

Deformable 3D Gaussian Splatting for Animatable Human Avatars

HyunJun Jung¹, Nikolas Brasch¹, Jifei Song², Eduardo Pérez-Pellitero², Yiren Zhou², Zhihao Li², Nassir Navab¹, Benjamin Busam^{1,3}

¹ Technical University of Munich

² Huawei Noah’s Ark Lab

³ 3dwe.ai

{hyunjun.jung, b.busam}@tum.de



Figure 1. **ParDy-Human** constitutes an explicit dynamic human avatar that can be re-posed via SMPL [35] parameters. It utilizes the design of a deformable version of 3D Gaussian Splatting [26]. ParDy-Human, unlike existing implicit methods, can be trained with significantly fewer camera views and less human poses. While being free of ground truth mask for training, it generalizes well to novel human poses as shown in the above reposed results on the individuals from ZJU-MoCap [47] and THuman4.0 [78] datasets.

Abstract

Recent advances in neural radiance fields enable novel view synthesis of photo-realistic images in dynamic settings, which can be applied to scenarios with human animation. Commonly used implicit backbones to establish accurate models, however, require many input views and additional annotations such as human masks, UV maps and depth maps. In this work, we propose ParDy-Human (Parameterized Dynamic Human Avatar), a fully explicit approach to construct a digital avatar from as little as a single monocular sequence. ParDy-Human introduces parameter-driven dynamics into 3D Gaussian Splatting where 3D Gaussians are deformed by a human pose model to animate the avatar. Our method is composed of two parts: A first module that deforms canonical 3D Gaussians according to SMPL vertices and a consecutive module that further takes their designed joint encodings and predicts per Gaussian deformations to deal with dynamics beyond SMPL vertex deformations. Images are then synthesized by a rasterizer. ParDy-Human constitutes an explicit model for realistic dynamic human avatars which requires significantly fewer training views and images. Our avatars learning is free of additional annotations such as masks and can be trained with

variable backgrounds while inferring full-resolution images efficiently even on consumer hardware. We provide experimental evidence to show that ParDy-Human outperforms state-of-the-art methods on ZJU-MoCap and THuman4.0 datasets both quantitatively and visually. Our code is available at <https://github.com/Junggy/pardy-human>.

1. Introduction

Creating animatable human avatars from images and videos is a popular tasks in computer vision and graphics due to its applications in animation, virtual reality and gaming. Early works [1, 16, 53] are developed around a CNN or MLP based generator that learns 3d human mesh prediction conditioned on a parameterized representation of humans. While providing a mesh, early approaches produce avatars with only low frequency details for both shape and texture. Recent works utilize implicit representations combined with volume rendering [8, 31, 46] which enables photo-realistic rendering with a significantly higher amount of details at the cost of denser camera views and human poses during training.

Most recently, Kerbl *et al.* [26] propose an explicit

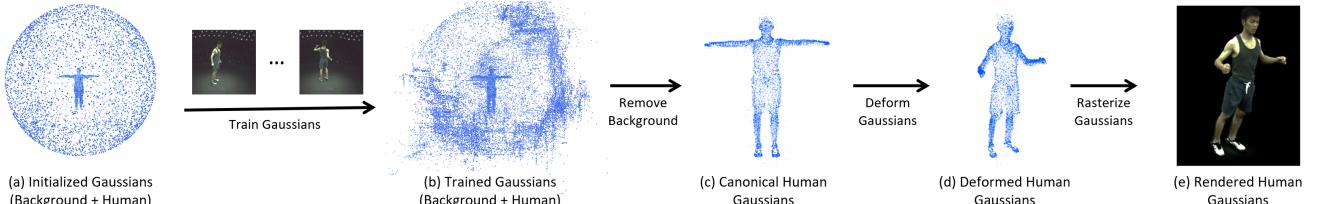


Figure 2. **Overview of Avatar Generation Framework.** (a) ParDy-Human starts by initializing Gaussians on a sphere for the background and a canonical SMPL [35] mesh for the human. (b) the Gaussians are updated during the training. (c) For inference, the background Gaussians are removed leaving only the avatar. (d) the Canonical human Gaussians are deformed according to the SMPL vertex deformations and learned residual refinement. (e) the deformed Gaussians are rasterized to synthesize an image output under a given pose.

method utilizing point based rendering with 3D Gaussians to enable real-time synthesis of static scenes at high quality. Other scholars [36] show its potential in dynamic scenarios involving humans.

In this paper, we present ParDy-Human (Parameterized Dynamic Human Avatar), a novel approach to leverage 3D Gaussian Splatting for avatar generation. We achieve this by updating the canonical T-pose Gaussians during training via posing and deposing with the help of SMPL parameters [35] and a specialized deformation module. We design human motion as two consecutive deformations, a naked body deformation followed by motion related cloth movement.

Canonicalized T-pose human Gaussians are deformed via SMPL parameters as a naked body deformation and a MLP associated with the joint’s geometric distances determines the residual deformation corrections arising from garment motion. We initialize the scene with two set of point clouds, one for canonical SMPL vertices representing the human and one randomly covering a spherical surface representing the background. This split allows the separation between the human and the background, enabling mask-free training while being capable of removing the background at inference time.

We show that ParDy-Human can be trained without a segmentation mask using significantly fewer cameras and images compared to existing methods. Experiments proof the realism of rendered avatars in novel poses even with as little as a single monocular video. We enable image synthesis of full-resolution even on consumer-grade hardware.

In summary, we contribute:

1. A method for **deformable 3D Gaussian splatting of human avatars**, a parametrized fully explicit representation for **dynamic animation** of humans that requires significantly **fewer views and images** for training.
2. ParDy-Human provides a **mask-free training** approach that automatically separates background and human.
3. An efficient pipeline that allows **significantly faster inference** for **full-resolution renderings**.

2. Related Work

2.1. View Synthesis of 3D Scenes

3D scenes are represented in a plethora of different ways. Classical explicit representations include point clouds, voxels, and meshes, while classical **implicit representations** utilize signed distance fields (SDFs). Historical approaches involve lightfields [28, 64] and coloured voxel grids [58, 63]. Implicit neural representations leveraging neural networks to encode the scene [38, 43]. **Neural Radiance Fields (NeRFs)** are a recent popular advancement due to their ability for high-fidelity view synthesis from RGB [2, 39, 40] or RGBD [11, 23, 51] images. Accurate camera poses are crucial to train NeRFs. Typical setups extract poses using structure-from-motion [56] which is limited in homogeneous regions and to visual ambiguities [37]. Relaxed methods like BARF [32], Nerf– [68], or dynamic SLAM [25] can help to robustify camera pose retrieval. In line with existing literature [8, 30, 44, 69], we adopt the convention of assuming either static cameras or posed images.

In comparison, **3D Gaussian splatting** has emerged very recently as a compelling **explicit method** to represent 3D scenes [26]. This technique capitalizes on the co-produced COLMAP point cloud to represent the scene using 3D Gaussians, optimizing their anisotropic covariance and incorporating visibility-aware rendering.

2.2. Dynamic Scenes

Dynamic scenes are also considered both to render novel camera views of humans [5] and their reconstruction [60]. A series of works by researchers from Microsoft [9, 12, 42] introduces content-agnostic streamable free-viewpoint systems with multi-view RGBD cameras.

Also NeRFs have been extended to the temporal domain from synchronized **multi-view** video [30, 49]. Nerfies [44] addresses the challenging case of a **moving monocular camera** with limited scene dynamics as video input and optimizes a 5D NeRF. HyperNeRF [45] offers a higher-dimensional representation that also accounts for topological changes. Efficient 4D NeRF rendering methods allow

interactive frame rates [59] with high resolution [73]. 3D Gaussian splatting has also been extended in time [70, 74] by representing generic 4D scenes and leveraging temporal connections. Additionally, 3D Gaussians can be densely tracked over time [36]. In our work, we take a different approach, utilizing a deformation field in conjunction with a parametric human model. This allows us to associate dynamic 3D Gaussians in a canonical representation, enabling the high-fidelity learning of 3D avatars that can be freely animated using an underlying parametric 3D human model.

2.3. Humans in 3D

The representation of humans in 3D space has seen significant advancements through the development and extension of spatial human body models in the past decade. Early methods focused on predicting major body joints [17] and keypoints including hand, face, and foot to retrieve human skeleton motion [4]. With SMPL [35] a widely adopted **parametric shape representation** has been introduced, utilizing skinning techniques and blend shapes optimized on real 3D body scans, providing realistic representations for non-clothed humans.

These advancements have been paralleled by substantial efforts in dataset acquisition, leveraging advanced hardware setups, accurate annotations, and synchronized multi-view capture. Early work by Kanade et al. [24] utilized a multi-camera dome for reconstruction. Subsequent datasets like Human3.6M [18], ZJU-Mocap [47], THUman4.0 [78], and ActorsHQ [19] build on these experiences and provide human captures with multi-view cameras. However, it is important to note that even with advanced hardware, obtaining high-quality 3D annotations remains a costly endeavor [19, 22, 65].

In the context of registering 3D shapes, **scene-agnostic** methods typically rely on point cloud descriptors [13, 15, 52, 55] that can be used for both 3D and 4D descriptor matching [10, 50, 54, 75, 76], or on functional maps applied to known shapes [3, 33]. Establishing the connection to mentioned parametric models and finding correspondences to canonical parts plays a crucial role in enabling subsequent 3D avatar animation.

2.4. Animatable Human Avatars

The pursuit of animatable human avatars has seen a progression from static 3D human surface recovery to more intricate neural approaches guided by 3D human models. Deepcap [16] focuses on deformable pre-scanned humans for monocular human performance capture, leveraging weak multi-view supervision, while Monoperfcap [72] provided the first marker-less method for temporally coherent 3D performance capture of a clothed human from monocular video. The emergence of neural methods specializing in human rendering marked a significant shift. Models like

Neural Body [49] optimize latent features for each mesh vertex, leveraging an SMPL model. Approaches, including Neural Articulated Radiance Field [41], A-NeRF [61], and TAVA [29] provide solutions to learn human shape, appearance, and pose using skeleton input. PoseVocab [31] further learns an optimal pose embedding for dynamic human motion from multi-view RGB by factorizing global pose into embeddings for each joint together with a strategy to interpolate features for fine-grained human appearance.

For monocular video, Selfrecon [20] combines explicit and implicit representations using SDFs for 3D human body representations, while NeuMan [21] splits the scene content to train two NeRFs one for the static background and one for the human. Animatable NeRF [46] and HumanNerf [69] integrates motion priors in the form of SMPL [35] parameters to regularize field prediction. The canonical representation is warped into the target frame by Arah [66], SNARF [7] as well as Scanimate [53] while others [34, 71] use backward sampling from the observation. MonoHuman [77] models deformation bi-directionally to retrieve view-consistent avatars. Typically garment deformation is tied to the model, however, Zheng et al. [78] use a residual model correction to fit dynamic cloth deformations with local fields attached to pre-defined SMPL nodes. Inherent texture fidelity limitations have been additionally addressed with generative models [62, 67]. All these approaches prioritize rendering quality but also grapple with high computational costs for both training and testing. NeRFs are great to generate texture on the fixed geometry [6]. UV Volumes [8] leverages this to establish a rendering pipeline for human avatars by generating texture and its mapping separately. The popularity of implicit neural fields for scene representations undoubtedly lead to impressive results. However, these methods inherently struggle to represent empty space efficiently and acceleration is limited by efficient sampling for ray-marching and the structured representations of the scene. Using explicit 3D Gaussians allows us to overcome these bottlenecks with an unstructured representation for more efficient training with higher quality and faster inference speed.

Concurrent to ParDy-Human, D3GA [79] has just proposed to compose 3D Gaussians embedded in tetrahedral cages for cloth layer decomposition of human avatars. They use multi-view video captures from 200 synchronized cameras as input while ParDy-Human works with as little as one monocular view.

3. Method

To achieve lifelike animatable human avatars, we need to map low-frequency features such as human pose and shape to high-frequency appearance details. ParDy-Human leverages 3D Gaussian Spatting [26] (3D-GS) as the foundational framework for three primary reasons. **1.** 3D Gaus-

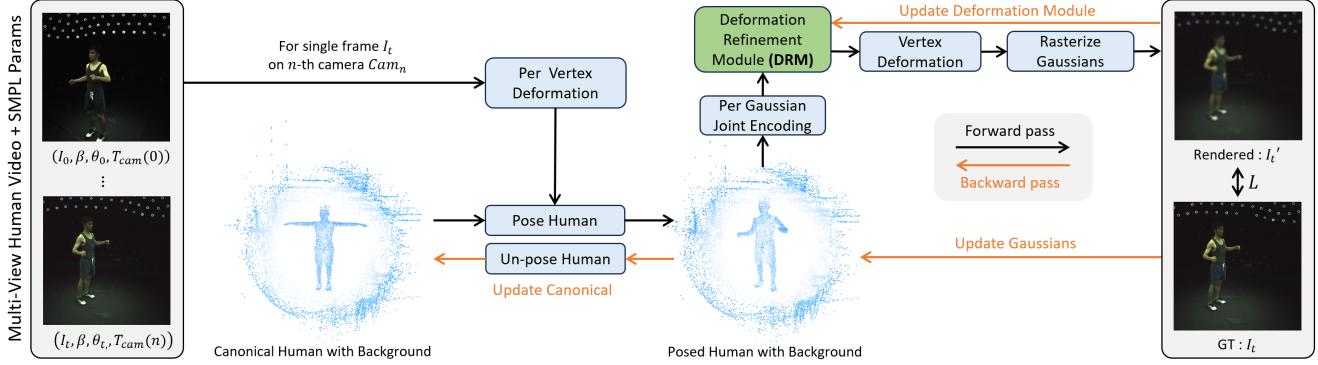


Figure 3. Training Pipeline Overview. ParDy-Human is a fully explicit animatable human representation based on 3D Gaussians [26]. Information from images I_j , $1 \leq j \leq t$ of the n -th camera Cam_n are integrated into the avatar by using the camera pose $T_{Cam}(j)$, human shape β , and pose θ_j parameters (left). Correspondences between Gaussians of a Canonical and Posed Human are established by a Per Vertex Deformation Module (centre to left, black arrows). Residual corrections of Gaussians are performed using a Deformation Refinement Module (DRM) (centre to right, black arrows) before image synthesis through rasterization (right). The rendered output can then be compared to ground truth input images to calculate gradients and update both the DRM and human avatar (orange arrows)

sians, initialized from coarse point clouds can be associated to vertices of a parametric human model. **2.** The point cloud structure of Gaussians facilitates straightforward deformation based on human parameters. **3.** 3D Gaussians enables swift and efficient rendering of high-frequency details. Our pipeline overview is depicted in Fig. 3.

3.1. 3D Gaussian Initialization

3D Gaussians are an explicit scene representation that contains the following attributes: geometric center $\mathbf{P}_i \in \mathbb{R}^3$, rotation $\mathbf{R}_i \in SO(3)$, size, scale, opacity, spherical harmonics (SH). Images can be rendered from this representation with a point-based rendering scheme. While being similar to surfels [48], Gaussians incorporate a 3-dimensional scale vector to create elliptical shapes of different size and shape and SH can model view dependent color variations. Originally, Gaussians are first initialized with SfM [56, 57] point clouds and then trained with a specific adaptive density control scheme that splits, clones and prunes the Gaussians during training depending on their size and gradient magnitude. This takes an important role in optimizing the number of Gaussians that is denser in detailed regions and coarser in larger homogeneous regions.

Unlike the original implementation [26], we use two different point clouds (Fig. 2 (a)) to initialize Gaussians and include additional features: *parent index i* and *surface normal n_i* (only for human). For the background, we initialize random points on a spherical surface and label parents with $i = b$.

For the human, we generate a parametric human mesh in canonical T-pose $M_{\theta_c}^{SMPL}$ using SMPL [35], then initialize the canonical Gaussians G_c at the center of each mesh face and assign extra features such as the face normal and parent as an index of the face in the mesh.

3.2. Posing Gaussians

After initialization, each Gaussian is deformed according to its parent using Per Vertex Deformation (PWD). Given the t -th human pose parameter θ_t , a new human mesh $M_{\theta_t}^{SMPL}$ is generated from SMPL [35] and per face human deformation D_t^i is calculated for all faces $i \in \mathcal{F}$ of the SMPL model by comparing with the canonical human face:

$$\mathbf{D}_t = (D_t^1, \dots, D_t^F) = PWD(M_{\theta_c}^{SMPL}, M_{\theta_t}^{SMPL}). \quad (1)$$

The deformed Gaussians G_d can then be obtained by transforming the canonical Gaussians G_c into their new locations according to \mathbf{D}_t . Associated locations $\mathbf{P}_i \in \mathbb{R}^3$ and rotations $\mathbf{R}_i \in SO(3)$ can be established through D_t^i via:

$$G_d(\mathbf{P}_i, \mathbf{R}_i) = D_t^i(G_c(\mathbf{P}_i, \mathbf{R}_i)).$$

3.3. Deformation Refinement

For humans with tight clothing, it is enough to deform with an SMPL [35] model. However, in general garment motion causes more deformations dependent on the human posture. To obtain higher fidelity renderings, we include a Deformation Refinement Module (DRM). The module is designed with a small set of MLPs that take the distance between the center of the Gaussian \mathbf{P}_i and the human joints \mathbf{J}_t of the SMPL model to predict a residual refinements for PVD deformations. In practice, using shorthand notation $G_d(\mathbf{P}_i)$ for the location of the Gaussians after the deformation, the feature

$$\mathbf{E}_d^i = G_d(\mathbf{P}_i) - \mathbf{J}_t \quad (2)$$

is flattened and fed into an MLP to calculate the corrected per Gaussian transformation $\mathbf{D}_r = (D_r^1, \dots, D_r^F)$ with

$$D_r^i = DRM(\mathbf{E}_d^i). \quad (3)$$

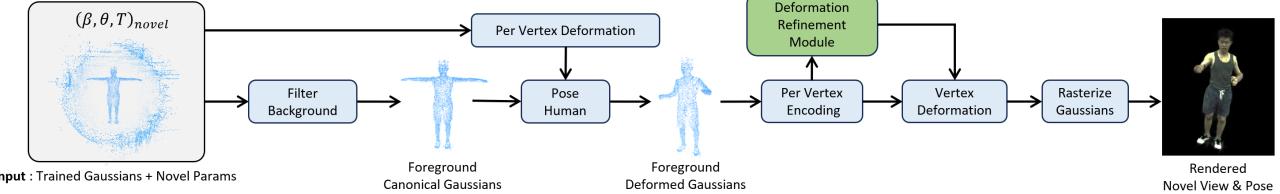


Figure 4. Inference Pipeline Overview. During inference time, we first filter out the background Gaussians (left) and then deform the canonical human avatar. A coarse deformation is done first using SMPL [35] parameters followed by the DRM correction (centre to right). The output is an animated human without background (right).

The detailed architecture is specified in the supplementary material. Finally D_r^i is applied to G_d to obtain the refined Gaussians G_r as

$$G_r(\mathbf{P}_i, \mathbf{R}_i) = D_r^i(G_d(\mathbf{P}_i, \mathbf{R}_i)). \quad (4)$$

This refinement step plays a crucial role in producing high fidelity renderings of clothed human avatars. Fig. 7 shows the impact of this residual correction step.

3.4. Spherical Harmonics Direction

In 3D-GS [26], Spherical Harmonics (SH) are used to incorporate viewing angle dependent effects. The direction $\mathbf{d}_i \in \mathbb{R}^3$ is calculated as the relative vector from the camera center $\mathbf{c} \in \mathbb{R}^3$ to each Gaussian via

$$\mathbf{d}^i = \frac{\mathbf{c} - \mathbf{P}_i}{\|\mathbf{c} - \mathbf{P}_i\|_2}. \quad (5)$$

While this assumption holds for static scenes, it fails in dynamic scenarios where the same point can have the identical relative direction to a camera but might be viewed from a different angle due to surface rotations. We correct this by including surface normal information calculated from the SMPL [35] prior mesh (Sec. 3.1). We apply the rotations $\mathbf{R}_i^t, \mathbf{R}_i^r$ from the deformations $\mathbf{D}_t, \mathbf{D}_r$ sequentially on the surface normal \mathbf{n}_i to obtain more realistic corrected directions $\mathbf{d}_c^i \in \mathbb{R}^3$ by

$$\mathbf{d}_c^i = \mathbf{R}_i^r \cdot \mathbf{R}_i^t \cdot \mathbf{n}_i. \quad (6)$$

We use both direction definitions for rasterization. Eq. 5 is used on the static background while Eq. 6 is used for the dynamic foreground.

3.5. Un-Posing Gaussians and Updating Parents

Once the Gaussians are updated, they are transformed back to the canonical space in order to be posed again with the next set of parameters. Un-posing is done by following the deformation in the inverse order. From the updated Gaussians, we apply the inverse of \mathbf{D}_t and \mathbf{D}_r sequentially to transform G_r into the updated canonical Gaussians G_c .

During training, the number of Gaussians and their centers may change and the parent index i has to be updated

accordingly. When Gaussians are added via splitting and cloning [26], the parent index is passed to the new Gaussians. Parent assignments are updated every 1000-th iteration by finding the face with the shortest Euclidean distance in the canonical pose. This prevents Gaussians from diverging too far from their parents during the update of \mathbf{P}_i . Gaussians with a distance above a threshold τ from the surface are assigned to the background.

3.6. Loss Design and Inference Pipeline

Three losses are applied throughout the training: L1 loss \mathcal{L}_1 , structural similarity loss \mathcal{L}_{SSIM} , and perceptual similarity loss \mathcal{L}_{LPIPS} with AlexNet [27] as the base. The losses are balanced by the weights $\lambda_{L1}, \lambda_{SSIM}$, and λ_{LPIPS} .

The overall training objective reads as follows:

$$\mathcal{L} = \lambda_{L1}\mathcal{L}_1 + \lambda_{SSIM}\mathcal{L}_{SSIM} + \lambda_{LPIPS}\mathcal{L}_{LPIPS}. \quad (7)$$

During inference time (cf. Fig. 4), our pipeline filters the background Gaussians based on their parents. Taking a novel camera pose and human pose as input the Gaussians are deformed based on the SMPL model first and then adjusted via the DRM. The deformed Gaussians are then passed to a rasterizer to render the human under a given camera pose.

4. Training and Implementation Details

For training, we calculate the loss by comparing the rendered image with the ground truth image and then update the Gaussian features [26] as well as the DRM based on the resulting gradients. As both updates involve displacements of Gaussians, we split the training into multiple stages and mutually update these modules.

4.1. Training Schedule

We begin by updating both DRM and Gaussians at the same time until 10k-iterations to initialize them. After that, we mutually fix either the Gaussians or DRM and update its counterpart. We switch the weight updates between them every 5k-iterations until 100k-iterations. During the Gaussian part, alongside the Gaussian features update, we follow the original Gaussian adaptive update schemes [26] until 50k iterations.

5. Experiments

We first describe the experimental setup including datasets and baseline methods, then show qualitative and quantitative results as well as train and test efficiency comparisons.

For all our experiments, we use the parameters $\tau = 10$ cm as well as weights $\lambda_{L1} = 0.6$, $\lambda_{SSIM} = 0.4$, and $\lambda_{LPIPS} = 0.4$.

5.1. Data Selection and Evaluation Setup

To illustrate the background handling of ParDy-Human, we choose two human datasets with different background setups (ZJU-MoCap [47] and THUman4.0 [78]). The **ZJU-MoCap** [47] dataset features a motion capture sphere setup with 21-23 cameras with 9 human subjects performing actions of different length (600-2500 frames) in front of a relatively dark background. The data helps to understand how our method can differentiate the human from the background in a low light scenario where human subjects often blend into the dark background. The data of **THUman4.0** [78] features 24 cameras with only 3 human subjects performing relatively long sequences of actions (2500-5060 frames) in an uncontrolled and partially unbounded indoor scene. We specifically select this dataset to evaluate our performance in less constrained, more realistic scenes.

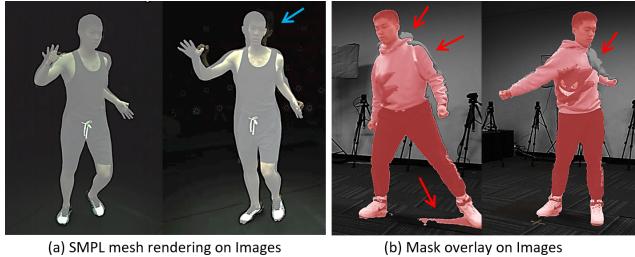


Figure 5. Issues in Datasets. Some scenes in the ZJU dataset [47] suffer from inaccurate extrinsic calibration. The rendered SMPL meshes on these images are not consistent over cameras (a). On the other hand, the mask in the THUman dataset [78] suffers from over- and under-segmentation artifacts depending on the lighting conditions (b).

To analyse the performance with only a few views and sparse image sampling, we train the method using seven views and use every 50-th frame per camera as training input. We further analyse unseen human poses that are significantly different from the training poses (cf. Sec. 5.3). Since the calibration quality of the camera views in the ZJU dataset [47] varies severely (cf. Fig 5), we select the best seven views upon visual inspection. We equally take seven cameras for THUman4.0 [78] to have a consistent number of training views. We specify the cameras that are used for training and testing in the supplementary material.

5.2. Baseline Selections

Two of the most recent state-of-the-art methods are selected as baselines: UV-Volumes [8] and PoseVocab [31]. Although both methods tackle dynamic human reconstruction and reposing using implicit volume rendering technique, they are based on orthogonal approaches.

UV-Volumes [8] encodes SMPL [35] vertices directly into a volume using a sparse 3D CNN. This approach breaks down the human rendering task into two smaller tasks, volume and texture generation. The volume generator conditioned on SMPL [35] vertices first generates a UV volume via a sparse 3D CNN that renders a UV map from a given camera pose. A consecutive texture generator predicts the texture with SMPL [35] parameters and body parts mapped to the image using a UV map. Training requires a UV map and body part segmentation as predictions of DensePose [14].

PoseVocab [31] in comparison encodes features in a canonical pose and samples corresponding features when a human pose is given. Given human poses with joint angles, a pose vocabulary (PoseVocab) in canonical human pose is constructed and takes pose embeddings as queries. When a query pose vector is given, the pose is converted into an embedding and features are sampled from the vocabulary. A volume rendering module synthesizes a human image under the corresponding pose. PoseVocab requires depth maps for full training, which forces the training to be split into two parts. Depth prediction is learned before the full network is trained.

5.3. Quantitative and Qualitative Evaluation

For a quantitative evaluation, we follow the literature and use three metrics: Signal to noise ratio (**PSNR**, higher is better), Structural Similarity (**SSIM**, higher is better), Perceptual Loss (**LPIPS**, lower is better). For the evaluation, we use the furthest in between frames from the training sequence (i.e. every $(50n + 25)$ -th frame) and evaluate with a mixture of used and novel camera views. See supplementary material for more details.

Ours vs Baselines. The result in Tab. 1 shows that ParDy-Human gives much better LPIPS score and similar or better SSIM compared to the baselines indicating that novel pose synthesis is visually perceived as a more human like and maintains good details. The difference is clearly visible in Fig. 6. The PSNR metric on the ZJU dataset [47] for ParDy-Human, however, is lower on average compared to UV-Volumes [8]. The ZJU dataset is recorded in dim lights that leads to a varying shade effect depending on the orientation and pose of the human performer. In this case, UV-Volumes [8]'s dedicated texture generation network has an advantage. It remembers the orientation and pose dependent shading which are reproduced at inference time while the shading in our pipeline does not vary severely.

ZJU 313, 315 in Fig. 6 indicate that our method pro-

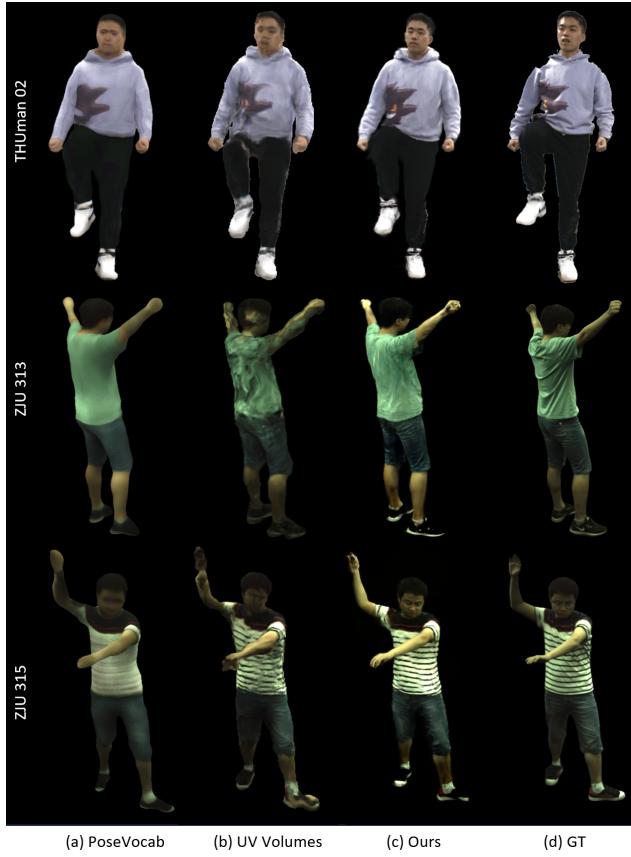


Figure 6. Qualitative Evaluation in a sparse view training setting. PoseVocab [31] loses details if provided with a sparse setup. UV-Volumes [8] produces more realistic shading by leveraging a texture map, however, suffers from geometric and texture artifacts especially if the inferred human pose is far from the training poses. Ours is limited on generating exact shading (ZJU [47] 313, 315), but produces more realistic and detailed humans in novel poses.

duces an average shading that can be brighter than GT image. While UV-Volumes [8]'s replication of a similar shading leads to better PSNR values, it fails to correctly synthesize the human shape. In comparison, when training in bright lighting (i.e. THUman [78], 00-02 in Tab. 1), ParDy-Human outperforms the PSNR values of UV-Volumes [8] by a large margin.

Baseline Comparison. UV-Volumes [8] directly encodes feature volumes using prior SMPL [35] vertices. It specializes to replicate seen human poses while it struggles with larger pose deviations. PoseVocab [31] encodes features in a canonical volume which generalize better to novel human poses, but requires many human poses to train. This can be observed in Tab. 1 and Fig. 6. UV-Volume replicates shading easily (PSNR \uparrow) while the human shape breaks (LPIPS \uparrow). PoseVocab's rendering lack details (PSNR \downarrow) due to the limited training images while the overall human shape is preserved (LPIPS \downarrow).

Table 1. Quantitative Evaluation for Novel Pose Synthesis. We compare our method against two recent state of the art works, UV-Volumes [8] (UVV) and PoseVocab [31] (PV), on ZJU-Mocap [47] (row 1-4) and THUman4.0 [78] (row 5-7) dataset.

Scene	PSNR \uparrow			SSIM \uparrow			LPIPS \downarrow		
	UVV	PV	Ours	UVV	PV	Ours	UVV	PV	Ours
313	29.18	22.40	28.33	0.960	0.961	0.960	0.056	0.048	0.043
315	26.49	21.96	24.51	0.957	0.954	0.955	0.043	0.044	0.034
377	29.32	23.30	28.44	0.969	0.970	0.974	0.047	0.035	0.033
394	29.12	22.74	29.28	0.957	0.962	0.964	0.059	0.047	0.040
00	28.33	26.30	30.32	0.961	0.970	0.975	0.048	0.038	0.028
01	25.88	25.07	28.65	0.944	0.962	0.963	0.064	0.061	0.047
02	24.49	23.97	26.54	0.942	0.959	0.961	0.080	0.062	0.057

Table 2. Ablation Study. We run two ablations for DRM and masks. Ours wo.D denotes ParDy-Human trained without DRM while w.M indicates training with mask. In general, our full pipeline (Full) does not perform best quantitatively under these metrics, however, gives clearly better visual results (Fig. 7, 8).

Scene	PSNR \uparrow			SSIM \uparrow			LPIPS \downarrow		
	Full	wo.D	w.M	Full	wo.D	w.M	Full	wo.D	w.M
315	24.54	25.45	25.82	0.955	0.953	0.962	0.034	0.038	0.029
377	28.44	28.96	27.17	0.984	0.976	0.960	0.033	0.031	0.058
02	26.54	26.77	26.67	0.961	0.962	0.960	0.057	0.056	0.046

Rendering Efficiency For all tests, we use a consumer grade laptop (i9 + RTX3080 max-Q) and measure the inference time. ParDy-Human and PoseVocab [31] were capable of rendering at full resolution. ParDy-Human takes roughly 0.3 sec while PoseVocab requires 15-25 sec per image, which is a speed up of $> 50\times$. Due to the high GPU memory requirement of UV-Volumes, we were not able to render at full resolution. At 30% resolution, the method shows a similar inference time compared to our approach.

5.4. Ablation Study

Residual Correction. The DRM module plays a crucial role in producing texture details when rendering clothed humans. Canonical Gaussians that only deform with the SMPL model are not flexible enough to deal with garment deformation. If trained without this correction, texture details become blurry (Fig. 7a,c) and ghosting effects at surface boundaries arise (Fig. 7b). This obvious improvement is not well quantified using current metrics (Tab. 2, Full vs wo.D) despite its clear visual advantage (Fig. 7, green vs red).

Background The annotation of masks in dynamic sequences is non-trivial resulting in imperfect masks (Fig. 5). These impact training and evaluation. ParDy-Human can be trained without masks. We perform an extra experiment utilizing masked images to show the impact of the masks on the final result. Fig. 8 shows that unreliable masks can create black (red box) or white Gaussians (orange box) around the human. This artifact disappears in the mask-free training setup. Since evaluation is performed on incorrectly masked images of the dataset, this is not quantified in Tab. 2

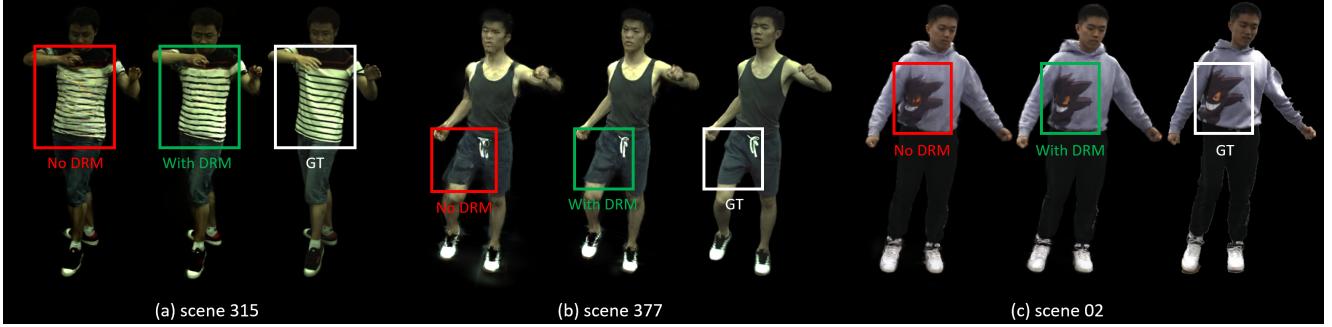


Figure 7. **Effect of the Deformation Refinement Module.** The residual correction of the DRM plays an important role in our pipeline to cope with garment deformations. Training without it (red) results in loss of texture details (a). DRM training (green) reduces ghosting effects (b) and improves texture boundaries (c). The ground truth is illustrated in white.

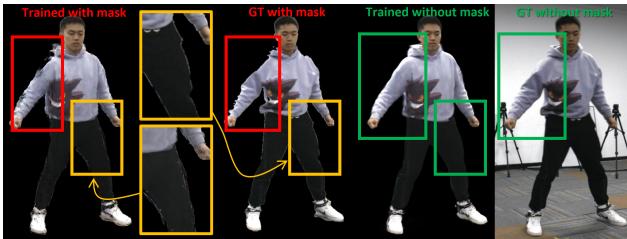


Figure 8. **Training with Mask Annotations.** When trained with human masks (left), mask errors (second from left) manifest in black (red box) and white (orange box) Gaussians around the body. If ParDy-Human is trained without these annotations (second from right), it produces higher quality geometry of the human shape.

(Full vs w.M.).

Training with a Single Camera. We have seen that our method requires significantly fewer views and human poses for training. However, setting up synchronized and calibrated cameras for data capture can be challenging (cf. Fig. 5a). We train our method with only a monocular fixed camera to simulate a real-life recording scenario. To compensate for the 7 times fewer views, we train with every 7th frame of the video sequence. Fig. 9 shows that our method successfully creates an animatable avatar when the subject covers a 360 degree motion. On the other hand, our explicit method does not hallucinate unseen sides. A full animated sequence can be found in our supplementary video.

6. Conclusion and Limitations

In this paper, we propose ParDy-Human, an explicit method for animatable human avatar generation from RGB images. We use deformable 3D Gaussians controlled via SMPL parameters and show that our method can be trained with significantly fewer images, views and without masks compared to prior works. Even without mask annotations, the method generates high quality 3D digital twins of humans that can be re-posed and rendered from arbitrary viewpoints. Still, uni-coloured garments with

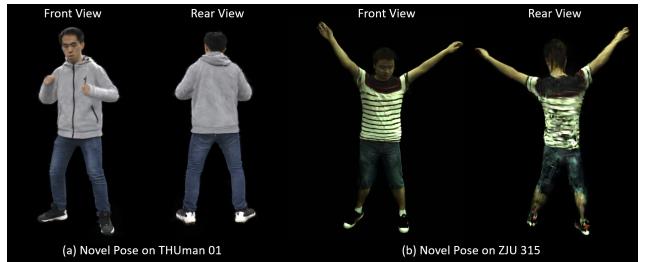


Figure 9. **Monocular Training Setup.** Our method can be trained with only a single static camera. Our avatar can be re-posed well in novel views (a) if the individual turns around fully during the training. Unseen visual appearances from areas not seen during training are not recovered (b).



Figure 10. **Gaussian Spike Artifact.** Training on uni-coloured garments can cause a reduction of the number of Gaussians while increasing their size. This manifests in Gaussian spikes arising at bent geometry boundaries in unseen poses.

large deformations can pose problems for animations of out-of-distribution poses as shown in Fig. 10. Our method efficiently infers full-resolution images even on a consumer laptop which helps democratizing the work, but also poses ethical concerns due to the fact that a single input video is enough to generate a 3D digital human copy. It shares the issue with all other 3D avatar generators that can be abused to create fake videos of real people. We still firmly believe that the idea of deformable 3D Gaussian fused with a parametric model can inspire other works.

References

- [1] Thiendo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1186, 2019. [1](#)
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. [2](#)
- [3] Lennart Bastian, Alexander Baumann, Emily Hoppe, Vincent Bürgin, Ha Young Kim, Mahdi Saleh, Benjamin Busam, and Nassir Navab. S3m: Scalable statistical shape modeling through unsupervised correspondences. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 459–469. Springer, 2023. [3](#)
- [4] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. [3](#)
- [5] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM transactions on graphics (TOG)*, 22(3):569–577, 2003. [2](#)
- [6] Hanzhi Chen, Fabian Manhardt, Nassir Navab, and Benjamin Busam. Texpose: Neural texture learning for self-supervised 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4841–4852, 2023. [3](#)
- [7] 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/CVF International Conference on Computer Vision*, pages 11594–11604, 2021. [3](#)
- [8] Yue Chen, Xuan Wang, Xingyu Chen, Qi Zhang, Xiaoyu Li, Yu Guo, Jue Wang, and Fei Wang. Uv volumes for real-time rendering of editable free-view human performance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16621–16631, 2023. [1, 2, 3, 6, 7](#)
- [9] 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](#)
- [10] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 195–205, 2018. [3](#)
- [11] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. [2](#)
- [12] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. [2](#)
- [13] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010. [3](#)
- [14] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018. [6](#)
- [15] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020. [3](#)
- [16] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5052–5063, 2020. [1, 3](#)
- [17] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepcut: A deeper, stronger, and faster multi-person pose estimation model. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI 14*, pages 34–50. Springer, 2016. [3](#)
- [18] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. [3](#)
- [19] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *arXiv preprint arXiv:2305.06356*, 2023. [3](#)
- [20] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5605–5615, 2022. [3](#)
- [21] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *European Conference on Computer Vision*, pages 402–418. Springer, 2022. [3](#)
- [22] HyunJun Jung, Shun-Cheng Wu, Patrick Ruhkamp, Hannah Schieber, Pengyuan Wang, Giulia Rizzoli, Hongcheng Zhao, Sven Damian Meier, Daniel Roth, Nassir Navab, et al. Housecat6d—a large-scale multi-modal category level 6d object pose dataset with household objects in realistic scenarios. *arXiv preprint arXiv:2212.10428*, 2022. [3](#)
- [23] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou,

- Anil Armagan, Slobodan Ilic, et al. On the importance of accurate geometry data for dense 3d vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–791, 2023. 2
- [24] Takeo Kanade, Peter Rander, and PJ Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia*, 4(1):34–47, 1997. 3
- [25] Mert Asim Karaoglu, Hannah Schieber, Nicolas Schischka, Melih Görgülü, Florian Grötzner, Alexander Ladikos, Daniel Roth, Nassir Navab, and Benjamin Busam. Dynamon: Motion-aware fast and robust camera localization for dynamic nerf. *arXiv preprint arXiv:2309.08927*, 2023. 2
- [26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 1, 2, 3, 4, 5
- [27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012. 5
- [28] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, page 31–42, New York, NY, USA, 1996. Association for Computing Machinery. 2
- [29] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *European Conference on Computer Vision*, pages 419–436. Springer, 2022. 3
- [30] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2
- [31] Zhe Li, Zerong Zheng, Yuxiao Liu, Boyao Zhou, and Yebin Liu. Posevocab: Learning joint-structured pose embeddings for human avatar modeling. In *ACM SIGGRAPH Conference Proceedings*, 2023. 1, 3, 6, 7
- [32] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021. 2
- [33] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017. 3
- [34] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM transactions on graphics (TOG)*, 40(6):1–16, 2021. 3
- [35] 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):1–16, 2015. 1, 2, 3, 4, 5, 6, 7
- [36] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 2, 3
- [37] Fabian Manhardt, Diego Martin Arroyo, Christian Ruprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6841–6850, 2019. 2
- [38] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [39] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2
- [40] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2
- [41] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5762–5772, 2021. 3
- [42] Sergio Orts-Esculano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th annual symposium on user interface software and technology*, pages 741–754, 2016. 2
- [43] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [44] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2
- [45] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 2
- [46] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021. 1, 3
- [47] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes

- for novel view synthesis of dynamic humans. In *CVPR*, 2021. 1, 3, 6, 7, 2
- [48] Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, page 335–342, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 4
- [49] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2, 3
- [50] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11143–11152, 2022. 3
- [51] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 2
- [52] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011. 3
- [53] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. Scanimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2886–2897, 2021. 1, 3
- [54] Mahdi Saleh, Shun-Cheng Wu, Luca Cosmo, Nassir Navab, Benjamin Busam, and Federico Tombari. Bending graphs: Hierarchical shape matching using gated optimal transport. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11757–11767, 2022. 3
- [55] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 3
- [56] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4, 3
- [57] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 4, 2, 3
- [58] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35:151–173, 1999. 2
- [59] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerf-player: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 3
- [60] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE computer graphics and applications*, 27(3):21–31, 2007. 2
- [61] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Advances in Neural Information Processing Systems*, 34:12278–12291, 2021. 3
- [62] David Svitov, Dmitrii Gudkov, Renat Bashirov, and Victor Lempitsky. Dinar: Diffusion inpainting of neural textures for one-shot human avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7062–7072, 2023. 3
- [63] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 517–524. IEEE, 1998. 2
- [64] Richard Szeliski, Steven Gortler, Radek Grzeszczuk, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on computer graphics and interactive techniques (SIGGRAPH 1996)*, pages 43–54, 1996. 2
- [65] Pengyuan Wang, HyunJun Jung, Yitong Li, Siyuan Shen, Rahul Parthasarathy Srikanth, Lorenzo Garattoni, Sven Meier, Nassir Navab, and Benjamin Busam. Phocal: A multi-modal dataset for category-level object pose estimation with photometrically challenging objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21222–21231, 2022. 3
- [66] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdbs. In *European conference on computer vision*, pages 1–19. Springer, 2022. 3
- [67] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9168–9178, 2021. 3
- [68] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2
- [69] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humanerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition*, pages 16210–16220, 2022. 2, 3
- [70] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 3
- [71] Tianhan Xu, Yasuhiro Fujita, and Eiichi Matsumoto. Surface-aligned neural radiance fields for controllable 3d human synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15883–15892, 2022. 3
- [72] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian

- Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (ToG)*, 37(2):1–15, 2018. 3
- [73] Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 4k4d: Real-time 4d view synthesis at 4k resolution, 2023. 3
- [74] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 3
- [75] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Advances in Neural Information Processing Systems*, 34:23872–23884, 2021. 3
- [76] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5384–5393, 2023. 3
- [77] Zhengming Yu, Wei Cheng, Xian Liu, Wayne Wu, and Kwan-Yee Lin. Monohuman: Animatable human neural field from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16943–16953, 2023. 3
- [78] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3, 6, 7, 2
- [79] Wojciech Zienonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars, 2023. 3

Deformable 3D Gaussian Splatting for Animatable Human Avatars

Supplementary Material

7. Deformation Refinement Module (DRM)

The DRM is a fully connected (FC) layer based network that takes care of the motion dependent cloth deformation in our pipeline. The DRM is composed of 13 FC layers. The first 12 FC layers are followed by a ReLu activation. After the 5th and 9th layer there are skip connections. The last layer is a FC layer followed by a postprocessing layer to convert the output vector into rotation and translation. A detailed composition is shown in Fig. 11.

The last layer outputs a 7 channel vector that is parameterized as axis angle rotation (4 channels) and translation (3 channels). We use the first 3 channels as translation and the later channels for rotation. For the rotation, the first channel is used as the angle, and the later 3 channels are normalized to form the rotation axis. For the ZJU mocap dataset [47], we apply a sigmoid activation to limit the translation to $\pm 10\text{cm}$ and rotation to $\pm 30^\circ$. For the THUman 4.0 dataset [78], we use an unbounded output that is scaled by 0.01 for the translation ($1 \rightarrow 1\text{cm}$) and π^{-1} for the rotation ($1 \rightarrow 18.25^\circ$). We find the results for the THUman dataset [78] are better with an unbounded prediction as the human performers wears hoodies that involve more deformation than the ZJU [47] dataset.

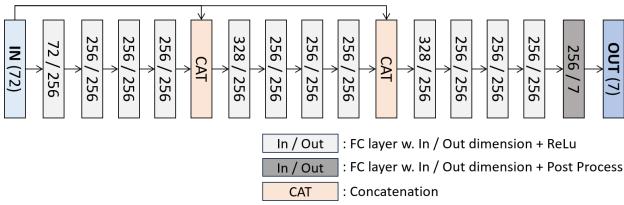


Figure 11. **DRM definition** The DRM is based on 13 FC layers. Each of the 12 layers is followed by a ReLu activation and the last layer is followed by a postprocessing that converts the 7 channel vector into translation and rotation. The 5th and 9th layer are skip connection layers that concatenate the previous layer’s output feature and input vector.

8. Camera Selection for Training and Evaluation

We use 7 camera views for training for both the ZJU mocap dataset [47] and the THUman 4.0 dataset [78]. Tab. 3 shows the exact cameras that are used for training and testing. For the ZJU mocap dataset [47], slightly different training and testing cameras are used depending on the scene due to issues with camera calibration (see Fig. 5 (a)). While for the THUman 4.0 dataset [78], we fix the cameras among all

scenes for training and testing. For both dataset, we use mixed cameras for testing. For the ZJU mocap dataset [47], 3 seen views and 4 unseen views are used for testing, while 3 seen and 3 unseen views are used for the THUman 4.0 dataset [78].

Table 3. **Cameras Used for Training and Testing.** We use different cameras in the ZJU mocap dataset [47] among the scenes as the calibration quality differs depending on the scene. In comparison, the same camera combinations are used in all scenes in THUman 4.0 dataset [78]. For testing we use a combination of seen and unseen cameras for both datasets.

Dataset	Scene	Train Camera	Test Camera
ZJU Mocap [47]	313	1,3,7,10,13,16,19	1,5,10,13,16,18,21
	315	1,3,7,10,13,16,19	1,5,10,13,16,18,21
	377	2,3,7,10,14,15,16	2,4,7,10,14,17,22
	394	2,5,7,10,15,16,22	3,4,7,10,15,16,23
THUman 4.0 [78]	00	1,5,9,13,17,21,23	3,7,11,17,21,23
	01	1,5,9,13,17,21,23	3,7,11,17,21,23
	02	1,5,9,13,17,21,23	3,7,11,17,21,23

9. Detailed Speed Analysis

ParDy-Humanfeatures 3D Gaussian Splatting [26] as a backbone, which allows faster rendering as it renders images directly with pointclouds instead of shooting rays from the camera along each pixel. However, our pipeline requires multiple steps that deform the gaussians as well as calculating per point deformation which slows down the rendering speed compared to the original 3D Gaussian Splatting [26] pipeline. Tab. 4 shows detailed speed analysis of each major step. A consumer level laptop with an i9 and a RTX3080 Max-Q is used for measuring the speed.

In general, the THUman dataset [78] requires more time than the Mocap dataset [47] for the animation and rendering. This is because the human performers from the Mocap dataset [47] in general have short sleeve and short pants that require less gaussians, while the actors in the THUman dataset [78] wear hoodies and long pants that require more gaussians to replicate more texture, shape and shading. Furthermore, we find one of the main bottle necks is the Per Vertex Deformation (calculating the SMPL based deformation per gaussian). For the Per Vertex Deformation calculation, we use a per mesh face SVD calculation between the canonical and deformed SMPL mesh instead of using the SMPL deformation field directly for compatibility reasons, for example the Mocap dataset [47] uses a non-conventional SMPL definition¹, while the THUman dataset [78]) uses the

¹https://github.com/zju3dv/EasyMocap/blob/master/doc/02_output.md

Table 4. Detailed time analysis. We break down each step of our pipeline during inference to give detailed information regarding speed and also insight for future optimization. In general, generating an human avatar from the THUman dataset [78] takes more time than from the Mocap dataset [47]. We find that, short sleeves and short pants require less gaussians than long sleeves, hoodies and long pants. For this reason, the rendering speed on scene 313-377 is slightly faster than scene 394, and the Mocap dataset [47] is faster than the THUman dataset [78] on average.

Scene	Per Vertrex Deformation Calculation	Posing (I)	Deformation Refinement Calculation	Posing (II)	Rendering	Img Save	Total Time (w. Save / w.o. Save)	FPS (w. Save / w.o. Save)
313	0.0414	0.0069	0.0479	0.0026	0.0068	0.041	0.147 / 0.106	6.80 / 9.43
315	0.0437	0.0074	0.0536	0.0028	0.0072	0.042	0.158 / 0.116	6.33 / 8.62
377	0.0434	0.0080	0.0434	0.0030	0.0074	0.043	0.148 / 0.105	6.75 / 9.53
394	0.0410	0.0093	0.0568	0.0034	0.0084	0.042	0.161 / 0.119	6.21 / 8.40
AVG	0.0424	0.0079	0.0504	0.0030	0.0075	0.042	0.153 / 0.111	6.54 / 9.01
00	0.0652	0.0041	0.0369	0.0017	0.0091	0.049	0.166 / 0.117	6.02 / 8.55
01	0.0659	0.0051	0.0525	0.0022	0.0120	0.055	0.193 / 0.138	5.18 / 7.25
02	0.0659	0.0046	0.0448	0.0019	0.0110	0.056	0.182 / 0.126	5.49 / 7.94
AVG	0.0657	0.0046	0.0447	0.0019	0.0107	0.053	0.181 / 0.128	5.52 / 7.81

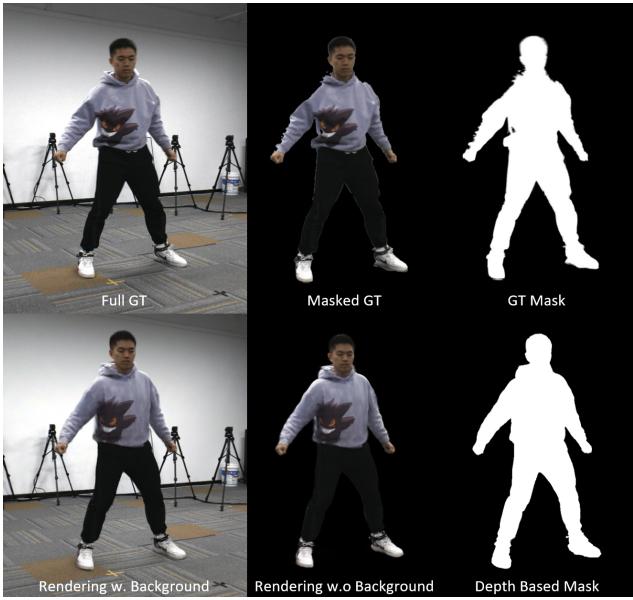


Figure 12. Mask Comparison. ParDy-Humanpipeline learns to build avatars together with the background to avoid any artifact related to mask annotations. During the inference the background can easily be filtered out so that the avatar can be rendered without any background.

original definition. We believe that optimizing our pipeline by adapting each dataset’s specific format to be compatible with the SMPL deformation field directly could speed up our pipeline up to 30% in the future.

10. Rendering Examples

In this section, we show more examples of rendered images, such as mask rendering and rendering without background in different camera poses.

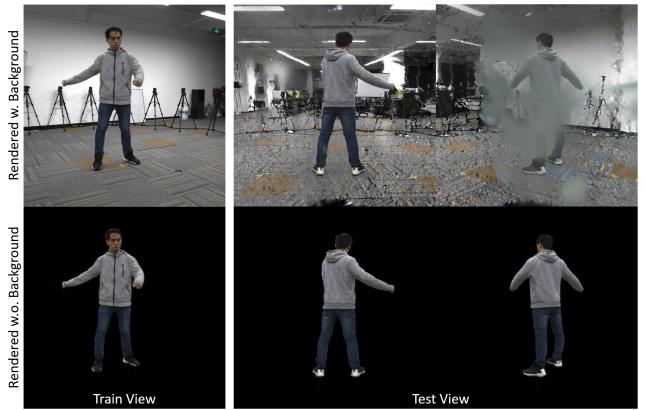


Figure 13. Rendering without background. As our pipeline trains with less camera views, we initialize the background gaussians with rather random points on a spherical surface around the world center instead of using SfM points [56, 57]. With this, the learning of the background is done by overfitting on the training views, creating artifacts on unseen camera views (first row). However, this artifacts can be easily resolved by filtering the background gaussians using their parent information (second row).

10.1. Mask Comparison

Annotating masks of a human performer in a long sequence is often challenging, especially when the background is uncontrolled. We show in Fig. 5 show that masks on THUman 4.0 [78] often over-segment and under-segment depending on the light condition in the frame and show in Fig. 8 that this issue can create strong artifacts when these masks are used during avatar generation. Here in Fig. 12, we show that ours trained without masks can generate better quality masks than what is provided as GT, thus shows a big advantage of our work that can circumvent the issue of annotating good quality masks and still learns the shape of a human in

a good quality. To generate the mask, we first remove the background gaussians, then render the depth by applying some depth rendering modification to the rendering script of Ziyi Yang² followed by masking with a distance threshold.

10.2. Rendering without Background

For training, we initialize the background gaussians as a random pointcloud along a spherical surface instead of using a SfM [56, 57] pointcloud, such that our pipeline stays simple and can run with fewer viewpoints (i.e. 7 views or 1 view). For this reason, the background overfits on the training views and produces artifacts in the rendering, such as the breaking of background geometry as well as generating floating points that block the camera. Fig. 13 (first row) shows a visualization of the aforementioned artifacts. These artifacts can be easily removed by filtering the background using the parent information of the gaussians (Fig. 13, second row). We believe that a monocular video sequence that moves around an human performer could allow a SfM [56, 57] pipeline to generate a pointcloud of the background and give better results for the background, allowing the rendering of novel views of the human avatar together with the background.

²<https://github.com/graphdeco-inria/diff-gaussian-rasterization/pull/3>