AHA! ANIMATING HUMAN AVATARS AND DYNAMIC OBJECTS IN 3D SCENES WITH GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review



Figure 1: **Animating avatars and dynamic objects in 3DGS scenes.** We introduce a unified Gaussian-based representation for humans, objects, and environments, enabling synthesis of photorealistic human–scene and dynamic human–object interactions entirely within the Gaussian Splatting framework.

ABSTRACT

We present AHA!, a framework that animates human avatars navigating and interacting with dynamic objects in 3D scenes, with 3D Gaussian Splatting (3DGS) as the only 3D representation for rendering. Existing animation pipelines rely on meshes or point clouds for scenes and humans. These provide the surfaces, distance fields, and walkable regions that motion synthesis and contact modeling need, but they do not provide photorealistic renderings and are hard to recover from casually captured video. 3DGS offers photorealism and reconstruction from a single mobile-phone video, yet provides none of these geometric quantities. We close this gap with four components that operate directly on opacity-filtered Gaussians: occupancy maps for path planning, an egocentric walkability grid that lets a locomotion policy trained only in synthetic mesh rooms run zero-shot in 3DGS scenes, a differentiable SDF proxy for synthesizing scene interactions, and a differentiable contact refinement for human–scene, human–object, and object–scene contact. Because maps and contact targets are functions of the current object poses, the same stack composes long animations in which an avatar walks through a scene, sits down, lifts objects, carries them, and puts them down, re-planning as its own actions change the layout. On SuperSplat and DL3DV scenes, human and VLM preference studies favor our renderings over the strongest mesh pipelines, and our motion quality is within 5% of a clean-mesh upper bound. Full-resolution videos are available at `https://anonymous.4open.science/w/AHAICLR27/`.

1 INTRODUCTION

Human animation in 3D scenes is essential for applications ranging from video gaming and computer-generated imagery (CGI) to robotics. Recent research has made significant progress on generating humans in 3D scenes (Hassan et al., 2019; Jiang et al., 2024; Hassan et al., 2021a; Zhao et al., 2023; Hwang et al., 2025), where humans are typically represented as either 3D skeletons or meshes, while background scenes are represented as meshes or point clouds. These representations are compact and versatile, capable of modeling a wide variety of surfaces. However, a fundamental limitation persists: there is almost always a domain gap between rendered results and real images.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In parallel, neural scene representations such as NeRF (Mildenhall et al., 2020) and 3DGS (Kerbl et al., 2023) have emerged; they render scenes, humans, and objects photorealistically from novel viewpoints and in novel poses, yet have seen little adoption in animation. 3DGS as the sole representation for rendering in animation has natural advantages: renderings of humans interacting with their surroundings are more photorealistic, and a 3DGS scene can be reconstructed from a single casually captured mobile-phone video (Ling et al., 2024), whereas recovering a usable mesh from such footage remains hard (Wang et al., 2025). The price is that 3DGS provides none of the geometric quantities that animation methods rely on. This paper asks two questions: *Can Gaussian Splatting replace meshes across the human–scene animation stack, from navigation and locomotion to interaction and contact? Does the same Gaussian-only stack extend to dynamic objects that the avatar lifts, carries, and puts down, so that scene, human, and object are all Gaussians?* (Fig. 1).

Two obstacles stand in the way. The first is data: human–scene motion datasets (Hassan et al., 2019; Jiang et al., 2024; Bhatnagar et al., 2022) provide motion in mesh form only, and no paired motion–scene data exists in 3DGS. We sidestep this by decoupling rendering from motion synthesis: scenes, humans, and objects are reconstructed as Gaussians independently, motion is synthesized on the SMPL body (Loper et al., 2015), and Gaussians are assembled only at render time (Sec. 3). The second is geometry. Every stage of an animation stack queries the scene, for occupancy when planning, walkable ground when stepping, distance when reaching, and a surface when making contact, and a 3DGS reconstruction is a cloud of anisotropic primitives with no surface, no distance field, and floaters left in free space by reconstruction noise. The SMPL body serves as scaffolding for the avatar, a kinematic skeleton and a proxy surface for contact queries, but is never rendered; scenes and objects remain Gaussians, with no mesh ever extracted from them.

We address the second obstacle with four components that let a mesh-based animation stack run on Gaussians without retraining any of its learned parts (Fig. 2). Opacity culling and orthographic projection of the Gaussians yield occupancy maps for A* planning, free of the phantom obstacles that floaters would otherwise create (Sec. 3.1). An egocentric walkability grid computed from nearest-neighbor distances to Gaussian centers gives a reinforcement learning (RL) locomotion policy trained only in synthetic mesh rooms the one observation it needs, so the policy runs zero-shot in scenes reconstructed from a single phone video (Sec. 3.2). A differentiable SDF proxy, the nearest-neighbor distance to scene Gaussians minus their extent, lets scene interactions such as sitting be synthesized by latent-space optimization in the latent motion model (Sec. 3.3). A soft version of the same distance supplies gradients for contact: a 3DGS avatar maps pose to Gaussians through a neural network that is intractable to optimize through and offers no surface to snap to, so we instead optimize per-Gaussian offsets that pull contact Gaussians onto the target and keep the rest a margin away (Sec. 3.4). On the project page we also show that our framework works with multiple avatars animated in the same scene.

Dynamic objects raise the bar. Once the avatar can lift a chair and put it down elsewhere, the layout changes with every interaction, so nothing about the scene can be precomputed. Because an object is a Gaussian set like the scene, the same components apply: occupancy maps and walkability grids are recomputed *on the fly* from the current object poses (Sec. 3.1), and contact refinement runs against the dynamic objects. For dynamic interaction synthesis, we follow a retrieval-based approach: we retrieve interaction clips, place them in the scene, and chain them with locomotion through motion inbetweening, updating the map after every interaction (Sec. 3.5).

We evaluate on ScanNet++ (Yeshwanth et al., 2023), SuperSplat (SuperSplat, 2026), and DL3DV (Ling et al., 2024) scenes, spanning indoor scans, artist-made environments, and casually captured outdoor video, with avatars and objects reconstructed from multi-view capture. Human and VLM preference studies favor our rendering quality over the strongest mesh pipelines, including one that extracts meshes from the very same Gaussians; our motion quality lands within 5% of a clean-mesh motion upper bound; and ablations confirm the contribution of each component. In summary, we make 3DGS the sole rendered representation for animating humans and dynamic objects, without extracting a mesh from scene or objects and without human–scene paired data, through five contributions: (i) occupancy maps for planning from opacity-filtered Gaussians, (ii) zero-shot transfer of a mesh-trained RL locomotion policy through an egocentric walkability grid, (iii) interaction synthesis with a differentiable SDF proxy, (iv) differentiable contact refinement for human–scene, human–object, and object–scene contact, and (v) composition of long animations in changing scenes through dynamic maps and retrieval-based chaining.

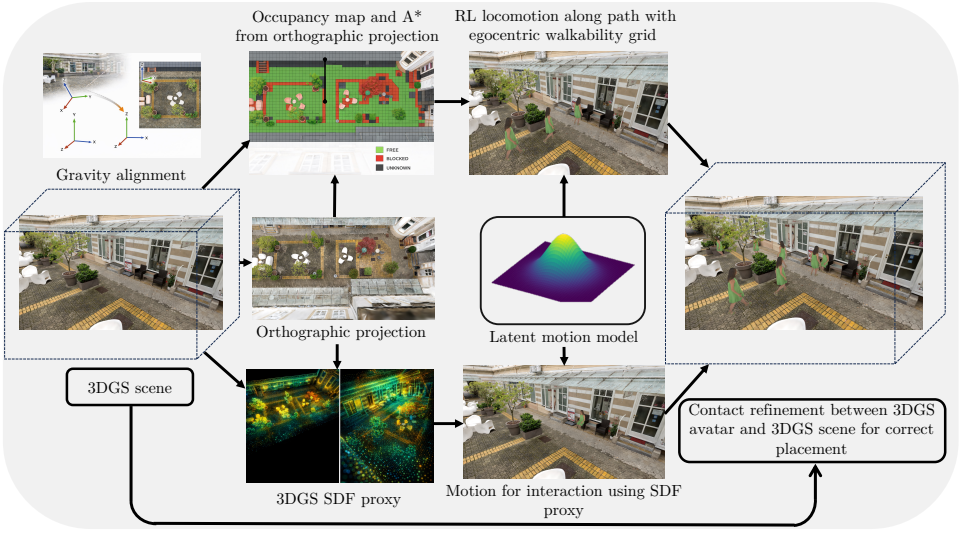

Figure 2: **Method overview for static interactions.** Given 3DGS reconstructions of a scene, an avatar, and objects, we (1) build occupancy maps from opacity-filtered Gaussians and plan paths (Sec. 3.1); (2) synthesize locomotion with a locomotion policy that reads an egocentric walkability grid (Sec. 3.2); (3) synthesize scene interactions by latent optimization against a differentiable SDF proxy (Sec. 3.3); (4) refine human–scene, human–object, and object–scene contact in Gaussian space (Sec. 3.4). All stages query and render Gaussians only.

2 Related Work

Neural Rendering of Scenes and Humans. NeRF (Mildenhall et al., 2020) and its variants (Müller et al., 2022; Barron et al., 2023; Xie et al., 2022b) render photorealistic novel views but are slow. 3DGS (Kerbl et al., 2023) achieves real-time rendering with explicit primitives and splatting (Westover, 1991), and has been extended to dynamic scenes (Luiten et al., 2024; Wu et al., 2024; Li et al., 2024b), SLAM (Keetha et al., 2024), mesh extraction (Huang et al., 2024; Guédon & Lepetit, 2024), and sparse views (Mihajlovic et al., 2024). For humans, mesh templates (Loper et al., 2015; Pavlakos et al., 2019), implicit functions (Chen et al., 2021; Saito et al., 2020; Alldieck et al., 2021), and monocular methods (Alldieck et al., 2018; Mir et al., 2020) recover shape but lack photorealism or cannot be reposed. NeRF avatars (Peng et al., 2021; Weng et al., 2022; Guo et al., 2023; Zhu et al., 2024) and 3DGS avatars (Kocabas et al., 2024; Mir et al., 2026; Qian et al., 2024; Moreau et al., 2024; Li et al., 2024c; Pang et al., 2024; Hu et al., 2024; Moon et al., 2024; Junkawitsch et al., 2025) are controllable and photorealistic, but animate humans in isolation. Extensions that model humans with their surroundings (Xue et al., 2024; Zhan et al., 2025) assume a static scene or treat the object as an extension of the hand. Mir et al. (2025) reconstruct separate animatable human and object Gaussians from interaction captures; we build on such reconstructions but address the orthogonal problem of animating them in 3D scenes.

Humans, Objects, and Scenes. Datasets of humans in scenes and with objects (Hassan et al., 2021a; Guzov et al., 2021; Hassan et al., 2019; Savva et al., 2016; Taheri et al., 2020; Bhatnagar et al., 2022; Jiang et al., 2024; Zhang et al., 2022) have driven joint human–object reconstruction (Xie et al., 2022a; 2023; Zhang et al., 2020) and object- and scene-conditioned motion synthesis (Starke et al., 2019; Hassan et al., 2021b; Mir et al., 2024; Zhao et al., 2023; Hwang et al., 2025), including generative human–object interaction models (Li et al., 2024a; Diller & Dai, 2024; Jiang et al., 2024) that synthesize manipulation conditioned on object and scene meshes. All represent humans, objects, and scenes as meshes and thus cannot operate on raw Gaussians and cannot render photorealistically.

Human Animation. Classical approaches include motion matching (Clavet, 2016; Holden et al., 2020), motion graphs (Kovar et al., 2002; Safonova & Hodgins, 2007), and neural controllers (Holden et al., 2017; Starke et al., 2020). Diffusion models (Tevet et al., 2023) extend to physics (Yuan et al., 2023) and fine-grained control (Karunratanakul et al., 2024; Xie et al., 2024; Wang et al., 2026). Reinforcement learning (Zhang & Tang, 2022; Zhao et al., 2023) is widely used for navigation, and latent motion models (Zhao et al., 2025; 2026) combine RL with latent-space optimization. All operate on clean mesh scenes. We adapt this paradigm to 3DGS.

3 METHOD

Problem Statement. Given a 3DGS scene, a 3DGS avatar, 3DGS objects with captured interaction clips, and a user script of L scripted locations $\mathbf{p}_\ell \in \mathbb{R}^3$ with action labels a_ℓ (e.g., "sit", "reach", "lift", "put down"), we produce for every frame t posed human and object Gaussians that take the avatar from a start to a goal position through the scripted interactions, avoid collisions, make contact where the motion implies it, and composite with the scene for rendering from any viewpoint.

Overview. In Secs. 3.1 to 3.4 we build the animation stack for a static scene (Fig. 2), and in Sec. 3.5 we compose these stages into long animations in scenes whose layout changes as objects are moved (Fig. 3). Every component operates directly on the Gaussians, and the learned components transfer zero-shot to 3DGS: the locomotion policy is trained only in synthetic mesh rooms, so the transfer to Gaussian scenes rests entirely on the geometric structures we extract from them.

Representations. We represent scenes, humans, and objects as independent Gaussian sets that are composed only at render time, so no paired human–scene data in 3DGS form is needed: motion is synthesized on the SMPL body and mapped to Gaussians afterward. A static scene is $\mathcal{G}^S = \{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i, \mathbf{c}_i)\}_{i=1}^{N_S}$ with centers $\boldsymbol{\mu}_i \in \mathbb{R}^3$, covariances $\boldsymbol{\Sigma}_i$ parameterized by scale vectors $\mathbf{s}_i \in \mathbb{R}^3$ and rotations $\mathbf{q}_i \in \mathbb{R}^4$, opacities $\alpha_i \in (0, 1)$, and colors $\mathbf{c}_i$ (Kerbl et al., 2023). An avatar is a network f_ζ with weights ζ (Li et al., 2024c) that maps SMPL shape $\boldsymbol{\beta}$ and pose $\boldsymbol{\theta}_t$ at frame t to canonical Gaussians $\mathcal{G}_t^{H,\mathsf{C}}$ with centers $\boldsymbol{\mu}_k^{\mathsf{C}}$, $k = 1, \ldots, N_H$, which linear blend skinning (LBS) turns into posed Gaussians $\mathcal{G}_t^{H,\mathsf{P}}$ with centers $\boldsymbol{\mu}_{k,t}^{\mathsf{P}}$; because $\boldsymbol{\theta}_t \mapsto \mathcal{G}_t^{H,\mathsf{P}}$ passes through f_ζ, optimizing pose for contact is intractable (Sec. 3.4). A rigid object (one per interaction) is a canonical set $\mathcal{G}^O$ with centers $\boldsymbol{\mu}_p$, $p = 1, \ldots, N_O$, placed by a 6D pose $\boldsymbol{\phi}_t$ through the rigid transform $\mathcal{T}$ as $\mathcal{G}_t^{O,\mathsf{P}} = \mathcal{T}(\mathcal{G}^O; \boldsymbol{\phi}_t)$; objects and tracked interaction clips of paired poses $(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$ come from Mir et al. (2025). Reconstruction, rendering, skinning, and pose projection details are in Apps. F to J.

3.1 OCCUPANCY MAPS FOR PLANNING

Gravity Alignment. We first align the scene to a canonical axis. We compute PCA of the scene centers $\{\boldsymbol{\mu}_i\}$. Real scenes extend further horizontally than vertically, so the smallest principal axis is taken as the gravity direction and the other two span the ground plane. The resulting rotation $\mathbf{R}_{\text{align}}$ brings the scene into a canonical top-down view without camera metadata or manual annotation; the floor height is a low percentile of the vertical coordinates (App. O).

Opacity Culling. A naive orthographic projection of all Gaussians yields a map cluttered by floaters, low-opacity Gaussians produced by reconstruction noise that do not correspond to surfaces. We retain only Gaussians whose opacity exceeds a threshold, $\alpha_i > \tau_\alpha$ (Fig. 8, App. A).

Occupancy Map. The retained scene Gaussians and every object at its current pose $\boldsymbol{\phi}_t$ are projected orthographically along the gravity axis onto an $M \times M$ grid, giving a binary occupancy map $\mathcal{O}_t \in \{0, 1\}^{M \times M}$ in which a cell is marked as an obstacle (value 1) if its projected Gaussian density exceeds a threshold. Since scene and objects are separate Gaussian sets, the map decomposes as

$$\mathcal{O}_t = \mathcal{O}^S \vee \mathcal{O}_t^O, \tag{1}$$

where the scene term is computed once, and the object term is re-projected and dilated by a small safety margin whenever an object pose changes (Fig. 3, left). We refer to $\mathcal{O}_t$ as the dynamic map. A* (Hart et al., 1968) on $\mathcal{O}_t$ yields collision-free waypoints from the current position to the next scripted location. Because the map is a function of the current object poses, planning remains valid after the avatar has moved an object (Fig. 5); a static map would route the avatar through it.

3.2 GAUSSIAN-ALIGNED LOCOMOTION

Locomotion follows the A* waypoints with an RL locomotion policy that acts in the latent space of a latent motion model, DartControl (Zhao et al., 2025), whose denoiser $\mathcal{F}$ turns a latent noise sample into a motion primitive, a short clip of F frames, given the motion history $\mathbf{H}$ and a text embedding $\mathbf{e}$ (App. K). Existing scene-aware policies (Zhao et al., 2023; Hassan et al., 2021a) are trained in synthetic mesh scenes; our insight is to replace this mesh interface with a grid computed from Gaussians.

Egocentric Walkability Grid. At policy step n, we build an $N_g \times N_g$ grid $\mathcal{W}_n$ centered on the pelvis and aligned with the heading. A cell u is walkable, $\mathcal{W}_n(u) = 1$, if the planar distance from

its center to the nearest retained scene or posed object Gaussian center exceeds a threshold τ_{occ}, and blocked otherwise. During training in synthetic rooms, the same grid is filled from the navigation mesh; at deployment, it is filled from Gaussians. The grid is the only channel through which the policy sees the scene, which is what makes the transfer from meshes to 3DGS possible.

Policy. The policy π_ω with parameters ω observes

$$o_n = (\mathbf{H}_n,\ \mathbf{g}_n,\ \mathcal{W}_n,\ \mathbf{e}_n), \tag{2}$$

the motion history $\mathbf{H}_n$, a goal cue toward the next waypoint $\mathbf{g}_n$, the walkability grid $\mathcal{W}_n$, and a text embedding $\mathbf{e}_n$ ("walk" by default), and outputs the initial noise $\mathbf{z}_{\text{RL},n}$ of the denoising chain, which the frozen denoiser maps to the clean latent and the next motion primitive,

$$\widehat{\mathbf{z}}_{0,n} = \mathcal{F}(\mathbf{z}_{m,n},\, m,\, \mathbf{H}_n,\, \mathbf{e}_n;\, w_{\text{cfg}}), \qquad \mathbf{z}_{m_{\max},n} = \mathbf{z}_{\text{RL},n}, \tag{3}$$

where m is the diffusion step, iterated from $m_{\max}$ down to 1 with $m_{\max}$ the number of DDIM steps, and w_{cfg} is the classifier-free guidance scale (App. P). The denoiser is scene- and goal-agnostic; the goal and the grid shape the motion only through the policy's choice of $\mathbf{z}_{\text{RL},n}$. Rewards encourage waypoint progress, obstacle avoidance, and kinematic plausibility (App. K).

3.3 Interaction Synthesis

When locomotion reaches a scripted location, control passes to interaction synthesis. Interactions with the static scene (sitting, reaching) are synthesized by latent-space optimization against a differentiable proxy for the scene's SDF; interactions with dynamic objects are treated in Sec. 3.5.

Differentiable SDF Proxy. 3DGS has no signed distance field. We approximate one by the distance from a query point $\mathbf{x}$ to the nearest retained Gaussian center of an entity $\Upsilon \in \{S, O\}$, minus that Gaussian's extent. Writing $\boldsymbol{\mu}_j^\Upsilon$ for the retained centers of Υ, that is $\{\boldsymbol{\mu}_i\}$ for the scene and $\{\boldsymbol{\mu}_p\}$ for the object,

$$\rho(\mathbf{x}; \Upsilon) = \|\mathbf{x} - \boldsymbol{\mu}_{j^*}^\Upsilon\| - r_{j^*}, \qquad j^* = \arg\min_j \|\mathbf{x} - \boldsymbol{\mu}_j^\Upsilon\|, \qquad r_j = \|\mathbf{s}_j^\Upsilon\|_\infty, \tag{4}$$

where r_j, the largest entry of the scale vector $\mathbf{s}_j^\Upsilon$, is the effective radius of Gaussian j. Positive values indicate free space and negative values penetration; the proxy is differentiable almost everywhere.

Scene Interactions by Latent Optimization. With a deterministic DDIM sampler, the latent motion model maps an initial noise $\mathbf{z}_{m_{\max}}$ to a motion primitive $\widehat{\mathbf{X}} = (\widehat{\mathbf{X}}_1, \ldots, \widehat{\mathbf{X}}_F)$ of F frames given a seed history and a text cue such as "sit" (App. L). Starting from the locomotion end state $\widehat{\mathbf{X}}^{\text{loc}}$, we optimize $\mathbf{z}_{m_{\max}}$ by gradient descent on

$$\mathcal{L} = \mathcal{L}_{\text{reach}} + \lambda_{\text{stop}}\mathcal{L}_{\text{stop}} + \lambda_{\text{start}}\mathcal{L}_{\text{start}} + \lambda_{\text{coll}}\mathcal{L}_{\text{coll}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}}, \tag{5}$$

where $\mathcal{L}_{\text{reach}}$ and $\mathcal{L}_{\text{stop}}$ are the squared distance of a task-specific anchor joint (pelvis for sitting, hand for reaching) to the scripted location $\mathbf{p}_\ell$ and its squared velocity at the final frame, $\mathcal{L}_{\text{start}}$ pins the first frame to $\widehat{\mathbf{X}}^{\text{loc}}$, $\mathcal{L}_{\text{coll}} = \sum_{\mathbf{b}\in\mathcal{B}_F}[-\rho(\mathbf{b}; S)]_+^2$ with $[\cdot]_+ = \max(0, \cdot)$ penalizes penetration of the body surface points $\mathcal{B}_F$ sampled from SMPL at the final frame, $\mathcal{L}_{\text{smooth}}$ penalizes frame-to-frame differences, and the λ are scalar weights (App. P). The scripted location may lie at any height, so the same objective covers sitting on a chair and reaching for a shelf. After the action, the same formulation synthesizes the exit transition (e.g., sit to stand).

3.4 Differentiable Contact Refinement

Motion synthesis (Secs. 3.2 and 3.3) yields SMPL poses $\{\boldsymbol{\theta}_t\}$, which are projected into the avatar's pose subspace (App. J), mapped through f_ζ, and skinned to $\mathcal{G}_t^{H,\mathsf{P}}$; objects are rigidly transformed to $\mathcal{G}_t^{O,\mathsf{P}}$. Compositing these with the scene directly produces, for example, floor penetration, floating feet (Fig. 3, right), and hands that miss the carried object, because the SMPL body used for synthesis and the reconstructed avatar differ in shape and height. With meshes one would optimize pose through LBS or project vertices onto surfaces; here pose optimization is intractable (Sec. 3) and there is no surface. We instead optimize per-Gaussian translation offsets $\mathbf{T}_{k,t}$ on a small set of contact Gaussians, pulling them onto a target entity Υ when contact is expected and keeping them a margin away otherwise. Three instances share one objective: human–scene contact ($\Upsilon = S$; feet and pelvis), human–object contact ($\Upsilon = O$; hands), and object–scene contact ($\Upsilon = S$; the object Gaussians closest to the scene, indexed by p in place of k, take the role of the human Gaussians).

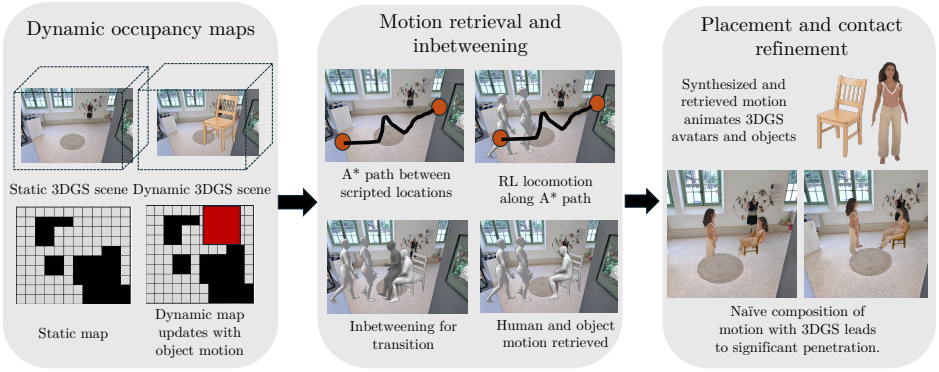

Figure 3: **Composing animations with dynamic objects. Left:** dynamic maps: object Gaussians are re-projected after every interaction and the path is re-planned on the updated map. **Middle:** retrieval and inbetweening: captured interaction clips are placed at the scripted locations and chained with locomotion. **Right:** placement and contact refinement: human and object Gaussians are composited with the scene and contact is enforced; without the SDF proxy and refinement, interactions penetrate the scene.

Contact Gaussians. Contact frames are detected from the synthesized SMPL motion by thresholding the vertical velocity and acceleration of each contact joint, giving an indicator $\delta_{c,t} \in \{0, 1\}$ for each contact joint c (feet, pelvis, hands; App. P). 3DGS avatars have no fixed template correspondence, so we map each SMPL contact vertex $\mathbf{m}$ of joint c to its nearest canonical Gaussian in the rest pose, $\arg\min_k \|\boldsymbol{\mu}_k^{\mathsf{C}} - \mathbf{m}\|$, once per avatar. This gives an index set $\mathcal{C}$ that adapts to any avatar regardless of its Gaussian count, and assigns every $k \in \mathcal{C}$ to a joint $c(k)$.

Soft Proximity. The hard proxy of Eq. 4 suffices for collision avoidance, but pulling Gaussians onto a surface needs smoother gradients. We use the soft nearest-neighbor distance to the retained centers of the target entity,

$$d_\kappa(\mathbf{x}; \Upsilon) = -\tfrac{1}{\kappa} \log \textstyle\sum_j \exp\big(-\kappa \|\mathbf{x} - \boldsymbol{\mu}_j^{\Upsilon}\|\big), \tag{6}$$

where $\kappa > 0$ controls the softness. Unlike ρ, d_κ measures distance to centers rather than to surfaces; since both the contact Gaussians and the target Gaussians lie on surfaces, coincident centers correspond to touching surfaces.

Objective. For each contact Gaussian $k \in \mathcal{C}$ at frame t, the refined center is $\tilde{\boldsymbol{\mu}}_{k,t}^{\mathsf{P}} = \boldsymbol{\mu}_{k,t}^{\mathsf{P}} + \mathbf{T}_{k,t}$. Below, $\mathbf{x} = \boldsymbol{\mu}_{k,t}^{\mathsf{P}} + \mathbf{T}$ is the displaced center, $\delta = \delta_{c(k),t}$ the contact indicator of its joint, and $\boldsymbol{\mu}_j^{\Upsilon}$ the target center nearest to the Gaussian at frame t. The offset minimizes four terms,

$$\mathbf{T}_{k,t} = \arg\min_{\mathbf{T}} \; \lambda_{\text{snap}}\, \delta \, \|\mathbf{x} - \boldsymbol{\mu}_j^{\Upsilon}\|^2 + \lambda_{\text{prox}}\, \psi(\mathbf{x}, \delta) + \lambda_{\text{reg}} \, \|\mathbf{T}\|^2 + \lambda_{\text{temp}} \, \|\mathbf{T} - \mathbf{T}_{k,t-1}\|^2, \tag{7}$$

$$\psi(\mathbf{x}, 1) = d_\kappa(\mathbf{x}; \Upsilon)^2, \qquad \psi(\mathbf{x}, 0) = \max\big(0,\, \epsilon - d_\kappa(\mathbf{x}; \Upsilon)\big)^2. \tag{8}$$

Intuitively, the first term snaps the Gaussian onto its nearest target center at contact frames. The second pulls it toward the target surface when in contact ($\delta = 1$) and keeps it at least a margin $\epsilon > 0$ away otherwise ($\delta = 0$). The last two keep the offset small and smooth over time; the λ terms are scalar weights (App. P). When the target moves, as a carried object does, its centers are updated every frame before the objective is evaluated. Refined human, object, and scene Gaussians are composited and rendered with the standard 3DGS rasterizer.

3.5 Animating Dynamic Objects: Retrieval and Chaining

Object Interactions by Retrieval. Interactions with dynamic objects (lifting, carrying, putting down, pushing) are not synthesized but retrieved from a corpus of captured interaction clips (App. M), each short and covering a single interaction. For a scripted location $\mathbf{p}_\ell$ with an object action label, we retrieve a clip $\mathcal{K}_\ell$ of paired human and object poses $(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$ with a matching label and place it by a rigid transform that aligns its start to $\mathbf{p}_\ell$ and its floor to the scene floor (App. N). The object follows its tracked poses during the clip. Once lifted, it is attached to the avatar's wrist and follows SMPL forward kinematics until it is put down, after which it is static at its new pose.

Composition in Changing Scenes. Scene interactions, object interactions, and locomotion are composed into one loop, illustrated in Fig. 3 and given as an algorithm in App. L. For each scripted location $\ell = 1, \ldots, L$ we (i) plan a path on the current occupancy map $\mathcal{O}_t$ of Eq. 1 and generate

Table 1: **Motion quality against mesh pipelines.** Baseline A plans on a mesh extracted from the same Gaussians with MiLo, Baseline B on a VGGT mesh. The ScanNet++ clean-mesh row is an upper bound; shading marks our method. Metrics in App. A.

Scenes	Method	Skat.↓	Pen. (mm)↓	Jerk↓	Goal↑
DL3DV	Bas. A (MiLo)	2.62	78	0.78	62.7
DL3DV	Bas. B (VGGT)	2.84	89	0.83	58.6
DL3DV	**Ours**	**0.92**	**14**	**0.49**	**93.8**
SuperSplat	Bas. A (MiLo)	2.23	66	0.66	72.1
SuperSplat	**Ours**	**0.87**	**14**	**0.46**	**95.6**
ScanNet++	Clean mesh	0.85	11	0.45	96.5
ScanNet++	**Ours**	**0.89**	**12**	**0.47**	**95.2**

Table 2: **Rendering preference (win rate, %).** Fraction of judgments preferring our 3DGS renderings over each mesh baseline. Baseline C renders the same scene reconstruction, avatar, and motion as ours with a mesh renderer.

Evaluator	Bas. C↑ same constit.	Bas. D↑ HQ vs HQ	Bas. B↑ mono.
Human study	86.4	82.1	72.9
GPT-5.6	88.1	76.1	81.3
Gemini 3.1	84.1	72.9	78.2

locomotion with the policy of Sec. 3.2; (ii) synthesize the interaction (Sec. 3.3) or retrieve it as above; and (iii) if the interaction moved an object, update ϕ_t and hence $\mathcal{O}_t^O$ before planning the next path. Locomotion and interaction segments rarely meet in matching poses, so consecutive segments are bridged by inbetweening with the latent motion model, seeded with the last frames of one segment and targeted at the first frame of the next. The assembled motion is

$$\mathrm{loco}_0 \to \mathrm{in}_1 \to \mathcal{K}_1 \to \mathrm{out}_1 \to \mathrm{loco}_1 \to \cdots \to \mathrm{in}_L \to \mathcal{K}_L \to \mathrm{out}_L \to \mathrm{loco}_L, \tag{9}$$

where loco_ℓ walks from $\mathbf{p}_\ell$ to $\mathbf{p}_{\ell+1}$ (from the start position for $\ell = 0$ and to the goal for $\ell = L$), in_ℓ and out_ℓ are inbetweening transitions, and $\mathcal{K}_\ell$ denotes the ℓ-th interaction segment, a retrieved clip or a scene interaction synthesized as in Sec. 3.3. Contact refinement (Sec. 3.4) then runs over the assembled motion, with the human–object and object–scene instances active while an object is carried or put down. Every scene-aware locomotion method we are aware of (Hassan et al., 2021a; Zhao et al., 2023; Hwang et al., 2025) assumes a static layout. Because the occupancy map, the walkability grid, and the contact targets are all functions of the current object poses, the avatar routes around, and makes contact with, objects that its own earlier actions displaced.

4 Experiments

We evaluate along three axes. *Is animating on Gaussians necessary*, or does extracting a mesh from the Gaussians and running a mesh pipeline suffice (Sec. 4.1)? *Rendering quality*: does 3DGS produce more photorealistic renderings than mesh pipelines (Sec. 4.2)? *Ablations*: what does each component contribute, for static scenes and for dynamic objects (Sec. 4.3)? In the appendix we provide metric definitions and ablation visualizations (App. A), interaction-specific metrics, contact refinement statistics, and the SDF proxy stress test (App. B), detailed baseline results and explanations (App. C), evaluation protocol validation (App. D), and additional results (App. E).

Data. Scenes come from ScanNet++ (Yeshwanth et al., 2023), where both 3DGS and mesh reconstructions are available, the SuperSplat library (SuperSplat, 2026), and DL3DV (Ling et al., 2024) mobile-phone videos. Avatars are reconstructed from ActorsHQ (Işık et al., 2023) (48 cameras), AvatarReX (Zheng et al., 2023), DNA-Rendering (Cheng et al., 2023), and BEHAVE (Bhatnagar et al., 2022) (4 cameras). Objects and interaction clips are reconstructed from BEHAVE, DNA-Rendering, and NeuralDome (Zhang et al., 2023) captures following Mir et al. (2025).

4.1 Is Animating on Gaussians Necessary?

The most direct test of our premise is whether one could simply extract a mesh from the scene Gaussians and hand it to an existing mesh pipeline. *Baseline A (extracted mesh)* does exactly this: the scene mesh is extracted from our own scene Gaussians with MiLo (Guédon et al., 2025), motion is synthesized on it with DIMOS (Zhao et al., 2023), and the result is rendered with the 3DGS scene. *Baseline B (VGGT mesh)* instead reconstructs each DL3DV video as a mesh with the feed-forward VGGT (Wang et al., 2025) and runs the same DIMOS pipeline; both see the same frames as our 3DGS reconstruction, so the asymmetry is feed-forward mesh extraction versus photometric 3DGS optimization. Both baselines yield artifacts. Construction details and failure cases are in App. C.

Tab. 1 evaluates the synthesized motion against mesh geometry (VGGT meshes on DL3DV, approximate SuGaR meshes on SuperSplat). Both mesh pipelines inherit reconstruction artifacts (Figs. 9 and 10). On DL3DV they show about $3\times$ higher skating and 5 to $6\times$ higher penetration than our

Table 3: **Ablation of navigation and locomotion on ScanNet++.** Motion is synthesized on 3DGS and evaluated against the mesh.

Method	Skat.↓	Pen. (mm)↓	Jerk↓	Goal↑
Ours (full)	**0.89**	**12**	**0.47**	**95.2**
w/o opacity culling	1.45	42	0.78	42.6
w/o walkability grid	1.37	21	0.71	84.5

Table 4: **Ablation of object interaction and chaining.** Penetration and contact distance; the user study reports the preference for the full method over each ablation.

Method	Pen. (mm)↓	Contact (mm)↓	Full pref. (%)↑
Ours (full)	**1.8**	**8.2**	–
w/o contact refin.	2.1	23.7	79.2
w/o dynamic maps	6.9	14.8	85.6
w/o inbetweening	1.9	8.6	72.4



Figure 4: **Effect of the egocentric walkability grid. Left:** without it the policy walks through the wall. **Right:** with the grid extracted from scene Gaussians the avatar avoids the obstacle.



Figure 5: **Dynamic maps.** Without dynamically computing occupancy maps the avatar hits placed objects.

method; on SuperSplat, where only Baseline A applies, the gaps are $2.5\times$ and $5\times$. On ScanNet++ we also run the full pipeline on the clean ground-truth mesh as an upper bound: our Gaussian-space pipeline lands within 5% of this ceiling on all four metrics, so little motion quality is sacrificed.

4.2 Rendering Evaluation

Baselines. *Baseline C (same constituents).* To evaluate rendering quality, we render our own result with a mesh pipeline: the scene Gaussians are meshed with MiLo, our 3DGS avatar is converted to a mesh with SuGaR (Guédon & Lepetit, 2024), and scene and avatar are rendered with mesh rasterization, so that only the renderer differs from ours (App. C). *Baseline D (cross-scene, HQ vs HQ)* is the highest-quality mesh pipeline we could assemble: Replica (Straub et al., 2019) scenes with a RenderPeople (RenderPeople, 2026) scan animated by DIMOS, against our best 3DGS setting (SuperSplat scenes, AvatarReX avatars). A within-scene comparison is impossible here because the two modalities require different capture setups. We also include Baseline B of Sec. 4.1, with its avatar naively composited into the 3DGS scene without our walkability, interaction, or refinement modules, as a within-scene monocular comparison (App. C).

Protocols. *Protocol I (human study):* 21 participants (19 for Baseline C) perform pairwise forced-choice judgments on 5 video pairs per baseline, selecting the more photorealistic rendering. *Protocol II (VLM study):* GPT-5.6 (OpenAI, 2026) and Gemini 3.1 (Google DeepMind, 2026) judge 10 matched frame pairs per video. We validate the VLM judges on synthetic-versus-real pairs, order flips, inter-VLM agreement, and agreement with human consensus in App. D.

Results. Tab. 2 reports win rates. Against the same-constituent pipeline (Baseline C), participants prefer our renderings 86.4% of the time, so the preference is attributable to the representation rather than to content (Fig. 11). Against the best mesh pipeline (Baseline D) the preference is 82.1%, and on the within-scene monocular comparison (Baseline B) 72.9%; there the mesh baseline suffers from noisy VGGT reconstructions that block paths and cause penetration. Both VLM judges prefer our renderings on every comparison, in 72.9% to 88.1% of pairs.

4.3 Ablations

Navigation and Locomotion. Tab. 3 isolates each component on ScanNet++, with metrics evaluated against the mesh. Opacity culling has the largest effect: without it, floaters create phantom obstacles that cut goal reaching from 95.2% to 42.6% and more than triple penetration. Removing the walkability grid lowers goal reaching to 84.5%, since the policy loses local obstacle awareness and attempts paths through walls (Fig. 4). Contact refinement only offsets Gaussians at render time and leaves all four metrics, unchanged. We measure its effect in Gaussian space in Tab. 4.

Object Interaction and Chaining. Tab. 4 ablates the components specific to dynamic objects on chained sequences in SuperSplat scenes, using penetration depth and contact distance measured in Gaussian space and a user study with 13 raters on 7 video pairs. Replacing dynamic maps with the



Figure 6: **Qualitative results. Left:** avatars interacting with dynamic objects in a 3DGS scene. **Right:** navigation and scene interaction in a static SuperSplat scene.



Figure 7: **Free-viewpoint rendering of monocular videos with inserted animated avatars.** Each scene is a DL3DV video reconstructed as 3DGS; the avatar was never in the footage and is composited in Gaussian space.

static map increases penetration depth almost fourfold, as the avatar walks through objects placed by earlier interactions (Fig. 5). Disabling contact refinement triples the contact distance between hands and carried objects. Removing inbetweening barely changes the geometric metrics but is clearly visible to raters, who prefer the full method 72.4% of the time. Interaction-specific accuracy and penetration for synthesized sit and reach actions, offset statistics of the refinement, and a stress test of the SDF proxy under sparse reconstructions are reported in App. B.

4.4 Qualitative Results

Fig. 6 shows results across SuperSplat scenes with avatars interacting with static scenes or with dynamic objects in the scene. On the left, avatars interact with a table; on the right, avatars navigate and interact with static scenes. Fig. 7 shows free-viewpoint rendering of DL3DV mobile-phone videos edited with animated avatars that were never in the footage, a capability unique to the 3DGS representation. Additional video results are on the project page.

5 Conclusion

We have presented AHA!, to our knowledge the first method that animates human avatars and dynamic objects in diverse 3D environments with 3D Gaussian Splatting as the only 3D representation for rendering. Occupancy maps for planning, a walkability grid that transfers a mesh-trained locomotion policy to 3DGS scenes without retraining, a differentiable SDF proxy for interaction synthesis, and contact refinement all operate on opacity-filtered Gaussians, and they extend to dynamic objects as well. Our result is preferred over the strongest mesh pipelines in rendering quality, and matches a clean-mesh motion ceiling to within 5% in motion quality. As our method works on scenes reconstructed from mobile-phone video, it opens the door to free-viewpoint video editing and to gaming and AR/VR content in environments captured with consumer cameras. Our method has limitations as well: avatars and objects keep the lighting of their capture. We do not enforce physics, so interactions can look plausible yet violate stability or momentum. Object interactions are retrieved from a corpus of captured interaction clips rather than synthesized, since existing generative human–object models (Li et al., 2024a; Diller & Dai, 2024; Jiang et al., 2024) condition on object meshes, which our objects never have; the set of object motions is therefore bounded by what was captured, and objects are assumed rigid and the floor is assumed planar. Avatar and object reconstruction requires multi-view video. Future work includes relightable composition, learned physics priors, generative synthesis of object interactions conditioned on Gaussian objects through the SDF proxy, and avatar reconstruction from monocular video.

AI USE STATEMENT

We used AI-based tools in a supporting role only. A large language model assisted with language editing of author-written text and with routine code completion during implementation. The research question, method design, experimental protocol, and interpretation of results were produced by the authors without generative AI. Vision-language models (GPT-5.6 and Gemini 3.1) are used as an evaluation instrument in Sec. 4.2, with the protocol and its validation reported in App. D. All AI-assisted text and code were reviewed by the authors, who take full responsibility for the content of this work.

ETHICS STATEMENT

This work animates photorealistic avatars of real people in reconstructed environments. All avatars, objects, and scenes come from publicly released datasets whose subjects consented to recording and redistribution under the respective licenses; we capture no new human data and show only subjects whose datasets permit publication of renders. Photorealistic drivable avatars can be misused to depict a person doing something they did not do. We recommend that any deployment couple avatar creation with explicit subject consent and provenance signaling on rendered output. Human raters in our preference studies participated voluntarily, and no personal data was collected. The authors have read and adhere to the ICLR Code of Ethics.

REFERENCES

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. imGHUM: Implicit generative models of 3D human shape and articulated pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. BEHAVE: Dataset and method for tracking human object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. SNARF: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Wei Cheng, Ruixiang Chen, Siming Fan, Wanqi Yin, Keyu Chen, Zhongang Cai, Jingbo Wang, Yang Gao, Zhengming Yu, Zhengyu Lin, Daxuan Ren, Lei Yang, Ziwei Liu, Chen Change Loy, Chen Qian, Wayne Wu, Dahua Lin, Bo Dai, and Kwan-Yee Lin. DNA-Rendering: A diverse neural actor repository for high-fidelity human-centric rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Simon Clavet. Motion matching and the road to next-gen animation. In *Game Developers Conference*, 2016.

Christian Diller and Angela Dai. CG-HOI: Contact-guided 3D human-object interaction generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Google DeepMind. Gemini 3.1 Pro model card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, 2026.

Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned Gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Antoine Guédon, Diego Gomez, Nissim Maruani, Bingchen Gong, George Drettakis, and Maks Ovsjanikov. MILo: Mesh-in-the-loop Gaussian splatting for detailed and efficient surface reconstruction. *ACM Transactions on Graphics*, 44(6), 2025.

Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2Avatar: 3D avatar reconstruction from videos in the wild via self-supervised scene decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human POSEitioning system (HPS): 3D human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 1968.

Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. Stochastic Scene-Aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021a.

Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D scenes by learning human-scene interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics*, 36(4), 2017.

Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Transactions on Graphics*, 39(4), 2020.

Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. GaussianAvatar: Towards realistic human avatar modeling from a single video via animatable 3D Gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH Conference Proceedings*, 2024.

Inwoo Hwang, Bing Zhou, Young Min Kim, Jian Wang, and Chuan Guo. SceneMI: Motion inbetweening for modeling human-scene interactions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. HumanRF: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics*, 42(4), 2023.

Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. Scaling up dynamic human-scene interaction modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Hendrik Junkawitsch, Guoxing Sun, Heming Zhu, Christian Theobalt, and Marc Habermann. EVA: Expressive virtual avatars from multi-view videos. In *ACM SIGGRAPH Conference Proceedings*, 2025.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. HUGS: Human Gaussian splats. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3), 2002.

Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C. Karen Liu. Controllable human-object interaction synthesis. In *European Conference on Computer Vision (ECCV)*, 2024a.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime Gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable Gaussians: Learning pose-dependent Gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024c.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6), 2015.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision (3DV)*, 2024.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Marko Mihajlovic, Sergey Prokudin, Siyu Tang, Robert Maier, Federica Bogo, Tony Tung, and Edmond Boyer. SplatFields: Neural Gaussian splats for sparse 3D and 4D reconstruction. In *European Conference on Computer Vision (ECCV)*, 2024.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

Aymen Mir, Thiemo Alldieck, and Gerard Pons-Moll. Learning to transfer texture from clothing images to 3D humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Aymen Mir, Xavier Puig, Angjoo Kanazawa, and Gerard Pons-Moll. Generating continual human motion in diverse 3D scenes. In *International Conference on 3D Vision (3DV)*, 2024.

Aymen Mir, Arthur Moreau, Helisa Dhamo, Zhensong Zhang, Gerard Pons-Moll, and Eduardo Pérez-Pellitero. GASPACHO: Gaussian splatting for controllable humans and objects. *arXiv preprint arXiv:2503.09342*, 2025.

Aymen Mir, Riza Alp Guler, Xiangjun Tang, Peter Wonka, and Gerard Pons-Moll. AHOY! animatable humans under occlusion from YouTube videos with Gaussian splatting and video diffusion priors. In *European Conference on Computer Vision (ECCV)*, 2026.

Gyeongsik Moon, Takaaki Shiratori, and Shunsuke Saito. Expressive whole-body 3D Gaussian avatar. In *European Conference on Computer Vision (ECCV)*, 2024.

Arthur Moreau, Jifei Song, Helisa Dhamo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Human Gaussian splatting: Real-time rendering of animatable avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4), 2022.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. `https://openai.com/index/gpt-5-6/`, 2026.

Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. ASH: Animatable Gaussian splats for efficient and photoreal human rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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 of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3DGS-Avatar: Animatable avatars via deformable 3D Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

RenderPeople. RenderPeople. `https://renderpeople.com/`, 2026. Accessed: 2026-09-17.

Alla Safonova and Jessica K. Hodgins. Construction and optimal search of interpolated motion graphs. *ACM Transactions on Graphics*, 26(3), 2007.

Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. PiGraphs: Learning Interaction Snapshots from Observations. *ACM Transactions on Graphics*, 35(4), 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Transactions on Graphics*, 38(6), 2019.

Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics*, 39(4), 2020.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

SuperSplat. SuperSplat. `https://superspl.at/`, 2026. Accessed: 2026-09-17.

Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. In *International Conference on Learning Representations (ICLR)*, 2023.

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 (CVPR)*, 2025.

Tingwu Wang, Olivier Dionne, Michael De Ruyter, David Minor, Davis Rempe, Kaifeng Zhao, Mathis Petrovich, Ye Yuan, Chenran Li, Zhengyi Luo, Brian Robison, Xavier Blackwell, Bernardo Antoniazzi, Xue Bin Peng, Yuke Zhu, and Simon Yuen. MotionBricks: Scalable real-time motions with modular latent generative model and smart primitives. *ACM Transactions on Graphics*, 45(4), 2026.

Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Lee Alan Westover. *Splatting: a parallel, feed-forward volume rendering algorithm*. PhD thesis, University of North Carolina at Chapel Hill, 1991.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D Gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xianghui Xie, Bharat Lal Bhatnagar, and Gerard Pons-Moll. CHORE: Contact, human and object REconstruction from a single RGB image. In *European Conference on Computer Vision (ECCV)*, 2022a.

Xianghui Xie, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Visibility aware human-object interaction tracking from single RGB camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022b.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. OmniControl: Control any joint at any time for human motion generation. In *International Conference on Learning Representations (ICLR)*, 2024.

Lixin Xue, Chen Guo, Chengwei Zheng, Fangjinhua Wang, Tianjian Jiang, Hsuan-I Ho, Manuel Kaufmann, Jie Song, and Otmar Hilliges. HSR: holistic 3D human-scene reconstruction from monocular videos. In *European Conference on Computer Vision (ECCV)*, 2024.

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

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. PhysDiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Yifan Zhan, Qingtian Zhu, Muyao Niu, Mingze Ma, Jiancheng Zhao, Zhihang Zhong, Xiao Sun, Yu Qiao, and Yinqiang Zheng. ToMiE: Towards explicit exoskeleton for the reconstruction of complicated 3D human avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics*, 2018.

Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3D human-object spatial arrangements from a single image in the wild. In *European Conference on Computer Vision (ECCV)*, 2020.

Juze Zhang, Haimin Luo, Hongdi Yang, Xinru Xu, Qianyang Wu, Ye Shi, Jingyi Yu, Lan Xu, and Jingya Wang. NeuralDome: A neural modeling pipeline on multi-view human-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. COUCH: Towards controllable human-chair interactions. In *European Conference on Computer Vision (ECCV)*, 2022.

Yan Zhang and Siyu Tang. The wanderings of odysseus in 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. DIMOS: Synthesizing diverse human motions in 3D indoor scenes. In *Proceedings of the IEEE/CVF 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 *International Conference on Learning Representations (ICLR)*, 2025.

Kaifeng Zhao, Mathis Petrovich, Haotian Zhang, Tingwu Wang, Siyu Tang, and Davis Rempe. ARDY: Autoregressive diffusion with hybrid representation for interactive human motion generation. *ACM Transactions on Graphics (TOG)*, 45(4), 2026.

Zerong Zheng, Xiaochen Zhao, Hongwen Zhang, Boning Liu, and Yebin Liu. AvatarReX: Real-time expressive full-body avatars. *ACM Transactions on Graphics*, 42(4), 2023.

Heming Zhu, Fangneng Zhan, Christian Theobalt, and Marc Habermann. TriHuman: A real-time and controllable tri-plane representation for detailed human geometry and appearance synthesis. *ACM Transactions on Graphics*, 2024.

Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting. In *ACM SIGGRAPH Conference Proceedings*, 2001.

APPENDIX

This appendix provides additional experiments, background, and implementation details. It is organized as follows:

- App. A: Motion metric definitions and ablation visualizations.
- App. B: Interaction-specific metrics, contact refinement analysis, and SDF proxy stress test.
- App. C: Baseline pipelines and failure analysis.
- App. D: VLM evaluation protocol and validation.
- App. E: Additional qualitative results.
- App. F: Gaussian reconstruction of scenes, humans, and objects.
- App. G: Background on Gaussian Splatting.
- App. H: Background on linear blend skinning.
- App. I: Background on the 3DGS avatar model.
- App. J: Pose projection strategy.
- App. K: RL-based locomotion details.
- App. L: Rollout of the motion model and the composition loop.
- App. M: Interaction clip corpus and retrieval.
- App. N: Dynamic map construction, chaining, and interaction placement.
- App. O: Coordinate frame and height normalization.
- App. P: Hyperparameters.

A MOTION METRICS AND ABLATION VISUALIZATIONS

In Sec. 4 and Tab. 1 we refer to the appendix for formal metric definitions and ablation details. We provide them here.

Mesh-Space Motion Metrics (Tabs. 1 and 3).

1. **Skating.** Following Zhang et al. (2018), foot velocities are accumulated when the foot height h is within a threshold $\tau_h^{\text{eval}} = 2.5\,\text{cm}$ of the floor. Each horizontal velocity magnitude v is weighted as $v\,(2 - 2^{h/\tau_h^{\text{eval}}})$, which decreases from v at the floor to zero at the threshold, and the skating score is the mean of these weighted magnitudes over the sequence.
2. **Penetration.** Per frame, we average the penetration depth over the SMPL body vertices, $\frac{1}{|\mathcal{V}|}\sum_{\mathbf{m}\in\mathcal{V}} |\min(0, \text{SDF}(\mathbf{m}))|$, where $\mathcal{V}$ is the vertex set and SDF the signed distance field of the evaluation mesh: the ground-truth mesh on ScanNet++, the VGGT mesh on DL3DV, and an approximate mesh extracted with SuGaR (Guédon & Lepetit, 2024) on SuperSplat, which has no ground-truth mesh. We report the average over frames in millimeters.
3. **Jerk.** The third derivative of the body joint trajectories, averaged over time and joints.
4. **Goal reaching.** The percentage of sequences whose final pelvis position lies within a threshold of the goal position.

Gaussian-Space Interaction Metrics (Tab. 4). Let $\boldsymbol{\mu}_{k,t}^{\mathsf{P}}$ be the center of posed human Gaussian k at frame t after contact refinement (Eq. 7; the unrefined center when refinement is ablated), and let $\Upsilon \in \{O, S\}$ denote the target entity with retained centers $\boldsymbol{\mu}_j^{\Upsilon}$ (updated per frame for moving objects) and radii r_j as in Eq. 4.

1. **Penetration depth.** Measures how deeply human Gaussians enter object or scene Gaussians. For each human Gaussian whose distance to its nearest target Gaussian is smaller than that Gaussian's radius r_j, we compute $\max(0, r_j - \|\boldsymbol{\mu}_{k,t}^{\mathsf{P}} - \boldsymbol{\mu}_j^{\Upsilon}\|)$ and average over penetrating Gaussians and frames.

2. **Contact distance.** Measures how close the hand contact Gaussians are to the carried object at detected contact frames: $\sum_{k\in\mathcal{C}_{\text{hand}}} \delta_{c(k),t} \min_j \|\boldsymbol{\mu}^{\text{P}}_{k,t} - \boldsymbol{\mu}^{O}_{j}\| \,/\, \sum_{k\in\mathcal{C}_{\text{hand}}} \delta_{c(k),t}$, averaged over frames with at least one hand in contact, where $\mathcal{C}_{\text{hand}} \subset \mathcal{C}$ are the contact Gaussians of the hand joints (Sec. 3.4).
3. **User study.** Thirteen raters were shown 7 pairs of rendered videos, the full method against one ablated variant, and asked to select the more realistic one; we report the preference rate for the full method.

Opacity Culling. Fig. 8 visualizes the effect of opacity culling on the top-down occupancy map. Without culling, floaters project to blocky phantom obstacles that block viable paths.



Figure 8: **Effect of opacity culling on occupancy maps.** (a) Naive orthographic projection of all Gaussians produces blocky structures. (b) Projection after opacity culling. (c) A* path in the filtered map.

B Interaction Metrics, Contact Refinement Analysis, and SDF Stress Test

In Sec. 4.3 we refer to the appendix for interaction-specific metrics, offset statistics of the contact refinement, and a stress test of the SDF proxy. We provide them here (Tabs. 5–8).

Table 5: **Interaction-specific metrics on ScanNet++.** Contact accuracy and penetration of synthesized sit and reach segments.

Action	Contact acc. (%)↑	Pen. (mm)↓
Sit	93.7	14.1
Reach	88.4	17.4

Table 6: **SDF proxy stress test.** Interaction penetration when 1, 2, 3, or 4 frames are dropped between consecutive kept frames of the DL3DV input videos before 3DGS reconstruction.

Frames dropped	1	2	3	4
Pen. (mm)↓	15.1	17.6	19.1	19.3

Table 7: **Contact refinement offsets.** Statistics of the offsets $\mathbf{T}$ over all contact Gaussians and the mean offset per body part.

Statistic	Offset $\|\mathbf{T}\|$ (cm)
Mean, all contact Gaussians	0.5
95th percentile, all contact Gaussians	2.1
Maximum, all contact Gaussians	8.4
Mean, feet	0.6
Mean, pelvis	0.4
Mean, hands	0.3

Table 8: **Contact detector validation.** Precision and recall on 200 hand-labeled frames at the default thresholds, and the largest change of either under the τ_v, τ_a sweep.

Check	Value
Precision	0.89
Recall	0.84
Max. change under sweep	$\leq 7\%$

Interaction-Specific Metrics. The metrics of Tab. 3 summarize whole sequences. To isolate the synthesized interactions of Sec. 3.3, we evaluate sit and reach segments on ScanNet++ with the same protocol as the ablation. *Contact accuracy* is the percentage of segments whose anchor joint (pelvis for sitting, hand for reaching) ends within 2 cm of the scripted location $\mathbf{p}_\ell$, and *penetration* is the mean SDF penetration over the interaction segment. Both actions reach their scripted location in about nine out of ten cases with penetration below 20 mm (Tab. 5).

Contact Refinement Analysis. The refinement of Sec. 3.4 moves Gaussians by 0.5 cm on average and by 2.1 cm at the 95th percentile; the largest offsets occur at the feet (Tab. 7). Offsets of this size do not produce visible shape distortion or texture drift. To validate the heuristic contact detector, we hand-labeled 200 frames as contact or non-contact and measured precision 0.89 and recall 0.84 at the default thresholds. Sweeping τ_v and τ_a over $[0.05, 0.20]$ changes precision and recall by at most 7% (Tab. 8), so the detector is not sensitive to the exact thresholds.

SDF Proxy Stress Test. The proxy of Eq. 4 collapses each Gaussian's anisotropic extent into a single radius. To test how this approximation degrades with reconstruction quality, we keep only every 2nd, 3rd, 4th, or 5th frame of the DL3DV input videos, dropping 1 to 4 frames in between, before 3DGS reconstruction, which yields fewer, narrower, and more elongated Gaussians. Interaction penetration rises only from 15 mm to 19 mm (Tab. 6), so the proxy is robust to the Gaussian shapes encountered in practice.

C Baseline Pipelines and Failure Analysis

In Secs. 4.1 and 4.2 we refer to the appendix for the construction of Baselines A to D and for a failure analysis of the mesh baselines. We provide them here. Baselines A, B, and D synthesize motion with DIMOS (Zhao et al., 2023) and differ in where the scene geometry comes from and how the result is rendered; Baseline C keeps the motion of our pipeline and changes only the renderer (Tab. 9). Baselines A and B answer the motion question of Sec. 4.1: can a mesh extracted from the same input match motion planned directly on the Gaussians? Baselines C and D answer the rendering question of Sec. 4.2: is the preference for our renderings due to the representation, or to the content we happen to render?

Table 9: **Summary of the four mesh baselines.** Baselines A, B, and D synthesize motion with DIMOS; Baseline C uses the motion of our pipeline. Baselines A and B are evaluated for motion quality (Tab. 1), Baselines C and D for rendering quality, and Baseline B additionally serves as the within-scene monocular rendering comparison (Tab. 2).

Baseline	Scene geometry	Avatar	Rendering	Evaluated for
A (extracted mesh)	MiLo mesh from our Gaussians	3DGS avatar	3DGS	Motion
B (VGGT mesh)	VGGT mesh from video	3DGS avatar	3DGS	Motion, rendering
C (same constituents)	MiLo mesh of our Gaussians (rendering only)	SuGaR mesh of ours	Mesh rasterization	Rendering
D (HQ vs HQ)	Replica mesh	RenderPeople scan	Mesh rasterization	Rendering

Baseline A (Mesh Extracted from the Same Gaussians). The most direct alternative to our pipeline is to extract a mesh from the scene Gaussians and hand it to an existing mesh pipeline. Baseline A does this: input video $\rightarrow$ 3DGS $\rightarrow$ surface extraction with MiLo (Guédon et al., 2025) $\rightarrow$ mesh $\rightarrow$ DIMOS motion planned and synthesized on the mesh $\rightarrow$ compositing of the 3DGS avatar into the 3DGS scene. The avatar, scene, and renderer are therefore identical to ours, and only the geometry that the animation stack queries differs. Fig. 9a shows the extracted mesh of a DL3DV living room. Surface extraction from Gaussians inherits every floater and every under-reconstructed region: floaters become spurious blobs of geometry in free space, thin and glossy structures such as the table top and the plant are fragmented, and regions seen from few views are missing altogether. DIMOS treats the resulting surface as ground truth. Where the floor is bumpy the navigation mesh is cut into disconnected islands, where geometry is missing the planner walks through furniture, and where blobs float in free space the planner detours around obstacles that do not exist. Fig. 9b shows the outcome: the avatar walks into the coffee table and the plant because the extracted mesh has no closed surface there. Our pipeline, which never extracts a mesh from the scene, walks around them (Fig. 9c). Quantitatively, Baseline A shows about $3\times$ the skating and $5\times$ the penetration of our method on DL3DV, and $2.5\times$ and $5\times$ on SuperSplat (Tab. 1). We also tried computing our walkability grid from the extracted mesh instead of from the Gaussians; the penetrations persist, since the mesh itself is wrong, which is why every geometric query in Sec. 3 is made against the Gaussians directly.



(a) MiLo mesh (b) Baseline A (c) Ours

Figure 9: **Baseline A: mesh extracted from our scene Gaussians.** (a) The MiLo mesh of a DL3DV living room has holes, fragmented thin structures, and spurious geometry where floaters were meshed. (b) Motion planned on that mesh and rendered in the 3DGS scene penetrates the coffee table and the plant. (c) Our pipeline plans on the Gaussians and walks around them.

Baseline B (VGGT Mesh). Baseline B tests the other natural mesh pipeline: skip 3DGS altogether and reconstruct the scene mesh directly from the video with a feed-forward reconstruction model. Each DL3DV video is passed through VGGT (Wang et al., 2025), the predicted point map is meshed, and DIMOS plans and synthesizes motion on the result. The 3DGS avatar is then composited into the 3DGS scene without our walkability grid, interaction synthesis, or contact refinement. Both Baseline B and our method see exactly the same frames, so the asymmetry is feed-forward mesh prediction versus photometric 3DGS optimization. Fig. 10a shows a VGGT mesh of the same living room. The red circle marks the region where the sofa, the coffee table, and the floor merge into a single noisy surface. The floor itself is uneven, and the far end of the room is warped by accumulated depth error. These artifacts produce the three failure modes we observe throughout: (1) paths that pass through furniture, because merged geometry leaves no free corridor and the planner picks the least bad one, (2) interaction poses that penetrate the scene, because the sitting target is placed on a surface that does not coincide with the rendered sofa, and (3) unnatural motion, because the policy compensates for an uneven floor with steps that skate and jerk. Fig. 10b shows the avatar walking through the couch region marked in (a), while ours reaches the same goal along the free corridor (Fig. 10c). Baseline B is the weakest of the motion baselines on every metric (Tab. 1), although it is scored against its own VGGT mesh, which if anything favors it.

Baseline B also appears in the rendering study (Tab. 2) as the within-scene monocular comparison: raters see the full Baseline B pipeline and our pipeline on the same DL3DV scene. Because both are composited into the same 3DGS scene, this column measures the end-to-end result on one input video, including the penetrations and floating contacts that the mesh pipeline introduces, rather than the renderer in isolation. Human raters prefer our result in 72.9% of pairs, the two VLM judges in 81.3% and 78.2%.



(a) VGGT mesh (b) Baseline B (c) Ours

Figure 10: **Baseline B: VGGT mesh from monocular video.** (a) The feed-forward VGGT mesh of the same living room merges the sofa, the coffee table, and the floor into one noisy surface (red circle) and warps the far end of the room. (b) DIMOS motion planned on that mesh walks through the marked region. (c) Our pipeline, reconstructed from the same frames, keeps to the free corridor.

Baseline C (Same Constituents, Mesh Rendering). Baselines A and B are rendered with 3DGS, so they say nothing about rendering quality. Baseline C closes this gap while keeping the content fixed: it takes the motion synthesized by our full pipeline on the Gaussians and converts only the final Gaussians for rendering, the scene Gaussians to a mesh with MiLo (Guédon et al., 2025) and the posed avatar Gaussians to a textured mesh with SuGaR (Guédon & Lepetit, 2024), which are then rendered with standard mesh rasterization. No mesh is involved in motion synthesis. The

comparison therefore shares the scene reconstruction, the avatar capture, and the motion with our pipeline, and only the representation used for rendering changes. Fig. 11 shows a frame pair. The mesh rendering loses texture sharpness across the scene, flattens the view-dependent appearance of the floor and the television, and blurs thin structures such as the plant leaves and picture frames. The avatar suffers in the same way: SuGaR extraction bakes the Gaussians into a fixed surface with a single texture, which removes the view-dependent shading and the soft silhouette of the 3DGS avatar. In the human study 19 participants prefer our renderings in 86.4% of pairwise judgments, and the VLM judges in 88.1% (GPT-5.6) and 84.1% (Gemini 3.1), the strongest preference of the three rendering comparisons. Since the constituents are identical, this preference is attributable to the representation itself.



(a) Baseline C (b) Ours

Figure 11: **Baseline C: same scene, same avatar, mesh rendering.** (a) The MiLo scene mesh and the SuGaR mesh of our avatar rendered with mesh rasterization. (b) The same scene and avatar rendered as Gaussians. Texture detail, view-dependent appearance, and thin structures are lost in the mesh rendering.

Baseline D (Cross-Scene, HQ vs HQ). Baseline C gives the mesh pipeline our reconstructed content, which one could argue is unfair to meshes, since Gaussians were the source. Baseline D removes this objection by giving the mesh pipeline the best content available to it: Replica (Straub et al., 2019) scene meshes, which are artist-cleaned reconstructions with high-resolution textures, and a RenderPeople (RenderPeople, 2026) scan, a commercial photogrammetry avatar, animated by DIMOS and rendered with a mesh renderer. It is compared against our best 3DGS setting, SuperSplat (SuperSplat, 2026) scenes with AvatarReX (Zheng et al., 2023) avatars. A within-scene comparison is impossible here, because Replica has no multi-view video from which to train Gaussians and SuperSplat scenes have no ground-truth mesh, so the two pipelines are shown on different scenes and raters are asked which video is more photorealistic. Fig. 12 shows two such pairs. The Replica renderings are clean but carry the signature of mesh rendering: baked lighting that does not respond to the avatar, flat shading on the scan, and hard silhouettes where avatar and floor meet. Our renderings preserve reflections and soft shadows that were captured in the Gaussians. Human raters prefer our renderings in 82.1% of pairs, the VLM judges in 76.1% and 72.9%. The preference is smaller than for Baseline C, as expected when the mesh side is given clean content, but it remains clear.



(a) Baseline D (b) Ours (c) Baseline D (d) Ours

Figure 12: **Baseline D: the highest-quality mesh pipeline against our best 3DGS setting.** (a,c) Replica scenes with a RenderPeople scan animated by DIMOS and rendered as meshes. (b,d) SuperSplat scenes with AvatarReX avatars animated and rendered by our pipeline. The pipelines require different capture setups, so the scenes differ; raters judge photorealism.

D VLM Evaluation Protocol and Validation

In Sec. 4.2 we refer to the appendix for the VLM evaluation protocol and its validation. We provide them here.

Protocol. For the pairwise VLM study, the model 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 (halos, floaters). Respond with* `A`*,* `B`*, or* `Tie`*."* Each video pair contributes 10 matched frame pairs, and the order of the two images is randomized.

Validation. Tab. 10 validates the VLM judges. On 100 image pairs contrasting synthetic BEDLAM (Black et al., 2023) renderings with real DL3DV frames, the VLMs pick the real image 92.1% of the time. Flipping the presentation order changes the verdict in only 6.5% of pairs, ruling out positional bias. The agreement checks use the 10 video pairs of Baselines C and D; GPT-5.6 and Gemini 3.1 agree on 9 of these 10 video-level verdicts. On the 8 of these video pairs where at least 75% of participants agreed, both VLMs match the human consensus on all 8.

Table 10: **VLM judge validation.** Checks on the GPT-5.6 and Gemini 3.1 judges used in Tab. 2.

Check	Value
Real (DL3DV) preferred over synthetic (BEDLAM), 100 pairs	92.1%
Verdict changes under order flip	6.5%
Inter-VLM agreement (GPT-5.6 vs Gemini 3.1), Baselines C and D videos	9 / 10
VLM agreement with human consensus ($\geq 75\%$ of raters)	8 / 8

E Additional Qualitative Results

In Sec. 4 we refer to the appendix for additional qualitative results. We provide them here: Fig. 13 shows further animations in indoor and outdoor 3DGS scenes.

F Gaussian Reconstruction of Scenes, Humans, and Objects

In Sec. 3 of the main paper we refer to the appendix for details of the scene, human, and object reconstructions. We provide them here; rendering and skinning background follow in Apps. G and H.

Scene Gaussians. Given a casually captured mobile-phone video or multi-view video of a static scene, we model the environment as N_S anisotropic Gaussians $\mathcal{G}^S = \{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i, \mathbf{c}_i)\}_{i=1}^{N_S}$ with center $\boldsymbol{\mu}_i \in \mathbb{R}^3$, covariance $\boldsymbol{\Sigma}_i$, opacity $\alpha_i \in (0,1)$, and color $\mathbf{c}_i$, optimized with a photometric loss following Kerbl et al. (2023). The covariance is parameterized by a scale vector $\mathbf{s}_i \in \mathbb{R}^3$ and a rotation (App. G).

Human Gaussians. We learn pose-conditioned avatars from multi-view video following Animatable Gaussians (Li et al., 2024c). A per-identity StyleUNet f_ζ maps SMPL position maps $\mathcal{P}_t = \mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta}_t)$ to canonical Gaussians $\mathcal{G}_t^{H,\mathsf{C}} = f_\zeta(\mathcal{P}_t)$ anchored on the SMPL surface via a front-back UV atlas, where $\mathcal{M}$ is the SMPL forward model, $\boldsymbol{\beta}$ the shape, and $\boldsymbol{\theta}_t$ the pose at frame t (App. I). Linear blend skinning (LBS) with SMPL joint transformations yields posed Gaussians $\mathcal{G}_t^{H,\mathsf{P}} = \{(\boldsymbol{\mu}_{k,t}^{\mathsf{P}}, \boldsymbol{\Sigma}_{k,t}^{\mathsf{P}}, \alpha_k, \mathbf{c}_k)\}_{k=1}^{N_H}$. At test time, synthesized poses are projected onto the top-K PCA components of the training position maps (App. J). The mapping $\boldsymbol{\theta}_t \mapsto \mathcal{G}_t^{H,\mathsf{P}}$ passes through the network f_ζ, which makes direct pose optimization for contact intractable (Sec. 3.4).

Object Gaussians and Interaction Clips. For dynamic objects we use the reconstructions of Mir et al. (2025), which recover from multi-view interaction captures a canonical object Gaussian set $\mathcal{G}^O = \{(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p, \alpha_p, \mathbf{c}_p)\}_{p=1}^{N_O}$ together with tracked rigid object poses $\{\boldsymbol{\phi}_t\}$ and the accompanying human poses $\{\boldsymbol{\theta}_t\}$. The object is placed in the scene by a rigid transform, $\mathcal{G}_t^{O,\mathsf{P}} = \mathcal{T}(\mathcal{G}^O; \boldsymbol{\phi}_t)$. The tracked pose pairs $(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$ form a corpus of short interaction clips (lifting, carrying, putting down, pushing) that our method retrieves and chains (Sec. 3.5, App. M).



Figure 13: **Additional qualitative results across diverse 3DGS environments. Top:** avatar navigating an indoor living room while avoiding furniture. **Bottom:** avatar navigating an outdoor courtyard with planters, benches, and chairs. Both results are rendered entirely within the Gaussian Splatting framework.

G Background: Gaussian Splatting

In Sec. 3 we refer to the appendix for rendering and skinning details. We provide the rendering background here.

3D Gaussian Splatting (Kerbl et al., 2023) represents a static scene by a set of Gaussian primitives $\mathcal{G}$ that can be rendered in real time by rasterization. Each Gaussian is parameterized by its center $\boldsymbol{\mu} \in \mathbb{R}^3$, covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$, opacity $\alpha \in (0, 1)$, and view-dependent color $\mathbf{c}$. To ensure positive semi-definiteness, the covariance is decomposed into a rotation $\mathbf{R}$ and a scale matrix $\mathbf{S}$,

$$\boldsymbol{\Sigma} = \mathbf{RSS}^\top \mathbf{R}^\top, \tag{10}$$

where the rotation is stored as a quaternion $\mathbf{q} \in \mathbb{R}^4$ and the scale matrix is diagonal with entries $\mathbf{s} \in \mathbb{R}^3$. Each Gaussian i is thus specified by $(\boldsymbol{\mu}_i, \mathbf{q}_i, \alpha_i, \mathbf{s}_i, \mathbf{c}_i)$.

The Gaussians are projected to the image plane and alpha-blended. Under camera projection Π_t, the 2D covariance in screen space (Zwicker et al., 2001) is $\mathbf{\Sigma}_{i,t}^{2\mathrm{D}} = (\mathbf{A}_t\, \mathbf{\Sigma}_i\, \mathbf{A}_t^\top)_{1:2,1:2}$, where $\mathbf{A}_t$ is the Jacobian of the affine approximation of the projective transformation. The color of pixel $\mathbf{u}$ is obtained by blending the depth-sorted splats that overlap it,

$$\hat{\mathbf{C}}_t(\mathbf{u}) = \sum\nolimits_i \Big(\hat{\alpha}_{i,t}(\mathbf{u}) \prod\nolimits_{i'<i} (1 - \hat{\alpha}_{i',t}(\mathbf{u}))\Big)\mathbf{c}_i, \tag{11}$$

where $\hat{\alpha}_{i,t}(\mathbf{u}) = \alpha_i\, g_{i,t}(\mathbf{u})$ is the opacity weighted by the Gaussian kernel $g_{i,t}$ evaluated at $\mathbf{u}$. Parameters are optimized against ground-truth images with a photometric loss, and the number of Gaussians is controlled by periodic densification and pruning.

H Background: Linear Blend Skinning for Posing

In Sec. 3 we refer to the appendix for skinning details. We provide them here.

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

Canonical Gaussians and Anchoring. Each canonical Gaussian is $(\boldsymbol{\mu}_k^{\mathsf{C}}, \mathbf{\Sigma}_k^{\mathsf{C}}, \alpha_k, \mathbf{c}_k)$ with center $\boldsymbol{\mu}_k^{\mathsf{C}} \in \mathbb{R}^3$, covariance $\mathbf{\Sigma}_k^{\mathsf{C}}$, opacity α_k, and color $\mathbf{c}_k$. Gaussians are anchored to the SMPL surface by nearest-neighbor association to mesh vertices.

Skinning Weights and Bone Transformations. Let B be the number of SMPL joints. Gaussian k inherits skinning weights $w_{k,b} \in [0,1]$ with $\sum_{b=1}^{B} w_{k,b} = 1$ from its associated vertex. Given pose $\boldsymbol{\theta}_t$, SMPL provides per-bone rigid motions $\mathbf{R}_b(\boldsymbol{\theta}_t) \in \mathrm{SO}(3)$ and $\mathbf{t}_b(\boldsymbol{\theta}_t) \in \mathbb{R}^3$ relative to the rest pose.

Deformation. The posed center and covariance of Gaussian k are

$$\boldsymbol{\mu}_{k,t}^{\mathsf{P}} = \sum_{b=1}^{B} w_{k,b} \left(\mathbf{R}_b(\boldsymbol{\theta}_t)\, \boldsymbol{\mu}_k^{\mathsf{C}} + \mathbf{t}_b(\boldsymbol{\theta}_t)\right), \tag{12}$$

$$\mathbf{\Sigma}_{k,t}^{\mathsf{P}} \approx \Big(\sum_{b=1}^{B} w_{k,b}\, \mathbf{R}_b(\boldsymbol{\theta}_t)\Big) \mathbf{\Sigma}_k^{\mathsf{C}} \Big(\sum_{b=1}^{B} w_{k,b}\, \mathbf{R}_b(\boldsymbol{\theta}_t)\Big)^\top. \tag{13}$$

The covariance update ignores shear and higher-order effects, but is accurate in practice and mirrors vertex LBS. The posed set $\mathcal{G}_t^{H,\mathsf{P}} = \{(\boldsymbol{\mu}_{k,t}^{\mathsf{P}}, \mathbf{\Sigma}_{k,t}^{\mathsf{P}}, \alpha_k, \mathbf{c}_k)\}_{k=1}^{N_H}$ is rendered with the standard rasterizer. Together with the pose-conditioned prediction f_ζ, this defines the differentiable mapping $\boldsymbol{\theta}_t \mapsto \mathcal{G}_t^{H,\mathsf{P}}$ used throughout the paper. Object Gaussians use a single rigid transform $\mathcal{T}(\mathcal{G}^O; \boldsymbol{\phi}_t)$ in place of the blended bone transforms.

I Background: 3DGS Avatar Model

In App. F we summarize the human reconstruction pipeline. Here we describe it in more detail. Our avatars follow Animatable Gaussians (Li et al., 2024c): a pose-dependent network maps SMPL body poses to Gaussian primitives, which are articulated via LBS (App. H).

Template Reconstruction. Given multi-view video of a person, a per-identity canonical template is reconstructed first. Rather than relying on the naked SMPL mesh, which cannot represent loose garments, a canonical signed distance field and color field are optimized from a near-A-pose frame, and a mesh is extracted via Marching Cubes. SMPL skinning weights are diffused from the body surface to the template surface, so the template can be deformed via LBS while capturing the geometry of worn clothing.

Position Maps. For each pose $\boldsymbol{\theta}_t$, the canonical template is deformed via LBS and rendered under front and back orthographic projections, producing two position maps $\mathcal{P}_{\mathrm{fr}}(\boldsymbol{\theta}_t)$ and $\mathcal{P}_{\mathrm{bk}}(\boldsymbol{\theta}_t)$; together they constitute the position map $\mathcal{P}_t = \mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta}_t)$ of App. F. Each pixel stores the 3D coordinates of the corresponding deformed template point. The two views form an approximate UV atlas of the body: nearly every surface point is visible in at least one, which gives a dense surface-to-image correspondence without the seams of a full unwrapping.

Gaussian Maps. A StyleUNet f_ζ, a U-Net with StyleGAN-based modulation layers, maps the position maps and a view-direction encoding $\mathbf{v}_{\text{dir}}$ to Gaussian maps $\mathcal{G}_{\text{fr}}(\boldsymbol{\theta}_t), \mathcal{G}_{\text{bk}}(\boldsymbol{\theta}_t) \leftarrow f_\zeta(\mathcal{P}_{\text{fr}}(\boldsymbol{\theta}_t), \mathcal{P}_{\text{bk}}(\boldsymbol{\theta}_t), \mathbf{v}_{\text{dir}})$. Each foreground pixel encodes one Gaussian: a position offset $\Delta\boldsymbol{\mu}$ relative to the template surface, scales $\mathbf{s}$, a rotation quaternion $\mathbf{q}$, opacity α, and view-dependent color $\mathbf{c}$. The union of foreground pixels yields the canonical set $\mathcal{G}_t^{H,\mathsf{C}}$.

Training. The network is trained end-to-end from multi-view video (4 to 48 cameras in our experiments). For each frame the predicted canonical Gaussians are posed via LBS and rendered, and the parameters are optimized with an L1 and perceptual photometric loss against the ground-truth images.

J Pose Projection Strategy

In Secs. 3 and 3.4 we refer to the appendix for the projection of synthesized poses into the avatar's pose subspace. We provide it here.

Following Li et al. (2024c), we constrain test-time poses to the training distribution with PCA to improve generalization to poses unseen during avatar training. Let $\{\mathcal{P}_t\}_{t=1}^{N_P}$ be the N_P training position maps, vectorized as $\mathbf{y}_t \in \mathbb{R}^D$ with D the number of map pixels times channels, and stacked into $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{N_P}]$. PCA on $\mathbf{Y}$ gives the mean $\bar{\mathbf{y}}$, the top K components $\mathbf{Q}_K \in \mathbb{R}^{D\times K}$, and standard deviations $\{\sigma_q\}_{q=1}^{K}$. A synthesized pose map, vectorized as $\mathbf{y}$, is projected by

$$\boldsymbol{\xi} = \mathbf{Q}_K^\top(\mathbf{y} - \bar{\mathbf{y}}), \qquad \tilde{\mathbf{y}} = \bar{\mathbf{y}} + \mathbf{Q}_K\boldsymbol{\xi}, \tag{14}$$

where each coefficient ξ_q is clipped to $[-2\sigma_q, 2\sigma_q]$ before reconstruction. The reconstructed map $\tilde{\mathbf{y}}$ is fed to f_ζ and skinned to posed Gaussians. Poses synthesized by our locomotion and interaction modules are therefore interpolated smoothly within the training distribution of the avatar, which is what makes it possible to drive avatars captured performing everyday motions with novel scene-aware motion.

K RL-Based Locomotion for Navigating 3DGS Scenes

In Sec. 3.2 we refer to the appendix for the motion backbone, reward terms, and policy learning. We provide them here.

Latent Motion Backbone. The transformer VAE of Zhao et al. (2025) learns a motion primitive space from AMASS (Mahmood et al., 2019; Punnakkal et al., 2021). Given history $\mathbf{H}$ and a future segment $\mathbf{X}$, the encoder outputs a Gaussian posterior $q_{\text{enc}}(\mathbf{z} \mid \mathbf{H}, \mathbf{X}) = \mathcal{N}(\boldsymbol{\nu}, \text{diag}(\boldsymbol{\sigma}^2))$ with mean $\boldsymbol{\nu}$ and standard deviation $\boldsymbol{\sigma}$, sampled as $\mathbf{z} = \boldsymbol{\nu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The decoder reconstructs $\widehat{\mathbf{X}} = \mathcal{D}(\mathbf{H}, \mathbf{z})$. The diffusion prior in this latent space has forward process $q(\mathbf{z}_m \mid \mathbf{z}_{m-1}) = \mathcal{N}(\sqrt{1-\chi_m}\,\mathbf{z}_{m-1}, \chi_m\mathbf{I})$ over diffusion steps $m = 1, \ldots, m_{\max}$ with noise schedule χ_m, and the denoiser $\mathcal{F}$ predicts the clean code as in Eq. 3.

Policy and Actions. The locomotion policy is trained with PPO (Schulman et al., 2017) and an actor-critic architecture; both networks are 4-layer MLPs of width 512 with residual connections. The policy output is the diffusion start noise

$$\mathbf{z}_{\text{RL},n} = 4\,\tanh\left(\mathbf{z}_{\text{raw},n}\right) \in [-4, 4]^{n_z}, \tag{15}$$

where n_z is the latent dimension and $\mathbf{z}_{\text{raw},n}$ the unconstrained network output at policy step n. The bounded parameterization stabilizes exploration and keeps the noise in the range expected by the frozen denoiser and decoder.

Observations. At step n the policy observes $o_n = (\mathbf{H}_n, \mathbf{g}_n, \mathcal{W}_n, \mathbf{e}_n)$ as in Eq. 2: the motion history, a goal cue $\mathbf{g}_n$, the goal $\mathbf{g}$ clamped to a 5 m egocentric horizon and 120° field of view, the egocentric walkability grid $\mathcal{W}_n \in \{0,1\}^{N_g\times N_g}$ with $N_g = 16$ covering 1.6×1.6 m around the pelvis, and a text cue defaulting to "walk". During training in synthetic rooms the grid is filled from the navigation mesh and object collisions; at deployment it is computed from opacity-filtered scene and object Gaussians as in Sec. 3.2.

Rewards. Let $\mathbf{x}_n^{\text{pel}} \in \mathbb{R}^3$ be the pelvis position at step n and $\mathbf{g} \in \mathbb{R}^3$ the goal. With planar distance $d_n = \|(\mathbf{x}_n^{\text{pel}} - \mathbf{g})_{xz}\|_2$, where $(\cdot)_{xz}$ drops the vertical component, progress and success are

$$\varrho_{\text{dist}} = d_{n-1} - d_n, \qquad \varrho_{\text{succ}} = \mathbf{1}[d_n < 0.3]. \tag{16}$$

Orientation alignment is $\varrho_{\text{ori}} = \frac{1}{2}(\langle \mathbf{x}_n^{\text{pel}} - \mathbf{x}_{n-1}^{\text{pel}}, \mathbf{g} - \mathbf{x}_{n-1}^{\text{pel}} \rangle + 1)$ with normalized direction vectors. Kinematic plausibility penalizes skating and floor contact errors,

$$\varrho_{\text{skate}} = -\, v_f \left(2 - 2^{h/\tau_h}\right) \mathbf{1}[h < \tau_h], \qquad \varrho_{\text{floor}} = -\left[|h_f| - \tau_f\right]_+, \tag{17}$$

where v_f is the horizontal foot displacement, h the foot height, $\tau_h = 0.03\,\text{m}$ a height threshold, h_f the signed height of the lowest foot vertex above the floor, $\tau_f = 0.03\,\text{m}$ a contact tolerance, and $[\cdot]_+ = \max(0, \cdot)$. The floor term penalizes the lowest foot vertex leaving a band of $\pm\tau_f$ around the floor, that is, both floating above it and penetrating below it. Obstacle avoidance discourages overlap between the blocked cells of $\mathcal{W}_n$ and the planar body footprint $\mathcal{B}_{xz}(\mathbf{X})$, the ground-plane projection of the body surface points of the primitive, $\varrho_{\text{pene}} = \exp(-|\{u : \mathcal{W}_n(u) = 0\} \cap \mathcal{B}_{xz}(\mathbf{X})|)$, where $|\cdot|$ counts overlapping cells. The total reward is

$$\varrho_n = w_{\text{dist}}\varrho_{\text{dist}} + w_{\text{succ}}\varrho_{\text{succ}} + w_{\text{ori}}\varrho_{\text{ori}} + w_{\text{skate}}\varrho_{\text{skate}} + w_{\text{floor}}\varrho_{\text{floor}} + w_{\text{pene}}\varrho_{\text{pene}}, \tag{18}$$

with weights listed in App. P.

Training Environments. We procedurally generate rectangular rooms with edges between 2 and 7 m, filled with ShapeNet furniture (chairs, beds, sofas, desks, tables), aligned with the vertical axis and with expanded margins to preserve traversability. Navigation meshes yield collision-free start and goal pairs and waypoint paths, and randomized initial poses and headings encourage robustness. The policy never sees a 3DGS scene during training; the walkability grid is the only interface to the environment, which is what allows deployment on Gaussians.

L Rollout and Composition Loop

In Sec. 3.3 we refer to the appendix for the rollout of the latent motion model, and in Sec. 3.5 for the composition loop as an algorithm. We provide both here. Alg. 1 originates from Zhao et al. (2025) and is used unchanged inside our locomotion, interaction synthesis, and inbetweening; Alg. 2 is the loop of Sec. 3.5.

Algorithm 1 Autoregressive rollout with the latent motion model.

Require: Decoder $\mathcal{D}$, denoiser $\mathcal{F}$, motion seed $\mathbf{H}_{\text{seed}}$, history length F_h, number of primitives N, text embeddings $\mathbf{e}_1, \ldots, \mathbf{e}_N$, diffusion steps $m_{\max}$, guidance scale w_{cfg}, DDIM sampler DDIM
Optional: Initial noises $\mathbf{z}_{m_{\max},1}, \ldots, \mathbf{z}_{m_{\max},N}$
Ensure: Motion sequence $\mathbf{M}$

$\mathbf{H}_1 \leftarrow \mathbf{H}_{\text{seed}}$, $\mathbf{M} \leftarrow \mathbf{H}_{\text{seed}}$
for $n \leftarrow 1$ to N **do** ▷ number of motion primitives
  if latent noise not provided **then**
    Sample $\mathbf{z}_{m_{\max},n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  else
    Use provided $\mathbf{z}_{m_{\max},n}$
  end if
  $\widehat{\mathbf{z}}_{0,n} \leftarrow \text{DDIM}(\mathcal{F}, \mathbf{z}_{m_{\max},n}, m_{\max}, \mathbf{H}_n, \mathbf{e}_n, w_{\text{cfg}})$ ▷ DDIM denoising with guidance, Eq. 3
  $\widehat{\mathbf{X}} \leftarrow \mathcal{D}(\mathbf{H}_n, \widehat{\mathbf{z}}_{0,n})$ ▷ decode latent to motion primitive
  $\mathbf{M} \leftarrow \text{CONCAT}(\mathbf{M}, \widehat{\mathbf{X}})$
  $\mathbf{H}_{n+1} \leftarrow \text{CANONICALIZE}(\widehat{\mathbf{X}}_{F-F_h+1:F})$ ▷ update history with the last F_h of the F frames
end for
return $\mathbf{M}$

M Interaction Clip Corpus and Retrieval

In Sec. 3.5 we retrieve object interactions from a corpus of captured interaction clips and refer to the appendix for the corpus and the retrieval rule. We provide them here.

Algorithm 2 Composition of long animations in changing scenes (Sec. 3.5).

Require: Retained scene Gaussians, avatar f_ζ, object Gaussians $\mathcal{G}^O$ with initial pose $\boldsymbol{\phi}_0$, clip corpus (App. M), start and goal positions, script $(\mathbf{p}_\ell, a_\ell)_{\ell=1}^{L}$
Ensure: Posed human and object Gaussians for every frame t

$\mathcal{O}^S \leftarrow$ project retained scene Gaussians (Sec. 3.1); $\mathcal{O}_t \leftarrow \mathcal{O}^S \vee \mathcal{O}_t^O$ from $\boldsymbol{\phi}_0$ (Eq. 1)
$\mathbf{M} \leftarrow \emptyset$; current position $\leftarrow$ start
for $\ell \leftarrow 1$ to $L+1$ **do**
  target $\leftarrow \mathbf{p}_\ell$ if $\ell \leq L$ else goal
  waypoints $\leftarrow$ A* on $\mathcal{O}_t$ from current position to target
  $\mathrm{loco}_{\ell-1} \leftarrow$ locomotion policy along waypoints with grid $\mathcal{W}_n$ from scene and posed object Gaussians (Sec. 3.2, Alg. 1)
  $\mathbf{M} \leftarrow \mathrm{CONCAT}(\mathbf{M}, \mathrm{loco}_{\ell-1})$ ▷ the preceding out transition bridges into the locomotion
  if $\ell \leq L$ **then**
    if a_ℓ is a scene action **then**
      $\mathcal{K}_\ell \leftarrow$ latent optimization of Eq. 5 against $\rho(\cdot; S)$ (Sec. 3.3)
    else
      $\mathcal{K}_\ell \leftarrow$ retrieve clip with label a_ℓ, place at $\mathbf{p}_\ell$ (App. N); object follows tracked $\boldsymbol{\phi}_t$, then the wrist if carried
    end if
    $\mathbf{M} \leftarrow \mathrm{CONCAT}(\mathbf{M}, \mathrm{in}_\ell, \mathcal{K}_\ell, \mathrm{out}_\ell)$
    if $\mathcal{K}_\ell$ moved the object **then**
      update $\boldsymbol{\phi}_t$, recompute $\mathcal{O}_t^O$ and $\mathcal{O}_t$ (Eq. 1)
    end if
    current position $\leftarrow$ pelvis at the end of out_ℓ
  end if
end for
Project poses of $\mathbf{M}$ to the avatar subspace, map through f_ζ, skin; transform $\mathcal{G}^O$ by $\boldsymbol{\phi}_t$ (Sec. 3.4)
Contact refinement, Eq. 7, for human–scene, human–object, and object–scene contact
return refined human and object Gaussians per frame

Sources. Every clip originates from a multi-view capture of a person interacting with a rigid object, taken from BEHAVE (Bhatnagar et al., 2022) (4 Kinect cameras), DNA-Rendering (Cheng et al., 2023) (60 cameras, of which 48 are used), and NeuralDome (Zhang et al., 2023). From each capture, GASPACHO (Mir et al., 2025) reconstructs a canonical object Gaussian set $\mathcal{G}^O$ and, jointly, the SMPL pose sequence $\{\boldsymbol{\theta}_t\}$ of the person and the rigid object pose sequence $\{\boldsymbol{\phi}_t\}$, with the human–object contact of the capture preserved in the tracked poses (App. F). We do not use the human Gaussians of these captures for animation: a clip contributes only its pose pairs $(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$, and the avatar that performs the clip is any avatar of Sec. 3, which is what lets one clip drive avatars from ActorsHQ, AvatarReX, DNA-Rendering, or BEHAVE alike. The object Gaussians, in contrast, are always taken from the same capture as the clip, so that the tracked hand–object contact matches the object geometry.

Clips and Labels. Each capture is cut into short clips that cover a single interaction with the object and are labeled with an action, one of lift, carry, put down, and push, and with the object of the capture (e.g., chair, box, trashcan). Clips are stored in the capture frame together with the object Gaussians and are converted to the gravity-aligned scene frame at placement time (App. N).

Retrieval. The user script assigns an action label a_ℓ to each scripted location $\mathbf{p}_\ell$ (Sec. 3). For an object action, the retrieved clip $\mathcal{K}_\ell$ is the clip of the object currently placed in the scene whose label equals a_ℓ; the object itself is chosen when the scene is set up, so retrieval reduces to a label lookup within that object's clips. A retrieved clip is placed by the rigid transform of App. N, its object follows the tracked poses during the clip, and the composition loop of Alg. 2 attaches the object to the wrist after a lift and releases it once it is put down. Scene actions (sit, reach) are never retrieved but synthesized by the optimization of Sec. 3.3.

N Dynamic Map Construction, Chaining, and Interaction Placement

In Sec. 3.5 we refer to the appendix for the placement of retrieved clips. We provide it here, together with construction details of the dynamic maps of Sec. 3.1 and of the inbetweening transitions of Sec. 3.5. The clip corpus itself and the retrieval rule are described in App. M.

Dynamic Map Construction. After the PCA alignment of App. O, we discretize the ground plane into an $M \times M$ grid with $M = 4096$. The opacity-filtered scene Gaussians are projected orthographically onto this grid, and a cell is an obstacle in the static map $\mathcal{O}^S$ if its projected Gaussian density exceeds a threshold. For the object map $\mathcal{O}_t^O$ we transform the posed object Gaussians by $\mathbf{R}_{\text{align}}$, project them with the same orthographic camera, mark the covered cells as obstacles, and dilate the footprint by a small kernel as a safety margin. The dynamic map is the union of obstacles, $\mathcal{O}_t = \mathcal{O}^S \vee \mathcal{O}_t^O$. Whenever an interaction moves the object we recompute $\mathcal{O}_t^O$ and re-plan with A*. The raw A* path is smoothed by B-spline fitting with collision checking to remove sharp turns before it is handed to the locomotion policy as waypoints.

Locomotion Between Interactions. Locomotion between scripted locations follows the waypoints on $\mathcal{O}_t$. Locomotion is generated by the policy of Sec. 3.2, which consumes the waypoints and the walkability grid computed on the current map, for the static scenes and for the chained object sequences of Tab. 4 alike. Because the path is planned on the dynamic map, locomotion avoids objects moved by earlier interactions.

Motion Inbetweening. Locomotion and interaction segments generally do not meet in matching poses. We use the latent motion model to generate the transitions in_ℓ and out_ℓ of Eq. 9: $n_{\text{seed}} = 3$ frames from the end of the preceding segment seed the model, and the first frame of the next segment is the target. A text prompt ("walk" or "turn") sets the style of the transition and is chosen automatically from the angle between the avatar's facing direction and the direction to the target.

Interaction Placement and Floor Alignment. Retrieved clips live in the coordinate frame of the capture system, which differs from the gravity-aligned scene frame. Converting the capture frame to the scene frame with a rotation $\mathbf{R}_{\text{coord}}$ is straightforward for the object, $\mathbf{R}^O_{\text{new}} = \mathbf{R}_{\text{coord}}\mathbf{R}^O_{\text{old}}$ and $\mathbf{t}^O_{\text{new}} = \mathbf{R}_{\text{coord}}\mathbf{t}^O_{\text{old}}$. For SMPL a subtlety arises: LBS rotates about the pelvis joint $\mathbf{J}_0$, not the origin, so naively rotating the translation introduces an offset proportional to $(\mathbf{R}_{\text{coord}} - \mathbf{I})\mathbf{J}_0$ and breaks the human–object contact of the clip. We correct for it with

$$\mathbf{R}_{\text{new}} = \mathbf{R}_{\text{coord}}\,\mathbf{R}_{\text{old}}, \tag{19}$$

$$\mathbf{t}_{\text{new}} = \mathbf{R}_{\text{coord}}\,(\mathbf{t}_{\text{old}} + \mathbf{J}_0(\boldsymbol{\beta})) - \mathbf{J}_0(\boldsymbol{\beta}), \tag{20}$$

where $\mathbf{R}_{\text{old}}, \mathbf{R}_{\text{new}}$ are the global orientations and $\mathbf{t}_{\text{old}}, \mathbf{t}_{\text{new}}$ the root translations before and after the change of frame, and $\mathbf{J}_0(\boldsymbol{\beta})$ the pelvis location for shape $\boldsymbol{\beta}$. The clip is then placed at the scripted location $\mathbf{p}_\ell$ by a rigid transform in the ground plane. Because locomotion and retrieved clips may assume slightly different floor heights, we run an SMPL forward pass on the boundary frames of both segments, compute the lowest body vertex, and shift the clip vertically so that the feet land on the scene floor. Remaining discrepancies are absorbed by the contact refinement of Sec. 3.4.

O Coordinate Frame and Height Normalization

In Sec. 3.1 we refer to the appendix for coordinate normalization. We provide it here, together with the avatar height alignment it enables.

Scene, human, and object Gaussians are reconstructed independently and live in different frames with different scales. Before compositing we align them to a shared metric frame.

Gravity Alignment via PCA. We compute PCA of the scene Gaussian centers $\{\boldsymbol{\mu}_i\}_{i=1}^{N_S}$ and take the eigenvector with the smallest eigenvalue as the vertical direction, since real scenes are wider than they are tall. The resulting rotation $\mathbf{R}_{\text{align}}$ is applied to all scene centers and covariances, $\boldsymbol{\mu}'_i = \mathbf{R}_{\text{align}}\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}'_i = \mathbf{R}_{\text{align}}\boldsymbol{\Sigma}_i\mathbf{R}_{\text{align}}^\top$, after which the vertical axis is aligned with gravity and the remaining two axes span the ground plane. The same $\mathbf{R}_{\text{align}}$ is reused for every orthographic projection of scene and object Gaussians. On elongated scenes such as hallways the two horizontal components may swap, which is harmless since only the vertical axis is used.

Ground Plane Estimation. Among Gaussians retained after opacity culling, we take the 5th percentile of the vertical coordinates as the floor height y_{floor}, which is robust to sub-floor outliers, and shift all centers so that the floor sits at zero.

Metric Scale. SMPL avatars are metric, whereas scene Gaussians can be at an arbitrary scale, especially for monocular reconstructions. For ScanNet++ the reconstruction is already metric. For SuperSplat and DL3DV scenes we estimate a scale factor $\eta = h_{\text{ref}}/h_{\text{scene}}$, where h_{ref} is the known height of a reference structure such as a door or standard furniture and h_{scene} its measured height in the reconstruction.

Avatar Placement. Given the normalized scene, the vertical component of the SMPL root translation at each frame is set so that the lowest body vertex touches the floor, $\mathbf{t}_t^y = \Delta y_t$, where Δy_t is the height of the pelvis above the lowest body vertex in the current pose and the floor sits at zero after normalization. This places the feet on the estimated ground before the contact refinement of Sec. 3.4 adjusts individual Gaussians.

P HYPERPARAMETERS

In Secs. 3.2, 3.3, and 3.4 and App. K we introduce loss and reward weights without numerical values and refer to the appendix for contact detection details. We provide them here.

Contact Detection. From the synthesized SMPL motion, let $\mathbf{x}_{c,t}$ be the position of contact joint c (feet, pelvis, hands) at frame t, with velocity $\mathbf{v}_{c,t} = (\mathbf{x}_{c,t} - \mathbf{x}_{c,t-1})/\Delta t$ and acceleration $\ddot{\mathbf{x}}_{c,t} = (\mathbf{v}_{c,t} - \mathbf{v}_{c,t-1})/\Delta t$, where Δt is the frame interval. A frame is a contact frame if the joint has small vertical velocity, its vertical acceleration does not exceed a threshold, and it is close to the target surface,

$$\delta_{c,t} = \big(|v_{c,t}^y| < \tau_v\big) \wedge \big(\ddot{x}_{c,t}^y < \tau_a\big) \wedge \big(\min_{\mathbf{m}\in\mathcal{V}_{c,t}} \rho(\mathbf{m};\Upsilon) < \epsilon\big), \tag{21}$$

where y is the vertical axis, τ_v, τ_a are thresholds (Tab. 11), $\mathcal{V}_{c,t}$ are the posed SMPL contact vertices of joint c at frame t, and ϵ is the separation margin of Eq. 8. The proximity term prevents a foot at the apex of a step, where the vertical velocity is also near zero, from being marked as in contact. For hand–object contact during retrieved clips, the contact frames are those in which the tracked hand joint lies within the separation margin ϵ of the object Gaussians.

Table 11: **Hyperparameters for all pipeline stages.** Grouped by stage: occupancy maps and walkability grid, RL locomotion reward weights (Eq. 18), interaction latent-space optimization (Eq. 5), and contact refinement (Eq. 7).

Stage / symbol	Description	Value
Occupancy maps and walkability grid		
τ_α	opacity culling threshold	0.5
τ_{occ}	walkability distance threshold (m)	0.05
M	occupancy map resolution	4096
N_g	egocentric grid resolution (1.6×1.6 m)	16
RL locomotion rewards		
w_{dist}	distance progress	1.0
w_{succ}	goal success	1.0
w_{ori}	orientation alignment	0.1
w_{skate}	foot skating penalty	100.0
w_{floor}	floor contact penalty	100.0
w_{pene}	scene penetration	1.0
Interaction optimization		
λ_{stop}	velocity at goal	1.0
λ_{start}	continuity with locomotion	1.0
λ_{coll}	collision avoidance	0.1
λ_{smooth}	temporal smoothness	0.1
learning rate	Adam step size	0.01
optimization steps	gradient iterations	100
guidance scale w_{cfg}	classifier-free guidance	5.0
$m_{\max}$	DDIM sampling steps	10
n_{seed}	inbetweening seed frames	3
Contact refinement		
τ_v	contact velocity threshold (m/s)	0.09
τ_a	contact acceleration threshold (m/s^2)	0.2
λ_{snap}	snap to nearest target Gaussian	1.0
λ_{prox}	contact / separation penalty	0.1
λ_{reg}	translation regularization	0.01
λ_{temp}	temporal smoothness	0.1
ϵ	separation margin (m)	0.05
κ	soft nearest-neighbor temperature	100