AVATARPILOT: LEARNING A CAMERA POLICY FOR ACTIVE 4D HUMAN AVATAR RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Capturing a moving person for 4D reconstruction typically relies on studio rigs with dozens of fixed cameras, which are costly and confine capture to the studio. An autonomous camera that moves around the person would be a far more flexible and cheaper alternative, but the quality of the reconstruction then depends on where the camera goes. We formulate this as active 4D human avatar reconstruction, a task in which a physically constrained camera decides at every step where to move next using only its onboard observations, to achieve a better 4D reconstruction from the images it captures. We solve this task by learning a camera policy with reinforcement learning. To train it, we turn hundreds of existing multi-view captures of real people into renderable 4D avatar assets made of 3D Gaussians, and build on them a pipeline from simulated onboard observations to camera actions, along with an evaluation protocol. We define the training reward directly on the avatars' surface Gaussians, which encourages the policy to see the body from new viewing directions. Experiments show that on held-out subjects, the learned policy outperforms every preset trajectory we compare against. When the person walks, the reconstructions from its captures exceed those from the strongest preset trajectory by 2.5 dB PSNR. When the person stays in place, one camera under our policy matches the reconstruction quality of six to eight fixed cameras.

1 INTRODUCTION

Many applications, from film production to telepresence, need 4D reconstructions of people, that is, captures of a moving person that can be viewed from any angle afterwards. Traditionally, such captures are obtained in a dedicated studio, where dozens of synchronised cameras surround the subject (sixty in DNA-Rendering (Cheng et al., 2023)). While effective, these multi-camera rigs are expensive and confined to a fixed location, so the subject has to come to the studio and perform within a limited capture space. A far more flexible and cheaper alternative is an autonomous camera, for example on a drone or a rail system, that moves around the person and follows them wherever they go. Such a setup needs no dedicated studio and allows 4D capture in far more environments.

Yet with one moving camera, the quality of the reconstruction is governed by the viewpoints it captures, and whether a viewpoint is worth taking depends on what the camera has already captured, on the person's current state and on what the camera can still reach. The capture history is which parts of the body have been seen and from which directions. The state is the person's pose and position, which decide which parts a viewpoint can see. Among these factors, dependence on the capture history is the basis of next-best-view methods for static scenes (Connolly, 1985; Scott et al., 2003; Jiang et al., 2024), which pick the next viewpoint from what has already been seen. For a moving person, the best viewpoint also depends on the person's current state. Moreover, the camera cannot jump to the most valuable viewpoint but moves at a bounded speed, so each move constrains which viewpoints come next and the choice becomes a sequential decision.

Existing systems that capture people with a moving camera, however, typically rely on a hand-designed preset trajectory, such as an orbit around the person (Zhou et al., 2018) or a spiral that sweeps the elevation (Xu et al., 2018; Kotcov et al., 2024). A preset trajectory fixes every viewpoint before the recording starts, and therefore takes neither the capture history nor the person's state into account. An orbit at one elevation never sees the body from elevations above or below its own, whatever has been captured so far. Nor does it know that the person has moved, so it easily loses

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.

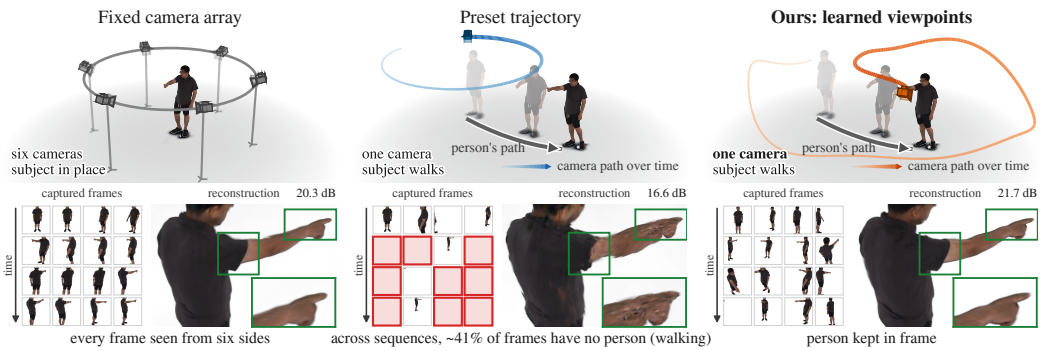


Figure 1: **Capturing a person in motion: a fixed array, a preset trajectory and a learned camera. Top row:** six cameras with the person in place, or one camera while the person walks. **Captured frames:** frames sampled uniformly over the episode, red where the person is not in the frame. **Reconstruction:** in place, same view and frame, by template-free 4DGS, with that view's PSNR.

them (Tallamraju et al., 2020). In our setting, these limitations lead to two failure cases. When the person walks, the presets miss the person in a substantial share of the frames. Even when the person stays in place, parts of the body are cut off at some of their viewpoints when the camera is close. Fig. 1 shows both cases. The preset trajectory reconstructs the person worse in place and loses them once they walk, whereas the learned camera avoids both. To capture well, the camera must therefore choose its next viewpoint online, from what it has already captured and from what it observes now.

We formulate this problem as active 4D human avatar reconstruction, in which a physically constrained camera decides at every step where to move next, using only its onboard observations, to achieve a better reconstruction from the images it captures. The observations are a depth map, a person mask and the camera pose, all available on a real camera, and include no ground-truth pose or position of the person. The camera's motion is limited by its platform, in our case the thrust of a consumer drone, so every viewpoint must be reached step by step, not chosen freely.

Designing such a decision rule by hand is difficult, because the value of a viewpoint depends jointly on all three factors above and the person's future motion is unknown. Once the physical limits of a drone are imposed, a hand-designed criterion that picks the best reachable viewpoint step by step ended up close to a plain orbit in prior work (Kiciroglu et al., 2020). This motivates reasoning about the long-term consequences of each viewpoint decision. We therefore learn a camera policy with reinforcement learning, optimised over whole captures rather than one step at a time. Training such a policy requires rendering a real person from wherever the camera moves, so we reconstruct hundreds of existing studio capture sequences into 4D avatars (Hu et al., 2024) of 3D Gaussians (Kerbl et al., 2023) attached to a body model, renderable from any viewpoint at any moment. Around these avatars we build a simulator and the full pipeline from onboard observations to camera motion. We further define an evaluation protocol that reconstructs the person from the captured images with two methods (Qian et al., 2024; Wu et al., 2024) and scores it at fixed test viewpoints against references built from all studio cameras. To make the policy see every part of the body from as many directions as possible, we build the reward on a property of the Gaussian avatars: the same Gaussians persist across frames, so each surface Gaussian can record the directions from which it has been seen. The policy is then rewarded only when some part of the body is seen from a new direction.

At our main operating point, the learned policy outperforms every preset trajectory we compare against (fixed camera, random walk, orbit and spiral) on held-out subjects under two reconstruction methods. When the person stays in place, one camera under our policy matches the reconstruction quality of six to eight fixed cameras. The policy learns to choose better viewpoints and to cover the body more completely than any preset, and its quality holds when the person walks or the camera comes closer, where every preset degrades (Sec. 4.3). It also transfers to a second studio dataset without retraining (Sec. 4.3). Our results are obtained in simulation, and we see them as a step toward autonomous cameras that capture moving people in the real world.

Our contributions are: **(1)** We formulate a new task, active 4D human avatar reconstruction, in which a physically constrained moving camera uses only its onboard observations to decide online where

to capture a dynamic human. **(2)** We build the complete pipeline, turning hundreds of real studio captures into a simulation and evaluation platform and learning a camera policy, *AvatarPilot*, on it with a reward defined on the avatars' surface Gaussians. **(3)** One camera under this policy matches six to eight fixed cameras when the person stays in place and outperforms every preset trajectory.

2 Related Work

Active view selection. Next-best-view selection chooses where to look next from what has been captured so far, classically from the geometry acquired (Connolly, 1985; Scott et al., 2003) and, for neural reconstruction, from the model's uncertainty or expected information gain (Pan et al., 2022; Ran et al., 2023; Jiang et al., 2024; Feng et al., 2024; Jin et al., 2025; Xie et al., 2025). Learned policies extend this to scanning objects and exploring indoor scenes with a coverage reward (Peralta et al., 2020; Chen et al., 2024; 2025), but the scene is still static and the target is geometry or coverage. Dynamic scenes have only begun to be addressed. Li et al. (2026b) select training frames offline from pre-recorded multi-camera video, DynActiveGS (Duan et al., 2026) reconstructs a scene while suppressing pedestrians as motion disturbance, and Li et al. (2026a) plan views for a rigid object moving in the plane. None of them controls a physical camera online around a moving person to reconstruct that person's appearance, which is the task we address.

Human capture with moving cameras. Active human capture has mostly targeted pose. ActiveMoCap (Kiciroglu et al., 2020) steers one drone around a person by choosing at each step the candidate viewpoint with the lowest predicted pose uncertainty. Multi-drone systems estimate pose and shape with optimised formations (Saini et al., 2019; 2022; Nägeli et al., 2018; Ho et al., 2021) or learn the formation with reinforcement learning (Tallamraju et al., 2020; Ci et al., 2023), and aerial cinematography learns shot selection and camera motion for cinematic objectives (Gschwindt et al., 2019; Bonatti et al., 2020; Huang et al., 2019). Gärtner et al. (2020) select among the fixed cameras of a dome, and Zhou et al. (2018) orbit one drone around the person. Closer to reconstruction, iHuman3D (Cheng et al., 2018) plans the views of one RGB-D drone by information gain in a TSDF volume for a person who may move only slightly, FlyCap (Xu et al., 2018) scans a template on a spiral and then follows the person with three drones, and AirNeRF (Kotcov et al., 2024) flies a fixed spiral around a standing person to fit a NeRF (Mildenhall et al., 2020). DroNeRF (Patel et al., 2023) and UAV4D (Choi et al., 2026) place drones around a static object or a still person, or render people from footage. In these systems the body is reconstructed only while the person holds still, and the systems that follow a moving person recover their pose or geometry but not their appearance.

Our problem differs in four ways. They optimise joint or surface error, whereas we focus on the quality of the 4D appearance reconstructed from the captured images. They pick the next view one step ahead, fly a preset trajectory or coordinate a formation, whereas we learn a policy for one physically constrained camera that is optimised over the whole capture. Their learned policies are given the person's 3D position as an input (Tallamraju et al., 2020; Ci et al., 2023), whereas ours sees only an onboard depth map, a person mask and the camera's pose. Their experiments use synthetic characters (Kiciroglu et al., 2020; Tallamraju et al., 2020; Ci et al., 2023), whereas our simulator and references are built from real multi-view captures. We therefore pose this as a new task, provide a platform on which it can be trained and compared, and learn a policy on it.

Dynamic human reconstruction. Dynamic humans are reconstructed with or without a body template. Template-based methods drive a neural field or a set of Gaussians with SMPL (Loper et al., 2015; Pavlakos et al., 2019) and learn it from multi-view or monocular video (Weng et al., 2022; Hu et al., 2024; Qian et al., 2024; Li et al., 2024; Moon et al., 2024), whereas template-free methods model the dynamic scene directly (Wu et al., 2024). Both kinds of method reconstruct from an existing capture, whereas we ask which views to capture so that they reconstruct the person better.

3 Method

3.1 Problem Formulation

Task and episode. One camera records a moving person for T time steps and decides at every step where to move next, using only what it has observed so far, and the $T + 1$ images it captures are then used to reconstruct the person in 4D. Time is discretised into $T = 75$ steps of $\Delta t = 2/15\,\mathrm{s}$,

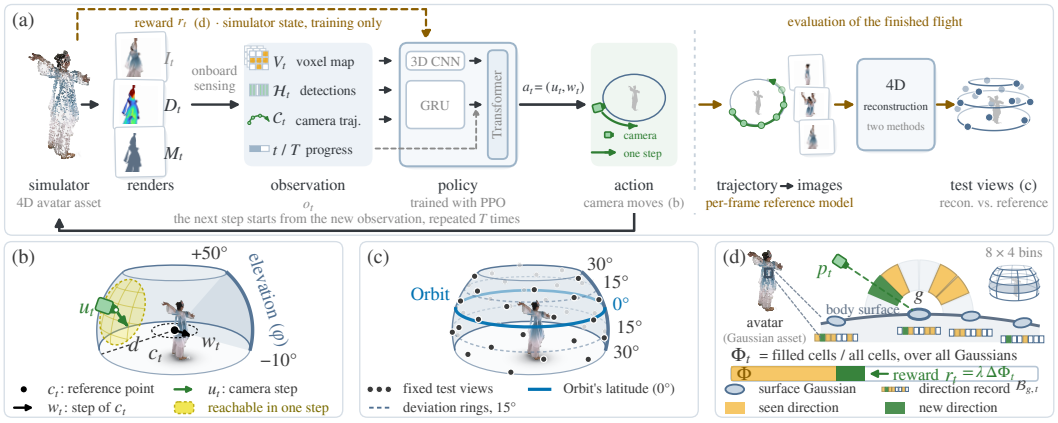


Figure 2: **Method overview.** **(a)** One step: the avatar is rendered at the current viewpoint, the observation o_t is built from depth and mask, and the policy outputs the next camera motion; the reward (dashed) exists only in training. **(b)** The action, one step on the sphere of radius d around c_t within the elevation limits. **(c)** The test views and the rings of Fig. 4c. **(d)** Each surface Gaussian records the directions it has been seen from; the reward is the gain in Φ_t.

so that one episode covers the 150 frames of a 15 fps recording at two frames per step (Sec. 4.1), and at each step t the camera records an RGB image I_t and a depth map D_t, both of which can be obtained from an onboard RGB-D camera. The outcome of an episode is the sequence of camera poses $\tau = (p_0, \ldots, p_T)$ together with the images taken at them. These images are given to a reconstruction method, which returns a 4D reconstruction $R(\tau)$, and its quality $Q(\tau)$ is measured by rendering it at a fixed set of test views $\mathcal{V}$ and comparing the renderings with references (Sec. 3.4). Ideally the policy π maximises $\mathbb{E}_{\tau\sim\pi}[Q(\tau)]$, but evaluating Q requires training a reconstruction for every capture, which is far too expensive for the many thousands of episodes that reinforcement learning needs and would give only one reward per episode. We therefore train with a proxy reward r_t that can be computed at every step and rewards what a good capture needs, seeing every part of the body from as many directions as possible (Sec. 3.3).

Camera state and actions. The camera moves on a sphere of radius d (the capture distance) around a reference point c_t and always faces it, which keeps the task simple: a viewpoint is just an azimuth and an elevation, and the camera's orientation needs no control of its own. The reference point lies on a horizontal plane at a fixed height and changes only through the policy's action, so the policy has to move it along when the person walks away. The action is thus $a_t = (u_t, w_t)$: u_t moves the camera along the sphere, and w_t moves the reference point (Fig. 2b; the exact update is given in App. A.1). The camera's position is $p_t = c_t + d\,\mathbf{n}(\theta_t, \phi_t)$, where $\mathbf{n}$ is the unit vector at azimuth θ_t and elevation $\phi_t \in [-10°, 50°]$, above the floor and short of top-down views. The capture distance d is fixed within an experiment, so that every strategy is compared on a sphere of the same radius.

Physical limits. We impose the motion limits of a real drone, so that the trajectories the policy learns are physically feasible and every strategy is compared under the same constraint. The camera's displacement per step in the world, including the motion of the sphere's centre, is bounded by $\|p_{t+1} - p_t\| \le s_{\max}$, since a drone has one motion budget. The limit is derived from the thrust-to-weight ratio of a consumer multirotor rather than chosen by hand. Because a wider circle needs less acceleration at the same speed, $s_{\max}$ grows with $\sqrt{d}$ (App. A.2). The capture distance can thus be varied while modelling the same drone, and every strategy at every distance obeys the same limit.

3.2 Simulator from Real Captures

The simulator has to render the person from wherever the camera goes, and we build it from two assets derived from studio captures. We use DNA-Rendering (Cheng et al., 2023), in which 60 synchronised cameras record a person acting in place for 10 s at 15 fps, with per-frame SMPL-X fits and foreground masks. The first asset is a 4D avatar $\mathcal{A}$ for simulation, fitted to each sequence with GauHuman (Hu et al., 2024), in which 3D Gaussians attached to the SMPL-X body are deformed by

linear blend skinning. It can be rendered from any viewpoint at any frame, $(I_t, D_t, \alpha_t) = \mathcal{A}(p_t, t)$, returning an RGB image, the expected depth and an alpha map. Because the same Gaussians persist across frames, every surface element keeps its identity throughout the episode, which is what lets the reward of Sec. 3.3 record for each element the directions from which it has been seen. The second asset is a per-frame reference model $\mathcal{R}_t$ that stands in for a real camera at evaluation. It is a static 3D Gaussian reconstruction fitted to every frame of every held-out sequence from all 60 views with MaskGaussian (Liu et al., 2025) without any shared template, and it renders each frame more faithfully than the avatar. Being fitted per frame, it has no surface element that persists across frames, so it serves evaluation rather than training. At evaluation the policy flies in the simulator built on $\mathcal{A}$, and its trajectory is re-rendered from $\mathcal{R}_t$ at the same camera poses to produce the images given to the reconstruction; the references it is compared against at the test views also come from $\mathcal{R}_t$ (Sec. 3.4). The policy thus only ever sees avatar renderings, while the reconstruction receives images close to what a real camera would have photographed along the trajectory.

3.3 Policy Learning

Observation. The observation carries the capture history, the current state of the person and the camera's own past poses. The policy uses only the depth map D_t, the person mask M_t and the camera's own pose (in simulation $M_t = \{p : \alpha_t(p) > 0.5\}$ comes from the rendered alpha map, and on a real camera D_t and M_t would come from a depth sensor and a segmentation model (Kirillov et al., 2023)). Back-projecting the mask pixels with D_t gives the visible surface points $\mathcal{P}_t$, whose centroid $\hat{c}_t$ is the centre of the visible surface. A step with too few mask pixels counts as a missed detection. The capture history is a voxel map V_t, a 48^3 grid centred on the visible-surface centre $\hat{c}_t$ and aligned with the world axes, updated cumulatively with the visible surface points $\mathcal{P}_t$ at every step. Each cell records three channels, how many times it has been seen, from how many directions and how recently. The direction channel is the number of azimuth and elevation bins ($K = 8 \times 4$) in which the camera has observed the cell. The state of the person is a detection history $\mathcal{H}_t$ over the last H steps, recording the offset of the visible-surface centre $\hat{c}_t$ from the reference point and whether the person was detected, so the policy knows where the person is relative to the sphere's centre and when it has lost sight of them (App. A.1). The final observation is $o_t = (V_t, \mathcal{H}_t, \mathcal{C}_t, t/T)$, where $\mathcal{C}_t$ holds the camera's own poses over the same H steps and t/T is the episode progress (Fig. 2a). Every entry of o_t thus comes from the onboard depth stream and the camera's own poses, with no privileged information such as the person's pose or true position (Table A10, row d).

Reward. To turn the goal of seeing every part of the body from as many directions as possible into a reward, we record for each surface element the directions from which it has been seen and reward each new direction. Let $\mathcal{G}$ be the surface Gaussians of the avatar, and let the camera's azimuth and elevation range be divided evenly into $K = 8 \times 4 = 32$ bins, one per viewing direction. Each Gaussian g keeps the set $\mathcal{B}_{g,t} \subseteq \{1, \dots, K\}$ of bins it has been seen from (Fig. 2d), updated as

$$\mathcal{B}_{g,t} = \mathcal{B}_{g,t-1} \cup \{b_t\} \text{ if } g \text{ is visible at step } t, \qquad \mathcal{B}_{g,t} = \mathcal{B}_{g,t-1} \text{ otherwise,} \tag{1}$$

where b_t is the bin of the camera's azimuth and elevation (θ_t, ϕ_t) at step t. Keeping this record per element is possible because the same Gaussians persist across frames (Sec. 3.2). The potential Φ_t is the fraction of directions a Gaussian has been seen from, averaged over all Gaussians, and the reward is its increase from one step to the next,

$$\Phi_t = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \frac{|\mathcal{B}_{g,t}|}{K}, \qquad r_t = \lambda \, (\Phi_t - \Phi_{t-1}). \tag{2}$$

The rewards of an episode add up to $\lambda(\Phi_T - \Phi_0)$, so maximising the return is the same as maximising the directional diversity reached at the end, while every step still receives an immediate signal. Reward is earned only by seeing part of the body from a direction it has not been seen from, so keeping the person in the frame is rewarded without a separate term. The reward uses information that only the simulator has, namely which surface elements are visible at each step. It enters the training signal only, and the deployed policy relies on onboard observations alone, as is common in simulation-trained control (Pinto et al., 2018; Chen et al., 2020). We reward new directions rather than binary surface coverage, which is nearly complete after one pass around the person, whereas a new direction still scores after every part has been seen once (ablated in Table A10, row a).

Training. To map the observation o_t to an action, we use a small network of standard components, similar to those used for learned exploration of static scenes (Chen et al., 2025). A 3D convolutional

encoder turns the voxel map V_t into a set of tokens, and the two histories $\mathcal{H}_t$ and $\mathcal{C}_t$ are projected per step and summarised by a GRU into one token with the progress t/T. A two-layer Transformer (Vaswani et al., 2017) fuses all tokens into a feature from which the action distribution and a value estimate are read out. We train with PPO (Schulman et al., 2017), with actor and critic sharing the encoder and both seeing only o_t, so privileged information never enters the network. Since studio recordings show people acting in place, we simulate everyday walking by carrying the person along a walking path in randomly chosen training episodes (Sec. 4.1), so one policy serves both conditions.

3.4 Evaluation Protocol

Reconstruction quality as the metric. The value of a capture lies in what can be reconstructed from it, so we score every strategy by the reconstruction trained on its images, which makes $R(\tau)$ and $Q(\tau)$ of Sec. 3.1 concrete. The $T + 1$ images along a trajectory, rendered from $\mathcal{R}_t$ (Sec. 3.2), are handed to the reconstruction method with their masks and camera poses. The resulting reconstruction $R(\tau)$ is rendered at the test views $\mathcal{V}$ and compared with renderings of $\mathcal{R}_t$ at the same views by PSNR, SSIM and LPIPS, all computed on a crop around the person taken from the reference mask (App. B.2). We use two reconstruction methods. The template-based 3DGS-Avatar (Qian et al., 2024) attaches Gaussians to a body model driven by the SMPL parameters that come with the dataset, whereas the template-free 4D Gaussian Splatting (4DGS) (Wu et al., 2024) uses only the images and camera poses. A body template fills in parts the camera never saw and relies on SMPL parameters that a deployed system would not have, so it can partly mask differences between strategies. These parameters are a privilege of the scoring side rather than of the policy, and Sec. 4.3 checks that estimated ones change the comparison little (App. C.2). The template-free method serves as an independent check, and we report both throughout.

Test views and fair comparison. For a fair comparison, the test views $\mathcal{V}$ are fixed in advance and shared by every strategy. They lie on a sphere of radius d around the person and are spread over the elevation band so that each view covers an equal area (Fig. 2c). In the walking condition the test views move with the person frame by frame, using the true displacement, which serves only the scoring side and never reaches the policy. Each preset also obeys the limits of Sec. 3.1 and starts from the policy's azimuth, and the orbit and the spiral travel the same total distance as the policy.

4 Experiments

4.1 Experimental Setup

Data and conditions. We build the simulator and the evaluation from DNA-Rendering (Cheng et al., 2023), with 368 sequences for training and 24 held-out sequences from 10 subjects absent from the training set for evaluation, so every result is on unseen people. Each sequence is an episode of 75 camera steps. In the in-place condition the sequence is used as recorded, and in the walking condition the same recorded motion is carried along a path of waypoints inside a 3 m disc at 0.3–0.9 m/s, by applying one rigid motion of the whole body to both assets. Training assigns each episode the walking condition with probability 0.5, and the walking paths used in evaluation never appear in training. The main operating point is a capture distance of 2.0 m.

Baselines. We compare against four preset capture strategies that obey the same physical limits as the policy (the same per-step motion budget, capture distance and elevation range). A fixed camera stays at one viewpoint at 20° elevation, the midpoint of the elevation range. A random walk moves at every step in a random direction along the sphere with a random step length within the same per-step budget. An orbit circles the person at the constant elevation of 20°. A spiral circles the person while its elevation sweeps the elevation range once, from the bottom to the top and back. The orbit and the spiral are the trajectories flown by prior human capture systems (Zhou et al., 2018; Xu et al., 2018; Kotcov et al., 2024). Their number of turns is set so that they travel the same total distance as our policy, and they start from the same azimuth (Sec. 3.4, App. B.1).

Metrics and statistics. The images captured by each strategy are scored by the reconstruction built from them, under the template-based 3DGS-Avatar (Qian et al., 2024) and the template-free 4DGS (Wu et al., 2024) (Sec. 3.4). Each reconstruction is rendered at the same 48 test views for every strategy, spread uniformly by area over the elevation band and moving with the walking person, so absolute values are comparable across strategies and capture distances. We report PSNR and LPIPS

Table 1: **Comparison with preset trajectories.** Reconstruction quality at the test views, in place and while the person walks, with the template-based 3DGS-Avatar and the template-free 4DGS. Per column, the best, second and third values are shaded.

Reconstruction	3DGS-Avatar				4DGS			
	In place		Walking		In place		Walking	
Capture	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
Fixed camera	13.26⋆	0.375⋆	13.29⋆	0.374⋆	8.32⋆	0.619⋆	9.58⋆	0.565⋆
Random walk	17.82⋆	0.248⋆	15.78⋆	0.308⋆	15.05⋆	0.346⋆	12.39⋆	0.419⋆
Orbit	17.88⋆	0.202⋆	16.25⋆	0.251⋆	16.00⋆	0.288⋆	13.67⋆	0.360⋆
Spiral	19.92⋆	0.190⋆	17.62⋆	0.248⋆	16.68⋆	0.260	13.92⋆	0.347⋆
OURS	**20.18**	**0.184**	**20.16**	**0.184**	**17.26**	**0.255**	**17.26**	**0.256**

in the main text and SSIM in App. C.1. Differences are paired per sequence over the 24 held-out sequences. Our policy is trained with three seeds, and a ⋆ marks a difference whose 95% bootstrap confidence interval excludes zero for each seed separately (Agarwal et al., 2021), so the gain is consistent across sequences and seeds. Settings are in Table A1 (App. A.1). **All differences Δ are signed so that a positive value means our policy is better**, i.e. ours minus the baseline for PSNR and SSIM and the baseline minus ours for LPIPS.

4.2 Main Results

Comparison with preset trajectories. Table 1 compares our policy with the four preset strategies at the main operating point of 2.0 m. One policy, trained on both conditions, beats every preset strategy in both conditions and under both reconstruction methods. All sixteen PSNR comparisons are significant on all three seeds, and LPIPS and SSIM (App. C.1) agree in direction. The closest preset is the spiral, which our policy leads by $+0.27$ dB⋆ in the in-place condition and by $+2.55$ dB⋆ in the walking condition, where it leads the orbit by $+3.92$ dB⋆. Under the same limits the random walk does no better than the orbit, so moving alone is not enough. Fig. 3 shows these differences, with further examples and per-pixel error in App. C.4 and a video in the supplementary material. The presets lose or blur parts of the body and clothing in the in-place condition and lose or break whole parts in the walking condition, whereas ours stays close to the reference in both. Under the template-free method every comparison keeps its direction, so the advantage does not come from the body template. When the person walks, the template-free method is fitted in a coordinate frame that moves with the person (App. C.1), and there our policy leads the spiral by $+3.34$ dB⋆.

Comparison with a greedy planner. We test two planners of the kind used as baselines in next-best-view work (Chen et al., 2024; 2025; Jiang et al., 2024), hand-designed rules that maximise our reward one step or three steps ahead. Both read the simulator's coverage record and the true position of the person, neither of which enters our policy's observation, and obey the same motion budget (App. B.3). Our policy reconstructs 0.7–0.9 dB better than either planner in both conditions on every sequence (Table A4 in App. B.3), although both reach a higher reward on average (App. D.2).

Comparison with fixed multi-camera arrays. The usual way to capture a person is to surround them with fixed cameras, so we compare our one moving camera with fixed-camera arrays of $N \in \{1, 2, 3, 4, 6, 8\}$ cameras spaced evenly on the 20° ring at the same capture distance, in the in-place condition (Fig. 4a). The reconstruction quality of the fixed-camera arrays rises with N and reaches that of our one moving camera between six and eight cameras at the main operating point. Our one camera matches four to six fixed cameras at $d = 3.0$ m and six to eight at $d = 2.0$ m. As the capture distance shrinks, a camera on that ring keeps less of the body inside its image and the arrays lose quality, whereas our policy stays within 0.35 dB across the four distances.

4.3 Analysis

Learned behaviour. The reward pays only for seeing part of the body from a new direction, and the policy turns it into two behaviours. It sweeps elevation while it circles the person instead of staying on one ring (Fig. 4b), and sees the person from 92% of the 8×4 direction bins on average, against

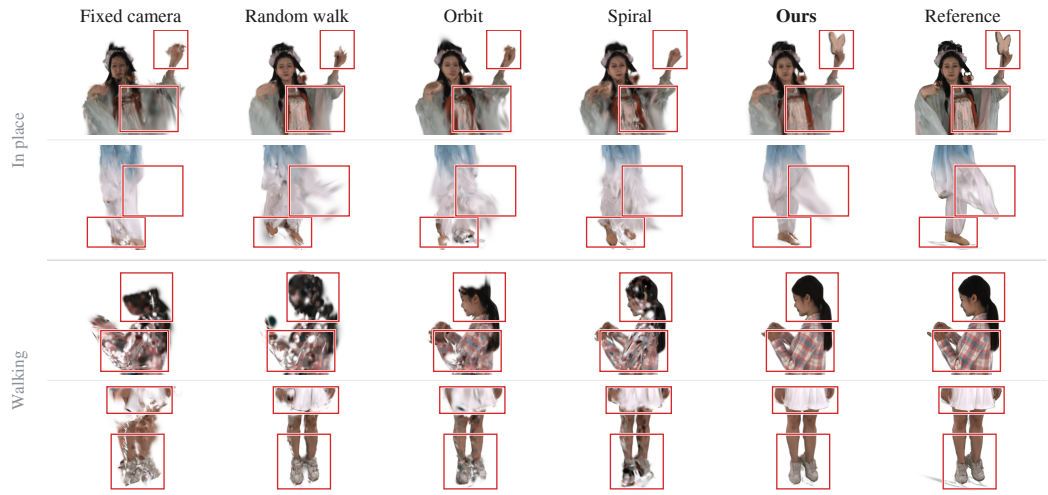


Figure 3: **Reconstruction quality under each capture strategy.** Novel views of each reconstruction; rows 1–2 captured in place, rows 3–4 while the person walks. Same frame and view per row.

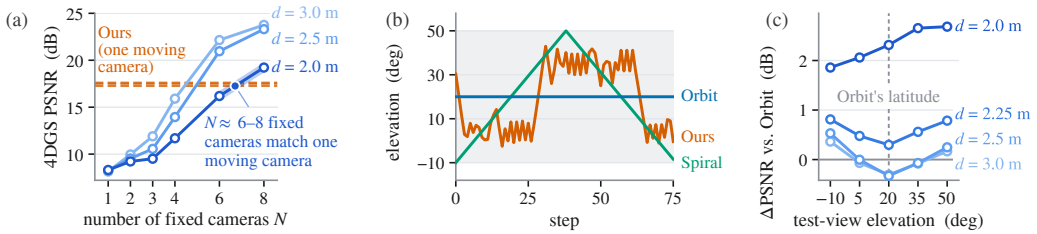


Figure 4: **Fixed-camera arrays and gain by viewing direction, in-place condition. (a)** Reconstruction quality (template-free) for N fixed cameras on the 20° ring (blue) and for our one moving camera (orange), at three capture distances; every fixed camera records every frame. One moving camera under our policy matches six to eight fixed cameras at the main operating point. **(b)** Elevation over time for one sequence in the in-place condition, first training seed. **(c)** PSNR gain over the orbit by elevation ring (template-based), at four capture distances; dashed at the orbit's ring.

25% for the orbit and 83% for the spiral. It also keeps the walking person in the image although nothing in the reward refers to following (Fig. A6).

Gain by viewing direction. In the in-place condition a gain over a preset trajectory comes from choosing better viewpoints rather than from following the person. The orbit stays at one elevation, so we group the test views into rings of constant elevation and compare our policy with the orbit ring by ring at the four capture distances with the template-based method (Fig. 4c). At the two loosest distances the orbit is ahead only on its own ring at 20°, the two are about equal within 15° of that ring, and our policy is ahead everywhere farther away. At the two tightest distances the whole curve lifts, and at 2.0 m our policy leads by more than 1.8 dB at every ring within 30° of the orbit's.

Response to motion. From the in-place to the walking condition in Table 1, the orbit and the spiral drop by 1.6–2.8 dB, whereas our policy drops by at most 0.02 dB. At the main operating point the orbit and the spiral capture no person at all in about 41% of their frames and our policy in under 0.1% (Fig. 1 and App. B.2). When each preset is given our own estimate of the person's position and moves the centre of its path with it under the same motion budget, the empty frames vanish and the orbit gains 1.7 dB and the spiral 2.3 dB. Even so, our policy leads these strengthened presets by 2.17 and 0.27 dB (App. B.3). Beyond following the person, our policy also responds better to the motion of the body itself, since at the loosest distance it leads the spiral by 0.80 dB more in the most-moving quarter of the frames than in the least-moving quarter (Table A8 in App. C.2).

Effect of capture distance. Close capture resolves finer detail, but it is also where a preset trajectory struggles to keep the body in frame, so we evaluate at four capture distances from 3.0 m to 2.0 m. At

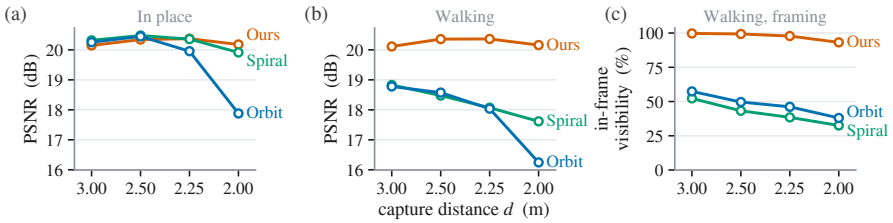


Figure 5: **Effect of capture distance.** Reconstruction quality (template-based) **(a)** in place and **(b)** walking, and **(c)** the fraction of the body surface facing the camera that falls inside the image.

each distance one policy is trained and the presets fly on that sphere, distance-matched, and every capture is scored at that distance's test views, so the values in Fig. 5 are comparable across distances. Our policy outperforms both presets at every distance while walking (Fig. 5b), and in place the three strategies are close at the looser distances and our advantage grows as the framing tightens (Fig. 5a). From 3.0 m to 2.0 m the orbit loses 2.4 dB and the spiral 0.4 dB in place, and 2.5 dB and 1.2 dB while walking, whereas our policy stays within 0.05 dB in both. The template-free 4DGS gives the same pattern in the in-place condition (Fig. A1 in App. C.2). Fig. 5c shows why for the walking condition: the presets keep only 32–57% of the body surface facing the camera inside the image because they keep circling the person's starting position, whereas our policy keeps 93–99.7% (App. B.2). To check that the comparison does not depend on where reconstructions are scored, we also score at four test-view radii, which moves the differences by at most 0.14 dB (App. C.2).

Robustness. Adding noise to the onboard observations at test time costs at most 0.41 dB at the heaviest of four noise levels (App. B.4). At that level the policy still leads a distance-matched spiral by 2.2 dB while walking, whereas in place its lead disappears. Replacing the SMPL parameters that the template-based method takes from the dataset by noisy versions or by GVHMR estimates from one fixed camera (Shen et al., 2024) lowers every strategy but preserves the ranking (App. C.2). Trained on DNA-Rendering alone, the policy transfers zero-shot to MVHumanNet (Xiong et al., 2024) and beats the orbit and the spiral there by margins close to those in Table 1 (App. C.3).

Ablations. Each design choice of Sec. 3 was ablated against the same final policy (Table A10 in App. D.1). Rewarding binary surface coverage instead of directional diversity (a), as learned exploration policies for static scenes do (Chen et al., 2024; 2025), lowers reconstruction quality in both conditions. That policy moves on as soon as a surface has been seen once. Two smoothness penalties added on top of our reward make no measurable difference (b, c). Replacing the onboard observations by a privileged map registered to the ground-truth body changes quality by at most 0.17 dB (d), so little is lost by relying on onboard observations alone. A Cartesian velocity action (e) reaches the same peak reward but collapses in every run, after which the camera stays at one spot, so we report no reconstruction score for it. Letting the policy turn the camera by up to $\pm 30^\circ$ instead of always facing the reference point (f) costs reconstruction quality in both conditions with the same budget. The policy spends that freedom on aiming to keep the person near the optical axis and follows the person less, so we keep the look-at lock.

5 Conclusion and Limitations

We posed the task of active 4D human avatar reconstruction, turned hundreds of studio captures into a simulation and evaluation platform, and learned a camera policy that acts on onboard observations alone, with a reward defined on the avatars' surface Gaussians. On held-out subjects the policy outperforms every preset trajectory we compare against, and one camera under it matches six to eight fixed cameras when the person stays in place. Its reconstruction quality is nearly unchanged by the person's motion and by the capture distance, where every preset trajectory degrades.

Limitations. All results are obtained in simulation, under conditions cleaner than real ones. The walking condition moves a person recorded in place along a path rather than recording real walking. Real flight, multi-camera coordination and multi-person capture are the next steps.

AI USE STATEMENT

We used generative AI tools for research ideation and execution, including feedback on experiment design, implementation of code and interpretation of results, for finding related work, and for drafting and polishing the text. We did not use them to generate data or to formulate or prove mathematical claims. All AI-assisted content was reviewed by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses the DNA-Rendering and MVHumanNet datasets under their terms of use for research, collects no new data on people and does not redistribute the recordings.

REPRODUCIBILITY STATEMENT

Sec. 3 specifies the task, the simulator, the policy and the evaluation protocol, Sec. 4.1 and App. A.1 the settings and the statistical protocol, and App. B the preset trajectories and the evaluation details. The appendix lists the data split, the training seeds and every ablation setting. Code, the trained policies and the evaluation scripts will be released upon publication.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Rogerio Bonatti, Wenshan Wang, Cherie Ho, Aayush Ahuja, Mirko Gschwindt, Efe Camci, Erdal Kayacan, Sanjiban Choudhury, and Sebastian Scherer. Autonomous aerial cinematography in unstructured environments with learned artistic decision-making. *Journal of Field Robotics*, 37 (4):606–641, 2020.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on robot learning*, pp. 66–75. PMLR, 2020.

Xiao Chen, Quanyi Li, Tai Wang, Tianfan Xue, and Jiangmiao Pang. Gennbv: Generalizable next-best-view policy for active 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16436–16445. IEEE, 2024.

Xiao Chen, Tai Wang, Quanyi Li, Tao Huang, Jiangmiao Pang, and Tianfan Xue. Gleam: Learning generalizable exploration policy for active mapping in complex 3d indoor scenes. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5558–5568. IEEE, 2025.

Wei Cheng, Lan Xu, Lei Han, Yuanfang Guo, and Lu Fang. ihuman3d: Intelligent human body 3d reconstruction using a single flying camera. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1733–1741, 2018.

Wei Cheng, Ruixiang Chen, Siming Fan, Wanqi Yin, Keyu Chen, Zhongang Cai, Jingbo Wang, Yang Gao, Zhengming Yu, Zhengyu Lin, et al. Dna-rendering: A diverse neural actor repository for high-fidelity human-centric rendering. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19925–19936. IEEE, 2023.

Jaehoon Choi, Dongki Jung, Chris Maxey, Sungmin Eum, Yonghan Lee, Dinesh Manocha, and Heesung Kwon. Uav4d: Dynamic neural rendering of human-centric uav imagery using gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3372–3380, 2026.

Hai Ci, Mickel Liu, Xuehai Pan, Fangwei Zhong, and Yizhou Wang. Proactive multi-camera collaboration for 3d human pose estimation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CPIy9TWFYBG.

C. Connolly. The determination of next best views. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pp. 432–435. IEEE, 1985.

Hongbo Duan, Pengting Luo, Chengzhi Zhao, Yuanhao Chiang, Fangming Liu, and Xueqian Wang. Dynactivegs: Active gaussian splatting for dynamic scene reconstruction. *arXiv preprint arXiv:2608.01178*, 2026.

Ziyue Feng, Huangying Zhan, Zheng Chen, Qingan Yan, Xiangyu Xu, Changjiang Cai, Bing Li, Qilun Zhu, and Yi Xu. Naruto: Neural active reconstruction from uncertain target observations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21572–21583. IEEE, 2024.

Erik Gärtner, Aleksis Pirinen, and Cristian Sminchisescu. Deep reinforcement learning for active human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10835–10844, 2020.

Mirko Gschwindt, Efe Camci, Rogerio Bonatti, Wenshan Wang, Erdal Kayacan, and Sebastian Scherer. Can a robot become a movie director? learning artistic principles for aerial cinematography. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1107–1114. IEEE, 2019.

Cherie Ho, Andrew Jong, Harry Freeman, Rohan Rao, Rogerio Bonatti, and Sebastian Scherer. 3d human reconstruction in the wild with collaborative aerial cameras. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5263–5269. IEEE, 2021.

Shoukang Hu, Tao Hu, and Ziwei Liu. Gauhuman: Articulated gaussian splatting from monocular human videos. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20418–20431. IEEE, 2024.

Chong Huang, Chuan-En Lin, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Cheng. Learning to film from professional human motion videos. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4239–4248. IEEE, 2019.

Wen Jiang, Boshu Lei, and Kostas Daniilidis. FisherRF: Active view selection and mapping with radiance fields using Fisher information. In *European Conference on Computer Vision*, pp. 422–440. Springer, 2024.

Liren Jin, Xingguang Zhong, Yue Pan, Jens Behley, Cyrill Stachniss, and Marija Popović. Activegs: Active scene reconstruction using gaussian splatting. *IEEE Robotics and Automation Letters*, 10 (5):4866–4873, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), 2023.

Sena Kiciroglu, Helge Rhodin, Sudipta N Sinha, Mathieu Salzmann, and Pascal Fua. Activemocap: Optimized viewpoint selection for active human motion capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 103–112, 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Alexey Kotcov, Maria Dronova, Vladislav Cheremnykh, Sausar Karaf, and Dzmitry Tsetserukou. Airnerf: 3d reconstruction of human with drone and nerf for future communication systems. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pp. 1067–1072. IEEE, 2024.

Karen Li, Mattia Mantovani, Robert J Wood, Lorenzo Sabattini, and Stephanie Gil. Motion-uncertainty-aware next-best-view planning for moving object reconstruction. *arXiv preprint arXiv:2605.17593*, 2026a.

Yiqian Li, Wen Jiang, and Kostas Daniilidis. Next best view selections for semantic and dynamic 3d gaussian splatting. In *2026 International Conference on 3D Vision (3DV)*, pp. 1916–1926. IEEE, 2026b.

Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19711–19722. IEEE, 2024.

Yifei Liu, Zhihang Zhong, Yifan Zhan, Sheng Xu, and Xiao Sun. Maskgaussian: Adaptive 3d gaussian representation from probabilistic masks. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 681–690. IEEE, 2025.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

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*, 2020.

Gyeongsik Moon, Takaaki Shiratori, and Shunsuke Saito. Expressive whole-body 3d gaussian avatar. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Tobias Nägeli, Samuel Oberholzer, Silvan Plüss, Javier Alonso-Mora, and Otmar Hilliges. Flycon: Real-time environment-independent multi-view human pose estimation with aerial vehicles. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *European Conference on Computer Vision*, pp. 230–246. Springer, 2022.

Dipam Patel, Phu Pham, and Aniket Bera. Dronerf: Real-time multi-agent drone pose optimization for computing neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5050–5055. IEEE, 2023.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA 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*, pp. 10975–10985, 2019.

Daryl Peralta, Joel Casimiro, Aldrin Michael Nilles, Justine Aletta Aguilar, Rowel Atienza, and Rhandley Cajote. Next-best view policy for 3d reconstruction. In *European Conference on Computer Vision Workshops*, pp. 558–573. Springer, 2020.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Robotics: Science and Systems (RSS)*, 2018.

Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5020–5030. IEEE, 2024.

Yunlong Ran, Jing Zeng, Shibo He, Jiming Chen, Lincheng Li, Yingfeng Chen, Gimhee Lee, and Qi Ye. Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters*, 8(2):1125–1132, 2023.

Nitin Saini, Eric Price, Rahul Tallamraju, Raffi Enficiaud, Roman Ludwig, Igor Martinovic, Aamir Ahmad, and Michael J Black. Markerless outdoor human motion capture using multiple autonomous micro aerial vehicles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 823–832, 2019.

Nitin Saini, Elia Bonetto, Eric Price, Aamir Ahmad, and Michael J Black. Airpose: Multi-view fusion network for aerial 3d human pose and shape estimation. *IEEE Robotics and Automation Letters*, 7(2):4805–4812, 2022.

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

William R Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys (CSUR)*, 35(1):64–96, 2003.

Zehong Shen, Huaijin Pi, Yan Xia, Zhi Cen, Sida Peng, Zechen Hu, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. World-grounded human motion recovery via gravity-view coordinates. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Rahul Tallamraju, Nitin Saini, Elia Bonetto, Michael Pabst, Yu Tang Liu, Michael J Black, and Aamir Ahmad. Aircaprl: autonomous aerial human motion capture using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4):6678–6685, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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*, pp. 16210–16220, 2022.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320. IEEE, 2024.

Yuhan Xie, Yixi Cai, Yinqiang Zhang, Lei Yang, and Jia Pan. Gauss-mi: Gaussian splatting shannon mutual information for active 3d reconstruction. In *Proceedings of Robotics: Science and Systems*, 2025.

Zhangyang Xiong, Chenghong Li, Kenkun Liu, Hongjie Liao, Jianqiao Hu, Junyi Zhu, Shuliang Ning, Lingteng Qiu, Chongjie Wang, Shijie Wang, et al. MVHumanNet: A large-scale dataset of multi-view daily dressing human captures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19801–19811, 2024.

Lan Xu, Yebin Liu, Wei Cheng, Kaiwen Guo, Guyue Zhou, Qionghai Dai, and Lu Fang. Flycap: Markerless motion capture using multiple autonomous flying cameras. *IEEE transactions on visualization and computer graphics*, 24(8):2284–2297, 2018.

Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025.

Xiaowei Zhou, Sikang Liu, Georgios Pavlakos, Vijay Kumar, and Kostas Daniilidis. Human motion capture using a drone. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 2027–2033. IEEE, 2018.

APPENDIX

A IMPLEMENTATION DETAILS

A.1 SIMULATOR AND TRAINING

Table A1: **Experimental settings.**

Item	Setting
Dataset & split	368 training / 24 held-out sequences (10 subjects), subject-disjoint
Episode	76 captures over 10 s, two frames of a 15 fps recording per step ($\Delta t = 2/15$ s), i.e. 75 camera steps
Conditions	in place, and walking (waypoints in a 3 m disc at 0.3–0.9 m/s); training mixes them with probability 0.5
Action space & physical limits	sphere of radius d, 2.0 m at the main operating point and 3.0, 2.5, 2.25 m in Sec. 4.3; elevation $[-10^\circ, 50^\circ]$; per-step budget $s_{\max}$ shared by sphere motion and reference-point motion, 0.776 m at $d = 2.0$ m (thrust-to-weight 2.09, App. A.2); reference point moves at most 0.3 m per step
Onboard camera	256×256, 40° field of view, colour and depth
Policy & training	PPO; 368 parallel environments $\times$ 128 steps, minibatch 2048, 3 epochs, learning rate 0.0001, clip 0.2, entropy 0.01, target KL 0.2, 300 iterations; observation history 24 steps
Reward	gain in directional diversity, 8×4 bins per surface Gaussian in the world frame, $\lambda = 60$
Test views	48 views, area-uniform over the elevation band, identical for every strategy; under walking the whole set translates with the person
Reconstruction methods	3DGS-Avatar (template-based) and 4DGS (template-free)
Baselines	fixed camera, random walk, orbit (20°), spiral
Distance matching	orbit and spiral matched to our policy on total distance travelled in the world frame, within 3% for all but one training seed (App. B.1)
Statistics	Δ = ours $-$ baseline (positive favours ours), paired per sequence over 24 held-out sequences; three training seeds paired independently; * = every seed's 95% bootstrap CI excludes zero

The simulator rests on two assets, the 4D avatars $\mathcal{A}$ the policy flies around and the per-frame reference models $\mathcal{R}_t$ the captures are scored against. Each avatar is fitted with GauHuman for 30k iterations at a resolution of 512×612, and each reference model with MaskGaussian for 10k iterations at 2048×2448 on a white background with a mask loss. We keep the sequences that reach 150 frames and whose body fits in a box of 2.2 m. The renderer is built on gsplat (Ye et al., 2025) and batches all environments into one call. The images handed to the reconstruction are re-rendered from $\mathcal{R}_t$ at the recorded poses, at a resolution of 1024×1024 on a white background with masks. The person mask is thresholded from the rendered alpha and back-projected with the depth map to give $\mathcal{P}_t$, and a frame with fewer than 32 mask pixels counts as a missed detection.

The camera command u_t of Sec. 3.1 is clamped into the unit disc and moves the camera by

$$\theta_{t+1} = \theta_t + \frac{u_{t,1}\, s_{\max}}{d \cos \phi_t}, \qquad \phi_{t+1} = \phi_t + \frac{\kappa\, u_{t,2}\, s_{\max}}{d},$$

with $\kappa = 0.6$, after which the elevation is clipped to $[-10^\circ, 50^\circ]$. The reference point moves horizontally, $c_{t+1} = c_t + \delta_c w_t$ with $\delta_c = 0.3$ m and each component of w_t in $[-1, 1]$. The budget applies to the total displacement $p_{t+1} - p_t$, and whenever it exceeds $s_{\max}$ both u_t and w_t are scaled down until it fits. The voxel map covers a cube of 2.2 m, and its three channels are accumulated from the surface points $\mathcal{P}_t$ and the camera's own pose alone, all in $[0, 1]$ and zero where nothing has been seen. The hit channel counts the steps at which a point of $\mathcal{P}_t$ fell into the cell, entered as $\log(1 + n)/\log(1 + 32)$ and clipped at one. Each hit also records the camera's direction bin at that step, taken from its own pose. The direction channel is the fraction of the 8×4 bins a cell has been hit from. The recency channel is $1 - (t - t_{\text{seen}})/T$, where t_{seen} is the last step at which the cell received a point. Each slot of the pose history $\mathcal{C}_t$ holds one camera pose, and each slot of the detection history $\mathcal{H}_t$ holds the offset of $\hat{c}_t$ from the reference point, a detection flag, the distance from

Table A2: **Framing.** For every capture distance and condition, the percentage of frames in which the person is not in the image at all, and the share of the body facing the camera that lands inside the image. Panel (c) of Fig. 5 plots the in-frame rows for the walking condition.

		No person (% of frames)			Body in frame (%)		
Distance	Condition	OURS	Orbit	Spiral	OURS	Orbit	Spiral
d =3.00 m	In place	0.00	0.00	0.00	99.79	100.00	100.00
d =3.00 m	Walking	0.00	26.86	28.02	99.69	57.38	52.29
d =2.50 m	In place	0.00	0.00	0.00	99.44	99.74	99.52
d =2.50 m	Walking	0.00	33.94	35.25	99.32	49.64	43.23
d =2.25 m	In place	0.00	0.00	0.00	98.11	97.91	97.68
d =2.25 m	Walking	0.00	37.12	37.55	97.80	46.13	38.44
d =2.00 m	In place	0.00	0.00	0.00	93.70	84.85	88.63
d =2.00 m	Walking	0.04	41.01	41.01	93.18	38.00	32.53

the person to the image border and the reference point's own last step. The voxel encoder has three 3D convolutions and a 1×1 convolution and gives 64 tokens of width 64, and the two Transformer layers use one attention head with a feed-forward width of 256. The reward counts a Gaussian as seen when it lies no more than 5 cm behind the surface the camera sees. Training one policy takes about 5.4 hours on one NVIDIA RTX PRO 6000 GPU, and at test time one step, from the depth map and mask to the next camera command, takes about 8 ms on an NVIDIA RTX 6000 Ada GPU, not counting the segmentation and depth sensing of a real camera.

A.2 PHYSICAL LIMITS

We bound the displacement per step by the tilt a multirotor can sustain. A multirotor moves sideways by tilting, and a tilt of β gives a sideways acceleration of $g \tan \beta$ while the rest of the thrust holds the aircraft up. The largest tilt it can hold is set by its thrust-to-weight ratio $1/\cos\beta$, which we fix at 2.09, a value within the range of consumer multirotors. Circling the person at speed v on a horizontal radius of $d\cos\bar{\phi}$ needs an acceleration of $v^2/(d\cos\bar{\phi})$, so $\tan\beta = v^2/(g\,d\cos\bar{\phi})$, with $\bar{\phi} = 20^\circ$ the middle of the elevation band. With the ratio fixed the bound grows with $\sqrt{d}$, giving 0.950, 0.867, 0.823 and 0.776 m at $d = 3.0$, 2.5, 2.25 and 2.0 m.

B EVALUATION PROTOCOL

B.1 PRESETS AND DISTANCE MATCHING

Each preset follows a path fixed before the episode begins that does not depend on where the person is, so in the walking condition the centre of its sphere stays at the person's starting position. The random walk is the one preset whose distance is not matched, since it is whatever its random steps add up to. The policy's first viewpoint is drawn at random for each sequence, and the presets are aligned to it so that none is favoured by where it starts: the fixed camera, the orbit and the spiral start at the policy's azimuth, and the random walk at the policy's viewpoint itself. For the orbit and the spiral we solve for the number of turns that gives the same total distance in the world frame as the policy at each capture distance and condition. One preset path serves all three training seeds, whose policies travel slightly different distances, so the match is within 3% except for one seed at $d = 2.0$ m, where the policy travels 5% more than the presets.

B.2 METRICS

All three metrics are computed on the same crop of the reference and the reconstruction, the bounding box of the reference mask with an eight-pixel margin. Table A2 reports how often the person is not in the image at all and how much of the body facing the camera falls inside the image. Values are averaged over the frames of a sequence, then over the 24 held-out sequences, and over the three training seeds for our policy, while the presets do not depend on the seed. The share of the body in frame counts the body vertices that face the camera, so self-occluded parts are left out.

Table A3: **Presets that follow the person.** Each preset moves the centre of its path toward our own estimate of the person's position, with the motion budget and total distance matched to our policy. Template-based method, walking condition, three training seeds.

Capture	PSNR ↑	LPIPS ↓
Orbit	16.25	0.251
Orbit + tracker	17.99	0.196
Spiral	17.62	0.248
Spiral + tracker	19.89	0.187
OURS	**20.16**	**0.184**

Table A4: **Comparison with a greedy planner.** Template-based method; the planners maximise our reward one or three steps ahead with privileged information. Reward is the final Φ of the episode.

	In place			Walking		
Capture	PSNR ↑	LPIPS ↓	Reward Φ	PSNR ↑	LPIPS ↓	Reward Φ
One-step planner	19.24*	0.204*	0.549	19.37*	0.202*	0.556
Three-step planner	19.29*	0.201*	0.579	19.47*	0.196*	0.608
OURS	**20.18**	**0.184**	0.511	**20.16**	**0.184**	0.506

Table A5: **Observation noise at test time.** Noise is added to the depth map, person mask, detections and pose estimate in four levels. All numbers use one training seed.

	Noise					In place			Walking		
Level	Mask (px)	Depth σ (m)	Holes	Missed	Pose σ ($^\circ$/m)	PSNR ↑	Ours − spiral	Ours − orbit	PSNR ↑	Ours − spiral	Ours − orbit
Clean	0	0	0	0	0/0	20.252	+0.338	+2.373	20.204	+2.588	+3.958
Light	3	0.01	0.0	0.05	0.25/0.01	20.247	+0.332	+2.367	20.204	+2.588	+3.958
Medium	5	0.03	0.1	0.10	0.50/0.02	20.161	+0.212	+2.261	20.133	+2.439	+3.742
Heavy	7	0.05	0.1	0.20	1.00/0.04	19.842	−0.106	+1.943	19.905	+2.211	+3.514

Its denominator is taken on a canvas larger than the image, so it does not change with the capture distance.

B.3 ADDITIONAL BASELINES

Greedy planners. At each step the planner scores every viewpoint it can reach within the motion budget and takes the one with the largest gain of our reward. Its action has the two parts of ours, each discretised: the camera moves in one of sixteen directions at full or half speed, and the reference point either stays where it is or steps toward the person. The planner reads the simulator's record of which surface elements have been seen and from where, and the true position of the person, neither of which enters our policy's observation. A second variant searches three steps ahead with a beam of eight and executes the first step of the best sequence. Both run in the simulator our policy is trained in, with the same kinematics, budget, visibility test and direction bins.

Presets that follow the person. Each preset runs the same perception chain as our policy from its own viewpoint to estimate where the person is, and the centre of its path moves toward that estimate at every step, keeping the last estimate when the person is not detected. Each shares our per-step budget, with its number of turns solved per seed so that it travels the same total distance as that seed's policy (Table A3). As these presets are our own construction rather than an existing method, Table A3 reports means only, without significance marks. In the in-place condition there is nothing to follow, and these presets score within 0.1 dB of the plain ones.

B.4 OBSERVATION NOISE

We add noise to the trained policy's observations at test time, without retraining: to the depth map, the person mask and the camera's estimate of its own pose, which the policy uses to build its pose history and voxel map, and we drop a random fraction of the detections to simulate missed detec-

Table A6: **Comparison with preset trajectories, SSIM.** The layout of Table 1 with SSIM (↑) in place of PSNR and LPIPS; shading as in Table 1.

Reconstruction	3DGS-Avatar		4DGS	
Capture	In place	Walking	In place	Walking
Fixed camera	0.596⋆	0.599⋆	0.438⋆	0.439⋆
Random walk	0.719⋆	0.661⋆	0.645⋆	0.568⋆
Orbit	0.768⋆	0.716⋆	0.708	0.628⋆
Spiral	0.778	0.715⋆	0.712	0.624⋆
OURS	**0.779**	**0.779**	**0.715**	**0.714**

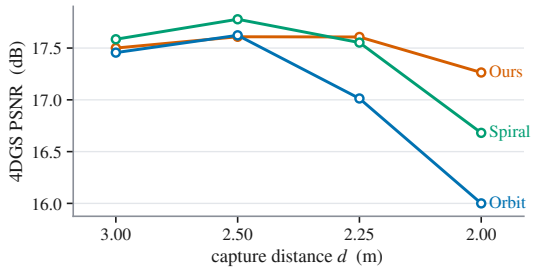


Figure A1: **Effect of capture distance under the template-free method.** 4DGS reconstruction quality from the views each strategy captured, in the in-place condition; the counterpart of Fig. 5(a).

tions. The noise changes only what the policy sees; the captured images are rendered at the true camera poses as before. The presets follow fixed paths and use no observation, so they are unaffected. The four levels were fixed before the experiment, and each level increases the noise in every channel at once (Table A5). Under noise the policy travels further, so at the medium and heavy levels the spiral is re-matched to the noisy policy's total distance. At the heaviest level in the in-place condition the policy and the spiral are no longer distinguishable.

C COMPLETE RESULTS

C.1 MAIN RESULTS

Table A6 repeats the main comparison in SSIM.

Fitted in the world frame, the template-free method collapses to 8–9 dB for every capture strategy when the person walks. Since an image depends only on where the camera is relative to the person, we describe every camera pose in a coordinate frame attached to the person's position, by moving the pose back by the person's displacement at that step. The camera poses of the captured images are shifted in this way before 4DGS is fitted, and the test views are shifted in the same way before the reconstruction is rendered at them. In this frame the person's position does not move and only the camera does, and 4DGS is fitted with every setting unchanged. The displacement is the same one the template-based method receives as the global translation of its body-model parameters, and the captures themselves are unchanged. In this frame the template-free method ranks the strategies as the template-based one does (Tables 1 and A6): our policy keeps its in-place score while the orbit and the spiral fall well below theirs. Refitted the same way, the eight-camera array scores about 1 dB below our policy while walking; larger arrays were not tested in this condition.

C.2 ANALYSIS

Table A7 shows how the template-based comparison changes as the body template is degraded. We add noise to the dataset's fitted SMPL parameters (Loper et al., 2015; Pavlakos et al., 2019) and also replace them by estimates from one fixed camera with the off-the-shelf estimator GVHMR (Shen

Table A7: **Sensitivity to the body template.** Template-based method with the dataset template degraded by noise or replaced by an estimate from one fixed camera; joint error is measured against the dataset ground truth. All numbers use one training seed.

Template	Joint error	PSNR (dB)			ΔPSNR, ours −	
	(mm)	Ours	Orbit	Spiral	Orbit	Spiral
In place						
dataset template	0	20.25	17.83	19.90	+2.42	+0.35
+ 20 mm noise	20	17.33	16.21	17.14	+1.11	+0.19
+ 40 mm noise	40	15.92	15.23	15.73	+0.69	+0.19
+ 80 mm noise	80	14.39	14.08	14.23	+0.31	+0.16
single-camera estimate	122	18.06	16.35	17.81	+1.71	+0.25
Walking						
dataset template	0	20.20	16.25	17.65	+3.95	+2.56
+ 20 mm noise	20	17.29	14.78	15.33	+2.51	+1.95
+ 40 mm noise	40	15.88	13.76	13.93	+2.12	+1.96
+ 80 mm noise	80	14.39	12.50	12.51	+1.89	+1.88
single-camera estimate	122	18.00	14.49	15.20	+3.51	+2.80

Table A8: **Motion-resolved gain over *spiral*.** Template-based method; frames of each walking episode are ranked by body motion (App. C.2) and split into the least- and most-moving quarters.

Capture	ΔPSNR vs. Spiral (dB)		
distance	least-moving 25%	most-moving 25%	**most − least**
3.00 m	+0.46	+1.26*	**+0.80**
2.50 m	+1.22*	+1.75*	**+0.54**
2.25 m	+1.66*	+2.03*	**+0.37**
2.00 m	+2.15*	+2.24*	**+0.09**

Table A9: **Zero-shot transfer to MVHumanNet.** The policy trained on DNA-Rendering, run without any change on 24 held-out MVHumanNet sequences, in both conditions at 2.0 m and with the same test-view layout and distance matching as Table 1.

Reconstruction	3DGS-Avatar				4DGS			
	In place		Walking		In place		Walking	
Capture	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
Orbit	18.18*	0.188*	16.36*	0.226*	15.55*	0.292*	13.84*	0.330*
Spiral	20.02*	0.177*	17.45*	0.227*	16.30*	0.256	14.35*	0.314*
OURS	**20.30**	**0.172**	**20.30**	**0.172**	**16.74**	**0.252**	**16.77**	**0.252**

et al., 2024), and use the degraded parameters for both fitting and rendering. The four test-view radii used in Sec. 4.3 are 3.0, 2.5, 2.25 and 2.0 m, all with the template-based method in the in-place condition.

For Table A8 the frames of each walking episode are split into quarters by body motion. Body motion is the mean displacement of the surface points between consecutive frames, not counting the walking translation, so the least-moving quarter still contains walking frames. The difference shrinks as the framing tightens, since at 2.0 m the presets lose quality in every frame.

C.3 TRANSFER TO ANOTHER STUDIO DATASET

MVHumanNet (Xiong et al., 2024) is a second studio dataset, recorded with 48 cameras in three rings. We run the policy trained on DNA-Rendering on 24 of its sequences without any change. The sequences are the first 24 of Part 01 whose assets could be built. Assets and references are built as for DNA-Rendering (Sec. 3.2), with the SMPL-X parameters provided with the dataset. The

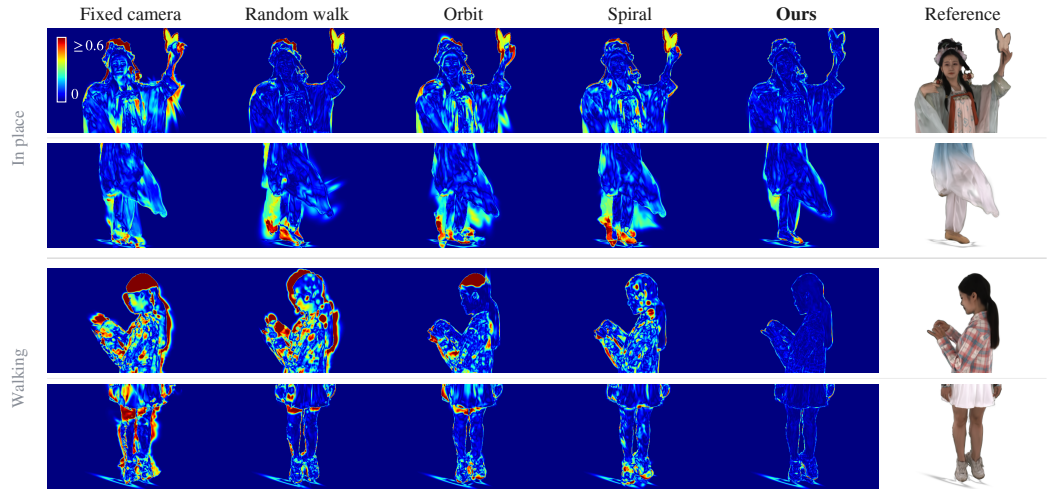


Figure A2: **Per-pixel error of the reconstructions in Fig. 3.** Mean absolute colour difference to the reference at every pixel of the same crops; the last column is that reference.

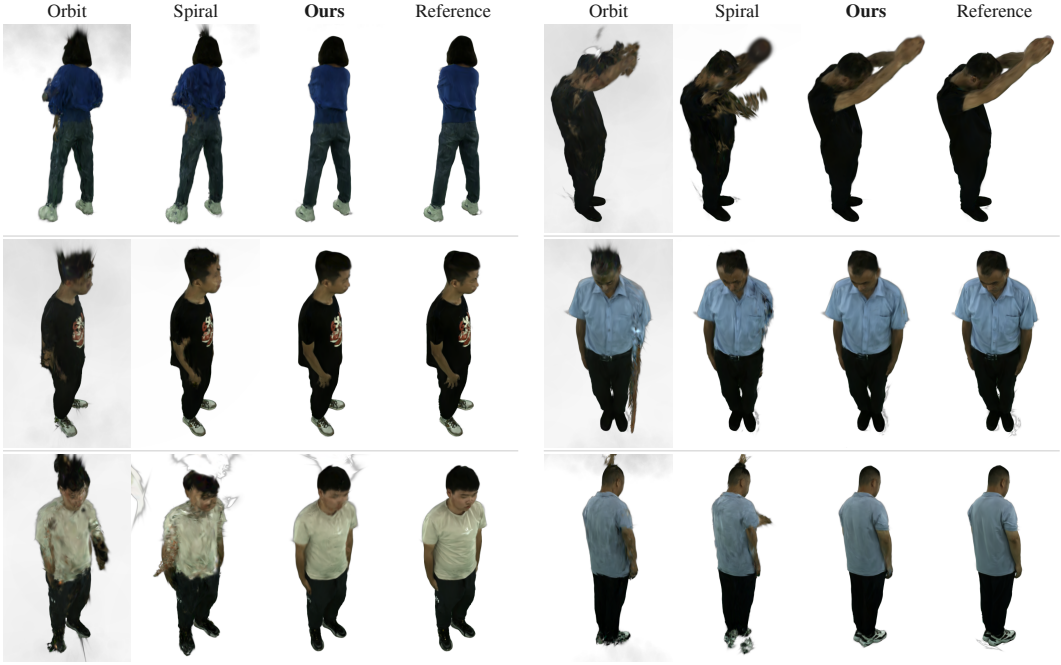


Figure A3: **Zero-shot transfer to MVHumanNet, 4DGS reconstructions.** Six of the MVHumanNet sequences of Table A9, captured in place at 2.0 m by the orbit, the spiral and our policy, next to the reference. Same frame and view per row, first training seed.

recordings are sampled at 5 fps, so the body moves about 1.7 times as far per step as in training. The walking condition carries each sequence along the evaluation paths of DNA-Rendering, by the same displacement per step. Table A9 reports both conditions at 2.0 m with both reconstruction methods, in the format of Table 1, and Fig. A3 shows 4DGS reconstructions for six of these sequences. Our policy leads the orbit and the spiral in both conditions under both reconstruction methods.



Figure A4: **Reconstruction quality under each capture strategy, further sequences.** The layout of Fig. 3 for eight further examples from seven held-out sequences.

C.4 Additional qualitative results

A video in the supplementary material shows the task, the camera paths of each capture strategy and the resulting reconstructions in both conditions. All qualitative figures use the first training seed at 2.0 m; Figs. A2 and A4 show the template-based method as Fig. 3 does, and Fig. A3 the template-free method. Fig. A2 gives the per-pixel error for Fig. 3, and Fig. A4 shows eight further examples.

Table A10: **Design ablations.** Each alternative against our final policy, both conditions, $d = 2.0\,\mathrm{m}$, template-based method; $\Delta > 0$ means our choice is better (Sec. 4.1). Seeds are ours $\times$ alternative; cells are not starred because seed counts differ.

		ΔPSNR (dB)		ΔLPIPS		Seeds
	Alternative tested	in place	walking	in place	walking	ours×alt.
(a)	Reward: coverage instead of directional diversity	$+0.578$	$+0.780$	$+0.0169$	$+0.0200$	3×1
(b)	Reward: + a trajectory smoothness penalty	$+0.014$	$+0.009$			1×1
(c)	Reward: + a temporal smoothness penalty	-0.016	-0.044			1×1
(d)	Observation: privileged map instead of onboard	-0.166	-0.126			2×2
(e)	Action: Cartesian velocity instead of spherical	diverged in every run				3×3
(f)	Orientation: free yaw/pitch $\pm 30^\circ$ instead of look-at	$+0.678$	$+0.617$	$+0.0205$	$+0.0191$	3×1

D Design ablations and the reward

D.1 Design ablations

(a) Reward. We replace our directional-diversity reward by the binary coverage reward used by learned exploration policies for static scenes (Chen et al., 2024; 2025). It pays a surface Gaussian once, the first time it is seen from any direction, whereas ours pays it each time it is seen from a new direction bin. A policy trained with it therefore has no reason to return to a surface it has already seen and moves on as soon as a surface has been seen once; it no longer beats the spiral in the in-place condition, whereas ours does (Table 1).

(b, c) Smoothness penalties. We add a trajectory smoothness penalty to our reward: at every step it lowers the reward by an amount that grows with the angle between the camera's last two moves. The policy learns to avoid it and turns by two thirds less, while reconstruction quality does not move (Table A10). We also add, separately, a temporal smoothness penalty, which lowers the reward by how much the reward changed from the previous step. The policy trained with it hardly reduces the penalty, and quality does not move either.

(d) Observation. We replace the onboard voxel map by a privileged one: the simulator's own record of which surface Gaussians have been seen and from where, registered to the ground-truth body and free of detection and depth error. The onboard policy trails it by at most 0.17 dB (Table A10).

(e) Action coordinates. We replace the spherical action by a Cartesian one: the policy outputs a velocity vector (v_x, v_y, v_z), and the new position is projected back onto the sphere, so the camera can reach the same positions as before and only the coordinates of the action change. Trained this way, the policy reaches the same peak reward as ours and then collapses in every run, after which the camera hardly moves.

(f) Orientation. We add two action dimensions that let the policy turn the camera by up to $\pm 30^\circ$ in yaw and pitch away from the reference point. This variant loses on every sequence in both conditions on all three metrics. The range is wide enough to point the person out of the frame, but the policy keeps the person in view and spends the freedom on aiming; its reference point drifts away from the person even in place and hardly follows the person while walking.

D.2 Reward and reconstruction quality

Fig. A5 plots the reward against reconstruction quality for every capture strategy. The reward is the potential Φ at the end of the episode, obtained by replaying every strategy in the simulator so that the presets and the planners are measured the same way as our policy. From the fixed camera to our policy, a higher reward goes with better reconstruction. Above our policy's reward the relation breaks: the two planners reach an even higher reward and reconstruct worse. The reward is what the policy trains on and not what we judge a capture by, so every comparison in this paper is decided by reconstruction quality.

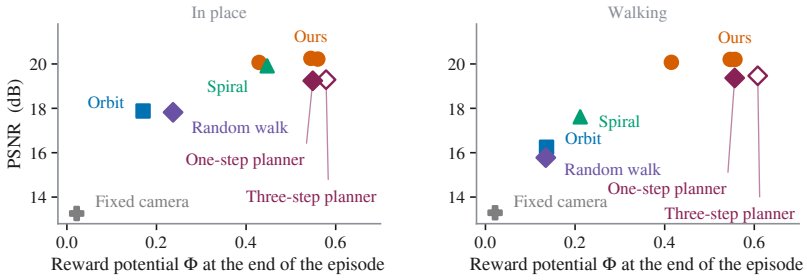


Figure A5: **Reward and reconstruction quality.** One point per capture strategy, averaged over the 24 held-out sequences; our policy is shown for each of three training seeds.

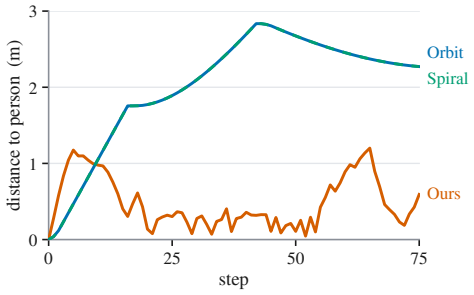


Figure A6: **Following the walking person.** Horizontal distance from the centre of each strategy's path to the person, for one sequence in the walking condition at $d = 2.0$ m, first training seed.

D.3 Following the walking person

Fig. A6 shows, for one walking sequence, the distance from the centre of each strategy's path to the person. The centre of a preset stays at the starting position, whereas ours moves with the person.