Supplementary Material
for
InstantAvatar: Learning Avatars from Monocular Video in 60 Seconds

Tianjian Jiang[1]*, Xu Chen[1,2]*, Jie Song[1] , Otmar Hilliges[1]
[1] ETH Zürich [2] Max Planck Institute for Intelligent Systems, Tübingen

Abstract

*In this **supplementary document**, we first provide implementation details of our method in Section. 1. We then provide details regarding the evaluation protocol in Section. 2. Finally, we show additional qualitative results in Section. 3. In addition, we invite the reader to watch our **supplementary video** in which we show our training progression and more animation results. Finally, we include the **anonymous manuscript of Fast-SNARF [3]** for reference, which is currently under submission and is used in this paper.*

1. Implementation Details

Network Architecture We implement our model with PyTorch [9] and tiny-cuda-nn [7]. Fig. 1 shows the architecture for the canonical radiance field. We follow a similar architecture as [8] but omit the view-dependent feature. We also replace the activation function of the last layer with rectified linear unit (ReLU), because we empirically notice it leads to better rendering quality. However, the ReLU activation function is known to be vulnerable to the "dying ReLU" problem, where ReLU neurons are no longer updated and only output 0. As a result, the model occasionally gets stuck at undesired local minimums, as can be seen in Fig. 2. To address this issue, we follow the original NeRF paper [6] and add random Gaussian noise with zero mean and unit variance to the output sigma values before feeding them to the ReLU activation function.

Importance Sampling As the human subjects usually take up only a small fraction of the whole image, uniform sampling strategy becomes inefficient. Therefore, we use the human mask to improve sampling efficiency during training. Specifically, we sample pixels inside the masked region, near the mask boundary, and in the rest part of the image with probability 0.8, 0.1 and 0.1 respectively.

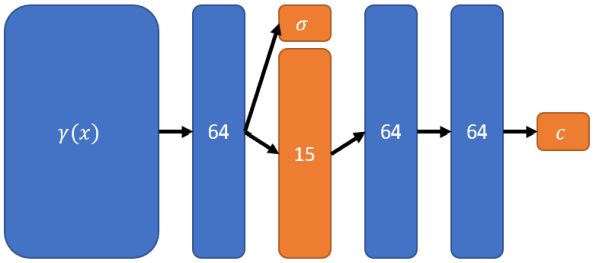


Figure 1. **Architecture of Canonical Radiance Field.** Important variables are marked in orange and components of network are marked in blue. Incoming points are first fed to the multi-resolution hash table γ to generate multi-resolution features. The features are then sent to shallow MLP with 1 hidden layer and 64 neurons to produce the estimated density σ and 15-dim shape features. The shape features are finally passed to another shallow MLP with 2 hidden layer to produce the color prediction c.

Figure 2. **Problem with ReLU Activation Function.** When using ReLU activation function, the model occasionally produces degenerated results if no random noise on sigma is applied.

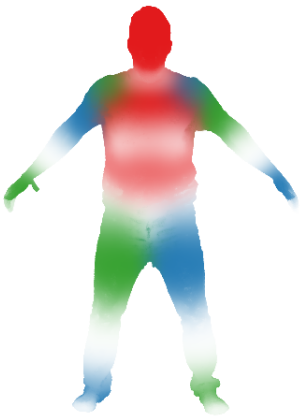


Figure 3. **Skinning Weight Field in Canonical Pose.** We use A-pose as the canonical pose.

Background Augmentation During training, we randomly replace the background with a Gaussian noise image to facilitate the separation of foreground and background.

Choice of Canonical Pose We choose A-pose as the canonical pose because we empirically find that it leads to better animation quality. The canonical pose and the skinning weights are illustrated in Fig. 3.

Losses As described in the main paper, our training objective consists of a huber loss L_{rgb} between the predicted and ground-truth RGB value, a L1 loss L_{alpha} between between the predicted and ground-truth opacity, a hard surface loss L_{hard} and an occupancy-based regularizer L_{reg}:

$$L = L_{rgb} + \lambda_1 L_{alpha} + \lambda_2 L_{hard} + \lambda_3 L_{reg} \quad (1)$$

We use $\lambda_1 = 0.1$, $\lambda_2 = 0.1$ and $\lambda_3 = 10$ in all experiments. Note that the occupancy-based regularization L_{reg} is only applied after 500 training iterations.

Hyperparameters We use the same hyperparameters for all the experiments. For optimizer we use Adam [4], with initial $lr = 10^{-2}$, $\beta = (0.9, 0.99)$ and $\epsilon = 10^{-15}$. During training we exponentially decay the learning rate to 10^{-5}. For the multi-resolution hash table, we use 16 levels with 2 features per level.

2. Evaluation Details

2.1. PeopleSnapshot

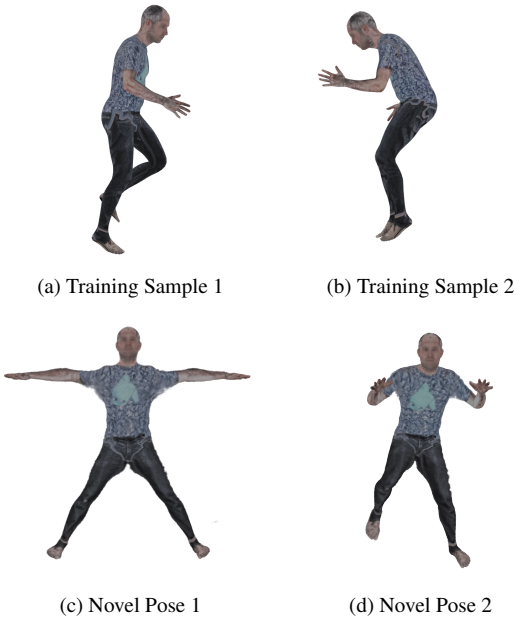
For the PeopleSnapshot dataset, we follow [2] and evaluate our method on 4 sequences. For each sequence, we subsample the images to produce training, validation and test sets. The training set of each sequence contains around 80 to 120 frames.

2.2. SURREAL

For the synthetic setting, we create 3 synthetic sequences with textured SMPL from SURREAL [10] and poses from AMASS [5]. For the training sequence we use the same self-rotating poses as is in PeopleSnapshot, but evaluate our method on challenging motions like kicking. Similar to PeopleSnapshot, we use around 100 images for training, and around 50 images for test.

3. Additional Results

3.1. Training with Challenging Poses



(a) Training Sample 1 (b) Training Sample 2

(c) Novel Pose 1 (d) Novel Pose 2

Figure 4. **Qualitative Results on Synthetic Data with Challenging Training Poses.** Our method can reconstruct avatars also from challenging training poses and generalize to realistic novel poses.

For both PeopleSnapshot and the synthetic data we discussed in the main paper, the training sequence contains self-rotation poses only. To verify our method on more challenging training poses, we create textured meshes using SURREAL [10] and drive them using poses from AMASS [5], such as running, as our training data. The result is shown in Fig. 4. As can be seen, our method is capable of learning high-fidelity avatar even from monocular video of challenging poses.

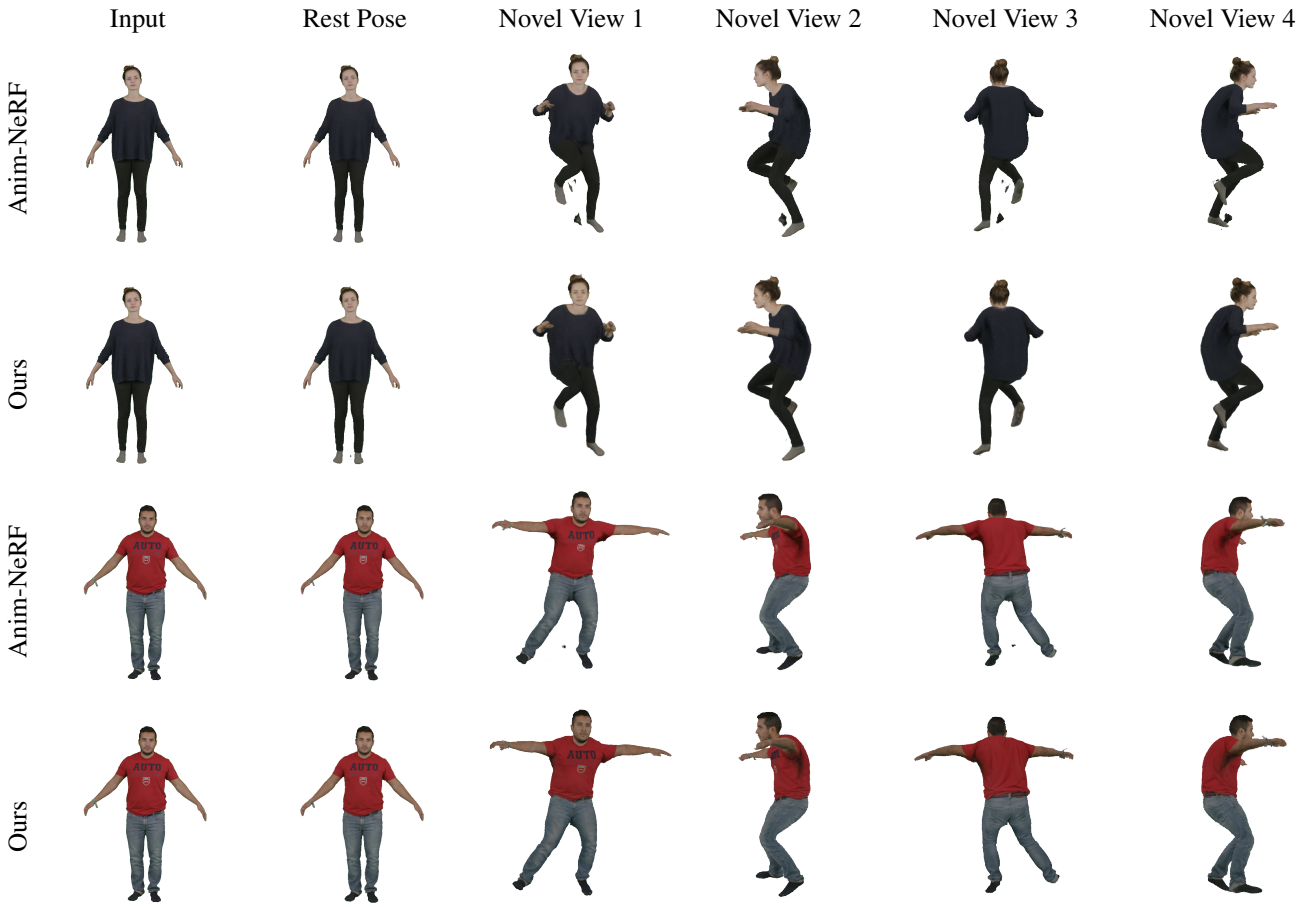


Figure 5. **Additional Qualitative Results on PeopleSnapshot Dataset [1].** We show additional novel views of novel poses on reconstructed avatars of PeopleSnapshot.

3.2. Additional Results

We show additional qualitative comparison and animation results in Fig. 6 and Fig. 7.

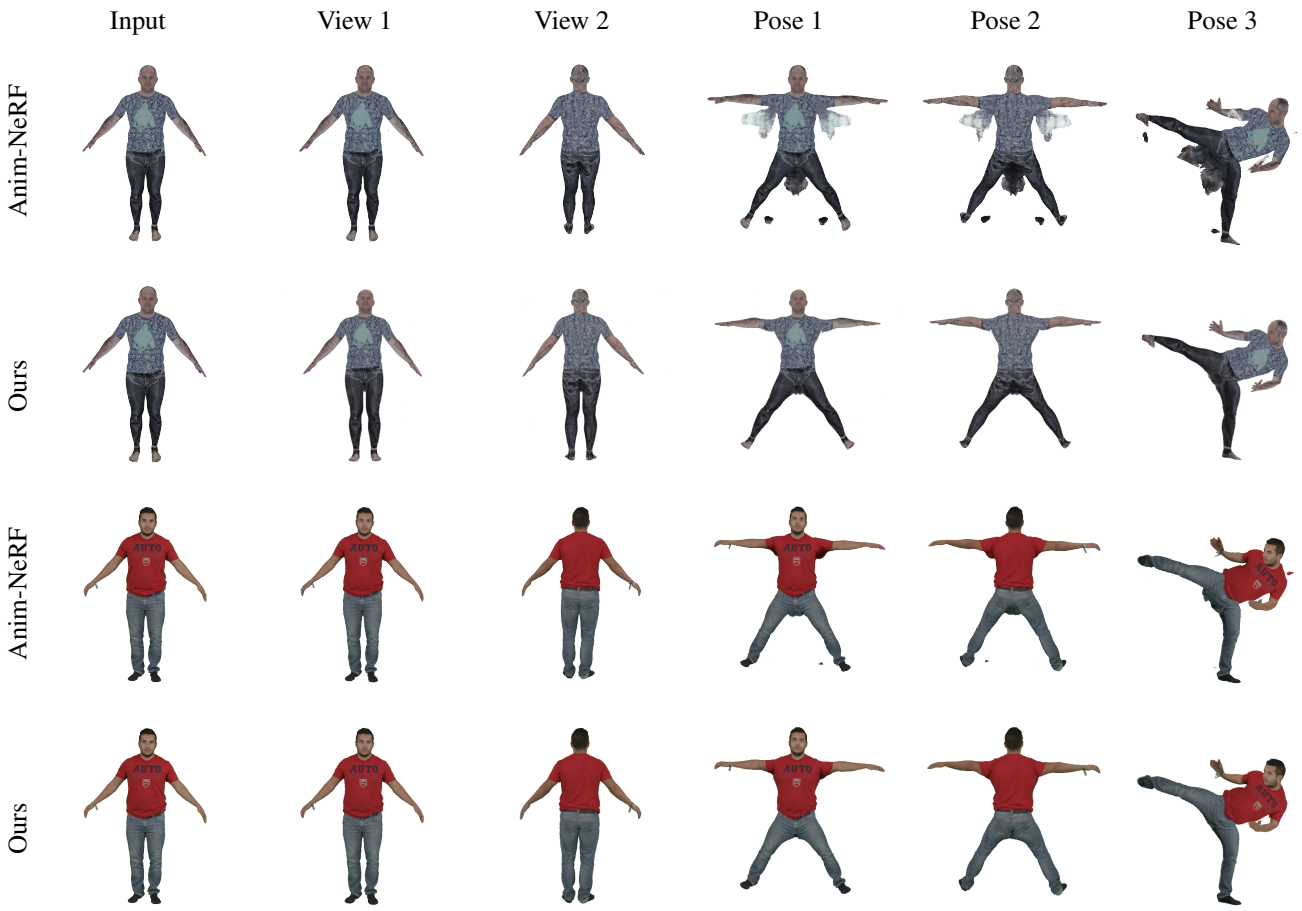


Figure 6. **Additional Qualitative Results on SURREAL [10] and PeopleSnapshot Dataset [1].** We show reconstructed avatars on SURREAL (top) and PeopleSnapshot (bottom) from different viewpoints (column 2-3) and in various poses (column 4-6).

Figure 7. **Additional Animation Results.**

References

[1] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2018. 3, 4

[2] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, Xu Jia, and Huchuan Lu. Animatable neural radiance fields from monocular rgb videos. *arXiv.org*, 2021. 2

[3] Xu Chen, Tianjian Jiang, Jie Song, Max Rietmann, Andreas Geiger, Michael J. Black, and Otmar Hilliges. Fast-SNARF: A fast deformer for articulated neural fields. *arXiv*, abs/2211.15601, 2022. 1

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[5] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019. 2

[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[7] Thomas Müller. tiny-cuda-nn, 4 2021. 1

[8] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 1

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[10] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 2, 4