Relightable and Animatable Neural Avatars from Videos

Wenbin Lin, Chengwei Zheng, Jun-Hai Yong, Feng Xu

School of Software and BNRist, Tsinghua University
lwb20@mails.tsinghua.edu.cn, zhengcw18@gmail.com, yongjh@tsinghua.edu.cn, xufeng2003@gmail.com

Abstract

Lightweight creation of 3D digital avatars is a highly desirable but challenging task. With only sparse videos of a person under unknown illumination, we propose a method to create relightable and animatable neural avatars, which can be used to synthesize photorealistic images of humans under novel viewpoints, body poses, and lighting. The key challenge here is to disentangle the geometry, material of the clothed body, and lighting, which becomes more difficult due to the complex geometry and shadow changes caused by body motions. To solve this ill-posed problem, we propose novel techniques to better model the geometry and shadow changes. For geometry change modeling, we propose an invertible deformation field, which helps to solve the inverse skinning problem and leads to better geometry quality. To model the spatial and temporal varying shading cues, we propose a pose-aware part-wise light visibility network to estimate light occlusion. Extensive experiments on synthetic and real datasets show that our approach reconstructs high-quality geometry and generates realistic shadows under different body poses. Code and data are available at https://wenbin-lin.github.io/RelightableAvatar-page/.

1 Introduction

Human digitizing has been rapidly developed in recent years, in which the reconstruction and animation of 3D clothed human avatars have many applications in telepresence, AR/VR, and virtual try-on. One important goal here is to render the human avatar in desired lighting environment with desired poses. Therefore, the human avatars need to be both relightable and animatable and achieve photorealistic rendering quality. Usually, the generation of these high-quality human avatars relies on high-quality data like the ones recorded by Light Stages (**?**) which are complicated and expensive.

Recently, the emergence of Neural Radiance Fileds (NeRF) (**?**) opens a new window to generate animatable and relightable 3D human avatars just from the daily recorded videos. NeRF-based methods have achieved remarkable success in 3D object representation and photorealistic rendering of both static and dynamic objects including human bodies (**??????????**). Also, NeRF can be used for intrinsic decomposition to achieve impressive relighting results for static objects (**???????**). However, NeRF-based dynamic object relighting is rarely studied. One key challenge is that the dynamics cause dramatic changes in object shading, which is hard to model with the current NeRF techniques.

In this work, we propose to reconstruct both relightable and animatable 3D human avatars from sparse videos recorded under uncalibrated illuminations. To achieve this goal, we need to reconstruct the body geometry, material, and environmental light. The dynamic body geometry is modeled by a static geometry in a canonical space and the motion to deform it to the shape in the observation space of each frame. We propose an invertible neural deformation field that builds a bidirectional mapping between points of the canonical space and all observation spaces. With this bidirectional mapping, we can easily leverage the body mesh extracted in the canonical pose to better solve the inverse linear blend skinning problem, thus achieving high-quality geometry reconstruction. After the geometry reconstruction of all frames, we propose a light visibility estimation module to better model the dynamic self-occlusion effect for material and light reconstruction. We transfer the global pose-related visibility estimation task into multiple, part-wise, local ones, which dramatically simplifies the complexity of light visibility estimation. This model has good generalization capability with limited training data benefiting from the part-wise architecture, and thus successfully estimates the light visibility under various body poses and lighting conditions. Finally, we optimize the body material and lighting parameters, and then our method can render photorealistic images under any desired body pose, lighting, and viewpoint. In summary, the contributions include:

- the first method that is able to reconstruct both relightable and animatable human avatars with plausible shadow effects from sparse multiview videos,
- an invertible deformation field that better solves the inverse skinning problem, leading to accurate dense correspondence between different body poses,
- part-wise light visibility networks that better estimate pose and light-related shading cues with high generalization capability.

2 Related Work

2.1 Neural Human Avatars

In recent years, neural radiance fields (NeRF) (**?**) have shown great abilities in photorealistic rendering. And many methods have successfully combined NeRF with human parametric models for human body reconstruction (**??**) and animatable human body modeling (**????????**) with sparse videos. For dynamic body motion modeling, people usually leverage linear blend skinning (LBS) (**?**) to drive the body to different poses and use neural displacement fields to model the non-rigid deformations. Among these works, the deformation fields only model single-direction displacement, either forward deformation (canonical to observation) (**??**) or backward deformation (observation to canonical) (**???**). Different from them, our method proposes an invertible deformation field to solve the correspondence between canonical and observation space bidirectionally, which helps to better solve the inverse skinning problem, and leads to better geometry reconstruction. Recent work MonoHuman (**?**) also models bidirectional deformations, but unlike the compact single invertible network in our approach, they use two non-invertible neural networks to model the deformations separately. Additionally, these methods model body appearance using view-dependent color without decomposing it into lighting and reflectance. In contrast, our method enables relighting by reconstructing the environment lighting and the surface material.

2.2 Human Relighting

Some methods have been proposed to enable relighting of human images (**??????**). However, these image-based methods do not support changing the viewpoints and human poses. To further enable novel view relighting, 3D reconstruction techniques have been leveraged to model the human geometry (**?**). For video-based human relighting, Relighting4D (**?**) enables free-viewpoint relighting from only human videos under unknown illuminations by using a set of neural fields of normal, occlusion, diffuse, and specular maps. But it is hard to relight the human with novel poses as it involves per-frame latent features which are not generalizable for novel poses. RANA (**?**) proposed a generalizable relightable articulated neural avatars creation method based on SMPL+D (**?**) model with albedo, normal map refinement techniques. But their method did not model specular reflection and cast shadows. In this paper, we present the first method that can reconstruct relightable and animatable human avatars from videos under unknown illuminations, while providing physically correct shadows.

2.3 Invertible Neural Network

Invertible Neural Networks (INNs) (**?????**) are are capable of performing invertible transformations between the input and output space. They are widely used in generative models like Normalizing Flows (**?**) for density estimation. Moreover, the ability of INNs to maintain cycle consistency between two spaces makes them suitable for modeling the deformation field of 3D objects. As a result, INNs have been used for 3D shape completion (**????**), geometry processing (**?**), dynamic scenes reconstruction (**?**), and building animatable avatars with 3D scans (**?**). However, for video-based dynamic body deformation modeling, existing works only use non-invertible single-directional deformation. In this work, we leverage the invertibility of the INNs to model the dynamic body motions and reconstruct high-quality dynamic body geometry.

3 Method

Given multi-view videos of a user with some arbitrary motions, our goal is to reconstruct a relightable and animatable avatar of the user. The key challenge of this task is disentangling the geometry, material of the clothed body, and lighting, which is a highly ill-posed problem. To tackle this problem, we first reconstruct the body geometry from the input videos using the neural rendering techniques, where the geometry is modeled by a neural signed distance function (SDF) field and the dynamics of the human body are modeled with a rigid bone transformation of SMPL (**?**) model plusing an invertible neural deformation field (Sec.**??**, top left of Fig.**??**). Then, with the reconstructed geometry, we train a pose-aware part-wise light visibility estimation network, which is able to predict the light visibility of any query point under any light direction and body pose (Sec.**??**, bottom left of Fig.**??**). Finally, with the visibility information, we achieve the disentangling of the material of the human body and the illumination parameters (Sec.**??**, top right of Fig.**??**). Therefore, we can render a free-viewpoint video of the human with any target pose and illumination.

3.1 Geometry and Motion Reconstruction

Dynamic body deformation consists of articulated rigid motion and neural non-rigid deformation. Correspondingly, we propose a mesh-based inverse skinning method and an invertible neural deformation field to map points between the canonical and the observation space bidirectionally.

Mesh-based Inverse Skinning. The rigid motion is computed using linear blend skinning (LBS) algorithm (**?**). For point $\mathbf{x}_c$ in the canonical space, we use the bone transformation matrices $\{\mathbf{B}_b\}_{b=1}^{24}$ of the SMPL (**?**) model to transform $\mathbf{x}_c$ to $\mathbf{x}_o$ in the observation space (we omit the transformations of homogeneous coordinates for simplicity of notation):

$$\mathbf{x}_o = \sum_{b=1}^{24} \left(w_b(\mathbf{x}_c)\mathbf{B}_b\right) \mathbf{x}_c \tag{1}$$

where $w_b(\mathbf{x}_c)$ is the skinning weights of $\mathbf{x}_c$, and $\sum_{b=1}^{24} w_b(\mathbf{x}_c) = 1$. Similarly, for $\mathbf{x}_o$ in the observation space, we can transform it back to the canonical space by:

$$\mathbf{x}_c = \sum_{b=1}^{24} \left(w_b(\mathbf{x}_o)\mathbf{B}_b\right)^{-1} \mathbf{x}_o \tag{2}$$

Similarly, for a query view direction ω_o in the observation space, we can apply the same backward transformation to get the view direction ω_c in the canonical space.

In volume rendering, we need to transform sampled ray points in the observation space to the canonical space (i.e.

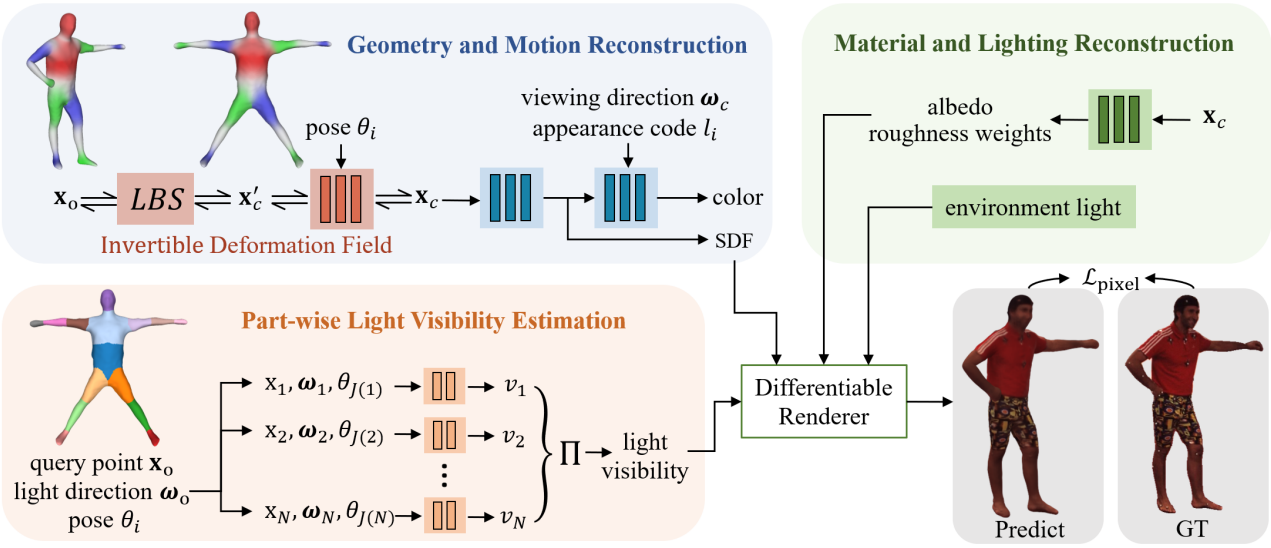

Figure 1: The pipeline of our method. The invertible deformation field in *Geometry and Motion Reconstruction* contributes to reconstruct more accurate dynamic body geometry (Sec.**??**). Then the networks in *Part-wise Light Visibility Estimation* are trained to estimate pose-aware light visibility in an effective manner (Sec.**??**). With these two parts fixed, the networks and lighting coefficients in *Material and Light Estimation* are trained and optimized by the photometric losses (Sec.**??**).

solve the inverse skinning problem) to query their SDF and color values. However, determining the skinning weights of points in the observation space is non-trivial as w_b is calculated in the canonical space rather than the observation space. Many existing works, such as (**???**), rely on the posed SMPL mesh and use the skinning weights of neighboring SMPL mesh points to compute the inverse skinning weights of the ray points. However, the naked SMPL mesh differs from the body surface, resulting in inaccurate weights. Differently, we leverage the extracted explicit body mesh to compute the inverse skinning weights. We first extract an explicit mesh of the body in the canonical space and compute the skinning weights of mesh vertices. Then, we use the LBS algorithm to deform the mesh to the observation space. For any points in the observation space, we compute their skinning weights by the skinning weights of the nearest neighbor on the deformed body mesh. As the deformed body mesh fits the actual body surface better than the naked SMPL mesh, our method does not suffer from the inaccuracies of skinning weights.

Invertible Deformation Field. Since only rigid bone transformation is not enough for modeling the body motion, we use an invertible neural displacement field to model the non-rigid motions. As shown in Fig.**??**, on the one hand, we apply non-rigid motion to the explicit mesh in canonical space. On the other hand, for sampled ray points in observation space, we need to map them back to the canonical space. Therefore, the neural displacement field should be able to transform points bidirectionally and ensure the cycle consistency of the transformation. So, we involve an invertible neural network to represent the non-rigid motion. For a point $\mathbf{x} = [u, v, w]$ in the canonical space, we use the invertible network D to apply displacement to it:

$$\mathbf{x}' = D(\mathbf{x}) = [u', v', w'] \tag{3}$$

Besides, the invertible network D can also transform $\mathbf{x}'$ back while keep the cycle consistency:

$$\mathbf{x} = D^{-1}(\mathbf{x}') = D^{-1}(D(\mathbf{x})) \tag{4}$$

To keep the cycle consistency, we design a network similar to Real-NVP (**?**). Specifically, we split the coordinates $[u, v, w]$ into two parts, for example, $[u, v]$ and $[w]$. During forward deformation, we assume the displacement of $[w]$ is decided by the value of $[u, v]$:

$$[w'] = [w] + f([u, v]) \tag{5}$$

and then the displacement of $[u, v]$ is decided by $[w']$:

$$[u', v'] = [u, v] + g([w']) \tag{6}$$

With this two-step forward deformation D, we can directly get an invertible backward deformation D^{-1} which deforms point $[u', v', w']$ to $[u, v, w]$ as follows:

$$\begin{aligned} [u, v] &= [u', v'] - g([w']) \\ [w] &= [w'] - f([u, v]) \end{aligned} \tag{7}$$

The functions $f(\cdot), g(\cdot)$ are implemented as MLPs, they form a transformation block of the invertible network D. As the aforementioned $f(\cdot)$ makes the deformation decided by $[u, v]$ only, we stack more transformation blocks and change the split of $[u, v, w]$ in these blocks. We assume that the non-rigid deformations are pose-dependent, for the ith frame, we use the body pose θ_i as the condition of the network D. Besides, we found that it is hard to learn the

deformation using only the pose and coordinates as conditions. Thus, we use the skinning weights of the query points $\mathbf{W}(\mathbf{x}) \in \mathbb{R}^{24}$ as an additional condition, which leads to better results. The displacement field D can be formulated as $\mathbf{x}' = D(\mathbf{x}, \theta_i, \mathbf{W}(\mathbf{x}))$. The use of skinning weights will slightly affect the cycle consistency of the deformation network, as $\mathbf{W}(\mathbf{x})$ is not strictly equal to $\mathbf{W}(\mathbf{x}')$. But we found the skinning weights field in the canonical pose is smooth, and the deformations are relatively small, so they are almost the same, the sacrifice on cycle consistency is negligible.

Network Training. To supervise these neural fields with videos, we use the technique proposed in VolSDF (**?**) to convert the SDF values to density and conduct volume rendering. For the color field, we introduce learnable per-frame appearance latent codes $\{l_i\}_{i=1}^N$ to model the dynamic appearance, where N is the number of frames. Besides, we optimize the pose vectors $\{\theta_i\}_{i=1}^N$, as the initial poses may not be accurate. In sum, the training parameters contain the SDF network, the color network, the deformation network D, the appearance latent codes $\{l_i\}_{i=1}^N$ and the pose vectors $\{\theta_i\}_{i=1}^N$. The training loss consists of rendering photometric loss and multiple regularizers:

$$\mathcal{L} = \lambda_{\text{pixel}}\mathcal{L}_{\text{pixel}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{eik}}\mathcal{L}_{\text{eik}} + \lambda_{\text{disp}}\mathcal{L}_{\text{disp}} \quad (8)$$

where $\mathcal{L}_{\text{pixel}}$ is an L2 pixel loss for predicted color, $\mathcal{L}_{\text{mask}}$ is a binary cross-entropy loss for the rendering object mask and input mask, $\mathcal{L}_{\text{eik}}$ is the Eikonal regularization term (**?**), $\mathcal{L}_{\text{disp}}$ is an L2 regularizer for the output displacements. For more details about network architecture and training, please refer to the supplemental document.

3.2 Part-wise Light Visibility Estimation

With the reconstructed geometry, we then conduct pose-aware light visibility estimation. Modeling the visibility allows for the extraction or generation of shadows on images, which helps to better disentangle material and lighting from input images as well as produce physically plausible shadow effects in rendered images. Given a query point $\mathbf{x}$ and a query light direction ω, our goal is to train a network to predict whether the query point will be lighted or occluded by the body in a certain a pose and light direction.

Traditionally, estimating light visibility is solved by performing ray tracing. However, for implicit neural network-based methods, tracing a path of light requires numerous queries, as we need to trace all possible lighting directions for one 3D point, which is very time-consuming. Thus, existing methods (**???**) use MLPs to re-parameterize and speed up this process as $V(\mathbf{x}, \omega) \mapsto v$, where $v = 1$ indicates the point is visible to the light from ω direction. However, with the motion of the human body, light visibility changes dramatically. Relighting4D (**?**) leverages temporally-varying latent codes to model these changes, but it is limited to seen poses as there is no latent code for unseen motions. To solve this problem, we need to make it pose-aware for light visibility estimation. A naive approach is to use the pose vectors as the condition of the visibility network, but we found this approach does not work well as the relationship among pose, lighting, and shadow is too complex to be modeled.

Our observation is that how light rays are blocked is determined by the object geometry, even though the human body as a whole can be in different complex shapes caused by pose changes, for a single body part, its geometry changes are relatively small among different poses. So, we divide the human body into $N(= 15)$ parts as shown in the orange rectangle in Fig.**??**, where different colors denote different body parts. Then for each body part, we train a neural network respectively to predict how the body part blocks the lights. Finally, we combine the light visibility of all body parts by multiplying all the predicted visibility. Thus, our method achieves light visibility prediction of any query points, light directions, and body poses.

To be specific, given the query point $\mathbf{x}_o$ and light direction ω_o in observation space, we first transform them to the local coordinate of each body part:

$$\mathbf{x}_i = \mathbf{B}_i^{-1}\mathbf{x}_o, \quad \omega_i = \mathbf{B}_i^{-1}\omega_o \quad (9)$$

where $\mathbf{B}_i$ is the bone transformation of the ith body part. Besides, although the geometry changes of body parts are relatively small, there are still some pose-dependent deformations. And the geometry of a body part is majorly affected by the poses of its neighboring joints, so we use them as the networks' condition. We denote the neighboring joints of body part i as $J(i)$. So, the visibility network of a body part i can be formulated as:

$$V_i\left(\mathbf{x}_i, \omega_i, \theta_{J(i)}\right) \mapsto v_i \quad (10)$$

For network training, we sample different query points, light directions, and body poses, then we perform ray tracing to compute the ground truth light visibility of each body part. We impose binary cross-entropy loss to train the networks for visibility estimation.

3.3 Material and Light Estimation

At this stage, we fix the geometry and light visibility estimation modules and optimize the material network and light parameters as shown in the green rectangle in Fig.**??**. Here, we parameterize the material using the Disney BRDF (**?**) model and use albedo and roughness to represent the material. However, we found that directly optimizing the roughness is difficult. Similar to (**???**), we use a weighted combination of specular bases. Each basis is defined by a different roughness value. For a query point in the canonical space, we use an implicit neural network M to predict its albedo value and roughness weights. For environment light, we parameterize it using $L = 128$ spherical Gaussians (SGs) (**?**):

$$E(\omega_i) = \sum_{j=1}^{L} G(\omega_i; \xi_j, \lambda_j, \mu_j) \quad (11)$$

where $\omega_i \in \mathbb{S}^2$ is the query lighting direction, $\xi_j \in \mathbb{S}^2$ is the lobe axis, $\lambda_j \in \mathbb{R}_+$ is the lobe sharpness, $\mu_j \in \mathbb{R}^3$ is the lobe amplitude. To compute the visibility of an SG light, we sample 4 directions around the lobe axis based on the distribution defined by λ_j. Then we predict their visibilities by the trained light visibility estimation network and use the weighted sum of these samples as the visibility of the SG light.

With geometry, material, environment light, and light visibility, we can render images of the human body using a differentiable renderer. The rendering equation computes the outgoing radiance L_o at point $\mathbf{x}$ viewed from ω_o:

$$L_o\left(\mathbf{x}, \omega_o\right)=\int_{\Omega} L_i\left(\mathbf{x}, \omega_i\right) R\left(\mathbf{x}, \omega_i, \omega_o, \mathbf{n}\right)\left(\omega_i \cdot \mathbf{n}\right) \mathrm{d} \omega_i \tag{12}$$

where $L_i\left(\mathbf{x}, \omega_i\right)$ is the incident radiance of point $\mathbf{x}$ from direction ω_i, which is determined by the environment light E and masked by the light visibility. $R\left(\mathbf{x}, \omega_i, \omega_o, \mathbf{n}\right)$ is the Bidirectional Reflectance Distribution Function (BRDF) which is determined by the albedo values and roughness weights predicted by M.

To train the material network M and the light parameters of E, we use L1 pixel loss between the rendered images and the recorded images. However, there are strong ambiguities in solving material and lighting, so we apply some regularization strategies. First, the material network is designed as an encoder-decoder architecture following (**?**), so that we can impose constraints on the latent space to ensure the sparsity of albedo and roughness weights. We denote the encoder and decoder of M as M_E and M_D, for a query point $\mathbf{x}$ in the canonical space, its latent vector is $\mathbf{z}=M_E(\mathbf{x}) \in \mathbb{R}^N$. For K latent codes in a batch $\{\mathbf{z}_i\}_{i=0}^K$, we impose Kullback-Leibler divergence loss to encourage the sparsity of the latent space:

$$\mathcal{L}_{\mathrm{kl}}=\sum_{j=1}^{N} \mathrm{KL}\left(\rho \| \hat{\rho}_j\right) \tag{13}$$

where $\hat{\rho}_j$ is the average of the jth channel of $\{\mathbf{z}_i\}_{i=0}^K$, ρ is set to 0.05. Moreover, we apply smooth loss to both the latent vectors and the output albedo and roughness weights by adding perturbations:

$$\begin{aligned}\mathcal{L}_{\mathrm{smooth}} =& \lambda_{\mathbf{z}}\left\|M_D(\mathbf{z})-M_D\left(\mathbf{z}+\xi_{\mathbf{z}}\right)\right\|_1+ \\ & \lambda_{\mathbf{x}}\left\|M(\mathbf{x})-M\left(\mathbf{x}+\xi_{\mathbf{x}}\right)\right\|_1\end{aligned} \tag{14}$$

where $\xi_{\mathbf{z}}$ and $\xi_{\mathbf{x}}$ are the perturbations of the latent code $\mathbf{z}$ and the query point $\mathbf{x}$, which is sampled from a Gaussian distribution with zero mean and 0.01 variance.

In sum, the full training loss for this stage is:

$$\mathcal{L}=\lambda_{\mathrm{pixel}} \mathcal{L}_{\mathrm{pixel}}+\lambda_{\mathrm{kl}} \mathcal{L}_{\mathrm{kl}}+\lambda_{\mathrm{smooth}} \mathcal{L}_{\mathrm{smooth}} \tag{15}$$

With the trained geometry field, the deformation field, the light visibility estimation networks V, and the material network M, we can render the avatar in novel poses, lightings, and viewpoints. Thus we achieve a relightable and animatable neural avatar.

4 Experiments

In this section, we evaluate the performance of our method qualitatively and quantitatively. First, we introduce the used datasets. Then we compare our method with the state-of-the-art human relighting method Relighting4D (**?**). Since our geometry reconstruction is improved by the proposed invertible deformation field, we also compare our method with the state-of-the-art video-based human geometry reconstruction methods ARAH (**?**) and **?**. Next, we perform ablation studies to validate our key design choices. Finally, we show the synthesized results on various characters with various body motions under various lightings, and video results can be seen in the supplemental video.

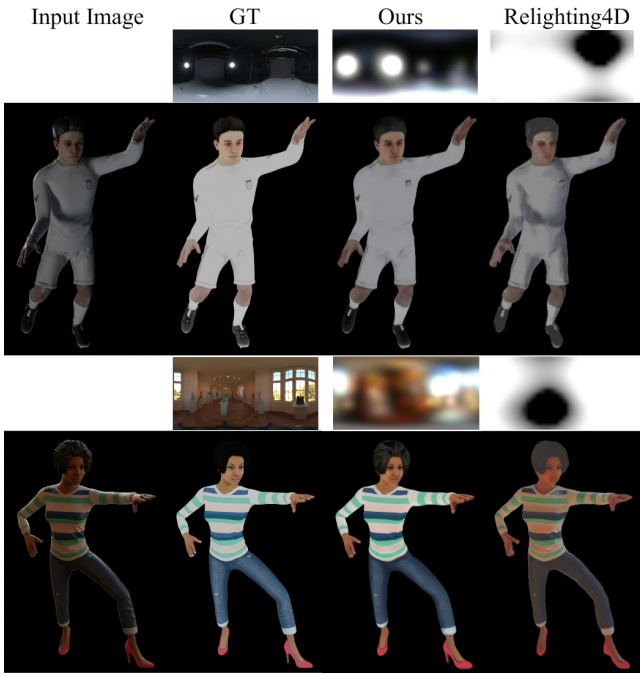

Figure 2: Qualitative comparison of the reconstructed albedo and lighting on synthetic data. Environment lighting is shown on top of the albedo in each result.

4.1 Datasets

We use both real and synthetic datasets for comparisons and evaluations. For the real dataset, we use multi-view dynamic human datasets including the ZJU-MoCap (**?**), Human3.6M (**?**), DeepCap (**?**) and PeopleSnapshot (**?**) dataset. To perform a quantitative evaluation, we create a new synthetic dataset. We leverage 4 rigged characters from Mixamo[1] and transfer the body motion from the ZJU-MoCap dataset to generate motion sequences. Each sequence contains 100 frames. Then, we use Blender[2] to render multi-view videos under different illuminations with HDRI environment maps from PolyHaven[3]. Besides, we use 4 OLAT light sources for relighting evaluations.

4.2 Comparisons

Since Relighting4D (**?**) is the state-of-the-art for video-based human motion relighting, we compare our full method with it on albedo estimation, lighting reconstruction, and relighting under training/novel poses. Body geometry is an intermediate result of our method, we also compare it with the

[1] https://www.mixamo.com/
[2] https://www.blender.org/
[3] https://polyhaven.com/

Method	Albedo Map			Relighting (Training poses)			Relighting (Novel poses)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Relighting4D (**?**)	21.5103	0.8320	0.2299	19.7323	0.7568	0.2721	16.7475	0.6729	0.3330
Ours w/o visibility	24.7611	0.8918	0.1655	23.7758	0.8376	0.2223	18.9768	0.7333	0.2638
Ours w/o part-wise visibility	**25.2150**	**0.8921**	0.1652	24.7064	0.8462	0.2173	19.7119	0.7452	0.2580
Ours	25.1666	0.8919	**0.1645**	**25.3477**	**0.8546**	**0.2124**	**19.8622**	**0.7518**	**0.2533**

Table 1: Quantitative comparison of the reconstructed albedo and the relighting results on synthetic data.

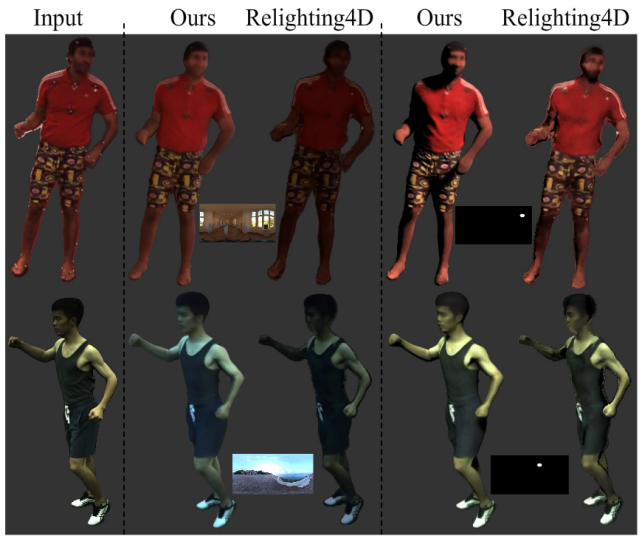

Figure 3: Qualitative comparison of relighting results on real data. The environment lighting of the rendered results is shown at the bottom.

state-of-the-art video-based human geometry reconstruction methods **?** and ARAH (**?**).

Material Estimation and Relighting. The comparison results with Relighting4D (**?**) are shown in Fig.**??**. Relighting4D cannot disentangle the lighting and appearance very well, as we can see noticeable errors on both the estimated environment map and the reconstructed albedo. For example, there are shadows wrongly baked into albedo in the result on the top right side. Besides, some lighting information is backed into albedo in the result on the bottom right side. For numerical comparisons, we use Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) (**?**), and Learned Perceptual Image Patch Similarity (LPIPS) (**?**) as metrics. In Tab.**??**, we show the numerical albedo estimation result on synthetic data, which also indicates our improvement in albedo estimation. Tab.**??** also shows the final relighting results for both training poses and novel poses, both show that we achieve noticeably better results than Relighitng4D. Note that for novel poses, there are misalignments in the geometry between the animated geometry and the ground truth geometry, which leads to noticeable performance drops for novel poses. Besides, qualitative results for relighting on real datasets are shown in Fig.**??** (please zoom in for better comparison). The overall lighting effect is better rendered by our method as shown on the left, and the spatial

	Method	S1	S2	S3	S4	Avg
P2S↓	Peng et al.	0.387	0.359	0.339	0.339	0.356
	ARAH	0.317	0.340	0.325	0.280	0.316
	Ours w/o MIS	0.241	0.230	0.234	0.241	0.237
	Ours w/o $\mathbf{W}$	0.246	0.247	0.243	0.256	0.248
	Ours	**0.185**	**0.179**	**0.182**	**0.184**	**0.182**
CD↓	Peng et al.	0.656	0.864	0.521	0.528	0.642
	ARAH	0.531	0.714	0.477	0.441	0.541
	Ours w/o MIS	0.462	0.666	0.423	0.391	0.485
	Ours w/o $\mathbf{W}$	0.479	0.700	0.427	0.424	0.507
	Ours	**0.395**	**0.609**	**0.358**	**0.363**	**0.431**

Table 2: Quantitative comparison of the reconstructed geometry on synthetic data.

variant effect caused by point light is also correctly generated by our method as shown on the right due to the success of visibility modeling. More video results can be seen in our supplementary video. Notice that as Relighting4D relies on per-frame latent codes to model the dynamics, it does not support novel poses synthesis by design. So, when performing relighting for a novel pose, we find the closest pose in its training poses and use its latent code for inference.

Geometry Reconstruction. We evaluate different methods on our synthetic dataset with ground truth geometry and use point-to-surface distance (P2S) and Chamfer Distance (CD) as metrics. The results as shown in Tab.**??**, our method outperforms the compared two methods on all test sequences. We also show qualitative comparisons of rendering images in the real dataset in Fig.**??**. We can find that there are obvious artifacts in the results of **?** and ARAH in the elbow and hand regions. While the result of our method does not suffer from the artifacts, as our mesh-based inverse skinning helps to find accurate correspondences between the observation space and the canonical space. In contrast, **?** use posed SMPL models to compute the backward skinning weights which leads to worse correspondences, especially for regions with body contacts. ARAH involves iterative root-finding to compute the correspondences, but the optimization sometimes fails to converge, thus also leading to artifacts.

4.3 Ablation Study

Here, we evaluate our two key components: mesh-based inverse skinning (MIS) and part-wise visibility estimation. The MIS based on the invertible deformation makes it possible to deform the more accurate mesh in the canonical space to the observation space to calculate skinning weights. Otherwise, the naked SMPL mesh with large geometry er-

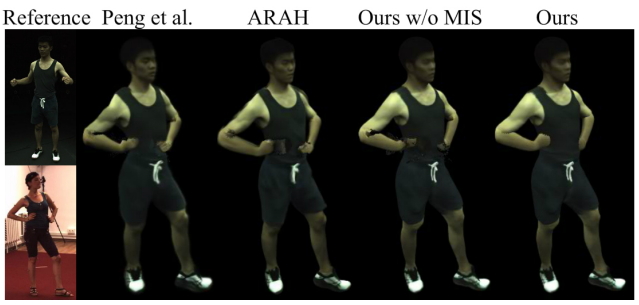

Figure 4: Qualitative results of novel poses synthesis on real data. This novel pose results reflect the accuracy of the reconstructed geometry to a certain extent.

rors has to be used. So, we compare our method with using the SMPL mesh in weights calculation. Besides, using the part-wise design achieves accurate light visibility estimation, which is crucial to generate self-occlusion effects on bodies. To evaluate it, we compare it with two alternatives, removing the light visibility module and using only one neural network to predict the light visibility.

Mesh-based Inverse Skinning. As shown in Tab.**??**, the reconstruction errors without the mesh-based inverse skinning are consistently larger. We also show the qualitative result in Fig.**??**, using SMPL mesh to compute the skinning weights leads to artifacts in the contact regions.

Besides, we evaluate the effect of the condition of skinning weights $\mathbf{W}$ (in Sec. **??**) in the invertible deformation network. As shown in Tab.**??**, we can also find that removing the condition of $\mathbf{W}$ also leads to worse results.

Part-wise Visibility Estimation. We show quantitative comparisons in Tab.**??**, the results show that although the albedo map reconstruction qualities are similar, our method with part-wise visibility estimation achieves the best results on relighting. Furthermore, We show qualitative results in Fig.**??**. We can see that without light visibility modeling (results in the fourth column), the self-occlusion effect cannot be generated at all. With the baseline light visibility modeling, self-occlusion can be partly generated for some poses (results of the third column). For our final solution, the received lighting for different body regions on the novel poses are well modeled and thus the relighting results are consistent with the ground truth rendering.

5 Limitations

The network training takes about 2.5 days in total on a single RTX 3090 GPU, and it takes about 40 seconds to render an image with a resolution of 512×512 during inference (more details in the supplemental document). Integrating instant training techniques like Instant-NGP (**?**) may improve the efficiency of our technique. It is still hard for our method to animate pose-dependent wrinkle deformations (especially for loose clothing) or generate global illumination effects, which are also open problems in this topic. Our method only considers the body motion rather than the face and hands, while recent works (**??**) provide possibilities to handle them.

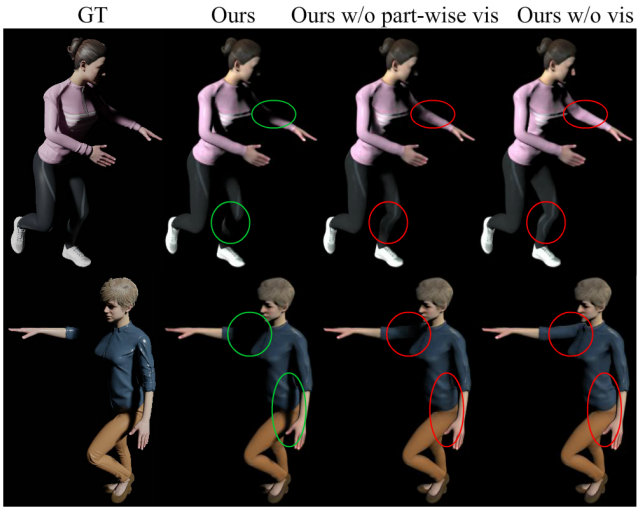

Figure 5: Ablation study on part-wise light visibility. See our method synthesizing plausible self-occlusions.

6 Conclusion

This is the first work that reconstructs relightable and animatable neural avatars with plausible shadow effects from sparse human videos. For dynamic body geometry modeling, the proposed invertible deformation field provides a novel and effective way to solve the inverse skinning problem. Besides, the part-wise light visibility modeling solves the problem of dynamic object relighting based on neural fields. Benefiting from the two techniques, our method succeeds in disentangling the geometry, material of the clothed body, and lighting, thus building a relightable and animatable neural avatar in a lightweight setting.

7 Acknowledgments

This work was supported by the National Key R&D Program of China (2018YFA0704000), the NSFC (No.62021002), and the Key Research and Development Project of Tibet Autonomous Region (XZ202101ZY0019G). This work was also supported by THUIBCS, Tsinghua University, and BLBCI, Beijing Municipal Education Commission. Jun-Hai Yong and Feng Xu are the corresponding authors.

A Overview

In the appendix, we first present the implementation details of our method, including the network architectures, the training process, and the used datasets. Second, we evaluate of the cycle consistency of the deformation field. Then, we show the results of the visualization of the correspondences. Next, we compare our method with state-of-the-art non-relightable neural avatar methods. Finally, we show additional results of our method in different datasets to demonstrate its effectiveness.

B Implementation Details

In this section, we provide more details about the network architectures, the training process and the used datasets.

B.1 Network Architectures

First, we show the structure of the geometry and color network used in the geometry and motion reconstruction stage, illustrated in Fig. **??**. The orange part is the geometry network, the green part is the color network. The $\gamma_{\mathbf{x}}(\cdot), \gamma_{\omega}(\cdot)$ are the positional encoding functions for query points and view directions. The frequencies are 10 for positions, and 4 for directions, which is the same for all the neural networks in our method. In Fig. **??**, $\mathbf{z}(\mathbf{x}_c)$ is the feature vector with a size of 256, $\mathbf{n}(\mathbf{x}_c)$ is the normal of $\mathbf{x}_c$, calculated through the gradient of the SDF field. Additionally, l_i is the appearance latent code of the ith frame with a size of 128.

Next, we show the basic transformation block of the invertible displacement network in Fig. **??**, with the procedures for both forward and backward deformation. $\mathbf{W}(\mathbf{x}_c) \in \mathbb{R}^{24}$ is the skinning weights vector, $\gamma(\cdot)$ is the positional encoding function. The full displacement network consists of 3 blocks, and (u, v, w) are split in 3 different orders.

In the light visibility estimation module, each subnetwork is an MLP with 3 hidden layers and a width of 128. In the ablation study, the light visibility estimation network without part-wise design is an MLP with 4 hidden layers and 256 in width. It should be noted that the size of this network is the same as the light visibility network of Relighting4D (**?**).

In the material network M, the encoder is an MLP with 4 hidden layers and a width of 512, the dimension of the latent space is 32, and the decoder is an MLP with 2 hidden layers and a width of 128. The roughness weights in M contain 9 different specular bases defined by different roughness values.

B.2 Training Details

For geometry and motion reconstruction, the loss weights are $\lambda_{\text{pixel}} = 1, \lambda_{\text{mask}} = 1, \lambda_{\text{eik}} = 0.01, \lambda_{\text{disp}} = 0.02$. The $\mathcal{L}_{\text{mask}}$ term is implemented in a coarse to fine manner as in IDR (**?**). For a sampled ray r, we find the minimum SDF value s_r along the ray and apply binary-cross-entropy loss as follow:

$$\mathcal{L}_{\text{mask}} = \sum_{r \in \mathcal{R}} \text{BCE}\left(\text{sigmoid}(-\alpha s_r), M(r)\right) \tag{16}$$

where $M(r) \in \{0, 1\}$ is the ground truth mask for the ray r, α is initially set to 50 and multiplied by a factor of 2 every 20 epochs. The number of multiplications of α is up to 5.

At the geometry and motion reconstruction stage, we train the network for 400 epochs. For the first 40 epochs, we use the posed SMPL (**?**) mesh to compute the skinning weights in the observation space, and the output displacements are set to zero. After this warm-up process, we replace the SMPL mesh with the extract body mesh, and use the displacement network to apply bidirectional deformations. The body mesh is extracted from the canonical SDF field using marching cubes algorithm (**?**).

For the training of the light visibility network, we sample 2000 poses from the AIST++ (**?**) dataset. For each pose, we randomly sample 4 light directions and 16,384 3D points and compute their light visibility. The networks are trained for 32 epochs. Please note that the training and novel poses in Tab. 1 of the main paper are both unseen for the light visibility network. The training poses in the table are the poses seen in the input videos, and the novel poses are just used to evaluate the animatable capability.

At the material and lighting optimization stage, the loss weights are $\lambda_{\text{pixel}} = 1, \lambda_{\text{kl}} = 0.001, \lambda_{\text{smooth}} = 0.01, \lambda_{\mathbf{z}} = 1.0, \lambda_{\mathbf{x}} = 0.05$. The models are trained for 200 epochs.

During volume rendering, we first uniformly sample 256 points per ray. For the geometry and motion reconstruction stage, we only keep the ray points close to the body mesh for color integration. For the material and light estimation stage, as the trained geometry is fixed, the material network only needs to query the sampled points with non-zero weights for volumetric rendering.

We implement our model using PyTorch and use the Adam (**?**) optimizer for training. The learning rate is $5e-4$ for the first and third stages and $1e-3$ for the second stage. All the models are trained on a single NVIDIA RTX 3090 GPU. It takes about a day each to train the first and third stages, and the light visibility estimation network takes approximately 12 hours to train.

During inference, our method with the part-wise visibility network takes about 40s to render an image of 512×512 resolution. Besides, it takes about 18s and 22s for our method without the visibility network and with a single visibility network respectively. The additional time for querying 15 body parts for light visibility is acceptable and the part-wise light visibility module achieves significantly better results than the single network. In contrast, computing light visibility through ray tracing takes about 430s per frame, causing the training of the network to be more than 10 days.

B.3 Dataset Details

Here we provide more details about the used datasets. For the synthetic dataset, we synthesize 4 body motion sequences as the training set, each containing 100 frames. We sample 10 frames evenly from each sequence for evaluation. For the novel pose evaluation, we sample 10 out-of-distribution poses from the AIST++ (**?**) dataset. The videos in the dataset are generated under 8 different views, with 4 views used for training and the other 4 views used for evaluation.

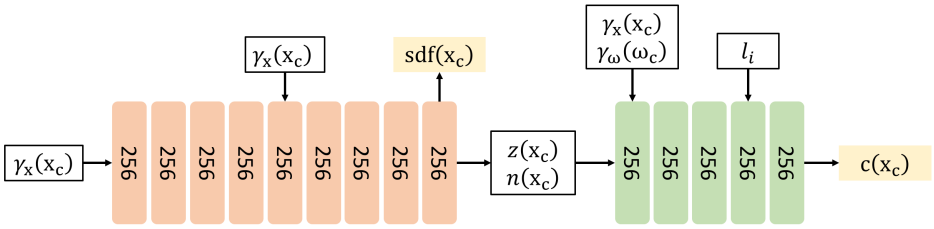

Figure 6: Architecture of the geometry and color network.

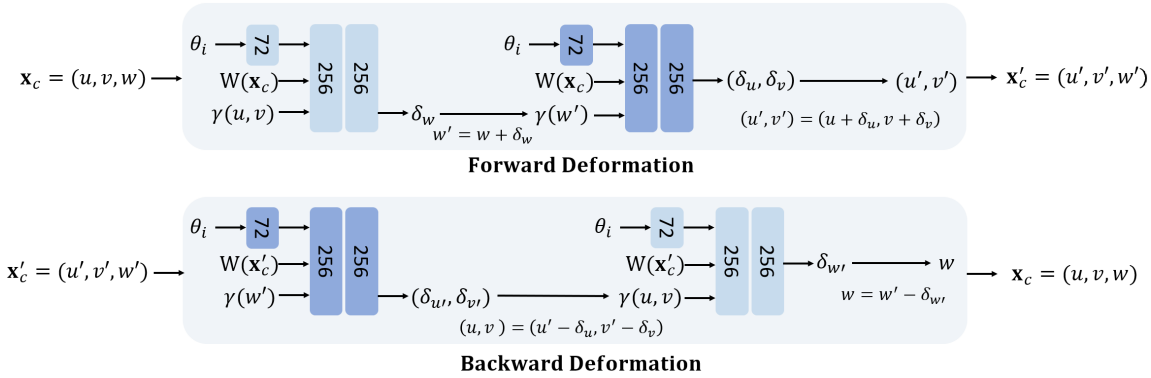

Figure 7: Architecture of the basic block in the invertible deformation network.

For evaluation on real datasets, we follow the data preprocessing method and the training-test split of **?** for the ZJU-MoCap (**?**), Human3.6M (**?**), DeepCap (**?**) and PeopleSnapshot (**?**) datasets. In our experiments, there are 4 input views used in the ZJU-MoCap and DeepCap datasets, 3 input views used in the Human3.6M dataset, and a single input view used in the PeopleSnapshot dataset. In the ZJU-MoCap, DeepCap and PeopleSnapshot datasets, the number of frames in the input videos is 300. In the Human3.6M dataset, the number of training frames ranged from 150 to 260.

C Evaluation of the Deformation Cycle Consistency

We propose an invertible deformation field that allows for bidirectional deformation while maintaining cycle consistency. For ordinary invertible neural networks, the cycle consistency is strictly satisfied. However, we involve the skinning weights $\mathbf{W}(\mathbf{x})$ as an additional condition of the deformation network to improve the geometry reconstruction results, which slightly sacrifices the cycle consistency of the deformation field. In this section, we evaluate the cycle consistency of the proposed invertible deformation network.

To evaluate the cycle consistency, we first sample some points on the body mesh. Then for a sampled point $\mathbf{x}$, we apply the forward displacement $f(\mathbf{x})$ to obtain the deformed point $\mathbf{x}' = \mathbf{x} + f(\mathbf{x})$. Then, we apply the backward displacement $f^{-1}(\mathbf{x}')$ to $\mathbf{x}'$ to obtain the final transformed point $\mathbf{x}'' = \mathbf{x}' + f^{-1}(\mathbf{x}')$. The goal is for $\mathbf{x}''$ to be close to $\mathbf{x}$, indicating that the forward and backward deformations are consistent. We use the relative deformation error as the metric:

$$\mathcal{L} = \frac{2\|\mathbf{x}'' - \mathbf{x}\|_2}{\|\mathbf{x}' - \mathbf{x}\|_2 + \|\mathbf{x}'' - \mathbf{x}'\|_2} \tag{17}$$

We compare our invertible network with the single directional deformation network of **?**, which is a 9 hidden layer, 256 width MLP. For the single directional MLP, we assume that the backward deformation is simply the negative of the forward deformation: $f^{-1}(\mathbf{x}) = -f(\mathbf{x})$. We find that the average deformation error of the single directional MLP is 9.45%, while the deformation error of our invertible network is only 1.66%. This indicates that our network is significantly better at maintaining cycle consistency.

D Correspondence Visualization

We show the estimated correspondences in Fig.**??**, the results correspond to Fig.**??**. The left part shows the overlay between the mesh and the image and the right part shows the correspondences between the canonical space and the observation space, where color encodes the correspondences. Since the explicit mesh fits the body surface well, the computed inverse skinning weights find better correspondences between the canonical and observation spaces, and using only SMPL will lead to artifacts due to the miss matches as shown in the results of "Ours w/o MIS". Besides, the correspondences of other methods also mix the arms with the body.

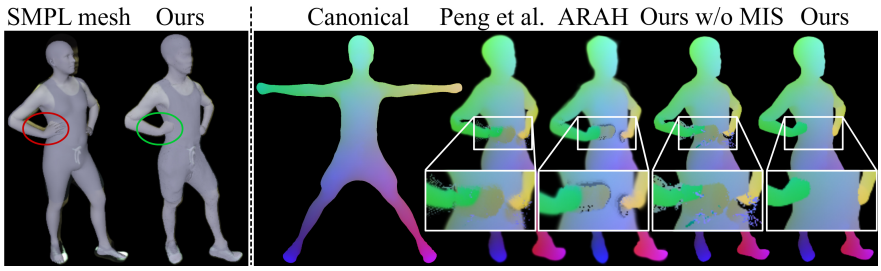

Figure 8: Visualization of estimated correspondences.

Figure 9: Qualitative comparisons of novel pose synthesis on the Human3.6M dataset.

E Comparisons with Non-relightable Methods

In this section, we compare our method with state-of-the-art non-relightable methods on the Human3.6M (**?**) dataset for novel view and novel pose synthesis. We follow the test split of Anim-NeRF (**?**) and show quantitative results of novel view and novel pose synthesis in Tab. **??** and Tab. **??** respectively. The numerical results of these methods come from the recent work NPC (**?**). We can see that our method achieves comparable results with state-of-the-art non-relightable methods and our results are better than NeuralBody (**?**), Anim-NeRF (**?**) and A-NeRF (**?**). Note that these non-relightable methods directly predict view-dependent color, while the results of our method are rendered with the reconstructed material and lighting. So the results of these methods cannot be relight under novel scenes like ours. We further show qualitative comparisons of novel pose synthesis with NeuralBody (**?**) and Anim-NeRF (**?**) in Fig. **??**. Our method achieves better visual quality than the other two methods.

F More Results

In this section, we present more results of our method. First, we show results on the DeepCap (**?**) dataset in Fig. **??**. Note that our method also reconstructs geometry details like cloth wrinkles as shown in the normal maps. Besides, our method also works well with only single-view videos as input, we show the results of our method on the PeopleSnapshot (**?**) dataset in Fig. **??**. Moreover, our method also produces plausible results on the ZJU-MoCap (**?**) dataset with single input view, as shown in Fig. **??**. The input monocular videos from the ZJU-MoCap dataset contain 500 frames. Note that these subjects from different datasets are captured under different lighting conditions, indicating that our method is robust to different shooting environments. For video results, please refer to the supplemental video.

Method	S1			S5			S9		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeuralBody (?)	22.88	0.897	0.139	24.61	0.917	0.128	24.29	0.911	0.122
Anim-NeRF (?)	22.74	0.896	0.151	23.40	0.895	0.159	24.86	0.911	0.145
A-NeRF (?)	23.93	0.912	0.118	24.67	0.919	0.114	25.58	0.916	0.126
ARAH (?)	24.53	0.921	0.103	24.67	0.921	0.115	25.43	0.924	0.112
DANBO (?)	23.95	0.916	0.108	24.86	0.924	0.108	26.15	0.925	0.108
TAVA (?)	**25.28**	**0.928**	0.108	24.00	0.916	0.122	26.20	0.923	0.119
NPC (?)	24.81	0.922	**0.097**	**24.92**	**0.926**	**0.100**	**26.39**	**0.930**	**0.095**
Ours	24.45	0.910	0.115	24.59	0.911	0.119	26.24	0.920	0.111

Table 3: Novel-view synthesis comparisons on the Human3.6M dataset.

Method	S1			S5			S9		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeuralBody (?)	22.10	0.878	0.143	23.52	0.897	0.144	23.05	0.885	0.150
Anim-NeRF (?)	21.37	0.868	0.167	22.29	0.875	0.171	23.73	0.886	0.157
A-NeRF (?)	22.67	0.883	0.159	22.96	0.888	0.155	24.16	0.889	0.164
ARAH (?)	23.18	0.903	0.116	22.91	0.894	0.133	24.15	0.896	0.135
DANBO (?)	23.03	0.895	0.121	**23.66**	0.903	0.124	24.79	0.904	0.130
TAVA (?)	**23.83**	**0.908**	0.120	22.89	0.898	0.135	24.80	0.901	0.138
NPC (?)	23.39	0.901	**0.109**	23.63	**0.906**	**0.113**	24.86	**0.907**	**0.115**
Ours	23.25	0.890	0.127	23.39	0.894	0.132	**25.05**	0.900	0.131

Table 4: Novel-pose synthesis comparisons on the Human3.6M dataset.

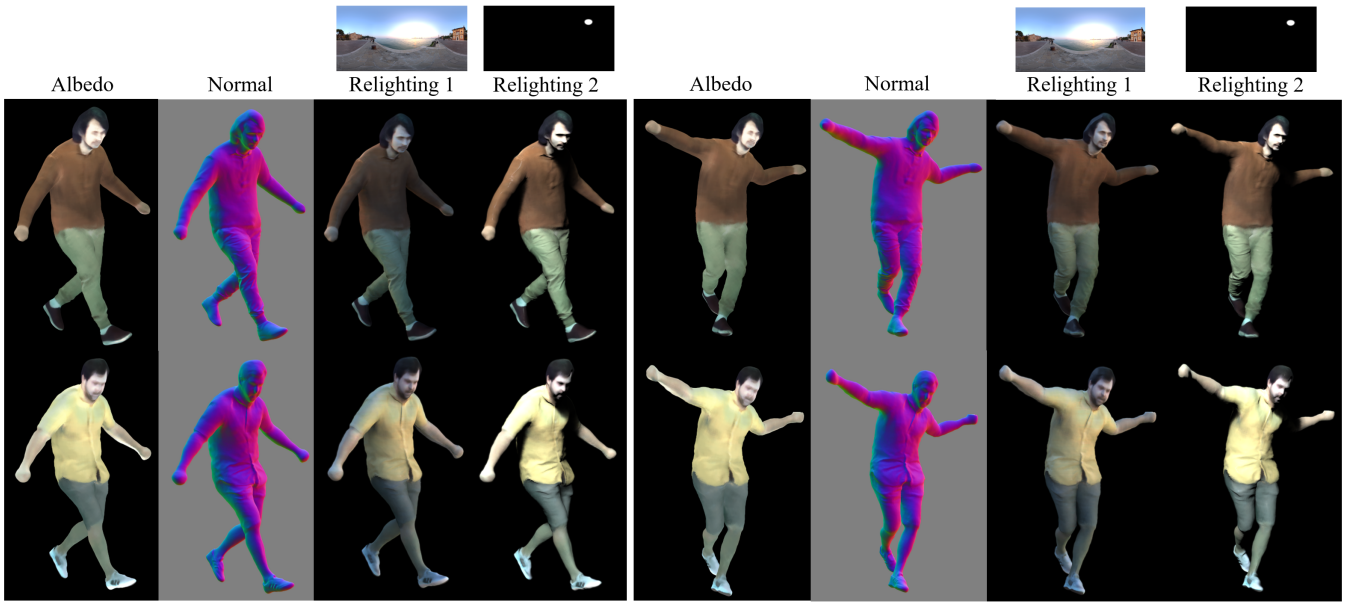

Figure 10: Results of our technique on the DeepCap dataset. From left to right of each result: the albedo of an animated pose, the corresponding normal in this pose, and two relighting results.

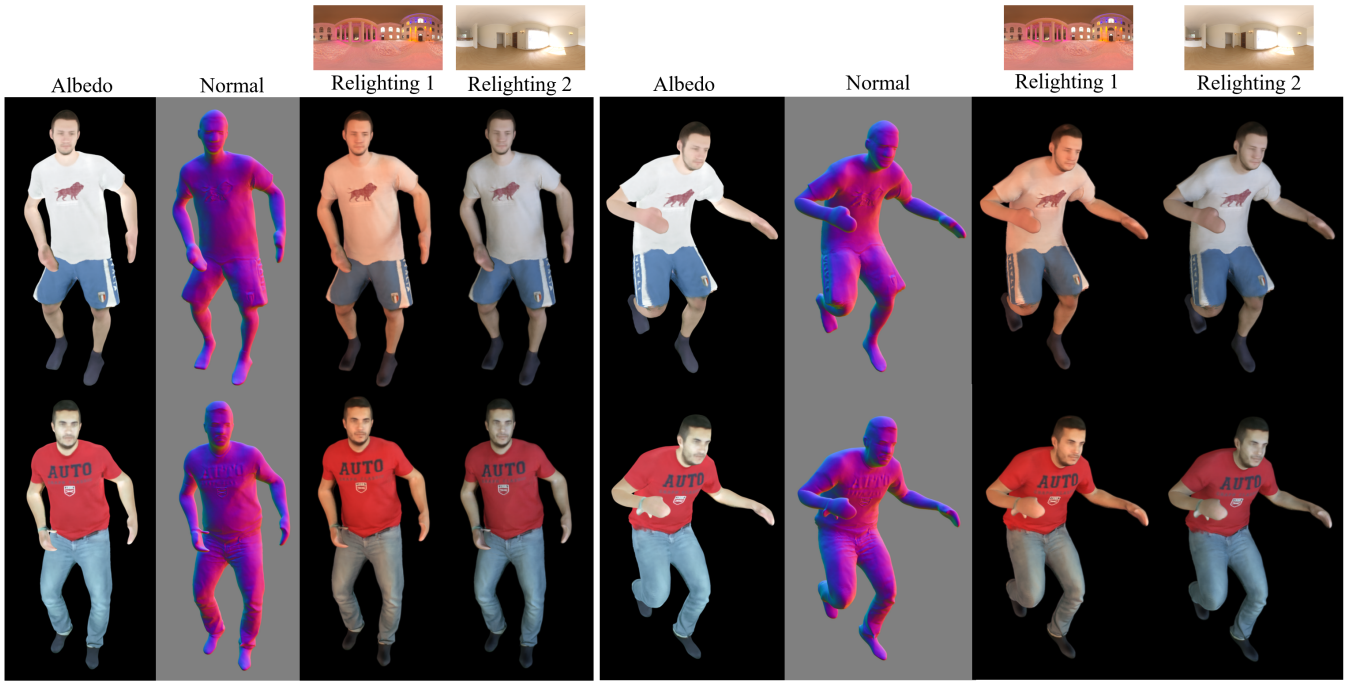

Figure 11: Results of our technique on the PeopleSnapshot dataset. From left to right of each result: the albedo of an animated pose, the corresponding normal in this pose, and two relighting results.

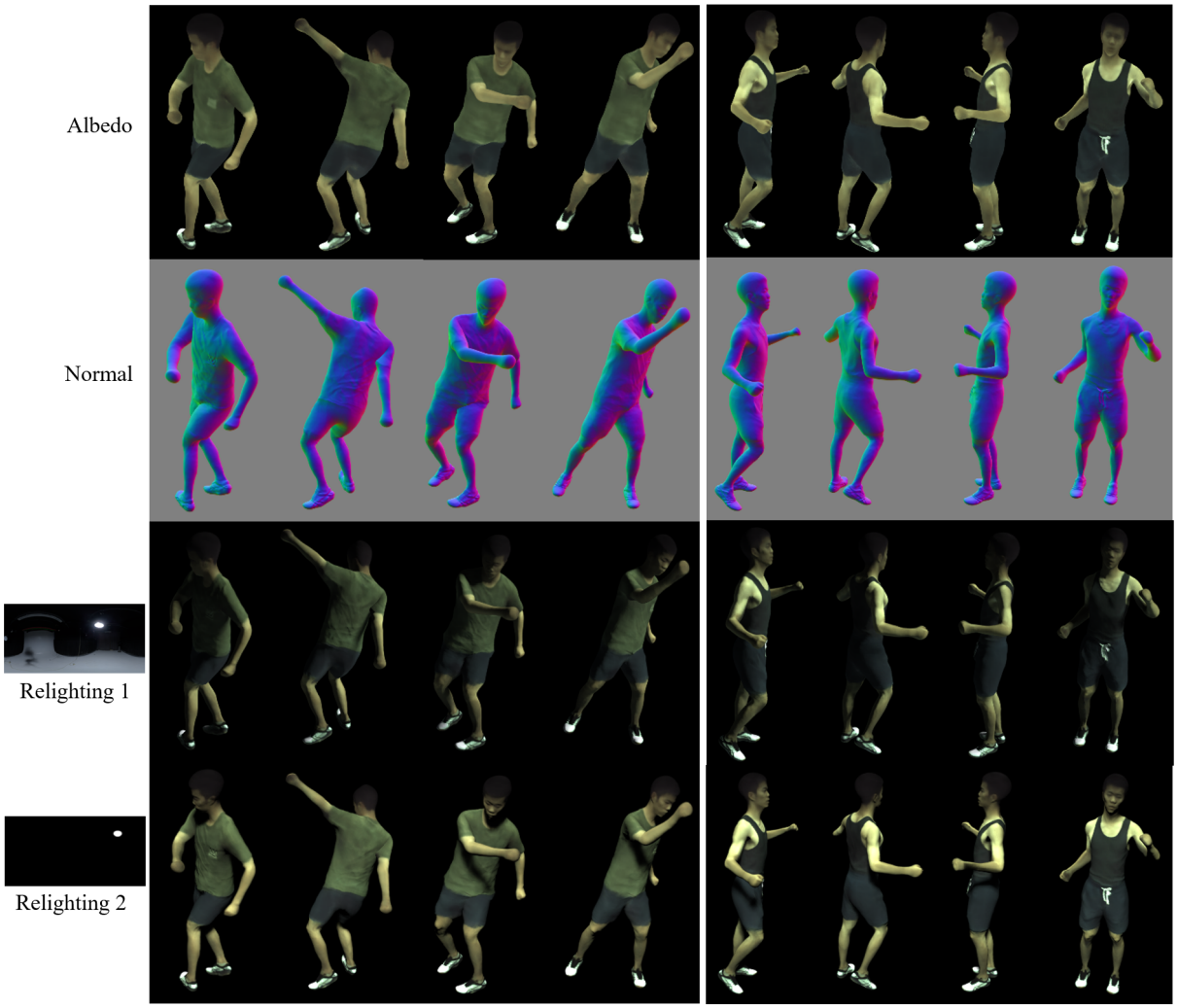

Figure 12: Results on the ZJU-MoCap dataset with single-view input. From top to bottom: the reconstructed albedo, normal of the reconstructed geometry, and two relighting results.