3DGS-Avatar: Animatable Avatars via Deformable 3D Gaussian Splatting

Supplementary Material

A. Loss Definition

In Sec. 4.4 of the main paper we describe our loss term which can be formulated as follows:

$$\mathcal{L} = \lambda_{l1}\mathcal{L}_{l1} + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{mask}\mathcal{L}_{mask} + \lambda_{skin}\mathcal{L}_{skin} + \lambda_{isopos}\mathcal{L}_{isopos} + \lambda_{isocov}\mathcal{L}_{isocov} \tag{1}$$

We describe how each loss term is defined below:

RGB Loss: We use an $l1$ loss to compute pixel-wise error and a perceptual loss to provide robustness to local misalignments, which is critical for the monocular setup. Following [9], we optimize LPIPS as the perceptual loss with VGG as the backbone. However, unlike NeRF-based methods which train on random ray samples, we render the whole image via rasterization and thus do not require patch sampling. For computational efficiency, we crop the tight enclosing bounding box with the ground truth mask and compute the VGG-based LPIPS as our perceptual loss.

Mask Loss: To boost the convergence of 3D Gaussian positions, we use an explicit mask loss. For each pixel p, we compute the opacity value O_p by summing up the sample weights in the rendering equation Eq. (3) in the main paper , namely:

$$O_p = \sum\nolimits_i \alpha'_i \prod\nolimits_{j=1}^{i-1}(1-\alpha'_j) \tag{2}$$

We thus supervise it with the ground truth foreground mask via an $l1$ loss. Experiments show that the $l1$ loss provides faster convergence than the Binary Cross Entropy (BCE) loss.

Skinning Loss: We leverage SMPL prior by sampling 1024 points $\mathbf{X}_{skin}$ on the surface of the canonical SMPL mesh and regularizing the forward skinning network with corresponding skinning weights $\mathbf{w}$ interpolated with barycentric coordinates.

$$\mathcal{L}_{skin} = \frac{1}{|\mathbf{X}_{skin}|}\sum_{\mathbf{x}_{skin}\in\mathbf{X}_{skin}} ||f_{\theta_r}(\mathbf{x}_{skin}) - \mathbf{w}||^2 \tag{3}$$

As-isometric-as-possible Loss: Please refer to the second paragraph of Sec. 4.4 in the main paper for details.

We set $\lambda_{l1} = 1, \lambda_{perc} = 0.01, \lambda_{mask} = 0.1, \lambda_{isopos} = 1, \lambda_{isocov} = 100$ in all experiments. For λ_{skin}, we set it to 10 for the first $1k$ iterations for fast convergence to a reasonable skinning field, then decreased to 0.1 for soft regularization.

B. Implementation Details

We initialize the canonical 3D Gaussians with $N = 50k$ random samples on the SMPL mesh surface in canonical pose. During optimization, we follow the same strategy from [4] to densify and prune the 3D Gaussians, using the view-space position gradients derived from the transformed Gaussians $\mathcal{G}_o$ in the observation space as the criterion for densification.

We then describe the network architectures of our learned neural components. For the forward skinning network f_{θ_r}, we use an MLP with 4 hidden layers of 128 dimensions which takes $\mathbf{x}_c \in \mathbb{R}^3$ with no positional encoding and outputs a 25-dimension vector. This vector is further propagated through a hierarchical softmax layer that is aware of the tree structure of the human skeleton to obtain the skinning weights $\mathbf{w}$ that sum up to 1. To normalize the coordinates in the canonical space, we proportionally pad the bounding box enclosing the canonical SMPL mesh instead of using the same length in all axes as in [8]. This allows us to use a lower resolution in the flat z-dimension of the human body.

For the non-rigid deformation network $f_{\theta_{nr}}$, the 3D position $\mathbf{x}_d$ is normalized with the aforementioned bounding box and first encoded into representative features with a multi-level hash grid, whose parameters are defined in Tab. 1. The concatenation of the hash grid features and the pose latent code $\mathcal{Z}_p$ then go through a shallow MLP with 3 hidden layers of 128 dimensions to decode pose-dependent local deformation.

The details of our color network structure $\mathcal{F}_{\theta_c}$ are well elaborated in Sec. 4.3 of the main paper. For frames outside the training set, we follow [8] and use the latent code of the last frame in the training sequence.

Parameter	Value
Number of levels	16
Feature dimension per level	2
Hash table size	2^{16}
Coarsest resolution	16
Finest resolution	2048

Table 1. Hash table parameters.

To reduce overfitting, we add noise to the pose and viewing direction input. Specifically, we add a noise drawn from the normal distribution $\mathcal{N}(0, 0.1)$ to the SMPL pose parameters θ with a probability of $p = 0.5$ during training. The viewing direction d is first canonicalized to the canonical space and then augmented with a random rotation derived from uniformly sampled roll, pitch, and yaw degrees $\in [0, 45)$. Adding noise to training signals helps the model to better generalize to novel poses and views.

Our model is trained for a total of $15k$ iterations on the ZJU-MoCap dataset in 30 minutes and $30k$ iterations on PeopleSnapshot in 45 minutes on a single NVIDIA RTX 3090 GPU. We use Adam [5] to optimize our model and the per-frame latent codes with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate of 3D Gaussians is exactly the same as the original implementation from [4]. We set the learning rate for forward skinning network θ_r to 1×10^{-4} and 1×10^{-3} for all the others. An exponential learning rate scheduler is employed to gradually decrease the learning rate by a factor of 0.1 on neural networks. We also apply a weight decay with a weight of 0.05 to the per-frame latent codes.

Following prior works [9, 10], we split the training stage and learn the whole model in a coarse-to-fine manner. In the first $1k$ iterations, we freeze everything except the forward skinning network f_{θ_r} to learn a coarse skinning field with $\mathcal{L}_{skin}$ and prevent the noisy gradients from moving the 3D Gaussians away from the initialization. We then enable optimization on the 3D Gaussians after $1k$ steps. To decouple rigid and non-rigid motion, we start to optimize the non-rigid deformation network $f_{\theta_{nr}}$ after $3k$ iterations. Lastly, we turn on pose correction after $5k$ iterations.

C. Implementation Details for Baselines

In this section, we elaborate on the implementation details of baselines used for comparison to our proposed method, *i.e.* NeuralBody [7], HumanNeRF [9], ARAH [8], Instant-NVR [2], MonoHuman [11] and InstantAvatar [3].

C.1. NeuralBody

For the quantitative evaluation, we use the results of NeuralBody [7] reported in HumanNeRF [9] which follows the same data split.

C.2. HumanNeRF

We use pre-trained models provided by the official code repository[1] for both quantitative and qualitative evaluation.

C.3. ARAH

For the quantitative evaluation, we use the same setup as HumanNeRF (*i.e.* same data split with a reduced image size of 512×512) and train the models using the code from official code repository[2] for 500 epochs. All other hyperparameters remain unchanged. The trained models are then used for qualitative evaluation and out-of-distribution pose animation.

C.4. Instant-NVR

For quantitative and qualitative evaluation, we retrain the models using the code from official code repository[3] on the refined ZJU-MoCap dataset provided by the author. We change the data split to match other baselines while keeping all other hyperparameters the same.

[1] https://github.com/chungyiweng/humannerf

[2] https://github.com/taconite/arah-release

[3] https://github.com/zju3dv/instant-nvr

Table 2. **Additional Ablation Study on ZJU-MoCap [7].** We present the average metrics over 6 sequences.

Metric:	PSNR↑	SSIM↑	LPIPS↓
Full model	**30.61**	**0.9703**	**29.58**
w/o $\mathcal{L}_{mask}$	30.58	**0.9703**	29.90
Random initialization	**30.61**	0.9701	30.90
$7k$ iterations	30.56	0.9698	31.73

C.5. MonoHuman

We note that MonoHuman uses a different data split from HumanNeRF with the last fifth of the training frames being used for novel pose synthesis evaluation instead. For fair comparison we retrain the model from official code repository[4] on the same data split of HumanNeRF with the provided configs for $400k$ iterations and recompute the metrics on novel view synthesis. The trained models are then used for qualitative evaluation and out-of-distribution pose animation.

C.6. InstantAvatar

We follow the original setup and use the provided poses optimized by Anim-NeRF [6] without further pose correction. For quantitative results we copy the metrics from their table, while for qualitative results we train the model from official code repository[5] as they do not release pretrained checkpoints.

D. Ablation Study

We conduct additional ablation study and report the average metrics on ZJU-MoCap in Tab. 2.

D.1. Ablation on Network Components

To showcase the effect of each MLP component in our model on both training efficiency and quality, we additionally ablate respective network-free variants: (1) shallow color MLP $\mathcal{F}_{\theta_c}$ is replaced by spherical harmonics function, (2) no non-rigid deformation $\mathcal{F}_{\theta_{nr}}$, (3) learned skinning field $\mathcal{F}_{\theta_r}$ is replaced by querying the skinning weight of the nearest SMPL vertex. The results are shown in Tab. 3. We surprisingly find that using the SMPL nearest neighbor skinning does not harm the quality while further reducing the training time on ZJU-MoCap dataset. The result is not sensitive to the skinning field possibly due to subsequent compensation of non-rigid deformation. However, we keep to learn the skinning field for its flexibility and generalization to diverse clothing.

	Full	(1)	(2)	(3)	(1)(2)(3)
Time	0:24	0:24	0:20	0:19	0:12
LPIPS	29.58	31.24	32.31	29.54	32.67

Table 3. **Balance between quality and efficiency.** We present the average LPIPS over 6 sequences and the respective training time under each setting.

D.2. Ablation on Color MLP

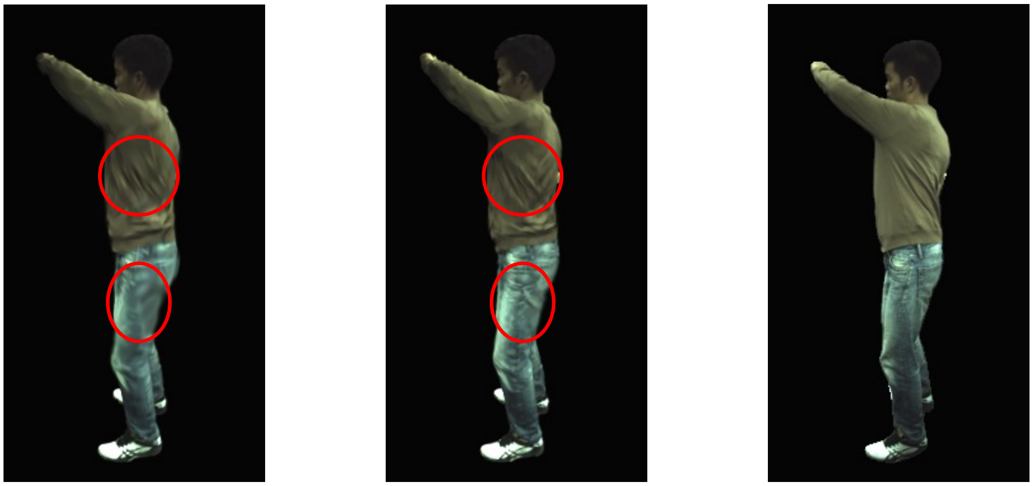
We show in Tab. 5 of the main paper that our proposed color MLP produces rendering with higher quality compared to learning spherical harmonics coefficients. We hereby show qualitative comparison to corroborate this enhancement in Fig. 1. Our proposed color MLP helps generate more realistic cloth wrinkles and sharper textures with pose-dependent feature $\mathbf{z}$ and per-frame latent code $\mathcal{Z}_c$ as additional inputs.

D.3. Ablation on Pose Correction

We additionally show the visualization of pose correction in Fig. 2. Following ARAH and HumanNeRF, we refine the inaccurate SMPL estimation during training, which helps improve the quality of avatar modeling.

[4] https://github.com/Yzmblog/MonoHuman

[5] https://github.com/tijiang13/InstantAvatar



w/o color MLP

Full Model

Ground Truth

Figure 1. **Qualitative Ablation of Color MLP.**

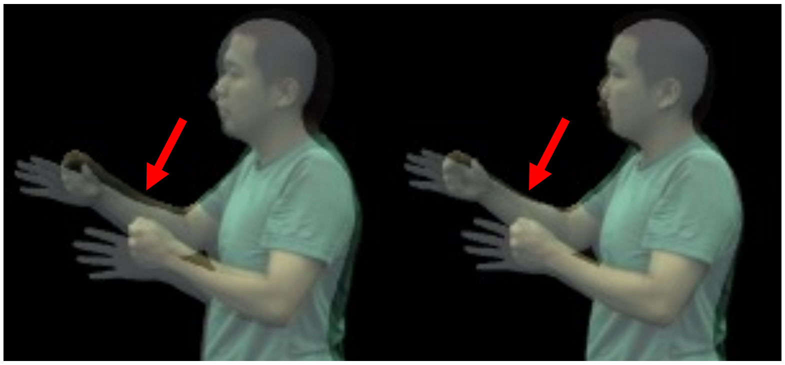


Figure 2. **Qualitative Ablation of Pose Correction.** *left:* before pose correction, *right:* after pose correction. The SMPL mesh aligns better with the ground-truth image after pose optimization.

D.4. Additional Ablation on AIAP Regularization

While forward LBS naturally generalizes to out-of-distribution poses, the pose-dependent non-rigid deformation module can be underconstrained and noisy without proper regularization. To improve generalization, AIAP loss enforces local consistent deformation of Gaussians, thus removing scattered Gaussian artifacts away from the human body. Similar effects can also be observed in novel pose synthesis results on PeopleSnapshot, as shown in Fig. 3. While the AIAP loss shows marginal improvement on novel-view synthesis benchmark, it helps stabilize the Gaussian position and shape on unseen poses.

D.5. Ablation on Mask Supervision

Explicit supervision from ground-truth foreground masks only seems to gain slight improvement, as shown in Tab. 2. However, we observe that the mask loss is useful for removing floating blobs in the empty space. Fig. 4 shows an example for this, without mask loss, the floating Gaussian with the background color could occlude the subject in novel views.

D.6. Ablation on Gaussian Initialization

Instead of initializing the canonical 3D Gaussians from a SMPL mesh surface, we tried to perform random initialization. Specifically, we randomly sample $N = 50k$ points in the enclosing bounding box around the canonical SMPL mesh. Experimental results from Tab. 2 demonstrate that our method could as well converge starting from random initialization, with little

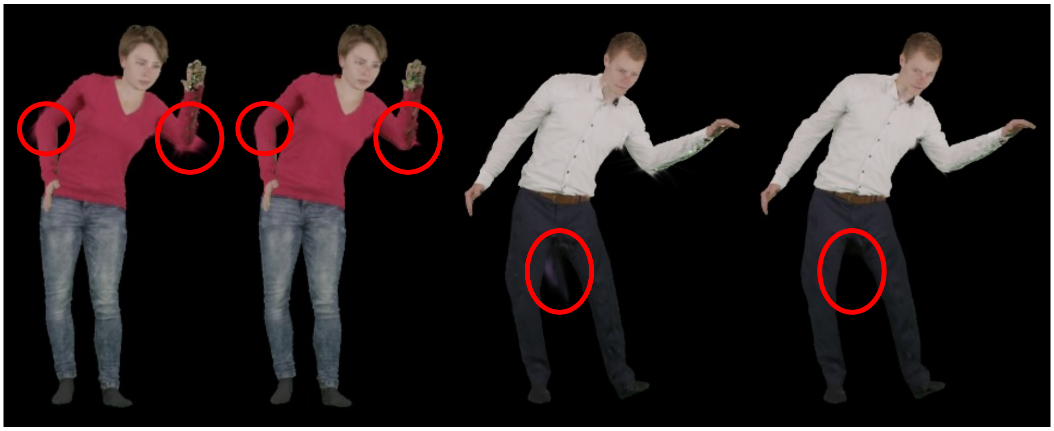
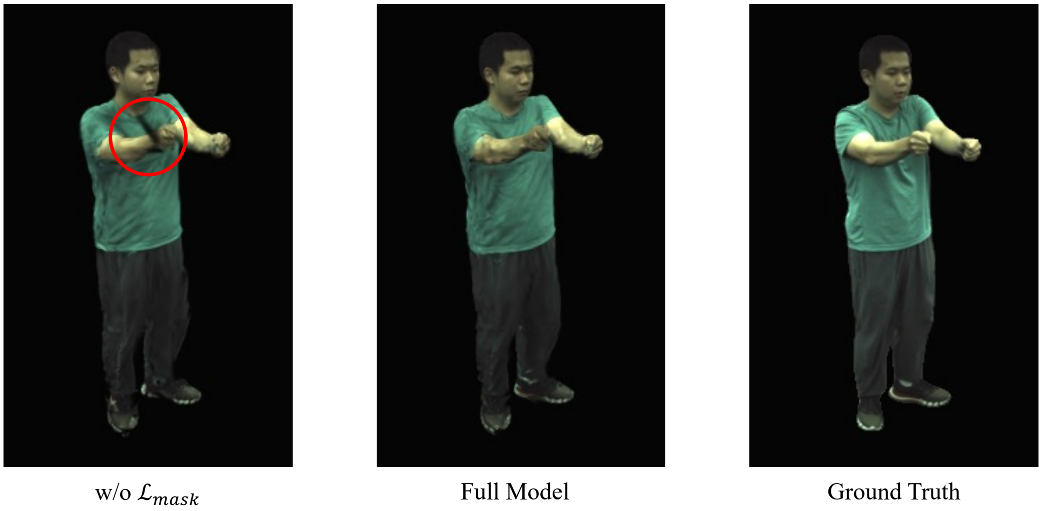


Figure 3. **Qualitative Ablation of AIAP regularization on PeopleSnapshot.** For each subject, *left:* w/o AIAP loss, *right:* w/ AIAP loss. Red circles highlight where Gaussian deformations become noisy without enforcing the AIAP constraint.



w/o $\mathcal{L}_{mask}$ Full Model Ground Truth

Figure 4. **Qualitative Ablation of Mask Loss.**

performance drop compared to the SMPL initialization scheme. Despite this interesting observation, we decide to use SMPL initialization as it is more intuitive and does not incur any overhead.

D.7. Ablation on Training Iterations

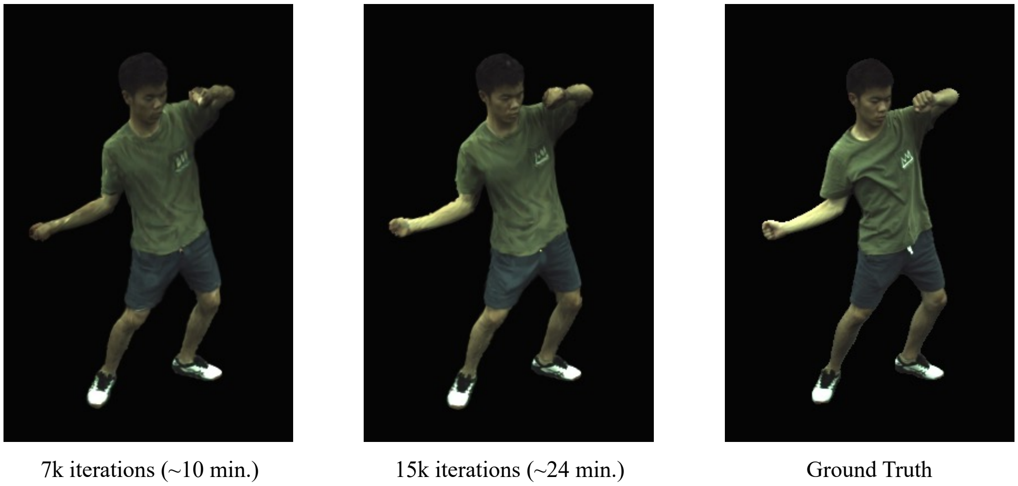
Training for $15k$ iterations on ZJU-MoCap takes precisely around 24 minutes. We further show that our method can already achieve high-quality results at $7k$ iterations in Tab. 2, which takes around 10 minutes, not far away from [3] and [2] that claim instant training within 5 minutes. Qualitative comparison is shown in Fig. 5.

E. Additional Qualitative Results

We show more qualitative results in this section. **For better visualization, we strongly recommend to check our supplementary video.**

E.1. Qualitative Results of Novel View Synthesis on ZJU-MoCap

Additional qualitative comparison of novel view synthesis on ZJU-MoCap is shown in Fig. 6. HumanNeRF and MonoHuman preserves sharp details, but often produces undesired distortions and cloud-like effect around the contour. ARAH gives more


7k iterations (~10 min.)

15k iterations (~24 min.)

Ground Truth

Figure 5. **Qualitative Ablation of Training Iterations.**

rigid body thanks to their explicit modeling of geometry, while they show misalignment and lack fine details. Instant-NVR synthesizes blurry appearance and obvious artifacts on the limbs. Overall, our method can generate high-quality images with realistic cloth deformations.

E.2. Qualitative Results of Out-of-distribution Poses on ZJU-MoCap

We present qualitative comparison of extreme out-of-distribution pose animation in Fig. 7. Our method does not produce obvious artifacts compared to baselines, demonstrating good generalization to unseen poses.

E.3. Qualitative Results on PeopleSnapshot

We show qualitative results on the test set of PeopleSnapshot in Fig. 8. Compared to InstantAvatar, our method produces sharper results, especially in the face region.

F. Limitations

While our proposed approach achieves state-of-the-art rendering quality of clothed human avatars with an interactive frame rate of rendering, the training time of our model still does not match those fast grid-based methods [2, 3]. On the other hand, our method may produce blurry results in areas with high-frequency texture or repetitive patterns, such as striped shirts. Lastly, our method does not provide accurate geometry reconstruction of the avatar, unlike ARAH [8]. Despite reasonable rendering quality, our method generates noisy surface normal resulting from the inconsistency of Gaussian splat depth. It would be particularly interesting to study how to extract a smooth, detailed geometry from the 3DGS avatar model, possibly by applying regularization to the normal map or attaching 3D Gaussians to an underlying mesh.

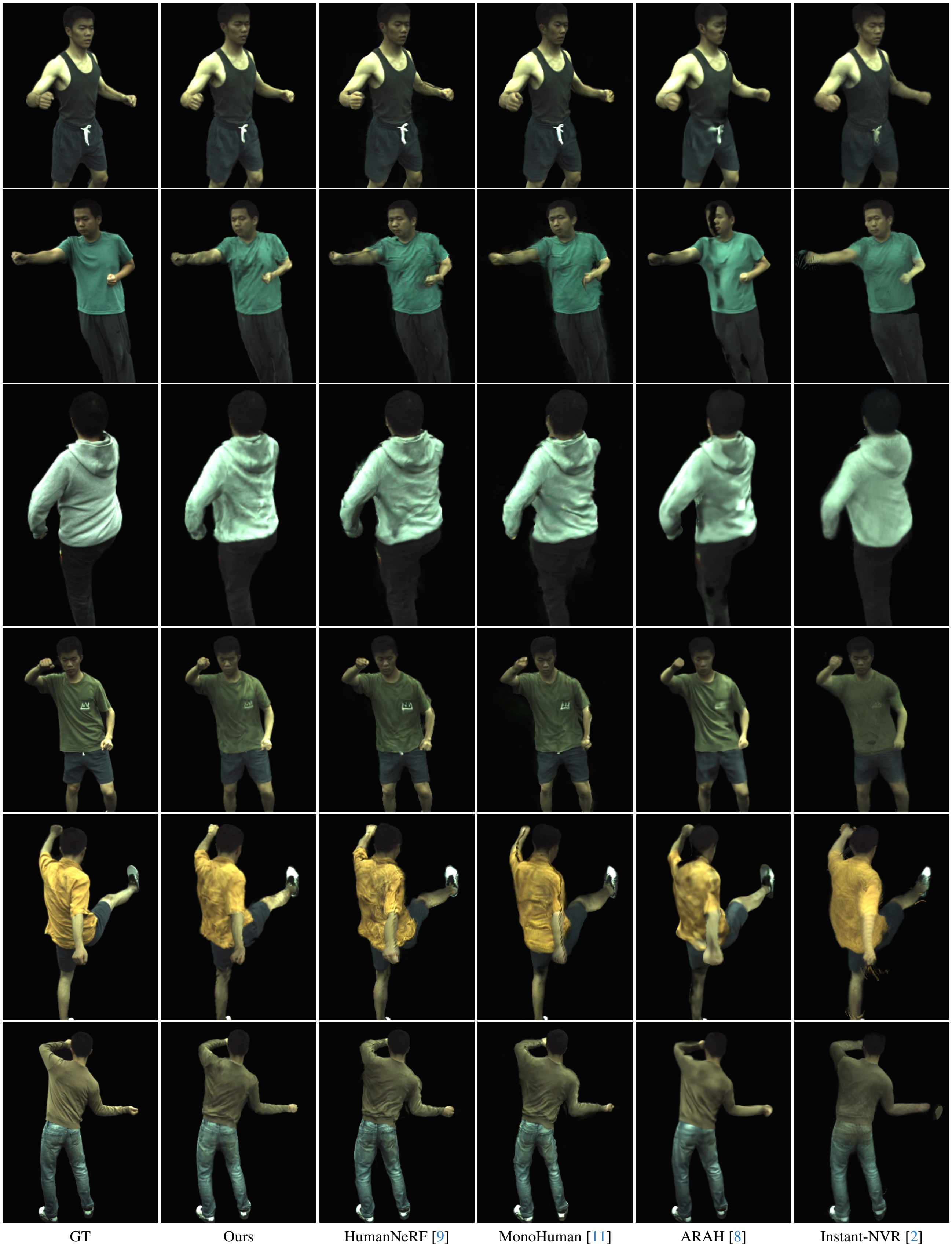
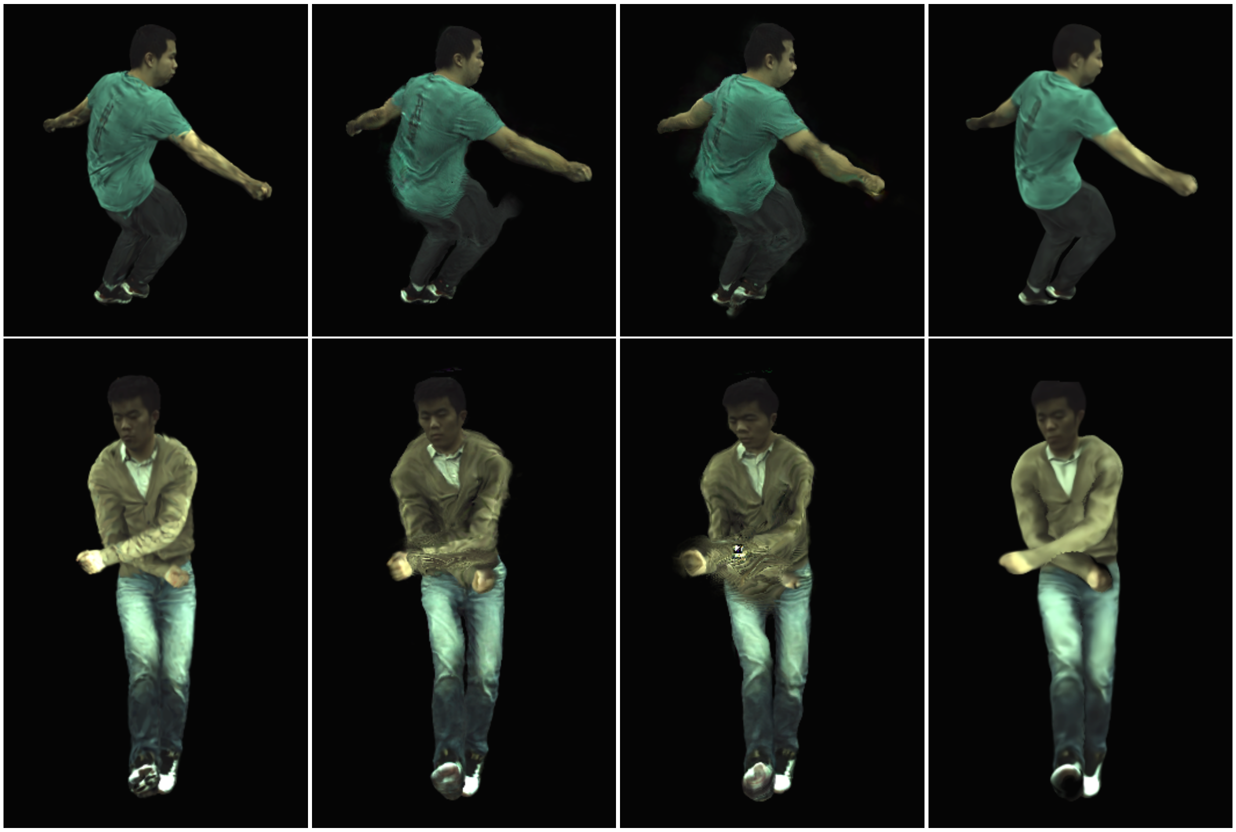
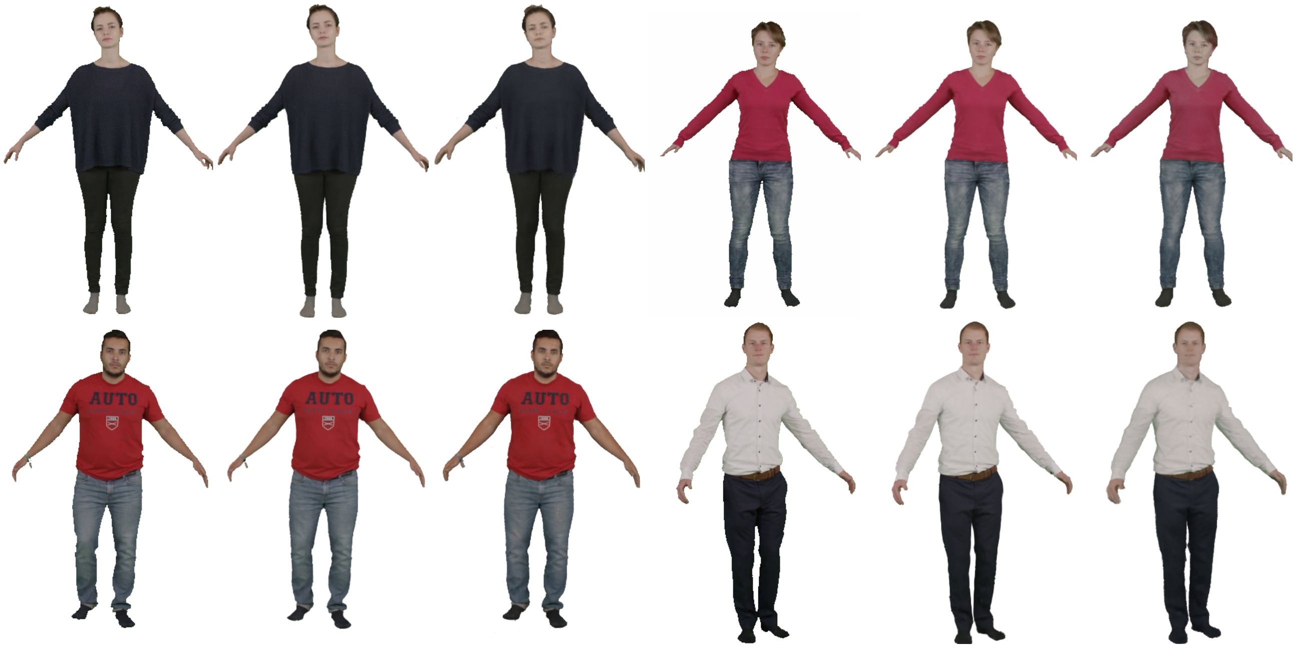


Figure 6. **Qualitative Comparison of Novel View Synthesis on ZJU-MoCap.**



Ours HumanNeRF [9] MonoHuman [11] ARAH [8]

Figure 7. **Qualitative Comparison of Out-of-distribution Pose Animation on ZJU-MoCap.**



GT Ours InstantAvatar [3] GT Ours InstantAvatar [3]

Figure 8. **Qualitative Comparison on PeopleSnapshot [1]. Best viewed zoomed-in.**

References

[1] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *Proc. of CVPR*, 2018. 8

[2] Chen Geng, Sida Peng, Zhen Xu, Hujun Bao, and Xiaowei Zhou. Learning neural volumetric representations of dynamic humans in minutes. In *Proc. of CVPR*, 2023. 2, 5, 6, 7

[3] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proc. of CVPR*, 2023. 2, 5, 6, 8

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1, 2

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015. 2

[6] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proc. of ICCV*, 2021. 3

[7] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proc. of CVPR*, 2021. 2, 3

[8] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *Proc. of ECCV*, 2022. 1, 2, 6, 7, 8

[9] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proc. of CVPR*, 2022. 1, 2, 7, 8

[10] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 2

[11] Zhengming Yu, Wei Cheng, xian Liu, Wayne Wu, and Kwan-Yee Lin. MonoHuman: Animatable human neural field from monocular video. In *Proc. of CVPR*, 2023. 2, 7, 8