

Sequential Gaussian Avatars with Hierarchical Motion Context

Supplementary Material

7. Details of Skeleton Motion ΔP

The pose P is represented as the rotation relative to the parent node of all joints:

$$P = \{\theta_1, \theta_2, \dots, \theta_K\}, \quad (26)$$

where $\theta_j \in \mathbb{R}^3$ describes the j -th joint's relative rotation with its parent in axis-angle form [43]. We follow Dyco [8] to derive each joint's rotation variation $\Delta\theta$ between adjacent frames as its skeleton motion representation. Specifically, the rotation of j -th joint in matrix form can be obtained according to *Rodrigues formula* [43]:

$$\mathbf{R}_j = \text{Rodrigues}(\theta_j), \quad (27)$$

where \mathbf{R}_j is a 3×3 rotation matrix. Given 2 consecutive poses P^t and P^{t-s} , the temporal variation of j -th joint's rotation is computed as a relative transformation matrix $\Delta\mathbf{R}_j^t$:

$$\Delta\mathbf{R}_j^t = \mathbf{R}_j^t \mathbf{R}_j^{t-s-1}. \quad (28)$$

Then we convert the 3×3 matrix $\Delta\mathbf{R}_j^t$ back to the axis-angle form $\Delta\theta_j^t \in \mathbb{R}^3$. Now, the skeleton motion at time t is

$$\Delta P^t = \{\Delta\theta_1^t, \Delta\theta_2^t, \dots, \Delta\theta_K^t\}, \quad (29)$$

which is simplified as $\Delta P^t = \delta(P^t, P^{t-s})$ in Eq. (7).

8. Datasets

I3D-Human Dataset [8]. We use images at a resolution of 512×512 for experiments and follow Dyco[8] for both training and testing splits.

DNA-Rendering Dataset [9]. We use images with a resolution of 512×612 for training and 750×1024 for testing in our experiments. For the temporal split, we select 100 consecutive time steps (pose IDs 0–99) and use the multi-view images of all selected steps for training. During testing, we evaluate on a total of 20 time steps sampled every 5 steps from this training range, using multi-view images from held-out camera views. For the view split, we uniformly sample 24 views from camera IDs 0–48 for training and 6 additional views uniformly sampled from camera IDs 48–60 for novel view evaluation.

ZJU-MoCap [53]. We use images with resolution 512×512 for experiments and follow 3DGS-Avatar[56] for both training and testing splits.

9. Motion Embeddings for Non-Rigid MLP

The embeddings $f_{\Delta P}$ and f_V in Eq. (17) describe the skeleton and local region motions, respectively. For each frame,

all Gaussian primitives $\{\mathcal{G}_i\}$ share the same skeleton motion embedding $f_{\Delta P} \in \mathbb{R}^{32}$ and we concat it with \mathcal{G}_i 's point-wise motion embedding $f_{V_i} \in \mathbb{R}^{96}$ as a new condition $f'_i \in \mathbb{R}^{128}$. f'_i is then input to an MLP $\mathcal{E}_{non-rigid}$ to predict the corresponding non-rigid deformation. $\mathcal{E}_{non-rigid}$ consists of 3 hidden layers and one output layer, and each hidden layer is followed by a ReLU activation.

10. Details of Loss Functions

\mathcal{L}_{color} is the L_1 loss between the rendered image I and the ground truth I_{gt} :

$$\mathcal{L}_{color} = |I - I_{gt}|. \quad (30)$$

We use \mathcal{L}_{ssim} to constrain the structure similarity between the rendered image and the ground truth, which is given by

$$\mathcal{L}_{ssim} = 1 - \text{SSIM}(I, I_{gt}), \quad (31)$$

where $\text{SSIM}(\cdot)$ is the SSIM metric. Additionally, we use the LPIPS loss to ensure the perceptual similarity:

$$\mathcal{L}_{lips} = \text{LPIPS}(I, I_{gt}), \quad (32)$$

where $\text{LPIPS}(\cdot)$ is the LPIPS metric. Following GauHuman [22] and 3DGS-Avatar [56], we employ a mask loss to ensure that Gaussian primitives are accurately localized within their designated regions:

$$\mathcal{L}_{mask} = \|M - M_{gt}\|_2 \quad (33)$$

where M_{gt} is the foreground mask and M is the accumulated α value:

$$M = \sum \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (34)$$

Similar to [54, 56], we also utilize the as-isometric-as-possible constraint [28] to enforce neighboring distance similarity of 3D Gaussian centers and covariance matrices between canonical space and observation space:

$$\mathcal{L}_{isopos} = \sum_{i=1}^N \sum_{j \in \text{KNN}(i)} |d(\mathbf{x}^i, \mathbf{x}^j) - d(\mathbf{x}_o^i, \mathbf{x}_o^j)|, \quad (35)$$

$$\mathcal{L}_{isocov} = \sum_{i=1}^N \sum_{j \in \text{KNN}(i)} |d(\Sigma^i, \Sigma^j) - d(\Sigma_o^i, \Sigma_o^j)|, \quad (36)$$

where N is the number of Gaussian primitives. \mathbf{x} , Σ and \mathbf{x}_o , Σ_o are the Gaussian primitive's position and covariance matrix in canonical space and observation space, respectively. KNN denotes the $K = 5$ nearest points.

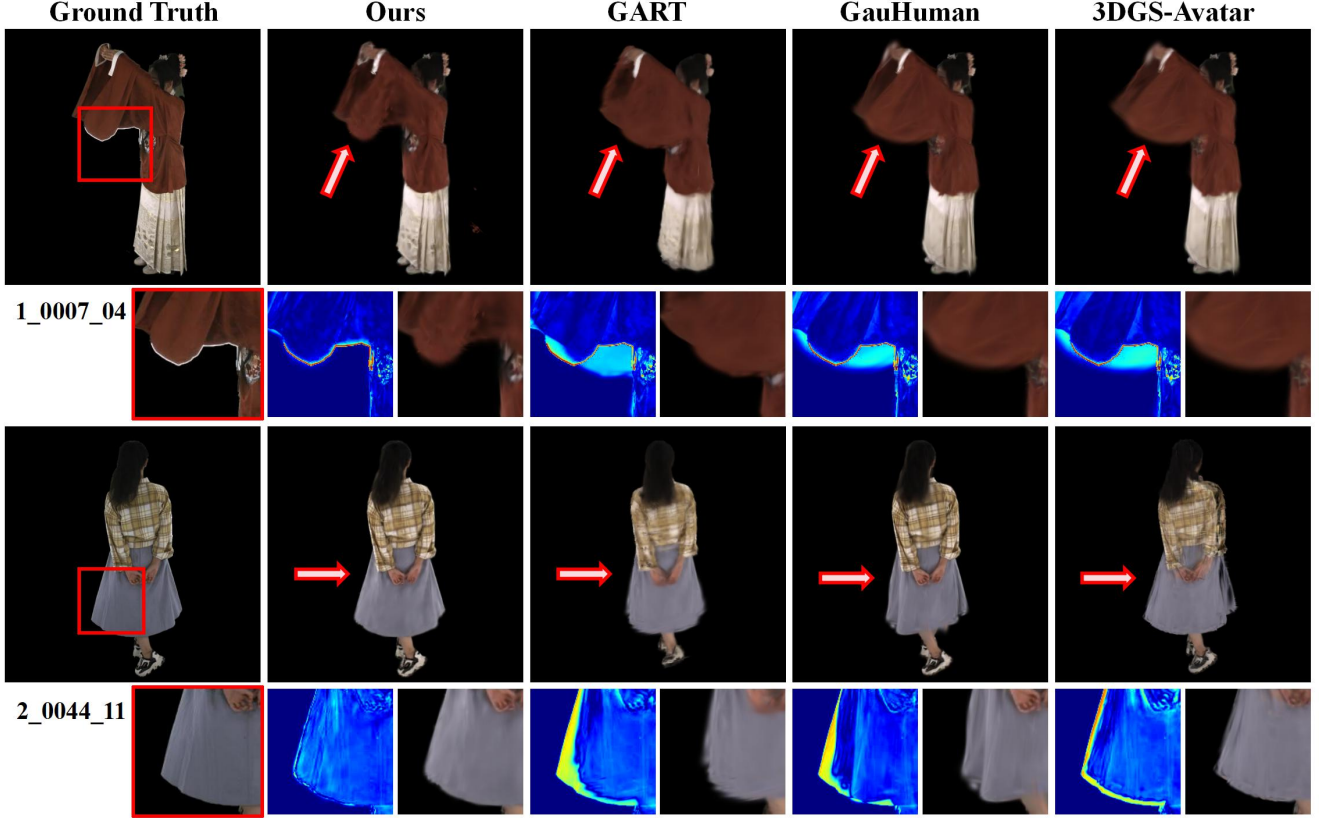


Figure 7. **Novel View Qualitative Results on DNA-Rendering.** We provide both the complete image and localized error map comparisons against other methods for a more comprehensive visualization.

11. Details of Optimization

In our experiments, the number $|\mathcal{S}|$ (in Eq. (14)) of sequence sampled in different scales is set to 3, and each sequence length L (in Eq. (6)) is set to 8. We use the $\tau = 8$ nearest SMPL template vertexes’ velocities to get each Gaussian primitive’s local velocity embedding e_i (in Eq. (12)). Additionally, we utilize the adaptive densification in [27] to control the number of Gaussian primitives.

For the DNA-Rendering dataset, we use the SMPL-X [51] model with pose dimension $P \in \mathbb{R}^{25 \times 3}$ in our experiments. We adopt the SMPL-X template with $N = 10475$ vertexes as Gaussian initialization. We optimize 30k iterations and set the loss weights $\lambda_0 = 1.0, \lambda_1 = 0.01, \lambda_2 = 0.01$. The initial sampling step and the increasing interval are $s_0 = 1$ and $\Delta s = 2$, respectively. The final sample scales are $\mathcal{S} = \{1, 3, 5\}$ (in Eq. (14)). For the I3D-Human dataset, we follow Dyco [8] to use the SMPL [43] model in our experiments for fairness. The number of SMPL template vertexes used for Gaussian initialization is $N = 6890$ and the corresponding pose dimension is $P \in \mathbb{R}^{23 \times 3}$. We optimize 15k iterations and set the loss weights $\lambda_0 = 1.0, \lambda_1 = 0.1, \lambda_2 = 0.1$. We set the increas-

ing interval $\Delta s = 9$. The initial sampling step is $s_0 = 24$ and the final sample scales are $\mathcal{S} = \{24, 33, 42\}$ (in Eq. (14)). Note that the experiment in Tab. 4 (d) adopts single sampling step that $\mathcal{S} = \{s_0\}$.

12. Additional Results

12.1. Experiments on ZJU-MoCap

ZJU-MoCap [53] is a common benchmark dataset in human avatar, which is mainly collected under controlled speeds and tight-fitting garments. We use 6 sequences (377, 386, 387, 392, 393, 394) for experiments following the dataset split in 3DGS-Avaar[56].

Comparison on ZJU-MoCap. Results on DNA-Rendering and I3D-Human demonstrate that our methods achieve better performance in complex cases with loose clothing and uncontrolled speeds. Furthermore, Tab. 5 and Fig. 9 show that our method also achieves competitive metrics in simpler controlled scenarios, which verifies the generalizability of the proposed methods.

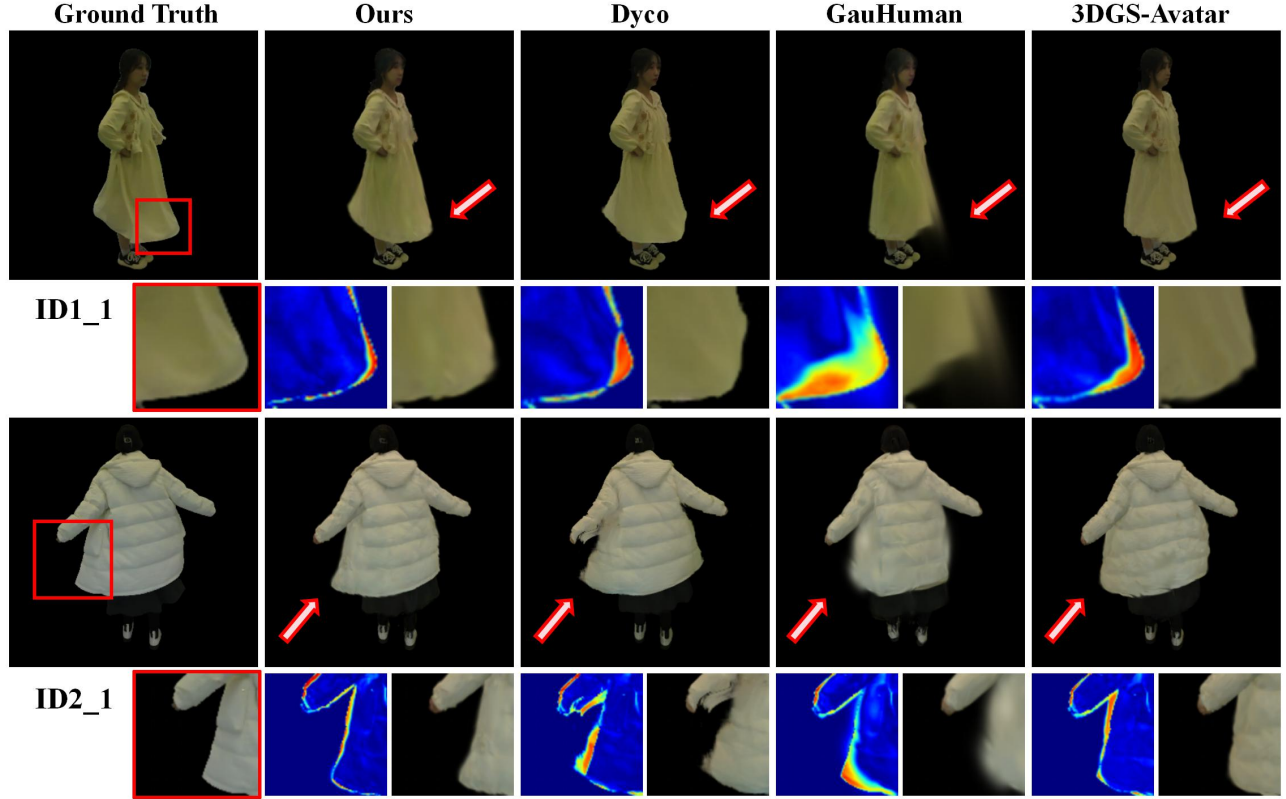


Figure 8. Novel View Qualitative Results on I3D-Human.

Table 5. Quantitative Results on ZJU-MoCap.

Methods	PSNR \uparrow	SSIM \uparrow	LPIPS* \downarrow
Dyco [8]	30.37	0.9599	29.47
3DGS-Avatar [56]	30.59	0.9609	27.09
GauHuman [22]	31.04	0.9620	31.81
Ours	31.02	0.9619	28.89

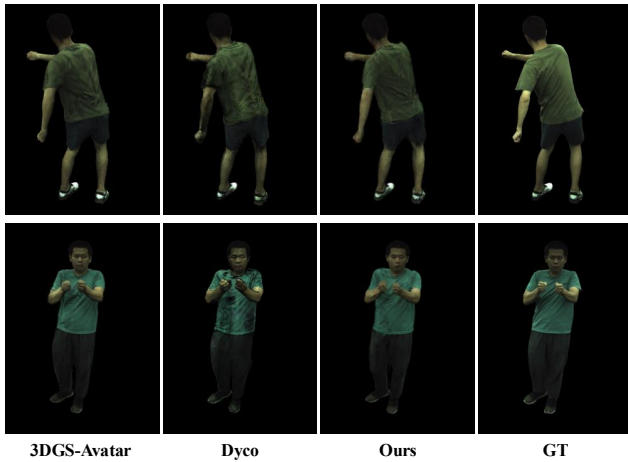


Figure 9. Qualitative Results on ZJU-MoCap.

12.2. Visual Comparisons On DNA-Rendering

Fig. 7 shows additional qualitative comparisons with other SOTA methods on the DNA-Rendering dataset [9]. Our approach excels in regions where motion causes notable appearance variations, producing results that conform to the true shape more accurately, as exemplified by the sleeve (top row) and the skirt hem (bottom row) in Fig. 7.

12.3. Video Quality

We provide rendered videos on the DNA-Rendering dataset [9], along with the corresponding error maps similar to Fig. 3. Please refer to the attached videos for comparison. In the videos, the novel view results are rendered on temporal frames with pose IDs 0–99, using camera view-points that are held out during training.

13. Additional Ablations

Different Sample Steps Tab. 6 shows the quantitative metrics with different sequential sampling steps. The results demonstrate that combining all coarse-to-fine temporal motions as the condition leads to more robust non-rigid deformation and better performance.

Different KNN Sampled Vertexes Tab. 7 presents the quantitative results on I3D-Human dataset [8] with differ-

Table 6. **Impact of Different Sampling Steps on I3D-Human.**

Sampling Step \mathcal{S}	PSNR \uparrow	SSIM \uparrow	LPIPS* \downarrow
$\mathcal{S} = \{24\}$	32.01	0.9655	30.71
$\mathcal{S} = \{33\}$	32.09	0.9656	30.88
$\mathcal{S} = \{42\}$	32.11	0.9658	30.81
$\mathcal{S} = \{24, 33, 42\}$	32.24	0.9664	29.78

ent numbers of KNN vertexes when deriving the velocity e_i of each Gaussian primitive (Eq. 12). It suggests that considering the motion states of all local regions helps obtain a more robust Gaussian primitive’s velocity embedding, contributing to enhancing performance.

Table 7. **Impact of Different KNN Vertexe Number τ .**

KNN num. τ	PSNR \uparrow	SSIM \uparrow	LPIPS* \downarrow
$\tau = 1$	32.14	0.9659	30.41
$\tau = 5$	32.15	0.9660	30.32
$\tau = 8$	32.24	0.9664	29.79
$\tau = 10$	32.20	0.9663	30.11

14. Computational Cost Analysis

Table 8. **Computational Efficiency.**

Mehods	Train Time	FPS	Train Mem.
Dyco	6 h	0.7	20 GB
3DGS-Avatar	25 m	25	6 GB
Ours (<i>3k iter</i>)	5 m	62	8 GB

We provide computational cost analysis for experiments on ZJU-Mocap (conducted on a single 4090 GPU). Tab. 8 shows that our approach achieves competitive computational efficiency. The additional computation time for each frame’s Vertex Motion Template (multi-scale sampling number $|\mathcal{S}| = 3$, sequence length $L = 8$) is about 0.09 s, and it only needs to be computed once during preprocessing before training.