SM4RT: LEARNING STRUCTURED MOTION GEOMETRY FOR 4D RECONSTRUCTION

Anonymous authors
Paper under double-blind review



Figure 1: **Structured motion geometry. SM4RT** represents scene dynamics with *Structure-of-Motion*: a set of motion bases to describe how each component evolves over time and time-shared point-to-base assignments to bind source points to motion bases. They produce dense rigid transform maps that recover world-coordinate tracks and uncover the structured motion geometry underlying the dynamic 4D scenes.

ABSTRACT

Geometry Foundation Models (GFMs) have substantially advanced monocular 3D reconstruction, yet extending this capability to 4D dynamic understanding remains a fundamental challenge. Most existing motion perception methods (e.g., sparse tracking, dense point-wise flow) treat motion as independent point-wise displacements, ignoring the structured nature of physical motion. However, real-world objects usually obey rigid-body kinematics, and points thus usually move collectively, not in isolation. Motion itself possesses geometric structure: physical objects undergo a set of rigid-body transformations governed by $\mathrm{SE}(3)$, rather than unstructured point-wise displacements. Building on this insight, we propose **SM4RT**, a **S**tructured **M**otion **4**D **R**econstruction **T**ransformer for end-to-end 3D reconstruction and structured motion perception. SM4RT introduces *Structure-of-Motion* (SoM) to represent scene dynamics, where scene motion is decomposed into a compact set of motion bases, each represented as a temporal sequence of 6D twists in $\mathfrak{se}(3)$. Dense scene motion is then recovered by sparse, time-shared per-pixel assignment weights over these bases, ensuring points on the same object share a common rigid-body motion trajectory. SM4RT introduces a parallel motion geometry encoder and decoder that jointly infer 3D geometry, world-coordinate motion, and scene kinematic structure in a single forward pass from monocular RGB video. SM4RT achieves strong motion reconstruction performance while preserving the geometric structure of scene motion.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Recent advances in Geometry Foundation Models (GFMs) (Wang et al., 2025b; Zhuo et al., 2025; Wu et al., 2025a; Wang et al., 2025d; Wu et al., 2026) have demonstrated remarkable capability in reconstructing 3D scene geometry from monocular video, facilitating applications from embodied intelligence (Wu et al., 2025b) to autonomous driving (Zuo et al., 2026a;b). Beyond static geometry, 4D reconstruction aims to recover scene structure and motion jointly (Zhang et al., 2026).

Most existing methods primarily estimate where points move over time without explicitly modeling the shared motion structure that explains how they move together. They usually frame motion perception as point-wise trajectory estimation: either sparse point tracking (Xiao et al., 2025; Wang et al., 2025a) or dense point-wise flow estimation (Karhade et al., 2026; Ngo et al., 2025a). Both treat motion as a collection of independent point-wise displacements, overlooking the relational dependencies among points. While empirically effective for localization tasks, this formulation fails to capture the *structured nature of physical motion*: accurate point-wise displacement does not directly imply meaningful dynamic understanding. *We reflect on how humans perceive motion.* Humans perceive motion via sparse kinematic cues (Johansson, 1973): not at the pixel or point level, but at the level of objects and parts, enabling rapid and parsimonious motion understanding. 3D points that machines track are not atomic entities but samples drawn from continuous, physically grounded objects (Liu et al., 2019). Real-world objects usually obey rigid-body kinematics: points move *collectively*, not in isolation. Consider pure rotation: under a displacement view, rotating points appear as disorganized coordinate shifts, yet they share a single common motion governed by $\mathrm{SE}(3)$. This motivates a shift in perspective: from *geometry **with** motion* to the *geometry **of** motion*, where motion itself possesses geometric structure (Wu & Carricato, 2020) encoded in the symmetry groups of physical transformations, and any motion representation must respect these group-theoretic priors.

We propose **SM4RT**, a **S**tructured **M**otion **4**D **R**econstruction **T**ransformer that jointly infers 3D scene geometry, world-coordinate dense motion, and scene motion structure from monocular RGB video in a single forward pass, as shown in Figure 1. Rather than predicting independent point-wise displacements, SM4RT introduces *Structure-of-Motion* (SoM): decomposing scene motion into a compact set of N motion bases, each represented as a temporal sequence of 6D twists in $\mathfrak{se}(3)$. Dense motion is recovered by combining these bases through sparse, time-shared per-pixel assignment weights, ensuring consistent motion throughout the scene. Unlike classical SfM, which recovers structure from motion, SoM models the structure of motion itself: assignments determine which points move together, while motion bases describe how the corresponding latent objects or parts evolve over time. Architecturally, SM4RT introduces a parallel motion geometry encoder that distills dynamic cues from intermediate geometry features and a motion geometry decoder that jointly predicts the motion bases and per-pixel assignment map. Entropy regularization and a pseudo-background constraint encourage compact, unambiguous decompositions. Shared kinematic parameters bind points into coherent entities, promoting motion consistency and preserving object structure over time. Beyond point tracking, the twist representation supports structured motion interpolation and future state prediction grounded in rigid-body kinematics. SM4RT achieves state-of-the-art performance on existing motion perception benchmarks, demonstrating that structured motion modeling and accurate point-wise tracking are mutually reinforcing.

2 RELATED WORK

Geometry Foundation Models. Building on large-scale vision transformers (Dosovitskiy et al., 2021; Oquab et al., 2024), geometry foundation models learn geometric priors for feed-forward 3D reconstruction without per-scene optimization. Multi-view models recover camera parameters and dense scene geometry (Wang et al., 2025b; 2026; Lin et al., 2026), while recurrent and streaming models extend reconstruction to sequential inputs (Wang et al., 2025d; Zhuo et al., 2025; Wu et al., 2025a). Subsequent work also explores metric reconstruction (Keetha et al., 2026), implicit representations for mesh reconstruction (Wu et al., 2026), and instance-aware semantic reconstruction (Li et al., 2026). SM4RT extends this foundation from static geometry to structured 4D motion.

Motion Perception. Motion perception methods estimate point-wise trajectories through 2D tracking (Doersch et al., 2023; Karaev et al., 2025; Lai et al., 2026; Karaev et al., 2024), sparse 3D tracking with geometric priors (Wang et al., 2025a; Xiao et al., 2025), or dense 3D motion recon-

struction from monocular video (Ngo et al., 2025b;a; Feng et al., 2025; Liu et al., 2025; Karhade et al., 2026; Sucar et al., 2026; Luo et al., 2026). D4RT (Zhang et al., 2026) further enables efficient, queryable 4D reconstruction with a feed-forward transformer and lightweight decoder. However, they produce independent point-wise motion estimates without any notion of object-level structure, making them fundamentally limited for structured dynamic scene understanding. Rather than focusing only on efficient point-time querying, SM4RT models motion with rigid kinematic entities and exposes object/part-level motion structure.

Motion Decomposition. Scene motion can be decomposed into a compact set of rigid components (Teed & Deng, 2021; Wang et al., 2025c; Zhou & Lee, 2025), echoing classical multi-body motion segmentation (Vidal, 2011). RAFT-3D (Teed & Deng, 2021) predicts dense $\mathrm{SE}(3)$ fields from RGBD image pairs, while Motion4D (Zhou & Lee, 2025) alternates motion and semantic optimization using SAM2 masks. Shape of Motion (Wang et al., 2025c) is closely related to our representation: it exploits the low-dimensional structure of dynamic scenes through compact $\mathrm{SE}(3)$ motion bases and soft point-to-basis decomposition. However, it recovers this structure through per-scene optimization with external priors such as monocular depth and long-range 2D tracks. SM4RT learns Structure-of-Motion within a geometry foundation model, directly predicting time-shared assignments and temporal twist-sequence motion bases from monocular RGB video in one forward pass, without auxiliary inputs or test-time optimization.

3 Proposed Approach

3.1 Geometry with Motion to Geometry of Motion

Geometry Foundation Models. Given a monocular RGB video $\boldsymbol{I}$ of S frames, a GFM $\mathcal{F}_{\text{geo}}$ infers per-frame camera intrinsics $\mathbf{K}$, extrinsics $[\mathbf{R} \mid \mathbf{t}]$, and depth maps $\boldsymbol{D}$ in a single forward pass:

$$(\boldsymbol{D},\, \mathbf{K},\, [\mathbf{R} \mid \mathbf{t}]) = \mathcal{F}_{\text{geo}}(\boldsymbol{I}). \tag{1}$$

These outputs unproject to a world-coordinate point cloud $\boldsymbol{P}$ for each frame, constituting a complete static 3D representation of the scene.

Existing Motion Perception. Existing methods (Karhade et al., 2026; Sucar et al., 2026; Luo et al., 2026) usually formulate motion as point-wise trajectory estimation. Let $t = 0$ denote the source frame and u denote a source-frame position. Dense methods predict a displacement field $\boldsymbol{\Delta}$ over all target frames:

$$\hat{\boldsymbol{p}}_{u,t} = \boldsymbol{p}_{u,0} + \boldsymbol{\Delta}_{u,t}, \quad t = 1, \ldots, S. \tag{2}$$

Each pixel u is predicted independently with no self-constrained structural constraint coupling it to its neighbors. In the sparse case, a subset of query points $\{\boldsymbol{q}_k\}_{k=1}^{K}$ is tracked individually as $\hat{\boldsymbol{p}}_{k,t} = f_\theta(\boldsymbol{I}, \boldsymbol{q}_k)$, again not exploiting inter-point relations.

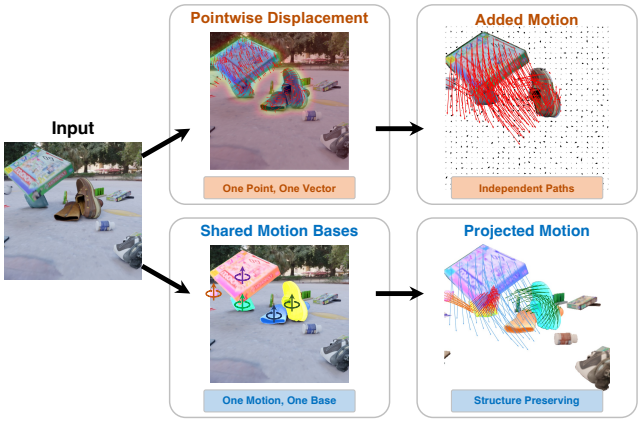


Figure 2: **Comparison of motion representations.** The proposed representation preserves the structure of object motion, which provides a more structured description than standard per-pixel displacement and encouraging intra-object homogeneity.

The Structural Gap. This independent displacement formulation places no constraint between pixels of the same physical object. Yet for any rigid body undergoing transformation $\mathbf{T} \in \mathrm{SE}(3)$, all points on the object satisfy the coupling constraint:

$$\boldsymbol{p}_{u,t} = \mathbf{T}_t\, \boldsymbol{p}_{u,0}, \quad \forall u \in \text{object}. \tag{3}$$

A scene with N rigid objects thus has only $6N$ motion degrees of freedom per target frame, far fewer than the $3HW$ degrees assumed by point-wise displacement. Ignoring this structure forces the model to independently fit HW unconstrained displacements, which is a severely over-parameterized and physically inconsistent formulation.

Our Paradigm. As shown in Figure 2, we model scene motion as a sparse mixture of shared motion bases. Time-shared point-to-base assignments determine *which points move together*, while temporal motion bases describe *how each group moves over time*.

3.2 Structured Motion Geometry

We define the geometry of motion as the structured organization of scene dynamics induced by shared physical transformations. SoM captures this structure through time-shared point-to-base assignments and temporal twist-sequence motion bases.

Twist Representation. A twist $\boldsymbol{\xi} = [\boldsymbol{\rho}, \boldsymbol{\phi}] \in \mathbb{R}^6$ encodes translational and rotational components $\boldsymbol{\rho}, \boldsymbol{\phi} \in \mathbb{R}^3$. The hat operator maps it to $\hat{\boldsymbol{\xi}} \in \mathfrak{se}(3)$, whose exponential gives a rigid transformation in $\mathrm{SE}(3)$ (Murray et al., 1994). The matrix definition is provided in Section A.1.

Structure-of-Motion via Motion Bases. For a video $\boldsymbol{I}$, we predict N temporal motion bases $\{\boldsymbol{\xi}_{i,t}\}$ and a time-shared assignment map $\boldsymbol{W} = \{w_{u,i}\}$, where $w_{u,i} \geq 0$ and $\sum_i w_{u,i} = 1$. Each basis is a sequence of 6D twists describing a latent component's motion, and each source point uses the same assignment weights across target frames. We combine the bases in twist space and apply the resulting rigid transform to the source point:

$$\boldsymbol{\Xi}_{u,t} = \sum_{i=1}^{N} w_{u,i}\, \boldsymbol{\xi}_{i,t}, \mathbf{T}_{u,t} \quad = \mathrm{Exp}(\hat{\boldsymbol{\Xi}}_{u,t}), \qquad \boldsymbol{p}_{u,t} = \mathbf{T}_{u,t}\boldsymbol{p}_{u,0}. \tag{4}$$

Homogeneous coordinates are omitted for clarity. Points with identical assignments share the same temporal rigid transformation, coupling their motion through the shared bases. Blending local rigid motions also allows the representation to model articulated and deformable objects. Instead of predicting $3SHW$ point-wise displacement values, SoM predicts $NHW + 6SN$ assignment and motion-base parameters, with assignments shared across time. The architecture below learns both components jointly from RGB video.

Emerging Capability of Structured Motion. We interpolate or extrapolate temporal motion bases in twist space while keeping point-to-base assignments fixed. As shown in Figure 3, shared bases produce more coherent intermediate and future states than point-wise trajectory interpolation or extrapolation (Liu et al., 2025). The two formulations are equivalent for pure translation, but differ when rotation is involved. This parameter-free procedure does not explicitly model collisions, rebounds, or contact dynamics. A detailed explanation is provided in Section A.2.

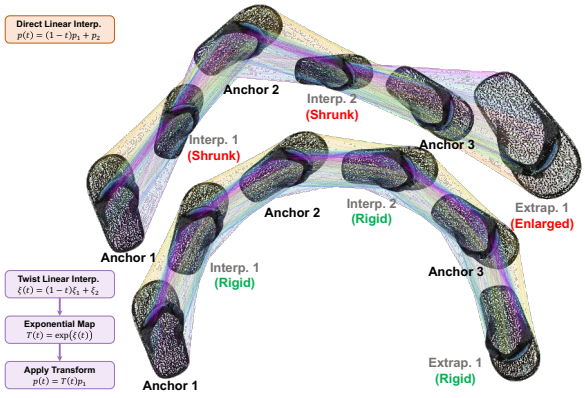


Figure 3: **Motion interpolation and extrapolation.** Track-space operations (top) deform object geometry, while motion-base operations (bottom) preserve rigid structure.

3.3 SM4RT: Structured Motion 4D Reconstruction Transformer

SM4RT processes an input video $\boldsymbol{I}$ through two parallel branches: a *scene geometry* branch that recovers 3D structure and camera parameters, and a *motion geometry* branch that infers the motion bases and per-pixel assignments jointly. The two branches share a common geometry-aware backbone and interact through cross-attention, allowing motion reasoning to be grounded in 3D scene structure. The overall architecture is illustrated in Figure 4.

Scene Geometry Branch. We adopt the pretrained DepthAnythingV3 (Lin et al., 2026) geometry backbone and its camera and depth heads to extract geometry tokens $\boldsymbol{G}$ and predict per-frame camera parameters and depth maps $\boldsymbol{D}$. Unprojecting the source-frame depth yields source points $\boldsymbol{P}^0$. World coordinates are defined in the source camera coordinate system. Encoder and unprojection details are provided in Section A.3.

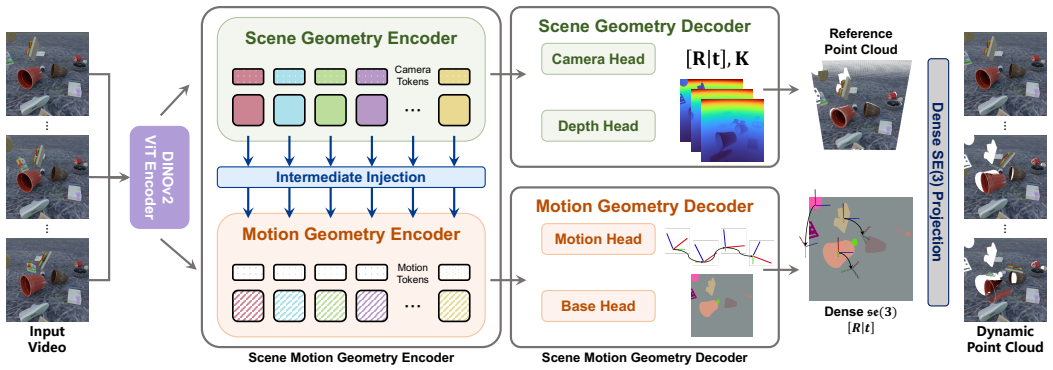


Figure 4: **Overview of our SM4RT.** Given a monocular input video, SM4RT extracts latent geometry tokens with a DINOv2 backbone, then decouples them into scene geometry and motion geometry tokens. The scene decoder predicts camera parameters and depth, while the motion decoder estimates sparse $\mathfrak{se}(3)$ transforms and motion base weights for dense SE(3) projection.

Motion Geometry Encoder. While the geometry-aware backbone encodes rich scene structure, it lacks motion-specific inter-frame cues. We therefore introduce a parallel motion geometry encoder that uses four intermediate geometry token layers $\{\boldsymbol{G}'_i \mid i = 0, 1, 2, 3\}$ as auxiliary references. N motion tokens $\boldsymbol{m}$, initialized with per-frame time embeddings, first attend to $\boldsymbol{G}_0$ via cross-attention to obtain $\boldsymbol{m}'$, and then concatenated as $[\boldsymbol{m}', \boldsymbol{G}'_0]$ and fed through 4 layers of frame motion attention with positional embeddings. After each layer, the corresponding intermediate geometry token is injected via a projection layer as a structural reference. This design allows motion tokens to aggregate dynamic cues from the geometry stream while capturing long-range temporal dependencies.

Motion Geometry Decoder. The *base head* applies a Dense Prediction Transformer (DPT) (Ranftl et al., 2021) to produce per-pixel base assignment weights $\boldsymbol{W} \in \mathbb{R}^{H \times W \times N}$. SparseMax$(\cdot)$ (Martins & Astudillo, 2016) is applied along N to induce sparsity and promote hard assignments. The *motion head* applies an MLP to decode motion tokens $\boldsymbol{m}$ into temporal twist sequences $\{\boldsymbol{\xi}_{i,t}\}$. Together with source points $\boldsymbol{P}^0$, these outputs recover dense tracks through Equation (4) in a single forward pass without test-time optimization.

Post Processing (Optional). *SM4RT (Adaptive)* refines the assignment map by clustering and averaging assignments within each region before recomputing motion with Equation (4). Results using this optional refinement are labeled *Adaptive*. Details are provided in Section A.4.

3.4 Training

The total training objective combines geometry supervision, motion point supervision, and motion base regularization:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{geo} + \lambda_2 \mathcal{L}_{point} + \lambda_3 \mathcal{L}_{base}, \tag{5}$$

where λ are balancing weights.

Geometry Supervision. We supervise depth and camera parameters with normalized depth and camera ℓ_1 losses, and regularize scene scale to constrain scale ambiguity. These terms form $\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{cam}} + \mathcal{L}_{\text{scale}}$. Detailed geometry loss definitions are provided in Section A.5.

Motion Point Supervision. Datasets with dense annotations allow foreground and background pixels to be decoupled via mask to avoid domination by static points, whereas sparse tracking datasets lack exhaustive per-pixel labels and do not require such decoupling. We therefore apply different supervision strategies for $\mathcal{L}_{point}$ depending on data type:

$$\begin{aligned} \mathcal{L}_{dense} &= \mathcal{L}_{fg} + \alpha \cdot \mathcal{L}_{bg} = \frac{1}{|\mathcal{M}|} \sum_{u \in \mathcal{M}} \ell(\mathbf{p}_u, \mathbf{p}'_u) + \frac{\alpha}{|\neg\mathcal{M}|} \sum_{u \in \neg\mathcal{M}} \ell(\mathbf{p}_u, \mathbf{0}), \\ \mathcal{L}_{sparse} &= \sum_{u \in \mathcal{M}} \ell(\mathbf{p}_u, \mathbf{p}'_u), \end{aligned} \tag{6}$$

where $\mathcal{M} \in \{0, 1\}$ is the selected mask, $\mathbf{p}_u$ and $\mathbf{p}'_u$ are the deviatoric positions from source to target of prediction and ground-truth at pixel u, and α downweights the background.

Table 1: **Performance on video depth estimation.** We follow the protocols of Zhang et al. (2025), reporting AbsRel (↓) and $\delta_{<1.25}$(↑).

	Bonn		Sintel		KITTI		7Scenes		ScanNetV2	
Method	AbsRel ↓	$\delta_{<1.25}$ ↑	AbsRel ↓	$\delta_{<1.25}$ ↑	AbsRel ↓	$\delta_{<1.25}$ ↑	AbsRel ↓	$\delta_{<1.25}$ ↑	AbsRel ↓	$\delta_{<1.25}$ ↑
MonST3R (Zhang et al., 2025)	0.072	95.48	0.309	54.23	0.095	91.59	0.132	84.57	0.092	92.25
MapAnythingV1 (Keetha et al., 2026)	0.068	96.80	0.275	59.58	0.066	95.81	0.076	92.82	0.039	98.34
VGGT (Wang et al., 2025b)	0.051	97.18	0.306	62.68	**0.051**	96.88	0.073	91.79	**0.023**	98.84
DepthAnythingV3 (Lin et al., 2026)	**0.046**	97.38	0.186	70.66	0.052	**97.35**	**0.066**	92.74	0.032	98.04
SM4RT	0.054	**97.49**	**0.169**	**75.51**	0.056	97.20	0.067	**92.91**	0.029	**98.86**

Table 2: **Performance on 3D reconstruction.** We adopt DA3Bench (Lin et al., 2026), reporting performance on Pose Estimation (Pose) and Unposed Reconstruction (Rec.). For Pose, we report Auc3 / Auc30 (↑). For Rec., we report CD (↓) for DTU, and F1 Score (↑) for the rest.

	HiRoom		ETH3D		DTU		7Scenes		ScanNet++	
Model	Pose	Rec.	Pose	Rec.	Pose	Rec.	Pose	Rec.	Pose	Rec.
MapAnythingV1 (Keetha et al., 2026)	22.13 / 83.51	27.48	21.96 / 79.36	44.89	5.56 / 77.89	7.469	18.36 / 81.24	39.46	27.75 / 87.59	34.62
Pi3 (Wang et al., 2026)	66.88 / 94.78	67.68	35.30 / 87.30	71.61	62.54 / 94.82	3.388	26.34 / 86.18	43.43	50.36 / 92.42	58.54
VGGT (Wang et al., 2025b)	49.03 / 87.96	57.20	26.33 / 80.80	55.95	79.19 / 85.27	2.030	23.63 / 84.91	47.14	59.48 / 94.43	66.98
DepthAnythingV3 (Lin et al., 2026)	79.39 / 95.26	86.08	**47.84** / 90.81	**78.69**	93.25 / 99.30	1.819	27.46 / 86.68	49.85	**83.90 / 98.17**	78.30
Any4D (Karhade et al., 2026)	11.22 / 78.87	15.80	4.63 / 61.46	28.19	2.15 / 74.04	6.206	11.16 / 78.32	28.38	6.43 / 74.62	24.35
VDPM (Sucar et al., 2026)	28.15 / 85.45	41.12	11.41 / 73.99	55.50	42.68 / 92.66	2.940	20.98 / 84.88	53.81	27.22 / 87.70	56.41
4RC (Luo et al., 2026)	85.53 / 98.13	86.53	37.79 / 89.97	71.97	92.07 / 99.20	1.380	26.45 / 86.66	54.14	78.96 / 97.58	73.90
SM4RT	**86.16 / 98.36**	**89.89**	43.88 / **91.56**	74.86	**93.30 / 99.32**	**1.153**	**27.91 / 86.76**	**54.31**	81.45 / 97.88	**78.50**

Motion Base Supervision. Multiple auxiliary losses regularize the base assignment map $\boldsymbol{W}$ to encourage physically meaningful motion assignment.

Entropy regularization promotes sparse and unambiguous assignments by penalizing diffuse distributions over bases:

$$\mathcal{L}_{entropy} = -\sum \boldsymbol{W} \odot \log(\boldsymbol{W}). \tag{7}$$

Pseudo-background regularization addresses the tendency of the static background base to bleed into dynamic regions. Since the background occupies the largest spatial area in most scenes, we identify the pseudo-background (PB) base as the mode of the dominant assignment indices, and penalize its assignment in non-background regions:

$$\mathcal{L}_{pseudo} = \left\| \mathbf{1}_{(\arg\max_N \boldsymbol{W}) \neq n_{pseudo}} \odot (\boldsymbol{W}_{n_{pseudo}}) \right\|_2^2, \qquad n_{pseudo} = \text{mode}(\arg\max_N \boldsymbol{W}). \tag{8}$$

Twist singularity regularization. A dense SE(3) motion field can collapse into translation-only motion, weakening the uniform-motion property. To tackle this, we introduce a translation supervision parallel to point supervision:

$$\mathcal{L}_{singular} = \frac{1}{|\mathcal{M}|} \sum_{u \in \mathcal{M}} \ell(\mathbf{t}_u, \mathbf{t}'_u), \tag{9}$$

where $\mathbf{t}_u$ and $\mathbf{t}'_u$ are the translation components of the predicted and ground-truth dense SE(3) transforms at pixel u. This term is used for samples with dense SE(3) annotations.

4 Experiments

4.1 Experimental Setup

Implementation. We initialize SM4RT from Depth Anything 3 (Lin et al., 2026), including its DINOv2 backbone with depth and camera heads. We use $N = 10$ motion tokens and input clips of 12–24 frames. The model contains 1.38 billion parameters, including 0.17 billion in motion-specific modules. More details are provided in Section B.1.

Training. We first pretrain on Kubric (Movi-F) (Greff et al., 2022) to learn base assignments, then train jointly on Kubric, Stereo4D (Jin et al., 2025), PointOdyssey (Zheng et al., 2023), and DynamicReplica (Karaev et al., 2023). Finally, we fine-tune depth and camera prediction on static and dynamic scenes, adding ARKitScenes (Baruch et al., 2021), HyperSIM (Roberts et al., 2021), ScanNet++ (Yeshwanth et al., 2023), VKitti2 (Cabon et al., 2020), and Waymo (Sun et al., 2020) for geometry supervision. Data preparation and the full training schedule are given in Section B.2.

Table 3: **Performance on world-coordinate tracking.** We report Average Percent of Points within Delta (APD) and End-Point Error (EPE), where higher APD and lower EPE indicate better performance. Numbers are presented in fractions of All / Dynamic. Predictions are aligned to the ground truth using Global Alignment over 64 frames.

Task	World-Coordinate Tracking							
Dataset	ADT		DS		PO		PStudio	
Model	APD (↑)	EPE (↓)	APD (↑)	EPE (↓)	APD (↑)	EPE (↓)	APD (↑)	EPE (↓)
MonST3R (Zhang et al., 2025)	74.35 / 67.92	0.2721 / 0.1578	58.06 / 51.86	0.4387 / 0.5313	33.47 / 39.36	0.9021 / 0.6452	51.32 / 51.32	0.4568 / 0.4568
SpatialTracker (Xiao et al., 2024)	45.65 / 67.65	0.8530 / 0.1628	54.85 / 58.65	0.9274 / 1.0828	38.54 / 51.20	0.7499 / 0.4695	62.59 / 62.59	0.3094 / 0.3094
St4RTrack (Feng et al., 2025)	76.00 / 75.34	0.2680 / 0.1212	73.74 / 68.13	0.2682 / 0.2961	67.94 / 68.71	0.3140 / 0.2970	69.67 / 69.67	0.2637 / 0.2637
TraceAnything (Liu et al., 2025)	76.08 / 73.05	0.2469 / 0.1268	60.66 / 61.23	0.5756 / 0.4422	39.84 / 47.39	1.0595 / 0.7224	71.32 / 71.32	0.2727 / 0.2727
Any4D (Karhade et al., 2026)	58.65 / 70.66	0.4048 / 0.1313	79.08 / 68.81	0.2203 / 0.3112	66.20 / 66.55	0.3741 / 0.3991	64.44 / 64.44	0.2977 / 0.2977
VDPM (Sucar et al., 2026)	85.96 / 72.10	0.1666 / 0.1359	74.27 / **77.66**	0.2548 / **0.2123**	81.28 / **81.85**	**0.1899** / **0.1749**	78.73 / 78.73	0.1812 / 0.1812
4RC (Luo et al., 2026)	83.22 / 72.27	0.1905 / 0.1422	82.17 / 73.92	0.1946 / 0.2747	79.35 / 75.94	0.2762 / 0.3517	70.91 / 70.91	0.2443 / 0.2433
D4RT (Zhang et al., 2026)	72.20 / 73.25	0.2758 / 0.3199	72.48 / 74.88	0.2959 / 0.2755	67.99 / 74.25	0.3178 / 0.2593	79.60 / 79.60	**0.1753** / **0.1753**
SM4RT	**88.56** / **80.68**	**0.1339** / 0.0935	**82.61** / 71.62	**0.1918** / 0.2912	**81.62** / 73.68	0.2578 / 0.3808	**79.98** / **79.98**	0.1754 / 0.1754
SM4RT (Adaptive)	88.32 / 78.99	0.1363 / 0.0995	81.95 / 69.79	0.1970 / 0.3051	80.77 / 71.76	0.2617 / 0.3897	78.16 / 78.16	0.1883 / 0.1883

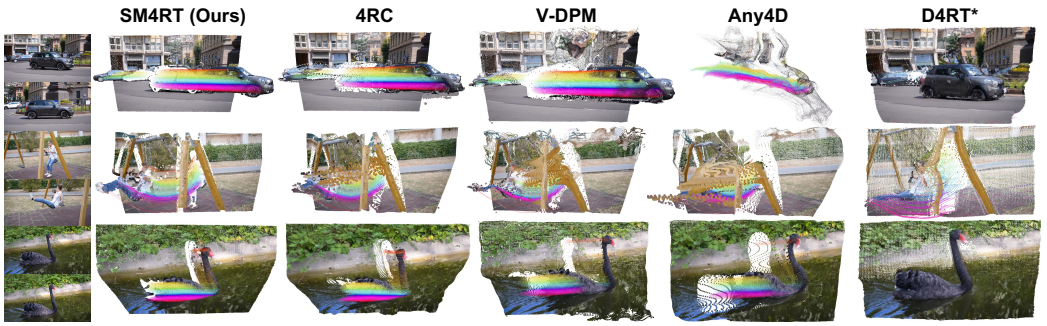


Figure 5: **Video track visualization.** We draw foreground tracklines and terminal positions of initial points. We use the reproduced version of D4RT (Li & RHOS lab, 2026) and sample input queries at a quarter resolution, while the other methods take dense input at resolution 294×518.

4.2 3D Geometry Reconstruction

Video Depth Estimation. Table 1 reports the video depth estimation performance following the protocol of MonST3R (Zhang et al., 2025). Our method, SM4RT, achieves competitive results across all benchmarks, reaching optimal or suboptimal in most datasets. These results demonstrate that SM4RT maintains strong depth estimation accuracy across diverse indoor and outdoor scenes.

3D Reconstruction. Table 2 presents the 3D reconstruction results on DA3Bench (Lin et al., 2026) to evaluate both the pose estimation (Auc3 / Auc30) and unposed reconstruction quality. SM4RT achieves the best overall performance across all five benchmarks, and consistently improves reconstruction quality across all datasets while maintaining comparable pose estimation performance, validating the effectiveness of our method for joint depth and pose estimation in 3D reconstruction pipelines.

Table 4: **TapVid3D benchmark results on the minval split.** We report 3D Average Jaccard (AJ) and Average Percent of Points within Delta (APD) to evaluate tracking performance. * indicates that the results are from Liu et al. (2025).

Dataset	ADT		DriveTrack		PStudio	
Model	AJ	APD	AJ	APD	AJ	APD
SpatialTracker (Xiao et al., 2024)	9.2	15.1	5.8	10.2	9.8	17.7
DELTAv1 (Ngo et al., 2025b)	20.3	20.4	18.0	25.2	**18.5**	26.1
DELTAv2 (Ngo et al., 2025a)	20.5	20.9	19.1	26.2	17.9	26.2
SpatialTrackerV2 (Xiao et al., 2025)	18.1	28.2	14.2	20.9	16.9	**26.3**
St4RTrack (Feng et al., 2025)*	13.4	15.2	7.4	8.5	6.9	7.2
TraceAnything (Liu et al., 2025)*	15.6	20.5	9.6	15.5	10.8	16.3
D4RT (Zhang et al., 2026)	24.3	31.5	14.7	19.3	17.5	23.5
V-DPM (Sucar et al., 2026)	28.3	35.3	23.1	28.6	17.6	25.5
4RC (Luo et al., 2026)	30.2	38.5	23.1	28.3	16.2	22.8
SM4RT	**32.5**	**40.6**	**24.3**	**30.1**	16.7	23.6

4.3 3D Tracking

Evaluations. We evaluate SM4RT under two standard 3D tracking settings. For TapVid3D (Koppula et al., 2024), we follow the official minval split, downsampling to 48 frames, and report 3D Average Jaccard (AJ) and Average Percent of Points within Delta (APD) to measure tracking performance. As shown in Table 4, SM4RT achieves the best results on ADT and DriveTrack across both metrics, outperforming prior methods such as V-DPM and 4RC. On PStudio, SM4RT remains competitive, although the best performance is achieved by other baselines.

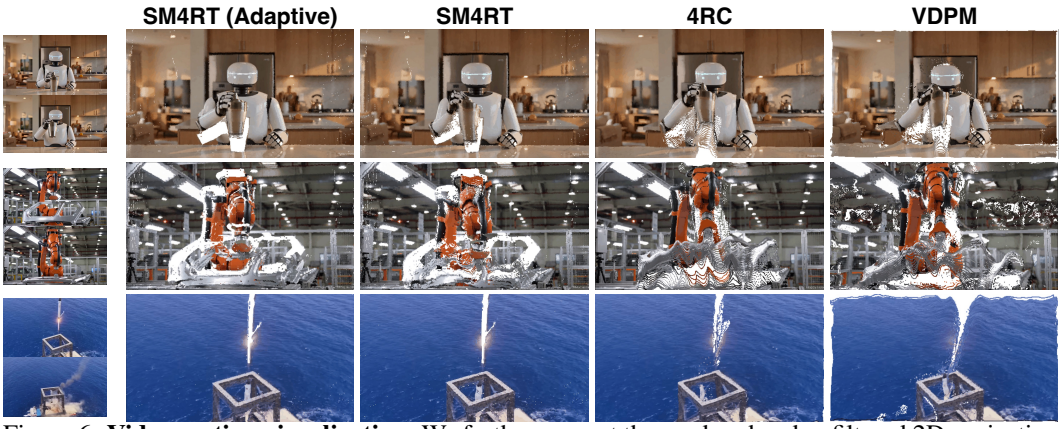


Figure 6: **Video motion visualization.** We further present the rendered and unfiltered 2D projection of source-frame points in the target frame, with the reference source (top) and target (bottom) frames shown on the left. *SM4RT* indicates that the motion is directly obtained from the output, while *SM4RT (Adaptive)* automatically groups similar regions to obtain more structured motion.

Table 5: **Entity deformation and dense tracking on Kubric-Rigid (24 Frames).**

Type	Model	Dynamic		Initial	Tracking	
		VM^{fg} (↓)	VM^{bg} (↓)	VM^{fg} (↓)	AJ (↑)	APD (↑)
Camera	DELTAv1 (Ngo et al., 2025b)	0.7137 / 0.6795	0.3223	0.4686	46.18	55.27
	DELTAv2 (Ngo et al., 2025a)	0.7503 / 0.7052	0.3405	0.4627	46.08	55.17
World	SpatialTrackerV2 (Xiao et al., 2025)	0.3291 / 0.2821	N/A	0.7437	N/A	N/A
	St4RTrack (Feng et al., 2025)	1.1597 / 0.9441	0.6971	1.0008	23.91	30.52
	TraceAnything (Liu et al., 2025)	0.1887 / 0.1761	0.1210	1.0081	76.33	85.61
	Any4D (Karhade et al., 2026)	0.5053 / 0.4056	0.1385	0.6530	68.49	79.78
	VDPM (Sucar et al., 2026)	0.1952 / 0.1710	0.1811	**0.2322**	59.26	70.93
	4RC (Luo et al., 2026)	0.2930 / 0.2503	0.2383	0.2520	82.25	89.62
World	SM4RT	0.1905 / 0.1876	0.0122	0.2452	**84.37**	**90.62**
	SM4RT (Adaptive)	**0.0254 / 0.0191**	**0.0119**	0.2499	82.54	89.56

We further evaluate world-coordinate tracking following WorldTrack (Feng et al., 2025). In this protocol, predicted 3D point positions are aligned to the ground truth using global alignment over 64 frames. Since D4RT (Zhang et al., 2026) is not open-source, we adapt OpenD4RT (Li & RHOS lab, 2026) to report its results under the same setting. As shown in Table 3, SM4RT achieves the best performance on ADT and strong results on DS and PO, showing accurate and stable 3D motion.

Visualizations. Figure 5 shows visual comparisons of video tracklines on foreground objects. Compared with 4RC (Luo et al., 2026), V-DPM (Sucar et al., 2026), Any4D (Karhade et al., 2026), and D4RT (Zhang et al., 2026; Li & RHOS lab, 2026), SM4RT produces more coherent tracklines that better adhere to object structure. While other methods suffer from scattered points or structural collapse, our method preserves moving-object geometry.

In Figure 6, instead of showing indirect tracklines, we render the unfiltered 2D projection of source-frame points in the target frame, with the reference source and target frames shown on the left. SM4RT visualizes the motion directly predicted by the model, while SM4RT (Adaptive) applies adaptive grouped averaging according to the motion base assignment. Compared with 4RC (Luo et al., 2026) and V-DPM (Sucar et al., 2026), SM4RT produces more coherent projected motion with fewer scattered points and better preservation of object structure, especially near motion boundaries.

4.4 Experimental Analysis

Rigid Entity Deformation Analysis. We introduce Kubric-Rigid-Eval to assess rigid-object structure preservation beyond point-wise tracking accuracy. We use the average von Mises deformation score (VM) to measure non-rigid distortion from affine alignment between source and target point clouds. As shown in Table 5, raw SM4RT achieves the best tracking accuracy and low background deformation. Adaptive grouping further reduces dynamic foreground VM from 0.1905 to 0.0254, with a small APD drop from 90.62 to 89.56. This verifies the effectiveness of learning the motion structure. More details are provided in Section C.

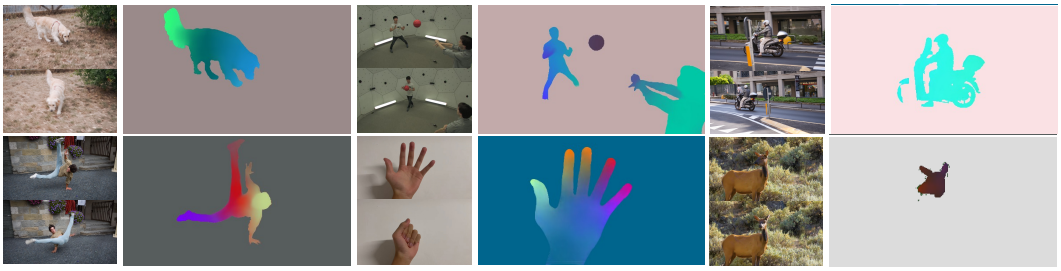


Figure 7: **Visualization of motion base assignment.** We present the start and end frame of input video, and the corresponding PCA visualization of motion base assignment shared across the video.

Table 6: **Backbone and injection ablation.** We compare VGGT and DA3 backbones and feature injection strategies using tracking APD.

Backbone	Injection Strategy	ADT	DS
VGGT	Linear	80.79 / 77.73	72.48 / 71.39
DA3	Linear	**88.56 / 80.68**	**82.61 / 71.62**
DA3	MLP	87.65 / 78.28	81.05 / 69.26
DA3	Replace	85.00 / 78.51	80.58 / 69.14

Table 7: **Fine-tuning data ablation.** We compare rigid, non-rigid, and mixed data using tracking APD and deformation VM.

Setting	ADT	DS	Kubric (VM)
Rigid	87.76 / 79.74	81.39 / 69.46	**0.1847 / 0.1203**
Non-Rigid	86.01 / 81.45	82.38 / **71.69**	0.2347 / 0.2186
All	**88.56 / 80.68**	**82.61** / 71.62	0.1905 / 0.1876

Motion Base Visualization. Our method does not directly produce motion segmentation as in Huang et al. (2025). However, the weight assignment map can provide similar output. In fact, the weight assignment output is a structural segmentation that can be simplified into a dynamic mask. Some existing evaluations focus on segmenting entire motion entities (e.g., humans), making them unsuitable for evaluating our method because static parts of an entity would not be assigned as dynamic. As shown in Figure 7, our assignment map provides physically grounded, skeleton-aware structure, clearly delineating the articulated parts undergoing movement rather than treating the object as a monolithic dynamic blob. The weight assignment is shared by all frames and mounted on the reference frame.

Ablation Study. We investigated the impact of different backbones and intermediate layer injection strategies on model performance. As shown in Table 6, the results demonstrate that the DA3 backbone outperforms VGGT largely, achieving higher scores, highlighting the effective of our backbone design. Among the fusion strategies, the linear injection method yields the best performance, surpassing both MLP and Replace variants in both metrics. This suggests that a simple linear transformation is sufficient and effective for integrating intermediate features.

In Table 7, we compare different fine-tuning settings for final stage. "Rigid" targets on structure-keeping using singularity loss, while "Non-Rigid" applies traditional point supervision on non-Rigid datasets, and "All" mix them together. On Tracking, the "All" setting achieves the best overall performance, indicating that mixed-data finetuning effectively improves tracking accuracy. On Deformity, while the "Non-Rigid" setting to some extend harness structural rigidity. In contrast, the "All" setting provides a more balanced performance on both criterion.

5 Conclusion

We have presented SM4RT, a framework that extends Geometry Foundation Models from static 3D reconstruction to structured 4D dynamic understanding. The central contribution is a shift in the motion representation paradigm. Rather than predicting independent point-wise displacements, SM4RT introduces Structure-of-Motion (SoM): scene dynamics are decomposed into a compact set of N motion bases, each represented as a temporal sequence of 6D twists in $\mathfrak{se}(3)$. Dense scene motion is then recovered by sparse, time-shared per-pixel assignment weights over these bases. This reduces the effective motion degrees of freedom from $3HW$ to $6N$, enforces object-level kinematic coherence by construction, and yields a representation that is both physically interpretable and parsimonious. Built on a pretrained GFM backbone with a parallel motion geometry encoder and decoder, SM4RT jointly infers 3D geometry, world-coordinate motion, and scene kinematic structure in a single forward pass from monocular RGB video. It achieves advanced performance on motion reconstruction and uniquely enables dynamic motion structure perception.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and for guidance on figure layout and presentation, to improve the readability of this paper. We have not used generative AI tools for any tasks with required disclosure, including generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. The remaining required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, sensitive data, or privacy concerns. All datasets used are publicly available, and we foresee no harmful applications or conflicts of interest arising from this research.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our work. The architecture and training details of the model are fully described in main content. All datasets used in our experiments are publicly available. The code and checkpoint of our model will be revealed in the article upon acceptance.

REFERENCES

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv: 2001.10773*, 2020.

Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD*. Springer, 2013.

Carlos Agelet de Saracibar. *Nonlinear Continuum Mechanics: An Engineering Approach*. Springer Cham, 2023.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *ICCV*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *ICCV*, 2025.

Vivek Gopalakrishnan. pytorchse3: Numerically stable implementation of batched se(3) exponential and logarithmic maps. https://github.com/eigenvivek/pytorchse3, 2023.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *CVPR*, 2022.

Nan Huang, Wenzhao Zheng, Chenfeng Xu, Kurt Keutzer, Shanghang Zhang, Angjoo Kanazawa, and Qianqian Wang. Segment any motion in videos. In *CVPR*, 2025.

Swarnim Jain and Shangzhe Wu. Rigidbench: Evaluating rigid-body physics in video generation models. In *ICLR Workshop on World Models*, 2026.

Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4d: Learning how things move in 3d from internet stereo videos. In *CVPR*, 2025.

Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *CVPR*, 2023.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In *ICCV*, 2025.

Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. In *CVPR*, 2026.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. Mapanything: Universal feed-forward metric 3D reconstruction. In *3DV*, 2026.

Skanda Koppula, Ignacio Rocco, Yi Yang, Joseph Heyward, Joao Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. Tapvid-3d: A benchmark for tracking any point in 3d. In *NeurIPS Datasets and Benchmarks Track*, 2024.

Zihang Lai, Eldar Insafutdinov, Edgar Sucar, and Andrea Vedaldi. Cowtracker: Tracking by warping instead of correlation. *arXiv preprint arXiv: 2602.04877*, 2026.

Hao Li, Zhengyu Zou, Fangfu Liu, Xuanyang Zhang, Fangzhou Hong, Yukang Cao, Yushi LAN, Manyuan Zhang, Gang Yu, Dingwen Zhang, and Ziwei Liu. IGGT: Instance-grounded geometry transformer for semantic 3d reconstruction. In *ICLR*, 2026.

Jiaxin Li and RHOS lab. OpenD4RT: An unofficial pytorch/gpu implementation of d4rt for 4d reconstruction and tracking. https://github.com/Lijiaxin0111/Open-d4rt, 2026. Accessed: 2026-06-20.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, 2026.

Xinhang Liu, Yuxi Xiao, Donny Y. Chen, Jiashi Feng, Yu-Wing Tai, Chi-Keung Tang, and Bingyi Kang. Trace anything: Representing any video in 4d via trajectory fields. In *ICLR*, 2025.

Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *CVPR*, 2019.

Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4rc: 4d reconstruction via conditional querying anytime and anywhere. In *ICML*, 2026.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*, 2016.

Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1st edition, 1994.

Tuan Duc Ngo, Ashkan Mirzaei, Guocheng Qian, Hanwen Liang, Chuang Gan, Evangelos Kalogerakis, Peter Wonka, and Chaoyang Wang. Deltav2: Accelerating dense 3d tracking. In *NeurIPS*, 2025a.

Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. In *ICLR*, 2025b.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-dpm: 4d video reconstruction with dynamic point maps. In *CVPR*, 2026.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *CVPR*, 2021.

René Vidal. A tutorial on subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

Bo Wang, Jian Li, Yang Yu, Li Liu, Zhenping Sun, and Dewen Hu. Scenetracker: Long-term scene flow estimation network. *TPAMI*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025b.

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *ICCV*, 2025c.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025d.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Pi3: Permutation-equivariant visual geometry learning. In *ICLR*, 2026.

Yuanqing Wu and Marco Carricato. Persistent manifolds of the special euclidean group se(3): A review. *Computer Aided Geometric Design*, 79:101872, 2020. ISSN 0167-8396.

Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. In *NeurIPS*, 2025a.

Yuqi Wu, Wenzhao Zheng, Sicheng Zuo, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Embodiedocc: Embodied 3d occupancy prediction for vision-based online scene understanding. In *ICCV*, 2025b.

Yuqi Wu, Tianyu Hu, Wenzhao Zheng, Yuanhui Huang, Haowen Sun, Jie Zhou, and Jiwen Lu. IVGT: Implicit visual geometry transformer for neural scene representation. *arXiv preprint arXiv: 2605.16258*, 2026.

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *CVPR*, 2024.

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Iurii Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In *ICCV*, 2025.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one d4rt at a time. In *CVPR*, 2026.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *ICLR*, 2025.

Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023.

Haoran Zhou and Gim Hee Lee. Motion4d: Learning 3d-consistent motion and semantics for 4d scene understanding. In *NeurIPS*, 2025.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. In *ICLR*, 2025.

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Shengyin Jiang, Long Chen, Zhi-Xin Yang, and Jiwen Lu. Dvgt: Driving visual geometry transformer. In *CVPR*, 2026a.

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Hanbing Li, Long Chen, Zhi-Xin Yang, and Jiwen Lu. Dvgt-2: Vision-geometry-action model for autonomous driving at scale. *arXiv preprint arXiv: 2604.00813*, 2026b.

Appendix

A Additional Method Details

A.1 Twist Representation

A rigid body motion can be described by how it translates and how it rotates. We use a *twist* for this purpose: a compact 6D motion vector $\boldsymbol{\xi} = [\boldsymbol{\rho}, \boldsymbol{\phi}] \in \mathbb{R}^6$, where $\boldsymbol{\rho} \in \mathbb{R}^3$ denotes the translational component and $\boldsymbol{\phi} \in \mathbb{R}^3$ denotes the rotational component. In other words, one twist specifies one rigid-body motion pattern shared by a group of points. Formally, the twist $\boldsymbol{\xi}$ corresponds to an element of the Lie algebra $\mathfrak{se}(3)$ via:

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} [\boldsymbol{\phi}]_\times & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathfrak{se}(3), \tag{10}$$

where $[\boldsymbol{\phi}]_\times \in \mathfrak{so}(3)$ is the skew-symmetric matrix of $\boldsymbol{\phi}$. The exponential map converts this 6D motion description into a finite rigid transformation in $\mathrm{SE}(3)$ (Murray et al., 1994), which maps a source point $\boldsymbol{p}$ to a target point $\boldsymbol{p}'$:

$$\boldsymbol{p}' = \mathrm{Exp}(\hat{\boldsymbol{\xi}})\boldsymbol{p}, \tag{11}$$

where homogeneous coordinates are omitted.

A.2 Motion Interpolation and Extrapolation

Given anchor frames, interpolation estimates intermediate states and extrapolation predicts states beyond the observed interval. SM4RT operates on temporal motion bases in twist space with a fixed assignment map $\boldsymbol{W}$, then reconstructs point positions using Equation (4).

Points with identical assignments share a rigid transformation at each interpolated or extrapolated time, preserving their relative geometry. Independent interpolation of point positions lacks this constraint and can shrink or enlarge rotating objects, as illustrated by the track-space results in Figure 3 (top). The two formulations coincide for pure translation with the same anchor states and interpolation weights. With rotation, exponentiating interpolated twists still yields rigid transformations for points sharing those twists, as shown by the motion-base results (bottom).

This parameter-free procedure extends inferred motion without explicitly modeling collisions, rebounds, or contact dynamics, which can change motion beyond the observed interval.

A.3 Scene Geometry Branch

Scene Geometry Encoder. Each frame $\boldsymbol{I}_t$ is patchified and encoded via a pretrained DINOv2 (Oquab et al., 2024) backbone into patch features $\boldsymbol{F}$. A learnable camera token and register tokens are appended per frame, and the full sequence is fed into a geometry-aware backbone, a pretrained DepthAnythingV3 (Lin et al., 2026) with local-global attention, to produce geometry tokens $\boldsymbol{G}$ enriched with 3D geometric cues, epipolar geometry, and cross-frame consistency.

Scene Geometry Decoder. The scene geometry decoder comprises a *camera head* and a *depth head*, both adopted from DepthAnythingV3 (Lin et al., 2026). The camera head outputs per-frame extrinsics and intrinsics. The depth head outputs depth maps $\boldsymbol{D}$. Together, they yield anchored world-coordinate source points via unprojection:

$$\boldsymbol{P}^0 = \mathrm{Reproject}(\boldsymbol{D}^0, \mathbf{K}^0), \tag{12}$$

where world coordinates are defined in the camera coordinate system of the source frame.

A.4 Adaptive Grouping

To enhance structured representation, we include an adaptive grouping strategy used in segmentation (Li et al., 2026) denoted as *SM4RT (Adaptive)*. Upon obtaining $\boldsymbol{W}$, hdbscan (Campello et al., 2013) (implemented by `cuml`) is applied to adaptively determine clustering classes and masks. The base of each masked region is replaced with the average value of its assigned region. The updated $\boldsymbol{W}'$ is then fed into Equation (4) and subsequent steps.

A.5 Geometry Supervision

Geometry Supervision. We supervise the predicted depth $\boldsymbol{D}$, camera pose $[\mathbf{R} \mid \mathbf{t}]$, and scene scale. The total loss is defined as $\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{cam}} + \mathcal{L}_{\text{scale}}$. Specifically, we employ a normalized ℓ_1 loss for depth estimation over valid pixels $\mathcal{M}$:

$$\mathcal{L}_{depth} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left| \text{norm}(\hat{\boldsymbol{D}}_i) - \text{norm}(\boldsymbol{D}_i^*) \right|. \tag{13}$$

For camera parameters, we apply a standard ℓ_1 loss against the ground truth. To constrain the inherent scale ambiguity, we compute the scale loss by minimizing the log-difference between the mean Euclidean distance of source-frame points $\boldsymbol{P}^0$ and a unit scale:

$$\mathcal{L}_{scale} = \left(\log \left(\text{mean}(|\boldsymbol{P}^0|_2) \right) - \log(1.0) \right)^2. \tag{14}$$

B Additional Experimental Details

B.1 Implementation Details

Our base model architecture follows Depth Anything 3 (Lin et al., 2026). We initialize **SM4RT** with pre-trained weights for the ① DINOv2 backbone, ② depth head, and ③ camera head. The number of motion tokens N is set to 10. Overall, the model has 1.38 billion parameters, with motion-specific modules accounting for 0.17 billion. Training was conducted on 8 NVIDIA RTX5880 GPUs for 5 days. We employed the AdamW optimizer and a hybrid schedule of linear warm-up (first 0.5 epochs) followed by cosine decay, with learning rate adjusted within [1e-7,1e-4]. For Lie group exponential mapping, we use PyTorchSE3 (Gopalakrishnan, 2023).

B.2 Training Details

We curated a collection of four dynamic datasets with motion annotations: Kubric (Movi-F) (Greff et al., 2022), Stereo4D (Jin et al., 2025), PointOdyssey (Zheng et al., 2023), and DynamicReplica (Karaev et al., 2023). Additionally, for fine-tuning, we include datasets with scene geometry annotations: ARKitScenes (Baruch et al., 2021), HyperSIM (Roberts et al., 2021), ScanNet++ (Yeshwanth et al., 2023), VKitti2 (Cabon et al., 2020), and Waymo (Sun et al., 2020).

For Kubric, we retrieve ① world-coordinate positions and transform them to the first-frame camera coordinate system using the Procrustes algorithm, together with ② background segmentation. For Stereo4D, we filter high-quality data with denser annotations ($> 10K$ valid points per scene) to avoid under-supervision. Point coordinates are transformed and normalized in world coordinates. The number of input frames varies from 12 to 24. The scene scale is normalized such that the mean Euclidean distance from all valid 3D points to the origin equals 1.

We employ a multi-stage training strategy. In stage I, we pretrain the model on Kubric for 100 epochs, which is necessary for learning base assignments. In stage II, we include all motion datasets for 150 epochs. Each epoch contains 8000 episodes. In stage III, we fine-tune depth and camera prediction on static and dynamic scenes for 25 epochs.

B.3 Efficiency Analysis

We compare inference time, computational cost, GPU memory, and tracking accuracy under the same input resolution and frame count. Measurements use BF16 on a single NVIDIA RTX5880 (48GB) and are averaged over the 147 samples of the Kubric evaluation set. Table 8 includes the full baseline comparison, with separate rows for DELTAv2 iteration counts and V-DPM execution modes. SM4RT takes 2.86 seconds per sample and uses 15.72GB of GPU memory, with an APD of 90.62. Adaptive grouping increases runtime to 11.64 seconds and memory to 16.65GB, with an APD of 89.56. Its effect on structure preservation is evaluated in Table 5.

V-DPM's full execution takes 64.79 seconds per sample, while its single-pass cost is 4.30 seconds. We report both modes to distinguish the cost of a single pass from the complete procedure. Single-pass APD is unavailable in this comparison.

Table 8: **Comparison of inference efficiency among different methods.** Results are measured under the same input resolution and frame number on BF16 precision averaged over the 147 samples of the Kubric evaluation set. Inference is conducted on a single NVIDIA RTX5880 (48G).

Method	Time (s/sample)	TFLOPS	Memory (GB)	APD
DELTAv2 (iter1) (Ngo et al., 2025a)	6.35	68.91	36.02	54.72
DELTAv2 (iter4) (Ngo et al., 2025a)	7.31	79.61	36.02	55.17
DELTAv2 (iter6) (Ngo et al., 2025a)	11.05	142.27	36.02	55.47
St4RTrack (Feng et al., 2025)	2.54	61.58	15.77	30.52
TraceAnything (Liu et al., 2025)	1.74	109.71	19.91	85.61
Any4D (Karhade et al., 2026)	4.53	103.95	16.88	79.78
V-DPM (All Pass) (Sucar et al., 2026)	64.79	6467.91	29.78	70.93
V-DPM (Single Pass) (Sucar et al., 2026)	4.30	438.00	24.22	N/A
4RC (Luo et al., 2026)	4.24	278.40	15.22	89.62
SM4RT	2.86	177.74	15.72	90.62
SM4RT (Adaptive)	11.64	210.85	16.65	89.56

C Rigid Entity Deformation Details

Motivation. Previous work on evaluating rigid-body motion focuses on 2D video generation models (Jain & Wu, 2026). We introduce Kubric-Rigid-Eval as a benchmark for 3D rigid-body motion reconstruction. Existing point-wise tracking metrics (Koppula et al., 2024) do not directly capture object deformation. VM complements these metrics by testing whether each moving object remains a coherent rigid entity.

Metric derivation. We measure rigid-entity deformation with a von Mises deformation score (VM) derived from continuum mechanics (de Saracibar, 2023, pp. 90–95). For each rigid segment, we first align the source point cloud to the target point cloud with an affine transformation:

$$(\boldsymbol{F}^*, \boldsymbol{c}^*) = \underset{\boldsymbol{F},\boldsymbol{c}}{\operatorname{argmin}} \sum_{i=1} \left\| \boldsymbol{x}_{tgt}^{(i)} - \left(\boldsymbol{F}\boldsymbol{x}_{src}^{(i)} + \boldsymbol{c} \right) \right\|_2^2, \tag{15}$$

where $\boldsymbol{F}^* = \boldsymbol{R}\boldsymbol{U}$ contains rotation $\boldsymbol{R}$ and stretch $\boldsymbol{U}$. Since this estimates a relative transform, the resulting $\boldsymbol{F}$ is scale-invariant. We then compute the right Cauchy-Green deformation tensor $\boldsymbol{C} = (\boldsymbol{F}^*)^\top \boldsymbol{F}^* = \boldsymbol{U}^\top \boldsymbol{U}$ and derive the Green-Lagrange strain tensor $\boldsymbol{E}$:

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}) = \frac{1}{2}\left(\boldsymbol{U}^\top \boldsymbol{U} - \boldsymbol{I}\right), \tag{16}$$

where $\boldsymbol{I}$ is the identity tensor. The tensor $\boldsymbol{E}$ captures stretching and shearing components in the global transformation, enabling the assessment of structural deformation. We compute the von Mises equivalent Green-Lagrange strain ϵ_{vM} from the deviatoric part of the strain tensor:

$$\epsilon_{\text{vM}} = \sqrt{\frac{2}{3}\boldsymbol{E}' : \boldsymbol{E}'} = \sqrt{\frac{2}{3}\sum_{i,j} \boldsymbol{E}'_{ij}\boldsymbol{E}'_{ij}}, \boldsymbol{E}' = \boldsymbol{E} - \frac{1}{3}\operatorname{tr}(\boldsymbol{E})\boldsymbol{I}. \tag{17}$$

This scalar quantifies the intensity of non-rigid distortion. Given video sequences of rigid-body scenes with ground-truth instance segmentations, we compute ϵ_{vM} weighted by the number of segment points and frames to obtain VM.

Evaluation protocol. Kubric-Rigid-Eval contains 147 video samples of rigid objects with ground-truth segmentation and world-centric motion. To compare camera- and world-coordinate trackers fairly, all methods are aligned by foreground-object scale and evaluated with both deformation and tracking metrics (Koppula et al., 2024). The evaluation reports ① initial foreground deformation, ② dynamic foreground deformation without and with edge filtering, and ③ dynamic background deformation. Dynamic object deformation measures how the target point cloud deforms from the source point cloud, while initial object deformation measures how well the initial position aligns with the ground-truth point cloud. Dynamic deformation is scale-invariant because it is measured within each sample, while initial deformation is aligned to the initial ground truth by object-wise scale. Together, these terms quantify deformation from complementary sources.

Discussion. As shown in Table 5, raw SM4RT outputs achieve the highest tracking AJ and APD among the compared methods, together with low dynamic background deformation. Adaptive grouping substantially reduces dynamic foreground deformation, with VM decreasing from 0.1905 to 0.0254 without edge filtering and from 0.1876 to 0.0191 with edge filtering. This improvement comes with a small tracking decrease, from 84.37 to 82.54 in AJ and from 90.62 to 89.56 in APD. One possible explanation is that inconsistent base assignments for some objects cause grouping to preserve structure while shifting predictions slightly farther from the ground truth.

D LIMITATIONS

The rigid-body assumption underlying SoM does not account for physical scene interactions such as collisions, contact dynamics, or fluid deformation, limiting the fidelity of future state prediction in highly complex scenarios. Additionally, the number of motion bases N is fixed at inference time. An adaptive or hierarchical decomposition could better handle scenes with highly variable numbers of moving objects. Therefore, SM4RT may also perform less reliably on highly deformable objects, leaving the balance between rigid and deformable motion as an open issue. We view SM4RT as establishing a foundation for structured 4D scene understanding and hope it inspires further work on physically grounded, interpretable dynamic representations.