

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:

$$\mathcal{L}_c = \|I_{\text{rend}} - I_{\text{gt}}\|_1 \quad (1)$$

For segmentation lifting, I_{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^{seg} and $\mathbf{s}_i^{\text{smplx}}$ the scale of its corresponding SMPL-X Gaussian. We use an L_1 loss:

$$\mathcal{L}_s = \frac{1}{N_{\text{seg}}} \sum_{i=1}^{N_{\text{seg}}} \|\mathbf{s}_i^{\text{seg}} - \mathbf{s}_i^{\text{smplx}}\|_1 \quad (2)$$

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^{seg} , with label $\ell_i^{\text{seg}} = \arg \max(\mathbf{p}_i)$. 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_{seg} Gaussians:

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

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

$$\mathcal{L}_m = \|M_{\text{rend}} - M_{\text{gt}}\|_1 \quad (4)$$

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:

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

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:

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

A.2. Topology-Aware Label Refinement Algorithm

We provide the exact algorithm for the topology-aware refinement stage.

After Stage 1, each Gaussian is assigned a label ℓ_i . These labels can be noisy. We project the labels onto the SMPL-X mesh by associating each Gaussian with its corresponding face f_i , and refine them on the mesh topology to obtain ℓ_i^{ref} .

Let A_i denote the area of face f_i . We define a face adjacency graph $\mathcal{G} = (\mathcal{F}, \mathcal{E})$, where $(f_i, f_j) \in \mathcal{E}$ if the two faces share an edge. The neighbors of a face f_i are $\mathcal{N}(i) = \{j \mid (f_i, f_j) \in \mathcal{E}\}$.

We extract connected components $C \subset \mathcal{F}$ of faces sharing the same label. Let $A(C) = \sum_{i \in C} A_i$. We introduce an area threshold τ and treat components with $A(C) < \tau$ as spurious, reassigning them to the dominant label of their neighboring faces. This enforces spatial consistency while preserving large regions, yielding refined labels $\{\ell_i^{\text{ref}}\}$. The full procedure is summarized in Alg. 1.

Algorithm 1 Topology-Aware Label Refinement

- 1: **Input:** faces $\{f_i\}$, labels $\{\ell_i\}$, areas $\{A_i\}$, threshold τ
 - 2: Build adjacency graph \mathcal{G}
 - 3: Initialize $\ell_i^{\text{ref}} \leftarrow \ell_i$
 - 4: **repeat**
 - 5: // same-label regions
 - 6: Extract connected components $C \subset \mathcal{F}$ such that $\ell_i^{\text{ref}} = \ell_j^{\text{ref}} \forall i, j \in C$
 - 7: **for each component** C **do**
 - 8: // compute area
 - 9: $A(C) \leftarrow \sum_{i \in C} A_i$
 - 10: **if** $A(C) < \tau$ **then**
 - 11: // find neighbors
 - 12: $\mathcal{N}(C) \leftarrow \{j \notin C \mid \exists i \in C, j \in \mathcal{N}(i)\}$
 - 13: // majority vote
 - 14: $\ell^* \leftarrow \text{mode}(\{\ell_j^{\text{ref}} \mid j \in \mathcal{N}(C)\})$
 - 15: // reassign labels
 - 16: $\ell_i^{\text{ref}} \leftarrow \ell^* \quad \forall i \in C$
 - 17: **end if**
 - 18: **end for**
 - 19: **until** no change in ℓ^{ref}
 - 20: **Output:** refined labels $\{\ell_i^{\text{ref}}\}$
-

A.3. 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}^{seg} for 10k iterations (~ 3 min) and enable the label smoothness loss \mathcal{L}_ℓ after 5k iterations. In Stage 3, we optimize each semantic layer independently for 2k iterations (~ 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. Evaluation Metrics

Let $\mathcal{V}^{\text{gt}} = \{\mathbf{v}_i^{\text{gt}}\}_{i=1}^N$ denote the set of ground-truth scan vertices and $\mathcal{V}^{\text{rec}} = \{\mathbf{v}_j^{\text{rec}}\}_{j=1}^M$ the set of reconstructed 3D points, represented as Gaussian means for Gaussian-based methods or mesh vertices for mesh-based methods. The body surface is represented by $\mathcal{V}^{\text{body}} = \{\mathbf{v}_k^{\text{body}}\}_{k=1}^K$ with corresponding outward normals $\{\mathbf{n}_k\}_{k=1}^K$. All quantities are evaluated in the posed space.

Geometric Accuracy. Geometric accuracy is quantified using the two-way Chamfer distance:

$$\begin{aligned} \text{CD} = & \frac{1}{N} \sum_{i=1}^N \min_j \|\mathbf{v}_i^{\text{gt}} - \mathbf{v}_j^{\text{rec}}\|_2 \\ & + \frac{1}{M} \sum_{j=1}^M \min_i \|\mathbf{v}_j^{\text{rec}} - \mathbf{v}_i^{\text{gt}}\|_2. \end{aligned} \quad (7)$$

capturing bidirectional proximity and ensuring that reconstructed points cover the ground truth points while remaining close to them.

Physical Plausibility. Physical plausibility is evaluated via the signed distance d_j of each reconstructed point $\mathbf{v}_j^{\text{rec}} \in \mathcal{V}^{\text{rec}}$ to the body surface:

$$d_j = \min_k (\mathbf{v}_j^{\text{rec}} - \mathbf{v}_k^{\text{body}}) \cdot \mathbf{n}_k, \quad (8)$$

which indicates whether a point lies outside the body or penetrates it along the local surface normal. Penetration depth is defined as:

$$\text{PD} = \frac{1}{|\{j \mid d_j < 0\}|} \sum_{j: d_j < 0} (-d_j), \quad (9)$$

capturing the average extent of interpenetration. The penetration rate is given by:

$$\text{PR} = \frac{|\{j \mid d_j < 0\}|}{M}, \quad (10)$$

reflecting the proportion of reconstructed points that lie inside the body.

C. Additional Loss Ablations

We ablate \mathcal{L}_a , \mathcal{L}_d , and \mathcal{L}_r to study their individual effects. The loss \mathcal{L}_a prevents Gaussian shrinkage or explosion, while \mathcal{L}_d and \mathcal{L}_r stabilize weakly supervised regions (e.g., underarms), reducing noisy geometry during animation (Fig. 1).

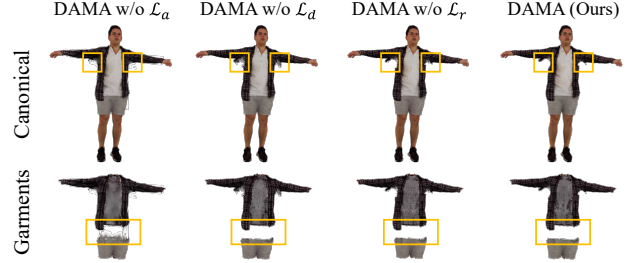


Figure 1. **Additional Loss Ablations.** Effect of removing \mathcal{L}_a , \mathcal{L}_d , and \mathcal{L}_r .

D. Additional Applications and Results

Hair Transfer. Our representation naturally extends to hair. Fig. 2 illustrates transferring hair from a source subject to a target, along with reordering its layer.



Figure 2. **Hair Transfer.** Hair transferred from a source subject and reordered.

Additional Results. We further present SMPL-X-driven animation of stacked Gaussian garments with preserved layer ordering (Fig. 3). We also include additional 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.

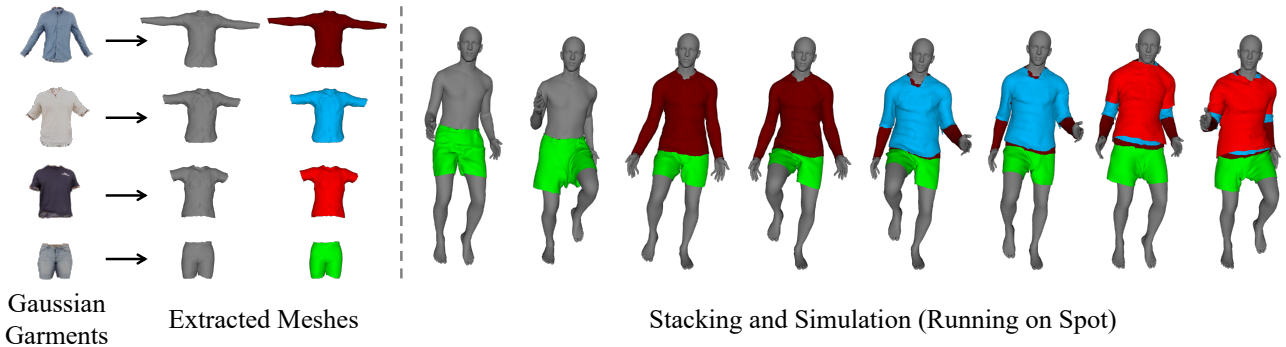


Figure 4. **Additional Clothing Simulation Example.** We show an additional example with one lower garment and three upper garments. (Left) Simulation-ready meshes extracted from the Gaussian layers. (Right) CLO3D [1] simulation driven by a running-on-spot motion sequence from AMASS [3]. The garments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

References

- [1] CLO Virtual Fashion. *CLO3D (Version 2025.2.368)*. CLO Virtual Fashion, Seoul, South Korea, 2026. Updated March 19, 2026. [3](#)
- [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. [3](#)
- [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](#)