

RoGSplat: Learning Robust Generalizable Human Gaussian Splatting from Sparse Multi-View Images — Supplementary Material

1. Implementation Details

Feature extractor and depth refiner. We employ the same network architecture for both the image feature extractor and the depth refiner, with differences only in their input and output channels. The feature extractor processes the input image and generates a 32-channel feature map while preserving the input image resolution. Conversely, the depth refiner takes both the original depth map and the image as input and produces a refined depth map as output. The network architecture is illustrated in Figure 3. Each residual block in the network comprises two 3×3 convolutional layers, each followed by ReLU activation and group normalization.

Sampling voxel-level features. Given a set of target points (estimated SMPL points \mathbf{P} or pixel-wise points \mathbf{P}' in our work), we aim to sample their voxel-level features from a constructed feature volume to capture 3D-aware geometric information. To achieve this, we first construct a feature volume using sparse 3D points (unprojected points derived from refined depth or prior points \mathbf{P}^o in our work). Inspired by [4, 6], we utilize SparseConvNet [2, 5] to diffuse the features of these sparse 3D points. The network architecture is detailed in Table 3. Initially, we compute the 3D bounding box of the sparse points and divide it into small voxels, each measuring $5mm \times 5mm \times 5mm$, resulting in a volumetric representation. SparseConvNet processes this volumetric input using 3D sparse convolutions, diffusing the features of the sparse points into the surrounding 3D space and producing output features. The multi-scale outputs from the 5th, 9th, 13th, and 17th layers of SparseConvNet are resized and concatenated to form the final feature volume. Voxel-level features are then sampled from this output volume using tri-linear interpolation.

Gaussian predictor. As shown in Figure 4, our Gaussian predictor networks take point-related features as input, which include pixel-level image features and point features from the SPD network (or pixel-level depth features). These networks output Gaussian properties. Each predictor head is implemented as a 3-layer MLP, with each layer (except the final one) producing 256-dimensional features.

Offset estimator. The architecture of the offset estimator is illustrated in Figure 5. It takes the voxel-level feature f'_v of pixel-wise points as input and outputs per-point offset. Each layer generates 128-dimensional features with ReLU activations, except for the final layer, which employs Tanh activations.

Num of view	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
1	21.31	0.9123	0.1026
2	23.57	0.9255	0.0857
3	26.32	0.9478	0.0530
4	28.94	0.9615	0.0433
5	30.98	0.9711	0.0341

Table 1. **Ablation study on the number of input views.** We train and test our method given different input views. The performance improves with more available observations.

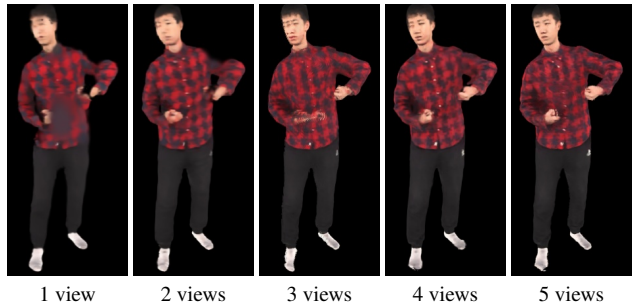


Figure 1. **Ablation study on the number of input views.**



Figure 2. **Ablation study on alternatives for Gaussian position prediction.**

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
position	18.34	0.8915	0.1271
depth	24.25	0.9405	0.0666
Ours	28.94	0.9615	0.0433

Table 2. **Ablation study on alternatives for Gaussian position prediction.**

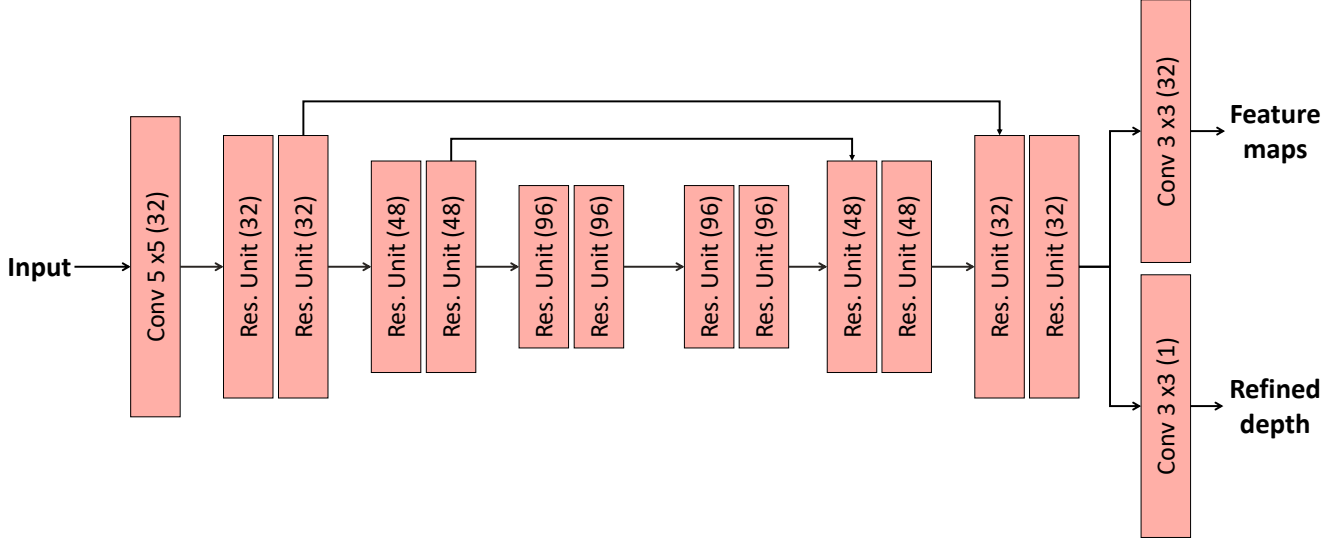


Figure 3. Architecture of feature extractor and depth refiner.

	Layer Description	Output Dim.
	Input volume	$D' \times H' \times W' \times 32$
1-2	$(3 \times 3 \times 3 \text{ conv}, 32 \text{ features, stride } 1) \times 2$	$D' \times H' \times W' \times 32$
3	$(3 \times 3 \times 3 \text{ conv}, 32 \text{ features, stride } 2)$	$D'/2 \times H'/2 \times W'/2 \times 32$
4-5	$(3 \times 3 \times 3 \text{ conv}, 32 \text{ features, stride } 1) \times 2$	$D'/2 \times H'/2 \times W'/2 \times 32$
6	$(3 \times 3 \times 3 \text{ conv}, 32 \text{ features, stride } 2)$	$D'/4 \times H'/4 \times W'/4 \times 64$
7-9	$(3 \times 3 \times 3 \text{ conv}, 64 \text{ features, stride } 1) \times 3$	$D'/4 \times H'/4 \times W'/4 \times 64$
10	$(3 \times 3 \times 3 \text{ conv}, 64 \text{ features, stride } 2)$	$D'/8 \times H'/8 \times W'/8 \times 128$
11-13	$(3 \times 3 \times 3 \text{ conv}, 128 \text{ features, stride } 1) \times 3$	$D'/8 \times H'/8 \times W'/8 \times 128$
14	$(3 \times 3 \times 3 \text{ conv}, 128 \text{ features, stride } 2)$	$D'/16 \times H'/16 \times W'/16 \times 128$
15-17	$(3 \times 3 \times 3 \text{ conv}, 128 \text{ features, stride } 1) \times 3$	$D'/16 \times H'/16 \times W'/16 \times 128$
	Resize & Concat. outputs of layer 5, 9, 13, and 17	$D'/16 \times H'/16 \times W'/16 \times 352$

Table 3. Architecture of SparseConvNet. Each layer consists of sparse convolution, batch normalization and ReLU.

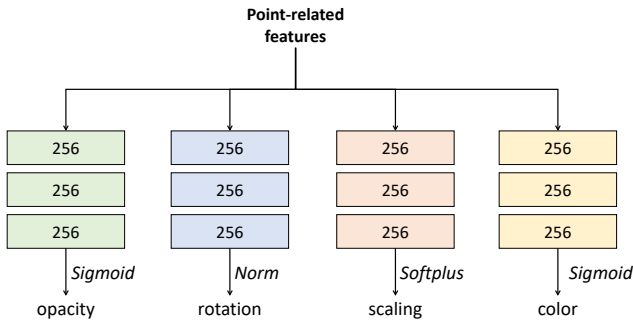


Figure 4. Architecture of Gaussian predictor.

2. Additional Qualitative Results

Figure 6 showcases additional qualitative comparison of in-domain generalization results. Compared to all baseline methods, our method can preserve more reasonable geometry

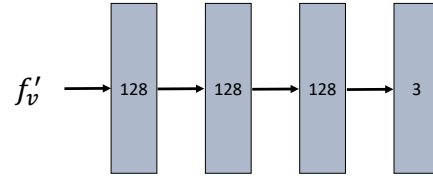


Figure 5. Architecture of offset estimator.

and high-fidelity appearance details. Figure 7 provides additional qualitative comparison of cross-domain generalization, where we show the results on RenderPeople [1, 3] dataset using model trained on THuman2.0 [7] dataset, demonstrating that our method outperforms others on cross-dataset generalization. For further results, please refer to our supplementary video.



Figure 6. More qualitative comparison of in-domain generalization.

3. Additional Ablation Studies

Ablation study on the number of input views. We evaluate our method with varying numbers of input views, as presented in Table 1 and Figure 1. The results indicate that performance improves as the number of input views increases, providing more observations.

Ablation study on alternatives for Gaussian position prediction. To validate the effectiveness of our proposed coarse-to-fine pixel-wise Gaussian prediction method, which leverages refined prior 3D points to regress fine-grained 3D Gaussians, we compare it against two alternative approaches: jointly regressing Gaussian positions or depth maps alongside other Gaussian properties via Gaussian predictor, de-

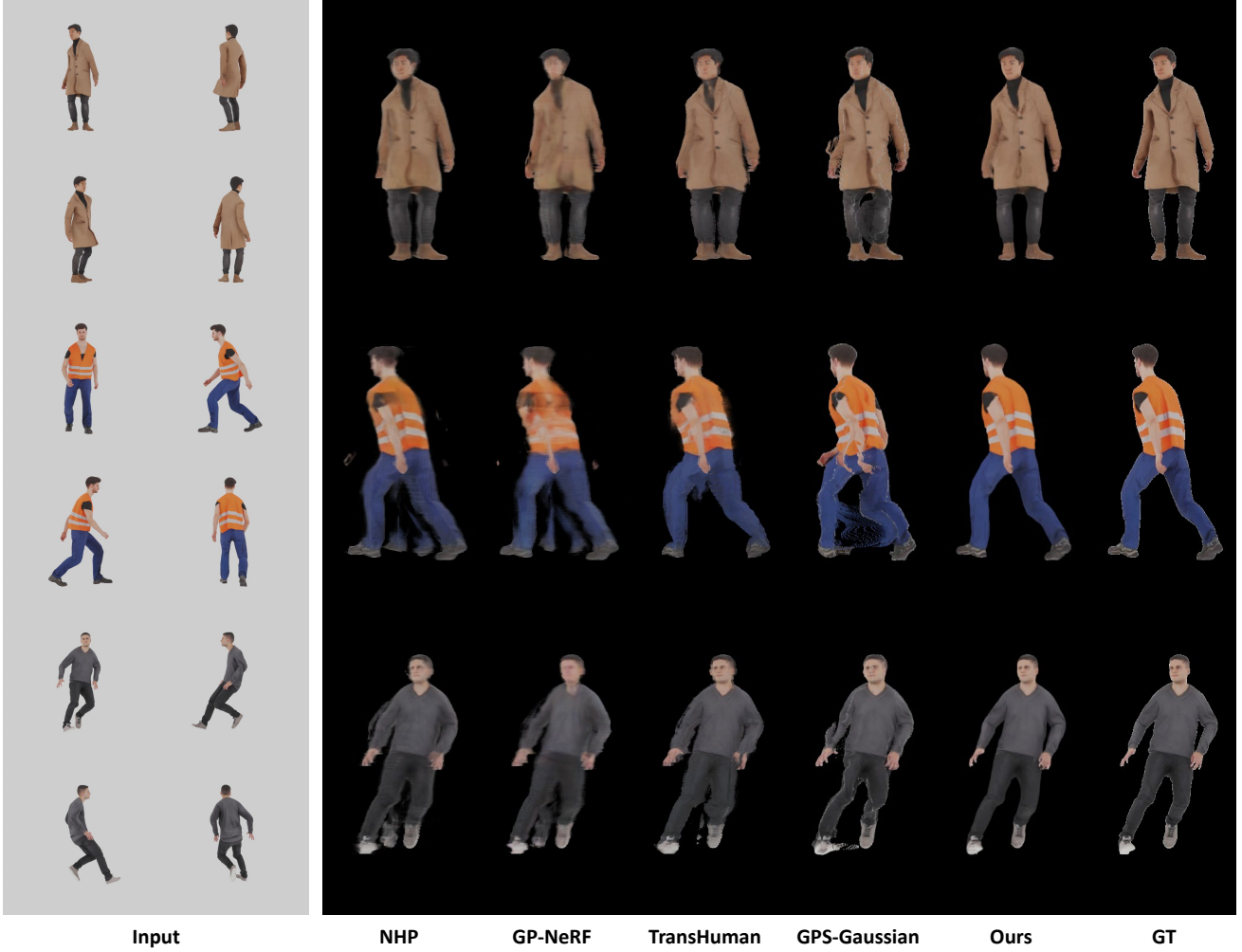


Figure 7. More qualitative comparison of cross-domain generalization.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
w/o depth refiner	27.27	0.9524	0.0602
w/o coarse Gaussian predictor	26.34	0.9471	0.0678
Ours Full	28.94	0.9615	0.0433

Table 4. More geometry-related ablation studies.

noted as “position” and “depth” respectively. To ensure valid numerical results, we clamp the predicted position and depth values to the range $[0, 1]$ and scale these values according to the bounding box of the fitted SMPL model. As shown in Table 2 and Figure 2, our proposed strategy demonstrates superior performance compared to the other two alternatives.

Ablation studies on depth refiner and coarse Gaussian predictor. In table 4, we conduct ablation studies on THuman2.0 dataset to quantitatively evaluate the contribution of our depth refiner and coarse Gaussian prediction, where we

can see that the two geometry-related designs benefit our method to obtain better results.

4. Influence of pose distribution in the training set.

Voxel-level features depend on the pose distribution in the training set, which may adversely affect our generalizability to unseen poses. However, this influence is effectively mitigated by incorporating pixel-level features and performing coarse-to-fine Gaussian prediction, as demonstrated by the results on the RenderPeople dataset in Table 5. As shown, although the RenderPeople dataset contains diverse challenging poses, our method still outperforms the compared methods, manifesting its advantage in generalizability.

Method	PSNR↑	SSIM↑	LPIPS↓
NHP	26.01	0.9384	0.0726
GP-NeRF	25.33	0.9326	0.0792
TransHuman	26.37	0.9451	0.0579
w/o pixel-level features	26.42	0.9522	0.0546
w/o coarse-to-fine Gaussian prediction	26.58	0.9519	0.0594
Ours	27.00	0.9530	0.0519

Table 5. Results on RenderPeople dataset.

5. Ethics Statement

The datasets utilized in our research are sourced from publicly available repositories, including THuman2.0 [7], RenderPeople [1, 3], ZJU-MoCap [6], and one real-world data [8]. Our research centers on the development of a method for free-viewpoint rendering of unseen human avatars, a technology poised to have significant implications in various domains, particularly within virtual environments such as the metaverse and video games. However, it is crucial to recognize the potential misuse and ethical concerns associated with this technology. The ability to manipulate digital representations of individuals, particularly in photo-realistic and indistinguishable ways, raises legitimate concerns regarding privacy, identity theft, and the perpetration of fraudulent activities. In light of these considerations, we advocate for the responsible and ethical use of our research findings.

References

- [1] Renderpeople. <https://renderpeople.com/>. 2, 5
- [2] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 1
- [3] Shoukang Hu, Fangzhou Hong, Liang Pan, Haiyi Mei, Lei Yang, and Ziwei Liu. SHERF: Generalizable human nerf from a single image. *ICCV*, 2023. 2, 5
- [4] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural Image-based Avatars: Generalizable radiance fields for human avatar modeling. In *ICLR*, 2023. 1
- [5] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pinsky. Sparse convolutional neural networks. In *CVPR*, 2015. 1
- [6] 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 *CVPR*, 2021. 1, 5
- [7] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4D: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *CVPR*, 2021. 2, 5
- [8] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. GPS-Gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *CVPR*, 2024. 5