

7. Appendix

Due to space constraints in the main paper, we provide additional information and details in this supplementary material. These include:

- A more detailed discussion of the preliminaries in Section 7.1.
- Comprehensive training details of the Identity Encoding Loss in Section 7.2.
- Implementation details outlined in Section 7.3.
- Analysis of failure cases in Section 7.4.
- Ablation of initialization strategies in Section 7.5
- Examples of hair tagging in Section 7.6
- Additional visualizations presented in Section 7.9.
- Detailed information on the demo video [Disco4D-demo.mp4] in Section 7.10.

7.1. Preliminary

3D Gaussian Splatting utilizes explicit 3D Gaussian points as the core elements for rendering. Each 3D Gaussian point is defined by the function:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)},$$

where μ represents the spatial mean, and Σ denotes the covariance matrix. Additionally, each Gaussian is assigned an opacity value α and a view-dependent color c , parameterized by spherical harmonic coefficients f . During rendering, these 3D Gaussians are projected onto the 2D view plane via a splatting technique. The 2D projection is computed using the projection matrix, while the 2D covariance matrices are approximated as: $\Sigma' = J_g W_g \Sigma W_g^T J_g^T$, where W_g is the viewing transformation, and J_g is the Jacobian of the affine approximation for perspective projection. The final pixel color is obtained through alpha-blending of N layered 2D Gaussians from front to back $C = \sum_{i \in N} T_i \alpha_i c_i$, with $T_i = \prod_{j=1}^i (1 - \alpha_j)$.

The opacity α is determined by multiplying γ with the contribution of the 2D covariance, derived from Σ' and the pixel coordinate in image space. The covariance matrix Σ is parameterized using a quaternion q and a 3D scaling vector v to aid in optimization.

SMPL-X parameterization [71] extends the original SMPL body model [62] by incorporating detailed face and hand deformations to capture more expressive human movements. **SMPL-X** expands **SMPL** joint set by including additional joints for facial features, toes and fingers, enabling a more accurate representation of complex body movements. **SMPL-X** is defined by a function $M(\beta, \theta, \psi) : \mathbb{R}^{|\beta| \times |\theta| \times |\psi|} \rightarrow \mathbb{R}^{3N}$, where $\theta \in \mathbb{R}^{3K}$ represents the pose (with K being the number of body joints), $\beta \in \mathbb{R}^{|\beta|}$ represents body shape, and $\psi \in \mathbb{R}^{|\psi|}$ captures facial expressions. Further details can be found in [71].

7.2. Training details of Identity Encoding loss

To optimize the introduced Identity Encoding of each Gaussian, we render these encoded identity vectors into 2D images in a differentiable manner following [111]. We adapt the differentiable 3D Gaussian renderer from [47], approaching the rendering process similarly to the color optimization using spherical harmonic (SH) coefficients, as described in [47]. In this method, 3D Gaussian splatting utilizes neural point-based α' -rendering [52, 53], where the influence weight α' is calculated in 2D for each Gaussian and pixel. Following the approach in [47], the influence of all Gaussians on a pixel is computed by sorting them based on depth and blending the N ordered Gaussians that overlap with that pixel:

$$E_{id} = \sum_{i \in N} e_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (3)$$

Here, the rendered 2D mask identity feature E_{id} is the sum of the Identity Encoding e_i (of length 15) for each Gaussian, weighted by the Gaussian's influence factor α'_i on that pixel. The value of α'_i is determined by evaluating a 2D Gaussian with covariance $\Sigma 2D$, which is scaled by a learned per-point opacity α_i :

$$\Sigma 2D = J W \Sigma 2D^3 W^T J^T \quad (4)$$

where Σ^{3D} is the 3D covariance matrix, $\Sigma 2D$ represents the splatted 2D counterpart, J is the Jacobian of the affine approximation for the 3D-to-2D projection, and W is the world-to-camera transformation matrix.

To ensure consistency in the Identity Encoding e_i during training, we apply an unsupervised 3D regularization loss. This loss encourages the Identity Encodings of the top k -nearest 3D Gaussians to remain close in feature space, promoting spatial consistency. Using the softmax function F , we define the KL divergence loss with m sampled points as follows:

$$\mathcal{L}_{3d} = \frac{1}{m} \sum_{j=1}^m D_{KL}(P||Q) = \frac{1}{mk} \sum_{j=1}^m \sum_{i=1}^k F(e_j) \log \left(\frac{F(e_j)}{F'(e_j)} \right) \quad (5)$$

Here, P is the sampled Identity Encoding e of a 3D Gaussian, and Q consists of the k -nearest neighbors in 3D space, represented as e'_1, e'_2, \dots, e'_k . The total identity encoding loss is then defined as:

$$\mathcal{L}_{id} = \mathcal{L}_{2d} + \mathcal{L}_{3d} \quad (6)$$

7.3. Implementation details

The 3D generation experiments were conducted using a single 24GB RTX3090 GPU, while the 4D generation experiments utilized a single 48GB RTX6000 GPU. For the 3D generation process, the SMPL-X fitting was performed

with 3000 iterations in 3 minutes, followed by skin color inpainting on SMPL-X Gaussians for 100 iterations in 30 seconds. Reconstruction and disentanglement optimization required 3000 iterations, completed in 12 minutes. In video reconstruction, SMPL-X fitting aligned 14 frames in 6 minutes for in-the-wild videos. The 4D-Dress [96] experiments involved 1000 iterations for clothing deformation over 18 minutes.

7.4. Failure cases

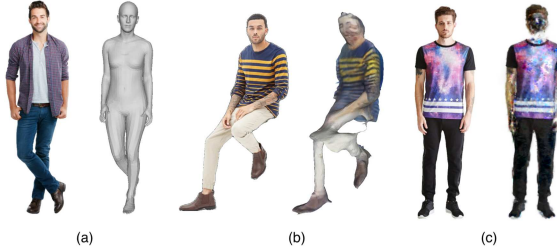


Figure 7. **Failure cases of Disco4D.** (a) Poor SMPL-X estimation (b) Poor visual hull initialization (c) Misclassification of clothing categories.

Disco4D relies on robust and pixel-aligned SMPL-X estimation, which is still an unsolved problem, especially for challenging poses. In Figure 7a, it is difficult to correct the pose with keypoints and segmentation mask due to depth ambiguity. Disco4D occasionally fails for poor visual hull initialization (7b), which is common for difficult poses. Lastly, poor disentanglement is a common problem due to misclassification of clothing category by the segmentation model. This is seen in Figure 7c where the arms are wrongly classified under the "top" category.

7.5. Initialization

We evaluate random, surface, and hull-based initialization strategies. Surface initialization on SMPL-X often produces inaccurate geometries for complex or loose garments, leading to elongated Gaussians and artifacts. Hull-based initialization better captures garment details, preserves pose consistency, and aligns closely with the true clothed body geometry, as seen in Figure 8.

7.6. Hair tagging

In our approach, hair Gaussians are tagged to head faces rather than the nearest face during reposing. Reposing hair Gaussians according to the nearest face, as commonly done in previous works, often results in artifacts such as disjointed hair (Figure 9). By leveraging the learned identity encoding, we assign a unified identity to hair Gaussians, enabling them to be reposed cohesively as a single entity, thereby preserving the structural integrity of the hair during transformations.

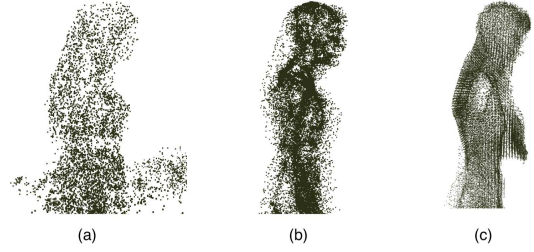


Figure 8. **Ablation of initialization.** (a) Random Initialization (b) SMPL-X Initialization (c) Visual Hull Initialization.

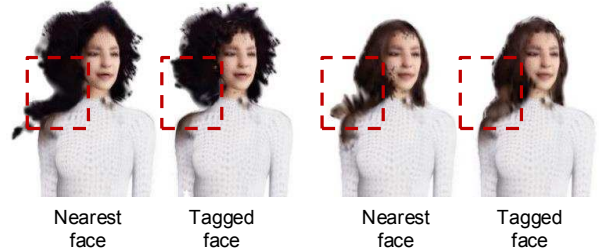


Figure 9. **Visualization of hair tagging.**

7.7. In-the-wild evaluation



Figure 10. **Qualitative evaluation on ITW images.**



Figure 11. **Qualitative evaluation on avatars clothed in dress.**

Our focus on studio and synthetic datasets (e.g., Synbody, CloSe, and 4DDress) was due to the availability of ground-

truth data from multiple views, enabling rigorous quantitative evaluation. ITW images lack such ground-truth data, making comparisons challenging. Nevertheless, our solution applies to ITW images, with some examples shown in Fig. 10. Examples of avatars clothed in dress are added in Fig 11, driven with poses from subjects in Fig. 10.

7.8. Facial detail

Additional visualizations of well known individuals are provided in Fig. 10 and Fig. 11.

7.9. Extra visualizations

Figure 12 presents visual comparisons with 2D animation methods. Figure 13 illustrates ablation results for 4D reconstruction. Finally, ablation studies on point geometry and editing are provided in Figure 14.

7.10. Demo video

Extended visualizations and results showcasing 3D generation and disentanglement, pose-driven animation, video-to-4D reconstruction, and fine-grained editing of animated outputs are demonstrated in the accompanying demo video [\[Disco4D-demo.mp4\]](#). A sample of the video is shown in Figure 15.



Figure 12. **Comparison to 2D animation methods.** Compared to Magic-Animate and Animate-Anyone, we have better preservation of body shape and details. Compared to CHAMP, we have better geometry and consistency.



Figure 13. 4D reconstruction results on 4D-Dress Dataset.

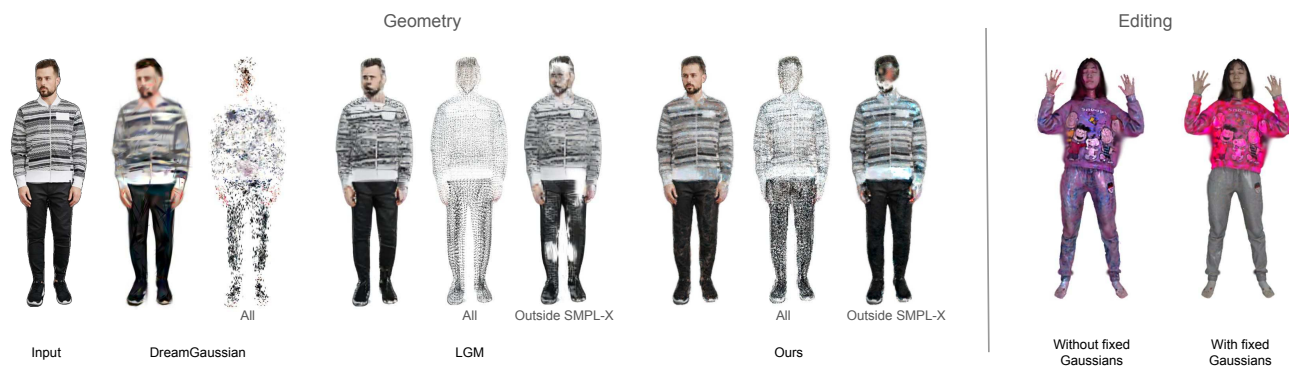


Figure 14. Ablation of points geometry (left) and editing results (right). Points ("All") are visualised with a Gaussian Scale of 0.1.



Figure 15. Additional visualizations showcasing generation, disentanglement, animation, and editing. Full demonstrations are available in the accompanying demo video [\[Disco4D-demo.mp4\]](#). This figure provides a sample from the demo.