FLOW4RT: SEEING MOTION THROUGH GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing and tracking dynamic scenes requires choosing a reference against which motion is measured: the background and independently moving objects support different rigid alignments across views. Directly regressing global geometry or camera poses makes this choice implicit and difficult to change after prediction. We present Flow4RT, a feedforward model that extends Flow4R's local geometry and relative motion formulation from image pairs to joint multi-view inference. The model aggregates information across all input views and predicts motion in both directions between each view and an anchor, supporting reconstruction and tracking within one formulation. Our key insight is that motion is displaced geometry. We predict local geometry for each view, then condition its features on a target view to predict where the same points move in viewpoint and time. A single dense head decodes both local and displaced point maps with their respective confidence and rigidity weights, avoiding duplicated geometry predictions and separate output heads. A closed-form, rigidity-weighted alignment recovers the transformations needed for reconstruction and tracking without a pose regression head. Adjusting the alignment weights selects a different rigid reference object while reusing the same dense predictions. On WorldTrack, Flow4RT achieves the best average tracking performance, improving APD by 2.4 points and reducing EPE by 17% over the strongest prior result for each metric, while ranking second in reconstruction on both metrics.

1 INTRODUCTION

Feedforward models such as DUSt3R (Wang et al., 2024) and VGGT (Wang et al., 2025a) have made dense 3D reconstruction possible directly from images. Extending this success to dynamic scenes requires recovering both scene geometry and the trajectories of its points while the camera and objects move independently. Methods such as MonST3R (Zhang et al., 2025b) adapt static reconstruction models to moving scenes, while PAGE-4D (Zhou et al., 2026) suppresses dynamic regions when estimating camera pose. These approaches highlight a fundamental question: *relative to what should motion be measured?*

In a static scene, visible surfaces support a common rigid alignment between views. In a dynamic scene, the background and each independently moving rigid object support different alignments. For the street scene in Fig. 1, aligning the ground reveals the car's motion, whereas aligning the car makes the surroundings move around it. Choosing an anchor image fixes a viewpoint at one instant, but does not specify which scene region should remain stationary across time. A model that directly regresses global points or camera poses must resolve this choice implicitly, making its reference difficult to inspect or change after prediction.

Flow4R (Qian et al., 2026) makes that choice explicit through pixel-level predictions. Given two views, it predicts local geometry and relative motion, represented by a local point map and a displaced point map locating the same points at the other view's viewpoint and time. These maps do not require choosing a stationary scene region. A learned *rigidity map* then weights their rigid alignment to recover the relative camera pose. The reference choice is thus isolated in an interpretable weight map: changing the weights can select another rigid reference object without predicting geometry or motion again. However, Flow4R processes only two views at a time. Applying it independently to many pairs neither enforces agreement between their predictions nor lets intermediate observations help connect views with little overlap.

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.

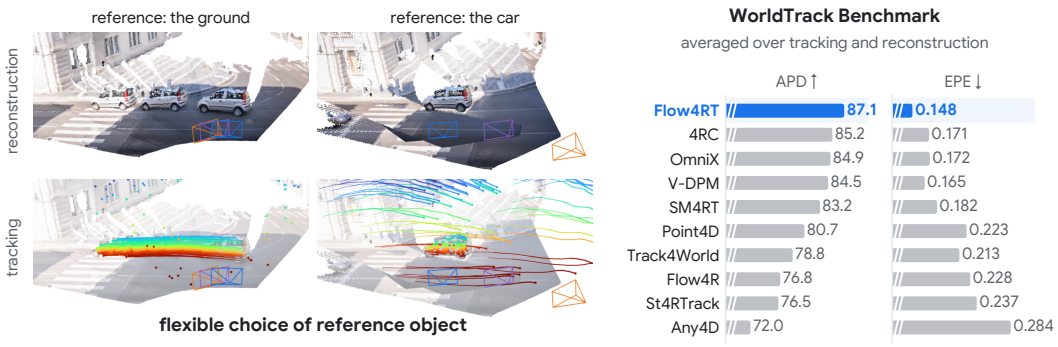


Figure 1: **Flow4RT predicts dense geometry and relative motion across views.** Left: aligning the ground shows the car at three positions along its path; aligning the car fuses its observations while the plaza moves around it. Both reconstructions and their corresponding trajectories use the same network predictions. Right: averaged over tracking and reconstruction on the WorldTrack benchmark, Flow4RT leads every prior method on both APD and EPE.

We introduce Flow4RT, a feedforward multi-view model that preserves this separation between dense prediction and reference selection while jointly reasoning over all input views. The challenge is to retain Flow4R's unified treatment of reconstruction and tracking without predicting a view's local geometry again each time it is paired with a different target view. Our solution combines bidirectional pairing with the key insight that *motion is displaced geometry*.

First, we pair each input view with an anchor view and predict relative motion in both directions. Reconstruction uses predictions from every other view *to* the anchor to align their local geometry in a common coordinate frame. Tracking uses predictions in the reverse direction, *from* the anchor to every other view, to follow the anchor's points across time. Both tasks therefore use the same bidirectional graph, with a rigid alignment on each edge determined by the selected reference region. Flow4RT predicts relative motion for each ordered view pair from features that already incorporate all input views, combining pairwise motion prediction with multi-view context.

Second, local geometry belongs to a source view and need not be predicted again for each target. We compute one local point map $\mathbf{P}$ and confidence map $\mathbf{C}$ per input view, then condition its geometric features on a target view to predict the displaced point map $\mathbf{P}_{vt}$ and rigidity map $\mathbf{W}$. The two output pairs, $(\mathbf{P}, \mathbf{C})$ and $(\mathbf{P}_{vt}, \mathbf{W})$, share a structure: a 3D point and a scalar weight at each pixel. Confidence and rigidity serve different purposes, but this structural symmetry allows a single dense head to decode both pairs, with separate activations for the weights. This is the architectural principle behind *seeing motion through geometry*: geometry blocks aggregate information within and across views, while motion blocks condition those features on a target at multiple depths. View pairing is confined to the motion branch, avoiding redundant local geometry predictions; sharing the output head matches the accuracy of separate heads in our ablation (Table 3).

A closed-form, rigidity-weighted alignment of the predicted point maps yields the transformations needed for both world-space reconstruction and tracking, without a pose regression head. Adjusting the alignment weights turns reference ambiguity into flexibility: the same network predictions can describe a scene relative to the ground or a moving rigid object (Fig. 1). On WorldTrack, Flow4RT achieves the best average tracking performance, improving APD by 2.4 points and reducing EPE by 17% relative to the strongest prior result for each metric. It ranks second in reconstruction on both metrics and is the only evaluated method in the top two for both tasks (Tables 1 and 2).

Our contributions are:

- A multi-view extension of local geometry and relative motion that unifies reconstruction and tracking through bidirectional relations with joint multi-view context and a selectable rigid reference.
- An architecture that predicts local geometry once per view and decodes target-conditioned motion with the same dense head, exploiting the structural symmetry between local and displaced point maps.

- A feedforward model that achieves state-of-the-art average tracking performance on World-Track and ranks second in reconstruction, recovering both through closed-form alignment without a pose regression head.

2 Related Work

Feedforward 3D reconstruction. Classical structure-from-motion systems such as COLMAP (Schönberger & Frahm, 2016) reconstruct scenes through feature matching, triangulation, and bundle adjustment. DUSt3R (Wang et al., 2024) instead predicts dense point maps from image pairs, and MASt3R (Leroy et al., 2024) augments this representation with features for matching. Subsequent methods support image streams, including Spann3R (Wang & Agapito, 2025), CUT3R (Wang et al., 2025b), and StreamVGGT (Zhuo et al., 2026), or joint multi-view inference, including Fast3R (Yang et al., 2025), MUSt3R (Cabon et al., 2025), VGGT (Wang et al., 2025a), and VGGT-Ω (Wang et al., 2026a). Other developments include metric reconstruction with optional geometric inputs in MapAnything (Keetha et al., 2026), depth-ray prediction in Depth Anything 3 (Lin et al., 2026a), and permutation-equivariant geometry prediction in π^3 (Wang et al., 2026b).

Point tracking. PIPs (Harley et al., 2022) recasts pixel tracking as long-range trajectory estimation through occlusions, and TAPIR (Doersch et al., 2023), LocoTrack (Cho et al., 2024), CoTracker (Karaev et al., 2024), and CoTracker3 (Karaev et al., 2025) improve its accuracy and speed in the image plane. TAPVid-3D (Koppula et al., 2024) benchmarks the task in 3D, where SpatialTracker (Xiao et al., 2024), DELTA (Ngo et al., 2025), and TAPIP3D (Zhang et al., 2025a) lift pixels into 3D using given depth and track them there, TAPIP3D optionally with given camera poses. These trackers either stay in the image plane or take geometry from elsewhere, whereas Flow4RT infers geometry and motion together from the images alone.

Dynamic reconstruction. MonST3R (Zhang et al., 2025b) fine-tunes DUSt3R on dynamic scenes, Align3R (Lu et al., 2025) conditions it on monocular depth for temporally consistent video depth, and MegaSaM (Li et al., 2025) folds motion cues into bundle adjustment. Easi3R (Chen et al., 2025) and VGGT4D (Hu et al., 2026) mine motion cues from the attention of DUSt3R and VGGT without training, while PAGE-4D (Zhou et al., 2026) fine-tunes VGGT with a dynamics-aware mask that suppresses moving regions for pose estimation. Geo4D (Jiang et al., 2025) instead repurposes a video diffusion model to predict point, disparity, and ray maps. Their primary outputs are per-frame geometry and camera poses; dense point trajectories require modeling the motion of scene points across time.

4D reconstruction and tracking. Dynamic Point Maps (Sucar et al., 2025), D^{2}USt3R (Han et al., 2025), and POMATO (Zhang et al., 2025c) extend the point map to carry time, regressing geometry at more than one timestamp. Multi-view successors encode a video once and decode geometry at a requested time, either densely per target time, as in V-DPM (Sucar et al., 2026) and 4RC (Luo et al., 2026), or per point query, as in D4RT (Zhang et al., 2026), UniQuery4R (Chen et al., 2026), and Point4D (Jeon et al., 2026). Any4D (Karhade et al., 2026) instead factors multi-view predictions into per-view depth, camera parameters, and scene flow. Other approaches use dedicated motion representations: OmniX (Jiang et al., 2026a) predicts trajectory fields from dynamic tokens, while SM4RT (Lin et al., 2026b) combines rigid $SE(3)$ motion bases with per-pixel assignment weights. Track4World (Lu et al., 2026) estimates dense flow between frame pairs using 3D correlation, and SpatialTrackerV2 (Xiao et al., 2025) jointly estimates geometry, camera motion, and point trajectories. Flow4R (Qian et al., 2026) instead predicts local geometry and relative motion for image pairs, then recovers camera pose by weighting the rigid alignment of its dense predictions. Flow4RT extends this formulation to joint multi-view inference, conditioning source geometry features on the target view to predict displaced geometry with a shared dense head. Its alignment weights make the reference region explicit and adjustable, allowing the same dense predictions to support reconstruction and tracking relative to different rigid scene regions (Fig. 1).

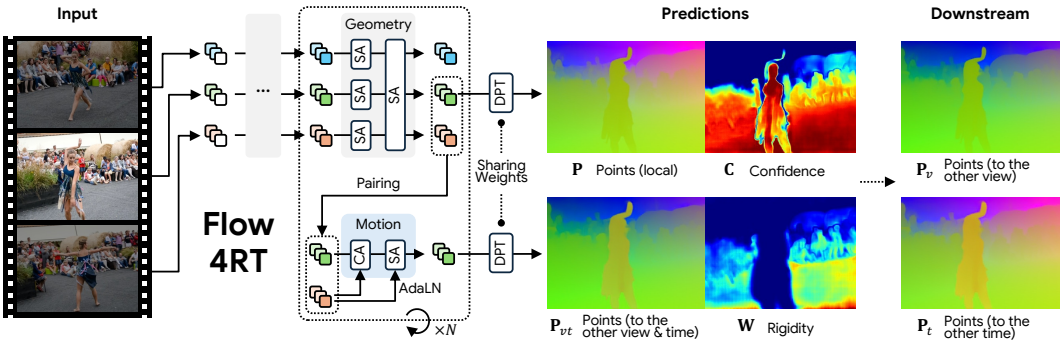


Figure 2: **Flow4RT predicts pixel-aligned local geometry and relative motion for every view.** Geometry blocks attend within and across views to predict a local point map $\mathbf{P}$ and its confidence $\mathbf{C}$. Motion blocks condition these features on a target view to predict displaced points $\mathbf{P}_{vt}$ and rigidity $\mathbf{W}$. One shared dense head decodes both branches at four read-out depths.

3 Method

Flow4RT jointly processes the input views to predict local geometry and relative motion between selected view pairs.

3.1 Local Geometry and Relative Motion

Input. Flow4RT takes a set of N images $\{\mathbf{I}^n\}_{n=1}^N$ of a dynamic scene, with $\mathbf{I}^n \in \mathbb{R}^{H\times W\times 3}$. We denote the viewpoint of view n by v_n and its capture time by t_n.

Two pairs of predictions. For every view, the geometry branch predicts a local point map $\mathbf{P}$, which places each pixel at that view's own viewpoint and time, and a confidence map $\mathbf{C}$. Toward a target view, the motion branch predicts a displaced point map $\mathbf{P}_{vt}$, which places the same pixels at the target's viewpoint and time, and a rigidity map $\mathbf{W}$:

$$\mathbf{P}, \mathbf{P}_{vt} \in \mathbb{R}^{H\times W\times 3}, \qquad \mathbf{C} \in (1,\infty)^{H\times W}, \qquad \mathbf{W} \in (0,1)^{H\times W}.$$

Both point maps are indexed by source-view pixels, so their entries correspond even when the associated point is not visible in the target image. The rigidity is normalized so that $\sum_i \mathbf{W}^{(i)} = 1$, where i indexes source-view pixels. For clarity, we omit the source and target indices until Section 3.3.

Shared output structure. Our key insight is that *motion is displaced geometry*: $\mathbf{P}$ and $\mathbf{P}_{vt}$ describe the same points at different viewpoints and times. Both are point maps, so both can be decoded by the same dense prediction head from differently conditioned features. Their accompanying weights have distinct roles: confidence $\mathbf{C}$ weights point regression during training, whereas rigidity $\mathbf{W}$ weights the rigid alignment used to estimate camera motion. Both output pairs contain a 3D point and a scalar weight per pixel, allowing a shared head with separate scalar activations (Section 3.4).

3.2 Recovering Rigid Alignment and Scene Motion

Rigidity-weighted alignment. For points on a stationary scene region, the change from $\mathbf{P}$ to $\mathbf{P}_{vt}$ is explained by a single rigid transformation. We estimate this transformation by aligning corresponding entries of the two point maps, weighted by $\mathbf{W}$:

$$\hat{\mathbf{T}} = \underset{\mathbf{T}\in SE(3)}{\arg\min} \sum_{i=1}^{HW} \mathbf{W}^{(i)} \left\| \mathbf{P}_{vt}^{(i)} - \mathbf{T}\mathbf{P}^{(i)} \right\|_2^2. \tag{1}$$

This is a weighted Procrustes problem, solved in closed form by the singular value decomposition of the $\mathbf{W}$-weighted cross-covariance of the two point sets. We train $\mathbf{W}$ by differentiating through

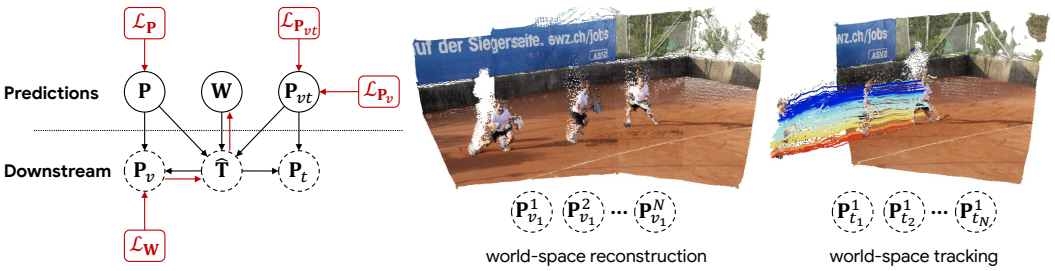


Figure 3: **Reconstruction and tracking from rigid alignment.** Left: predicted quantities (solid), derived quantities (dashed), and supervision paths (red). Rigidity $\mathbf{W}$ is supervised through the alignment solve (Eq. (1)). Right: with view 1 as the anchor, reconstruction aligns the court while preserving the player's positions across time; tracking follows the anchor's points through the sequence (Eq. (4)).

this solve (Section 3.5). When the weights select stationary scene points, $\hat{\mathbf{T}}$ estimates the relative camera pose without a pose regression head. Changing the weights to select a moving rigid object instead yields an alignment relative to that object.

Reconstruction and tracking from the same alignment. The forward transformation places source-time geometry in the target camera frame, while its inverse expresses target-time geometry in the source camera frame:

$$\mathbf{P}_v = \hat{\mathbf{T}}\mathbf{P}, \qquad \mathbf{P}_t = \hat{\mathbf{T}}^{-1}\mathbf{P}_{vt}. \tag{2}$$

Thus, $\mathbf{P}_v$ supports reconstruction by aligning the source point cloud with the target viewpoint, and $\mathbf{P}_t$ supports tracking by locating source points at the target time in the source coordinate frame. The total displacement $\mathbf{F} = \mathbf{P}_{vt} - \mathbf{P}$ can also be decomposed into the displacement induced by camera motion and the residual due to scene motion:

$$\mathbf{F}_v = \mathbf{P}_v - \mathbf{P}, \qquad \mathbf{F}_t = \mathbf{F} - \mathbf{F}_v. \tag{3}$$

3.3 World-Space Reconstruction and Tracking

We now apply these pairwise operations across all input views to reconstruct a scene and track its points in a shared coordinate frame. The geometry branch runs once over all N views; the motion branch runs on whichever ordered pairs are requested at inference. We choose one input view as the anchor a and express all reconstructed points and trajectories in its camera coordinate system. Using stationary scene regions as the alignment reference, world-space reconstruction and tracking from a are the collections

$$\text{Reconstruction: } \left\{\mathbf{P}^{n}_{v_a}\right\}_{n=1}^{N}, \qquad \text{Tracking: } \left\{\mathbf{P}^{a}_{t_n}\right\}_{n=1}^{N}, \tag{4}$$

which require the pairs (n, a) and (a, n), respectively; both are illustrated in Fig. 3. Here, $\mathbf{P}^{n}_{v_a}$ is view n's geometry transformed to the anchor viewpoint at its original time, while $\mathbf{P}^{a}_{t_n}$ locates the anchor's points at time t_n in the anchor coordinate frame. Reconstruction uses N predictions toward the anchor, while tracking uses N predictions from the anchor. Both sets include the self-pair (a, a): $\mathbf{P}^{a}_{v_a}$ supplies the anchor's reconstruction, and $\mathbf{P}^{a}_{t_a}$ supplies its initial track positions. Local geometry is computed once per view and reused across pairs, which can be processed in parallel in a single batch. The anchor and requested pairs are selected at inference, so the same trained model supports either task or both.

Choosing the reference region. The anchor view specifies the output coordinates at one instant, while the rigidity weights specify which scene region remains stationary across time. By adjusting $\mathbf{W}$ to select another rigid region and recomputing the alignment, we can express reconstruction and tracking relative to a moving object using the same dense predictions (Fig. 1).

3.4 ARCHITECTURE

Geometry blocks. We convert the images into patch tokens and process them jointly with a vision transformer composed of geometry blocks (Fig. 4). Early blocks attend within each view; from a chosen depth onward, blocks alternate within-view and cross-view self-attention. Cross-view attention provides shared context while each output point map remains in its own camera frame. Tokens are read out at four depths to predict local geometry.

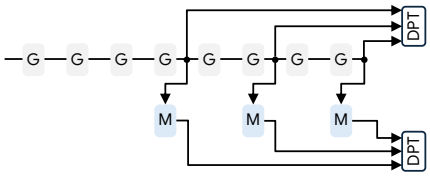


Figure 4: **Backbone layout.** Every backbone block is a geometry block (G), and features from several depths are fused to predict local geometry. Motion blocks (M) branch off at the same depths to predict relative motion. Block and read-out counts are schematic.

Motion blocks. At each of the four read-out depths, motion blocks map the source view's geometric tokens toward the target view (Fig. 2). Each block has a cross-attention layer, with source tokens as queries and target tokens as keys and values, followed by a self-attention layer with AdaLN (Peebles & Xie, 2023). AdaLN is modulated by a token representing the target view's time. This token starts from the same learned embedding in every view at the first cross-view block, then becomes view-specific as it is updated alongside the image tokens. Because motion blocks branch off the geometry blocks, each pairwise prediction has access to context from all input views, including observations outside the source–target pair.

Shared head. A single DPT head fuses tokens from the four depths coarse-to-fine into pixel-aligned maps. It decodes the geometry branch into $(\mathbf{P}, \mathbf{C})$ and the motion branch into $(\mathbf{P}_{vt}, \mathbf{W})$, using the same head parameters for both. Separate scalar activations produce confidence and rigidity in their respective ranges, and rigidity is normalized over pixels. Local geometry is decoded once per view, while displaced geometry is decoded for each requested ordered pair.

3.5 TRAINING

We train jointly on static and dynamic data using four objectives:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\mathbf{P}} + \lambda_2 \mathcal{L}_{\mathbf{P}_{vt}} + \lambda_3 \mathcal{L}_{\mathbf{P}_v} + \lambda_4 \mathcal{L}_{\mathbf{W}}, \tag{5}$$

where $\mathcal{L}_{\mathbf{P}}$ supervises local geometry and $\mathcal{L}_{\mathbf{P}_{vt}}$ supervises displaced geometry from scene flow and 3D tracks, both weighted by confidence $\mathbf{C}$. The auxiliary term $\mathcal{L}_{\mathbf{P}_v}$ supervises displacement induced by the ground-truth camera pose; on dynamic data, detached rigidity weights suppress moving regions. Finally, $\mathcal{L}_{\mathbf{W}}$ trains rigidity through the differentiable alignment in Eq. (1), with gradients to the point maps blocked, so no rigidity annotations are needed. We use $\lambda_1 = 1$ and $\lambda_2 = \lambda_3 = \lambda_4 = 0.5$, with scale-normalized point maps and confidence regularization weight $\alpha = 0.2$. For each training sample, a random derangement assigns every view a different target, and we predict both directions of each pair. Section D gives the full loss definitions; Fig. 3 shows their supervision paths.

4 EXPERIMENTS

We evaluate whether a shared geometry and motion representation supports accurate world-space reconstruction and tracking. We also assess video depth, camera pose, and runtime, and ablate the design and initialization of the dense prediction head.

4.1 SETUP

Training data. We train on a mixture of static and dynamic, real and synthetic datasets, listed with their supervision and samples per epoch in Table 5 (Section B).

Implementation details. We initialize the backbone from 4RC (Luo et al., 2026) and train the motion blocks and shared DPT head from scratch. The model has 1.73 B parameters and is trained

Table 1: **World-space 3D tracking on WorldTrack.** Flow4RT achieves the highest average APD and lowest average EPE. Shading marks the top three per column.

	ADT		DR		PO		PS		**Average**	
	APD↑	EPE↓	APD↑	EPE↓	APD↑	EPE↓	APD↑	EPE↓	APD↑	EPE↓
St4RTrack (Feng et al., 2025)	76.0	0.268	73.7	0.268	68.0	0.314	69.7	0.264	71.8	0.278
Flow4R (Qian et al., 2026)	78.6	0.237	78.5	0.223	71.1	0.279	64.3	0.313	73.1	0.263
Any4D (Karhade et al., 2026)	58.5	0.408	77.8	0.231	65.2	0.413	63.8	0.300	66.3	0.338
V-DPM (Sucar et al., 2026)	85.7	0.170	75.3	0.249	81.3	0.189	78.8	0.180	80.3	0.197
4RC (Luo et al., 2026)	83.2	0.190	82.3	0.195	79.3	0.282	69.4	0.256	78.6	0.231
Track4World (Lu et al., 2026)	79.9	0.217	71.2	0.274	66.3	0.363	73.0	0.232	72.6	0.272
OmniX (Jiang et al., 2026a)	91.7	0.103	78.3	0.230	75.7	0.303	77.3	0.199	80.8	0.209
SM4RT (Lin et al., 2026b)	87.5	0.146	80.9	0.205	77.5	0.297	79.3	0.182	81.3	0.207
Point4D (Jeon et al., 2026)	90.6	0.134	77.8	0.244	69.5	0.370	74.0	0.228	78.0	0.244
Flow4RT	88.9	0.132	85.4	0.166	81.3	0.189	79.2	0.170	83.7	0.164

for 100 epochs on 8 H100 GPUs, with 352 views per step at a long side of 518 pixels. Architecture settings and the training schedule are detailed in Section C.

Benchmark. We evaluate in world coordinates on the WorldTrack benchmark (Feng et al., 2025). Tracking is evaluated on two real datasets, Aria Digital Twin (ADT) (Pan et al., 2023) and Panoptic Studio (PS) (Joo et al., 2015), and two synthetic ones, Point Odyssey (PO) (Zheng et al., 2023) and Dynamic Replica (DR) (Karaev et al., 2023), each with 50 sequences over their first 64 frames. Reconstruction is evaluated on the synthetic Point Odyssey and the real TUM-Dynamics (TUM) (Sturm et al., 2012). Section F.1 further reports video depth and camera pose under the MonST3R protocol (Zhang et al., 2025b).

Metrics. Following St4RTrack (Feng et al., 2025), we report the Average Percentage of 3D Points within Delta (APD): the fraction of points within $\delta_{3D} \in \{0.1, 0.3, 0.5, 1.0\}$ of the ground truth, averaged over thresholds and frames. We also report the Euclidean End-Point Error (EPE), which measures how large the residual is rather than only whether it crosses a threshold. Before scoring, predictions and ground truth are each normalized by their global median per sequence, so thresholds and errors are in normalized units.

Baselines. We compare against feedforward methods that output world-space geometry and motion without per-scene optimization, including St4RTrack (Feng et al., 2025) and Flow4R (Qian et al., 2026) as representative two-view methods, and Any4D (Karhade et al., 2026), V-DPM (Sucar et al., 2026), 4RC (Luo et al., 2026), Track4World (Lu et al., 2026), OmniX (Jiang et al., 2026a), SM4RT (Lin et al., 2026b), and Point4D (Jeon et al., 2026) as multi-view methods, which differ in how time and viewpoint enter. All multi-view methods receive frames resized to a long side of 518 pixels, except Track4World, which keeps the canvases of its own evaluation, up to 360×640. St4RTrack and Flow4R run at their native long side of 512 pixels.

4.2 World-Space Tracking and Reconstruction

Tracking. Table 1 reports world-space 3D point tracking. Scoring points in world coordinates jointly evaluates local geometry, camera motion, and object motion. Flow4RT attains the best average on both metrics, 83.7 APD and 0.164 EPE. It leads SM4RT by 2.4 APD points and lowers the best prior EPE, V-DPM's 0.197, by 17%. Flow4RT achieves these results with a dense prediction head shared between geometry and motion.

Reconstruction. Table 2 evaluates the reconstructed point clouds in world space. 4RC leads on average with 91.9 APD and 0.112 EPE, driven by Point Odyssey (97.0 APD); Flow4RT is second at 90.4 APD and 0.131 EPE. On the real TUM sequences, Flow4RT has the highest APD, 90.3, and Track4World the lowest EPE, 0.125. Flow4RT is the only evaluated method in the top two on both average metrics for both tasks, demonstrating balanced performance across reconstruction and

Table 2: **World-space 3D reconstruction on WorldTrack.** Flow4RT ranks second on both averages while recovering camera pose via rigid alignment. Shading marks the top three per column.

	PO		TUM		**Average**	
	APD↑	EPE↓	APD↑	EPE↓	APD↑	EPE↓
St4RTrack (Feng et al., 2025)	78.7	0.205	83.4	0.185	81.1	0.195
Flow4R (Qian et al., 2026)	81.0	0.182	79.9	0.202	80.4	0.192
Any4D (Karhade et al., 2026)	75.1	0.221	80.2	0.240	77.6	0.230
V-DPM (Sucar et al., 2026)	87.5	0.130	90.2	0.136	88.8	0.133
4RC (Luo et al., 2026)	97.0	0.070	86.7	0.154	91.9	0.112
Track4World (Lu et al., 2026)	81.0	0.185	89.0	0.125	85.0	0.155
OmniX (Jiang et al., 2026a)	91.5	0.108	86.8	0.163	89.1	0.135
SM4RT (Lin et al., 2026b)	90.8	0.117	79.4	0.196	85.1	0.156
Point4D (Jeon et al., 2026)	81.5	0.184	85.3	0.218	83.4	0.201
Flow4RT	90.6	0.110	90.3	0.153	90.4	0.131

Table 3: **Ablation of the DPT head.** ✓ marks one head shared by $(\mathbf{P}, \mathbf{C})$ and $(\mathbf{P}_{vt}, \mathbf{W})$, which matches two separate heads. Training the head from scratch outperforms initializing it from 4RC. All variants are trained for 25 epochs under the same protocol.

Shared	Initialization	Tracking		Reconstruction		**Average**	
		APD↑	EPE↓	APD↑	EPE↓	APD↑	EPE↓
	4RC	82.9	0.174	88.4	0.159	85.6	0.166
	From scratch	83.6	0.166	90.6	0.146	87.1	0.156
✓	From scratch	83.5	0.164	90.7	0.145	87.1	0.155

tracking. Averaged over the two, it reaches 87.1 APD and 0.148 EPE and leads every baseline on both metrics (Fig. 1).

Qualitative comparison. Fig. 5 compares Flow4RT with 4RC and OmniX on clips from WorldTrack and DAVIS (Perazzi et al., 2016). The Panoptic Studio basketball clip has three independently moving entities, and Flow4RT's trajectories follow the ground truth most closely. In the egocentric Aria clip, the observer carries an object around an apartment. Flow4RT aligns the room despite low covisibility and tracks the object until it leaves the view, whereas 4RC loses it entirely. In the DAVIS example, all three methods produce coherent reconstructions, while Flow4RT better preserves the tracks under large camera and object motion through the final frame. More examples are in Section F.2.

4.3 Ablation Studies

One head or two. We test whether the shared output structure of $(\mathbf{P}, \mathbf{C})$ and $(\mathbf{P}_{vt}, \mathbf{W})$ allows one dense head to predict both pairs without sacrificing accuracy. With random initialization, shared and separate heads both reach 87.1 average APD, while the shared head yields slightly lower average EPE (0.155 versus 0.156; Table 3).

Head initialization. With separate heads, random initialization improves average APD from 85.6 to 87.1 and reduces EPE from 0.166 to 0.156 compared with initialization from 4RC. Under this training protocol, transferring the pretrained head provides no benefit, so our final model initializes the shared head randomly while retaining the pretrained backbone.

4.4 Additional Evaluation

Video depth and camera pose. Under the MonST3R evaluation protocol, Flow4RT ranks among the top three on both depth metrics across Sintel, Bonn, and KITTI and achieves the lowest errors on all three ScanNet pose metrics (Table 8). On TUM, it is competitive in translational relative pose error but trails the leading methods in absolute trajectory and rotational relative pose error. The full protocol is in Section F.1.

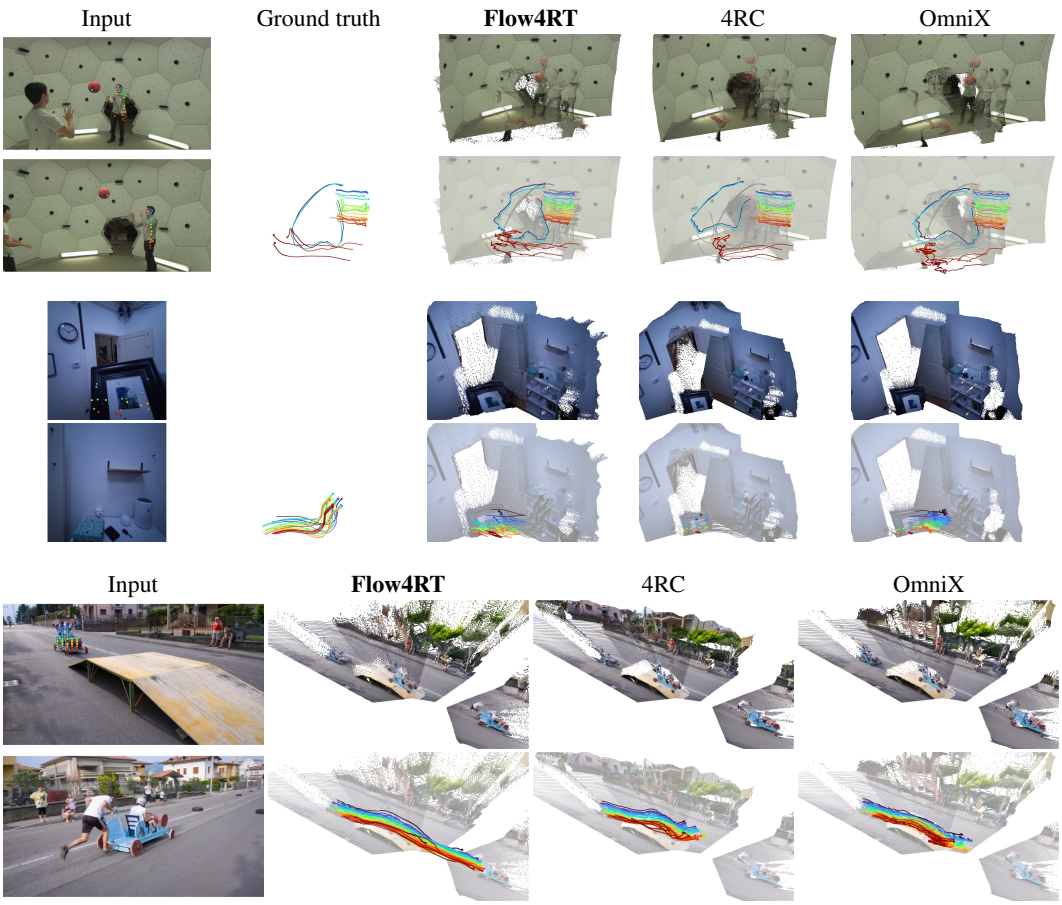


Figure 5: **4D reconstruction and world-space tracking.** For each clip, the upper row stacks the predicted point maps of the first, middle, and last frames in world space, and the lower row draws the predicted world trajectories of first-frame query points. The input column shows the first frame above the last, with the query points colored to match their trajectories. Gray curves are the ground-truth tracks; an empty reconstruction cell marks a clip without dense ground-truth geometry. DAVIS clips have no ground truth, so their query points come from the first-frame object mask. All methods share one virtual camera, one set of query points, and one global scale, so the panels differ only in the predictions. Flow4RT keeps tracking through large camera and object motion.

Runtime. For a 64-view clip on one H100 GPU, Flow4RT produces reconstruction and tracking outputs at 28.3 views/s with 20.6 GB peak GPU memory (Table 6). Its throughput ranks second to SM4RT (30.6 views/s); Section E gives the protocol and scaling results.

5 Conclusion

We presented Flow4RT, a feedforward multi-view reconstruction and tracking model built on *motion as displaced geometry*: source and target geometric features interact to predict displaced points using the same head as local points. Rigidity-weighted alignment recovers both tasks without pose regression and lets the reference region change after prediction. Flow4RT ranks first in tracking and second in reconstruction on WorldTrack, with competitive depth and pose estimates and the second-highest throughput among evaluated multi-view methods.

Limitations and Outlook. Flow4RT does not explicitly model temporal continuity, making trajectories during prolonged occlusion challenging; temporal priors could help when visual evidence is unavailable. Metric applications require an external scale reference. Reference selection also depends on reliable geometry: changing alignment weights cannot correct errors in the point maps.

AI USE STATEMENT

We used generative AI tools in the following parts of this work: providing feedback on the research methodology and experimental design, implementing methods and editing software code, supporting qualitative analysis, creating and modifying scientific figures and other artifacts, searching for information and identifying relevant literature, summarizing existing literature, drafting parts of the paper, editing the paper for readability, and formatting references. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims or their proofs, propose or refine hypotheses, interpret results, or clean and reformat datasets; translation was not needed. All AI-assisted work was reviewed by the authors, who take full responsibility for the content of this paper, including text, claims, code, and figures produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 specifies the formulation, the closed-form rigidity-weighted solve (Eq. (1)), and the architecture; Section D defines the training losses, with notation summarized in Table 4. Sections C and 4.1 specify the backbone and its initialization, parameter counts, input resolutions, augmentations, batch size, hardware, optimizer, and learning-rate schedule. All training datasets are publicly available, and Table 5 gives the supervision each provides and the number of samples drawn from it per epoch. Evaluation uses the public WorldTrack benchmark (Feng et al., 2025) and the MonST3R protocol (Zhang et al., 2025b), with WorldTrack metrics and normalization described in Section 4.1 and the depth and pose protocol in Section F.1; all baselines receive the same frames at the resolutions stated there. The runtime protocol, including what is timed and on which hardware, is described in Section E. We will release the code and model weights upon publication.

REFERENCES

Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *ECCV*, 2022.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, 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*, 2021.

Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

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

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. MUSt3R: Multi-view network for stereo 3D reconstruction. In *CVPR*, 2025.

Tiancheng Chen, Sheng Tang, Wenhua Jin, Weiqi Zhang, Juntong Fang, Junsheng Zhou, and Zesong Li. UniQuery4R: Unified 4D scene reconstruction from a single query. *arXiv preprint arXiv:2608.17283*, 2026.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3R: Estimating disentangled motion from DUSt3R without training. In *ICCV*, 2025.

Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *ECCV*, 2024.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017.

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.

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.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *CVPR*, 2022.

Jisang Han, Honggyu An, Jaewoo Jung, Takuya Narihira, Junyoung Seo, Kazumi Fukuda, Chaehyun Kim, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. D^2USt3R: Enhancing 3D reconstruction for dynamic scenes. In *NeurIPS*, 2025.

Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, 2022.

Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. VGGT4D: Mining motion cues in visual geometry transformers for 4D scene reconstruction. In *CVPR Findings*, 2026.

Minsik Jeon, Jay Karhade, Deva Ramanan, and Shubham Tulsiani. Point4D: Long-range 4D motion reconstruction. *arXiv preprint arXiv:2609.09145*, 2026.

Yanqin Jiang, Tengfei Wang, Zhengwei Wang, Chenjie Cao, Junta Wu, Wenhan Luo, Weiming Hu, Jin Gao, and Chunchao Guo. OmniX: Any-view and any-time 4D reconstruction via feed-forward trajectory fields. In *ECCV*, 2026a.

Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4D: Leveraging video generators for geometric 4D scene reconstruction. In *ICCV*, 2025.

Zeren Jiang, Yushi Lan, Yihang Luo, Yufan Deng, Zihang Lai, Edgar Sucar, Christian Rupprecht, Iro Laina, Diane Larlus, Chuanxia Zheng, and Andrea Vedaldi. Syn4D: A multiview synthetic 4D dataset. In *ECCV*, 2026b.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic Studio: A massively multiview system for social motion capture. In *ICCV*, 2015.

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, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling 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, Joe Heyward, João Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. TAPVid-3D: A benchmark for tracking any point in 3D. In *NeurIPS Datasets and Benchmarks*, 2024.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *ECCV*, 2024.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast, and robust structure and motion from casual dynamic videos. In *CVPR*, 2025.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *ICLR*, 2026a.

Shing Ho J. Lin, Wenzhao Zheng, Dong Zhuo, Yuqi Wu, Jie Zhou, and Jiwen Lu. SM4RT: Learning structured motion geometry for 4D reconstruction. *arXiv preprint arXiv:2607.22534*, 2026b.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *CVPR*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3R: Aligned monocular depth estimation for dynamic videos. In *CVPR*, 2025.

Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4World: Feedforward world-centric dense 3D tracking of all pixels. In *ECCV*, 2026.

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

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*, 2025.

Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguère, and Cyrill Stachniss. ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals. In *IROS*, 2019.

Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Carl Yuheng Ren. Aria Digital Twin: A new benchmark dataset for egocentric 3D machine perception. In *ICCV*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

Shenhan Qian, Ganlin Zhang, Shangzhe Wu, and Daniel Cremers. Flow4R: Unifying 4D reconstruction and tracking with scene flow. In *ECCV*, 2026.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. 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.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012.

Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic Point Maps: A versatile representation for dynamic 3D reconstruction. In *ICCV*, 2025.

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

Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. SMD-Nets: Stereo mixture density networks. In *CVPR*, 2021.

Hengyi Wang and Lourdes Agapito. 3D reconstruction with spatial memory. In *3DV*, 2025.

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

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *CVPR*, 2026a.

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

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IROS*, 2020.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *ICLR*, 2026b.

Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. RGBD objects in the wild: Scaling real-world 3D object learning from RGB-D videos. In *CVPR*, 2024.

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, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. SpatialTrackerV2: 3D point tracking made easy. In *ICCV*, 2025.

Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In *CVPR*, 2025.

Bowei Zhang, Lei Ke, Adam W. Harley, and Katerina Fragkiadaki. TAPIP3D: Tracking any point in persistent 3D geometry. In *NeurIPS*, 2025a.

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*, 2025b.

Songyan Zhang, Yongtao Ge, Jinyuan Tian, Guangkai Xu, Hao Chen, Chen Lv, and Chunhua Shen. POMATO: Marrying pointmap matching with temporal motions for dynamic 3D reconstruction. In *ICCV*, 2025c.

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.

Kaichen Zhou, Yuhan Wang, Grace Chen, Xinhai Chang, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. PAGE-4D: Disentangled pose and geometry estimation for VGGT-4D perception. In *ICLR*, 2026.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4D visual geometry transformer. In *ICLR*, 2026.

A NOTATION

Table 4 summarizes the notation of Section 3.

Table 4: **Summary of notation.** Predictions and derived quantities are written for one view toward one target view.

Symbol	Meaning
Input	
$\{\mathbf{I}^n\}_{n=1}^N$	N input images of a dynamic scene, $\mathbf{I}^n \in \mathbb{R}^{H\times W\times 3}$
Predictions	
$\mathbf{P}$	Local point map, placing each pixel at the view's own viewpoint and time
$\mathbf{C}$	Confidence map, $\mathbf{C} \in (1,\infty)^{H\times W}$, weights geometry and motion regression
$\mathbf{P}_{vt}$	Displaced point map, placing the same pixels at the target's viewpoint and time
$\mathbf{W}$	Rigidity map, $\mathbf{W} \in (0,1)^{H\times W}$ with $\sum_i \mathbf{W}^{(i)} = 1$
Derived	
$\hat{\mathbf{T}}$	Relative camera pose in $SE(3)$, solved from $\mathbf{P}$, $\mathbf{P}_{vt}$, and $\mathbf{W}$ (Eq. (1))
$\mathbf{P}_v$	Point map at the target's viewpoint and the view's own time, $\mathbf{P}_v = \hat{\mathbf{T}}\mathbf{P}$
$\mathbf{P}_t$	Point map at the view's own viewpoint and the target's time, $\mathbf{P}_t = \hat{\mathbf{T}}^{-1}\mathbf{P}_{vt}$
$\mathbf{F}$	Scene flow, $\mathbf{F} = \mathbf{P}_{vt} - \mathbf{P}$
$\mathbf{F}_v$, $\mathbf{F}_t$	Camera-induced displacement and scene-motion residual of $\mathbf{F}$ (Eq. (3))
Training	
$\bar{\mathbf{P}}$, $\bar{\mathbf{P}}_{vt}$, $\bar{\mathbf{T}}$	Ground-truth local point map, displaced point map, and relative pose
$\bar{\mathbf{P}}_v$	Ground-truth point map carried to the target viewpoint, $\bar{\mathbf{P}}_v = \bar{\mathbf{T}}\bar{\mathbf{P}}$
$\mathbf{M}_{\mathbf{P}}$, $\mathbf{M}_{\mathbf{P}_{vt}}$	Masks of pixels with valid ground-truth depth and displacement

B TRAINING DATA

Table 5 lists the training datasets with their annotations and the number of samples drawn from each per epoch.

Table 5: **Training data.** Datasets are grouped by content and domain and listed with the supervision each provides and the number of samples drawn from it per epoch. Depth and camera parameters supervise the local point map $\mathbf{P}$ and, through the ground-truth relative pose, the displaced point map $\mathbf{P}_{vt}$ (Eq. (8)) and the rigidity $\mathbf{W}$ (Eq. (9)); scene flow and 3D point tracks supervise $\mathbf{P}_{vt}$ directly (Eq. (7)).

Dataset	Camera Param.	Depth	Scene flow	3D tracks	Samples/epoch
Static, real					
WildRGB-D (Xia et al., 2024)	✓	✓			5,000
CO3Dv2 (Reizenstein et al., 2021)	✓	✓			5,000
ARKitScenes (Baruch et al., 2021)	✓	✓			5,000
ScanNet (Dai et al., 2017)	✓	✓			5,000
Map-free (Arnold et al., 2022)	✓	✓			5,000
DL3DV (Ling et al., 2024)	✓	✓			5,000
Static, synthetic					
TartanAir (Wang et al., 2020)	✓	✓			5,000
UnrealStereo4K (Tosi et al., 2021)	✓	✓			500
Hypersim (Roberts et al., 2021)	✓	✓			2,000
Dynamic, synthetic					
Virtual KITTI 2 (Cabon et al., 2020)	✓	✓	✓		2,000
Point Odyssey (Zheng et al., 2023)	✓	✓		✓	5,000
Dynamic Replica (Karaev et al., 2023)	✓	✓		✓	5,000
Kubric (Greff et al., 2022)	✓	✓		✓	5,000
Syn4D (Jiang et al., 2026b)	✓	✓		✓	6,000

C Implementation Details

The backbone is the 40-layer ViT-g/14 encoder of 4RC (Luo et al., 2026), initialized from its public weights; the motion blocks and the DPT head are initialized randomly. Within-view and cross-view attention alternate from layer 14 onward, and tokens are read out after layers 20, 28, 34, and 40. Each read-out depth feeds two motion blocks of width 1536, for 1.73 B parameters in total, 0.57 B of them in the motion blocks. We train in a single stage for 100 epochs at a long side of 518 pixels, with aspect ratios from 518×392 to 518×168, random crops, and color jitter. Each step processes 352 views, 44 per GPU on 8 H100 GPUs. We use AdamW (Loshchilov & Hutter, 2019) with $\beta = (0.9, 0.95)$ and weight decay 0.05, a linear warmup over 2 epochs to a peak learning rate of 5×10^{-5}, cosine decay to 10^{-6}, gradient clipping at norm 10, and bfloat16 mixed precision.

D Training Objectives

We supervise every ordered pair the model predicts; the terms below are written for one pair, and each is averaged over the valid pixels of all pairs in a batch. Following DUSt3R (Wang et al., 2024), point maps are normalized to make every term scale-invariant. The left panel of Fig. 3 shows which prediction each term reaches.

Geometry. The local point map is supervised against the back-projected ground-truth depth $\bar{\mathbf{P}}$ with the standard confidence weighting:

$$\mathcal{L}_{\mathbf{P}} = \frac{1}{|\mathbf{M}_{\mathbf{P}}|} \sum_i \mathbf{M}_{\mathbf{P}}^{(i)} \left(\mathbf{C}^{(i)} \left\| \mathbf{P}^{(i)} - \bar{\mathbf{P}}^{(i)} \right\|_2 - \alpha \log \mathbf{C}^{(i)} \right), \tag{6}$$

where $\mathbf{M}_{\mathbf{P}}$ masks pixels with valid depth.

Motion. Ground-truth 3D displacement is scarce, so we convert every available correspondence—scene flow and 3D point tracks—into the displaced point map it implies and supervise $\mathbf{P}_{vt}$ directly:

$$\mathcal{L}_{\mathbf{P}_{vt}} = \frac{1}{|\mathbf{M}_{\mathbf{P}_{vt}}|} \sum_i \mathbf{M}_{\mathbf{P}_{vt}}^{(i)} \left(\mathbf{C}^{(i)} \left\| \mathbf{P}_{vt}^{(i)} - \bar{\mathbf{P}}_{vt}^{(i)} \right\|_2 - \alpha \log \mathbf{C}^{(i)} \right). \tag{7}$$

Both losses measure 3D point errors for the same source pixels, and we use the source confidence map $\mathbf{C}$ to weight both.

Static data supply displacement too. For a static scene, the target displaced point map is $\bar{\mathbf{P}}_v = \bar{\mathbf{T}}\bar{\mathbf{P}}$, obtained by applying the ground-truth relative camera pose $\bar{\mathbf{T}}$ to the local geometry. Every static pair thus becomes motion supervision:

$$\begin{aligned} \mathcal{L}_{\mathbf{P}_v} &= \frac{1}{|\mathbf{M}_{\mathbf{P}}|} \sum_i \mathbf{M}_{\mathbf{P}}^{(i)} \left(w^{(i)} \mathbf{C}^{(i)} \left\| \mathbf{P}_{vt}^{(i)} - \bar{\mathbf{P}}_v^{(i)} \right\|_2 - \alpha \log \mathbf{C}^{(i)} \right), \\ w^{(i)} &= \begin{cases} \mathrm{sg}(\mathbf{W}^{(i)}) \cdot HW & \text{dynamic,} \\ 1 & \text{static.} \end{cases} \end{aligned} \tag{8}$$

On dynamic data, this term would contradict Eq. (7) wherever the scene moves, so the predicted rigidity gates it. The gate is rescaled by HW to undo the normalization of $\mathbf{W}$ and detached by $\mathrm{sg}(\cdot)$ so that it does not interfere with the rigidity term below.

Rigidity. A binary motion mask does not fully specify which points provide a reliable rigid alignment. Static foreground objects can support camera pose estimation, while uncertain geometry can make even stationary points unreliable. We therefore supervise $\mathbf{W}$ through the alignment it produces, backpropagating through Eq. (1):

$$\mathcal{L}_{\mathbf{W}} = \frac{1}{|\mathbf{M}_{\mathbf{P}}|} \sum_i \mathbf{M}_{\mathbf{P}}^{(i)} \left\| \mathbf{P}_v^{(i)} - \bar{\mathbf{P}}_v^{(i)} \right\|_2, \tag{9}$$

where $\mathbf{P}_v = \hat{\mathbf{T}}\mathbf{P}$ is the predicted counterpart of $\bar{\mathbf{P}}_v$. Gradients to $\mathbf{P}$ and $\mathbf{P}_{vt}$ are blocked, as both are already supervised, so this term adjusts only $\mathbf{W}$. This objective learns weights that make the recovered transformation agree with the ground-truth camera motion, without requiring rigidity annotations.

The loss weights and pair sampling follow Section 3.5.

E RUNTIME EFFICIENCY

Method	# par. (B)	views/s ↑	VRAM (GB) ↓
V-DPM	1.66	0.6	32.4
4RC	1.52	23.1	17.5
Track4World	1.72	15.1	15.9
OmniX	1.33	6.6	32.8
SM4RT	1.38	30.6	21.5
Point4D	1.45	1.9	21.3
Any4D	0.61	14.4	30.7
Flow4RT	1.73	28.3	20.6

Table 6: **Runtime efficiency.** One 64-view forward pass at a 518-pixel long side on one H100 GPU, producing both the world-space point maps and the world-space tracks. A method that needs separate passes for the two is timed for both. Times cover the GPU work from each method's preprocessed inputs to its outputs.

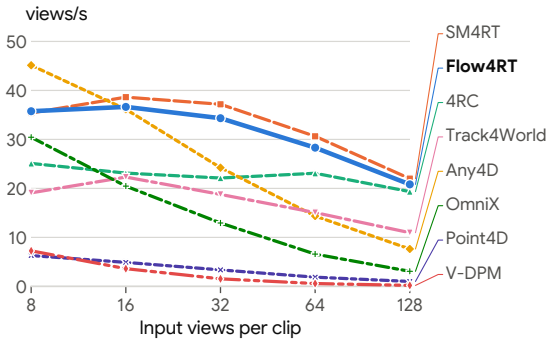


Figure 6: **Throughput against clip length.** Views per second against the number of input views, on a log-scaled view axis. Cost linear in the clip length gives a horizontal line; downward slopes indicate super-linear cost. Values are listed in Table 7.

Table 7: **Throughput against clip length** in views/s, under the protocol of Table 6. The top three entries per column are shaded; Fig. 6 plots these values.

Method	8 views	16 views	32 views	64 views	128 views
V-DPM	7.3	3.6	1.5	0.6	0.2
4RC	25.1	23.1	22.1	23.1	19.3
Track4World	19.1	22.3	18.8	15.1	11.0
OmniX	30.4	20.5	12.9	6.6	3.1
SM4RT	35.4	38.6	37.2	30.6	22.0
Point4D	6.3	4.9	3.4	1.9	1.0
Any4D	45.1	36.2	24.2	14.4	7.6
Flow4RT	35.8	36.6	34.3	28.3	20.8

Cost per clip. Table 6 reports the cost of producing both world-space point maps and world-space tracks for a 64-view clip on one H100 GPU. We time GPU computation from each method's preprocessed inputs to its outputs, including separate passes when required for reconstruction and tracking. Flow4RT reaches 28.3 views/s, second to SM4RT at 30.6 views/s and ahead of 4RC at 23.1 views/s; the remaining methods achieve at most 15.1 views/s. Peak memory is 20.6 GB, below Point4D (21.3 GB), SM4RT (21.5 GB), Any4D (30.7 GB), V-DPM (32.4 GB), and OmniX (32.8 GB), but above Track4World (15.9 GB) and 4RC (17.5 GB).

Scaling with clip length. Fig. 6 and Table 7 report throughput for clips of 8–128 views. All methods encode the full clip jointly. Track4World performs its subsequent flow refinement in overlapping 16-frame windows, matching the configuration used for its accuracy evaluation. In Flow4RT, cross-view attention becomes more expensive as the clip grows. Each motion block attends only between one source and one target view, so at a fixed image resolution the motion branch scales linearly with the number of requested pairs. The bidirectional anchor configuration requires a number of pairs linear in the clip length, with a closed-form alignment for each pair. In our evaluation, V-DPM reruns its time-conditioned decoder over the clip for each query time to produce tracks. Producing both outputs for 64 views takes 110 s for V-DPM and 2.3 s for Flow4RT. Flow4RT reuses the geometry features across requested pairs, avoiding repeated full-clip decoding.

F Additional Results

Section F.1 evaluates video depth and camera pose on standard benchmarks, and Section F.2 extends the qualitative comparison of Section 4.2.

F.1 Video Depth and Camera Pose

We additionally evaluate video depth and camera pose under the MonST3R protocol (Zhang et al., 2025b). Video depth is evaluated on Sintel (Butler et al., 2012), Bonn (Palazzolo et al., 2019), and KITTI (Geiger et al., 2012), and camera pose on ScanNet (Dai et al., 2017) and TUM-Dynamics (Sturm et al., 2012). Depth is aligned to the ground truth by one scale-and-shift fit per sequence over the stacked clip, with maximum depth capped at 70 m on Sintel and Bonn and uncapped on KITTI. Pose is measured with `evo` under a $\mathrm{Sim}(3)$ alignment. On ScanNet, all methods run the same 94 of 100 scenes; the other six have malformed ground-truth pose files. The input long side is 518 pixels for Flow4RT, OmniX, Point4D, V-DPM, 4RC, and SM4RT, and 512 pixels for Track4World. Track4World's correlation pyramid needs a short side of at least 256 pixels, which the 155×512 aspect-preserving resize of KITTI does not meet. For its KITTI entry, we therefore downsample each frame to 155×512 and upsample it to a canvas the model accepts, so the input carries no more image information than at a 512-pixel long side. Flow4RT ranks among the top three on both depth metrics for all three datasets and achieves the lowest errors on all three ScanNet pose metrics (Table 8). On TUM, it ties for third in translational relative pose error but does not lead in absolute trajectory or rotational relative pose error.

Table 8: **Video depth and camera pose evaluation.** δ is $\delta < 1.25$, RPE_t and RPE_r are the translational and rotational relative pose errors at a one-frame offset, and the sequence count of each split is given in the header. All methods are evaluated on identical sequence sets and all pass the whole clip through in one forward pass.

	Video depth						Camera pose					
	Sintel (23)		Bonn (5)		KITTI (13)		ScanNet (94)			TUM (8)		
	Abs Rel ↓	δ ↑	Abs Rel ↓	δ ↑	Abs Rel ↓	δ ↑	ATE ↓	RPE_t ↓	RPE_r ↓	ATE ↓	RPE_t ↓	RPE_r ↓
V-DPM (Sucar et al., 2026)	0.203	0.699	0.047	0.977	0.070	0.938	0.0342	0.0141	0.4041	0.0144	0.0103	0.3132
4RC (Luo et al., 2026)	0.268	0.661	0.048	0.973	0.052	0.959	0.0315	0.0131	0.3848	0.0107	0.0091	0.3129
Track4World (Lu et al., 2026)	0.236	0.659	0.039	0.980	0.032	0.985	0.0347	0.0134	0.3826	0.0121	0.0091	0.3085
OmniX (Jiang et al., 2026a)	0.348	0.731	0.046	0.975	0.075	0.941	0.0345	0.0133	0.4895	0.0123	0.0101	0.3208
SM4RT (Lin et al., 2026b)	0.269	0.661	0.048	0.973	0.052	0.959	0.0346	0.0137	0.5132	0.0129	0.0096	0.3245
Point4D (Jeon et al., 2026)	0.169	0.748	0.045	0.975	0.043	0.979	0.0368	0.0135	0.4879	0.0268	0.0134	0.3940
Flow4RT	0.179	0.750	0.044	0.975	0.040	0.979	0.0280	0.0115	0.3426	0.0136	0.0096	0.3227

F.2 Qualitative Results

Fig. 7 shows five further clips: two from WorldTrack—Point Odyssey and Dynamic Replica, both of which provide dense ground-truth geometry—and three from DAVIS. Gray curves are the ground truth, available in full for the two WorldTrack clips. The DAVIS clips have no ground truth, so their query pixels come from the first-frame object mask and their last frame is left unmarked.

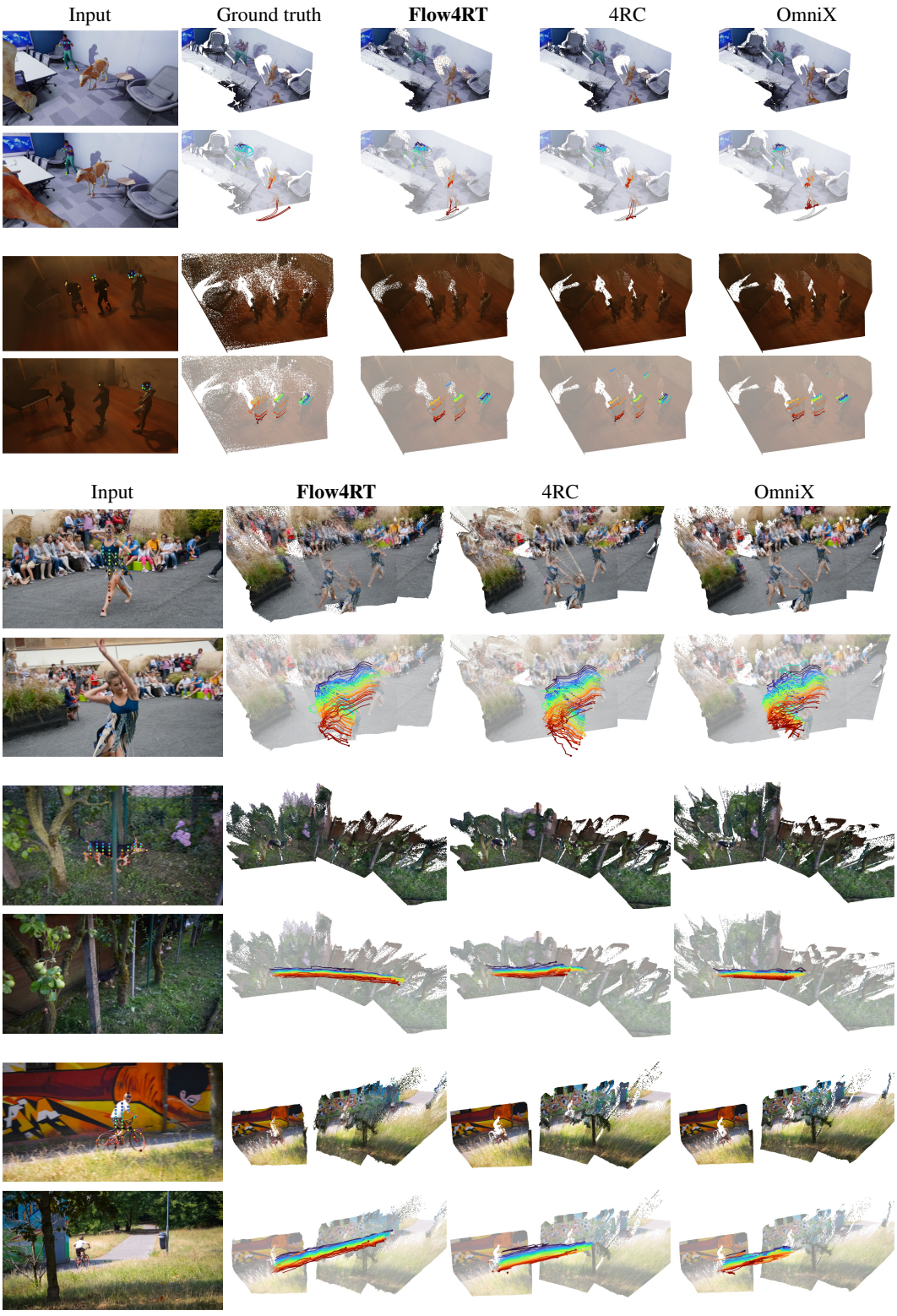


Figure 7: **Additional 4D reconstruction and world-space tracking results.** Layout and protocol follow Fig. 5. Flow4RT keeps tracking through large camera and object motion.