Easi3R: Estimating Disentangled Motion from DUSt3R Without Training

Xingyu Chen[1,2] Yue Chen[1,2] Yuliang Xiu[2,3] Andreas Geiger[4] Anpei Chen[2,4]

[1]Zhejiang University [2]Westlake University [3]Max Planck Institute for Intelligent Systems
[4]University of Tübingen, Tübingen AI Center

easi3r.github.io

Abstract

Recent advances in DUSt3R have enabled robust estimation of dense point clouds and camera parameters of static scenes, leveraging Transformer network architectures and direct supervision on large-scale 3D datasets. In contrast, the limited scale and diversity of available 4D datasets present a major bottleneck for training a highly generalizable 4D model. This constraint has driven conventional 4D methods to fine-tune 3D models on scalable dynamic video data with additional geometric priors such as optical flow and depths. In this work, we take an opposite path and introduce Easi3R, a simple yet efficient training-free method for 4D reconstruction. Our approach applies attention adaptation during inference, eliminating the need for from-scratch pre-training or network fine-tuning. We find that the attention layers in DUSt3R inherently encode rich information about camera and object motion. By carefully disentangling these attention maps, we achieve accurate dynamic region segmentation, camera pose estimation, and 4D dense point map reconstruction. Extensive experiments on real-world dynamic videos demonstrate that our lightweight attention adaptation significantly outperforms previous state-of-the-art methods that are trained or fine-tuned on extensive dynamic datasets.

1. Introduction

Recovering geometry and motions from dynamic image collections is still a fundamental challenge in computer vision, with broad downstream applications in novel view synthesis, AR/VR, autonomous navigation, and robotics. The literature commonly identifies this problem as Structure-from-Motion (SfM) and has been the core focus in 3D vision over decades, yielding mature algorithms that perform well under stationary conditions and wide baselines. However, these algorithms often fail when applied to dynamic video input.

The main reason for the accuracy and robustness gap between static and dynamic SfM is object dynamics, a com-
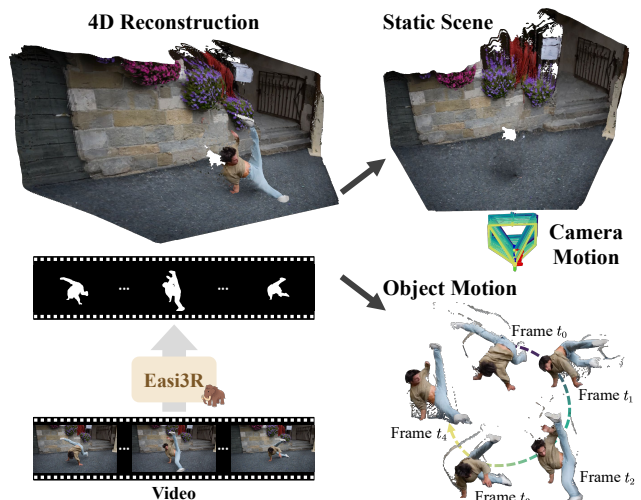

Figure 1. We present Easi3R, a training-free, plug-and-play approach that efficiently disentangles object and camera motion, enabling the adaptation of DUSt3R for 4D reconstruction.

mon component in real-world videos. Moving objects violate fundamental assumptions of homography and epipolar consistency in traditional SfM methods [37, 48]. In addition, in dynamic videos, where camera and object motions are often entangled, these methods struggle to disentangle the two motions, often causing the motion with rich texture to mainly contribute to camera pose estimation erroneously. Recent efforts, such as MonST3R [73] and CUT3R [63], have made strides to address these challenges. However, their success is based on extensive training data [19, 25, 63, 68, 73] or task-specific prior models [22, 25, 73, 74], such as the depth, optical flow, and object mask estimators. These limitations motivate us to innovate further to minimize the gap between static and dynamic reconstruction.

We ask ourselves if there are lessons from human perception that can be used as design principles for dynamic 4D reconstruction: Human beings are capable of perceiving body motion and the structure of the scene, identifying

dynamic objects, and disentangling ego-motion from object motion through the inherent attention mechanisms of the brain [58]. Yet, the learning process rarely relies on explicit dynamic labels.

We observe that DUSt3R implicitly learned a similar mechanism, and based on this, we introduce Easi3R, a training-free method to achieve dynamic object segmentation, dense point map reconstruction, and robust camera pose estimation from dynamic videos, as shown in Figure 1. DUSt3R uses attention layers at its core, taking two image features as input and producing pixel-aligned point maps as output. These attention layers are trained to directly predict pointmaps in the reference view coordinate space, implicitly matching the image features between the input views [4] and estimating the rigid view transformation in the feature space. In practice, performance drops significantly when processing pairs with object dynamics [73], as shown in Figure 2. By analyzing the attention maps in the transformer layers, we find that regions with less texture, under-observed, and dynamic objects can yield low attention values. Therefore, we propose a simple yet effective decomposition strategy to isolate the above components, which enables long-horizon dynamic object detection and segmentation. With this segmentation, we perform a second inference pass by applying a re-weighting [17] in the cross-attention layers, enabling robust dynamic 4D reconstruction and camera motion recovery without fine-tuning on a dynamic dataset, all at minimal additional cost to DUSt3R.

Despite its simplicity, we demonstrate that our inference-time scaling approach for 4D reconstruction is remarkably robust and accurate on in-the-wild casual dynamic videos. We evaluate our Easi3R adaptation on the DUSt3R and MonST3R backbones in three task categories: camera pose estimation, dynamic object segmentation, and pointcloud reconstruction in dynamic scenes. Easi3R performs surprisingly well across a wide range of datasets, even surpassing concurrent methods (e.g., CUT3R [63], MonST3R [73], and DAS3R [68]) that are trained on dynamic datasets.

2. Related Work

SfM and SLAM. Structure-from-Motion (SfM) [2, 41, 42, 48, 51, 52] and Simultaneous Localization and Mapping (SLAM) [9, 13, 32, 34] have long been the foundation for 3D structure and camera pose estimation. These methods are done by associating 2D correspondences [5, 10, 28, 32, 47] or minimizing photometric errors [12, 13], followed by bundle adjustment (BA) [3, 6, 55, 57, 59, 62] to refine structure and motion estimates. Although highly effective with dense input, these approaches often struggle with limited camera parallax or ill-posed conditions, leading to performance degeneracy. To overcome these limitations, DUSt3R [64] introduced a learning-based approach that di-

rectly predicts two pointmaps from an image pair in the coordinate space of the first view. This approach inherently matches image features and rigid body view transformation. By leveraging a Transformer-based architecture [11] and direct point supervision on large-scale 3D datasets, DUSt3R establishes a robust Multi-View Stereo (MVS) foundation model. However, DUSt3R and the follow-up methods [27, 33, 56, 61] assume primarily static scenes, which can lead to significant performance degradation when dealing with videos with dynamic objects.

Pose-free Dynamic Scene Reconstruction. Modifications to SLAM for dynamic scenes involve robust pose estimation to mitigate moving object interference, dynamic map management for updating changing environments, including techniques like semantic segmentation [72], optical flows [75], and enhance SLAM's resilience in dynamic scenarios. Another line of work focuses on estimating stable video depth by incorporating geometric constraints [29] and generative priors [18, 49]. These methods enhance monocular depth accuracy but lack global point cloud lifting due to missing camera intrinsics and poses. For joint pose and depth estimation, optimization-based methods such as CasualSAM [74] fine-tune a depth network [45] at test time using pre-computed optical flow [66]. Robust-CVD [22] refines pre-computed depth [45] and camera pose by leveraging masked optical flow [16, 66] to improve stability in occluded and moving regions. Concurrently, MegaSaM [25] further enhances pose and depth accuracy by integrating DROID-SLAM [57], optical flow [66], and depth initializations from [40, 71], achieving state-of-the-art results. Alternatively, point-map-based approaches like MonST3R [73] extend DUSt3R to dynamic scenes by fine-tuning with dynamic datasets and incorporating optical flow [66] to infer dynamic object segmentation. DAS3R trains a DPT [44] on top of MonST3R, enabling feedforward segmentation estimation. CUT3R [63] fine-tunes MASt3R [24] on both static and dynamic datasets, achieving feedforward reconstruction but without predicting dynamic object segmentation, thereby entangling the static scene with dynamic objects. Although effective, these methods require costly training on diverse motion patterns to generalize well.

In contrast, we take an opposite path, exploring a training-free and plug-in-play adaptation that enhances the generalization of DUSt3R variants for dynamic scene reconstruction. Our method requires no fine-tuning and comes at almost no additional cost, offering a scalable and efficient alternative for handling real-world dynamic videos.

Motion Segmentation. Motion segmentation aims to predict dynamic object masks from video inputs. Classical approaches generally rely on optical flow estimation [26, 31, 67, 70] and point tracking [7, 21, 35, 50, 69] to distinguish moving objects from the background. Being trained solely on 2D data, they often struggle with occlusions and

distinguishing between object and camera motion. To improve robustness, RoMo [15] incorporates epipolar geometry [30] and SAM2 [46] to better disambiguate object motion from camera motion. While RoMo successfully combines COLMAP [48] for accurate camera calibration, it focuses primarily on removing dynamic objects and reconstructing static scene elements only. For the complete 4D reconstruction, MonST3R [73] integrates optical flow [66] with estimated pose and depth to predict dynamic object segmentation. DAS3R builds on MonST3R and trains a DPT [44] for segmentation inference. Using this segmentation, they align static components globally while preserving dynamic point clouds from each frame, enabling temporally consistent reconstruction of moving objects.

In this work, we discover that dynamic segmentation can be extracted from pre-trained 3D reconstruction models like DUSt3R. We propose a simple, yet robust strategy to isolate this information from the attention layers, without the need for optical flow or pre-training on segmentation datasets.

3. Method

Given a casually captured video sequence $\{I^t \in \mathbb{R}^{W \times H \times 3}\}_{t=1}^T$, our goal is to estimate object and camera movements, as well as the canonical point clouds present in the input video. Object motion is represented as the segmentation sequence $\mathbf{M}^t$, camera motion as the extrinsic and intrinsic pose sequences $\mathbf{P}^t, \mathbf{K}^t$, and point clouds $\mathcal{X}^*$. First, we formulate how the DUSt3R model handles videos (Section 3.1). Next, we explore the mechanisms of attention aggregation in spatial and temporal dimensions (Section 3.2). Finally, we introduce how aggregated cross-attention maps can be leveraged to decompose dynamic object segmentation (Section 3.3), which in turn helps re-weight attention values for robust point cloud and camera pose reconstruction (Section 3.4).

3.1. DUSt3R with Dynamic Video

DUSt3R is designed for pose-free reconstruction, taking two RGB images - I^a, I^b, where $a, b \in [1, \ldots, T]$ - as input and output two pointmaps in the *reference* view coordinate space, $X^{a \to a}, X^{b \to a} \in \mathbb{R}^{W \times H \times 3}$:

$$X^{a \to a}, X^{b \to a} = \text{DUSt3R}(I^a, I^b) \quad (1)$$

Here, $X^{b \to a}$ denotes the pointmap of input I^b predicted in the view a coordinate space. In particular, both pointmaps are expressed in the *reference* view coordinate, i.e., view a in this example.

Given multi-view images, DUSt3R processes them in pairs and globally aligns the pairwise predictions into a joint coordinate space using a connectivity graph across all views. However, this approach introduces computational redundancy for video sequences, as the view connectivity is largely known. Instead, we process videos using a
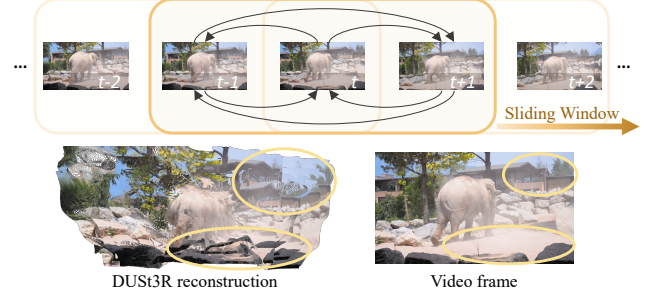

Figure 2. **DUSt3R with Dynamic Video.** We process videos using a sliding window and infer the DUSt3R network pairwise. Reconstruction degrades with misalignment when dynamic objects occupy a considerable portion of the frames.

sliding temporal window and infer the network for pair set $\varepsilon^t = \{(a, b) \mid a, b \in [t - \frac{n-1}{2}, \ldots, t + \frac{n-1}{2}], a \neq b\}$ within the symmetric temporal window of size n centered at time t, as illustrated in the top row of Figure 2.

With pairwise predictions, it recovers globally aligned pointmaps $\{\mathcal{X}^t \in \mathbb{R}^{W \times H \times 3}\}_{t=1}^T$ by optimizing the transformation $\mathbf{P}_i^t \in \mathbb{R}^{3 \times 4}$ from the coordinate space of each pair to the world coordinate, and a scale factor $\mathbf{s}_i^t$ for the i-th pair within the set of pairs ε^t:

$$\mathcal{X}^* = \underset{\mathcal{X}, \mathbf{P}, \mathbf{s}}{\arg \min} \sum_{t \in T} \sum_{i \in \varepsilon^t} \|\mathcal{X}^a - \mathbf{s}_i^t \mathbf{P}_i^t X^{a \to a}\|_1$$
$$+ \|\mathcal{X}^b - \mathbf{s}_i^t \mathbf{P}_i^t X^{b \to a}\|_1 \quad (2)$$

Note that the above optimization process assumes a reliable pairwise reconstruction and that global content can be registered by minimizing the linear equations in Eq. 2. However, since DUSt3R are learned from RGB-D images of static scenes, dynamic objects disrupt the learned epipolar matching policy. As a result, registration fails when dynamic content occupies a considerable portion of pixels, as shown in Figure 2.

3.2. Secrets Behind DUSt3R

We now examine the network architecture to identify the components that cause failures for dynamic video input. As shown in Figure 3, DUSt3R consists of two branches: the top one for the reference image I^a and the bottom one for the source image I^b. The two input images are first processed by a weight-sharing ViT encoder [11], producing token representations $\mathbf{F}_0^a = \text{Encoder}(I^a)$ and $\mathbf{F}_0^b = \text{Encoder}(I^b)$. Next, two decoders, composed of a sequence of decoder blocks, exchange information both within and between views. In each block, self-attention is applied to the token outputs from the previous block, while cross-attention is performed using the corresponding block outputs from the other branch.
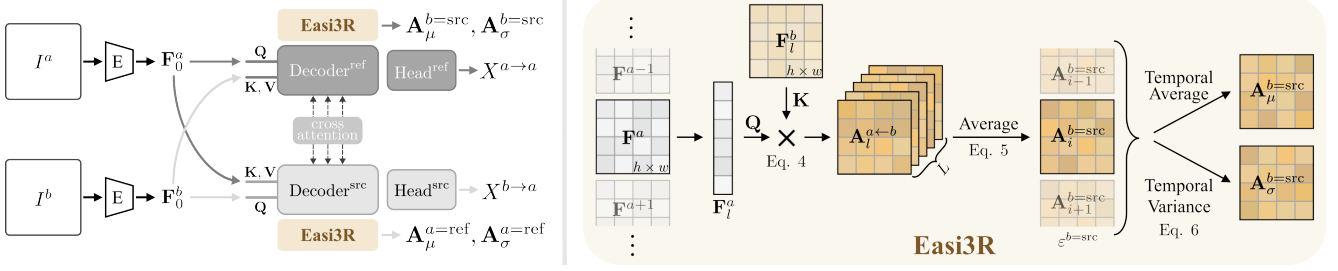
Figure 3. **DUSt3R** and our **Easi3R** adaptation. DUSt3R encodes two images I^a, I^b into feature tokens $\mathbf{F}_0^a, \mathbf{F}_0^b$, which are then decoded into point maps in the reference view coordinate space using two decoders. Our Easi3R aggregates the cross-attention maps from the decoders, producing four semantically meaningful maps: $\mathbf{A}_\mu^{b=\mathrm{src}}, \mathbf{A}_\sigma^{b=\mathrm{src}}, \mathbf{A}_\mu^{a=\mathrm{ref}}, \mathbf{A}_\sigma^{a=\mathrm{ref}}$. These maps are then used for a second inference pass to enhance reconstruction quality.
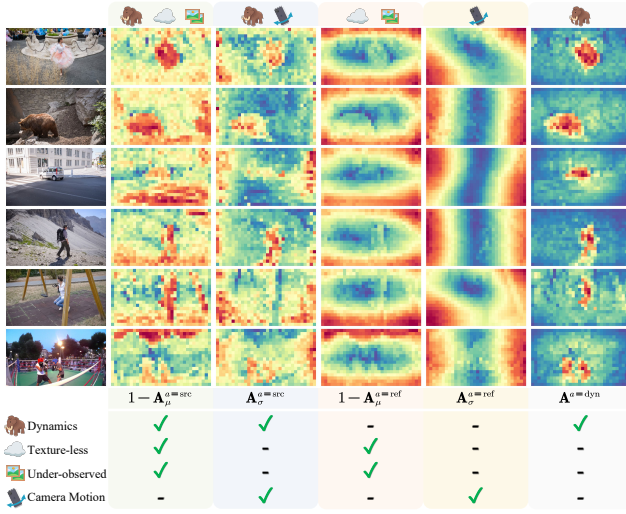

Figure 4. **Visualization for Cross-Attention Maps.** We color the *normalized* values of attention maps, ranging from one to zero. We highlight the patterns captured by each type of attention map using relatively high values. For a more detailed demonstration, we invite reviewers to visit our webpage under easi3r.github.io.

$$\mathbf{F}_l^a = \mathrm{DecoderBlock}_l^{\mathrm{ref}}\left(\mathbf{F}_{l-1}^a, \mathbf{F}_{l-1}^b\right)$$
$$\mathbf{F}_l^b = \mathrm{DecoderBlock}_l^{\mathrm{src}}\left(\mathbf{F}_{l-1}^b, \mathbf{F}_{l-1}^a\right) \quad (3)$$

where $l = 1, \ldots, L$ is the block index. Using the feature tokens, two regression heads produce pointmap predictions: $X^{a\rightarrow a} = \mathrm{Head}^{\mathrm{ref}}\left(\mathbf{F}_0^a, \ldots, \mathbf{F}_L^a\right)$ and $X^{b\rightarrow a} = \mathrm{Head}^{\mathrm{src}}\left(\mathbf{F}_0^b, \ldots, \mathbf{F}_L^b\right)$, respectively. The blocks are trained by minimizing the Euclidean distance between the predicted and ground-truth pointmaps.

Observation. Our key insight is that DUSt3R implicitly learns rigid view transformations through its cross-attention layers, assigning low attention values to tokens that violate epipolar geometry constraints, such as texture-less, under-observed, and dynamic regions. By aggregating cross-attention outputs across spatial and temporal dimensions,

we extract motions from the attention layers.

Spatial attention maps. As illustrated in the Figure 3 (left), the image features $\mathbf{F}$ are projected into a query matrix for their respective branch with $\mathbf{Q} = \ell_{\mathbf{Q}}(\mathbf{F}) \in \mathbb{R}^{(h\times w)\times c}$, while also serving as a key and value matrix for the other matrices, $\mathbf{K} = \ell_{\mathbf{K}}(\mathbf{F}) \in \mathbb{R}^{(h\times w)\times c}$, where c is the feature dimenson. The projections are obtained using trainable linear functions $\ell_{\mathbf{Q}}(\cdot), \ell_{\mathbf{K}}(\cdot)$. As illumistracted in the right side of Figure 3, this results in the cross-attention map:

$$\mathbf{A}_l^{a\leftarrow b} = \mathbf{Q}_l^a \mathbf{K}_l^{b\,T}/\sqrt{c}, \quad \mathbf{A}_l^{b\leftarrow a} = \mathbf{Q}_l^b \mathbf{K}_l^{a\,T}/\sqrt{c} \quad (4)$$

in which the cross-attention map $\mathbf{A}_l^{a\leftarrow b}$, $\mathbf{A}_l^{b\leftarrow a} \in \mathbb{R}^{(h\times w)\times h\times w}$ are used to guide the warping of the value matrix $\mathbf{V} = \ell_{\mathbf{V}}(\mathbf{F}) \in \mathbb{R}^{(h\times w)\times c}$, and the cross-attention output in the reference view branch is given by $\mathrm{softmax}(\mathbf{A}_l^{a\leftarrow b})\mathbf{V}^b$. Intuitively, the attention map $\mathbf{A}_l^{a\leftarrow b}$ determines how the information is aggregated from the view b to the view a in the l-th decoder block.

To evaluate the spatial contribution of each token in view b to all tokens in view a, we average the attention values between different tokens along the query and layer dimensions. This is given by,

$$\mathbf{A}^{b=\mathrm{src}} = \sum_l \sum_x \mathbf{A}_l^{a\leftarrow b}(x, y, z)/(L \times h \times w)$$
$$\mathbf{A}^{a=\mathrm{ref}} = \sum_l \sum_x \mathbf{A}_l^{b\leftarrow a}(x, y, z)/(L \times h \times w) \quad (5)$$

where $\mathbf{A}^{b=\mathrm{src}}, \mathbf{A}^{a=\mathrm{ref}} \in \mathbb{R}^{h\times w}$, representing the averaged attention maps, capturing the overall influence of tokens from one view to another across all decoder layers. i.e., $\mathbf{A}^{b=\mathrm{src}}$ denotes the overall contribution of view b to the reference view when it serve as source view.

Temporal attention maps. In the following, we extend the above single-pair formulation to multiple pairs to explore their temporal attention correlations. For a specific frame I^t, it pairs with multiple frames, resulting in $2(n-1)$ attention maps per frame. As shown in the upper row of Figure 2,

window size of 3 corresponds to 4 pairs. To aggregate the pairwise cross-attention maps temporally, we compute the mean and variance over pairs that the view serves as source and reference:

$$\mathbf{A}_\mu^{b=\text{src}} = \text{Mean}(\mathbf{A}_i^{b=\text{src}}), \ \mathbf{A}_\sigma^{b=\text{src}} = \text{Std}(\mathbf{A}_i^{b=\text{src}}) \quad (6)$$

where $i \in \varepsilon^{b=\text{src}}$ and

$$\varepsilon^{b=\text{src}} = \{(a, b) \mid \text{src} = b, a \in [t - n, \dots, t + n], a \neq b\} \quad (7)$$

Similarly, we compute $\mathbf{A}_\mu^{b=\text{ref}}$ and $\mathbf{A}_\sigma^{b=\text{ref}}$ for the set of pairs where view a acts as the source view:

$$\varepsilon^{b=\text{ref}} = \{(b, a) \mid \text{ref} = b, a \in [t - n, \dots, t + n], a \neq b\} \quad (8)$$

Secrets. We visualize the aggregated temporal cross-attention maps in Figure 4. Recall that DUSt3R infers pointmaps from two images in the reference view coordinate frame, implicitly aligning points from the source view to the reference view.

(i) The reference view serves as the registration standard and is assumed to be static. As a result, the average attention map $\mathbf{A}_\mu^{a=\text{ref}}$ tends to be smooth, with texture-less regions (e.g., ground, sky, swing supports, boxing fences) and under-observed areas (e.g., image boundary) naturally exhibiting low attention values, since DUSt3R believes that they are less useful for registration. These regions can be highlighted and extracted using $(1 - \mathbf{A}_\mu^{a=\text{ref}})$, as shown in the "☁🖼" column of Figure 4.

(ii) By calculating the standard deviation of $(1 - \mathbf{A}_\mu^{a=\text{ref}})$ between neighboring frames, we have $\mathbf{A}_\sigma^{a=\text{ref}}$, e.g., the column "📱", representing the changes of the token contribution in the image coordinate space. Pixels perpendicular to the direction of motion generally share similar pixel flow speeds, resulting in consistent deviations that allow us to infer camera motion from the attention pattern. For example, in the "Walking Man" case in the fourth row of the Figure 4, with the camera motion from left to right, we can observe pixels along a column sharing similar attention values.

(iii) Similar to the reference view, we also compute the average invert attention map in the source view, $1 - \mathbf{A}_\mu^{a=\text{src}}$. As shown in the "🐘☁🖼" column, the result not only indicates areas with less texture and underobserved areas but also highlights dynamic objects because they violate the rigid body transformation prior which DUSt3R has learned from the 3D dataset, resulting in low $\mathbf{A}_\mu^{a=\text{src}}$ values.

(iv) The column "🐘📱" shows the standard deviation of the source view attention map, $\mathbf{A}_\sigma^{a=\text{src}}$. It highlights both camera and object motion, as the attention of these areas continuously changes over time, leading to high deviation in image space.

3.3. Dynamic Object Segmentation

By observing the compositional properties of the derived cross-attention maps, we propose extracting dynamic object segmentation for free, which provides a key for bridging static and dynamic scene reconstruction. To this end, we identify attention activations attributed to object motion. We infer the dynamic attention map for frame a by computing the joint attention of the first two attention columns in Figure 4 using the element-wise product: $(1 - \mathbf{A}_\mu^{a=\text{src}}) \cdot \mathbf{A}_\sigma^{a=\text{src}}$. To further mitigate the effects of texture-less regions, under-observed areas, and camera motion (as shown in the third and fourth columns of Figure 4), we incorporate the outputs with their inverse attention, resulting in the final formula:

$$\mathbf{A}^{a=\text{dyn}} = (1 - \mathbf{A}_\mu^{a=\text{src}}) \cdot \mathbf{A}_\sigma^{a=\text{src}} \cdot \mathbf{A}_\mu^{a=\text{ref}} \cdot (1 - \mathbf{A}_\sigma^{a=\text{ref}}) \quad (9)$$

we then obtain per-frame dynamic object segmentation $\mathbf{M}^t = \left[\mathbf{A}^{t=\text{dyn}} > \alpha\right]$ using Eq. 9 and $\mathbf{M}^t \in \mathbb{R}^{h \times w}$, α denotes a pre-defined attention threshold and the $[\cdot]$ is the Iverson bracket. Note that the segmentation is processed frame by frame. To enhance temporal consistency, we apply a feature clustering method that fuses information across all frames; see the supplementary materials for more details.

3.4. 4D Reconstruction

With dynamic object segmentation, the most intuitive way to adapt static models to dynamic scenes is by masking out dynamic objects during inference at both the image and token levels. This can be done by replacing dynamic regions with black pixels in the image and substituting the corresponding tokens with mask tokens. In practice, this approach significantly degrades reconstruction performance [73], mainly because black pixels and mask tokens lead to out-of-distribution input. This motivates us to apply masking directly within the attention layers instead of modifying the input images.

Attention re-weighting. We propose to modify the cross-attention maps by weakening the attention values associated with dynamic regions. To achieve this, we perform a second inference pass through the network, masking the attention map for assigned dynamic regions. This results in zero attention for those regions while keeping the rest of the attention maps unchanged:

$$\text{softmax}(\tilde{\mathbf{A}}_l^{a \leftarrow b}) = \begin{cases} 0 & \text{if } \mathbf{M}^{a \leftarrow b} \\ \text{softmax}(\mathbf{A}_l^{a \leftarrow b}) & \text{otherwise} \end{cases} \quad (10)$$

here, $\mathbf{M}^{a \leftarrow b} = (1 - \mathbf{M}^a) \otimes \mathbf{M}^{b^T}$, where $\mathbf{M}^{a \leftarrow b} \in \mathbb{R}^{(h \times w) \times (h \times w)}$ and $\otimes$ denotes outer product. This results in tokens from dynamic regions in view b that do not contribute to static regions in view a. It is important to note that re-weighting is applied only to the reference view decoder, as source view requires a static reference (i.e., the

reference view), as described in the secret (i). To achieve this, the source view decoder must perform cross-attention with all tokens from the reference view. Re-weighting dynamic attention on both branches could result in the loss of static standard, leading to noisy outputs. We conducted an ablation study on this insight.

Global alignment. We align the predicted pointmaps from the sliding windows with the global world coordinate using Eq. 2. Moreover, thanks to dynamic region segmentation, our method also supports segmentation-aware global alignment with optical flow. In particular, we incorporate a reprojection loss to ensure that the projected point flow remains consistent with the optical flow estimation [66]. Specifically, given an image pair (a, b), we compute the camera motion from frame a to frame b, denoted by $\hat{\mathcal{F}}^{a \rightarrow b}$, by projecting the global point map $\mathcal{X}^b$ from camera $(\mathbf{P}^a, \mathbf{K}^a)$ to camera $(\mathbf{P}^b, \mathbf{K}^b)$. We then enforce the consistency between the computed flow and the estimated optical flow $\mathcal{F}^{a \rightarrow b}$ in static regions $(1 - \mathbf{M}^t)$:

$$
\mathcal{L}_{\text{flow}} = \sum_{t \in T} \sum_{i \in \varepsilon^t} (1 - \mathbf{M}^a) \cdot \|\hat{\mathcal{F}}_i^{a \rightarrow b} - \mathcal{F}_i^{a \rightarrow b}\|_1
$$
$$
+ (1 - \mathbf{M}^b) \cdot \|\hat{\mathcal{F}}_i^{b \rightarrow a} - \mathcal{F}_i^{b \rightarrow a}\|_1 \quad (11)
$$

where $\cdot$ indicates element-wise product. By incorporating flow constraint into the optimization process in Eq. 2, we achieve a more robust output in terms of global pointmaps $\mathcal{X}^*$ and pose sequences $\mathbf{P}^t, \mathbf{K}^t$. Note that this term is used optionally to ensure a fair comparison with the baseline that does not incorporate the flow-estimation model.

4. Experiments

We evaluate our method in a variety of tasks, including dynamic object segmentation (Section 4.1), camera pose estimation (Section 4.2) and 4D reconstruction (Section 4.3). We performed ablation studies in supplementary.

Baselines. We compare Easi3R with state-of-the-art pose free 4D reconstruction method, including DUSt3R [64], MonST3R [73], DAS3R [68], and CUT3R [63]. Among these works, the latter three are concurrent works that also aim to extend DUSt3R to handle dynamic videos, but take a different approach by fine-tuning on dynamic datasets, such as [20, 54, 65], and optimization under the supervision of optical flow [66]. Unlike previous work, our method performs a second inference pass on top of the pre-trained DUSt3R or MonST3R model without requiring fine-tuning or optimization on additional data.

4.1. Dynamic Object Motion

We represent object motion as a segmentation sequence and evaluate performance on the video object segmenta-
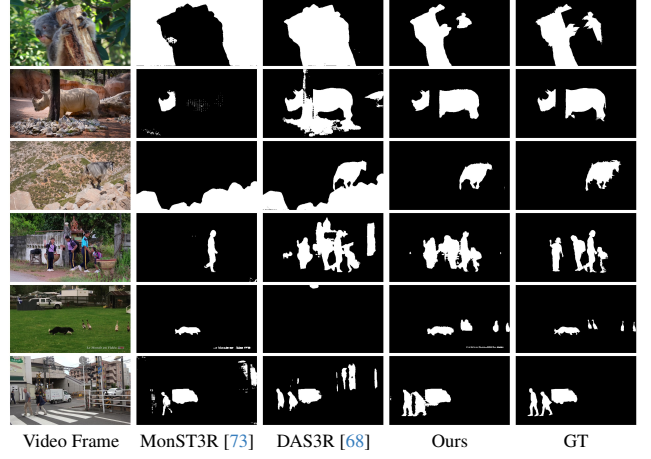

Figure 5. **Qualitative Results of Dynamic Object Segmentation.** "Ours" refers to the Easi3R$_{\text{monst3r}}$ setting. Here, we present the enhanced setting, where outputs from different methods serve as prompts and are used with SAM2 [46] for mask inference.

Table 1. **Dynamic Object Segmentation** on the DAVIS dataset. The best and second best results are **bold** and underlined, respectively. Easi3R $_{\text{dust3r/monst3r}}$ denotes the Easi3R experiment with the backbones of MonST3R/DUSt3R.

Method	Flow	DAVIS-16				DAVIS-17				DAVIS-all			
		w/o SAM2		w/ SAM2		w/o SAM2		w/ SAM2		w/o SAM2		w/ SAM2	
		JM↑	JR↑	JM↑	JR↑	JM↑	JR↑	JM↑	JR↑	JM↑	JR↑	JM↑	JR↑
DUSt3R [64]	✓	42.1	45.7	58.5	63.4	35.2	35.3	48.7	50.2	35.9	34.0	47.6	48.7
MonST3R [73]	✓	40.9	42.2	64.3	73.3	38.6	38.2	56.4	59.6	36.7	34.3	51.9	54.1
DAS3R [68]	✗	41.6	39.0	54.2	55.8	43.5	42.1	57.4	61.3	43.4	38.7	53.9	54.8
Easi3R$_{\text{dust3r}}$	✗	53.1	60.4	67.9	71.4	49.0	56.4	60.1	65.3	44.5	49.6	54.7	60.6
Easi3R$_{\text{monst3r}}$	✗	**57.7**	**71.6**	**70.7**	**79.9**	**56.5**	**68.6**	**67.9**	**76.1**	**53.0**	**63.4**	**63.1**	**72.6**

tion benchmark DAVIS-16 [39], more challenging DAVIS-17 [43], and DAVIS-all. We present two experiment settings: direct evaluation of network outputs and an enhanced setting where outputs serve as prompts for SAM2 [46], improving results. These settings are denoted as w/ and w/o SAM2 in Table 1. Following DAVIS [39], we evaluate performance using IoU mean (JM) and IoU recall (JR) metrics. Since DUSt3R originally does not support dynamic object segmentation, we extend it as a baseline by incorporating the flow-guided segmentation as MonST3R. By applying our attention-guided decomposition, both DUSt3R and MonST3R show improved segmentation, without the need for flow, even surpassing DAS3R, which is explicitly trained on dynamic mask labels.

Qualitative Results. Figure 5 presents the qualitative comparison between our method and existing approaches. Since MonST3R relies on optical flow estimation, it struggles in textureless regions, failing to disentangle dynamic objects from the background (e.g., koala, rhino, sheep). On the other hand, DAS3R learns a mask head for dynamic segmentation but tends to over-segment in most cases. Our

Table 2. **Benefits of Easi3R on Camera Pose Estimation** on the DyCheck, ADT and TUM-dynamics datasets. The best and second best results are **bold** and <u>underlined</u>, respectively. Easi3R $_{dust3r/monst3r}$ denotes the Easi3R experiment with the backbones of MonST3R/DUSt3R.

Method	Flow	DyCheck			ADT			TUM-dynamics		
		ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓
DUSt3R [64]	✗	0.035	0.030	2.323	<u>0.042</u>	0.025	1.212	0.100	0.087	2.692
Easi3R$_{dust3r}$	✗	<u>0.029</u>	0.025	<u>1.774</u>	**0.040**	<u>0.021</u>	0.880	0.093	0.076	<u>2.366</u>
DUSt3R [64]	✓	<u>0.029</u>	<u>0.021</u>	1.875	0.076	0.030	0.974	<u>0.071</u>	0.067	3.711
Easi3R$_{dust3r}$	✓	**0.021**	**0.014**	**1.092**	0.042	**0.015**	0.655	**0.070**	**0.061**	**2.361**
MonST3R [73]	✗	0.040	0.034	1.820	<u>0.045</u>	0.024	0.759	0.183	**0.148**	6.985
Easi3R$_{monst3r}$	✗	0.038	0.032	1.736	<u>0.045</u>	<u>0.024</u>	0.715	0.184	<u>0.149</u>	<u>6.311</u>
MonST3R [73]	✓	<u>0.033</u>	<u>0.024</u>	<u>1.501</u>	0.055	0.025	0.776	<u>0.170</u>	0.155	6.455
Easi3R$_{monst3r}$	✓	**0.030**	**0.021**	**1.390**	**0.039**	**0.016**	**0.640**	**0.168**	0.150	**5.925**

Table 3. **Quantitative Comparisons of Camera Pose Estimation** on the DyCheck, ADT and TUM-dynamics datasets. The best and second best results are **bold** and <u>underlined</u>, respectively.

Method	Flow	DyCheck			ADT			TUM-dynamics		
		ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓
DUSt3R [64]	✗	0.035	0.030	2.323	0.042	0.025	1.212	0.100	<u>0.087</u>	<u>2.692</u>
CUT3R [63]	✗	<u>0.029</u>	<u>0.020</u>	<u>1.383</u>	0.084	0.025	<u>0.490</u>	<u>0.079</u>	0.088	10.41
MonST3R [73]	✓	0.033	0.024	1.501	0.055	0.025	0.776	0.170	0.155	6.455
DAS3R [68]	✓	0.033	0.022	1.467	<u>0.040</u>	0.017	0.685	0.173	0.157	8.341
Easi3R$_{monst3r}$	✓	0.030	0.021	1.390	**0.039**	<u>0.016</u>	<u>0.640</u>	0.168	0.150	5.925
Easi3R$_{dust3r}$	✓	**0.021**	**0.014**	**1.092**	0.042	**0.015**	0.655	**0.070**	**0.061**	**2.361**

Table 4. **Benefits of Easi3R on Point Cloud Reconstruction** on the DyCheck dataset. The best and second best results are **bold** and <u>underlined</u>, respectively. Easi3R $_{dust3r/monst3r}$ denotes the Easi3R experiment with the backbones of MonST3R/DUSt3R.

Method	Flow	Accuracy↓		Completeness↓		Distance↓	
		Mean	Median	Mean	Median	Mean	Median
DUSt3R [64]	✗	0.802	<u>0.595</u>	1.950	0.815	0.353	0.233
Easi3R$_{dust3r}$	✗	0.772	0.596	1.813	0.757	0.336	0.219
DUSt3R [64]	✓	0.738	0.599	<u>1.669</u>	0.678	<u>0.313</u>	0.196
Easi3R$_{dust3r}$	✓	**0.703**	**0.589**	**1.474**	**0.586**	0.301	**0.186**
MonST3R [73]	✗	0.855	0.693	1.916	1.035	0.398	0.295
Easi3R$_{monst3r}$	✗	<u>0.846</u>	<u>0.660</u>	1.840	0.983	0.390	0.290
MonST3R [73]	✓	0.851	0.689	<u>1.734</u>	<u>0.958</u>	<u>0.353</u>	**0.254**
Easi3R$_{monst3r}$	✓	0.834	0.643	1.661	0.916	0.350	<u>0.255</u>

Table 5. **Quantitative Comparisons of Point Cloud Reconstruction** on the DyCheck dataset. The best and second best results are **bold** and <u>underlined</u>, respectively.

Method	Flow	Accuracy↓		Completeness↓		Distance↓	
		Mean	Median	Mean	Median	Mean	Median
DUSt3R [64]	✗	0.802	0.595	1.950	0.815	0.353	0.233
CUT3R [63]	✗	**0.458**	**0.342**	<u>1.633</u>	<u>0.792</u>	<u>0.326</u>	<u>0.229</u>
MonST3R [73]	✓	0.851	0.689	1.734	0.958	0.353	0.254
DAS3R [68]	✓	1.772	1.438	2.503	1.548	0.475	0.352
Easi3R$_{monst3r}$	✓	0.834	0.643	1.661	0.916	0.350	0.255
Easi3R$_{dust3r}$	✓	<u>0.703</u>	<u>0.589</u>	**1.474**	**0.586**	**0.301**	**0.186**

method, built on DUSt3R and enhanced with our Easi3R attention-guided decomposition, accurately segments dynamic objects while maintaining robustness in handling textureless regions (e.g., trunks, rocks, walls), small dynamic objects (e.g., goose), and casual motions (e.g., girls, pedestrian). The results provide a surprising insight that 3D models, such as DUSt3R in our case, may inherently possess a strong understanding of the scene and can generalize well to standard 2D tasks.

4.2. Camera Motion

We evaluate camera motion by using the estimated extrinsic sequence on three dynamic benchmarks: DyCheck [14], TUM-dynamics [53], and ADT [23, 38] datasets. Specifically, the ADT dataset features egocentric videos, which are out-of-distribution for DUSt3R's training set. The DyCheck dataset includes diverse, in-the-wild dynamic videos captured from handheld cameras. The TUM-dynamics dataset contains major dynamic objects in relatively simple indoor scenarios. Instead of evaluating video clips as in previous methods, we adopt a more challenging setting by processing **entire sequences**. Specifically, we downsample frames at different rates: every 5 frames for ADT, 10 for DyCheck, and 30 for TUM-dynamics, resulting in approximately 40 frames. We use standard error metrics: Absolute Translation Error (ATE), Relative Translation Error (RTE), and Relative Rotation Error (RRE), after applying the Sim(3) alignment [60] on the estimated camera trajectory to the GT.

Benefits from Easi3R. We use DUSt3R and MonST3R without optical flow as the backbone settings, independently analyzing the benefits that Easi3R offers for each. We show qualitative comparisons of the estimation of the camera trajectory (Figure 7) and quantitative pose accuracy in Table 2, including w/ and w/o flow settings. Easi3R demonstrates more accurate and robust camera pose and trajectory estimation over both backbones and settings. Our Easi3R effectively leverages the inherent knowledge of DUSt3R with just a few lines of code, even achieving an improvement compared to models with optical flow prior.

Comparison. In Table 3, we compare Easi3R with state-of-the-art variants of DUSt3R. Unlike the plug-and-play comparison in Table 2, where we optionally disable the optical flow prior for a fair evaluation. Here, we report baseline performance using their original experimental settings, i.e., whether the flow model is used is specified in the second column. For clarity, we denote the setting "MonST3R +Easi3R" as Easi3R $_{monst3r}$ and "DUSt3R +Easi3R" as Easi3R $_{dust3r}$. Notably, our approach achieves significant improvements and delivers the best overall performance among all methods, without ANY fine-tuning on additional dynamic datasets or mask labels.

4.3. 4D Reconstruction

We evaluate 4D reconstruction on DyCheck [14] by measuring distances to ground-truth point clouds. Following prior work [1, 8, 61], we use accuracy, completeness, and dis-
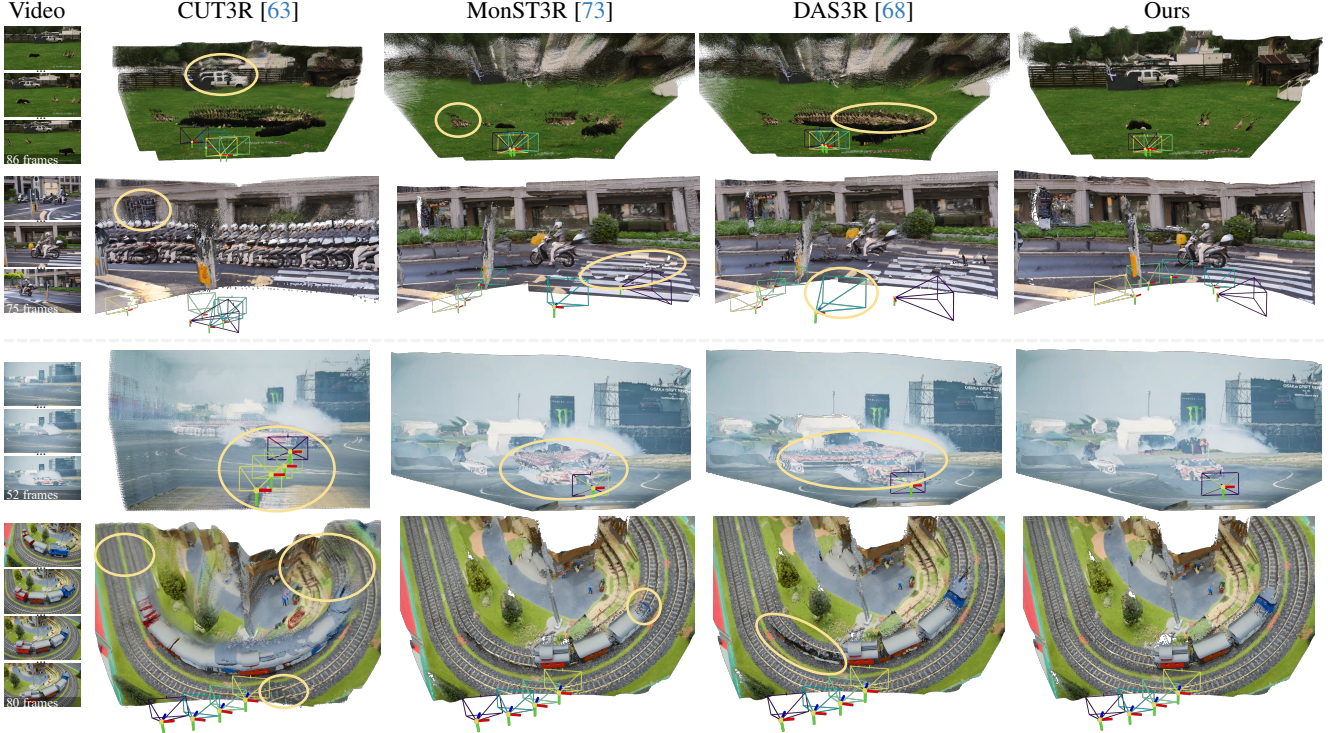
Figure 6. **Qualitative Comparison.** We visualize cross-frame globally aligned static scenes with dynamic point clouds at a selected timestamp. Notably, instead of using ground truth dynamic masks in previous work, we apply the estimated per-frame dynamic masks to filter out dynamic points at other timestamps for comparison. Our method (top two and bottom two rows as Easi3R $_{dust3r/monst3r}$, respectively) achieves temporally consistent reconstruction of both static scenes and moving objects, whereas baselines suffer from static structure misalignment and unstable camera pose estimation, and ghosting artifacts due to inaccuracy estimation of dynamic segmentation.
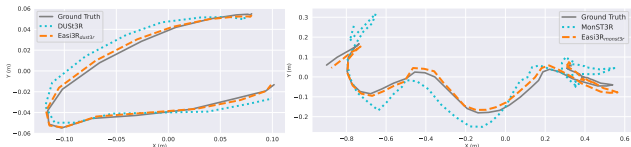

Figure 7. **Visualization of estimated camera trajectories.** Our robust estimated camera trajectory (orange) deviates less from the ground truth (gray) compared to the original backbones (blue).

tance metrics. Accuracy is the nearest Euclidean distance from a reconstructed point to ground truth, completeness is the reverse, and distance is the Euclidean distance based on ground-truth point matching.

Quantitative Results. We observe benefits from Easi3R in Table 4 and Table 5, Easi3R demonstrates more accurate reconstruction and outperforms most baselines, even comparable to concurrent CUT3R [63], which are trained with many extensive datasets.

Qualitative Results. We also compare the reconstruction quality of our method with CUT3R [63], MonST3R [73] and DAS3R [68] in Figure 6. All baselines struggle with misalignment and entanglement of dynamic and static re-

constructions, resulting in broken geometry, distortions, and ghosting artifacts. The key to our success lies in: (1) attention-guided segmentation for robust motion disentanglement, (2) attention re-weighting for improved pairwise reconstruction, and (3) segmentation-aware global alignment for enhanced overall quality.

5. Conclusion

We presented Easi3R, an adaptation to DUSt3R, which introduces the spatial and temporal attention mechanism behind DUSt3R, to achieve training-free and robust 4D reconstruction. We found the compositional complexity in attention maps, and propose a simple yet effective decomposition strategy to isolate the textureless, under-observed, and dynamic objects components and allowing for robust dynamic object segmentation. With the segmentation, we perform a second inference pass by applying attention re-weighting, enabling robust dynamic 4D reconstruction and camera motion recovery, and at almost no additional cost on top of DUSt3R. Surprisingly, our experimental results demonstrate that Easi3R outperforms state-of-the-art methods in most cases. We hope that our findings on attention map disentanglement can inspire other tasks.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2016. 7

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *ACM Communications*, 2011. 2

[3] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010. 2

[4] Honggyu An, Jinhyeon Kim, Seonghoon Park, Jaewoo Jung, Jisang Han, Sunghwan Hong, and Seungryong Kim. Cross-view completion models are zero-shot correspondence estimators. *arXiv*, 2412.09072, 2024. 2

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 2008. 2

[6] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2

[7] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010. 2

[8] Yue Chen, Xingyu Chen, Anpei Chen, Gerard Pons-Moll, and Yuliang Xiu. Feat2gs: Probing visual foundation models with gaussian splatting. *arXiv*, 2412.09606, 2024. 7

[9] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2007. 2

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[12] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2017. 2

[13] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014. 2

[14] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 2022. 7

[15] Lily Goli, Sara Sabour, Mark Matthews, Marcus Brubaker, Dmitry Lagun, Alec Jacobson, David J Fleet, Saurabh Saxena, and Andrea Tagliasacchi. Romo: Robust motion segmentation improves structure from motion. *arXiv preprint arXiv:2411.18650*, 2024. 3

[16] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[18] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. 2025. 2

[19] Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4d: Learning how things move in 3d from internet stereo videos. *arXiv*, 2412.09621, 2024. 1

[20] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

[21] Laurynas Karazija, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Learning segmentation from point trajectories. *arXiv preprint arXiv:2501.12392*, 2025. 2

[22] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[23] Skanda Koppula, Ignacio Rocco, Yi Yang, Joe Heyward, Joao Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. Tapvid-3d: A benchmark for tracking any point in 3d, 2024. *arXiv preprint arXiv:2407.05921*. 7

[24] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with MASt3R. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2

[25] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: accurate, fast, and robust structure and motion from casual dynamic videos. 2025. 1, 2

[26] Long Lian, Zhirong Wu, and Stella X Yu. Bootstrapping objectness from videos by relaxed common fate and visual grouping. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[27] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yanchao Yang, Qingnan Fan, and Baoquan Chen. SLAM3R: real-time dense scene reconstruction from monocular RGB videos. *arXiv*, 2412.09401, 2024. 2

[28] David G Lowe. Distinctive image features from scale-

invariant keypoints. *International journal of computer vision*, 2004. 2

[29] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Trans. on Graphics*, 2020. 2

[30] Quan-Tuan Luong and Olivier D Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 1996. 3

[31] Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy. Em-driven unsupervised learning for efficient motion segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2022. 2

[32] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 2015. 2

[33] Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: real-time dense SLAM with 3d reconstruction priors. *arXiv*, 2412.12392, 2024. 2

[34] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2011. 2

[35] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2013. 2

[36] Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975. 1

[37] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[38] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 7

[39] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[40] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: universal monocular metric depth estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[41] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 1999. 2

[42] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 2004. 2

[43] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[44] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2, 3

[45] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2020. 2

[46] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 6

[47] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[48] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 3

[49] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Vitor Guizilini, Yue Wang, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[50] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2009. 2

[51] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*. 2006. 2

[52] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 2008. 2

[53] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 2012. 7

[54] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[55] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 2

[56] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. MV-DUSt3R+: single-stage scene reconstruction from sparse views in 2 seconds. *arXiv*, 2412.06974, 2024. 2

[57] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in Neural Information Processing Systems (NIPS)*, 2021. 2

[58] Stefan Treue and John HR Maunsell. Attentional modulation of visual motion processing in cortical areas mt and mst.

Nature, 1996. 2

[59] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece*, 2000. 2

[60] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1991. 7

[61] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv*, 2408.16061, 2024. 2, 7

[62] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[63] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. 2025. 1, 2, 6, 7, 8, 4

[64] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: geometric 3d vision made easy. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 6, 7

[65] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2020. 6

[66] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2, 3, 6

[67] Junyu Xie, Weidi Xie, and Andrew Zisserman. Segmenting moving objects via an object-centric layered representation. *Advances in neural information processing systems*, 2022. 2

[68] Kai Xu, Tze Ho Elden Tse, Jizong Peng, and Angela Yao. DAS3R: dynamics-aware gaussian splatting for static scene reconstruction. *arXiv*, 2412.19584, 2024. 1, 2, 6, 7, 8, 4

[69] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2006. 2

[70] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2

[71] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[72] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Fei Qiao. DS-SLAM: A semantic visual SLAM towards dynamic environments. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2018. 2

[73] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: a simple approach for estimating geometry in the presence of motion. 2025. 1, 2, 3, 5, 6, 7, 8, 4

[74] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Noah Snavely, Michael Rubinstein, and William T. Freeman. Structure and motion from casual videos. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 1, 2

[75] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 2

Easi3R: Estimating Disentangled Motion from DUSt3R Without Training

Supplementary Material

In this **supplementary document**, we first present additional method details on temporal consistency dynamic object segmentation in Appendix A. Next, we conduct ablation studies of Easi3R in Appendix B and analysis limitations in Appendix C. Lastly, we report additional qualitative results in Appendix D. We invite readers to easi3r.github.io for better visualization.

A. Dynamic Object Segmentation

We have presented dynamic object segmentation for a single frame in Section 3.3, now we introduce how to ensure consistency along the temporal axis. Given image feature tokens $\mathbf{F}_0^t$ for frames at t, output from the image encoder, we concatenate them along the temporal dimension,

$$\bar{\mathbf{F}} = [\mathbf{F}_0^1; \mathbf{F}_0^2; \dots; \mathbf{F}_0^T] \in \mathbb{R}^{(T \times h \times w) \times c} \quad (12)$$

where c is the feature dimension of the tokens. This allows us to apply k-means clustering to group similar features across frames, producing cluster assignments,

$$C = \text{KMeans}(\bar{\mathbf{F}}, k), \quad C^t(x, y) \in \{1, \dots, k\}, \ \forall t, x, y \quad (13)$$

where k is the number of clusters, we use $k = 64$ for all experiments.

For each cluster $c \in \{1, \dots, k\}$, we compute a dynamic score s_c by averaging the base dynamic attention values of all tokens within that cluster:

$$s_c = \frac{\sum_t \sum_{i,j} \mathbb{1}[C^t(x, y) = c] \cdot \mathbf{A}^{t=\text{dyn}}(x, y)}{\sum_t \sum_{x,y} \mathbb{1}[C^t(x, y) = c]} \quad (14)$$

where $\mathbb{1}[\cdot]$ denotes the indicator function. We then use these scores to generate a cluster-fused dynamic attention map, mapping each pixel's cluster assignment back to its corresponding dynamic score,

$$\mathbf{A}_{\text{fuse}}^{t=\text{dyn}}(x, y) = s_{C^t(x,y)} \quad (15)$$

The refined dynamic attention map $\mathbf{A}_{\text{fuse}}^{t=\text{dyn}} \in \mathbb{R}^{h \times w}$ is used to infer the dynamic object segmentation by,

$$\mathbf{M}^t(x, y) = \mathbb{1}[\mathbf{A}_{\text{fuse}}^{t=\text{dyn}}(x, y) > \alpha] \quad (16)$$

where α is an automatic image thresholding using Otsu's method [36]. This refinement enforces temporal consistency by ensuring similar features across frames receive consistent dynamic scores, as shown in Figure 8. The resulting dynamic object segmentation is further utilized in the second inference pass and global optimization.
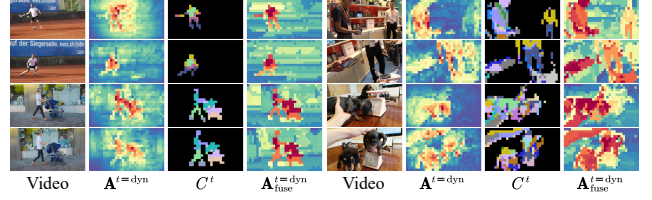

Figure 8. **Benefits of Cross-frame Feature Clustering.** We visualize the dynamic attention map $\mathbf{A}^{t=\text{dyn}}$, cluster assignments C^t, and cluster-fused dynamic attention map $\mathbf{A}_{\text{fuse}}^{t=\text{dyn}}$. Features from the DUSt3R encoder exhibit temporal consistency, as cluster assignments (C^t) remain unchanged across frames, thereby enhancing temporal consistency in dynamic segmentation ($\mathbf{A}^{t=\text{dyn}}$fuse) through clustering-guided temporal fusing. For better visual intuition, we invite readers to easi3r.github.io.

Table 6. **Ablation of Dynamic Object Segmentation** on DAVIS.

Backbone	Ablation	DAVIS-16		DAVIS-17		DAVIS-all	
		JM↑	JR↑	JM↑	JR↑	JM↑	JR↑
DUSt3R	w/o $\mathbf{A}_\mu^{a=\text{src}}$	45.1	45.2	42.8	39.9	42.2	38.5
	w/o $\mathbf{A}_\sigma^{a=\text{src}}$	42.3	50.0	35.0	37.0	30.9	28.3
	w/o $\mathbf{A}_\mu^{a=\text{ref}}$	33.3	28.4	31.5	27.9	32.5	29.7
	w/o $\mathbf{A}_\sigma^{a=\text{ref}}$	47.7	54.1	46.2	54.3	43.7	48.6
	w/o Clustering	40.0	38.5	38.3	38.3	34.3	30.5
	Full	**53.1**	**60.4**	**49.0**	**56.4**	**44.5**	**49.6**
MonST3R	w/o $\mathbf{A}_\mu^{a=\text{src}}$	47.2	51.5	44.4	46.7	40.9	41.5
	w/o $\mathbf{A}_\sigma^{a=\text{src}}$	49.7	60.1	48.7	57.8	44.9	49.6
	w/o $\mathbf{A}_\mu^{a=\text{ref}}$	46.4	54.0	47.4	55.9	45.3	50.7
	w/o $\mathbf{A}_\sigma^{a=\text{ref}}$	50.7	62.6	51.0	60.2	50.3	56.8
	w/o Clustering	45.5	46.7	45.3	48.1	42.1	43.5
	Full	**57.7**	**71.6**	**56.5**	**68.6**	**53.0**	**63.4**

B. Ablation Study

Our ablation lies in two folds: dynamic object segmentation and 4D reconstruction. For dynamic object segmentation, as shown in Table 11 we ablate the contribution of four aggregated temporal cross-attention maps, $\mathbf{A}_\mu^{a=\text{src}}, \mathbf{A}_\sigma^{a=\text{src}}, \mathbf{A}_\mu^{a=\text{ref}}, \mathbf{A}_\sigma^{a=\text{ref}}$, and feature clustering. The ablation results show that (1) Disabling any temporal cross-attention map leads to a performance drop, indicating that all attention maps contribute to improved dynamic object segmentation; and (2) Features from the DUSt3R encoder exhibit temporal consistency and enhance dynamic segmentation through cross-frame clustering.

Table 7 presents ablation studies on 4D reconstruction, evaluating two key design choices: (1) the impact of two-branch re-weighting (applying attention re-weighting to both reference and source decoders) and (2) global alignment using optical flow with and without segmentation. The

Table 7. **Ablation Study of Camera Pose Estimation and Point Cloud Reconstruction** on the DyCheck dataset.

| Backbone	Re-weighting	Flow-GA	Pose Estimation			Reconstruction					
			ATE↓	RTE↓	RRE↓	Accuracy↓		Completeness↓		Distance↓	
						Mean	Median	Mean	Median	Mean	Median
DUSt3R	Ref + Src	✗	0.030	0.026	1.777	0.775	0.596	1.848	0.778	0.342	0.224
	Ref	✗	0.029	0.025	1.774	0.772	0.596	1.813	0.757	0.336	0.219
	Ref	w/o Mask	0.026	0.017	1.472	0.940	0.831	1.654	0.685	0.336	0.220
	Ref	w/ Mask	**0.021**	**0.014**	**1.092**	**0.703**	**0.589**	**1.474**	**0.586**	**0.301**	**0.186**
MonST3R	Ref + Src	✗	0.040	0.032	1.751	0.848	0.744	1.850	1.003	0.398	0.292
	Ref	✗	0.038	0.032	1.736	0.846	0.660	1.840	0.983	0.390	0.290
	Ref	w/o Mask	0.033	0.023	1.495	0.969	0.796	1.752	0.998	0.368	0.273
	Ref	w/ Mask	**0.030**	**0.021**	**1.390**	**0.834**	**0.643**	**1.661**	**0.916**	**0.350**	**0.255**

DUSt3R [64]   Ours
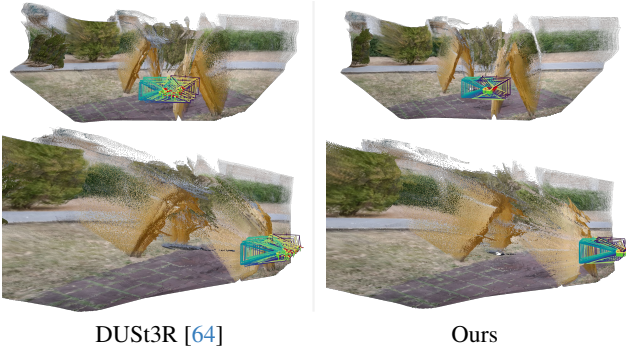
Figure 9. **Limitation.** We visualize static reconstructions from two different viewpoints in the top and bottom rows. Easi3R improves camera pose estimation and point cloud reconstruction (top row), enhancing alignment in structures like swing supports through attention re-weighting and segmentation-aware global alignment. However, from another viewpoint (bottom row), Easi3R still produces floaters near object boundaries.

ablation results show that (1) Re-weighting only the reference view decoder outperforms re-weighting both branches. Since the reference and source decoders serve different roles, and the reference view acts as the static standard, this aligns with our design intuition (i); and (2) Incorporating segmentation in global alignment consistently improves 4D reconstruction quality.

C. Limitations

Despite strong performance on various in-the-wild videos, Easi3R can fail when the DUSt3R/MonST3R backbones produce inaccurate depth predictions. While Easi3R effectively improves camera pose estimation and point cloud reconstruction, as shown in Table 5 of the main paper, it provides clear improvements in completeness and distance metrics, which are measured on the global point cloud. However, a noticeable gap remains in depth accuracy, which

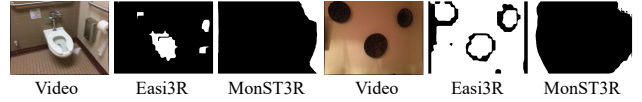Video   Easi3R   MonST3R   Video   Easi3R   MonST3R

Figure 10. **Dynamic masks in static scene.** Easi3R tends to reweight low confident regions in static scenes, leading to better pose estimation, as shown in Tab. 9.

Alignment	Method	Sintel		BONN		KITTI	
		Abs Rel ↓	δ<1.25 ↑	Abs Rel ↓	δ<1.25 ↑	Abs Rel ↓	δ<1.25 ↑
Per-sequence scale & shift	Marigold	0.532	51.5	0.091	93.1	0.149	79.6
	DepthAnythingV2	0.367	55.4	0.106	92.1	0.140	80.4
	NVDS	0.408	48.3	0.167	76.6	0.253	58.8
	ChronoDepth	0.687	48.6	0.100	91.1	0.167	75.9
	DepthCrafter	**0.292**	**69.7**	0.075	<u>97.1</u>	0.110	88.1
	Robust-CVD	0.703	47.8	-	-	-	-
	CasualSAM	0.387	54.7	0.169	73.7	0.246	62.2
	MASt3R	0.327	<u>59.4</u>	0.167	78.5	0.137	83.6
	Spann3R	0.508	50.8	0.157	82.1	0.207	73.0
	CUT3R	0.454	55.7	0.074	94.5	<u>0.106</u>	<u>88.7</u>
	DUSt3R	0.531	51.2	0.156	83.1	0.135	81.8
	MonST3R	0.333	59.0	<u>0.066</u>	96.4	0.157	73.8
	Easi3Rdust3r	0.435	59.1	0.085	91.1	0.155	76.1
	Easi3Rmonst3r	<u>0.316</u>	59.3	**0.057**	**97.2**	**0.092**	**90.6**
Per-sequence scale	MASt3R	0.641	43.9	0.252	70.1	0.183	74.5
	Spann3R	0.622	42.6	0.144	81.3	0.198	73.7
	CUT3R	0.421	47.9	0.078	93.7	<u>0.118</u>	**88.1**
	DUSt3R	0.656	45.2	0.155	83.3	0.144	81.3
	MonST3R	<u>0.378</u>	<u>55.8</u>	<u>0.067</u>	**96.3**	0.168	74.4
	Easi3Rdust3r	0.577	51.9	0.086	90.3	0.170	74.2
	Easi3Rmonst3r	**0.377**	**55.9**	**0.059**	97.0	0.102	<u>91.2</u>

Table 8. **Video Depth Evaluation**. We use the evaluation results from CUT3R for baselines.

is evaluated on per-view outputs. This is because our method focuses mainly on improving dynamic regions and global alignment rather than correcting depth predictions in static parts, as illustrated in Figure 9. We leave per-view depth correction for future work.

D. Addtional Results

More Evaluation. We evaluated on full-length sequences with downsampling only for GPU fit, leading to more dynamic and challenging motion than prior short-clip settings. For a more general evaluation, we also include MonST3R and CUT3R evaluation protocols. Tab. 8 and Tab. 9 confirm the effectiveness.

Category	Method	Sintel			TUM-dynamics			ScanNet (static)		
		ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓
Pose only	DROID-SLAM	0.175	0.084	1.912	-	-	-	-	-	-
	DPVO	0.115	0.072	1.975	-	-	-	-	-	-
	Particle-SfM	0.129	**0.031**	**0.535**	-	-	-	0.136	0.023	0.836
	LEAP-VO	**0.089**	0.066	1.250	0.068	0.008	1.686	**0.070**	0.018	0.535
Pose & Depth	Robust-CVD	0.360	0.154	3.443	0.153	0.026	3.528	0.227	0.064	7.374
	CasualSAM	0.141	**0.035**	**0.615**	0.071	**0.010**	1.712	0.158	0.034	1.618
	MASt3R	0.185	0.060	1.496	**0.038**	0.012	0.448	0.078	0.020	0.475
	Spann3R	0.329	0.110	4.471	0.056	0.021	0.591	0.096	0.023	0.661
	CUT3R	0.213	0.066	0.621	0.046	0.015	0.473	0.099	0.022	0.600
	DUSt3R	0.417	0.250	5.796	0.083	0.017	3.567	0.081	0.028	0.784
	MonST3R	0.111	0.044	0.869	0.098	0.019	0.935	0.077	0.018	0.529
	Easi3R$_{dust3r}$	0.402	0.098	0.876	0.134	0.017	1.077	0.067	0.018	0.670
	Easi3R$_{monst3r}$	**0.110**	0.042	0.758	0.105	0.022	1.064	**0.061**	**0.017**	0.525

Table 9. **Camera Pose Evaluation.** We use the evaluation results from CUT3R for baselines.

Supervision	Method	Input			Output		DAVIS-16		DAVIS-17	
		RGB	Optical Flow	Point Tracks	Mask	4D Reconstruction	JM↑	JM-M↑	JM↑	JM-M↑
Supervised	SFL	✓	✓		✓		67.4	-	-	-
	SIMO	✓	✓		✓		67.8	-	-	-
	OCLR-flow	✓	✓		✓		72.0	70.0	-	69.9
	OCLR-TTA	✓	✓		✓		80.8	80.2	-	76.0
	FlowSAM	✓	✓		✓		87.1	85.7	-	-
	SegAnyMo	✓		✓	✓		**90.6**	**89.2**	-	**90.0**
	DAS3R	✓			✓	✓	54.2	51.6	57.4	55.5
Unsupervised	SAGE	✓	✓		✓		42.6	-	-	-
	CUT	✓	✓		✓		55.2	-	-	-
	FTS	✓	✓		✓		55.8	-	-	-
	CIS	✓	✓		✓		70.3	67.6	-	-
	Motion Grouping	✓	✓		✓		68.3	-	-	-
	EM	✓	✓		✓		69.3	76.2	-	-
	RCF-Stage1	✓	✓		✓		80.2	78.6	-	-
	RCF-All	✓	✓		✓		82.1	81.0	-	-
	LRTL	✓	✓	✓	✓		**82.2**	-	-	-
Zero-shot	MonST3R	✓	✓		✓	✓	64.3	61.4	56.4	59.0
	Easi3R$_{dust3r}$	✓			✓	✓	67.9	67.4	60.1	62.0
	Easi3R$_{monst3r}$	✓			✓	✓	**70.7**	**71.1**	**67.9**	**67.7**

Table 10. **Comparisons of Dynamic Object Segmentation** on DAVIS with 2D dynamic segmentation methods.

Ablation	Variants	DAVIS-16		DAVIS-17		DAVIS-all	
		JM↑	JR↑	JM↑	JR↑	JM↑	JR↑
Window Size	3	76.0	89.2	70.8	82.4	65.7	76.2
	5*	70.7	79.9	67.9	76.1	63.1	72.6
	7	66.9	76.9	63.9	73.3	61.0	68.8
Number of Clusters	16	67.4	79.1	64.6	73.7	60.7	65.2
	32	71.6	83.9	68.0	78.3	65.1	75.2
	64*	70.7	79.9	67.9	76.1	63.1	72.6
	128	69.9	79.9	66.3	76.2	62.9	73.1
Thresholding Values	0.5	61.6	64.5	61.0	65.2	61.6	67.8
	0.7	70.2	85.0	62.8	71.8	58.1	67.0
	Otsu's method*	70.7	79.9	67.9	76.1	63.1	72.6

Table 11. **More ablations on segmentation quality using DAVIS.** * denotes the value used in the submission.

Behavior for Static Scenes. Interestingly, our method also improves on static scenes (ScanNet in Tab. 9 and Fig. 10), owing to our attention reweighting.

Comparison with 2D Baselines. We also include a comparison with 2D baselines. As shown in Tab. 10, Easi3R achieves SOTA segmentation in a zero-shot manner with only the image as input.

More Ablation. We further ablate the default settings - window size of 5, 64 clusters, and Otsu's method in Tab. 11. Using the recent segmentation method SegAnyMo, pose accuracy improves by 9.62% and depth accuracy improves by 4.11% on the Sintel dataset.

Runtime Our method runs at almost the same speed as MonST3R. MonST3R runs at 0.33 FPS, while Easi3R achieves 0.31 FPS for 512×144 image resolution on an NVIDIA RTX 4090 GPU.

Qualitative Results We report additional qualitative results of disentangled 4D reconstruction in Figure 11, Figure 12 and Figure 13. We find that MonST3R tends to predict under-segmented dynamic masks, while DAS3R tends to predict over-segmented dynamic masks. CUT3R, although it produces more accurate depth estimation, is prone to being affected by dynamic objects, leading to misaligned static structures, unstable camera pose estimation, and ghosting artifacts due to the lack of dynamic segmentation prediction. In contrast, Easi3R achieves more accurate segmentation, camera pose estimation, and 4D reconstruction, resulting in renderings with better visual quality.
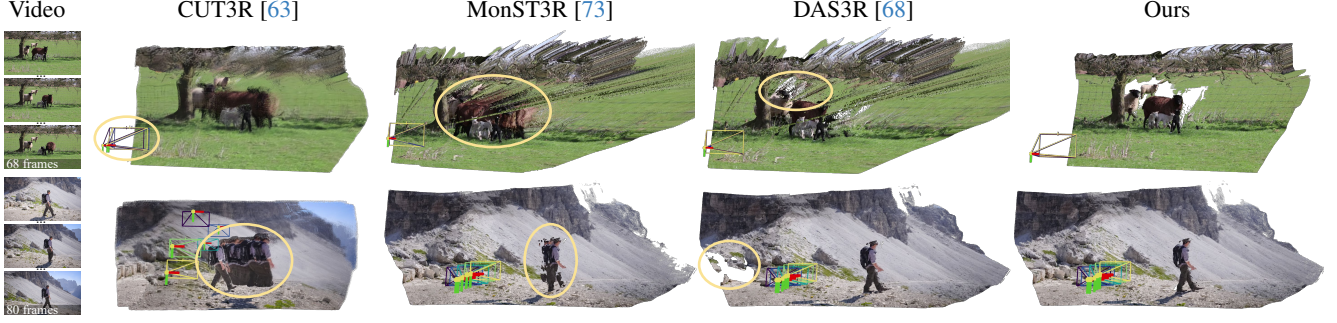
Figure 11. **Qualitative Comparison.** We visualize cross-frame globally aligned static scenes with dynamic point clouds at a selected timestamp. Notably, instead of using ground truth dynamic masks in previous work, we apply the estimated per-frame dynamic masks to filter out dynamic points at other timestamps for comparison. Top and bottom rows are Easi3R $_{dust3r/monst3r}$, respectively.
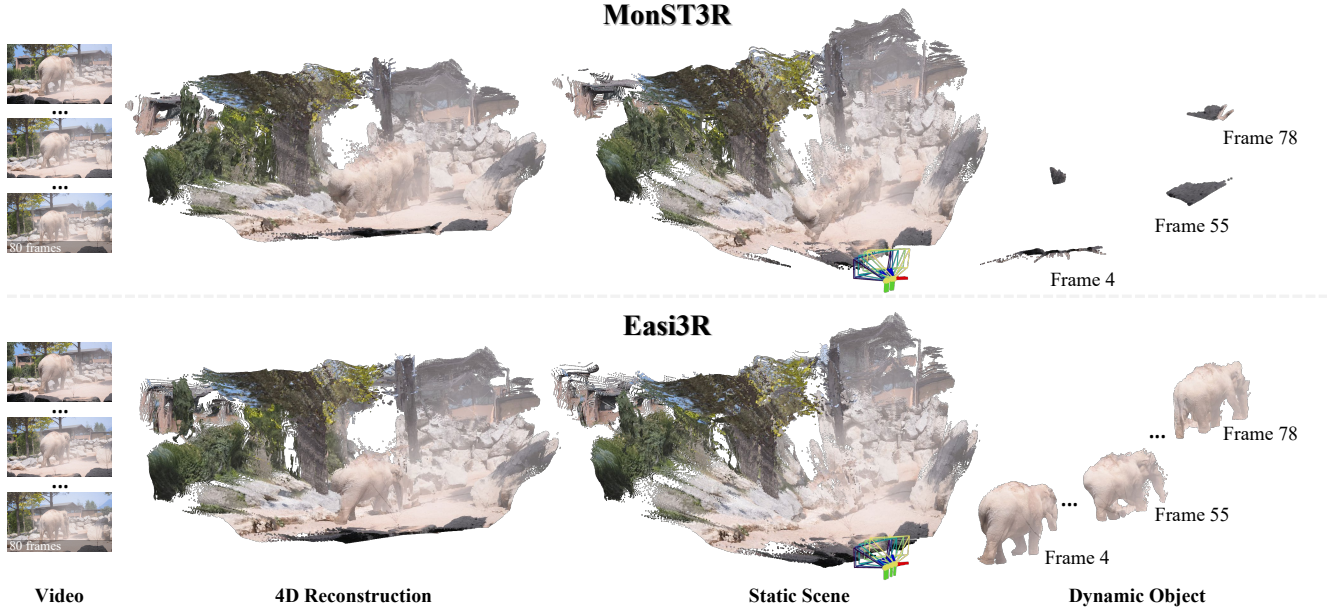

Figure 12. **Disentanglement vs. MonST3R [73].** We visualize the disentangled 4D reconstruction, static scene and dynamic objects at different frames. MonST3R tends to predict under-segmented dynamic masks.
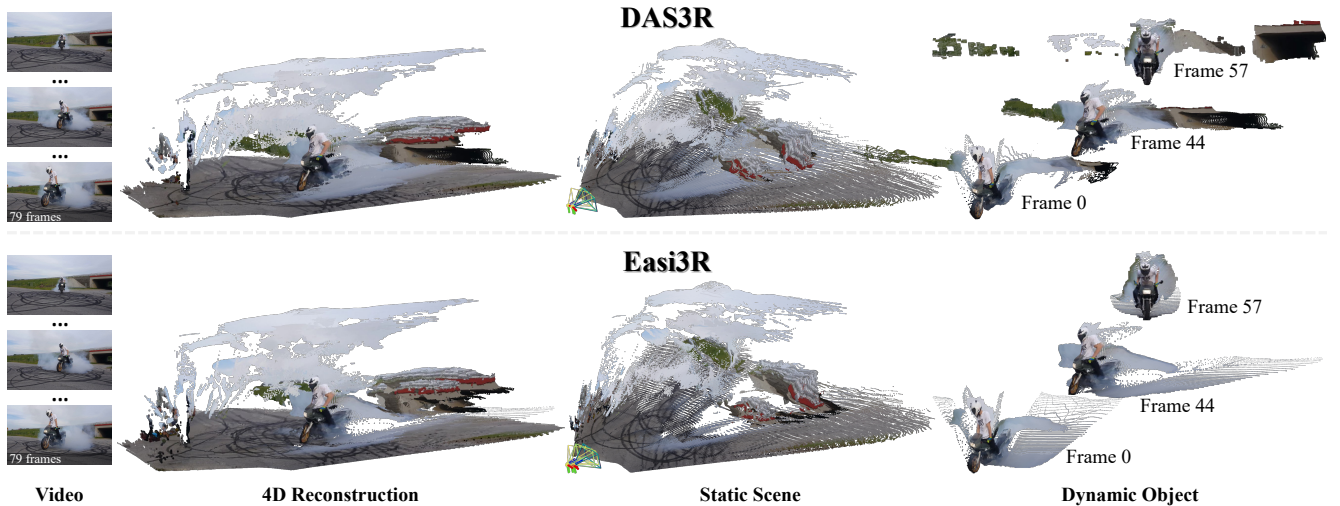

Figure 13. **Disentanglement vs. DAS3R [68].** We visualize the disentangled 4D reconstruction, static scene and dynamic objects at different frames. DAS3R tends to predict over-segmented dynamic masks.