Mem4D: Decoupling Static and Dynamic Memory for Dynamic Scene Reconstruction

Xudong Cai[1], Shuo Wang[1], Peng Wang[1], Yongcai Wang[1*], Zhaoxin Fan[2*], Wanting Li[1], Tianbao Zhang[3], Jianrong Tao[4], Yeying Jin[5], Deying Li[1*]

[1]Renmin University of China,
[2]Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University,
[3]Shanghai Jiao Tong University,
[4]Zhejiang University,
[5]Tencent

Abstract

Reconstructing dense geometry for dynamic scenes from a monocular video is a critical yet challenging task. Recent memory-based methods enable efficient online reconstruction, but they fundamentally suffer from a Memory Demand Dilemma: The memory representation faces an inherent conflict between the long-term stability required for static structures and the rapid, high-fidelity detail retention needed for dynamic motion. This conflict forces existing methods into a compromise, leading to either geometric drift in static structures or blurred, inaccurate reconstructions of dynamic objects. To address this dilemma, we propose Mem4D, a novel framework that decouples the modeling of static geometry and dynamic motion. Guided by this insight, we design a dual-memory architecture: 1) The Transient Dynamics Memory (TDM) focuses on capturing high-frequency motion details from recent frames, enabling accurate and fine-grained modeling of dynamic content; 2) The Persistent Structure Memory (PSM) compresses and preserves long-term spatial information, ensuring global consistency and drift-free reconstruction for static elements. By alternating queries to these specialized memories, Mem4D simultaneously maintains static geometry with global consistency and reconstructs dynamic elements with high fidelity. Experiments on challenging benchmarks demonstrate that our method achieves state-of-the-art or competitive performance while maintaining high efficiency.

1 Introduction

The reconstruction of dense dynamic scene geometry from monocular video plays a key role in various real-world applications, such as autonomous driving (Cai et al. 2024a), virtual reality (Jiang et al. 2024), and robotic navigation (Wang et al. 2025c). This task is challenging due to the inherent complexity of dynamic scenes, where camera ego-motion is entangled with the independent movement of dynamic elements. Traditional SfM (Schonberger and Frahm 2016) and SLAM (Campos et al. 2021) methods struggle in dynamic scenes, which violate their core static-world assumption. Classical approaches handle dynamic scenes by segmenting and filtering out moving objects or modeling their motion independently (Yu et al. 2018), but this multi-stage process is complex and prone to error accumulation.

To overcome these limitations, recent research has shifted towards end-to-end learning, producing powerful pointmap regression frameworks. For instance, DUSt3R (Wang et al. 2024b) directly predicts pair-wise 3D pointmaps from image pairs and conducts a global alignment to assemble a global point cloud, demonstrating impressive performance for static scene reconstruction. Subsequent works like MonST3R (Zhang et al. 2024), D2USt3R (Han et al. 2025), and St4RTrack (Feng et al. 2025) adapt this architecture for dynamic scenes, achieving notable gains by fine-tuning on dynamic data, adding separate supervision for static and dynamic elements or jointly predicting pointmaps with 3D tracks. However, their reliance on the costly global alignment remains a bottleneck. To bypass this limitation, some methods process all images in a single pass (Wang et al. 2025a; Yang et al. 2025) at the cost of significant computational overhead and offline processing. Memory-based methods (Wang et al. 2025b; Wang and Agapito 2025) offer a more practical alternative, enabling online reconstruction of long video by incrementally updating a persistent feature memory. While efficient, their unified memory design struggles to effectively model dynamic scenes, leading to a trade-off between geometric stability and motion fidelity.

We find the core issue causing this trade-off is a "Memory Demand Dilemma": a unified memory is inherent inability to simultaneously provide the long-term consistency required for static structures and the high-fidelity plasticity needed for dynamic motion. As shown in Figure 1, we can consider the scenario of a person performing a street dance in front of a wall. To preserve the geometric stability of the wall, the memory must enforce long-term consistency, inevitably smearing the dancer's swift movements into a blurry mess. In contrast, to capture these fine movements with high fidelity, the memory must be highly responsive to transient details, ultimately causing the static wall to distort and drift. Consequently, methods relying on a unified memory face an inescapable compromise between geometric stability and motion fidelity.

Inspired by this analysis, we introduce Mem4D, a novel

*Corresponding authors

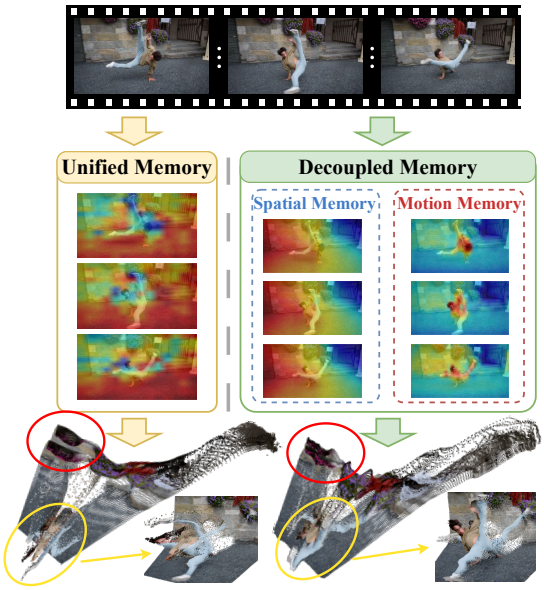

Figure 1: Illustration of the Memory Demand Dilemma. We visualize the memory feature maps of several frames. A Unified Memory (left) struggles to represent both static and dynamic elements, resulting in entangled memory features. This leads to geometric drift on the wall (red circle) and severe motion blur on the dancer (yellow circle). In contrast, our Decoupled Memory (right) resolves this conflict by explicitly decoupling the memory into a Spatial Memory for static structures and a Motion Memory for the dancer's movements. Our method yields a reconstruction that is both geometrically stable and rich in sharp dynamic details.

framework for online dynamic reconstruction that resolves the Memory Demand Dilemma. The basic idea is illustrated in Figure 1. Our key insight is that the static geometry and dynamic motion have different properties and should be modeled separately. Specifically, we introduce a dual-memory architecture that consists of: (1) a Transient Dynamics Memory (TDM) that captures high-frequency fine-grained motion details by computing 4D cost volumes between current and recent frames, ensuring short-term fidelity. (2) a Persistent Structure Memory (PSM) that maintains a feature bank of the scene's geometry over time, and employs a spatio-temporal attention mechanism to encode and compress the low-frequency static geometry, ensuring long-term stability. (3) a Temporal Context Aggregator (TCA) that aggregates rich spatio-temporal context from a local history into the current frame, providing a motion-aware input for our dual memories.

In contrast to previous methods, our decoupled design eliminates the memory conflict inherent in unified memories. This allows Mem4D to synthesize a superior reconstruction by fusing the global stability from the PSM with the sharp motion details from the TDM. Experiments on challenging benchmarks show that Mem4D achieves competitive or state-of-the-art performance in dynamic scene reconstruction, while keeping high efficiency.

The main contributions of our work are as follows:

- We introduce Mem4D, a novel online dual-memory framework for dynamic scene reconstruction by decoupling static geometry and dynamic motion.
- We introduce the Transient Dynamics Memory to preserve high-fidelity motion details and the Persistent Structure Memory with spatio-temporal compression to ensure long-term geometric stability.
- Our method achieves state-of-the-art or competitive performance on challenging benchmarks, delivering superior accuracy for static structures while preserving fine-grained dynamic motion.

2 Related Work

2.1 Static 3D Reconstruction

Classical Static 3D reconstruction primarily relies on Structure from Motion (SfM) (Schonberger and Frahm 2016) and Simultaneous Localization and Mapping (SLAM) (Campos et al. 2021). These methods are largely rooted in multi-stage geometric pipelines such as image matching (Lowe 2004) and bundle adjustment (Agarwal et al. 2010). Despite their widespread success, they remain susceptible to failure in degenerate scenarios (e.g., low-texture regions or views with minimal overlap).

Deep learning has progressively revolutionized the field. Initially, this involved replacing handcrafted modules with learnable ones, such as feature matching (Cai et al. 2024b). While improving specific stages, the overall process remained fragmented and susceptible to error propagation. This motivated a paradigm shift towards end-to-end frameworks. While early works (Wang et al. 2024a) introduced differentiable SfM pipelines, they often struggled with generalization and global consistency. A significant breakthrough came with pointmap regression models like DUSt3R (Wang et al. 2024b). They directly map images to 3D pointmaps using transformer (Vaswani et al. 2017), achieving impressive results on static scenes. However, their pairwise design requires a costly global alignment to assemble multiple pointmaps, limiting their scalability.

To address this, some works process all images in a single pass (Wang et al. 2025a; Yang et al. 2025; Wang et al. 2025d) at the cost of significant computational overhead and offline processing. In contrast, online methods have emerged as a more practical paradigm. Spann3R (Wang and Agapito 2025) pioneered this direction by introducing an external memory bank that is incrementally updated as new frames arrive. Subsequent works (Wang et al. 2025b; Liu et al. 2025; Wu et al. 2025; Cabon et al. 2025) focused on developing more elaborate designs for the unified memory architecture. While effective for static scenes, these methods struggle with dynamic scenes. We address this limitation by introducing a dual-memory architecture that decouples the modeling of static geometry and dynamic motion, achieving superior performance.

2.2 Dynamic Scene Reconstruction

Traditional approaches for dynamic scenes often rely on complex multi-stage pipelines involving explicit object seg-

mentation and tracking (Yu et al. 2018) or per-scene optimization with depth priors (Zhang et al. 2022; Li et al. 2025). While capable, their complexity makes them prone to error accumulation.

To address this issue, the recent paradigm of end-to-end pointmap regression has been adapted for dynamic scenes in various ways, such as introducing new training strategies (Zhang et al. 2024; Han et al. 2025), geometry and generative priors (Jiang et al. 2025), attention adaptation during inference (Chen et al. 2025) or joint tracking formulations (Feng et al. 2025; Zhang et al. 2025). However, these methods still require a costly global alignment, limiting their practicality for long videos. To bypass this limitation, recent works have begun to incorporate memory design: Driv3R (Fei et al. 2024) extends spatial memory to the temporal dimension, CUT3R (Wang et al. 2025b) uses recurrent states to encode scene history, and MUSt3R (Cabon et al. 2025) leverages a multi-layer memory mechanism to reduce the computational complexity. Concurrent work Point3R (Wu et al. 2025) proposes explicit 3D pointer-based memories and StreamVGGT (Zhuo et al. 2025) leverages implicit key-value caches from causal transformers.

Despite these varied and elaborate designs, a common limitation persists: they all rely on a single, unified memory to encode both static geometry and dynamic motions. This unified memory inevitably faces a fundamental conflict between preserving long-term geometric stability and capturing high-fidelity motion. We propose a novel dual-memory architecture to resolve this conflict by decoupling the modeling of static geometry and dynamic motion, resulting in superior reconstruction performance.

2.3 Memory Bank in Computer Vision

Memory banks are widely studied in computer vision for their ability to store and retrieve historical information. This ability makes them suitable for sequential tasks like video object segmentation (Cheng and Schwing 2022), video recognition (Wu et al. 2022), optical flow estimation (Dong and Fu 2024) and video generation (Zhang and Agrawala 2025). For instance, FramePack (Zhang and Agrawala 2025) introduces a memory structure that progressively compresses input frames based on their importance to improve video generation fidelity. This concept shares a similar spirit with our Persistent Structure Memory, which compresses the historical memory features based on their temporal distance for long-term stability. However, FramePack compresses the input frames to improve efficiency, while our PSM compresses past pointmaps for geometry stability.

3 Method

Given a monocular video stream $\mathcal{I} = \{I_1, I_2, \dots, I_T\}$ of a dynamic scene, our goal is to reconstruct a sequence of dense 3D pointmaps $\mathcal{X} = \{X_1, X_2, \dots, X_T\}$ and camera parameters $\mathcal{P} = \{(K_1, T_1), (K_2, T_2), \dots, (K_T, T_T)\}$ within a consistent global coordinate frame and K_i is the camera intrinsic. Figure 2 illustrates the overview of Mem4D. Our key contribution is a dual-memory architecture that models static geometry and dynamic motion separately, allowing the model to draw upon distinct information sources for geometric stability and motion fidelity.

In the following sections, we first introduce the Temporal Context Aggregator, which enriches each input frame with recent motion context. Next, we elaborate on the core of our framework: the Transient Dynamics Memory for capturing high-fidelity dynamics, and the Persistent Structure Memory for maintaining long-term structural stability. Finally, we present our training strategy and objectives.

3.1 Temporal Context Aggregator (TCA)

A single video frame is often insufficient to resolve motion ambiguities. Therefore, to create a more rich input for the subsequent memory fusion, we first enrich the current frame's features with its temporal neighborhood using TCA.

For each incoming frame I_t, we first extract a feature map $F_t \in \mathbb{R}^{H \times W \times C}$ using a ViT (Dosovitskiy et al. 2020) encoder. The TCA enriches F_t by aggregating context from a local window of k_t past frames $\{F_{t-j}\}_{j=1}^{k_t}$. To efficiently summarize this history, we employ a distance-aware temporal compression scheme. Each past feature map F_{t-j} is spatially downsampled using a 2D convolution Conv_{s_j}, where the stride $s_j = \phi(j)$ is a function of its temporal distance j:

$$s_j = \phi(j) = \begin{cases} 1, & \text{if } 0 \le j < 2 \\ 2, & \text{if } 2 \le j < 4 \\ 4, & \text{if } j \ge 4 \end{cases} \tag{1}$$

The compressed feature map $\tilde{F}_{t-j}$ is thus given by:

$$\tilde{F}_{t-j} = \mathrm{Conv}_{\phi(j)}(F_{t-j}) \quad \forall j \in \{1, \dots, k_t\} \tag{2}$$

This scheme progressively summarizes older features while preserving the high-fidelity detail of recent ones. Then, the current image tokens F_t are concatenated with the compressed past tokens $\left\{\tilde{F}_{t-j}\right\}_{j=1}^{k_t}$ and processed by four self-attention layers. To accurately capture the spatio-temporal relationships that are crucial for motion, we incorporate 3D Rotary Position Embeddings (3DRoPE) (Su et al. 2024) to jointly encode each token's position. The self-attention allows the current frame tokens to draw relevant context from the aggregated history, producing the enriched tokens T'_t.

3.2 Decoupled Static and Dynamic Memory

The core of Mem4D is a dual-memory architecture that models static geometry and dynamic motion separately, enabling drift-free and motion-fidelity reconstruction. We first describe how each memory is constructed, then describe the memory readout process.

Transient Dynamics Memory (TDM) Motion consists of high-frequency signals whose relevance is local in time. To capture fine-grained motion details without loss, we introduce the Transient Dynamics Memory. The TDM stores k_d motion-aware feature maps, denoted as $\mathcal{M}^D \in \mathbb{R}^{k_d \times H \times W \times C_m}$. Each feature map is derived from a 4D correlation volume $C_{t,t-j}$. $C_{t,t-j}$ is computed by taking the dot

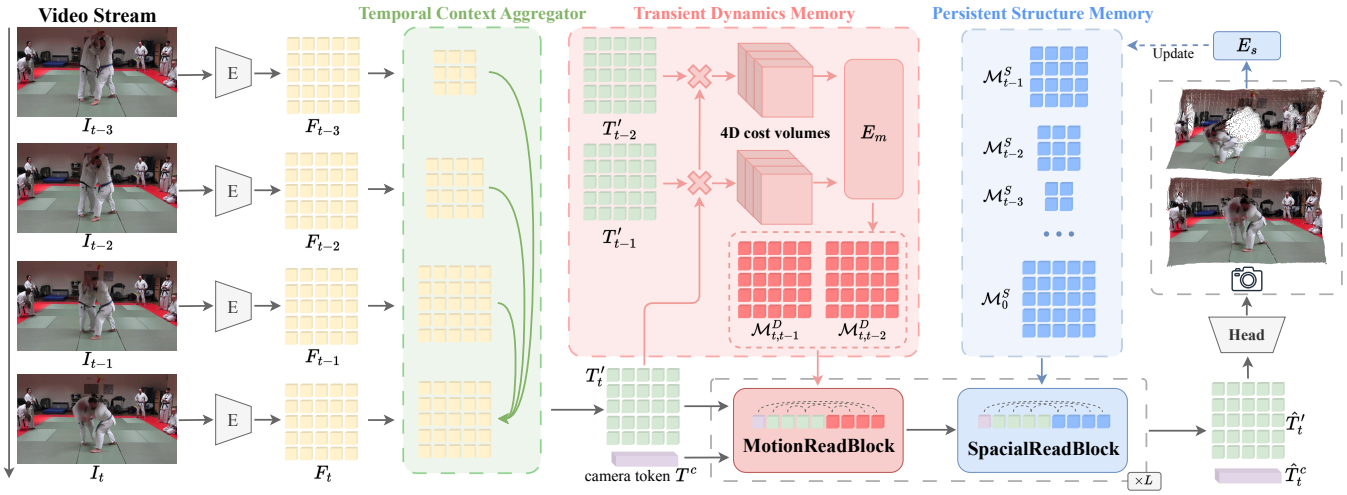

Figure 2: Framework of Mem4D. For each incoming frame I_t, the TCA first enriches the ViT-encoded F_t by aggregating features from a local history to create a motion-aware representation T'_t. Then, T'_t and a learnable camera token T^c are iteratively refined by interleaved readouts from the TDM and PSM via self-attention. The TDM is computed on-the-fly from 4D correlation volumes with recent feature maps, capturing fine-grained motion cues. Concurrently, the PSM maintains a long-term FIFO feature bank of the scene's geometry to ensure global consistency. All features but the initial one $\mathcal{M}_0^S$ are compressed based on their temporal distance to t. Finally, the refined tokens $\hat{T}_t^c$ and $\hat{T}'_t$ are passed to prediction heads to output the global and self-view pointmaps, along with camera parameters for the current frame. The global pointmap is encoded to update the PSM.

product between the current features T'_t and a past feature T'_{t-j} from the set of past frames $\{T'_{t-j}\}_{j=1}^{k_d}$:

$$C_{t,t-j} = T'_t \times (T'_{t-j})^T \in \mathbb{R}^{H\times W\times H\times W}. \quad (3)$$

Following RAFT (Teed and Deng 2020), we build a multi-scale correlation pyramid $\{C^l_{t,t-j}\}_{l=1}^{L}$ by pooling the last two dimensions of the correlation volume at different scales. This pyramid, which captures both large and small displacements, is then concatenated along the channel dimension and projected by an MLP to produce a motion feature map:

$$M_{t,t-j} = \text{MLP}\left(C^1_{t,t-j} \oplus C^2_{t,t-j}, \ldots, C^L_{t,t-j}\right). \quad (4)$$

where $\oplus$ denotes the concatenation operation. A motion feature encoder E_m consisting of four self-attention layers with 3DRoPE, further refines $M_{t,t-j}$ to yield the final motion memory feature $\mathcal{M}^D_{t,t-j} = E_m(M_{t,t-j}) \in \mathbb{R}^{H\times W\times C_m}$. Unlike a persistent memory, the TDM is computed on-the-fly at each timestep t, storing motion-aware features relevant only to the current context.

Persistent Structure Memory (PSM) To ensure global consistency and prevent long-term drift, the Persistent Structure Memory maintains a feature bank of the scene's structure over time, denoted as $\mathcal{M}^S \in \mathbb{R}^{k_s\times H\times W\times C_s}$. At each timestep t, the PSM is updated by appending a new feature map $\mathcal{M}^S_t$. $\mathcal{M}^S_t$ is generated using a lightweight encoder E_s which is built primarily of four self-attention blocks. E_s processes the final predicted global pointmap $\hat{X}^{global}_t$ and transforms it into a compact feature representation that encapsulates the scene's essential geometric structure: $\mathcal{M}^S_t = E_s\left(\hat{X}^{global}_t\right) \in \mathbb{R}^{H\times W\times C_s}$. This memory is managed as a First-In-First-Out (FIFO) queue of size k_s. Note that the first memory frame $\mathcal{M}^S_0$ is never removed to serve as a stable global anchor against drift.

Memory Readout The final reconstruction is produced by an iterative fusion decoder. It progressively refines the current frame's tokens T'_t and a learnable camera token T^c_t over L stages of interleaved readouts from the TDM and PSM, as detailed in Algorithm 1. Each block first employs a MotionReadBlock to refine the current tokens by attending to the TDM to fuse high-frequency motion features. Subsequently, a SpatialReadBlock updates the tokens by querying the PSM, grounding the prediction in a stable, long-term geometric context. Both blocks are implemented as self-attention layers enhanced with 3DRoPE to effectively encode spatio-temporal relationships. The alternating design enables the model to first resolve fine-grained dynamics and then anchor them within a globally consistent framework.

Prior to the readout stage, the two memories are handled differently. The TDM is directly used for readout to retain maximum detail as it captures high-frequency, transient motion information. In contrast, the PSM undergoes a temporally-aware compression to filter noise and reduce redundancy, as it stores low-frequency, long-term geometric information. We apply 3D convolutions with varying kernel sizes s_d to the memory frame $\mathcal{M}^S_j$ based on its temporal distance $d = t - j$ to the current frame t:

$$s_d = \begin{cases} (4,8,8), & \text{if } d \geq 6 \\ (2,4,4), & \text{if } 4 \leq d < 6 \\ (1,2,2), & \text{if } 2 \leq d < 4 \\ (1,1,1), & \text{otherwise} \end{cases} \quad (5)$$

This policy enables the model to effectively keep long-term

Alignment	Method	Type	Sintel		BONN		KITTI	
			Abs Rel ↓	δ<1.25 ↑	Abs Rel ↓	δ<1.25 ↑	Abs Rel ↓	δ <1.25 ↑
Per-Scene	DUSt3R-GA	PW	0.656	<u>45.2</u>	<u>0.155</u>	<u>83.3</u>	**0.144**	<u>81.3</u>
	MASt3R-GA	PW	0.641	43.9	0.252	70.1	0.183	74.5
	MonST3R-GA	PW	**0.378**	**55.8**	**0.067**	**96.3**	0.168	74.4
	Fast3R	FF	<u>0.628</u>	43.8	0.193	77.3	<u>0.152</u>	**84.1**
	Spann3R	OL	0.622	42.6	0.144	81.3	0.198	73.7
	CUT3R	OL	**0.421**	**47.9**	<u>0.078</u>	<u>93.7</u>	**0.118**	**88.1**
	Ours	OL	<u>0.520</u>	<u>43.1</u>	**0.072**	**95.7**	<u>0.140</u>	<u>82.0</u>
Metric	MASt3R-GA	PW	<u>1.022</u>	14.3	0.272	70.6	0.467	15.2
	CUT3R	OL	1.029	**23.8**	<u>0.103</u>	<u>88.5</u>	<u>0.122</u>	**85.5**
	Ours	OL	**0.846**	<u>22.3</u>	**0.086**	**95.7**	**0.117**	<u>79.5</u>

Table 1: Video Depth Evaluation. We compare scale-invariant depth (Per-Scene alignment) and metric depth (no alignment) results on Sintel, Bonn, and KITTI datasets. PW denotes pair-wise methods, OL denotes online methods, FF denotes feed-forward methods, and GA denotes global alignment. The best results are in **bold** and the second best are <u>underlined</u>.

structural cues and preserve high-fidelity detail for recent ones. Crucially, the initial frame $\mathcal{M}_0^S$ is never compressed, serving as a stable, high-resolution anchor to mitigate drift. After L iterations, the current frame tokens $\hat{T}_t' = T_t'^{(L)}$

Algorithm 1: Iterative Memory Readout

Input: Current frame tokens $T_t'^{(0)}$, Pose Token $T_t^{c(0)}$, TDM $\mathcal{M}^{D(0)}$ and compressed PSM $\mathcal{M}^{S(0)}$.
for $l = 1$ **to** L **do**
  $\tilde{T}_t^{c(l)}, \tilde{T}_t'^{(l)}, \mathcal{M}_t^{D(l)} \leftarrow$MotionReadBlock($T_t^{c(l-1)}$, $T_t'^{(l-1)}$, $\mathcal{M}^{D(l-1)}$)
  $T_t^{c(l)}, T_t'^{(l)}, \mathcal{M}_t^{S(l)} \leftarrow$SpacialReadBlock($\tilde{T}_t^{c(l)}$, $\tilde{T}_t'^{(l)}$, $\mathcal{M}^{S(l-1)}$)
end for
return $T_t^{c(L)}, T_t'^{(L)}$

are processed by two DPT heads (Ranftl, Bochkovskiy, and Koltun 2021) to predict global and self-view coordinate pointmaps $\hat{X}_t^{global}$ and $\hat{X}_t^{self}$, along with their corresponding confidence maps C_t^{global} and C_t^{self}. The camera token $\hat{T}_t^c = T_t^{c(L)}$ is fed to an MLP head to regress the intrinsics (FoV) and camera pose (quaternion and translation vector):

$$\begin{aligned} \hat{X}_t^{global}, C_t^{global} &= \text{Head}_{global}\left(\hat{T}_t'\right), \\ \hat{X}_t^{self}, C_t^{self} &= \text{Head}_{self}\left(\hat{T}_t'\right), \\ \hat{K}_t, \hat{T}_t &= \text{Head}_{camera}\left(\hat{T}_t^c\right). \end{aligned} \tag{6}$$

3.3 Training Objective and Strategy

Our model is trained end-to-end on sequences of N images. The total loss function is a weighted sum of losses for pointmap reconstruction and camera parameter estimation.

3D regression loss. We denote the predicted pointmaps as $\mathcal{X} = \left\{\hat{X}_1^{global}, \ldots, \hat{X}_N^{global}, \hat{X}_1^{self}, \ldots, \hat{X}_N^{self}\right\}$ and their corresponding confidence scores as $\mathcal{C}$. We adopt the confidence-aware regression loss following MASt3R (Leroy, Cabon, and Revaud 2024).

$$\mathcal{L}_{\text{conf}} = \sum_{(\hat{\boldsymbol{x}},c)\in(\hat{\mathcal{X}},\mathcal{C})} \left(c \cdot \left\| \frac{\hat{\boldsymbol{x}}}{s} - \frac{\boldsymbol{x}}{s} \right\|_2 - \alpha \log c \right) \tag{7}$$

The scale normalization factor s is from the groundtruth, enabling the model to learn metric-scale pointmaps.

Camera Loss. The camera parameter is supervised with two components. First, we apply a L2 loss on the predicted absolute pose (quaternion $\hat{\boldsymbol{q}}_t$ and translation $\hat{\boldsymbol{\tau}}_t$) and camera intrinsic $\hat{K}$ for each frame against their ground-truth values:

$$\mathcal{L}_{\text{abspose}} = \sum_{t=1}^{N} \left(\|\hat{\boldsymbol{q}}_t - \boldsymbol{q}_t\|_2 + \left\| \frac{\hat{\boldsymbol{\tau}}_t}{s} - \frac{\boldsymbol{\tau}_t}{s} \right\|_2 + \beta \left\| K_t - \hat{K}_t \right\| \right) \tag{8}$$

To improve temporal consistency, we introduce an L2 loss on the relative pose between consecutive frames. We compute the predicted relative pose $\hat{T}_{t,t-1} = \hat{T}_t \times \hat{T}_{t-1}^{-1}$ and supervise it using the ground-truth relative:

$$\mathcal{L}_{\text{relpose}} = \sum_{t=2}^{N} \left(\|\hat{\boldsymbol{q}}_{t,t-1} - \boldsymbol{q}_{t,t-1}\|_2 + \left\| \frac{\hat{\boldsymbol{\tau}}_{t,t-1}}{s} - \frac{\boldsymbol{\tau}_{t,t-1}}{s} \right\|_2 \right) \tag{9}$$

The final training objective is the weighted sum of these components:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{conf}} + \lambda_2 \mathcal{L}_{\text{abspose}} + \lambda_3 \mathcal{L}_{\text{relpose}} \tag{10}$$

Curriculum Training We train Mem4D on a diverse mix of 10 datasets, including both synthetic and real-world data such as ARKitScenes (Baruch et al. 2021) and Spring (Mehl et al. 2023). See the supplementary material for more details. The image resolution is set to a maximum of 512 pixels on the longer side. The training is divided into two stages. We first train the model on fixed-length sequences by sampling 11 frames per video sequence. In the second stage, we train the model on longer sequences by randomly sampling 11 to 48 frames from each video sequence, improving the model's ability to handle longer videos.

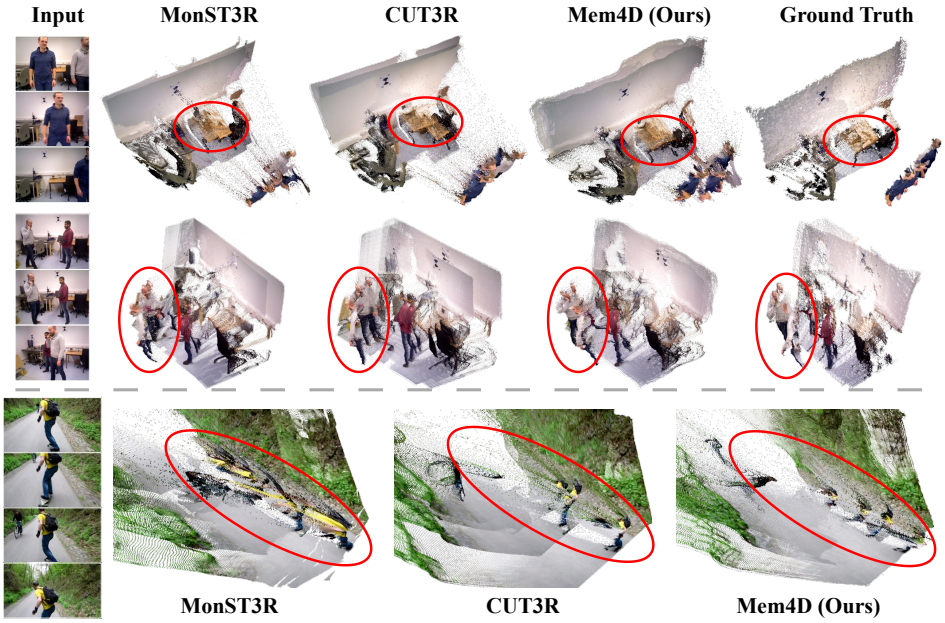

Figure 3: Qualitative results of dynamic reconstruction on Bonn (top) and DAVIS (down) dataset. Compared to MonST3R and CUT3R, our Mem4D achieves the best qualitative results.

Method	Type	Stereo4D			Sintel			TUM-dynamics			FPS ↑
		ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓	
DUSt3R-GA	PW	0.931	0.896	4.541	0.417	0.250	5.796	0.083	0.017	3.567	**0.33**
MASt3R-GA	PW	**0.382**	**0.1951**	**0.687**	0.185	0.060	1.496	**0.038**	**0.012**	**0.448**	0.15
MonST3R-GA	PW	0.414	0.631	1.052	**0.111**	**0.044**	**0.869**	0.098	0.019	0.935	0.11
Spann3R	OL	0.653	0.321	1.353	0.329	0.110	4.471	0.056	0.021	0.591	9.07
CUT3R	OL	0.506	0.244	0.730	**0.213**	**0.066**	**0.621**	**0.046**	**0.015**	**0.473**	**17.91**
Ours	OL	**0.495**	**0.229**	**0.641**	0.263	0.091	0.812	0.061	0.020	0.517	16.12

Table 2: Camera Pose Estimation Evaluation on ScanNet, Sintel, and TUM-dynamics datasets. PW denotes pair-wise method, OL denotes online method and GA denotes global alignment. The best results are in **bold** and the second best are underlined. We report FPS on Stereo4D at 512×384 resolution for all methods, except Spann3R which only supports 224×224 images.

Implementation Details The image encoder is a ViT-Large (Dosovitskiy et al. 2020) initialized from CUT3R (Wang et al. 2025b) weights and is frozen during training. The memory feature of TDM and PSM is set to 768 channels. The window size k_t of TCA is 5 and the memory sizes k_d and k_s are set to 2 and 100, respectively. We use AdamW (Loshchilov and Hutter 2017) optimizer with an initial learning rate of $1e^{-5}$. Linear warmup and cosine decay are applied. Training is conducted on 8 NVIDIA A100 GPUs with a batch size of 4 per GPU.

4 Experiment

We evaluate our method on video depth estimation, camera pose estimation, and 3D scene reconstruction tasks. We compare Mem4D against leading methods which can be broadly categorized into two groups: (1) Offline methods need the entire set of views to form a complete reconstruction, such as DUSt3R (Wang et al. 2024b), MASt3R (Leroy, Cabon, and Revaud 2024), MonST3R (Zhang et al. 2024) and Fast3R (Yang et al. 2025); (2) Online methods that process the input frames sequentially, such as Spann3R (Wang and Agapito 2025) and CUT3R (Wang et al. 2025b).

4.1 Video Depth Estimation Performance

Following the protocol of (Zhang et al. 2024; Wang et al. 2025b), we evaluate depth prediction for long dynamic videos on KITTI (Geiger et al. 2013), Sintel (Butler et al. 2012) and Bonn (Palazzolo et al. 2019) benchmarks. We report the absolute relative error (Abs Rel) and percentage of inlier points $\delta < 1.25$. We evaluate Mem4D both with alignment (Per-Scene scale) and without alignment (Metric).

As shown in Table 1, under Per-Scene alignment, Mem4D performs competitively against CUT3R, notably surpassing it on the challenging Bonn dataset (0.072 vs. 0.078 Abs Rel), and clearly outperforms Spann3R across all benchmarks. While MonST3R achieves state-of-the-art results, its performance relies on a costly global alignment process and requires additional inputs such as optical flow (Teed and Deng 2020) and semantic segmentation (Ravi et al. 2024).

Method	7-Scenes				NRGBD			
	Acc ↓		Comp ↓		Acc ↓		Comp ↓	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
DUST3R-GA	**0.146**	**0.077**	0.181	0.067	**0.144**	**0.019**	**0.154**	**0.018**
MonST3R-GA	0.248	0.185	0.266	0.167	0.272	0.114	0.287	0.110
Fast3R	0.155	0.104	**0.125**	**0.052**	0.366	0.239	0.198	0.097
Spann3R	0.298	0.226	0.205	0.112	0.416	0.323	0.417	0.285
CUT3R	**0.126**	**0.047**	**0.154**	**0.031**	**0.099**	**0.031**	**0.076**	**0.026**
Ours	0.185	0.137	0.178	0.082	0.271	0.196	0.212	0.113

Table 3: 3D Reconstruction Evaluation on 7-scenes and NRGBD datasets. The best results are in **bold** and the second best are underlined.

More importantly, in the challenging Metric (no alignment) evaluation, Mem4D establishes a new state of the art, significantly outperforming CUT3R on major benchmarks. On Sintel and Bonn, we improve the absolute relative error by 21.6% and 19.7%, respectively. This substantial improvement validates the effectiveness of our design choices. Figure 3 shows three qualitative examples, demonstrating the superior performance of our method.

4.2 Camera Pose Estimation Performance

Following standard evaluation protocols (Zhang et al. 2024; Wang et al. 2025b), we evaluate camera pose estimation on the Sintel (Butler et al. 2012) and TUM dynamics (Sturm et al. 2012). We also conduct evaluations on the challenging Stereo4D (Jin et al. 2024) dataset. For Stereo4D, we report results on 50 sequences randomly sampled from the official test split, sampling keyframes every 5 images. For all evaluations, we report Absolute Translation Error (ATE), Relative Translation Error (RPE trans), and Relative Rotation Error (RPE rot) after Sim(3) alignment with the ground truth.

It is crucial to note that Mem4D is a fully online method without any post-processing, while top methods like MonST3R-GA achieve superior pose accuracy via costly, offline global alignment. As shown in Table 2, Mem4D achieves competitive performance against CUT3R, for instance, delivering a strong accuracy on Stereo4D. We argue this trade-off enables significant efficiency gains: Mem4D runs at 16 FPS, making it suitable for real-time applications.

4.3 3D Reconstruction Performance

To demonstrate Mem4D's generalizability on the static scenes, we evaluate our method on the 7-scenes (Shotton et al. 2013) and NRGBD (Azinović et al. 2022) benchmarks. Following the evaluation protocols (Wang et al. 2025b), we use sparse inputs: 3-5 frames for 7-scenes and 2-4 frames for NRGBD and report accuracy (Acc) and completion (Comp). Table 3 shows Mem4D achieves results comparable to leading online methods, demonstrating the strong generalizability of our method for static scenes.

4.4 Ablation Study

We validate our design choices with the ablation study presented in Table 4. This table evaluates several smaller-scale model variants on the Sintel dataset for video depth and pose accuracy, with each variant corresponding to a specific row. Row one is our full implementation. All models were trained on ARKitScenes (Baruch et al. 2021) and PointOdyssey (Zheng et al. 2023) under consistent settings.

(1)	(2)	(3)	(4)	(5)	Poses			Depth	
					ATE ↓	RTE ↓	RRE ↓	Abs Rel ↓	δ<1.25 ↑
✓	✓	✓	✓	✓	0.451	0.172	1.173	0.513	34.27
	✓	✓	✓	✓	0.466	0.150	1.267	0.518	33.75
		✓	✓	✓	0.491	0.197	1.367	0.520	33.37
			✓	✓	0.465	0.184	1.427	0.573	32.40
				✓	0.501	0.198	1.389	0.569	32.10
					0.524	0.194	1.677	0.550	31.99

Table 4: Ablation study on Sintel dataset.

(1) Impact of Second Stage Training. The variant in the second row omits the second stage training on longer sequences, resulting in a general performance degradation. This highlights the importance of long-range finetuning to handle extended sequences and mitigate cumulative drift.

(2) Effect of Relative Pose Loss. In the third row, removing $\mathcal{L}_{relpose}$ impairs pose estimation, highlighting its crucial role in ensuring trajectory consistency.

(3) Contribution of TDM. The variant in the fourth row removes the TDM, resulting in the most severe drop in depth accuracy. It shows that motion information is essential for capturing high-frequency motion details.

(4) Influence of PSM. As shown in the fifth row, removing PSM severely degrades performance across most metrics. This result underscores the PSM's critical role as a long-term geometric anchor against global drift.

(5) Importance of TCA. The variant in the sixth row removes the TCA. It leads to significant drop in pose accuracy. This demonstrates that providing the memories with a rich temporal context is vital for resolving motion ambiguities.

5 Conclusion

In this work, we present Mem4D, a novel online framework for dynamic scene reconstruction from monocular video. Mem4D resolves the Memory Demand Dilemma in existing memory-based methods by decoupling the modeling of low-frequency static geometry and high-frequency dynamic motion into two separate memories: the Persistent Structure Memory that ensures long-term geometric stability, and the Transient Dynamics Memory that captures fine-grained motion details. Extensive experiments demonstrate that Mem4D achieves impressive performance across multiple tasks, validating the effectiveness of our method.

Limitations As a feed-forward online method, Mem4D can still be susceptible to drift accumulation over extremely long video sequences. Integrating a Bundle Adjustment module could further enhance long-term accuracy. Besides, our approach relies on supervised training, yet dense 4D ground-truth data is scarce and difficult to acquire, which limits scalability and generalization. Exploring self-supervised training strategies could help address this issue.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61972404, No. 12071478 and No. 62441617. It was supported by the Postdoctoral Fellowship Program and China Postdoctoral Science Foundation under Grant No. 2024M764093 and Grant No. BX20250485, the Beijing Natural Science Foundation under Grant No. 4254100, the Fundamental Research Funds for the Central Universities under Grant No. KG16336301, and by Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing. Specifically, Dr. Wang is supported in part by the National Natural Science Foundation of China Grant No. 61972404, Public Computing Cloud, Renmin University of China, and the Blockchain Lab. School of Information, Renmin University of China. Dr. Li is supported in part by the National Natural Science Foundation of China Grant No. 12071478.

References

Agarwal, S.; Snavely, N.; Seitz, S. M.; and Szeliski, R. 2010. Bundle adjustment in the large. In *European conference on computer vision*, 29–42. Springer.

Azinović, D.; Martin-Brualla, R.; Goldman, D. B.; Nießner, M.; and Thies, J. 2022. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6290–6301.

Baruch, G.; Chen, Z.; Dehghan, A.; Dimry, T.; Feigin, Y.; Fu, P.; Gebauer, T.; Joffe, B.; Kurz, D.; Schwartz, A.; et al. 2021. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*.

Butler, D. J.; Wulff, J.; Stanley, G. B.; and Black, M. J. 2012. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, 611–625. Springer.

Cabon, Y.; Stoffl, L.; Antsfeld, L.; Csurka, G.; Chidlovskii, B.; Revaud, J.; and Leroy, V. 2025. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 1050–1060.

Cai, X.; Wang, Y.; Huang, Z.; Shao, Y.; and Li, D. 2024a. Voloc: Visual place recognition by querying compressed lidar map. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 10192–10199. IEEE.

Cai, X.; Wang, Y.; Luo, L.; Wang, M.; Li, D.; Xu, J.; Gu, W.; and Ai, R. 2024b. PRISM: PRogressive dependency maxImization for Scale-invariant image Matching. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 5250–5259.

Campos, C.; Elvira, R.; Rodríguez, J. J. G.; Montiel, J. M.; and Tardós, J. D. 2021. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6): 1874–1890.

Chen, X.; Chen, Y.; Xiu, Y.; Geiger, A.; and Chen, A. 2025. Easi3r: Estimating disentangled motion from dust3r without training. *arXiv preprint arXiv:2503.24391*.

Cheng, H. K.; and Schwing, A. G. 2022. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European conference on computer vision*, 640–658. Springer.

Dong, Q.; and Fu, Y. 2024. Memflow: Optical flow estimation and prediction with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19068–19078.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fei, X.; Zheng, W.; Duan, Y.; Zhan, W.; Tomizuka, M.; Keutzer, K.; and Lu, J. 2024. Driv3r: Learning dense 4d reconstruction for autonomous driving. *arXiv preprint arXiv:2412.06777*.

Feng, H.; Zhang, J.; Wang, Q.; Ye, Y.; Yu, P.; Black, M. J.; Darrell, T.; and Kanazawa, A. 2025. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. *arXiv preprint arXiv:2504.13152*.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11): 1231–1237.

Han, J.; An, H.; Jung, J.; Narihira, T.; Seo, J.; Fukuda, K.; Kim, C.; Hong, S.; Mitsufuji, Y.; and Kim, S. 2025. Dˆ 2USt3R: Enhancing 3D Reconstruction with 4D Pointmaps for Dynamic Scenes. *arXiv preprint arXiv:2504.06264*.

Jiang, Y.; Yu, C.; Xie, T.; Li, X.; Feng, Y.; Wang, H.; Li, M.; Lau, H.; Gao, F.; Yang, Y.; and Jiang, C. 2024. VR-GS: A Physical Dynamics-Aware Interactive Gaussian Splatting System in Virtual Reality. *arXiv preprint arXiv:2401.16663*.

Jiang, Z.; Zheng, C.; Laina, I.; Larlus, D.; and Vedaldi, A. 2025. Geo4d: Leveraging video generators for geometric 4d scene reconstruction. *arXiv preprint arXiv:2504.07961*.

Jin, L.; Tucker, R.; Li, Z.; Fouhey, D.; Snavely, N.; and Holynski, A. 2024. Stereo4d: Learning how things move in 3d from internet stereo videos. *arXiv preprint arXiv:2412.09621*.

Leroy, V.; Cabon, Y.; and Revaud, J. 2024. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, 71–91. Springer.

Li, Z.; Tucker, R.; Cole, F.; Wang, Q.; Jin, L.; Ye, V.; Kanazawa, A.; Holynski, A.; and Snavely, N. 2025. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 10486–10496.

Liu, Y.; Dong, S.; Wang, S.; Yin, Y.; Yang, Y.; Fan, Q.; and Chen, B. 2025. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 16651–16662.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2): 91–110.

Mehl, L.; Schmalfuss, J.; Jahedi, A.; Nalivayko, Y.; and Bruhn, A. 2023. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4981–4991.

Palazzolo, E.; Behley, J.; Lottes, P.; Giguere, P.; and Stachniss, C. 2019. ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7855–7862. IEEE.

Ranftl, R.; Bochkovskiy, A.; and Koltun, V. 2021. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12179–12188.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; and Fitzgibbon, A. 2013. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2930–2937.

Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; and Cremers, D. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 573–580. IEEE.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, 402–419. Springer.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.; and Agapito, L. 2025. 3d reconstruction with spatial memory. In *International Conference on 3D Vision 2025*.

Wang, J.; Chen, M.; Karaev, N.; Vedaldi, A.; Rupprecht, C.; and Novotny, D. 2025a. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 5294–5306.

Wang, J.; Karaev, N.; Rupprecht, C.; and Novotny, D. 2024a. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 21686–21697.

Wang, Q.; Zhang, Y.; Holynski, A.; Efros, A. A.; and Kanazawa, A. 2025b. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 10510–10522.

Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; and Revaud, J. 2024b. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20697–20709.

Wang, S.; Wang, Y.; Li, W.; Cai, X.; Wang, Y.; Chen, M.; Wang, K.; Su, Z.; Li, D.; and Fan, Z. 2025c. Aux-Think: Exploring Reasoning Strategies for Data-Efficient Vision-Language Navigation. *Advances in Neural Information Processing Systems*.

Wang, Y.; Zhou, J.; Zhu, H.; Chang, W.; Zhou, Y.; Li, Z.; Chen, J.; Pang, J.; Shen, C.; and He, T. 2025d. π^3: Scalable Permutation-Equivariant Visual Geometry Learning. arXiv:2507.13347.

Wu, C.-Y.; Li, Y.; Mangalam, K.; Fan, H.; Xiong, B.; Malik, J.; and Feichtenhofer, C. 2022. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 13587–13597.

Wu, Y.; Zheng, W.; Zhou, J.; and Lu, J. 2025. Point3R: Streaming 3D Reconstruction with Explicit Spatial Pointer Memory. *arXiv preprint arXiv:2507.02863*.

Yang, J.; Sax, A.; Liang, K. J.; Henaff, M.; Tang, H.; Cao, A.; Chai, J.; Meier, F.; and Feiszli, M. 2025. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 21924–21935.

Yu, C.; Liu, Z.; Liu, X.-J.; Xie, F.; Yang, Y.; Wei, Q.; and Fei, Q. 2018. DS-SLAM: A semantic visual SLAM towards dynamic environments. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 1168–1174. IEEE.

Zhang, J.; Herrmann, C.; Hur, J.; Jampani, V.; Darrell, T.; Cole, F.; Sun, D.; and Yang, M.-H. 2024. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*.

Zhang, L.; and Agrawala, M. 2025. Packing input frame context in next-frame prediction models for video generation. *arXiv preprint arXiv:2504.12626*.

Zhang, S.; Ge, Y.; Tian, J.; Xu, G.; Chen, H.; Lv, C.; and Shen, C. 2025. POMATO: Marrying Pointmap Matching with Temporal Motion for Dynamic 3D Reconstruction. *arXiv preprint arXiv:2504.05692*.

Zhang, Z.; Cole, F.; Li, Z.; Rubinstein, M.; Snavely, N.; and Freeman, W. T. 2022. Structure and motion from casual videos. In *European Conference on Computer Vision*, 20–37. Springer.

Zheng, Y.; Harley, A. W.; Shen, B.; Wetzstein, G.; and Guibas, L. J. 2023. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19855–19865.

Zhuo, D.; Zheng, W.; Guo, J.; Wu, Y.; Zhou, J.; and Lu, J. 2025. Streaming 4D Visual Geometry Transformer. *arXiv preprint arXiv:2507.11539*.