# DAMA: Disentangled Body-Anchored Gaussians for Controllable Multi-Layered Avatars

Supplementary Material

## A. Implementation Details

### A.1. Losses

**Color Loss.** We use an $L_1$ loss between the rendered image and the ground-truth image:

$$\mathcal{L}_c = \|I_{\text{rend}} - I_{\text{gt}}\|_1. \tag{1}$$

For segmentation lifting, $I_{\text{rend}}$ contains the rendered label colors assigned to semantic classes. For appearance optimization, it contains the rendered RGB colors.

**Scale Loss.** We keep the scales of segmentation Gaussians close to the scales of the corresponding SMPL-X Gaussians to preserve similar surface coverage. Let $\mathbf{s}_i^{\text{seg}}$ denote the scale of Gaussian $g_i^{\text{seg}}$ and $\mathbf{s}_i^{\text{smplx}}$ the scale of its corresponding SMPL-X Gaussian. We use an $L_1$ loss:

$$\mathcal{L}_s = \frac{1}{N_{\text{seg}}} \sum_{i=1}^{N_{\text{seg}}} \left\| \mathbf{s}_i^{\text{seg}} - \mathbf{s}_i^{\text{smplx}} \right\|_1. \tag{2}$$

**Normal Loss.** $\mathcal{L}_n$ aligns Gaussian normals with normals estimated from rendered depth maps. We use the same formulation and implementation as the normal regularization introduced in 2DGS [2].

**Label Smoothness Loss.** Let $\mathbf{p}_i$ denote the label probability vector of Gaussian $g_i^{\text{seg}}$, with label $\ell_i^{seg} = \arg\max(\mathbf{p}_i)$. We encourage neighboring Gaussians to share similar label distributions. For precomputed neighbors $\mathcal{N}(i)$ we compute the KL divergence and average over all $N_{\text{seg}}$ Gaussians:

$$\mathcal{L}_\ell = \frac{1}{N_{\text{seg}}} \sum_{i=1}^{N_{\text{seg}}} \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} D_{\text{KL}}(\mathbf{p}_i^{seg} \,\|\, \mathbf{p}_j^{seg}). \tag{3}$$

**Mask Loss.** We use an $L_1$ loss between the rendered layer mask and the ground-truth layer mask:

$$\mathcal{L}_m = \|M_{\text{rend}} - M_{\text{gt}}\|_1. \tag{4}$$

**Anisotropic Loss.** We use the anisotropic regularizer $\mathcal{L}_a$ introduced in PhysGaussian [4].

**Canonical Distance Loss.** We use an $L_2$ loss to keep Gaussians close to the SMPL-X surface in canonical space. Let $\boldsymbol{\mu}_i^l$ be the Gaussian mean belonging to layer $l$ and $\boldsymbol{\mu}_i^{\text{smplx}}$ the center of its bound SMPL-X face:

$$\mathcal{L}_d = \frac{1}{N_l} \sum_{i=1}^{N_l} \|\boldsymbol{\mu}_i^l - \boldsymbol{\mu}_i^{\text{smplx}}\|_2. \tag{5}$$

**Canonical Rotation Loss.** We align Gaussian orientations with the orientation of their bound SMPL-X face in canonical space. Let $\mathbf{q}_i^l$ denote the Gaussian rotation belonging to layer $l$ and $\mathbf{q}_i^{\text{smplx}}$ the SMPL-X face rotation, both in canonical space:

$$\mathcal{L}_r = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(1 - \langle \mathbf{q}_i^l, \mathbf{q}_i^{\text{smplx}} \rangle\right). \tag{6}$$

### A.2. Optimization and Runtime

We set the loss weights as follows: $\lambda_c$=1, $\lambda_s$=10, $\lambda_n$=0.1, $\lambda_\ell$=0.1, $\lambda_a$=100, $\lambda_d$=1, and $\lambda_r$=100. All experiments run on a single NVIDIA A100 GPU. In Stage 1, we optimize $\mathcal{G}^{\text{seg}}$ for 10k iterations ($\sim$3 min) and enable the label smoothness loss $\mathcal{L}_\ell$ after 5k iterations. In Stage 3, we optimize each semantic layer independently for 2k iterations ($\sim$1.5 min per layer), followed by a final joint optimization of all layers for 2k iterations. The full method takes about 10–15 minutes depending on the number of layers.

## B. Additional Loss Ablations

We ablate $\mathcal{L}_a$, $\mathcal{L}_d$, and $\mathcal{L}_r$. $\mathcal{L}_a$ prevents Gaussian shrinkage/explosion, while $\mathcal{L}_d$ and $\mathcal{L}_r$ stabilize weakly supervised regions (e.g., underarms), preventing noisy geometry during animation (Fig.1).

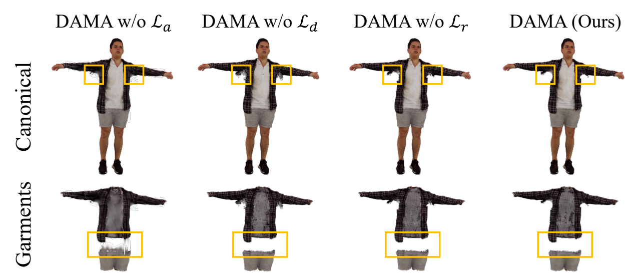


Figure 1. **Additional Loss Ablations.** Removing $\mathcal{L}_a$, $\mathcal{L}_d$, $\mathcal{L}_r$.

## C. Additional Applications and Results

**Hair Transfer.** Our representation extends to hair. Fig. 2 shows transferring hair from a source subject to a target and changing its layer order.


Figure 2. **Hair Transfer.** Source hair transferred and reordered.

**Additional Results.** We show SMPL-X–driven animation of stacked Gaussian garments with preserved layer ordering (Fig. 3). We also show further simulation results of stacked garment meshes extracted from the Gaussians (Fig. 4).


Figure 3. **SMPL-X–Driven Avatar Animation.** We animate the reconstructed avatar with transferred and stacked garments using SMPL-X motion sequences from AMASS [3]. The sequence shows that the layered garments deform consistently with the body while preserving their ordering and separation throughout the motion.

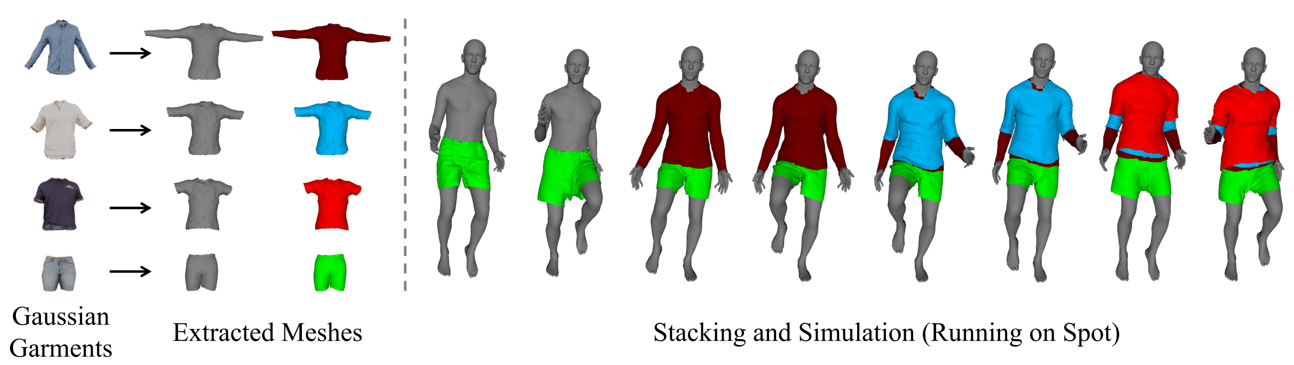


Figure 4. **Additional Clothing Simulation Example.** We show an additional example with one lower garment and three upper garments. (Left) Simulation-ready meshes extracted from the Gaussian layers. (Right) CLO3D [1] simulation driven by a running-on-spot motion sequence from AMASS [3]. The garments are progressively stacked, showing that the extracted meshes preserve layer ordering and remain stable during simulation.

# References

[1] CLO Virtual Fashion. Clo 2024.0, 2024. 3D Fashion Design Software. 2

[2] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 1

[3] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, 2019. 2

[4] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 1