

SUPPLEMENTARY AHA! ANIMATING HUMAN AVATARS IN DIVERSE SCENES WITH GAUSSIAN SPLATTING

Anonymous authors

Paper under double-blind review

In the supplementary material, we begin by ablating the various components of our method, highlighting the contribution of each to the overall performance. Next, we provide a deeper analysis of Baseline A, focusing on its failure modes and the reasons behind its limitations. Finally, we include additional background information to further clarify and contextualize the main content of the paper.

1 FURTHER EXPERIMENTS:

Motion Metrics. We evaluate using the following metrics:

1. **Foot Contact Score.** Following prior work, we compute the scaled foot skating when in contact with the floor Zhang et al. (2018) : The foot velocities are added if a position height h is within a maximum threshold of $H = 2.5$ cm. Each velocity magnitude v in the horizontal plane is further weighted by an exponential interpolation

$$s = v^{(2-2\frac{h}{H})}$$

to estimate the amount of skating s during motion, where the exponent is clamped between 0 and 1.

2. **Interaction Penetration.** For each frame, the penetration score is defined as

$$s_{\text{inter.pene}} = \sum_{v_i \in V} |(\Psi_O(v_i))_-|, \quad (1)$$

where Ψ_O is the signed distance field (SDF) of the scene objects, $(\cdot)_-$ clips all positive values to zero, and V denotes body vertices. We report both average penetration over time and maximum penetration within a sequence.

3. **Jerk.** To capture temporal smoothness, we compute the *jerk* as the third derivative of the body joint trajectories, averaged over time and joints.
4. **Goal Reaching.** We measure the success rate of whether the synthesized motion successfully reaches the designated goal position - the goal is reached if final pose is within a threshold of the user specified goal location.

1.1 ABLATION EVALUATION

Our motion synthesis framework operates directly on 3D Gaussians. However, most existing metrics are defined in the mesh space. Therefore, for ablation studies, we adopt the ScanNet++ dataset Yeshwanth et al. (2023), where both mesh reconstructions of the scenes and their corresponding 3D Gaussian splatting (3DGS) representations are available. Motion is synthesized using 3DGS, while the mesh geometry is only used for evaluation purposes.

Ablation Settings. To analyze the contribution of each component in our framework, we design the following ablation experiments:

- **Without Opacity Culling.** In this setting, we disable our proposed opacity culling scheme and instead rely on a naïve projection of all Gaussians into the scene representation. As a consequence, overlapping or redundant Gaussians are not filtered out, which leads to spurious geometry in collision reasoning. This causes the motion policy to perceive phantom



Figure 1: a) Naive orthographic projection of Gaussians leads to blocky structures b) Orthographic projections With opacity culling c) Path in filtered map



Figure 2: w/o and with Gaussian walkability map

obstacles, often resulting in high penetration values and degraded goal reaching performance. See Fig 1

- **Without Optimization Refinement.** Here, we remove the refinement stage after Gaussian placement. The absence of this refinement step produces slight inconsistencies such as penetration of the human hip or feet Gaussians through the scene Gaussians. See Fig 3
- **Without Walkability Map.** Instead of using our modified walkability map to guide reinforcement learning, we allow the agent to explore the scene without explicit walkability constraints. While the system can still synthesize plausible motion, the lack of walkability guidance creates penetration during locomotion. As a result, characters occasionally attempt infeasible paths (e.g., walking through thin obstacles), leading to increased penetration and reduced robustness of the generated motion. See Fig. 2

Table 1: Ablation study on ScanNet++. Motion is synthesized using 3DGS; metrics are evaluated in mesh space. Lower is better for skating, penetration, and jerk; higher is better for goal reaching.

Method	Foot Contact (cm/s) ↓	Penetration (m) ↓	Jerk (m/s^3) ↓	Goal Reaching (%) ↑
Ours (full)	0.89	0.012	0.47	95.2
w/o opacity culling	1.45	0.042	0.78	42.6
w/o walkability map	1.37	0.021	0.71	84.5
w/o optimization refinement	1.11	0.016	0.55	91.3

1.2 MOTION EVALUATION: BASELINE A

For baseline A we also report standard motion evaluation metrics: foot skate, human scene penetration. Here we use the VGGT Wang et al. (2025) meshes as proxies for the 3D scene. We synthesize motion for our method using 3DGS scenes, for baseline A using VGGT scenes but report numbers on SMPL parameters in relation to VGGT meshes. Please note through this experiment, we don't claim to improve upon the motion quality of existing frameworks in general cases but our claim is more limited: that existing motion frameworks, as they rely on meshes, fail when deployed on meshes



Figure 3: With (left) and right (without) refinement of Gaussians



Figure 4: VGGT mesh reconstruction from monocular video

reconstructed from monocular RGB videos, while our method does not. It should also be noted that these papers themselves do not claim to work in this challenging setting.

Table 2: Baseline comparison on VGGT meshes (monocular setting) . Motion is synthesized using 3DGS for ours and VGGT meshes for Baseline A; metrics are evaluated in mesh space using VGGT meshes. Lower is better for skating, penetration, and jerk; higher is better for goal reaching.

Method	Foot Contact (cm/s) ↓	Penetration (m) ↓	Jerk (m/s^3) ↓	Goal Reaching (%) ↑
Ours (3DGS → VGGT eval)	0.92	0.014	0.49	93.8
Baseline A(VGGT)	2.84	0.089	0.83	58.6

2 FAILURE MODES OF BASELINE A: ANALYSIS

Please see supplementary video for results of Baseline A. Here (Fig. 4) we show that the underlying mesh recovered from VGGT contains blocked paths and significant noisy structures. This results in 1) navigation paths that penetrate through the scene 2) interaction poses with significant penetration with the scene 3) unnatural motion. Note that for this baseline we do not apply any refinement in Gaussian space but use naive composition for rendering.

3 VLM GUARDRAILS

For the pairwise VLM study, the VLM is instructed to ignore artistic style and focus on physical plausibility: “Which image appears more photorealistic? Consider geometry (straight lines, depth cues), materials (BRDF, speculars), lighting/shadows, and absence of artifacts (flicker, halos, floaters). Respond with A, B, or Tie.”

4 BACKGROUND: GAUSSIAN SPLATTING

3D Gaussian Splatting Kerbl et al. (2023) uses a set of 3D Gaussian primitives \mathcal{G} to represent static 3D scenes which can be rendered in real-time using rasterization. Each 3D Gaussian in the set \mathcal{G} is parametrized by its mean \mathbf{x} , covariance Σ , opacity α and color parametrized by spherical harmonics coefficients \mathbf{p} . To ensure positive semi-definiteness, the covariance matrix is decomposed into this scaling \mathbf{S} and rotation components \mathbf{R} .

$$\Sigma = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T \quad (2)$$

A quaternion $\mathbf{q} \in \mathbb{R}^4$ is used to represent rotation, which can be trivially converted to a valid covariance matrix to ensure that the matrix represents a valid rotation. The scale matrix is a diagonal matrix parametrized by a scaling vector \mathbf{s} . In this paper we use the following notations to denote a set of standard 3D Gaussian primitives: $\{\mathbf{x}_k, \mathbf{q}_k, \alpha_k, \mathbf{s}_k, \mathbf{p}_k\}_{k=1}^K$ where K is the total number of Gaussian primitives for a given scene.

The 3D Gaussians are projected to the 2D image plane and alpha-blended. Given a viewing transformation \mathbf{W} and the Jacobian of the affine approximation of the projective transformation \mathbf{J} , the 2D covariance matrix in camera coordinates Zwicker et al. (2001) is given by $\Sigma' = (\mathbf{J}\mathbf{W}\Sigma\mathbf{W}^T\mathbf{J}^T)_{1:2,1:2}$. The pixel color C is computed by blending 3D Gaussian splats that overlap at the given pixel, sorted according to their depth:

$$C = \sum_i \left(\alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \right) c_i, \quad (3)$$

where α'_i denotes the learned opacity α_i weighted by the probability density of i -th projected 2D Gaussian at the target pixel location. c denotes the view-dependent color obtained from Spherical Harmonics coefficients \mathbf{p} . The parameters of the individual 3D Gaussians are optimized by comparing rendered pixels against provided ground truth images via a photometric loss. During optimization, 3DGS adaptively controls the number of 3D Gaussians via periodic densification and pruning.

5 BACKGROUND: LINEAR BLEND SKINNING FOR POSING

We deform canonical, SMPL-anchored Pavlakos et al. (2019); Loper et al. (2015) Gaussians into posed space using linear blend skinning (LBS).

Canonical Gaussians and anchoring. Each canonical Gaussian is

$$(\mathbf{x}_k^C, \Sigma_k^C, \alpha_k, \mathbf{c}_k),$$

with center $\mathbf{x}_k^C \in \mathbb{R}^3$, covariance $\Sigma_k^C \in \mathbb{R}^{3 \times 3}$, opacity $\alpha_k \in (0, 1)$, and color \mathbf{c}_k . Gaussians are anchored to the SMPL surface by nearest-neighbor association to mesh vertices.

Skinning weights. Let B be the number of SMPL bones (joints). For Gaussian k , we use skinning weights

$$w_{kb} \in [0, 1], \quad \sum_{b=1}^B w_{kb} = 1,$$

inherited from its associated SMPL vertex.

Bone transformations. Given pose parameters θ_t , SMPL provides per-bone rigid motions

$$\mathbf{R}_b(\theta_t) \in \text{SO}(3), \quad \mathbf{t}_b(\theta_t) \in \mathbb{R}^3,$$

relative to the canonical rest pose.

Position deformation. The posed center of Gaussian k at time t is the LBS blend of bone transforms:

$$\mathbf{x}_k^P(\boldsymbol{\theta}_t) = \sum_{b=1}^B w_{kb} (\mathbf{R}_b(\boldsymbol{\theta}_t) \mathbf{x}_k^C + \mathbf{t}_b(\boldsymbol{\theta}_t)). \quad (4)$$

Covariance deformation. We approximate the posed covariance by applying the same blended rotational action:

$$\boldsymbol{\Sigma}_k^P(\boldsymbol{\theta}_t) \approx \left(\sum_{b=1}^B w_{kb} \mathbf{R}_b(\boldsymbol{\theta}_t) \right) \boldsymbol{\Sigma}_k^C \left(\sum_{b=1}^B w_{kb} \mathbf{R}_b(\boldsymbol{\theta}_t) \right)^\top. \quad (5)$$

This first-order treatment ignores shear and higher-order effects, but is accurate in practice and mirrors vertex LBS behavior.

Posed Gaussians and rendering. The posed human Gaussians are

$$\mathcal{G}_t^P = \{(\mathbf{x}_k^P, \boldsymbol{\Sigma}_k^P, \alpha_k, \mathbf{c}_k)\}_{k=1}^{N_H},$$

which we render with the standard 3D Gaussian Splatting rasterizer. This defines a differentiable mapping

$$\mathcal{H} : \boldsymbol{\theta}_t \mapsto \mathcal{G}_t^P,$$

used during training and inference in combination with the pose-conditioned canonical prediction f_ϕ .

6 POSE PROJECTION STRATEGY

To improve generalization to unseen poses, following Li et al. (2023), we constrain test-time poses to remain within the distribution of training poses using Principal Component Analysis (PCA). Let $\{P_t\}_{t=1}^T$ denote the set of training pose maps, each rasterized as described in Sec 3.2. We vectorize each P_t into $\mathbf{p}_t \in \mathbb{R}^D$, where $D = HWC$ is the flattened pose-map dimension, and stack them into a data matrix $X = [\mathbf{p}_1, \dots, \mathbf{p}_T]$.

We perform PCA on X , obtaining the mean $\bar{P} \in \mathbb{R}^D$ and the top K principal components $Q_K \in \mathbb{R}^{D \times K}$ with corresponding standard deviations $\{\sigma_i\}_{i=1}^K$. A novel test-time pose map, vectorized as \mathbf{p} , is projected into this subspace by

$$z_y = Q_K^\top (\mathbf{p} - \bar{P}), \quad (6)$$

where each coefficient $z_{y,i}$ is clipped to the interval $[-2\sigma_i, 2\sigma_i]$ to enforce plausibility. The low-dimensional reconstruction is then

$$\tilde{P}_y = \bar{P} + Q_K z_y, \quad (7)$$

which is reshaped back into a pose map of size $H \times W \times C$.

As detailed in Sec 3.2 of the main paper, the reconstructed map \tilde{P}_y is fed into the StyleUNet f_ϕ and subsequently deformed with LBS to yield posed Gaussians \mathcal{G}_y^P . This projection strategy ensures that novel poses are smoothly interpolated within the training distribution, improving robustness and generalization.

7 REINFORCEMENT LEARNING BASED LOCOMOTION FOR NAVIGATING 3DGS SCENES

We now give the detailed instantiation of the RL setup summarized in the method, specifying the policy architecture, observation design, reward terms, and synthetic training environments.

Policy and actions. We train locomotion, following Zhao et al. (2023), using PPO with an actor-critic architecture; both networks are 4-layer MLPs (width 512) with residual connections. The policy’s output corresponds to the diffusion start-noise $\mathbf{z}_{\text{RL},i}^{(\tau_{\max})}$ in the latent space of our adopted motion model, parameterized as

$$\mathbf{z}_{\text{RL},i}^{(\tau_{\max})} = 4 \tanh(\mathbf{z}_{\text{raw},i}^{(\tau_{\max})}) \in [-4, 4]^{d_z}, \quad (8)$$

where $\mathbf{z}_{\text{raw},i}^{(\tau_{\max})}$ is the unconstrained vector output of the policy network before squashing. This bounded parameterization stabilizes exploration and ensures consistency with the pretrained denoiser-decoder $(\mathcal{G}, \mathcal{D})$, which maps $\mathbf{z}_{\text{RL},i}^{(\tau_{\max})}$ into short motion clips.

Observations. At each step the agent receives motion history \mathbf{H}_i , a goal cue (clamped to a 5 m egocentric horizon and 120° field of view), a fixed text cue “walk,” and a body-centric *egocentric occupancy grid* $\mathcal{M} \in \{0, 1\}^{16 \times 16}$ covering a 1.6×1.6 m region around the pelvis. Each cell encodes walkability: $\mathcal{M}(u) = 1$ if the cell is traversable, and $\mathcal{M}(u) = 0$ if blocked by obstacles. During training in synthetic mesh-based rooms, occupancy is computed from the navigation mesh and object collisions; during deployment in 3DGS, it is derived from projected, opacity-thresholded Gaussians as described in the Sec 3 of the paper.

Rewards in synthetic training scenes. Let $p_i \in \mathbb{R}^3$ denote the pelvis and $g \in \mathbb{R}^3$ the goal. With planar distance $d_i = \|(p_i - g)_{xy}\|_2$, we shape progress and success as

$$r_{\text{dist}}^{\text{prog}} = d_{i-1} - d_i, \quad r_{\text{succ}} = \mathbf{1}[d_i < 0.3]. \quad (9)$$

Orientation alignment is encouraged via

$$r_{\text{ori}} = \frac{1}{2} (\langle p_i - p_{i-1}, g - p_{i-1} \rangle + 1), \quad (10)$$

while kinematic plausibility is promoted by penalizing foot skating and floor contact errors:

$$r_{\text{skate}} = -\text{disp}\left(2 - 2\frac{h}{0.03}\right), \quad r_{\text{floor}} = -(|l_f| - 0.03)_+. \quad (11)$$

Obstacle avoidance is realized by discouraging overlap between the occupancy grid \mathcal{M} and the body footprint $\mathcal{B}_{xy}(X)$:

$$r_{\text{pene}} = \exp\left(-|\mathcal{M}_0 \cap \mathcal{B}_{xy}(X)|\right), \quad (12)$$

where \mathcal{M}_0 are non-walkable cells.

The total reward is

$$r_i = w_{\text{dist}} r_{\text{dist}}^{\text{prog}} + w_{\text{succ}} r_{\text{succ}} + w_{\text{ori}} r_{\text{ori}} + w_{\text{skate}} r_{\text{skate}} + w_{\text{floor}} r_{\text{floor}} + w_{\text{pene}} r_{\text{pene}}. \quad (13)$$

Training environments. Training for RL-based locomotion follows Zhao et al. (2023). We procedurally generate rectangular rooms (edge 2–7 m) filled with ShapeNet furniture (chairs, beds, sofas, desks, tables) with z-up alignment and expanded margins to preserve traversability. Navigation meshes yield collision-free start/goal pairs and waypoint paths, and randomized initial poses/headings encourage robustness. These synthetic environments, together with the reward terms in equation 9–equation 13, teach policies to reach goals while avoiding obstacles and maintaining physically plausible motion.

8 ROLLOUT

Here we describe the rollout algorithm originally introduced in, Zhao et al. (2025) which we use for our transition synthesis framework.

REFERENCES

- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>.
- Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. *arXiv preprint arXiv:2311.16096*, 2023.
- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, October 2015.

Algorithm 1 Autoregressive rollout generation using latent motion primitive model

Require: Primitive decoder \mathcal{D} , latent denoiser \mathcal{G} , motion seed \mathbf{H}_{seed} , text prompts $\mathbf{C} = [c_1, \dots, c_N]$, diffusion steps τ_{max} , classifier-free guidance scale w , sampler S Latent noises $\mathbf{Z}_{\tau_{\text{max}}} = [z_{\tau_{\text{max}}}^1, \dots, z_{\tau_{\text{max}}}^N]$

Ensure: Motion sequence \mathbf{M}

```

0:  $\mathbf{H} \leftarrow \mathbf{H}_{\text{seed}}, \mathbf{M} \leftarrow \mathbf{H}_{\text{seed}}$ 
0: for  $i \leftarrow 1$  to  $N$  do {number of motion primitives}
0:   if latent noise not provided then
0:     Sample  $z_{\tau_{\text{max}}}^i \sim \mathcal{N}(0, I)$ 
0:   else
0:     Use provided  $z_{\tau_{\text{max}}}^i$ 
0:   end if
0:    $\hat{z}_0^i \leftarrow S(\mathcal{G}, z_{\tau_{\text{max}}}^i, \tau_{\text{max}}, \mathbf{H}, c_i, w)$  {DDIM/DDPM denoising with guidance}
0:    $\hat{X} \leftarrow \mathcal{D}(\mathbf{H}, \hat{z}_0^i)$  {decode latent to motion primitive}
0:    $\mathbf{M} \leftarrow \text{CONCAT}(\mathbf{M}, \hat{X})$ 
0:    $\mathbf{H} \leftarrow \text{CANONICALIZE}(\hat{X}_{F-H+1:F})$  {update history with last  $H$  frames}
0: end for
0: return  $\mathbf{M}$ 

```

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)*, pp. 10975–10985, 2019.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018.

Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. DIMOS: Synthesizing diverse human motions in 3d indoor scenes. In *International conference on computer vision (ICCV)*, 2023.

Kaifeng Zhao, Gen Li, and Siyu Tang. DartControl: A diffusion-based autoregressive motion model for real-time text-driven motion control. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pp. 371–378, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 158113374X. doi: 10.1145/383259.383300. URL <https://doi.org/10.1145/383259.383300>.