Disentangled Clothed Avatar Generation with Layered Representation

Supplementary Material

This is the supplementary material for *Disentangled Clothed Avatar Generation via Layered Representation*. A video (Sec 1) is included summarizing our method and exhibiting more visualization results. We introduce the implementation details of our method in Sec 2. Additional experimental results and limitation discussions are provided in Sec 3.

1. Supplementary Video

We provide a supplementary video for quick understanding of our method. The video includes:

- A brief introduction of our method;
- Results of unconditional generation and decomposition;
- Results of novel pose animation;
- Results of component transfer.

2. Implementation Details

2.1. Network Architecture

Layered UV Feature Plane and Shared Decoders. The size of layered UV feature plane is $12 \times 128 \times 384$, where we concatenate the three-layer Gaussian-based UV feature plane width-wise instead of channel-wise following [14]. Two shared decoders $\mathcal{D}_g$ and $\mathcal{D}_t$ are utilized to decode the UV feature plane to attribute maps, where attribute (position μ, opacity α, rotation $\mathbf{r}$, scale $\mathbf{s}$, color $\mathbf{c}$) of 3D Gaussians [5] can be extracted via bilinear interpolation. $\mathcal{D}_g$ predict geometry-related attributes (position and opacity) while $\mathcal{D}_t$ outputs texture-related attributes (color, rotation and scale). The architecture of these two shared decoders is shown in Fig. A. $\mathcal{D}_g$ and $\mathcal{D}_t$ both are shallow decoder with two layers. For the first layer, we apply SiLU [2] as the activation function, while for the last layer, Sigmoid is utilized except for the offset $\Delta\mu$ prediction layer. No activation function is utilized for the offset prediction layer. Both the offset prediction layer and covariance prediction layer (predict $\Delta\mathbf{r}$ and $\Delta\mathbf{s}$) are initialized with weight conform to $\mathcal{U}(-1 \times 10^{-5}, +1 \times 10^{-1})$. Biases are initialized to be 0.

Denoising UNet. Following [12], the denoising UNet [13] has UNet architecture with attention modules. The network architecture is shown in Tab. A.

2.2. Training Details

The whole training framework is built upon Pytorch. We utilize Adam [7] as the optimizer. For shared decoder and denoising UNet, the learning rate is 1×10^{-4} and 1×10^{-4} separately. The learning rate for UV feature plane is 0.04. All the UV feature planes are first normalized through a Tanh function and then sent into the shared decoder. The total batch size is 4 scenes per GPU. During one iteration, we select 2 views of each scene for supervision. The whole training process takes 6 days on two RTX 3090 GPUs. The loss weight for each fitting loss is as follows: $\lambda_{\text{color}} = 18$, $\lambda_{\text{mask}} = 9$, $\lambda_{\text{per}} = 0.05$, $\lambda_{\text{seg}} = 9$, $\lambda_{\text{maskin}} = 5$,

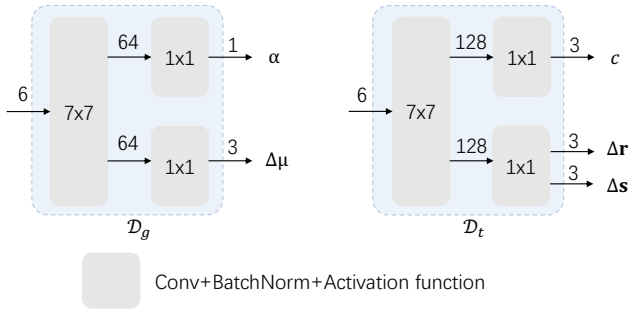
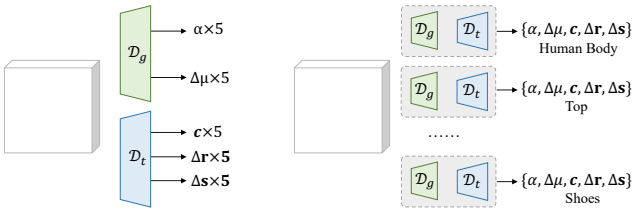


Figure A. Network Architecture of Shared Decoder. The number above the arrow represents the input and output channels of each block. Each block consists of one 2D convolutional layer, one batchnorm layer, and one activation function. The number on the block is the kernel size of the convolutional layer. We utilize SiLU as the activation function and for the last layer, sigmoid is used as the activation function except for the $\Delta\mu$ output.



(a) Single-Layer-Multi-Output (b) Single-Layer-Multi-Network

Figure B. Structure of Single Layer representations. We design two kinds of single-layer representations for ablation study.

Table A. Model Configuration of denoising UNet.

Key	Value
Number of timesteps	1000
Noise configuration	Linear
Input size	$128 \times 384 \times 12$
Base channel	128
Channels configuration	[0.5, 1, 2, 2, 4, 4]
Resblocks per downsample	2
dropout	0
Number of attention heads	4
Attention resolution	[16, 8, 4]

$\lambda_{\text{skin}} = 0.5$, $\lambda_{\text{offset}} = 5$, and $\lambda_{\text{smooth}} = 0.5$. For ablation study, we propose single-layer representations with a size of $128 \times 128 \times 12$, the whole architecture is shown in Fig. B. Other parameters are the same as layered UV feature plane.

2.3. Data Preprocessing

For Tightcap [1], we convert the estimated SMPL [10] parameters to SMPLX [11] to fit our template. Masks for each component (top, bottom, shoes) are provided by the dataset. Since the dataset does not separate hair from body, we combine hair and body components during training. For THuman2.0, THuman2.1 [15] and CustomHuman [3], we first optimize the SMPL-X parameters to make them underneath the clothing layer. We then obtain the segmentation maps for each view through Sapiens [6] estimation and merge the original 20 segmentation labels into 5 (hair, body, top, bottom, and shoes).

3. Limitations and Discussions

3.1. Limitations

(1) Due to the segmentation-map-based supervision and SMPLX-based templates, the performance of our method is affected by the accuracy of the estimated segmentation map and SMPLX parameters. Eliminating inaccurate segmentation results as in [4] and optimizing SMPLX parameters during training can help mitigate the impact.
(2) The collision between body and clothing is a long-existing problem when representing the human body and clothing separately. Introducing post-processing to optimize the position of 3G Gaussians on clothing and body via collision-avoiding loss might eliminate this problem.
(3) Animation of loose clothing is prone to artifacts. Using video data instead of multi-view images or introducing pre-trained networks for cloth deformation might help.
(4) While our method effectively disentangles core components, a promising direction for future research is to extend it to handle general accessories (e.g., glasses, bags) by incorporating additional template layers or adopting other representations.
(5) Animation of loose clothing is prone to artifacts. Using video data instead of multi-view images or introducing pre-trained networks for cloth deformation might help. Introducing physical constraints by converting current representation to meshes [8] or combining UV feature plane with sewing pattern similar to GarmageNet [9] would be interesting future work.

3.2. Additional Experiments

Shape Adaptation. Benefiting from the SMPL-X-based template of our representation, we can adapt clothing to various body shapes as shown in Fig. C which facilitates seamless transfer of components among subjects.

Figure C. Clothing Adaptation to Body Shape

Sensitivity on noisy semantic masks. Our method is robust to noisy masks thanks to multi-view supervision, which can correct errors present in individual views. During training, we filter out samples with obvious noise in over three consistent views. As shown in Fig. D, our method can handle noise that does not persist across multiple views.

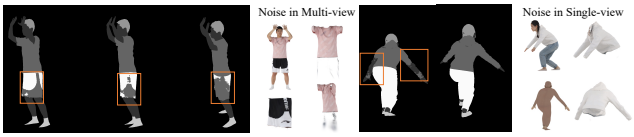


Figure D. Robustness to noisy semantic masks.

Extension to multi-layer outfits. As shown in Fig. E, our method can composite multi-layer clothes to enable more composition.

Figure E. Multi-layer garment composition.

More Generation Results. By training on the composition of three datasets (THuman2.0, THuman2.1, and CustomHuman), our method is capable of generating more diverse avatars and challenging clothing which demonstrates the promise of our methods to achieve better results on larger datasets, as illustrated in Fig. F and Fig. G.

Figure F. More diverse generation results.

Generalization capability. As shown in Fig H, our method can generalize to various clothing types (dress) and poses (hitting) exhibiting distinctive fingers, facial details, and

Figure G. Dress and skirt generation results.

disentanglement capability. We believe that better results can be obtained when extending to a larger training set.

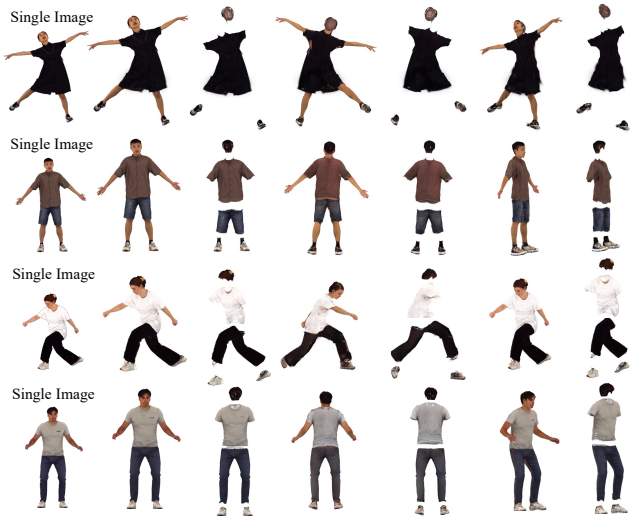


Figure H. Single image reconstruction on unseen 4DDress dataset.

Ablation on TVLoss. Shown in Fig. F, without TVLoss, the inner body color will be affected by outer components.

Figure I. Ablation on feature map resolution and decoder depth.

Ablation on UV feature plane resolution and decoder depth. We conduct a simple experiment to explore the influence of UV feature plane resolution and decoder depth on the results. We reconstruct digital avatars utilizing multi-view images with different resolutions and decoder depths. Demonstrated by Fig. I, the increase of feature map resolution can enhance the final results by a small margin. Increasing decoder depth, however, hinders optimization and reduces results. We consider imbalanced data distribution(over 90% plain texture clothing) and occlusion from limited poses might be the primary constraint of the generation quality for our method. Incorporating video and synthetic data might be helpful to construct a more powerful model.

References

[1] Xin Chen, Anqi Pang, Yang Wei, Wang Peihao, Lan Xu, and Jingyi Yu. Tightcap: 3d human shape capture with clothing tightness field. *ACM Transactions on Graphics (Presented at ACM SIGGRAPH)*, 2021. 2

[2] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. 1

[3] Hsuan-I Ho, Lixin Xue, Jie Song, and Otmar Hilliges. Learning locally editable virtual humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21024–21035, 2023. 2

[4] Zeren Jiang, Chen Guo, Manuel Kaufmann, Tianjian Jiang, Julien Valentin, Otmar Hilliges, and Jie Song. Multiply: Reconstruction of multiple people from monocular video in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[5] 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

[6] Rawal Khirodkar, Timur Bagautdinov, Julieta Martinez, Su Zhaoen, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. *arXiv preprint arXiv:2408.12569*, 2024. 2

[7] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[8] Maria Korosteleva and Sung-Hee Lee. Neuraltailor: Reconstructing sewing pattern structures from 3d point clouds of garments. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 2

[9] Siran Li, Chen Liu, Ruiyang Liu, Zhendong Wang, Gaofeng He, Yong-Lu Li, Xiaogang Jin, and Huamin Wang. Garmagenet: A multimodal generative framework for sewing pattern design and generic garment modeling. *arXiv preprint arXiv:2504.01483*, 2025. 2

[10] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 2

[11] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 1

[14] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4563–4573, 2023. 1

[15] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2021)*, 2021. 2