

4D-DRESS: A 4D Dataset of Real-World Human Clothing With Semantic Annotations

Wenbo Wang^{*1} Hsuan-I Ho^{*1} Chen Guo¹ Boxiang Rong¹ Artur Grigorev^{1,2}
 Jie Song¹ Juan Jose Zarate^{†1} Otmar Hilliges¹

Department of Computer Science, ETH Zürich
 Max Planck Institute for Intelligent Systems, Tübingen

<https://ait.ethz.ch/4d-dress>

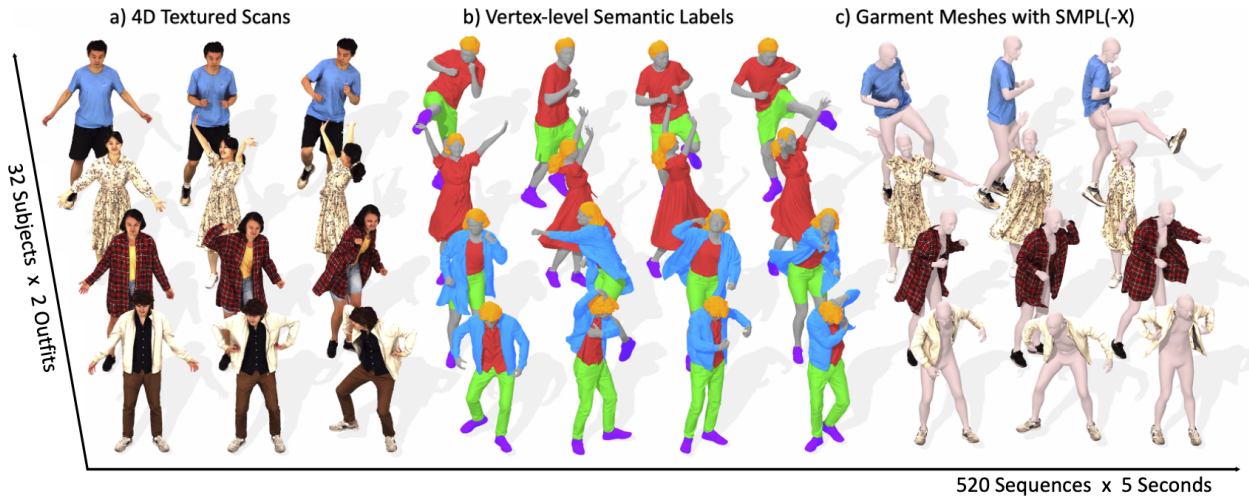


Figure 1. **Overview of 4D-DRESS.** We propose the first real-world 4D dataset of human clothing, capturing 64 human outfits in more than 520 motion sequences. These sequences include a) high-quality 4D textured scans; for each scan, we annotate b) vertex-level semantic labels, thereby obtaining c) the corresponding garment meshes and fitted SMPL(-X) body meshes.

Abstract

The studies of human clothing for digital avatars have predominantly relied on synthetic datasets. While easy to collect, synthetic data often fall short in realism and fail to capture authentic clothing dynamics. Addressing this gap, we introduce 4D-DRESS, the first real-world 4D dataset advancing human clothing research with its high-quality 4D textured scans and garment meshes. 4D-DRESS captures 64 outfits in 520 human motion sequences, amounting to 78k textured scans. Creating a real-world clothing dataset is challenging, particularly in annotating and segmenting the extensive and complex 4D human scans. To address this, we develop a semi-automatic 4D human parsing pipeline. We efficiently combine a human-in-the-loop process with automation to accurately label 4D scans in di-

verse garments and body movements. Leveraging precise annotations and high-quality garment meshes, we establish several benchmarks for clothing simulation and reconstruction. 4D-DRESS offers realistic and challenging data that complements synthetic sources, paving the way for advancements in research of lifelike human clothing.

1. Introduction

Human clothing is crucial in various applications such as 3D games, animations, and virtual try-on. Researchers are actively investigating algorithms for clothing reconstruction [14, 26, 36] and simulation [4, 5, 17], to achieve realistic clothing behavior, enhance user engagement, and enable cross-industry applications. These algorithms are frequently developed and assessed using synthetic datasets [3, 7, 57], since they comprise a) meshes covering various garment types and outfits and b) parametric body mod-

^{*} Equal contributors [†] Corresponding author

Dataset	# of Outfits	# of Frames	Data Format	Textured	Semantic Labels	Loose Garments
TailorNet [37]	9	5.5k	SMPL + Garments		✓	
ReSynth [35]	24	30k	SMPLX + Point Clouds			✓
CLOTH3D [3]	8.5k	2.1M	SMPL + Garments	✓	✓	✓
CLOTH4D [57]	1k	100k	Mesh + Garments	✓	✓	✓
BEDLAM [7]	111	380k	SMPL-X + Garments	✓	✓	✓
D-LAYERS [43]	5k	700k	SMPL + Garments		✓	✓
BUFF [55]	6	14k	Scans + SMPL	✓		
CAPE [34]	15	140k	SMPL+D			
ActorsHQ [25]	8	39k	Scans			✓
X-Humans [44]	20	35k	Scans + SMPL(-X)	✓		
4DHumanOutfit [2]	14	459k	Scans + SMPL	✓		✓
4D-DRESS (Ours)	64	78k	Scans + SMPL(-X) + Garments	✓	✓	✓

Table 1. **Summary of 4D clothed human datasets.** The datasets highlighted in gray color are synthetic datasets while the others are real-world scans. # of Outfits: number of outfits included; # of Frames: total number of 3D human frames; Data Format: 3D representations of human bodies and garments; Textured: with textured map or not; Semantic Labels: with semantic labels for clothing or not; Loose Garments: containing challenging loose clothing such as dresses or not. 4D-DRESS demonstrates outstanding features against others.

els with diverse motions. While synthetic datasets lead in outfit quantity and the number of frames provided (refer to Tab. 1), there also presents a significant challenge in bridging the domain gap between the synthetic and real garments. Despite the recently released real-world 4D human datasets such as X-Humans [44], ActorsHQ [25], and 4DHumanOutfit [2], a key limitation persists: they lack accurately segmented garment meshes, offering only raw human scans. Moreover, these datasets are limited in the number of loose garments (e.g., jackets and dresses) or dynamic motions, which reduces their applicability as test benches. These challenges highlight the need for a real-world 4D dataset that provides semantic annotations and captures diverse garments across various body motions.

In this work, we contribute 4D-DRESS, the first real-world dataset of human clothing with 4D semantic segmentation. We aim to provide an evaluation testbench with real-world data for tasks related to human clothing in computer vision and graphics. We capture over 520 human motion sequences featuring 64 distinct real-world human outfits in a high-end multi-view volumetric capture system, similar to the one used in [12]. The complete dataset comprises a total of 78k frames, each composed of an 80k-face triangle mesh, a 1k resolution textured map, and a set of 1k resolution multi-view images. As illustrated in Fig. 1, we provide a) high-quality 4D textured scans, b) vertex-level semantic labels for various clothing types, such as upper, lower, and outer garments, and c) garment meshes along with their registered SMPL(-X) body models.

Capturing real-world 4D sequences of humans wearing various clothing and performing diverse motions requires dedicated high-end capture facilities. Moreover, processing these clips into accurately annotated and segmented 4D human scans presents significant challenges. To develop our

dataset, we tackled the task of labeling 78k high-resolution meshes at the vertex level. Given that the mesh topologies of consecutive frames do not inherently correspond, consistently propagating 3D vertex labels from one frame to the next is non-trivial. While previous methods [6, 38] attempted to fit a fixed-topology parametric body model to the scans, these template-based approaches still struggle with scenarios such as a jacket being lifted to reveal a shirt or the emergence of new vertices on a flowing coat as illustrated in the example shown in Fig. 3. Consequently, we opted for an alternative approach. We developed a semi-automatic and template-free 4D human parsing pipeline. Leveraging semantic maps from a 2D human parser [16] and a segmentation model [29], we extended these techniques to 4D, considering both multi-view and temporal consistency. Our pipeline accurately assigns vertex labels without manual intervention in 96.8% of frames. Within the remaining scans, only 1.5% of vertices require further rectification, addressed via a human-in-the-loop process.

The quality of the ground-truth data in 4D-DRESS allows us to establish several evaluation benchmarks for diverse tasks, including clothing simulation, reconstruction, and human parsing. Our evaluation and analysis demonstrate that 4D-DRESS offers realistic and challenging human clothing that cannot be readily modeled by existing algorithms, thereby opening avenues for further research. In summary, our contributions include:

- the first real-world 4D human clothing dataset comprising 4D textured scans, vertex-level semantic labels, garment meshes, and corresponding parametric body meshes.
- a semi-automatic and template-free 4D human parsing pipeline for efficient data annotation.
- evaluation benchmarks showing the utility of our dataset.

2. Related Work

4D clothed human dataset. Datasets featuring clothed humans can be divided into two categories. Firstly, synthetic datasets [3, 7, 35, 37, 43, 57] create large volume of synthetic data using graphic engines [48] and simulation tools [11] (Tab. 1 top). These datasets are easy to scale with ground truth semantic labels available by design. However, they often lack realism in human appearances, clothing deformations, and motion dynamics. Even though recent work [7, 50] attempted to achieve photorealistic human textures with manual efforts, it is challenging to precisely mimic the way real-world clothing moves and deforms. Therefore, it is essential to create datasets of real-world human clothing by capturing these intricate details.

The second category (Tab. 1 bottom) involves using multi-view volumetric capture systems [12, 28] to collect datasets of people dressed in real-world clothing [2, 19, 22, 24, 25, 34, 44, 45, 47, 54, 55]. However, the resources required for capturing, storing, and processing this data are substantial, which limits the size of these publicly available datasets [2, 44, 55]. Moreover, these methods do not inherently provide labeled annotations, offering only temporally uncorrelated scans. This makes the raw data on these datasets less suitable for research focusing on human clothing. 4D-DRESS gathers a variety of human subjects and outfits providing accurate semantic labels of human clothing, garment meshes, and SMPL/SMPL-X fits.

Human parsing. Human parsing [53] is a specific task within semantic segmentation aimed at identifying detailed body parts and clothing labels. Conventionally, this challenge is tackled using deep neural networks, trained on images with their corresponding semantic labels [9, 15, 31]. Although these methods have been successful in 2D [16, 20, 21, 30, 32, 49], applying them to annotate 3D and 4D scans is still a challenge. Previous work has explored it using two distinct strategies. One strategy, used by SIZER [47] and MGN [6], involves rendering multi-view images and projecting parsing labels onto 3D meshes through a voting process. While this method considers consistency across multiple views, it overlooks temporal consistency and falls short of accurately labeling 4D scans. Another approach, used by ClothCap [38], registers all scans to a fixed-topology SMPL model [33] with per-vertex displacements. Yet, this method struggles with handling large motions and complex clothing due to limited template resolutions and model-fitting capabilities. This results in noisy labels near boundaries and loose garments. In contrast, our approach combines multi-view voting and optical warping in a template-free pipeline, achieving both multi-view and temporal consistency.

3. Methodology

To accurately label each vertex within our 4D textured scan sequences, we leverage a semi-automatic parsing pipeline

that incorporates but minimizes manual efforts during the labeling process. Fig. 2 depicts the overall workflow of our pipeline. We first render 24 multi-view images of the current frame textured scan. We combine those images with the previous frame’s multi-view images and labels to deploy three state-of-the-art tools to vote candidate labels for each rendered pixel (Sec. 3.1): a) human image parser, b) optical flow transfer, and c) segmentation masks. Next, we re-project and fuse all the 2D label votes via a Graph Cut optimization to obtain vertex-level semantic labels, considering neighboring and temporal consistency (Sec. 3.2). For those challenging frames where further labeling refinement is needed (around 3% in our dataset), we refined their semantic labels with a manual rectification step that we feed back into the optimization (Sec. 3.3). We describe the details of the pipeline within this section.

3.1. Multi-view Parsing

At each frame $k \in \{1, \dots, N_{frame}\}$, we render the 3D-mesh into a set of multi-view images, consisting of twelve horizontal, six upper, and six lower uniformly distributed views. We note this as $I_{img,n,k}$ with $n \in \{1, \dots, N_{view} = 24\}$. Within the multi-view space, we tackle the problem of assigning a label vote l to each pixel p using multi-view image-based models. The label l varies for human skin, hair, shoes, upper clothing (shirts, hoodies), lower clothing (shorts, pants), and outer clothing (jackets, coats). For clarity, we omit the frame index (k) in the following unless they are strictly needed. Please refer to Fig. 2 and the Supp. Mat. for more label definitions and the versatility of our parsing method with new labels like belts and socks.

Human image parser (PAR). Our primary source of labels is a deep-learning image parser, which provides pixel-level votes for body parts and clothes. Specifically, we apply Graphonomy [16] to each view n and store the labels as a new set of images $\{I_{par}\}$ (see Fig. 2). These labels are then accessible by the vote function $f_{par,n}(p, l)$ that checks if the image $I_{par,n}$ matches the value l at the pixel p , in which case returns 1, or 0 otherwise. This vote function and the other two defined below will be crucial later when setting our full-mesh optimization (Sec. 3.2).

Optical flow transfer (OPT). This block leverages the previous frame’s multi-view labels to provide temporal consistency. Specifically, we use the optical flow predictor RAFT [46] to transfer multi-view labels in the $k - 1$ frame to the current k frame using the texture features on the rendered multi-view images. Similarly to the image parser above, the optical flow output goes to a set $\{I_{opt}\}$. These labels are accessible via the vote function $f_{opt,n}(p, l)$, which checks $I_{opt,n}$ and returns 1 if label l is in p and 0 otherwise.

Segmentation masks (SAM). The multi-view votes generated by the Human Image Parser sometimes lack 3D consistency, particularly when dealing with open garments un-

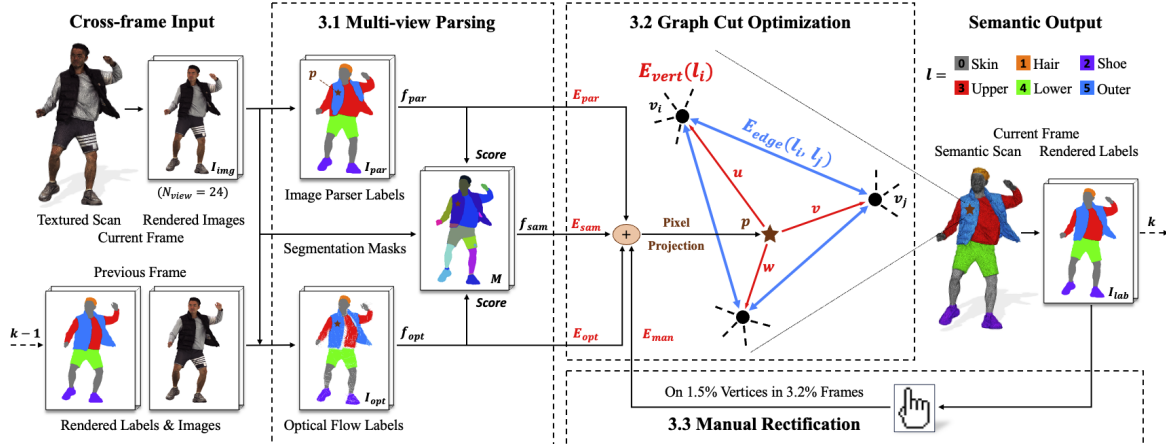


Figure 2. **4D Human parsing method.** We first render current and previous frame scans into multi-view images and labels. Then collect multi-view parsing results from the image parser, optical flows, and segmentation masks (Sec. 3.1). Finally, we project multi-view labels to 3D vertices and optimize vertex labels using the Graph Cut algorithm with vertex-wise unary energy and edge-wise binary energy (Sec. 3.2). The manual rectification labels can be easily introduced by checking multi-view rendered labels. (Sec. 3.3).

der dynamic motions (cf. Fig. 3). While the votes derived from the optical flows provide a cross-frame prior, they may not accurately track every human part and can’t identify newly emerging regions. Therefore, we introduce segmentation masks to regularize the label consistency within each masked region. We apply the *Segment Anything Model* [29] to each rendered image and obtain a self-define group of masks $M_{m,n}$, with the index $m \in \{1, \dots, N_{mask,n}\}$. Within a mask $M_{m,n}$ we compute the score function $\mathcal{S}(l, M_{m,n})$ that fuses the votes of the image parser and the optical flow, normalized by the area of the mask:

$$\mathcal{S}(l, M_{m,n}) = \frac{\sum_{p \in M_{m,n}} [f_{par,n}(p, l) + \lambda_{po} f_{opt,n}(p, l)]}{\sum_{p \in M_{m,n}} (1 + \lambda_{po})}, \quad (1)$$

where the factor λ_{po} weights the contribution of *OPT* over *PAR*. We now define a check function, $\mathcal{C}(p, M_{m,n})$, that returns 1 if the input evaluation pixel p is in the mask $M_{m,n}$ and 0 otherwise. Finally, we obtain the corresponding vote function by summing over all the masks in the image:

$$f_{sam,n}(p, l) = \sum_{m \in 1:N_{mask,n}} \mathcal{C}(p, M_{m,n}) * \mathcal{S}(l, M_{m,n}). \quad (2)$$

3.2. Graph Cut Optimization for Vertex Parsing

The next step in our semi-automatic process is combining all the labels obtained in Sec. 3.1 to assign a unique label to each scan vertex v_i , with $i \in \{1, \dots, N_{vert}\}$. We frame this 3D semantic segmentation problem as a graph cut optimization: each 3D frame is interpreted as a graph G , where vertices are now nodes and mesh edges are connections. Note that in a traditional Graph Cut, the values of the nodes are fixed, and the optimization computes only the

cost of breaking a connection. In our case, we have several *votes* for a vertex label, coming from three different tools and from concurrent multi-view projections. We define our cost function that consists of two terms,

$$E(L) = \sum_{i \in 1:N_{vert}} E_{vert}(l_i) + \sum_{i,j \in 1:N_{vert}} E_{edge}(l_i, l_j), \quad (3)$$

where $L = \{l_i\}$ represents all the vertex labels in current frame. As described below, the term E_{vert} combines the different votes into a single cost function, while E_{edge} evaluates neighboring labels for consistent 3D segmentation. We follow an approach similar to [8].

Vertex-wise unary energy. The cost function per node or *Unary* energy comes from combining the different votes obtained in the multi-view image processing (see Sec. 3.1):

$$E_{vert}(l_i) = \sum_{n \in 1:N_{view}} \frac{\lambda_p E_{par,n} + \lambda_o E_{opt,n} + \lambda_s E_{sam,n}}{N_{view}}, \quad (4)$$

where we combine the human image parser (E_{par}), the cross-frame optical prior (E_{opt}), and the segmentation masks regularization (E_{sam}) contributions. All these energy terms can be written with the same equation by using the notation $\mathcal{X} = \{par, opt, sam\}$:

$$E_{\mathcal{X},n}(l_i) = \sum_{p \in P(v_i,n)} -w_{\mathcal{X}}(p, v_i) f_{\mathcal{X},n}(p, l_i), \quad (5)$$

meaning that energy of the method \mathcal{X} , calculated for a proposed label l_i , is obtained by summing over those pixels $p \in P(v_i, n)$ whose projections are within a triangle of v_i . The weights for the cases of E_{par} and E_{opt} are set to the barycentric distance from the projected pixel p to the vertex v_i , which means $w_{par} = w_{opt} = u$ as in Fig. 2. For E_{sam} instead, we set the weight w_{sam} to the constant value 1 given that we look for an across-vertex regularization.

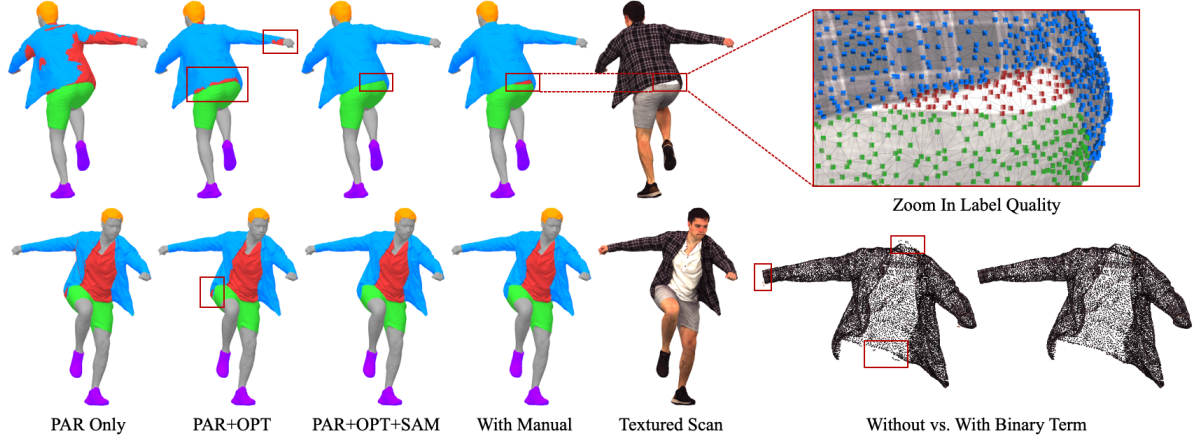


Figure 3. **Qualitative ablation study.** We visualize the effectiveness of our 4D human parsing method on our 4D-DRESS dataset. From left to right, we show the improvements after adding the optical flow labels and mask scores to the multi-view image parser labels. The manual rectification efforts can be easily introduced from multi-view rendered labels, with which we achieve high-quality vertex annotations. The problem of isolated labels can be relieved by introducing the edge-wise binary energy term.

Edge-wise binary energy. The *Binary* energy term penalizes the case of adjacent vertices with different labels, encouraging neighboring vertices to take the same label. Being A the adjacency matrix of the graph G and δ the Dirac delta function, the edge cost can be calculated as follows:

$$E_{edge}(l_i, l_j) = \lambda_b A_{i,j} (1 - \delta(l_i, l_j)), \quad (6)$$

which increases the energy by λ_b in the case that the adjacent vertices v_i, v_j take different labels $l_i \neq l_j$.

3.3. Manual Rectification of 3D Labels

When manual rectification is needed, we introduce it back into the multi-view space as an additional 2D annotation, and we recalculate the steps in Sec. 3.2. Concretely, we ran the graph cut optimization for the first time. Then, we rendered the vertex labels into multi-view labels, from which we let a person introduce corrections by comparing the resulting labels with the textured multi-view images. Similarly to the vote functions of the image parser and optical flow, we create a vote function $f_{man}(p, l)$ that accesses this set of images with rectified annotations and returns 1 if the label l is assigned to the pixel p and 0 otherwise.

Similar to previous cases, we define a per-view manual energy (E_{man}) by using the variable $\mathcal{X} = man$ in Eq. (5), and we added it to the global per-node energy E_{vert} in Eq. (4). We use a constant large weight for w_{man} to favor the manual annotation over other sources of voting where we rectified the labels. The final vertex labels $L^* = \{l^*_i\}$ are obtained after the second round of graph cut optimization. This manual rectification process finally changed 1.5% of vertices within 3.2% of all frames. The rectification process is detailed in Supp. Mat.

4. Experiments

To validate the effectiveness of our method, we conducted controlled experiments on two synthetic datasets,

	CLOTH4D [57]	BEDLAM [7]	
Method	Inner	Inner	Outer
SMPL+D [38]	0.872	0.846	0.765
PAR Only [47]	0.961	0.910	0.714
PAR+OPT	0.969	0.963	0.942
PAR+OPT+SAM	0.995	0.993	0.988

Table 2. **Baseline and ablation study.** Mean accuracy of 4D human parsing methods applied on synthetic datasets. The **Inner** and **Outer** outfits are selected according to our definition in Sec. 5

CLOTH4D [57] and **BEDLAM** [7], where ground-truth semantic labels are available. We first compare our parsing method with a template-based baseline [38], that uses a semantic template (SMPL model with per-vertex displacements) to track and parse the clothed human scans. Due to the limited resolution and the fixed topology nature of the SMPL+D model, its parsing accuracy is lower than 90% on all synthetic outfits (see Tab. 2).

We then compare our 4D parsing pipeline with several ablations and report them in Tab. 2. We use an example scan from 4D-DRESS to support the visualization of the ablation study in Fig. 3. Using PAR only shows reasonable results for upper and lower clothes. Yet, it predicts inconsistent labels at open garments like jackets and coats (Fig. 3 PAR Only), resulting in only 71.4% parsing accuracy on the BEDLAM dataset. The optical flow labels from the previous frame can serve as a cross-frame prior, yet accuracy may vary, particularly in fast-moving arms and cloth boundaries (Fig. 3 PAR+OPT). By fusing both of the previous multi-view labels via the segmentation masks, we achieve better boundary labels (Fig. 3 PAR+OPT+SAM), with 98.8% accuracy on the outer outfits in BEDLAM, with challenging open garments. Finally, we show the effect of

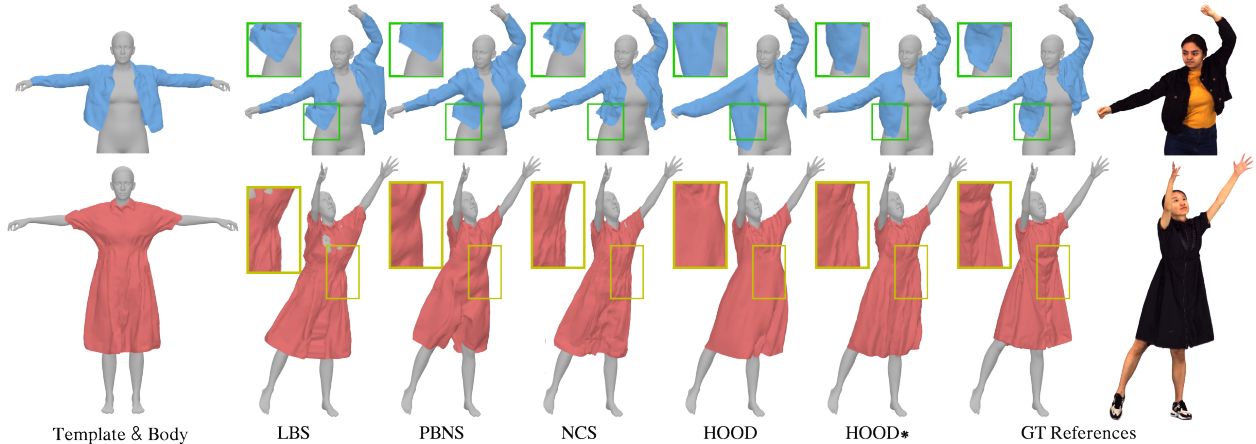


Figure 4. **Qualitative examples for clothing simulation methods.** On the left are templates used for simulations. On the right are ground-truth geometries and original scans, LBS baseline results in body penetrations and overly stretched areas. Compared to other methods, HOOD better models dresses and jackets and, with tuned material parameters, HOOD* achieves simulations closest to the ground truth.

introducing manual efforts to rectify incorrect labels (Fig. 3 With Manual). Our parsing method can also be deployed to annotate other existing 4D human datasets. We present examples of BUFF[55], X-Humans [44], and ActorsHQ[25] and additional qualitative results in Supp. Mat.

5. Dataset Description

4D-DRESS contains 520 motion sequences (150 frames at 30 fps) in 64 real-world human outfits with a total of 78k frames. Each frame consists of multi-view images at 1k resolution, an 80k-face triangle 3D mesh with vertex annotations, and a 1k-resolution texture map. We also provide each garment with its canonical template to benefit the clothing simulation study. Finally, each 3D scan is accurately registered by SMPL/SMPL-X body models.

To record 4D-DRESS we recruited 32 participants (18 female), with an average age of 24. The dataset consists of 4 dresses, 30 upper, 28 lower, and 32 outer garments. Participants were instructed to perform different dynamic motions for each 5-second sequence. For each participant, we capture two types of outfits: **Inner Outfit** comprising the inner layer dress/upper, and lower garments; and **Outer Outfit** with an additional layer of garment, such as open jackets or coats. A unique feature of 4D-DRESS is the challenging clothing deformations we captured. To quantify these deformations, we compute the mean distances from the garments to the registered SMPL body surfaces. The inner and outer outfits exhibit distance ranges up to 7.12 cm and 14.76 cm over all frames. This is twice as much as what we observed in the X-Humans dataset [44], for example. In the 10% most challenging frames, this increases to 20.09 cm for outer outfits, highlighting the prevalence of challenging garments. Please refer to Supp. Mat. for dataset details.

6. Benchmark Evaluation

With high-quality 4D scans and diverse garment meshes in dynamic motions, 4D-DRESS serves as an ideal ground

	Lower		Upper		Dress		Outer	
Method	CD ↓	E_{str} ↓	CD ↓	E_{str} ↓	CD ↓	E_{str} ↓	CD ↓	E_{str} ↓
LBS	1.767	0.333	2.167	0.095	4.461	1.293	4.626	0.811
PBNS [4]	1.885	0.107	2.687	0.040	4.869	0.643	4.859	0.107
NCS [5]	1.716	0.017	2.112	0.016	4.548	0.031	4.738	0.025
HOOD [17]	2.070	0.008	2.668	0.013	4.292	0.010	5.355	0.011
HOOD*	0.924	0.010	1.308	0.015	2.463	0.009	2.833	0.009

Table 3. **Clothing simulation benchmark.** CD is Chamfer Distance between the simulation and ground truth. E_{str} denotes stretching energy with respect to the template.

truth for a variety of computer vision and graphics benchmarks. In our work, we outline several standard benchmarks conducted in these fields using our dataset. Our primary focus is on tasks related to clothing simulation (Sec. 6.1) and clothed human reconstruction (Sec. 6.2). Additionally, benchmarks on human parsing and human representation learning are included in our Supp. Mat.

6.1. Clothing Simulation

Experimental setup. We introduce a new benchmark for clothing simulation, leveraging the garment meshes from 4D-DRESS, which capture dynamical real-world clothing deformations. This benchmark evaluates three methods for modeling garment dynamics: **PBNS** [4], Neural Cloth Simulator (**NCS** [5]), and **HOOD** [17], as well as a baseline method that applies SMPL-based linear blend-skinning (**LBS**) to the template. We ran the simulations using T-posed templates extracted from static scans and compared the results to the ground-truth garment meshes across various pose sequences. Our evaluation metrics include the Chamfer Distance (**CD**), which compares the resulting mesh sequences with ground-truth point clouds, and the average stretching energy (E_{str}) calculated by measuring the difference in edge lengths between the simulated and template meshes. The experiments were conducted across four categories of garments (Lower, Upper, Dress, and Outer),

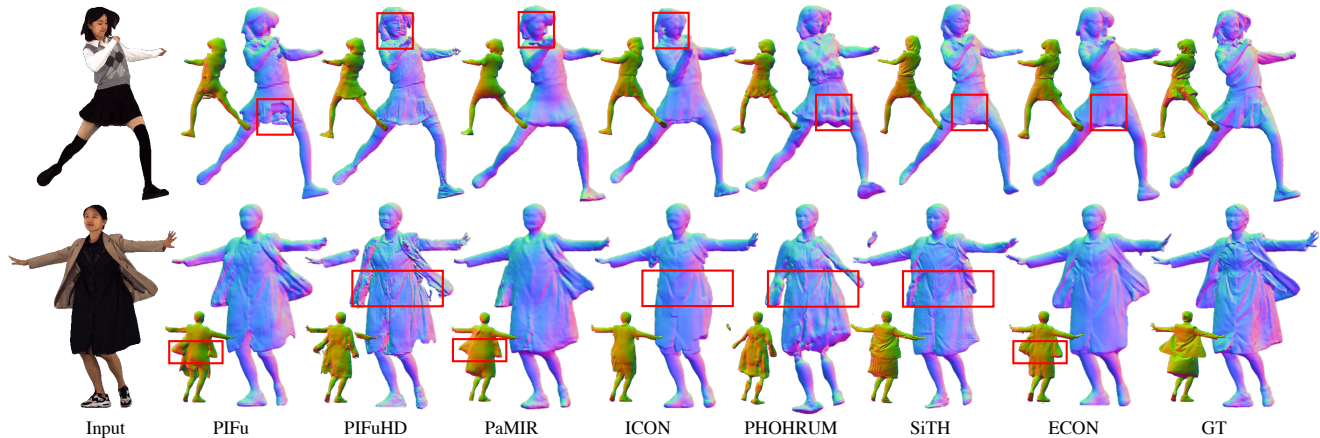


Figure 5. **Examples of clothed human reconstruction on 4D-DRESS.** We evaluate state-of-the-art methods using both inner (*Top*) and outer (*Bottom*) outfits. We show that existing methods generally struggle with the challenging loose garments. Moreover, these approaches cannot faithfully recover realistic details such as clothing wrinkles.

with four garment templates in each category. We simulated clothing deformation for each garment in six different pose sequences, providing a comprehensive comparison of their ability to generate realistic motions.

Fine-tuning material parameters. To demonstrate the advantages of real-world garment meshes in 4D-DRESS, we also introduce a simple optimization-based strategy for inverse simulation using HOOD. Specifically, we optimize the material parameters fed into the HOOD model to minimize the simulations’ Chamfer Distance to the ground-truth sequences and their stretching energy. This optimized version is denoted as **HOOD***. For more details on the material optimization experiments, please refer to Supp. Mat.

Evaluation results. The quantitative and qualitative comparisons of the clothing simulation methods are presented in Tab. 3 and Fig. 4 respectively. The LBS baseline and LBS-based approaches (PBNS and NCS) perform better with upper and lower garments, which exhibit limited free-flowing motions compared with the dress and outer garments. Conversely, HOOD excels with dresses, generating more natural, free-flowing motions and achieving lower stretching energy. However, if HOOD fails to generate realistic motions for a single frame, this error propagates to all subsequent frames. This issue does not occur in the LBS-based methods, which generate geometries independently for each frame. With finely-tuned material parameters, HOOD* produces garment sequences that more faithfully replicate real-world behavior. We anticipate that future research in learned garment simulation will increasingly focus on modeling real-world garments made from complex heterogeneous materials. This will be a major step in creating realistically animated digital avatars, and we believe 4D-DRESS will be highly instrumental in this task.

Method	Inner			Outer		
	CD↓	NC↑	IoU↑	CD↓	NC↑	IoU↑
PIFu [40]	2.696	0.792	0.690	2.783	0.759	0.697
PIFuHD [41]	<u>2.426</u>	0.793	0.739	<u>2.393</u>	0.763	0.743
PaMIR [56]	2.520	<u>0.805</u>	0.706	2.608	<u>0.777</u>	0.715
ICON [51]	2.473	0.798	<u>0.752</u>	2.832	0.762	0.756
PHORHUM [1]	3.944	0.725	0.580	3.762	0.705	0.603
ECON [52]	2.543	0.796	0.736	2.852	0.760	0.728
SiTH [23]	2.110	0.824	0.755	2.322	0.794	<u>0.749</u>

Table 4. **Clothed human reconstruction benchmark.** We computed Chamfer distance (CD), normal consistency (NC), and Intersection over Union (IoU) between ground truth and reconstructed meshes obtained from different baselines.

6.2. Clothed Human Reconstruction

Experimental setup. We create a new benchmark for evaluating state-of-the-art clothed human reconstruction methods on the 4D-DRESS dataset. This benchmark is divided into three subtasks. First, we evaluate **single-view human reconstruction** utilizing images and high-quality 3D scans from our dataset. In addition, benefiting from the garment meshes in our dataset, we establish the first real-world benchmark for evaluating **single-view clothing reconstruction**. Finally, we assess **video-based human reconstruction** approaches leveraging the sequences in 4D-DRESS that capture rich motion dynamics of both human bodies and garments. In all the experiments, we report 3D metrics including Chamfer Distance (CD), Normal Consistency (NC), and Intersection over Union (IoU) to compare the predictions with ground-truth meshes.

Single-view human reconstruction. We use the two test sets defined in Sec. 5 (denote as **Outer** and **Inner**) to evaluate the following single-view reconstruction methods: PIFu [40], PIFuHD [41], PaMIR [56], ICON [51], PHORHUM [1], ECON [52], and SiTH [23]. The evaluation results are summarized in Fig. 5 and Tab. 4. We ob-

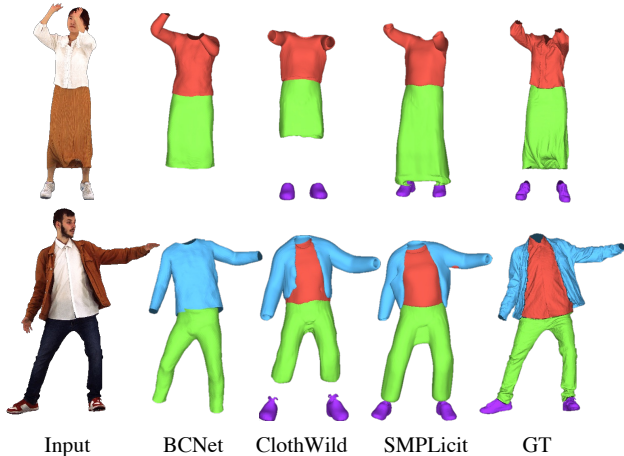


Figure 6. **Examples of clothing reconstruction on 4D-DRESS.** We visualize the reconstructed garment meshes from different approaches. These methods trained on synthetic datasets failed to predict accurate clothing sizes and detailed wrinkles.

Method	Shoes		Lower		Upper		Outer	
	CD ↓	IoU ↑	CD ↓	IoU ↑	CD ↓	IoU ↑	CD ↓	IoU ↑
BCNet [26]	-	-	2.533	0.675	2.079	0.700	3.600	0.639
SMPLicit [14]	2.619	0.621	2.101	0.698	2.452	0.617	3.359	0.618
ClothWild [36]	3.657	0.548	2.690	0.582	3.279	0.533	4.163	0.588

Table 5. **Clothing reconstruction benchmark.** We report Chamfer Distance (CD), and Intersection over Union (IoU) between the ground-truth garment meshes and the reconstructed clothing.

served that methods leveraging SMPL body models as guidance (i.e., ICON, ECON, SiTH) performed better in reconstructing inner clothing. However, their performance significantly declined when dealing with outer garments. On the other hand, end-to-end models like PIFu and PIFuHD demonstrated more stability with both clothing types. This leads to an intriguing research question: whether the human body prior is necessary for reconstructing clothing. Qualitatively, we see that even the best-performing methods cannot perfectly reconstruct realistic free-flowing jackets as shown in Tab. 4. We believe 4D-DRESS will offer more valuable insights for research in clothed human reconstruction.

Single-view clothes reconstruction. Clothes reconstruction has received relatively little attention compared to full-body human reconstruction. Leveraging the garment meshes in 4D-DRESS, we introduce the first real-world benchmark to assess prior art, including **BCNet** [26], **SMPLicit** [14], and **ClothWild** [36]. The results of different clothing types, as shown in Fig. 6, indicate a significant gap between the reconstructed and real clothing. Firstly, the clothing sizes produced by these methods are often inaccurate, suggesting a lack of effective use of image information for guidance. Moreover, the results typically lack geometric details like clothing wrinkles compared to full-body reconstruction. We report quantitative results in Tab. 5. We ob-



Figure 7. **Video-based human reconstruction.** Qualitative results of video-based human reconstruction methods on 4D-DRESS. Prior works struggle to reconstruct 3D human with challenging outfits and cannot recover the fine-grained surface details.

Method	Inner			Outer		
	CD ↓	NC ↑	IoU ↑	CD ↓	NC ↑	IoU ↑
SelfRecon [27]	3.180	0.729	0.754	4.027	0.683	0.745
Vid2Avatar [18]	2.870	0.750	0.772	3.014	0.725	0.787

Table 6. **Video-based human reconstruction.** Results of video-based human reconstruction methods on 4D-DRESS.

served that the data-driven method (BCNet) performs better with inner clothing, while the generative fitting method (SMPLicit) shows more robustness to outer clothing, such as coats. However, none of these methods is designed for or trained on real-world data. The domain gap between synthetic and real data still limits their capability to produce accurate shapes and fine-grained details. We expect our benchmark and dataset will draw more research attention to the topic of real-world clothing reconstruction.

Video-based human reconstruction Leveraging the sequential 4D data in our dataset, we create a new benchmark for evaluating video-based human reconstruction methods. We applied **Vid2Avatar** [18] and **SelfRecon** [27] to obtain 4D reconstructions and compared them with the provided ground-truth 4D scans. As observed in Fig. 7, both methods struggle with diverse clothing styles and face challenges in reconstructing surface parts that greatly differ in topology from the human body, such as the open jacket. Moreover, there remains a noticeable discrepancy between the real geometry and the recovered surface details. Quantitatively, the existing methods cannot achieve satisfactory reconstruction results with outer garments, as demonstrated by a large performance degradation in Tab. 6. We believe 4D-DRESS provides essential data for advancing video-based human reconstruction methods, particularly in achieving detailed geometry recovery for challenging clothing.

6.3. Clothed Human Parsing

We design a benchmark for the image-based human parser. Concretely, we project each scan frame’s vertex labels to the multi-view captured images using corresponding camera parameters and rasterizer, which provide the ground-truth pixel labels for evaluating the image-based human pars-



Figure 8. **Human representation learning.** Qualitative results of the novel pose synthesis of state-of-the-art human representation learning approaches together with the GT of 4D-DRESS. All Baseline methods fail to learn the large non-rigid surface deformations and are bounded by the skeletal deformations.

Method	Inner		Outer	
	mAcc.↑	mIoU↑	mAcc.↑	mIoU↑
SCHP [30]	0.908	0.832	0.863	0.768
CDGNet [32]	0.922	0.853	0.887	0.790
Graphonomy [16]	0.968	0.859	0.915	0.810

Table 7. **Image-based human parsing.** Results of image-based human parsers on 4D-DRESS.

Method	Inner			Outer		
	CD↓	NC↑	IoU↑	CD↓	NC↑	IoU↑
SCANimate [42]	0.965	0.854	0.918	1.237	0.828	0.912
SNARF [10]	1.158	0.843	0.907	1.248	0.827	0.930
X-Avatar [44]	1.008	0.861	0.954	1.177	0.841	0.946

Table 8. **Human representation learning.** Results of human representation learning approaches on 4D-DRESS.

ing methods: **SCHP** [30], **CDGNet** [32], and **Graphonomy** [16]. In Tab. 7, we report the mean Pixel Accuracy (mAcc.) and mean Intersection over Union (mIoU) between the prediction and the ground-truth labels. We conducted our human image parsing experiments on one subset of our 4D-DRESS dataset, which contains 128 sequences of 64 outfits (2 sequences for each of the inner and outer outfits). The qualitative parsing results are shown in Fig. 9.

6.4. Human Representation Learning

We design a new benchmark for evaluating the human representation learning task. Unlike physics-based methods, this line of work directly takes 3D human scans as training input and obtains an animation-ready human avatar. We follow the split strategy mentioned before and evaluate prior works, **SCANimate** [42], **SNARF** [10], **X-Avatar** [44] on the novel-pose synthesis. Fig. 8 shows that state-of-the-art human representation learning approaches cannot correctly learn the large non-rigid surface deformations (e.g., folded skirt) due to the strong skeletal dependency and the lack of modeling for temporal dynamics. This effect can also be

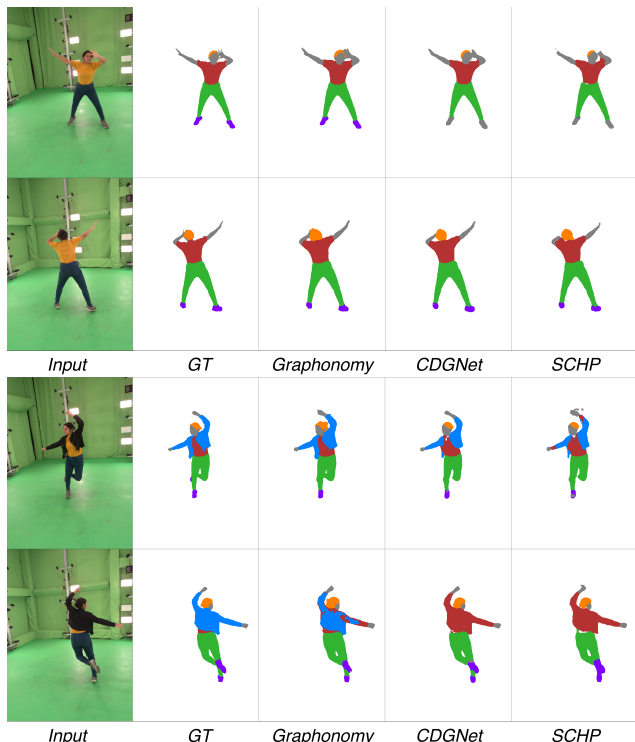


Figure 9. **Human parsing comparison.** We use the ground-truth semantic labels to evaluate state-of-the-art human parsing models. These methods generally failed to predict correct clothing labels from different view angles.

reflected in Tab. 8 quantitatively where all baseline methods produce higher errors on the split of more challenging garments (outer outfits).

7. Discussion

Limitations. Our current pipeline requires substantial computational time. The offline manual rectification process and garment mesh extraction also demand expertise in 3D editing and additional human efforts. These factors constrain the scalability of our dataset. With a goal of expanding more diverse subjects and clothing, real-time 4D anno-

tation and rectification/editing will be exciting future work. **Conclusion.** 4D-DRESS is the first real-world 4D clothed human dataset with semantic annotations, aiming to bridge the gap between existing clothing algorithms and real-world human clothing. We demonstrate that 4D-DRESS is not only a novel data source but also a challenging benchmark for clothing simulation, reconstruction, and other related tasks. We believe that 4D-DRESS can support a wide range of endeavors and foster research progress by providing high-quality 4D data in life like human clothing.

Acknowledgements. This work was partially supported by the Swiss SERI Consolidation Grant "AI-PERCEIVE". AG was supported in part by the Max Planck ETH CLS.

References

- [1] Thiemo Alldieck, Mihai Zanfir, and Cristian Sminchisescu. Photorealistic monocular 3d reconstruction of humans wearing clothing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 7
- [2] Matthieu Armando, Laurence Boissieux, Edmond Boyer, Jean-Sebastien Franco, Martin Humenberger, Christophe Legras, Vincent Leroy, Mathieu Marsot, Julien Pansiot, Sergi Pujades, Rim Rekić, Gregory Rogez, Anilkumar Swamy, and Stefanie Wuhrer. 4dhumanoutfit: a multi-subject 4d dataset of human motion sequences in varying outfits exhibiting large displacements. *Computer Vision and Image Understanding*, 2023. 2, 3
- [3] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 344–359. Springer, 2020. 1, 2, 3
- [4] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Pbn: Physically based neural simulation for unsupervised garment pose space deformation. *ACM Transactions on Graphics (TOG)*, 40(6), 2021. 1, 6, 17
- [5] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Neural cloth simulation. *ACM Transactions on Graphics (TOG)*, 41(6):1–14, 2022. 1, 6, 17
- [6] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019. 2, 3
- [7] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8737, 2023. 1, 2, 3, 5, 15
- [8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11): 1222–1239, 2001. 4
- [9] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3
- [10] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 9
- [11] CLO. <https://www.clo3d.com>, 2022. 3
- [12] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015. 2, 3, 16
- [13] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 17
- [14] Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 8
- [15] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 770–785, 2018. 3
- [16] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 9, 13
- [17] Artur Grigorev, Bernhard Thomaszewski, Michael J. Black, and Otmar Hilliges. Hood: Hierarchical graphs for generalized modelling of clothing dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16965–16974, 2023. 1, 6, 17
- [18] Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8
- [19] Sang-Hun Han, Min-Gyu Park, Ju Hong Yoon, Ju-Mi Kang, Young-Jae Park, and Hae-Gon Jeon. High-fidelity 3d human digitization from single 2k resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [20] Haoyu He, Jing Zhang, Qiming Zhang, and Dacheng Tao. Grapy-ml: Graph pyramid mutual learning for cross-dataset human parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 3
- [21] Haoyu He, Jing Zhang, Bhavani Thuraisingham, and Dacheng Tao. Progressive one-shot human parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 3
- [22] Zhu Heming, Cao Yu, Jin Hang, Chen Weikai, Du Dong, Wang Zhangye, Cui Shuguang, and Han Xiaoguang. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In *Proceedings of the Euro-*

- pean Conference on Computer Vision (ECCV), pages 512–530. Springer International Publishing, 2020. 3
- [23] Hsuan-I Ho, Jie Song, and Otmar Hilliges. Sith: Single-view textured human reconstruction with image-conditioned diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 7
- [24] Jie Song Hsuan-I Ho, Lixin Xue and Otmar Hilliges. Learning locally editable virtual humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [25] Mustafa İşık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 2, 3, 6
- [26] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020. 1, 8
- [27] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8
- [28] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 3
- [29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2, 4, 13
- [30] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 3, 9
- [31] X. Liang, K. Gong, X. Shen, and L. Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(04):871–885, 2019. 3
- [32] Kunliang Liu, Ouk Choi, Jianming Wang, and Wonjun Hwang. Cdnet: Class distribution guided network for human parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4473–4482, 2022. 3, 9
- [33] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248:1–248:16, 2015. 3
- [34] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to Dress 3D People in Generative Clothing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3
- [35] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J. Black. The power of points for modeling humans in clothing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 3
- [36] Gyeongsik Moon, Hyeongjin Nam, Takaaki Shiratori, and Kyoung Mu Lee. 3d clothed human reconstruction in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 8
- [37] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020. 2, 3
- [38] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4), 2017. 2, 3, 5, 15
- [39] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 13
- [40] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 7
- [41] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7
- [42] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 9
- [43] Yidi Shao, Chen Change Loy, and Bo Dai. Towards multi-layered 3d garments animation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 2, 3
- [44] Kaiyue Shen, Chen Guo, Manuel Kaufmann, Juan Zarate, Julien Valentin, Jie Song, and Otmar Hilliges. X-avatar: Expressive human avatars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 6, 9, 17
- [45] Zhaoqi Su, Tao Yu, Yangang Wang, and Yebin Liu. Deepcloth: Neural garment representation for shape and style editing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(2):1581–1593, 2023. 3
- [46] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020. 3, 13
- [47] Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020. 3, 5
- [48] Unreal Engine 5. <https://www.unrealengine.com>, 2022. 3

- [49] Wenguan Wang, Hailong Zhu, Jifeng Dai, Yanwei Pang, Jianbing Shen, and Ling Shao. Hierarchical human parsing with typed part-relation reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [3](#)
- [50] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3681–3691, 2021. [3](#)
- [51] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. ICON: Implicit Clothed humans Obtained from Normals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [7](#)
- [52] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. ECON: Explicit Clothed humans Optimized via Normal integration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [7](#)
- [53] Lu Yang, Wenhe Jia, Shan Li, and Qing Song. Deep learning technique for human parsing: A survey and outlook. *arXiv preprint arXiv:2301.00394*, 2023. [3](#)
- [54] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgb-d sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [3](#)
- [55] Chao Zhang, Sergi Pujades, Michael J. Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#), [3](#), [6](#)
- [56] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. [7](#)
- [57] Xingxing Zou, Xintong Han, and Waikeng Wong. Cloth4d: A dataset for clothed human reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12847–12857, 2023. [1](#), [2](#), [3](#), [5](#), [15](#)

4D-DRESS: A 4D Dataset of Real-World Human Clothing With Semantic Annotations

Supplementary Material



Figure 10. **Example of 24 rendered views.** We render 24 views to ensure the visibility of each scan vertex and consider the computational cost of human parsing.

8. Implementation Details

8.1. Multi-view Parsing

Multi-view rendering. For each frame $k \in \{1, \dots, N_{frame}\}$, we render twelve horizontal, six upper, and six lower images $I_{img,n,k}$ that are uniformly distributed on a sphere by rasterizing the textured scan with Pytorch3D [39], where $n \in \{1, \dots, N_{view} = 24\}$. Each scan is centralized according to its bounding box center and then placed at the camera sphere center. The rendered images have a resolution of 512×512 . Examples of 24-view rendered images are shown in Fig. 10.

4D-DRESS	Graphonomy (LIP)
(-1) other	background
(0) skin	torso-skin, face, glove left-arm, right-arm, left-leg, right-leg
(1) hair	hat, hair, sunglasses
(2) shoe	socks, left-shoe, right-shoe
(3) upper	upper-clothing, dress, scarf
(4) lower	pant, skirt
(5) outer	coat

Table 9. **Label mapping between 4D-DRESS and LIP dataset.** We define 6 label categories based on LIP dataset.

Human image parser (PAR). We apply the pre-trained Graphonomy [16] to each rendered image $I_{img,n,k}$ and save the label results as a new image $I_{par,n,k}$. Concretely, we manually classify the 20 classes of Graphonomy labels into 6 classes that are used in our dataset: skin (0), hair(1), shoes(2), upper(3), lower(4), and outer(5) clothes. The corresponding labels between Graphonomy (LIP) and ours are shown in Tab. 9. Specifically, we map the background label from Graphonomy to our setting with a label value -1, and the color code of white. These background labels will return 0 in the vote function $f_{par,n}(p, l)$.

Optical flow transfer (OPT). To establish connections with previous frames, we project previous frame vertex labels to multi-view labels $I_{lab,n,k-1}$ using the same rendering cameras and rasterizer from Pytorch3D. Then, we warp these previous multi-view labels to the current frame $I_{opt,n,k}$ using the optical flow vectors predicted by the RAFT [46] model. The vertex labels at the first frame do not involve this process thanks to our first-frame initialization (see Sec. 8.3). Concretely, each pixel label with location p within $I_{lab,n,k-1}$ will be warped to a new pixel location $p + v$ at the current frame, through the optical flow vector $v = RAFT(I_{img,n,k-1}, I_{img,n,k}, p)$. The new labels at the current frame are determined by voting. If there is no corresponding label found in the previous frame, the new label will be set to -1.

Segmentation masks and scores (SAM). We use Segment Anything Model [29] to segment each rendered image $I_{img,n}$ into a group of masks $M_{m,n}$ without any ex-

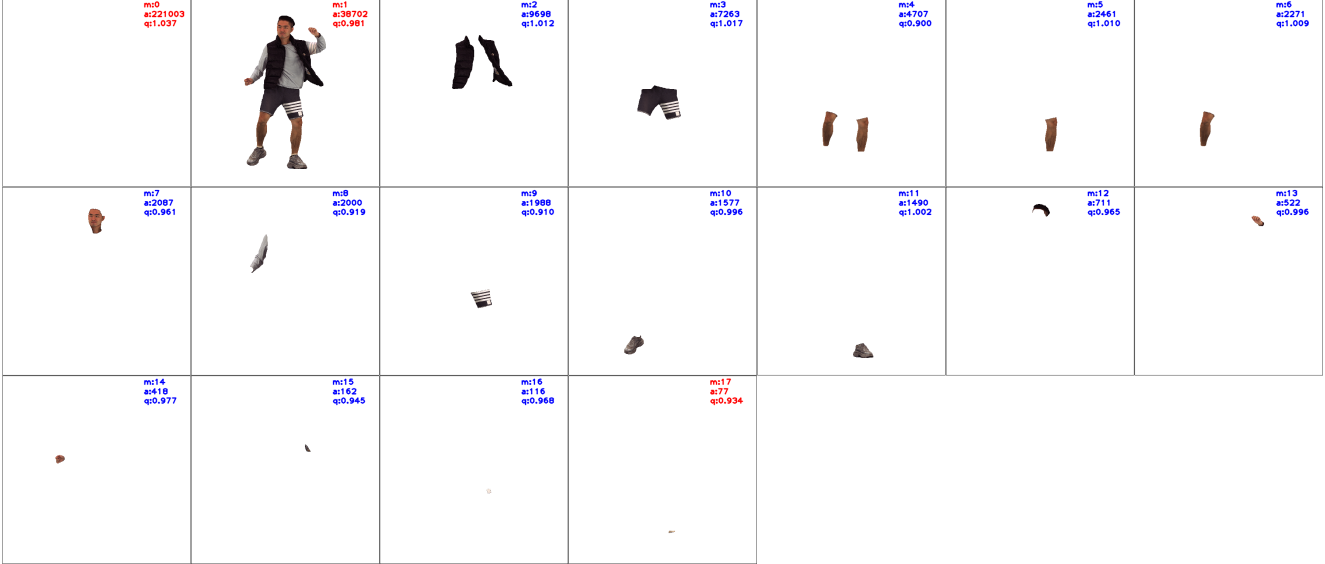


Figure 11. **Example of SAM predictions.** The input image is the first view (upper-left) of Fig. 10. We filter out the segmentation masks that contain background, full body, and only small regions (marked as red).

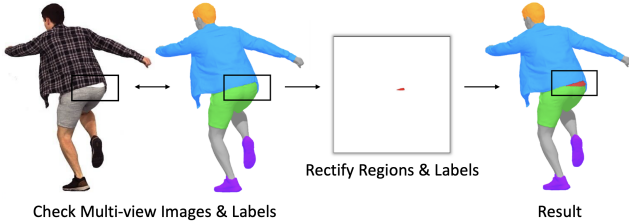


Figure 12. **Example of manual rectification.** An annotator selects a region in the rendered images and gives a correct label. The label is projected to 3D and used for correcting the 3D vertices through a second round of graph cut optimization.

tra prompts, where $m \in \{1, \dots, M_{mask,n}\}$. Then we compute the score function $S(l, M_{m,n})$ within each mask for each label by fusing the votes from the image parser and optical flow, normalized by the area of the mask. Fig. 11 depicts the predicted segmentation masks from a rendered image. A pixel p within the rendered image $I_{img,n}$ may belong to multiple segmentation masks. In this case, the SAM vote function $f_{sam,n}(p, l)$ is calculated by summing all the scores of masks that contain this pixel.

8.2. Graph Cut Optimization

The energy Eq. (5) in the main paper is optimized through the graph cut algorithm (alpha-expansion). The vertex-wise unary energy is normalized among all labels and then added to the edge-wise binary energy. The weights are empirically set as $\lambda_p = 0.5$, $\lambda_o = 0.5$, $\lambda_{po} = 1.5$, $\lambda_s = 1$, and $\lambda_b = 1$.

8.3. Manual Rectification Process

Manual rectification on segmentation masks. In our dataset, each scan mesh has around 80k vertices. Manually annotating their vertex labels on the 3D scans is very expensive and time-consuming. Thus, we introduce a manual rectification process within the 2D image space. After the first graph cut optimization, we render vertex labels to multi-view images, from which we let an annotator correct labels with the segmentation masks and a painting tool. More specifically, the annotator is asked to identify an incorrectly labeled region by checking the multi-view images and labels. Once an incorrect labeling is found, the annotator will look for its corresponding segmentation masks for label correction. If such a mask does not exist, the annotator will manually paint the region using a painting tool. Finally, the images with rectified labels are projected to 3D vertices and are formulated as the manual vote function $f_{man,n}(p, l)$. The energy $E_{man,n}$ term will be added to the second round of graph cut optimization, with a large weight $w_{man} = 10$. We note that for each 150-frame 4D sequence, the rectification process takes about 30 minutes on a desktop with an RTX 2080Ti GPU whereas the human parsing and the graph cut optimization take two and one hour, respectively. An example of our rectification process is shown in Fig. 12.

First-frame initialization of vertex labels. To ensure a good label initialization, the motion sequences always start from the A pose, which is easier for human parsing and pose registration. We obtain the first-frame vertex labels

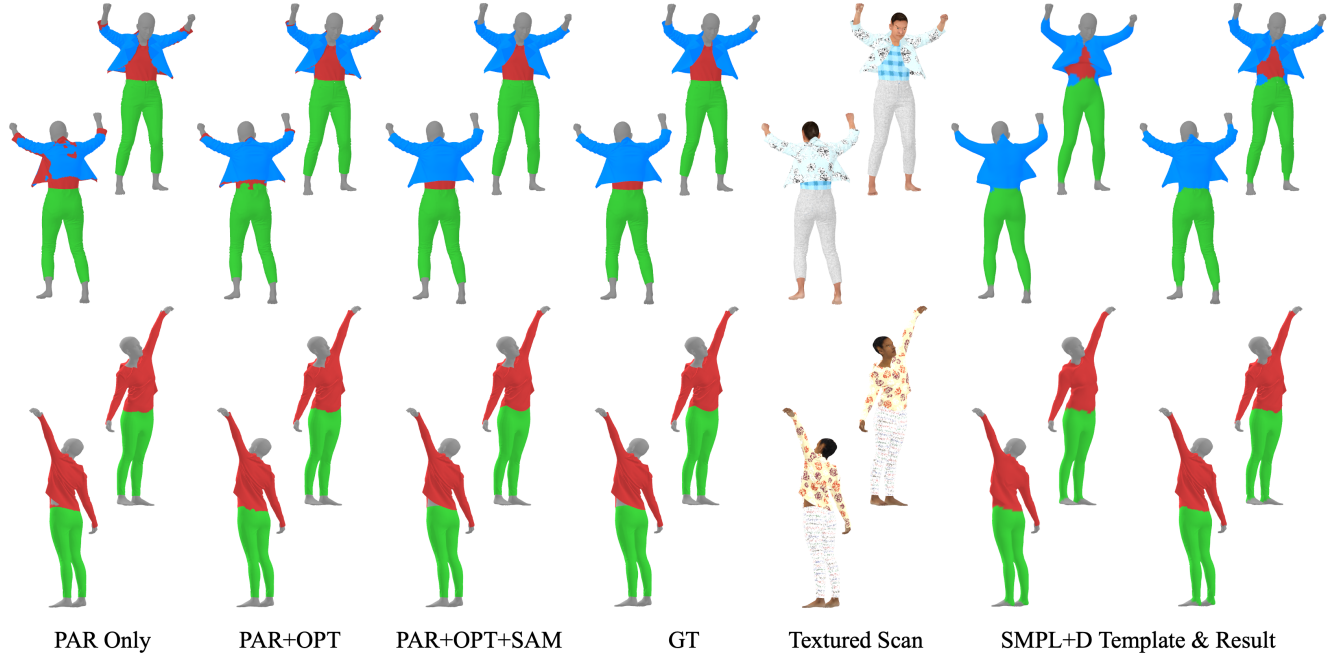


Figure 13. **Ablation study and baseline comparison on the BEDLAM dataset.** We conducted ablative experiments on the synthetic BEDLAM dataset where ground-truth semantic labels are available.

using the edge-wise binary energy and the multi-view unary energy calculated only from the image parser (E_{par}) and manual rectifications (E_{man}).

9. Additional Parsing Experiments

9.1. 4D Parsing on Synthetic Datasets

We conducted controlled 4D parsing experiments on two synthetic datasets, CLOTH4D [57] and BEDLAM [7], where the cloth meshes are simulated from cloth templates on top of the parameterized body models. Since within these synthetic datasets, some inner body and cloth vertices are always invisible from the outside, we report our labeling accuracy only on the vertices that are visible from our 24 views of rendered images.

Baseline comparison. We first compare our 4D human parsing method with a template-based baseline method [38] that utilizes a semantic SMPL+D template to first track the clothed human shape, and then project the template labels to neighboring scan vertices. Since ClothCap [38] didn't release their 4D parsing code, we implemented their parsing method following their descriptions. We first register the SMPL+D model to all frames. Then we initialize the first frame template label using the nearby scan vertex labels obtained through our first-frame initialization process. At each frame, we update the template labels using the body prior, previous frame prior, and the Gaussian Mixture

Model trained from the vertex colors of each labeled category. Finally, the scan vertex labels are assigned from the nearest template label. The quantitative parsing results from this baseline method are shown in the main paper. Here, we show more qualitative results in Fig. 13.

The main issue of this template-based baseline method is fitting the SMPL+D template to loose human outfits. The spatial mismatch between template and loose garments generates incorrect labels, especially in the open area of the jackets. Besides this, precisely updating the template labels using the Gaussian mixture model of labeled vertex colors is also difficult, especially in front of garments that have similar colors. The limited template resolution also results in noisy boundary labels at the higher-resolution clothed human meshes. The parsing accuracy from this baseline method is below 90% for all synthetic outfits.

Ablation studies. We then compare our 4D human parsing method (without manual rectifications) with several ablations of the multi-view parsing inputs (PAR Only, PAR+OPT, PAR+OPT+SAM), as shown in Fig. 13. Similar to Fig. 3 in the main paper, we observed similar qualitative results on the synthetic datasets.

9.2. 4D Parsing on Other Datasets

Our 4D human parsing method takes the input as scan mesh sequences and multi-view videos and thus can be applied to the existing real-world 4D human datasets, such as BUFF,

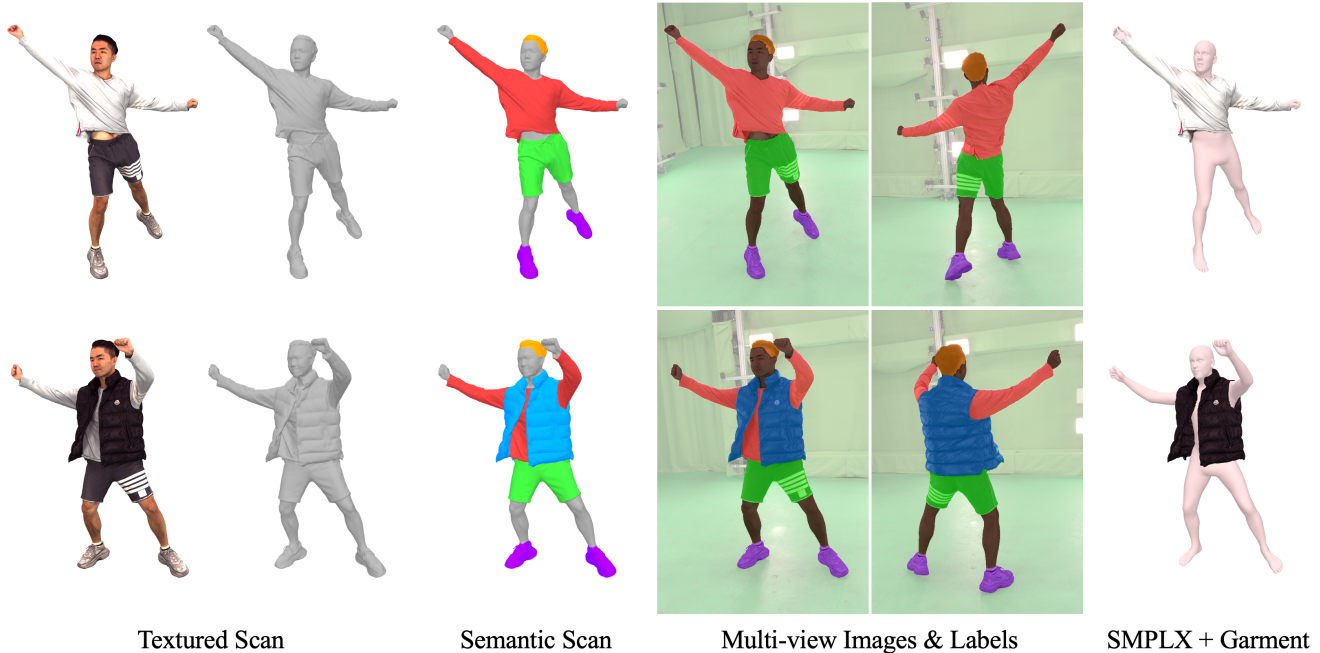


Figure 14. **Data provided in the 4D-DRESS dataset.** We provide high-quality 4D textured scans. For each scan, we annotate vertex-level semantic labels, thereby obtaining the corresponding garment meshes and fitted SMPL(-X) body meshes.

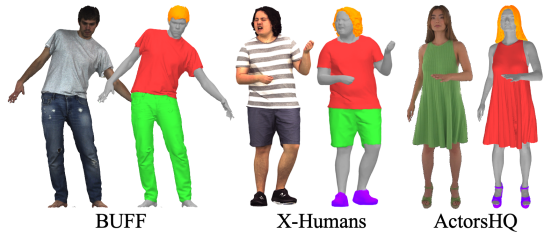


Figure 15. **4D human parsing on other real-world datasets.**

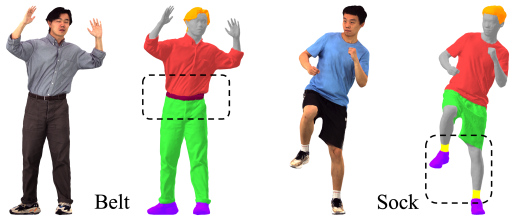


Figure 16. **4D human parsing with new labels.**

X-Humans, and ActorsHQ, as shown in Fig. 15.

9.3. 4D Parsing with New Labels

The six classes in our 4D-DRESS are strategically defined to ensure a consistent benchmark evaluation for clothing simulation and reconstruction. We showcase the generalization ability of our parsing method with new labels in Fig. 16, by effectively distinguishing a belt from pants and socks from shoes. Initiated during the first-frame initialization, these new labels can integrate into the 4D parsing pipeline. However, refining labels for these smaller clothes and objects may entail additional manual efforts for rectifi-

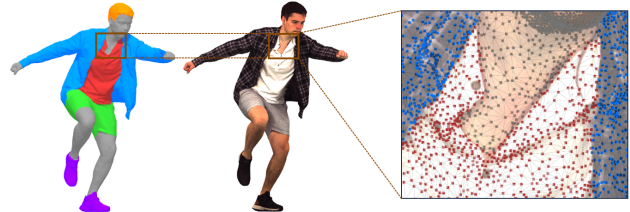


Figure 17. **Vertex-level semantic annotations.** Our dataset contained precise vertex-level semantic labels of clothing categories.

cation.

10. Additional Dataset Description

10.1. Data Capturing Steup

We captured our dataset with a volumetric capture system [12] equipped with 106 synchronized cameras (53 RGB and 53 IR cameras). The sequences are filmed at 12 MP, 30 FPS, and within an effective capture volume of 2.8 m in diameter and 3 m in height. Each frame consists of a mesh with 80k faces and a texture map.

10.2. Dataset Contents

Our 4D-DRESS dataset provides the following data, examples are shown in Fig. 14:

- **4D textures scans.** High-quality 4D textured scans of 32 subjects, 64 human outfits (32 Inner and 32 Outer), with 520 motion sequences and 78k frames in total.

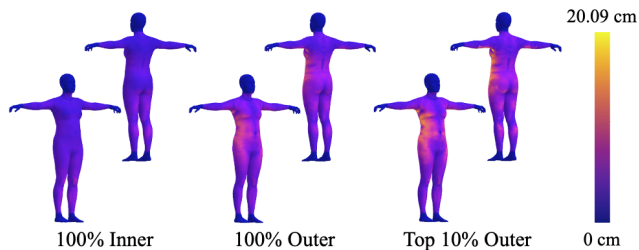


Figure 18. **Visualization of 4D-DRESS outfits distance.** The mean distance distribution from garment outfits to SMPL bodies.

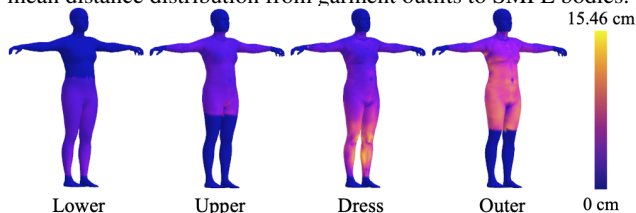


Figure 19. **Visualization of 4D-DRESS outfits distance.** The mean distance distribution from garment meshes to SMPL bodies.

- **Vertex-level annotations.** We offer accurate vertex-level annotations through our 4D human parsing pipeline. An example of our label quality is shown in Fig. 17. Using these labels, we also provide multi-view images with semantic labels in 2D.
- **Parametric body models.** We register precise SMPL and SMPL-X body models for each frame.
- **Garment meshes.** We extract 3D garment meshes based on the vertex labels.

10.3. Clothing Distribution

We compute the mean distances from the outfits to the registered SMPL body surfaces. The inner and outer outfits exhibit distance ranges of up to 7.12 cm and 14.76 cm, respectively, over all frames. The distribution of the distance on the SMPL body is shown in Fig. 18. In the 10% most challenging frames that have a larger Chamfer distance between scan mesh and SMPL mesh, the distance range increases to 20.09 cm for outer outfits. We further visualize the mean distances of each garment category, as shown in Fig. 19. The average Chamfer distance between the clothed human scans and SMPL body meshes are 3.30 cm and 5.28 cm for the inner and outer outfits in our 4D-DRESS dataset, and 2.21 cm in the X-Humans dataset [44].

11. Experimental Details

11.1. Clothing Simulation

4D-Dress provides diverse garments and challenging human pose sequences, which serves as a great asset for future research in clothing simulation. Unlike the synthesized garment templates with smooth surfaces and simple topologies, we provide templates extracted from scans, with realistic

wrinkles and complex structures. Using these templates, we evaluated the performance of recent unsupervised cloth simulators, including PBNS [4], Neural Cloth Simulator (NCS) [5] and HOOD [17], and a baseline method, linear blend-skinning. We quantitatively and qualitatively compared the generated garments with our scanned garments. We also demonstrated the potential of HOOD by simply optimizing the material parameters, which again confirmed the value of our dataset. In the following sections, we elaborate on each step of our experiments.

11.2. Template Extraction

Current clothing simulation algorithms rely on a predefined garment template, deforming it to generate realistic simulations under various poses. They typically utilized synthesized garment templates, with unnaturally smooth surfaces and basic topologies. In our work, we provided templates directly extracted from real-world scans, offering a more realistic foundation for deformation.

Firstly, we select from pose sequences the frames closest to the canonical pose, in other words, “T-pose”. We also make sure that the body in this frame is static and garments are in rest status. Then we apply inverse LBS to convert the scans into exact canonical pose. After extracting garment meshes from the unposed scans, we made some manual efforts to recover the garment shape in Blender [13]. Specifically, we erased unwanted faces, solved penetrations between clothing and body, and smoothed rigid wrinkles and coarse boundaries. Synthesized templates used by current simulators usually have 4-5k vertices. We observed in experiments that too many vertices in the template are computationally expensive for simulation and may erode performance. Therefore, we downsampled each template to 30-50%, which now has 3-8k vertices in total depending on each garment’s surface area, while keeping them in their original shapes. To use lower garments in simulators, like pants and lower skirts, pinned vertices are compulsory for them to stay on the body. We extract the loop around the waist as pinned vertices and provide their indexes.

11.3. Evaluation Details

In the clothing simulation benchmark, we compared four different clothing simulators: LBS, PBNS [4], NCS [5], and HOOD [17]. The training and evaluation of each method were conducted using the SMPLX model, which provides more details in visualization. The final evaluation is done on four types of garments (Upper, Outer, Dress, and Lower), with each having 2 garments and 6 sequences in total. For qualitative evaluation, we employed Chamfer distance and stretching energy, scaling vertex positions by a factor of 100 to use centimeters as the unit.

The Chamfer distance, shown in equation 7, is computed by summing the squared distances between nearest neigh-

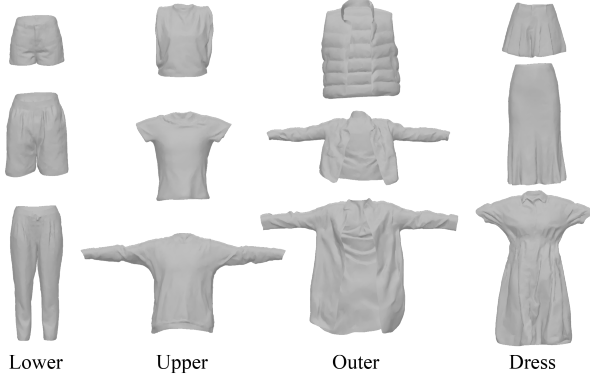


Figure 20. **Garment templates used for clothing simulation.** We extract four types of Garment templates from T-pose scans.

bor correspondences of two-point clouds. We denote the sampled points on simulation and ground-truth meshes as X and Y , respectively, with N_* representing the amount of sampled points, set to 100,000 in our experiment.

$$d_{CD} = \frac{1}{N_x} \sum_{x \in X} \min_{y \in Y} \|x - y\|_2^2 + \frac{1}{N_y} \sum_{y \in Y} \min_{x \in X} \|x - y\|_2^2 \quad (7)$$

The stretching energy, widely used in mass-spring-based simulators, is computed as equation 8, where N_e is the total number of edges, e_i and \bar{e}_i are the lengths of the edge i in the current frame and the template respectively.

$$E_{str} = \frac{1}{N_e} \sum_i \|e_i - \bar{e}_i\|^2 \quad (8)$$

We provide more details on implementing each method: **LBS** blends joint transforms with skinning-weights. For each garment template, we find the nearest body node on the canonical SMPLX human, and get the skinning weights on this point. Then, we follow the same forward LBS process in SMPLX to get deformed template meshes.

PBNS and **NCS**, both are deformation-based methods, predict vertex-wise deformation on the template and employ LBS to transform the deformed garment into desired poses. Given their "One model for one garment" nature, we trained each garment from scratch. We also used identical AMASS sequences mentioned in the NCS paper to ensure fairness. As both PBNS and NCS developed using SMPL, we made slight adjustments to the data-loading pipeline to ensure their compatibility with SMPLX. And we assigned zero poses to joints that are exclusive in SMPLX.

Meanwhile, we also kept the same training settings used in their original papers. For PBNS, default parameters were used, and each garment underwent training for 20-50 epochs to ensure convergence. For NCS, a batch size of 2048 was employed across all training instances, as suggested in their paper. In the case of tight garments, default

parameters were maintained with a temporal window size of 0.5 and 10 iterations for blend weights smoothing. In the case of loose garments like outerwear and dresses, we made slight parameter adjustments for stable training, typically using a temporal window size of 0.75 and 1, with 50 iterations for blend weights smoothing, as suggested by the author in a GitHub issue.

HOOD, as a simulation-based method, predicts physically realistic fabric dynamics and is agnostic to garment topology. Hence, we directly used a pre-trained publicly available model to evaluate our garments. Unlike the deformation-based methods, which convert the template in canonical pose to any pose instantly, HOOD predicts garment motion frame by frame. Therefore, to apply our canonical template for simulating each sequence, we have to convert the template into the pose of the first frame. In the HOOD paper, they used LBS to convert templates, which works for tight synthesized garments. However, for our real-world garments, it usually results in large stretching on mesh, especially around joint areas. Therefore, alternatively, we insert extra frames from the canonical pose to the first frame and simulate the prolonged sequence to get a natural transform from the canonical pose. The first poses for all sequences in our dataset are in A-pose. Generally, we insert 30 frames to transfer from canonical to A-pose, which makes it slow enough for the garment to stay in rest status with minimum dynamics.

11.4. HOOD*: Material Optimization

HOOD provides 4 local material parameters for each vertex, including μ and λ evaluating the ability of stretching and area preservation, mass m computed from the fabric density, and the bending coefficient $k_{bending}$ penalizing folding and wrinkles. For each edge, there are three material parameters, including μ , λ , and $k_{bending}$. Assuming we have v vertices, e edges, and coarse edges in total, we define the material parameters as $\mathcal{M} \in \mathbb{R}^{4v+3e}$.

In the fine-tuning process, we freeze the pre-trained HOOD model \mathcal{H} and only update material parameters \mathcal{M} . Using all 6 sequences of each garment for training, we feed them into model f to get simulated outputs. Then, with Ground Truth garment mesh G , we compute Chamfer distance and stretching energy, as described in equation 9.

$$\mathcal{L} = \mathcal{L}_{CD}(f(\mathcal{M}, \mathcal{H}), G) + w \mathcal{L}_{Estr}(f(\mathcal{M}, \mathcal{H}), G) \quad (9)$$

We used the stretching energy from HOOD and set w as 1 in our experiments. Chamfer distance \mathcal{L}_{CD} is described in equation 10, measuring the average distance between simulation and ground-truth garment. We use V_* , ($* \in [s, g]$) to represent the simulated and ground truth vertices and use N_* as the total number of vertices.

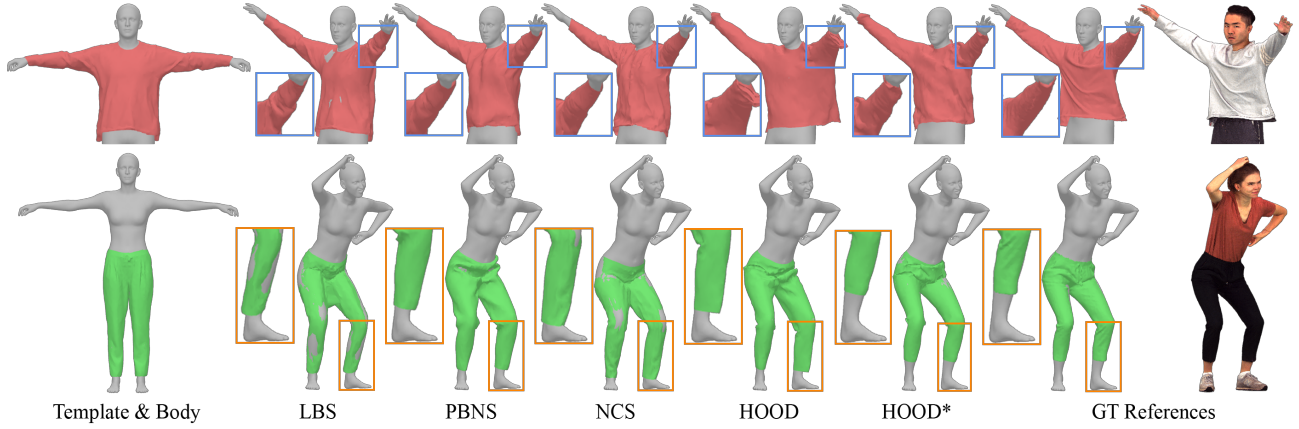


Figure 21. **Additional qualitative results for clothing simulation.** Left are templates used for simulations. Right are simulations and ground-truth scans. HOOD presents more dynamic while getting overly stretched. HOOD* matches well with ground truth.

$$\mathcal{L}_{CD} = \frac{1}{N_s} \sum_{x \in V_s} \min_{y \in V_g} \|x - y\|^2 + \frac{1}{N_g} \sum_{y \in V_g} \min_{x \in V_s} \|x - y\|^2 \quad (10)$$

For each garment, we trained with Adam Optimizer with a learning rate of $5e-4$. And it usually takes 50 epochs to converge. Generally, HOOD* gets a much lower distance compared to ground truth mesh quantitatively, and also performs more natural fabric dynamics qualitatively.