GALA: Generating Animatable Layered Assets from a Single Scan Supplementary Materials

Taeksoo Kim[1*] Byungjun Kim[1*] Shunsuke Saito[2] Hanbyul Joo[1]

[1]Seoul National University [2]Codec Avatars Lab, Meta

{taeksu98, byungjun.kim, hbjoo}@snu.ac.kr shunsukesaito@meta.com

https://snuvclab.github.io/gala/

A. Implementation Details

A.1. Network Architectures

We implement our networks for predicting SDF and offsets, $\mathbf{\Psi}_h$ and $\mathbf{\Psi}_o$, as a 2-layer MLP network with 32 hidden units and ReLU activations except for the last layer. As inputs, each network takes the 3D Cartesian coordinates of the vertices, X_T, of the designated canonical DMTet grid, (X_T, T). The coordinates are normalized between 0 to 1, and encoded using a hash positional encoding [19] with 16 resolution levels and a maximum resolution of 1024. The networks for predicting vertex colors, $\mathbf{\Gamma}_h$ and $\mathbf{\Gamma}_o$, are implemented using a 1-layer MLP network with 32 hidden units and ReLU activations except for the last layer that uses sigmoid activations. As inputs, each network takes the 3D Cartesian coordinates of the vertices of the canonical human mesh and object mesh, $\mathcal{M}_h^c$ and $\mathcal{M}_o^c$. The coordinates are similarly normalized between 0 to 1, and encoded using a hash positional encoding with 16 resolution levels and a maximum resolution of 2048.

A.2. Optimization Details

The total loss, $\mathcal{L}_{geo}$, for geometry modeling is as follows:

$$\begin{aligned}\mathcal{L}_{geo} = &\lambda_{h_{geo}}^{rec}\mathcal{L}_{h_{geo}}^{rec} + \lambda_{o_{geo}}^{rec}\mathcal{L}_{o_{geo}}^{rec} + \lambda_{comp}^{seg}\mathcal{L}_{comp}^{seg} \\ &+ \lambda_{h_{geo}}^{SDS}\mathcal{L}_{h_{geo}}^{SDS} + \lambda_{o_{geo}}^{SDS}\mathcal{L}_{o_{geo}}^{SDS},\end{aligned} \tag{1}$$

where $\lambda_{h_{geo}}^{rec} = 5 \times 10^3$, $\lambda_{o_{geo}}^{rec} = 5 \times 10^3$, $\lambda_{comp}^{seg} = 1 \times 10^5$, $\lambda_{h_{geo}}^{SDS} = 1$, and $\lambda_{o_{geo}}^{SDS} = 1$. We use AdamW optimizer with a learning rate of 0.001 and optimize for 1600 steps, after 400 steps of the initialization process with $\mathcal{L}_h^{init}$ and $\mathcal{L}_o^{init}$.

The total loss, $\mathcal{L}_{tex}$, for appearance modeling is,

$$\begin{aligned}\mathcal{L}_{tex} = &\lambda_{h_{tex}}^{rec}\mathcal{L}_{h_{tex}}^{rec} + \lambda_{o_{tex}}^{rec}\mathcal{L}_{o_{tex}}^{rec} \\ &+ \lambda_{h_{tex}}^{SDS}\mathcal{L}_{h_{tex}}^{SDS} + \lambda_{o_{tex}}^{SDS}\mathcal{L}_{o_{tex}}^{SDS},\end{aligned} \tag{2}$$

where $\lambda_{h_{tex}}^{rec} = 1 \times 10^8$ and $\lambda_{o_{tex}}^{rec} = 1 \times 10^8$. $\lambda_{h_{tex}}^{SDS} = 0$ and $\lambda_{o_{tex}}^{SDS} = 0$ for the first 400 steps, and $\lambda_{h_{tex}}^{SDS} = 1$ and $\lambda_{o_{tex}}^{SDS} = 1$ otherwise. We use AdamW optimizer with a learning rate of 0.01 and optimize for 2000 steps. Each stage takes about 20 minutes on a single NVIDIA RTX 3090.

A.3. Additional Details

Prompts for the SDS loss. For $\boldsymbol{y}_h$ in $\nabla_{\mathbf{\Psi}_h}\mathcal{L}_{h_{geo}}^{SDS}$ and $\nabla_{\mathbf{\Gamma}_h}\mathcal{L}_{h_{tex}}^{SDS}$, we use "A photo of a man/woman" as the positive prompt and "{*target object*}" as the negative prompt. Note that we use "man" or "woman" based on the gender provided by RenderPeople [21] and CAPE Dataset [17]. For $\boldsymbol{y}_{comp}$ in $\nabla_{\mathbf{\Psi}_o}\mathcal{L}_{o_{geo}}^{SDS}$ and $\nabla_{\mathbf{\Gamma}_o}\mathcal{L}_{o_{tex}}^{SDS}$, we use "A photo of a man/woman wearing {*target object*}" as the positive prompt and do not use any negative prompt. Following DreamFusion [20], we incorporate view directions by concatenating "front/side/back view" to each prompt based on the viewing angle of the sampled camera.

Camera Sampling. We set the camera center using spherical coordinate system, (r, θ, ϕ), where r denotes the radial distance from the origin, θ denotes the elevation, and ϕ denotes the azimuth angle. We set $r = 3$, and sample cameras facing the origin from $\theta \in [-\frac{\pi}{18}, \frac{\pi}{9}]$, and $\phi \in [0, 2\pi]$. We also sample the field of view from $\mathcal{U}(\frac{\pi}{7}, \frac{\pi}{4})$. We additionally use zoomed-in views to capture fine details of human faces and hands and to effectively synthesize the missing regions where human and target object interact. To render zoomed-in images, we translate and scale the input mesh before the rendering process. For the zoomed-in views for faces and hands, we translate the input mesh using the corresponding joint information of the SMPL-X mesh such that each joint locates at the origin, and scale the input mesh by factor of 5 for rendering the face and 10 for rendering the hands. For the zoomed-in views for regions where human and target object interact, we utilize the bounding box information of the target object. Specifically,

*Equal contribution

given the object bounding box $\boldsymbol{x}_l = (x_{min}, y_{min}, z_{min})$ to $\boldsymbol{x}_r = (x_{max}, y_{max}, z_{max})$, we first translate the input mesh by $t \sim \mathcal{U}(\frac{\boldsymbol{x}_r+3\boldsymbol{x}_l}{4}, \frac{3\boldsymbol{x}_r+\boldsymbol{x}_l}{4})$. We then scale the input mesh by the factor of $s \sim \mathcal{U}(\frac{1}{0.6max(\boldsymbol{x}_r-\boldsymbol{x}_l)}, \frac{1}{0.3max(\boldsymbol{x}_r-\boldsymbol{x}_l)})$.

B. Evaluation Details

B.1. Decomposition

Baselines. To the best of our knowledge, there is no existing work that tackles the decomposition of a 3D scan. Hence, we use the recent text-based 3D editing methods as baselines: Instruct-NeRF2NeRF [8] and Vox-E [25]. For evaluation, we use the official implementation for both methods. We train nerfacto model [18] for Instruct-NeRF2NeRF and ReLU field [13] for Vox-E with each scan. Since Instruct-NeRF2NeRF is based on Instruct-Pix2Pix [3], the prompt is given in the form of "instruction"; hence, the basic form of prompts for Instruct-NeRF2NeRF is "Remove {*target object*} from him/her" or "Change his/her {*target object*} to a white t-shirt/shorts" to avoid getting naked body for single-layered clothing. For Vox-E, the basic form of prompts is "A photo of a man/woman without {*target object*}".

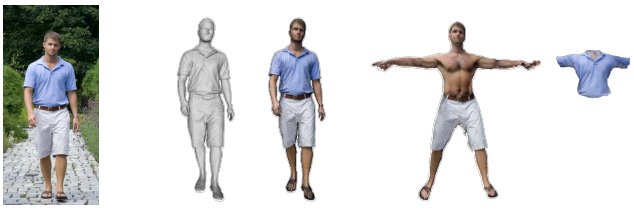
POR metric. We propose a novel metric named pixel-wise object removal score (POR Score) for quantitatively evaluating the decomposition performance. Specifically, we render 30 images per subject using the camera views with equally distributed yaw angles. Then, we run the off-the-shelf open-vocabulary image segmentation method, SAM [14], to get the segmentation of the target object specified by the prompt. Ideally, if the target object is properly decomposed or removed, there should be no pixel classified as the target object for the images rendered after decomposition. Hence, we compute the ratio of the number of pixels classified as the target object in the images after editing and the images rendered from the input scan as follows:

$$POR = \frac{1}{|\mathbf{K}|} \sum_{\boldsymbol{k} \in \mathbf{K}} \frac{\sum_{(i,j) \in \mathbf{M}_{\boldsymbol{k}}^{input}} \mathbb{1}(\text{SAM}(\mathbf{I}_{\boldsymbol{k}}^{edit})_{ij} = 1)}{|\mathbf{M}_{\boldsymbol{k}}^{input}|}, \quad (3)$$

where $\mathbf{K}$ is a set of cameras for rendering, $\mathbf{I}_{\boldsymbol{k}}^{input}$ and $\mathbf{I}_{\boldsymbol{k}}^{edit}$ are images rendered from the input mesh and the edited result, and $\mathbf{M}_{\boldsymbol{k}}^{input}$ is a segmentation mask of the $\mathbf{I}_{\boldsymbol{k}}^{input}$ which is defined as $\mathbf{M}_{\boldsymbol{k}}^{input} = \{(i,j) | \text{SAM}(\mathbf{I}_{\boldsymbol{k}}^{input})_{ij} = 1\}$.

B.2. Canonicalization

Baselines. For Fast-SNARF [7], we use the official implementation with the default hyperparameters except for the skinning mode where we use the "preset" mode which uses the nearest neighbor skinning weights, instead of the original "mlp" mode which learns the skinning weights. This is due to the training instability with limited training data as mentioned in the main paper.



(a) Image (b) 3D reconstruction (c) Decomposition

Figure 1. **Decomposing single-view 3D reconstructions.** Our method enables the generation of animatable layered assets from 2D images via 2D-to-3D reconstruction methods [1].

(a) 3D avatar (b) Decomposition

Figure 2. **Decomposing diffusion-generated 3D assets.** Our method enables the generation of animatable layered assets from texts via text-to-3D generation methods [16]. We show the result of applying GALA on the avatar generated with the prompt, "Vincent Van Gogh".

Ablation. In our ablation study, we utilize the CAPE dataset [17]. Since the dataset doesn't provide texture data, we employ an off-the-shelf mesh texturing tool [22] to add color information to the input mesh and perform segmentation, which we find challenging to perform on the rendered geometry or normals.

C. Additional Qualitative Results

In this section, we present additional qualitative results of our method. Please refer to the supplementary video for animated results.

Decomposing User-generated 3D Assets. GALA can decompose user-generated 3D assets from single-view 3D reconstruction methods [1, 2, 11, 23, 24, 27, 28] or 3D avatar generation methods [4, 10, 15, 16, 29]. Fig. 1 shows the decomposition result of the 3D human mesh reconstructed from a 2D image with Human-SGD [1] and Fig. 2 shows the decomposition result of the 3D avatar generated from text with TADA [16]. These results demonstrate that GALA enables the intuitive scenario for the users to create their own reusable 3D assets from their images or text guidance.

Comparison on Canonicalization. We compare canonicalization results with baseline methods in Fig. 3.

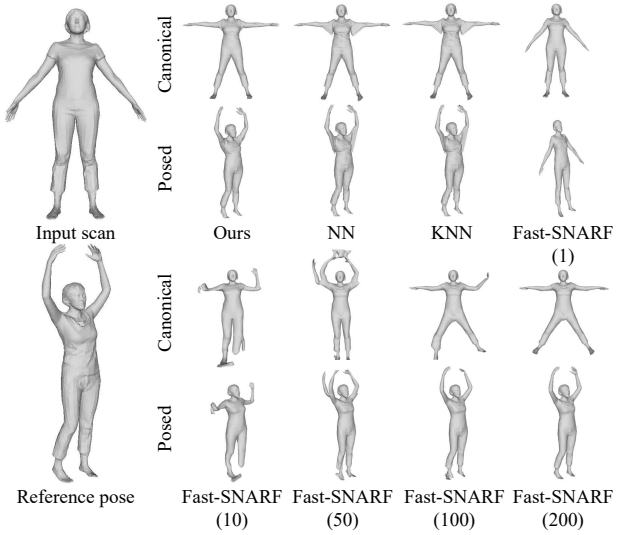


Figure 3. **Qualitative comparison on canonicalization.** We present the results of single-scan canonicalization in the top two rows. The bottom two rows depict the results of Fast-SNARF [7], with varying numbers of training scans denoted in the parenthesis.

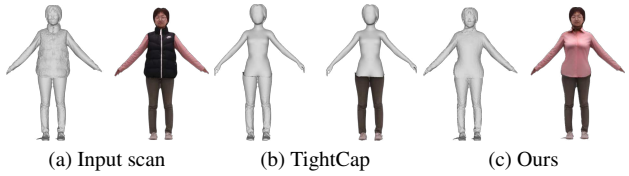


Figure 4. **Qualitative comparison with TightCap [6].** GALA allows higher-quality decomposition compared to TightCap.

Comparison on Decomposition with TightCap [6]. TightCap [6] enables the decomposition of unclothed humans and clothing by leveraging paired 3D data of clothed and unclothed human scans. We demonstrate a qualitative comparison between GALA and TightCap in Fig. 4. As shown, GALA produces higher-quality decomposition along with texture and allows the separate decomposition of multi-layered garments. Notably, GALA doesn't require any 3D training data whereas TightCap relies on hundreds of paired clothed and unclothed 3D human scans.

Decomposition and Canonicalization. Fig. 12 is an extended figure of Fig. 5 in the main paper, which shows the results of decomposition and canonicalization of input scans.

Layered Decomposition. Fig. 5 is an extended figure of Fig. 1 in the main paper, which shows the strength of our method to generate "layered" assets by applying series of decomposition to the input scan. By composing back the decomposed assets, our method enables the decomposition of specific layers of clothing.

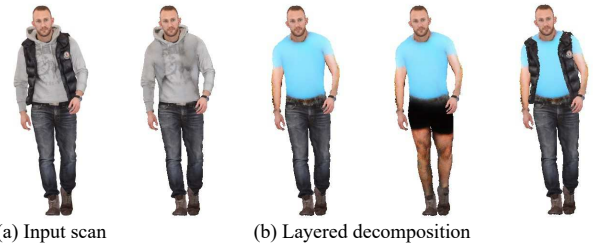
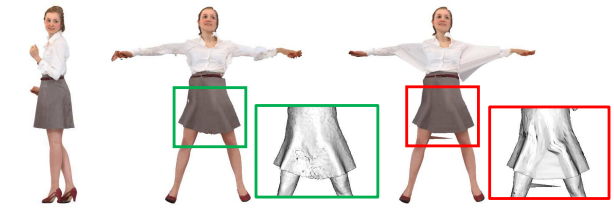


Figure 5. **Layered decomposition.** Our method enables the layered decomposition of the input scan. Note that we can remove the specific layer of clothing by recomposing the decomposed assets.

Figure 6. **Composition.** Our method enables creation of newly-dressed avatars which are fully animatable, by combining various combinations of decomposed assets.

Composition. Fig. 6 is an extended figure of Fig. 1 in the main paper, depicting the ability of our method for 3D garment transfer and reposing.

Loose Clothing. Fig. 7 is an extended figure of Fig. 7 in the main paper, which shows the advantage of our method for modeling canonical shapes of loose clothing compared to simple canonicalization methods [9, 12].

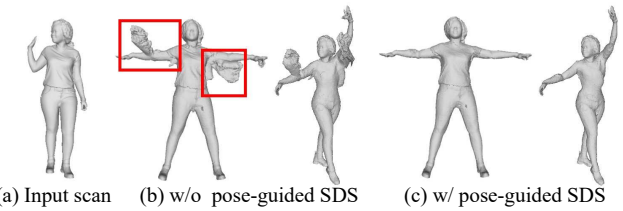


(a) Input scan (b) Ours (c) NN

Figure 7. **Loose clothing.** Our method successfully models canonical shapes of loose clothing.

(a) $\beta_0 = -2$ (b) $\beta_0 = -1$ (c) $\beta_0 = 0$ (d) $\beta_0 = 1$ (e) $\beta_0 = 2$

Figure 8. **Size changes of decomposed assets.** Our method enables effortless size changes of decomposed assets by switching the SMPL-X shape parameters.



(a) Input scan (b) w/o pose-guided SDS (c) w/ pose-guided SDS

Figure 9. **Canonicalization via pose-guided SDS loss.** Applying our pose-guided SDS loss in the canonical space enables robust canonicalization from a single scan.

Size Changes. Fig. 8 shows the ability of our method to efficiently change the shapes of decomposed assets by altering the SMPL-X shape parameters.

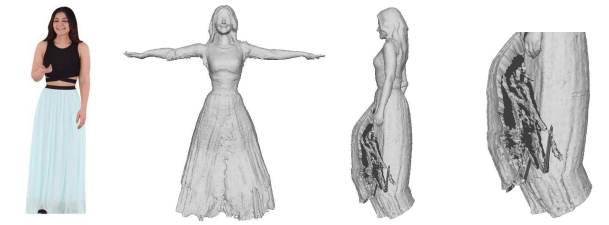
Pose-guided SDS Loss. Fig. 9 is an extended figure of Fig. 10 in the main paper. Our pose-guided SDS loss applied in the canonical space effectively removes artifacts in the canonical shape and enables correct canonicalization from a single scan.

D. Discussion

SDS loss to Composition Mesh. In order to complete geometry and appearance of the object, we apply our pose-guided SDS loss to the composite mesh of human and object instead of the object mesh itself. This is due the fact that OpenPose [5] ControlNet [30] is trained to generate pose-guided human images. Hence, when given the positive prompt "{*target object*}", and the negative prompt, "a person", it fails to exclusively generate the object without humans as shown in Fig. 13. We also present a quantitative comparison on canonicalization between applying SDS loss to the composite mesh and to the object mesh in Tab. 1.

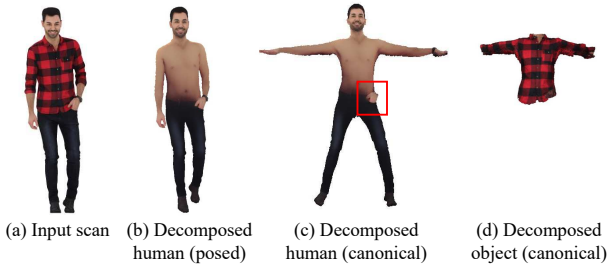
Method	IoU↑	Chamfer↓
Composite	**83.59%**	**1.184**
Object	83.50 %	1.205

Table 1. **SDS loss to composite mesh.** We show the effect of applying SDS loss to composite mesh instead of object mesh.



(a) Input scan (b) Canonical geometry (c) Reposed geometry (d) Zoomed-in view

Figure 10. **Failure case of reposing loose clothing.** Since our method generates static canonical shape, reposing a human with loose clothing may result in severe artifacts between the legs.



(a) Input scan (b) Decomposed human (posed) (c) Decomposed human (canonical) (d) Decomposed object (canonical)

Figure 11. **Failure case of canonicalization.** Our method suffers from correctly canonicalizing scans with hands in their pockets.

Limitations and Future Work. As mentioned in the main paper, GALA currently models a static canonical shape without considering pose-dependent deformations. Fig. 10 illustrates a failure case of reposing a human with loose clothing, where severe artifacts of the dress appear between the legs. Jointly modeling pose-dependent deformation of clothing from a single scan can be a potential direction for future work. Additionally, our method may encounter challenges when canonicalizing input scans with difficult poses such as humans with their hands in their pockets. As shown in Fig. 11 (c), the hand partially remains inside the pocket after decomposition, limiting the reuse of the decomposed human.

Nonetheless, the decomposed human can still be used in the pose of the input scan as depicted in Fig. 11 (b), and the decomposed object of Fig. 11 (d) can be utilized as any other decomposed asset. The dependency on accurate 2D segmentation can be also problematic if the 2D segmentation module fails. Self-discovering each layer without requiring 2D segmentation is also an interesting future work.

Societal Impact. GALA decomposes a single static scan into reusable and animatable assets, *e.g*. target apparel and the underlying human body. Similar to other recent generative models and editing methods, our method may have both positive and negative societal impacts depending on the usage. On the positive side, GALA can immediately generate diverse reusable assets from existing 3D assets that have entangled geometry, without template registration, additional scanning, or editing by 3D designers. For the metaverse applications, GALA enables users to easily digitize their assets and clothe their avatars in the virtual world. On the negative side, GALA may generate a naked underlying body for the human scan with single-layered clothing unless the input prompts are properly given. Since GALA utilizes SDS loss [20] to leverage the prior from the pre-trained 2D diffusion model, this problem can be alleviated via the NSFW filter. Nonetheless, there are still potential problems, *e.g*. privacy violations, fake news, online sexual harassment, etc., like deepfake [26]. In our code release, we will specify the correct use of our method. We believe that the malicious use of generative models should be dealt with through both legal regulation and technology to detect misuse cases. We hope that our work invokes a serious discussion on such issues.

Figure 12. **Decomposition and Canonicalization.** In each set, we show the decomposition and canonicalization results of the leftmost input scan.

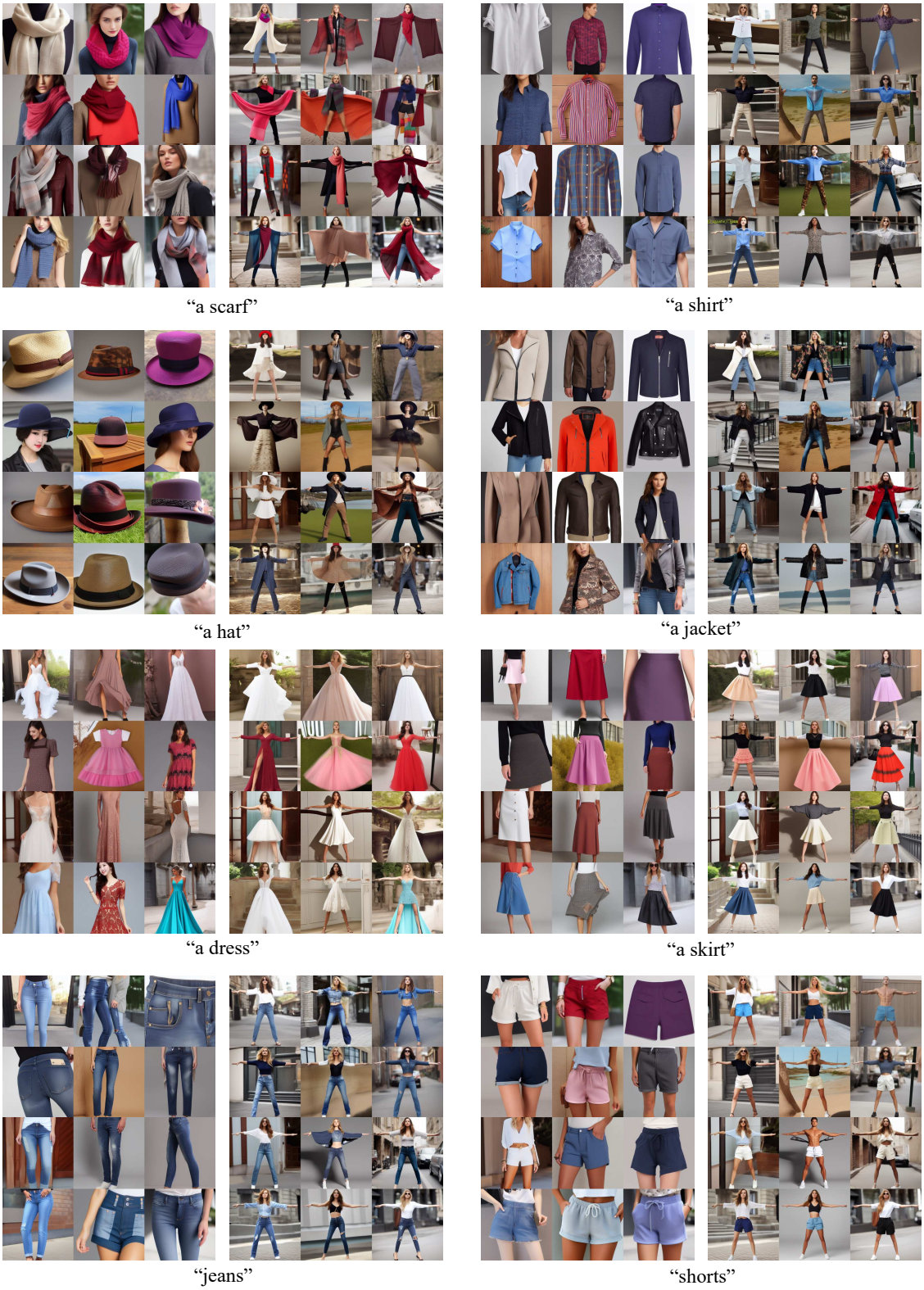


Figure 13. **Pose-guided Generation.** In each set, we show the generated images of the target objects without OpenPose ControlNet on the left, and with OpenPose ControlNet on the right. Diffusion model fails to exclusively generate target objects without humans when OpenPose ControlNet is used for pose-guided SDS loss.

References

[1] B. AlBahar, S. Saito, H.-Y. Tseng, C. Kim, J. Kopf, and J.-B. Huang. Single-image 3d human digitization with shape-guided diffusion. In *Proc. ACM SIGGRAPH Asia*, 2023. 2

[2] T. Alldieck, M. Zanfir, and C. Sminchisescu. Photorealistic monocular 3d reconstruction of humans wearing clothing. In *Proc. CVPR*, 2022. 2

[3] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proc. CVPR*, 2023. 2

[4] Y. Cao, Y.-P. Cao, K. Han, Y. Shan, and K.-Y. K. Wong. Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models. *arXiv preprint arXiv:2304.00916*, 2023. 2

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE TPAMI*, 2019. 4

[6] X. Chen, A. Pang, W. Yang, P. Wang, L. Xu, and J. Yu. Tightcap: 3d human shape capture with clothing tightness field. *ACM TOG*, 2021. 3

[7] X. Chen, T. Jiang, J. Song, M. Rietmann, A. Geiger, M. J. Black, and O. Hilliges. Fast-snarf: A fast deformer for articulated neural fields. *IEEE TPAMI*, 2022. 2, 3

[8] A. Haque, M. Tancik, A. Efros, A. Holynski, and A. Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proc. ICCV*, 2023. 2

[9] T. He, Y. Xu, S. Saito, S. Soatto, and T. Tung. Arch++: Animation-ready clothed human reconstruction revisited. In *Proc. ICCV*, 2021. 3

[10] X. Huang, R. Shao, Q. Zhang, H. Zhang, Y. Feng, Y. Liu, and Q. Wang. Humannorm: Learning normal diffusion model for high-quality and realistic 3d human generation. *arXiv preprint arXiv:2310.01406*, 2023. 2

[11] Y. Huang, H. Yi, Y. Xiu, T. Liao, J. Tang, D. Cai, and J. Thies. TeCH: Text-guided Reconstruction of Lifelike Clothed Humans. In *Proc. 3DV*, 2024. 2

[12] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung. Arch: Animatable reconstruction of clothed humans. In *Proc. CVPR*, 2020. 3

[13] A. Karnewar, T. Ritschel, O. Wang, and N. Mitra. Relu fields: The little non-linearity that could. In *Proc. ACM SIGGRAPH*, 2022. 2

[14] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. In *Proc. ICCV*, 2023. 2

[15] N. Kolotouros, T. Alldieck, A. Zanfir, E. G. Bazavan, M. Fieraru, and C. Sminchisescu. Dreamhuman: Animatable 3d avatars from text. *arXiv preprint arXiv:2306.09329*, 2023. 2

[16] T. Liao, H. Yi, Y. Xiu, J. Tang, Y. Huang, J. Thies, and M. J. Black. TADA! Text to Animatable Digital Avatars. In *Proc. 3DV*, 2024. 2

[17] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black. Learning to dress 3d people in generative clothing. In *Proc. CVPR*, 2020. 1, 2

[18] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. ECCV*, 2020. 2

[19] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 1

[20] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *Proc. ICLR*, 2023. 1, 5

[21] Renderpeople, 2018. https://renderpeople.com/3d-people. 1

[22] E. Richardson, G. Metzer, Y. Alaluf, R. Giryes, and D. Cohen-Or. Texture: Text-guided texturing of 3d shapes. *ACM TOG*, 2023. 2

[23] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proc. ICCV*, 2019. 2

[24] S. Saito, T. Simon, J. Saragih, and H. Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proc. CVPR*, 2020. 2

[25] E. Sella, G. Fiebelman, P. Hedman, and H. Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *Proc. ICCV*, 2023. 2

[26] M. Westerlund. The emergence of deepfake technology: A review. *Technology innovation management review*, 9(11), 2019. 5

[27] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black. Icon: implicit clothed humans obtained from normals. In *Proc. CVPR*, 2022. 2

[28] Y. Xiu, J. Yang, X. Cao, D. Tzionas, and M. J. Black. ECON: Explicit Clothed humans Optimized via Normal integration. In *Proc. CVPR*, 2023. 2

[29] H. Zhang, B. Chen, H. Yang, L. Qu, X. Wang, L. Chen, C. Long, F. Zhu, K. Du, and M. Zheng. Avatarverse: High-quality & stable 3d avatar creation from text and pose. *arXiv preprint arXiv:2308.03610*, 2023. 2

[30] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proc. ICCV*, 2023. 4