EgoLifter: Open-world 3D Segmentation for Egocentric Perception

Qiao Gu[1,2⋆], Zhaoyang Lv[2], Duncan Frost[2], Simon Green[2], Julian Straub[2], and Chris Sweeney[2]

[1] University of Toronto
[2] Meta Reality Labs

Abstract. In this paper we present *EgoLifter*, a novel system that can automatically segment scenes captured from egocentric sensors into a complete decomposition of individual 3D objects. The system is specifically designed for egocentric data where scenes contain hundreds of objects captured from natural (non-scanning) motion. *EgoLifter* adopts 3D Gaussians as the underlying representation of 3D scenes and objects and uses segmentation masks from the Segment Anything Model (SAM) as weak supervision to learn flexible and promptable definitions of object instances free of any specific object taxonomy. To handle the challenge of dynamic objects in ego-centric videos, we design a transient prediction module that learns to filter out dynamic objects in the 3D reconstruction. The result is a fully automatic pipeline that is able to reconstruct 3D object instances as collections of 3D Gaussians that collectively compose the entire scene. We created a new benchmark on the Aria Digital Twin dataset that quantitatively demonstrates its state-of-the-art performance in open-world 3D segmentation from natural egocentric input. We run *EgoLifter* on various egocentric activity datasets which shows the promise of the method for 3D egocentric perception at scale.

Keywords: Egocentric Perception · Open-world Segmentation · 3D Reconstruction

1 Introduction

The rise of personal wearable devices has led to the increased importance of egocentric machine perception algorithms capable of understanding the physical 3D world around the user. Egocentric videos directly reflect the way humans see the world and contain important information about the physical surroundings and how the human user interacts with them. The specific characteristics of egocentric motion, however, present challenges for 3D computer vision and machine perception algorithms. Unlike datasets captured with deliberate "scanning" motions, egocentric videos are not guaranteed to provide complete coverage of the scene. This makes reconstruction challenging due to limited or missing multi-view observations.

⋆ Work done during internship at Reality Labs, Meta.

Fig. 1: *EgoLifter* solves 3D reconstruction and open-world segmentation simultaneously from egocentric videos. *EgoLifter* augments 3D Gaussian Splatting [16] with instance features and lifts open-world 2D segmentation by contrastive learning, where 3D Gaussians belong to the same objects are learned to have similar features. In this way, *EgoLifter* solves the multi-view mask association problem and establishes a consistent 3D representation that can be decomposed into object instances. *EgoLifter* enables multiple downstream applications including detection, segmentation, 3D object extraction and scene editing. See supplementary material for animated visualizations.

The specific content found in egocentric videos also presents challenges to conventional reconstruction and perception algorithms. An average adult interacts with hundreds of different objects many thousands of times per day [4]. Egocentric videos capturing this frequent human-object interaction thus contain a huge amount of dynamic motion with challenging occlusions. A system capable of providing useful scene understanding from egocentric data must therefore be able to recognize hundreds of different objects while being robust to sparse and rapid dynamics.

To tackle the above challenges, we propose *EgoLifter*, a novel egocentric 3D perception algorithm that simultaneously solves reconstruction and open-world 3D instance segmentation from egocentric videos. We represent the geometry

of the scene using 3D Gaussians [16] that are trained to minimize photometric reconstruction of the input images. To learn a flexible decomposition of objects that make up the scene we leverage SAM [20] for its strong understanding of objects in 2D and lift these object priors into 3D using contrastive learning. Specifically, 3D Gaussians are augmented with additional N-channel feature embeddings that are rasterized into feature images. These features are then learned to encode the object segmentation information by contrastive lifting [1]. This technique allows us to learn a flexible embedding with useful object priors that can be used for several downstream tasks.

To handle the difficulties brought by the dynamic objects in egocentric videos, we design *EgoLifter* to focus on reconstructing the static part of the 3D scene. *EgoLifter* learns a transient prediction network to filter out the dynamic objects from the reconstruction process. This network does not need extra supervision and is optimized together with 3D Gaussian Splatting using solely the photometric reconstruction losses. We show that the transient prediction module not only helps with photorealistic 3D reconstruction but also results in cleaner lifted features and better segmentation performance.

EgoLifter is able to reconstruct a 3D scene while decomposing it into 3D object instances without the need for any human annotation. The method is evaluated on several egocentric video datasets. The experiments demonstrate strong 3D reconstruction and open-world segmentation results. We also showcase several qualitative applications including 3D object extraction and scene editing. The contributions of this paper can be summarized as follows:

- We demonstrate *EgoLifter*, the first system that can enable open-world 3D understanding from natural dynamic egocentric videos.
- By lifting output from recent image foundation models to 3D Gaussian Splatting, *EgoLifter* achieve strong open-world 3D instance segmentation performance without the need for expensive data annotation or extra training.
- We propose a transient prediction network, which filters out transient objects from the 3D reconstruction results. By doing so, we achieve improved performance on both reconstruction and segmentation of static objects.
- We set up the first benchmark of dynamic egocentric video data and quantitatively demonstrate the leading performance of *EgoLifter*. On several large-scale egocentric video datasets, *EgoLifter* showcases the ability to decompose a 3D scene into a set of 3D object instances, which opens up promising directions for egocentric video understanding in AR/VR applications.

2 Related Work

2.1 3D Gaussian Models

3D Gaussian Splatting (3DGS) [16] has emerged as a powerful algorithm for novel view synthesis by 3D volumetric neural rendering. It has shown promising performance in many applications, like 3D content generation [3, 47, 58], SLAM [15, 28, 53] and autonomous driving [54, 60]. Recent work extend 3DGS

to dynamic scene reconstruction [6, 25, 52, 55, 56]. The pioneering work from Luiten *et al*. [25] first learns a static 3DGS using the multi-view observations at the initial timestep and then updates it by the observations at the following timesteps. Later work [52, 56] reconstructs dynamic scenes by deforming a canonical 3DGS using a time-conditioned deformation network. Another line of work [6, 55] extends 3D Gaussians to 4D, with an additional variance dimension in time. While they show promising results in dynamic 3D reconstruction, they typically require training videos from multiple static cameras. However, in egocentric perception, there are only one or few cameras with a narrow baseline. As we show in the experiments, dynamic 3DGS struggles to track dynamic objects and results in floaters that harm instance segmentation feature learning.

2.2 Open-world 3D Segmentation

Recent research on open-world 3D segmentation [8, 13, 14, 17, 21, 22, 29, 36, 42, 43, 48, 50] has focused on lifting outputs from 2D open-world models - large, powerful models that are trained on Internet-scale datasets and can generalize to a wide range of concepts [20, 33, 38, 40]. These approaches transfer the ability of powerful 2D models to 3D, require no training on 3D models, and alleviate the need for large-scale 3D datasets that are expensive to collect. Early work [14, 17, 36] lifts dense 2D feature maps to 3D representations by multi-view feature fusion, where each position in 3D is associated with a feature vector. This allows queries in fine granularity over 3D space, but it also incurs high memory usage. Other work [11, 24, 46] builds object-decomposed 3D maps using 2D open-world detection or segmentation models [20, 23], where each 3D object is reconstructed separately and has a single feature vector. This approach provides structured 3D scene understanding in the form of object maps or scene graphs but the scene decomposition is predefined and the granularity does not vary according to the query at inference time. Recently, another work [1] lifts 2D instance segmentation to 3D by contrastive learning. It augments NeRF [31] with an extra feature map output and optimizes it such that pixels belonging to the same 2D segmentation mask are pulled closer and otherwise pushed apart. In this way, multi-view association of 2D segmentation is solved in an implicit manner and the resulting feature map allows instance segmentation by either user queries or clustering algorithms.

Concurrent Work. We briefly review several recent and unpublished preprints that further explore topics in this direction using techniques similar to ours. Concurrently, OmniSeg3D [59] and GARField [18] follow the idea of [1], and focus on learning 3D hierarchical segmentation. They both take advantage of the multi-scale outputs from SAM [20] and incorporate the scales into the lifted features. GaussianGrouping [57] also approaches the open-world 3D segmentation problem but they rely on a 2D video object tracker for multi-view association instead of directly using 2D segmentation via contrastive learning. Similar to our improvement on 3DGS, FMGS [61] and LangSplat [37] also augment 3DGS with feature rendering. They learn to embed the dense features from

foundation models [34,38] into 3DGS such that the 3D scenes can be segmented by language queries. While the concurrent work collectively also achieves 3D reconstruction with the open-world segmentation ability, *EgoLifter* is the first to explicitly handle the dynamic objects that are commonly present in the real-world and especially in egocentric videos. We demonstrate this is a challenge in real-world scenarios and show improvements on it brought by *EgoLifter*.

2.3 3D Reconstruction from Egocentric Videos

NeuralDiff [51] first approached the problem of training an egocentric radiance field reconstruction by decomposing NeRF into three branches, which capture ego actor, dynamic objects, and static background respectively as inductive biases. EPIC-Fields [49] propose an augmented benchmark using 3D reconstruction by augmenting the EPIC-Kitchen [5] dataset using neural reconstruction. They also provide comprehensive reconstruction evaluations of several baseline methods [9,27,51]. Recently, two datasets for egocentric perception, Aria Digital Twin (ADT) dataset [35] and Aria Everyday Activities (AEA) Dataset [26], have been released. Collected by Project Aria devices [7], both datasets feature egocentric video sequences with human actions and contain multimodal data streams and high-quality 3D information. ADT also provides extensive ground truth annotations using a motion capture system. Preliminary studies on egocentric 3D reconstruction have been conducted on these new datasets [26,45] and demonstrate the challenges posed by dynamic motion. In contrast, this paper tackles the challenges in egocentric 3D reconstruction and proposes to filter out transient objects in the videos. Compared to all existing work, we are the first work that holistically tackles the challenges in reconstruction and open-world scene understanding, and set up the quantitative benchmark to systematically evaluate performance in egocentric videos.

3 Method

3.1 3D Gaussian Splatting with Feature Rendering

3D Gaussian Splatting (3DGS) [16] has shown state-of-the-art results in 3D reconstruction and novel view synthesis. However, the original design only reconstructs the color radiance in RGB space and is not able to capture the rich semantic information in a 3D scene. In *EgoLifter*, we augment 3DGS to also render a feature map of arbitrary dimension in a differentiable manner, which enables us to encode high-dimensional features in the learned 3D scenes and lift segmentation from 2D to 3D. These additional feature channels are used to learn object instance semantics in addition to photometric reconstruction.

Formally, 3DGS represents a 3D scene by a set of N colored 3D Gaussians $\mathcal{S} = \{\mathbf{\Theta}_i | i = 1, \cdots, N\}$, with location and shape represented by a center position $\mathbf{p}_i \in \mathbb{R}^3$, an anisotropic 3D covariance $\mathbf{s}_i \in \mathbb{R}^3$ and a rotation quaternion $\mathbf{q}_i \in \mathbb{R}^4$. The radiance of each 3D Gaussian is described by an opacity parameter $\alpha_i \in \mathbb{R}$

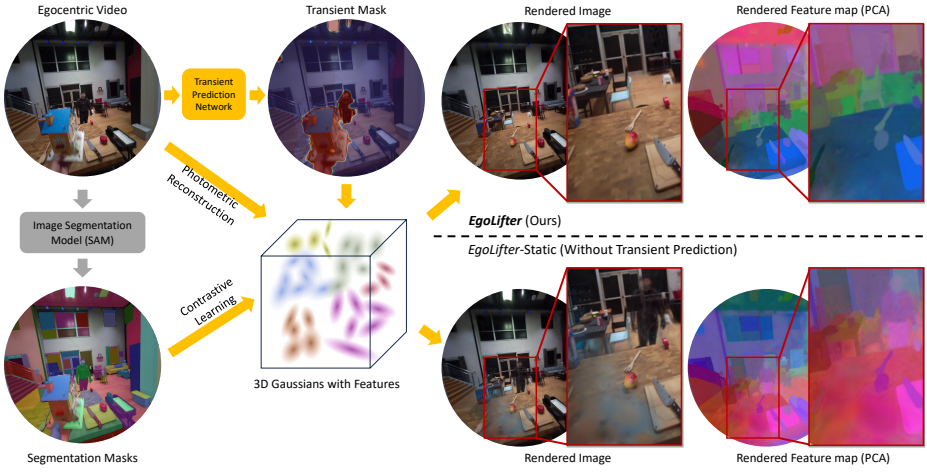

Fig. 2: Naive 3D reconstruction from egocentric videos creates a lot of "floaters" in the reconstruction and leads to blurry rendered images and erroneous instance features (bottom right). *EgoLifter* tackles this problem using a transient prediction network, which predicts a probability mask of transient objects in the image and guides the reconstruction process. In this way, *EgoLifter* gets a much cleaner reconstruction of the static background in both RGB and feature space (top right), which in turn leads to better object decomposition of 3D scenes.

and a color vector $\mathbf{c}_i$, parameterized by spherical harmonics (SH) coefficients. In *EgoLifter*, we additionally associate each 3D Gaussian with an extra feature vector $\mathbf{f} \in \mathbb{R}^d$, and thus the optimizable parameter set for i-th Gaussian is $\mathbf{\Theta}_i = \{\mathbf{p}_i, \mathbf{s}_i, \mathbf{q}_i, \alpha_i, \mathbf{c}_i, \mathbf{f}_i\}$.

To train 3DGS for 3D reconstruction, a set of M observations $\{\mathbf{I}_j, \theta_j | j = 1, \cdots, M\}$ is used, where $\mathbf{I}_j$ is an RGB image and θ_j is the corresponding camera parameters. During the differentiable rendering process, all 3D Gaussians are splatted onto the 2D image plane according to θ_j and α-blended to get a rendered image $\hat{\mathbf{I}}_j$. Then the photometric loss is computed between the rendered image $\hat{\mathbf{I}}_j$ and the corresponding ground truth RGB image $\mathbf{I}_j$ as

$$\mathcal{L}_{\text{RGB}}(\mathbf{I}_j, \hat{\mathbf{I}}_j) = \mathcal{L}_{\text{MSE}}(\mathbf{I}_j, \hat{\mathbf{I}}_j) = \sum_{u \in \Omega} \|\mathbf{I}_j[u] - f(\hat{\mathbf{I}}_j[u])\|_2^2, \tag{1}$$

where $\mathcal{L}_{\text{MSE}}$ is the mean-squared-error (MSE) loss, Ω is set of all coordinates on the image and $\mathbf{I}_j[u]$ denotes the pixel value of $\mathbf{I}_j$ at coordinate u. $f(\cdot)$ is an image formation model that applies special properties of the camera (e.g. vignetting, radius of valid pixels) on the rendered image. By optimizing $\mathcal{L}_{RGB}$, the location, shape, and color parameters of 3D Gaussians are updated to reconstruct the geometry and appearance of the 3D scene. A density control mechanism is also used to split or prune 3D Gaussians during the training process [16].

In *EgoLifter*, we also implement the differentiable feature rendering pipeline similar to that of the RGB images, which renders to a 2D feature map $\hat{\mathbf{F}} \in \mathbb{R}^{H \times W \times d}$ according to the camera information. During the training process, the

feature vectors are supervised by segmentation output obtained from 2D images and jointly optimized with the location and color parameters of each Gaussian. We also include gradients of feature learning for the density control process in learning 3DGS. More details may be found in the supplementary material.

3.2 Learning Instance Features by Contrastive Loss

Egocentric videos capture a huge number of different objects in everyday activities, and some of them may not exist in any 3D datasets for training. Therefore, egocentric 3D perception requires an ability to generalize to unseen categories (open-world) which we propose to achieve by lifting the output from 2D instance segmentation models. The key insight is that 2D instance masks from images of different views can be associated to form a consistent 3D object instance and that this can be done together with the 3D reconstruction process. Recent work has approached this problem using linear assignment [44], video object tracking [57], and incremental matching [11, 24, 46].

To achieve open-world 3D segmentation, we use $\mathbf{f}$ as instance features to capture the lifted segmentation and their similarity to indicate whether a set of Gaussians should be considered as the same object instance. Inspired by Contrastive Lift [1], we adopt supervised contrastive learning, which pulls the rendered features belonging to the same mask closer and pushes those of different masks further apart. Formally, given a training image $\mathbf{I}_j$, we use a 2D segmentation model to extract a set of instance masks $\mathcal{M}_j = \{\mathbf{M}_j^k | k = 1, \cdots, m_i\}$ from $\mathbf{I}_j$. The feature map $\hat{\mathbf{F}}_j$ at the corresponding camera pose θ_j is then rendered, and the contrastive loss is computed over a set of pixel coordinates $\mathcal{U}$, for which we use a uniformly sampled set of pixels $\mathcal{U} \subset \Omega$ due to GPU memory constraint. The contrastive loss is formulated as

$$\mathcal{L}_{\text{contr}}(\hat{\mathbf{F}}_j, \mathcal{M}_j) = -\frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \log \frac{\sum_{u' \in \mathcal{U}^+} \exp(\text{sim}(\hat{\mathbf{F}}_j[u], \hat{\mathbf{F}}_j[u']; \gamma)}{\sum_{u' \in \mathcal{U}} \exp(\text{sim}(\hat{\mathbf{F}}_j[u], \hat{\mathbf{F}}_j[u']; \gamma)}, \tag{2}$$

where $\mathcal{U}^+$ is the set of pixels that belong to the same instance mask as u and $\hat{\mathbf{F}}_j[u]$ denotes the feature vector of the $\hat{\mathbf{F}}_j$ at coordinate u. We use a Gaussian RBF kernel as the similarity function, i.e. $\text{sim}(f_1, f_2; \gamma) = \exp(-\gamma \|f_1 - f_2\|_2^2)$.

In the contrastive loss, pixels on the same instance mask are considered as positive pairs and will have similar features during training. Note that since the 2D segmentation model does not output consistent object instance IDs across different views, the contrastive loss is computed individually on each image. This weak supervision allows the model to maintain a flexible definition of object instances without hard assignments and is key to learning multi-view consistent instance features for 3D Gaussians that enables flexible open-world 3D segmentation.

3.3 Transient Prediction for Egocentric 3D Reconstruction

Egocentric videos contain a lot of dynamic objects that cause many inconsistencies among 3D views. As we show in Fig. 2, the original 3DGS algorithm

on the egocentric videos results in many floaters and harms the results of both reconstruction and feature learning. In *EgoLifter*, we propose to filter out transient phenomena in the egocentric 3D reconstruction, by predicting a transient probability mask from the input image, which is used to guide the 3DGS reconstruction process.

Specifically, we employ a transient prediction network $G(\mathbf{I}_j)$, which takes in the training image $\mathbf{I}_j$ and outputs a probability mask $\mathbf{P}_j \in \mathbb{R}^{H \times W}$ whose value indicates the probability of each pixel being on a transient object. Then $\mathbf{P}_j$ is used to weigh the reconstruction loss during training, such that when a pixel is considered transient, it is filtered out in reconstruction. Therefore the reconstruction loss from Eq. (1) is adapted to

$$\mathcal{L}_{\text{RGB-w}}(\mathbf{I}_j, \hat{\mathbf{I}}_j, \mathbf{P}_j) = \sum_{u \in \Omega} (1 - \mathbf{P}_j[u]) \|\mathbf{I}_j[u] - \hat{\mathbf{I}}_j[u]\|_2^2, \tag{3}$$

where the pixels with lower transient probability will contribute more to the reconstruction loss. As most of the objects in egocentric videos remain static, we also apply an L-1 regularization loss on the predicted $\mathbf{P}_j$ as $\mathcal{L}_{\text{reg}}(\mathbf{P}_j) = \sum_{p \in \mathbf{P}_j} |p|$. This regularization also helps avoid the trivial solution where $\mathbf{P}_j$ equals zero and all pixels are considered transient. The transient mask $\mathbf{P}_j$ is also used to guide contrastive learning for lifting instance segmentation, where the pixel set $\mathcal{U}$ is only sampled on pixels with the probability of being transient less than a threshold δ. As shown in Fig. 2 and Fig. 3, this transient filtering also helps learn cleaner instance features and thus better segmentation results.

In summary, the overall training loss on image $\mathbf{I}_j$ is a weighted sum as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{RGB-w}}(\mathbf{I}_j, \hat{\mathbf{I}}_j, \mathbf{P}_j) + \lambda_2 \mathcal{L}_{\text{contr}}(\hat{\mathbf{F}}_j, \mathcal{M}_j) + \lambda_3 \mathcal{L}_{\text{reg}}(\mathbf{P}_j), \tag{4}$$

with λ_1, λ_2 and λ_3 as hyperparameters.

3.4 Open-world Segmentation

After training, instance features $\mathbf{f}$ capture the similarities among 3D Gaussians, and can be used for open-world segmentation in two ways, query-based and clustering-based. In query-based open-world segmentation, one or few clicks on the object of interest are provided and a query feature vector is computed as the averaged features rendered at these pixels. Then a set of 2D pixels or a set of 3D Gaussians can be obtained by thresholding their Euclidean distances from the query feature, from which a 2D segmentation mask or a 3D bounding box can be estimated. In clustering-based segmentation, an HDBSCAN clustering algorithm [30] is performed to assign 3D Guassians into different groups, which gives a full decomposition of the 3D scene into a set of individual objects. In our experiments, query-based segmentation is used for quantitative evaluation, and clustering-based mainly for qualitative results.

4 Experiments

Implementation. We use a U-Net [41] with the pretrained MobileNet-v3 [12] backbone as the transient prediction network G. The input to G is first resized to 224×224 and then we resize its output back to the original resolution using bilinear interpolation. We use feature dimension $d = 16$, threshold $\delta = 0.5$, temperature $\gamma = 0.01$, and loss weights $\lambda_1 = 1$, $\lambda_2 = 0.1$ and $\lambda_3 = 0.01$. The 3DGS is trained using the Adam optimizer [19] with the same setting and the same density control schedule as in [16]. The transient prediction network is optimized by another Adam optimizer with an initial learning rate of 1×10^{-5}. *EgoLifter* is agnostic to the specific 2D instance segmentation method, and we use the Segment Anything Model (SAM) [20] for its remarkable instance segmentation performance.

Datasets. We evaluate *EgoLifter* on the following egocentric datasets:

- **Aria Digital Twin (ADT) [35]** provides 3D ground truth for objects paired with egocentric videos, which we used to evaluate *EgoLifter* quantitatively. ADT dataset contains 200 egocentric video sequences of daily activities, captured using Aria glasses. ADT also uses a high-quality simulator and motion capture devices for extensive ground truth annotations, including 3D object bounding boxes and 2D segmentation masks for all frames. ADT does not contain an off-the-shelf setting for scene reconstruction or open-world 3D segmentation. We create the evaluation benchmark using the GT 2D masks and 3D bounding boxes by reprocessing the 3D annotations. Note that only the RGB images are used during training, and for contrastive learning, we used the masks obtained by SAM [20].
- **Aria Everyday Activities (AEA) dataset [26]** provides 143 egocentric videos of various daily activities performed by multiple wearers in five different indoor locations. Different from ADT, AEA contains more natural video activity recordings but does not offer 3D annotations. For each location, multiple sequences of different activities are captured at different times but aligned in the same 3D coordinate space. Different frames or recordings may observe the same local space at different time with various dynamic actions, which represent significant challenges in reconstruction. We group all daily videos in each location and run *EgoLifter* for each spatial environment. The longest aggregated video in one location (Location 2) contains 2.3 hours of video recording and a total of 170K RGB frames. The dataset demonstrates our method can not only tackle diverse dynamic activities, but also produce scene understanding at large scale in space and time.
- **Ego-Exo4D [10] dataset** is a large and diverse dataset containing over one thousand hours of videos captured simultaneously by egocentric and exocentric cameras. Ego-Exo4D videos capture humans performing a wide range of activities. We qualitatively evaluate *EgoLifter* on the egocentric videos of Ego-Exo4D.

We use the same process for all Project Aria videos. Since Aria glasses use fisheye cameras, we undistort the captured images first before training. We use

Table 1: Quantitative evaluation of 2D instance segmentation (measured in mIoU) and novel view synthesis (measured in PNSR) on the ADT dataset. The evaluations are conducted on the frames in the novel subset of each scene.

Evaluation	mIoU (In-view)			mIoU (Cross-view)			PSNR		
Object set	Static	Dynamic	All	Static	Dynamic	All	Static	Dynamic	All
SAM [20]	54.51	32.77	50.69	-	-	-	-	-	-
Gaussian Grouping [57]	35.68	30.76	34.81	23.79	11.33	21.58	21.29	14.99	19.97
EgoLifter-Static	55.67	**39.61**	52.86	51.29	18.67	45.49	21.37	15.32	20.16
EgoLifter-Deform	54.23	38.62	51.49	51.10	18.02	45.22	21.16	**15.39**	19.93
***EgoLifter* (Ours)**	**58.15**	37.74	**54.57**	**55.27**	**19.14**	**48.84**	**22.14**	14.37	**20.28**

the image formation function $f(\cdot)$ in Eq. (1) to capture the vignetting effect and the radius of valid pixels, according to the specifications of the camera on Aria glasses. We use high-frequency 6DoF trajectories to acquire RGB camera poses and the semi-dense point clouds provided by each dataset through the Project Aria Machine Perception Services (MPS).

Baselines. We compare *EgoLifter* to the following baselines.

- **SAM [20]** masks serve as input to *EgoLifter*. The comparison on segmentation between *EgoLifter* and SAM shows the benefits of multi-view fusion of 2D masks. As we will discuss in Sec. 4.1, SAM only allows prompts from the same image (in-view query), while *EgoLifter* enables segmentation prompts from different views (cross-view query) and 3D segmentation.
- **Gaussian Grouping [57]** also lifts the 2D segmentation masks into 3D Gaussians. Instead of the contrastive loss, Gaussian Grouping uses a video object tracker to associate the masks from different views and employs a linear layer for identity classification. Gaussian Grouping does not handle the dynamic objects in 3D scenes.

Ablations. We further provide two variants of *EgoLifter* in particular to study the impact of reconstruction backbone.

- ***EgoLifter*-Static** disabled the transient prediction network. A vanilla static 3DGS [16] is learned to reconstruct the scene. We use the same method to lift and segment 3D features.
- ***EgoLifter*-Deform** uses a dynamic variant of 3DGS [56] instead of the transient prediction network to handle the dynamics in the scene. Similar to [56], *EgoLifter*-Deform learns a canonical 3DGS and a network to predict the location and shape of each canonical 3D Gaussian at different timestamps.

4.1 Benchmark Setup on ADT

We use the ADT dataset [35] for the quantitative evaluation, We use 16 video sequences from ADT, and the frames of each sequence are split into **seen** and **novel** subsets. The seen subset are used for training and validation, while the novel subset contains a chunk of consecutive frames separate from the seen subset

and is only used for testing. The evaluation on the novel subset reflects the performance on novel views. The objects in each video sequence are also tagged **dynamic** and **static** according to whether they move. Each model is trained and evaluated on one video sequence separately. We evaluate the performance of the query-based open-world 2D instance segmentation and 3D instance detection tasks, as described in Sec. 3.4. For the Gaussian Grouping baseline [57], we use their learned identity encoding for extracting query features and computing similarity maps. Please refer to supplementary material for more details of the evaluation settings, the exact sequence IDs and splits we used.

Open-world 2D instance segmentation. We adopt two settings in terms of query sampling for 2D evaluation, namely **in-view** and **cross-view**. In both settings, a similarity map is computed between the query feature and the rendered feature image. The instance segmentation mask is then obtained by cutting off the similarity map using a threshold that maximizes the IoU with respect to the GT mask, which resembles the process of a human user adjusting the threshold to get the desired object. In the **in-view** setting, the query feature is sampled from one pixel on the rendered feature map in the same camera view. For a fair comparison, SAM [20] in this setup takes in the rendered images from the trained 3DGS and the same query pixel as the segmentation prompt. For each prompt, SAM predicts multiple instance masks at different scales, from which we also use the GT mask to pick one that maximizes the IoU for evaluation. The **cross-view** setting follows the prompt propagation evaluation used in the literature [2,32,39,59]. We randomly sample 5 pixels from the training images (in the seen subset) on the object, and their average feature is used as the query for segmentation on the novel subset. To summarize, the in-view setting evaluates how well features group into objects after being rendered into a feature map, and the cross-view setting evaluates how well the learned feature represents each object instance in 3D and how they generalize to novel views.

Open-world 3D instance detection. For 3D evaluation, we use the same query feature obtained in the above cross-view setting. The similarity map is computed between the query feature and the learned 3D Gaussians, from which a subset of 3D Gaussians are obtained by thresholding, and a 3D bounding box is estimated based on their coordinates. The 3D detection performance is evaluated by the IoU between the estimated and the GT 3D bounding boxes. We also select the threshold that maximizes the IoU with the GT bounding box. We only evaluate the 3D static objects in each scene.

Novel view synthesis. We evaluate the synthesized frames in the novel subset using PSNR metric. We use "All" to indicate full-frame view synthesis. We also separately evaluate the pixels on dynamic and static regions using the provided the 2D ground truth dynamic motion mask.

4.2 Quantitative Results on ADT

The quantitative results are reported in Tab. 1 and 2. As shown in Tab. 1, *EgoLifter* consistently outperforms all other baselines and variants in the reconstruction and segmentation of static object instances in novel views. Since transient

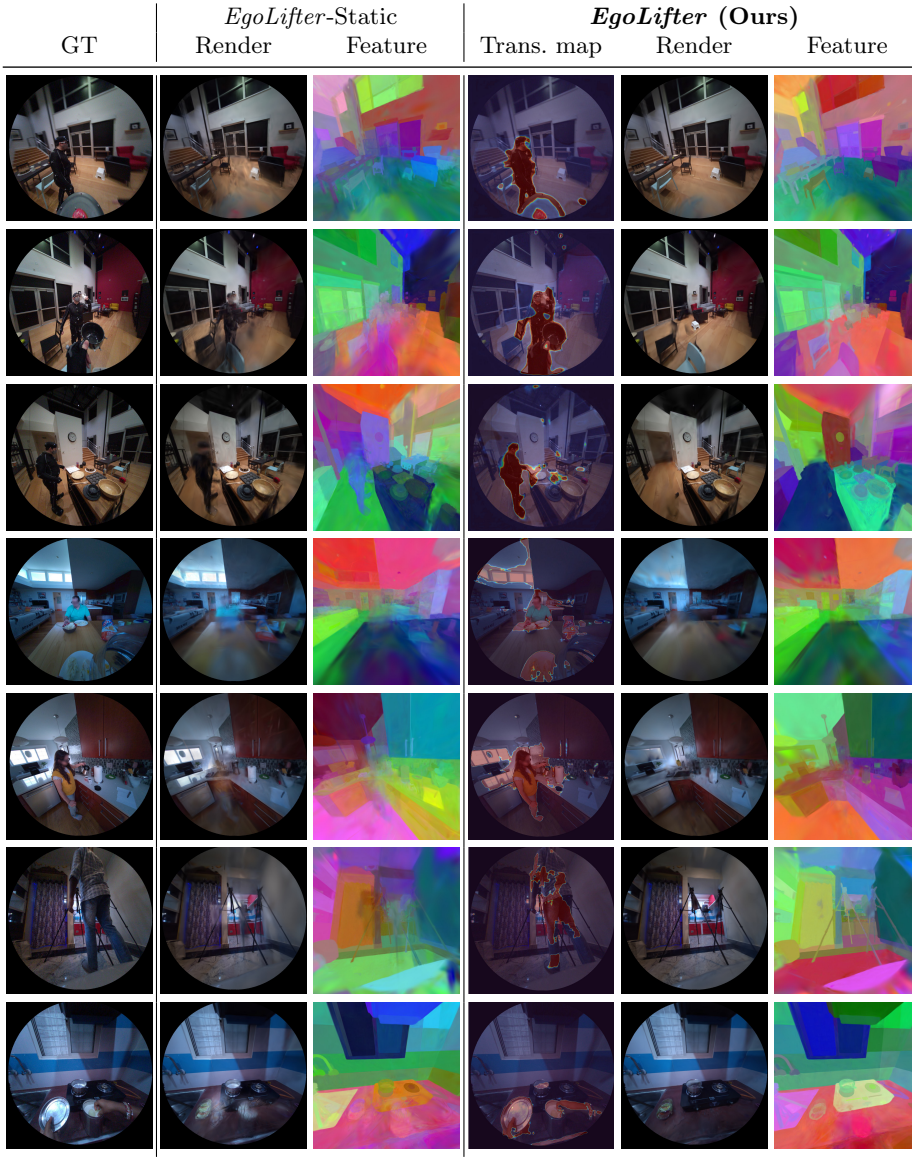

Fig. 3: RGB images and feature maps (colored by PCA) rendered by the *EgoLifter* Static baseline and *EgoLifter*. The predicted transient maps (Trans. map) from *EgoLifter* are also visualized, with red color indicating a high probability of being transient. Note that the baseline puts ghostly floaters on the region of transient objects, but *EgoLifter* filters them out and gives a cleaner reconstruction of both RGB images and feature maps. Rows 1-3 are from ADT, rows 4-5 from AEA, and rows 6-7 from Ego-Exo4D.

Table 2: 3D instance detection performance for the static objects in the ADT dataset.

Method	mIoU
Gaussian Grouping [57]	7.48
EgoLifter-Static	21.10
EgoLifter-Deform	20.58
***EgoLifter* (Ours)**	**23.11**

objects are deliberately filtered out during training, *EgoLifter* has slightly worse performance on dynamic objects, However, the improvements on static objects outweigh the drops on transient ones in egocentric videos and *EgoLifter* still achieves the best overall results in all settings. Similarly, this trend also holds in 3D, and *EgoLifter* has the best 3D detection performance as shown in Tab. 2.

4.3 Qualitative Results on Diverse Egocentric Datasets

In Fig. 3, we visualize the qualitative results on several egocentric datasets [10, 26, 35]. Please refer to supplementary material for the videos rendered by *EgoLifter* with comparison to baselines. As shown in Fig. 3, without transient prediction, *EgoLifter*-Static creates 3D Gaussians to overfit the dynamic observations in some training views. However, since dynamic objects are not geometrically consistent and may be at a different location in other views, these 3D Gaussians become floaters that explain dynamics in a ghostly way, harming both the rendering and segmentation quality. In contrast, *EgoLifter* correctly identifies the dynamic objects in each image using transient prediction and filters them out in the reconstruction. The resulting cleaner reconstruction leads to better results in novel view synthesis and segmentation, as we have already seen quantitatively in Sec. 4.2. We also compare the qualitative results with Gaussian Grouping [57] in Fig. 4, from which we can see that Gaussian Grouping not only struggles with floaters associated with transient objects but also has a less clean feature map even on the static region. We hypothesize this is because our contrastive loss helps learn more cohesive identity features than the classification loss used in [57]. This also explains why *EgoLifter*-Static significantly outperforms Gaussian Grouping in segmentation metrics as shown in Tab. 1 and 2.

4.4 3D Object Extraction and Scene Editing

Based on the features learned by *EgoLifter*, we can decompose a 3D scene into individual 3D objects, by querying or clustering over the feature space. Each extracted 3D object is represented by a set of 3D Gaussians which can be photo-realistically rendered. In Fig. 5, we show the visualization of 3D objects extracted from a scene in the ADT dataset. This can further enable scene editing applications by adding, removing, or transforming these objects over the 3D space. In Fig. 1, we demonstrate one example of background recovery by removing all segmented 3D objects from the table.

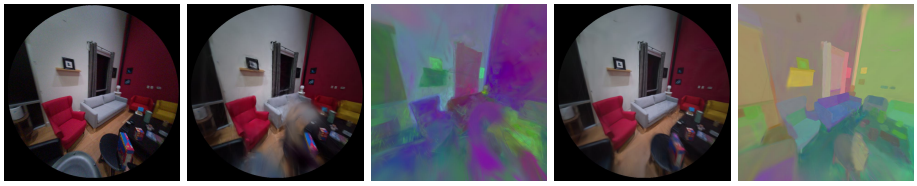

Fig. 4: Rendered images and feature maps (visualised in PCA colors) by Gaussian Grouping [57] and *EgoLifter* (Ours).

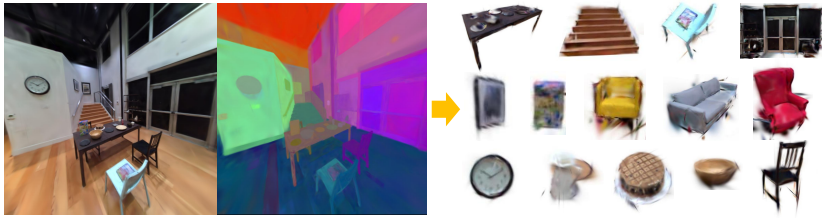

Fig. 5: Individual 3D object can be extracted by querying or clustering over the 3D features from *EgoLifter*. Note object reconstructions are not perfect since each object might be partial observable in the egocentric videos rather than scanned intentionally.

4.5 Limitations

We observe the transient prediction module may mix the regions that are hard to reconstruct with transient objects. As shown in rows (4) and (5) of Fig. 3, the transient prediction module predicts a high probability for pixels on the windows, which have over-exposed pixels that are hard to be reconstructed from LDR images. In this case, *EgoLifter* learns to filter them out to improve reconstruction on that region. Besides, the performance of *EgoLifter* may also be dependent on the underlying 2D segmentation model. *EgoLifter* is not able to segment an object if the 2D model consistently fails on it.

5 Conclusion

We present *EgoLifter*, a novel algorithm that simultaneously solves the 3D reconstruction and open-world segmentation problem for in-the-wild egocentric perception. By lifting the 2D segmentation into 3D Gaussian Splatting, *EgoLifter* achieves strong open-world 2D/3D segmentation performance with no 3D data annotation. To handle the rapid and sparse dynamics in egocentric videos, we employ a transient prediction network to filter out transient objects and get more accurate 3D reconstruction. *EgoLifter* is evaluated on several challenging egocentric datasets and outperforms other existing baselines. The representations obtained by *EgoLifter* can also be used for several downstream tasks like 3D object asset extraction and scene editing, showing great potential for personal wearable devices and AR/VR applications.

Potential Negative Impact: 3D object digitization for egocentric videos in the wild may pose a risk to privacy considerations. Ownership of digital object rights of physical objects is also a challenging and complex topic that will have to be addressed as AR/VR becomes more ubiquitous.

Appendix

A1 Video Qualitative Results

Please refer to the video submitted together with this PDF. The supplementary video contains:

- Videos qualitative results of the multiple applications of *EgoLifter* (corresponding to Fig. 1).
- Video qualitative results on the ADT dataset, comparing *EgoLifter* and its variants (corresponding to Fig. 3).
- Video qualitative results on the ADT dataset, comparing with Gaussian Grouping [57] (corresponding to Fig. 4).
- Video qualitative results on the AEA and Ego-Exo4D datasets. (corresponding to Fig. 3).
- Demonstration video of the interactive visualization and segmentation system.

A2 Experiment Details

A2.1 Image Formation Model for Project Aria

Aria Glasses [7] use a fisheye camera, and thus recorded images have a fisheye distortion and vignette effect, but 3DGS uses a linear camera model and does not have a vignette effect. Therefore we account for these effects in training 3D Gaussian models using the image formation model $f(\cdot)$ in Eq. 1, such that not the raw rendered image but a processed one is used for loss computation. Specifically, we apply an image processing pipeline as shown in Fig. A.1. In the pipeline, the raw recorded images are first rectified to a pinhole camera model using `projectaria_tools`[3], and then multiplied with a valid-pixel mask that removes the pixels that are too far from the image center. The rendered image from 3DGS is multiplied with a vignette mask and also the valid-pixel masks. Then the photometric losses are computed between the processed rendered image and the processed GT image during training. This pipeline models the camera model used in Aria glasses and leads to better 3D reconstruction. Empirically we found that without this pipeline, 3DGS will create a lot of floaters to account for the vignette effect in the reconstruction and significantly harm the results.

[3] Link

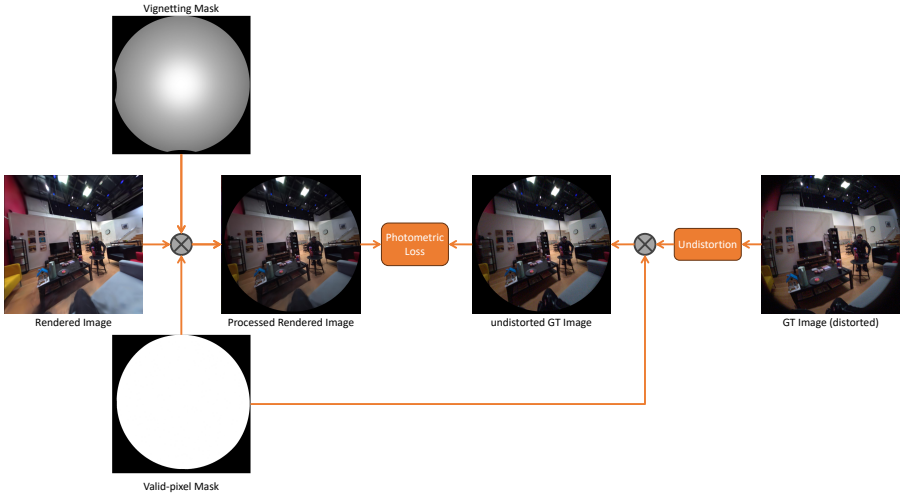

Fig. A.1: Image processing pipeline during training. The ⊗ symbol indicates element-wise multiplication.

A2.2 Additional Training Details

Due to the GPU memory constraint, we sampled at most $|\mathcal{U}| = 4096$ pixels within the valid-pixel mask for computing the contrastive loss in Eq. 2. Note that for *EgoLifter* where the transient prediction is used, the samples are additionally constrained to be pixels with transient probability less than $\delta = 0.5$.

For the segmentation masks generated by SAM, some masks may have overlapped with each other. In our experiments, we discarded the information about overlapping and simply overlaid all masks on the image space to get a one-hot segmentation for each pixel. While making use of these overlapping results leads to interesting applications like hierarchical 3D segmentation as shown in [18,59], this is beyond the scope of *EgoLifter* and we left this for future exploration. The images used for training are of resolution of 1408×1408 and segmentation masks from SAM are in the resolution of 512×512. Therefore, during training, two forward passes are performed. In the first pass, only the RGB image is rendered at the resolution of 1408×1408 and in the second, only the feature map is rendered at 512×512. The losses are computed separately from each pass and summed up for gradient computation. Note that the view-space gradients from both passes are also summed for deciding whether to split 3D Gaussians.

For optimization on the 3D Gaussian models, we adopt the same setting as used in the original implementation [16], in terms of parameters used in the optimizer and scheduler and density control process. The learning rate for the additional per-Gaussian feature vector $\mathbf{f}_i$ is 0.0025, the same as that for updating color $\mathbf{c}_i$. All models are trained for 30,000 iterations on each scene in the ADT dataset, and for 100,000 iterations on scenes in the AEA and Ego-Exo4D

datasets, as these two datasets contain more frames in each scene. In the latter case, the learning rate scheduler and density control schedule are also proportionally extended.

A2.3 ADT Dataset Benchmark

Sequence selection Based on the 218 sequences in the full ADT datasets [35], we filter out the sequences that have too narrow baselines for 3D reconstruction (sequences with name starting with `Lite_release_recognition`) or do not have segmentation annotation on human bodies. From the rest of the sequences, we select 16 sequences for evaluation, where 6 of them contain recordings of Aria glasses from two human users in the scene (sequences with `multiskeleton` in the name), and the rest 10 only have recordings from one user, although there may be multiple two persons in the scene (sequences with `multiuser` in the name). The names of the selected sequences are listed as follows:

```
Apartment_release_multiskeleton_party_seq121
Apartment_release_multiskeleton_party_seq122
Apartment_release_multiskeleton_party_seq123
Apartment_release_multiskeleton_party_seq125
Apartment_release_multiskeleton_party_seq126
Apartment_release_multiskeleton_party_seq127
Apartment_release_multiuser_cook_seq114
Apartment_release_multiuser_meal_seq140
Apartment_release_multiuser_cook_seq143
Apartment_release_multiuser_party_seq140
Apartment_release_multiuser_clean_seq116
Apartment_release_multiuser_meal_seq132
Apartment_release_work_skeleton_seq131
Apartment_release_work_skeleton_seq140
Apartment_release_meal_skeleton_seq136
Apartment_release_decoration_skeleton_seq137
```

Subset Splitting For sequences that only have a recording from one pair of Aria glasses, the first 4/5 of the video is considered as seen views and the rest are considered as novel ones. For sequences that have videos from two pairs, the video from one pair is considered as seen views and the other is considered as novel views. During training, every 1 out of 5 consecutive frames in the seen views are used for validation the rest 4 are used for training. The entire novel subset is hidden from training and solely used for evaluation. For evaluation on 2D instance segmentation, we uniformly sampled at most 200 frames from each subset for fast inference. The objects in each video sequence are also split into dynamic and static subsets, according to whether their GT object positions have changed by over 2cm over the duration of each recording. Humans are always considered dynamic objects.

Table A.1: 2D instance segmentation results (measured in mIoU) and novel view synthesis results (measured in PNSR) on **seen** subsets in the ADT dataset.

Evaluation	mIoU (In-view)			mIoU (Cross-view)			PSNR		
Object set	Static	Dynamic	All	Static	Dynamic	All	Static	Dynamic	All
SAM [20]	62.74	52.48	61.00	-	-	-	-	-	-
Gaussian Grouping [57]	40.86	42.24	41.09	32.26	26.23	31.24	27.97	19.13	25.53
EgoLifter-Static	64.34	**57.71**	**63.21**	62.20	**35.39**	57.64	27.65	19.60	25.64
EgoLifter-Deform	63.33	57.11	62.27	62.24	34.91	57.59	**28.60**	**19.89**	**26.24**
***EgoLifter* (Ours)**	**65.08**	52.12	62.88	**63.65**	33.70	**58.56**	26.86	16.02	23.34

A2.4 Results on ADT Seen Subset

For completeness, we also report the 2D instance segmentation and photometric results on the **seen** subset of ADT in Tab. A.1. Note that the frames used for evaluation in the seen subset are closer to those for training, and therefore these results mostly reflect how well the models overfit to the training viewpoints in each scene, rather than generalize to novel views. As we can see from Tab. A.1, *EgoLifter* outperforms the baselines in segmenting static objects using both in-view and cross-view queries. When both static and dynamic objects are considered (the "All" column), *EgoLifter* still achieves the best results in cross-view, which is a harder setting for open-world segmentation. *EgoLifter* also has the second place in the in-view setting.

References

1. Bhalgat, Y., Laina, I., Henriques, J.F., Zisserman, A., Vedaldi, A.: Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. arXiv preprint arXiv:2306.04633 (2023) 3, 4, 7
2. Cen, J., Zhou, Z., Fang, J., Shen, W., Xie, L., Jiang, D., Zhang, X., Tian, Q.: Segment anything in 3d with nerfs. Advances in Neural Information Processing Systems **36** (2024) 11
3. Chen, Z., Wang, F., Liu, H.: Text-to-3d using gaussian splatting. arXiv preprint arXiv:2309.16585 (2023) 3
4. Crabtree, A., Tolmie, P.: A day in the life of things in the home. In: Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing. pp. 1738–1750 (2016) 2
5. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Scaling egocentric vision: The epic-kitchens dataset. In: Proceedings of the European conference on computer vision (ECCV). pp. 720–736 (2018) 5
6. Duan, Y., Wei, F., Dai, Q., He, Y., Chen, W., Chen, B.: 4d gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. arXiv preprint arXiv:2402.03307 (2024) 4
7. Engel, J., Somasundaram, K., Goesele, M., Sun, A., Gamino, 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. arXiv preprint arXiv:2308.13561 (2023) 5, 15

8. Engelmann, F., Manhardt, F., Niemeyer, M., Tateno, K., Pollefeys, M., Tombari, F.: Open-set 3d scene segmentation with rendered novel views (2023) 4
9. Gao, H., Li, R., Tulsiani, S., Russell, B., Kanazawa, A.: Monocular dynamic view synthesis: A reality check. Advances in Neural Information Processing Systems **35**, 33768–33780 (2022) 5
10. 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) 9, 13
11. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., et al.: Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. arXiv preprint arXiv:2309.16650 (2023) 4, 7
12. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1314–1324 (2019) 9
13. Huang, C., Mees, O., Zeng, A., Burgard, W.: Audio visual language maps for robot navigation. arXiv preprint arXiv:2303.07522 (2023) 4
14. Jatavallabhula, K.M., Kuwajerwala, A., Gu, Q., Omama, M., Iyer, G., Saryazdi, S., Chen, T., Maalouf, A., Li, S., Keetha, N.V., Tewari, A., Tenenbaum, J.B., de Melo, C.M., Krishna, K.M., Paull, L., Shkurti, F., Torralba, A.: ConceptFusion: Open-set multimodal 3d mapping. In: Robotics: Science and Systems (2023). https://doi.org/10.15607/RSS.2023.XIX.066 4
15. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. arXiv preprint arXiv:2312.02126 (2023) 3
16. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (ToG) **42**(4), 1–14 (2023) 2, 3, 5, 6, 9, 10, 16
17. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: LERF: Language embedded radiance fields. In: International Conference on Computer Vision (ICCV) (2023) 4
18. Kim, C.M., Wu, M., Kerr, J., Goldberg, K., Tancik, M., Kanazawa, A.: Garfield: Group anything with radiance fields. arXiv preprint arXiv:2401.09419 (2024) 4, 16
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
20. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. ICCV (2023) 3, 4, 9, 10, 11, 18
21. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. NeurIPS **35**, 23311–23330 (2022) 4

22. Liu, K., Zhan, F., Zhang, J., Xu, M., Yu, Y., Saddik, A.E., Theobalt, C., Xing, E., Lu, S.: 3d open-vocabulary segmentation with foundation models. arXiv preprint arXiv:2305.14093 (2023) 4
23. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023) 4
24. Lu, S., Chang, H., Jing, E.P., Boularias, A., Bekris, K.: OVIR-3d: Open-vocabulary 3d instance retrieval without training on 3d data (2023), https://openreview.net/forum?id=gVBvtRqU1_ 4, 7
25. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. arXiv preprint arXiv:2308.09713 (2023) 4
26. 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.M., Gu, Q., Nardi, R.D., Cheng, S., Saarinen, S., Baiyya, V., Zou, Y., Newcombe, R.A., Engel, J.J., Pan, X., Ren, C.: Aria everyday activities dataset. arXiv preprint arXiv:2402.13349 (2024) 5, 9, 13
27. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7210–7219 (2021) 5
28. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. arXiv preprint arXiv:2312.06741 (2023) 3
29. Mazur, K., Sucar, E., Davison, A.J.: Feature-realistic neural fusion for real-time, open set scene understanding. IEEE (2023) 4
30. McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. The Journal of Open Source Software **2**(11), 205 (2017) 8
31. 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) 4
32. Mirzaei, A., Aumentado-Armstrong, T., Derpanis, K.G., Kelly, J., Brubaker, M.A., Gilitschenski, I., Levinshtein, A.: Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20669–20679 (2023) 11
33. OpenAI: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 4
34. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) 5
35. 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. pp. 20133–20143 (2023) 5, 9, 10, 13, 17
36. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 815–824 (2023) 4
37. Qin, M., Li, W., Zhou, J., Wang, H., Pfister, H.: Langsplat: 3d language gaussian splatting. arXiv preprint arXiv:2312.16084 (2023) 4
38. 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: ICML. PMLR (2021) 4, 5

39. Ren, Z., Agarwala, A., Russell, B., Schwing, A.G., Wang, O.: Neural volumetric object selection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6133–6142 (2022) 11
40. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 4
41. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015) 9
42. Shafiullah, N.M.M., Paxton, C., Pinto, L., Chintala, S., Szlam, A.: Clip-fields: Weakly supervised semantic fields for robotic memory. In: Bekris, K.E., Hauser, K., Herbert, S.L., Yu, J. (eds.) Robotics: Science and Systems (2023). https://doi.org/10.15607/RSS.2023.XIX.074 4
43. Shen, W., Yang, G., Yu, A., Wong, J., Kaelbling, L.P., Isola, P.: Distilled feature fields enable few-shot manipulation (2023), https://openreview.net/forum?id=Rb0nGIt_kh5 4
44. Siddiqui, Y., Porzi, L., Bulò, S.R., Müller, N., Nießner, M., Dai, A., Kontschieder, P.: Panoptic lifting for 3d scene understanding with neural fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9043–9052 (2023) 7
45. Sun, J., Qiu, J., Zheng, C., Tucker, J., Yu, J., Schwager, M.: Aria-nerf: Multimodal egocentric view synthesis. arXiv preprint arXiv:2311.06455 (2023) 5
46. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: Openmask3d: Open-vocabulary 3d instance segmentation. arXiv preprint arXiv:2306.13631 (2023) 4, 7
47. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023) 3
48. Tsagkas, N., Mac Aodha, O., Lu, C.X.: Vl-fields: Towards language-grounded neural implicit spatial representations. arXiv preprint arXiv:2305.12427 (2023) 4
49. Tschernezki, V., Darkhalil, A., Zhu, Z., Fouhey, D., Laina, I., Larlus, D., Damen, D., Vedaldi, A.: Epic fields: Marrying 3d geometry and video understanding. arXiv preprint arXiv:2306.08731 (2023) 5
50. Tschernezki, V., Laina, I., Larlus, D., Vedaldi, A.: Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In: International Conference on 3D Vision (3DV). IEEE (2022) 4
51. Tschernezki, V., Larlus, D., Vedaldi, A.: Neuraldiff: Segmenting 3d objects that move in egocentric videos. In: 2021 International Conference on 3D Vision (3DV). pp. 910–919. IEEE (2021) 5
52. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. arXiv preprint arXiv:2310.08528 (2023) 4
53. Yan, C., Qu, D., Wang, D., Xu, D., Wang, Z., Zhao, B., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. arXiv preprint arXiv:2311.11700 (2023) 3
54. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians for modeling dynamic urban scenes. arXiv preprint arXiv:2401.01339 (2024) 3
55. Yang, Z., Yang, H., Pan, Z., Zhu, X., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint arXiv:2310.10642 (2023) 4

56. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. arXiv preprint arXiv:2309.13101 (2023) 4, 10
57. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023) 4, 7, 10, 11, 13, 14, 15, 18
58. Yi, T., Fang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., Wang, X.: Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. arXiv preprint arXiv:2310.08529 (2023) 3
59. Ying, H., Yin, Y., Zhang, J., Wang, F., Yu, T., Huang, R., Fang, L.: Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. arXiv preprint arXiv:2311.11666 (2023) 4, 11, 16
60. Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., Yang, M.H.: Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. arXiv preprint arXiv:2312.07920 (2023) 3
61. Zuo, X., Samangouei, P., Zhou, Y., Di, Y., Li, M.: Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. arXiv preprint arXiv:2401.01970 (2024) 4