Differentiable Stock Constraints for Continuous Optimization of Frame Structures by Reuse

Zhouyuan Chen
Centre Inria d'Université Côte d'Azur
Sophia Antipolis, France
zhouyuan.chen@inria.fr

Marzia Riso
Centre Inria d'Université Côte d'Azur
Sophia Antipolis, France
marzia.riso@inria.fr

Berend Baas
Centre Inria d'Université Côte d'Azur
Sophia Antipolis, France
berend.baas@inria.fr

Adriana Schulz
Brown University
Providence, USA
adriana_schulz@brown.edu

Adrien Bousseau
Centre Inria d'Université Côte d'Azur
Sophia Antipolis, France
adrien.bousseau@inria.fr

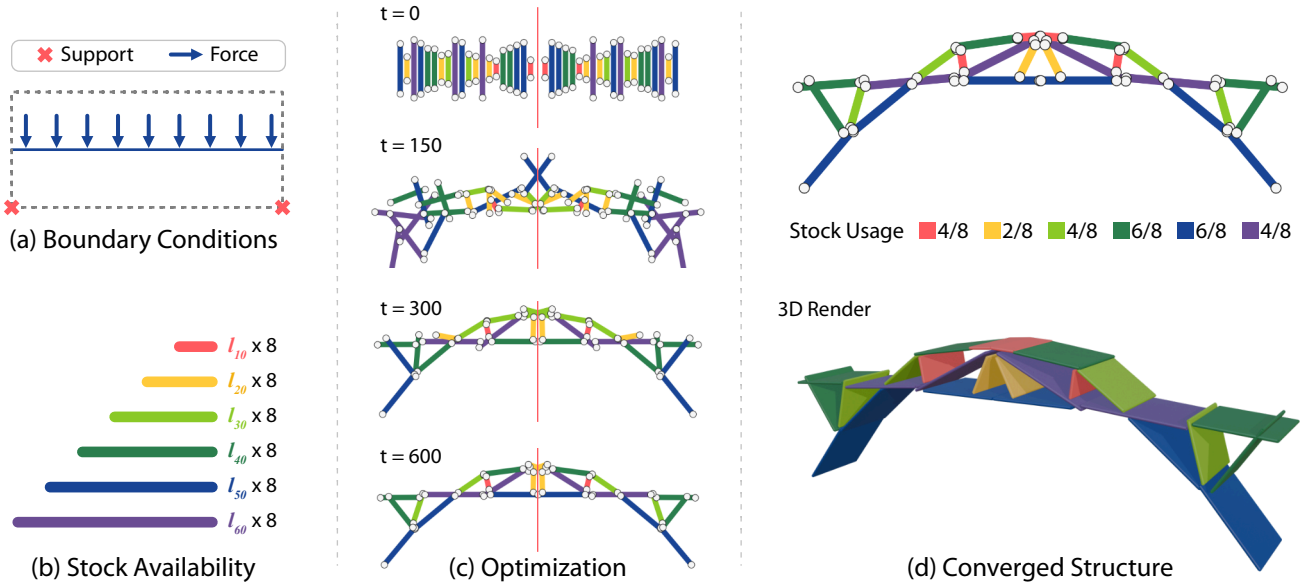


Figure 1: Starting from the prescribed boundary conditions (a) and a stock of beams to be reused (b), our differentiable optimization continuously selects and place beams to form a load-bearing frame structure (c), while maintaining symmetry along the vertical axis. The optimized structure (d) respects the stock constraints.

Abstract

The growing awareness of resource limitations has driven increasing interest in reusing existing materials for structural design. Existing methods formulate reuse as an assignment problem, incorporating the discrete constraints on the dimension and availability of reclaimed materials into complex combinatorial problems, which are often computationally expensive and restricted to predefined structural layouts.

We present a continuous gradient-based formulation for the design of frame structures from a stock of decommissioned beams. Building on differentiable density-based solvers, our method enables end-to-end optimization of beams rotation, translation and length, without requiring a specific layout. By expressing the discrete beam types and stock availability as differentiable objective terms, our system converges to solutions that respect the stock constraints while discovering alternative layouts with structural properties comparable to those produced by previous approaches.

We also demonstrate the flexibility of our continuous formulation by incorporating additional objective terms, such as image-based guidance for structural design, and by extending it to support other structural elements, like curved arcs.

This work is licensed under a Creative Commons Attribution 4.0 International License.

SCF '26, Tokyo, Japan

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2821-1/2026/12
https://doi.org/10.1145/3828611.3845645

CCS Concepts

• **Computing methodologies** → *Modeling and simulation*; *Computer graphics.*

Keywords

Circular design, structural reuse, frame structures, topology optimization, differentiable optimization

ACM Reference Format:

Zhouyuan Chen, Marzia Riso, Berend Baas, Adriana Schulz, and Adrien Bousseau. 2026. Differentiable Stock Constraints for Continuous Optimization of Frame Structures by Reuse. In *Proceedings of the 11th ACM Symposium on Computational Fabrication (SCF '26), December 06–08, 2026, Tokyo, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3828611.3845645

1 Introduction

The circular economy aims to reduce material consumption and waste through reuse and repair. In particular, *structural reuse* consists of creating new products by reusing elements reclaimed from decommissioned products [Iacovidou and Purnell 2016], for example to produce furniture from decommissioned wind turbine blades [Joustra et al. 2021], or truss structures from disassembled electric pylons [Brütting et al. 2019]. But structural reuse requires a paradigm shift in the way products are designed: in contrast to traditional manufacturing where raw materials are processed to fit the dimensions of the envisioned product, reuse implies that the dimensions of the product are dictated by the stock of reclaimed elements – *form follows availability* [Brütting et al. 2019; Gorgolewski 2018].

From a digital design and manufacturing standpoint, this change of paradigm typically translates into combinatorial optimization problems, where the goal is to select elements from the available stock to best approximate a target design. In this paper, we consider the problem of designing frame structures composed of reclaimed beams (Fig. 1). Prior work formulates beam reuse as an assignment problem solved with *discrete* combinatorial optimization tools such as the Hungarian algorithm, genetic algorithms, or mixed-integer programming [Brütting et al. 2019; Huang et al. 2021]. A first difficulty in developing such discrete approaches is to model both the structural objectives and the stock objectives as linear equality and inequality constraints over binary selection variables. A second difficulty is that the complexity of combinatorial optimization quickly grows with the number of candidate beams in the structure, which is why existing methods often restrict the beam assignment to a coarse pre-defined structural layout. In this paper, we demonstrate the use of an alternative *continuous* formulation based on gradient descent, which is easy to implement in modern auto-differentiation frameworks, can incorporate diverse objective terms, and can discover arbitrary layouts of beams.

The first challenge we face is that the stock constraints imposed by reuse are inherently discrete. First, beams in a reclaimed stock only cover a finite set of dimensions. Second, each type of beam is only available in a limited number. We show how to relax these discrete constraints into continuous objective terms by associating each beam type with a basis function, such that the beam dimensions can be interpolated continuously and the total stock usage of a solution can be evaluated by summing the (possibly fractional) contribution of each beam type. We complement this scheme with a regularization term that encourages beam dimensions to snap to their closest values in the stock. The second challenge is to evaluate the physical performance of the frame structure in a differentiable manner. Our solution is to express individual beams as fuzzy density fields, which accumulate to form the complete structure. This formulation allows us to leverage an existing differentiable density-based solver originally developed for gradient-based topology optimization [Hoyer et al. 2019]. The key difference is that, instead of optimizing the density of individual grid cells as in topology optimization, we optimize the position, rotation, and dimensions of individual beams that we rasterize over that grid.

We compare the structures obtained with our approach to the ones documented in prior work, as well as to the optimal structures produced by topology optimization and to a baseline that consists in extracting the medial axis of these structures and approximating it with beams. Furthermore, we showcase the flexibility of our gradient-based formulation by incorporating user guidance on the overall shape of the structure in the form of an image-matching loss, and by extending our algorithm to the reuse of circular arcs rather than straight beams.

2 Related Work

Computational tools for sustainable shape design. Several strategies exist to promote sustainability in design and manufacturing [Allwood et al. 2011], and diverse digital tools have been proposed to support these strategies for shape design. Early work has considered promoting modularity in architectural structures, typically by rationalizing surfaces such that they are made of the same elements [Brütting et al. 2021; Chen et al. 2023; Fu et al. 2010; Liu et al. 2023; Singh and Schaefer 2010], or built from the same molds [Eigensatz et al. 2010; Pellis et al. 2021]. Another strategy consists in designing shapes that can be manufactured with minimal material waste [Koo et al. 2016; Wu et al. 2019; Zhang et al. 2024]. Closer to our work are digital tools to design new structures through reuse. While some papers consider the reuse of surface panels [Baas et al. 2025; Pupping et al. 2025; Qi et al. 2025], the majority of existing work focuses on the reuse of beams and forks for truss, frame and grid shell structures [Amtsberg et al. 2020; Brütting et al. 2020; Brütting et al. 2019; Bukauskas et al. 2017; Favilli et al. 2025; Gordon and De Wolf 2024; Larsson et al. 2019; Seats et al. 2024; Van Marcke et al. 2024]. We also contribute to the problem of beam reuse, specifically to design planar frame structures, i.e., structures made of beams connected by rigid joints. In contrast to truss structures made of triangles, frame structures can include polygons of more than three sides, and as such are subject to bending and shearing in addition to axial forces [Brütting et al. 2020]. While most prior work casts beam reuse as a discrete optimization problem, we show how to express stock constraints as soft terms within a continuous formulation.

Discrete structural optimization. Designing an optimal beam structure requires finding both the layout of the structure (number of beams, nodal connections between beams) and its dimensions (length and cross-section of the beams). In this context, existing reuse approaches seek to best approximate a target structure by selecting beams from a stock to fit elements of the structure (formulated as an *assignment problem*), sometimes also cutting the available beams to produce smaller beams while minimizing waste (formulated as a *cutting stock problem*). Both cases are combinatorial optimization problems and are typically solved with greedy

algorithms, the Hungarian algorithm, or mixed integer linear programming [Huang et al. 2021; Tomczak et al. 2023]. To keep the problem tractable, these methods need to restrict the search space, either by assuming that the layout of the structure is given, or that it is discovered as a subset of a pre-defined over-complete structure.

In the first category, the topology of the design domain is largely fixed, and the problem actually reduces to a stock assignment problem. Tomczak et al. [2023] directly match the existing stock to the target structure, while Bukauskas et al. [2017] assign the existing stock to a predefined structural form through bin-packing. Brütting et al. [2020] simultaneously optimize element sizing and stock assignment for a prescribed frame layout. More recently, Favilli et al. [2025] start from a user-provided grid-shell mesh and optimize the vertex positions and stock assignment through a differentiable procedure, while keeping the connectivity and overall shape of the original mesh.

A second group does allow the layout to change, but only within a design space fixed beforehand. The most common route is the Ground Structure Method [Dorn et al. 1964; Zegard and Paulino 2014], where a dense set of candidate bars is placed over a regular grid and the layout is recovered as a subset of it. [Brütting et al. 2019] build on this to find stock-feasible layouts, alternating between discrete assignment and topology steps and continuous geometry optimization. However, the grid must stay coarse, or the number of candidates explodes, a familiar obstacle in discrete truss optimization [Stolpe 2016], which keeps the tractable problem sizes small. Other work restricts the design in different ways: to a low-dimensional family of geodesic domes [Huang et al. 2021], to a designer-guided graphic-statics construction [Seats et al. 2024], or to a vocabulary of reclaimed triangular modules aggregated inside a prescribed outline [Van Marcke et al. 2024].

Compared with these previous methods, we optimize the individual beams' geometry (position, orientation, and length) directly within a continuous search domain, recovering both the layout and the dimensions of the structure without committing to a ground structure, a parametric family, or a fixed connectivity graph.

Continuous structural optimization. Our approach is inspired by density-based topology optimization, where the design domain is treated as a dense grid whose cells store continuous material density and the distribution of density is optimized to achieve minimal compliance under some target load [Sigmund 2001], allowing the method to discover structures of arbitrary topology through continuous optimization. Recent work has proposed differentiable implementations of topology optimization, allowing the use of gradient descent to represent the design domain as a continuous neural network rather than a discrete grid [Doosti et al. 2021; Hoyer et al. 2019]. However, topology optimization produces organic structures that require additional processing to be converted into frame structures made of beams. Our key idea is to represent individual beams as elongated density fields, and to employ a differentiable density-based method to optimize the dimensions and position of these beams rather than the density of grid cells. Through gradient-based optimization, the beams evolve smoothly to form connected structures of minimal compliance, without being limited to a pre-defined layout. Importantly, we also introduce soft objectives to model the stock constraints within this continuous framework. Our use of gradient-based optimization also relates to the work of Keith J. Lee and Caitlin Mueller [Lee 2025] who leverage automatic differentiation to optimize beam structures using gradient descent, and optimal transport as a differentiable relaxation of the assignment problem for reuse. While their method considers more structural objectives than compliance, it takes as input a pre-defined beam layout and does not modify it during optimization.

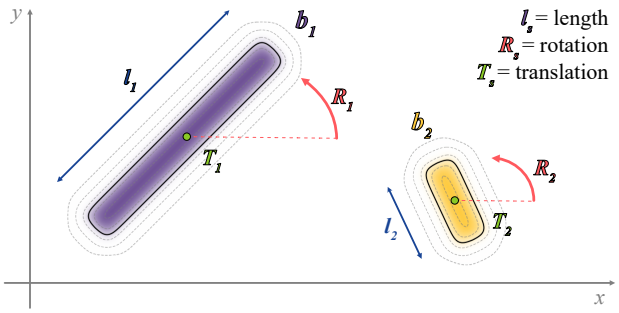


Figure 2: We treat a beam b_s as a soft box-shaped density field, parametrized by a translation T_s, rotation R_s and length l_s.

Our approach is also inspired by the work of Kodnongbua et al. [2025] who propose guidelines to design shape grammars that are well suited to gradient-based optimization, and who applied these principles to the optimization of modular displays made of a small set of binary elements [Kodnongbua et al. 2026], where they employ shape interpolation to allow the elements to morph smoothly throughout optimization. Similarly, we equip each beam in the stock with a basis function to allow the length of the beams to vary smoothly as an interpolation of the admissible stock lengths. Finally, our approach also draws inspiration from the success of 3D Gaussian Splatting in reconstructing complex 3D scenes by optimizing the placement of fuzzy primitives using gradient descent [Kerbl et al. 2023].

3 Continuous optimization for beam reuse

3.1 Problem formulation

We consider a reuse scenario where we are given a stock of reclaimed beams and we want to design a planar load-bearing frame structure by reusing some of these beams. For simplicity, we assume that all beams have the same cross-sectionSimilarly to prior work [Brütting et al. 2019; Favilli et al. 2025], we aim at reusing the beams as is to avoid cut-off waste. The optimization variables are thus the number of beams, their position in the domain, and their length subject to stock constraints.

In contrast to prior work that represents frame structures as networks of beams, we propose a continuous formulation where the structure is modeled as a set of independent beams $\{b_{s=1..S}\}$. Each beam is represented by a density field $\rho_s : \Omega \rightarrow [0, 1]$, which encodes the beams occupancy over the design domain. Specifically, a density value of 0 corresponds to empty areas while a value of 1 corresponds to areas fully covered by material. The total density field of the structure at a point p in the domain Ω is then obtained

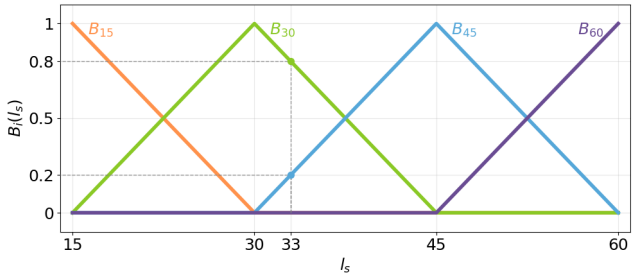


Figure 3: Interpolating basis functions B_i for a stock with admissible beam lengths $\{15, 30, 45, 60\}$. In this example, a beam length of 33 is expressed as a the linear interpolation $0.8B_{30} + 0.2B_{45}$.

by summing over of all beams:

$$\rho(p) = \sum_{i=1}^{S} \rho_s(p), \tag{1}$$

This formulation allows us to rely on existing topology optimization algorithms to compute the compliance $c(\rho, f)$ of a structure ρ under boundary conditions f [Hoyer et al. 2019]. Our goal is to optimize the position, orientation and length of all beams to minimize:

$$\mathcal{L}_{\text{compliance}} = c(\rho, f). \tag{2}$$

To avoid the trivial solution of filling the domain densely, we balance the minimization of compliance with the minimization of the total length of the structure:

$$\mathcal{L}_{\text{length}} = \sum_{s=1}^{S} l_s, \tag{3}$$

where l_s denotes the length of beam b_s.

Finally, similar to prior work on structural reuse [Brütting et al. 2020], we consider the case where beams of the stock fall into a finite set of admissible lengths $\{l_{i=1..N}\}$. The stock defines the number n_i of beams available for each length, and we denote as m_i the number of beams of length l_i used in a given structure. To be valid, a structure must not use more beams than available in the stock, i.e:

$$m_i \leq n_i, \forall i. \tag{4}$$

In the remainder of this section, we describe how we model individual beams as density fields, and how we formulate soft objective terms to model constraints on the admissible lengths of the beams and on their availability in the stock.

3.2 Modeling individual beams

We represent beams as soft box-shaped density fields parameterized by the beam translation T_s, rotation R_s, and length l_s, as shown in Figure 2. Specifically, for a given point $p(p_x, p_y)$ in the domain, we express the density of beam b_s as the product of two sigmoid functions:

$$\rho_s(p') = \sigma_w(p'_x) \cdot \sigma_{l_s}(p'_y) \cdot \left(1 - e^{-\eta_b l_s}\right), \tag{5}$$

$$\sigma_w(p'_x) = \frac{1}{1 + e^{-\eta_b \left(\frac{w}{2} - |p'_x|\right)}}, \tag{6}$$

$$\sigma_{l_s}(p'_y) = \frac{1}{1 + e^{-\eta_b \left(\frac{l_s}{2} - |p'_y|\right)}}, \tag{7}$$

where $p' = R_s^\top (p - T_s)$ expresses the point in the local coordinate frame of b_s. Intuitively, the sigmoid softens a hard-edged box with a transition band in which the density gradually fades from solid to empty. This fuzziness is critical to provide directional gradients to the optimization, effectively attracting beams to regions where compliance can decrease. The parameter η_b controls the extent of this soft transition (we use $\eta_b = 2$ in all our results, for domains of resolution 180×120).

The gating term $\left(1 - e^{-\eta_b l_s}\right)$ ensures that a beam of vanishing length contributes zero density everywhere. Allowing the length to vanish provides us with a continuous mechanism to remove unnecessary beams during optimization.

This formulation is fully differentiable with respect to the beam parameters T_s, R_s and l_s. Gradients of the compliance with respect to the beam parameters are obtained by chaining the gradients inherited from the density-based compliance solver with those computed from the differentiable density field:

$$\frac{\partial c}{\partial T_s} = \frac{\partial c}{\partial \rho}\frac{\partial \rho}{\partial T_s}, \qquad \frac{\partial c}{\partial R_s} = \frac{\partial c}{\partial \rho}\frac{\partial \rho}{\partial R_s}, \qquad \frac{\partial c}{\partial l_s} = \frac{\partial c}{\partial \rho}\frac{\partial \rho}{\partial l_s}, \tag{8}$$

enabling end-to-end gradient-based optimization of both the layout and dimensions of the structure.

3.3 Modeling an assignment of beams

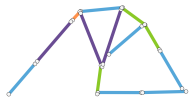
In contrast to prior work that solves beam reuse as a discrete assignment problem, our gradient-based formulation requires a continuous mechanism to enforce stock constraints. Our solution to this challenge is to relax the problem through interpolation and regularization. Specifically, we associate each admissible beam length l_i with a basis function B_i such that a beam of arbitrary length l_s can be expressed as an interpolation of the admissible beams:

$$l_s = \sum_{i=1}^{N} B_i(l_s)\, l_i. \tag{9}$$

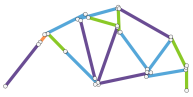
This interpolation scheme allows us to model, in a continuous manner, how much a beam b_s consumes from the stock. We complement this scheme with penalty terms to encourage l_s to snap to an admissible length, and to prevent the solution to exceed the number n_i of available beams of each length l_i, as described next.

To achieve this goal, the basis functions need to satisfy several requirements:

(1) **Interpolation.** Each basis function should peak exactly at its own stock length and should be nonzero only between the two neighboring stock lengths: $B_i(l_s) = 1$ when $l_s = l_i$, and $B_i(l_s) \neq 0$ for $l_s \in (l_{i-1}, l_{i+1})$.
(2) **Monotonicity.** Away from l_i, the basis function should decay on both sides: $B_i(l_s)$ is non-decreasing for $l_s \leq l_i$ and non-increasing for $l_s \geq l_i$, ensuring that the further a beam



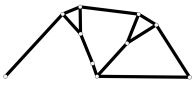
$\mathcal{L}_c = 4{,}539 \quad \mathcal{L}_l = 370$
$\Delta_l = 0.24\%$
(a) Random Initialization



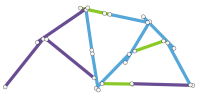
$\mathcal{L}_c = 2{,}201 \quad \mathcal{L}_l = 490$
$\Delta_l = 0.22\%$
(b) Uniform Initialization



Density Method



Skeleton Extraction



$\mathcal{L}_c = 2{,}517 \quad \mathcal{L}_l = 429$
$\Delta_l = -1.36\%$
(c) Skeleton Initialization

Figure 4: Evaluation of three initialization strategies. While using Random (a) the optimization converges to a weaker structure, both Uniform (b) and Skeleton-based (c) initializations perform comparably well.

length strays from a stock length, the smaller its contribution.

(3) **Partition of Unity.** At any given length l_s, the basis functions across all stock lengths should sum to one: $\sum_i B_i(l_s) = 1$, so that the soft assignment always forms a valid probability distribution for beam b_s regardless of where l_s falls relative to the admissible stock lengths.

(4) **Differentiability.** $B_i(l_s)$ should have a valid derivative, allowing gradients to flow during optimization.

Following these requirements, we adopt the simple linear hat basis function [Hughes 2003]:

$$B_i(l_s) = \begin{cases} 0 & l_s \le l_{i-1} \\ \dfrac{l_s - l_{i-1}}{l_i - l_{i-1}} & l_{i-1} < l_s \le l_i \\ \dfrac{l_{i+1} - l_s}{l_{i+1} - l_i} & l_i < l_s \le l_{i+1} \\ 0 & l_s > l_{i+1} \end{cases}$$

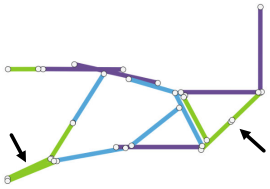
Figure 3 illustrate these basis functions on a simple example where the stock contains four admissible lengths. Intuitively, a beam of arbitrary length l_s is treated as a linear interpolation between its neighboring admissible lengths l_i and l_{i+1} in the stock.

Equipped with these basis functions, we can define a continuous penalty term on the usage of the stock. The total consumption of length l_i across all S beams of a structure is $m_i = \sum_{s=1}^{S} B_i(l_s)$. We penalize violation of the stock limit via a softplus barrier function:

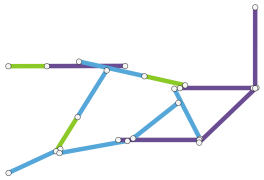
$$\mathcal{L}_{\text{stock}} = \sum_{i=1}^{N} \log\left(1 + e^{\eta_s\left(\sum_{s=1}^{S} B_i(l_s) - n_i\right)}\right). \tag{10}$$

Where η_s controls the sharpness of this barrier, we use $\eta_s = 2$ in all our results.

Our second penalty term $\mathcal{L}_{\text{rounding}}$ drives the optimized lengths toward the discrete set of admissible stock lengths. To this end,



(a) Optimization result



(b) Beam merging result

Figure 5: After optimization, the beam merging post-processing removes duplicated beams and reduces the number of joints. Two changes are highlighted: The two parallel green beams at the bottom-left corner are replaced by a blue beam. Similarly, the two green beams on the right are merged into a single purple one.

we construct a shifted set of linear basis functions, where $B'_{i,i+1}$ is supported on $[l_i, l_{i+1}]$, vanishes at all stock lengths outside this interval, and attains its maximum at the midpoint:

$$B'_{i,i+1}(l_s) = 0 \quad l_s \le l_i \text{ or } l_s \ge l_{i+1}, \qquad B'_{i,i+1}\left(\frac{l_i + l_{i+1}}{2}\right) = 1. \tag{11}$$

We then define our rounding penalty term as:

$$\mathcal{L}_{\text{rounding}}(\ell) = \sum_{i=1}^{N-1} \sum_{s=1}^{S} (B'_{i,i+1}(l_s))^2. \tag{12}$$

Finally, we prevent beams from overlapping by penalizing excessive accumulation of density via another softplus barrier:

$$\mathcal{L}_{\text{overlap}} = \sum_{p \in \Omega} \log\left(1 + e^{\eta_o\,(\rho(p) - \tau)}\right), \tag{13}$$

where $\eta_o = 5$ controls the sharpness of the penalty and $\tau = 0.74$ is a threshold on the maximum overlap allowed.

Our complete objective function is a weighted sum of all terms:

$$\mathcal{L}_{\text{total}} = w_c\,\mathcal{L}_{\text{compliance}} + w_o\,\mathcal{L}_{\text{overlap}} + w_l\,\mathcal{L}_{\text{length}} + w_s\,\mathcal{L}_{\text{stock}} + w_r\,\mathcal{L}_{\text{rounding}}. \tag{14}$$

Based on our experiments, we set high weights $w_l = 100$ and $w_s = 1000$ to the length and stock terms respectively, and lower weights $w_c = 5$, $w_o = 10$, and $w_r = 10$ to the other regularization terms. A smaller value of w_l can be used to achieve more detailed structures. A larger w_r encourages beam lengths to round almost exactly to stock lengths, but it may also trap the result in a local minimum; therefore, we suggest increasing w_r only after the optimization has converged with the initial set of weights.

In all the following experiments, we report the values of the compliance and length objective terms, denoted by $\mathcal{L}_c$ and $\mathcal{L}_l$ respectively. We also report Δ_l, defined as the maximum difference between the optimized beam length and the closest length available in the stock, expressed as a percentage of the stock length.

3.4 Initialization

The initial beam configuration can significantly influence both the final arrangement layout and its structural performance. We evaluated three different initialization strategies, with growing complexity. The simplest strategy initializes the optimization with 20

beams whose rotation, translation and length are randomly sampled. Despite a straightforward implementation, the optimization initialized with the Random schema converges to structures with relatively high compliance. We identified the main limitation of such approach to be the arbitrary spatial distribution, which could leave large portions of the computational domain uncovered, while introducing several overlaps in others.

To address this issue, we introduced an Uniform initialization, which evenly distributes vertically rotated beams across the computational domain. This ensures a better spatial coverage, while preventing the beams to overlap. The beam length is still randomly samples, thus preserving stochasticity in the method.

Finally, we considered a more refined initialization procedure, exploiting prior structural information. The Skeleton initialization extracts the medial axis from the density-based topology optimizations solution, executed under a volume fraction of 0.15. Then, such skeleton is converted into our beam representation, identifying junctions and replacing the segments with straight beams of variable length.

Figure 4 compares the effect of these initialization schemes after optimizing beams using our method. While the Random initialization converges to structurally poor layouts, both the Uniform and Skeleton initializations perform comparably well. In particular, the Uniform achieves a good structural layout while also benefiting from a simpler implementation.

3.5 Optimization Scheme

Since our problem is non-convex and ill-conditioned, we combine a preconditioned update with a nonmonotone line search, inspired by Galli et al. [2023]. More specifically, we use Adam [Kingma and Ba 2014] as a diagonal preconditioner to produce a trial step, then backtrack on this step: the step is accepted as long as the energy does not rise above the previous value by more than a small relative tolerance, and is otherwise shrunk toward the previous iterate. If no step passes within a fixed number of backtracks, we fall back to the full preconditioned step. We first run the optimization for 600 iterations with learning rate $lr = 1$, reducing it my a factor of 2 every 200 iterations and clamping it to a minimum value of 0.1. We then apply a post-processing refinement, as described in the next section.

3.6 Beam merging

While $\mathcal{L}_{\text{overlap}}$ penalizes beam overlap, the optimized layouts may still contain pairs of beams that are nearly parallel and separated only by a tiny gap. Although this configuration has a small impact on the compliance objective, it introduces an higher number of joints in the structure, thus increasing its complexity.

To address this issue, we apply a simple post-processing step after optimization. Specifically, we detect pairs of beams which are close in space. If one beam completely overlaps with the another, it is removed. On the other hand, if the beams partially overlap, the two are replaced by a single longer beam spanning the two extreme endpoints. If the resulting beam exceeds the admissible stock limits, it is repeatedly split in half until the constraint is respected again. We then run the optimization for 200 iterations, allowing the beam to adapt to the modified layout. After each refinement stage, the weight associated to the rounding objective term is multiplied by 10, to gradually encourage discrete stock assignments, while the learning rate is reduced by a factor of 2 every 200 iterations. This process is repeated until no further beam pairs can be merged or a maximum of 1200 iterations is reached.

4 Results

In this section, we first describe the experimental setup we used, followed by the analysis of the results obtained using our method. We then compare our method's performance against a baseline and the most closely related approaches. Finally, we present an ablation study of the different components of our pipeline.

Implementation Details and Experimental Setup. Our frameworks is implemented in PyTorch [Paszke et al. 2019], using the Adam optimizer for parameter updates and the Autograd module to perform the differentiable optimization. We adopted the SIMP formulation as our density-based solver [Rozvany 2000], setting the Young's modulus to 1, the Poisson's ratio to 0.3, and the penalization exponent to 3. We set all applied force magnitudes to one unit, since, under linear elasticity, uniformly scaling the applied loads scales the compliance but does not change the relative comparison between designs. We use a density filter radius of 1 for our method and 2 for the density-based baseline. In our context the beam width is fixed to 2, so we rasterize the beam layout into a density field and apply a threshold of 0.4 to obtain the final binary structure for the final evaluation. In all experiments, we used a computational domain with a resolution of (180×120) and set the maximum number of beams to 20. All experiments are conducted on a MacBook Air equipped with an Apple M4 processor. Running the optimization for 600 iterations takes around 6 minutes on this hardware.

Convergence. Figure 6 illustrates the behavior of our method on the MBB beam problem. As the optimization progresses, the initial beams evolve to form a structure that is competitive with the one produced by the classical density-based topology optimization method. Under the same boundary conditions and a volume fraction constraint of 0.1, the density-based method takes approximately 30 seconds to achieve a final compliance of 611. Our method takes around six minutes to reach a compliance of 635 while selecting 15 beams that satisfy the stock constraints. Figure 7 plots the evolution of compliance, number of beams that exceed the stock availability, and average rounding error of beam length throughout the optimization, with min and max values measured over 12 runs. Note that some beam lengths do not exactly match the available stock lengths after convergence of the continuous optimization. Nevertheless, while we tried to round the beam lengths to the closest stock lengths and optimize the structure further, we observe no noticeable change in the structure and its compliance after rounding, indicating that enforcing the discrete stock lengths strictly has little impact on the structural performance of the optimized designs.

Symmetry. Our framework can be easily adapted to enforce structural symmetry with respect to a specific axis. As illustrated in Fig. 1, the vertical red axis divides the computational domain into two halves. Under the prescribed boundary conditions, we solve the optimization problem only on one half of the domain using half of the available stock. The optimized beam layout is then mirrored

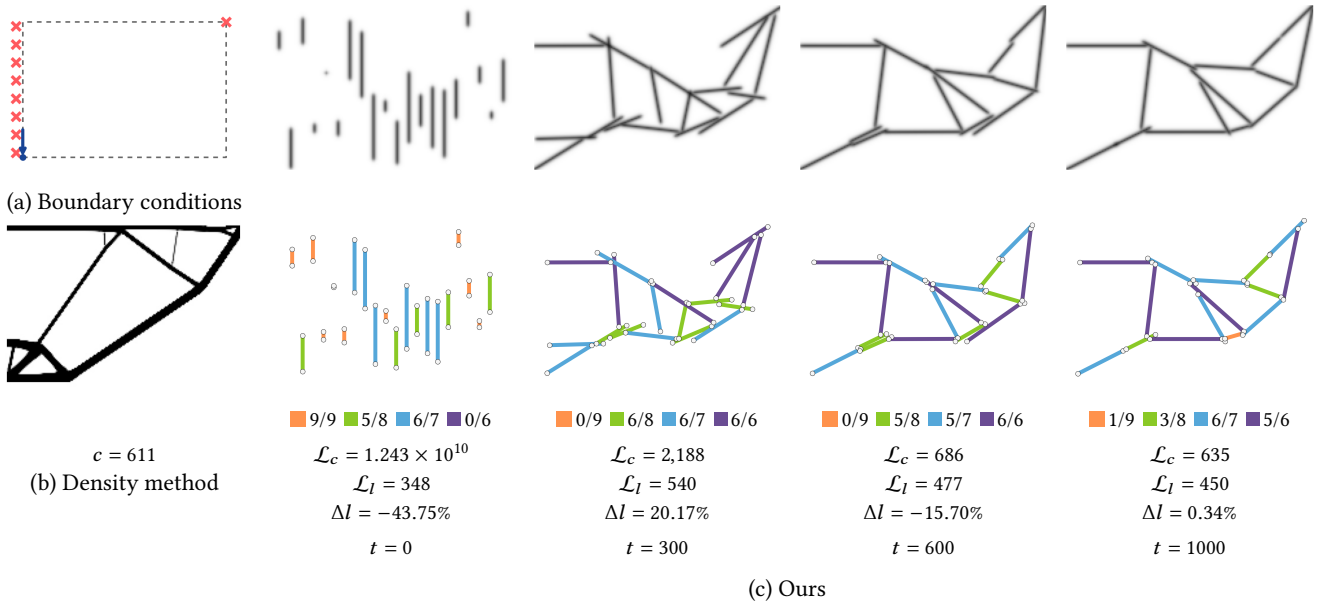


Figure 6: Under the boundary conditions shown in (a), the density method with a volume fraction of 0.1 obtains a structure with a final compliance of 611 (b). Our method (c) iteratively optimizes the beam parameters to minimize compliance and material usage while satisfying the available stock constraints. Here we visualize both the rasterized density field (c, top) and the corresponding individual beams (c, bottom). The optimization converges to a structure with a compliance of 635 using only 15 beams, without exceeding the stock limits and deviating by at most 0.34% from the stock lengths.

across the symmetry axis and the density field is reconstructed over the entire domain. To avoid duplicate beams along the symmetry axis, the beams that intersect the centerline are trimmed before mirroring.

4.1 Comparison

Although several methods have been proposed for beam reuse and discrete structural optimization, to the best of our knowledge there is no prior work with available code on frame structure optimization under stock reuse constraints. We therefore compare our method with the closest available approaches on four benchmark boundary conditions, as shown in Fig. 9. All experiments use a common 180×120 computational domain with identical boundary conditions. The stock inventory is only considered by our method and consists of beam lengths $\{15, 30, 45, 60\}$ with available quantities $\{9, 8, 7, 6\}$, respectively. We report the mean and standard deviation of the $\mathcal{L}_c$, $\mathcal{L}_l$, and Δl metrics for our method with Uniform initialization, computed over 8 experimental runs. Among the 32 experimental runs in total, 28 successfully satisfied the stock availability constraint. We first compare our optimized layouts to those obtained by the Density Method, upon which our method is built. Structures generated by the density method are characterized by complex layouts due to the independent optimization of each point in the computational domain. While our method is constrained to use the fixed geometries and availabilities in the stock, our optimized structures achieve comparable compliance and layout to those discovered by the density method.

We also compare against the results reported by [Kanno 2016] and [Fujita and Kanno 2018], which formulate the problem as a mixed integer optimization without accounting for stock constraints. To ensure a fair comparison, we reproduced the configurations that assume a constant cross section and uniform material properties. Then, we recomputed the compliance values of their structures by converting their beams into our density field representation and using our density-based solver. Our optimized structures obtain similar compliance values in most of the experiments. In example C, the Skeleton-based method obtains a better compliance than the Uniform one. However, the Uniform initialization converges to a different layout that consumes less material, as reported by the $\mathcal{L}_l$ term. More generally, our approach is not restricted to predefined layouts as the MILP ones and can discover alternative layouts that respect both the stock constraints while minimizing the compliance or material usage. This limitation of the Ground Structure Method used by the MILP baseline is evident in Figure 9D, where the grid layout does not include beams at the required angle, resulting in higher material usage than necessary to form a hat-shaped structure.

Finally, we report the results for the Δ_l metric, which measures the maximum deviation of the length of optimized beams compared to the available stock lengths. In all our results, the optimized beam length deviated from the stock length by at most 4.21% for the Skeleton initialization and 7.64% for the Uniform one. This result demonstrates that our method respects the stock constraints both in terms of lengths and stock availability.

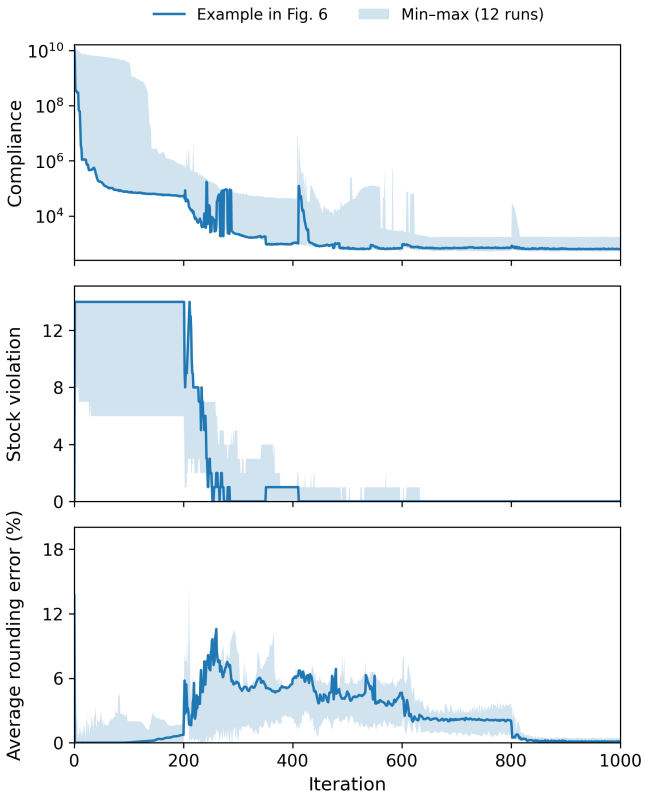


Figure 7: Convergence of our method for the same example as in Figure 6, along with min and max values measured on 12 runs of the optimization. We report the evolution of the compliance (top), number of violations of the stock availability (middle), and average rounding error (bottom). The optimization tends to first reduce compliance, and then enforces the stock limits while maintaining low rounding error.

4.2 Analysis

In this section, we further analyze the proposed method by evaluating the contribution of its individual components and optimization strategies. Unless specified otherwise, all experiments are performed using a stock composed by a set of admissible lengths $\{15, 30, 45, 60\}$ with available quantities of $\{9, 8, 7, 6\}$ respectively.

Loss function. We first investigate the contribution of each individual term in the proposed loss function through an ablation study on a representative boundary condition, as shown in Figure 8. Using $\mathcal{L}_{\text{compliance}}$ (Fig. 8a), the optimization produces a structure with low compliance that respects the imposed boundary conditions but uses an excessive amount of material, thus presenting beam extrusions and overlaps. Therefore, the $\mathcal{L}_{\text{length}}$ (Fig. 8b) term complements the initial loss function by regularizing the total beam length, reducing material usage while still converging to a structurally stable layout. However, beams tend to accumulate in localized regions, such as the bottom part of the computational domain. To mitigate this issue, $\mathcal{L}_{\text{overlap}}$ (Fig. 8c) is introduces to discourage density accumulation due to multiple beams overlap. At this stage, however, none of the optimized structures satisfy the constraints imposed by the given stock, both in terms of beams length or availability. Finally, $\mathcal{L}_{\text{stock}}$ (Fig. 8d) explicitly provides control over such constraints, thus driving the optimization towards manufacturable configurations that are consistent with the provided beam stock.

Number of beams. Under the MBB-beam boundary condition of Fig 6, we further investigate the effect of the initial number of beams on the optimization results, as shown in Figure 10. Specifically, we initialize the optimization with between 5 and 20 beams, with lengths sampled between the shortest and the longest beam in the stock. As expected, a small number of initial beams yield to relatively simple structures with higher compliance. On the other hand, increasing the initial number of beams consistently produces more complex structural layouts with lower compliance, resulting in improved structural performance.

Stock behavior. The beam lengths and their availability in the stock strongly influence in the structural layouts found through optimization. Figure 11 compares the results obtained with different stocks composed exclusively of short $\{10, 20\}$, medium $\{30, 40\}$ and long $\{50, 60\}$ beams. Stocks containing only short beams only tends to produce weaker structured while requiring most of the available material. Oppositely, stocks with medium or long beams yield simpler and yet more structurally efficient layouts. Overall, a mixed stock is a valuable trade-off by enabling good reuse of available material, while obtaining a satisfactory compliance.

Compliance vs. Length. Our method optimizes a beam structure by balancing between structural performance and material efficiency. This trade-off is controlled by the relative weights associated to the $\mathcal{L}_{compliance}$ and $\mathcal{L}_{length}$ terms in the loss function. Figure 12 illustrates how the optimization is affected while variating the two weights. Increasing the weights of the compliance term encourages the optimization to converge to more complex and yet performant structures composed of a larger number of beams. Oppositely, assigning a higher weight to the length term favors simpler structure that use less material, at the expense of an increased compliance.

5 Additional Applications

Our framework can be naturally extended to support a variety of additional applications. Thanks to its gradient-based optimization formulation, new objective terms can be easily integrated to address specific requirements. Furthermore, the method can be extended to other kinds of geometries, adopting a proper parameterization and basis function definition. In this section, we demonstrate two applications more in detail.

Image Guidance. Due to our continuous optimization based on gradient descent, our framework can easily incorporate additional objective terms that drive the optimization to meet other requirements. In Figure 13, we guide the optimization to match a given input image. Specifically, the boundary of a relevant object in the image is extracted using the open-source library rembg [Gatis 2026]. Then, the distance field of the extracted boundary is computed and incorporated into the overlap objective term as an additional guidance, encouraging the beams to align with this boundary.

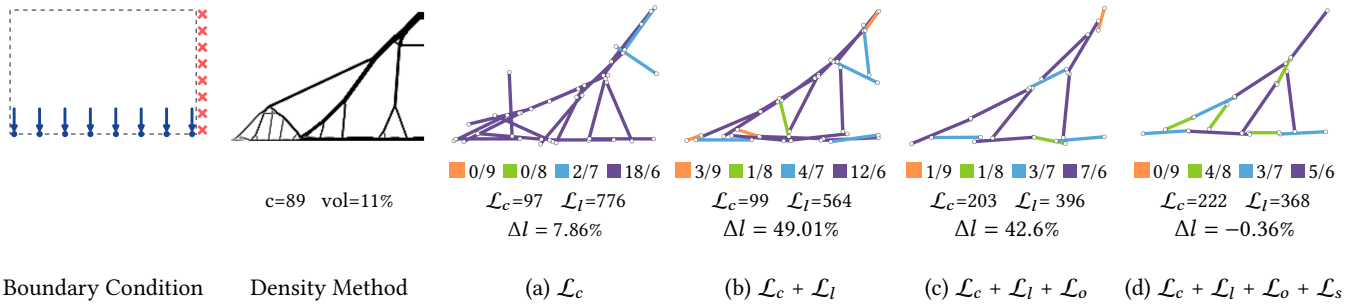


Figure 8: Contribution of the different terms in the objective function. Starting from a compliance-based optimization only (a), each additional term improves the resulting structures by reducing the amount of used material (b), preventing overlap between the beams (c) and enforcing the stock constraints (d).

Other Geometries. Although our framework is demonstrated using straight beams, it can be naturally extended to support other geometric primitives, which we illustrate with circular arcs. This extension requires defining a new family of interpolating basis function tailored to the geometry parameterization, while respecting the requirements defined in Section 3.3. Specifically, each circular arc element is parametrized by its length l and its curvature k, as shown in Figure 14a. Therefore, a two-dimensional hat basis function B_{ij} is assigned with each pair of arc length l_i and curvature k_j in the stock. Similarly to the one-dimensional family described for straight beams, this family of basis functions respects the interpolation, monotonicity and differentiability requirements, while also forming a partition of unity. Figure 14b shows the basis function B_{ij} associated with the circular arc characterized by $l = 30$ and $k = 0.06$, which peaks at the corresponding parameter pair and decreases towards each of its neighboring pairs. Finally, Figure 14c shows a structure optimized using circular arcs, demonstrating the flexibility of our framework and its potential support of a broader set of primitives for structural reuse.

6 Limitation and Future Works

A limitation of our formulation is that it does not explicitly enforce endpoint coincidence or node connectivity, and therefore adjacent beams are not guaranteed to meet at common joints. Manufacturing the resulting structures would thus require the design of custom joints, and sometimes cutting slight excess in beam length, which diminishes the material gain achieved through reuse. A related limitation is that our method produces frame structures with non-triangular polygons, which requires rigid joints compared to triangular truss structures that can be assembled with pin joints if the beam endpoints coincide.

A first option to improve connectivity would be to use our method to obtain an initial layout from which nearby beam endpoints could be detected. A subsequent optimization could then adjust the beam position and orientation to bring these endpoints as close-as-possible to form a single node, similar in spirit to the geometric optimization described in prior work [Brütting et al. 2019; Favilli et al. 2025]. But such a post-process optimization would be restricted to the layout discovered by our method, which might not be optimal after endpoint snapping. A more ambitious direction would be to incorporate additional attraction terms in our continuous formulation to encourage nearby endpoints to merge into common joints during the optimization.

Importantly, similarly to standard topology optimization [Sigmund 2001], our formulation relies solely on compliance to measure structural efficiency. Fabricating our structures in the real world would require evaluation of other aspects, such as the risk of sudden buckling, or the strength of nodal connections.

We demonstrated our continuous formulation in a simplified setting where beams can only vary in length. Yet, our experiment with circular arcs shows how our formulation can incorporate additional dimensions of the elements by using higher-dimensional interpolating functions (Figure 14), which could be used to also optimize the beam width, as well as other properties such as material stiffness. However, distinguishing between small changes in beam width might require a higher-resolution density grid. Our formulation also seeks to match the admissible dimensions as accurately as possible to avoid any cut-off waste. Alternatively, one could allow the beams to be shorter than the admissible length – but never longer – to expand the design space at the cost of material waste.

Another promising direction is to extend the proposed framework to three-dimensional domains. Such extension would enable many exciting applications, such as fabrication-aware design involving complex geometries. Moreover, given the growing interest in implicit shape representations, we believe that our framework provides a useful foundation to address a broader range of problems in the structural optimization and fabrication domains.

Lastly, our current implementation has not been optimized for computational efficiency, leaving considerable room for improvement. This limitation becomes particularly relevant when extending the framework to the 3D domain, where both the computational cost and memory consumption scale cubically with the domain resolution. Developing more efficient algorithms and implementations will therefore be an important direction for future work.

7 Conclusion

We presented a *continuous* formulation for optimizing frame structures starting from a stock of reclaimed beams. Unlike existing approaches, which model material reuse as a *discrete* assignment

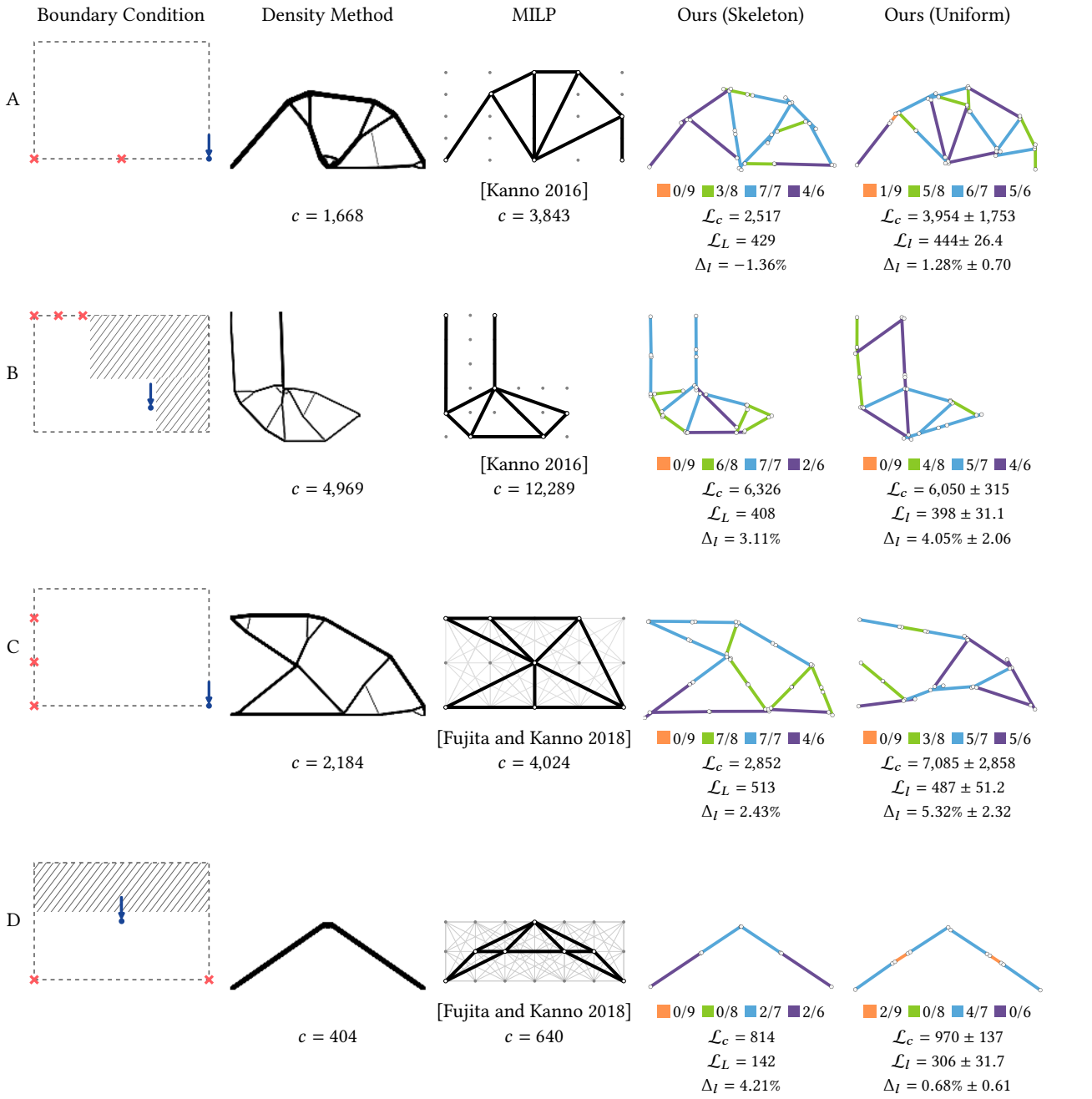
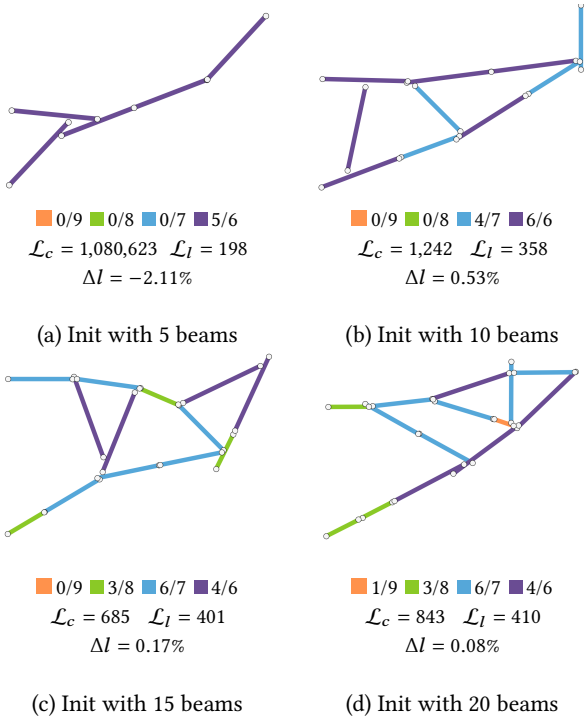


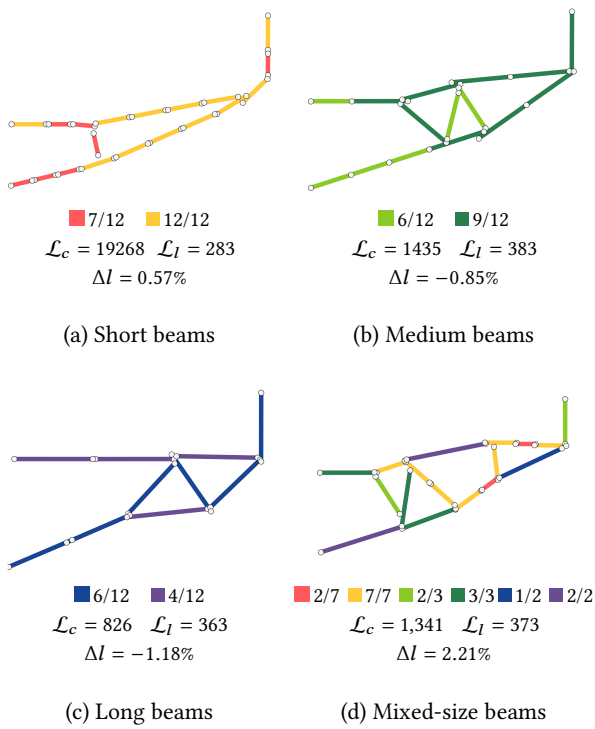
Figure 9: Comparison with prior and baseline methods on four benchmark boundary conditions. From left to right: the density-based topology optimization, the MILP approaches [Fujita and Kanno 2018; Kanno 2016] (with their ground structure grids), and our method initialized with the Skeleton and Uniform strategies. For the latter, we report the mean and standard deviation of the computed metrics over 8 experimental runs, while showing the output from one representative run. While satisfying the stock constraints, our method achieves compliance comparable to the baselines, producing layouts that are not restricted to a predefined structure.



(a) Init with 5 beams (b) Init with 10 beams

(c) Init with 15 beams (d) Init with 20 beams

Figure 10: Optimization results obtained with an initialization of 5, 10, 15 and 20 beams respectively. Increasing the number of beams allows the optimization to discover more complex structures with improved structural performance.



(a) Short beams (b) Medium beams

(c) Long beams (d) Mixed-size beams

Figure 11: Effect of different stock compositions on optimization. Stocks containing only short beams produce weaker structures while requiring majority of the available beams. In contrast, stock made of long or mixed-size beams yield more efficient structures while using less material.

problem, our method formulates it in terms of a differentiable optimization process based on gradient descent. It simultaneously optimizes the rotation, translation and length of each beam while respecting the constraints imposed by the stock through differentiable objective terms. This approach alleviates the computational complexity of previous combinatorial methods and is not tailored to respect a specific layout, thus producing structures that could not be found otherwise.

Our formulation of beams as fuzzy density fields naturally integrates with existing differentiable density-based solvers, thus allowing gradients to flow from the structural analysis to each individual beam parameters. By leveraging off-the-shelf automatic differentiation tools for gradient computations, our framework could be easily extended with additional objective terms, like enforcing connectivity properties. It could also be extended to other structural elements and potentially scale up to 3D components, by accordingly redefining their density fields and corresponding interpolating basis function while respecting the properties we identified. We believe the flexibility of our approach makes it a promising foundation for future research and application in structural reuse.

Acknowledgments

This work was supported by the ANR-NSF grant NaturalCAD (ANR-23-CE94-0003, NSF #2315354) and by the Circularity 4.0 for Aerospace funded by TU Delft France Initiative (TDFI).

References

Julian M. Allwood, Michael F. Ashby, Timothy G. Gutowski, and Ernst Worrell. 2011. Material efficiency: A white paper. *Resources, Conservation and Recycling* 55, 3 (2011), 362–381.

Felix Amtsberg, Yijiang Huang, Daniel Marshall, K Moreno Gata, and Caitlin Mueller. 2020. Structural upcycling: Matching digital and natural geometry. *Advances in Architectural Geometry* (2020).

Berend Baas, David Bommes, and Adrien Bousseau. 2025. Shape Approximation by Surface Reuse. *Computer Graphics Forum (Proc. SGP)* (2025).

Jan Brütting, Gennaro Senatore, and Corentin Fivet. 2019. Form Follows Availability Designing Structures Through Reuse. *Journal of the International Association for Shell and Spatial Structures* 60, 4 (2019), 257–265. doi:doi:10.20898/j.iass.2019.202.033

Jan Brütting, Gennaro Senatore, and Corentin Fivet. 2021. Design and fabrication of a reusable kit of parts for diverse structures. *Automation in Construction* 125 (May 2021), pp. 103614. doi:10.1016/j.autcon.2021.103614

Jan Brütting, Gennaro Senatore, Mattias Schevenels, and Corentin Fivet. 2020. Optimum design of frame structures from a stock of reclaimed elements. *Frontiers in Built Environment* 6 (2020), 57.

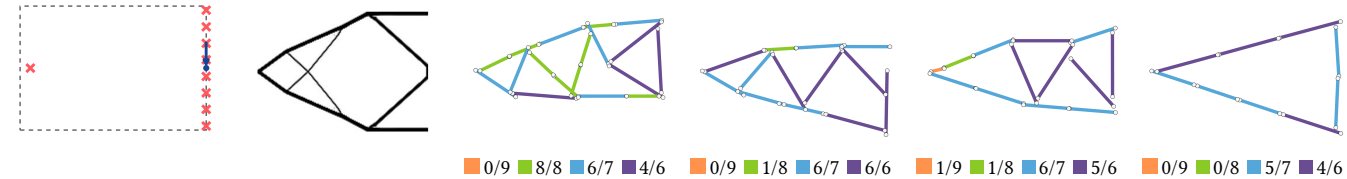
Jan Brütting, Joseph Desruelle, Gennaro Senatore, and Corentin Fivet. 2019. Design of Truss Structures Through Reuse. *Structures* 18 (April 2019). doi:10.1016/j.istruc.2018.11.006

Aurimas Bukauskas, Paul Shepherd, Pete Walker, Bhavna Sharma, and Julie Bregulla. 2017. Form-Fitting strategies for diversity-tolerant design. In *Proceedings of IASS annual symposia*, Vol. 2017. International Association for Shell and Spatial Structures (IASS), 1–10.

Rulin Chen, Pengyun Qiu, Peng Song, Bailin Deng, Ziqi Wang, and Ying He. 2023. Masonry Shell Structures with Discrete Equivalence Classes. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 42, 4 (2023). doi:10.1145/3592095

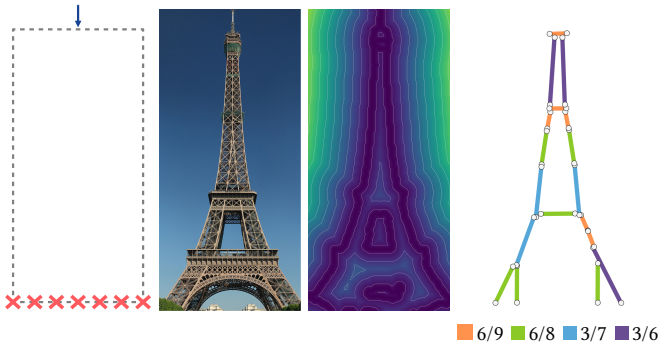
Nikan Doosti, Julian Panetta, and Vahid Babaei. 2021. Topology optimization via frequency tuning of neural design representations. In *Proceedings of the 6th Annual ACM Symposium on Computational Fabrication*. 1–9.

William S Dorn, Ralph E Gomory, and H. J. Greenberg. 1964. Automatic design of optimal structures. *Journal de Mecanique* 3 (1964), 25–52.



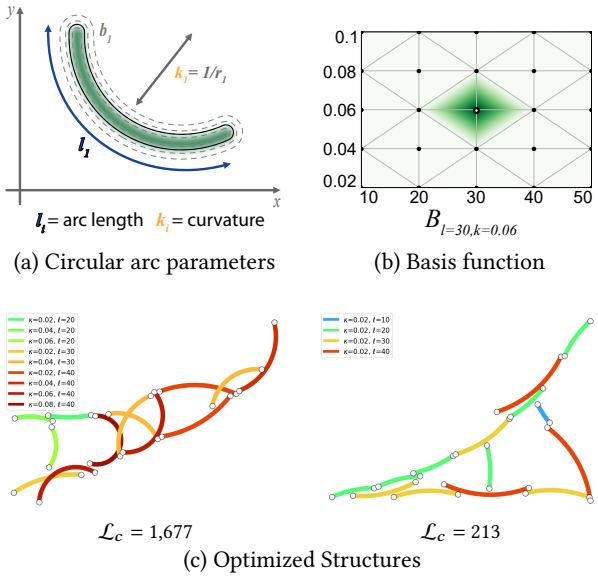
Boundary Condition | Density Method | (a) w_c/w_l = 10 | (b) w_c/w_l = 1 | (c) w_c/w_l = 0.1 | (d) w_c/w_l = 0.01

Figure 12: Effect of the optimization weights. Increasing the weight w_c of the compliance term yields more complex beam layouts, while increasing the weight w_l to the sum of lengths term results in simpler structures.



(a) Boundary Conditions | (b) Image and Guidance | (c) Optimized Structure

Figure 13: Given an image (a), we extract a soft guidance image (b) that we directly integrate in the $\mathcal{L}_{overlap}$ term of our framework to guide the optimization (c). *Photo: © Benh Lieu Song, Wikimedia Commons.*



(a) Circular arc parameters | (b) Basis function

(c) Optimized Structures

Figure 14: Given a proper parameterization (a) and interpolating basis functions (B), our framework can be extended to optimize structures with different geometries (c).

Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. 2010. Paneling architectural freeform surfaces. *ACM Transactions on Graphics* 29, 4, Article 45 (Jul. 2010), 10 pages. doi:10.1145/1778765.1778782

Andrea Favilli, Francesco Laccone, Paolo Cignoni, Luigi Malomo, and Daniela Giorgi. 2025. Optimizing Free-Form Grid Shells with Reclaimed Elements under Inventory Constraints. *Computer Graphics Forum (Proc. Eurographics)* (2025). doi:10.1111/cgf.70047

Chi-Wing Fu, Chi-Fu Lai, Ying He, and Daniel Cohen-Or. 2010. K-set tilable surfaces. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 29, 4 (2010). doi:10.1145/1778765.1778781

Shinnosuke Fujita and Yoshihiro Kanno. 2018. Global topology optimization of structural frames considering constructability. In *Proceedings of IASS Annual Symposia*, Vol. 2018. International Association for Shell and Spatial Structures (IASS), 1–8.

Leonardo Galli, Holger Rauhut, and Mark Schmidt. 2023. Don't be so monotone: Relaxing stochastic line search in over-parameterized models. *Advances in Neural Information Processing Systems* 36 (2023), 34752–34764.

Daniel Gatis. 2026. rembg: Remove Image Backgrounds. https://github.com/danielgatis/rembg. accessed 2026-07-24.

Matthew Gordon and Catherine De Wolf. 2024. Optimisation goals for efficient construction from reused materials towards a circular built environment. *Developments in the Built Environment* 19 (2024), 100509.

Mark Gorgolewski. 2018. *Implications for Design.* John Wiley & Sons, Ltd, Chapter 6, 249–275. doi:10.1002/9781118928806.ch6

Stephan Hoyer, Jascha Sohl-Dickstein, and Sam Greydanus. 2019. Neural reparameterization improves structural optimization. *arXiv preprint arXiv:1909.04240* (2019).

Yijiang Huang, Latifa Alkhayat, Catherine De Wolf, and Caitlin Mueller. 2021. Algorithmic circular design with reused structural elements: method and tool. In *fib Symposium Proceedings*, Vol. 55. International Federation for Structural Concrete, 457–468.

Thomas JR Hughes. 2003. *The finite element method: linear static and dynamic finite element analysis.* Courier Corporation.

Eleni Iacovidou and Phil Purnell. 2016. Mining the physical infrastructure: Opportunities, barriers and interventions in promoting structural components reuse. *Science of the total environment* 557 (2016), 791–807.

Jelle Joustra, Bas Flipsen, and Ruud Balkenende. 2021. Structural reuse of wind turbine blades through segmentation. *Composites Part C: Open Access* 5 (2021).

Yoshihiro Kanno. 2016. Mixed-integer second-order cone programming for global optimization of compliance of frame structure with discrete design variables. *Structural and Multidisciplinary Optimization* 54, 2 (2016), 301–316.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)* 42, 4 (2023).

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

Milin Kodnongbua, Zihan Zhang, Nicholas Sharp, and Adriana Schulz. 2025. Design for Descent: What Makes a Shape Grammar Easy to Optimize?. In *SIGGRAPH Asia Conference Papers.* doi:10.1145/3757377.3764004

Milin Kodnongbua, Zihan Zhang, Shishi Xiao, Vivian Li, Heather Robertson, Rulin Chen, David Laidlaw, and Adriana Schulz. 2026. Diceplay: A Modular Canvas for

Physical Image Composition. In *SIGGRAPH Conference Papers* (Los Angeles, CA, USA). 11 pages.

Bongjin Koo, Jean Hergel, Sylvain Lefebvre, and Niloy J Mitra. 2016. Towards zero-waste furniture design. *IEEE transactions on visualization and computer graphics* 23, 12 (2016), 2627–2640.

Maria Larsson, Hironori Yoshida, and Takeo Igarashi. 2019. Human-in-the-loop fabrication of 3D surfaces with natural tree branches. In *Proceedings of the 3rd Annual ACM Symposium on Computational Fabrication*. 1–12.

Keith Janghyun Lee. 2025. Geometric interpretations of structural demand for the analysis and reduction of design complexity. https://hdl.handle.net/1721.1/164582

Yuanpeng Liu, Ting-Uei Lee, Antiopi Koronaki, Nico Pietroni, and Yi Min Xie. 2023. Reducing the number of different nodes in space frame structures through clustering and optimization. *Engineering Structures* 284 (2023), 116016. doi:10.1016/j.engstruct.2023.116016

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*. 8024–8035.

Davide Pellis, Martin Kilian, Helmut Pottmann, and Mark Pauly. 2021. Computational design of weingarten surfaces. *ACM Transactions on Graphics* 40, 4, Article 114 (July 2021), 11 pages. doi:10.1145/3450626.3459939

Jesse Pupping, Marzia Riso, Mariana Popescu, Adrien Bousseau, and Jelle Joustra. 2025. From blades to tracks: a case study in structural reuse of curved surfaces for circular design. In *Proceedings of the ACM Symposium on Computational Fabrication*. 1–11.

Anran Qi, Nico Pietroni, Maria Korosteleva, Olga Sorkine-Hornung, and Adrien Bousseau. 2025. Rags2Riches: Computational Garment Reuse. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–11.

George Rozvany. 2000. The SIMP method in topology optimization-theoretical background, advantages and new applications. In *8th symposium on multidisciplinary analysis and optimization*. 4738.

Daniel Campbell Seats, Joshua A Schultz, and Josephine Voigt Carstensen. 2024. Automatic design generation of trusses from a reused steel stock library using graphic statics. *Journal of Building Engineering* 98 (2024), 111166.

Ole Sigmund. 2001. A 99 line topology optimization code written in Matlab. *Structural and multidisciplinary optimization* 21, 2 (2001), 120–127.

Mayank Singh and Scott Schaefer. 2010. Triangle surfaces with discrete equivalence classes. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 29, 4 (2010). doi:10.1145/1778765.1778783

Mathias Stolpe. 2016. Truss optimization with discrete design variables: a critical review. *Structural and Multidisciplinary Optimization* 53, 2 (2016), 349–374.

A. Tomczak, S. M. Haakonsen, and M. Luczkowski. 2023. Matching algorithms to assist in designing with reclaimed building elements. *Environmental Research: Infrastructure and Sustainability* 3, 3 (Sept. 2023). Publisher: IOP Publishing. doi:10.1088/2634-4505/acf341

Albertine Van Marcke, Vittoria Laghi, and Josephine Voigt Carstensen. 2024. Automated planar truss design with reclaimed partially disassembled steel truss components. *Journal of Building Engineering* 84 (May 2024). doi:10.1016/j.jobe.2024.108458

Chenming Wu, Haisen Zhao, Chandrakana Nandi, Jeffrey I Lipton, Zachary Tatlock, and Adriana Schulz. 2019. Carpentry compiler. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–14.

Tomás Zegard and Glaucio H Paulino. 2014. GRAND—Ground structure based topology optimization for arbitrary 2D domains using MATLAB. *Structural and multidisciplinary optimization* 50, 5 (2014), 861–882.

Ruowang Zhang, Stefanie Mueller, Gilbert Louis Bernstein, Adriana Schulz, and Mackenzie Leake. 2024. WasteBanned: Supporting Zero Waste Fashion Design Through Linked Edits. In *Proc. ACM Symposium on User Interface Software and Technology (UIST)*. doi:10.1145/3654777.3676395