# Supplementary Material for Animatable Implicit Neural Representations for Creating Realistic Avatars from Videos

Sida Peng, Zhen Xu, Junting Dong, Qianqian Wang, Shangzhan Zhang,
Qing Shuai, Hujun Bao and Xiaowei Zhou

✦

**Overview.** The supplementary material has the following contents:

- Section 1 provides descriptions of baseline methods, datasets, and evaluation metrics.
- Section 2 describes the implementation details, including the derivation of transformation matrices, network architectures, volume rendering process, and training strategy.
- Section 3 provides more discussions, which aim to sufficiently evaluate our method.

## 1 Experimental details

### 1.1 Baseline methods

We compare with state-of-the-art image synthesis methods [1], [2], [3], [4] that also utilize SMPL priors. Same to our method, these methods train a separate network for each video. 1) NHR [1] extracts 3D features from input point clouds and renders them into 2D feature maps, which are then transformed into images using 2D CNNs. Since dense point clouds are difficult to obtain from sparse camera views, we take SMPL vertices as input point clouds. 2) Neural body [2] anchors a set of latent codes on the vertices of SMPL and uses a network to regress neural radiance fields from the latent codes, which are then rendered into images using volume rendering. 3) D-NeRF [3] decomposes the dynamic human into a canonical human model and a deformation field. The human model is represented as a neural radiance field, and the deformation field is predicted by an MLP network that takes time index and spatial location as input. 4) A-NeRF [4] constructs the skeleton-relative embedding for input 3D points to represent the animatable human model and jointly optimizes the input skeleton poses and network parameters during training.

| subject | S1 | S5 | S6 | S7 | S8 | S9 | S11 |
|---|---|---|---|---|---|---|---|
| training | 150 | 250 | 150 | 300 | 250 | 260 | 200 |
| test | 49 | 127 | 83 | 200 | 87 | 133 | 82 |

TABLE 1: **The number of training frames and test frames of the Human3.6M dataset.**

| subject | Twirl | Taichi | Swing1 | Swing2 | Swing3 | Warmup | Punch1 | Punch2 | Kick |
|---|---|---|---|---|---|---|---|---|---|
| training | 60 | 400 | 300 | 300 | 300 | 300 | 300 | 300 | 400 |
| test | 1000 | 1000 | 356 | 559 | 358 | 317 | 346 | 354 | 700 |

TABLE 2: **The number of video frames for each subject in the ZJU-MoCap dataset.**

### 1.2 Dataset details

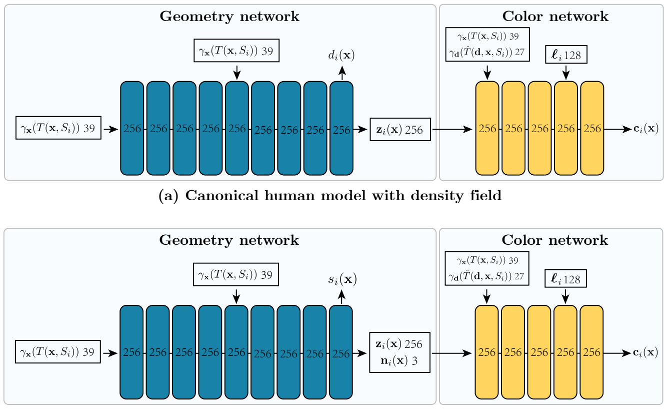
**Human3.6M [5].** Following [6], we use three camera views for training and test on the remaining view. [6] select video clips from the action "Posing" of S1, S5, S6, S7, S8, S9, and S11. The number of training frames and test frames is described in Table 1.

**MonoCap [7].** It consists of two videos "Lan" and "Marc" from DeepCap dataset [8], and two videos "Olek" and "Vlad" from DynaCap dataset [9]. "Lan" is selected from 620-th frame to 1220-th frame in the original video. "Marc" is selected from 35000-th frame to 35600-th frame. "Olek" is selected from 12300-th frame to 12900-th frame. "Vlad" is selected from 15275-th frame to 15875-th frame. Each clip has 300 frames for training and 300 frames for evaluating novel pose synthesis, respectively. We use the 0-th camera as the training view for "Lan" and "Marc". The 44-th camera is selected as the training view for "Olek". The training view of "Vlad" is the 66-th camera. We uniformly select ten cameras from the remaining cameras for test.

**ZJU-MoCap [2].** It records multi-view videos with 21 cameras and collects human poses using the marker-less motion capture system. Table 2 lists the number of training and test frames in the ZJU-MoCap dataset.

**SyntheticHuman [7].** It contains 7 human characters. Subjects S1, S2, S3, and S4 perform rotation with A-pose, which are rendered into monocular videos. Subjects S5, S6 and S7 perform random actions, which are rendered into 4-view videos. The number of video frames is listed in Table 3.

- *S. Peng, Z. Xu, J. Dong, S. Zhang, Q. Shuai, H. Bao and X. Zhou are affiliated with the State Key Lab of CAD&CG, the College of Computer Science, Zhejiang University, China.*
- *Q. Wang is with the College of Computer Science, Cornell University, USA.*
- *Corresponding authors: Xiaowei Zhou.*


Geometry network
Color network
$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 39
$d_i(\mathbf{x})$
$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 39
256
$\mathbf{z}_i(\mathbf{x})$ 256
$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 39
$\gamma_{\mathbf{d}}(\hat{T}(\mathbf{d}, \mathbf{x}, S_i))$ 27
$\boldsymbol{\ell}_i$ 128
256
$\mathbf{c}_i(\mathbf{x})$
Geometry network
Color network
$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 39
$s_i(\mathbf{x})$
$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 39
256
$\mathbf{z}_i(\mathbf{x})$ 256
$\mathbf{n}_i(\mathbf{x})$ 3
$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 39
$\gamma_{\mathbf{d}}(\hat{T}(\mathbf{d}, \mathbf{x}, S_i))$ 27
$\boldsymbol{\ell}_i$ 128
256
$\mathbf{c}_i(\mathbf{x})$

(a) Canonical human model with density field

(b) Canonical human model with signed distance field

Fig. 1: **Canonical human model.** We present two types of canonical human models. (a) One models the human geometry with density field, and (b) the other one models the geometry with the signed distance field. All layers are linear layers with softplus activations except for the final layer. The dimension of the input is shown in each block.

| subject | S1 | S2 | S3 | S4 | S5 | S6 | S7 |
|---|---|---|---|---|---|---|---|
| training | 69 | 300 | 70 | 100 | 100 | 100 | 70 |

TABLE 3: **The number of video frames for each subject in the SyntheticHuman dataset.**

### 1.3 Evaluation metrics

We follow [2] to calculate the metrics of image synthesis. Specifically, the 3D human bounding box is first projected to produce a 2D mask. Then, we calculate the PSNR metric based on the pixels inside the 2D mask. Since the SSIM metric require the image input, we compute the 2D box that bounds the 2D mask and crop the image within the box, which is used to calculate the SSIM metric. For the SyntheticHuman dataset, we calculate the reconstruction metrics every 10-th frame. For the Human3.6M and MonoCap datasets, we calculate the metrics of image synthesis every 30-th frame.

## 2 Implementation Details

### 2.1 Derivation of transformation matrices

Given the human skeleton, the LBS model [10] calculates the transformation matrices of body parts to produce the deformation field. We represent the human skeleton as $(\mathbf{J}, \boldsymbol{\theta})$, where $\mathbf{J} \in \mathbb{R}^{K \times 3}$ denotes the joint locations of $K$ joints and $\boldsymbol{\theta} \in \mathbb{R}^{3(K+1) \times 1} = [\boldsymbol{\omega}_0^T, ..., \boldsymbol{\omega}_K^T]$ denotes the $(K+1)$ relative rotation of body part with respect to its parent part in a kinematic tree using the axis-angle representation. Then, the transformation matrix of part $k$ from canonical pose $\boldsymbol{\theta}_c$ to target pose $\boldsymbol{\theta}_t$ can be represented as

$$G_k = A_k(\mathbf{J}, \boldsymbol{\theta}_t) A_k(\mathbf{J}, \boldsymbol{\theta}_c)^{-1}, \tag{1}$$

$$A_k(\mathbf{J}, \boldsymbol{\theta}) = \prod_{i \in P(k)} \begin{bmatrix} R(\boldsymbol{\omega}_i) & \mathbf{j}_i \\ 0 & 1 \end{bmatrix}, \tag{2}$$

where $R(\boldsymbol{\omega}_i) \in \mathbb{R}^{3 \times 3}$ is the converted rotation matrix of $\boldsymbol{\omega}_i$ via the Rodrigues formula, $\mathbf{j}_i$ is the $i$-th joint center, and $P(k)$ is the ordered set of parent joints of joint $k$. In practice, we adopt the SMPL skeleton [11], which has $K = 24$ parts, but this idea applies to other human skeletons [5], [12].

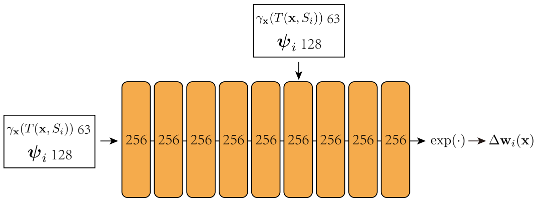

$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 63
$\boldsymbol{\psi}_i$ 128
$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 63
$\boldsymbol{\psi}_i$ 128
256
$\exp(\cdot)$
$\Delta \mathbf{w}_i(\mathbf{x})$

Fig. 2: **Neural blend weight field.** All layers are linear layers with ReLU activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T(\mathbf{x}), S_i)$ and the per-frame latent code for $\psi_i$ as input.

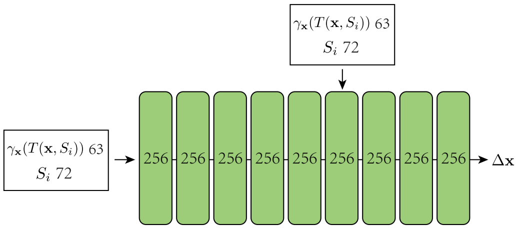

$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 63
$S_i$ 72
$\gamma_{\mathbf{x}}(T(\mathbf{x}, S_i))$ 63
$S_i$ 72
256
$\Delta \mathbf{x}$

Fig. 3: **Pose-dependent displacement field.** All layers are linear layers with ReLU activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T(\mathbf{x}), S_i)$ and the human pose $S_i$ as input.

### 2.2 Network architectures

Figures 1, 2 and 3 illustrate network architectures of canonical human model, neural blend weight field $F_{\Delta \mathbf{w}}$, and pose-dependent displacement field $F_{\Delta \mathbf{x}}$, respectively. We perform positional encoding [13] to the spatial point and viewing direction. For the canonical human model, 6 frequencies are used when encoding spatial position, and 4 frequencies are used when encoding viewing direction. For the blend weight field and displacement field, 10 frequencies are used when encoding spatial position. The dimension of appearance code $\boldsymbol{\ell}_i$ is 128.

The color network $F_{\mathbf{c}}$ takes the canonical-space viewing direction as input to better approximate the radiance function. To obtain the canonical-space viewing direction, we transform the observation-space viewing direction $\mathbf{d}$ to the canonical space based on the LBS model. Denote the weighted sum of transformation matrices in the LBS model as $[R_i^*(\mathbf{x}); t_i(\mathbf{x})] = \sum w_i^k(\mathbf{x}) G_i^k$. The deformation $\hat{T}(\mathbf{d}, \mathbf{x}, S_i)$ that transforms the viewing direction to the canonical space is defined as:

$$\hat{T}(\mathbf{d}, \mathbf{x}, S_i) = R_i^*(\mathbf{x}) \mathbf{d}, \tag{3}$$

where $R_i^*(\mathbf{x})$ is a $3 \times 3$ matrix. To validate the benefit of using the canonical-space viewing direction, we evaluate our model with the world-space viewing direction on the subject "S9" of Human3.6M dataset, which gives 23.65 PSNR and 0.887 SSIM on novel pose synthesis. In contrast, our model with the canonical-space viewing direction gives 24.45 PSNR and 0.898 SSIM, indicating that using the canonical-space viewing improves the performance.

### 2.3 Volume rendering

We can use volume rendering techniques [13], [14] to render the animatable implicit neural representation from particular viewpoints. Given a pixel at frame $i$, we emit the camera ray and calculate the near and far bounds by intersecting the camera ray with the 3D bounding box of the SMPL model. Then, we use a stratified sampling approach [13] to sample $N_k$ points $\{\mathbf{x}_k\}_{k=1}^{N_k}$. The number of sampled points $N_k$ is set as 64 in all experiments. These points are fed into the proposed pipeline to predict the densities $\sigma_i(\mathbf{x}_k)$ and colors $\mathbf{c}_i(\mathbf{x}_k)$, which are accumulated into the pixel color $\tilde{\mathbf{C}}_i(\mathbf{r})$ using the numerical quadrature:

$$\tilde{\mathbf{C}}_i(\mathbf{r}) = \sum_{k=1}^{N_k} \alpha_i(\mathbf{x}_k) \prod_{j<k} (1 - \alpha_i(\mathbf{x}_j)) \mathbf{c}_i(\mathbf{x}_k), \tag{4}$$

where $\alpha_i(\mathbf{x}_k) = 1 - \exp(-\sigma_i(\mathbf{x}_k)\delta_k)$, and $\delta_k$ is the distance between adjacent sampled points $||\mathbf{x}_{k+1} - \mathbf{x}_k||_2$.

When the human geometry is represented by the signed distance field, we first convert the predicted signed distances into volume densities and then perform the volume rendering, as [15], [16] do. Following [15], we convert signed distance $s_i(\mathbf{x}_k)$ into volume density using

$$\sigma_i(\mathbf{x}) = \begin{cases} \frac{1}{\beta}\left(1 - \frac{1}{2}\exp\left(\frac{s_i(\mathbf{x})}{\beta}\right)\right) & \text{if } s_i(\mathbf{x}) < 0, \\ \frac{1}{2\beta}\exp\left(-\frac{s_i(\mathbf{x})}{\beta}\right) & \text{if } s_i(\mathbf{x}) \geq 0, \end{cases} \tag{5}$$

where $\beta$ is a learnable parameter.

### 2.4 Losses functions

In experiments, we evaluate three types of animatable implicit neural presentations, including NeRF-NBW, NeRF-PDF, and SDF-PDF, which are optimized based on different loss functions. For NeRF-NBW, the combination of the rendering loss $L_{\text{rgb}}$ and consistency loss $L_{\text{nsf}}$ is used for training, which is defined as:

$$L_{\text{NeRF-NBW}} = L_{\text{rgb}} + L_{\text{nsf}}. \tag{6}$$

For NeRF-PDF, we use the combination of the rendering loss $L_{\text{rgb}}$ and regularization term $L_{\Delta\mathbf{x}}$, which is defined as:

$$L_{\text{NeRF-PDF}} = L_{\text{rgb}} + 0.01 L_{\Delta\mathbf{x}}. \tag{7}$$

For SDF-PDF, we use the combination of the rendering loss $L_{\text{rgb}}$, mask loss $L_{\text{mask}}$, Eikonal term $L_{\text{E}}$, and regularization term $L_{\Delta\mathbf{x}}$, which is defined as:

$$L_{\text{SDF-PDF}} = L_{\text{rgb}} + L_{\text{mask}} + 0.01 L_{\text{E}} + 0.01 L_{\Delta\mathbf{x}}. \tag{8}$$

### 2.5 Training

In all experiments, we use the Adam optimizer for the training, and the learning rate starts from $5e^{-4}$ and decays exponentially to $5e^{-5}$ along the optimization. Animatable implicit neural representations with the pose-dependent displacement field requires a single stage training on the input video, while the neural blend weight field requires the additional optimization on novel human poses based on the loss function $L_{\text{new}}$, which is described in the Section 3.4 of the main paper. To improve the capacity of our model, the neural blend weight field $\mathbf{w}^{\text{new}}$ of novel human poses does not share network parameters with the blend weight field $\mathbf{w}^{\text{can}}$ of the canonical human pose.

| | NHR | D-NeRF | NB | A-NeRF | NeRF- | NeRF- | SDF- |
|---|---|---|---|---|---|---|---|
| | [1] | [3] | [2] | [4] | NBW | PDF | PDF |
| Params. | 18.68 | 1.21 | 4.34 | 1.78 | 1.41 | 1.38 | 1.38 |

TABLE 4: **Number of network parameters.** Our model has fewer parameters than [1], [2]. The unit is in million.

## 3 Discussions

We provide more discussions on possible design choices and interesting experiments, aiming to show more insights.

**Combining neural blend weight field with pose-dependent displacement field.** The neural blend weight field can be used together with the pose-dependent displacement field to produce the deformation field. Given a human pose $S_i$ and a 3D point $\mathbf{x}$ in the observation space, we first compute the neural blend weight using $\mathbf{w}_i(\mathbf{x})$ and then leverage the LBS model to transform the observation-space point to the canonical space, resulting in the transformed point $\mathbf{x}'$. Then, the pose-dependent displacement field $F_{\Delta\mathbf{x}}$ takes $\mathbf{x}'$ as input and output the displacement to deform this point. The final point is fed into the canonical human model to predict the geometry and color. Here we use neural radiance field to represent the canonical human model. Experiments on the subject "S9" of the Human3.6M dataset show that this strategy does not perform as well as NeRF-PDF, which gives 25.94 PSNR and 0.911 SSIM on training poses, while NeRF-PDF gives 26.03 PSNR and 0.917 SSIM. The reason is that the articulated motions could be modeled by the displacement field, leading to the local minima, as discussed in [17]. A possible solution is using the coarse-to-fine optimization strategy [17].

**Performance of pose-dependent blend weight field.** We define an MLP network that maps the 3D point and the human pose to the residual vector of blend weight, denoted as $F'_{\Delta\mathbf{w}} : (\mathbf{x}, S) \rightarrow \Delta\mathbf{w}$. Then the residual vector of blend weight is used to update the SMPL blend weight based on

$$\mathbf{w}(\mathbf{x}, S) = \text{norm}(F'_{\Delta\mathbf{w}}(\mathbf{x}, S) + \mathbf{w}^{\text{s}}(\mathbf{x}, S)). \tag{9}$$

We combine this deformation field with canonical neural radiance field to represent the dynamic human. On the subject "S9" of the Human3.6M dataset, this representation gives 0.877 PSNR on novel poses. In contrast, NeRF-NBW gives 0.885 PSNR, indicating that pose-dependent blend weight field does not generalize well to novel human poses.

**Comparison of the number of network parameters.** Table 4 compares the number of network parameters of our and other methods. Our method has a smaller model size than NHR [1] and Neural Body [2].

**Running time analysis.** We test the running time of NeRF-NBW, NeRF-PDF, and SDF-PDF that render a 512 × 512 image on a desktop with an Intel i7 3.7GHz CPU and a GTX 2080 Ti GPU. Table 5 lists the results of running time. Because the number of points sampled along the ray is only 64 and the scene bound of a human is small, the rendering speed of our method is relatively fast.

## References

[1] M. Wu, Y. Wang, Q. Hu, and J. Yu, "Multi-view neural human rendering," in *CVPR*, 2020.

| | NeRF-NBW | NeRF-PDF | SDF-PDF |
|---|---|---|---|
| Deformation | 0.50 | 0.64 | 0.64 |
| Canonical human model | 1.11 | 1.05 | 2.38 |
| Volume rendering | 0.02 | 0.02 | 0.02 |
| Total | 1.63 | 1.71 | 3.04 |

TABLE 5: **Running time.** This table analyzes the running time of three representations. The unit is in second. "Deformation" means predicting the pose-driven deformation field and transformating the observation-space points to the canonical space. "Canonical human model" means predicting the geometry and appearance of the canonical human model. "Volume rendering" means accumulating the predicted geometry and appearance into pixel colors. Because SDF-PDF needs to additionally calculate the poinr normal, it takes more time to predict the geometry and color.

[2] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," in *CVPR*, 2021.

[3] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *CVPR*, 2021.

[4] S.-Y. Su, F. Yu, M. Zollhöfer, and H. Rhodin, "A-nerf: A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose," in *NeurIPS*, 2021.

[5] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *PAMI*, 2013.

[6] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, X. Zhou, and H. Bao, "Animatable neural radiance fields for modeling dynamic human bodies," in *ICCV*, 2021.

[7] S. Peng, S. Zhang, Z. Xu, C. Geng, B. Jiang, H. Bao, and X. Zhou, "Animatable neural implict surfaces for creating avatars from videos," *arXiv preprint arXiv:2203.08133*, 2022.

[8] M. Habermann, W. Xu, M. Zollhofer, G. Pons-Moll, and C. Theobalt, "Deepcap: Monocular human performance capture using weak supervision," in *CVPR*, 2020.

[9] M. Habermann, L. Liu, W. Xu, M. Zollhoefer, G. Pons-Moll, and C. Theobalt, "Real-time deep dynamic characters," in *SIGGRAPH Asia*, 2021.

[10] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *SIGGRAPH*, 2000.

[11] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM TOG*, 2015.

[12] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *PAMI*, 2018.

[13] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[14] J. T. Kajiya, "The rendering equation," in *SIGGRAPH*, 1986.

[15] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *NeurIPS*, 2021.

[16] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *NeurIPS*, 2021.

[17] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman, "Humannerf: Free-viewpoint rendering of moving people from monocular video," in *CVPR*, 2022.