

Animatable Neural Radiance Fields from Monocular RGB-D

Tiantian Wang^{1*}, Nikolaos Sarafianos², Ming-Hsuan Yang¹, Tony Tung²

¹University of California, Merced, ²Meta Reality Labs Research, Sausalito

Abstract

This paper aims at representing animatable photo-realistic humans under novel views and poses. Recent work has shown significant progress with dynamic scenes by exploring shared canonical neural radiance fields. However learning a user-controlled model for novel poses remains a challenging task. To tackle this problem, we introduce a novel method to integrate observations across frames and encode the appearance at each individual frame by utilizing the human pose that models the body shape and point clouds which cover partial part of the human as the input. Specifically, our method simultaneously learns a shared set of latent codes anchored to the human pose among frames, and learns an appearance-dependent code anchored to incomplete point clouds generated by monocular RGB-D at each frame. A human pose-based code models the shape of the performer whereas a point cloud-based code predicts details and reasons about missing structures at the unseen poses. To further recover non-visible regions in query frames, we utilize a temporal transformer to integrate features of points in query frames and tracked body points from automatically-selected key frames. Experiments on various sequences of humans in motion show that our method significantly outperforms existing works under unseen poses and novel views given monocular RGB-D videos as input.

1 Introduction

3D human digitization has drawn significant attention in recent years, with a wide range of applications such as photo editing, video games and immersive technologies. To obtain photo-realistic renders of free-viewpoint videos, existing approaches [5, 33] require complicated equipment with expensive synchronized cameras, which makes them difficult to be applied to realistic scenarios. To date, modeling detailed appearance of dynamic clothed humans such as cloth wrinkles and facial details such as eyes from videos remains a challenging problem.

To represent static scenes, neural radiance field-based models (NeRF) [22] learn an implicit representation using neural networks, which has enabled photo-realistic rendering of shape and appearance from images. With dense multi-view observations as input, NeRF encodes density and color as a function of 3D coordinates and viewing directions by the MLPs along with a differentiable renderer to synthesize novel views. While it shows unprecedented visual quality on static scenes, applying it to high quality free-viewpoint rendering of humans in dynamic videos remains a

*This work was conducted during an internship at Meta Reality Labs Research.

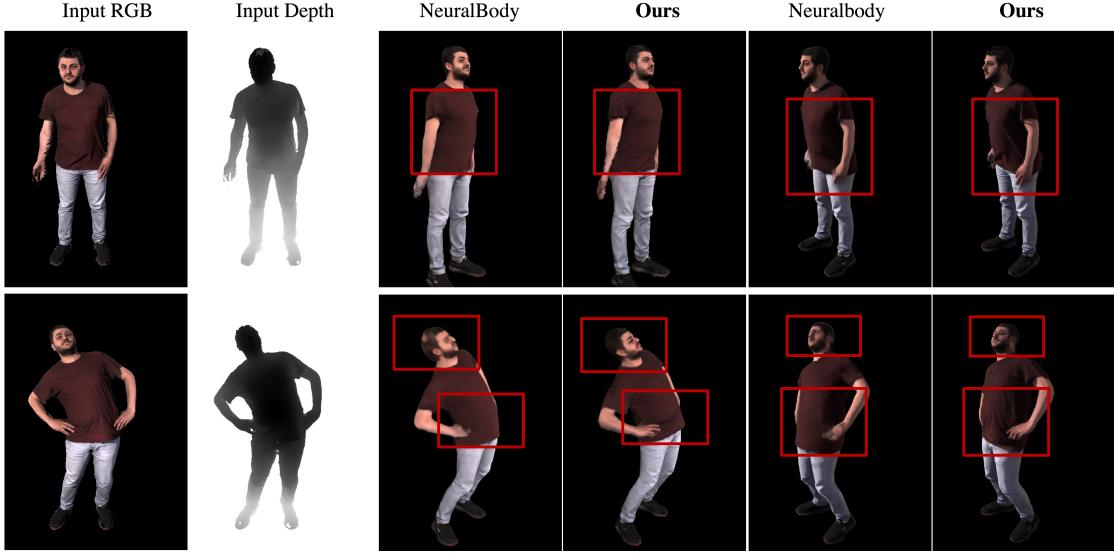


Figure 1: **NeRF with monocular RGB-D.** Given one RGB-D video as input, we predict novel views with body poses unseen from training with fine-level details (wrinkles, facial characteristics) that prior works such as Neuralbody [25] struggles to obtain.

challenging task. To generalize NeRF from static scenes to dynamic videos, D-NeRF [28] encodes a time step t to differentiate motions across frames and converts scenes from the observation space to a shared canonical space in order to model the neural radiance field. As such, they can handle dynamic scenes to some extent but the poses remain uncontrollable by users. Furthermore, some approaches [19, 26] introduce human pose as an additional input to serve as a geometric guidance for different frames. However, they either cannot generalize to novel poses or need more than one input view.

To overcome these limitations, we propose a novel approach by learning implicit radiance fields based on pose and appearance representations for high fidelity novel view and pose synthesis. Motivated by [26], we leverage the human pose extracted from the parametric body model as a geometric prior to model motion information across frames. Shared latent codes anchored to the human poses are optimized, which can integrate information across frames. However, a model which only formulates latent codes in a shared space will not generalize well to unseen poses without test-time optimization of the latent codes. To generalize the model to unseen poses, we propose to model the appearance information with the assist of the single-view RGBD image. Our model learns the appearance code anchored to incomplete point clouds in the 3D space. Point clouds can be obtained by using single-view depth information to lift the RGB image to the 3D space, which provides partial visible parts of the human body. The learned implicit representation enables reasoning of the unknown regions and complements the missing details on the human body.

To further leverage the temporal information from multiple frames, we introduce a temporal transformer that aggregates the trackable information. To achieve this, we use the parametric body model to track points from the query frame to the key frames. Then, based on the learned implicit representation, we extract the pose code across frames and forward it into the temporal

transformer for feature aggregation. Our method is extensively evaluated against state-of-the-art techniques on several sequences of humans in motion and exhibits significantly higher rendering quality of novel view and novel pose synthesis. In addition, we are able to reconstruct fine-level details such as cloth wrinkles, face details or logos at a resolution and fidelity that the prior works such as NeuralBody [26] was not able to achieve (see Fig. 1). The contributions of this work are summarized below:

1. We introduce a new framework with monocular RGB-D as input and show significant improvement on the unseen poses compared to existing methods, with high-fidelity reconstruction of fine-level facial, cloth and body details.
2. We combine pose and appearance representations by modeling shared information across frames and specific information at each individual frame. These two representations help the model to generalize better to novel poses compared to only utilizing the pose representation.
3. A temporal transformer is introduced to combine information across frames, which helps to recover non-visible details in the query frame (at unseen views).

2 Related Work

3D Neural Representations. Early 3D shape representation works can be classified into three categories: point-based methods [1, 29], voxel-based methods [4, 37] and mesh-based methods [2, 11, 39]. Implicit representations are then used to represent shapes by reconstructing a continuous surface geometry, which utilizes the spatial coordinates as the input and outputs the signed distances or occupancy values. With advances in differential rendering methods, geometry and appearance can be learned from multi-view observations. Related works can be categorized into static [20, 22, 34, 35, 40, 41] and dynamic scenes [3, 8–10, 13, 16, 17, 23–26, 28, 31, 36].

Static Scene Representations. SRN [35] represents scenes as continuous functions that maps 3D coordinates to a feature representation of local scene properties and formulates the image as a differentiable ray-marching algorithm. NSVF [20] utilizes a sparse voxel octree to represent a set of voxel-bounded implicit fields. A differentiable ray-marching operation is adopted to render views from a set of posed RGB images. NeRF [22] achieves state-of-the-art renderings for novel view synthesis by optimizing a neural radiance field for a scene, which maps 3D coordinates and viewing directions to density and color using a neural network. While NeRF can render photo-realistic images given dense images as input, it is limited mostly to static scenes.

Dynamic Scene Representations. Dynamic NeRFs [26, 28] extend NeRF to dynamic scenes by introducing a latent deformation field or human poses. D-NeRF [28] learns a mapping from the observation space to the canonical space at a particular time. NeuralBody [26] proposes a set of latent codes shared across all frames anchored to a human body model in order to replay character motions from arbitrary view points under training poses. These methods, where the deformations are learned by neural networks, allow to handle general deformation, and synthesize novel poses by using interpolation in the latent space. However, the human poses cannot be controlled by users or the synthesis fails under novel poses. Human pose based representation can model the body shape at any time step but will fail to capture detailed appearance. To overcome this problem, we propose to construct the appearance-based representation by utilizing the 2D features anchored to the point clouds as an input.

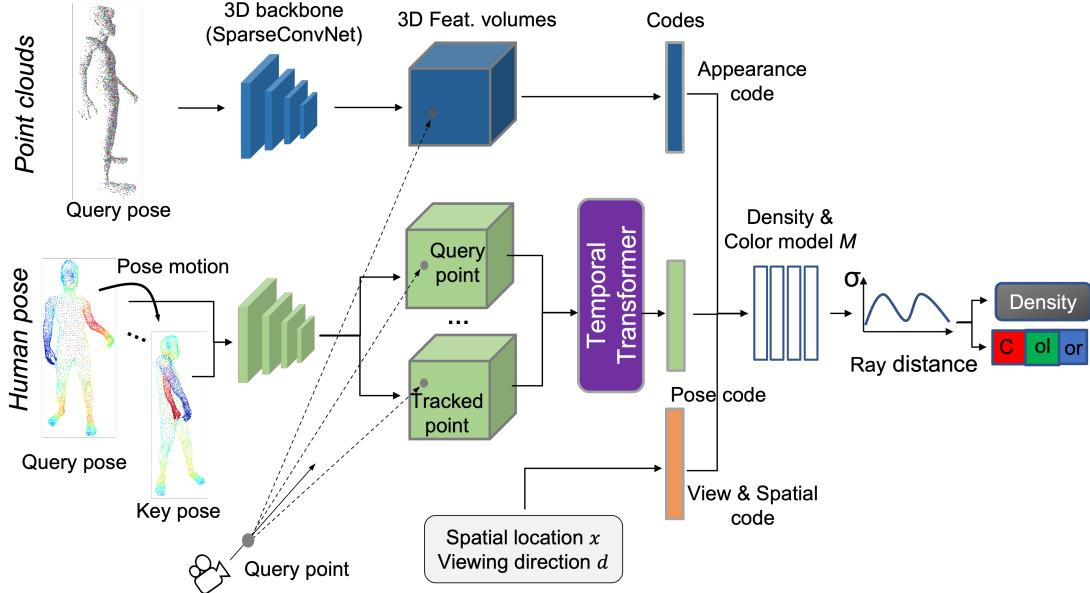


Figure 2: **Overview.** Given a query point as input, our method learns pose and appearance codes which can simultaneously integrate shared information across frames and model the appearance information at each frame. The pose and appearance codes are anchored to the human pose and point clouds, respectively. In addition, we use the body motion to track 3D points from the query frame to the key frames and extract the pose code from the key frames. A transformer fusion module is proposed to combine the pose codes of all frames. Similar to NeRF [22], we use spatial location and viewing direction as the extra inputs and train the model to predict the density and color for each 3D point.

Dynamic Scene Fusion. To model the temporal cues across frames, previous works [7, 16, 17, 42] combine motion information. Li *et al.* [17] propose to learn dense scene flow fields that aim to learn 3D offset vectors from a point in time t to the same point in time $t-1$ and time $t+1$. The offsets are implicitly supervised with 2D optical flow. Kwon *et al.* [16] employ a temporal transformer to integrate skeletal features across different frames. The vertices of the human body are first reprojected to the 2D image plane and then image features are sampled to obtain the skeletal features. Although both the work by Kwon *et al.* [16] and our method use the temporal transformer, the way we use the transformer differs from [16]. Kwon *et al.* [16] use the transformer to combine pixel-aligned skeletal features. The features are obtained by first projecting the vertices to the 2D image plane and then sampled from 2D images using bilinear interpolation. In addition, they require multiple views as input as inaccurate features are extracted when projecting the 3D vertices into single view. Instead of combining the skeletal features, we propose to use the transformer to combine the pose codes for any 3D point and its tracked points. Our method optimizes the 3D feature volumes and does not require multiple views as input.

3 Methodology

Given a monocular RGB-D video of a human in motion, we aim to synthesize free-viewpoint videos of the person under novel views and new poses. We denote the set of video frames as $\{I_t | t = 1, \dots, N_f\}$, where t represents the frame index and N_f is the number of frames. To avoid the influence of background changes due to the camera movements, we use a mask to remove the background color and only focus on the human in the foreground.

We start by briefly introducing the notation and concept of NeRF for static scenes in Sec. 3.1. In Sec. 3.2, NeRF is extended to dynamic human performers which are conditioned on the pose code shared across all frames. In Sec. 3.3, the appearance representation is introduced by using a single RGB-D image as the cue for each frame. To integrate the motion information across frames, we propose to employ a temporal transformer as discussed in Sec. 3.4 to help recover the non-visible parts. Volume rendering and the corresponding objective functions are introduced in Sec. 3.5 and Sec. 3.6. The overview of our approach is illustrated in Fig. 2.

3.1 Neural Radiance Fields

Neural Radiance Fields (NeRFs) [22] represent a static scene as a radiance field and render the color using classical volume rendering [14]. NeRF utilizes the 3D location $\mathbf{x} = (x, y, z)$ and 2D viewing direction \mathbf{d} as input and outputs color \mathbf{c} and volume density σ with a neural network for any 3D point,

$$F_\Theta : (\gamma_x(\mathbf{x}), \gamma_d(\mathbf{d})) \rightarrow (\mathbf{c}, \sigma). \quad (1)$$

To render the color of an image pixel, NeRF uses the volume rendering integral equation by accumulating volume densities and colors for all sampled points along the camera ray. Let \mathbf{r} be the camera ray emitted from the center of projection to a pixel on the image plane. The expected color of that pixel bounded by h_n and h_f is then given by:

$$\tilde{C}(\mathbf{r}) = \int_{h_n}^{h_f} T(h) \sigma(\mathbf{r}(h)) \mathbf{c}(\mathbf{r}(h), \mathbf{d}) dh, \quad (2)$$

where $T(h) = \exp(-\int_{h_n}^h \sigma(\mathbf{r}(s)) ds)$. The function $T(h)$ denotes the accumulated transmittance along the ray from h_n to h . NeRF is trained on a collection of images for each static scene with known camera parameters, and can render scenes with photo-realistic quality. To extend NeRF to model the dynamic human body, we propose to learn the implicit representation to represent the shape and appearance information of the human. Specifically, we introduce a pose-conditioned representation *shared by all frames* and an appearance-conditioned representation *specified by each individual frame*.

3.2 Pose-conditioned Representation

For each frame, we assume the 3D human model is given. We first extract the vertices from the posed 3D mesh and aim to learn a set of pose codes $\mathbf{Z} = \{\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^{N_m}\}$ anchored to the vertices of the human body model at frame t . Here N_m denotes the number of codes. Similar to [26], the dimension of each pose code is set to 16. Then the implicit representation is learned by forwarding the pose code into a neural network, which aims to represent the geometry and shape of a human performer. The pose space is shared across all frames, which can be treated as a common canonical

space and enables the representation of a dynamic human based on the NeRF. Finally a neural network is utilized to learn the density and color for any 3D point and volume rendering is used to render the RGB value for each pixel.

The pose codes anchored to the body model are relatively sparse in the 3D space and directly calculating the pose codes using trilinear interpolation would lead to less effective features for most points. To overcome this challenge, motivated by [26, 27, 32, 38], a SparseConvNet is used which propagates the codes defined on the mesh surface to the nearby 3D space. Specifically, to acquire the pose code for each point sampled along the camera ray, we utilize the trilinear interpolation to query the code at continuous 3D locations. Here the pose code for point \mathbf{x}_t^i at frame t is represented by $\phi(\mathbf{x}_t^i, \mathbf{Z})$ and will then be forwarded into a neural network to predict the density and color.

The pose codes learned in the shared space on all frames can model the human shape well in both known and unseen poses. However, the synthesized views still lack details under novel poses without optimizing the pose codes. To model the details on each frame, we propose an appearance-conditioned implicit representation by utilizing the single view RGB-D as the reference input.

3.3 Appearance-conditioned Representation

A monocular RGB image can serve as the appearance prior for the the human body under one view. To learn detailed information at each individual frame, we propose to learn the appearance code anchored to the point clouds. The points clouds are generated by the 2D image lifted to the 3D space using the depth image. The point clouds generated in this way model the partial body of the human performer and show details such as wrinkles on the clothes. Given a 2D pixel \mathbf{p}_t^i and corresponding depth value d_t^i , the point clouds generation process can be formulated as:

$$\mathbf{p}_t^{s_i} = F_\varphi(\mathbf{p}_t^i, d_t^i). \quad (3)$$

Here $\mathbf{p}_t^{s_i}$ is the 3D point generated by the 2D pixel for frame t . $F(\dots)$ is the function generating a 3D point given a 2D pixel and camera pose. Different from the pose-conditioned latent codes that are shared across all frames, here the proposed appearance-conditioned codes are anchored to the point clouds, which are obtained from the pixel-aligned features extracted from the image encoder E . To take advantage of the rich semantic and detailed cues from images, we use a 2D convolution network (ResNet34 [12]) to encode the image feature map $E(I_t)$ for the given image I_t . Specifically, we first extract features from the ResNet34, and then utilize three convolutional layers to reduce the dimension followed by a SparseConvNet to encode the features anchored to the sparse point clouds. To obtain the appearance code for each point sampled along the camera ray, we utilize the trilinear interpolation to query the code at the continuous 3D locations. $\psi(\mathbf{x}_t^i, E)$ is adopted to represent the appearance code for point \mathbf{x}_t^i . The appearance code together with the pose code will be forwarded into a neural network to predict the density and color. The appearance code learned on each single frame can model the details on the human body and help recover some missing pixels in the 3D space.

3.4 Temporal Fusion Module

Frames from different time steps can provide complementary information to the query frame. Given the features extracted from the key frames, a temporal transformer is utilized to effectively integrate

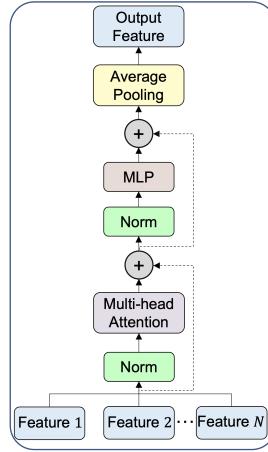


Figure 3: **Temporal Fusion Module.** Given the pose codes from the query point and tracked points as input, temporal fusion aims to aggregate the codes by using a transformer-based structure.

the features. To obtain the corresponding pixels in the key frame, we use the body model extracted from each frame to track the points.

Point Tracking. First, N_s points on each face of the mesh are randomly sampled, which result in $N_s \times N_m$ points on the whole surface of a human body. Here N_m represents the number of faces. We calculate the distance between a 3D point sampled on the camera ray and all points on the surface at the query frame I_t . Here we only keep each sample \mathbf{x}_t^i close to the surface for rendering the color if $\min_{v \in \mathcal{V}_t} \|\mathbf{x}_t^i - v\|_2 < \gamma$ and obtain the nearest point $\hat{\mathbf{x}}_t^i$ on the surface at frame I_t , where \mathcal{V}_t is the set of sampled points. In addition, we can track the points at different frames that match $\hat{\mathbf{x}}_t^i$ by the body motion, and assign the feature of the tracked points to \mathbf{x}_t^i .

Key Frame Selection. We automatically select three key frames from the training frames. We first calculate the distances between all training poses and the pose for the query frame \mathbf{S}_t by $\|\mathbf{S}_t - \mathbf{S}_j\|_2 (j \in N_f)$ and keep the frames with the K -NN distances. Here \mathbf{S} are the coordinates of the vertices extracted from the body mesh and K is set to 2. In addition, we select the first frame as the fixed key frame. For simplicity, the key frame selection strategy is not trained with the whole model.

Temporal Fusion. After obtaining the pose code from N frames ($K+1$ key frames and one query frame), a transformer based structure [6] is employed to take the N features as input and utilize a multi-head attention mechanism and feed-forward multi-layer perceptron (MLP) for feature aggregation. The fusion module is described in Fig. 3.

Here we introduce the multi-head self-attention mechanism, which generates a trainable associate memory with a query, key and value to an output via linearly transforming the input. Given the input feature $\phi(\mathbf{x}_t^i, \mathbf{Z})$, the query, key and value are represented by $f_q(\phi(\mathbf{x}_t^i, \mathbf{Z}))$, $f_k(\phi(\mathbf{x}_t^i, \mathbf{Z}))$ and $f_v(\phi(\mathbf{x}_t^i, \mathbf{Z}))$ respectively. The query and the key are used to calculate the attention map using the multiplication operation, which represents the correlation between all the features. The attention map is used to retrieve and combine the feature in the value. Formally, the attention

weight for point \mathbf{x}_t^i in frame t and tracked point \mathbf{x}_k^i in frame k is calculated by:

$$a_{t,k}^i = \psi\left(\frac{f_q(\phi(\mathbf{x}_t^i, \mathbf{Z})) \cdot f_k(\phi(\mathbf{x}_k^i, \mathbf{Z}))^\top}{\sqrt{d}}\right), \quad (4)$$

where \sqrt{d} is a scaling factor based on the network depth, and $\psi(\cdot)$ denotes the softmax operation. The aggregated feature for input $\phi(\mathbf{x}_t^i, \mathbf{Z}_t)$ is formulated as:

$$\phi'(\mathbf{x}_t^i, \mathbf{Z}) = \sum_{k \in \mathcal{K}} \phi(\mathbf{x}_t^i, \mathbf{Z}) \cdot a_{t,k}^i + f_v(\phi(\mathbf{x}_t^i, \mathbf{Z})), \quad (5)$$

where \mathcal{K} denotes the index set of the combined frames. In this work, multi-head self-attention is adopted by running multiple self-attention operations, in parallel. The results from different heads are integrated to obtain the final output. After the self-attention mechanism, each input feature contains its original information and also takes into account the information from all other frames. As such, the information from key frames and the query frame are combined together. Average pooling is then employed to integrate all features, which serves as the output of the temporal fusion module. In our implementation, we do not adopt any positional encoding on the input feature sequence.

3.5 Density and Color Regression

Figure 2 shows the prediction of density and color that are represented by a neural network. For each frame, the network takes the pose code, appearance code, spatial location and viewing direction as the input and outputs the density and color for each point in the 3D space. Similar to [22, 30], we apply positional encoding to both the viewing direction \mathbf{d} and the spatial location \mathbf{x} by mapping the inputs to a higher dimensional space. For frame t , the volume density and color at point \mathbf{x}_t^i is predicted as a function of the latent codes, which is defined as:

$$(\sigma_t^i, \mathbf{c}_t^i) = M(\phi(\mathbf{x}_t^i, \mathbf{Z}), \psi(\mathbf{x}_t^i, E), \gamma_d(\mathbf{d}_t^i), \gamma_x(\mathbf{x}_t^i)), \quad (6)$$

where $M(\cdot)$ represents a neural network. $\gamma_d(\mathbf{d}_t^i)$ and $\gamma_x(\mathbf{x}_t^i)$ are the positional encoding functions for viewing direction and spatial location, respectively. Details of the network architecture can be found in the supplementary material.

3.6 Objective Functions

We optimize the proposed model using the following objective function:

$$\mathcal{L} = \mathcal{L}_{c1} + \mathcal{L}_{c2}, \quad (7)$$

where \mathcal{L}_{c1} and \mathcal{L}_{c2} denote the reconstruction loss for the rendered pixels and image loss for the image decoder network D . The image decoder comprises multiple Conv2D layers behind the ResNet34, which aims to reconstructed the input image. We render the color of each ray using both the coarse and fine set of samples, and minimize the mean squared error between the rendered pixel color $\tilde{C}_c(\mathbf{r})$ and ground-truth color $C(\mathbf{r})$ for training:

$$\mathcal{L}_{c1} = \sum_{\mathbf{r} \in R} \|\tilde{C}_c(\mathbf{r}) - C(\mathbf{r})\|_2^2 + \|\tilde{C}_f(\mathbf{r}) - C(\mathbf{r})\|_2^2, \quad (8)$$

Models	Train Views	Sequence1		Sequence2		Sequence3		Sequence4		Sequence5	
		PSNR↑	SSIM↑								
NeRF+pose	✓	16.87	0.59	20.75	0.67	12.76	0.32	19.94	0.54	23.81	0.77
NeRF+pose+depth	✓	17.32	0.66	20.93	0.72	13.08	0.34	20.65	0.61	23.91	0.80
NeuralBody	✓	19.08	0.77	22.83	0.79	14.12	0.46	23.84	0.77	24.66	0.83
NeuralBody+depth	✓	19.67	0.77	23.81	0.80	14.82	0.48	24.01	0.76	24.70	0.83
Ours	✓	19.21	0.75	24.91	0.81	15.72	0.49	24.96	0.77	24.95	0.84
NeRF+pose		16.88	0.59	20.03	0.65	12.05	0.29	19.87	0.52	22.15	0.70
NeRF+pose + depth		17.65	0.71	21.60	0.74	12.25	0.30	21.01	0.54	22.47	0.75
NeuralBody		19.91	0.79	22.76	0.79	13.52	0.41	23.81	0.77	23.17	0.78
NeuralBody+depth		19.92	0.78	23.80	0.80	13.86	0.43	23.92	0.75	23.49	0.79
Ours		19.06	0.73	24.89	0.82	15.01	0.45	24.63	0.74	23.91	0.81

Table 1: **Quantitative Comparison** on training (first five rows) and novel views (last five rows) under novel poses across all sequences.

where R is the set of rays. \tilde{C}_c and \tilde{C}_f denote the prediction of the coarse and fine networks.

$$\mathcal{L}_{c2} = \sum_{\mathbf{p} \in \mathcal{I}} \|\tilde{I}(\mathbf{p}) - I(\mathbf{p})\|_2^2. \quad (9)$$

The symbol $\tilde{I}(\mathbf{p})$ and $I(\mathbf{p})$ represent the reconstructed and ground truth colors for pixel \mathbf{p} . \mathcal{I} is the set of pixels.

4 Implementation Details

Network details. For the encoder E , we extract a feature pyramid [18] from each image similar to [41]. A ResNet34 backbone pretrained on ImageNet is utilized for our experiments. The output feature of the decoder has 1/4 the spatial resolution compared to the input image. Multi-scale features are extracted prior to the fourth pooling layer. We extract pixel-aligned features using bilinear interpolation, and then concatenate them to form a latent vector of size 256. To construct the image decoder D , we simply connect several convolutional layers with upsampling operation to reconstruct the input image.

For the transformer network, we utilize three heads for the self-attention module, which has a similar structure as [6]. Following NeRF [22], we perform hierarchical volume sampling and simultaneously optimize a coarse and fine network with identical network architecture. At the coarse scale, we sample a set of M_c points using stratified sampling,. With the prediction of the coarse network, we then sample another set of points along each camera ray, where samples are more likely to be located at relevant regions for rendering. We sample additional M_f locations and use all $M_c + M_f$ locations to render the fine results, where M_c and M_f are set to 64 respectively.

Training details. We train all layers using Adam [15]. The base learning rates for the encoder-decoder network and other layers are 10^{-3} and 5×10^{-4} , which decay exponentially during the optimization. Additional network architecture as well as implementation details will be provided in the supplementary material.

Model Variant	PSNR↑	SSIM↑	# frames	PSNR↑	SSIM↑
W/o appearance-based representation	23.81	0.80	1	24.05	0.80
W/o temporal transformer	23.74	0.79	3 (Ours)	24.89	0.82
Ours	24.89	0.82	5	24.92	0.82

Table 2: **Left:** Ablation Study of appearance-based representation and transformer fusion module on *Sequence2* **Right:** Effect of the number of key frames on *Sequence2*

5 Experiments

Datasets. To train our method, we rely on four sequences of real humans in motion that captured with a 3dMD full-body scanner as well as a single sequence of a synthetic human in motion. The 3dMD body scanner comprises 18 calibrated RGB cameras that can capture a human in motion performing various actions and facial expressions and output a reconstructed 3D geometry and material image file per frame. These scans tend to be noisy but can capture facial expressions and fine-level details like cloth wrinkles. The synthetic scan is a high-res animated 3D human model with synthetic clothes (t-shirt and pants) that were simulated. Unlike the 3dMD scans, this 3D geometry is very clean but it lacks facial expressions. We render RGB and Depth for all real and synthetic sequences from 31 views at 2048×2048 resolution that cover the whole hemisphere (very similar to the way that NeRF data is generated) at 6 fps using Blender Cycles.

The number of video frames in our experiments varies between 200 to 600 depending on the sequence. The image resolution for the training and test is set to 1024×1024 . To train our model, we select the first half of the frames for training and the rest frames for inference. Both training and test frames contain large variations in terms of the motion and facial expressions. At training and test stages, a single RGBD image at each frame is used as the input. All the input RGBD images at different frames share the same camera pose. In addition, 29 more views with different camera poses are used to train the network. The output is a rendered view given any camera pose (not including the camera pose of the input RGBD image).

Evaluation Metrics. Following existing approaches [22, 26], we evaluate the performance on the proposed dataset using two metrics, including the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM).

Baselines. We evaluate the proposed model against state-of-the-art methods on view synthesis (training and novel views) under unseen poses:

- NeRF+pose: Similar to [21], we extend NeRF [22] from the static scenes to a pose-conditioned NeRF by utilizing the pose vector as an extra input.
- NeRF+pose+depth: We extend NeRF+pose by using the depth information. Specifically, single RGBD image is first forwarded into the ResNet34 to extract the 2D image feature. Given any 3D point sampled on the camera ray, the pixel-aligned feature is extracted which serves as an extra input.
- NeuralBody: NeuralBody [26] extends NeRF to dynamic scenes using the latent codes anchored to the human pose as an extra input besides the coordinate and viewing direction.
- NeuralBody+depth: 2D image feature is first extracted by using the single RGBD image as the input. The vertices on the human body are projected to the 2D space to extract the skeletal feature. The skeletal feature together with the latent codes are integrated, which are anchored to the vertices.

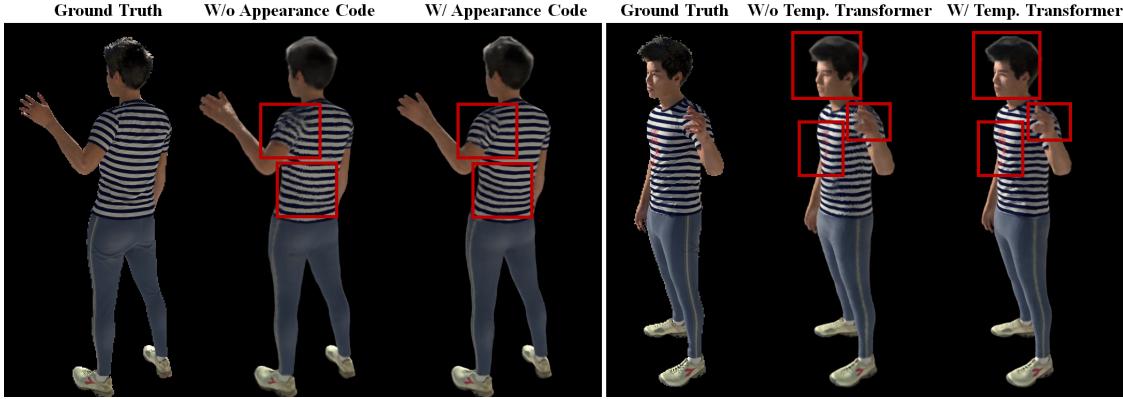


Figure 4: **Ablation Studies.** Effect of the appearance code and temporal transformer on synthesized images. Both components can help recover better details on the human.

For training and inference, these baselines use the same camera views and human poses as ours for fair comparison.

5.1 Experimental Results

We evaluate all methods on unseen new poses. Results on novel as well as training views are presented.

Novel view and novel pose. Table 1 shows quantitative results. The proposed method clearly outperforms other competitive approaches on most of the video sequences. All baseline methods, have a hard time generating realistic renders when the test poses deviate significantly from what was seen during training.

Comparing with baseline models, the proposed method can generate better details on the human body, which indicates that the point cloud-based representation and the temporal transformer can help reason the missing structure and recover the non-visible parts at the unseen poses.

Training view and novel pose. The quantitative results on training views and novel poses are shown in Table 1. On most video sequences, the proposed method performs better compared to all other baselines.

Qualitative Results. We show visual comparison with other methods under novel poses in Fig. 6. With the human pose as the geometric guidance, NeuralBody can predict the body shape well but fails to render fine-level details on the human body. NeuralBody does not generalize well to novel poses because the shared latent codes across all frames are not optimized during the test stage. More specifically, this model overfits to the training poses and generates blurry results for unseen poses, as one can observe in Fig 6 for the details on the body (1st row) and head (2nd row).

5.2 Ablation Studies

We analyze the effects of key components in our method with ablation studies.

Contribution of appearance-based representation. We analyze the effect of the appearance-based representation in Table 2 and Fig. 4. Using the appearance code brings performance improve-

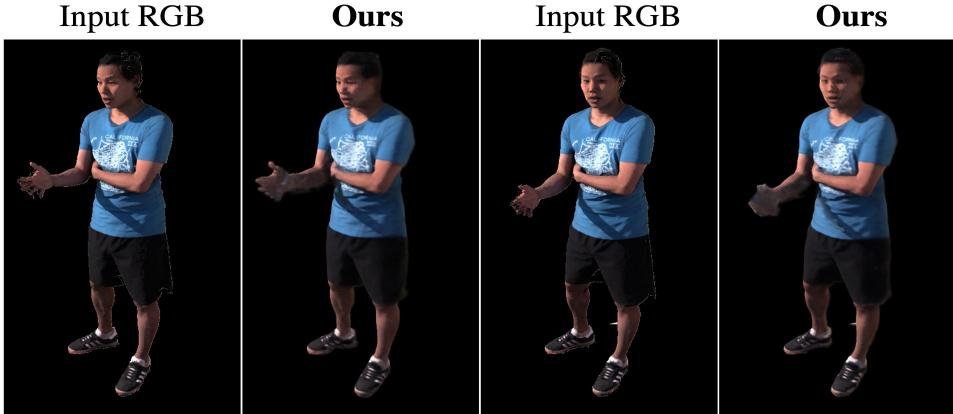


Figure 5: **Failure Cases** on *Sequence5*. The proposed method does not render smooth facial details when the unseen poses and training poses have significantly different expressions.

ment on the fine structures in different parts of the body, which demonstrates that the appearance code anchored to the points clouds can help recover the missing pixels in the query view.

Contribution of the temporal transformer. As shown in Table 2 and Fig. 4, utilizing the temporal fusion module can help the model generate better rendering performance. We observe that the details like logos on the shirt are finer, the hands are cleaner and the face is significantly more crisp.

Effect of the number of key frames. To evaluate the impact of the number of key frames, we report the performance in Table 2. The performance increases with more key frames and saturates with 5 frames.

5.3 Limitations

Our method leverages a pose code anchored to the human pose and an appearance code anchored to the point clouds that can model the shape of the human and can help fill in the missing parts in the body. The temporal transformer module can help recover more non-visible pixels in the body. Without encoding the facial expressions, our method can handle humans without substantial facial expression variations. As shown in Fig. 5, the proposed method predicts blurred expressions on the face of *Sequence5*, in which the query frame has facial expression different from the key frames and combining key frames with the query frame makes the network unable to differentiate the specific facial expression in the query frame. To overcome this, future work will encode facial expression for each frame as a separate code and thus being able to render such diverse facial expressions under new views.

6 Conclusion

In this paper, we build upon recent advances of neural radiance fields pertaining to digital humans and address key challenges that existing human body based methods suffer from, preventing them to generalize to unseen poses. Towards that direction, we propose to integrate a pose code and an

appearance code to synthesize humans in novel views and different poses with high fidelity. The pose code that is anchored to the human pose can help model the human shape whereas the appearance code anchored to the points clouds can help infer the details and recover the missing parts. To leverage temporal information, we further propose to utilize the body motion to track points from the query frame to a few automatically selected key frames and adopt a temporal transformer to aggregate information across multiple frames. The transformer-based fusion module can recover the non-visible part in the query frame. The proposed method achieves significantly better results against the state-of-the-art methods under novel views and poses with quality that has not been observed in prior work. Fine-level information such as fingers, logos, cloth wrinkles and face details are rendered with high fidelity.

References

- [1] Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: ICML (2018) [3](#)
- [2] Ben-Hamu, H., Maron, H., Kezurer, I., Avineri, G., Lipman, Y.: Multi-chart generative surface modeling. TOG (2018) [3](#)
- [3] Chen, J., Zhang, Y., Kang, D., Zhe, X., Bao, L., Jia, X., Lu, H.: Animatable neural radiance fields from monocular rgb videos. arXiv preprint arXiv:2106.13629 (2021) [3](#)
- [4] Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV (2016) [3](#)
- [5] Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., Sullivan, S.: High-quality streamable free-viewpoint video. ACM TOG (2015) [1](#)
- [6] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) [7](#), [9](#)
- [7] Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S.R., Kowdle, A., Escolano, S.O., Rhemann, C., Kim, D., Taylor, J., et al.: Fusion4d: Real-time performance capture of challenging scenes. ACM TOG (2016) [4](#)
- [8] Du, Y., Zhang, Y., Yu, H.X., Tenenbaum, J.B., Wu, J.: Neural radiance flow for 4d view synthesis and video processing. In: ICCV (2021) [3](#)
- [9] Gafni, G., Thies, J., Zollhofer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: CVPR (2021) [3](#)
- [10] Gao, C., Saraf, A., Kopf, J., Huang, J.B.: Dynamic view synthesis from dynamic monocular video. arXiv preprint arXiv:2105.06468 (2021) [3](#)
- [11] Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: CVPR (2018) [3](#)
- [12] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) [6](#)

- [13] Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: ICCV (2021) [3](#)
- [14] Kajiya, J.T., Von Herzen, B.P.: Ray tracing volume densities. SIGGRAPH (1984) [5](#)
- [15] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) [9](#)
- [16] Kwon, Y., Kim, D., Ceylan, D., Fuchs, H.: Neural human performer: Learning generalizable radiance fields for human performance rendering. arXiv preprint arXiv:2109.07448 (2021) [3](#), [4](#)
- [17] Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: CVPR (2021) [3](#), [4](#)
- [18] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. pp. 2117–2125 (2017) [9](#)
- [19] Lior, Y., Yoni, K., Dror, M., Meirav, G., Matan, A., Ronen, B., Yaron, L.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: NeurIPS (2020) [2](#)
- [20] Liu, L., Gu, J., Lin, K.Z., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. In: NeurIPS (2020) [3](#)
- [21] Liu, L., Habermann, M., Rudnev, V., Sarkar, K., Gu, J., Theobalt, C.: Neural actor: Neural free-view synthesis of human actors with pose control. arXiv preprint arXiv:2106.02019 (2021) [10](#)
- [22] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) [1](#), [3](#), [4](#), [5](#), [8](#), [9](#), [10](#)
- [23] Nguyen, P., Sarafianos, N., Lassner, C., Heikkila, J., Tung, T.: Human view synthesis using a single sparse rgb-d input. arXiv preprint arXiv:2112.13889 (2021) [3](#)
- [24] Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Brualla, R.M.: Deformable neural radiance fields. arXiv preprint arXiv:2011.12948 (2020) [3](#)
- [25] Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Q., Zhou, X., Bao, H.: Animatable neural radiance fields for modeling dynamic human bodies. In: ICCV (2021) [2](#), [3](#)
- [26] Peng, S., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Bao, H., Zhou, X.: Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: CVPR (2021) [2](#), [3](#), [5](#), [6](#), [10](#)
- [27] Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: ECCV (2020) [6](#)
- [28] Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: CVPR (2021) [2](#), [3](#)
- [29] Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017) [3](#)

- [30] Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: ICML (2019) 8
- [31] Raj, A., Zollhoefer, M., Simon, T., Saragih, J., Saito, S., Hays, J., Lombardi, S.: Pva: Pixel-aligned volumetric avatars. arXiv preprint arXiv:2101.02697 (2021) 3
- [32] Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: CVPR (2020) 6
- [33] Shysheya, A., Zakharov, E., Aliev, K.A., Bashirov, R., Burkov, E., Iskakov, K., Ivakhnenko, A., Malkov, Y., Pasechnik, I., Ulyanov, D., et al.: Textured neural avatars. In: CVPR (2019) 1
- [34] Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: Deepvoxels: Learning persistent 3d feature embeddings. In: CVPR (2019) 3
- [35] Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: NeurIPS (2019) 3
- [36] Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: ICCV (2021) 3
- [37] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR (2015) 3
- [38] Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors (2018) 6
- [39] Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Interpretable unsupervised learning on 3d point clouds. arXiv preprint arXiv:1712.07262 (2017) 3
- [40] Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: PlenOctrees for real-time rendering of neural radiance fields. In: ICCV (2021) 3
- [41] Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021) 3, 9
- [42] Yu, T., Zheng, Z., Guo, K., Zhao, J., Dai, Q., Li, H., Pons-Moll, G., Liu, Y.: Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In: CVPR (2018) 4



Figure 6: **Qualitative Results.** Our proposed approach achieves significantly better render quality compared to NeuralBody (+Depth) across various poses, viewpoints and subjects. All poses are unseen and have not been used for training.