

EFM3D: A Benchmark for Measuring Progress Towards 3D Egocentric Foundation Models

Julian Straub*, Daniel DeTone**, Tianwei Shen**, Nan Yang**,
Chris Sweeney, Richard Newcombe

Meta Reality Labs Research

Abstract. The advent of wearable computers enables a new source of context for AI that is embedded in egocentric sensor data. This new egocentric data comes equipped with fine-grained 3D location information and thus presents the opportunity for a novel class of spatial foundation models that are rooted in 3D space. To measure progress on what we term Egocentric Foundation Models (EFMs) we establish EFM3D, a benchmark with two core 3D egocentric perception tasks. EFM3D is the first benchmark for 3D object detection and surface regression on high quality annotated egocentric data of Project Aria. We propose Egocentric Voxel Lifting (EVL), a baseline for 3D EFMs. EVL leverages all available egocentric modalities and inherits foundational capabilities from 2D foundation models. This model, trained on a large simulated dataset, outperforms existing methods on the EFM3D benchmark.

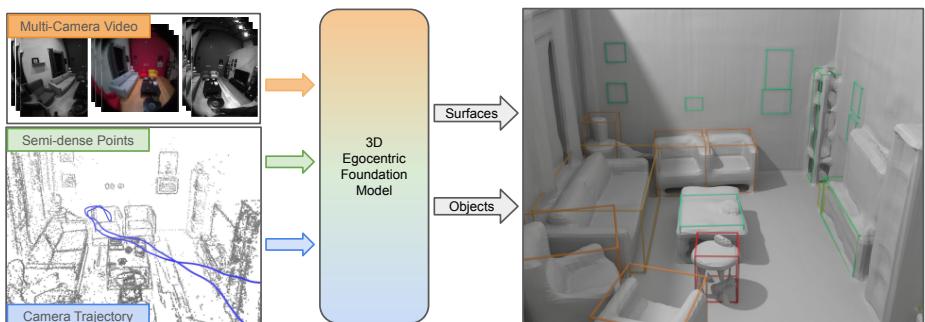


Fig. 1: 3D Egocentric Foundation Models leverage spatial priors from egocentric data to power core 3D tasks such as 3D object detection and reconstruction.

1 Introduction

Foundation models trained on Internet-scale text, image and video datasets have demonstrated the potential in using large-scale self-supervised learning approaches to build backbones that are useful for numerous downstream tasks,

* Project lead.

** Equal contribution in alphabetic order.

through both fine-tuning and zero-shot learning. The advent of wearable spatial computers enables a new source of context from egocentric sensor data. Key to unlocking this context is understanding the environment of the wearer. This new egocentric data source comes equipped with fine-grained 3D location information [13, 19] and thus presents the opportunity for a novel class of spatial foundation models that are rooted in 3D space. This class of 3D egocentric foundation models (EFMs) can leverage strong priors from egocentric data like camera poses, calibration and semi-dense point information. As with 2D image foundation models [1, 22, 28, 42], the availability of large amounts of data is critical for training such models and high quality annotations to measuring their performance. While there are now significant amounts of 2D data with 2D annotations [12, 20, 34], and a large body of 3D scene dataset [3, 9, 50, 58], and autonomous vehicles (AV) datasets [6, 16], the equivalent for egocentric data from wearable devices with 3D annotations is only just starting to become available [13, 19, 53].

To enable measuring progress towards EFMs we propose the EFM3D benchmark which contains two tasks: 3D bounding box detection and surface regression. To set up the first baseline model on EFM3D, we introduce the Egocentric Voxel Lifting (EVL) model which relies on frozen 2D foundation features to set a competitive performance. EVL leverages all egocentric modalities from Aria including posed and calibrated RGB and greyscale video streams and semidense points. When trained on our new large-scale simulated dataset EVL generalizes well to the real-world EFM3D benchmark and significantly outperforms the current state-of-the-art 3D scene understanding models even when they are retrained on the same dataset. To summarize our contributions:

Dataset. We release more annotations including 3D object bounding boxes (OBBs) and groundtruth (GT) meshes built on top of Aria Synthetic Environments dataset [53] and real Project Aria sequences to enable research on the two foundational tasks of 3D object detection and surface reconstruction.

Benchmark. We set up the EFM3D benchmark with the first two tasks, namely 3D object detection and surface reconstruction, to advance continuous research in the areas of egocentric machine perception.

Method. We introduce a baseline model, EVL, to solve both tasks at the state-of-the-art level by leveraging explicit volumetric representation, a full suite of egocentric signals, and 2D features from vision foundation models.

2 Related Work

We categorize the related work into datasets, 3D foundation models, object detection, and surface reconstruction methods using video and point cloud inputs.

Datasets. As shown in Table 1, egocentric datasets [10, 18, 19, 29, 41] have been typically designed to enable activity recognition, video narration and 2D understanding tasks. The most recent egocentric datasets, through the use of Project Aria [13], are starting to enable 3D perception by releasing accurate camera cali-

	sim/real	motion	video	points	poses	# scenes	# sequences	#3D OBBs	surface
Ego4D [18]	real	egocentric	✓			74	3k hrs	✗	✗
Ego-Exo4D [19]	real	egocentric	✓	✓	✓	131	5,625	✗	✗
AEA [36]	real	egocentric	✓	✓	✓	5	143	✗	✗
ScanNet [9]	real	scanning	✓		✓	707	1,513	36k	fused depth
ARKITScenes [3]	real	scanning	✓		✓	1,661	5,048	52k	fused depth
Hypersim [47]	sim	random	✓		✓	461	774	54k	CAD mesh
ASE [53]	sim	egocentric	✓	✓	✓	100k	100k	3M	CAD mesh
ADT [43]	real	egocentric	✓	✓	✓	1	6	281	CAD mesh
AEO	real	egocentric	✓	✓	✓	20	26	584	✗

Table 1: Related datasets either are egocentric or have 3D OBB and surface annotations. We fill this gap and contribute 3D OBB annotations for ASE, CAD meshes for a validation subset of ASE and ADT as well as AEO—an egocentric validation dataset with high quality 3D OBB annotations.

brations, pose and semi-dense points information. Examples are Ego-Exo4D [19], which also contains localized third-person perspective information, the Aria Everyday Activities (AEA) [36] dataset as well as the Aria Digital Twin (ADT) [43] dataset. Only the ADT dataset contains 3D OBBs and CAD mesh GT annotations for real-world Project Aria data. However all data is recorded in a single space, making diversity too low for OBB tasks. Thus far 3D object detection and surface reconstruction datasets used by the community are either real RGB-D scanning sequences of indoor spaces (e.g. ScanNet [2, 9], ScanNet++ [58], and ARKitScenes [3]) or simulations (e.g. Hypersim [47], InterioNet [32], and 3D-Front [14]). While these RGB-D datasets have sufficient scale for training, there is substantial modality difference from RGB-D datasets to egocentric data from Project Aria. Project Aria provides one high-resolution RGB and two grey-scale video streams whereas RGB-D datasets come with just a single RGB stream. RGB-D datasets contain dense, uniformly sampled depth and surface information whereas egocentric data from Project Aria comes only with sparse depth via semi-dense point clouds from the SLAM system. Additionally, the motion in scanning datasets is different than egocentric motion because the aim is to “cover” and observe all surfaces of the scene, which is not the case in typical egocentric data. We find these differences in modalities and motion patterns are challenging and necessitate substantial differences in model design which opens up new research. To close these gaps we provide new object and mesh annotations for existing Project Aria datasets [43, 53] and a small high-quality object detection benchmark.

3D Foundation Models. Likely hindered by the availability of sufficient 3D training data, there are few related 3D foundation models and none designed for egocentric data. 3D-LFM [8] generates 3D skeletons from 2D point inputs for more than 30 categories of 2D-3D point sets. Most relevant is Ponder [23, 61] which uses a point-cloud encoder on unprojected RGB-D frames from ScanNet to produce a dense feature volume that is trained via a NERF [37] rendering loss. The architecture shares the lifting into a feature volume [21] but does so via a sparse point encoder instead of leveraging 2D foundation models and dense lifting. As we find in our experiments 3D point-encoder-based methods like 3DETR [38] struggle with the non-uniform egocentric point clouds.

3D Object Detection. The most similar method to the proposed EVL model is ImVoxelNet [48]. After unprojecting 2D image features into a 3D feature volume, a 3D CNN head regresses 3D bounding box parameters at multi-scales. Similarly, Raytran [54] lifts 2D features into a 3D grid via a sparse attention mechanism to detect 3D OBBs. Neither use additional semi-dense point information. In the AV community 3D OBB detection is also investigated based on video and LiDAR points [7, 59]. LiDAR points sparsely but uniformly provide depth input in contrast to egocentric semidense points. The OBB distribution is also more constrained due to the fact that the car and objects move essentially in the same, locally-2D plane. DETR3D [55] and extensions such as PARQ [56] explore how to directly perform 3D OBB detection from multi-view image input. Cube R-CNN [5] is following the Mask R-CNN-paradigm and directly regresses 3D OBBs from single frames. Similarly, MOLTR [31] and ODAM [30] detect 3D OBBs per frame and additionally learn to track and fuse 3D OBBs. Point-cloud based methods for 3D object detection such as VoteNet [45] and more recent Transformer-based 3DETR [38], typically leverage either dense depth or fused RGB-D reconstructions as inputs. We find that the sparse, non-uniform nature of egocentric point clouds is challenging for this class of methods.

3D Surface Estimation Aside from the vast amount of traditional stereo methods [15, 17], learning-based surface reconstruction can be categorized into volumetric methods, depth-based methods, and scene-specific methods [25, 37]. Among depth-based approaches, the premise of monocular depth estimation is to learn strong priors about scenes and environments to regress depth without any multi-view information. ZoeDepth [4] is a high-performing monocular depth estimator that is first trained on relative depth and then fine-tuned on datasets with metric depth. Typically mono-depth methods have problems with the scale ambiguity of natural images. ConsistentDepth [26] addresses this issue by leveraging the global sparse points to achieve temporally consistent depths. With multi-view inputs, MVSNet [57] builds a 3D cost volume via differentiable homography warping to regress depths. At the boundary between multi-view methods and mono-depth methods that infer depth from multiple views is SimpleRecon [49], which constructs a frustum cost volume but then squashes it into a 2D feature map for depth regression. ATLAS [39] proposes the idea of unprojecting 2D image features into a 3D feature grid for semantic surface regression. and Lift-Splat-Shoot [44] in the AV literature concurrently uses a similar idea to lift to a bird-eye-view (BEV) grid. NeuralRecon [51] expands on this idea by first regressing local sparse TSDF and then fusing local reconstructions into a global volume using a learned fusion module. As such it is the most similar method to the proposed EVL model. None of these methods is designed for egocentric data modalities nor has been evaluated on egocentric data.

3 Dataset Contributions

To facilitate novel research in 3D Egocentric Foundation Models, we release new data for both training and evaluation, spanning both synthetic and real egocentric Aria data. We summarize the three types of this new data below.

Large-scale Synthetic OBB Metadata. We release the 3D oriented bounding boxes (OBBs) corresponding to the ASE dataset. This consists of approximately 3 million OBBs across 43 object classes. This includes visibility metadata for ASE which enables image-based detector training and evaluation. We define the visibility of an OBB on a certain image based on its *observability* and *occlusion*.

Small-scale Real world OBB Dataset. We introduce a small real-world egocentric validation dataset, termed Aria Everyday Objects (AEO). It has high quality 3D OBB annotations to allow evaluation on real Project Aria sequences. One key aspect of this dataset is that the sequences were collected by non computer vision experts, without any specific guidance to scan the scene. This ensures realistic egocentric motion. The annotations are made by human annotators that use the semi-dense point depths, camera poses, calibrated multi-camera rig to label the 3D OBBs in a 3D viewer. The dataset contains 26 diverse scenes with 584 OBB instances across 9 classes: Chair, Table, Sofa, Bed, WallArt, Lamp, Plant, Window and Mirror. These semantic classes overlap with the ASE dataset and enable testing of sim-to-real generalization.

Ground-truth Meshes. For 3D reconstruction training the ASE dataset already contains groundtruth depth images. We release the 3D groundtruth mesh for the simulated ASE validation dataset as well as for the real-world ADT dataset to enable benchmarking 3D reconstruction methods.

4 Egocentric Voxel Lifting (EVL)

We design EVL as a universal 3D backbone that lifts 2D foundation features from posed and calibrated video streams into a 3D gravity-aligned voxel grid of features. Before processing this 3D feature volume with a 3D U-Net we concatenate masks derived from semi-dense points that indicate surface points as well as freespace. As shown in Fig. 2, on this feature volume we can run different kinds of 3D-CNN heads to regress desired 3D quantities such as 3D OBB parameters and occupancy values.

2D-3D Lifting Encoder We run a 2D foundation model [42] with frozen weights on T posed input images from each video stream. The resulting 2D foundation features are up-sampled to the full input image resolution using stacked 2D CNN & upsample layers. We instantiate a local, gravity-aligned 3D voxel grid in front of the most recent Aria camera rig. We choose the last RGB camera pose and gravity align its rotation to obtain the anchor pose of the voxel grid. Using known camera poses and calibrations, we project the voxel grid centers into each image and sample the 2D feature maps using bi-linear interpolation while respecting the valid radius of the distorted camera models. This yields a feature volume of $T \times F \times D \times H \times W$ for every stream. We aggregate features

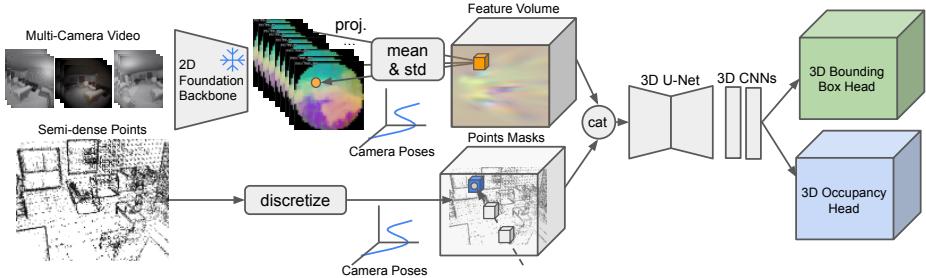


Fig. 2: EVL lifts 2D features extracted from frozen image foundation models into an local gravity-aligned 3D voxel grid of features. After concatenating point masks a U-Net processes the 3D feature volume before running 3D CNN heads. These heads regress target parameters such as 3D OBBs and occupancy values.

across all streams and the time dimension using mean and standard deviation to obtain a lifted feature volume of shape $2F \times D \times H \times W$. The metric size and resolution of the voxel grid can be defined freely. Note that this lifting formulation is camera model agnostic and supports online-calibrated video streams from Aria. Please refer to the supplemental for the experiments of EVL using different camera models.

Semi-Dense Point Encoder. We additionally input the semi-dense 3D point cloud provided by the Aria research toolkit [13] to provide a geometric prior for the lifting backbone. Together with the visibility information from where each point was observed this point cloud contains not only information about the surfaces in the scene but also the freespace. To provide this information to EVL we compute a point and a free-space mask tensor to concatenate to the lifted feature volume. The point mask indicates where there are surface points. It is computed by discretizing points into a binary mask sized $D \times H \times W$ corresponding to the voxel grid resolution. The freespace mask indicates the voxel locations between the camera centers to the known surface points. We compute it by sampling S points along each ray within the bounds of the feature volume. We concatenate both masks to the lifted 3D features resulting in a 3D feature volume shaped $(F + 2) \times D \times H \times W$.

3D U-Net. Once the 3D feature volume is instantiated, we run a 3D U-Net to allow 3D processing. The U-Net reduces the spatial resolution by a factor of 8x before upsampling back to the full 3D resolution. See a more detailed model diagram in the supplemental.

4.1 3D Bounding Box Detection

We assume all object bounding boxes are gravity-aligned. We take inspiration from the ImVoxelNet [48] to build a proposal-free and anchor-free 3D detection head. This head runs on top of the output of EVL and regresses the class distribution (v^{cls}) and eight values defining the geometric properties of a 3D bounding box. It first predicts a centerness score of $[0, 1]$ for each voxel v . The centerness

(v^c) of each voxel defines the probability of containing any object bounding box center. Then the head regresses the seven parameters (v^{obb}) that describe the bounding box: three for the height, width and depth of the box, three that predict a small offset for the bounding box center to account for discretization of the voxel grid, and lastly a single yaw rotation parameter that determines the rotation of the bounding box around the gravity direction. The predictions are then filtered with Non-Maximum Suppression (NMS) using 3D Intersection over Union (IoU) and the centerness confidence score. There are three losses used for training, one for each head. The centerness and the classifier heads are trained with focal loss (FL) [33], and the parameters head is trained using gravity oriented 3D IoU loss. With N_v being the number of voxels:

$$L_{obj} = \frac{1}{N_v} \sum_n^{N_v} w_c \text{FL}(v_n^c, \widehat{v_n^c}) + w_{iou} \text{IoU}(v_n^{obb}, \widehat{v_n^{obb}}) + w_{cls} \text{FL}(v_n^{cls}, \widehat{v_n^{cls}}), \quad (1)$$

4.2 3D Surface Regression

To regress 3D surfaces in the local gravity-aligned voxel grid we regress an occupancy value for every voxel. We supervise occupancy using the GT depth maps. For each GT depth value we obtain three kinds of occupancy supervisions: one surface point, one free-space point, and one occupied point. We compute the surface point p_{surf} by unprojecting the GT depth values using the camera calibrations. The free-space point p_{free} is sampled in front of the surface, while the occupied point p_{occ} is sampled behind the surface. Both p_{free} and p_{occ} are randomly sampled up to δ distance from the surface, which we choose as the length of a voxel. At these locations we sample into the predicted occupancy grid via tri-linear interpolation to obtain predicted values for all three kinds of points. The final loss is then computed via the focal loss [33] by defining the ground-truth probability of free-space point, surface point, and occupied point being 0.0, 0.5, and 1.0, respectively:

$$L_{surf} = \frac{1}{N} \sum_n^N \text{FL}(p_{free}^n, 0.0) + \text{FL}(p_{surf}^n, 0.5) + \text{FL}(p_{occ}^n, 1.0), \quad (2)$$

where $\text{FL}(\cdot)$ is the focal loss and N is the number of unprojected points from the GT depth maps. Note that we use focal loss to make the optimization to focus less on the easy samples like flat surfaces while more on the hard samples like sharp object boundaries. We further add a total variation (TV) regularization loss on the predicted occupancy volume to encourage local smoothness.

4.3 Implementation and Training Details

Input Snippet Data Preparation. We sample 1s snippets from the RGB and greyscale video streams of Aria running at 10Hz. The input resolution for RGB and greyscale videos are $240 \times 240 \times 3$ and $320 \times 240 \times 1$, which has approximately the same angular resolution per pixel. We leave exploring other image sampling strategies like key-framing [56] for future research. For the GT OBB labels, we use the OBBs which are visible inside the snippet and discretize the centers into

the volume to obtain the centerness supervision signal. We further filter out the OBBs which are heavily occluded in the snippet as described in Sec. 3.

2D Foundation Model Encoder. While we could use other 2D foundation models like CLIP [46] to encode the video streams, we find the base DinoV2.5 [11, 42] model works well for our tasks and is reasonably efficient.

Feature Volume We choose a $4m^3$ voxel grid. We adapt the grid resolution to the task. For 3D object detection we find a $6.25cm$ voxel size ($D = H = W = 64$) is sufficient and for surface regression we push to $4cm$ ($D = H = W = 96$) to be comparable with related work [51].

Hyperparameters. The 3D Bounding Box Detection head is trained on ASE 10k sequences (with 526k snippets) for 10 epochs and the 3D Surface Regression head is trained on the ASE 1k sequences (with 54k snippets). Both heads were trained with a base learning rate of $2e^{-4}$ using Adam [27] optimizer with cosine annealing scheduler. We set $w_{cent} = 100$, $w_{iou} = 10$, and $w_{class} = 1.0$ for training 3D bounding box detection. For the inference, we use the centerness threshold of 0.2 to select the candidate detections and performs simple 3D NMS with radius of 2 voxels based on the 3D IoU. The focal loss parameters in the 3D Surface training is set to $\alpha = 0.25$ and $\gamma = 2.0$.

5 The EFM3D Benchmark and Experiments

In this section we describe the EFM3D benchmark, which consists of two egocentric tasks: 3D oriented bounding box detection and 3D surface regression. For the object detection task, we use the Aria Everyday Objects (AOE) dataset. For the surface estimation task, we use the Aria Digital Twin (ADT) dataset. To demonstrate the value of this benchmark, we evaluate the proposed EVL model together with state-of-the-art-approaches on the 3D OBB detection and the 3D surface reconstruction tasks. These tasks on egocentric data can leverage the video streams, poses and camera calibration as well as semi-dense point outputs extracted from the video streams via a SLAM [13] system. For both tasks we train on the large-scale simulated ASE dataset and evaluate performance both in domain on ASE as well as on real sequences with ground-truth annotations in ADT and AEO recorded with Project Aria.

Field-of-view matters. While EVL uses a non-linear camera model to leverage the full field of view (FoV) of the Aria cameras, various related methods [5, 48, 49, 51] rely on linear cameras. For the off-the-shelf (OTS) models trained on ScanNet, we undistort the images into the ScanNet intrinsics in order to provide the data in the format the models were trained with. For other OTS models [4, 26] that are not constrained by camera models, we provide the largest possible linear camera model (max-linear). As can be seen in Fig. 3, there is a substantial difference in FoV between the two linear models. The max-linear model covers almost the same FoV as the distorted camera. As we show in the following experiments, larger FoV provides better coverage of the scene thus better performance.

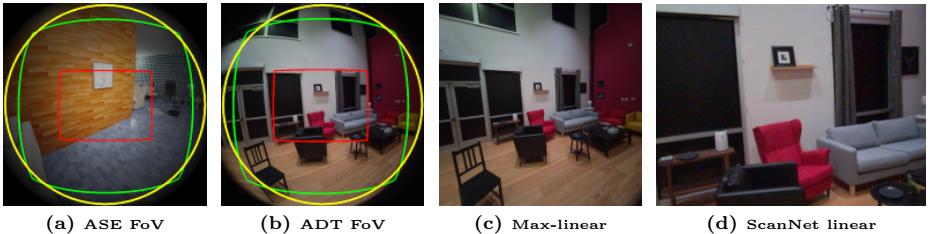


Fig. 3: We visualize (c) maximum linear (region in green) and (d) ScanNet linear (region in red), as well as the original fisheye (with a valid region in yellow) camera models for (a) ASE and (d) ADT.

Persisting 3D Predictions. Inferring the 3D scene state from individual frame or snippet-level 3D inferences is not only useful for downstream systems but also enables comparing different algorithms and measuring how consistent these 3D inferences are across the full sequence. In the following experiments, we use tracking and fusion systems for both 3D OBBs as well as surface predictions to enable accumulation and improvement of 3D OBBs and 3D surfaces reconstructions. Predicted 3D OBBs are associated to the scene OBBs using a set of metrics including 3D intersection over union (IoU) and then fused via a running average over OBB parameters. For 3D reconstruction we use the implementation from [60] with minor improvements to fuse depth maps into a truncated signed distance field (TSDF) [40]. The occupancy volume predicted by EVL is fused using a per voxel running average in a global grid. The meshes are extracted using the marching cubes algorithm [35] at the iso-level of 0 for TSDF fusion and 0.5 for occupancy fusion. See supplemental for more details.

5.1 3D Bounding Box Detection and Persistence

Broadly speaking there are three different types of ML architectures that can run on these modalities: 2D CNNs running on individual frames [5], 3D CNNs running on 3D geometric quantities [48] and Transformers that compute on the point cloud [38] or snippets [56]. For Cube R-CNN [5], we rectify the images using the max-linear camera setting. For ImVoxelNet [48] we show performance of the base configuration following the training recipe for OBB training. We provide max-linear rectified RGB snippets. We freeze the ResNet50 image encoder to help sim-to-real transfer. 3DETR [38] is trained on the per snippet visible point cloud using the author’s OBB training recipe and the random point-cloud crop augmentation from the paper. All models are trained on 10k sequences of ASE containing 600k snippets for 10 epochs except for the transformer-based models which are trained for 50 epochs, as we found its convergence to be slow. We report the mean Average Precision (mAP) averaged over IoU thresholds in the range $[0.0, 0.05, \dots, 0.5]$ and semantic classes following Cube R-CNN [5].

We compare a representative set of related approaches that cover different subsets of modalities and architecture types in Tab. 2 quantitatively. Across all

	Train Set	Modality	Decoder	ASE mAP Snippet	ASE mAP Sequence	AEO mAP Sequence
Cube R-CNN	OTS	frame	2D CNN	0.01	0.02	0.05
Cube R-CNN	ASE	frame	2D CNN	0.21	0.36	0.08
ImVoxelNet	ASE	snippet	3D CNN	0.30	0.64	0.15
3DETR	ASE	pts	Transformer	0.24	0.33	0.16
EVL (ours)	ASE	snip+pts	3D CNN	0.40	0.75	0.22

Table 2: The OTS Cube R-CNN model does not generalize well to egocentric data. Four models trained on ASE training data are evaluated on the simulated ASE validation dataset as well as real-world AEO dataset.



Fig. 4: ASE validation scenes overlaid with camera trajectories and the 3D OBB predictions from EVL and ImVoxelNet [48].

methods the OBB tracker increases the mAP—in most cases by almost a factor of $2\times$ —when comparing the snippet to the sequence-level mAP on ASE. For the sequence results we perform a search over tracker instantiation thresholds for each model and report the best mAP. Interestingly, when comparing performance on the challenging AEO real-world dataset, the image based models suffer the strongest from the sim-to-real gap: Cube R-CNN (-32 mAP), ImVoxelNet (-49 mAP) and EVL (-48 mAP). The points only method 3DETR has a smaller drop in mAP (-17 mAP). We attribute this to the fact that point-clouds are lower dimensional data than video and are less sensitive to photorealistic rendering artifacts in synthetic renderers. EVL performs best in both synthetic and real data scenarios. We note that although the Cube R-CNN OTS (off-the-shelf) model is trained a diverse dataset it does not generalize well to egocentric data.

We show overhead views of the sequence level OBB prediction results on held out synthetic scenes for the GT, EVL and ImVoxelNet in Fig. 4. Both

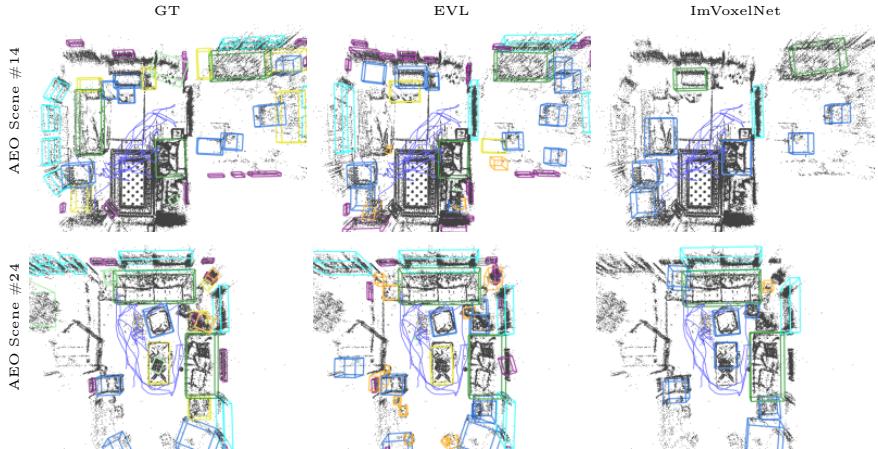


Fig. 5: Aria Everyday Object (AEO) scenes overlaid with the camera trajectories (blue) and the colored 3D bounding boxes from Ground Truth (left), EVL (middle), and ImVoxelNet (right).

approaches perform relatively well on the synthetic data. ImVoxelNet has more false positives and the EVL has tighter bounding boxes. We attribute this gap mostly to the additional pointcloud input that the EVL has that provides a strong geometric cue for localizing the OBBs in 3D space.

In Fig. 5 we show a similar comparison but on real data on the AEO dataset. Textured meshes are not available for this real data so we show the OBBs overlaid on top of the semi-dense point clouds. In this visualization the sim-to-real gap is more apparent. For example, there are never any bay windows in the ASE dataset, but in the top row (AEO Scene #14) shown in cyan there are four bay windows. EVL does a reasonable job to approximate them with a single window, whereas ImVoxelNet fails to detect any window. Another example of the sim-to-real gap is in the bottom-most row (AEO Scene #24) there is a plant shown in a light green OBB on top of the table shown in a yellow OBB. The ASE dataset does not have plants on top of other objects, so it is reasonable that both models fail to detect the plant on top of the table.

We perform a sensitivity analysis of EVL in Table 3. Geometric augmentation at training time is more important than photometric augmentation. We find that standard deviation aggregation is more important than mean aggregation, but using both is best. Removing the freespace encoding has a smaller effect than remove the point mask, but using both is best.

5.2 3D Surface Estimation and Reconstruction

We evaluate the performance of the proposed EVL with occupancy output in comparison to a set of most relevant baseline methods on the final fused reconstructions. The representative methods are from different surface regression

geom. aug.	photo aug.	ASE mAP Snippet	ASE mAP Sequence	mean	std	ASE mAP Snippet	ASE mAP Sequence	pts	free	ASE mAP Snippet	ASE mAP Sequence
✓		0.26	0.52	✓		0.26	0.52	✓		0.38	0.72
✓	✓	0.38	0.67	✓	✓	0.37	0.66	✓	✓	0.36	0.69
✓	✓	0.38	0.68	✓	✓	0.39	0.71	✓	✓	0.39	0.71

(a) **Augmentation:** photometric and geometric augmentations help generalization.

(b) **Aggregation:** feature aggregation using mean and standard deviation (std) and mean, and concatenating both a points and freespace mask helps best.

(c) **Usage of Points:** concatenating a points and freespace mask helps generalization.

Table 3: We modify different hyperparameter settings based on the RGB stream & points EVL. Turning on all augmentations, aggregating features in time using standard deviation (std) and mean, and concatenating both a points and freespace mask leads to the best model.

categories: monocular depth [4] estimation, multi-view depth [26, 49] estimation, volumetric surface reconstruction [51]. For all depth-based methods, we adjust the depth values by aligning the depth output to the semidense points. We use the learned fusion module for NeuralRecon [51] instead of the aforementioned fusion system to generate the final outputs. Since NeuralRecon is the most similar method to EVL, we also re-train it on the same ASE training dataset. For other methods, we use off-the-shelf (OTS) models. In addition, we also evaluate the GT depths and semi-dense points that are provided with the datasets.

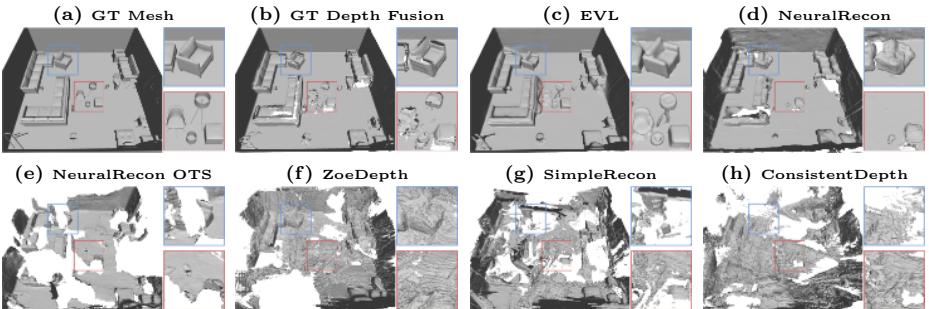


Fig. 6: Surface reconstruction on ASE-100077. 3D volumetric methods like NeuralRecon and EVL reconstruct flat floors and orthogonal walls with explicit 3D priors. EVL delivers a more complete and detailed reconstruction. Depth-based methods fail to produce flat surfaces and result in major holes.

Surface reconstructions are evaluated against GT meshes. We follow [24] to compute mesh-to-mesh metrics as point-cloud-to-mesh metrics, with *Acc* being the mean distance of points sampled from prediction to GT mesh triangles, and *Comp* the other way around (sampled points from GT to predicted surfaces). Please see the supplemental for a rigorous definition of the metrics.

Figure 6 and 7 show qualitative examples. The first three rows in Table 4 demonstrate the quality of the GT depth maps, with a comparison of different camera models.

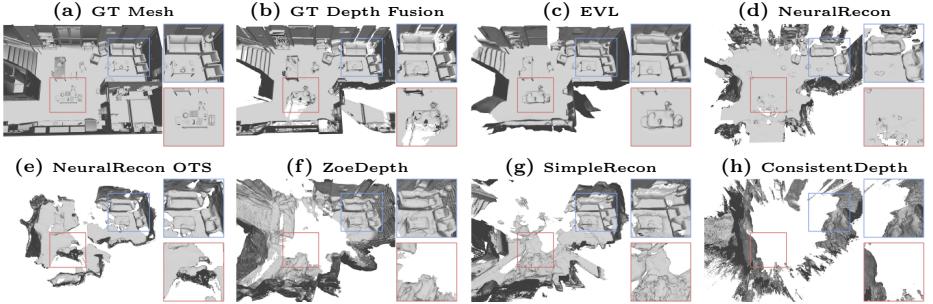


Fig. 7: Surface reconstruction on ADT-106. It shows how models trained on simulation generalize to real data. EVL has the best generalization ability among all the methods.

Using ScanNet-linear camera makes the fusion results much worse on completeness as it has limited FoV as shown in Fig. 3. Fusing monocular depth methods generally does not work well as the scale of the per-frame depth maps can change due to inherent scale ambiguity which leads to incorrect fusion.

Fusion of multi-view depth methods visually looks more consistent as shown in Fig. 6 and 7, but does not have a good result quantitatively, mainly due to noisy depth estimation on low-textured regions such as walls. Volumetric methods generally perform better than depth-based methods. We test both the OTS NeuralRecon model trained on ScanNet, and the model re-trained on ASE dataset. Since the input to the OTS NeuralRecon model is ScanNet-linear images, the input has limited FoV and hence has worse completeness.

	Model	Modality	ASE Acc ↓	ASE Comp ↓	ASE Prec ↑	ASE Recal ↑	ADT Acc ↓	ADT Comp ↓	ADT Prec ↑	ADT Recal ↑
<i>GT-Depth-Fisheye</i>	-	frame	0.011	0.473	0.945	0.568	0.060	1.392	0.793	0.360
<i>GT-Depth-MaxLinear</i>	-	frame	0.014	0.487	0.944	0.507	0.052	1.463	0.807	0.357
<i>GT-Depth-ScanNet</i>	-	frame	0.015	1.183	0.952	0.386	0.020	2.744	0.907	0.178
<i>Semidense Points</i>	-	pts	0.034	-	0.943	-	0.122	-	0.720	-
ZoeDepth [4]	OTS	frame	0.368	1.225	0.290	0.130	0.417	2.127	0.200	0.076
ConsistentDepth [26]	OTS	frame	0.349	1.304	0.277	0.125	0.603	2.759	0.145	0.045
SimpleRecon [49]	OTS	snippet	0.539	3.064	0.257	0.064	0.326	3.063	0.257	0.064
NeuralRecon [51]	OTS	snippet	0.110	1.952	0.491	0.160	0.183	3.905	0.371	0.043
NeuralRecon [51]	ASE	snippet	0.212	1.103	0.512	0.241	0.307	3.383	0.474	0.061
EVL (ours)	ASE	snip+pts	0.057	0.877	0.822	0.405	0.182	3.105	0.594	0.106

Table 4: We show in-domain performance on ASE validation dataset as well as sim-to-real generalization performance on the ADT dataset. Accuracy and completeness are measured in meters. Precision and recall are given at a 5cm threshold. We train NeuralRecon and EVL on the ASE dataset. For other baseline methods we use the off-the-shelf (OTS) models. Depth-based methods such as ZoeDepth have better completeness on ADT as they are not limited by a 3D bounding volume.

Both visually and quantitatively, EVL produces much better reconstruction with more smooth surfaces in contrast to the noisy fusion results from depth-based methods. Compared with NeuralRecon trained on the same data, EVL also adds more finer object details shown in zoom-in views. We hypothesise that EVL

performs better than NeuralRecon for two reasons. First, EVL takes advantages of the input semi-dense points, which provide a strong cue for surface prediction and have high accuracy as shown in the *semi-dense points* row. Second, EVL uses DINOv2.5 as the 2D backbone.

6 Limitations and Societal Impact

The EVL 3D feature lifting mechanism and semi-dense point geometry inputs assume a mostly static world. Though the model is robust to some dynamics and scene changes (as they are present in the ADT and AEO real-world datasets), the model cannot handle large scene dynamics. Additionally, the EVL model has a limited 3D viewing frustum (by default we use $4m \times 4m \times 4m$), which cannot process far away scene geometry. We believe that with long enough egocentric data capture we will observe the majority of indoor scenes with this assumption.

All real-world Project Aria sequences go through an anonymization pipeline that blurs all personally identifying information. In addition all data collectors gave consent to the publishing of their data. We believe EFM_s are poised to provide a profound impact for bringing the context of the physical world to egocentric devices such as AR glasses. While our datasets are collected in fully consented environments, the profound fidelity at which EVL already reconstructs objects and scenes reveals the pervasive nature of egocentric sensor data. It is important as EFM_s continue to improve and evolve that strong privacy and consent models are incorporated into the models.

7 Conclusion

In this manuscript, we introduce the concept of 3D Egocentric Foundation Models that integrate egocentric sensor data for 3D scene understanding. We identify two core tasks for 3D EFM_s—3D object detection and surface regression—and create a benchmarks for each task using high quality annotations of datasets captured with Project Aria glasses [13]. When evaluating these tasks over an entire sequence of egocentric data (as opposed to a single frame), existing methods exhibit poor 3D consistency in their predictions that leads to poor performance on the EFM3D benchmark. To address this, we design a simple but effective 3D backbone for egocentric data, EVL, that leverages semi-dense points and image features to produce a 3D voxel grid of features. EVL outperforms all other methods when evaluated on the proposed EFM3D benchmark. The simplicity of this architecture underscores the effectiveness of the 3D inductive biases in EVL. We encourage the development of more sophisticated models that can exploit the richness of egocentric 3D data even more effectively, including the incorporation of dynamic scene understanding and user interaction modeling. Such modeling advancements could improve the performance even further and extend the applicability of 3D EFM_s to a wider range of real-world scenarios.

8 Acknowledgements

We would like to thank Suvam Patra, Armen Avetisyan, Samir Aroudj, Chris Xie, and Henry Howard-Jenkins for help with the ASE dataset. And Yang Lou, Kang Zheng, Shangyi Cheng, and Xiaqing Pan for helping with the dataset releases, as well as Thomas Whelan for help with tuning the fusion system. We would like to thank Numair Khan for help with evaluating the TemporallyConsistentDepth pipeline and Campbell Orme for narrating the supplemental video. Finally we would like to thank Dan Barnes, Raul Mur Artal, Lingni Ma, Austin Kukay, Rowan Postyeni, Abha Arora and Luis Pesqueira for help with the 3D OBB annotations.

References

1. Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., Le-Cun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture. arXiv preprint arXiv:2301.08243 (2023) [2](#)
2. Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Niessner, M.: Scan2cad: Learning cad model alignment in rgb-d scans. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) [3](#)
3. Baruch, G., Chen, Z., Dehghan, A., Dimry, T., Feigin, Y., Fu, P., Gebauer, T., Joffe, B., Kurz, D., Schwartz, A., Shulman, E.: ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1) (2021), https://openreview.net/forum?id=tjZjv_qh_CE [2](#), [3](#)
4. Bhat, S.F., Birk, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth (2023). <https://doi.org/10.48550/ARXIV.2302.12288>, <https://arxiv.org/abs/2302.12288> [4](#), [8](#), [12](#), [13](#)
5. Brazil, G., Kumar, A., Straub, J., Ravi, N., Johnson, J., Gkioxari, G.: Omni3d: A large benchmark and model for 3d object detection in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13154–13164 (June 2023) [4](#), [8](#), [9](#), [25](#)
6. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Lioung, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020) [2](#)
7. Chen, Y., Li, Y., Zhang, X., Sun, J., Jia, J.: Focal sparse convolutional networks for 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022) [4](#)
8. Dabhi, M., Jeni, L.A., Lucey, S.: 3d-lfm: Lifting foundation model. arXiv preprint arXiv:2312.11894 (2023) [3](#)
9. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017) [2](#), [3](#)
10. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Scaling egocentric vision: The epic-kitchens dataset. In: European Conference on Computer Vision (ECCV) (2018) [2](#)
11. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers (2023) [8](#)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) [2](#)
13. Engel, J., Somasundaram, K., Goesele, M., Sun, A., Gumno, A., Turner, A., Talattof, A., Yuan, A., Souti, B., Meredith, B., Peng, C., Sweeney, C., Wilson, C., Barnes, D., DeTone, D., Caruso, D., Valleroy, D., Ginjupalli, D., Frost, D., Miller, E., Mueggler, E., Oleinik, E., Zhang, F., Somasundaram, G., Solaira, G., Lanaras, H., Howard-Jenkins, H., Tang, H., Kim, H.J., Rivera, J., Luo, J., Dong, J., Straub, J., Bailey, K., Eckenhoff, K., Ma, L., Pesqueira, L., Schwesinger, M., Monge, M., Yang, N., Charron, N., Raina, N., Parkhi, O., Borschowa, P., Moulon, P., Gupta, P., Mur-Artal, R., Pennington, R., Kulkarni, S., Miglani, S., Gondi, S., Solanki,

- S., Diener, S., Cheng, S., Green, S., Saarinen, S., Patra, S., Mourikis, T., Whelan, T., Singh, T., Balntas, V., Baiyya, V., Dreewes, W., Pan, X., Lou, Y., Zhao, Y., Mansour, Y., Zou, Y., Lv, Z., Wang, Z., Yan, M., Ren, C., Nardi, R.D., Newcombe, R.: Project aria: A new tool for egocentric multi-modal ai research (2023) 2, 6, 8, 14
- 14
14. Fu, H., Cai, B., Gao, L., Zhang, L.X., Wang, J., Li, C., Zeng, Q., Sun, C., Jia, R., Zhao, B., et al.: 3d-front: 3d furnished rooms with layouts and semantics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10933–10942 (2021) 3
15. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence* **32**(8), 1362–1376 (2009) 4
16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 2
17. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: 2007 IEEE 11th International Conference on Computer Vision. pp. 1–8. IEEE (2007) 4
18. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamberger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S.K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E.Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erappalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Gebreselasie, A., Gonzalez, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolar, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Puentes, P.R., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhao, Z., Zhu, Y., Arbelaez, P., Crandall, D., Damen, D., Farinella, G.M., Fuegen, C., Ghanem, B., Ithapu, V.K., Jawahar, C.V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H.S., Rehg, J.M., Sato, Y., Shi, J., Shou, M.Z., Torralba, A., Torresani, L., Yan, M., Malik, J.: Ego4d: Around the world in 3,000 hours of egocentric video (2022) 2, 3
19. Grauman, K., Westbury, A., Torresani, L., Kitani, K., Malik, J., Afouras, T., Ashutosh, K., Baiyya, V., Bansal, S., Boote, B., et al.: Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. arXiv preprint arXiv:2311.18259 (2023) 2, 3
20. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5356–5364 (2019) 2
21. Harley, A.W., Lakshminikanth, S.K., Li, F., Zhou, X., Tung, H.Y.F., Fragkiadaki, K.: Learning from unlabelled videos using contrastive predictive neural 3d mapping. arXiv preprint arXiv:1906.03764 (2019) 3
22. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022) 2
23. Huang, D., Peng, S., He, T., Yang, H., Zhou, X., Ouyang, W.: Ponder: Point cloud pre-training via neural rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16089–16098 (2023) 3
24. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 406–413 (2014) 12, 29

25. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* **42**(4) (July 2023), <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/> 4, 34, 35
26. Khan, N., Penner, E., Lanman, D., Xiao, L.: Temporally consistent online depth estimation using point-based fusion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9119–9129 (2023) 4, 8, 12, 13
27. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014) 8
28. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. *arXiv:2304.02643* (2023) 2
29. Lee, Y.J., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: *2012 IEEE conference on computer vision and pattern recognition*. pp. 1346–1353. IEEE (2012) 2
30. Li, K., DeTone, D., Chen, Y.F.S., Vo, M., Reid, I., Rezatofighi, H., Sweeney, C., Straub, J., Newcombe, R.: Odam: Object detection, association, and mapping using posed rgb video. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 5998–6008 (2021) 4
31. Li, K., Rezatofighi, H., Reid, I.: Moltr: Multiple object localization, tracking and reconstruction from monocular rgb videos. *IEEE Robotics and Automation Letters* **6**(2), 3341–3348 (2021) 4
32. Li, W., Saeedi, S., McCormac, J., Clark, R., Tzoumanikas, D., Ye, Q., Huang, Y., Tang, R., Leutenegger, S.: Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In: *British Machine Vision Conference (BMVC)* (2018) 3
33. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*. pp. 2980–2988 (2017) 7
34. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V* 13. pp. 740–755. Springer (2014) 2
35. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. Association for Computing Machinery (1998) 9, 28
36. Lv, Z., Charron, N., Moulon, P., Gamino, A., Peng, C., Sweeney, C., Miller, E., Tang, H., Meissner, J., Dong, J., Somasundaram, K., Pesqueira, L., Schwesinger, M., Parkhi, O., Gu, Q., Nardi, R.D., Cheng, S., Saarinen, S., Baiyya, V., Zou, Y., Newcombe, R., Engel, J.J., Pan, X., Ren, C.: Aria everyday activities dataset (2024) 3
37. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021) 3, 4
38. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 2906–2917 (2021) 3, 4, 9
39. Murez, Z., van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., Rabinovich, A.: Atlas: End-to-end 3d scene reconstruction from posed images. In: *ECCV* (2020), <https://arxiv.org/abs/2003.10432> 4, 29

40. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality. pp. 127–136. Ieee (2011) [9](#), [28](#)
41. Northcutt, C., Zha, S., Lovegrove, S., Newcombe, R.: EgoCom: A multi-person multi-modal egocentric communications dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020) [2](#)
42. Oquab, M., Darabet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023) [2](#), [5](#), [8](#)
43. Pan, X., Charron, N., Yang, Y., Peters, S., Whelan, T., Kong, C., Parkhi, O., Newcombe, R., Ren, Y.C.: Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 20133–20143 (October 2023) [3](#)
44. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 194–210. Springer (2020) [4](#)
45. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9277–9286 (2019) [4](#)
46. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) [8](#)
47. Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10912–10922 (2021) [3](#)
48. Rukhovich, D., Vorontsova, A., Konushin, A.: Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2397–2406 (2022) [4](#), [6](#), [8](#), [9](#), [10](#), [25](#)
49. Sayed, M., Gibson, J., Watson, J., Prisacariu, V., Firman, M., Godard, C.: Simplercon: 3d reconstruction without 3d convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022) [4](#), [8](#), [12](#), [13](#)
50. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019) [2](#)
51. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: NeuralRecon: Real-time coherent 3D reconstruction from monocular video. CVPR (2021) [4](#), [8](#), [12](#), [13](#), [29](#)
52. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023) [34](#)

53. Team, A.: Aria synthetic environments dataset. <https://www.projectaria.com/datasets/ase/>, accessed: 2024-02-26 2, 3
54. Tyszkiewicz, M.J., Maninis, K.K., Popov, S., Ferrari, V.: Raytran: 3d pose estimation and shape reconstruction of multiple objects from videos with ray-traced transformers. In: European Conference on Computer Vision. pp. 211–228. Springer (2022) 4
55. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022) 4
56. Xie, Y., Jiang, H., Gkioxari, G., Straub, J.: Pixel-aligned recurrent queries for multi-view 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 18370–18380 (October 2023) 4, 7, 9
57. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018) 4
58. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the International Conference on Computer Vision (ICCV) (2023) 2, 3
59. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021) 4
60. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: CVPR (2017) 9, 28
61. Zhu, H., Yang, H., Wu, X., Huang, D., Zhang, S., He, X., He, T., Zhao, H., Shen, C., Qiao, Y., Ouyang, W.: Ponderv2: Pave the way for 3d foundation model with a universal pre-training paradigm. arXiv preprint arXiv:2310.08586 (2023) 3

In the following we provide additional details for the models we trained on ASE including EVL, the ASE and AEO datasets we release, more details for the EFM3D benchmark, and finally some additional experiments.

A Model and Training Details

We first describe more details for the EVL model before detailing training procedures for all models.

A.1 EVL Model Details

EVL Architecture Details: We use a frozen base DINOv2.5 backbone with an output feature dimension of 768. We upsample it using a simple 2D upsample network. The 2D upsample network alternates 2D upsampling by a factor of $2 \times$, 3×3 2D convolution, ReLU activation, and batch-norm. In the upsampling we reduce the input feature dimension to an output dimension of 64. These upsampled 64-dimensional features are at the same resolution of the input image. After resampling and aggregating these image features into the feature volume, we run a simple 3D U-Net to process the features in 3D. The 3D U-Net down-samples the 3D volume three times down to 1/8th of the input resolution by iterating 3×3 3D convolutions, batch norm, ReLU activation and max pooling. The upsampling uses tri-linear $2 \times$ upsampling layers that get combined with skip connections and then processed with a sequence of 3×3 3D convolutions, batch norm and ReLU. All intermediate feature dimensions are kept at 256. We currently keep the 3D object detection model and surface reconstruction model separate. The DINOv2.5 base model has 86.6M non-trainable parameters. On top of that the EVL model has 16.7M trainable parameters.

Computing Gravity-Aligned Voxel Grid We define a gravity-aligned voxel grid pose by aligning the camera pose to gravity as follows. Given camera rotation $R_{wc} = [r_x, r_y, r_z]$ and unit-length gravity direction g_w in world coordinates we compute the gravity aligned rotation R_{wg} as:

$$R_{wg} = [g_w, \lceil d_z \times g_w \rceil, \lceil d_z \rceil] \quad \text{where} \quad d_z = r_z - g_w^T r_z g_w, \quad (3)$$

and $\lceil x \rceil$ normalizes a vector x to unit length.

A.2 Model Training Details

Overall the ASE training dataset contains 100k sequences. For OBB training we use a 10k sequence subset which contains 600k 1s snippets. For surface regression training we use a 1k sequence subset of 60k 1s snippets. We find these subsets are sufficiently large for solid performance at reasonable training times. For OBBs we find that the validation set performance saturates at 10k sequences with the current model architecture of EVL (see Sec. D.2). More details in Sec. B.

EVL OBB training: We train all configurations for 3D OBB regression for 10 epochs on the ASE 10k training dataset. We use a max learning rate of 5e-4

with an effective batch size of 160. The learning rate is linear warmed up for 8% and the cosine annealed. We set the loss weights to $w_c = 100$, $w_{iou} = 10$ and $w_{cls} = 1$.

EVL Occupancy training: We train all configurations for occupancy regression for 60 epochs on the ASE 1k training dataset. We use a max learning rate of 4e-4 with an effective batch size of 128. Similar to the OBB training the learning rate is linear warmed up until 8% for the first 5 epochs. The surface loss by a factor of 1.0 and the TV-loss by 0.01.

3DETR training: We choose the base configuration of 3DETR targeted for SUN-RGBD with a GIoU matching cost of 5.0 to train for oriented bounding box regression. We train for 50 epochs on the ASE training dataset and a cosine learning rate schedule with linear warmup. The max learning rate is 7e-4 and the effective batch size 320. We use the same point cloud cropping data augmentation as in the original implementation.

ImVoxelNet training: We choose the base configuration of ImVoxelNet with the SUB-RGBD config to train for oriented bounding box regression. We freeze the ResNet50 backbone to help with sim-to-real transfer. We train for 10 epochs on the ASE training dataset and a step learning rate schedule. The starting learning rate is 1e-4 and gets reduced by a factor of 10x after 67% and again after 83% of training. The effective batch size is 320. We provide max-linear rectified images.

Cube R-CNN training: We use the DLA backbone and train it on the ASE training dataset for 10 epochs. A max learning rate of 1e-4 is cosine rate scheduled. The effective batch size is 640. We train on max-linear rectified images.

NeuralRecon training: Among the surface reconstruction baseline methods, NeuralRecon is the only one we did the retraining. We train NeuralRecon using the default hyper-parameters on the ASE 1k training subset. It has a similar voxel configuration to EVL Occupancy model, with a [96, 96, 96] local volume and a 0.04m voxel size. We use a start learning rate of 2e-4. NeuralRecon has a two-phase training scheme, i.e. TSDF local volume regression and learned fusion training. We train each phase for 10 epochs.

B Dataset Details

In the following we describe details of the Aria Synthetic Environment (ASE) training dataset as well as the real-world Aria Everyday Objects (AEO) 3D OBB validation dataset.

B.1 Aria Synthetic Environment (ASE) Details

Subselecting ASE Sequences. As described earlier in Section D.2, we found that scaling the training beyond 10k sequences resulted in diminishing returns in terms of 3D Object Detection results on holdout ASE sequences. We thus report all OBB experimental results using a 10k sequence subset of the 100k sequences to reduce the engineering and infrastructure requirements for training.

Subselecting ASE Classes. We sub-select 27 total semantic classes in the ASE dataset by combining similar and filtering out certain classes from the original 42 semantic labels. We removed pure structural elements such as “Floor”, “Wall”, “Ceiling” and “Stairs”, because we assume that the full extent of the object can be reasonably visible within a short video snippet. We combined certain similar categories such as "Ottoman", "SittingChair" and "bench" into a single class called "Chair". We also removed some vague classes such as "ElectricalCable" and "Decoration". "Cutlery" and "BatteryCharger" are also removed because they are filtered by the absolute pixel counting. In the end we were left with 27 total classes.

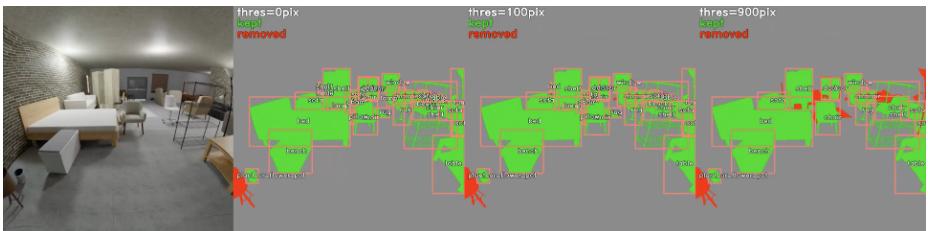


Fig. 8: Heuristic for assigning 2D-3D observability. For a given RGB image (left), we visualize various thresholds for 2D-3D observability, going from more objects (left-middle) to fewer objects (right). We ultimately used the 100px threshold (middle-right).

Visibility Annotation Details. In order to train a 3D object detector that operates on short video snippets, it is necessary that we select which 3D objects to assign as "visible" within the video snippet. With a simulated dataset such as ASE, we have per pixel observability information for each camera. We first associate the 3D OBBs with the 2D instance masks of the images to determine the observability. As no OBBs in ASE intersects with each other, we achieve this by simply unprojecting the pixels inside an instance mask using ground-truth depth maps and the camera calibrations, then associate the mask with the OBB which contains the unprojected points. We further filter out the OBBs of which the mask contains too few pixels to account for the occlusion and very far away objects. In this paper, we use a simple heuristic which is to filter out the OBB observations with less than 100 pixels in the 704x704 RGB camera image.

ASE Per Class Statistics. A histogram of the per-class 3d orientend bounding box counts are shown for 10k ASE sequences in Fig. 9, for a total of about 251k OBBs.

B.2 Aria Everyday Objects (AEO) Details

AEO More Details. The AEO dataset consists of short (1-2 minute) recordings from Project Aria, collected by non-computer vision expert wearers. This results

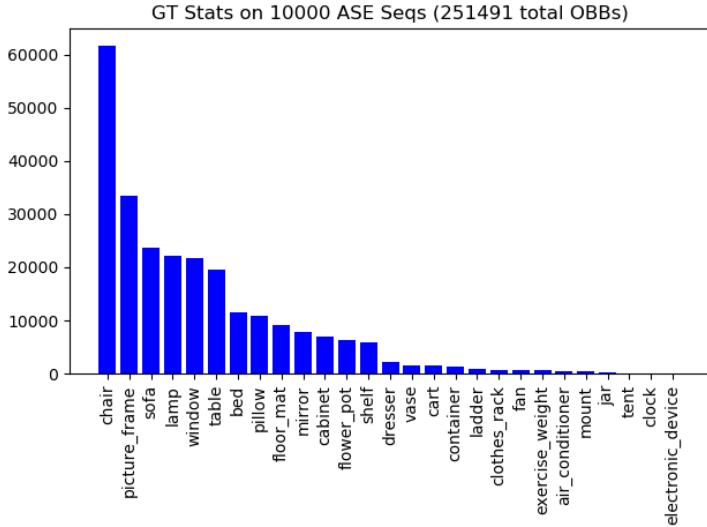


Fig. 9: 3D Object Bounding box per class frequency histogram for ASE.

in trajectories that do not scan entire scenes but rather explore scenes in a more natural way, often focused on one or a handful of objects. The dataset consists of 20 unique scenes across 26 sequences with 584 total OBBs. Most sequences are collected indoors, though there are a few collected in backyards and a garage. Top down views of the dataset are visualized in Fig. 11 and visualizations with the 3D OBBs projected onto the three Project Aria images are shown in Fig. 12.

Remapping ASE Classes to AEO. The ASE taxonomy does not perfectly match the AEO taxonomy. The classes "Couch", "Table", "Bed", "Chair", "Window", "Mirror" and "Lamp" have the same name so we match those directly. The classes "PictureFrameOrPainting" is mapped to "wall art" and "PlantOrFlowerPot" is mapped to "plant". We note that the definition of each OBB differs slightly, for example "lamp" in AEO covers floor lamps, chandeliers and recessed overhead lighting for example, while in ASE it typically only covers floor lamp. These slight differences in taxonomy definition are another potential source of limitations for sim-to-real generalization. Per-class object detection performance differences shown in Sec. D.1 indicate as much. We leave open-taxonomy OBB detection to future work which should mitigate this issue.

AEO Per Class Statistics. A histogram of the per-class 3d orientend bounding box counts are shown for 26 AEO sequences in Fig. 10, for a total of 584 OBBs.

C EFM3D Benchmark Details

We first provide more details for the tracking and fusion of 3D OBBs as well as scene surfaces before describing the surface error metrics in more detail.

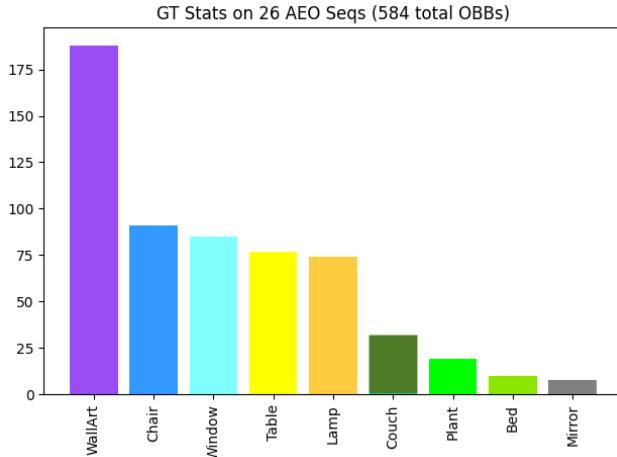


Fig. 10: 3D Object Bounding box per class frequency histogram for AEO. The colored bars for AEO correspond to the classes visualized throughout this paper.

C.1 Tracking and Fusion of 3D Bounding Box Predictions

Sequential object detectors predict an independent set of 3D OBBs per frame (e.g., CubeRCNN [5]) or per snippet (e.g., ImVoxelNet [48]). Directly appending all the independent sets of 3D OBBs to the scene state results in noisy and duplicated objects. Therefore, we propose a simple object tracker named *ObbTracker* to track and update the objects in the scene from the sequential predictions.

ObbTracker gets the raw predictions as the inputs and maintains a scene state which consists of a set of tracked 3D OBBs. Note that the raw predictions from the object detector are transformed into the world coordinate system using the camera poses. After filtering out the low-confidence predictions, ObbTracker performs 3 key steps: Association, Update, and Removal, which we will explain in detail.

Association. We match the predictions that clear a matching score threshold of p_{assoc} and the tracked OBBs in the scene using Hungarian Matching with the cost function:

$$C_{assoc} = w_1 \cdot C_{class} + w_2 \cdot C_{bbox2} + w_3 \cdot C_{bbox3} + w_4 \cdot C_{iou2d} + w_5 \cdot C_{iou3d}, \quad (4)$$

where C_{class} measures the discrepancy between the classes of the predictions and the tracked 3D OBBs, which is implemented as the probability of tracked OBB's class evaluated on the predicted class distribution, C_{bbox2} is the distance between the 2D bounding box centers, C_{bbox3} is the distance between the 3D bounding box centers, C_{iou2d} is the 2D Intersection over Union (IoU) between the 2D bounding boxes, and C_{iou3d} is the 3D IoU between the 3D bounding boxes. Specifically we set $w_1 = 8$, $w_2 = 0$, $w_3 = 1$, $w_4 = 2$, $w_5 = 0$ for all experiments. Then we further filter out the matches where the 2D and 3D IoUs

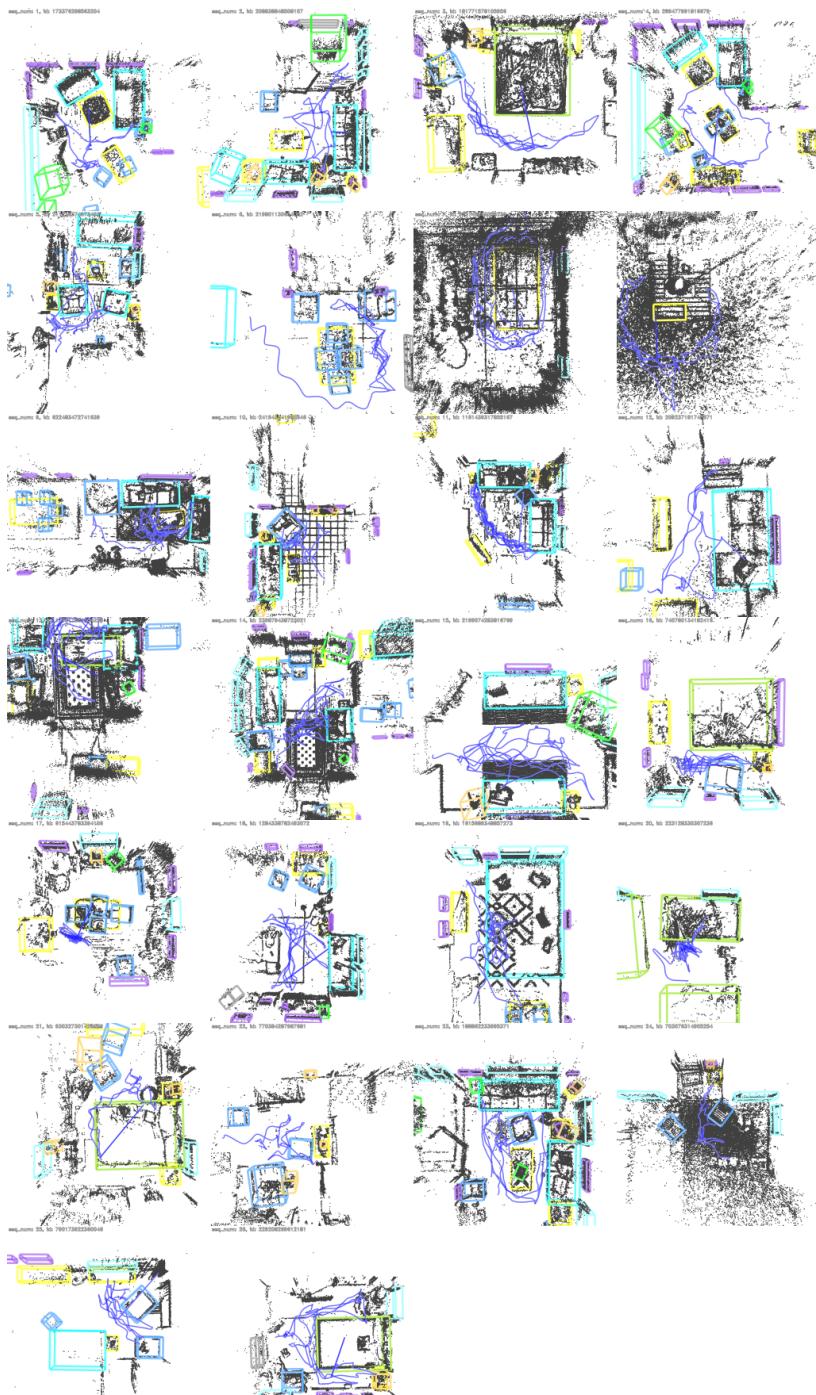


Fig. 11: Top-down view of each of the 26 scenes from the AEO dataset show the OBBs (colored by class), point cloud (black) and Project Aria trajectory (blue).

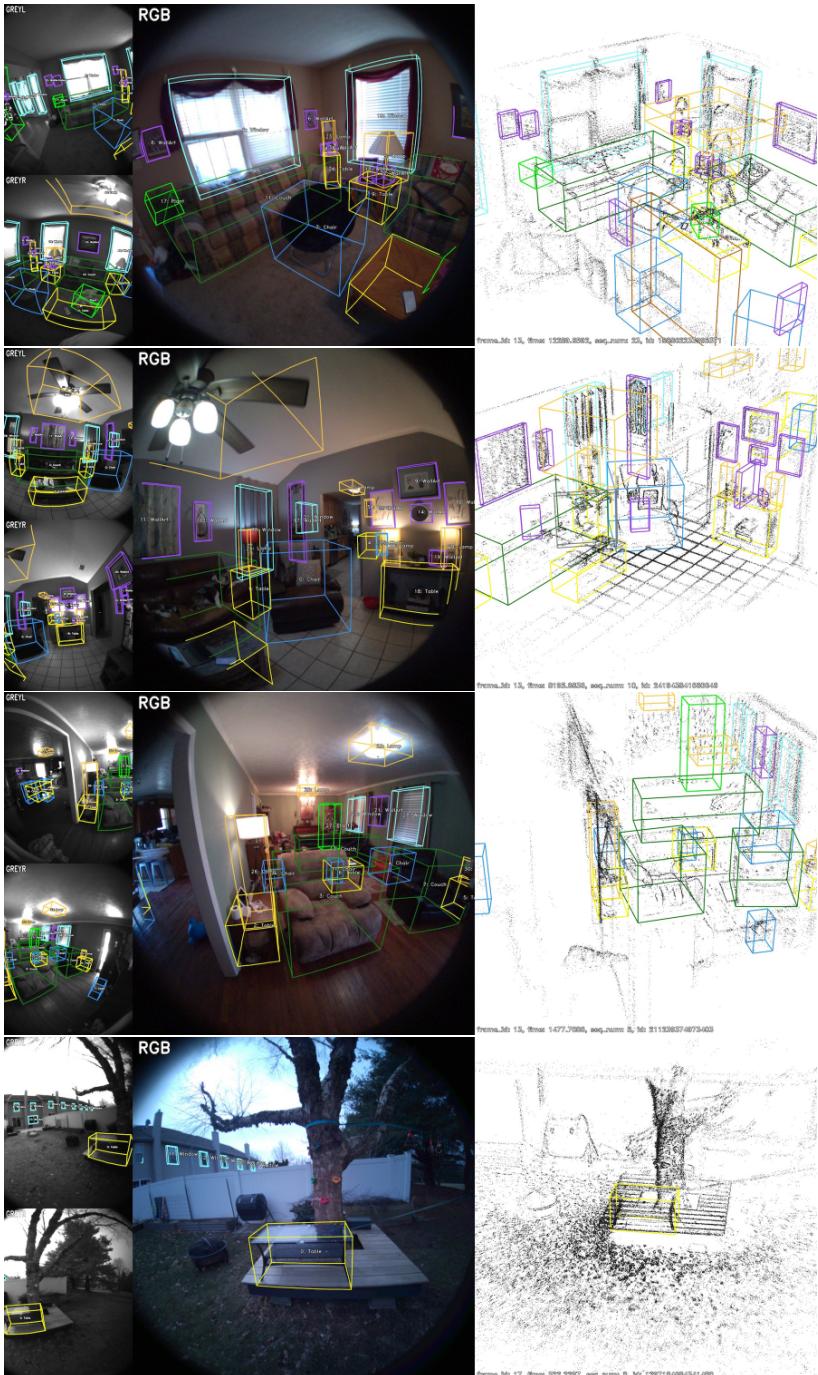


Fig. 12: Example Ground Truth 3D Object Bounding Box Labels for AEO, each row is from a different sequence. The left two images show the OBBs visualized in the greyscale images, the middle shows the OBBs in the RGB image, and the right side shows a 3D visualization of the OBBs alongside the full scene point cloud

are higher than predefined threshold of 0.2 for both 2D and 3D IoU. Those good matches will be used for updating the tracked OBBs. The remaining predictions are added to the scene as new candidates if they clear an instantiation score threshold p_{inst} . We tune the instantiation and association score thresholds but typically keep $p_{assoc} = p_{inst} - 0.05$.

Update. The states of the tracked OBBs are updated with the associated predictions using running average. We define the states of an OBB as its scales $s \in \mathbb{R}^3$, pose in the world coordinate frame $T_{obj}^w \in SE(3)$, and the class probability distribution $c \in \mathbb{R}^k$ where k is the number of total categories in the taxonomy. Given a predicted set of parameters s' , c' , and $T_{obj}^{w'}$, the running average of the object parameters is computed as

$$n' = n + 1 \quad (5)$$

$$s = (sn + s')/n' \quad (6)$$

$$c = (cn + c')/n' \quad (7)$$

$$T_{obj}^w = T_{obj}^w \cdot \exp(T_{obj}^{w'} \boxminus T_{obj}^w/n'), \quad (8)$$

where $T_{obj}^{w'}$ is the predicted object pose, n is the number of observations of the tracked OBB, and $T_a \boxminus T_b = \log(T_a^{-1}T_b)$ is the difference between two poses on the pose manifold SE3 and \exp is the exponential map to SE3.

Removal. After the update step, we remove the tracked OBBs which do not get enough observations w_{min} within a certain period of time t_{inst} after it's added to the scene. We set $n_{min} = 2$ and $t_{inst} = 1s$. We further suppress duplicate scene OBBs by using both 3D non-maximum suppression (NMS) with IoU threshold of 0.1 and 2D NMS with IoU threshold of 0.5.

C.2 Fusion of Surface Predictions

To persist local surface observations into a globally consistent estimation, a fusion system is needed for either depth-based or volumetric methods.

Depth map fusion. When evaluating depth-map-based surface reconstruction, we use a simple implementation from [60] to fuse per-frame observations into an TSDF volume [40]. The method iteratively projects global volume to depth maps, and computes TSDF values by merging the current depth observations in a global averaging way. After getting the global TSDF volume, the meshes are extracted using the marching cubes algorithm [35].

Occupancy fusion. For volumetric methods like EVL, a local occupancy volume is predicted per snippet. We fuse the local occupancy grids into a global grid. Each local occupancy grid is iteratively fused into the global volume, by transforming the global volume to local ones and sampling from them. Similar to depth map fusion, We extract mesh surfaces using marching cubes algorithm.

Implementation details. Both surface fusion algorithms follow an iterative style by feeding the current observations to a global TSDF (for depths) or occupancy (for EVL) volume. We use the iso-level of 0 for TSDF fusion and 0.5

for occupancy fusion. In surface extraction, we set a minimal number of observations for a voxel to be considered valid, which is 2 for TSDF fusion and 5 for occupancy fusion.

C.3 Surface Error Metrics

Unlike the previous evaluations [39, 51] that degenerates the metrics to point-cloud-to-point-cloud distance, we strictly evaluate mesh-to-mesh distance, since having GT meshes is one of the unique values of the proposed datasets. We follow Jensen *et al.* [24] to compute accuracy and completeness errors, denoted as *Acc* and *Comp*. Accuracy is the distance of the reconstruction (i.e. the input) to the ground truth (i.e. the reference); completeness is the distance from ground truth to reconstruction. When input and reference are meshes, both accuracy and completeness is computed. When the input is the semi-dense points, only accuracy is computed. To compute mesh-to-mesh distance, the implementation first samples a fixed number ($=10k$) of points on the input mesh, and then computes the distance of these points to the closest face of the reference mesh. Like previous evaluations, we also compute *Precision@5cm* (*Prec*) and *Recall@5cm* (*Recal*) which are the ratio of point-to-mesh distances that are within 5cm. These metrics are mathematically defined in Table 5, where P and P^* denote the sampled point sets from the predicted and GT mesh. T and T^* denote the triangles for the predicted and GT meshes. $\text{Dist}(p, f)$ is the orthogonal distance from a point p to a triangle face f .

Acc	$\text{mean}_{p \in P}(\min_{f \in T^*} \text{Dist}(p, f))$
Comp	$\text{mean}_{p \in P^*}(\min_{f \in T} \text{Dist}(p, f))$
Prec	$\text{mean}_{p \in P}(\min_{f \in T^*} \text{Dist}(p, f) < 0.05)$
Recal	$\text{mean}_{p \in P^*}(\min_{f \in T} \text{Dist}(p, f) < 0.05)$

Table 5: Surface metric definitions.

D Additional Experiments and Details

We provide more details for EVL OBB detection performance by drilling into per-class mAPs, show more qualitative experiments on AEO, and illustrate some failure cases for OBB detection. Additionally, we provide additional sensitivity studies for surface regression with EVL and show failure modes of Gaussian Splats for surface estimation on egocentric data. The later justifies why we did not compare against Gaussian Splats.

D.1 EVL per-class Object Detection Performance

In Table 6 we show per-class mAPs for the four models on AEO and in Table 7 on the ASE validation dataset. Unlike on ASE, no model does well on

the Mirror, Lamp or Plant/FlowerPot class on AEO which is likely due to substantial sim-to-real gap in the class examples in the ASE training dataset. For the Lamp and Plant/FlowerPot class the typical position on top of other objects in the real world may also contribute to the sim-to-real gap since in ASE all objects are placed on the floor. Both ImVoxelNet and 3DETR struggle with the WallArt/Picture and Window classes which EVL performs well on. Larger, relatively well defined objects such as Bed, Chair, Sofa/Couch and Table are detected reasonably well by all detectors.

	Bed	Chair	Couch	Lamp	Mirror	Plant	Table	WallArt	Window
Cube R-CNN	0.32	0.08	0.33	0.00	0.00	0.00	0.02	0.00	0.00
ImVoxelNet	0.49	0.22	0.46	0.05	0.00	0.00	0.08	0.01	0.05
3DETR	0.24	0.23	0.66	0.03	0.00	0.00	0.10	0.06	0.07
EVL (ours)	0.46	0.25	0.65	0.04	0.01	0.00	0.14	0.21	0.18

Table 6: Four models trained on ASE are evaluated the real-world AEO dataset.

	Bed	Chair	FlowerPot	Lamp	Mirror	Picture	Sofa	Table	Window
Cube R-CNN	0.78	0.62	0.32	0.37	0.04	0.02	0.71	0.54	0.19
ImVoxelNet	0.94	0.92	0.68	0.83	0.19	0.11	0.95	0.91	0.61
3DETR	0.90	0.78	0.18	0.34	0.10	0.13	0.87	0.62	0.36
EVL (ours)	0.95	0.91	0.77	0.86	0.44	0.37	0.94	0.88	0.73

Table 7: Four models trained on ASE are evaluated on the ASE validation dataset.

D.2 EVL Object Detection Scaling Experiments

What training dataset size is big enough to saturate the current EVL model? We train EVL on increasingly large subsets of the ASE training dataset and measure detection performance on the same validation set. We find that EVLs performance saturates at 10k sequences. This is the foundation of our choice to train on the 600k 1s snippets of a subset of 10k sequences from ASE. We expect that transformer-based methods would exhibit better scaling behavior and leave this investigation for future work.

D.3 EVL Failure Cases and Limitations

EVL has only been trained on the simulated ASE training dataset and therefore has only seen a limited set of scene variations. Particularly it has only seen indoor scenes with all objects located on the ground.

Failure cases. In Fig. 15 we show that EVL has problems detecting objects on top of other objects. In this case the plant object on the table is not detected.

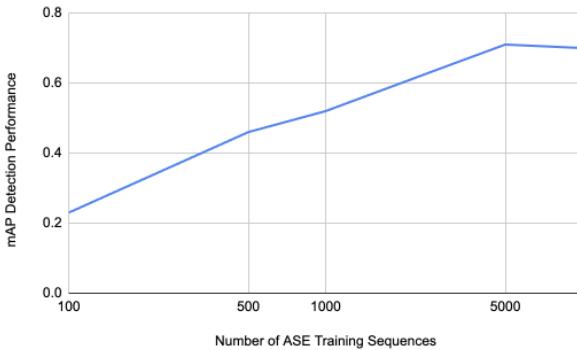


Fig. 13: EVL performance on the same validation set saturates at 10k ASE sequences (the last data point on the log-scaled x-axis).

This domain gap in relative object poses contributes to the substantially lower mAPs seen in objects that are typically not located on the floor (see Sec. D.1). In Fig. 14 we show that the presence of mirrors in the ASE dataset leads to erroneous reflected detections outside the room behind the mirrors. This shows that the model has not yet learned to correctly reason about reflections via mirrors.

Limitations. We show EVL running on an outdoor sequence from AEO in Fig. 16 . As noted before EVL has only seen indoor scenes during training which means that trees and open sky are completely out of the training distribution. EVL reconstruction quality degrades gracefully on this sequence by still predicting smooth surfaces for the tree, table and grass, but struggles with the fine detail of the tree branches. Another drawback is that the EVL model has a limited 3D viewing frustum ($4m \times 4m \times 4m$ by default), which cannot process distant scene geometry which is more common outdoors.

D.4 EVL Sensitivity Study for Surface Estimation

Camera Model Sensitivity: We test EVL surface regression when changing the camera model. Specifically, we rectify the original Fisheye camera model to linear camera models and use the rectified images as the inputs to the EVL model. As shown in Table 8, the performance does not degrade substantially with different camera models. Camera models with smaller field of view like the ScanNet-linear camera model do lead to the largest degradation. Insensitivity to the particular camera model is a valuable property since it theoretically allows training the same model on different kinds of cameras. Likely the semi-dense points inputs are important in keeping performance relatively high independent of the camera model.

Ablation Studies: We show the ablation studies of EVL on Surface regression. Specifically, in Table 9 we show the effectiveness of using the points mask and the freespace mask. We can see that using point masks from the semi-dense points

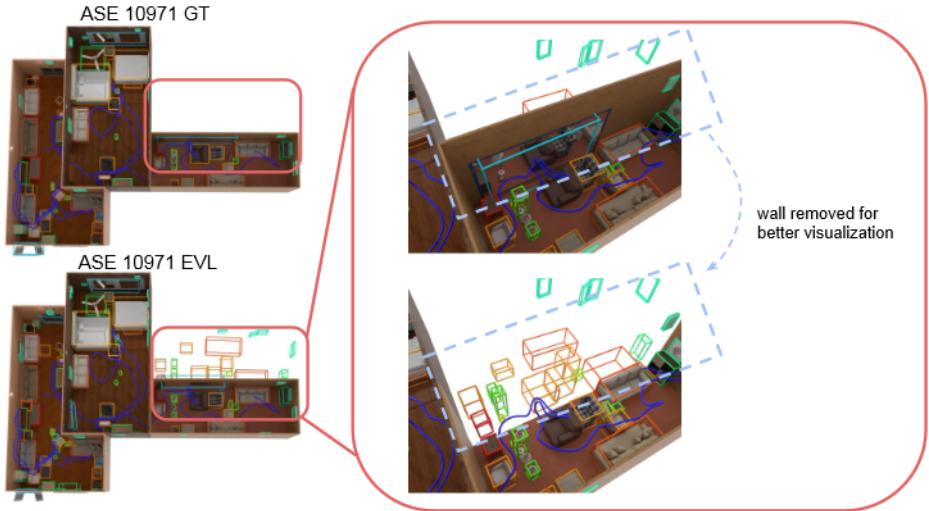


Fig. 14: EVL fails to reason about the reflections and predicts false-positives on the reflected image from the mirror on the wall.

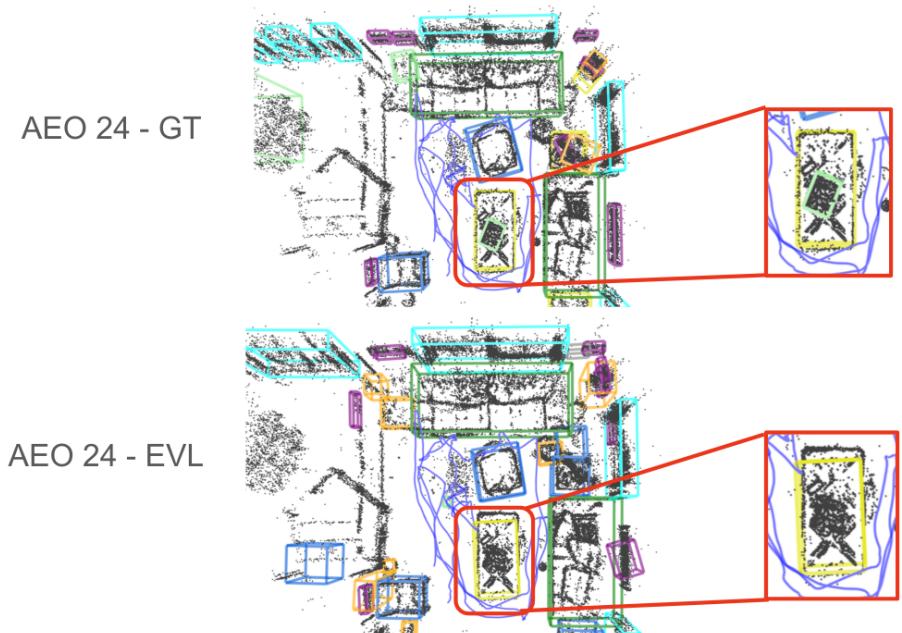


Fig. 15: EVL fails to detect the plant on the table (shown in the green OBB). This is due to a limitation in the training data where there are no objects on top of or another in ASE.

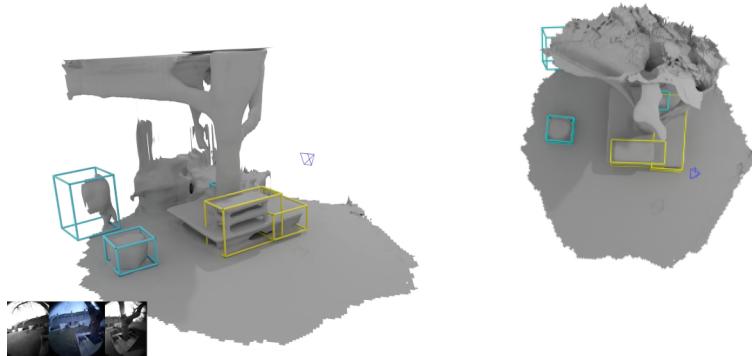


Fig. 16: EVL was trained only on indoor data from ASE yet generalizes reasonably well on outdoor real Project Aria data. However, the limited FoV and lack of detail on tree branches remain problematic, and the 3D viewing frustum ($4m \times 4m \times 4m$) limits the range of reconstructions that EVL can deliver.

Camera Model	ASE Prec ↑	ASE Recal ↑	ADT Prec ↑	ADT Recal ↑
Fisheye	0.73	0.35	0.44	0.077
Max. Linear	0.73	0.35	0.42	0.076
ScanNet Linear	0.69	0.33	0.41	0.070

Table 8: EVL Surface regression performance when changing the camera model.

gives a large boost on the performance and by modeling the freespace it gets further improved.

points	freespace	ASE Prec \uparrow	ASE Recal \uparrow	ADT Prec \uparrow	ADT Recal \uparrow
✓	✓	0.82	0.41	0.59	0.11
✓		0.73	0.35	0.44	0.077
		0.47	0.18	0.14	0.023

Table 9: EVL Surface regression ablation study. Adding point occupancy and freespace mask improves performance.

D.5 Gaussian Splats Experiments

We evaluated 3D Gaussian Splatting [25] as a surface estimator using NerfStudio’s Splatfacto model [52]. The images from the RGB stream were undistorted from fisheye to linear cameras using the Max. Linear camera model before training and the Gaussian point cloud is optimized for 30K iterations with the default settings.

The rendered results of the optimized 3D Gaussian point cloud are shown in Fig. 17. Notice the inconsistent floor texture with many areas of dark patches or fog. These areas exactly correspond to the areas where the device-wearer moved while capturing the video. When the video used to train the 3D Gaussian point cloud contains observations of the wearer (as shown in Fig. 18) the model will attempt to create new Gaussians at the along the wearer’s trajectory to explain those observations (incorrectly) as static Gaussians. This is particularly a problem when computing the depth via rendering because these artifacts create false surfaces or otherwise degrade the depth value output by the rasterization process (see Fig. 18 right). The ADT dataset sequences contain significant dynamics of this nature, and so we did not evaluate Gaussian Splatting results for the surface estimation task. We expect that this rapidly evolving area will provide significant opportunity for future EFM algorithms.



Fig. 17: Top-down view of ADT-136 "work" reconstructed using 3D Gaussian Splatting [25]. Egocentric data contains sparse, dynamic observations of the user throughout the scene. The inability for 3D Gaussians to model these sparse dynamics leads to "fog" throughout the scene placed exactly at the user's trajectory. This "fog" degrades the accuracy of the rendered depth from Gaussian Splatting.

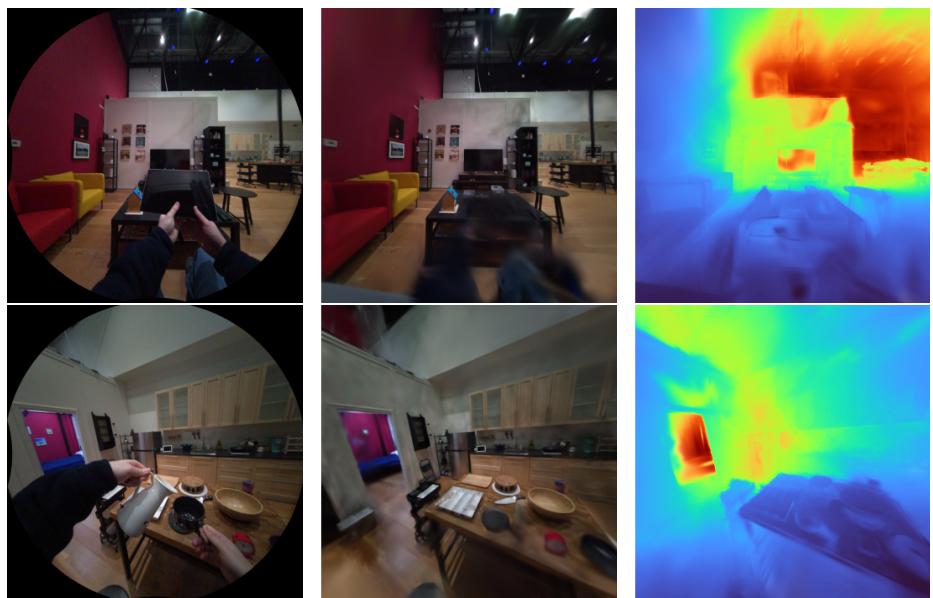


Fig. 18: **Left:** Two video frames from ADT-136 "work" containing observations of the user dynamically interacting with the scene. **Center:** The rendered RGB image of the same viewpoint using the optimized 3D Gaussian point cloud. **Right:** The rendered depth maps using the optimized 3D Gaussian point cloud.