GauHuman: Articulated Gaussian Splatting from Monocular Human Videos — *Supplementary Material*

Shoukang Hu Tao Hu Ziwei Liu
S-Lab, Nanyang Technological University

1. Implementation Details

Implementation Details of GauHuman Our pose refinement module consists of 4 fully connected layers, *i.e.*, an input layer, 2 hidden layers, and one output layer. Each layer is followed by a ReLU activation. The dimension of the hidden layer is 128, while the input and output dimension of the pose refinement module is 69. The LBS offset module adopts 5 fully connected layers with an input layer, 3 hidden layers, and one output layer. The ReLU activation is used after each layer. Positional encoding is applied to the 3D Gaussian positions before they are fed into the input layer. The input and output dimensions of the LBS offset module are 63 and 24 (number of joints). We use Adam optimizer with a learning rate 10^{-5} to optimize the above two modules. We set the threshold of KL divergence as 0.4 to perform split/clone operations. Other training details for 3D Gaussians are the same as [5].

Evaluation Metrics. To quantitatively evaluate the quality of rendered novel view and novel pose images, we report the peak signal-to-noise ratio (PSNR) [15], structural similarity index (SSIM) [17] and Learned Perceptual Image Patch Similarity (LPIPS) [21].

Details of Comparable Methods. 1). Subject-specific optimization-based methods. Neural Body (NB) [12] encodes latent codes in SMPL vertex points and uses them to learn the neural radiance fields. Animatable NeRF (AN) [11] learns a canonical human NeRF through skeleton-driven deformation and learned blend weight fields. AS[13] further extends [11] by learning a signed distance field and a pose-dependent deformation field for residual information and geometric details of dynamic 3D humans. HumanNeRF [18] incorporates a pose refinement module, LBS field, and non-rigid deformation module to optimize a volumetric representation of 3D humans in the canonical space. DVA [14] extends mixtures of volumetric primitives [9] to articulated 3D humans for high-quality telepresence. InstantNVR [1] and InstantAvatar [4] propose to use multi-hashing encoding for fast training of 3D humans. 2). Generalizable methods. PixelNeRF [20] learns a neural network to infer the radiance field based on the input image. Neural Human Performer (NHP) [7] aggregates pixel-aligned features at each time step and temporally-fused features to learn generalizable neural radiance fields. For generalizable methods, we evaluate each subject (*e.g.*, one subject of MonoCap) by first pre-training the model on the other data set (e.g., ZJU_Mocap data set) and then fine-tuning it on the evaluated subject.

Efficient Implementation of KL Divergence The Kullback–Leibler (KL) divergence of two 3D Gaussians is computed as follows:

$$\begin{aligned} KL(G(\boldsymbol{x}_0)|G(\boldsymbol{x}_1)) = \frac{1}{2}(tr(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + \ln\frac{\det\boldsymbol{\Sigma}_1}{\det\boldsymbol{\Sigma}_2} \\ + (\boldsymbol{p}_1-\boldsymbol{p}_0)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{p}_1-\boldsymbol{p}_0) - 3), \end{aligned} \tag{1}$$

where $\boldsymbol{p}_0$, $\boldsymbol{\Sigma}_0$, $\boldsymbol{p}_1$, $\boldsymbol{\Sigma}_1$ are the position and covariance matrix of two 3D Gaussians $G(\boldsymbol{x}_0)$ and $G(\boldsymbol{x}_1)$.

As the covariance matrix is decomposed into the product of rotation and scaling matrices $\boldsymbol{\Sigma} = \boldsymbol{RSS}^T\boldsymbol{R}^T$, we simplify the computation of matrix inverse and determinant operations, *i.e.*,

$$\begin{aligned} \boldsymbol{\Sigma}_1^{-1} &= (\boldsymbol{RSS}^T\boldsymbol{R}^T)^{-1} = \boldsymbol{RS}^{-1}\boldsymbol{S}^{-1}\boldsymbol{R}^T, \\ \det\boldsymbol{\Sigma}_1 &= \det(\boldsymbol{RSS}^T\boldsymbol{R}^T) = \det(\boldsymbol{S}) * \det(\boldsymbol{S}) \end{aligned} \tag{2}$$

Since scaling matrix $\boldsymbol{S}$ is a diagonal matrix, the inverse and determinant of a diagonal matrix can be easily derived by inversing and prodding the diagonal elements respectively. Meanwhile, the inverse of the orthogonal rotation matrix is the transpose of the original matrix. The above simplification saves the computation time for matrix inverse and determinant operation.

2. Details of Loss Functions

Photometric Loss. Given the ground truth target image C and predicted image $\hat{C}$, we apply the photometric loss as follows:

$$\mathcal{L}_{color} = ||\hat{C} - C||_2. \tag{3}$$

Mask Loss. We also leverage the human region masks for Human NeRF optimization. The mask loss is defined as:

$$\mathcal{L}_{mask} = ||\hat{M} - M||_2, \tag{4}$$

where $\hat{M}$ is the accumulated volume density and M is the ground truth binary mask label.

SSIM Loss. We further employ SSIM to ensure the structural similarity between ground truth and synthesized images, *i.e.*,

$$\mathcal{L}_{SSIM} = \text{SSIM}(\hat{C}, C). \quad (5)$$

LPIPS Loss. The perceptual loss LPIPS is also utilized to ensure the quality of rendered image, *i.e.*,

$$\mathcal{L}_{LPIPS} = \text{LPIPS}(\hat{C}, C). \quad (6)$$

In summary, the overall loss function contains four components, *i.e.*,

$$\mathcal{L} = \mathcal{L}_{color} + \lambda_1 \mathcal{L}_{mask} + \lambda_2 \mathcal{L}_{SSIM} + \lambda_3 \mathcal{L}_{LPIPS}, \quad (7)$$

where λ's are loss weights. Empirically, we set $\lambda_1 = 0.5$, $\lambda_2 = \lambda_3 = 0.01$ to ensure the same magnitude for each loss.

3. Rotating Spherical Harmonic coefficients

When transforming 3D Gaussians from canonical space to posed space, the SH coefficients should also be rotated for view-dependent color effects. The above is achieved by first computing a Wigner D-matrix [19] and then rotating SH coefficients with the Wigner D-matrix. In our implementation, we find that rotating SH coefficients has little effect on the final performance[1], so we do not consider it in our work.

4. Further Analysis

Evaluations on novel poses. For each subject in ZJU_MoCap [12] and MonoCap [2, 3, 13], we collect 20 frames for novel pose synthesis by sampling 1 frame every 10 frames. We show the performance comparison of novel pose synthesis as follows. As shown in the table, GauHuman outperforms baselines. Note that PixelNeRF, NeuralBody, and InstantNVR are unsuitable for novel pose synthesis.

Table 1. Quantitative Novel Pose evaluation of our GauHuman and baseline methods on the ZJU_MoCap and MonoCap data sets. LPIPS* = 1000 × LPIPS. **For a fair comparison, we do not conduct test-time optimization of SMPL parameters with images from the test set on InstantAvatar [4].**

Method	ZJU_MoCap			MonoCap		
	PSNR↑	SSIM↑	LPIPS*↓	PSNR↑	SSIM↑	LPIPS*↓
AN	28.64	0.952	47.74	30.67	0.981	19.14
AS	30.42	0.963	37.70	32.78	**0.984**	16.27
HumanNeRF	30.47	0.962	27.31	31.63	0.983	14.18
DVA	29.31	0.955	38.79	32.27	0.982	16.44
InstantAvatar	29.50	0.934	76.37	27.75	0.945	68.20
GauHuman	**31.29**	**0.965**	**29.89**	**33.00**	**0.984**	**13.95**

Scale to in-the-wild datasets. We generalize GauHuman to an in_the_wild monocular online video. We use EasyMocap [16] to predict SMPL pose parameters and SAM [6] to predict human masks for the monocular online video. We sample 1 frame every 5 frames and collect 100 frames for training. To evaluate novel view and novel pose synthesis results, we follow the same setting as ZJU_Mocap and MonoCap. As shown in Fig. 1, GauHuman produces plausible results and surpasses the state-of-the-art InstantAvatar baseline. Note that the pose refinement module can help refine human pose parameters (e.g., foot) for accurate 3D human reconstruction. One example is shown in Fig. 2.

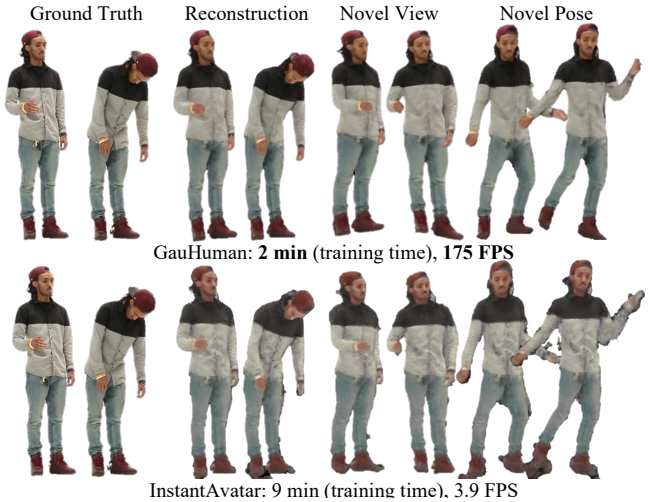

Figure 1. Visualization results produced by our GauHuman and the state-of-the-art InstantAvatar baseline method on an in_the_wild data set. The bottom lines show the training time and rendering speed. Zoom in for the best view.

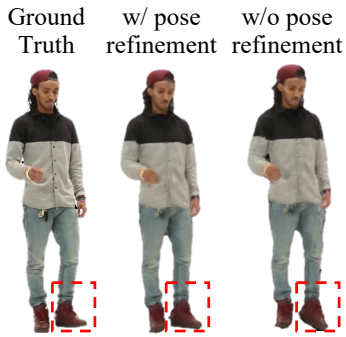

Figure 2. Visualization comparison results produced by our GauHuman w/ and w/o pose refinement module. Zoom in for the best view.

Effects of the number of 3D Gaussians on quality and speed. According to our experiments, increasing the number of 3D Gaussians leads to limited performance improvement, but consumes more computation time. For example, it takes 1 hour for 100k Gaussian points to converge to a 28.01 PSNR; while our GauHuman only needs 55s to achieve a 28.08 PSNR with 13k Gaussians.

Number of 3D Gaussians across different datasets. According to our analysis, the number of 3D Gaussians varies on different datasets, depending on the resolution and accuracy of SMPL and camera parameters. For six subjects

[1]We also find that the implementation of Wigner D-matrix using Pytorch [10] is time-consuming due to the matrix exponential operation.

(512x512p) in Zju_Mocap, the number of 3D Gaussians is around 13k; for two subjects belonging to DeepCap (1024x1024p) in MonoCap dataset, the number of 3D Gaussians is around 16k; for the remaining two subjects belonging to DynaCap (1285×940p) in MonoCap dataset, the number is around 22k.

Using constant KL to perform only splitting and cloning operations. We experiment on sequence 386 of the Zju_Mocap dataset. If only constant KL is used to perform split and clone operations, this will lead to a large number (up to 200k) of 3D Gaussians with a final performance PSNR of 27.88 (vs 28.08 PSNR of our GauHuman). The magnitude of the scaling matrix and gradients of positions are also important metrics for performing split or clone operations.

Convergence speed of initializing the scene with 13k points from SMPL(initial points can share same points from SMPL vertices) without performing splitting, cloning, and pruning operations. We ablate the experiments on sequence 386 of the Zju_Mocap dataset and find that it takes about 6 times more time to converge to a worse performance (PSNR: 27.35) than ours (PSNR: 28.08) when initializing the scene with 13k points without performing splitting, cloning, and pruning operations.

Additional experiments on DNA-Rendering We further evaluate the performance of our GauHuman and two representative baseline methods on a DNA-Rendering data set. We select two sequences (0012_09 and 0025_11 from part 1) from the DNA-Rendering data set and collect 100 frames for training. Similar to ZJU_MoCap and MonoCap, one camera is used for training. For evaluation purposes, we use four nearby camera views as testing views. As shown in Tab. 2 and Fig. 3, AS [13] and InstantAvatar [4] struggle to produce photorealistic renderings due to the complex clothing and fast-moving human actors recorded on the DNA-Rendering data set. In comparison, our GauHuman learns high-quality 3D human performers with fast training and rendering speed, which verifies the flexibility and efficiency of 3D Gaussian Splatting.

Table 2. Quantitative comparison of our GauHuman and baseline methods on the DNA-Rendering data set. LPIPS* = 1000 × LPIPS. Frames per second (FPS) are measured on an RTX 3090.

Method	DNA-Rendering				
	PSNR↑	SSIM↑	LPIPS*↓	Train	FPS
AS [13]	27.67	0.954	50.99	10h	0.14
InstantAvatar [4]	24.77	0.922	78.55	20m	0.48
GauHuman(Ours)	**29.11**	**0.961**	**37.68**	**4m**	**152**

Comparison with concurrent work GART [8] Our concurrent work GART [8] also extends Gaussian Splatting to 3D human modelling with monocular videos. It achieves comparable novel view synthesis performance when compared with state-of-the-art baseline methods on ZJU_MoCap data set while improving the rendering speed to 77 FPS with the efficient 3D Gaussian Splatting technique. We reproduce the result of GART with their released code and show the comparison results in Tab. 3. For a fair comparison, we do not conduct test-time optimization of SMPL parameters

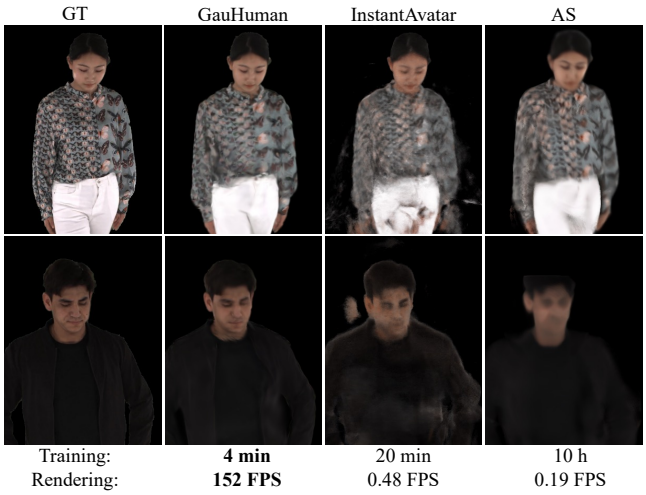

Figure 3. Novel view synthesis results produced by our GauHuman and baseline methods on DNA Rendering data set. The bottom lines show the training time and rendering speed of each method on the DNA Rendering data set. Zoom in for the best view.

Table 3. Quantitative comparison of our GauHuman and GART on the ZJU_MoCap data set. LPIPS* = 1000 × LPIPS. Frames per second (FPS) is measured on an RTX 3090. **For a fair comparison, we do not conduct test-time optimization of SMPL parameters with images from the test set on GART [8].**

Method	ZJU_MoCap (Avg)					
	PSNR↑	SSIM↑	LPIPS*↓	#Gau	Train	FPS
GART	30.91	0.9615	31.83	53.4k	3m	77
GauHuman (Ours)	**31.34**	**0.9647**	**30.51**	**11.8k**	**1m**	**189**
my_377						
GART	31.90	0.9747	**18.8**	55.0k		
GauHuman (Ours)	**32.24**	**0.9757**	18.9	**12.6k**		
my_386						
GART	33.50	0.9669	29.9	51.4k		
GauHuman (Ours)	**33.72**	**0.9693**	**29.0**	**13.1k**		
my_387						
GART	27.74	0.9518	40.3	52.9k		
GauHuman (Ours)	**28.19**	**0.9564**	**39.3**	**9.9k**		
my_392						
GART	31.92	0.9637	32.6	51.6k		
GauHuman (Ours)	**32.27**	**0.9669**	**30.2**	**11.3k**		
my_393						
GART	29.34	0.9540	37.9	51.7k		
GauHuman (Ours)	**30.24**	**0.9584**	**35.2**	**11.0k**		
my_394						
GART	31.08	0.9577	31.5	57.7k		
GauHuman (Ours)	**31.42**	**0.9611**	**30.6**	**12.8k**		

with images from the test set on GART [8]. In comparison with GART, our GauHuman produces slightly better novel view synthesis performance with both faster training (*1m vs. 3m*) and rendering (*189FPS vs. 77FPS*) speed. Specifically, we achieve fast optimization of GauHuman by initializing and pruning 3D Gaussians with 3D human prior, while splitting/cloning via KL divergence guidance, along with a novel merge operation for further speeding up. Notably, without sacrificing rendering quality, GauHuman can fast model the 3D human performer with ∼13k 3D Gaussians.

References

[1] Chen Geng, Sida Peng, Zhen Xu, Hujun Bao, and Xiaowei Zhou. Learning neural volumetric representations of dynamic humans in minutes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8770, 2023. 1

[2] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5052–5063, 2020. 2

[3] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. *ACM Transactions on Graphics (ToG)*, 40(4):1–16, 2021. 2

[4] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16922–16932, 2023. 1, 2, 3

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4): 1–14, 2023. 1

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[7] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. *Advances in Neural Information Processing Systems*, 34:24741–24752, 2021. 1

[8] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. *arXiv preprint arXiv:2311.16099*, 2023. 3, 4

[9] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 1

[10] 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. 2

[11] 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 *ICCV*, 2021. 1

[12] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021. 1, 2

[13] Sida Peng, Shangzhan Zhang, Zhen Xu, Chen Geng, Boyi Jiang, Hujun Bao, and Xiaowei Zhou. Animatable neural implict surfaces for creating avatars from videos. *arXiv preprint arXiv:2203.08133*, 4(5), 2022. 1, 2, 3

[14] Edoardo Remelli, Timur Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabian Prada, Jason Saragih, et al. Drivable volumetric avatars using texel-aligned features. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1

[15] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019. 1

[16] Qing Shuai, Chen Geng, Qi Fang, Sida Peng, Wenhao Shen, Xiaowei Zhou, and Hujun Bao. Novel view synthesis of human interactions from sparse multi-view videos. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2

[17] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 1

[18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16210–16220, 2022. 1

[19] EP Wigner. Gruppentheorie und ihre anwendungen auf die quantenmechanik der atomspektren. friedr. vieweg und sohn akt. *Ges., Braunschweig*, 1931. 2

[20] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 1

[21] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1