More Stiffness with Less Fiber: End-to-End Fiber Path Optimization for 3D-Printed Composites

ABSTRACT
In 3D printing, stiff fibers (e.g., carbon fiber) can reinforce thermoplastic polymers with limited stiffness. However, existing commercial digital manufacturing software only provides a few simple fiber layout algorithms, which solely use the geometry of the shape. In this work, we build an automated fiber path planning algorithm that maximizes the stiffness of a 3D print given specified external loads. We formalize this as an optimization problem: an objective function is designed to measure the stiffness of the object while regularizing certain properties of fiber paths (e.g., smoothness). To initialize each fiber path, we use finite element analysis to calculate the stress field on the object and greedily "walk" in the direction of the stress field. We then apply a gradient-based optimization algorithm that uses the adjoint method to calculate the gradient of stiffness with respect to fiber layout. We compare our approach, in both simulation and real-world experiments, to three baselines: (1) concentric fiber rings generated by Eiger, a leading digital manufacturing software package developed by Markforged, (2) greedy extraction on the simulated stress field (i.e., our method without optimization), and (3) the greedy algorithm on a fiber orientation field calculated by smoothing the simulated stress fields. The results show that objects with fiber paths generated by our algorithm achieve greater stiffness while using less fiber than the baselines—our algorithm improves the Pareto frontier of object stiffness as a function of fiber usage. Ablation studies show that the smoothing regularizer is needed for feasible fiber paths and stability of optimization, and multi-resolution optimization helps reduce the running time compared to single-resolution optimization.

1 INTRODUCTION
Additive manufacturing has revolutionized the ability to fabricate three-dimensional objects of high geometric complexity, with a variety of applications including in healthcare, automotive, and aerospace industries [Shahrubudin et al. 2019]. However, the increasing flexibility in manufacturing has outstripped our ability to produce designs that optimally take advantage of 3D printers. This has motivated research on computational fabrication pipelines that augment human specification of goals with computational optimization of designs that best realize those goals, for problems ranging from ensuring structural integrity through controlling appearance and fine-tuning the fabrication process [Attene et al. 2018].

In this work, we address the problem of producing structurally-sound parts that are capable of bearing nontrivial load. We aim to exploit the capabilities of devices such as the Markforged Mark Two [Markforged 2023a], which is based on conventional fused deposition modeling (FDM) using thermoplastic nylon, but augments this with the ability to extrude and deposit continuous fibers. Options for the latter include carbon fiber, Kevlar, fiberglass, and HSHT (High Strength High Temperature) fiberglass, all of which offer the ability to selectively strengthen printed parts with respect to tensile loads. In effect, this process creates fiber-reinforced plastic (FRP) composites [Kabir et al. 2020], but with the ability to control fiber placement to achieve specific tradeoffs in strength, weight, and cost.

The optimization of fiber layout is similar to problems traditionally considered in computational fabrication, such as topology optimization (i.e., removing material from certain regions) and spatially-varying assignment of different materials. Systems for
We incorporate our gradient descent-based optimization into a complete system that addresses three key challenges: (1) solving for the stress field of the object given external loads, (2) computing an optimization objective and its gradient based on the stress field, and (3) optimizing the path directly using gradient descent. This strategy is based on the adjoint method [Cao et al. 2003; Errico 1997], commonly used for PDE-constrained optimization, and exploits modern systems for automatic differentiation [Griewank and Walther 2008], which have evolved considerably in recent years to support a range of machine learning and general optimization problems. Our end-to-end optimization approach has the benefit of focusing directly on the final goal—maximizing stiffness with respect to external loads—rather than on indirect objectives such as minimizing strain throughout the object.

We show designs produced by our method on a number of illustrative case studies, demonstrating that our method yields higher stiffness with less fiber as compared to baseline paths produced by the Eiger software by Markforged [2023b]. We compare our results to greedy extraction based on either the stress field or optimized fiber direction field, as well as other ablations including omitting regularization or multi-resolution optimization. We print our designs (see Figure 1), using the method of Sun et al. [2021] to compute fiber extruder paths that compensate for fiber stiffness. Finally, we test our printed parts to verify that our method matches the predicted stiffness in the real world (subject to inherent print-to-print variations in material strength).

2 RELATED WORK

2.1 Fiber orientation optimization in 3D printing

A task that is similar to fiber path planning is fiber orientation optimization, where researchers discretize space into elements and optimize fiber orientations in them. Additional steps, such as greedy extraction, ODE solvers, or geometric methods, must be performed to produce fiber paths from the orientation field. Thus fiber orientation optimization can serve as the first step of fiber path planning, which we will discuss in §2.2. See Hu [2021] for a survey (called “free material optimization”). The most common approach for orientation optimization is to set density and orientation as design variables and optimize an objective such as compliance [Chu et al. 2021; da Silva et al. 2020; Jung et al. 2022] or the Tsai–Wu failure criterion [Ma et al. 2022] with a gradient-based optimizer. To address the checkerboard pattern issue (periodicity of the orientation variables), researchers usually use filtering [Andreasen et al. 2011] to smooth the design variable field (e.g., through a weighted average of neighboring elements). Another choice of design variable is the lamination parameters: Shafighfar et al. [2019] and Demir et al. [2019] proposed to first optimize lamination parameters, search for the best fitting fiber orientations from the optimized lamination parameters, and then perform an optimization on the orientation field while considering manufacturing constraints (e.g.,

Figure 1: Planned and 3D printed fiber paths with fiber lengths and average stiffness measured over four batches annotated, for a part with external tension applied between two holes. (a) (b): Concentric fiber rings generated by the Eiger baseline only consider geometry. (c) (d): Our optimized fiber paths, tuned for the applied loads, yield greater stiffness at lower fiber lengths.
A variety of path planning algorithms have been proposed for convergence. While mainly concentrating on orientation optimization, some approaches do ultimately generate fiber paths. For example, Fedulov et al. [2021] first optimized density and orientation and then used third-party software for printing trajectory generation; Schmidt et al. [2020] performed density and orientation optimization and generated streamlines using the 4th-order Runge-Kutta integrator for visualization.

2.2 Fiber path planning in 3D printing
A variety of path planning algorithms have been proposed for continuous fiber-reinforced plastics—see Zhang et al. [2020] for a survey. The most common approach is to first perform an optimization (topology, orientation, etc.), and then extract fiber paths from the result. As discussed, orientation optimization is one choice of the optimization (i.e., use density and orientation as the design variables), but there are different methods for path extraction. Wang et al. [2021] proposed to “walk” in the field along with the stress direction and consider the angle turned in every move to produce smoothed paths. Papapetrou et al. [2020] described three methods for path extraction: the offset method and the EQS (Equally-Space) method use the geometry of the optimized layout, and the streamline method fits the orientation field with streamlines. Safronov [2019] proposed to alternate between topology optimization and fiber orientation updates using an evolutionary heuristic method.

There are also more potential choices for the design variable. For example, one choice is to only optimize the density. Li et al. [2021] performed topology optimization of material density (without orientation) using regularizers that force the fiber material to form lines. However, they did not extract fiber paths explicitly at the end, so it is unclear whether the fibers are directly printable. Li et al. [2020] and Chen and Ye [2021] proposed to lay fibers along with the load transmission trajectories. Almeida Jr. et al. [2019] proposed to perform the SIMP (Solid Isotropic Material with Penalization) method first, designed the fiber pattern manually, and then used a genetic algorithm to determine the number of fiber rings/paths that would minimize compliance (defined as mass divided by stiffness). Sugiyama et al. [2020] proposed to calculate the stress field and update fiber paths so that they follow the direction of maximum principal stress, repeating this process until convergence. Apart from these two-stage approaches, there are also end-to-end approaches based on genetic algorithms. For example, Yamanaka et al. [2016] modeled fiber paths as streamlines and optimized them directly using a genetic algorithm.

In summary, most existing works perform fiber planning in two stages (topology/orientation optimization followed by path extraction). In contrast, our method performs an end-to-end optimization of the fiber layout, maximizing the regularized object stiffness via partial differential equations (PDEs) [De los Reyes 2015]. There are two common types of algorithms to solve PDE-constrained optimization problems: all-at-once and black-box [Hertzog and Kunisch 2010]. All-at-once treats both the design variable and the state variable as independent variables, so the method may not satisfy the constraints before the optimization finishes. A common all-at-once algorithm is SQP (sequential quadratic programming) [Boggs and Tolle 1995]. A disadvantage of the all-at-once approach is the dimension of the state variable can be very large, which makes the optimization costly. Black-box solves the problem in reduced form, by treating the design variable as the only independent variable, so that a gradient-based optimizer can be applied (e.g., gradient descent, Newton’s method). We formalize the fiber path planning task as a PDE-constrained optimization problem and use the black-box approach specifically the adjoint method, to solve it.

3 METHOD
The pipeline of our method is shown in Figure 2. Starting from a goal (a shape with some external loads), we first simulate the stress field using the finite element method (§ 3.1). We apply a greedy fiber extraction algorithm by “walking” in the stress field, and then downsample the greedy path (§ 3.2). We build and optimize an objective function based on the object’s stiffness and regularity conditions of the fiber paths, using the adjoint method to calculate the gradients of the objective (§ 3.3). These steps can be repeated until a desired number of fiber paths are extracted and optimized. We then perform a coarse-to-fine optimization by upsampling and optimizing the fiber paths a specified number of times (§ 3.4).

3.1 Simulation
In this subsection, we describe how we solve the stress field given a shape, some external loads, and a specified fiber layout. We denote the body as \( \Omega \), the stress tensor as \( \sigma \), the strain tensor as \( \epsilon \), the displacement vector as \( u \), and the stiffness tensor as \( C \). The linear elastic model can be written as

\[
-\nabla \cdot \sigma = f,
\]

\[
\epsilon = \frac{1}{2} (\nabla u + (\nabla u)^\top),
\]

\[
\sigma = C : \epsilon,
\]

where the colon is the dot product, \( f \) is the body force and we set it to \( 0 \). For certain regions on the boundary of \( \Omega \) (i.e., \( \partial \Omega \)), the value of \( u \) is given as input (i.e., Dirichlet boundary condition). For the remaining regions, we have \( \sigma \cdot n = T \) (i.e., Neumann boundary condition), where \( n \) is the outward unit normal vector, and \( T \) is the tractive force which we set to \( 0 \).

The constitutive equations \( \sigma = C : \epsilon \) can also be written in a matrix product form; under the assumption of in-plane stress, we have

\[
\begin{bmatrix}
\sigma_{11} \\
\sigma_{22} \\
\sigma_{12}
\end{bmatrix}
= \begin{bmatrix}
E_1 \delta_{11} & E_1 \delta_{12} & 0 \\
E_1 \delta_{12} & E_1 \delta_{22} & 0 \\
0 & 0 & \mu
\end{bmatrix}
\begin{bmatrix}
\epsilon_{11} \\
\epsilon_{22} \\
2\epsilon_{12}
\end{bmatrix},
\]

where \( E_1 \) and \( E_2 \) are Young’s moduli, \( \nu_{12} \) and \( \nu_{21} \) are the Poisson’s ratios, and \( \mu \) is the shear modulus. For simplicity, we assume both plastic and fiber are isotropic materials, and they have different Young’s moduli \( E_{\text{plastic}} \) and \( E_{\text{fiber}} \) and identical Poisson’s ratio \( \nu \).
The next issue is to calculate the Young’s modulus field. Consider a laminate of height \( h_{\text{object}} \), with some layers filled with just plastic and others containing both plastic and fiber. We assume that all layers with fiber, adding up to a total height of \( h_{\text{fiber}} \), have identical fiber paths, and omit plastic where fiber is present. The set of fiber paths is denoted as \( P \), and every path \( p \) in it is a sequence of vertices on the fiber path. For a point \( x \in \Omega \), for the purpose of differentiability, we define its “soft” Young’s modulus as

\[
E(x) := E_{\text{plastic}} \cdot \alpha_{\text{plastic}}(x) + E_{\text{fiber}} \cdot \alpha_{\text{fiber}}(x),
\]

where

\[
\alpha_{\text{fiber}}(x) := \sum_{p \in P} \exp\left( -\frac{(\text{dis}(p, x))^2}{w_{\text{fiber}}/2} \right) \cdot h_{\text{fiber}},
\]

\[
\alpha_{\text{plastic}}(x) := h_{\text{object}} - \min(\alpha_{\text{fiber}}(x), h_{\text{fiber}}).
\]

We allow fiber paths to overlap in this setting, as even in real prints from the Markforged Mark Two, we do not observe any problems. We then have \( \mu(x) = \frac{E(x)}{2(1+\nu)} \). Finally, we solve the PDE in Equation 1 using FEniCS with DOLFIN [Logg and Wells 2010] by solving its first-order condition on linear triangular finite elements. Figure 2 visualizes an example of the calculated stress field, using line integral convolution [Cabral and Leedom 1993].

### 3.2 Greedy fiber path extraction

In this subsection, we describe how we greedily extract a fiber path from a stress field along the directions of maximum tensile stress or perpendicular to the direction of maximum compressive stress. With the stress tensor \( \mathbf{\sigma} \) calculated from § 3.1, we first calculate the stress on plastic:

\[
\sigma_{\text{plastic}} := \sigma \cdot \frac{E_{\text{plastic}} \cdot \alpha_{\text{plastic}}}{E_{\text{plastic}} \cdot \alpha_{\text{plastic}} + E_{\text{fiber}} \cdot \alpha_{\text{fiber}}},
\]

For the stress matrix at any point \( x \in \Omega \), we can calculate its eigenvalue with the largest absolute value \( \lambda(x) \) and the corresponding eigenvector \( v(x) \). We then build a scalar field with \( |\lambda(x)| \) and randomly sample a starting point \( x_0 \) with the field as sampling weights. From the starting point, we walk in both directions along with \( \pm v(x_0) \) (or perpendicular to \( v(x_0) \) if \( \lambda(x_0) \) is negative) at a fixed step size of 0.5 mm. If we walk outside \( \Omega \) or within 1.3 mm to \( \partial \Omega \) (number measured from prints from Eiger), we retry at most 19 times with a random rotation uniformly sampled between \( -\pi/12 \) to \( \pi/12 \). The algorithm stops when a preset length limit is reached, or we cannot walk in both directions even after retries.

We then downsample the extracted fiber path by keeping 1 of every 20 vertices. We iterate every subsequence of the downsampled path and select the one that minimizes the objective function we will define in Section 3.3. We repeat this process 10 times (sampling 10 starting points) and keep the one that minimizes the objective function.

### 3.3 Gradient calculation and optimization

In this subsection, we describe how we design an objective function and minimize it using a gradient-based optimizer. We denote the optimized strain energy in Equation 1 as \( U \), and the set of fiber paths as \( P \). The objective \( \mathcal{L}(P) \) is defined as

\[
- U + \sum_{p \in P} \left( w_{\text{lap}} \cdot \mathcal{L}_{\text{lap}}(p) + w_{\text{min}_1} \cdot \mathcal{L}_{\text{min}_1}(p) + w_{\text{bdy}} \cdot \mathcal{L}_{\text{bdy}}(p) \right),
\]

where
We implement this end-to-end differentiation automatically using the BFGS implementation in SciPy [Virtanen et al. 2020]. where dis($p_i$) is a count of the total number of segments in all fiber paths (i.e., $\sum_{p \in P} |p| = |P|$). The reason to apply the $s(P)^3$ multiplier is because the Laplacian regularizer is sensitive to upsampling, which we discuss in § 3.4, and this multiplier keeps our Laplacian regularizer scale-invariant. The minimum-length regularizer $L_{\text{min}_l}$ penalizes infeasibly-short fiber paths:

$$L_{\text{min}_l}(p) := \max \left( l_{\text{min}} - \sum_{i=2}^{d} \| p_i - p_{i-1} \|, 0 \right)^2,$$

where $l_{\text{min}}$ is the minimum fiber length that can be printed by the 3D printer. The boundary regularizer $L_{\text{bdy}}(p)$ penalizes fiber paths outside $\Omega$ or too close to $\partial \Omega$:

$$L_{\text{bdy}}(p) := \sum_{i} \max(d_{\text{min}} - \text{dis}(p_i, \Omega), 0)^2,$$

where $\text{dis}(p_i, \Omega)$ measures the distance from $p$ to $\partial \Omega$ (positive for $p_i \in \Omega$, negative otherwise) and $d_{\text{min}}$ is the lower limit of distance from fiber to the boundary.

The next step is to calculate $\frac{dL(P)}{dp}$. Here we apply the adjoint method. Denote the first-order condition of Equation 1 as $F(u, P) = 0$. By the implicit function theorem (under proper regularity conditions) $u$ can be thought of a function of $F$, and the derivative $\frac{du}{dp}$ is well-defined. Taking the derivative of $F$ with respect to $P$, we have

$$\frac{dF}{dp} = \frac{\partial F}{\partial u} \frac{du}{dp} + \frac{\partial F}{\partial P} = 0,$$

which leads to

$$\frac{dL(P)}{dp} = -\frac{\partial L(P)}{\partial u} \left( \frac{\partial F}{\partial u} \right)^{-1} \frac{dF}{dp} + \frac{\partial L(P)}{\partial P}.$$

We implement this end-to-end differentiation automatically using dolfin-adjoint [Mitosch et al. 2019] and PyTorch [Paszke et al. 2019].

### 3.4 Coarse-to-fine optimization

To speed up the optimization, we perform multigrid optimization. As described in § 3.2, we initially downsample all the fiber paths. Then, for every fiber path $p$, we insert midpoints between every $p_i$ and $p_{i+1}$ by B-spline interpolation, using SciPy, and optimize all the fiber paths. This process can be repeated several times to generate the final fiber paths for 3D printing.

### 4 FABRICATION AND EXPERIMENTAL SETUP

In this section, we describe how we manufacture real 3D prints and measure their position-load curves. We use a Markforged Mark Two printer with nylon as the plastic material and carbon fiber as the reinforcing fiber material. We print laminates with a height of 2 mm and 16 layers, of which the 4th, 7th, 10th, and 13th layers are fiber layers. All layers without fiber and regions in fiber layers without fiber are filled with nylon (solid fill).

For the 2D shape, we use a 46 mm $\times$ 30 mm rectangle with two rounded isosceles trapezoidal holes, the same shape as shown in Figure 2. The isosceles trapezoids have two sides of 11 mm and 14 mm and a height of 11 mm, with every corner smoothed by an arc with a radius of 1 mm. We will reuse this shape in § 5, § 6.3, and § 7.

To measure the position-load curve of a print, we insert two square nuts into both its holes and apply tension to them using a universal testing machine (Instron 600DX), as shown in Figure 3. The machine is programmed to move at a speed of 20 mm/min until the object breaks or by a manual stop.

![Figure 3: A 3D printed part being tested on a universal testing machine (Instron 600DX), with square nuts in the trapezoidal holes. The machine moves at a speed of 20 mm/min and stops when the object breaks or by a manual stop.](image3)

![Figure 4: Nine different fiber layouts printed for moduli calculation (left to right, top to bottom): no fiber path, 1 to 3 inner rings, 1 to 3 outer rings, 1 to 2 rings for all walls.](image4)
We use Eiger to generate nine different layouts of carbon fiber paths: with loads between 150 N and 300 N, to calculate the stiffness. As described in Section 4, we test the prints and record their position-violates our in-plane stress assumption. We therefore measure the of 0.40 GPa for nylon and 20.1 GPa for carbon, with results shown between measured and simulated stiffness. The search yields moduli of the two holes on the rectangle. We calculate the strain energy of the object, then do a grid search for the values of the moduli of the material, we print all of them in one batch, as shown in Figure 4. Due to the variability of the printing process, we print three batches of these nine prints and pick the batch with the best printing quality.

5.1 3D prints for testing
We use Eiger to generate nine different layouts of carbon fiber paths: no fiber path, 1 to 3 inner rings, 1 to 3 outer rings, 1 to 2 rings for all walls. To reduce the bias introduced by the non-uniformity of the material, we print all of them in one batch, as shown in Figure 4. Due to the variability of the printing process, we print three batches of these nine prints and pick the batch with the best printing quality.

5.2 Stiffness measurement
As described in Section 4, we test the prints and record their position-load curves (Figure 5). Note that the beginning of every curve can be noisy as the part is not perfectly vertical, etc. Additionally, a large load can cause the part to buckle out of the 2D plane, which violates our in-plane stress assumption. We therefore measure the position change between a load of 150 N and a load of 300 N for every print, and calculate the stiffness by dividing load change (150 N) by position change (in mm). The results are shown in Figure 6, marked as “X”. Note that there is a factor of 0.5 when converting stiffness in N/mm to energy in N-mm at 1 mm displacement (e.g., a stiffness of 500 N/mm corresponds to having strain energy of 250 N-mm at 1 mm displacement).

5.3 Simulation and modulus calculation
For each measured data point, we apply Dirichlet boundary conditions corresponding to 1 mm displacement on the two shorter sides of the two holes on the rectangle. We calculate the strain energy of the object, then do a grid search for the values of the moduli of nylon and carbon that minimize the sum of squared distances between measured and simulated stiffness. The search yields moduli of 0.40 GPa for nylon and 20.1 GPa for carbon, with results shown in Figure 6. As we can see, the simulation results mostly match the real results, with small residuals relative to the energy.

6 EXPERIMENTS
In this section, we present detailed evaluations of the performance of our method in both simulation and real experiments on four case studies (§ 6.1–§ 6.4), then show several additional results in § 6.5. We start with two simple shapes—a rectangle and a “plus” shape—then move to more complex shapes: rectangles with two and four holes (Figure 7). The first baseline we use is concentric fiber rings from Eiger, which have three different types: inner, outer, and all walls. For the next two case studies, to better illustrate the effectiveness of our algorithm on complex shapes and loads, we include two additional baselines: (1) greedy, simplifying our algorithm by removing all the optimization components and directly generating results using the greedy algorithm; (2) field-opt-greedy, similar to greedy but with an additional step of field optimization before running the greedy algorithm. The latter baseline, intended to represent the approach of previous work on fiber orientation optimization (see § 2.1), optimizes a vector field that aligns to the stress direction, with a smoothing regularizer. Additional details about the field optimization can be found in the appendix. We refer to the results from our method as optimized. For all the experiments (unless otherwise specified), we use the BFGS optimizer and limit the maximum number of iterations to 500 and a gradient tolerance of $3 \times 10^{-9}$. The objective function is set with $w_{\text{up}} = 1 \times 10^{-8}$, $w_{\text{min}} = 1$, and $w_{\text{bdy}} = 1$. 
Figure 7: Four shapes we use in our case studies. (a) and (b) are relatively simple shapes, and the loads are applied on the two sides. For (c), a rectangle with two holes, tension is applied on the two shorter sides of the holes. (d) is designed to be a multi-functional rectangle with four holes, and the user can choose one hole from the left two holes and another hole from the right two holes to apply tension.

Figure 8: Step-by-step visualization of how our method extracts three fiber paths, optimizes them, and performs coarse-to-fine optimization on the rectangle shape. In the first row, we extract the first fiber path, optimize it, extract the second fiber path, and optimize both paths. The two paths curve and move up and down after the optimization, respectively. In the second row, we extract a third greedy path and optimize the three paths together. The last step does not help much since the task is relatively simple. As shown in Figure 9, for a fixed displacement of 1 mm, our algorithm uses less fiber while achieving higher energy, as the baseline lays vertical fibers that are much less useful than horizontal fibers.

Figure 9: Fiber paths and energy maps of outer and optimized at 1 mm displacement on the rectangle shape. We use less fiber while achieving higher energy, as the baseline lays vertical fibers that are much less useful than horizontal fibers.

Figure 10: Fiber paths, lengths, and strain energy at 1 mm displacement of outer and optimized on the plus shape. With the help of optimization, fiber paths automatically distribute themselves uniformly in the space as we increase the number of fiber paths. By laying slightly bending fibers in horizontal directions, we save fiber while increasing the energy, compared to outer, which lays fiber in unrelated regions.

6.1 Case 1: rectangle
In this case study, we show how our algorithm works step by step on a rectangle (45 mm × 30 mm), with tension applied to its two shorter sides (Figure 7a). As we can see in the first row in Figure 8, we first greedily extract a fiber path and optimize it. When we add and optimize a second fiber path, the two paths separate and curve. In the second row, we extract a third greedy path and optimize the three paths together. Finally, we double the number of points of both fiber paths and optimize the three paths together. The last step does not help much since the task is relatively simple. As shown in Figure 9, for a fixed displacement of 1 mm, our algorithm uses less fiber while achieving higher energy in simulation, compared to 1 outer concentric fiber ring, which lays fibers in vertical directions that are much less useful than fibers in horizontal directions.

6.2 Case 2: “plus” shape
As shown in Figure 7b, we use a “plus” shape whose edges are all of length 15 mm, and we apply tension to two sides of the shape. We compare fiber paths of outer and optimized in Figure 10, with three solutions from each strategy. As we can see, outer lays fibers in regions of low relevance to the loads applied, in contrast
Figure 11: Energy-fiber usage plot of outer and optimized at 1 mm displacement on the plus shape. Our method improves over the Pareto front of outer by laying fibers according to the external loads.

Table 1: Real (measured) stiffness of greedy (g), field-opt-greedy (f), and optimized (o) on the rectangle with two holes shape, measured between 150 N and 300 N (4 batches). Optimized performs consistently better than the baselines when using a similar or less amount of fiber.

| Stiffness (N/mm) | Solution 1 | Solution 2 |
|------------------|------------|------------|
| Batch 1 | g 490.1 f 574.6 o 625.5 | g 745.0 f 741.3 o 992.6 |
| Batch 2 | g 584.3 f 692.0 o 756.0 | g 801.0 f 741.3 o 985.2 |
| Batch 3 | g 483.5 f 485.0 o 656.7 | g 671.6 f 603.4 o 953.5 |
| Batch 4 | g 491.7 f 481.7 o 603.0 | g 670.0 f 670.9 o 970.4 |
| Average | g 512.4 f 558.3 o 660.3 | g 721.9 f 689.2 o 975.4 |
| Length (mm) | 400.0 | 400.0 |

paths tightly around the holes while aligned with stress direction, yielding larger strain energy when using a similar amount of fiber.

Real experiment results. To evaluate the quality of fiber paths, we perform real-world experiments by applying tension to 3D prints on a universal testing system (600DX from Instron). Due to the limited space on the printer bed, two sets of comparisons are performed separately: (1) inner, outer, and all walls vs. optimized; (2) greedy and field-opt-greedy vs. optimized. We thus printed eight batches, four for each set of comparisons. Again, as in Section 4, we measure the stiffness of a print by calculating the slope of its position-load curve, picking two points that have loads of 150 N and 300 N. The results of inner, outer, and all walls vs. optimized are shown in Figure 13. As we can see, our algorithm consistently provides significantly higher stiffness than the concentric baselines when using a similar or lower amount of fiber. Note that the fiber lengths may have slight discrepancies between simulation and real-world experiments since they are from different path generation algorithms (one from our re-implementation of Eiger, another from Eiger directly). The results of greedy and field-opt-greedy vs. optimized are shown in Table 1. Again, our algorithm consistently improves the stiffness over the two baselines while using a similar or lower amount of fiber.

6.4 Case 4: rectangle with four holes
As shown in Figure 7d, we also tested a rectangle (84 mm × 28 mm) with four rounded isosceles trapezoid holes. We design the shape to be multi-functional—if we label the holes from 1 to 4 from left to right, we assume the user uniformly chooses one of the four settings: 1) hole 1 and hole 3; 2) hole 1 and hole 4; 3) hole 2 and hole 3; 4) hole 2 and hole 4. To support this multi-functional shape, we simulate all four cases and calculate the average strain energy.

The fiber paths from all the methods are shown in Figure 14. Again, both greedy and field-opt-greedy produce fibers along stress directions with fiber paths from field-opt-greedy being slightly smoother. Optimized lays the first fiber over all holes and lays the second fiber around the middle two holes, due to the multi-functional nature of the shape. The energy-fiber usage plot is shown in Figure 15, where optimized improves upon the Pareto front of every baseline.

6.5 Results on additional shapes
In this subsection, we provide results from our method and baselines on several additional shapes. The shape designs are inspired by sketches from SketchGraphs [Seff et al. 2020], a large-scale dataset of sketches of real-world CAD models, as well as shapes from existing works [Ma et al. 2022; Shafighfarid et al. 2019]. We use a Laplacian regularizer weight \( w_{lap} = 5 \times 10^{-7} \), and the results are shown in Figure 16, with every dotted line a Dirichlet boundary condition. For the first shape, all methods use a similar amount of fiber but optimized achieves much higher energy than others. For the second shape, optimized uses a similar amount of fiber as concentric, less fiber than greedy and field-opt-greedy but achieves higher energy. For the third shape, optimized achieves comparable energy as concentric but saves approximately 70% of fiber. Compared to greedy and field-opt-greedy, optimized achieves much higher energy
while using slightly more fiber. For the fourth and fifth shapes, optimized uses less fiber or comparable fiber as other baselines while achieving significantly higher energy.

7 ABLATION STUDIES

Our algorithm without optimization has been studied in Section 6 as the greedy baseline. In this section, we study the effects of removing two other components of our method: the Laplacian regularizer and the multi-resolution optimization, using the shape rectangle with two holes (Figure 7c).

7.1 Ablation study of the Laplacian regularizer

As both the minimum-length regularizer and the boundary regularizer are intuitively necessary for fiber paths to be long enough for printing purposes and within the object boundary, we study the effect of removing the Laplacian regularizer from the optimization. We run our algorithm with the same hyper-parameter setting except for $w_{lap} = 0$. We extract one fiber path and upsample for one time. As shown in Figure 17, the optimizer successfully optimizes the low-resolution path as the number of points is still small (Figure 17a), but introduces jagged results with more degrees of freedom (Figure 17b), demonstrating the need for some form of regularization.

7.2 Ablation study of multi-resolution optimization

In this subsection, we study how the multi-resolution approach speeds up the optimization process. For the multi-resolution case,
Figure 14: Fiber paths, lengths, and strain energy at 1 mm displacement of inner, outer, all walls, greedy, field-opt-greedy, and optimized on the multi-functional rectangle with four holes shape. Similarly, greedy and field-opt-greedy lay fibers along stress directions, and field-opt-greedy provides slightly smoother fiber paths. Optimized lays the first fiber over all the holes, and the second fiber around the middle two holes, with paths tightly around the holes.

Figure 15: Energy-fiber usage plot (at 1 mm displacement) of all methods on the rectangle with four holes shape. The comparison between concentric fiber rings and optimized is shown on the left, and the comparison between greedy-based baselines and optimized is shown on the right. Optimized improves the Pareto front of all the baselines by laying fibers tightly around the holes.

In this work, we studied the task of fiber path planning in 3D printing for given external loads, aiming at maximizing the stiffness. We proposed an end-to-end optimization approach that optimizes regularized object stiffness directly to the fiber layout, rather than an intermediate fiber orientation field, with the help of the adjoint.

we extract one fiber path, downsample its resolution by a factor of 20, optimize it, and upsample and optimize it three times, with every optimization limited to 100 iterations. For the single-resolution case, we also extract one fiber path, downsample its resolution by a factor of 2, optimize it and limit the maximum number of optimization iterations to 400. For a fair comparison, we use the same random seed for both cases when sampling starting points of the greedy path extraction algorithm. As shown in Figure 18, both cases get similar fiber paths with similar strain energy, but multi-resolution optimization reduces the running time by approximately 40%.

8 DISCUSSION
In this work, we studied the task of fiber path planning in 3D printing for given external loads, aiming at maximizing the stiffness. We proposed an end-to-end optimization approach that optimizes regularized object stiffness directly to the fiber layout, rather than an intermediate fiber orientation field, with the help of the adjoint.
Figure 16: Fiber paths and energy at 1 mm displacement of all methods on additional shapes. Every dotted line indicates a Dirichlet boundary condition. For the first shape (from top to bottom), optimized achieves significantly higher energy while using slightly less fiber than all baselines. For the second shape, optimized achieves higher energy while using a similar amount of fiber as concentric and less fiber than greedy and field-opt-greedy. For the third shape, optimized saves approximately 70% of fiber usage while achieving similar energy as concentric. It also achieves much higher energy than greedy and field-opt-greedy while using slightly more fiber. For the last two shapes, compared to other baselines, optimized achieves significantly higher energy while using a less or comparable amount of fiber.

method and automatic differentiation. We perform planning by extracting fiber paths using a greedy algorithm that lays fiber paths along stress directions, followed by coarse-to-fine optimization. To apply our method, we first measure the effective moduli of plastic and fiber by manufacturing and testing real 3D prints. We then study our method with three baselines on four case studies and several additional shapes. The first baseline is concentric fiber rings from Eiger, a leading digital manufacturing software package developed by Markforged. The second baseline is our method with the optimization part removed, producing fiber paths from the greedy path extraction algorithm. The third baseline includes a fiber field optimization part which smooths the stress field before using it in the greedy algorithm. We demonstrated that, both in simulation and real experiments, our method could generate shorter fiber paths while achieving greater stiffness (i.e., we improved the Pareto front). We also studied the effects of removing the Laplacian regularizer and the multi-resolution optimization, showing the Laplacian regularizer is necessary for the optimization to be stable and multi-resolution optimization helps reduce the running time.
We would also like to mention some limitations of our method. First, our simulation simplifies the task by assuming linear elasticity, restricting to in-plane stress, and treating both plastic and fiber as isotropic materials with different Young’s moduli and identical Poisson’s ratio. Lifting these assumptions would introduce greater mathematical complexity, but would require no conceptual changes to our approach. Additionally, the planning is not performed in real time. For example, to plan fiber paths for the shape rectangle with two holes, our method uses 10 minutes and 18 minutes to generate the two studied solutions, respectively. Relative to the time required to design and print a part, this represents only a small increase. In this case, we save approximately 40% of running time by multi-resolution optimization.

We acknowledge the contributions of Civil and Environmental Engineering for providing discussions and feedback, Joseph Vokaturo and the Princeton Department of Civil and Environmental Engineering for providing the universal testing machine, as well as Markforged. This work is partially supported by the Princeton School of Engineering and Applied Science, as well as the U.S. National Science Foundation under grants #IIS-1815070, #IIS-2007278, and #OAC-2118201.

ACKNOWLEDGMENTS

We would like to thank Amit Bermano and members of the Princeton Laboratory for Intelligent Probabilistic Systems for valuable discussions and feedback, Joseph Vokaturo and the Princeton Department of Civil and Environmental Engineering.
Markforged. 2023b. Eiger 3D Printing Software. https://markforged.com/software.

Sebastian K Mitsuich, Samson W Punke, and Jørgen S Dokken. 2019. dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. Journal of Open Source Software 4, 38 (2019), 1292.

Vasilios S Papapetrou, Chitrang Patel, and Ali Y Tamjani. 2020. Stiffness-based optimization framework for the topology and fiber paths of continuous fiber composites. Composites Part B: Engineering 183 (2020), 107681.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Alexander A Salton. 2019. 3D topology optimization of continuous fiber-reinforced structures via natural evolution method. Composite Structures 215 (2019), 289–297.

Martin-Pierre Schmidt, Laura Couret, Christian Gout, and Claus BW Pedersen. 2020. Structural topology optimization with smoothly varying fiber orientations. Structural and Multidisciplinary Optimization 62, 6 (2020), 3105–3126.

Ari Seif, Yaniv Ovadia, Wenda Zhou, and Ryan P. Adams. 2020. SketchGraphs: A Large-Scale Dataset for Modeling Relational Geometry in Computer-Aided Design. In ICML 2020 Workshop on Object-Oriented Learning.

Torkan Shafighfard, Eralp Demir, and Mehtem Yildiz. 2019. Design of fiber-reinforced variable-stiffness composites for different open-hole geometries with fiber continuity and curvature constraints. Composite Structures 226 (2019), 111280.

Nurhalida Shahrubudin, Te Chuan Lee, and Rhaizan Ramlan. 2019. An overview on 3D printing technology: Technological, materials, and applications. Procedia Manufacturing 35 (2019), 1286–1296.

Kentaro Sugiyama, Ryosuke Matsuzaki, Andrei V Malakov, Alexander N Polilov, Masahito Ueda, Akira Todoroki, and Yoshinari Hirano. 2020. 3D printing of optimized composites with variable fiber volume fraction and stiffness using continuous fiber. Composites Science and Technology 186 (2020), 107905.

Xingyu Sun, Tianju Xue, Szymon Rusinkiewicz, and Ryan P. Adams. 2021. Amortized Synthesis of Constrained Configurations Using a Differentiable Surrogate. In Advances in Neural Information Processing Systems (NeurIPS).

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature methods 17, 3 (2020), 261–272.

Ting Wang, Nanya Li, Guido Link, John Jelonnek, Jürgen Fleischer, Jörg Dittus, and Daniel Kupzak. 2021. Load-dependent path planning method for 3D printing of continuous fiber reinforced plastics. Composites Part A: Applied Science and Manufacturing 140 (2021), 106181.

Yusuke Yamanaka, Akira Todoroki, Masahito Ueda, Yoshinari Hirano, Ryosuke Matsuzaki, et al. 2016. Fiber line optimization in single ply for 3D printed composites. Open Journal of Composite Materials 6, 04 (2016), 121.

Leen Zhang, Xiaoping Wang, Jingyi Pei, and Yu Zhou. 2020. Review of automated fibre placement and its prospects for advanced composites. Journal of Materials Science 55, 17 (2020), 7121–7155.
A APPENDIX: FIELD OPTIMIZATION

Given a stress field \( \sigma \), we would like to find a fiber field \( v : \Omega \to \mathbb{R}^2 \) such that (1) its direction is aligned with \( \sigma \); (2) it is smooth. We solve \( v \) by minimizing an objective function that reflects both properties:

\[
L(v; \sigma) := \alpha_{\text{stress}} \cdot L_{\text{stress}}(\hat{v}; \sigma) + \alpha_{\text{smooth}} \cdot L_{\text{smooth}}(\hat{v}),
\]

where \( \alpha_{\text{stress}} \) and \( \alpha_{\text{smooth}} \) are hyper-parameters, and

\[
\hat{\sigma}(x, y) := \sigma(x, y) / \| \sigma(x, y) \|,
\]

is the normalized \( \sigma \), as the objective function should be invariant regardless of the length of \( \sigma(x, y) \)'s. Note that the objective function should also be invariant if we randomly flip some \( \sigma(x, y) \)'s, which needs some special handling, as we will discuss below.

Consistent with \( \sigma \). For a specific point \((x, y) \in \Omega\), we calculate the tension in the stress field \( \sigma \) along \( \hat{\sigma}(x, y) \), which is

\[
\hat{\sigma}(x, y)^T \sigma(x, y) \hat{\sigma}(x, y).
\]

We then integrate it over \( \Omega \) and get

\[
L_{\text{stress}}(\hat{v}; \sigma) := -\iint_{\Omega} \hat{\sigma}(x, y)^T \sigma(x, y) \hat{\sigma}(x, y) \, dx \, dy,
\]

where the negative sign indicates we would like to maximize the tension along the field direction.

Smoothness. We penalize the squared Frobenius norm of the gradient of \( \hat{v} \):

\[
L_{\text{smooth}}(\hat{v}) := \iint_{\Omega} \| \nabla \hat{v}(x, y) \|_F^2 \, dx \, dy.
\]

Note that the penalty should be invariant to flips of \( \hat{v}(x, y) \)'s, so we handle this invariance when calculating the finite difference:

\[
\| \nabla \hat{v}(x, y) \|_F^2 := \min \left( \frac{\| \hat{v}(x + h, y) - \hat{v}(x, y) \|_F^2}{h}, \frac{\| \hat{v}(x, y + h) - \hat{v}(x, y) \|_F^2}{h} \right) + \min \left( \frac{\| \hat{v}(x - h, y) - \hat{v}(x, y) \|_F^2}{h}, \frac{\| \hat{v}(x, y - h) - \hat{v}(x, y) \|_F^2}{h} \right),
\]

where \( h \) is the step size.

In the experiments, we set \( \alpha_{\text{stress}} \) to 1 and \( \alpha_{\text{smooth}} \) to 0.02. We use the BFGS optimizer with a gradient tolerance of \( 1 \times 10^{-6} \) and set the maximum number of iterations to 100.