support structure generation for digital fabrication. *Eurographics Symposium on Geometry Processing* 33, 5 (2014), 117–125.
- A. von Dziegielewski, R. Erbes, and E. Schömer. 2010. Conservative swept volume boundary approximation. *Proceedings - 14th ACM Symposium on Solid and Physical Modeling, SPM'10*, 171–176.
- Jingping Wang, Tingrui Zhang, Qixuan Zhang, Chuxiao Zeng, Jingyi Yu, Chao Xu, Lan Xu, and Fei Gao. 2024. Implicit Swept Volume SDF: Enabling Continuous Collision-Free Trajectory Generation for Arbitrary Shapes. *ACM Transactions on Graphics* 43, 4 (2024), 1–14.
- Weiming Wang, Haiyuan Chao, Jing Tong, Zhouwang Yang, Xin Tong, Hang Li, Xiuping Liu, and Ligang Liu. 2015. Saliency-Preserving Slicing Optimization for Effective 3D Printing. *Computer Graphics Forum* 34, 6 (2015), 148–160.
- Wenping Wang, Bert Jüttler, Dayue Zheng, and Yang Liu. 2008. Computation of rotation minimizing frames. *ACM Transactions on Graphics* 27, 1 (2008), 1–18.
- Weiming Wang, Cédric Zanni, and Leif Kobbelt. 2016. Improved surface quality in 3D printing by optimizing the printing direction. *Computer Graphics Forum* 35, 2 (2016), 59–70.
- Chenming Wu, Chengkai Dai, Guoxin Fang, Yong-Jin Liu, and Charlie C.L. Wang. 2017. RoboFDM: A robotic system for support-free fabrication using FDM. *Proceedings of IEEE ICRA*, 1175–1180.
- Chenming Wu, Chengkai Dai, Guoxin Fang, Yong-Jin Liu, and Charlie C. L. Wang. 2020. General Support-Effective Decomposition for Multi-Directional 3-D Printing. *IEEE Transactions on Automation Science and Engineering* 17, 2 (2020), 599–610.
- Rundong Wu, Huaishu Peng, François Guimbretière, and Steve Marschner. 2016. Printing arbitrary meshes with a 5DOF wireframe printer. *ACM Transactions on Graphics* 35, 4 (2016), 1–9.
- Ran Yi, Chenming Wu, Yong-Jin Liu, Ying He, and Charlie C.L. Wang. 2018. Delta DLP 3-D Printing of Large Models. *IEEE Transactions on Automation Science and Engineering* 15, 3 (2018), 1193–1204.
- Ran Yi, Chenming Wu, Yong-Jin Liu, and Charlie C.L. Wang. 2016. RoboDLP: Large Object 3D Printing using Robotic Arm. *SA 2016 - SIGGRAPH ASIA 2016 Posters*, 1–2.
- Liang Yue, S. Macrae Montgomery, Xiaohao Sun, Luxia Yu, Yuyang Song, Tsuyoshi Nomura, Masato Tanaka, and H. Jerry Qi. 2023. Single-vat single-cure grayscale digital light processing 3D printing of materials with large property difference and high stretchability. *Nature Communications* 14, 1 (2023), 1251.
- Tianyu Zhang, Xiangjia Chen, Guoxin Fang, Yingjun Tian, and Charlie C.L. Wang. 2021. Singularity-Aware Motion Planning for Multi-Axis Additive Manufacturing. *IEEE Robotics and Automation Letters* 6, 4 (2021), 6172–6179.
- Tianyu Zhang, Guoxin Fang, Yuming Huang, Neelotpal Dutta, Sylvain Lefebvre, Zekai Murat Kilic, and Charlie C.L. Wang. 2022. S3-Slicer: A General Slicing Framework for Multi-Axis 3D Printing. *ACM Transactions on Graphics* 41, 6 (2022), 1–15.
- Xiaoting Zhang, Xinyi Le, Athina Panotopoulou, Emily Whiting, and Charlie C.L. Wang. 2015. Perceptual models of preference in 3D printing direction. *ACM Transactions on Graphics* 34, 6 (2015), 1–12.
- Xuejin Zhao, Yayue Pan, Chi Zhou, Yong Chen, and Charlie C.L. Wang. 2013. An integrated CNC accumulation system for automatic building-around-inserts. *Journal of Manufacturing Processes* 15, 4 (2013), 432–443.
- Fanchao Zhong, Yonglai Xu, Haisen Zhao, and Lin Lu. 2023a. As-Continuous-As-Possible Extrusion-Based Fabrication of Surface Models. *ACM Transactions on Graphics* 42, 3 (2023), 1–16.
- Fanchao Zhong, Haisen Zhao, Haochen Li, Xin Yan, Jikai Liu, Baoquan Chen, and Lin Lu. 2023b. VASCO: Volume and Surface Co-Decomposition for Hybrid Manufacturing. *ACM Transactions on Graphics* 42, 6 (2023), 1–17.
- Qingnan Zhou, Julian Panetta, and Denis Zorin. 2013. Worst-case structural analysis. *ACM Transactions on Graphics* 32, 4 (2013), 1–12.