Why is Sparse-View 4D Reconstruction Hard?

Zihan Wang Jeff Tan Tarasha Khurana* Neehar Peri* Deva Ramanan

Carnegie Mellon University

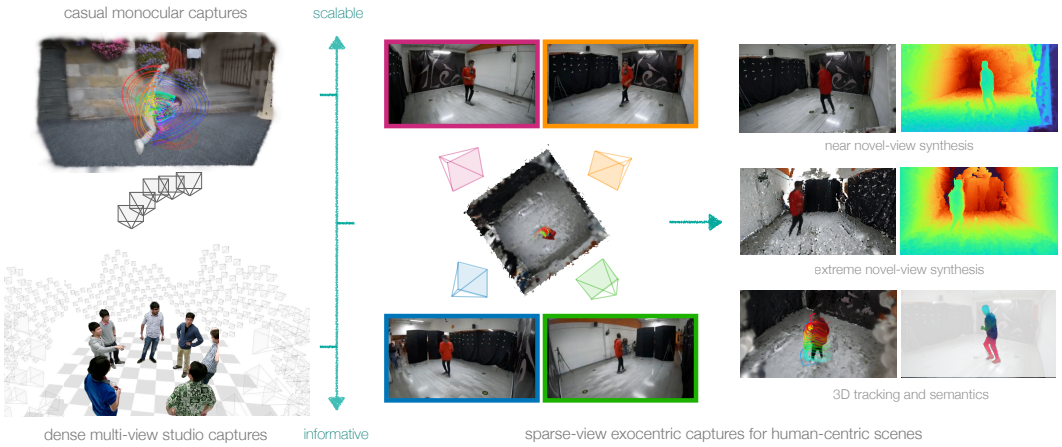

Figure 1. **Problem setup.** Given sparse-view videos of dynamic scenes, our approach reconstructs 3D geometry and motion, enabling extreme novel view synthesis, 3D tracking, and feature distillation. Our sparse-view (4-camera) setup strikes a balance between ill-posed reconstructions from casual monocular captures [18, 42] and well-constrained reconstructions from dense multi-view studio captures [24].

Abstract

We address the problem of dynamic scene reconstruction from sparse-view videos. Prior work often requires dense multi-view captures using dozens of calibrated cameras (e.g. Panoptic Studio), or short monocular videos with limited information (e.g. DAVIS). In contrast, we aim to reconstruct diverse dynamic human behaviors, such as repairing a bike or dancing from sparse-view videos. We repurpose state-of-the-art monocular reconstruction methods for sparse-view reconstruction and find that careful initialization from time- and view-consistent monocular depth estimators produces more accurate reconstructions. Specifically, our method predicts dense surface points across all training views, and uses confidence-aware pixel alignment to initialize scene geometry. We further distill per-point semantic features from 2D foundation models, and use feature clustering to encode a compact set of motion bases. Finally, we employ a gradient-based joint optimization framework to simultaneously learn scene geometry and motion. Notably, our approach achieves state-of-the-art performance on challenging sequences from the Ego-Exo4D dataset.

*Equal senior authorship

1. Introduction

Accurately reconstructing dynamic 3D scenes from one or more videos [34, 52] is of great interest to the vision community, with applications in AR/VR [48] and robotic manipulation [27]. Prior work often studies this problem in the context of casually-captured monocular videos [18, 42], which lack sufficient information to fully reconstruct the scene; or dense multi-view captures with dozens of calibrated cameras [24, 37], which require dedicated capture studios that are challenging to scale. In this paper, we aim to strike a balance between the ease and informativeness of capture, by reconstructing skilled human behaviors such as repairing a bike and dancing from sparse-view cameras.

Status quo. Despite recent advances in dynamic scene reconstruction [4, 16–18], current approaches often require dozens of calibrated cameras [24, 37], are category specific [59], or struggle to generate multi-view consistent geometry [34]. More recently, monocular reconstruction approaches [31, 53, 54, 64] have demonstrated high visual fidelity, but can only reconstruct video clips from a single viewpoint, and struggle with extreme novel view synthesis.

Problem setup. We study the problem of reconstructing dynamic human behaviors using the Ego-Exo4D dataset.

Ego-Exo4D provides egocentric video paired with multiple time-synchronized exocentric video streams across a range of skilled human activities. Unlike prior dynamic reconstruction datasets, Ego-Exo4D includes sparse-view videos of diverse environments from four cameras, 90° apart, posing unique challenges for 4D reconstruction.

Why is sparse-view 4D reconstruction hard? We argue that sparse-view reconstruction presents unique challenges not found in dense multi-view or monocular reconstruction. For dense multi-view captures, it is often sufficient to rely solely on geometric and photometric cues for reconstruction, often making use of classic techniques from (non-rigid) structure from motion [13]. As a result, these methods fail in sparse-view settings that are not sufficiently informative. In contrast, monocular reconstruction approaches often rely on priors to *hallucinate* geometry and regularize optimization. Indeed, such methods sometimes evaluate with generative metrics such as FID score [6, 21, 36], which capture correctness only in a distributional sense. Somewhat paradoxically, we find that adding additional information (via additional viewpoints) makes the problem *harder*; naively repurposing monocular approaches for sparse-view reconstruction often yields inconsistent image-level priors (e.g. monocular depth estimates) across views, resulting in local minima during 3D optimization. Finally, sparse-view reconstruction is typically evaluated with pixelwise metrics that compare novel-view renderings to the ground-truth.

Effectively incorporating priors. We observe that initializing with monocular geometry estimators that are both time- and view-consistent yields higher quality reconstructions. Our approach predicts dense surface points across all training views and uses confidence-based pixel alignment to initialize the scene's geometry. Additionally, we distill semantic features for each point from 2D foundation models and apply feature clustering to create a compact representation of motion bases. Finally, we use gradient-based joint optimization to simultaneously learn both the scene geometry and motion. Our experiments demonstrate that our approach achieves better novel time synthesis and novel extreme view synthesis compared to our baselines.

Contributions. We present three major contributions.

- We repurpose Ego-Exo4D for sparse-view reconstruction and highlight the challenge of reconstructing skilled human behaviors in dynamic environments.
- We demonstrate that monocular reconstruction methods can be extended to the sparse-view setting by carefully incorporating monocular depth and foundational priors.
- We extensively ablate our design choices and show that we achieve state-of-the-art performance on challenging sequences from Ego-Exo4D.

2. Related Work

Dynamics scene reconstruction. Dynamic scene reconstruction [4] has received significant interest in recent years. While classical work [11, 39] often relies on RGB-D sensors or strong domain knowledge [2, 8], recent approaches [33, 34] based on neural radiance fields [38] have progressed towards reconstructing dynamic scenes in-the-wild from RGB video alone. However, such methods are computationally heavy, can only reconstruct short video clips with limited dynamic movement, and struggle with extreme novel view synthesis. Recently, 3D Gaussian Splatting [26, 37] has accelerated radiance field training and rendering via an efficient rasterization process. Follow-up works [35, 56, 62] repurpose 3DGS to reconstruct dynamic scenes, often by optimizing a fixed set of Gaussians in canonical space and modeling their motion with deformation fields. However, as Gao et al. [18] points out, such methods often struggle to reconstruct realistic videos. Many works address this shortcoming by relying on 2D point tracking priors [53], fusing Gaussians from many timesteps [29], modeling isotropic Gaussians [49], or exploiting domain knowledge such as human body priors [30, 51]. However, these approaches study the reconstruction problem in the monocular setting. As 4D reconstruction from a single viewpoint is extremely under-constrained, the recovered geometry is often incorrect when observed from extreme viewpoints. We argue that a sparse multi-view capture setup strikes a reasonable balance between ease of capture and informativeness, and design a careful initialization procedure to ensure consistency over time and across cameras.

Novel-view synthesis from sparse views. Both NeRF and 3D Gaussian splatting require dense input view coverage, which hinders their real-world applicability. Recent works aim to reduce the number of required input views by adding additional supervision and regularization, such as depth [9, 40] or semantics [23, 44, 63]. FSGS [67] builds on Gaussian splatting by producing faithful static geometry from as few as 3 views by unpooling existing Gaussians and adopting extra depth supervision. GaussianObject [58], on the other hand, adds noise to Gaussian attributes and relies on a pre-trained ControlNet [65] to repair low-quality rendered images. Other works such as MVSplat [5] build a cost volume representation and predict Gaussian attributes in a feed-forward manner. However, they only show success in near-novel view synthesis with a small deviation from the nearest training view. For methods that rely on learned priors, high-quality novel view synthesis is often limited to images within the training distribution. Such methods cannot handle diverse real-world geometry. Diffusion-based reconstruction methods [19, 57] try to generate additional views consistent with the sparse input views, but often produce artifacts. In our case, four sparse view cameras are separated

around 90° apart, posing unique challenges.

Feed-forward geometry estimation. Learning-based methods, such as monocular depth networks, are able to reconstruct 3D objects and scenes by learning strong priors from training data. While early works [12, 14] focus on in-domain depth estimation, recent works build foundational depth models by scaling up the training data [45, 46, 60, 61], resolving the metric ambiguity from various camera models [22, 43, 55], or relying on priors such as Stable Diffusion [15, 25, 47]. Unfortunately, monocular depth networks are not scale or view consistent, and often require extensive alignment against ground-truth to produce meaningful metric outputs. To address these shortcomings, DUSt3R [54] and MonST3R [64] propose the task of point map estimation, which aims to recover scene geometry as well as camera intrinsics and extrinsics given a pair of input images. These methods unify single-view and multi-view geometry estimation, and enable consistent depth estimation across either time or space. While DUSt3R is multi-view consistent, it is not consistent over time and also has poor generalization outside its training data. MonST3R, on the other hand, predicts temporally consistent pointmaps of dynamic objects but is limited to a single video. In our work, we explore how to produce both spatially and temporally consistent depth predictions.

3. Towards Sparse-View 4D Reconstruction

Given sparse-view (i.e. 3-4) videos from stationary cameras as input, our method recovers the geometry and motion of a dynamic 3D scene. We model the scene as a set of canonical 3D Gaussians (Sec. 3.1), which translate and rotate via a linear combination of motion bases. As dynamic scene reconstruction from sparse views is extremely challenging, we present two key insights to initialize plausible geometry and motion: a) initializing consistent scene geometry via confidence-aware spatio-temporal alignment (Sec. 3.2), and b) initializing motion trajectories by clustering per-point 3D semantic features distilled from 2D foundation models (Sec. 3.3). We formulate a joint optimization which simultaneously recovers geometry and motion (Sec. 3.4). Fig. 2 provides a summary of our method.

3.1. 3D Gaussian Scene Representation

We represent the geometry and appearance of dynamic 3D scenes using 3D Gaussian Splatting [26], due to its efficient optimization and rendering. Each Gaussian in the canonical frame t_0 is parameterized by $(\mathbf{x}_0, \mathbf{R}_0, \mathbf{s}, \alpha, \mathbf{c})$, where $\mathbf{x}_0 \in \mathbb{R}^3$ is the position of the Gaussian in canonical frame, $\mathbf{R}_0 \in \mathbb{SO}(3)$ is the orientation, $\mathbf{s} \in \mathbb{R}^3$ is the scale, $\alpha \in \mathbb{R}$ is the opacity, and $\mathbf{c} \in \mathbb{R}^3$ is the color. The position and orientation are time-dependent, while the scale, opacity, and color are persistent quantities shared over time. Practically, the rotation matrix $\mathbf{R}$ is stored as a quaternion $\mathbf{q} \in \mathbb{R}^4$. We additionally assign a semantic feature $\mathbf{f} \in \mathbb{R}^N$ to each Gaussian (Sec. 3.3), where N is an arbitrary number representing the embedding dimension of the feature. Empirically, we find that fixing the color and opacity of Gaussians results in a better performance. In summary, for the i-th 3D Gaussian, the optimizable attributes are given by $\Theta^{(i)} = \{\mathbf{x}_0^{(i)}, \mathbf{q}_0^{(i)}, \mathbf{s}^{(i)}, \mathbf{f}^{(i)}\}$.

Rendering. To render 3D Gaussians from a camera with intrinsics $\mathbf{K}$ and world-to-camera extrinsics $\mathbf{W}$, we write the 3D Gaussian's covariance matrix $\Sigma = \mathbf{RSS}^\top\mathbf{R}^\top$, and project both the mean and covariance to 2D as follows:

$$\mu_0^{'}(\mathbf{K}, \mathbf{W}) = \Pi(\mathbf{KW}\mu_0) \in \mathbb{R}^2 \tag{1}$$

$$\Sigma_0^{'}(\mathbf{K}, \mathbf{W}) = \mathbf{J}\Sigma_0\mathbf{J}^\top \in \mathbb{R}^{2\times 2} \tag{2}$$

where Π is perspective projection and $\mathbf{J}$ is the Jacobian of the affine approximation of the projective transformation defined by $\mathbf{K}$ and $\mathbf{W}$. For each image pixel, we then compute the color C and feature value F by volumetric rendering, which is performed in front-to-back depth order:

$$C = \sum_{i\in\mathcal{N}} \mathbf{c}_i\alpha_i T_i, \quad F_s = \sum_{i\in\mathcal{N}} \mathbf{f}_i\alpha_i T_i, \tag{3}$$

where $\mathcal{N}$ is the set of sorted Gaussians overlapping with the given pixel and $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$ is the transmittance, defined as the product of opacity values across previous overlapping Gaussians. Following [66], both RGB image and feature map use the same tile-based rasterization procedure.

3D Gaussian initialization. Prior work [26, 37] in 3D Gaussian Splatting has shown the importance of good initialization for the position and appearance of 3D Gaussians. Due to the underconstrained nature of the reconstruction problem, a poor initialization can result in bad local minima where portions of input images are explained by floater Gaussians in front of specific training cameras. While prior works rely on sparse 3D points produced by structure-from-motion (SfM) pipelines [26], or measurements from depth cameras [37], neither SfM nor ground-truth depth cameras are available in our constrained sparse-view setup. Similar to recent methods [50, 53], we instead rely on data-driven monocular depth priors to initialize the position and appearance of 3D Gaussians over time. The following sections describe our insights to make this work in challenging real-world sparse-view scenarios.

3.2. Consistent Depth Initialization

Given the success of initializing 3DGS with monocular depth estimates in single-view settings [53], one might think to naturally extend this to multi-view settings by repeating monocular depth initialization for each view. However, this naive initialization yields conflicting geometry signals, even

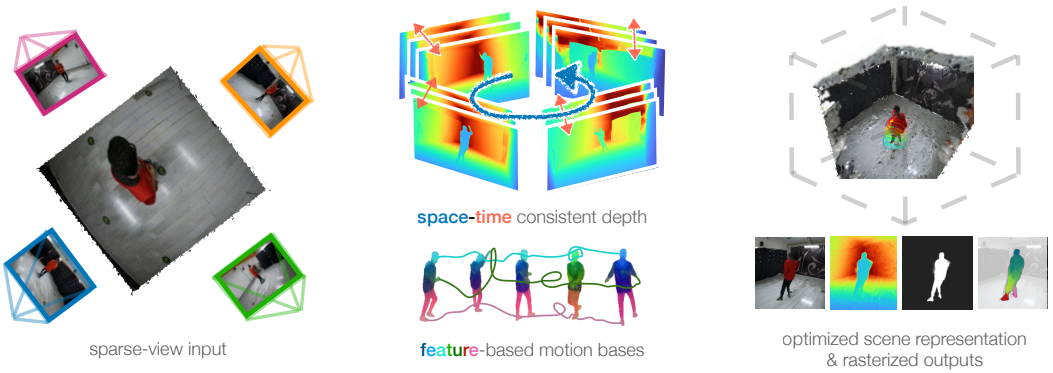

Figure 2. **Approach.** Given a set of sparse-view video sequences of a scene (left), we aim to optimize a 3D gaussian representation over time. First, we find that cross-view and time-consistent depth initialization are important for preventing the optimization from getting stuck in a local minima (middle, top). Second, we find that motion bases constructed from feature-clustering form a more geometrically consistent set of bases (middle, bottom), than those initialized by noisy 3D tracks. Our optimization yields a 4D scene representation from which we can rasterize RGB frames, depth maps, foreground silhoutte, and object features from novel views (right).

after scale and shift alignment. The unprojected monocular depths from separate views are inconsistent, and result in duplicated object parts. In this section, we show a promising step towards addressing this problem via consistent depth alignment across views and time.

Multi-view pointmap prediction. DUSt3R [54] aims to predict multi-view consistent pointmaps across multiple input images by first performing pairwise pointmap inference then optimizing a 3D global alignment objective. During the pairwise inference stage, DUSt3R is given a pair of RGB images $(\mathbf{I}_a, \mathbf{I}_b)$ as input, and predicts a pair of pixel-aligned pointmaps $\left(\mathbf{x}_a^{(a)}, \mathbf{x}_b^{(a)}\right)$ and confidences $\left(\mathbf{c}_a^{(a)}, \mathbf{c}_b^{(a)}\right)$, where (a) denotes that the predicted pointmaps for both images are in $\mathbf{I}_a$'s local coordinate frame. During DUSt3R's global alignment stage, a connectivity graph $G(\mathcal{V}, \mathcal{E})$ is first constructed from a set of T images $\mathcal{V} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_T\}$, where an edge $e = (a, b) \in \mathcal{E}$ denotes that images $\mathbf{I}_a$ and $\mathbf{I}_b$ observe an overlapping part of the scene. In order to place all pairwise predictions into a common global coordinate frame, DUSt3R solves for a pairwise pose $\mathbf{P}_e$ and scaling σ_e associated with each pair $e \in \mathcal{E}$. For clarity, given an edge $e = (a, b)$, define $\mathbf{x}_a^{(e)} := \mathbf{x}_a^{(a)}$ and $\mathbf{x}_b^{(e)} := \mathbf{x}_b^{(a)}$. Then, DUSt3R recovers a per-image pointmap $\{\chi_t \in \mathbb{R}^{H \times W \times 3}\}_{t=1}^T$ in world coordinates by solving the following optimization problem:

$$\underset{\chi, \mathbf{P}, \sigma}{\operatorname{argmin}} \sum_{e \in \mathcal{E}} \sum_{n \in e} \sum_{u,v}^{H,W} \mathbf{c}_n^{(e)}(u,v) \left\| \chi_n(u,v) - \sigma_e \mathbf{P}_e \mathbf{x}_n^{(e)}(u,v) \right\|^2 \tag{4}$$

Here, the pairwise pointmaps $\mathbf{x}_{n,t}^{(e)}$ and confidences $\mathbf{c}_{n,t}^{(e)}$ are fixed and predicted by DUSt3R pairwise inference. The optimization variables are per-image global pointmaps $\{\chi_t\}_{t=1}^T$, as well as per-pair similarity transforms $(\mathbf{P}_e, \sigma_e)$ which transform each pairwise inference from some local "pair" coordinate system to world coordinates.

Recovering multi-view consistent depth. By parameterizing the per-image global pointmaps in terms of camera intrinsics $\{\mathbf{K}_t\}_{t=1}^T$ and camera extrinsics $\{\mathbf{P}_t\}_{t=1}^T$, we can recover multi-view consistent depth maps $\{\mathbf{d}_t\}_{t=1}^T$ associated with the input images:

$$\chi_t(u,v) = \mathbf{P}_t^{-1} h\left(\mathbf{K}_t^{-1}[u,v,1]^\top \mathbf{d}_t(u,v)\right) \tag{5}$$

where h is a 3D point in homogeneous coordinates. As the camera parameters are assumed to be known during sparse-view capture, we can preset the cameras and keep them fixed throughout the global alignment optimization.

Recovering time-consistent depth. Though the approach described previously allows recovering multi-view consistent depth, it still does not address the problem of temporal consistency. Naively concatenating multi-view consistent depth maps across video frames yields flickering and disappearing geometry, e.g. people teleporting between the foreground and background.

Although MonST3R [64] addresses the problem of time-consistent pointmap prediction, it cannot simultaneously align more than 70 video frames on a consumer 24 GB GPU due to the large memory footprint of global alignment. MonST3R therefore cannot handle more than 15 frames from four cameras, making it impractical for estimating both time- and view-consistent depth.

Confidence-aware spatio-temporal alignment. To ensure both view and time consistency, we propose a two-stage confidence-aware procedure for spatio-temporal alignment. Let $t = \{1, \ldots, T\}$ over the number of video frames and $k = \{1, \ldots, K\}$ over cameras. First, we run

MonST3R separately on each input camera to obtain time-consistent (but not view-consistent) dynamic 3D pointmaps $\{\chi^{(\text{time})}_{t,k}\}$. Then, we run DUSt3R separately on each time (with fixed camera poses) to obtain view-consistent (but not time-consistent) metric 3D pointmaps: $\{\chi^{(\text{view})}_{t,k}\}$. We threshold only the confident pixels across both predictions, and search for per-camera scale factors that optimally align the time-consistent pointmaps with the view-consistent pointmaps:

$$s_k^* = \underset{s_k}{\text{argmin}} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{\text{confident } u,v} \left\| s_k \chi^{(\text{time})}_{t,k}[u,v] - \chi^{(\text{view})}_{t,k}[u,v] \right\|^2 \tag{6}$$

$$\chi^{(\text{time+view})}_{t,k} = s_k^* \chi^{(\text{time})}_{t,k} \tag{7}$$

The scaled time- and view-consistent pointmaps are used as 3DGS initialization. In practice, we find that even while running DUSt3R, it is beneficial to replace the DUSt3R checkpoint with the MonST3R checkpoint as there are very few human-centric scenes in DUSt3R's training data.

3.3. Grouping-based Motion Initialization

Beyond initializing time- and view-consistent geometry in the canonical frame, we also aim to initialize reasonable estimates of the scene motion. We model a dynamic 3D scene as a set of $\mathcal{N}$ canonical 3D Gaussians, along with time-varying rigid transformations $\mathbf{T}_{0\to t} = [\mathbf{R}_{0\to t}\mathbf{t}_{0\to t}] \in \mathbb{SE}(3)$ that warp from canonical space to time t:

$$\mathbf{x}_t = \mathbf{R}_{0\to t}\mathbf{x}_0 + \mathbf{t}_{0\to t} \quad \mathbf{R}_t = \mathbf{R}_{0\to t}\mathbf{R}_0 \tag{8}$$

Motion bases. Similar to Shape of Motion [53], we make the observation that in most dynamic scenes, the underlying 3D motion is often low-dimensional, and composed of simpler units of rigid motion. For example, the forearms tend to move together as one rigid unit, despite being composed of thousands of distinct 3D Gaussians. Rather than storing independent 3D motion trajectories for each 3D Gaussian (i), we define a set of B learnable basis trajectories $\{\mathbf{T}^{(i,b)}_{0\to t}\}_{b=1}^{B}$. The time-varying rigid transforms are written as a weighted combination of basis trajectories, using fixed per-point basis coefficients $\{w^{(i,b)}\}_{b=1}^{B}$:

$$\mathbf{T}^{(i)}_{0\to t} = \sum_{b=1}^{B} \mathbf{w}^{(i,b)} \mathbf{T}^{(i,b)}_{0\to t} \tag{9}$$

Motion bases via 3D tracking. To compute a sparse set of motion bases, Shape of Motion first obtains dense motion estimates by using monocular depth [43, 60] to lift off-the-shelf long-range 2D tracks [10] into 3D tracks. k-means clustering is used to group the vectorized velocities of 3D tracks, and a sparse set of motion bases is derived by taking a weighted average of the 3D tracks belonging to each group. In practice, we find that initializing motion bases from noisy 3D tracking results in low-quality and view-inconsistent motion. In particular, introducing a separate set of motion bases for each frame often results in duplicated structures that each move independently (e.g. multiple arms and legs). Thus, we aim to find a globally consistent method to sparsify the 3D scene motion.

Motion bases via feature clustering. Our key insight is that semantically grouping similar scene parts together can help regularize dynamic scene motion, without ever initializing trajectories from noisy 3D track predictions. Inspired by the success of robust and universal feature descriptors [41], we obtain pixel-level features for each input image by evaluating DINOv2 on an image pyramid. We average features across pyramid levels and reduce the dimension to 32 via PCA [1]. We choose the small DINOv2 model with registers, as it produces fewer peaky feature artifacts [7].

Given the consistent pixel-aligned pointmaps $\chi^{(\text{time+view})}_{t,k}$, we associate each pointmap with the 32-dim feature map $\mathbf{f}_{t,k}$ computed from the corresponding image. We perform k-means clustering on per-point features $\mathbf{f}$ to produce b initial clusters of 3D points. After initializing 3D Gaussians from pointmaps, we set the motion basis weight $\mathbf{w}^{(i,b)}$ to be the L2 distance between the cluster center and 3D Gaussian center. We initialize the basis trajectories $\mathbf{T}^{(b)}_{0\to t}$ to be identity, and optimize them via differentiable rendering.

3.4. Optimization

As observed in prior work [17, 32], using photometric supervision alone is insufficient to avoid bad local minima in a sparse-view setting. Our final optimization procedure is a combination of photometric losses, data-driven priors, and regularizations on the learned geometry and motions.

During each training step, we sample a random timestep t and camera k. Following Eq. 3, we render the image $\hat{\mathbf{I}}_{t,k}$, mask $\hat{\mathbf{M}}_{t,k}$, features $\hat{\mathbf{F}}_{t,k}$, and depth $\hat{\mathbf{D}}_{t,k}$. We compute reconstruction loss by comparing to off-the-shelf estimates:

$$\mathcal{L}_{\text{recon}} = \left\|\hat{\mathbf{I}} - \mathbf{I}\right\|_1 + \lambda_{\text{m}} \left\|\hat{\mathbf{M}} - \mathbf{M}\right\|_1 + \lambda_{\text{f}} \left\|\hat{\mathbf{F}} - \mathbf{F}\right\|_1 + \lambda_{\text{d}} \left\|\hat{\mathbf{D}} - \mathbf{D}\right\|_1 \tag{10}$$

sssss We additionally enforce a rigidity loss between randomly sampled dynamic Gaussians and their k nearest neighbors. Let $\hat{\mathbf{X}}_t$ denote the location of a 3D Gaussian at time t, and let $\hat{\mathbf{X}}_{t'}$ denote its location at time t'. Over neighboring 3D Gaussians i, we define:

$$\mathcal{L}_{\text{rigid}} = \sum_{\text{neighbors } i} \left\|\hat{\mathbf{X}}_t - \hat{\mathbf{X}}^{(i)}_t\right\|_2^2 - \left\|\hat{\mathbf{X}}_{t'} - \hat{\mathbf{X}}^{(i)}_{t'}\right\|_2^2 \tag{11}$$

4. Experimental Results

Implementation details. We optimize our representation with Adam [28]. We use 18k gaussians for the foreground

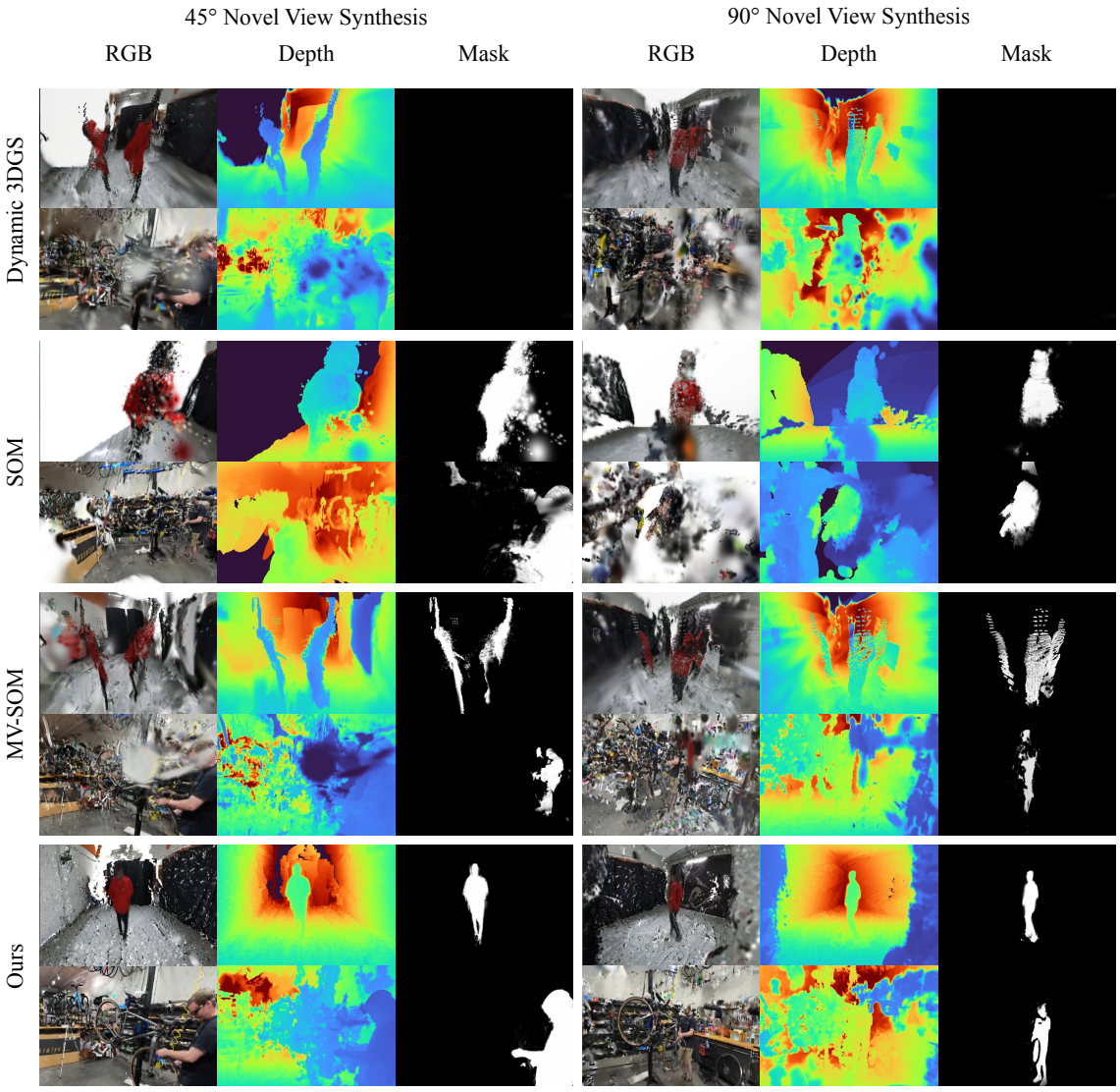

Figure 3. **Qualitative results.** In each 2×3 block, we visualize the rasterized RGB image, depth map, and foreground mask from each method for the dancing (top) and bike repair (bottom) sequences. We show 45° (left) and 90° (right) novel view synthesis results. Existing monocular methods and their extension to multi-view produce poor results rendered from a drastically different novel view. MV-SOM improves upon SOM in both 45° novel-view synthesis and 90° novel-view synthesis. Our method's careful point cloud initialization and feature-based motion bases further improve on MV-SOM. Note that Dynamic 3DGS does not separate foreground and background Gaussians, so we report a black foreground mask.

and 1.2M for the background. We fix the number of $\mathbb{SE}(3)$ motion bases to 40 and obtain these from feature clustering (Sec. 3.3). For the depth alignment, we use points above the confidence threshold of 95%. We train on 3-4 10-sec long videos at 30fps with a resolution of 512 × 288. Training takes about 3 hours on a single NVIDIA A6000 GPU. Our rendering speed is about 30fps.

Datasets. We repurpose Ego-Exo4D [20], which includes sparse-view videos of skilled human activities, for the task of 4D reconstruction. While many Ego-Exo4D scenarios are out of scope for dynamic reconstruction with existing methods (due to fine-grained object motion, specular surfaces, or excessive scene clutter), we find two challenging scenes with considerable object motion in controlled environments: *dance* and *bike repair*. We show additional results in the supplement. For each scene, we extract 300 frames of synchronized RGB video streams, captured from 4 different cameras with known parameters. We remove fisheye distortions from all RGB videos and assume a simple pinhole camera model after undistortion. We call this subset ExoRecon, and show results on these sequences.

	Method	Full Frame					Dynamic Only					
		PSNR ↑	SSIM ↑	LPIPS ↓	90°-NVS ↑	NTS ↑	PSNR ↑	SSIM ↑	LPIPS ↓	90°-NVS ↑	NTS ↑	IOU ↑
Pointmap	MonST3R [64]	20.10	0.65	0.436	9.15	—	21.51	0.657	0.277	10.46	—	0.33
3DGS-based	Dyn3D-GS [37]	10.59	0.475	0.601	7.51	9.70	11.36	0.479	0.544	8.19	10.09	—
	SOM [53]	12.37	0.475	0.574	6.09	11.03	13.14	0.485	0.537	6.67	11.38	0.23
	MV-SOM [53]	26.13	0.915	0.076	11.62	24.91	27.38	0.940	0.035	11.88	25.57	0.59
	Ours	**30.40**	**0.947**	**0.037**	**14.41**	**30.07**	**30.71**	**0.954**	**0.019**	**15.10**	**30.49**	**0.81**

Table 1. **Quantitative analysis.** We benchmark our method against state-of-the-art approaches by evaluating both the dynamic foreground region and the entire scene. Our results demonstrate improved performance across all metrics. Notably, all methods achieve better performance in the dynamic region, highlighting the challenge of reconstructing cluttered background scenes.

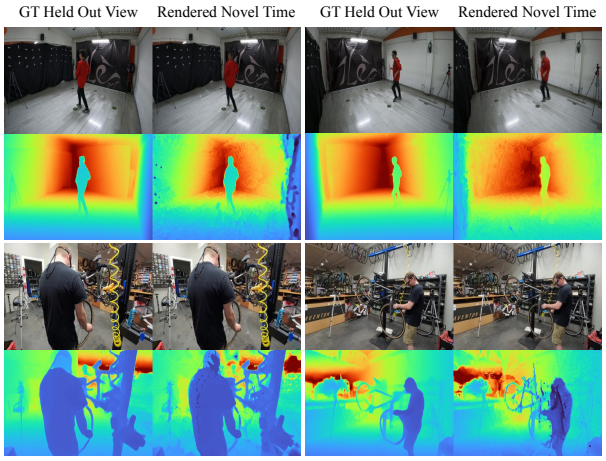

Figure 4. **Novel time synthesis.** We show qualitative novel-time synthesis results of our method on the bike repair (top) and dancing (bottom) sequences. We visualize the GT held out image on the left. The rendered novel-time RGB image closely matches ground truth. However, the depth map contains several floaters.

Metrics. We evaluate the perceptual and geometric quality of our reconstructions using a variety of metrics. First, we train on all 4 camera views and assess the quality of RGB training-view renderings, using PSNR, SSIM, and LPIPS. We also introduce the task of Novel Time Synthesis (NTS), which synthesizes motions at new timesteps by interpolating motion trajectories from training views. For 3DGS-based methods, we query per-Gaussian rotations and translations at timesteps t and $t+2$, and linearly interpolate the transformations at time $t+1$. We then render the interpolated Gaussians and report PSNR. We additionally evaluate the quality of the dynamic region silhouette by reporting mask IoU, computed as $(\hat{\mathbf{M}} \& \mathbf{M})/(\hat{\mathbf{M}} || \mathbf{M})$. Finally, we introduce the task of 90° Novel View Synthesis (90°-NVS), which trains our method on only 3 views and evaluates PSNR rendering metrics on the fourth held-out view after 4-fold cross-validation. Here, we evaluate on both the entire scene and the dynamic region only. We additionally sample a random camera at 5° or 45° distance from the training view cameras and qualitatively evaluate the view synthesis results from these views (5°-NVS and 45°-NVS).

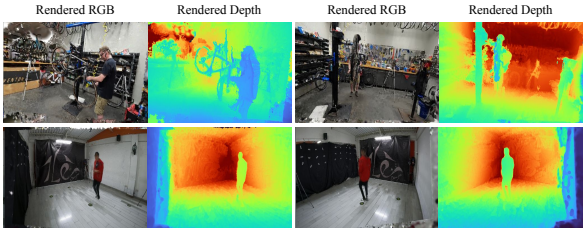

Figure 5. **5° novel view synthesis**. We show qualitative results for novel view synthesis on bike repair (top) and dancing (bottom) sequences. Notably, the rendered RGB and depth maps produce consistent reconstructions and plausible geometry.

Baselines. We compare our method with prior work on dynamic scene reconstruction from single or multiple views. Among methods that operate on monocular videos, we train Shape of Motion [53] on scenes from ExoRecon. We also compare against MonST3R [64] which estimates pointmaps from monocular videos. Finally, we consider two multi-view dynamic reconstruction baselines, Dynamic 3D Gaussians [37], and a naive multi-view extension of Shape of Motion (MV-SOM). To construct the latter baseline, we simply concatenate the Gaussians, motion bases, and optimization objectives as four separate instances of single-view SOM. We verify that all baselines reconstruct reasonable training views in the supplement.

4.1. Comparison to State-of-the-Art

In Tab. 1, we compare our method to recent dynamic scene reconstruction baselines [37, 53, 64]. Our method beats prior art across all metrics, when evaluated on training-view and novel-view rendering metrics across space (90°) and time (90°-NVS and NTS). While all methods achieve low photometric errors on training views, their performance on extreme 90° novel view synthesis plummets dramatically. This highlights the difficulty of learning plausible dynamic reconstructions from limited viewpoints.

Interestingly, we find that a multi-view geometry estimation method, MonST3R [64], performs surprisingly better than a monocular 4D reconstruction method, Shape of Mo-

tion (SOM) [53] and multi-view 4D reconstruction method, Dynamic 3D Gaussians[37]. We hypothesize that the foundational priors of MonST3R allow it to produce reasonable results in under-constrained scenarios, while test-time optimization methods, especially ones that do not always rely on data-driven priors like [37], can more easily fall into local optima (e.g. those caused by poor initialization) which are difficult to optimize out of via rendering losses alone.

4.2. Ablation Study

Space-time consistent depth. In Tab. 2, we ablate our choice of depth initialization. Our baseline is the depth alignment proposed in Shape of Motion [53], which aligns the high-fidelity relative depth estimates from Depth Anything v2 (DAv2) [60] with the metric depth estimates from UniDepth [43]. Our first observation is that these depth estimates are neither cross-view nor temporally consistent. We refine the alignment procedure in the following experiments. First, we align DAv2 estimates with multi-view consistent DUSt3R [54] instead of UniDepth, which slightly improves the photometric quality. Note that as the camera intrinsics and extrinsics are known in our setup, we can preset cameras in DUSt3R global alignment and extract metric depth. When DUSt3R depth is evaluated on its own however, we see a jump in performance which we attribute to DUSt3R's more accurate metric depth predictions. Interestingly, using depth estimates from a temporally-coherent depth estimator, MonST3R [64], on its own does not aid the optimization of cross-view consistent scene geometry as much. However, when the MonST3R estimates are aligned with metric DUSt3R estimates, we get the best photometric performance. We provide qualitative visuals in Fig. 10.

Pipeline components. We ablate the remaining design decisions in our pipeline in Tab. 3. Our proposed spacetime consistent depth plays a crucial role in learning accurate scene geometry and appearance (yielding a 3.4 PSNR improvement, Row 1 vs 3). Next, we find that the feature-metric loss $\mathcal{L}_{\text{feat}} = \left\| \hat{\mathbf{F}} - \mathbf{F} \right\|$ provides a trade-off between learning photometric properties vs.learning foreground motion and silhouette. Although the PSNR decreases, we see an increase in mask IoU (Row 1 vs 2 and Row 3 vs 4). Similarly, freezing the color of all Gaussians across frames aids learning the motion mask, as measured by mask IoU. Finally, our motion bases constructed from feature-clustering improve overall scene optimization (final row).

5. Conclusion

In this work, we address the problem of sparse-view 4D reconstruction of dynamic scenes. Existing 4D reconstruction methods are designed for either monocular casual captures (e.g. DAVIS) or dense multi-view setups (e.g. Panoptic Studio). In contrast to prior work, we explore the sparse-view setup with Ego-Exo4D. Our key insight is that carefully incorporating *priors*, in the form of consistent metric depth and feature-based motion clustering, are important to enable plausible and photorealistic 4D reconstructions of dynamic scenes. Our empirical analysis shows that we achieve state-of-the-art performance on novel spacetime synthesis.

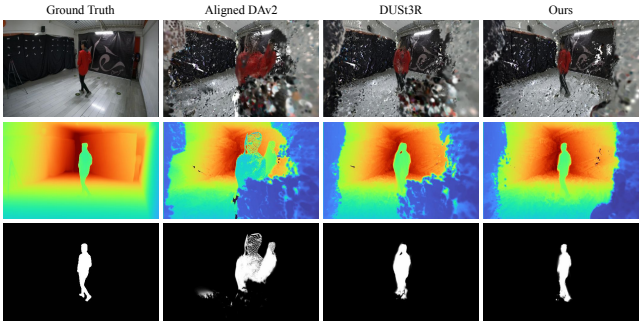

Figure 6. **Choice of depth estimates for initialization.** We qualitatively compare the depth optimized by three different variants of our method that use different depth initializations. We find that for the extreme novel-view synthesis setting, our spacetime consistent depth (far right) using MonST3R and DUSt3R achieves the best performance. Columns correspond to rows 1, 3 & 5 from Tab. 2.

$\mathfrak{D}_{src}$	$\mathfrak{D}_{tgt}$	$\mathfrak{A}_{cam}$	$\mathfrak{A}_t$	↑PSNR	↑SSIM	↓LPIPS	↑ 90°-NVS
DA-V2	Uni-D	✗	✗	26.19	0.915	0.077	11.9
DA-V2	DUSt3R	✓	✗	28.07	0.936	0.067	12.2
DUSt3R	✗	✓	✗	29.11	0.934	0.055	12.4
MonST3R	✗	✗	✓	28.63	0.928	0.059	12.5
MonST3R	DUSt3R	✓	✓	**30.40**	**0.947**	**0.037**	**14.4**

Table 2. **Ablation study in depth.** We ablate our choice of depth estimates used for initialization. We align pseudo-depth map from a source model $\mathfrak{D}_{src}$ with predicted metric depth estimates from a target model $\mathfrak{D}_{tgt}$. $\mathfrak{A}_{cam}$ and $\mathfrak{A}_t$ denote whether the final depth is cross-view / time consistent based on the choice of depth estimators. We find that using multi-view consistent depth alone (row 3) can improve sparse-view dynamic scene reconstruction. Aligning time- and view-consistent depth produces the best results.

Method	$\mathcal{L}_{\text{feat}}$	$\mathbf{d}_n$	$\mathbf{T}^{(b)}_{0\to t}$	↑PSNR	↑SSIM	↓LPIPS	↑IoU
Baseline	✗	✗	✗	26.19	0.915	0.077	0.60
+ $\mathcal{L}_{\text{feat}}$	✓	✗	✗	25.39	0.933	0.087	0.63
+ Our depth / no $\mathcal{L}_{\text{feat}}$	✗	✓	✗	29.55	0.944	0.037	0.73
+ Our depth / $\mathcal{L}_{\text{feat}}$	✓	✓	✗	29.31	0.941	0.041	0.75
+ Motion bases (**Ours**)	✓	✓	✓	**30.40**	**0.947**	**0.037**	**0.81**

Table 3. **Ablation study of pipeline components.** We ablate our choice of feature-metric loss, spacetime consistent depth, and feature-based motion bases. While the proposed depth and feature-based motion bases considerably improve 4D reconstruction (evaluated by photometric errors), we find that our feature loss helps learn better motion masks (evaluated by IoU).

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021. 5

[2] Joel Carranza, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM Trans. Graph.*, 22:569–577, 2003. 2

[3] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. *arXiv preprint arXiv:2304.12308*, 2023. 2

[4] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *arXiv preprint*, 2024. 1, 2

[5] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images, 2024. 2

[6] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Vedaldi, Tat-Jen Cham, and Jianfei Cai. MVSplat360: Feed-Forward 360 Scene Synthesis from Sparse Views, 2024. 2

[7] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023. 5

[8] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *SIGGRAPH*, 2008. 2

[9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021. 2

[10] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. *ICCV*, 2023. 5

[11] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2016*, 35, 2016. 2

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, pages 2366–2374, 2014. 3

[13] David A Forsyth and Jean Ponce. A modern approach. *Computer vision: a modern approach*, 17:21–48, 2003. 2

[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *CVPR*, pages 2002–2011, 2018. 3

[15] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, 2024. 3

[16] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[17] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021. 5

[18] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2

[19] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint 2405.10314*, 2024. 2

[20] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2023. 6

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 2

[22] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 3

[23] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 2

[24] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(1): 190–204, 2017. 1

[25] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 2, 3

[27] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. *arXiv preprint arXiv:2409.18121*, 2024. 1

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[29] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024. 2

[30] Jiahui Lei, Yufu Weng, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. In *CVPR*, 2024. 2

[31] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint 2406.09756*, 2024. 1

[32] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 5

[33] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[34] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[35] Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. *arXiv preprint*, 2023. 2

[36] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023. 2

[37] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 1, 2, 3, 7, 8

[38] 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. 2

[39] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[40] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. *arXiv preprint arXiv:2112.00724*, 2021. 2

[41] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 5

[42] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[43] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3, 5, 8

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[45] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 3

[46] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 3

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 3

[48] Chonghyuk Song, Gengshan Yang, Kangle Deng, Jun-Yan Zhu, and Deva Ramanan. Total-recon: Deformable scene reconstruction for embodied view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17671–17682, 2023. 1

[49] Colton Stearns, Adam W. Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel

view synthesis of casual monocular videos. *arXiv preprint arXiv:2406.18717*, 2024. 2

[50] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. 2024. 3

[51] Jeff Tan, Donglai Xiang, Shubham Tulsiani, Deva Ramanan, and Gengshan Yang. Dressrecon: Freeform 4d human reconstruction from monocular video. *arXiv preprint arXiv:2409.20563*, 2024. 2

[52] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 1

[53] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 1, 2, 3, 5, 7, 8

[54] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 3, 4, 8

[55] Hao Chen Zhipeng Cai Gang Yu Kaixuan Wang Xiaozhi Chen Chunhua Shen Wei Yin, Chi Zhang. Metric3d: Towards zero-shot metric 3d prediction from a single image. *ICCV*, 2023. 3

[56] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Wang Xinggang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint*, 2023. 2

[57] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, 2024. 2

[58] Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Gaussianobject: High-quality 3d object reconstruction from four views with gaussian splatting. *SIGGRAPH Asia*, 2024. 2

[59] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[60] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 3, 5, 8

[61] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 3

[62] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint*, 2023. 2

[63] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[64] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024. 1, 3, 4, 7, 8, 2

[65] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 3813–3824, 2023. 2

[66] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. *CVPR*, 2024. 3

[67] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: real-time few-shot view synthesis using gaussian splatting. *CoRR*, abs/2312.00451, 2023. 2

Why is Sparse-View 4D Reconstruction Hard?

Supplementary Material

A. Results on Additional Sequences

We provide novel-view synthesis results from additional sequences to show the generalization ability among all categories of EgoExo4D. Our additional results cover basic life scenarios, such as healthcare, dancing, cooking, music and bike repair. Specifically, we provide qualitative novel view rendering results from 5° and 45° novel views results. We also include 4D visualizations on the attached website.

B. Training View Renderings

To build confidence in our implementations, we validate every baseline we run by verifying that each method looks reasonable at training views. It is worth noticing that in each iteration of optimization, we sample a batch of frames out of the video to optimize the overall loss. As the loss is optimized as a global minimum averaged over all frames, it is possible that some artifacts remain for certain frames.

C. Training Details

In this section, we report the learning rate and loss weights of Gaussians in our optimization process. These hyperparameters are shared across every scene that we evaluated on. Specifically, $\mathcal{L}_{\text{smooth\_bases}}$ enforces smooth motion bases by penalizing high accelerations in rotations and translations. $\mathcal{L}_{\text{smooth\_tracks}}$ promotes smooth object tracks by penalizing large accelerations in object positions across frames. $\mathcal{L}_{\text{depth\_grad}}$ aligns the gradients of the predicted and ground truth depth maps to preserve structural details. $\mathcal{L}_{\text{z\_accel}}$ penalizes high accelerations along the depth axis to reduce jitter in depth estimation. $\mathcal{L}_{\text{scale\_val}}$ constrains the variance of scale parameters of Gaussians to achieve consistent representations.

Table 4. Learning Rates for Foreground (FG), Background (BG), and Motion Parameters

Parameter	FG LR	BG LR	Motion LR
means	1.6×10^{-4}	1.6×10^{-4}	–
opacities	1×10^{-2}	1×10^{-2}	–
scales	5×10^{-3}	1×10^{-3}	–
quats	1×10^{-3}	1×10^{-3}	–
colors	0	1×10^{-2}	–
feats	1×10^{-3}	1×10^{-3}	–
motion_coefs	1×10^{-3}	–	–
rots	–	–	1.6×10^{-4}
transls	–	–	1.6×10^{-4}

Table 5. Loss Weights Configuration

Loss Parameter	Weight	Loss Parameter	Weight
w_{rgb}	7.0	w_{mask}	5.0
w_{feat}	7.0	$w_{\text{smooth\_bases}}$	0.1
$w_{\text{depth\_reg}}$	1.0	$w_{\text{smooth\_tracks}}$	2.0
$w_{\text{depth\_const}}$	0.1	$w_{\text{scale\_var}}$	0.01
$w_{\text{depth\_grad}}$	0.1	$w_{\text{z\_accel}}$	1.0
w_{track}	2.0		

D. More Ablations

Velocity-based or feature-based motion bases? In the monocular setting, we empirically found that both designs performed equally well. However, in our 4 camera sparse view setting, we found that feature-based motion bases perform much better than velocity-based motion bases. The reason is that for velocity-based motion bases, we infer 3D velocity by querying the 2D tracking results plus depth per frame following Shape-of-Motion[53]. Thus, noisy foreground depth estimates where the estimated depth of the person flickers between foreground and backward will negatively influence the quality of velocity-based motion bases, causing rigid body parts to move erratically. In contrast, feature-based motion bases, where features are initialized from more reliable image-level observations, are more robust to noisy 3D initialization and force semantically-similar parts to move in similar ways. To validate our points, in Fig. 10 we use PCA analysis to visualize the inferred features and find that they are consistent not only on temporal axis but also across cameras.

Effect of different number of motion bases. When the number of motion bases is not expressive enough (in our experience when the number of motion bases < 20), there are often obvious flaws in the reconstruction, such as missing arms or the two legs joining together into a single leg. In reality, we do not observe that increasing the number of motion bases further hurts the performance. Empirically, the capacity of our design (which is **40** motion bases) can effectively handle different scene dynamics.

Feed-forward depth estimation models. In our problem setup where camera extrinsics are known, the additional flow-based loss and trajectory regularization introduced by MonST3R are not relevant. Therefore, MonST3R can be thought of as nothing more than DUSt3R with a fine-tuned

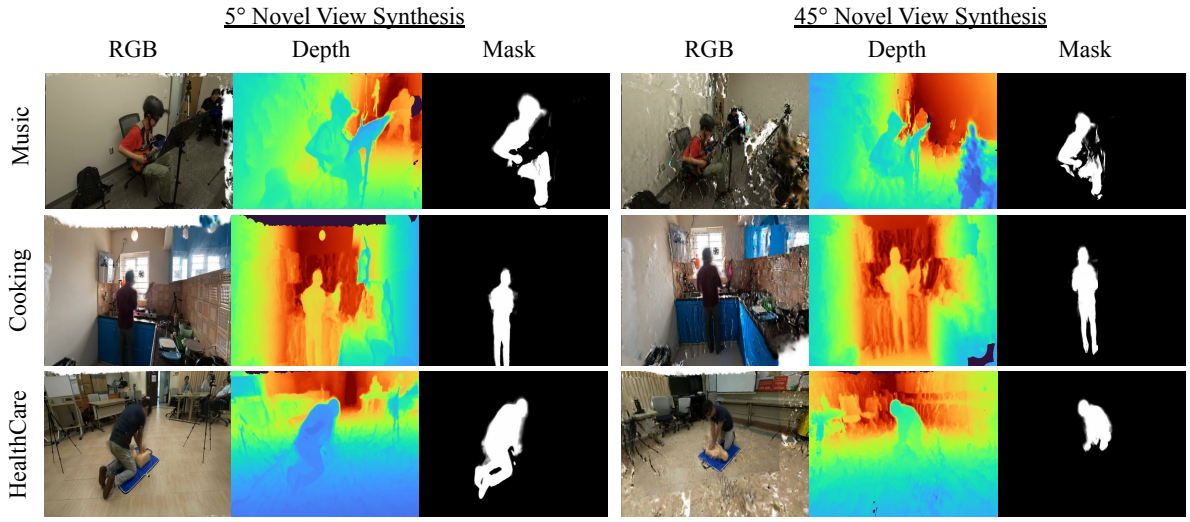

Figure 7. **Novel view synthesis results from more video sequences.** In each row, we visualize the rasterized RGB image, depth map, and foreground mask from our method for various diverse scene including music (top), cooking (middle), and healthcare (bottom). We include results for 5° (left) and 45° (right) novel view synthesis results. Notably, the rendered RGB and depth maps produce consistent reconstructions and plausible geometry.

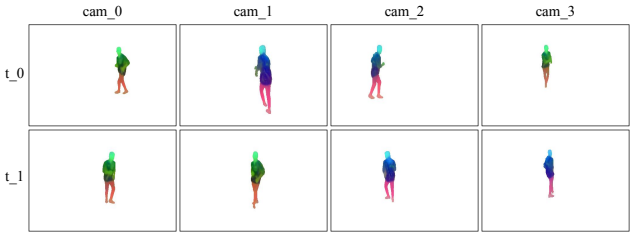

Figure 8. **Spatial-Temporal Visualization of feature PCA.** We perform PCA analysis and transform the 32-dim features from Sec. 3.3 down to 3 dimensions for visualization purposes. We find that the features are consistent across views and across time. Notably, when the person turns around between t_0 and t_1 in observations from cam_1 and cam_2, the feature remains robust and consistent. The semantic consistency of features aids explainability, provides a strong visual clue for tracking, and gives confidence in our feature-guided motion bases.

decoder and projection head, inferred on adjacent frames. To determine whether the MonST3R fine-tuning improves performance on EgoExo4D scenes, we conducted an experiment where we performed DUSt3R's global optimization routine but switched the checkpoint from DUSt3R to MonST3R. Specifically, at each timestep, we load images from the four cameras and create a complete scene graph (ending up with $4 \cdot 3 = 12$ edges) for global optimization. We observed that, although there was no obvious difference for background reconstruction, the results in Fig. 11 indicate that MonST3R is much better at foreground reconstruction. We hypothesize that this discrepancy exists because the dynamic foregrounds in EgoExo4D are out of distribution for the training data of DUSt3R, which consists largely of static indoor and outdoor scenes and lacks extensive human-centric observations.

E. Limitations and Future Work

We address two key limitations of our work. First, like previous methods, we rely heavily on 2D foundation models to estimate priors (e.g. depth and dynamic masks) for gradient-based differentiable rendering optimization. Thus, imprecise priors can harm the downstream rendering process. In addition, the current pipeline requires a user prompt to specify dynamic masks for each moving object [3], which can be labor-intensive for complex scenes. To solve this, distilling dynamic masks from foundation models or inferring dynamic masks from image level priors (as in [64]) could be beneficial.

Second, most off-the-shelf feed-forward depth estimation networks are trained on simple scene-level datasets, with few dynamic movers (e.g. people) in the foreground. In practice, we observe that the depth of humans in dynamic scenes is often incorrect when observed from other views. For example, DUSt3R often estimates the depth of a human to be the same as the depth of surrounding walls, causing the human to blend into the background. We believe that these fundamental problems with the depth predictions *cannot* be solved by any alignment in the output space. To mitigate this issue, we plan to further fine-tune DUSt3R or MonST3R on existing dynamic human datasets.

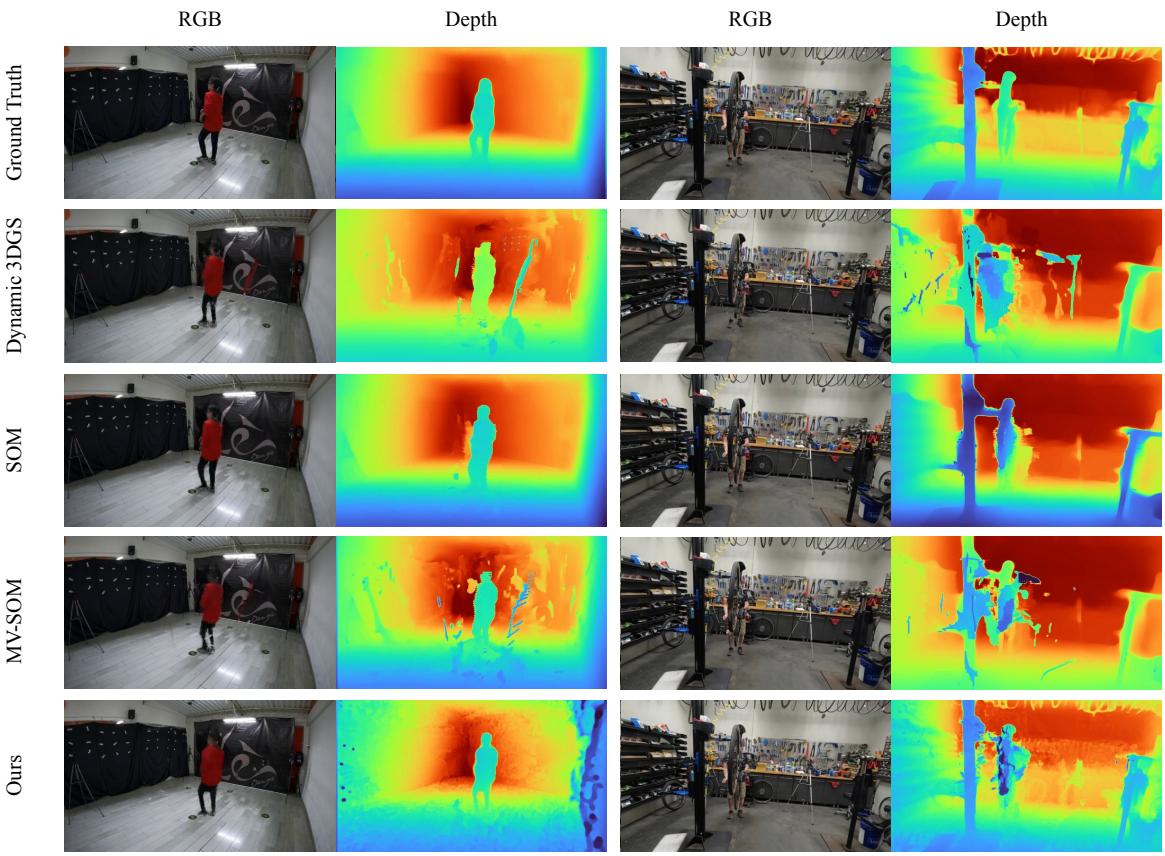

Figure 9. **Training view results.** We visualize the rasterized RGB image and depth map from each method for the dancing (left) and bike repair (right) sequences. All methods are capable of producing reasonable training views and depth maps. It is worth noticing that in each iteration of optimization, we sample a batch of frames out of the video to optimize the overall loss. As the loss is optimized as a global minimum averaged over all frames, it is possible that some artifacts remain for certain frames.

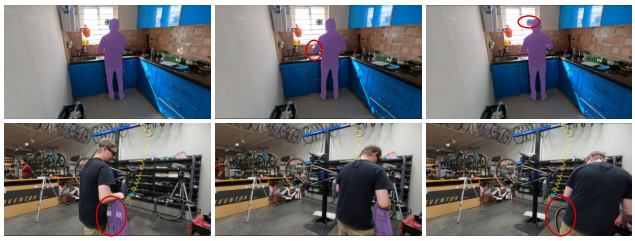

Figure 10. **Failure example of SAM-V2.** We qualitatively inspect the SAM-V2 dynamic foreground masks on the kitchen (top) and bike repair (bottom) scenes. The dynamic mask is highlighted in *purple*, and failures in dynamic mask estimation are highlighted in *red circle*. We observe that SAM-V2 can miss important body parts (e.g. the person's hands) or get confused by the background (as shown in top row). Long-term occlusion will also lead to tracking failure (as shown in bottom row). These failure cases suggest that dynamic mask tracking in complex scenes remains an open challenge.

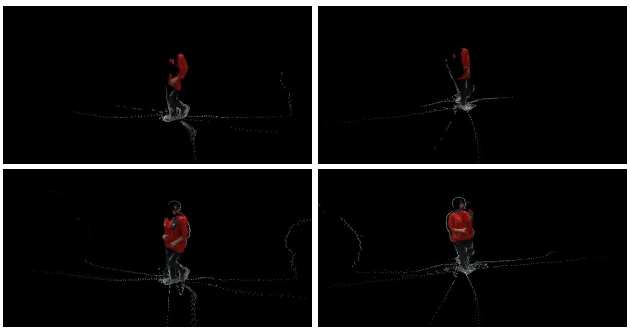

Figure 11. **Visualization of foreground projection for different checkpoints.** Here we show the projection by known cameras and ground-truth foreground masks, using the point cloud from DUSt3R (*top row*) and MonST3R (*bottom row*) for two selected cameras (each column represents one camera). Notably, although MonST3R is fine-tuned on temporal frame sequences instead of multi-view information, MonST3R benefits from the presence of dynamic foreground movers in its fine-tuning dataset and thus gives a better foreground result.