Supplementary Material

In the supplementary material, we provide the implementation details, network architecture, details of training and test data, the self-designed 1-piece texture map and the modification of shape parameters.

1 Implementation Details

The network of our radiance fields follows the original NeRF and extends it with pose as additional input. We only use the single-level NeRF and sample 64 points along each camera ray. We train the network for 200K iterations and evaluate on test data every 30 frames to save time. After projecting observation-space point $\mathbf{x}$ to NDF-space point $\tilde{\mathbf{u}}$, we normalize $\tilde{\mathbf{u}}$ to between 0 and 1. The hyper-parameter δ_N is set as 8 cm. The first three elements in the pose parameters representing global translation are set as 0 to only learn dynamics related to the pose rather than global translation.

2 Network Architecture

We present our network architecture details in Figure 1.

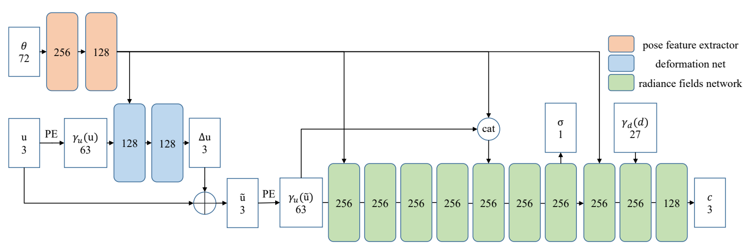


Fig. 1. The network architecture of the proposed method. The number inside each block denotes the vector's dimension. All layers are standard fully-connected layers followed by ReLU activation except for the output layers where we do not use activation to predict deformation $\triangle\mathbf{u}$, density σ, and color c. PE denotes position embedding.

3 Training and Test Data

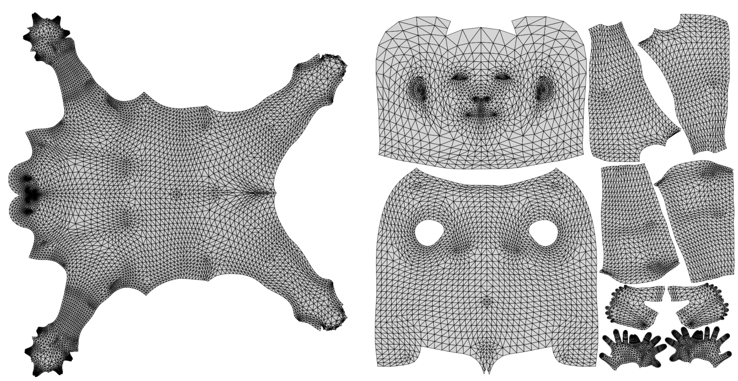
We show the detailed frame numbers for training and testing of each subject from the ZJU-MoCap dataset in Table 1. We adopt the same split strategy with Neural Body and Animatable NeRF for a fair comparison. For the subjects from the DynaCap dataset, we simply select the first 1000 frames for training and the following 1000 frames for testing.

Table 1. Frame numbers for training and test of each subject of the ZJU-MoCap dataset.

	313	315	377	386	387	390	392	393	394
training	60	400	300	300	300	300	300	300	300
test	1000	1000	200	346	354	700	256	258	559

4 Self-designed 1-piece Texture Map

We show the comparison of the self-designed 1-piece texture map and the SMPL's default texture map in Figure 2. The UV region corresponding to the face in the 1-piece texture map is much smaller than in the default texture map.



(a) 1-piece texture map (b) default texture map

Fig. 2. Self-designed 1-piece texture map and the default texture map of SMPL.

5 Avatar Shape Modification

In addition to novel view synthesis and novel pose synthesis, as shown in Figure 3, we can adjust the shape parameters of the SMPL to synthesize animations of the same performer in different shapes. Specifically, at inference time, we adjust the shape parameters of SMPL and calculate mesh vertices. Then these vertices are used to find the closest points for query points and unwrap them to the NDF space. Such a technique will be potentially useful for the movie industry or metaverse, i.e. one can arbitrarily change his shape in the virtual world.

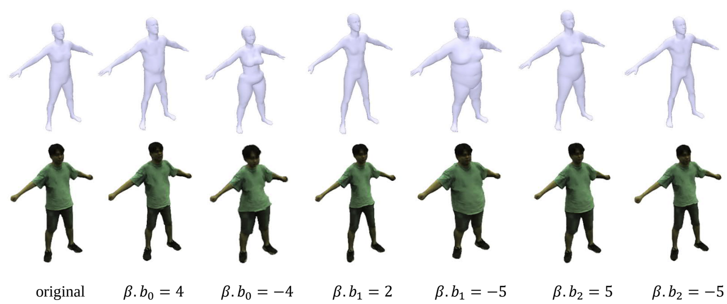


Fig. 3. Rendering results of our method for different body shape configurations of the same performer.