REFASHION: Reconfigurable Garments via Modular Design

Rebecca Lin
ryelin@mit.edu
MIT CSAIL
Cambridge, MA, USA

Michal Lukáč
lukac@adobe.com
Adobe Research
San Jose, CA, USA

Mackenzie Leake
leake@adobe.com
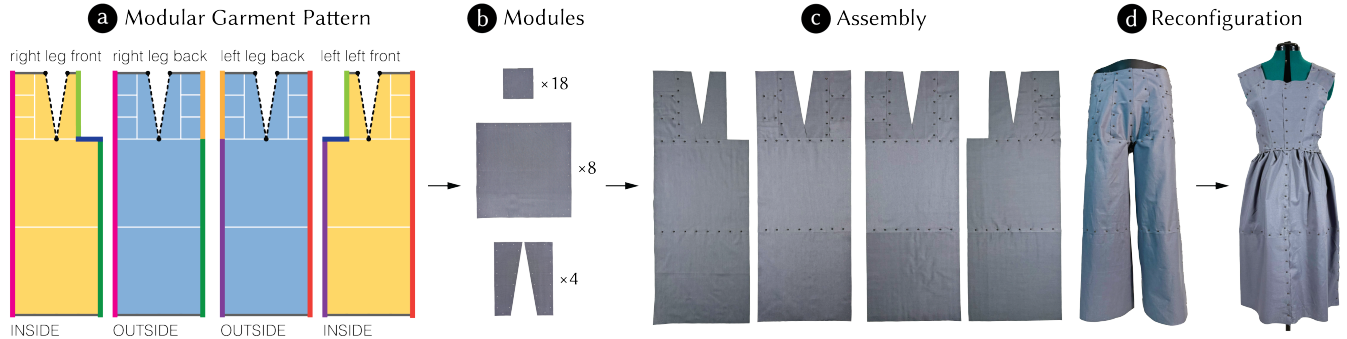Adobe Research
San Francisco, CA, USA

Figure 1: We introduce a system for designing modular garment patterns (a) that can be constructed from a shared set of fabric modules (b) into complete garments (c). These garments—such as the trousers shown—can be reconfigured to adjust fit and style, or even fully transformed into a different garment, such as the dress (d), all while reusing the same set of building blocks.

ABSTRACT

While bodies change over time and trends vary, most store-bought clothing comes in fixed sizes and styles and fails to adapt to these changes. Alterations can enable small changes to otherwise static garments, but these changes often require sewing and are non-reversible. We propose a modular approach to garment design that considers resizing, restyling, and reuse earlier in the design process. Our contributions include a compact set of modules and connectors that form the building blocks of modular garments, a method to decompose a garment into modules via integer linear programming, and a digital design tool that supports modular garment design and simulation. Our user evaluation suggests that our approach to modular design can support the creation of a wide range of garments and can help users transform them across sizes and styles while reusing the same building blocks.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

Modular design, fashion, design tools, fabrication, sustainability

1 INTRODUCTION

Most commercial garments are produced in a limited range of sizes and styles to streamline manufacturing. Individuals with sewing skills can adjust certain aspects of fit, such as shortening pant legs or reducing a skirt's volume, or perform simple repairs, such as patching a hole in jeans. However, these alterations remain difficult for the average consumer and often result in irreversible changes. For example, once pants are cut into shorts, restoring them without visible signs of alteration is nearly impossible.

While tailoring and mending play important roles in improving garment fit and lifespan, there are advantages to considering adaptability and reusability earlier in the garment design process. Rather than encouraging consumers to discard garments that no longer fit or suit their style, designing for easy resizing, restyling, and repair promotes longevity. Inspired by recent work in zero-waste garment design [59] and design for disassembly and reconfigurability in textiles and other domains [50, 53, 57], we propose a modular approach to fashion. This approach encourages the design of garment elements intended to be reconfigured across fits and styles. Modular design involves constructing systems from independent and often interchangeable *modules* connected through well-defined *interfaces*. This design paradigm has enabled scalability and reuse in many domains such as furniture design [22, 56, 58], architecture and construction [13], consumer electronics, and software. In this

paper, we consider how these principles can be applied to fashion to facilitate the creation of dynamic garment systems.

Reconfigurable and resizable garments have emerged as a growing area of interest in fashion through small-scale explorations and limited-run clothing lines. For example, Sofia Ilmonen [18] introduced expertly crafted apparel comprising uniform fabric modules connected with buttons and loops. Petit Pli [41] developed origami-inspired children's wear that grows with the child. Other commercially available reconfigurable garments feature detachable components [34], such as sleeves or hoods, or employ mix-and-match approaches, such as treating tops and bottoms as interchangeable modules [24]. However, such systems offer limited versatility and user customizability beyond simple variations of a base garment or a small set of predefined pairings.

While there are many possible interpretations of modularity in garment design, we identify three principles to characterize this design space: First, the modules should be *reusable* across a diverse range of garment designs and users. Second, the interfaces must support *reversible* yet secure connections to enable repeated assembly while ensuring wearer comfort. Lastly, both the modules and the resulting garments should be easy for users to fabricate and assemble. In response to these goals, we introduce REFASHION, a new system for modular garment design. Our contributions include:

(1) **Modules:** A compact, reusable set of modules that act as the building blocks for modular garments. Although they are standardized to reduce fabrication waste and promote easy assembly, these modules support expressive shaping techniques such as gathers, pleats, and darts (Sec. 4).

(2) **Decomposition Algorithm:** An optimization-based approach that uses integer linear programming to decompose garment patterns into a minimal set of modules for efficient assembly (Sec. 6.2).

(3) **Modular Design Tool:** A digital tool for creating, manipulating, and simulating modular garment patterns—enabling the exploration of design possibilities alongside their corresponding assembly instructions (Sec. 5).

Our results and user studies indicate that this modular approach to garment design supports the creation of many different garment types that can be resized, restyled, and repurposed from a shared set of core building blocks.

2 RELATED WORK

We build upon prior work in computational garment design, sustainable fabrication, and designing assemblies that support modularity.

2.1 Computational Garment Design Tools

Most garments are constructed by cutting, folding, and stitching 2D panels into 3D forms. Several computational tools have been developed to bridge the gap between 2D sewing patterns and 3D garment models—through simulation [7], bi-directional editing between formats [4, 5, 42, 51], and automatic pattern inference from 3D shapes [30, 40]. For garment shaping, prior work has investigated the placement of pleats [28] and darts [9], as well as the alignment of fabric prints across seams [55]. Other tools have focused on garment sizing. While commercial workflows typically rely on standardized "size grading" techniques, recent approaches explore

more personalized garment retargeting using optimization [54] and parameterized patterns for custom body measurements [8, 23]. Closest to modular garments is GarmentCode [23], which treats semantic garment components (e.g., sleeves, skirts, etc.) as swappable modules. However, such systems are typically constrained to specific garment types, making it difficult to repurpose modules across distinct categories (e.g., reusing parts of a skirt in a blouse). In contrast, we introduce a smaller set of more universal building blocks that can be combined to produce multiple types of garments.

2.2 Reducing Waste in Fabrication

Prior work in computational design and fabrication has explored a range of strategies to promote material-efficient practices. One common approach is enabling reuse of existing materials. For example, Scrappy [52] supports the reuse of leftover 3D printing materials as infill, while EcoEDA [32] facilitates the design of electronic circuits using recycled components by offering reuse suggestions and maintaining a component library. Other efforts have focused on biodegradable alternatives, such as heating pouches made from leaves [48], filament derived from used coffee grounds [44], and compostable threads for textiles [60]. Another area of focus has been integrating material awareness early in the design process. Fabricaide [45], for instance, helps users optimize material sheet usage during laser cutting. Similarly, Koo et al. [22] and Wu et al. [56] introduced tools for designing low-waste furniture layouts from flat sheets. Beyond specific tools, broader HCI research has examined themes such as reuse, salvage fabrication, and the practice of unmaking as part of sustainable design discourse [11, 19, 49].

Within the context of textiles, several systems have aimed to reduce waste and encourage reuse. Visible mending techniques have been explored as an aesthetically intentional form of repair [20], and ScrapMap [26] helps users incorporate fabric scraps into quilting projects. Knit and crochet textiles offer an inherently low-waste fabrication method, using only the necessary yarn. Recent work texTile [10] extends this efficiency by considering a modular approach to granny square garments—akin to our own—using custom connectors to support reuse and reconfiguration. In contrast, cut-and-sew garments often produce more waste due to the irregular shapes of fabric panels, which leave behind awkward scraps. InStitches [27] helps users repurpose these remnants to practice sewing techniques. The field of zero-waste fashion design [43] takes this challenge further by encouraging the creation of clothing that eliminates fabric waste entirely. WasteBanned [59] supports users in adhering to these constraints by providing tools that consider material usage throughout the design process. Our system contributes to this space by representing garments as assemblies of rectangular modules, which enable highly efficient packing on fabric sheets and minimize textile waste.

2.3 Designing Reusable Assemblies

Efficient physical assembly has been a long-standing goal in computational design, with strategies aimed at promoting material reuse, streamlining manufacturing, and reducing construction time. One foundational approach is shape decomposition—dividing a shape into meaningful subcomponents [46]. A well-known variant aimed

at maximizing reuse is geometric dissection, which seeks to transform one shape into another of equal area. Because this problem is NP-hard [6], researchers have proposed approximate methods that tolerate small discrepancies [12]. These solutions often account for practical assembly constraints, such as reversibility, by requiring connections that are secure yet non-permanent [29, 50]. We extend these ideas to garments by decomposing clothing into modular components that can be disassembled and reconfigured into new designs.

Another important design goal is minimizing part diversity to streamline fabrication. In architecture, for example, freeform surfaces are often constructed from a small number of standardized panels to reduce manufacturing complexity [13, 31]. In rapid prototyping, complex forms can be approximated using simple, reusable bricks, such as LEGO parts [21], as seen in systems like FaBrickation [36]. We apply a similar strategy to fashion: our system uses a small set of modules types that can be flexibly recombined to create a wide range of garment features and silhouettes.

Finally, fast, error-resistant construction is key to usability and accessibility. Prior work has introduced methods to reduce alignment errors in laser-cut assemblies [37] and accelerate fabrication with hinged or slot-fit joints [1, 2]. Inspired by these works, we employ bidirectional connectors that allow any aligned module edges to attach, enabling quick and simple garment construction.

3 SEWING BACKGROUND & TERMINOLOGY

Our modular garment system, REFASHION, borrows common concepts from conventional sewing. A *pattern* is a size-specific blueprint for constructing a desired garment. It comprises outlines for *panels* cut from fabric sheets and instructions for joining their edges in *seams* or leaving them as *free edges*. Each pattern also specifies a *seam allowance*—a margin between the panel edge and stitching line that provides stability.

Fabric manipulation techniques beyond basic seaming are used to alter a garment's 3D geometry (Fig. 2). *Gathers* compresses a longer edge into a shorter one through soft, evenly distributed folds to create fullness. They are commonly applied at waistlines, sleeves, cuffs, or for ruffles. *Pleats* double and secure fabric over itself in structured folds. They can span an entire panel, such as in a pleated skirt, or sit at select locations, such as along a trouser waistband. *Darts* remove triangular wedges of fabric to contour fabric for fit. They are commonly positioned at the bust, waist, or hips, with their width and height governing the resulting shaping. In practice, these operations require adjusting the original panel outlines—extending edges to accommodate future gathering, folding, or cutouts—while preserving the garment's intended finished dimensions.

4 MODULAR GARMENT DESIGN

REFASHION draws inspiration from several creative approaches to promoting garment longevity and reducing textile waste, including mending [20, 25], zero-waste clothing [43, 59], and capsule wardrobes [3, 17, 33, 38]. By assembling garments from a small set of rectangular modules, users can repair clothing by replacing only the affected parts, draft patterns with minimal fabric waste, and recombine elements to create new looks. Our *design goals* for modular garments include:
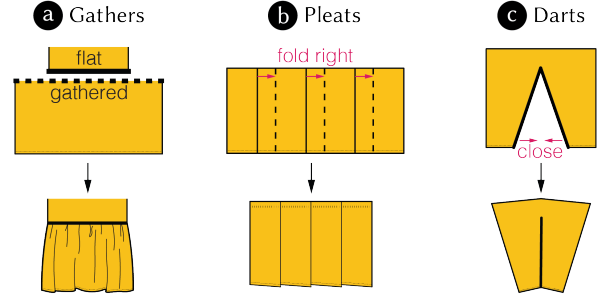

Figure 2: Three common fabric-shaping techniques: (a) gathers join a longer panel edge to a shorter one via soft, evenly spaced folds to add fullness to garments; (b) pleats double fabric back onto itself in structured and secured folds to vary garment volume and style; and (c) darts remove wedge-shaped sections of fabric to contour garments to the body.

(G1) *Versatility:* Support diverse garment shapes and styles with a small set of module types and sizes.

(G2) *Reconfigurability:* Enable easy disassembly and reconfiguration to adapt to changing bodies and contexts.

(G3) *Wearability:* Ensure garments fit properly, remain secure, and retain the appearance of conventional clothing.

Our approach to modular garment design differs from conventional sewing-based construction in two key ways: first, panels are assembled from discrete, reusable modules rather than cut from a single-use continuous sheet; second, seam edges are joined using specialized interfaces, which are carefully arranged snaps or fasteners, that enable reversible, no-sew assembly G2. The remainder of this section introduces these modules and interfaces, outlines the operations they support, and characterizes the resulting design space of modular garments.

4.1 Module Definitions

To support a diverse range of garments G1, we introduce three module types: foundation, pleat, and dart, which enable advanced shaping capabilities similar to those for conventional clothing (Sec. 3). Each module is sized as an integer multiple of a user-defined base unit, Δ, which defaults to 8 cm.

4.1.1 Foundation Modules. A *foundation module* is a square building block with dimensions $\Delta \times \Delta$ (Fig. 3) or an integer multiple thereof, $m\Delta \times m\Delta$, where $m \in \mathbb{N}$. Foundation modules alone are sufficient to construct complete garments, as demonstrated in Fig. 8.

4.1.2 Seam Interface. Each module edge has a *seam interface*: an array of double-sided snaps (inset) or fasteners (Fig. 14), centered along the edge and evenly spaced at intervals of $\Delta/(2k)$, where k is a positive integer that ensures an even number of connection points and modulates their density (e.g., $k = 1$ in Fig. 3). Similar to conventional seam allowances (Sec. 3), each module has a δ margin that creates an overlap between modules, preventing gaps when the garment is under tension.

The *discrete*, or *point-to-point*, nature of our seam interface enables two types of seams (Fig. 4). A *flat seam* joins edges of equal
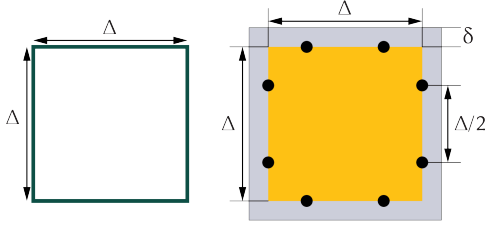
Figure 3: A $\Delta \times \Delta$ foundation module (left) and one possible implementation (right), featuring two connection points per edge and a seam allowance of δ.

length by matching fasteners one-to-one. A ***gathered seam*** joins edges in a 2:1 ratio: the longer edge is gathered so that every other fastener aligns with one on the shorter edge. A seam may be *partially* gathered if it combines both flat and gathered lengths, as seen in Fig. 11 and Fig. 18.
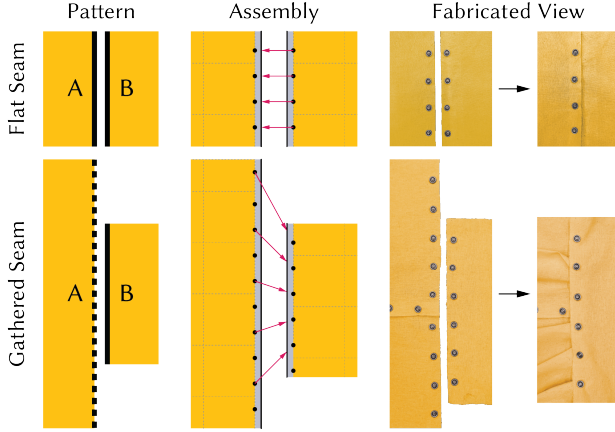

Figure 4: Our seam interface supports two types of seams. A *flat seam* joins two edges of equal length by aligning fasteners in a one-to-one correspondence. A *gathered seam* connects a longer edge (dashed) to a shorter edge by pairing two fasteners on the longer side with one on the shorter. Because the interface is bidirectional, either panel may lie on top. In the fabricated examples, panel A lies atop B for the flat seam, while B lies atop A for the gathered seam.

4.1.3 Pleat Modules. A ***pleat module*** is identical to the $\Delta \times \Delta$ foundation module except for an additional ***pleat interface*** that supports a directional fold. The interface comprises four sockets located on the front face of the module, immediately adjacent to the pins of the left and right seam interfaces (Fig. 5). Fig. 6 illustrates pleating effects produced by different arrangements of pleat modules.

4.1.4 Dart Modules. A ***dart module*** comprises two mirrored right-angle trapezoids whose slanted legs form a triangular gap; this gap is folded closed during assembly, creating 3D shaping. Each trapezoid is parameterized by a height parameter $h = n\Delta$, where $n \in \mathbb{N}$, a
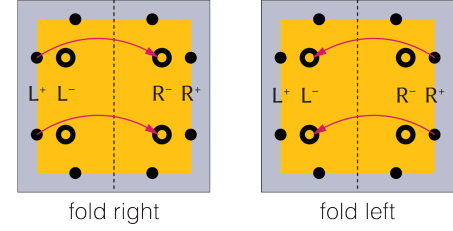

Figure 5: A pleat module can be folded in two directions by connecting seam interface pins to pleat interface sockets: (1) right by connecting left pins to right sockets ($L^+ \rightarrow R^-$), or (2) left by connecting right pins to left sockets ($R^+ \rightarrow L^-$).
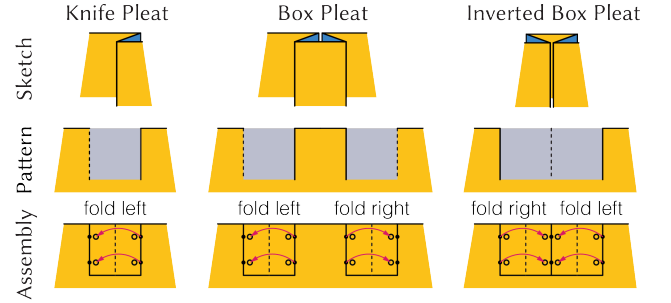

Figure 6: Arrangements of pleat modules to achieve knife, box, and inverted box pleats.

wide base of length Δ, and a narrow base of length $\Delta - \frac{w}{2}$ for some width parameter w, subject to $w \leq 2\Delta$. Once assembled, the dart module's narrow end measures $2\Delta - w$ (Fig. 7a). If $w = \Delta$, this length is Δ and is therefore compatible with the Δ-aligned dimensions of foundation and pleat modules, allowing it to participate freely in seams. We call this the ***universal*** case. Otherwise, for any other value of w, the module can either form a triangle dart along a free edge or pair with another dart module of the same width to create a diamond-shaped dart. Attaching a non-universal dart module to a Δ-aligned edge would result in a seam mismatch of length $\pm w$ and is therefore not permitted.
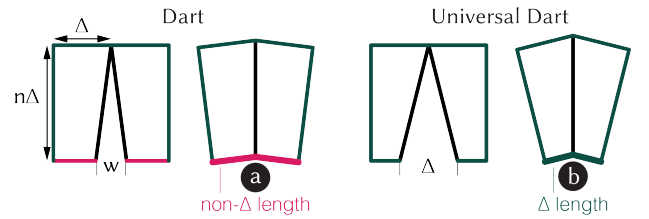

Figure 7: Left: A dart module with width w and height $n\Delta = 2\Delta$. Right: A *universal* dart module ($w = \Delta$) ensures that all assembled edge lengths remain integer multiples of Δ; edge (b), unlike (a), has length Δ, and so it may be part of a seam.
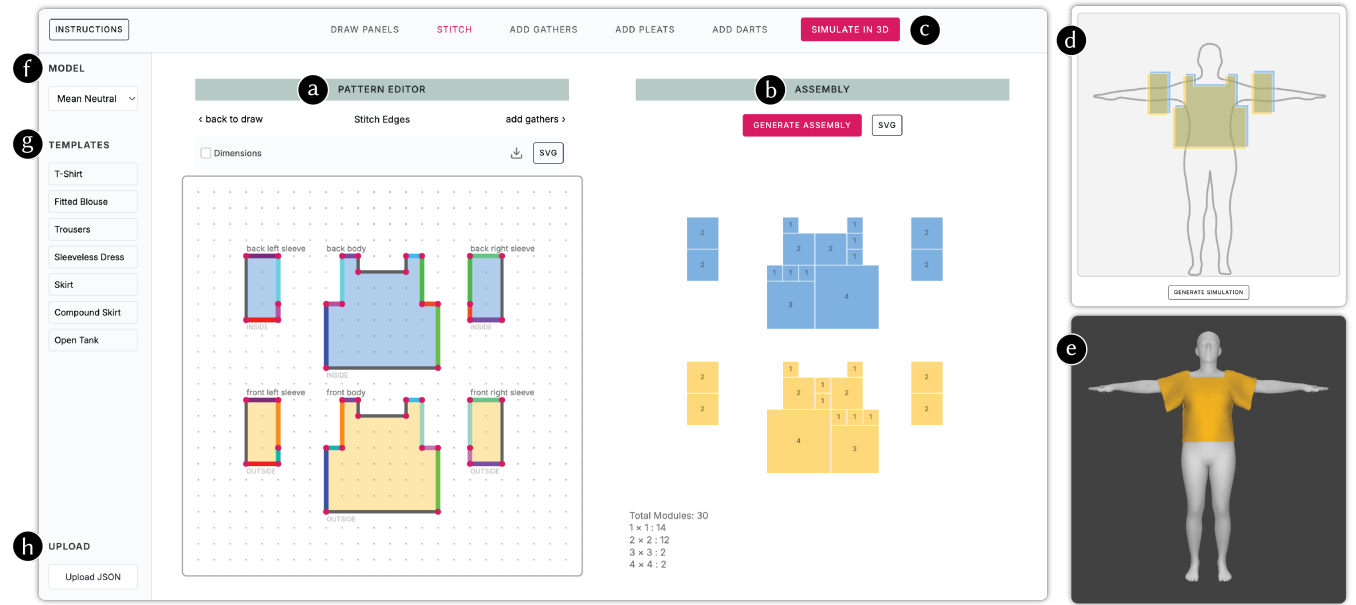
Figure 8: The Refashion interface consists of three main views: (a) the *Pattern Editor* for drawing and stitching garment panels and adding shaping features; (b) the *Assembly View* for decomposing the pattern into a modular assembly; and (c) the *Simulation View* for aligning (d) and previewing (e) the garment on a selected body model (f). Users can begin their design process by drawing a pattern from scratch (a), selecting from existing templates (g), or uploading a previously created pattern (h).

4.2 Modular Patterns

Similar to traditional garments, our modular garments are built by joining collections of 2D panels. However, in our approach, each panel is first assembled from a set of foundation, pleat, and dart modules. To ensure that a design can be realized using only these modules, we enforce two constraints on any **modular pattern**:

(C1) Panels, including any pleats or darts, must align with a regular grid of cell size Δ.

(C2) Seams may only join module edges whose lengths differ by at most a factor of two (Sec. 4.1.2).

An **assembly** is a decomposition of a modular pattern into modules (e.g., Fig. 8b) from which the pattern can be constructed. An assembly is considered efficient when it uses as few modules as possible, serving as a proxy for assembly effort. In other words, our goal is to divide the garment panels that comprise the pattern into grid-aligned regions, possibly with holes, with the fewest available modules. We formalize and solve this problem in Sec. 6.2.

5 REFASHION UI OVERVIEW

We present Refashion Studio, a digital tool for designing modular garment patterns. The UI comprises three main views: the *Pattern Editor View* (Fig. 8a) for creating and editing patterns, the *Assembly View* (Fig. 8b) for generating pattern assembly instructions, and the *Simulation View* (Fig. 8c) for previewing garments in 3D. The tool assumes no prior pattern-making experience and includes comprehensive user guides with video walkthroughs for pattern creation and shaping. Fig. 10 illustrates the step-by-step creation

of a compound skirt. In Phase 1, the base pattern is drawn and stitched. In Phase 2, gathers, pleats, and darts are applied to shape the garment (see App. A for full feature manipulation details).

5.1 Pattern Editor View

Users begin their design process by either selecting a pattern from the tool's built-in library (Fig. 8g), uploading a previously saved pattern (Fig. 8h), or creating a new pattern from scratch (Fig. 8a).

When creating a new pattern, users first draw garment panels as non-self-intersecting loops of horizontal and vertical segments aligned to a Δ-unit grid C1. The system automatically validates each panel's geometry. Users can rename panels, drag them into place, or flip their fabric orientation. These basic properties are locked once the user enters the stitching phase, where they select pairs of panel edges with matching flat lengths to form seams. If a single edge must be divided—for example, to participate in multiple seams or to leave part of it free—users can insert break points to split it into segments.

After creating a base pattern (Phase 1), users can add shaping features in the order of gathers, then pleats, and finally darts (Phase 2). This orders the features from the most to the least disruptive to a panel's dimensions: a gather doubles an edge's length, a pleat may introduce and then fold away a single unit of length, and a dart alters only local geometries.

5.1.1 Gathers. To gather an edge in a seam (Fig. 4), the system doubles its length (Fig. 9a). Gathering may affect adjacent seams; Refashion Studio automatically propagates these adjustments or,
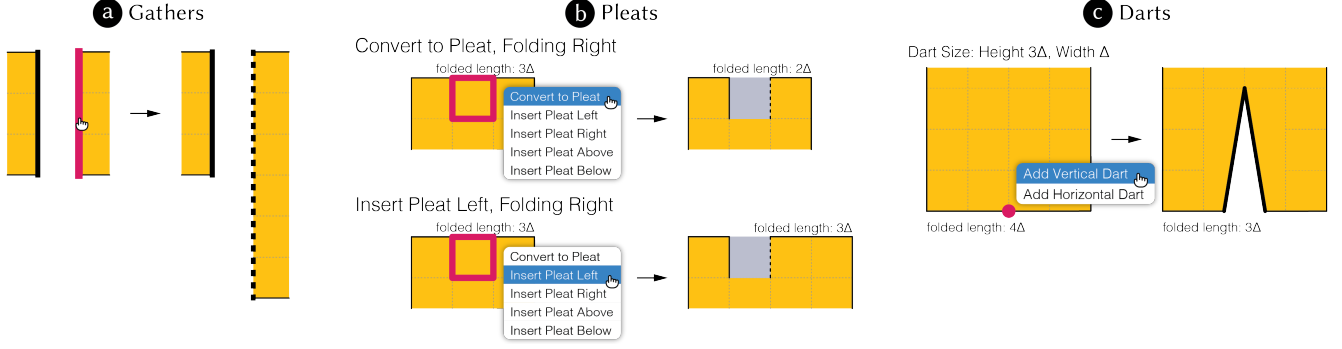
Figure 9: In the Pattern Editor View, (a) gathering an edge at a seam doubles its length. (b) Converting an existing unit into a pleat module reduces the edge's folded length, whereas inserting a new pleat preserves the folded length but increases the flat length. (c) Adding a dart replaces the local geometry with the specified dart module. More complex examples with cascading effects on seam relationships appear in App. A.

if a gather is not feasible at that location, alerts the user to the conflict (App. A.1).

5.1.2   Pleats. Users have two options when adding pleats: they can either convert an existing panel unit into a pleat, which reduces the panel edge's folded length, or insert a new pleat unit to maintain the current folded length (Fig. 9b). Using the menu above the pattern editor, users specify the pleat's fold direction (up, down, left, or right). Like gathers, pleats may require adjustments to existing seams (App. A.2). Refashion Studio validates pleat placement and alerts users if a location cannot support the desired pleat.

5.1.3   Darts. To create a dart, users click any grid point within a panel and choose between a horizontal or vertical orientation (Fig. 9c). A dropdown menu above the editor allows users to specify the dart's height and width. Refashion Studio validates the placement and adds the dart if possible (App. A.3), as shown in Fig. 9c and Fig. 10, or explains why the placement is invalid.

5.2   Assembly View

If the user has specified a set of available input modules, the Assembly View automatically decomposes the design into a pattern that only uses these modules. If the user does not specify existing modules, the system assumes the user will cut the required modules. Given a decomposition, users can review the suggested modules and download an SVG cutting guide to make what is needed.

5.3   Simulation View

The Simulation View enables interactive 3D visualization of garment designs (Fig. 8c). Refashion Studio provides a library of body models across multiple sizes [39] (Fig. 8f) and allows users to import custom T-pose OBJ meshes for personalized fitting. To initiate a simulation, users drag and align each pattern panel to the model's silhouette (Fig. 8d). Once the simulation is running, users can rotate and zoom the 3D scene to inspect the garment from any view.

6   IMPLEMENTATION

In this section, we discuss the underlying implementation for representing and decomposing modular sewing patterns. First, we present a grid-based pattern representation that supports modular pattern design under strict panel and seam constraints (Sec. 4.2). Next, we discuss our approach to generating efficient modular garment assemblies by formulating the problem as an integer linear program. For details on garment simulation, please refer to App. B.

6.1   Pattern Creation

A pattern is shown to the user at two levels of granularity: a coarse level for creating a base pattern (Phase 1) and a finer-grained level for localized feature (e.g., pleat and dart) manipulation (Phase 2). In Phase 1, each panel is defined by a list of edges and a binary orientation for the face of the module visible in the pattern. When users begin feature edits in Phase 2 (Fig. 10), we switch to a Δ-grid view to enforce the two key constraints from Sec. 4.2: panels must be Δ-grid-aligned polygons C1, and seams may differ in length by at most a factor of two C2. Each panel is decomposed into a collection of Δ-sized square units, and its edges into Δ-length segments, each tracking local connectivity and, if relevant, seam pairings (e.g., dotted lines in Fig. 11).

Panel Edits via Strip Operations. Working with gathers and pleats often requires modifying panel geometry to accommodate the additional material (Fig. 9). To preserve orthogonality and Δ-aligned dimensions, we perform these operations using ***strips***—maximal sets of connected squares along a single row or column.

When a panel is expanded or contracted through the addition or removal of gathers or pleats, we insert or delete entire strips, ensuring that all edges remain axis-aligned and multiples of Δ. For example, inserting a pleat duplicates the strip containing the selected square and places the copy adjacent to the original (Fig. 11). Similarly, gathering an edge duplicates all incident strips and so doubles its length.

The strip insertion process begins with a flood-fill operation from the strip's boundary to identify affected squares. We then shift these squares by one Δ unit, creating space for the duplicated strip, and update adjacency pointers to preserve the panel's topology. This localized approach prevents disconnections that might occur with full-column shifts, particularly in panels with cavities. While strip

Figure 10: To draw a two-layer compound skirt in Refashion Studio, the user starts by drawing and stitching panels to form the base pattern as shown in Phase 1 (Steps 1-2). In Phase 2, they manipulate its shape by adding features. First (Step 3), they gather edges (a) and (b) to add volume to the second layer of the skirt. Next (Step 4), they convert every other unit along its waistline (c) into right-folding pleats to cinch the waist. Finally (Step 5), they add darts (d) and (e), which align across the side seams, to refine the waist-to-hip shaping. For each step, we display both the resulting pattern (top) and its corresponding assembly (bottom). The simulation is in the upper left.
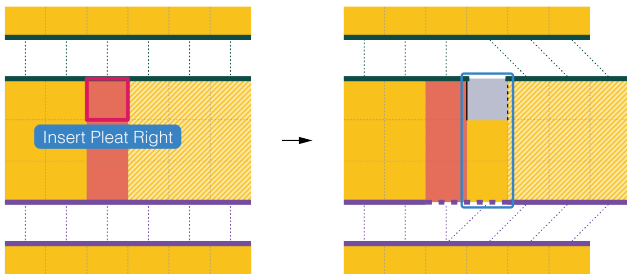

Figure 11: Inserting a pleat (gray square) to the right of a selected square (red outline) updates the panel geometry. Left: The system highlights the strip containing the selected square. Right: Adjacent squares (shaded) shift to make room for a duplicate strip (outlined). This insertion triggers seam rematching and introduces a gather (dashed) along the bottom seam to accommodate the added length.

deletion follows a similar process in reverse, we prohibit deletions that would disrupt existing seams or features.

Because inserting or deleting strips changes edge lengths, any affected seams must be rematched. If a proposed strip operation would violate seam matching constraints—for example, by attempting to fold away a portion of an edge whose match is already fully gathered—the operation is prohibited unless additional edits first resolve the conflict. Please refer to App. A for a more detailed discussion of these operations.

Seam Re-Matching Upon Panel Edits. At the start of feature edits, we mark each edge segment in a seam as active and establish a one-to-one, *planar* (non-crossing) correspondence between opposing segments—ensuring that the seam does not "twist" in our representation. These matchings evolve as the user makes edits: an active segment may gain a second match through gathering, lose all matches by becoming inactive in a fold, or retain a single match. If every segment of an edge already has two matches, we prohibit any feature operation that would shorten it or lengthen its counterpart. More generally, feature operations are allowed only if they preserve one or two matches per active segment across all affected edges. When updates are required, we adjust matchings as locally as possible while preserving planarity. The following algorithm enforces these constraints.

If inserting a new segment along an edge (e.g., via strip insertion; Fig. 11) results in an equal number of active segments on both edges, we establish new one-to-one correspondences directly. Otherwise, we create a sequence of slots—up to two per active segment—along

the edge and shift existing matches within these slots, based on the insertion location and seam orientation, to accommodate the new segment. These shifts may cascade to preserve matching invariants. Similarly, when deactivating a segment during folding or dart insertion, we redistribute its matches to neighboring segments, which can also trigger cascading adjustments (Fig. 12).
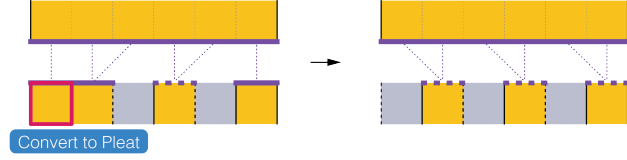

Figure 12: Pleating along a partially gathered seam may trigger cascading rematches to preserve seam constraints.

6.2 Assembly Formulation

To construct a modular pattern, we select a minimal set of modules that assemble into the desired layout. Once the user has set the pleat and dart locations, the remaining task is to fill the pattern with foundation modules. Our seam interface (Sec. 4.1.2) is designed to be fully *bidirectional*, allowing any two module edges to connect. As a result, all foundation modules of the same size are *interchangeable*. This reduces the task to square covering, which we formalize below as an integer linear program (ILP), using $\Delta = 1$ for illustration.

Problem Formulation. Let $\mathcal{S}$ be a set of foundation modules: squares of specified positive integer side lengths, and let $\mathcal{P} = \{P_1, \ldots, P_K\}$ denote the remaining garment pattern as a collection of grid-aligned polygons, possibly with holes. The objective is to cover $\mathcal{P}$ with a non-overlapping subset $U \subseteq \mathcal{S}$ that minimizes $|U|$.

This partitioning problem is generally intractable; even packing a maximal number of 2×2 arrays into a rectilinear grid is NP-complete [15, 16]. However, we formulate our task as an ILP and solve practical instances both optimally and efficiently. We empirically validate this performance across a range of pattern sizes and shapes in App. C.

ILP Formulation. Let n_ℓ denote the number of available squares of side length ℓ in the collection $\mathcal{S}$. A *candidate* for polygon P_k is specified by a side length ℓ (with $n_\ell > 0$) and a lower-left coordinate $(i, j) \in \mathbb{Z}^2$. The corresponding square is defined as

$$S(\ell, i, j) = [i, i + \ell] \times [j, j + \ell].$$

It must lie fully in P_k. Let C_k denote the set of all such candidates.

We introduce a binary variable per candidate $(\ell, i, j) \in C_k$:

$$x_{\ell,i,j}^{(k)} = \begin{cases} 1, & \text{if } S(\ell, i, j) \text{ is selected for } P_k, \\ 0, & \text{otherwise.} \end{cases}$$

These variables are defined by side length ℓ, not individual squares, to reduce variable count and improve ILP efficiency.

Let G_k be the set of unit grid cells contained in P_k. For each cell $(u, v) \in G_k$ and candidate $(\ell, i, j) \in C_k$, define a binary variable

$$A_{\ell,i,j}^{(u,v)} = \begin{cases} 1, & \text{if } [u, u+1] \times [v, v+1] \subseteq S(\ell, i, j), \\ 0, & \text{otherwise.} \end{cases}$$

This variable encodes whether the unit square in P_k with bottom-left corner at (u, v) is covered by candidate (ℓ, i, j).

The ILP formulation is as follows:

$$\min \quad \sum_{k=1}^{K} \sum_{(\ell,i,j) \in C_k} x_{\ell,i,j}^{(k)}$$

$$\text{subject to} \quad \sum_{(\ell,i,j) \in C_k} A_{\ell,i,j}^{(u,v)} x_{\ell,i,j}^{(k)} = 1, \quad \forall (u, v) \in G_k, \ k = 1, \ldots, K,$$

$$\sum_{k=1}^{K} \sum_{\substack{(i,j) \\ (\ell,i,j) \in C_k}} x_{\ell,i,j}^{(k)} \leq n_\ell, \quad \forall \ell \text{ with } n_\ell > 0,$$

$$x_{\ell,i,j}^{(k)} \in \{0, 1\}, \quad \forall k = 1, \ldots, K, \ (\ell, i, j) \in C_k.$$

The objective to be minimized is the number of squares used in the covering. The first set of constraints ensures that every unit cell is covered exactly once. The second set enforces that the number of selected squares of each side length does not exceed the available supply. In this formulation, if $x_{\ell,i,j}^{(k)} = 1$, then $S(\ell, i, j)$ is selected for P_k; that is, a square of side length ℓ is placed in P_k with its lower-left corner at (i, j).

We solve the ILP using PuLP [35] with the CBC solver [14]. Most instances solve in under a second on a MacBook Air (M2, 2023) with 8 GB of RAM. A detailed evaluation is provided in App. C.

7 RESULTS

With REFASHION we designed and fabricated a variety of modular garments to demonstrate a range of potential use cases (Sec. 7.2). We also invited additional designers to create their own modular garments (Sec. 7.3).

7.1 Module Fabrication

In developing our modular garments, we explored a range of different textiles and connection mechanisms. For clarity, we use "fabrication" to refer to cutting modules and installing their interfaces, and "assembly" to refer to joining modules into panels and garments. All modules can be fabricated without sewing.

7.1.1 Textiles. We used cotton muslin (136 gsm), an inexpensive prototyping fabric, for most of our modules. Fig. 1 features Kona cotton (147 gsm), a higher-quality, more tightly woven option. Figs. 17 and 18 employ much heavier, thicker, and more varied materials: denim from donated jeans and hand-crocheted granny squares.

7.1.2 Cutting. Unless otherwise stated, all modules adhere to a base unit size of $\Delta = 8$ cm with a seam allowance of $\delta = 1$ cm. Foundation modules range from 1Δ to 4Δ in size, while dart modules are fixed at Δ in width and either 2Δ or 3Δ in height. We found that this limited set is sufficient to efficiently form a variety of garments with adequate shaping, even across different sizes.

Modules can be cut using scissors or rotary tools, but for speed and precision, we primarily used a standard laser cutter. The regular geometries of the modules streamline cutting and reduce material waste. Foundation and pleat modules are square, while dart modules consist of trapezoid pairs that pack efficiently as rectangles. As a result, modules can be laid out and cut from fabric sheets with minimal (and sometimes even no) waste (see App. D).

7.1.3 Seam Interface. We explored two implementations of our seam interface: double-sided connectors and fasteners, which have trade-offs across fabrication time, assembly ease, and wearability.
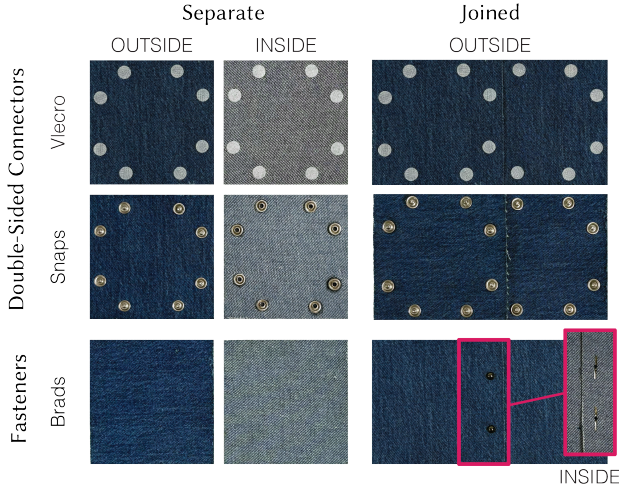

Figure 14: Fabricated examples of two seam interface implementations: double-sided connectors using Velcro dots and snap buttons (top), and symmetric fasteners—brads (bottom).

Double-sided Connectors. We tested two connector types (both 9 mm in diameter): adhesive Velcro dots for rapid prototyping and metal snap buttons for added durability. Velcro dots are applied with hooks facing outward and loops inward (~5 sec per installation), and snap buttons are installed by stamping pins onto the module exterior and hot-gluing sockets inside (~1 min per installation)

(Fig. 14). While snaps offer stronger tear resistance, they add bulk; joined seams stack four snaps, creating thick, heavy connections.

Fasteners. We tested brads (0.6 cm), which are inserted through module edges and bent open to secure the layers during assembly (Fig. 14). Although assembly takes two to three times longer than with snaps, the resulting garments are lighter, with more secure and less visible connectors (Figs. 1, 18–17). This method also allows modules to be flipped, as it does not distinguish between inside and outside surfaces, enabling more flexible or even reversible designs.

7.1.4 Prototype Modules. We built both lightweight prototypes and more durable garments made from heavier weight materials. The prototypes (e.g., Fig. 19) are assembled from inexpensive muslin modules using Velcro, applied at minimal density—two dots per Δ-length edge. Although the prototypes may not be suitable for everyday wear, they can serve as valuable, low-cost building blocks for iterating toward higher-quality garments—modular or otherwise (Sec. 7.4.2). The more polished constructions (e.g., Fig. 1) use higher-quality fabrics with fasteners applied at greater density.

7.2 Applications of Modular Garments

Refashion leverages key advantages of modular design, including ease of resizing, restyling, and repair. We demonstrate how these adjustments can be efficiently supported through our approach.

7.2.1 Resizing. Users can resize a garment by redrawing its pattern to match new target dimensions (Fig. 13) which would then be assembled from a new combination of modules. Fine-tuned fit adjustments can be made by adding gathers, pleats, and darts (Fig. 10).

7.2.2 Interchanging Parts. Our modular system allows users to exchange garment panels or parts within or across designs. For example, panels in maternity wear can be swapped for larger ones
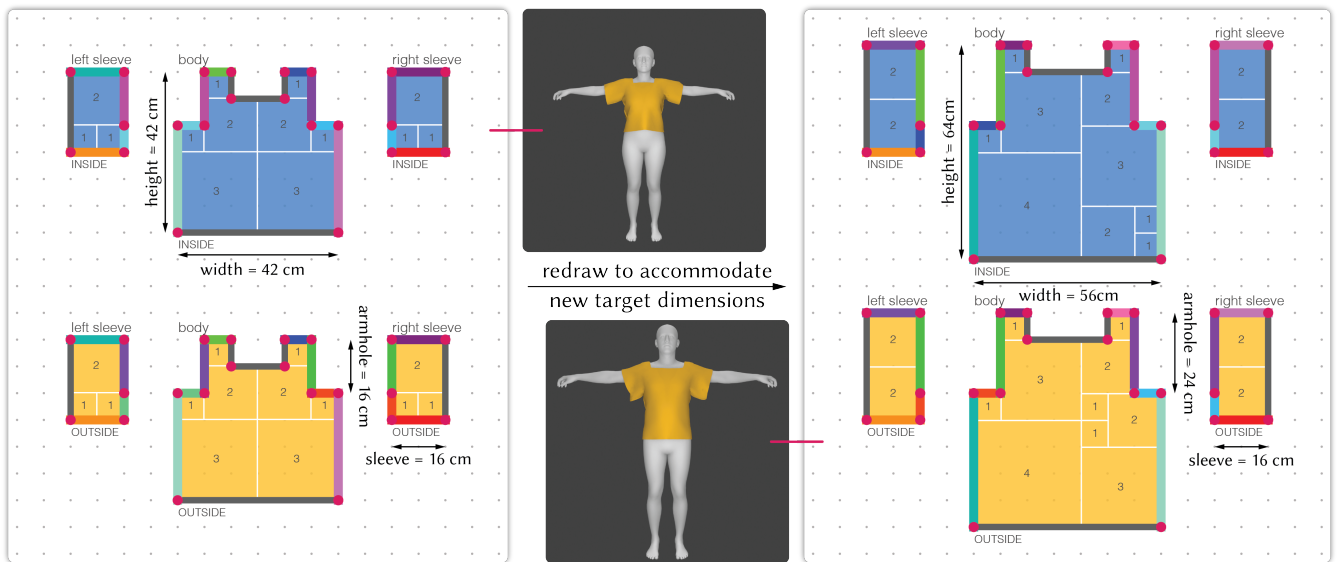

Figure 13: Given a fixed module set with base unit size Δ, resizing an existing garment pattern (left) involves redrawing it to match new target dimensions (right), rounded to increments of Δ.
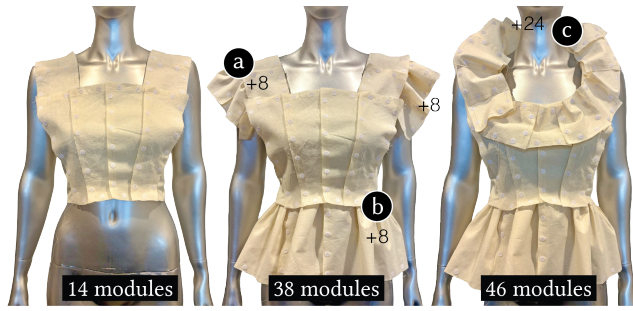
Figure 15: Restyling a fitted, sleeveless top (a) in multiple ways: first by adding short ruffled sleeves (b) and a peplum (c), then reconfiguring the sleeves into a gathered collar.

to accommodate a growing body without requiring an entirely new garment (Fig. 19-P2), and sleeves from different tops can be mixed and matched—expanding styling possibilities through combinations, even with a fixed set of components.

7.2.3 Style Adjustments. Users can easily remix a base garment by adding components such as stylized sleeves, collars, or cuffs. Fig. 15 shows a fitted, sleeveless top first augmented with short ruffled sleeves and a peplum, then alternatively styled with a gathered collar—each variation adding distinctive flair to the base design. Users may also extend a garment. The trousers and dress in Fig. 1 are complete and wearable at multiple stages of construction (e.g., they can be worn as shorts or a top before additional modules are added). This extensibility also enables detachability, useful in everyday contexts, such as shortening sleeves on a hot day.

7.2.4 Combining Garments. Users can combine garments to create new designs. For example, the top from Fig. 15 can be merged with a pleated skirt to form the dress shown in the inset, or with pants for a jumpsuit.

7.2.5 Transforming Between Garments. Our system supports both local and global transformations. Local transformations involve rearranging components of a garment or modules within a single component (as shown in the straps Fig. 16). Global transformations involve reconfiguring the entire garment, such as converting trousers into a dress (Fig. 1).
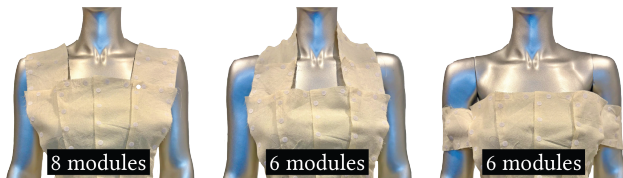

Figure 16: Transforming a square neckline (left) into a halter-neck style (middle) and an off-the-shoulder look (right) by shortening the straps and adjusting their position. The module numbers specify only the ones needed for the straps.

7.2.6 Textile Reuse. By supporting modules made from diverse materials, REFASHION enables the creative reuse of existing textiles. The sleeveless jacket in Fig. 17 is constructed from recycled jeans, while the backless halter dress in Fig. 18 incorporates hand-crocheted granny squares to add distinctive styling to different parts of the garment.

Figure 17: A sleeveless denim jacket is made from modules cut from donated jeans, preserving original seams and pockets. The model wore it for 10 hours during a workday.

Figure 18: A backless halter-neck top is constructed from denim and crocheted granny square modules. A larger base unit size ($\Delta = 11$ cm) is used so the fabric modules match the dimensions of the existing granny squares.

7.3 User-Created Designs

In our user evaluation, we invited participants to use REFASHION Studio to resize, restyle, and design their own modular garments.

7.3.1 Participants. We recruited six participants (mean age 28) from makerspaces at a local university. All had prior sewing experience, including making alterations and sewing garments from existing patterns. Three participants (P2, P4, and P6) had experience designing their own patterns. All had worked with CAD software

(e.g., SolidWorks), and five (all except P5) were proficient in SVG drawing tools (e.g., Illustrator). P2 and P6 had decades of design experience from architectural training and managing makerspaces.

7.3.2 Study Procedure. Each session lasted approximately two hours and involved three main tasks using REFASHION Studio:

(1) **Resize a T-shirt (15 min):** Participants resized a given T-shirt pattern to fit a different body model as in Fig. 13.
(2) **Add a Shaping Feature (15 min):** Participants selected a pattern and incorporated at least one garment feature.
(3) **Design a Garment (15 min digital design + 30 min physical assembly):** Participants designed a garment from scratch
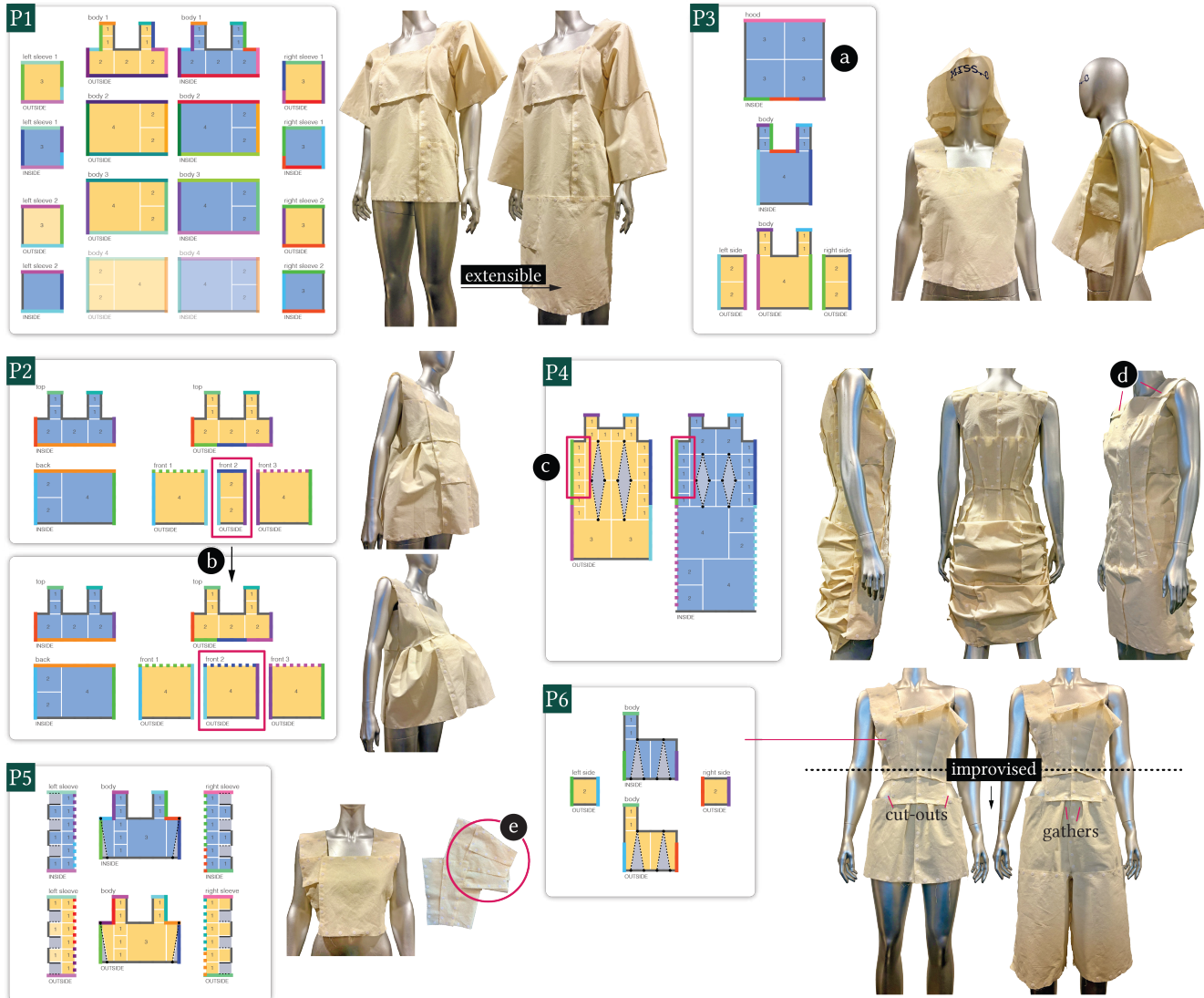

Figure 19: Participants designed and assembled their own modular garment patterns. P1 created a compound dress with extensible and detachable sleeves and skirt. P2 designed maternity wear that accommodates two stages of pregnancy via a single panel swap (b). P3 made a sleeveless hoodie inspired by the one they wore to the study session. P4 designed a formal dress with horizontal gathers at the back of the skirt. We manually improved its assembly by replacing the Δ-sized foundation modules along the sides (c) with larger 2Δ units that span the seams. We also observed that increasing connection density along the seam interface (Sec. 4.1.2) would yield cleaner joins than in the assembled prototype (d). P5 sketched a top with gathered *and* pleated sleeves, aiming to create a puff sleeve. However, due to a shortage of pre-fabricated modules, we created an ungathered version of the sleeve (e) instead. P6 drew an asymmetric top and improvised the design into a jumpsuit during assembly. They added a waistband with cutouts and gathers to reconcile dimensional differences between the top and bottom.

and constructed their designs from pre-fabricated prototyping modules guided by system-generated assemblies.

One participant (P2) returned later to create additional garments.

7.3.3   Task 1: Resizing a T-shirt. Participants redrew a T-shirt pattern to accommodate new target measurements (Fig. 13). Resizing was considered successful if the modified pattern fit correctly on the specified body model in simulation. All participants completed the task successfully within 15 minutes. While some found drawing and stitching the armhole edges somewhat tricky, all participants rated the overall task as "very easy" or "easy" on a 5-point Likert scale. For P2, *"solving the puzzle"* was their favorite part.

7.3.4   Task 2: Adding a Garment Feature. Participants selected and augmented a base garment by adding at least one feature—gathers, pleats, or darts. Some (P4, P5) recreated the compound skirt from Fig. 10, while others (P2, P3) explored more open-ended manipulations, such as gathering a single sleeve or repeatedly adding pleats at seams (e.g., Fig. 12). Four participants (P1, P3, P4, and P5) rated this task as "easy," while P2 and P6 found it "slightly difficult," primarily due to the challenge of understanding more complex feature interactions (App. A) and how they translate to fabric manipulation in the physical world.

7.3.5   Task 3: Designing a Garment. Participants designed and assembled a modular garment from scratch.

P1 created a compound dress with extensible sleeves and body (Fig. 19a), designed to adapt to changing temperatures throughout the day (e.g., transitioning from short to long sleeves in the evening).

P2 designed a piece of maternity wear featuring an interchangeable panel to accommodate different stages of pregnancy (Fig. 19c). They used gathers in the midsection to provide additional space while keeping the top of the garment unchanged. The garment was constructed in under 15 minutes and later modified in less than 5 minutes, demonstrating the efficiency of modular construction for both initial assembly and subsequent reconfiguration to achieve significant fit adjustments.

P3, after experimenting with simulation features, designed a sleeveless hoodie inspired by the one they wore to the study session (Fig. 19c). Although reverse-engineering the hood's pattern took some time, recreating and validating it in the studio was quick (15 min), as was the physical construction (10 min).

P4 designed a dress with a skirt gathered horizontally at the back (Fig. 19d). They iterated multiple times on the placement of the gathers, using the simulation to refine their design. P4 found the unit-based system helpful for quick measurements and counting (e.g., the widest part of the mannequin had a circumference of 12Δ), which simplified the design process compared to traditional sewing methods. Due to the complexity of the gathers, assembly took approximately 30 minutes.

P5 created a top featuring gathered and pleated sleeves (Fig. 19e), aiming for a short puff-sleeve design. To achieve the desired fullness and shaping, they combined gathers and pleats in parallel.

P6 began with an asymmetric top, aiming to challenge the more regular nature of our modular garments, and later improvised complementary pants to form a jumpsuit (Fig. 19f). The improvisation demonstrated both the flexibility of our modular system and the value of digital prototyping. To resolve the size mismatch when joining the top and bottom, the participant added a waistband of unit-sized modules with cutouts and gathered the pants to attach them. The top, based on a pre-designed digital pattern, took just 10 minutes to assemble, while the improvised pants required 40 minutes of trial and error. However, the arrangement of small modules and cutouts along the waistband inspired the participant to suggest a patchwork version if they created a higher-quality version of the garment from the muslin prototype.

7.4   User Feedback

We invited participants to provide feedback on their experience designing and assembling modular garments with REFASHION.

7.4.1   Adaptation. Participants highlighted the value of modular garments in adapting to evolving environments, bodies, and needs. P5 noted the garments' flexibility in responding to changing weather conditions and facilitating the shift from professional to casual attire. P2 described modularity as "ideal" for maternity wear, appreciating the ability to adjust rather than replace an entire wardrobe. They also pointed to children's apparel as a promising application, given how rapidly kids outgrow clothes. Looking beyond garments, they envisioned accessories like bags that could *"change size depending on how much stuff you need to carry that day"*.

7.4.2   Rapid Prototyping. Participants valued REFASHION's support for rapid prototyping. P1, P3, and P4 appreciated being able to iterate quickly digitally, guided by feedback from the simulation. P3 emphasized the efficiency of assembling mockups from simple, reusable modules, expressing enthusiasm for fast, low-cost iteration without the worry of fabric waste. P4 highlighted the ease of thinking in terms of modular "units" for both sizing and styling, as well as the ability to validate a concept digitally and physically without the need for cutting or sewing. P1 echoed this sentiment, noting that they would use the system for general pattern-making by rapidly prototyping a garment's baseline silhouette and style before fine-tuning the pattern.

7.4.3   Sewing-Free Construction. Participants emphasized that by abstracting and simplifying traditional garment-making, REFASHION is more approachable for those with limited sewing experience. P1 noted that the system could serve as an effective learning tool for garment construction—even beyond modular designs—by simplifying elements like seams, gathers, and pleats. Several participants appreciated that REFASHION enabled beginners to quickly produce functional garments; for example, P4 said, *"I learned [the system], made the pattern, and assembled it in two hours."* P4 further explained that, unlike traditional methods involving extensive sketching, proportion calculations, and multiple test pieces, REFASHION's streamlined approach simplifies the entire process. P3 likened its modeling experience to *"a Tinkercad aesthetic,"* suggesting it could serve a similar role as an approachable and playful platform for rapid garment design and learning.

7.4.4   Modularity as an Aesthetic. Participants expressed enthusiasm for modularity not only as a functional feature but also as an aesthetic choice. P3 posed the question of whether modular clothing should conceal its structure or instead highlight it to *"purposefully make a statement."* P2 advocated for embracing a bold modular

aesthetic, envisioning garments that *"push the box aesthetic"* as a counterpoint to fleeting fashion trends. P6 was particularly excited about using the system's constraints to explore visual patterns such as patchwork and cutouts. P2 also expressed interest in adding functional or decorative accessories—such as pockets and bows—either to existing connection points or via additional hardware.

8 DISCUSSION

Many insights from our user evaluation highlight broader themes related to designing for reconfiguration and reuse within fashion and across other domains.

8.1 Designing for Re-Assembly

To reduce assembly time, our method minimizes the number of modules required to construct a design, often favoring larger modules when possible. However, larger modules can restrict fabric variation within a region—limiting visual expression when modules are used as design features (e.g., Fig. 18)—and may increase the effort required to reconfigure garments, as fewer, larger blocks offer less flexibility for reshaping or rearranging. Several participants (P1, P2, P5) observed that fully disassembling and reassembling a garment is often unnecessary when transitioning between modular designs. For instance, clusters of modules forming larger panels—such as those forming a bodice or skirt—can often be reused with little to no breakdown. This insight suggests opportunities for "capsule panels" or guided assemblies that preserve and reuse multi-module groups. It may also be beneficial to design multiple modular garments as a cohesive set, selecting module sizes and materials to support efficient and expressive recombination.

8.2 Designing for Change

Our modular garments are intended to be reconfigured over time. While a wearer might switch between predefined styles—such as adding or removing a hood, or converting pants into shorts—it remains unclear how often such reversible transformations occur in practice. Alternatively, one could imagine creating a new configuration each day, never repeating the same garment form. These varied use cases suggest that design strategies should consider not only the end configurations but also the frequency and nature of change. Depending on how often and in what ways a garment is expected to be reconfigured, transitions can be optimized—for example, by designing specific transformation pathways and minimizing the steps or modules needed to move between them.

8.3 Simple Modules Promote Efficiency

The regular geometries of the modules significantly improve material efficiency compared to conventional garment patterns, whose irregular contours often leave large amounts of unused fabric when cut from fabric sheets, and their uniform and repetitive design streamlines assembly. Participants were able to construct garments by following consistent attachment patterns without the need for individualized instructions for each piece. Our set of modules, though small, achieved a wide range of shaping by supporting gathers, pleats, and darts through carefully designed interfaces and specialized blocks. The shaping modules, though more complex than the foundational components, efficiently transform 2D sheets into

structured 3D forms that fit the body and remain reusable across a variety of garment sizes and styles. Extending this set of modules to allow for even more customizable sizing and closures, such as drawstrings or zippers, would further bring modular clothing closer to the fits achieved by conventional clothing.

9 LIMITATIONS & FUTURE WORK

Our definition of modular garments represents just one possible interpretation of modularity. While our formulation of the design space supports a range of garments and use cases, there are many opportunities to extend these ideas and improve the system.

9.1 Sizing Control

While users can customize module dimensions, the unit size Δ controls the granularity of fit adjustments (e.g., with $\Delta = 8$ cm, all sizing occurs in 8 cm increments). We observe a trade-off between unit size and the number of modules—and thus the likely assembly time—required for a given garment. Although participants achieved a wide range of garment fits using our modules, more precise fitting could be supported by offering a more continuous range of module sizes or in-module adjustments. For example, flexible interfaces, such as drawstrings or laces, could enable fine-grained control. Finally, alternative module shapes (e.g., curved pieces) may help modular garments better emulate conventional clothing.

9.2 Supporting Multi-Fabric Designs

Several participants (P2, P3, P6) discussed embracing modularity by making the units more visually distinct through patchwork. Using modules in different colors, prints, and textures could give wearers greater control over a garment's appearance while highlighting its modular structure. Future work could explore interactive decomposition tools that allow users to place modules dynamically via a drag-and-drop interface to achieve custom patterns.

9.3 Wearability

While participants enjoyed designing and assembling modular garments, some (P2 and P3) raised concerns about longer-term wearability. P2 joked, *"If I bent over and the snaps unsnapped, that would be super embarrassing"*. In response, the first author wore the sleeveless denim jacket from Fig. 17 over the course of a day across various settings, including the office, machine shop, and a dinner outing with friends without noticeable decline in quality. Nevertheless, further work is required to ensure these garments meet the standards of comfort and durability required for daily wear and regular washing G3.

10 CONCLUSION

Modular garments offer a pathway toward clothing that adapts to evolving styles, fits, and needs—while supporting repair, reuse, and self-expression. Refashion demonstrates one approach to designing such dynamic garments. We hope this work inspires future efforts toward a more sustainable and creative future for fashion.

ACKNOWLEDGMENTS

REFERENCES

[1] Muhammad Abdullah, Romeo Sommerfeld, Laurenz Seidel, Jonas Noack, Ran Zhang, Thijs Roumen, and Patrick Baudisch. 2021. Roadkill: Nesting Laser-Cut Objects for Fast Assembly. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 972–984.

[2] Muhammad Abdullah, Romeo Sommerfeld, Bjarne Sievers, Leonard Geier, Jonas Noack, Marcus Ding, Christoph Thieme, Laurenz Seidel, Lukas Fritzsche, Erik Langenhan, et al. 2022. HingeCore: Laser-Cut Foamcore for Fast Assembly. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[3] Aurore Bardey, Madison Booth, Giuliana Heger, and Jonas Larsson. 2022. Finding yourself in your wardrobe: An exploratory study of lived experiences with a capsule wardrobe. *International Journal of Market Research* 64, 1 (2022), 113–131.

[4] Aric Bartle, Alla Sheffer, Vladimir G Kim, Danny M Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-driven pattern adjustment for direct 3D garment editing. *ACM Trans. Graph.* 35, 4 (2016), 50–1.

[5] Floraine Berthouzoz, Akash Garg, Danny M Kaufman, Eitan Grinspun, and Maneesh Agrawala. 2013. Parsing sewing patterns into 3D garments. *Acm Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

[6] Jeffrey Bosboom, Erik D Demaine, Martin L Demaine, Jayson Lynch, Pasin Manurangsi, Mikhail Rudoy, and Anak Yodpinyanee. 2015. k-Piece dissection is NP-hard. In *18th Japan Conf. on Discrete and Computational Geometry and Graphs*, Vol. 2.

[7] CLO. 2024. Clo3D. https://www.clo3d.com/

[8] Joost De Cock. 2019. Free Sewing. https://github.com/freesewing/freesewing.

[9] Charles de Malefette, Anran Qi, Amal Dev Parakkat, Marie-Paule Cani, and Takeo Igarashi. 2023. Perfectdart: Automatic dart design for garment fitting. In *SIGGRAPH Asia 2023 Technical Communications*. 1–4.

[10] Ashley Del Valle, Jennifer Jacobs, and Emilie Yu. 2025. texTile: Making and Re-making Crochet Granny Square Garments Through Computational Design and 3D-printed Connectors. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*. 2445–2464.

[11] Kristin N Dew, Samantha Shorey, and Daniela Rosner. 2018. Making within limits: Towards salvage fabrication. In *Proceedings of the 2018 Workshop on Computing within Limits*. 1–11.

[12] Noah Duncan, Lap-Fai Yu, Sai-Kit Yeung, and Demetri Terzopoulos. 2017. Approximate dissections. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

[13] Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J Mitra, Helmut Pottmann, and Mark Pauly. 2010. Paneling architectural freeform surfaces. In *ACM SIGGRAPH 2010 papers*. 1–10.

[14] John Forrest and Robin Lougee-Heimer. 2005. CBC User Guide. https://www.coin-or.org/Cbc/cbcuserguide.html In *Emerging Theory, Methods, and Applications*, pages 257–277. INFORMS, Catonsville, MD..

[15] Robert J Fowler, Michael S Paterson, and Steven L Tanimoto. 1981. Optimal packing and covering in the plane are NP-complete. *Inform. Process. Lett.* 12, 3 (1981), 133–137.

[16] Dorit S Hochbaum and Wolfgang Maass. 1985. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM (JACM)* 32, 1 (1985), 130–136.

[17] Wei-Lin Hsiao and Kristen Grauman. 2018. Creating capsule wardrobes from fashion images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7161–7170.

[18] Sofia Ilmonen. 2025. Sofia Ilmonen. https://www.sofiailmonen.com/. Accessed: 23 June 2025.

[19] Benjamin Jones, Yuxuan Mei, Haisen Zhao, Taylor Gotfrid, Jennifer Mankoff, and Adriana Schulz. 2021. Computational design of knit templates. *ACM Transactions on Graphics (TOG)* 41, 2 (2021), 1–16.

[20] Lee Jones and Audrey Girouard. 2021. Patching textiles: Insights from visible mending educators on wearability, extending the life of our clothes, and teaching tangible crafts. In *Proceedings of the 13th Conference on Creativity and Cognition*. 1–11.

[21] Jae Woo Kim. 2014. Survey on automated LEGO assembly construction. (2014).

[22] Bongmjin 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.

[23] Maria Korosteleva and Olga Sorkine-Hornung. 2023. GarmentCode: Programming Parametric Sewing Patterns. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–15.

[24] Flavia La Rocca. 2023. Flavia La Rocca. https://www.flavialarocca.com/. Accessed: 2025-06-23.

[25] Sabrina Lakhdhir, Charles Perin, and Sowmya Somanath. 2024. Expressive Clothing: Understanding Hobbyist-Sewers' Visions for Self-Expression Through Clothing. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–17.

[26] Mackenzie Leake and Ross Daly. 2024. ScrapMap: Interactive Color Layout for Scrap Quilting. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–17.

[27] Mackenzie Leake, Kathryn Jin, Abe Davis, and Stefanie Mueller. 2023. InStitches: Augmenting Sewing Patterns with Personalized Material-Efficient Practice. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[28] Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. 2018. Foldsketch: enriching garments with physically reproducible folds. *ACM Transactions on Graphics* 37, 4 (Aug. 2018), 1–13. https://doi.org/10.1145/3197517.3201310

[29] Shuhua Li, Ali Mahdavi-Amiri, Ruizhen Hu, Han Liu, Changqing Zou, Oliver Van Kaick, Xiuping Liu, Hui Huang, and Hao Zhang. 2018. Construction and fabrication of reversible shape transforms. *ACM Trans. Graph.* 37, 6 (2018), 190.

[30] Kaixuan Liu, Xianyi Zeng, Pascal Bruniaux, Xuyuan Tao, Xiaofeng Yao, Victoria Li, and Jianping Wang. 2018. 3D interactive garment pattern-making technology. *Computer-Aided Design* 104 (2018), 113–124.

[31] Yuanpeng Liu, Ting-Uei Lee, Anooshe Rezaee Javan, Nico Pietroni, and Yi Min Xie. 2024. Reducing the number of different faces in free-form surface approximations through clustering and optimization. *Computer-Aided Design* 166 (2024), 103633.

[32] Jasmine Lu, Beza Desta, K. D. Wu, Romain Nith, Joyce E Passananti, and Pedro Lopes. 2023. EcoEDA: Recycling E-Waste During Electronics Design. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 30, 14 pages. https://doi.org/10.1145/3586183.3606745

[33] Wendy Mak. 2017. *The Capsule Wardrobe: 1,000 Outfits from 30 Pieces*. Skyhorse.

[34] Marfa Stance. 2025. Marfa Stance. https://www.marfastance.com/. Accessed: 2025-06-23.

[35] Stuart Mitchell, Michael O'Sullivan, and Iain Dunning. 2011. PuLP: A Linear Programming Toolkit for Python. Available at https://optimization-online.org/2011/09/3178/.

[36] Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. 2014. faBrickation: fast 3D printing of functional objects by integrating construction kit building blocks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3827–3834.

[37] Keunwoo Park, Conrad Lempert, Muhammad Abdullah, Shohei Katakura, Jotaro Shigeyama, Thijs Roumen, and Patrick Baudisch. 2022. FoolProofJoint: Reducing Assembly Errors of Laser Cut 3D Models by Means of Custom Joint Patterns. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–12.

[38] Shubham Patil, Debopriyo Banerjee, and Shamik Sural. 2021. A graph theoretic approach for multi-objective budget constrained capsule wardrobe recommendation. *ACM Transactions on Information Systems (TOIS)* 40, 1 (2021), 1–33.

[39] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. 2019. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10975–10985.

[40] Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational pattern making from 3D garment models. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.

[41] Petit Pli. 2025. *Petit Pli*. https://www.roberts-wood.com/ Accessed 6 June 2025.

[42] Anran Qi and Takeo Igarashi. 2024. PerfectTailor: Scale-preserving 2-D pattern adjustment driven by 3-D garment editing. *IEEE Computer Graphics and Applications* 44, 4 (2024), 126–132.

[43] Timo Rissanen and Holly McQuillan. 2016. *Zero waste fashion design*. Bloomsbury Publishing.

[44] Michael L Rivera, S Sandra Bae, and Scott E Hudson. 2023. Designing a Sustainable Material for 3D Printing with Spent Coffee Grounds. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 294–311.

[45] Ticha Sethapakdi, Daniel Anderson, Adrian Reginald Chua Sy, and Stefanie Mueller. 2021. Fabricaide: Fabrication-aware design for 2d cutting machines. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[46] Ariel Shamir. 2008. A survey on mesh segmentation techniques. In *Computer graphics forum*, Vol. 27. Wiley Online Library, 1539–1556.

[47] Jonathan Richard Shewchuk. 1996. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. *Applied Computational Geometry Towards Geometric Engineering* (1996), 203–222.

[48] Katherine W Song, Aditi Maheshwari, Eric M Gallo, Andreea Danielescu, and Eric Paulos. 2022. Towards decomposable interactive systems: Design of a backyard-degradable wireless heating interface. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–12.

[49] Katherine W Song and Eric Paulos. 2021. Unmaking: Enabling and Celebrating the Creative Material of Failure, Destruction, Decay, and Deformation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[50] Peng Song, Chi-Wing Fu, Yueming Jin, Hongfei Xu, Ligang Liu, Pheng-Ann Heng, and Daniel Cohen-Or. 2017. Reconfigurable interlocking furniture. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–14.

[51] Nobuyuki Umetani, Danny M Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.* 30, 4 (2011), 90.

[52] Ludwig Wilhelm Wall, Alec Jacobson, Daniel Vogel, and Oliver Schneider. 2021. Scrappy: Using Scrap Material as Infill to Make Fabrication More Sustainable. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[53] Xin Wen, S. Sandra Bae, and Michael L. Rivera. 2025. Enabling Recycling of Multi-Material 3D Printed Objects through Computational Design and Disassembly by Dissolution. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 303, 21 pages. https://doi.org/10.1145/3706598.3714080

[54] Katja Wolff, Philipp Herholz, Verena Ziegler, Frauke Link, Nico Brügel, and Olga Sorkine-Hornung. 2023. Designing personalized garments with body movement. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, 180–194.

[55] Katja Wolff and Olga Sorkine-Hornung. 2019. Wallpaper Pattern Alignment along Garment Seams. *ACM Trans. Graph.* 38, 4, Article 62 (July 2019), 12 pages. https://doi.org/10.1145/3306346.3322991

[56] 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.

[57] Shanel Wu and Laura Devendorf. 2020. Unfabricate: designing smart textiles for disassembly. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[58] Jiaxian Yao, Danny M Kaufman, Yotam Gingold, and Maneesh Agrawala. 2017. Interactive design and stability analysis of decorative joinery for furniture. *ACM Transactions on Graphics (TOG)* 36, 2 (2017), 1–16.

[59] Ruowang Zhang, Stefanie Mueller, Gilbert Louis Bernstein, Adriana Schulz, and Mackenzie Leake. 2024. Wastebanned: Supporting zero waste fashion design through linked edits. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[60] Jingwen Zhu, Lily Winagle, and Hsin-Liu Kao. 2024. EcoThreads: Prototyping Biodegradable E-textiles Through Thread-based Fabrication. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–17.

A FEATURE CASEWORK

We provide detailed casework outlining the types of user interactions and control available when manipulating gathers, pleats, and darts in REFASHION Studio.

A.1 Gathers

The user initiates a gather by selecting an edge along a seam. Gathering doubles the length of the selected edge, and to maintain orthogonal panel geometry, the opposing edge must also double in length. Fig. 20 illustrates several scenarios that show whether and how the panel and surrounding seams are adjusted to accommodate the gathering.
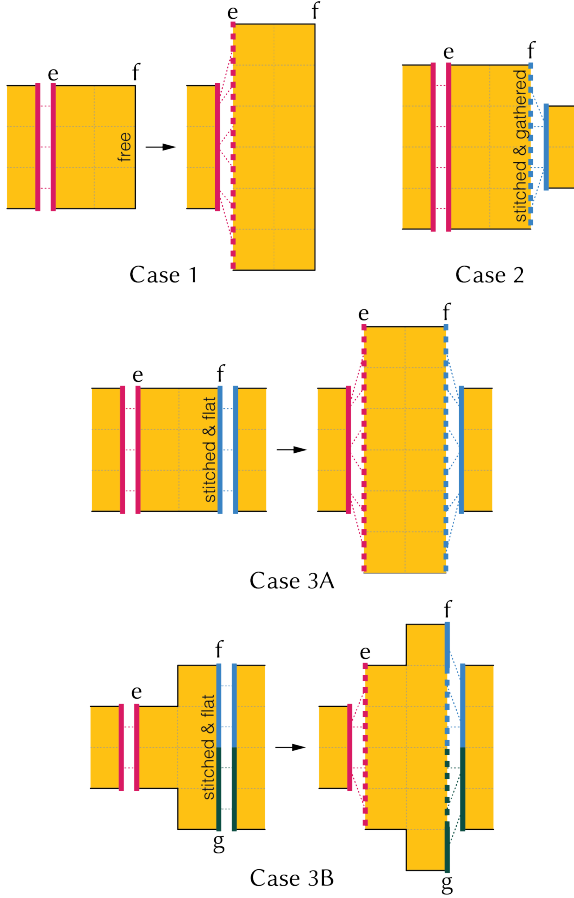
Figure 20: Scenarios for gathering edge e in the red seam, depending on the state of its opposite edge f. Case 1: If f is a free edge, both e and f double in length; Case 2: If f is already gathered, then gathering e would exceed f's maximum length and is therefore prohibited; Case 3AB: If f is part of a flat seam, both e the segments of f *directly opposing* e are gathered. Dotted lines indicate edge correspondences at the unit-segment-level.

By default, our system minimizes cascading changes across panels. In Case 3A, for example, we gather f rather than expanding

the adjacent panel it connects to. If the user prefers to keep f flat and instead double the connected panel, we provide interactions to support this alternative adjustment (Fig. 21c).

A.2 Pleats

To pleat, the user first specifies a fold direction—right, left, up, or down—and then either converts a selected panel unit into a pleat module or inserts a pleat module adjacent to it. The former may remove a unit-length edge through folding, requiring seam adjustments (e.g., Case 2 in Fig. 21). The latter can preserve edge dimensions after folding (e.g., Case 1 in Fig. 22) but alters the geometry of the panel. We illustrate the possible interactions for each operation in the following two figures, respectively.
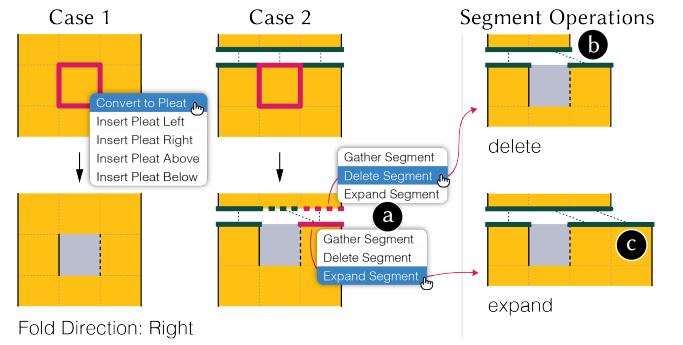

Figure 21: Converting a panel unit into a pleat results in one of two scenarios. Case 1: If the pleat is not adjacent to a seam *or* its fold direction is not orthogonal to the seam edge, no seam adjustments are needed. Case 2: If the pleat is flush against a seam *and* folds orthogonally—removing a unit edge segment—the opposing segment must be rematched, introducing a gather (a). Users can resolve undesired gathers by either (b) deleting the gathered segment on the opposite edge or (c) expanding a unit segment on the current edge to preserve the original seam length after folding.
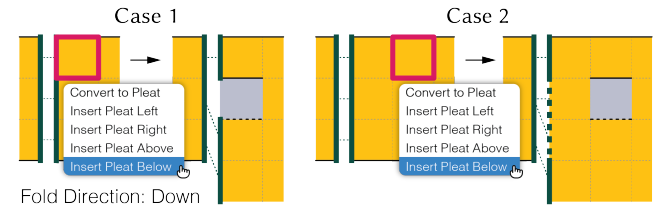

Figure 22: Inserting a pleat introduces a new unit-width strip of geometry to the panel. This may affect seams at the strip's endpoints—as seen in Case 2, where gathering is required. However, if the pleat is adjacent and folds perpendicular to the seam—removing the newly introduced edge length—then no rematching is needed, as shown in Case 1.

A.3 Darts

To add a dart, the user clicks any grid point within a panel—defining the midpoint of the cut-out's triangular base—and selects an orientation (vertical or horizontal) from the dropdown that appears. Dart height and width are chosen from a separate dropdown above the editor. A dart is added only if there is sufficient space in the panel and the operation respects constraints along the seams. These considerations are illustrated in the following two figures.
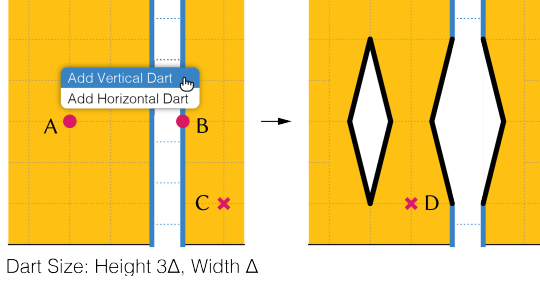

Dart Size: Height 3Δ, Width Δ

Figure 23: When a dart is added at an interior point of a panel, it is always a diamond dart—formed by aligning two dart modules—and the system checks only for sufficient space. Consider adding vertical diamond darts at points A, B, C, and D in sequence: A succeeds, as the dart fits entirely within the panel. B lies along a seam; since there is space across both panels, the dart is allowed. C is disallowed, even without B, as it would extend beyond the panel boundary. D is also disallowed, as it would intersect with the dart at B.
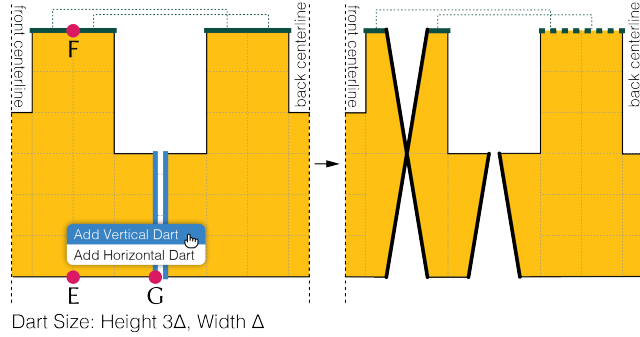

Dart Size: Height 3Δ, Width Δ

Figure 24: Adding darts along the edges of a panel involves more complex casework, often requiring coordination with seam constraints. E: A dart is added perpendicular to a free edge; F: A *universal* dart (with width Δ) is added perpendicular to a seam that is not already gathered; G: A dart is added at a corner, with its two parts joining across a seam.

B 3D SIMULATION

We implement the 3D garment simulation in Refashion Studio using Blender's open-source mass-spring cloth model.

B.1 Feature Modeling

To simulate fabric manipulations such as gathers and pleats, we modeled them directly as follows.

For gathers, we match seams at the *half*-unit level, as shown in Fig. 25a. Each half-unit segment along the flat edge is matched (and later stitched to) every other half-unit segment on the longer edge (red), while the remaining unmatched segments on the longer edge self-match to fold in half during simulation.

For pleats, we treat the modules as cut-outs or holes in the fabric as shown in Fig. 25b. Edges that align during folding are explicitly matched (red), while remaining edges self-match (blue). For pleats along panel edges, their matches may span adjacent panels. Darts are similarly treated as cut-outs, with their legs matched and sewn together during simulation (Fig. 25c).
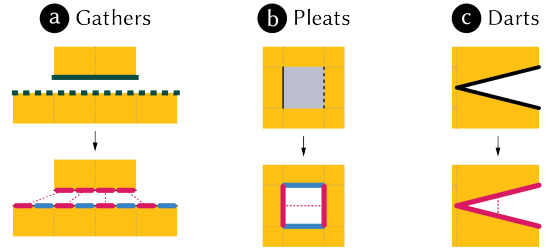

Figure 25: Modeling gathers, pleats, and darts for simulation. Red segments connected by dotted lines are matched and sewn together during simulation. Blue segments are self-matched and fold in half during simulation.

B.2 Meshing

Given a pattern represented as described above, we prepare a mesh as follows. First, we convert coordinates from the pattern alignment view (Fig. 8) to Blender's coordinate system. Panels marked as "outside-up" are placed in front of the body model (+0.2 m), and "inside-up" panels behind it (−0.2 m). Next, we subdivide each panel edge by inserting Steiner points at regular 1 cm intervals, then triangulate each panel using the Triangle library [47]. Finally, we add "sewing threads"—spring-based edges—between corresponding vertices of matching panel edges (see inset). Edge subdivision controls both triangle size and the density of sewing threads, which pull the panels together during simulation to wrap the garment around the body model.
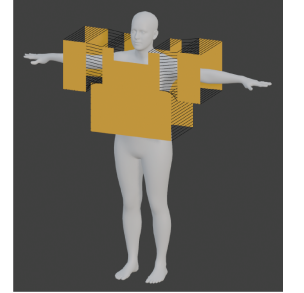
B.3 Results

We approximate a medium-weight woven fabric using Blender's cloth modifier, with balanced stiffness—tension, compression, shear, and bending all set to 3—and an areal density of 0.6 kg/m^2, which considers the additional weight contributed by garment connectors.

We use Blender's default cloth solver with a moderate simulation quality setting (7 substeps per frame). The simulation is run for

100 frames and typically converges within 30, requiring only 3–5 seconds on our MacBook Air, M2, 2023, with 8 GB memory.
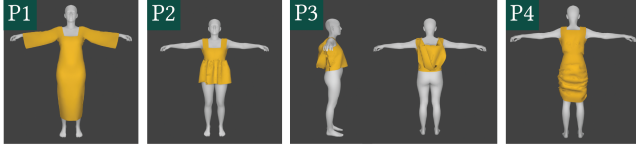

Figure 26: Screenshots of simulation results from our user study, showcasing a range of garments with varying features and complexities on different avatars.

C DECOMPOSITION EVALUATION

We conducted a brief evaluation of our ILP-based decomposition approach to understand how pattern characteristics—specifically area, irregularity, and panel count—affect solver runtime and assembly efficiency, measured by the number of modules. All experiments assumed unlimited availability of module sizes from 1 to 4 (i.e., $\Delta = 1$). We used Google Colab (Python 3 on a Google Compute Engine backend, with 12.7 GB of RAM). While preliminary, the results demonstrate sufficient performance for many practical use cases.

For patterns with regular panels (e.g., the square panel in Fig. 27, left), runtime tends to grow exponentially with size, with occasional spikes in difficulty. Introducing small irregularities—such as randomly removing 1% of units (Fig. 27, center)—can significantly increase runtime. However, more extensive removal (e.g., 10%) can lead to much faster solves (Fig. 27, right), likely due to the resulting smaller contiguous regions reducing the number of feasible module placements and, in turn, the number of ILP variables.
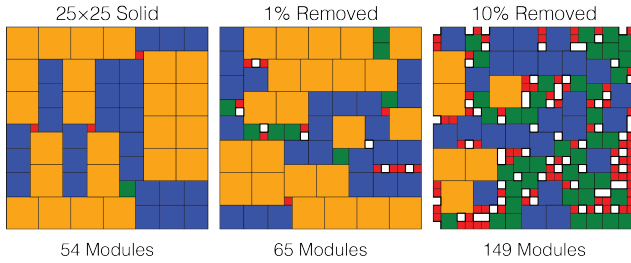

Figure 27: Decompositions for panels with increasing numbers of random cutouts, with modules colored by size. A 25×25 square panel (left) solves in 3.91 seconds. Removing just 1% of units (center) more than doubles the runtime to 8.68 seconds, while removing 10% (right) dramatically reduces it to 0.46 seconds.

For patterns composed of multiple panels, runtime scales approximately linearly with the number of components, as each panel can be solved independently. These findings suggest that preprocessing large patterns into smaller, disconnected regions can improve predictability and help avoid worst-case exponential runtimes.

D FABRICATION DETAILS

Before cutting, the fabric was ironed to remove wrinkles. Modules were compactly arranged to minimize material waste. For example, Fig. 28 shows an efficient layout of dart modules—packed as rectangles—and foundation modules. We used a Universal 120W laser cutter for fabrication. For light cottons, we used 50% power and 100% speed for cutting, and 12% power at 100% speed for optional engraving of markers. For denim, we used up to 100% power and 80% speed. These settings were tuned to avoid cutting completely through the material—preventing loose pieces during the process—while still allowing for easy removal after cutting.
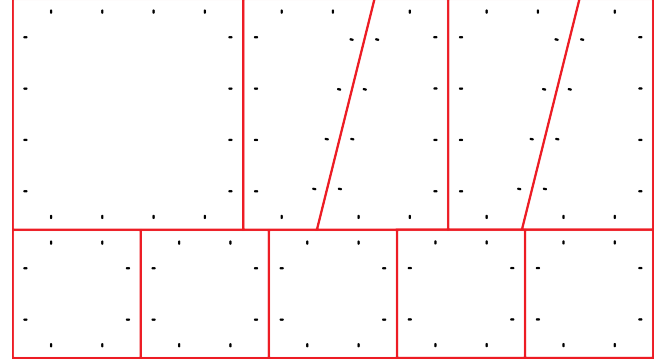

Figure 28: Dart modules can be packed into rectangles by reflecting one of their trapezoidal halves. Red lines denote module outlines, and black ticks mark connector positions.