# Intersection-Free Garment Retargeting

ZIZHOU HUANG, New York University and Roblox, USA
CHRYSTIANO ARAÚJO, Roblox, USA
ANDREW KUNZ, Roblox, USA
DENIS ZORIN, New York University, USA
DANIELE PANOZZO, New York University, USA
VICTOR ZORDAN, Roblox, USA

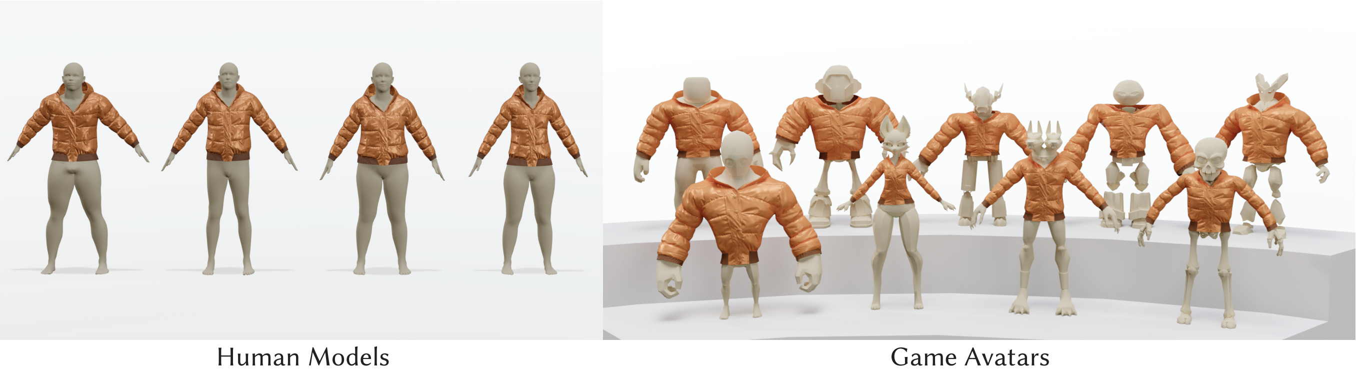

Fig. 1. We transfer an artist-designed puffer coat dressed on a human-shaped mannequin to a collection of SMPL humans (left) and virtual non-human characters (right). The undressed avatars and complete results are shown in Figure 11 and Figure 12.

Manual design of garments for avatars requires a large effort. Garment retargeting methods can save manual efforts by automatically deforming an existing garment design from one avatar to another. Previous methods are limited to human avatars with small variations in body shapes, while non-human avatars with unrealistic characteristics widely appear in games and animations. In this paper, the goal is to retarget artist-designed garments on a standard mannequin to a more general class of avatars. While there is a lack of training data of various avatars wearing garments, we propose a training-free method that performs optimizations on the mesh representation of the garments, with a combination of loss functions that preserve the geometrical features in the original design, guarantee intersection-free, and fit the garment adaptively to the avatars. Our method produces simulation-ready garment models that can be used later in avatar animations.

CCS Concepts: • **Computing methodologies** → **Shape modeling**.

Additional Key Words and Phrases: Animatable avatar, clothing transfer

**ACM Reference Format:**
Zizhou Huang, Chrystiano Araújo, Andrew Kunz, Denis Zorin, Daniele Panozzo, and Victor Zordan. 2025. Intersection-Free Garment Retargeting. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3721238.3730590

## 1 INTRODUCTION

Adapting artist-designed garments to new characters manually requires a lot of effort. Garment retargeting methods aim to automatically transfer existing garments from one avatar to another, saving the manual effort of editing the garment models. Existing methods [Lin et al. 2024; Pons-Moll et al. 2017; Wang 2018] have focused on transferring between human bodies with different body sizes, supporting relatively small deformation over the retargeted garments. However, in games and animation, avatars appear widely with extreme body proportions that defy human norms [Coros et al. 2010; Geijtenbeek et al. 2013; Hecker et al. 2008; Jiang et al. 2023; Li et al. 2023; Yamane et al. 2010]. Our aim is to perform garment retargeting in a manner that can allow for the variation and topology differences seen in game avatars.

In this paper, our goal is to support the retargeting of designed garments between avatars, especially those with unrealistic characteristics, including out-of-distribution proportions, voids, and detached components (Figure 1). Beyond the more extreme deformations anticipated, we wish to make no assumptions of global correspondence between avatar surfaces, breaking the requirements of previous works [Brouet et al. 2012; Lin et al. 2024; Pons-Moll et al. 2017]. Relaxing this is more general, but further, with such correspondences, garment deformation under extreme cases can become distorted due to the large disparity in body shapes of our target avatars – leading to distorted final garment shapes after transfer.

Authors' addresses: Zizhou Huang, zizhou@nyu.edu, New York University and Roblox, USA; Chrystiano Araújo, chrystianosaraujo@gmail.com, Roblox, USA; Andrew Kunz, akunz@roblox.com, Roblox, USA; Denis Zorin, dzorin@cs.nyu.edu, New York University, USA; Daniele Panozzo, panozzo@nyu.edu, New York University, USA; Victor Zordan, vbzordan@roblox.com, Roblox, USA.

*SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada*

ACM ISBN 979-8-4007-1540-2/2025/08
https://doi.org/10.1145/3721238.3730590

Instead, we purposefully enforce positional constraints over specific boundaries of the garments, e.g., cuffs and hems, which do not require manual effort to specify and are not too restrictive. Next, we preserve the look of the individual garment by penalizing local shape changes – maintaining the anticipated look embedded in the original model. Together, our technique provides an easily controllable balance between fitting on the avatar and the shape of the garment, to achieve the artist's look of the clothing on the final avatar.

The core of our approach is a multi-objective optimization of the garment geometry, represented as a triangular mesh, that preserves the geometric properties of the original design, avoids self-intersections and intersections between the garment and avatar, while adaptively fitting the transferred garment to the new avatar. The key features of our approach include:

- An intersection-free initialization for the optimization that embeds the target avatar into the original garment design.
- An optimization on the garment mesh with a combination of loss functions to preserve the geometric quality of the retargeted garment, avoiding intersections between the garment and the new avatar, and adaptive fit of the garment to the target avatar body, including a select set of guiding constraints.
- It avoids not only intersections between the avatar and garment but also intersections between garments and self-intersections, enabling multi-layer garment retargeting.

We demonstrate the effectiveness of our approach by transferring a set of garments of various types designed on a standard mannequin to a set of avatars with different body shapes. We also provide a small number of useful tuning controls to adapt the fit in the presence of ambiguity. While no previous research showcases like results, for comparison, we compare our results with those of a commercial tool that supports a wide variety of avatars.

## 2 RELATED WORK

We build off of a growing body of work in computer graphics on cloth modeling. Many techniques offer solutions tied to specific applications and their needs, for example, garment model reconstruction [Corona et al. 2021; Dong et al. 2023; Guo et al. 2021; Li et al. 2024; Pons-Moll et al. 2017; Xiu et al. 2023] from 3D scans or 2D images, sewing pattern generation [He et al. 2024; Liu et al. 2023; Pietroni et al. 2022] and arrangement [Liu et al. 2024], physics simulations of garments [Grigorev et al. 2024, 2023; Santesteban et al. 2022; Zheng et al. 2025], etc. However, our work is most informed by the specific focus of retargeting for garments [Brouet et al. 2012; Pons-Moll et al. 2017].

We further distinguish works in garment retargeting into two categories: re-fitting techniques for virtual try-on and those for clothing virtual characters, each serving distinct application areas with differing requirements. Virtual try-on aims to make real-world clothing fit realistically on human shapes in support of the commercial clothing market. In this area, a number of approaches perform retargeting directly from image to image, such as [Ge et al. 2021; Sekine et al. 2014; Yang et al. 2020]. For immersive 3D applications like open-world games and animations, unlike virtual try-ons, the input garments and avatars are normally in the form of 3D models, and it is desirable to have the fitted garments as 3D models, instead of images or animations, to allow interactions with surrounding environments and downstream applications like skeletal animation or physics simulation. Closer to our aims, there have been some recent works that output the fitted garments as 3D models [Brouet et al. 2012; Guan et al. 2012; Patel et al. 2020; Pons-Moll et al. 2017; Wang et al. 2018]. Still, these methods target dressing humans, where the body shapes can be represented as a smooth parametric space [Anguelov et al. 2023; Loper et al. 2015; Pavlakos et al. 2019] with finite dimensions. Virtual characters designed by artists have possibly unrealistic body shapes and much higher diversity that cannot be represented by a set of parameters, e.g., elementals [Hoffman et al. 2023], walking animals [Chun et al. 2023], and robots [Hopkins et al. 2024]. While there is plenty of data on images of dressed humans for training a model to retarget garments on humans, there is a lack of data on dressed characters in animations and games. Unlike human photos, dressed virtual characters require significant efforts from professional artists, making it difficult to adopt data-driven approaches.

In complement, some methods [Li et al. 2024; Wang 2018; Wang et al. 2018] for virtual try-on restrict the garment models to be fabricable, by assuming a pre-defined sewing pattern of the garment. The garment shape space is thus reduced to the boundary curve shapes of each patch. Although this is a valid requirement for real-world garments, garments for virtual characters are often more imaginative and do not adhere to these assumptions, which is why our method operates directly on the 3D garment model instead of 2D sewing patterns.

**Offset-based Retargeting.** A number of prior works [Guan et al. 2012; Lin et al. 2024; Ma et al. 2020; Pons-Moll et al. 2017] make use of the one-to-one correspondence between avatars and transfer garments by explicitly mapping the displacement between the input garment and avatar, to the target avatar. Although this kind of approach is efficient, it has major limitations: (a) The correspondence between avatars should not be distorted, limiting the variation in body shapes. (b) To enable an accurate displacement map, the garment should be close to the avatar surface, and both avatar surfaces should stay reasonably smooth. (c) Unlike optimization methods, explicit evaluation may introduce uncontrollable intersections and distortion.

**Optimization-based Retargeting.** Unlike offset methods, optimization methods allow more control over the retargeted garment from the users and can enforce constraints like intersection-free and shape preservation. Closest to our work, [Brouet et al. 2012] proposes a method to directly operate on the 3D model of the garment to transfer from one avatar to the other. To accomplish this, however, they require a cross-parametrization between the source and target avatars, which is a difficult problem on its own for models with large differences, thus requiring considerable constraints on the design of avatar bodies. Further, such global correspondence may not be possible at all, e.g., avatars with voids and disconnected pieces. Further, their shape preservation is enforced by preserving the triangle normals on the garment surface. In contrast, we relax this constraint to allow rotations and anisotropic scaling, providing more flexibility in the retargeting.

**Physics-aware Retargeting.** As a special class of optimization-based methods, physics-aware retargeting involves constraints on the physics simulation of garments. [Wang 2018] transfers garments between human bodies, preserving the geometry and stretching of garments in physics simulations. However, they are limited to body shapes with small differences and garments that fit tightly, unsuitable for loose clothes like skirts and jackets. [Wang et al. 2018] performs retargeting by optimizing 2D sewing patterns of garments. Their work preserves desired characteristics of clothing, e.g., fold patterns and draping effects, in physics simulations. In contrast, we sidestep the need for a sewing pattern of the garment and instead work on the 3D model directly. [Wolff et al. 2023] presents a method to design and optimize the shape of a garment for a range of poses, such that the garment respects the user input constraints, and fits comfortably but tightly in all poses. [Bartle et al. 2016] offers an editing tool that allows the user to modify the 3D garments in physics simulations directly and automatically updates the corresponding sewing patterns based on the 3D geometry change. Our work aims to preserve the pure geometry of the garment during retargeting, thus, the behavior in physics simulations is beyond the scope of this work.

## 3 METHOD

### 3.1 Problem Formulation

Our retargeting method directly modifies 3D garments, without assuming corresponding 2D sewing patterns, and performs optimizations to balance between shape preservation, fit, and collisions. We use $V,\ F,\ E$ to denote collections of vertices, triangular faces, and edges, respectively.

**Input.** The input of our method includes a source avatar dressed in an artist-designed garment and a target avatar. Each avatar model and garment model is represented as a triangular mesh $(V, F)$, where $V \in \mathbb{R}^{n\times 3},\ F \in \mathbb{N}^{m\times 3}$. Each row of $V$ contains the 3D coordinate of a vertex, and each row of $F$ contains the vertex indices of a triangular face, with $n,\ m$ being the number of vertices and faces, respectively. We denote the source avatar, the garment, and the target avatar mesh by $(V^{sa},\ F^{sa})$, $(V^{g},\ F^{g})$, $(V^{ta},\ F^{ta})$, respectively. We assume that the garment mesh is manifold and has no self-intersections, but the avatar meshes may be non-manifold, disconnected, possibly with self-intersections and degenerate triangles. This relaxed assumption on the target avatar allows our method to fit garments designed by professional artists to avatars created by non-professional users. In addition, we assume that avatars are rigged, i.e., have an animation skeleton. Our algorithm does not restrict the number of nodes in the skeleton, one can provide more nodes for finer control or fewer nodes to reduce manual effort. We denote the skeletons of the source and target avatars as $(V^{sb},\ E^{b})$ and $(V^{tb},\ E^{b})$ respectively.

The output of our method is a new garment mesh that shares the same connectivity as the source garment mesh, which preserves the shape of the source garment while fitting well to the target avatar, with the guarantee of no intersection between the avatar and the garment or self-intersection on the garment itself.

### 3.2 Algorithm Overview

Our algorithm optimizes the source garment geometry to minimize a collection of objectives, including shape preservation, fit, and collisions. We use IPC [Li et al. 2020] for collision handling instead of the more popular (in garment simulation) approaches based on penalty forces or signed distance as it allows us to robustly handle self-collisions of garments. Its guarantee of creating intersection-free results, even for avatars with complex geometry and thin features, is precious in this setting, as the results have higher visual fidelity and can be directly used in cloth simulation pipelines. Additionally, IPC can handle contact between co-dimensional (i.e. non-volumetric) objects [Li et al. 2021] such as surfaces and rods, opening the door to retargeting multiple and complex garments (Figure 6).

However, using IPC introduces an additional requirement: the source garment needs to be positioned on the target avatar without self-intersections. One of our key contributions is thus an algorithm to create this initial state based on projection and adaptive refinement to shrink the avatar to the skeleton. We then embed the shrunk avatar into the source garment and inflate the avatar to its original shape during the optimization while preserving the garment shape. In summary, the main steps of our algorithm are outlined as follows and are also summarized in Figure 2.

Step 1 Project the target avatar mesh onto its skeleton and embed the projected avatar into the source avatar (Section 3.3).

Step 2 Run an optimization with both the target avatar and garment meshes as DoFs. This step inflates the projected target avatar back to its original shape while maintaining the garment shape and guaranteeing the result remains intersection-free (Section 3.5).

Step 3 Fix the target avatar shape and run an optimization on the garment mesh only, to fit the garment to the target avatar while maintaining the garment shape and intersection-free condition (Section 3.6).

The objectives used in Steps 2 and 3 are discussed in Section 3.4.

### 3.3 Step 1. Initialization

To run the optimization, an initialization of the garment on the target avatar is required. IPC [Li et al. 2020] requires the input meshes to be intersection-free, so the garment cannot be directly overlaid on the target avatar. However, since we do not care about the self-intersection of the avatar, instead of deforming the garment, we can shrink the target avatar to fit inside the garment as the initial guess and inflate the target avatar back to its original shape using optimization. A naive way is to shrink the target avatar to a single point at the center of the source avatar. In this way, however, the limbs of the target avatar all lie at the trunk of the body at the beginning of the optimization, so there is a high probability that the limbs do not get into the correct sleeve or trouser leg during the inflation. Instead, we shrink the target avatar to its skeleton, making sure that the limbs overlap with the corresponding limb bones, and map the target skeleton to the source skeleton based on the correspondence (Figure 2). In this way, the limbs are surrounded by the corresponding sleeves or trouser legs at the beginning of the optimization.

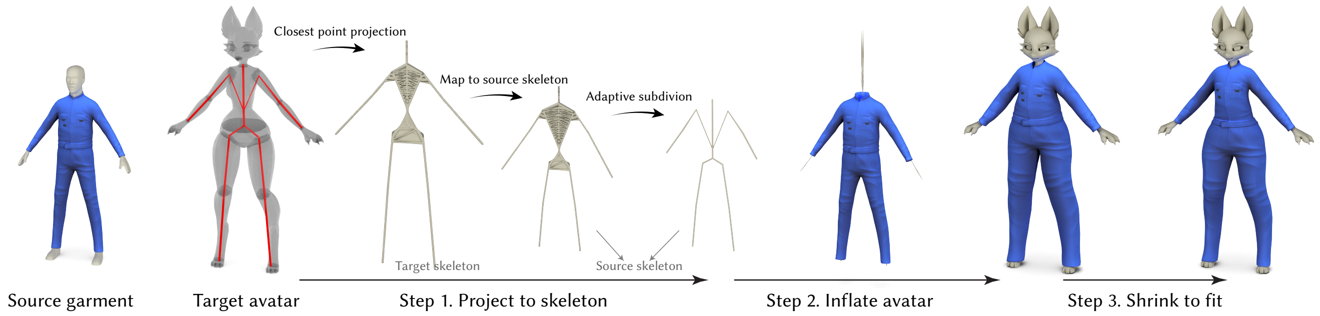

Fig. 2. Illustration of the algorithm pipeline.

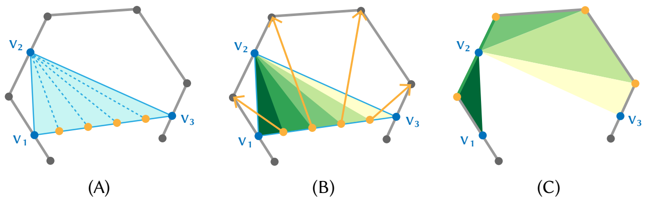

Fig. 3. Illustration of resolving a triangle spanning multiple bones (gray). (A) In the triangle $\triangle v_1 v_2 v_3$ (blue), there are 4 bone joints (dark gray) between $v_1$ and $v_3$, 3 between $v_2$ and $v_3$, and 1 between $v_1$ and $v_2$. Therefore, we pick edge $v_1 v_3$ and insert 4 vertices (yellow) uniformly on the edge. (B) The triangle is split into 5 smaller triangles, and each inserted vertex is moved to an intermediate bone joint between $v_1$ and $v_3$. (C) After the operations, all edges (including newly added ones) span over at most 3 joints.

To shrink the target avatar to its skeleton, for each vertex $v_i \in V^{ta}$ on the target avatar, we first find the closest point $\tilde{v}_i$ on the bone $e_i \in E^b$ that $v_i$ belongs to, and move $v_i$ to $\tilde{v}_i$. Although all vertices in $V^{ta}$ now lie on the target skeleton, the triangular mesh does not necessarily overlap with the skeleton, since one triangle may span multiple bones when its three vertices are projected to different bones (Figure 3A).

To resolve these triangles, we subdivide them and project the refined points onto the joints of skeleton bones. More specifically, within each triangle that spans over multiple bones, we count the number of bone joints between every two vertices. If we consider the bones as nodes and bone joints as edges, the skeleton forms a graph. The number of bone joints between two vertices on two different bones is essentially the shortest path length between the two bones in the graph, which can be precomputed using the Floyd–Warshall algorithm. As in Figure 3, we then adopt a greedy approach and pick the *farthest* pair of vertices, subdivide the edge into multiple segments by inserting intermediate vertices, and move each intermediate vertex to the intermediate bone joints. The original triangle is subdivided into multiple smaller triangles accordingly, and each new triangle spans over strictly less number of bones compared with the original triangle. Note that we do not subdivide the adjacent triangles accordingly, i.e., the subdivision is non-conforming, since we do not require the avatar mesh to be conforming in the optimization. We perform this operation for all triangles that span over multiple bones iteratively until all triangles exactly overlap with the skeleton.

*Remark.* The iterative process terminates in a finite number of iterations, bounded by the number of skeleton bones, since the number of bones that each triangle spans strictly reduces after every subdivision. In practice, it terminates in 2~4 steps in our examples.

## 3.4 Objectives

Before discussing the optimization algorithms in Steps 2 and 3, we first introduce the objectives used.

**Definitions.** Denote $\mathcal{G}$ as the garment surface, $\mathcal{X} \subset F^g \times F^g$ as the collection of all pairs of adjacent triangles on the garment. For variables defined on the current garment, we use $\cdot^s$ to represent the corresponding variable defined on the source garment (e.g., $v_i^s$ and $v_i$).

*3.4.1 Collision.* We adopt the collision barrier potential from IPC [Li et al. 2020] to handle collision. For completeness, we briefly repeat the formulation below. Given a triangular mesh $(V, F)$, denote $E$ as the edges of the mesh. Denote $\mathcal{T}$ as the set of all primitives (vertices, edges, faces), and $\mathcal{B} \subset \mathcal{T} \times \mathcal{T}$ as the set of all non-adjacent and non-incident primitive pairs. The intersection-free constraint boils down to enforcing the distance $d_k > 0$ between primitive pairs continuously in the optimization, for every pair $k \in C \subset \mathcal{B}$, where $C$ contains all non-incident point-triangle pairs and all non-adjacent edge-edge pairs in the mesh. Note that we only care about the self-collision of the garment and collisions between the garment and the avatar, so we do not include primitive pairs from the avatar itself.

The barrier potential is defined as

$$\mathcal{L}_{\text{contact}} := \sum_{k \in C} b(d_k), \tag{1}$$

where $b(\cdot) : \mathbb{R} \to \mathbb{R}$ is a $C^2$ smooth function that converges to infinity as the input tends to zero:

$$b(d) := \begin{cases} -(d-\hat{d})^2 \log\left(\frac{d}{\hat{d}}\right), & 0 < d < \hat{d} \\ 0, & d \geq \hat{d}. \end{cases}$$

We use $\hat{d} = 0.002$ and $w_{\text{contact}} = 10^8$ in our examples.

*Remark.* The barrier potential Equation (1) not only avoids intersections but also avoids triangles being degenerate, because in a degenerate triangle, one vertex overlaps with its opposite edge, i.e., the distance between the vertex and edge is zero. Thanks to this, we can safely assume that the triangle normals and edge tangents are always well-defined when constructing other loss terms.

*3.4.2 Surface shape.* The retargeted garment should preserve the surface shape of the input garment: e.g., a flat T-shirt should stay flat, wrinkles on the garment should stay as wrinkles, etc. On the other hand, some deformation should be allowed: e.g., the sleeves and trouser legs should elongate or shorten freely if needed, and the surface should rotate/translate freely to enable retargeting between avatars at different poses and scale freely to accommodate for body size change.

Inspired by [Araújo et al. 2023], we constrain the surface shape by penalizing the difference in deformations between every pair of adjacent triangles. For a pair of triangles $t_1 = \triangle v_1 v_2 v_3$ and $t_2 = \triangle v_2 v_1 v_4$ sharing the same edge $v_1 v_2$, the difference in deformations between the two triangles can be measured by considering their impact on the same vector $v_3^s v_4^s$. Suppose the deformation from $t_i^s$ to $t_i$ is $\tau_i$ $(i = 1, 2)$, which can be viewed as a $3 \times 3$ matrix, then the penalty is defined as

$$l_{\text{surf}}(t_1, t_2) := \left\| \tau_1 \frac{v_4^s - v_3^s}{\|v_4^s - v_3^s\|} - \tau_2 \frac{v_4^s - v_3^s}{\|v_4^s - v_3^s\|} \right\|^2,$$

where $\|\cdot\|$ is the Frobenius norm. The total similarity term is then defined as

$$\mathcal{L}_{\text{surf}} := \sum_{(t_1, t_2) \in \mathcal{X}} w_{\text{surf}}(t_1, t_2)\; l_{\text{surf}}(t_1, t_2), \tag{2}$$

where $w_{\text{surf}}(t_1, t_2)$ is the area of $t_1$ and $t_2$ on the source garment. The remaining question is how to compute the $\tau_i\, x$ for any vector $x \in \mathbb{R}^3$. We take $\tau_1$ as an example. Vectors $e_1 := v_2 - v_1$ and $e_2 := v_3 - v_1$ span the plane of triangle $t_1$, denote $n := e_1 \times e_2$ as the normal of $t_1$, then the deformation $\tau_1$ is uniquely defined by the mapping from $\{e_1^s,\ e_2^s,\ n^s\}$ to $\{e_1,\ e_2,\ n\}$. Suppose $x = c_1 e_1^s + c_2 e_2^s + c_3 n^s$ for some $c_i \in \mathbb{R},\ i = 1, 2, 3$, then $\tau_1 x = c_1 e_1 + c_2 e_2 + c_3 n$. $\{c_i\}$ can be computed by solving a linear system

$$[c_1,\ c_2,\ c_3]^\top = [e_1^s,\ e_2^s,\ n^s]^{-1}\, x.$$

The $\mathcal{L}_{\text{surf}}$ is defined in a way such that any affine transformation over the entire geometry does not change its value, which is critical when the target and source avatars have very different body ratios.

*Remark.* Our surface preservation term is different from the similarity energy used in [Araújo et al. 2023]. They treat $\tau_i$ as degrees of freedom and solve the nonlinear system by alternating steps between solving the vertex positions and transformations. They can accelerate the convergence using this method since each step becomes a quadratic problem that is easy to solve. However, in our case, due to the high nonlinearity of the collision potential in Section 3.4.1, we cannot gain better convergence by alternating steps, so instead, we define the energy using purely vertex positions and solve the complete problem using the Projected Newton's method.

*3.4.3 Positional.* The positional constraint aims to preserve the position of the garment relative to the underlying avatar, e.g., the sleeve (trouser leg) length should accommodate the change of arm (leg) length. Without a correspondence between the avatar surfaces, the relative location preservation in [Brouet et al. 2012] cannot be used. Instead, we use the position information relative to the skeleton bones. For a point $v^s$ on the source garment, suppose the bone in the source skeleton closest to $v^s$ is bone $e \in E^b$. We project $v^s$ onto $e$ and obtain the barycentric coordinate $w^s$. Similarly, for the corresponding point $v$ on the current garment, we project it to the same bone $e$ and obtain the current barycentric coordinate $w$. The positional constraint for this point is defined as

$$l_{\text{pos}}(v) := (w^s - w)^2.$$

Note that during the inflation step, the node positions of the current skeleton are a linear interpolation with coefficient $t$ (Section 3.5) between the source and target skeleton nodes. The total positional term is then defined as

$$\mathcal{L}_{\text{pos}} := \sum_{v \in \partial \mathcal{G}} w_{\text{pos}}(v) l_{\text{pos}}(v), \tag{3}$$

where $\partial \mathcal{G}$ represents the boundary of the garment surface, $w_{\text{pos(v)}}$ is the average length of edges at $v$ on the source garment. We use $w_{\text{pos}} = 10$ to prioritize the positional constraint.

*3.4.4 Fit.* We opt to enforce a similar level of tightness from the original garment in the retargeted garment. To measure the distance from the points on the garment to the avatar, we convert the avatar mesh to a Signed Distance Field (SDF). By evaluating the SDF on a point $p$ on the current garment, we obtain the distance $D(p)$ from $p$ to the avatar mesh. The fit loss can then be defined as

$$\mathcal{L}_{\text{fit}} = \int_{\mathcal{G}_{\text{fit}}} D(p)^2 \, dp, \tag{4}$$

where $\mathcal{G}_{\text{fit}} \subset \mathcal{G}$ is the part of the current garment surface that should fit, which can be decided based on the distance between the source avatar and garment. Note that we only include $\mathcal{L}_{\text{fit}}$ in the last step of our algorithm, i.e. when the target avatar mesh has been inflated to its original shape, since the intermediate avatar during inflation may not be a closed mesh that corresponds to a meaningful volume. The choice of $w_{\text{fit}}$ depends on the desired fit of the garment, we use $w_{\text{fit}} = 2$ in our examples. We study the influence of $w_{\text{fit}}$ in Section 4.6 and show the ability to control the fit of the retargeted garments.

*3.4.5 Curve Shape.* Apart from the distortion of the surface shape, the distortion of seamlines and boundary curves (e.g., cuffs and hems) also significantly influences the visual effects. Therefore, we enforce extra constraints to regularize the geometry of curve loops.

By the fundamental theorem of curves [Carmo 2016], the shape of a smooth 3D curve, with *non-zero curvature*, is uniquely determined by its curvature and torsion. Therefore, we can constrain the boundary curve shape by penalizing the change of the discretized curvature and torsion:

$$\mathcal{L}_{\text{curve}} := \mathcal{L}_{\text{curvature}} + \mathcal{L}_{\text{torsion}}.$$

Similar to $\mathcal{L}_{\text{surf}}$, we carefully define $\mathcal{L}_{\text{curve}}$ such that it is invariant under rotation, translation, and uniform scaling. However, it is undesirable to allow anisotropic scaling in $\mathcal{L}_{\text{curve}}$ – a circular cuff should not become an ellipse freely unless the fit term wants it to fit tightly to an elliptical wrist.

**Definition.** Each boundary curve of the garment mesh is supposed to be a closed polyline. Denote the vertices of the polyline as $v_1, v_2, \dots, v_m$, where $v_m = v_1$, and edge directions $e_i := v_{i+1} - v_i$ and $\hat{e}_i := e_i / \|e_i\|_{L_2}$ for $i = 1, \dots, m-1$.

**Curvature.** The curvature for the polyline under the discrete setting can be quantified using the dot product $\alpha_i$ between the two incident edges at every vertex $v_i$:

$$\alpha_i := -\hat{e}_{i-1} \cdot \hat{e}_i \quad (i = 1, \dots, m-1).$$

The curvature term penalizes the change of $\cos \alpha$

$$\mathcal{L}_{\text{curvature}} := \sum_{i=1}^{m-1} w_i (\alpha_i^s - \alpha_i)^2, \tag{5}$$

where $\alpha_i^s$, $\alpha_i$ are the dot products on the source and current garment, respectively, $w_i := (\|e_{i-1}^s\|_{L_2} + \|e_i^s\|_{L_2})/2$.

**Torsion.** Unlike curvature, the computation of discrete torsion requires every three consecutive edges. We define $v_0 := v_{m-1}$ for convenience. Define

$$P(v, w) := v - (w \cdot v)\, w,$$

which projects vector $v$ to the plane with unit normal $w$. We project edges $\hat{e}_{i-1}$, $\hat{e}_{i+1}$ to the plane with normal $e_i$, and compute

$$\beta_i := P(\hat{e}_{i+1}, \hat{e}_i) \cdot P(-\hat{e}_{i-1}, \hat{e}_i),$$
$$\gamma_i := (P(\hat{e}_{i+1}, \hat{e}_i) \times P(-\hat{e}_{i-1}, \hat{e}_i)) \cdot \hat{e}_i,$$

then the torsion term is defined as

$$\mathcal{L}_{\text{torsion}} := \sum_{i=1}^{m-1} w_i (\beta_i^s - \beta_i)^2 + w_i (\gamma_i^s - \gamma_i)^2,$$

where $\beta_i^s$, $\gamma_i^s$ are the values computed on the source garment.

*Remark.* On the one hand, $\mathcal{L}_{\text{curve}}$ is invariant under rigid body transformations and scaling; on the other hand, forcing $\mathcal{L}_{\text{curve}} = 0$ gives us a curve with the same shape up to a scaling and rigid body transformation.

## 3.5 Step 2. Avatar Inflation

In the inflation step of the optimization, we introduce one variable $t \in [0, 1]$ to linearly interpolate between the shrunk avatar vertices $V^{shink}$ and the target avatar vertices $V^{ta}$

$$V^a(t) := t\, V^{ta} + (1-t)\, V^{shink},$$

which is used as the current avatar shape in collision handling. Similarly, the current skeleton used in Section 3.4.3 is

$$V^b(t) := t\, V^{tb} + (1-t)\, V^{sb}.$$

We can consider $t = 0$ at the beginning of the optimization, and the inflation step is essentially trying to enforce an equality constraint of $t = 1$ in the optimization under collision constraints. We use the Augmented Lagrangian (AL) method to enforce this equality constraint, in which the loss becomes

$$\mathcal{L}_{\text{AL}} := \frac{\lambda_1}{2}(1-t)^2 + \lambda_2(1-t),$$

where $\lambda_1$, $\lambda_2$ are updated in the outer loop of AL to push $t$ towards 1. Please refer to the supplement for details. The total loss in Step 2 is then defined as

$$\begin{aligned}\mathcal{L}_{\text{Step 2}} := \mathcal{L}_{\text{AL}} + \mathcal{L}_{\text{surf}} + w_{\text{pos}}\, \mathcal{L}_{\text{pos}} \\ + w_{\text{contact}}\, \mathcal{L}_{\text{contact}} + \mathcal{L}_{\text{curve}}.\end{aligned} \tag{6}$$

Note that we do not include $\mathcal{L}_{\text{fit}}$ in Step 2 for two reasons: (a) The fit objective fights against the AL penalty on $t$ and slows down the convergence – the former term pushes the garment toward the avatar, while the latter forces the avatar to inflate. (b) The SDF used in $\mathcal{L}_{\text{fit}}$ changes as the avatar geometry changes, thus, the derivatives of the objective require differentiating through the mesh to the SDF conversion process, which is potentially non-differentiable.

Once the optimization can snap from the current $t$ value to $t = 1$ without causing intersections, even though the gradient is not close to zero, we snap $t$ to 1 and move to Step 3. When snapping $t$ to 1, it is critical to perform CCD and guarantee that this change in $t$ is intersection-free *continuously*, so that the garment does not pass through any thin features of the avatar.

## 3.6 Step 3. Garment Fit

In this step, we treat $t = 1$ in Step 2 as a constant, i.e., the target avatar is fully inflated to its original shape, and we keep its shape fixed in this step of the optimization. Now that the avatar shape is fixed, we convert the mesh to an SDF using OpenVDB [Museth 2013] to use in the fit term. We replace the $\mathcal{L}_{\text{AL}}$ with $\mathcal{L}_{\text{fit}}$ and minimize the following objective until convergence.

$$\begin{aligned}\mathcal{L}_{\text{Step 3}} := w_{\text{fit}}\mathcal{L}_{\text{fit}} + \mathcal{L}_{\text{surf}} + w_{\text{pos}}\, \mathcal{L}_{\text{pos}} \\ + w_{\text{contact}}\, \mathcal{L}_{\text{contact}} + \mathcal{L}_{\text{curve}}.\end{aligned} \tag{7}$$

Please refer to the supplement for the summary of weights.

## 3.7 Implementation

We use the Projected Newton's method [Li et al. 2020; Teran et al. 2005] to solve the nonlinear system in the inner loop of the Augmented Lagrangian method. We implemented our algorithm using Sympy [Meurer et al. 2017] for generating efficient C++ code to compute derivatives, IPC Toolkit [Ferguson et al. 2020] for computing contact potentials and robust CCD, OpenVDB [Museth 2013] for SDF evaluation, and Pardiso [Alappat et al. 2020; Bollhöfer et al. 2019, 2020] for solving linear systems. All our experiments are run on a MacBook Pro with the Apple M3 Max CPU, limited to 16 threads and 64GB of memory. Please refer to the supplement for additional algorithm details.

# 4 EVALUATION

To demonstrate the effectiveness of our method, we use it to retarget garments between both human models and game avatars, and compare our method with the fitting tool in Roblox Studio. We also include ablation studies on objectives.

## 4.1 Game Avatars

In Figure 11, we demonstrate our method by retargeting a collection of 6 garments designed on a mannequin to 9 avatars, in total 54 combinations are obtained. Both our garments and avatars are highly diverse. Our garments include a full-body jumpsuit that fits tightly to the body, a puffer coat that fits loosely, and a skirt that naturally drapes. Our avatars include skeletons made of weakly connected bones, elementals made of floating stones and crystals, giants, and a fox girl with unrealistic body ratios, a wolf man with double heads, and robotics with intricate wires and chips. Our method has shown robustness in these examples despite the variation in body shapes and the complexity of the models. Although we do not consider physics simulations in the retargeting, the draping of skirts, wrinkles of the sweaters, and bumps on the puffer jackets are visually preserved. For a more challenging example, we retarget the puffer jacket from a mannequin to a dinosaur (Figure 4).

The 54 combinations' average runtime is 97 seconds, with the maximum being 432 seconds. The number of vertices in avatars is between 2000 and 6000, and the number of vertices in garments is between 600 and 6000.

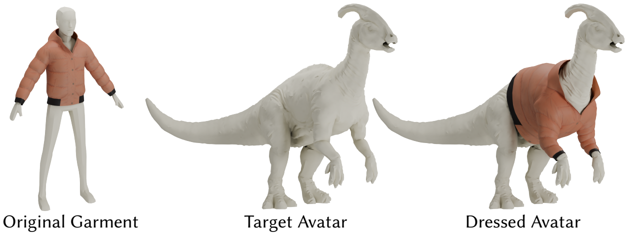

Fig. 4. Retargeting a puffer jacket from human to a dinosaur.

## 4.2 Comparison with Roblox Studio

Roblox's creator tools provide an accessory fitting tool [Roblox 2025] that retargets the garment dressed on one avatar to another avatar. The tool asks users to manually create a cage mesh to cover each garment and avatar tightly (Figure 5), which requires much more manual effort than embedding a skeleton. We perform a comparison with their method on the same set of caged avatars and garments. As shown in Figure 5, our method has less distortion around the sleeves and waist and better preserves the overall skirt shape. In addition, our method supports controllable tightness as shown in Figure 8, providing more flexibility to users.

## 4.3 Human Avatars

In Figure 12, we show the effectiveness of our method on retargeting between SMPL human avatars [Loper et al. 2015], which has been extensively studied in prior works [Brouet et al. 2012; Lin et al. 2024; Pons-Moll et al. 2017; Wang et al. 2018]. Our method generates high-quality garments for the target human bodies with different body shapes. Since these prior works do not have open-source code, we are not able to conduct a comparison on the same avatar and garment.

## 4.4 Multiple Layers of Garments

Since we do not assume the target avatar to be a volumetric object, our method can be easily extended to retarget multiple layers of garments on one avatar and maintain intersection-free between layers. Once a garment is retargeted to the avatar, we treat it as part of the avatar and rerun our algorithm to wear a new layer of garment over it. We show one such example in Figure 6 with three layers of garments. For simplicity, we treat the inner layers as fixed when we optimize the subsequent layer. However, it is possible to optimize the inner layers as well to accommodate the outer layer, e.g., if the outer layer is a tight garment that compresses the inner layers, which we leave as future work.

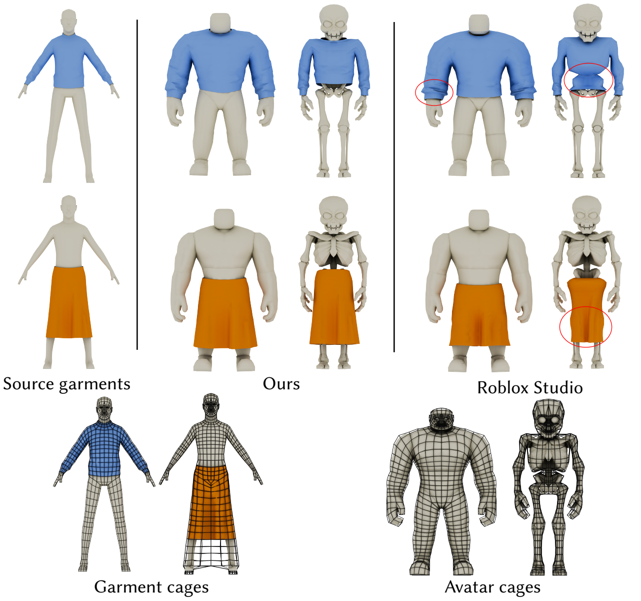

Fig. 5. Comparisons between our method and Roblox Studio. Top: The retargeting results of both methods. Bottom: Manually designed cage meshes (black) are needed as input for the Roblox Studio method. Artifacts are marked in red show excessive fit in the sleeves and overly tight regions elsewhere.

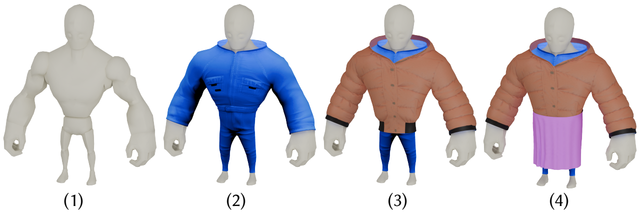

Fig. 6. Retargeting multiple layers of garments on one avatar. (1) Target avatar. (2) Target avatar wearing a jumpsuit. (3) Target avatar wearing the puffer jacket over the jumpsuit. (4) Wearing a skirt on top of (3).

## 4.5 Stress test

As a stress test for our method, we retarget a jumpsuit from the mannequin to an avatar with much longer arms (Figure 7). We visualize the element-wise $\mathcal{L}_{\text{surf}}$ on the optimized garment to show the geometric distortion.

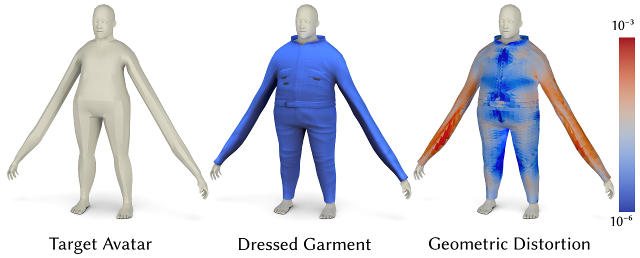

Fig. 7. **Stress test.** Retargeting the jumpsuit to an avatar with long arms. The geometric distortion is visualized on the right in a logarithmic scale.

## 4.6 Ablations

The optimization minimizes a collection of objectives that balance each other; a reasonable choice of weights is needed to obtain the desired results.

**Fit.** The weight $w_{\text{fit}}$ controls the tightness of the garment on the target avatar. We show the effect of $w_{\text{fit}}$ on the optimized garment in Figure 8. With small $w_{\text{fit}}$, the optimized garment tends to preserve the original shape, therefore the body part remains straight even though the target avatar has an inverted triangle body shape; with large $w_{\text{fit}}$, the optimized garment fits more tightly on the target avatar to accommodate the slim waist.

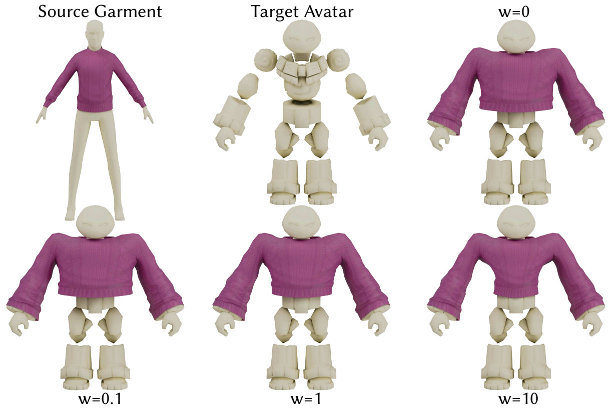

Fig. 8. Influence of $w_{\text{fit}}$ on the retargeting.

**Positional.** The positional constraint is based on the skeletons embedded in the avatars, therefore, by perturbing the node positions of an existing skeleton, one can obtain different results on the same avatar. In Figure 9, we show that by modifying the skeleton nodes corresponding to the neck and waist, one can change the height of the collar and hem. The original garment is shown in Figure 4.

**Curve Shape Constraint.** The $\mathcal{L}_{\text{curve}}$ objective allows one to preserve the shape of seamlines or boundary curves in addition to preserving the surface shape. As shown in Figure 10, without the curve shape constraint, the neck of the retargeted vest fails to retain its "V" shape due to the widened shoulder of the target avatar; with such constraint, the "V" neck is well-preserved without sacrificing the overall surface shape.

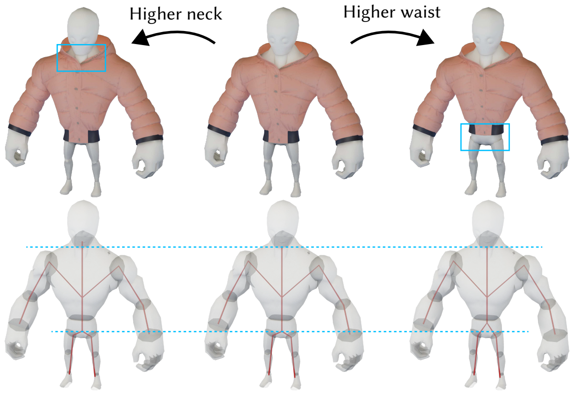

Fig. 9. Influence of skeleton node positions on the retargeting for the same avatar. The embedded skeleton (bottom) and the corresponding retargeted garment (top) are shown in each column.

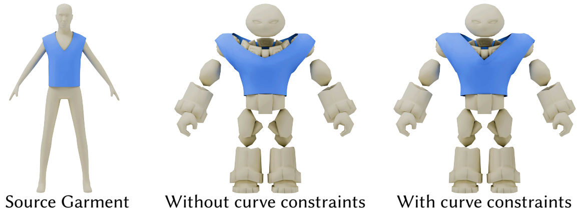

Fig. 10. Influence of $\mathcal{L}_{\text{curve}}$ on the retargeting. The undressed avatar is shown in Figure 8. The "V" shape at the neck is better preserved with the constraint.

## 5 CONCLUSIONS

We introduce a robust and flexible pipeline to automatically retarget garments from one avatar to many others, which greatly reduces manual efforts in applications including game design and animations. It is effective not only for human avatars with different body shapes but also for game avatars with possibly non-realistic body shapes, floating body parts, or voids, where correspondence between avatars with reasonable quality is impossible. We demonstrate the effectiveness of our method on all-pair combinations from a collection of avatars and garments with high diversity. The two key components of our method are a novel projection scheme that produces an intersection-free initialization for the subsequent optimizations, and a combination of objectives that meet the criteria of garment retargeting.

Our method has several limitations. First, we rely on the animation skeletons to position the garment onto the target avatar, assuming the target and source avatars have the same number of arms and legs and can be put in the same pose. Retargeting from humanoid avatars to non-humanoids like tetrapods or even octopuses can be an interesting topic given the rise in generative models. Second, our approach does not include physics simulations and thus ignores the physics of the garments. However, as shown by the figures, the output garments retain the physical effects like wrinkles and draping that appear on the original design. Lastly, our method

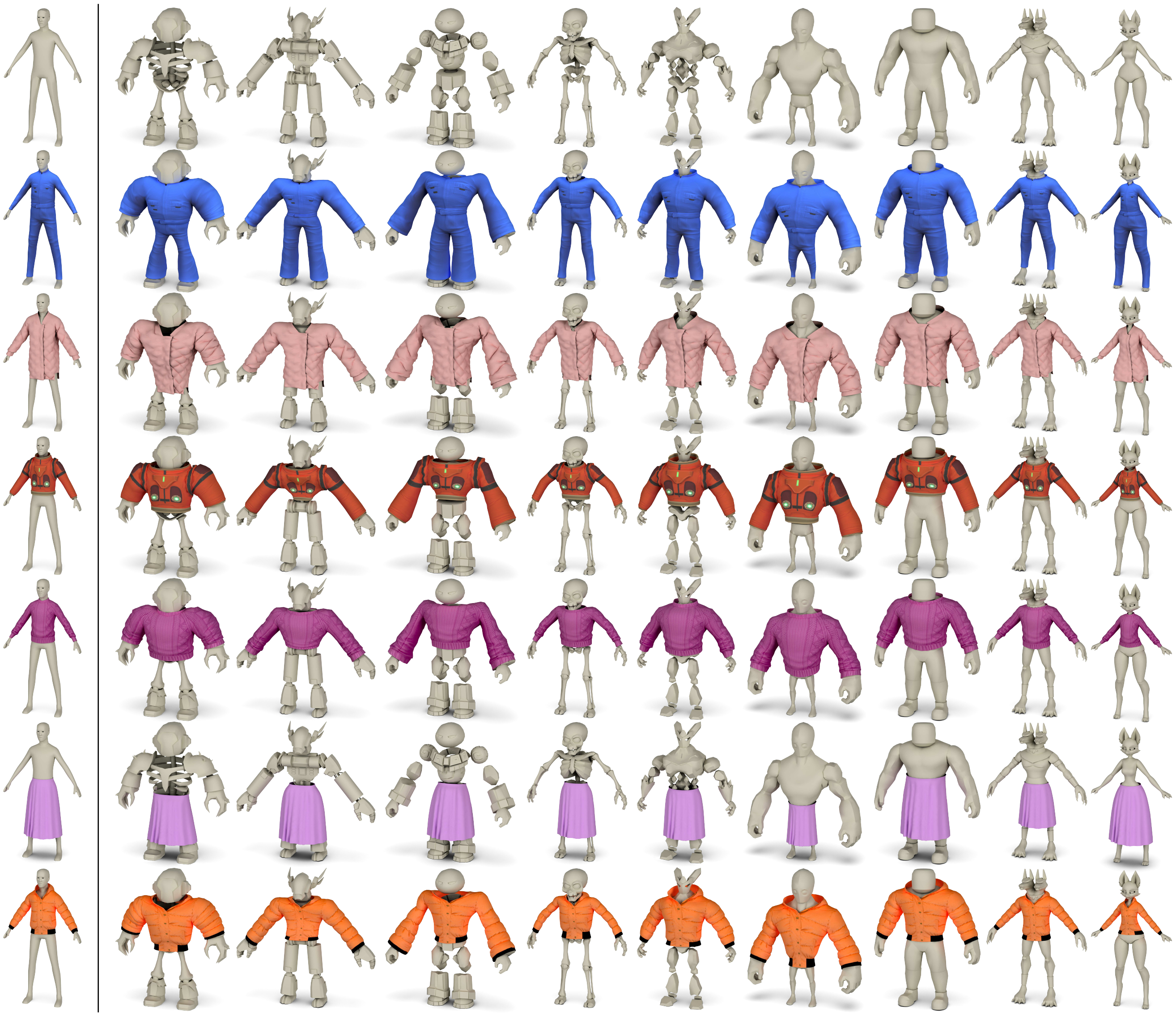

Fig. 11. Garments designed on a mannequin (left) are dressed onto various avatars. Undressed avatars are shown on top.

is limited to manifold meshes, which puts restrictions on the design process.

## ACKNOWLEDGMENTS

We would like to extend our thanks to the Layered Clothing Quality team at Roblox for their help and support, especially Kelvin Lau, Adam Burr, Ervin Teng, and Murilo Coutinho. We would also like to thank the Roblox Creator, BlueFoxClaw, for the use of their Anime Fox Girl model in our examples.

This work was also partially supported by the NSF CAREER award under Grant No. 1652515, the NSF grants OAC-2411349, OAC-1835712, CHS-1908767, CHS-1901091, IIS-2313156, a Sloan Fellowship, and a gift from Adobe Research.

## REFERENCES

Christie Alappat, Achim Basermann, Alan R. Bishop, Holger Fehske, Georg Hager, Olaf Schenk, Jonas Thies, and Gerhard Wellein. 2020. A Recursive Algebraic Coloring Technique for Hardware-Efficient Symmetric Sparse Matrix-Vector Multiplication. *ACM Trans. Parallel Comput.* 7, 3, Article 19 (June 2020), 37 pages. https://doi.org/10.1145/3399732

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2023. *SCAPE: Shape Completion and Animation of People* (1 ed.). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3596711.3596797

Chrystiano Araújo, Nicholas Vining, Silver Burla, Manuel Ruivo de Oliveira, Enrique Rosales, and Alla Sheffer. 2023. Slippage-Preserving Reshaping of Human-Made 3D Content. *ACM Transaction on Graphics* 42, 6 (2023). https://doi.org/10.1145/3618391

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, Article 50 (jul 2016), 11 pages. https://doi.org/10.1145/2897824.2925896

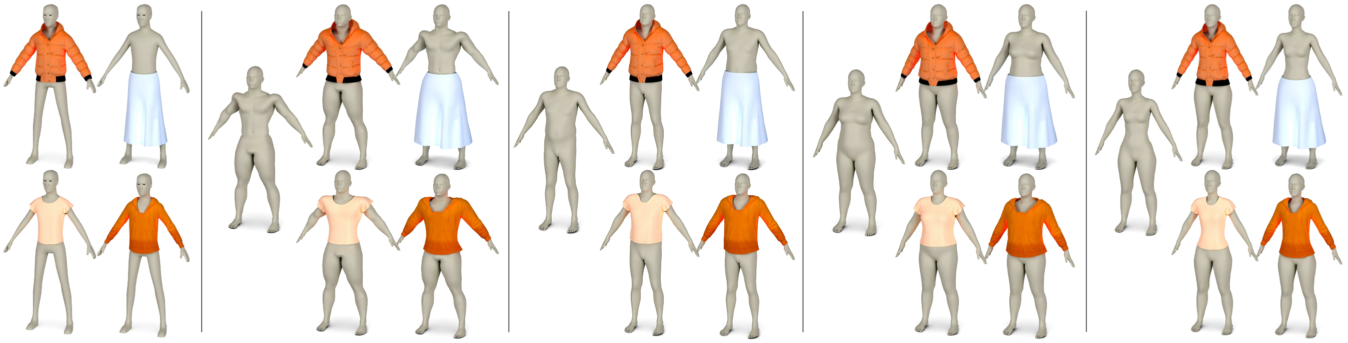

Fig. 12. Garments retargeting from a mannequin (left) onto SMPL human models [Loper et al. 2015].

Matthias Bollhöfer, Aryan Eftekhari, Simon Scheidegger, and Olaf Schenk. 2019. Large-scale Sparse Inverse Covariance Matrix Estimation. *SIAM Journal on Scientific Computing* 41, 1 (2019), A380–A401. https://doi.org/10.1137/17M1147615 arXiv:https://doi.org/10.1137/17M1147615

Matthias Bollhöfer, Olaf Schenk, Radim Janalik, Steve Hamm, and Kiran Gullapalli. 2020. State-of-the-Art Sparse Direct Solvers. (2020), 3–33. https://doi.org/10.1007/978-3-030-43736-7_1

Remi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. 2012. Design preserving garment transfer. *ACM Trans. Graph.* 31, 4, Article 36 (July 2012), 11 pages. https://doi.org/10.1145/2185520.2185532

Manfredo P. Do Carmo. 2016. *Differential Geometry of Curves and Surfaces* (2nd ed.). Dover Publications.

Courtney Chun, Jose Velasquez, and Haixiang Liu. 2023. Creating the Art-Directed Groom for Legend in Disney's "Strange World". In *ACM SIGGRAPH 2023 Talks* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 7, 2 pages. https://doi.org/10.1145/3587421.3595441

Enric Corona, Albert Pumarola, Guillem Alenya, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. SMPLicit: Topology-Aware Generative Model for Clothed People. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11875–11885.

Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. 2010. Generalized biped walking control. *ACM Trans. Graph.* 29, 4, Article 130 (July 2010), 9 pages. https://doi.org/10.1145/1778765.1781156

Zijian Dong, Xu Chen, Jinlong Yang, Michael J Black, Otmar Hilliges, and Andreas Geiger. 2023. Ag3d: Learning to generate 3d avatars from 2d image collections. In *Proceedings of the IEEE/CVF international conference on computer vision*. 14916–14927.

Zachary Ferguson et al. 2020. *IPC Toolkit*. https://ipc-sim.github.io/ipc-toolkit/

Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. 2021. Parser-free virtual try-on via distilling appearance flows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8485–8493.

Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen. 2013. Flexible muscle-based locomotion for bipedal creatures. *ACM Trans. Graph.* 32, 6, Article 206 (Nov. 2013), 11 pages. https://doi.org/10.1145/2508363.2508399

Artur Grigorev, Giorgio Becherini, Michael Black, Otmar Hilliges, and Bernhard Thomaszewski. 2024. Contourcraft: Learning to resolve intersections in neural multi-garment simulations. In *ACM SIGGRAPH 2024 Conference Papers*. 1–10.

Artur Grigorev, Bernhard Thomaszewski, Michael J Black, and Otmar Hilliges. 2023. HOOD: Hierarchical Graphs for Generalized Modelling of Clothing Dynamics. *Computer Vision and Pattern Recognition (CVPR)*.

Peng Guan, Loretta Reiss, David A Hirshberg, Alexander Weiss, and Michael J Black. 2012. Drape: Dressing any person. *ACM Transactions on Graphics (ToG)* 31, 4 (2012), 1–10.

Jingfan Guo, Jie Li, Rahul Narain, and Hyun Soo Park. 2021. Inverse Simulation: Reconstructing Dynamic Geometry of Clothed Humans via Optimal Control. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. 2024. DressCode: Autoregressively Sewing and Generating Garments from Text Guidance. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–13.

Chris Hecker, Bernd Raabe, Ryan W. Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. 2008. Real-time motion retargeting to highly varied user-created morphologies. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) *(SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 27, 11 pages. https://doi.org/10.1145/1399504.1360626

Jonathan Hoffman, Te Hu, Paul Kanyuk, Stephen Marshall, George Nguyen, Hope Schroers, and Patrick Witting. 2023. Creating Elemental Characters: From Sparks to Fire. In *ACM SIGGRAPH 2023 Talks* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 51, 2 pages. https://doi.org/10.1145/3587421.3595467

Michael A. Hopkins, Georg Wiedebach, Kyle Cesare, Jared Bishop, Espen Knoop, and Moritz Bächer. 2024. Interactive Design of Stylized Walking Gaits for Robotic Characters. *ACM Trans. Graph.* 43, 4, Article 137 (July 2024), 16 pages. https://doi.org/10.1145/3658227

Yu Jiang, Zhipeng Li, Mufei He, David Lindlbauer, and Yukang Yan. 2023. HandAvatar: Embodying Non-Humanoid Virtual Avatars through Hands. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 309, 17 pages. https://doi.org/10.1145/3544548.3581027

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Trans. Graph. (SIGGRAPH)* 39, 4, Article 49 (2020).

Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Trans. Graph. (SIGGRAPH)* 40, 4, Article 170 (2021).

Tianyu Li, Jungdam Won, Alexander Clegg, Jeonghwan Kim, Akshara Rai, and Sehoon Ha. 2023. ACE: Adversarial Correspondence Embedding for Cross Morphology Motion Retargeting from Human to Nonhuman Characters. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) *(SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 46, 11 pages. https://doi.org/10.1145/3610548.3618255

Yifei Li, Hsiao-yu Chen, Egor Larionov, Nikolaos Sarafianos, Wojciech Matusik, and Tuur Stuyck. 2024. DiffAvatar: Simulation-Ready Garment Optimization with Differentiable Simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4368–4378.

Siyou Lin, Zhe Li, Zhaoqi Su, Zerong Zheng, Hongwen Zhang, and Yebin Liu. 2024. LayGA: Layered Gaussian Avatars for Animatable Clothing Transfer. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 37, 11 pages. https://doi.org/10.1145/3641519.3657501

Chen Liu, Weiwei Xu, Yin Yang, and Huamin Wang. 2024. Automatic Digital Garment Initialization from Sewing Patterns. *ACM Trans. Graph.* 43, 4, Article 74 (July 2024), 12 pages. https://doi.org/10.1145/3658128

Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. 2023. Towards Garment Sewing Pattern Reconstruction from a Single Image. *ACM Trans. Graph.* 42, 6, Article 200 (Dec. 2023), 15 pages. https://doi.org/10.1145/3618319

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. 2020. Learning to Dress 3D People in Generative Clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. 2017. SymPy: symbolic computing in Python. *PeerJ Computer Science* 3 (Jan. 2017), e103. https://doi.org/10.7717/peerj-cs.103

Ken Museth. 2013. VDB: High-Resolution Sparse Volumes with Dynamic Topology. *ACM Trans. Graph.* 32, 3, Article 27 (jul 2013), 22 pages. https://doi.org/10.1145/2487228.2487235

Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. 2020. TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).* 10975–10985.

Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa A Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational pattern making from 3D garment models. *ACM Trans. Graph.* 41, 4 (2022), 157–1.

Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. 2017. ClothCap: seamless 4D clothing capture and retargeting. *ACM Trans. Graph.* 36, 4, Article 73 (jul 2017), 15 pages. https://doi.org/10.1145/3072959.3073711

Roblox. 2025. Accessory Fitting Tool | Documentation - Roblox Creator Hub — create.roblox.com. https://create.roblox.com/docs/art/accessories/accessory-fitting-tool. [Accessed 15-01-2025].

Igor Santesteban, Miguel A. Otaduy, and Dan Casas. 2022. SNUG: Self-Supervised Neural Dynamic Garments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 8140–8150.

Masahiro Sekine, Kaoru Sugita, Frank Perbet, Björn Stenger, and Masashi Nishiyama. 2014. Virtual fitting by single-shot body shape estimation. In *Int. Conf. on 3D Body Scanning Technologies*, Vol. 406. Citeseer, 413.

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '05)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/1073368.1073394

Huamin Wang. 2018. Rule-free sewing pattern adjustment with precision and efficiency. *ACM Trans. Graph.* 37, 4, Article 53 (jul 2018), 13 pages. https://doi.org/10.1145/3197517.3201320

Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popović, and Niloy J. Mitra. 2018. Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.* 37, 6, Article 203 (Dec. 2018), 13 pages. https://doi.org/10.1145/3272127.3275074

Katja Wolff, Philipp Herholz, Verena Ziegler, Frauke Link, Nico Brügel, and Olga Sorkine-Hornung. 2023. Designing Personalized Garments with Body Movement. *Computer Graphics Forum* 42, 1 (Jan. 2023), 180–194. https://doi.org/10.1111/cgf.14728

Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. 2023. ECON: Explicit Clothed Humans Optimized via Normal Integration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 512–523.

Katsu Yamane, Yuka Ariki, and Jessica Hodgins. 2010. Animating non-humanoid characters with human motion data. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Madrid, Spain) *(SCA '10)*. Eurographics Association, Goslar, DEU, 169–178.

Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. 2020. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 7850–7859.

Yang Zheng, Qingqing Zhao, Guandao Yang, Wang Yifan, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, et al. 2025. Physavatar: Learning the physics of dressed 3d avatars from visual observations. In *European Conference on Computer Vision.* Springer, 262–284.