EPIC Fields
Marrying 3D Geometry and Video Understanding

Vadim Tschernezki[★♥♦] **Ahmad Darkhalil**[★♣] **Zhifan Zhu**[★♣]
David Fouhey[♠] **Iro Laina**[♥] **Diane Larlus**[♦] **Dima Damen**[♣] **Andrea Vedaldi**[♥]

[♥]VGG, University of Oxford [♣]University of Bristol
[♠]New York University [♦]NAVER LABS Europe [★]: Equal Contribution

Abstract

Neural rendering is fuelling a unification of learning, 3D geometry and video understanding that has been waiting for more than two decades. Progress, however, is still hampered by a lack of suitable datasets and benchmarks. To address this gap, we introduce EPIC Fields, an augmentation of EPIC-KITCHENS with 3D camera information. Like other datasets for neural rendering, EPIC Fields removes the complex and expensive step of reconstructing cameras using photogrammetry, and allows researchers to focus on modelling problems. We illustrate the challenge of photogrammetry in egocentric videos of dynamic actions and propose innovations to address them. Compared to other neural rendering datasets, EPIC Fields is better tailored to video understanding because it is paired with labelled action segments and the recent VISOR segment annotations. To further motivate the community, we also evaluate three benchmark tasks in neural rendering and segmenting dynamic objects, with strong baselines that showcase what is not possible today. We also highlight the advantage of geometry in semi-supervised video object segmentations on the VISOR annotations. EPIC Fields reconstructs 96% of videos in EPIC-KITCHENS, registering 19M frames in 99 hours recorded in 45 kitchens, and is available from: http://epic-kitchens.github.io/epic-fields

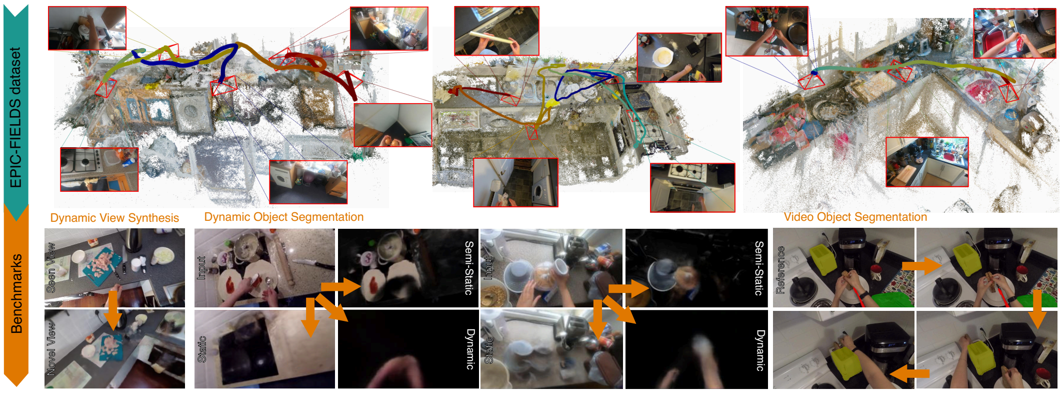


Figure 1: We propose EPIC Fields that extends EPIC-KITCHENS with 3D information, including full frame-rate camera pose trajectories (top). These are directly obtained from dynamic sequences of object interactions (sampled frames) without additional modalities or pre-scans. We showcase EPIC Fields through several benchmarks (bottom) that use the fusion of geometric and semantic cues.

37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks.

1 Introduction

Recent breakthroughs in neural rendering [57, 37] have enabled a deeper integration of machine learning in geometric tasks like 3D reconstruction and rendering, creating a new opportunity to bring 3D geometry and video understanding closer together. By representing videos in 3D we can explain away the variability induced by the camera motion, which is dominant especially in egocentric videos. We can also integrate information extracted from each frame independently into a global, consistent interpretation of the video, as demonstrated by semantic neural rendering [81, 63, 23, 74, 24, 14, 66]. However, such successes have been mostly limited to *static scenarios*, where only the camera moves. Indeed, 3D reconstruction still struggles with dynamic content and much work remains before we can have a 3D understanding of dynamic phenomena like actions and activities.

An obstacle to further progress in 3D video understanding is the lack of suitable development data. In this paper, we address this gap by introducing *EPIC Fields*, an extension of the popular EPIC-KITCHENS [7] dataset which adds reconstructed 3D cameras and new benchmark tasks assessing both 3D reconstruction and semantic video understanding.

We choose to build on EPIC-KITCHENS because it is an established benchmark for 2D video understanding with rich annotations. Furthermore, it contains egocentric videos which are likely to benefit from 3D understanding, but which also challenge existing 3D reconstruction techniques due to their highly dynamic content and long duration (up to one hour). Dynamics include the motion of the actor and of the objects that they manipulate, as well as object transformations (*e.g.*, slicing a carrot). Furthermore, most objects are mostly static, moving only during brief spells of active manipulation. These challenges push the limits of current dynamic 3D reconstruction methods, which are usually restricted to short videos or focus on one class of objects [49, 45, 43, 29, 25, 58].

Obtaining camera information for EPIC-KITCHENS is challenging since structure-from-motion methods often fail on such complex videos. By solving this problem, we make it much easier for other researchers to start on 3D video understanding even when they are not experts in 3D vision, similar to what the setup and data introduced in works like NeRF [37] has done for static 3D reconstruction. Furthermore, while we propose specific benchmark tasks, we anticipate that researchers will be able to use our dataset to investigate many more questions than the ones we investigate here.

In summary, our first contribution is to augment EPIC-KITCHENS with camera information. To overcome the limitations [19] of traditional structure-from-motion pipelines [53], which struggle with egocentric videos, we introduce a pre-processing step that intelligently sub-samples frames from these videos, resulting in higher reconstruction reliability and speed. Our second contribution is to introduce new benchmark tasks that require or can benefit from the 3D cameras: dynamic novel view synthesis (*i.e.*, reconstructing unseen frames given a subset of frames from a monocular video); identifying and segmenting objects that move independently from the camera; and semi-supervised video object segmentation. These benchmarks use and extend the VISOR [10] annotations to provide dense ground-truth semantic labels. We report a number of baselines and conclude that, while 3D reconstruction can indeed benefit video understanding, existing approaches are challenged by the dynamic aspects of EPIC Fields.

2 Related work

Egocentric action understanding using 3D. Some egocentric datasets [42, 9, 17] contain static 3D scans of the recording locations. These typically do not contain actions, or the environments are scanned post-hoc, usually with an additional step. For instance, [42] uses stereo egocentric cameras, but no activities, and in [9, 17], reconstruction is done afterwards via hardware or additional dedicated scans. These scans are costly, which is why just 13% of Ego4D [17] data comes with a 3D scan. In contrast, we provide a pipeline for estimating camera poses from egocentric data without additional hardware or scans, which we demonstrate on an existing, challenging dataset EPIC-KITCHENS.

Inferring cameras in egocentric videos. In this work, we perform the challenging task of reconstructing 3D camera poses from egocentric videos that show dynamic activities from a single camera. Since the EPIC-KITCHENS [8] dataset is unscripted, the videos show natural interactions by participants in their homes, who act swiftly due to familiarity. Prior work [19, 39, 59] on these videos highlights the challenge. In [19], where ORB-SLAM was used to find short clips where the camera pose was stable, the authors note that bundle adjustment failed and reconstructions lasted for just 7 second intervals. Using [19], [39] found hot-spots, but commented that just 44% of the frames could be registered. Others have used additional hardware information; for instance, [59] proposed using IMU data to establish short-term trajectories. In contrast, this work shows how to reconstruct cameras for *full* videos in EPIC-KITCHENS, without additional assumptions, data, or hardware.

Multi-view videos. A different approach to enabling neural rendering is calibrated multiview setups. Many of these datasets, however, capture humans in a “blank context”, including HumanEva [56], Human3.6M [21], AIST++ [65, 27], and ZJU-Mocap [47]. There are datasets capturing humans in complex environments, such as the Immersive Light Field dataset [2], NVIDIA Dynamic Scene Datasets [77], UCSD Dynamic Scene Dataset [32], and Plenoptic Video datasets [28]. However, these videos are short (1–2 min) and, due to the capture setup, show actions outside of their natural environment. In contrast, EPIC-KITCHENS is captured with an egocentric camera and shows long captures of indoor activities. Our contribution of reconstructing the cameras over time turns the egocentric data into the multiview data needed while retaining the naturalness of the data.

NeRF and dynamics. NeRF extensions to dynamic data can be roughly divided into approaches that add time as an additional dimension of the radiance fields [36, 64, 71, 15, 69, 28, 52, 3] and those that instead model explicitly 3D flow and reduce the reconstruction to a canonical (static) one [49, 43, 77, 44, 68, 62, 29, 11, 79, 58, 20, 13, 30, 34]. While these methods demonstrate successes, their success depends on the dominance of camera motion over scene motion [16]. Scene motion by dynamic objects is not always common in existing datasets. Our proposed EPIC Fields contains both camera motion and fast continuous motion by the actor visible in the camera's field of view.

NeRF and semantics. Authors have already noted that neural rendering and 3D geometry can be helpful allies of video understanding. For instance, Semantic NeRF [81, 66] proposes to predict dense semantic labels in addition to RGB colours, while [24, 14, 55, 67] consider panoptic segmentations (things and stuff). [63, 23, 31] propose to fuse semantic features from pre-trained ViTs [5, 26, 61] into a neural reconstruction. [74, 80] represent a scene as a composition of static objects given their 2D masks. Several studies employ neural rendering to separate scenes into objects and background either without or with weak supervisory signals [12, 72, 78, 64, 54, 41, 38, 70]. With a few exceptions [64, 31, 70], however, little work has been done on decomposing *dynamic* scenes into objects.

3 The EPIC Fields dataset

We introduce here the new *EPIC Fields* dataset. We first describe the content of the dataset and then the process of constructing it, including several technical innovations that made it possible.

3.1 EPIC Fields in a nutshell

EPIC Fields extends EPIC-KITCHENS to include camera pose information. EPIC-KITCHENS contains videos of cooking activities collected using a head-mounted camera in 45 different kitchens. It has semantic annotations for fine-grained actions and their action-relevant objects, including 90K start-end times of actions [8]. VISOR [10] adds 272K manually annotated masks and 9.9M interpolated masks of hands and active objects. With EPIC Fields, we further contribute camera extrinsic parameters for each video frame as well as camera intrinsic parameters. Using the technique described in Section 3.2, we successfully processed 671 videos spanning all 45 kitchens, resulting in 18,790,333 registered video frames with estimated camera poses.

Motivation. Our camera annotations facilitate reconstructing and interpreting videos in 3D. Figure 2 illustrates this point by mapping some 2D action annotations from EPIC-KITCHENS to the 3D space. Lifting annotations to 3D puts them in the wider context of the environment where actions occur, and enables studying the relevance of 3D egocentric trajectories to actions (for anticipation), objects (for understanding object state changes), and hand-object understanding. The figure also illustrates mapping hand meshes extracted using [51] to the 3D context of the kitchen.

Ethics, licensing, data protection. EPIC-KITCHENS was collected with ethics approval by the University of Bristol and explicit consent from the participants. The data does not contain personal identifiable information or offensive content and is provided under a non-commercial license. EPIC Fields is released under the same terms.

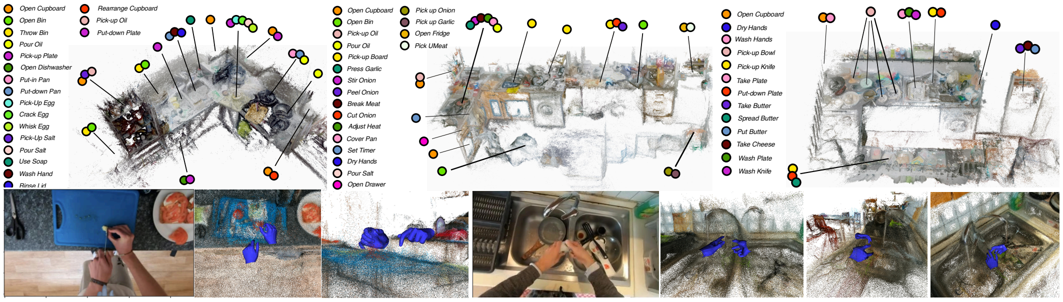


Figure 2: EPIC Fields unlocks applications that combine interactions with 3D information. We showcase examples of actions grounded in 3D (top row), and examples of integrating single-image 3D hands [51] into the kitchen reconstruction during interactions (bottom row).

3.2 Dataset construction

Because EPIC-KITCHENS videos were not collected with 3D reconstruction in mind they are difficult to reconstruct. For instance, they contain many dynamic objects: hands are visible in 95% of the frames and the focus of attention is often an object actively manipulated. Standard reconstruction pipelines operate under the assumption that the scene is static and are thus only moderately robust to dynamic objects. Other challenges include the video length (~9 mins on average) and the skewed distribution of viewpoints: videos alternate phases of small motion around hot-spots (*e.g.*, cooking at a hob or washing at the sink) and fast motion between hot-spots (*e.g.*, moving the pot to the sink).

We address these challenges by: (1) filtering videos to reduce the number of redundant frames, computational cost, and skew; (2) using structure from motion (SfM) to reconstruct the scene from the filtered frames; (3) registering the remaining frames to the sparse reconstruction. We accept a video's reconstruction if 70% or more of its frames are registered successfully. In this manner, we can reconstruct 96% of all EPIC-KITCHENS videos. We next describe each step, with details in the supplement.

Frame filtering. The goal of frame filtering is to downsample a video to reduce redundancy and skew while maintaining sufficient viewpoint coverage for accurate reconstruction. We filter frames by seeking temporal windows where frames have substantial visual overlap and then only keep one frame per window, similar to redundant frame mining [53, 60] and other SfM or SLAM pipelines. Overlap between frames is measured by estimating homographies by matching SIFT features [35]. Given a homography H between two frames, we define their visual overlap $\tilde{r}$ to be the fraction of the first frame area covered by the quadrilateral formed by warping the second frame corners by H. Windows are formed greedily, finding runs of frames $(i+1,\ldots,i+k)$ with overlap $\tilde{r} \geq 0.9$ to the first frame i and discarding them. Filtering discards on average about 82% of frames in each video while also retaining a sufficient number of frames in the critical transitions between hot-spots.

Sparse reconstruction. The filtered frames are fed to an off-the-shelf structure-from-motion pipeline. Among these, we found COLMAP [53] to be more effective than VINS-MONO [50], which suffered from frequent drifts and restarts.

In Table 1 we analyse the effectiveness of the homography-based filtering algorithm by comparing it to a naïve filter that subsamples frames uniformly. We use 30 randomly selected videos for this experiment and report two standard SfM metrics [53]: the average reprojection error and the number of 3D points in the reconstruction. The first metric is a proxy for the accuracy of the reconstruction, and the second for its coverage. Both filtering techniques reduce the number of frames equally and thus result in similar computational complexity. However, homography-based filtering also addresses the skew and results in a significantly better success rate, increased coverage, and reduced reprojection error compared to uniform subsampling. Besides considering the number of points reconstructed, Figure 3 shows qualitatively the notably improved coverage obtained by homography-based filtering.

Dense reconstruction, automated verification, and restart. After obtaining the sparse reconstruction from the filtered subset of video frames, we use COLMAP to register the remaining frames against it, which is computationally cheap. We accept the final reconstruction if ≥70% of the video's frames, at full frame rate, are registered successfully. This process succeeds in 90% (631 videos)

Table 1: **Impact of frame filtering on the reconstruction quality.** We compare the sparse reconstruction of 30 videos using either homography-based or uniform frame filtering. Naïve uniform sampling results in only 27 of the 30 videos being reconstructed successfully (*i.e.*, dense registration rate ≥ 70%). Furthermore, the successful reconstructions have significantly reduced coverage (-16.64%) and increased reprojection error (+4.76%) compared to homography-based filtering.

Frame sampling	**Avg. #3D Points**	**Avg. Repr. Error**	**Avg. Reg. Rate**	**Successful Reconstructions**
Homography-based (ours)	27,763	0.798	98.6%	30/30
Uniformly	23,142	0.836	89.0%	27/30
Relative change	-16.64%	4.76%	9.77%	-10 %

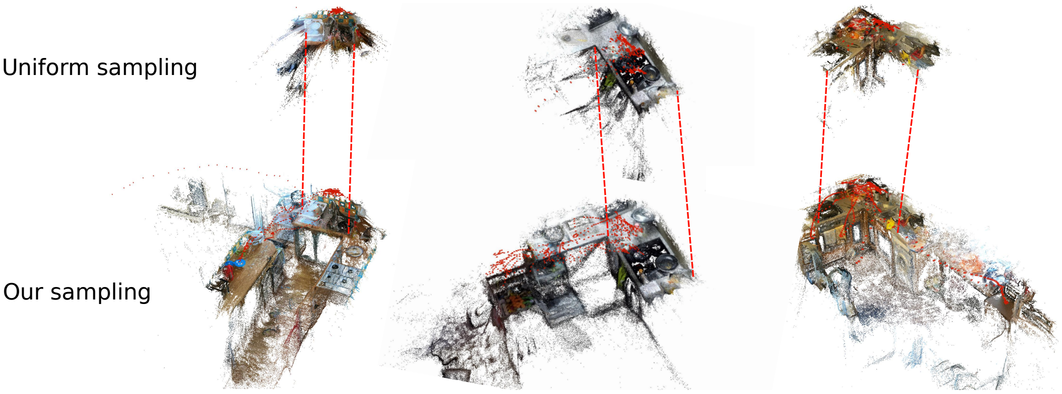


Figure 3: **3D reconstructions with different sampling.** We compare three scenes reconstructed using either uniform frame selection or our homography-based pipeline. Uniform sampling yields partial reconstructions with limited coverage. Ours demonstrates superior performance, resulting in better coverage by registering successfully more viewpoints.

of cases. When a video is rejected, the reconstruction process is attempted again with a higher threshold $\tilde{r} \geq 0.95$; this usually doubles the number of frames that COLMAP needs to process for the reconstruction, but increases the success rate to 96%. We discuss reasons for the failure of the last 29 EPIC-KITCHENS videos in the supplement.

Application to other egocentric videos. While we developed our reconstruction pipeline by considering the EPIC-KITCHENS data, the approach we obtained is general and applies equally well to other egocentric video collections such as Ego4D [18], at least for indoor locations. We give examples of these reconstructions in the supplement.

4 The EPIC Fields benchmarks, experiments and results

We define three benchmarks on EPIC Fields that probe 3D video understanding. Annotations, evaluation code and baselines are released as part of EPIC Fields; further details are in the supplement.

4.1 Dynamic New-View Synthesis (D-NVS)

Given a subset of video frames as input, the goal of dynamic new-view synthesis (D-NVS) is to predict other video frames given only their timestamps and camera parameters. While other D-NVS benchmarks exist, EPIC Fields is more challenging due to the first-person perspective and the large number of dynamic objects. In Table 2 we compare EPIC Fields to commonly used datasets in D-NVS. EPIC Fields offers a significant step up in complexity and scale with *significantly longer* videos and associated semantics. For detailed statistics, please refer to the supplement.

Video selection. Due to the computational cost of most D-NVS algorithms, we limit the D-NVS benchmark to a subset of 50 videos (14.7 hours and 2.86M registered frames) extracted from the train/val set of VISOR [10] (this selection includes 96.1% of the frames annotated in VISOR).

Table 2: Comparison of datasets commonly used in dynamic new-view synthesis.

Dataset	**#Scenes**	**Seq. Length**	**Monocular**	**Semantics**
Nerfies [44]	4	8–15 sec	✗	✗
D-NeRF [49]	8	1–3 sec	✗	✗
Plenoptic Video [28]	6	10–60 sec	✗	✗
NVIDIA Dynamic Scene Dataset [77]	12	1–5 sec	4 / 12	✗
HyperNeRF [45]	16	8–15 sec	13 / 16	✗
iPhone [16]	14	8–15 sec	7 / 14	✗
SAFF [31]	8	1–5sec	✗	✓
EPIC Fields [D-NVS] (ours)	50	6–37 min (Avg 22)	50 / 50	✓

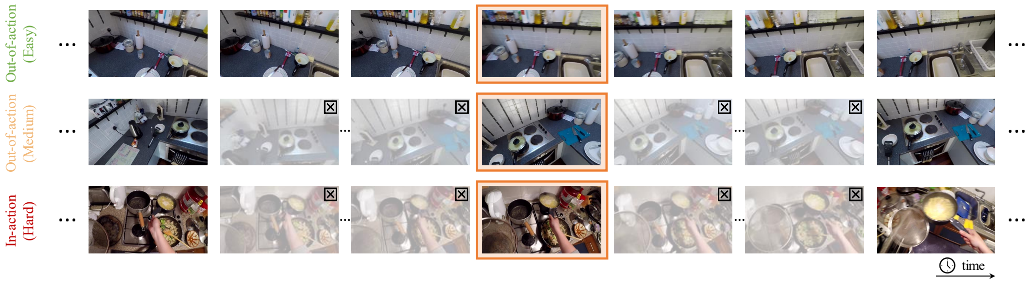


Figure 4: **Definition of the three difficulty levels for the task of dynamic new-view synthesis.** Validation and test frames are selected to meet three reconstruction difficulty levels. **In-Action frames (Hard)** happen during an action and are harder to reconstruct due to the dynamics. **Out-of-Action (Medium) frames** happen outside an action, but are far from a train frame. **Out-of-Action (Easy) frames** are near train frames. Frames in a bounding box (orange) represent either val/test frames. Frames marked with a cross are discarded to create a larger time gap around each val/test frame (medium and hard levels). All other frames can be used for training.

Frame selection. For each video in the D-NVS benchmark, we select the video frames to be used as input to the system (training) and those that remain unseen and are used for evaluation only (validation/testing). Specifically, we propose categorising evaluation frames into three tiers of difficulty (easy, medium, hard — visualised in Figure 4), determined by the type of motion and the temporal gap between the evaluation and training frames. **In-Action** frames correspond to common 'put', 'take', and 'cut' actions annotated in EPIC-KITCHENS, based on their start-stop times; they are characterised by substantial object motion due to hand-object interactions and are thus more difficult to reconstruct. In pursuit of a greater challenge, for the **In-Action (Hard)** set of frames, we exclude frames from the training set occurring within 1 second of a test frame. **Out-of-Action** frames occur outside action segments, where there is no appreciable motion except for the camera, making these frames generally easier to reconstruct. For the **Out-of-Action (Medium)** set, we sample 70% of the out-of-action frames with the same time gap as above. The **Out-of-Action (Easy)** set corresponds to the remaining 30% without removing the neighbouring training frames. The reasoning is that it is generally easier to predict a frame temporarily close to a training one. We assign every other evaluation frame to the validation and test sets, respectively. The average time gap between consecutive evaluation frames is 3.73 seconds. Further statistics are provided in the supplement.

Benchmark methods. To demonstrate how EPIC Fields can be used and to probe the limits of the state of the art in such challenging scenarios, we consider three neural rendering approaches: NeRF-W [36], NeuralDiff [64], and T-NeRF+, an extended version of T-NeRF [16].

NeuralDiff [64] is a method tailored to egocentric videos. It uses three parallel streams to separate the scene into the actor, the transient objects (that move at some point in the video), and the background that remains static. We combine the predictions of the actor and transient objects to predict our *dynamic* and *semi-static* objects, which will be relevant in Section 4.2.

NeRF-W [36] augments NeRF with the ability to 'explain' photometric and environmental (non-constant) variations by learning a low-dimensional latent space that can modulate scene appearance and geometry. As a result, NeRF-W also separates static and transient components. We follow the modification from [64] to render NeRF-W applicable to video frames and the D-NVS task.

Table 3: **Dynamic new-view synthesis**. We compare different neural rendering approaches for frames from different difficulty levels (easy, medium, hard). We report PSNR considering all pixels in each test frame. Given the mask annotations from VISOR for *In-Action* frames, we also report PSNR on background (BG) and foreground (FG) pixels separately for the hard (*In-Action*) setting.

Method	**Easy**	**Medium**	**Hard**		
			All	BG	FG
NeRF-W [36]	21.13	19.3	17.93	18.99	13.54
T-NeRF+ [16]	21.58	19.81	18.44	19.73	13.74
NeuralDiff [64]	22.14	19.88	18.36	19.54	13.37

T-NeRF+ [16] was proposed as a baseline to evaluate state-of-the-art NeRFs on dynamic scenes. It was shown to outperform other methods in terms of the quality of the synthesised images. We extend T-NeRF by adding another stream to the time-conditioned NeRF architecture that models the background (static parts of the scene).

Results. To measure performance on this task, we report the Peak Signal-to-Noise Ratio (PSNR) of the test frame reconstructions, which is a proxy for the quality of the underlying 3D reconstructions with the key advantage of not requiring 3D ground-truth for evaluation. We report results in Table 3 for the three levels of difficulty. There is a strong relationship between PSNR and difficulty: PSNR is consistently lower for all methods when rendering views during actions (hard) compared to outside actions (medium, easy). Some limitations of rendering these hard test frames are shown in Figure 5. For example, the bottom row shows that no 3D baseline renders the person's arm correctly, since all models struggle to interpolate the person's movement between frames. We further observe a significant gap in rendering quality if we calculate PSNR separately for foreground and background regions. We use the VISOR annotations of hands and active objects for In-Action frames to obtain this separation. These results not only highlight the existing limitations of current methods but also offer a valuable benchmark for assessing potential improvements in a targeted manner.

4.2 Unsupervised Dynamic Object Segmentation (UDOS)

The goal of Unsupervised Dynamic Object Segmentation (UDOS) is to identify which regions in each frame correspond to dynamic objects. This task can be approached in 2D only but is a good proxy to assess 3D methods as well, and can, in fact, be boosted by 3D modelling. Here, we extend the setting introduced in [64], using 20× more data and adopting a more nuanced evaluation protocol.

Video and frame selection. We use the same selection of videos as for the D-NVS task, but only use the In-Action frames with VISOR annotations, as they provide ground-truth dynamic object segmentations. We convert VISOR masks into a foreground-background mask for each frame in three ways, depending on objects that are currently moving, or those that have moved at a different time in the video. In the **dynamic objects only** setting, the foreground contains hands and other visible body parts as well as object masks only for objects that are currently being moved. We use the VISOR contact annotations to identify these objects and augment these with additional manual masks for visible body parts including torsos, legs, and feet. More details are in the supplement. In the **semi-static only** setting, we consider only objects that moved at some point during the video, but not during the current frame. We select these objects by watching the video and identifying all objects that have moved at least once. VISOR contains annotations of these objects only on frames where they are considered *active*. We employ an automated method to propagate the annotations to cover all frames, resulting in a set of semi-static object masks. This is the complete set of masks for all objects that have moved at any point in the video, even if they are temporarily static. More details can be found in the supplement. We combine both to report the **dynamic and semi-static** setting.

Benchmark methods. We use NeuralDiff and NeRF-W from the NVS task, since, by design, they decompose scenes into static and dynamic components. Additional considerations are necessary to make *T-NeRF+* applicable to UDOS. In order to disentangle the modelling of both radiance fields in terms of temporal variation, we apply the uncertainty modelling from [36] to model a change in observed colours of pixels that occur due to dynamic effects inside the scene. This extension enables *T-NeRF+* to learn a decomposed radiance field.

We also consider a 2D baseline, *Motion Grouping (MG)* [75], a state-of-the-art method for self-supervised video object segmentation. It trains a segmentation model using an autoencoder-like

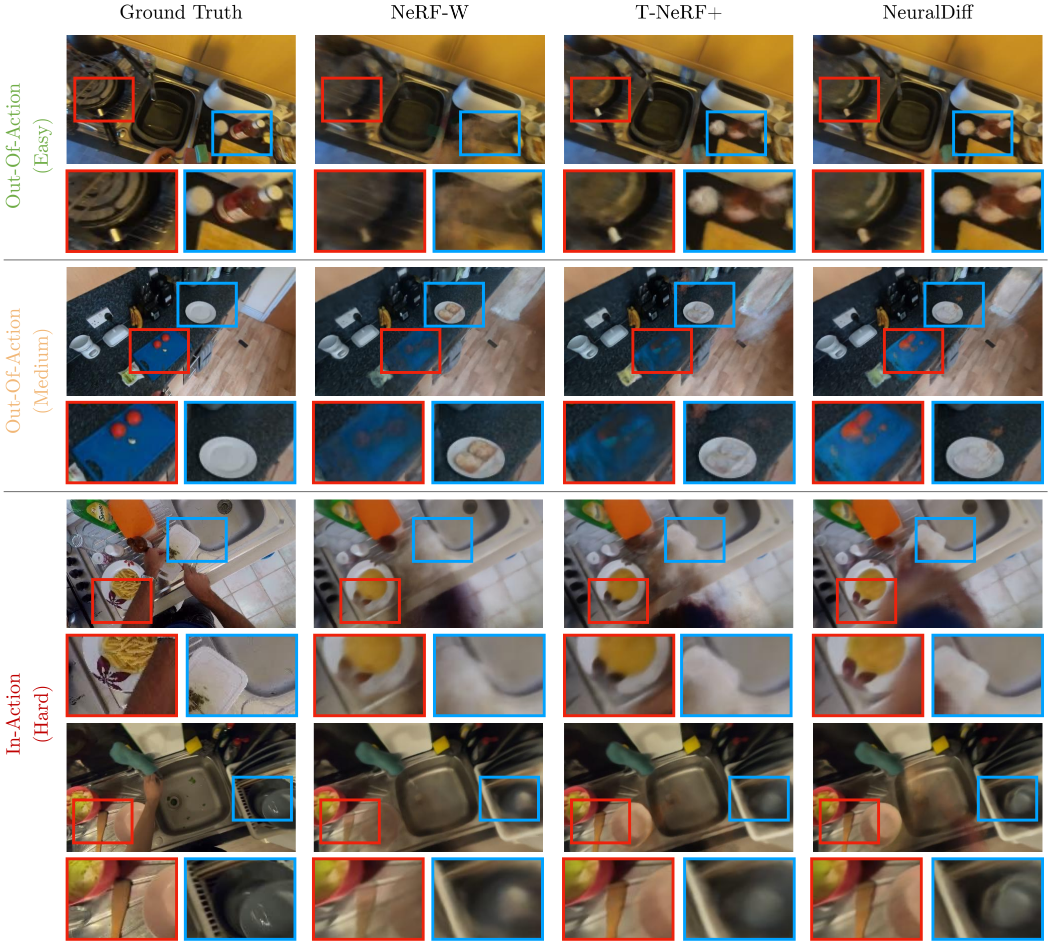


Figure 5: **Dynamic new-view synthesis.** We compare the outputs of 3D methods NeRF-W [36], T-NeRF+ [16], and NeuralDiff [63], for novel viewpoints, across three different complexity levels. The predictions are more accurate with less difficult motion as shown in the first and second row. The task becomes more challenging for our hard samples.

framework. The model has two output layers; one layer represents the background, and the other layer identifies one or more moving objects in the foreground, including their opacity layers. These layers are then linearly composed and optimised to reconstruct the input image. Since this approach is unsupervised, it can be compared fairly to the 3D baselines for this task.

Results. To evaluate performance, we measure 2D segmentation accuracy on test frames using mean average precision (mAP) as in [64]. Table 4 compares unsupervised 2D baselines and 3D baselines. Depending on the type of observed motion, 3D-based methods offer advantages over 2D methods and vice versa. For example, 3D-based methods are better suited for discovering semi-static objects that are not currently in motion, *i.e.*, they have been moved at different times within a video. This is evident by the improved segmentation performance when considering this type of motion (*i.e.*, *SS+D* and purely SS). However, we note that none of the 3D-based methods explicitly consider motion. Consequently, MG, which takes as input optical flow, performs better on purely dynamic motion, but struggles to segment objects that are temporarily not moving. This drawback of 3D-based methods, compared to 2D motion-based methods, underscores the current challenge in capturing dynamics in neural rendering. Addressing this limitation is an open question for future research.

Figure 6 shows qualitative results. We observe that MG performs particularly well on objects that are constantly in motion, for example, the moving body parts of the person. Among the 3D methods, NeuralDiff is better at capturing dynamic objects, and, unlike MG, both NeuralDiff and T-NeRF+ are able to segment various semi-static objects as well since they do not rely on continuous motion.

Table 4: **Unsupervised dynamic object segmentation**. We report the mean average precision (mAP) on segmenting the semi-static (SS) and dynamic components of the scene, and also their union (SS+Dyn). All methods are trained without explicit supervision, *i.e.*, no masks are used during training, only for evaluation.

Method	3D	SS+Dyn	SS	Dynamic
MG [75]	✗	55.53	12.78	64.27
NeRF-W [36]	✓	45.62	20.97	28.52
T-NeRF+ [16]	✓	64.91	24.48	44.27

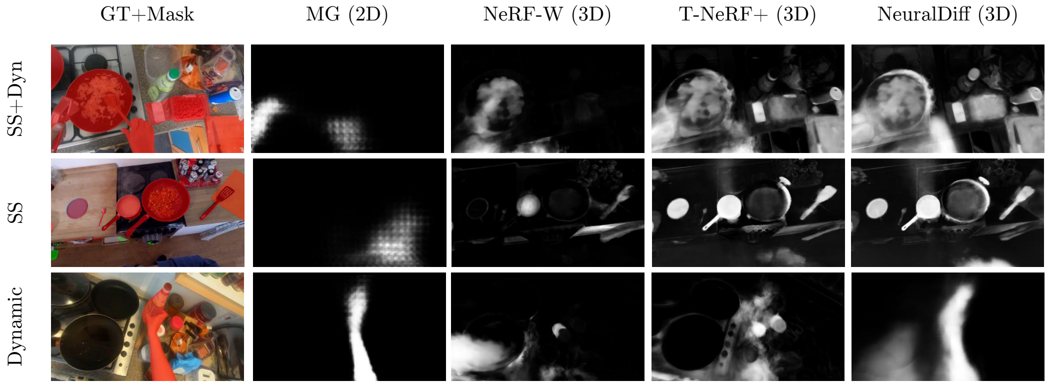


Figure 6: **Unsupervised dynamic object segmentation.** We compare three 3D baselines (NeRF-W [36], T-NeRF+ [16] and NeuralDiff [63]) and one 2D baseline (MG [75]) with three motion types. The 2D baseline captures the person and does well on the purely dynamic (short-range motion) setting. 3D models do this and also segment semi-static (SS) components (long-range motion, i.e., objects that were moved some time ago). In *SS+Dyn*, the evaluation includes SS and dynamic components.

4.3 Semi-Supervised Video Object Segmentation (VOS)

Semi-Supervised Video Object Segmentation (VOS) is a standard semi-supervised video understanding task: given the mask for one or more objects in a reference frame, the goal is to propagate the segments to subsequent frames. For this task, we use the train/val splits published as part of the VISOR VOS benchmark (See [10] Sec. 5.1). VOS is usually approached by using 2D models. Here, we explore how the 3D information in EPIC Fields can be used for it instead.

Benchmark methods. We evaluate two naïve baselines for this task, one in 2D and another in 3D. For completeness, we also compare these to existing, trained 2D VOS models.

Fixed in 2D. We make the assumption that the pixels in the first frame remain constant throughout the entire sequence. This naïve baseline is prone to failure when the camera undergoes movement.

Fixed in 3D. To better understand the potential of 3D information for VOS, we compare the 2D baseline above to a 3D one. In the 3D baseline, an object mask is projected to 3D and its position in 3D is fixed throughout the sequence. The mask is then re-projected to other frames using the available camera information. This works well for static objects and achieves two effects. First, objects can be reliably tracked over occlusions. Second, detecting when these objects are in or out of view is a by-product of estimated camera poses.

Trained 2D models. We also evaluate two state-of-the-art models for video object segmentation, STM [40] and XMEM [6]. These are trained on the train set of VISOR.

Results. We compare the baselines on the VISOR benchmark using the evaluation metrics defined in [48] which are the region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$. We also distinguish the set of objects that are static, such as 'fridge', 'floor', and 'sink', and report the above metrics separately for these and all other movable objects (SS+Dyn). Table 5 shows the results where the *Fixed in 3D* clearly outperforms the *Fixed in 2D* by a significant margin for the anticipated *static* objects

Table 5: **Semi-Supervised VOS**. We compare naive baselines in 2D and 3D, as well as pretrained/fine-tuned models on static and dynamic objects on the validation set of VISOR VOS. *: two videos from the validation set are excluded as they don't have successful reconstructions.

Method	3D	VISOR VAL[10]			Static			SS + Dyn		
		$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Fixed in 2D	✗	12.5	13.4	11.6	17.8	23.8	11.6	12.0	11.9	12.0
Fixed in 3D *	✓	31.3	30.5	32.2	48.4	52.2	44.6	29.6	27.8	31.5
Pretrained STM	✗	63.0	60.8	65.2	64.3	65.4	63.1	63.7	60.8	65.5
Fine-tuned STM	✗	76.4	74.2	78.6	76.8	77.7	76.0	76.6	73.8	79.5
Pretrained XMEM	✗	64.0	61.5	66.4	63.2	64.0	62.5	64.1	61.1	67.1
Fine-tuned XMEM	✗	77.3	75.2	79.4	77.0	77.7	77.4	78.0	75.3	80.7

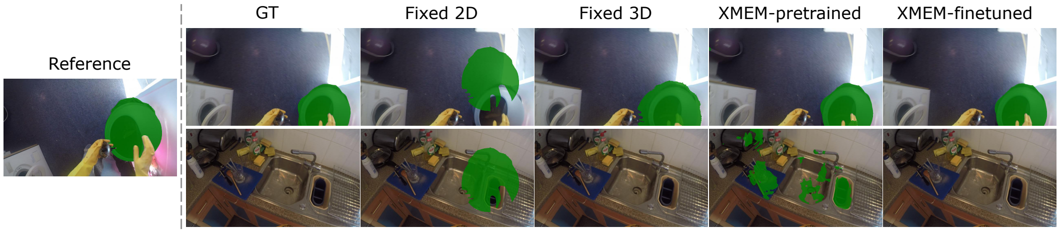


Figure 7: **Semi-Supervised Video Object Segmentation.** We compare our baselines on two frames from the same sequence. The *Fixed in 3D* baseline can track the bin over camera motion and recognise in/out-of view. Pretrained models usually suffer from false positives in the out-of-view scenes.

(+30.6%) but also improves results for the remaining *semi-static and dynamic* (+17.6%) objects. This is because such objects do remain unmoved for some duration of the videos. This highlights the additional value derived from representing objects in 3D. Figure 7 visualises one example where the bin is successfully propagated using the *Fixed in 3D* baseline, including when out of view. The pretrained models struggle to propagate masks for the novel objects in the dataset or for masks that go out of the scene. These are cases that the *Fixed in 3D* baseline successfully handles. However, the fine-tuned models are quantitatively and qualitatively superior as they are trained on the dataset. No prior work has utilised 3D information along with learnt models for the task of semi-supervised VOS. We hope our novel benchmark can trigger new VOS approaches that tackle the combined challenge of keeping track of static objects in 3D and dynamic objects through trained propagation of objects during motion and transformations.

5 Conclusions

We introduced EPIC Fields, a dataset to study 3D video understanding. We addressed the difficult challenge of reconstructing cameras in EPIC-KITCHENS videos, introducing filtering and other techniques that are portable to other similar reconstruction scenarios. Using these pre-computed cameras facilitates working on 3D video understanding even without significant expertise in photogrammetry.

With EPIC Fields we also defined three benchmark tasks: dynamic new-view synthesis, unsupervised dynamic object segmentation, and video object segmentation. Our results show that the performance of state-of-the-art dynamic neural reconstruction/rendering methods strongly depends on the type of motion. In particular, the gap in reconstruction quality between the dynamic and the static parts of the videos show that there is ample margin for further improvements in the handling of dynamic objects. Similar findings apply to the segmentation of objects, where 3D-based models can assist unsupervised video object segmentation and propagate masks of static objects over time. We hope that these results, the proposed benchmark data and code (comprising evaluation, camera reconstruction, and baselines) will assist the community in investigating further methods that combine geometry and video understanding.

Societal impact. While we expect that our benchmark will lead to positive impact, including applications to augmented and mixed reality including AR assistants, there are potential negative impacts as well: better AR may be used for deception and many capabilities powering an assistant may also aid surveillance.

Acknowledgements Project supported by EPSRC Program Grant Visual AI EP/T028572/1. A. Darkhalil is supported by EPSRC DTP program. Z. Zhu is supported by UoB-CSC Scholarship. I. Laina and A. Vedaldi are supported by ERC-CoG UNION 101001212. D. Damen is supported by EPSRC Fellowship UMPIRE EP/T004991/1. D. Fouhey is supported by a NSF CAREER #2142529.

We thank Sanja Fidler and Amlan Kar, from the University of Toronto, for contributions to the initial idea and discussions of this project. We also thank Rhodri Guerrier for assisting in manual annotations of the test set.

References

[1] Stefan Becker, Ronny Hug, Wolfgang Hübner, and Michael Arens. An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. *arXiv preprint arXiv:1805.07663*, 2018. D.3

[2] Michael Broxton, John Flynn, Ryan S. Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul E. Debevec. Immersive light field video with a layered mesh representation. *ACM Trans. on Graphics (TOG)*, 2020. 2

[3] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[4] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 387–404. Springer, 2020. D.3

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[6] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 640–658. Springer, 2022. 4.3

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2022. 2, 3.1, F

[9] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2014. 2

[10] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 3.1, 4.1, 4.3, C, D.3

[11] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[12] Zhiwen Fan, Peihao Wang, Yifan Jiang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. NeRF-SOS: Any-view self-supervised object segmentation on complex scenes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 2

[13] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *Proceedings of SIGGRAPH*, 2022. 2

[14] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. 1, 2

[15] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[16] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 2, 4.1, 5, 6

[17] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, D.3

[18] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video, 2021. 3.2, F

[19] Jiaqi Guan, Ye Yuan, Kris M Kitani, and Nicholas Rhinehart. Generative hybrid representations for activity forecasting with no-regret learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[20] Xiang Guo, Guanying Chen, Yuchao Dai, Xiaoqing Ye, Jiadai Sun, Xiao Tan, and Errui Ding. Neural deformable voxel grid for fast optimization of dynamic view synthesis. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2022. 2

[21] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 2

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. C

[23] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2

[24] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J. Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas A. Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[25] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[26] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 2

[27] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with AIST++: Music conditioned 3d dance generation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[28] Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard A. Newcombe, and Zhaoyang Lv. Neural 3D video synthesis from multi-view video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 2, 2

[29] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

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

[31] Yiqing Liang, Eliot Laidlaw, Alexander Meyerowitz, Srinath Sridhar, and James Tompkin. Semantic attention flow fields for monocular dynamic scene decomposition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 2, 2

[32] Kai-En Lin, Lei Xiao, Feng Liu, Guowei Yang, and Ravi Ramamoorthi. Deep 3d mask volume for view synthesis of dynamic scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. E

[34] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *arXiv.cs*, abs/2205.15723, 2022. 2

[35] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2), 2004. 3.2

[36] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 4.1, 4.2, 5, 6

[37] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[38] Ashkan Mirzaei, Yash Kant, Jonathan Kelly, and Igor Gilitschenski. Laterf: Label and text driven object radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[39] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[40] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9226–9235, 2019. 4.3

[41] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[42] Hyun Soo Park, Jyh-Jing Hwang, Yedong Niu, and Jianbo Shi. Egocentric future localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[43] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. *CoRR*, abs/2011.12948, 2020. 1, 2

[44] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2, 2

[45] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. on Graphics (TOG)*, 40(6), 2021. 1, 2

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019. E

[47] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. *CoRR*, abs/2012.15838, 2020. 2

[48] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 4.3

[49] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 2

[50] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 2018. 3.2

[51] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *IEEE International Conference on Computer Vision Workshops*, 2021. 3.1, 2

[52] Sara Fridovich-Keil and Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[53] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3.2, 3.2

[54] Prafull Sharma, Ayush Tewari, Yilun Du, Sergey Zakharov, Rares Andrei Ambrus, Adrien Gaidon, William T. Freeman, Fredo Durand, Joshua B. Tenenbaum, and Vincent Sitzmann. Neural groundplans: Persistent neural scene representations from a single image. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 2

[55] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[56] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision (IJCV)*, 2010. 2

[57] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1

[58] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 1, 2

[59] Shuhan Tan, Tushar Nagarajan, and Kristen Grauman. Egodistill: Egocentric head motion distillation for efficient video understanding. *arXiv preprint arXiv:2301.02217*, 2023. 2

[60] Chengzhou Tang, Oliver Wang, and Ping Tan. Gslam: Initialization-robust monocular visual slam via global structure-from-motion. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2017. 3.2

[61] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 2

[62] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[63] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural Feature Fusion Fields: 3D distillation of self-supervised 2D image representation. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. 1, 2, 5, 6

[64] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. NeuralDiff: Segmenting 3D objects that move in egocentric videos. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2021. 2, 2, 4.1, 4.2, 4.2, E

[65] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. AIST dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *Proc. ISMIR*, 2019. 2

[66] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi S. M. Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. NeSF: Neural semantic fields for generalizable semantic segmentation of 3D scenes. *arXiv.cs*, abs/2111.13260, 2021. 1, 2

[67] Bing Wang, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022. 2

[68] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv.cs*, abs/2105.05994, 2021. 2

[69] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier PlenOctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[70] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^2 nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. *arXiv preprint arXiv:2205.15838*, 2022. 2

[71] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[72] Christopher Xie, Keunhong Park, Ricardo Martin-Brualla, and Matthew Brown. Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 962–971. IEEE, 2021. 2

[73] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas S. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *CoRR*, abs/1809.03327, 2018. E

[74] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[75] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 4.2, 6

[76] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. C

[77] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 2, 2

[78] Hong-Xing Yu, Leonidas Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 2

[79] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. STaR: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[80] Xiaoshuai Zhang, Abhijit Kundu, Thomas Funkhouser, Leonidas Guibas, Hao Su, and Kyle Genova. Nerflets: Local radiance fields for efficient structure-aware 3d scene representation from 2d supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[81] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1, 2

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We believe that our abstract and introduction accurately describe the work.

(b) Did you describe the limitations of your work? [Yes] We have discussed the limitations in the experiments. Indeed, one of the points of our paper is to show current limitations in existing work.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] We have discussed potential negative impacts in the conclusions of the paper.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A] There are no theoretical results in the paper.

   (b) Did you include complete proofs of all theoretical results? [N/A] There are no theoretical results in the paper.

3. If you ran experiments (e.g., for benchmarks)...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We include all all of the data and instructions needed. We will release all code for our benchmarks no later than the publication date.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide information about the data splits and hyperparameters in the main paper and supplement.

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Our experiments are too computationally expensive to run multiple times to produce error bars. Our models are meant as baselines that subsequent methods ought to show improvement on.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We provide a detailed documentation of the compute cost of each of the steps in the supplement. Some steps, such as the registration of all cameras, are costly. However, believe that by doing them and providing the information to the community, we can collectively save substantial overall compute time across the community.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] We build exclusively on the EPIC-KITCHENS 100 dataset. Our use of the dataset is made clear throughout the paper.

   (b) Did you mention the license of the assets? [Yes] When introducing EPIC-KITCHENS, we introduce its license. Our new assets will be released under the same license.

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Yes, we provide additional assets that will provide to the reviewers privately and then share publicly upon acceptance.

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] When introducing EPIC-KITCHENS, we mention that it was collected with ethics approval and with clear consent from the people who are in the data.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] When introducing EPIC-KITCHENS, we discuss that the data does not contain PII, was reviewed by participants prior to publication, and does not contain offensive content.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not do any crowdsourced annotation.

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] There are no participant risks for this paper. The dataset we use, however, was collected with ethics board approval.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not have any paid participants or workers.

In this supplementary material, we first describe the companion video that provides an overview of our dataset (Appendix A) and then detail how the data was released (Appendix B) along with taking stock of additional information specifically promised in the checklist (Appendix C). Next, we provide additional details on the dataset construction (Appendix D) and on the benchmarks (Appendix E). We devote a final section (Appendix F) to showing that the EPIC Fields pipeline could be applied to reconstructing videos from the Ego4D dataset.

A Supplementary video

We provide a short video in the form of a trailer at https://youtu.be/RcacE26e0bE. It allows to visually assess how challenging the reconstruction problem is and hints at how frame filtering helps. The video also illustrates how the new camera poses complement the existing semantic annotations for this dataset (hands and active objects), showcasing the potential of marrying 3D geometry and video understanding. Additionally, we provide a couple of qualitative results for static and dynamic novel view synthesis, one of the benchmark tasks we describe in the paper.

B Released data

Our dataset is now publicly available with visualisation scripts that enable exploring all the reconstructions and camera poses.

The data can be downloaded from http://epic-kitchens.github.io/epic-fields. We released the camera parameters along with sparse point clouds (light-weight version of 10–20MB/video) as well as the full COLMAP database of dense registrations (heavy-weight version). The latter enables comparisons with the dense registrations in EPIC Fields, and also allows the use of the COLMAP library and interface for visualisation and exploration.

The webpage also includes links to the visualisation code and to the code to replicate training, inference and evaluation for our benchmarks.

C Dataset and benchmark details mentioned in the checklist

Data splits. We provide information about the data splits used in the benchmark in Appendix D.3.

Annotations. We offer two additional sets of manual annotations, on top of those available in VISOR [10] to facilitate the assessment of the MG, NeuralDiff, T-NeRF+, and NeRF-W benchmarks. We employ these annotations as ground truth for evaluation on the UDOS task; nevertheless, we anticipate that they may prove valuable for various applications in future research endeavors.

The first set of annotations serves the dynamic objects. We provide human body annotations for all evaluation frames, as VISOR exclusively annotates the hands but not the other visible parts of the body. To achieve this, we identify up to 3 frames per video with visible body parts of the camera wearer. Using manual points, we employ SAM [22] to generate a total of 143 automated human-body annotations. These frames serve as reference frames for the DeAOT [76] model pre-trained on YT-VOS to propagate the masks across all evaluation frames.

The second set of annotations is dedicated to semi-static objects. VISOR primarily addresses active objects within specific segments of the video, whereas our method aims to evaluate semi-static objects that may have moved at any point during the video. To achieve this, we utilize a fine-tuned MS-DeAOT [76] on VISOR along with a maximum of 10 VISOR ground truth annotations as reference frames to extend the coverage of semi-static objects across all evaluation frames. As a result, all objects that have moved during the video are annotated by a mask, on every evaluation frame.

Hyperparameters. We provide information about the baselines used in our benchmark and their hyperparameters in Appendix E.

Total compute used. Estimating the precise computational budget of a multi-institution project of this scope is challenging. However, we report the actual computational time specifying the machine used in each case. All resources used were local. The main components of this project were:

- *Reconstruction:* As described in Appendix D.2, the reconstruction corresponds to a total of 2264 hours of compute, 1695 hours for the sparse reconstructions and 569 for registration.

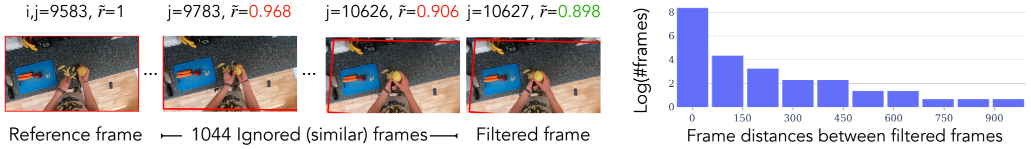


Figure 8: **Filtering frames before reconstruction.** We apply a 2D frame filtering technique to mitigate the oversampling of highly overlapping views (viewpoint distribution skews mentioned in Section 3.2 of the main paper) and to reduce the complexity of the SfM reconstruction. (Left) For a reference frame i, we show two of the ignored frames, the next frame after filtering, and their respective overlap r score with the original frame. Filtering discards 1044 frames (ca. 17 seconds) in this case. (Right) Histogram of the distances between frames after filtering (for one video).

This was parallelised across two machines with two GPUs each (two 11GB NVIDIA GeForce RTX 2080 Ti for the first machine, 12GB NVIDIA TITAN X and 11GB NVIDIA GeForce GTX 1080 Ti for the second machine).

- *NVS, UDOS Benchmarks:* We estimate that running the 3D baselines for D-NVS and UDOS benchmarks required 2400 GPU hours. Experiments on the D-NVS benchmark were carried out using several NVIDIA GPUs on a cluster, including P40, M40, V100, RTX8k and RTX6k. The training required up to 10GB of GPU memory for each experiment. The models for both benchmarks required a total of about 2400 GPU hours. We ran the experiments in parallel on 24 GPUs, resulting in a training time of 4.17 days. Both D-NVS and UDOS required each 50% of the total computation.
- *MG (UDOS Benchmark)*: We ran this baseline on a single 16GB V100 GPU. The total training time is about 5.5 days.
- *VOS Benchmark:* The *Fixed in 3D* baseline requires next-to-no compute — homography fitting on SIFT features is calculated during the reconstruction step. However, training STM and XMEM took 1.2 and 1.4 days respectively on a single 16GB V100 GPU.

We expect that by providing both sparse and dense reconstructions to the whole community, this effort will greatly reduce computation time for all the dataset users.

D Additional details on the dataset construction

D.1 Frame filtering

As discussed in Section 3.2 in the main paper, we downsampled videos to reduce the viewpoint skew that is common for ego-centric videos. The filtering discards on average 81.8% of all frames and allows the SfM pipeline to focus on more diverse views. Figure 8 visualises the filtering process using an example. The shown frame range contains many views that are similar to each other. The filtering discards 1044 redundant frames between frames $j = 9583$ and $j = 10627$. The figure also shows a histogram of distances between filtered frames.

D.2 Dataset statistics

How do we accept/reject a reconstruction? After producing the sparse reconstructions, we register all the frames; we then consider the videos with at least 70% dense registration rate. The histogram for both sparse and dense reconstructions is depicted in Figure 9. The majority of our reconstructions exhibit a dense registration rate exceeding 80%. In total, we successfully reconstructed 671 out of the 700 EPIC-KITCHENS videos, with average registration rates of 84.1% and 92.0% for the sparse and the dense reconstructions respectively. This is because we specifically select frames during transitions between kitchen hotspots for accurate reconstruction. This explains the higher registration rate for dense reconstructions.

Metrics for reconstruction quality. We use the common SfM metrics to assess the quality of the reconstructions. Figure 10 shows the histogram of the reprojection error of all the reconstructions. The average and maximum reprojection errors are 0.87px and 1.3px respectively. We use an image resolution of 456×256 to obtain the reconstructions and to calculate the reprojection errors.

How long does the reconstruction pipeline take? In Figure 11, for different video durations, we report the time required for the sparse reconstruction, for registration to obtain the dense reconstruction,

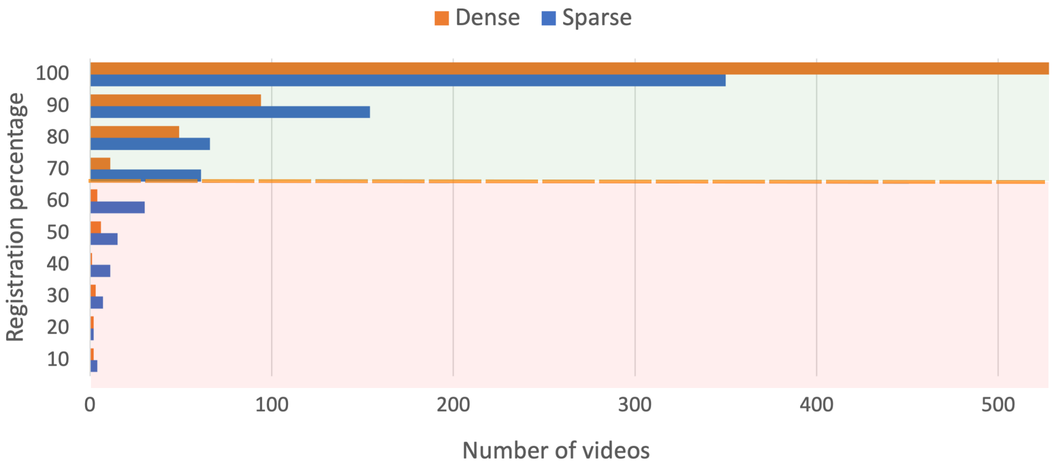


Figure 9: **Percentages of registered frames.** The dashed line specifies the threshold of the minimum dense registration rate to accept the reconstruction, otherwise, it would be considered a failure.

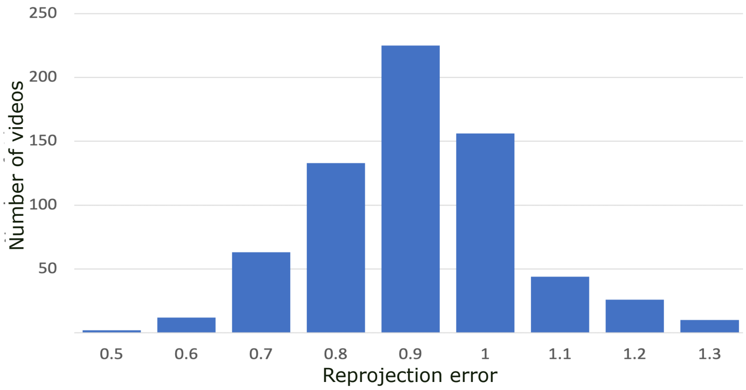


Figure 10: **Average reprojection error of EPIC Fields.** The majority of our reconstructions have an average reprojection error lower than 1.

and the total reconstruction time. As the length of the videos increases, the sparse reconstruction time follows a non-linear growth pattern. Overall, the sparse reconstruction and the registration processes took 1695 and 569 computation hours, respectively. We parallelise the pipeline on 2 machines with 2 GPUs each.

How large are these reconstructions? Figure 12 displays a histogram representing the number of 3D points in the sparse reconstructions, along with three example point clouds derived from reconstructions with varying numbers of points. These demonstrate the complexity of our reconstructions, which are capable of covering entire kitchens with fine-grained details. On average, each reconstruction consists of around 45,000 3D points.

Reasons for the reconstruction failures. While our reconstruction failure rate is only 4%, we examined the primary causes of these failures. These are mainly attributed to very short videos with large scene coverage, and challenging lighting conditions. (1) In the case of very short videos with large scene coverage, *e.g.*, a person just walking through the kitchen to retrieve one item and then walking out again, COLMAP often encounters difficulties due to the insufficient quantity of features and viewpoints. The median duration for the unsuccessful reconstructions is 1.5 minutes, compared to 6 minutes for the successful ones. This problem is exacerbated when the brief video captures a multitude of different locations within the kitchen, switching rapidly between these. (2) A couple of failure cases were linked to videos recorded under very low lighting, which led to a poor quality set of features to match. The average number of observed features per image for these unsuccessful videos was 198, compared to an average of 358 features per image for successful reconstructions.

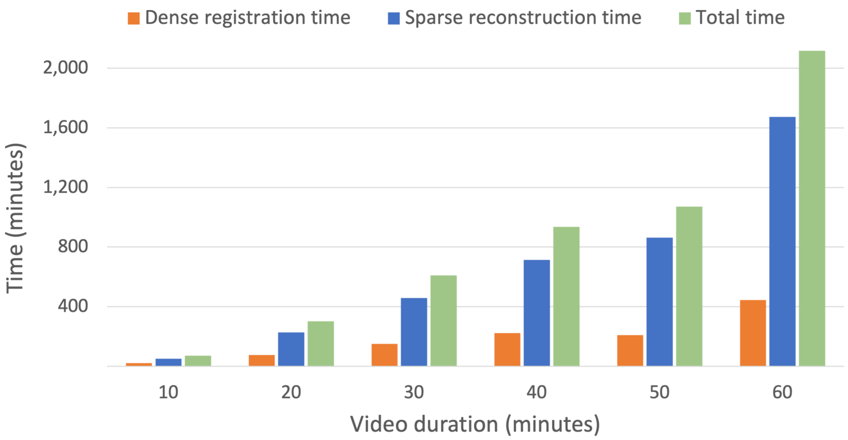


Figure 11: **Reconstruction time per video length.** We plot time for the sparse reconstruction (blue), registration time to obtain the dense camera poses (orange) and total reconstruction time (green) for different video durations. While the time for registration is almost linear, the reconstruction time increases non-linearly as a function of the video length, mainly because of bundle adjustment.

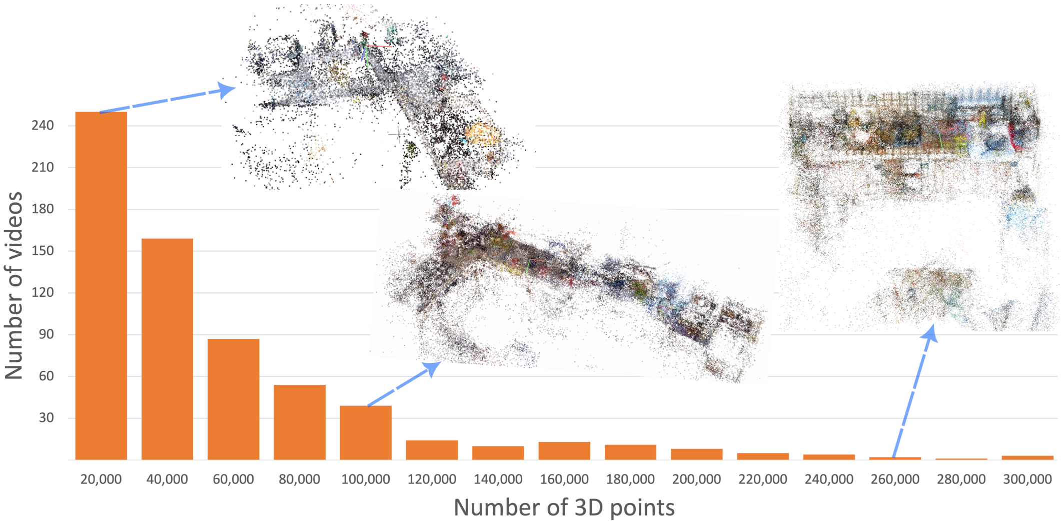
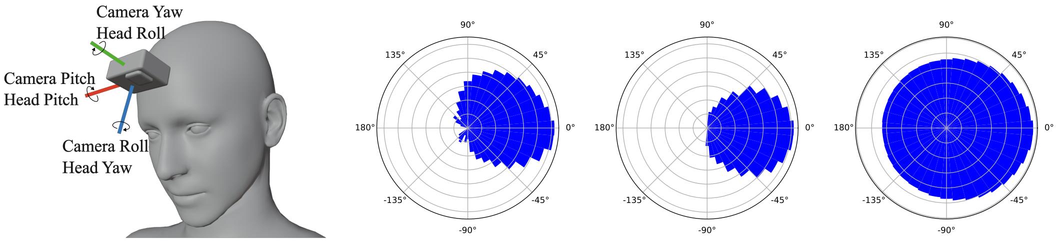


Figure 12: **Number of 3D points histogram.** The majority of our reconstructions generate fewer than 40,000 points that are enough to represent the kitchen. However, some reconstructions have more than 100,000 points, we include the point clouds for each points range showing the fine details covered by having more points.

Distribution of camera orientations. Figure 13 displays histograms representing the distribution of *relative* camera orientations of all EPIC Fields frames. Each frame uses the mean camera orientation within the video as reference. The histograms reveal that EPIC Fields contains diverse camera motions that are a result of natural head motions, such as looking up/down or tilting left/right. It is important to note the distinction between the camera orientations due to the particular camera mounting in EPIC-KITCHENS, illustrated in the figure. We thus particularly note camera motions and how they correspond to head motion given the specified mounting.

In summary, the figure shows larger head motion looking up (compared to the average camera orientation) than looking down, a balanced tilting as well as full 360 coverage of the kitchen by the body and/or head rotating in the scene.



(a) The correspondence of Yaw/Pitch/Roll between the camera and the head.

(b) Camera Pitch / Head Pitch / Head Raising Up-or-Down

(c) Camera Yaw / Head Roll / Head Tilting Left-or-Right

(d) Camera Roll / Head Yaw / Neck Turning

Figure 13: Camera mounting arrangement (a) and the log-scale polar histogram of the three camera orientation parameters in the dataset (b-d).

D.3 Statistics of the benchmark splits

We provide statistics of the splits for the D-NVS task of our benchmark in Table 6. In UDOS, the objective is to segment dynamic and semi-static objects in videos without relying on supervision from ground-truth segmentations during training. Thus, *all* frames are observed during training. Evaluation frames are the same as for the D-NVS task. For the VOS task, we use the train/val splits published as part of the VISOR VOS benchmark (See [10] Sec. 5.1).

For D-NVS, we divide the evaluation frames for each video equally between the validation and test sets, taking every other frame from both *In-Action* and *Out-of-Action* frames. Each video contains evaluation frames spanning all difficulty tiers (easy, medium, hard). The size of the validation and test sets corresponds to only a fraction of the number of training frames due to strict constraints on the sampling of evaluation frames, which include high variability in viewpoints and a minimum time gap between the training and test/validation frames as described in Section 4.1 of the main paper.

For the *Hard (In-Action)* and *Medium (Out-of-Action)* settings, this time gap is set to 1s, which introduces increased difficulty for rendering novel views, since a significant portion of an activity might have taken place and neural rendering approaches would have to interpolate motion to account for this. While this is indeed a challenging task, it provides a unique opportunity for further explorations in neural rendering. We can account for the ambiguity that this choice introduces in two ways: resort to an evaluation protocol that accounts for that (e.g., best-of-K prediction) or accept that pixel predictions will have to be approximate for dynamic pixels and still measure the PSNR score. While the latter is not perfect, it is still reasonable for most 1s gaps and is much simpler than alternatives. The preference for this choice is also common in other ambiguous prediction tasks; for example, in the GTA-IM benchmark, where 3D path error is estimated after 0.5, 1, 1.5, and 2s [4], the TrajNet benchmark, where prediction is estimated for 4.8s from the observed frame [1], and the future hand prediction task in Ego4D, which uses a time gap of 1.5s from the observed frame [17].

For the *Easy (Out-of-Action)* setting, there is no temporal gap between training and evaluation frames and no specific action taking place. Consequently, both the complexity for rendering novel views and the ambiguity in evaluation are reduced for this subset of frames. This simplified setup parallels existing NVS benchmarks.

E Additional training details for benchmarks

We now provide precise hyperparameters for the baselines used in the NVS and UDOS benchmarks. We provide full code for reproducing these results with the publication.

NeRF-W, T-NeRF+, NeuralDiff. We base our implementation of all 3D baselines on the codebase from NeuralDiff [64] and merge the other two approaches into the same PyTorch [46] codebase to align all training and evaluation details between models. We use the same training setup as in NeuralDiff, which involves training one model per baseline on each scene, taking approximately 12 hours using one NVIDIA Tesla P40 per experiment. Furthermore, the models are trained with hierarchical sampling (with a coarse and fine model as in the typical NeRF setting) and with a batch

Table 6: **EPIC Fields splits statistics**. We summarise the frame count and average frames per video for each split and for different difficulties (Easy, Medium, Hard). The number of frames for the validation and test sets is only a fraction of the training frames. This is due to strict constraints on the sampling of evaluation frames such as a high variety of viewpoints and the minimum time frame between train and test/validation frames. The train frames are fixed, regardless of the difficulty level.

	In-Action (Easy)		Out-of-Action (Medium)		Out-of-Action (Hard)		Total	
	#frames	average	#frames	average	#frames	average	#frames	average
Train	—	—	—	—	—	—	103,571	2071.42
Val	3,448	68.96	657	13.14	305	6.1	4,410	88.2
Test	3,461	69.22	695	13.9	289	5.78	4,445	88.9

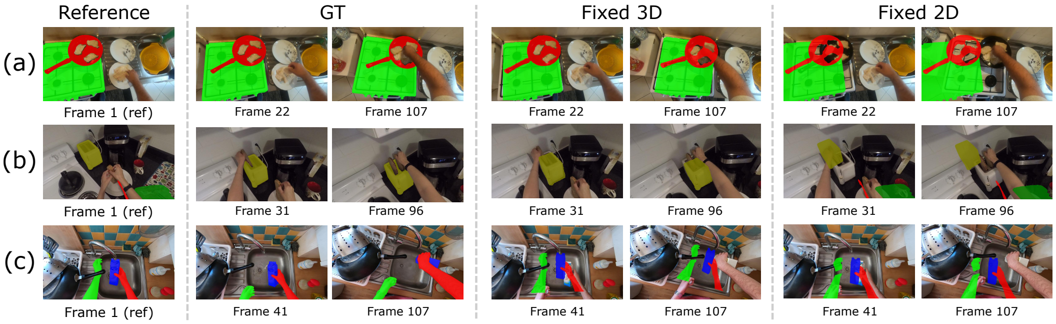


Figure 14: **Qualitative results for Semi-Supervised VOS.** We show samples with multiple objects. In scenarios (a) and (b), the *Fixed 3D* baseline effectively handles static objects, whereas the *Fixed 2D* falters due to camera movement. Conversely, in scenario (c), both strategies prove unsuccessful as the objects are in motion, invalidating the presumption of fixed objects in either 3D (*Fixed 3D*) or 2D (*Fixed 2D*), hence their failure.

size of 1024. We train with the Adam optimizer for 10 epochs and an initial learning rate of 5×10^{-4} that is adjusted during the training with a cosine annealing schedule.

MG. We use the provided code and train the model on our training split frames, jointly, for 135k iterations with a batch size of 32 and a learning rate of 5×10^{-4}.

STM and XMEM. For STM, we finetune a pretrained COCO [33] model on VISOR for 400K iterations with a batch size of 32 and a learning rate of 1×10^{-5}. For XMEM, we use the pretrained YoutubeVOS [73] model published in the XMEM paper and finetune it on VISOR for 100K iterations, with a batch size of 16 and a decaying learning rate initialised with 1×10^{-5}.

In the main paper, we include some qualitative results for the VOS challenge for a single object. We add more examples showing multi-object segmentation in Figure 14. The figure shows samples of failures of *Fixed 2D* in scenarios (a) and (b) and a case when both *Fixed 2D* and *Fixed 3D* fail to segment the dynamic objects (c).

F EPIC Fields pipeline for Ego4D videos

While our reconstruction pipeline addresses several difficulties that are inherent to the videos of EPIC-KITCHENS [8], we can also apply it to other ego-centric videos such as the ones from Ego4D [18]. Using the pipeline as is, we can estimate camera poses for Ego4D videos that are about cooking and construction/building. We showcase this through an example in Figure 15 and two videos of reconstructions and camera tracks:

- Task: Construction —– 35 minutes of decorating and refurbishment. The video at `https://youtu.be/EZlayZIwNgQ` contains situations of challenging camera pose estimation including the camera wearer on a ladder (01:29, 05:07), kneeling down (16:14), as well as

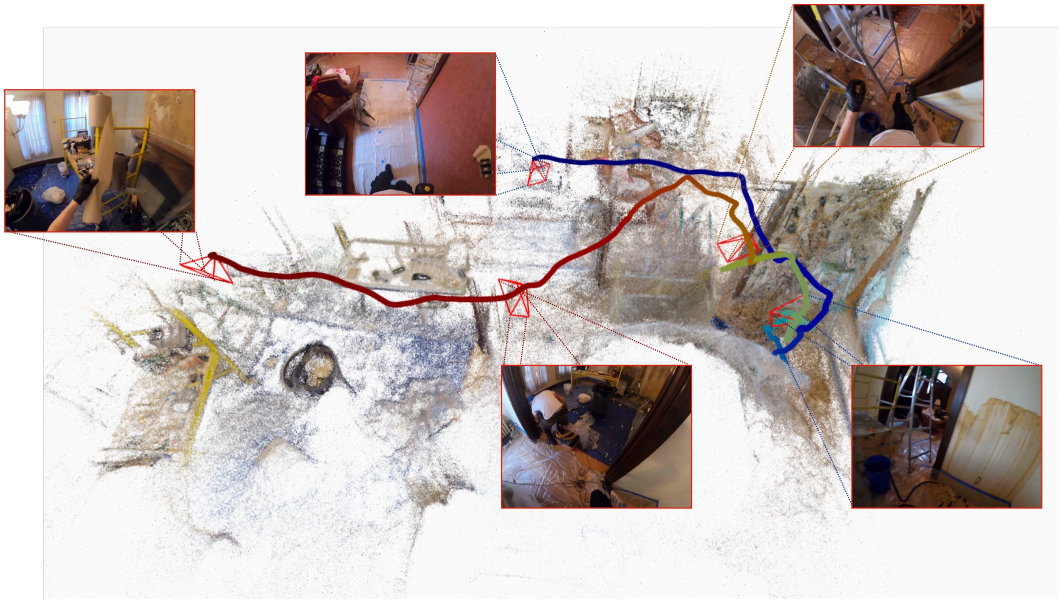


Figure 15: Visualisation of the 3D reconstruction for one video of the Ego4D dataset capturing building and refurbishment activities, with camera estimated using the EPIC Fields pipeline

drinking and navigating the scene (27:25) amongst many interesting poses. (Ego4D video a2dd8a8f-835f-4068-be78-99d38ad99625, source: CMU US)

- Task: Cooking —- 10 minutes. The corresponding video can be found at https://youtu.be/GfBsLnZoFGs (Ego4D video 18f5c2be-cb79-46fa-8ff1-e03b7e26c986).