The Dynamic Prior: Understanding 3D Structures for Casual Dynamic Videos

Zhuoyuan Wu[1], Xurui Yang[2], Jiahui Huang[3], Yue Wang[4], and Jun Gao[3,5]

[1] PKU [2] Independent Researcher [3] NVIDIA [4] USC [5] University of Michigan

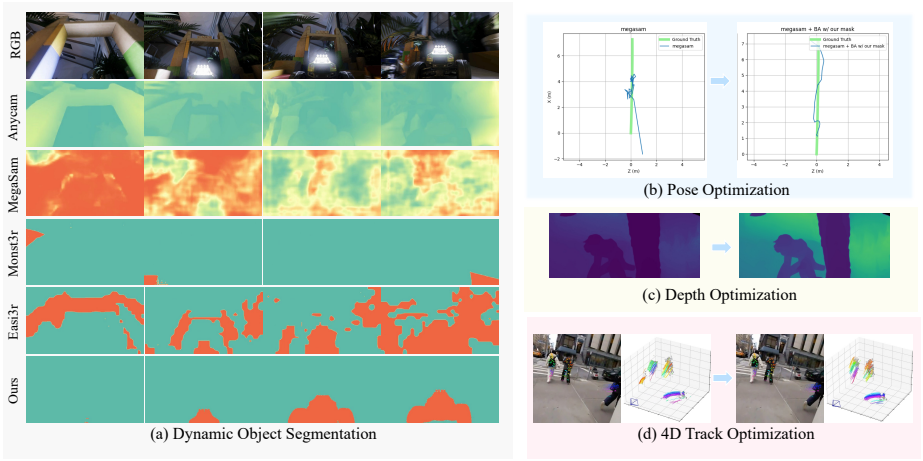

Fig. 1: Current works on estimating 3D structures from dynamic videos typically rely on segmenting out dynamic objects for robust bundle adjustment. However, the dynamic object masks from these works are typically inaccurate, as shown in Fig. (a). Our method can reason the dynamic scene and generate precise masks for dynamic objects. These masks can be seamlessly integrated into camera pose optimization (Fig. (b)), depth optimization (Fig. (c)), and 4D track optimization (Fig. (d)) for robust structural 3D understanding.

Abstract. Estimating accurate camera poses, 3D scene geometry, and object motion from in-the-wild videos is a long-standing challenge for classical structure from motion pipelines due to the presence of dynamic objects. Recent learning-based methods attempt to overcome this challenge by training motion estimators to filter dynamic objects and focus on the static background. However, their performance is largely limited by the availability of large-scale motion segmentation datasets, resulting in inaccurate segmentation and, therefore, inferior structural 3D understanding. In this work, we introduce the Dynamic Prior (DYNAPO) to robustly identify dynamic objects without task-specific training, leveraging the powerful reasoning capabilities of Vision-Language Models (VLMs) and the fine-grained spatial segmentation capacity of SAM2. DYNAPO can be seamlessly integrated into state-of-the-art pipelines for camera pose optimization, depth reconstruction, and 4D trajectory estimation. Extensive experiments on both synthetic and real-world videos demonstrate that DYNAPO not only achieves state-of-the-art performance on motion segmentation, but also significantly improves accuracy and robustness for structural 3D understanding. Code is available.

1 Introduction

Robustly estimating accurate 3D structures from real-world videos, including camera poses, scene geometry, and object motion, is a fundamental problem in computer vision, with applications across VR/AR, robotics, autonomous vehicles, and more broadly, spatial intelligence. With decades of research, classical methods, such as Structure from Motion (SfM) [17] and Simultaneous Localization And Mapping (SLAM) [14], have yielded mature algorithms for stationary scenes. However, their reliance on the epipolar constraints poses significant challenges to dynamic scenarios with moving objects, which are pervasive in real-world videos.

The fundamental problem is the ambiguity introduced by the moving objects, which violates the epipolar constraints. Many recent works [7,20,24,25,33,36,52,56,63,64,73] tackle this problem by training a neural network to identify the dynamic regions and filter them out in the bundle adjustment. However, these approaches are fundamentally limited by the availability of large-scale datasets for motion segmentation, and thus, produce inaccurate motion masks for in-the-wild videos (as shown in Fig. 1), leading to inferior 3D scene understanding results. BA-Track [7] jointly optimizes static and dynamic scene geometry through a motion decoupled 3D tracking estimator model. Yet, training this estimator model is also limited by the availability of large-scale training data.

Large vision foundation models, such as VLMs [3, 12, 23, 62], SAM [45], and Depth Anything [68], has demonstrated unprecedented success in vision perception. While many recent methods [7, 21, 24, 33, 63] have started integrating the prior knowledge from these models to improve the robustness and accuracy; they are mainly limited to monocular depth estimation and tracking [43, 55, 68]. In this work, we present the Dynamic Prior (DYNAPO), which leverages the powerful modern vision foundation model to overcome the generalization gap for motion segmentation, and in turn enhances structural 3D understanding. Specifically, we study the combination of Vision-Language Models (VLMs) [2, 3, 12, 23] and 2D segmentation models (SAM2) [45], where VLMs perform high-level reasoning to identify dynamic objects in the video and prompt SAM2 to segment out each dynamic object. Since both VLMs and SAM2 are pre-trained on vast internet-scale data, DYNAPO is robust and generalizable, and achieves state-of-the-art performance on motion segmentation.

We further integrate our DYNAPO in various state-of-the-art 3D structural understanding frameworks. **(I)** We integrate it into camera pose optimization pipelines [63] to refine the camera poses by using our dynamic mask for filtering in bundle adjustment. Our framework significantly improves the accuracy of pose estimation for a wide range of methods [8, 9, 33, 56, 58, 63, 64, 73] on both synthetic and real-world videos. **(II)** We integrate it with the consistent depth optimization pipeline from MegaSaM [33] for better recovery of 3D scene geometry. Using the dynamic mask from DYNAPO achieves significant improvement on the depth estimation across various datasets. **(III)** We extend it with Stereo4D [24] for 4D trajectory estimation. By simply replacing the motion mask from Stereo4D with DYNAPO, the same optimization pipeline can be significantly improved for better trajectory estimation. These extensive applications and experiments demonstrate the effectiveness of motion estimation from our DYNAPO, moving a step towards robust and accurate structural 3D understanding.

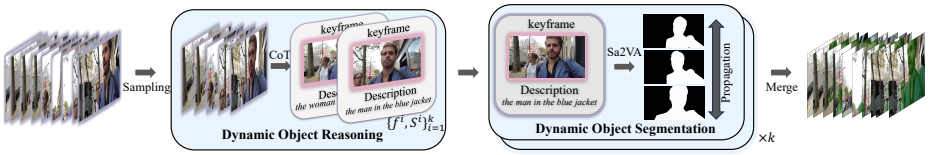

Fig. 2: Overview of DYNAPO. Given a video sequence, the **Dynamic Object Reasoning** takes input of sub-sampled keyframes, reasons all the dynamic objects within the video, and generates descriptions s^i and the frame number f^i for each dynamic object (in total, k objects). The **Dynamic Object Segmentation** then generates a mask sequence for each dynamic object, which we take an average to produce the final dynamic mask.

2 Related Work

2.1 Motion Segmentation

Motion segmentation aims to segment moving objects from static backgrounds. We broadly categorize prior work into flow-based and trajectory-based approaches. Flow-based methods [11, 37, 65, 67, 69] analyze short-term pixel displacements, and often fail with slow movements or significant camera motion. Trajectory-based methods [13, 30, 34, 41, 53] track points over longer sequences, offering more robustness to occlusion but at a higher computational cost and sensitivity to tracking errors. More recently, SegAnyMo [22] combines long-range trajectories with semantic features and leverages SAM2 [45] for mask densification. Romo [18] improves SfM by robustly combining optical flow and epipolar cues with a pre-trained video segmentation model. However, both paradigms primarily rely on geometric cues and struggle to incorporate semantic context. This limitation prevents them from understanding complex dynamic scenes where an object's motion is defined by its interaction with the environment. On the contrary, our method leverages chain-of-thought reasoning for complex video tasks [26], providing a deeper, context-aware understanding of dynamic scenes.

2.2 Dynamic Structure from Motion

Traditional SfM pipelines [1, 44, 47, 49] have excelled at reconstructing static scenes by matching local features across multiple views. However, the presence of dynamic objects violates the constraints, often leading to catastrophic failures. Recent advances in deep learning have led to three main paradigms for addressing this challenge. The first is feed-forward models [56, 61], which directly regress camera parameters and 3D structures from images. While being powerful, these models, along with their follow-up works [35, 40, 55], are typically trained on static datasets and struggle with dynamic content. To mitigate this, many methods [31, 58, 73] have explored explicitly handling moving objects by fine-tuning on dynamic scenes, but are limited by the availability of large-scale dynamic datasets. A separate line of work avoids fine-tuning altogether, repurposing the attention maps of a pretrained 3D foundation model at inference to expose motion: Easi3r [8], VGGT4D [19], and MUT3R [48] adapt DUSt3R, VGGT, and CUT3R respectively. Yet such attention signals still lack the semantic reasoning. The second paradigm integrates deep neural networks into SLAM systems. Models such as DROID-SLAM [52], MegaSAM [33], and LEAPVO [6] build upon neural

motion estimation to create robust systems that can operate in dynamic environments. Other methods like ParticleSfM [75], Robust-CVD [28] and CasualSAM [72] also aim to improve robustness by explicitly identifying and down-weighting the influence of dynamic elements during optimization. However, a common limitation is the accuracy of motion segmentation, which is limited by the training data and generalizes poorly for in-the-wild videos. Our work integrates the reasoning-based motion segmentation model into SfM, and significantly improves the performance of structural 3D understanding.

The third paradigm goes beyond pose and depth to jointly recover geometry, camera motion, and scene dynamics from casual videos. POMATO [74] and C4D [59] match temporal point maps for dynamic reconstruction, V-DPM [51] extends dynamic point maps to 4D video, and St4RTrack [15] jointly reconstructs and tracks the scene in a unified world frame. Being feed-forward networks, however, they inherit the dynamic-data dependency of trained reconstruction models and generalize poorly to in-the-wild motion. GFlow [60], like ours, is training-free, recovering a 4D world from optical-flow and epipolar cues; yet these low-level geometric cues degrade under slow or ambiguous motion, where a moving object cannot be told apart from a temporarily stationary one. In both cases the notion of motion is geometric rather than semantic. These methods are thus largely complementary to DYNAPO: our semantically grounded dynamic prior can be plugged into their optimization as a more accurate motion mask, and we provide quantitative comparisons with these concurrent methods in the supplementary material.

3 The Dynamic Prior

Our goal is to estimate camera poses, scene geometry and 4D trajectory from casual dynamic videos. We first discuss the details of DYNAPO for motion segmentation using dynamic prior in this section, and integrate it into structural 3D understanding pipelines in Sec. 4.

As shown in Fig. 2, with a video sequence $\mathbf{V} = \{I_t \in \mathbb{R}^{H\times W\times 3}\}_{t=1}^{T}$, DYNAPO estimates a binary mask sequence $\{\mathbf{M}_t \in \{0,1\}^{H\times W}\}_{t=1}^{T}$, capturing all the dynamic objects in the video. DYNAPO achieves this through a two-stage framework, each leveraging prior knowledge from pretrained vision foundation models. In the first stage (Sec. 3.1), we prompt a VLM to reason about the dynamic objects in the video, identify the keyframe where the object is most salient, and generate descriptions for each dynamic object. The descriptions and keyframes are then fed into the second stage (Sec. 3.2), which uses Multi-Modal Large Language Model (MLLM) with SAM2 to generate the dynamic mask. The whole process is fully automatic without post-training and is applicable to in-the-wild videos.

3.1 Dynamic Object Reasoning

The key question for reasoning dynamic objects is to understand *"which objects are moving"* in the video. The VLMs, such as GPT-4o [23], Gemini2.5 [12], and Qwen-VL [2, 3], have been trained on various video understanding tasks with emergent reasoning capabilities [23]. We design a prompting mechanism to let VLMs explicitly reason about dynamic objects that are most representative in the video.

First, to reduce token counts and improve efficiency, we uniformly subsample a set of keyframes from the video, $\mathcal{C} = \{I_t\}, t \in \mathcal{T}_{cand}$, where $\mathcal{T}_{cand}$ is the set of frame indices for keyframes. The VLMs are then prompted through a structured chain-of-thought process to analyze the keyframes $\mathcal{C}$ and reason about object motion. Specifically, **(i).** VLMs analyze each keyframe and identify moving objects across the frames. **(ii).** For each object, VLMs determine the occlusion, and start and end frames if object appears and disappears. **(iii).** VLMs finally generate the frame number where the object is most salient and a description for each object. The output is k tuples corresponding to k unique dynamic objects $\{(f^i, s^i)\}_{i=1}^k$, where $f^i \in \mathcal{C}$ is the selected frame for the i-th dynamic object, and s^i is a description for the object's location and appearance within the frame f^i. Further details are provided in the Appendix.

3.2 Dynamic Object Segmentation

With the identified dynamic objects and their descriptions, we first generate the mask for each object in the corresponding keyframe, and then propagate the mask from the keyframe into the whole video. The final binary mask $\{\mathbf{M}_t \in \{0,1\}^{H\times W}\}_{t=1}^T$ is obtained by averaging across all the dynamic object. We detail each step below:

Mask Segmentation at Keyframe. For each identified dynamic object i, we generate a binary mask $\mathbf{m}^i \in \{0,1\}^{H\times W}$ on the corresponding keyframe f^i, leveraging the description s^i. Our segmentation pipeline leverages pretrained Sa2VA [71], which synergizes a multi-modal large language model (MLLM) with SAM2 [45] decoder. We first encode both text and visual inputs into a unified token embedding space and process it through an MLLM, which generates specialized instruction tokens (i.e., "[SEG]" tokens) alongside conventional language outputs. The instruction token serves as a prompt for the SAM2 decoder, directing it to produce the segmentation masks for the selected dynamic object.

Mask Propagation from Keyframe to Video. The mask $\mathbf{m}^i$ at the keyframe needs to propagate to the entire video sequence to form the complete mask sequence $\{\mathbf{m}_t^i \in \{0,1\}^{H\times W}\}_{t=1}^T$ for the dynamic object i. We leverage the memory mechanism in SAM2 [45], without modification. We provide a rough description in the Appendix and refer the readers to the original paper for details. With this memory mechanism, SAM2 can generate a temporally coherent mask throughout the entire video sequence. Applying the same process to for all k identified dynamic instances, we obtain k separate mask sequences, $\{\mathbf{m}_t^1\}_{t=1}^T, \cdots, \{\mathbf{m}_t^k\}_{t=1}^T$.

Mask Merging. We merge all the instance-level video mask sequences into a unified video mask for the final output by simply computing the element-wise union of all individual instance masks: $\mathbf{M}_t = \bigcup_{i=1}^k \mathbf{m}_t^i$. This process generates the final binary mask sequence $\{\mathbf{M}_t\}_{t=1}^T$, where each mask identifies all dynamic regions in its corresponding frame.

4 3D Structure Understanding from Videos

The estimated dynamic masks from DYNAPO (Sec. 3) serve as the filter for the dynamic object, from where we can optimize the camera pose, scene geometry, and 4D

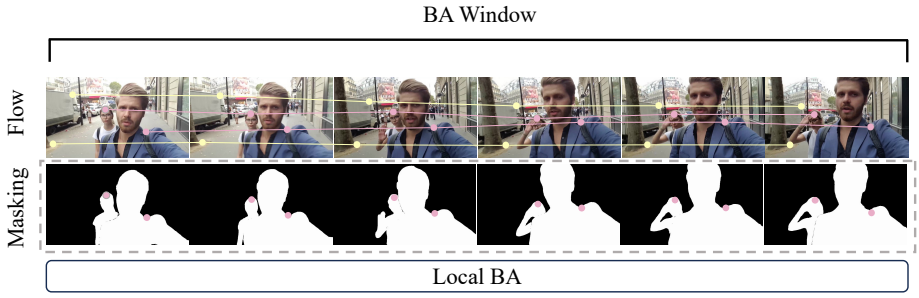

Fig. 3: Illustration of our BA. When calculating the reprojection loss, we effectively remove the dynamic objects (pink tracks) from the loss function using the mask from DYNAPO, and the BA can only focus on the static background for optimization.

tracking using conventional methods. We integrate DYNAPO into three state-of-the-art structural 3D understanding frameworks. We first integrate into camera pose optimization pipelines [63] to refine the camera poses with bundle adjustment in Sec. 4.1. We then integrate it with MegaSam [33] for consistent depth optimization in Sec. 4.2 to recover 3D scene geometry. Finally, we extend it with Stereo4D [24] for accurate 4D trajectories estimation in Sec. 4.3.

4.1 Camera Pose Optimization

Problem Setting. We incorporate DYNAPO into the bundle adjustment (BA) process from AnyCam [63] to optimize camera poses. This process acts as a plug-and-play post-processor that can be applied to refine the camera pose estimation from various methods. The inputs are the initial camera poses $\{\mathbf{P}_t\}$, intrinsics matrix K, and depth maps $\{\mathbf{D}_t\}$, and dense optical flow fields $\mathbf{F}^{t\to t+1}$ between adjacent frames from any off-the-shelf 3D estimation model. We optimize the camera pose with the dynamic object mask derived from DYNAPO.

Let $\pi_K(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^2$ denote the projection function that maps a 3D point $\mathbf{x}$ in the camera's coordinate system to a 2D pixel $\mathbf{p}$ on the image plane, parameterized by the intrinsics K. Accordingly, $\pi_K^{-1}(\mathbf{p}, d) : (\mathbb{R}^2, \mathbb{R}_+) \to \mathbb{R}^3$ denotes unprojecting a pixel $\mathbf{p}$ with depth d into 3D. Following AnyCam [63], we sample a grid of points in a starting frame t_0 and track them for 8 frames by chaining the dense optical flows $\mathbf{F}^{t_0\to t_0+1}, \dots, \mathbf{F}^{t_0+1\to t_0+8}$, forming a track: $((\mathbf{p}_1, \dots, \mathbf{p}_8), (m_1, \dots, m_8), d_1)$, where $\{\mathbf{p}_i\}$ are the pixel locations, $\{m_i\}$ are the corresponding binary motion mask values, and d_1 is the depth of the first point. We take the mask value from our generated motion mask from Sec. 3:

$$m_i = 1 - \mathbf{M}_{t_0+i-1}(\mathbf{p}_i), \tag{1}$$

where $\mathbf{M}_{t_0+i-1}(\mathbf{p}_i)$ denotes the mask value at pixel location $\mathbf{p}_i$. The initial depth is taken from the input depth map.

Optimization. We jointly optimize camera poses, intrinsics and depths of anchor points to minimize the reprojection error on the static regions, indicated by our motion masks.

Specifically, the loss for a single track is:

$$\mathcal{L}_{\text{Repr}} = \sum_{i=2}^{8} m_i \cdot \left\| \pi_K(\mathbf{P}^{1\to i}\pi_K^{-1}(\mathbf{p}_1, d_1)) - \mathbf{p}_i \right\|_1, \tag{2}$$

The term m_i ensures that the reprojection error only focuses on static points when $m_i = 0$, as shown in Fig. 3.

To ensure a plausible camera trajectory and prevent jerky movements, we additionally incorporate a smoothness regularizer. This term penalizes deviations from a constant-velocity camera motion by minimizing the difference between consecutive camera poses:

$$\mathcal{L}_{\text{smooth}} = \sum_{t=1}^{6} \left\| \left(\mathbf{P}^{t\to t+1}\right)^{-1} \mathbf{P}^{t+1\to t+2} - \mathbf{I}_4 \right\|_{1,1}, \tag{3}$$

where $\mathbf{I}_4$ is the 4×4 identity matrix. The final optimization objective is a weighted sum of the reprojection error and the smoothness term:

$$\mathcal{L}_{\text{BA}} = \mathcal{L}_{\text{Repr}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}}. \tag{4}$$

4.2 Depth Optimization

Background. We integrate DYNAPO with the Consistent Video Depth (CVD) optimization method from MegaSaM [33] to optimize depth predictions. We briefly introduce it here and refer readers to the original paper for further details. The objective in CVD is a weighed sum of pairwise 2D flow reprojection loss $\mathcal{L}_{\text{flow}}$, temporal depth consistency loss $\mathcal{L}_{\text{temp}}$, and mono-depth prior loss $\mathcal{L}_{\text{prior}}$:

$$\mathcal{L}_{\text{cvd}} = \lambda_{\text{flow}}\mathcal{L}_{\text{flow}} + \lambda_{\text{temp}}\mathcal{L}_{\text{temp}} + \lambda_{\text{prior}}\mathcal{L}_{\text{prior}}. \tag{5}$$

To handle dynamic objects, CVD jointly optimizes an uncertainty map $\{\boldsymbol{\Omega}_t\}_{t=1}^T$. Specifically, for a pixel $\mathbf{p}$ at frame t, its projected location at frame j is computed as $\mathbf{p}' = \pi_K(\mathbf{P}^{t\to j}\pi_K^{-1}(\mathbf{p}, d)))$. The reprojection loss $\mathcal{L}_{\text{flow}}$ computes the difference with the optical flow $\mathbf{F}^{t\to j}(\mathbf{p})$, weighted by the uncertainty $\boldsymbol{\Omega}_t(\mathbf{p})$:

$$\mathcal{L}_{\text{flow}}^{i\to j} = \boldsymbol{\Omega}_i(\mathbf{p}) \cdot ||\mathbf{p}' - (\mathbf{p} + \mathbf{F}^{i\to j}(\mathbf{p}))||_1 + \log(1/\boldsymbol{\Omega}_i(\mathbf{p})), \tag{6}$$

This loss encourages pixel disparity to be temporally consistent according to the estimated 2D optical flow. The uncertainty map $\{\boldsymbol{\Omega}_t\}_{t=1}^T$ is jointly optimized in the CVD pipeline.

Our Formulation. MegaSAM relies on the optimized uncertainty map $\{\boldsymbol{\Omega}_t\}_{t=1}^T$ to down-weight dynamic regions during optimization, which are typically inaccurate and prone to error. We replace it with our semantics-aware motion mask, $\mathbf{M}_t$ derived from Sec. 3, by simply initializing and finetuning it during optimization.

$$\boldsymbol{\Omega}_t^{\text{ours}} = 1 - \mathbf{M}_t. \tag{7}$$

This substitution is applied to both $\mathcal{L}_{\text{flow}}$ and $\mathcal{L}_{\text{temp}}$, which allows them to completely disregard pixels on moving objects and focus exclusively on the static parts of the scene. A visual comparison is shown in Fig. 4.

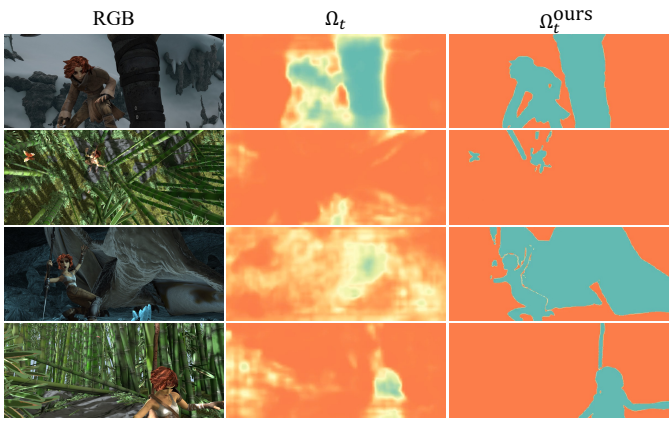

Fig. 4: Visualization of uncertainty map. MegaSaM produces inaccurate dynamic masks, deteriorating depth optimization, while our DYNAPO generates cleaner and more plausible dynamic masks.

4.3 4D Track Optimization

Background. Stereo4D [24] optimizes 4D trajectories from dynamic videos. For a 2D point track $\{\mathbf{p}_i\}_{i=1}^N$, with corresponding initial depth estimates $\{d_i\}_{i=1}^N$, the corresponding 3D points $\{\mathbf{x}_i\}_{i=1}^N$ can be obtained through unprojection $\mathbf{x}_i = \pi_K^{-1}(\mathbf{p}_i, d_i)$. The core idea of Stereo4D [24] is to optimize a per-frame scalar offset δ_i for each point along its corresponding camera ray, $\mathbf{r}_i = (\mathbf{x}_i - \mathbf{c}_i)/||\mathbf{x}_i - \mathbf{c}_i||$, where the camera center $\mathbf{c}_i$ is the translation component of the camera-to-world pose $\mathbf{P}_i$. The refined 3D position is then $\mathbf{x}_i' = \mathbf{x}_i + \delta_i \mathbf{r}_i$. The main objective in Stereo4D [24] is a combination of three terms:

$$\min_{\{\delta_i\}_{i=1}^N} \sigma(\mu)\mathcal{L}_{\text{static}} + (1-\sigma(\mu))\mathcal{L}_{\text{dynamic}} + \mathcal{L}_{\text{reg}}, \tag{8}$$

where $\mathcal{L}_{\text{static}}$ encourages stationary points to remain fixed in world space, $\mathcal{L}_{\text{dynamic}}$ promotes smooth trajectories for moving points, and $\mathcal{L}_{\text{reg}}$ ensures consistency to the initial depth estimates. $\sigma(\mu)$ is the function of motion score μ, defined as $\sigma(\mu) = \frac{1}{1+\exp(\mu-\mu_0)}$, with $\mu_0 = 20$ as a threshold.

While the formulation in Stereo4D [24] provides a solid foundation for track optimization, the objective heavily relies on the accuracy of the motion score. In the original paper, μ is defined as the as the 90^{th} percentile of the track's trail length across all frames:

$$\mu = \text{Percentile}_{i=1:N}^{90}\left[\max_{w=1:w_o} \|\pi_i(\mathbf{p}_i) - \pi_i(\mathbf{p}_{i-w})\|\right] \tag{9}$$

Such a definition is prone to noise from camera ego-motion, misclassifying points for motion and degrading trajectory quality, as shown in Fig. 5.

Our Formulation. Our key modification is to replace this heuristic motion score with a more robust, semantics-aware signal derived directly from our DYNAPO. Specifically,

we determine a single motion score for each track based on dynamic masks for all the 2D points along the track.

$$\mu^{\text{ours}} = \begin{cases} \mu_{dyn} & \text{if } \exists i \in \{1, \dots, N\} \text{ s.t. } \mathbf{M}_{t_0+i-1}(\mathbf{p}_i) = 1 \\ \mu_{stat} & \text{otherwise} \end{cases} \tag{10}$$

where $\mathbf{p}_i$ is the i-th point of track and we simply set $\mu_{dyn} = 25.0$ and $\mu_{stat} = 15.0$. This approach creates a decisive, binary switch for each track, pushing the sigmoid weighting function $\sigma(\mu^{\text{ours}})$ to its extremes. This allows the objective Eq. 8 to reasonably distinguish static or dynamic tracks, leading to significantly cleaner and more plausible 4D trajectories.

Fig. 5: Motion mask visualization. Stereo4D's [24] motion scores are inaccurate—some dynamic tracks get tiny scores, causing inappropriate loss weighting.

5 Experiments

We first evaluate the performance of DYNAPO for dynamic object segmentation in Sec. 5.1. The improved dynamic object masks are, in turn, applied to pose optimization, depth optimization, and 4D track optimization for evaluation in Sec. 5.2, Sec. 5.3, and Sec. 5.4, respectively.

5.1 The Dynamic Prior

Implementation Details. We use GPT-4o [23] (version "2024-05-13") as our dynamic object reasoner. The temperature of GPT-4o is set to 0.5, and the maximum number of output tokens is 4096. For the segmentation, we use Sa2VA-8B [71] and SAM2 [45] with the official pre-trained checkpoints.

Baselines. We compare our method against state-of-the-art approaches for moving object segmentation including SegAnyMo [22][1], RoMo [18], ABR [66], CIS [69], EM [37], RCF [34], OCLR [65], STM [38], SIMO [29] and LRTR [27].

Evaluation Datasets and Metrics. We evaluate on three well-established benchmarks for dynamic object segmentation: DAVIS-2016 [42], SegTrackv2 [32] and FBMS-59 [41]. We follow prior work and compute the region similarity ($\mathcal{J}$), also known as the Jaccard index, measures the overlap between the predicted and ground-truth masks. For scenarios with multiple objects, we consolidate all foreground objects into a single mask for evaluation following prior work [13, 65, 66, 69].

[1] we evaluate with the official code and checkpoints from https://github.com/nnanhuang/SegAnyMo

Table 1: Quantitative comparison on motion segmentation. Our method achieves state-of-the-art results in SegTrackv2 and FBMS-59, with comparable performance on the DAVIS-2016 dataset.

Methods	SegTrackv2	FBMS-59	DAVIS-2016
	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$
CIS [69]	62.0	63.6	70.3
SIMO [29]	62.2	-	67.8
OCLR-flow [65]	67.6	65.5	72.0
OCLR-TTA [65]	72.3	69.9	80.8
EM [37]	55.5	57.9	69.3
RCF-Stage1 [34]	76.7	69.9	80.2
RCF-All [34]	79.6	72.4	82.1
STM [38]	55.0	59.2	73.2
ABR [66]	76.6	81.9	71.8
LRTR [27]	81.2	79.6	**82.2**
SegAnyMo [22]	76.3	76.4	81.9
RoMo [18]	67.7	75.5	77.3
Ours	**83.3**	**84.0**	81.6

Experimental Results. We provide quantitative comparisons in Table 1 with qualitative examples in Fig. 6. Our approach achieves state-of-the-art results on FBMS-59 and SegTrackv2, with comparable performance on DAVIS-2016 The significant improvement gains on both SegTrackv2 and the more challenging FBMS-59 benchmark highlight the robustness and accuracy of our proposed dynamic object reasoning in identifying moving objects.

Ablation Studies. Due to limited space, we provide results in the appendix. We ablate key components of our method, including the choice of Vision Language Models (VLMs) for dynamic scene reasoning, different backbones for our segmentation module, as well as the impact of the video frame sampling rate.

5.2 Pose Optimization

We evaluate the performance of our pose optimization by refining the camera poses outputs from a wide range of state-of-the-art SfM methods and optimizing their poses.

Implementation Details. Following AnyCam [63], we use a two-stage refinement strategy to optimize the camera poses. The first stage employs a sliding-window optimization, where we use a window of eight frames, with a stride of one frame. We choose a small stride to densely refine the initial pose estimation. Within each window, we build short-term correspondences by sampling a uniform 16x16 grid of points and tracking them across the subsequent 8 frames using predicted optical flow. For each window, we perform 400 optimization steps. In the second stage, we optimize all the camera poses and static points jointly for 5000 steps to ensure long-range coherence.

Experimental Settings. We evaluate our pose optimization pipeline with DYNAPO by comparing with two categories of state-of-the-art methods. Optimization-based methods

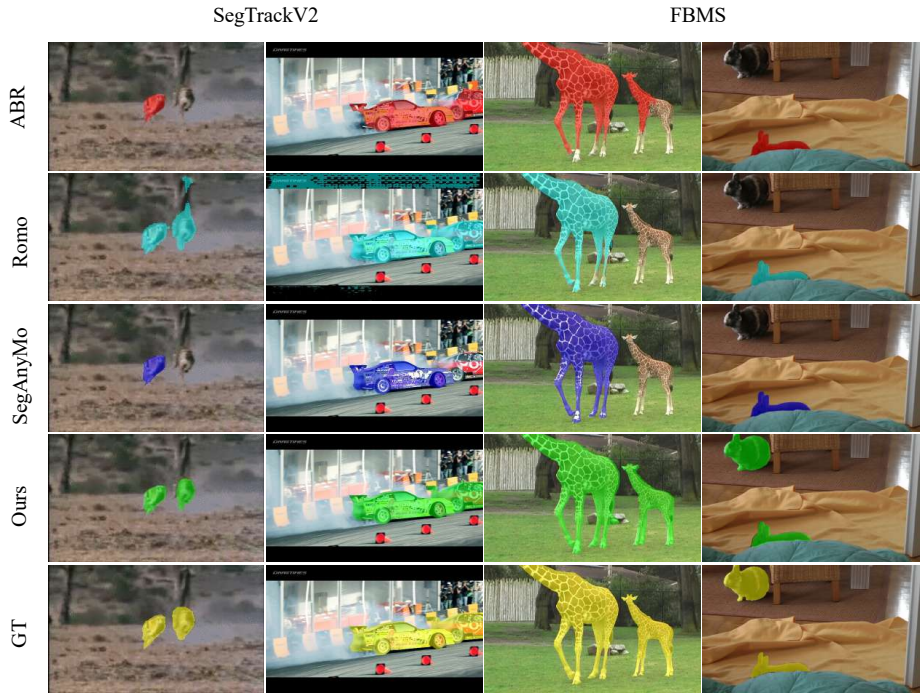

Fig. 6: Qualitative results on motion segmentation. DYNAPO generates precise masks for all the dynamic objects, while baselines can neglect some dynamic objects or generate inaccurate masks.

include MegaSaM [33], AnyCam [63], and Monst3r [73]. Feed-forword models include Easi3r [8] SpatialTrackerV2 [64], VGGT [56], Cut3r [58], and TTT3r [9]. For all the methods, we take their camera pose estimation as initialization and optimize with the masks predicted from DYNAPO. For a fair comparison, we also perform the same pose optimization process and do not apply the motion filtering (without using the mask). For the baselines that also predict the motion masks, including MegaSaM [33], AnyCam [63], Monst3r [73], and Easier [8], we perform additional experiments and optimize with their predicted motion masks for comparison.

Evaluation Datasets and Metrics. We evaluate our method on two benchmark datasets: Sintel [4] and LightSpeed [46]. During evaluation, we align the predicted trajectory with the ground-truth trajectory using Umeyama alignment [54] to handle the scale ambiguity of camera poses, following the practice. We compute three metrics: Absolute Trajectory Error (ATE) to measure the global consistency of the estimated trajectory, Relative Translation Error (RTE) to assess accuracy in translation, and Relative Rotation Error (RRE) to evaluate rotational.

Experimental Results. We provide quantitative results in Table 2 for the challenging LightSpeed dataset. Results on Sintel are provided in the Appendix. By leveraging masks from DYNAPO to effectively remove dynamic objects from bundle adjustment, our approach enhances the accuracy of various state-of-the-art methods for camera pose

Table 2: Quantitative comparison for camera pose estimation with different baselines on LightSpeed [46] dataset.

Methods that predict motion masks			
Method	ATE ↓	RTE ↓	RRE ↓
MegaSam [33]	0.073	0.032	0.773
+ BA w/o mask	0.090 (+0.017)	0.028 (-0.004)	1.003 (+0.230)
+ BA w/ their mask	0.072 (-0.001)	0.019 (-0.013)	2.504 (+1.731)
+ BA w/ our mask	0.052 (-0.021)	0.019 (-0.013)	0.910 (+0.137)
AnyCam [63]	0.139	0.032	1.101
+ BA w/o mask	0.133 (-0.006)	0.028 (-0.004)	1.057 (-0.044)
+ BA w/ their mask	0.131 (-0.008)	0.025 (-0.007)	0.914 (-0.187)
+ BA w/ our mask	0.125 (-0.014)	0.025 (-0.007)	0.884 (-0.217)
Monst3r [73]	0.112	0.029	0.998
+ BA w/o mask	0.096 (-0.016)	0.023 (-0.006)	0.962 (-0.036)
+ BA w/ their mask	0.093 (-0.019)	0.024 (-0.005)	0.969 (-0.029)
+ BA w/ our mask	0.092 (-0.020)	0.023 (-0.006)	0.935 (-0.063)
Easi3r [8]	0.214	0.058	2.153
+ BA w/o mask	0.135 (-0.079)	0.039 (-0.019)	1.076 (-1.077)
+ BA w/ their mask	0.150 (-0.064)	0.038 (-0.020)	1.079 (-1.074)
+ BA w/ our mask	0.118 (-0.096)	0.031 (-0.027)	1.027 (-1.126)

Methods that do not predict motion masks			
Method	ATE ↓	RTE ↓	RRE ↓
SpatialTrackerv2 [64]	0.188	0.094	2.487
+ BA w/o mask	0.168 (-0.020)	0.044 (-0.050)	1.573 (-0.914)
+ BA w/ our mask	0.103 (-0.085)	0.025 (-0.069)	1.249 (-1.238)
VGGT [56]	0.238	0.110	2.815
+ BA w/o mask	0.182 (-0.056)	0.053 (-0.057)	1.528 (-1.287)
+ BA w/ our mask	0.178 (-0.060)	0.051 (-0.059)	1.530 (-1.285)
Cut3r [58]	0.260	0.057	1.189
+ BA w/o mask	0.248 (-0.012)	0.050 (-0.007)	1.144 (-0.045)
+ BA w/ our mask	0.249 (-0.011)	0.050 (-0.007)	1.127 (-0.062)
TTT3r [9]	0.227	0.060	1.437
+ BA w/o mask	0.178 (-0.049)	0.037 (-0.023)	1.431 (-0.006)
+ BA w/ our mask	0.167 (-0.060)	0.036 (-0.024)	1.334 (-0.103)

estimation. The improvement is more significant on the challenging LightSpeed dataset, highlighting the efficacy of our method in dynamic environments. When comparing using the different sources of motion masks, optimizing with the masks from our DYNAPO achieves consistently better performance compared to optimizing with the masks from each baseline, highlighting the benefits of obtaining precise motion masks. A qualitative example is provided in Fig 7.

5.3 Depth Optimization

Implementation Details. We follow the settings from MegaSAM [33] to optimize video depth. We set the weights for the optical flow, prior, and temporal consistency losses to $\lambda_{\text{flow}} = 1.0$, $\lambda_{\text{prior}} = 1.0$, and $\lambda_{\text{temp}} = 0.2$, respectively. The optimization is conducted at a resolution of 336×144, and the final depth maps are upsampled to 672×288 for evaluation.

Evaluation Datasets and Metrics. We evaluate on three established benchmarks: MPI Sintel [4], DyCheck [16], and TUM-RGBD [50]. For Sintel, following prior work [72],

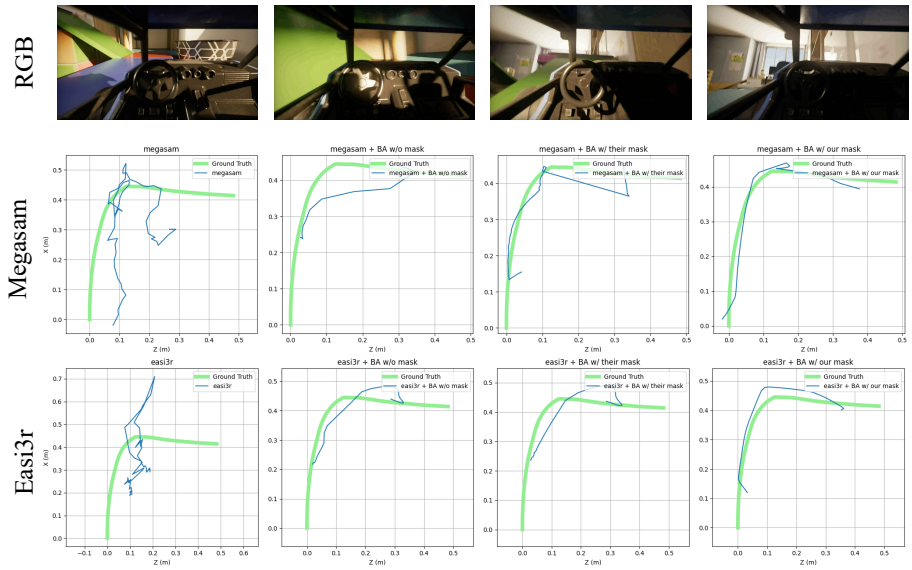

Fig. 7: Visualization of estimated camera trajectories.

Table 3: Quantitative comparisons of video depths. Lower is better for abs-rel and log-rmse, and higher is better for $\delta_{1.25}$.

	Sintel [4]			DyCheck [16]			TUM-RGBD [50]		
Method	abs-rel ↓	log-rmse ↓	$\delta_{1.25}$ ↑	abs-rel ↓	log-rmse ↓	$\delta_{1.25}$ ↑	abs-rel ↓	log-rmse ↓	$\delta_{1.25}$ ↑
MegaSam [33]	0.3081	0.4755	59.30	0.1411	0.2158	91.61	0.1777	0.2595	93.63
+ CVD w/ their mask	0.2369	0.4161	71.31	0.1083	0.2006	94.11	0.1598	0.2523	95.13
+ CVD w/ our mask	**0.2135**	**0.3957**	**72.77**	**0.1069**	**0.1998**	**94.53**	**0.1574**	**0.2519**	**95.17**

our evaluation is performed on the 18 sequences. For DyCheck, we use the ground truth camera poses and depth maps provided by Shape of Motion [57]. We compute standard depth estimation metrics for evaluation: absolute relative error (abs-rel), log root-mean-square error (log-rmse), and the threshold accuracy ($\delta_{1.25}$).

Experimental Settings. We compare with MegaSaM [33] by applying the same consistent video depth optimization process and only changing the source of the masks. We use the mask either from the original MegaSaM or use the mask predicted from DYNAPO.

Experimental Results. Table 3 shows the quantitative results, and Fig. 8 provides qualitative comparison. Our method consistently improves depth accuracy across all datasets. This performance gain is attributed to our discrete, semantics-aware motion mask, which replaces the learned, ambiguous uncertainty map used in the original CVD formulation. DYNAPO provides a clear signal that isolates static regions for the optimization, leading to more robust and accurate depth.

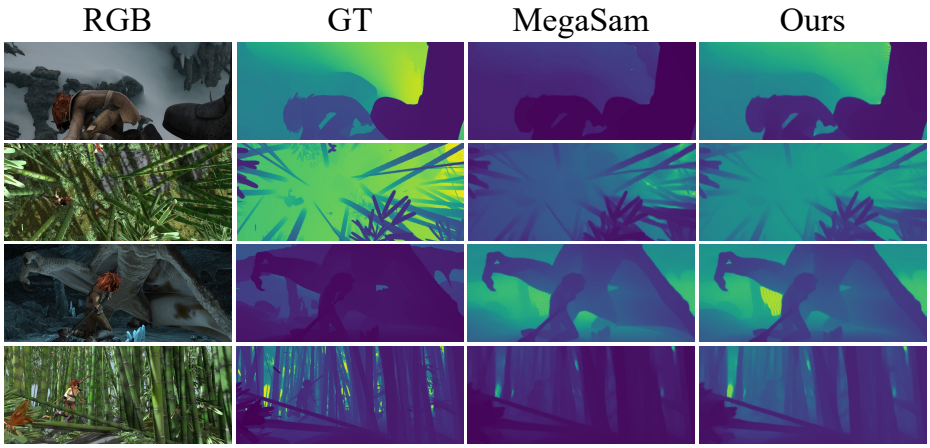

Fig. 8: Visual comparisons of depths. We compare with the original CVD [33] and CVD with DYNAPO.

5.4 4D Track Optimization

Experimental Settings. We follow Stereo4D's [24] experimental settings, replacing only its motion score with our dynamic mask (Sec. 4.3), and evaluate on the real-world videos it releases. The hyperparameters, initial camera parameters, depth, and optical flow are identical to Stereo4D.

Experimental Results. We provide qualitative comparisons with Stereo4D in Fig. 9. Our 4D track optimization produces more plausible trajectories on dynamic objects than the original Stereo4D [24], with more physically plausible motion and a more coherent relative structure. The key is our dynamic object masks: Stereo4D's motion score can be noisy, causing ambiguity in balancing the static and dynamic losses, whereas DYNAPO cleanly disentangles dynamic and static objects so the optimizer can apply the appropriate regularization to each region.

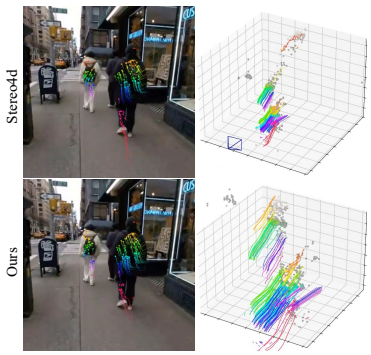

Fig. 9: Optimized tracks: DYNAPO's mask yields more plausible, coherent dynamic motion than Stereo4D.

6 Conclusion

Conclusion. In this work, we presented the Dynamic Prior (DYNAPO), a generalizable motion segmentation framework built on the reasoning of Vision Language Models and the segmentation capacity of SAM2. By producing accurate dynamic masks without task-specific training, DYNAPO enables more reliable camera pose estimation, depth reconstruction, and 4D trajectory recovery from casual real-world videos. Extensive ex-

periments show that it achieves state-of-the-art motion segmentation and significantly improves 3D scene understanding.

Limitations. While our method improves camera pose, depth, and tracking optimization, these tasks are performed independently; a cohesive framework for joint optimization is exciting future work. Since our dynamic mask only filters out dynamic objects, it can fail when a large dynamic object dominates the field of view. Finally, our text-prompted segmentation can be ambiguous when several objects share a semantic category (e.g., a group of animals); replacing this textual interface between the reasoning and segmentation models with implicit feature-level communication is a promising direction for future work.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54**(10), 105–112 (2011) 3
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025) 2, 4, 5
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025) 2, 4, 5
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: A. Fitzgibbon et al. (Eds.) (ed.) European Conf. on Computer Vision (ECCV). pp. 611–625. Part IV, LNCS 7577, Springer-Verlag (Oct 2012) 11, 12, 13, 8, 10
5. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., et al.: Sam 3: Segment anything with concepts. arXiv preprint arXiv:2511.16719 (2025) 7, 8
6. Chen, W., Chen, L., Wang, R., Pollefeys, M.: Leap-vo: Long-term effective any point tracking for visual odometry. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3
7. Chen, W., Zhang, G., Wimbauer, F., Wang, R., Araslanov, N., Vedaldi, A., Cremers, D.: Back on track: Bundle adjustment for dynamic scene reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4951–4960 (October 2025) 2
8. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Easi3r: Estimating disentangled motion from dust3r without training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9158–9168 (2025) 2, 3, 11, 12, 7, 8, 10
9. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Ttt3r: 3d reconstruction as test-time training. In: The Fourteenth International Conference on Learning Representations (2026) 2, 11, 12, 10
10. Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., et al.: Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. arXiv preprint arXiv:2412.05271 (2024) 5

11. Cho, S., Lee, M., Lee, S., Park, C., Kim, D., Lee, S.: Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision(WACV). pp. 5140–5149 (2023) 3
12. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025) 2, 4, 5
13. Dutt Jain, S., Xiong, B., Grauman, K.: Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3664–3673 (2017) 3, 9
14. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European conference on computer vision. pp. 834–849. Springer (2014) 2
15. Feng, H., Zhang, J., Wang, Q., Ye, Y., Yu, P., Black, M.J., Darrell, T., Kanazawa, A.: St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8503–8513 (2025) 4, 10, 11
16. 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) 12, 13
17. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: 2007 IEEE 11th International Conference on Computer Vision. pp. 1–8 (2007) 2
18. Goli, L., Sabour, S., Matthews, M., Brubaker, M.A., Lagun, D., Jacobson, A., Fleet, D.J., Saxena, S., Tagliasacchi, A.: Romo: Robust motion segmentation improves structure from motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6155–6164 (2025) 3, 9, 10, 7, 8
19. Hu, Y., Cheng, C., Yu, S., Guo, X., Wang, H.: Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 414–424 (2026) 3
20. Huang, J., Yang, S., Zhao, Z., Lai, Y.K., Hu, S.M.: Clusterslam: A slam backend for simultaneous rigid body clustering and motion estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5875–5884 (2019) 2
21. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., Ren, J., Xie, K., Biswas, J., Leal-Taixe, L., Fidler, S.: Vipe: Video pose engine for 3d geometric perception. In: NVIDIA Research Whitepapers arXiv:2508.10934 (2025) 2
22. Huang, N., Zheng, W., Xu, C., Keutzer, K., Zhang, S., Kanazawa, A., Wang, Q.: Segment any motion in videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 3406–3416 (2025) 3, 9, 10, 7, 8
23. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024) 2, 4, 9, 5
24. Jin, L., Tucker, R., Li, Z., Fouhey, D., Snavely, N., Holynski, A.: Stereo4d: Learning how things move in 3d from internet stereo videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10497–10509 (2025) 2, 6, 8, 9, 14, 13
25. Judd, K.M., Gammell, J.D.: Multimotion visual odometry. The International Journal of Robotics Research **43**(8), 1250–1278 (2024) 2
26. hong Kao, S., Tai, Y.W., Tang, C.K.: Cot-RVS: Zero-shot chain-of-thought reasoning segmentation for videos. In: The Fourteenth International Conference on Learning Representations (2026) 3

27. Karazija, L., Laina, I., Rupprecht, C., Vedaldi, A.: Learning segmentation from point trajectories. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024) 9, 10
28. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1611–1621 (2021) 4
29. Lamdouar, H., Xie, W., Zisserman, A.: Segmenting invisible moving objects. British Machine Vision Association (2021) 9, 10
30. Lee, Y.J., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: 2011 International conference on computer vision (ICCV). pp. 1995–2002. IEEE (2011) 3
31. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European Conference on Computer Vision. pp. 71–91. Springer (2024) 3
32. Li, F., Kim, T., Humayun, A., Tsai, D., Rehg, J.M.: Video segmentation by tracking many figure-ground segments. In: Proceedings of the IEEE international conference on computer vision (ICCV). pp. 2192–2199 (2013) 9
33. Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., Snavely, N.: Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10486–10496 (2025) 2, 3, 6, 7, 11, 12, 13, 14, 10
34. Lian, L., Wu, Z., Yu, S.X.: Bootstrapping objectness from videos by relaxed common fate and visual grouping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14582–14591 (June 2023) 3, 9, 10
35. Liu, Y., Dong, S., Wang, S., Yin, Y., Yang, Y., Fan, Q., Chen, B.: Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16651–16662 (2025) 3
36. McGann, D., Rogers, J.G., Kaess, M.: Robust incremental smoothing and mapping (risam). In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 4157–4163 (2023) 2
37. Meunier, E., Badoual, A., Bouthemy, P.: Em-driven unsupervised learning for efficient motion segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(4), 4462–4473 (2022) 3, 9, 10
38. Meunier, E., Bouthemy, P.: Unsupervised space-time network for temporally-consistent segmentation of multiple motions. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22139–22148 (2023) 9, 10
39. Mohamed, E., Ewaisha, M., Siam, M., Rashed, H., Yogamani, S., Hamdy, W., El-Dakdouky, M., El-Sallab, A.: Monocular instance motion segmentation for autonomous driving: Kitti instancemotseg dataset and multi-task baseline. In: 2021 IEEE Intelligent Vehicles Symposium (IV). pp. 114–121 (2021) 6, 7
40. Murai, R., Dexheimer, E., Davison, A.J.: Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16695–16705 (2025) 3
41. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. IEEE transactions on pattern analysis and machine intelligence **36**(6), 1187–1200 (2013) 3, 9, 4, 15
42. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Computer Vision and Pattern Recognition (2016) 9, 10, 11, 14, 15
43. Piccinelli, L., Yang, Y.H., Sakaridis, C., Segu, M., Li, S., Van Gool, L., Yu, F.: Unidepth: Universal monocular metric depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10106–10116 (2024) 2

44. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. International journal of computer vision **59**, 207–232 (2004) 3
45. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024) 2, 3, 5, 9, 7
46. Rockwell, C., Tung, J., Lin, T.Y., Liu, M.Y., Fouhey, D.F., Lin, C.H.: Dynamic camera poses and where to find them. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12444–12455 (2025) 11, 12, 10, 13
47. Schoenberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 3
48. Shen, G., Deng, T., Qin, X., Wang, N., Wang, J., Wang, Y., Chen, Y., Wang, H., Wang, J.: Mut3r: Motion-aware updating transformer for dynamic 3d reconstruction. arXiv preprint arXiv:2512.03939 (2025) 3
49. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM SIGGRAPH 2006 papers. pp. 835–846 (2006) 3
50. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 573–580 (2012) 12, 13
51. Sucar, E., Insafutdinov, E., Lai, Z., Vedaldi, A.: V-dpm: 4d video reconstruction with dynamic point maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14502–14511 (June 2026) 4, 10, 11
52. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In: Advances in Neural Information Processing Systems. vol. 34, pp. 16558–16569 (2021) 2, 3
53. Tokmakov, P., Alahari, K., Schmid, C.: Learning motion patterns in videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 3386–3394 (2017) 3
54. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE transactions on pattern analysis and machine intelligence **13**(4), 376–380 (1991) 11
55. Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: 2025 International Conference on 3D Vision (3DV). pp. 78–89. IEEE (2025) 2, 3
56. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025) 2, 3, 11, 12, 10
57. Wang, Q., Ye, V., Gao, H., Zeng, W., Austin, J., Li, Z., Kanazawa, A.: Shape of motion: 4d reconstruction from a single video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9660–9672 (2025) 13
58. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10510–10522 (2025) 2, 3, 11, 12, 10
59. Wang, S., Jiang, Z., Yang, X., Wang, X.: C4d: 4d made from 3d through dual correspondences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7570–7580 (2025) 4, 10
60. Wang, S., Yang, X., Shen, Q., Jiang, Z., Wang, X.: Gflow: Recovering 4d world from monocular video. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 7862–7870 (2025) 4, 7, 8

61. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20697–20709 (2024) 3
62. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025) 2, 5
63. Wimbauer, F., Chen, W., Muhle, D., Rupprecht, C., Cremers, D.: Anycam: Learning to recover camera poses and intrinsics from casual videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025) 2, 6, 10, 11, 12
64. Xiao, Y., Wang, J., Xue, N., Karaev, N., Makarov, Y., Kang, B., Zhu, X., Bao, H., Shen, Y., Zhou, X.: Spatialtrackerv2: 3d point tracking made easy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2025) 2, 11, 12, 10
65. Xie, J., Xie, W., Zisserman, A.: Segmenting moving objects via an object-centric layered representation. Advances in neural information processing systems **35**, 28023–28036 (2022) 3, 9, 10
66. Xie, J., Xie, W., Zisserman, A.: Appearance-based refinement for object-centric motion segmentation. In: European Conference on Computer Vision. pp. 238–256. Springer (2024) 9, 10
67. Xie, J., Yang, C., Xie, W., Zisserman, A.: Moving object segmentation: All you need is sam (and flow). In: Proceedings of the Asian conference on computer vision. pp. 162–178 (2024) 3
68. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. Advances in Neural Information Processing Systems **37**, 21875–21911 (2024) 2
69. Yang, Y., Loquercio, A., Scaramuzza, D., Soatto, S.: Unsupervised moving object detection via contextual information separation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 879–888 (2019) 3, 9, 10
70. Yao, D.Y., Zhai, A.J., Wang, S.: Uni4d: Unifying visual foundation models for 4d modeling from a single video. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1116–1126 (2025) 8
71. Yuan, H., Li, X., Zhang, T., Huang, Z., Xu, S., Ji, S., Tong, Y., Qi, L., Feng, J., Yang, M.H.: Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. arXiv (2025) 5, 9
72. Zhang, C.H., Cole, F., Li, A., Rubinstein, M., Snavely, N., Freeman, W.T.: Structure and motion from casual videos. In: European Conference on Computer Vision (ECCV) (2022) 4, 12
73. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Cole, F., Sun, D., Yang, M.H., et al.: Monst3r: A simple approach for estimating geometry in the presence of motion. In: International Conference on Learning Representations. vol. 2025, pp. 82863–82886 (2025) 2, 3, 11, 12, 10
74. Zhang, S., Ge, Y., Tian, J., Xu, G., Chen, H., Lv, C., Shen, C.: Pomato: Marrying pointmap matching with temporal motions for dynamic 3d reconstruction. Proceedings of the IEEE/CVF International Conference on Computer Vision pp. 5680–5689 (2025) 4, 10, 11
75. Zhao, W., Liu, S., Guo, H., Wang, W., Liu, Y.J.: Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In: European conference on computer vision (ECCV) (2022) 4
76. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025) 5