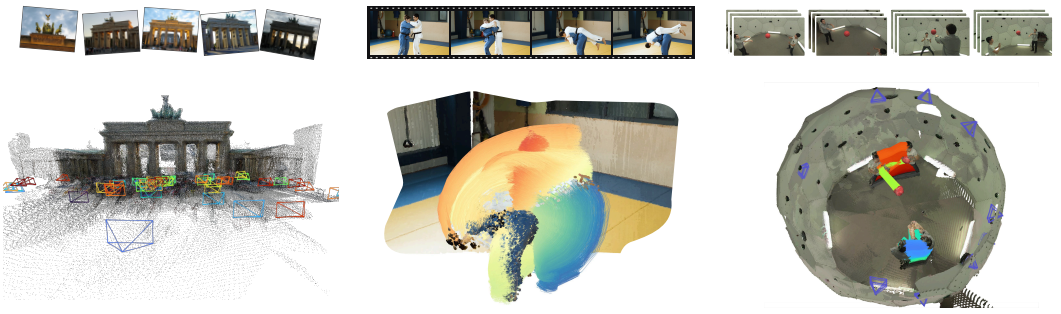
ARROW: ARBITRARY RECONSTRUCTION AND TRACKING OF 4D OBSERVATIONS IN THE WILD

Anonymous authors
Paper under double-blind review



Multi-view 3D Reconstruction | 3D Tracking | Multi-view 3D Tracking

Figure 1: **ARROW** reconstructs 4D scenes over arbitrary sets of input images. It sets a new state of the art in 3D point tracking & multi-view tracking and performs on par in 3D reconstruction.

ABSTRACT

Dynamic scenes may be captured by a moving camera, multiple video streams, or images taken at different times. These observations reveal complementary aspects of scene geometry and motion, yet bringing them together requires establishing correspondence across viewpoints, capture times, and visibility changes. We introduce ARROW, a feed-forward model that unifies 3D reconstruction and 3D point tracking from arbitrary image sets. At its core is a novel order-invariant querying approach, which allows the association of queries with observations across arbitrary inputs. We show that exposing the model to more diverse sets of inputs during training results in improved task performance. Moreover, the resulting model is capable of generalization to a wider range of tasks including multi-view tracking. Trained with this strategy, ARROW establishes a new state of the art in 3D tracking on WorldTrack and TAPVid-3D and outperforms dedicated multi-view trackers on an adapted RGB-only MVTracker benchmark, while remaining competitive across 3D reconstruction tasks. Code and weights will be released.

1 INTRODUCTION

Reconstruction recovers a scene's 3D structure, while tracking maintains the identity of physical points and estimates their positions across observations. Multi-view reconstruction has long considered images captured from diverse viewpoints, including unordered image collections (Schönberger & Frahm, 2016; Schönberger et al., 2016; Wang et al., 2024). Point tracking, in contrast, has primarily focused on temporal correspondence within video (Doersch et al., 2023; Karaev et al., 2025; Xiao et al., 2024). These are closely connected problems: geometry constrains correspondence, and correspondence relates structure across views and time, yet they have long been treated separately. Recent models such as St4RTrack and D4RT demonstrate the power of unifying geometry and tracking within a single model (Feng et al., 2025; Zhang et al., 2026). D4RT, in particular, encodes a video once and answers independent queries specifying a source pixel, target time, and reference camera, allowing reconstruction and tracking to share a single representation space. However, this unification retains the temporal scope of conventional tracking. Training primarily on temporally local video introduces an inductive bias toward motion and appearance continuity. This in turn

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.

narrows the correspondence problems and supervision encountered during learning and potentially limits generalization. Yet scene points can also be matched across unordered viewpoints without temporal continuity (Leroy et al., 2024).

To lift this restriction, we consider *arbitrary observations* of the same scene, differing in viewpoint, capture time, or both, without requiring temporal continuity. We exploit this broader observation space to diversify tracking supervision beyond continuous motion. Observations spanning different cameras and distant capture times expose a wider range of appearances and can pose more challenging correspondence problems. Motivated by positive transfer between related tasks (Caruana, 1997), we aim to support these broader configurations through a shared reconstruction and tracking interface.

To this end, we take inspiration from D4RT's query formulation: each query selects a pixel in a source observation (src) and requests its 3D position at a target observation's capture time (tgt), expressed in a reference observation's coordinate system (ref).

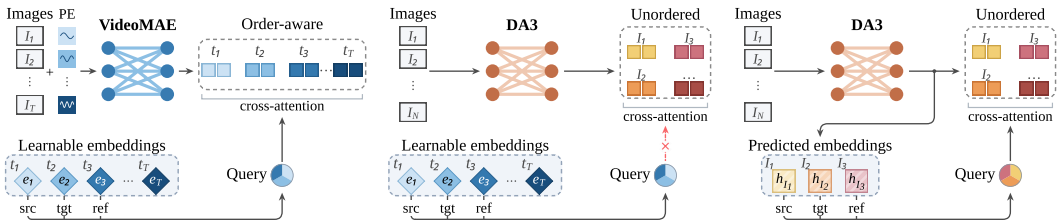
However, extending this query interface to arbitrary observations reveals an architectural limitation. Decoding from a shared representation requires associating queries with the intended observations. D4RT enables this association by explicitly imposing order on the global context via positional encodings (Figure 2a). Such order is natural for monocular video but does not apply to arbitrary observations: multi-view streams may share capture times, and static image collections have no inherent order. Thus, while this query formulation trivially extends to general image collections, D4RT's architecture imposes unwarranted order.

We seek to retain this query flexibility without imposing order. Geometry transformers such as DUSt3R, VGGT, and Depth Anything 3 (DA3) recover geometry from unposed image sets, learning relationships across diverse viewpoints (Wang et al., 2024; 2025a; Lin et al., 2026a). Such a prior is well-suited to relating observations that are not temporally adjacent. Replacing the encoder with an order-free image-set representation, however, removes the sequence-position cues that associate fixed timestep query embeddings with the intended observations (Figure 2b). Adding positional encoding (PE) could reestablish the required association, but maintains the order dependency. Instead, we propose to establish the correspondence between queries and the global context through content-derived identities. We utilize DA3 to generate an order-free global context from the input images and predict per-frame embeddings. Specifically, we repurpose DA3's per-image camera tokens as *identity tokens* (ID tokens) serving as input-derived embeddings unique to each frame. These ID tokens are used to indicate the queried source, target, and reference observations replacing the learned temporal embedding's role. Crucially, these ID tokens are order-invariant, as they are derived from the input images regardless of the frame position. Unlike the learned temporal embeddings, the ID tokens and the global context share the same embedding space and therefore can establish a meaningful order-invariant association (Figure 2c).

Based on this strategy, we introduce ARROW, a unified model for reconstruction and tracking across arbitrary unposed RGB observations. Each query independently attends to the full encoded context, allowing all available observations to inform its prediction. The same decoder supports the desired granularity an application needs, from sparse trajectories for interaction or control to dense geometry and motion for reconstruction. Content-derived identities natively accommodate multiple camera streams and remove the need for a fixed bank of learned timestep identities.

We exploit this architectural change through *non-sequential sampling* across camera views and beyond local temporal neighborhoods. We jointly train on static image collections, dynamic videos, and multi-view sequences. Thus, learning across arbitrary observations broadens supervision as well as the model's supported input settings.

Our cumulative ablations show the benefits of both the architectural change and the broader training strategy (Section 4). Specifically, content-derived addressing improves tracking and reconstruction, while exposing the model to a more challenging training distribution further improves performance. The resulting model sets a new state of the art in 3D tracking on WorldTrack and TAPVid-3D, and achieves competitive performance in 3D reconstruction and camera pose estimation. ARROW also outperforms dedicated multi-view trackers on the adapted RGB-only MVTracker benchmark. These results show that broadening training beyond temporally local observations can improve temporal tracking itself, while supporting correspondence across a wider range of scene observations.



(a) Order-dependent querying (b) Feature-query mismatch (c) Order-invariant querying (ours)

Figure 2: **From indexed to input-dependent correspondence.** (a) Order-aware global context enables position-based associations. (b) Order-invariant global context exposes no fixed order to exploit. (c) Content-derived identity embeddings enable order-invariant associations.

We summarize our contributions as follows:

- We introduce ARROW (Section 3), an open-source model trained on publicly available datasets, setting the state of the art on multiple benchmarks (Section 4).
- At its core is our proposed order-invariant querying approach (Figure 2c), aligning the embedding space of the queries and the global context.
- We demonstrate the benefits of diversified tracking supervision through extensive ablations (Section 4).

2 Related Work

3D geometry estimation. Classical structure-from-motion and multi-view stereo recover cameras and the geometry of a fixed world through feature matching, triangulation, and optimization (Schönberger & Frahm, 2016; Schönberger et al., 2016). Feed-forward models replaced this pipeline with direct point-map regression from unposed images (Wang et al., 2024; Leroy et al., 2024), and subsequent geometry transformers extended prediction from pairs to image sets (Wang et al., 2025a; Yang et al., 2025; Wang et al., 2026; Keetha et al., 2026; Lin et al., 2026a). Applying them to moving scenes requires distinguishing camera motion from changes in scene content, addressed through dynamic-scene fine-tuning, joint depth and pose optimization, or motion-aware masking (Zhang et al., 2025b; Li et al., 2025; Hu et al., 2025; Zhou et al., 2026). Recovering geometry at each observation, however, does not establish how the same physical objects move between observations.

3D point tracking. A parallel line of work pursues this correspondence directly. Image-plane trackers follow point identities through video (Doersch et al., 2023; Karaev et al., 2025) but cannot resolve depth or separate camera from scene motion, motivating trackers that consume depth or camera poses from an external estimator (Xiao et al., 2024; Ngo et al., 2025b;a; Zhang et al., 2025a).

Feed-forward models remove this dependency by jointly predicting geometry and motion or correspondence, either from image pairs (Sucar et al., 2025; Feng et al., 2025; Qian et al., 2026) or over a video (Xiao et al., 2025; Lu et al., 2026). Other works represent scene motion through compact parameterizations. Shape of Motion (Wang et al., 2025b) optimizes shared motion bases per scene, while Trace Anything (Liu et al., 2025) and SM4RT (Lin et al., 2026b) directly predict per-pixel B-spline trajectories and shared motion bases, respectively. These representations support temporal interpolation but model motion across a video timeline, irrespective of the requested point states.

Multi-view tracking. The trackers above follow points through a single camera stream. Multi-view trackers instead combine simultaneous views to reduce depth ambiguity and recover points occluded in individual views, but assume supplied camera calibration and synchronized videos (Rajič et al., 2025; Koo et al., 2026; Galoaa et al., 2026). Without calibrated capture, they therefore rely on an external reconstructor for cameras and, in MVTracker's case, depth (Table 3). Recently OmniX (Jiang et al., 2026a) seeks to jointly recover geometry and dense 3D trajectories from multiple videos and image–video combinations in a feed-forward pass, but it retains explicit timestamp, video-index, and local-frame encodings and does not expose sparse point queries. The recent TAPVid-MV benchmark reflects growing interest in this setting, yet finds that dedicated multi-view trackers do not

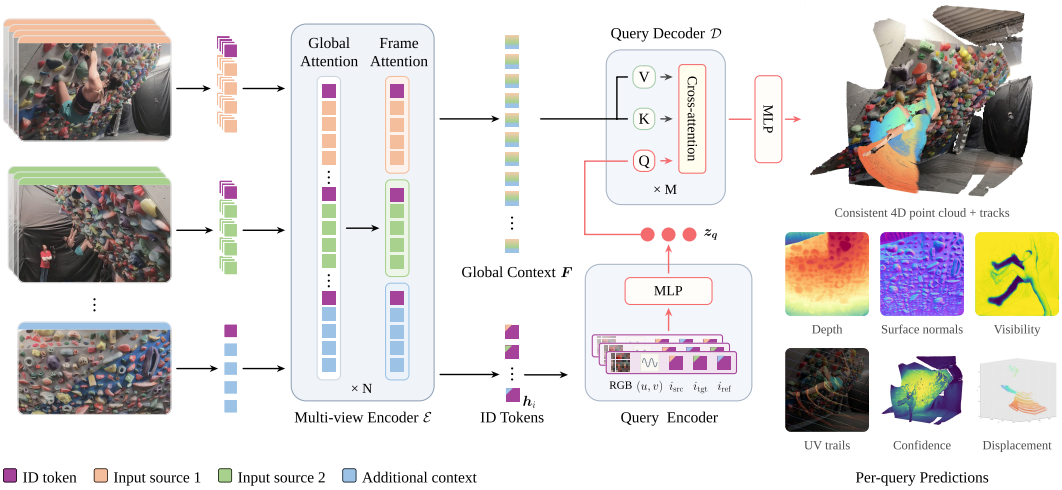


Figure 3: **ARROW architecture.** The multi-view encoder $\mathcal{E}$ tokenizes each frame independently and employs global attention to produce a shared global context $\boldsymbol{F}$ together with an ID token $\boldsymbol{h}_i$ for each frame. The query encoder combines a source-centered RGB crop, the Fourier-encoded (u, v) coordinate, and the ID tokens of the queried source, target, and reference view frames into a latent query token $\boldsymbol{z_q}$. Each latent query token independently cross-attends to the global context, so a single encoding serves arbitrarily many queries and yields a consistent 4D representation of the scene.

consistently outperform monocular methods, highlighting substantial room to better exploit additional views (Koppula et al., 2026). ARROW seamlessly handles dense and sparse multi-view tracking without external geometric inputs thanks to its content-derived query mechanism.

Conditional and query-based tracking. Sparse decoding offers finer control by predicting only the requested point states from a shared scene representation. L4P (Badki et al., 2026) shares a video backbone across geometry and tracking but retains task-specific output heads, whereas D4RT (Zhang et al., 2026) employs cross-attention to independently decode queries specified by a source pixel, a target time, and a reference camera. Its input side, however, remains video-oriented: a video-pretrained encoder, training on temporally sampled clips, and query identities drawn from learned timestep embeddings.

A growing body of recent and concurrent work adapts geometry transformers to 4D reconstruction and tracking, reporting promising results (Karhade et al., 2026; Sucar et al., 2026; Luo et al., 2026; Huang et al., 2026; Chen et al., 2026; Jeon et al., 2026; Jiang et al., 2026a). We discuss these methods in detail in Appendix A. In short, each of these approaches recovers part of D4RT's decoding flexibility, yet none combines *(i)* a single model accepting static image collections, dynamic videos, and multi-view sequences, *(ii)* sparse decoding with independently selectable source, target, and reference, and *(iii)* flexible representation free of explicit temporal or camera indexing. ARROW realizes all three with one shared encoder and a pointwise decoder.

3 Method

We consider an unordered set of RGB images $\mathcal{I} = \{I_i\}_{i=1}^{S}$ constituting observations of a 4D scene. Each observation $I_i \in \mathbb{R}^{H \times W \times 3}$ captures the scene from a particular viewpoint at a particular time. Unlike a frame index in a video, i only identifies an observation and implies neither an ordering nor temporal separation. Distinct observations may share the same capture time.

Query formulation. Inspired by the query interface of D4RT (Zhang et al., 2026), we define a query as a tuple of (u, v) coordinates and indices specifying source, target, and reference observations:

$$\boldsymbol{q} = (u, v, i_{\text{src}}, i_{\text{tgt}}, i_{\text{ref}}), \qquad (u, v) \in [0, 1]^2, \qquad i_{\text{src}}, i_{\text{tgt}}, i_{\text{ref}} \in \{1, \ldots, S\}. \tag{1}$$

The pixel coordinates (u, v) and the source observation $I_{i_{\text{src}}}$ describe a physical point at the capture time of $I_{i_{\text{src}}}$. The desired output is the 3D position of that point at the time specified by the target observation $I_{i_{\text{tgt}}}$ and in the coordinate system specified by the reference observation $I_{i_{\text{ref}}}$. This

formulation enables the inference of a diverse set of outputs such as tracks, point clouds, and camera parameters Zhang et al. (2026). Compared to D4RT, we explicitly support arbitrary input image sets.

3.1 ARROW ARCHITECTURE

Figure 3 provides an overview of ARROW, which consists of a multi-view encoder, a query encoder, and a sparse query decoder. The multi-view encoder receives an image set and outputs a global scene representation and a set of ID tokens. The query encoder translates a query's (u, v) coordinates and observation indices into a latent query token, using the ID tokens to encode observation identity. The query decoder then predicts the per-query outputs via cross-attention between the query token and the global scene representation. The network is trained end-to-end via per-query losses, without explicit supervision of observation identity.

Multi-view encoder. Our multi-view encoder $\mathcal{E}$ is initialized from DA3 (Lin et al., 2026a), a ViT (Dosovitskiy et al., 2021) that alternates frame-wise and global self-attention and is pretrained on multi-view image sets to predict per-frame point maps and camera poses. We repurpose the camera tokens as ID tokens allowing the decoder to match the latent query token to the patch tokens of the source, target, and reference observations. The ID tokens are neither interpreted as camera estimates nor tied to the later pose-recovery procedure.

Formally, the multi-view encoder $\mathcal{E}$ produces a global context $\boldsymbol{F}$ and one ID token $\boldsymbol{h}_i$ per observation:

$$\left(\boldsymbol{F}, \{\boldsymbol{h}_i\}_{i=1}^{S}\right) = \mathcal{E}(\mathcal{I}), \qquad \boldsymbol{F} \in \mathbb{R}^{N \times C}, \quad \boldsymbol{h}_i \in \mathbb{R}^{C}, \tag{2}$$

where N is the total number of patch tokens across the image set and C is the feature dimension. We add no temporal encoding to the encoder or context, since the input is treated as an order-agnostic set.

Query encoder. The query encoder combines local appearance, (u, v) coordinates, and the ID tokens of the queried observations into a latent query token $\boldsymbol{z_q} \in \mathbb{R}^C$:

$$\boldsymbol{z_q} = \phi_{\text{rgb}}(I_{i_{\text{src}}}, u, v) + \phi_{\text{uv}}(u, v) + \phi_{\text{rel}}\left([\boldsymbol{h}_{i_{\text{src}}}; \boldsymbol{h}_{i_{\text{tgt}}}; \boldsymbol{h}_{i_{\text{ref}}}]\right). \tag{3}$$

Here, ϕ_{rgb} embeds a small source-centered RGB crop, ϕ_{uv} embeds the Fourier-encoded normalized (u, v) coordinates, and ϕ_{rel} projects the concatenation of the ID tokens to the feature dimension of the decoder. All three ϕ are implemented as two-layer MLPs.

Sparse decoder. Each latent query token independently cross-attends to $\boldsymbol{F}$, with no self-attention between query tokens. The lightweight cross-attention decoder $\mathcal{D}$ predicts:

$$\hat{\boldsymbol{y}}_{\boldsymbol{q}} = \mathcal{D}(\boldsymbol{z_q}, \boldsymbol{F}) = \left(\hat{\boldsymbol{x}}_{i_{\text{tgt}}}^{i_{\text{ref}}}, \hat{c}, (\hat{u}_{i_{\text{tgt}}}, \hat{v}_{i_{\text{tgt}}}), \hat{o}_{i_{\text{tgt}}}, \hat{\boldsymbol{n}}_{i_{\text{tgt}}}^{i_{\text{ref}}}, \hat{\boldsymbol{d}}_{i_{\text{src}} \to i_{\text{tgt}}}^{i_{\text{ref}}}\right). \tag{4}$$

A superscript denotes the reference frame in which a 3D quantity is expressed and a subscript the target time; indices are suppressed in the text for brevity. The query decoder outputs a 3D position $\hat{\boldsymbol{x}} \in \mathbb{R}^3$, a confidence $\hat{c} \in \mathbb{R}_{>1}$, image coordinates $(\hat{u}, \hat{v}) \in \mathbb{R}^2$, a visibility logit $\hat{o} \in \mathbb{R}$, a unit surface normal $\hat{\boldsymbol{n}} \in \mathbb{R}^3$, and the source-to-target displacement $\hat{\boldsymbol{d}} \in \mathbb{R}^3$. All quantities are predicted at the time of capture of the target observation. The 3D position, surface normals, and displacement are predicted in the coordinate system of the reference observation. The point coordinates and displacement are normalized as described in Appendix B.1. UV predictions are unconstrained in normalized image coordinates, and the visibility probability is $\sigma(\hat{o})$. We recover camera poses from the predicted 3D points, as described in Appendix C.3. Since queries are independent from each other, the encoded context can be reused for sparse queries and dense decoding.

3.2 TRAINING

Image and query sampling. We use three complementary data regimes. For *static scenes*, known geometry provides trivial correspondence: the physical 3D point remains fixed while its appearance and camera coordinates change. For *dynamic scenes with tracks*, annotated tracks provide cross-time correspondence and motion supervision, while static background provides a crucial anchor for the scene. We sample both static and dynamic regions and choose observations from a candidate temporal window up to ten times larger than the requested set, with multi-view datasets potentially also contributing different cameras from the same scene. Finally, for *dynamic scenes without tracks*,

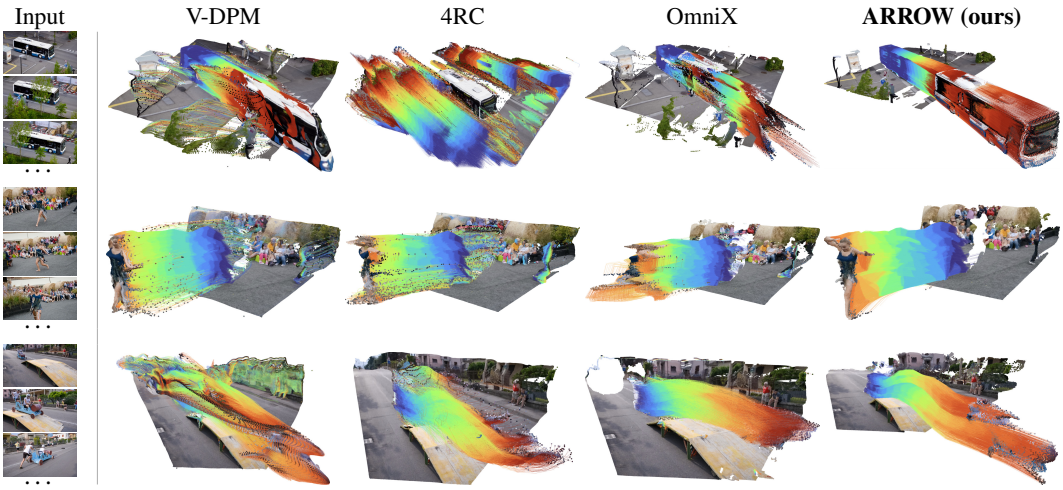


Figure 4: **4D reconstruction and tracking.** V-DPM, 4RC, OmniX, and ARROW on three DAVIS sequences. Input frames are shown on the left; each method panel shows its 3D reconstruction with trajectory overlays. ARROW shows clean reconstructions and consistent long-term tracks.

such as nuScenes (Caesar et al., 2020), we restrict queries to the same target and source time step ($i_{\text{tgt}} = i_{\text{src}}$), avoiding incorrect motion supervision while retaining geometric supervision. Dataset-level image policies and point-level edge, boundary, and motion sampling are detailed in Appendix B.3.

Losses. We apply a multi-task objective and apply each term only where its target is valid:

$$\mathcal{L} = \lambda_{\text{3D}}\mathcal{L}_{\text{3D}} + \lambda_{\text{conf}}\mathcal{L}_{\text{conf}} + \lambda_{\text{uv}}\mathcal{L}_{\text{uv}} + \lambda_{\text{vis}}\mathcal{L}_{\text{vis}} + \lambda_{\text{normal}}\mathcal{L}_{\text{normal}} + \lambda_{\text{disp}}\mathcal{L}_{\text{disp}}. \tag{5}$$

The primary term is a scale-invariant, confidence-weighted ℓ_1 loss after independently normalizing predicted and target point sets by their mean distance to the origin of their corresponding reference camera and applying a signed logarithm. Auxiliary terms use ℓ_1 for the target UV, binary cross-entropy for visibility, cosine distance for normals, and ℓ_1 for displacement. Every source, target, and reference combination is thus supervised by the same objective, without a dedicated tracking loss or a shared world frame. Exact formulation and normalization are further detailed in Appendix B.1.

4 Experiments

We train for 200k steps using DA3-giant, with ablations using 100k steps, DA3-Large, and six decoder layers (Appendix B.1). We evaluate single- and multi-view 3D tracking, camera pose, reconstruction, and video depth. Figure 4 shows qualitative 4D reconstruction and tracking results, while Figure 5 shows point correspondences across challenging image pairs.

Monocular 4D reconstruction and tracking. WorldTrack (Feng et al., 2025) benchmarks 3D point tracking over 64-frame sequences from Aria Digital Twin (ADT) (Pan et al., 2023), Dynamic Replica (Karaev et al., 2023), PointOdyssey Zheng et al. (2023), and Panoptic Studio (Joo et al., 2015), with queries initialized in the first frame. We express trajectories in the first camera's coordinate system and apply global median scale alignment per scene. Appendix C.1 details the evaluation thresholds and dynamic-track selection. ARROW achieves the highest APD and lowest EPE across all four datasets, for both overall and dynamic-only tracking (Table 1).

TAPVid-3D (Koppula et al., 2024) evaluates bidirectional tracking from arbitrary query frames over sequences of up to 300 frames taken from DriveTrack (Balasingam et al., 2024), ADT, and Panoptic Studio. Long sequences and arbitrary source frames require windowed methods to propagate tracks bidirectionally and align them across windows. ARROW instead queries every target against the full encoded sequence, choosing the target camera or first camera as reference (cf. Appendix D.2).

Following D4RT (Zhang et al., 2026), we evaluate on the mini-val split without ground-truth intrinsics in camera coordinates and a shared world frame (the first input camera). Both settings use the same per-sequence median-scale alignment formula as WorldTrack. We report APD, average Jaccard (AJ)

Table 1: **WorldTrack 3D point tracking.** We report the average percentage of points within distance thresholds (APD, %) and mean 3D end-point error (EPE, meters). All and Dyn. denote overall and dynamic-only scores. Results use global median scale alignment. ARROW sets a new state of the art.

Method	Aria Digital Twin				Dynamic Replica				PointOdyssey				Panoptic Studio			
	APD ↑		EPE ↓		APD ↑		EPE ↓		APD ↑		EPE ↓		APD ↑		EPE ↓	
	All	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.	All	Dyn.
SpatialTracker-V2	91.75	83.18	0.0964	0.0832	69.90	67.18	0.2986	0.3348	67.36	68.03	0.4299	0.4789	73.42	73.42	0.2397	0.2397
St4RTrack	72.89	77.98	0.2958	0.1087	74.29	76.85	0.2600	0.1973	67.45	69.89	0.3144	0.2722	70.79	70.79	0.2480	0.2480
TraceAnything	75.71	76.07	0.2517	0.1875	60.64	60.58	0.5751	0.4730	39.51	43.70	1.0546	0.8906	71.17	71.17	0.2788	0.2788
V-DPM	85.35	73.27	0.1763	0.1455	77.28	80.13	0.2306	0.1904	80.47	82.59	0.1886	0.1672	77.11	77.11	0.1881	0.1881
Any4D	57.88	69.59	0.4149	0.1420	79.45	69.74	0.2140	0.2976	66.87	67.86	0.3498	0.3574	63.98	63.98	0.2978	0.2978
4RC	83.19	71.83	0.1864	0.1427	84.41	76.27	0.1750	0.2515	83.15	83.83	0.2175	0.2351	68.19	68.19	0.2657	0.2657
OpenD4RT	71.92	72.68	0.2814	0.1627	76.59	82.18	0.2438	0.1754	67.61	70.09	0.3220	0.2969	78.98	78.98	0.1794	0.1794
Track4World	79.84	88.48	0.2173	0.0571	71.49	70.24	0.2709	0.2740	66.78	68.19	0.3505	0.3151	73.16	73.16	0.2265	0.2265
OmniX	91.05	86.58	0.1029	0.1108	79.91	72.64	0.2138	0.3071	78.68	78.47	0.2545	0.2779	77.23	77.23	0.1994	0.1994
SM4RT*	86.86	80.55	0.1515	0.0957	82.77	73.70	0.1848	0.2654	81.05	80.69	0.2345	0.2614	78.76	78.76	0.1788	0.1788
Point4D*	62.73	77.38	0.4462	0.1578	77.77	70.72	0.2356	0.3004	69.84	72.55	0.3604	0.3580	72.92	72.92	0.2354	0.2354
UniQuery4R*	87.18	90.11	0.1387	0.0501	79.25	81.95	0.2146	0.1633	83.84	85.47	0.1607	0.1510	84.76	84.76	0.1339	0.1339
ARROW	**93.55**	**90.63**	**0.0818**	**0.0481**	**88.00**	**91.47**	**0.1426**	**0.0896**	**85.26**	**86.72**	**0.1532**	**0.1417**	**85.76**	**85.76**	**0.1332**	**0.1332**

* Concurrent work first released on arXiv between July and September 2026.

Table 2: **TAPVid-3D evaluation in camera and world coordinates.** Without ground-truth intrinsics. L1 is measured after normalizing predictions and ground truth by their median distances to the reference camera. Baseline results are taken from D4RT (Zhang et al., 2026). ARROW achieves the best results in average AJ, APD, and normalized L1.

Method	Camera-coordinate 3D tracking												World-coordinate 3D tracking					
	DriveTrack			Aria Digital Twin			Panoptic Studio			Avg.			DriveTrack		ADT		Avg.	
	AJ ↑	APD ↑	OA ↑	AJ ↑	APD ↑	OA ↑	AJ ↑	APD ↑	OA ↑	AJ ↑	APD ↑	OA ↑	APD ↑	L1 ↓	APD ↑	L1 ↓	APD ↑	L1 ↓
St4RTrack	—	—	—	—	—	—	—	—	—	—	—	—	0.020	0.499	0.062	0.839	0.041	0.669
CoTracker3 + *UD2*	0.038	0.062	0.856	0.102	0.146	**0.937**	0.119	0.176	0.862	0.086	0.128	0.885	—	—	—	—	—	—
CoTracker3 + *VGGT*	0.129	0.189	0.856	0.132	0.190	**0.937**	0.045	0.070	0.862	0.102	0.150	0.885	0.205	0.170	0.192	0.736	0.199	0.453
SpatialTrackerV2	0.064	0.100	0.865	0.260	0.342	0.936	0.097	0.152	0.805	0.140	0.198	0.869	0.101	0.139	0.336	0.238	0.219	0.189
D4RT	**0.257**	0.345	**0.875**	0.240	0.319	0.926	0.138	0.186	**0.897**	0.212	0.283	**0.899**	**0.373**	**0.020**	0.319	0.096	0.346	0.058
ARROW	0.241	**0.350**	0.816	**0.360**	**0.473**	0.902	**0.141**	**0.201**	0.870	**0.248**	**0.341**	0.863	0.327	0.027	**0.442**	**0.054**	**0.385**	**0.041**

and occlusion accuracy (OA) in Table 2. ARROW achieves the best average camera-space AJ and APD, and world-space APD and L1, among the compared methods. It remains competitive in OA, slightly trailing the baselines. We observe visibility errors of a few pixels and hypothesize that training across varying aspect ratios might require longer optimization for precise UV localization.

Multi-view 4D reconstruction and tracking. We adapt the MVTracker benchmark (Rajič et al., 2025) to an RGB-only input setting for 3D tracking on Multi-View Kubric, DexYCB (Chao et al., 2021), and Panoptic Studio. The original protocol supplies depth, camera calibration, and 3D queries; our adaptation uses RGB images from four static cameras over 24 synchronized timesteps and 2D queries in visible source views. This requires joint scene reconstruction and tracking throughout the clip, before and after each query's source time. Trajectories are expressed in the first input camera's coordinate system and aligned by a single scale factor per scene. Table 3 reports APD and EPE on ground-truth-visible positions, alongside AJ and OA.

ARROW leads on 11 of the 12 dataset–metric combinations, with DexYCB EPE as the sole exception. MVTracker shows similarly elevated EPE with DA3 but not with VGGT or VGGT-Ω, suggesting a mismatch with DA3's pretraining distribution, and illustrates that the prior remains influential for tracking accuracy. Nevertheless, ARROW achieves the highest DexYCB APD and AJ, which are less sensitive than EPE to the magnitude of large errors.

3D reconstruction. Beyond tracking, we evaluate camera pose estimation and scene reconstruction. For pose estimation, we evaluate on Bonn (Palazzolo et al., 2019), Sintel (Butler et al., 2012), and Panoptic Studio; for reconstruction, we evaluate on NRGBD (Azinović et al., 2022) and 7-Scenes Shotton et al. (2013). We recover camera poses by aligning a sparse set of predicted 3D points

Table 3: **RGB-only multi-view 3D tracking.** We report APD and EPE on ground-truth-visible positions, as well as OA and AJ. + denotes methods requiring an external reconstruction model.

Method	Panoptic Studio				DexYCB				Multi-View Kubric				Avg.			
	APD ↑	EPE ↓	OA ↑	AJ ↑	APD ↑	EPE ↓	OA ↑	AJ ↑	APD ↑	EPE ↓	OA ↑	AJ ↑	APD ↑	EPE ↓	OA ↑	AJ ↑
MVTracker + *VGGT*	54.6	0.2369	69.5	37.0	39.7	0.1063	82.6	30.0	31.7	0.3070	72.4	21.2	42.0	0.2167	74.8	29.4
MVTracker + *DA3*	59.9	0.2054	69.1	39.1	32.4	0.1732	80.8	23.0	43.6	0.1595	76.4	30.4	45.3	0.1793	75.4	30.8
MVTracker + *VGGT*-Ω*	49.1	0.2923	63.6	28.5	49.3	0.1057	78.3	39.9	28.4	0.2279	72.5	18.0	42.3	0.2087	71.5	28.8
LAPA + *VGGT*-Ω*	41.4	0.3668	85.8	31.0	31.3	0.1543	85.2	24.4	13.6	0.3954	51.6	7.0	28.8	0.3055	74.2	20.8
TapIP3D + *VGGT*-Ω*	50.1	0.2835	87.6	38.8	50.2	**0.0991**	88.6	38.6	32.4	0.2002	83.5	21.8	44.3	0.1943	86.6	33.1
4RC†	56.9	0.1973	—	—	42.9	0.1151	—	—	32.1	0.2333	—	—	44.0	0.1819	—	—
OmniX†	68.4	0.1417	—	—	45.6	0.1215	—	—	30.0	0.2470	—	—	48.0	0.1701	—	—
ARROW	**79.2**	**0.0929**	**89.9**	**66.9**	**54.8**	0.1705	**89.4**	**44.2**	**49.0**	**0.1121**	**86.2**	**36.5**	**61.0**	**0.1252**	**88.5**	**49.2**

* August 2026 checkpoint. † No meaningful visibility prediction; AJ/OA are unavailable.

Table 4: **Pose estimation and 3D reconstruction.** We compare against dedicated 3D reconstruction models (top) and 4D reconstruction and tracking approaches (bottom). Although ARROW is not solely trained for pose prediction or 3D reconstruction, it still achieves competitive results.

Method	Camera pose									3D reconstruction			
	Bonn			Sintel			Panoptic Studio†			NRGBD		7-Scenes	
	ATE ↓	RPE-T ↓	RPE-R ↓	ATE ↓	RPE-T ↓	RPE-R ↓	ATE ↓	RPE-T ↓	RPE-R ↓	Chamfer ↓	NC ↑	Chamfer ↓	NC ↑
VGGT	0.0151	0.0066	0.223	0.1657	0.0512	0.484	0.4310	0.9849	16.914	0.0429	0.910	0.0538	0.754
π^3	0.0114	0.0063	0.225	0.0735	0.0417	0.263	0.1506	0.3755	5.663	0.0300	0.910	0.0721	0.736
VGGT-Ω*	**0.0081**	0.0041	0.213	0.0777	**0.0305**	**0.195**	0.3195	0.6475	8.947	0.0354	0.895	**0.0411**	0.754
DA3	0.0088	0.0047	0.212	0.1287	0.0531	0.357	0.2084	0.5030	6.037	0.0287	0.927	0.0598	0.752
V-DPM	0.0118	0.0064	0.219	0.1284	0.0544	0.461	0.2146	0.3906	3.797	0.0397	0.887	0.0739	0.730
Any4D	0.0277	0.0155	0.472	0.2228	0.1132	2.120	0.5788	1.3593	30.033	0.0691	0.845	0.0969	0.696
4RC	**0.0081**	0.0037	**0.211**	0.1292	0.0527	0.396	0.1414	0.2785	1.942	0.0374	0.911	0.0616	0.754
OpenD4RT	0.0289	0.0062	0.331	0.2328	0.1256	1.833	1.5576	2.8762	78.012	0.4077	0.592	0.2503	0.556
OmniX	0.0084	0.0040	0.214	0.0770	0.0352	0.291	0.0786	0.1459	1.824	0.0294	0.927	0.0481	**0.769**
SM4RT	0.0168	0.0075	0.240	0.1120	0.0445	0.422	0.1483	0.3366	3.281	0.1082	0.776	0.0831	0.691
ARROW	0.0084	**0.0036**	**0.211**	**0.0613**	0.0337	0.453	**0.0714**	**0.1301**	**1.406**	**0.0263**	**0.933**	0.0487	0.762

* August 2026 checkpoint. † Multi-camera setup capturing dynamic scenes simultaneously from multiple viewpoints.

across coordinate frames, without a dedicated pose head (Appendix C.3); pose accuracy thus also reflects the consistency of the predicted geometry. Results are shown in Table 4.

ARROW achieves the lowest ATE on Sintel and leads across all three pose metrics on Panoptic Studio, followed by OmniX. Both methods train with multi-view dynamic observations, suggesting that such training benefits multi-camera pose recovery in dynamic scenes. On Bonn, ARROW achieves the lowest relative translation error and ties for the lowest relative rotation error.

For scene reconstruction, ARROW achieves the lowest Chamfer distance and highest normal consistency on NRGBD and remains competitive on 7-Scenes. Video depth results (Table 5) further demonstrate its geometric accuracy: ARROW leads on both Bonn metrics, achieves the lowest AbsRel on KITTI, and attains the highest δ_1 on 7-Scenes. Figure 10 in the appendix provides a qualitative comparison of video depth predictions.

Cumulative Ablations. We evaluate cumulative progression from a D4RT-inspired baseline to the ARROW design across tracking, pose estimation, and reconstruction (Table 6). For brevity and concise comparisons, we average results across datasets within each task and input setting: monocular tracking uses WorldTrack, monocular pose uses Bonn and Sintel, and monocular reconstruction uses NRGBD. Multi-view results average Multi-View Kubric, DexYCB, and Panoptic Studio.

The baseline (**S1**) adopts D4RT's ordered-video formulation, replacing VideoMAE with our pretrained 3D backbone, i.e., a direct backbone swap. Sinusoidal sequence-position embeddings provide ordering information to the encoder's camera tokens, while learned discrete embeddings identify each query's source, target, and reference images. For inputs exceeding the 24-image training limit, S1 and S2 use either windowed inference (Win; anchored image-block pairs for multi-view tracking) or full-context inference with temporal encodings interpolated over the training range (Int, †). S3, S4, and S5 process the full context directly (Full). Reconstruction uses at most 24 images,

Table 5: **Video depth.** We report AbsRel and δ_1 (%) after per-sequence scale-only, depth-relative L1 alignment.

Method	Bonn		KITTI		7-Scenes	
	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
VGGT	0.053	0.975	0.056	0.961	0.212	0.896
π^3	0.042	0.986	0.037	0.986	0.219	0.876
VGGT-Ω*	0.115	0.937	0.038	**0.988**	0.210	0.910
DA3	0.042	0.982	0.051	0.976	0.209	0.919
V-DPM	0.049	0.985	0.068	0.943	0.210	0.921
Any4D	0.052	0.979	0.066	0.966	0.229	0.866
4RC	0.043	0.983	0.053	0.953	**0.204**	0.903
OpenD4RT	0.071	0.952	0.112	0.854	0.223	0.913
OmniX	0.042	0.984	0.079	0.929	0.214	0.870
SM4RT	0.051	0.979	0.037	0.977	0.211	0.904
ARROW	**0.041**	**0.987**	**0.036**	0.983	0.213	**0.924**

* August 2026 checkpoint.

Figure 5: **Point tracking.** Lines connect source queries to predictions on challenging image pairs from Nordström et al. (2026).

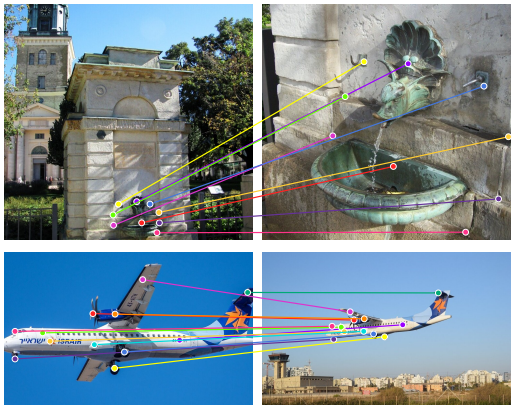


Table 6: **Cumulative ablations.** S5 corresponds to our training recipe and reports absolute scores; other rows report differences from S5. : order-dependent querying (D4RT); : order-invariant querying (ARROW). "video" means only the 15 dynamic/video datasets in Table 7.

Stage	Query Approach	PE	Sequential Sampling	Data	Eval	Tracking				Reconstruction			
						Monocular		Multi-view		Monocular		Multi-view	
						APD ↑	EPE ↓	APD ↑	EPE ↓	AUC@30 ↑	F1 ↑	AUC@30 ↑	F1 ↑
S1		✓	✓	video	windowed	−8.41	+0.0822	−34.23	+0.4416	−19.32	−31.36	−61.61	−26.72
		✓	✓	video	interpolated†	−3.25	+0.0322	−38.68	+0.6257	−7.62	−31.36	−62.63	−26.72
S2		✓	✓	video	windowed	−7.06	+0.0651	−30.37	+0.3995	−21.14	−17.86	−59.81	−26.53
		✓	✓	video	interpolated†	−2.18	+0.0213	−27.23	+0.3567	−8.71	−17.86	−56.94	−26.53
S3		✗	✓	video	full	−2.00	+0.0217	−28.89	+0.4142	−7.55	−18.97	−38.41	−20.33
S4		✗	✗	video	full	−1.52	+0.0144	−0.04	+0.0108	−0.91	−1.16	−0.88	−5.03
S5		✗	✗	all	full	**80.01**	**0.1983**	**48.49**	**0.1526**	**51.10**	**76.59**	**63.61**	**52.02**

† Indicates repeated evaluation with full context and interpolated temporal encodings.

so paired Win/Int rows share reconstruction scores. Replacing discrete query embeddings with identity-based conditioning (**S2**) improves monocular and multi-view tracking under both inference modes. The larger multi-view gain with interpolation supports content-derived observation identities for querying beyond the training context length. Removing the encoder's temporal embeddings (**S3**) leaves monocular tracking nearly unchanged relative to S2 with interpolation, which also uses full context. Effects on other tasks are mixed; overall, this design remains competitive without explicit temporal encoding. Non-sequential sampling (**S4**) introduces cross-view and non-contiguous training observations and yields the largest multi-view tracking gain: 28.85 percentage points in APD relative to S3. Monocular tracking, pose estimation, and reconstruction also improve. Adding four static multi-view datasets (**S5**) expands the mixture from 15 to 19 datasets and improves every reported metric, supporting broader training data for both monocular and multi-view inputs.

Additional design comparisons and 24-frame results appear in Appendix D.1; Appendix D.2 compares full-context and windowed inference on longer sequences.

5 Conclusion

In this work, we introduced ARROW, a feed-forward model capable of reconstructing and tracking across arbitrary observations. This capability is enabled by our introduced order-invariant query mechanism which relies on self-derived identity tokens, replacing order-dependent learnable embeddings. Our ablations show that broadening training beyond monocular video, together with the proposed query mechanism, improves generalization and task performance. Trained with this recipe, ARROW sets a new state of the art in multiple tracking benchmarks, while maintaining impressive results in 3D reconstruction. We discuss the limitations of ARROW in Appendix E. We will release the code and model weights to support continued progress in the community.

AI USE STATEMENT

In this work, we used generative AI tools to *implement methods*, to *clean and reformat datasets*, and to *create or edit software code*. In particular, we used agentic coding tools to accelerate the integration of baseline methods in our evaluation framework and to preprocess datasets into a shared format.

We have not used generative AI tools to *generate synthetic data sets*, *help develop theoretical models or conceptual frameworks*, *propose or refine hypotheses*, *design or provide feedback on research methodology or experiments*, *assist with translation*, *support qualitative and thematic data analysis*, *interpret results*.

The tasks *formulate mathematical claims*, *provide critical ingredients for proving mathematical claims*, and *assist in the writing of proofs* are not applicable to this work.

We have reviewed all AI-assisted work. All LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To enable reproducibility of the results, we will make both training and inference code available upon acceptance of the paper. Additionally, in Appendix B we provide further details about the losses used to train our model, architectural and implementation details, optimizer parameters, and dataset-specific sampling strategies. In Appendix C we provide details about the different evaluation protocols used for the tasks evaluated in the experimental section. All datasets used for training are publicly available, and we describe additional preprocessing and cleaning procedures in Appendix F.

REFERENCES

Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. SceneScript: Reconstructing scenes with an autoregressive structured language model. In *ECCV*, 2024.

Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *CVPR*, 2022.

Abhishek Badki, Hang Su, Bowen Wen, and Orazio Gallo. L4P: Towards unified low-level 4D vision perception. In *3DV*, 2026.

Arjun Balasingam, Joseph Chandler, Chenning Li, Zhoutong Zhang, and Hari Balakrishnan. DriveTrack: A benchmark for long-range point tracking in real-world videos. In *CVPR*, 2024.

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

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

Rich Caruana. Multitask learning. *Machine Learning*, 1997.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021.

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.

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.

Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *CVPR Workshops*, 2019.

Bishoy Galoaa, Xiangyu Bai, Shayda Moezzi, Utsav Nandi, Sai Siddhartha Vivek Dhir Rangoju, Somaieh Amraee, and Sarah Ostadabbas. Look around and pay attention: Multi-camera point tracking reimagined with transformers. In *3DV*, 2026.

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.

Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. VGGT4D: Mining motion cues in visual geometry transformers for 4D scene reconstruction. *arXiv preprint arXiv:2511.19971*, 2025.

Chengjie Huang, Guile Wu, Dongfeng Bai, and Bingbing Liu. Fast spatial tracking with visual geometry transformer. In *CVPR*, 2026.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *CVPR*, 2018.

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, 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.

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.

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, Yuri 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.

Jahyeok Koo, Inès Hyeonsu Kim, Mungyeom Kim, Junghyun Park, Seohyeon Park, Jaeyeong Kim, Jung Yi, Seokju Cho, and Seungryong Kim. MV-TAP: Tracking any point in multi-view videos. In *CVPR*, 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*, 2024.

Skanda Koppula, Frano Rajic, Abdullah Faiz Ur Rahman, Yi Yang, Ignacio Rocco, Jeet Thakwani, Rishabh Kabra, Andrew Zisserman, Joao Carreira, Siyu Tang, Carl Doersch, and Gabriel Brostow. TAPVid-MV: A benchmark for tracking any point in 3D across multiple views. *arXiv preprint arXiv:2609.01899*, 2026.

JoungBin Lee, Jaewoo Jung, Jongmin Lee, Tongmin Kim, Hyunsung Kim, Takuya Narihira, Kazumi Fukuda, Jahyeok Koo, Jisang Han, Yuki Mitsufuji, and Seungryong Kim. MVTrack4Gen: Multi-view point tracking as geometric supervision for 4D video generation. *arXiv preprint arXiv:2606.26087*, 2026.

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, Jun Hao 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.

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. *arXiv preprint arXiv:2510.13802*, 2025.

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

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. *arXiv preprint arXiv:2603.02573*, 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, Ashkan Mirzaei, Guocheng Qian, Hanwen Liang, Chuang Gan, Evangelos Kalogerakis, Peter Wonka, and Chaoyang Wang. DELTAv2: Accelerating dense 3D tracking. *arXiv preprint arXiv:2508.01170*, 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.

David Nordström, Johan Edstedt, Georg Bökman, Jonathan Astermark, Anders Heyden, Viktor Larsson, Mårten Wadenbäck, Michael Felsberg, and Fredrik Kahl. Loma: Local feature matching revisited. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, 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 Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3D machine perception. In *ICCV*, 2023.

Shenhan Qian, Ganlin Zhang, Shangzhe Wu, and Daniel Cremers. Flow4R: Unifying 4D reconstruction and tracking with scene flow. *arXiv preprint arXiv:2602.14021*, 2026.

Frano Rajič, Haofei Xu, Marko Mihajlovic, Siyuan Li, Irem Demir, Emircan Gündoğdu, Lei Ke, Sergey Prokudin, Marc Pollefeys, and Siyu Tang. Multi-view 3D point tracking. In *ICCV*, 2025.

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

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013.

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.

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.

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *ECCV*, 2024.

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

Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE RAL*, 2020.

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*, 2025b.

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

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*, 2026.

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: Advancing 3D point tracking with explicit camera motion. 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.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020.

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

Hao Yu, Haotong Lin, Jiawei Wang, Jiaxin Li, Yida Wang, Xueyang Zhang, Yue Wang, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. InfiniDepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields. In *CVPR*, 2026.

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.

Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A large photo-realistic dataset for structured 3D modeling. In *ECCV*, 2020.

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.

SUPPLEMENTARY MATERIAL

This supplementary material is structured as follows:

A Extended Related Work **15**

B Training Details **16**

B.1 Implementation Details 16

B.2 Training Data 17

B.3 Sampling Strategy 18

C Evaluation Details **20**

C.1 Evaluation Protocols 20

C.2 Baseline adaptation for multi-view tracking 20

C.3 Camera Pose Recovery 21

D Additional Results **21**

D.1 Further Ablations 21

D.2 Long-Range Tracking 22

E Limitations **23**

F Data Preprocessing **23**

G Additional Qualitative Results **25**

A EXTENDED RELATED WORK

This section complements Section 2 with a closer comparison to prior and concurrent reconstruction and tracking methods, providing additional details on their strengths and limitations.

Dense 4D prediction. Geometry transformers provide a starting point for 3D tracking through their learned multi-view priors, but require adaptations to specify motion between observations. Any4D (Karhade et al., 2026) builds on MapAnything to predict dense geometry and scene flow but remains limited to motion relative to a single frame. V-DPM (Sucar et al., 2026) encodes a video once, then repeats time-conditioned frame and global attention in its decoder for each requested target time. 4RC (Luo et al., 2026) also reuses a shared video encoding, but conditions dense decoding on a selected source–target pair. These designs differ in temporal conditioning while retaining dense map outputs.

Query composition and conditioning. Moving to point-level outputs, Fast Spatial Tracking (Huang et al., 2026) initializes queries from sampled multi-scale source features and exchanges information between global and frame-level branches: the former attends across the video, while the latter attends within each frame. UniQuery4R (Chen et al., 2026) similarly samples multi-scale source features, and implicitly conditions decoding on a selected source–target pair, similar to 4RC. Feature selection avoids fixed-length temporal embeddings, but 3D outputs lack the flexibility to express states in arbitrary viewpoints. Instead, ARROW forms queries from Fourier-encoded image coordinates, and content-derived source, target, and reference identities. Our ablations support this design over replacing appearance and coordinate encodings with interpolated encoder features (Table 9).

Long-range tracking adds the challenge of preserving point identity across occlusions and large viewpoint changes. Point4D (Jeon et al., 2026) addresses this by querying directly in 3D. It propagates predicted 3D endpoints across video chunks, avoiding image reprojection at handoffs and allowing queries during occlusion. Its formulation nevertheless remains centered on monocular video: target-time tokens are initialized from sinusoidal encodings of normalized frame positions, explicitly injecting order before contextual refinement. In this work we focus on retaining tracking performance within a single context. The content-derived identities of ARROW accommodate variable context lengths, while non-sequential training exposes the model to the large viewpoint changes encountered in long-range tracking. Figure 8 shows that retaining a longer context complements improved handoffs, for both 2D and 3D query handoffs.

Multi-view reconstruction and tracking. Beyond the scope of single-view tracking, multi-view scene capture can offer a more comprehensive understanding of the 3D scene, and can for example be used to improve tracking performance by providing additional viewpoints and reducing occlusions. Such reconstruction can also provide a stronger foundation for consistent novel-view video generation (Lee et al., 2026). One line of work assumes supplied camera calibration and synchronized videos: MVTracker (Rajič et al., 2025) fuses features across views into a shared 3D cloud using supplied or estimated depth, MV-TAP (Koo et al., 2026) uses camera geometry for multi-view 2D tracking, and LAPA (Galoaa et al., 2026) combines per-view tracks and appearance through camera-guided attention to recover 3D trajectories. MV-TAP and LAPA do not require depth inputs. A second line recovers geometry and motion together. OmniX (Jiang et al., 2026a) couples its DA3 geometry backbone with dedicated dynamic-token selection, transformation-basis prediction, and deformable sampling to produce dense trajectories. This specialized motion representation entails predicting a scene-wide trajectory field.

B Training Details

B.1 Implementation Details

Architecture. We train for 200k steps with AdamW Loshchilov & Hutter (2019) using DA3-giant weights as our encoder initialization. Its final-layer features provide context to an eight-layer pointwise decoder with hidden dimension 1280 and 16 attention heads, matching the decoder dimensions used by Open-D4RT (Zhang et al., 2026). Each query embeds a 9×9 source-centered RGB crop and Fourier features of $2(u, v) - 1 \in [-1, 1]^2$. For identity conditioning, we apply a shared LayerNorm to each ID token, concatenate them in source–target–reference order, and project with a two-layer MLP with GELU.

Training inputs and augmentation. We train on 32 NVIDIA GH200 GPUs, sampling sets of 2–24 images and adjusting the number of sets per batch to accommodate at most 24 images per GPU. We sample an aspect ratio uniformly from $[0.75, 3.0]$ and set the longer image dimension to 504 pixels. The query budget is 1024 per input image. We apply various augmentations, including perspective warping, color jitter, JPEG compression, and blur induced by downsampling and upsampling.

Optimization. We train for 200k steps using AdamW with weight decay 0.03 for the backbone and 0.003 for the decoder. Biases and one-dimensional parameters are excluded from weight decay, and gradients are clipped to a global norm of 20. The query modules use a peak learning rate of $\sqrt{2} \times 10^{-4}$. We apply layer-wise learning-rate decay to the backbone: for block $\ell \in \{0, \ldots, 39\}$, the peak learning rate is

$$\eta_\ell^{\text{peak}} = 5 \times 10^{-5}\, 0.96^{39-\ell}. \tag{6}$$

This ranges from approximately 1×10^{-5} in the first block to 5×10^{-5} in the final block, allowing larger updates in later layers. The backbone is frozen for the first 4000 steps, followed by 1000 steps of learning-rate warmup. Query modules warm up over the first 1000 steps. Both schedules subsequently follow cosine decay. We maintain an exponential moving average of the model parameters, with decay warmed toward 0.9999 over 60k updates. To reduce activation memory, we enable gradient checkpointing except in the backbone's global-attention blocks, where we retain activations to avoid recomputing attention across the full patch set.

Table 7: **ARROW training mixture.** "Multi-camera" denotes datasets with multiple video streams per scene. "Weight" denotes the probability to sample a scene from this dataset during training.

Dataset	Scene type	Tracks	Motion mask	Multi-camera	# Scenes	# Frames	Weight
Syn4D (Jiang et al., 2026b)	dynamic	✓	✓	✓	504	1.2M	13.50
Dynamic Replica (Karaev et al., 2023)	dynamic	✓	✓	✓	503	151K	12.50
Stereo4D (Jin et al., 2025)	dynamic	✓	✗	✗	31,420	6M	11.50
PointOdyssey (Zheng et al., 2023)	dynamic	✓	✓	✗	123	238K	10.50
Kubric-4D (Van Hoorick et al., 2024)	dynamic	✓	✓	✓	2,992	2.9M	8.00
CoTracker3-Kubric (Karaev et al., 2025)	dynamic	✓	✓	✗	5,837	700K	6.00
ParallelDomain-4D (Van Hoorick et al., 2024)	dynamic	✓	✗	✓	1,269	1.2M	5.00
Waymo DriveTrack (Sun et al., 2020; Balasingam et al., 2024)	dynamic	✓	✗	✓	664	658K	5.00
Kubric MOVi-F (Greff et al., 2022)	dynamic	✓	✓	✗	5,883	141K	4.00
nuScenes (Caesar et al., 2020)	dynamic	✗	✗	✓	850	205K	2.00
MidAir (Fonder & Van Droogenbroeck, 2019)	static	✗	✗	✗	138	61K	1.75
GTA-SfM (Wang & Shen, 2020)	static	✗	✗	✗	200	18K	1.75
MVS-Synth (Huang et al., 2018)	static	✗	✗	✗	120	12K	1.50
WildRGB-D (Xia et al., 2024)	static	✗	✗	✗	22,151	7.7M	1.50
Hypersim (Roberts et al., 2021)	static	✗	✗	✗	411	67K	4.00
ScanNet++ (Yeshwanth et al., 2023)	static	✗	✗	✗	902	897K	3.50
BlendedMVS (Yao et al., 2020)	static	✗	✗	✗	493	113K	3.25
Structured3D (Zheng et al., 2020)	static	✗	✗	✗	2,653	63K	2.50
Aria Synthetic Environments (Avetisyan et al., 2024)	static	✗	✗	✗	1,000	568K	2.25
Total					**78,113**	**22.8M**	**100.00**

Loss formulation. Our loss formulation largely follows D4RT (Zhang et al., 2026), as detailed below. The loss weights are $\lambda_{\text{3D}} = 1$, $\lambda_{\text{conf}} = 0.2$, $\lambda_{\text{uv}} = 0.1$, $\lambda_{\text{vis}} = 0.1$, $\lambda_{\text{normal}} = 0.5$, and $\lambda_{\text{disp}} = 0.1$. For each sample, let $\mathcal{V}$ index valid point queries and let $\boldsymbol{X} = [\boldsymbol{x}_i]_{i\in\mathcal{V}}$ contain their targets in the respective reference cameras. We normalize targets and predictions independently by their mean Euclidean distance from their respective reference-camera origins and apply an elementwise signed logarithm:

$$\mu(\boldsymbol{X}) = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \|\boldsymbol{x}_i\|_2, \qquad \psi(\boldsymbol{x}; \boldsymbol{X}) = \operatorname{sign}(\boldsymbol{x}) \odot \log\left(\mathbf{1} + \frac{|\boldsymbol{x}|}{\mu(\boldsymbol{X})}\right). \tag{7}$$

Independent normalization removes global scale ambiguity, while the logarithm reduces the influence of distant points. We use mean distance rather than mean depth as queries may use arbitrary reference cameras. The point and confidence terms in Equation (5) are

$$\mathcal{L}_{\text{3D}} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \hat{c}_i \|\psi(\hat{\boldsymbol{x}}_i; \hat{\boldsymbol{X}}) - \psi(\boldsymbol{x}_i; \boldsymbol{X})\|_1, \qquad \mathcal{L}_{\text{conf}} = -\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \log \hat{c}_i. \tag{8}$$

For $\mathcal{L}_{\text{uv}}$ and $\mathcal{L}_{\text{vis}}$, we supervise the source point's coordinates and visibility in the target image, respectively. Changing the reference observation leaves both supervision targets unchanged.

For displacement, $\boldsymbol{d}_i$ is the source-to-target world vector rotated into the reference camera and divided by the scene scale, defined as the mean distance of valid scene points from the first input camera. The head directly predicts its radial logarithm, which compresses motion magnitude while preserving direction:

$$\rho(\boldsymbol{d}) = \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|_2}\log\left(1 + \frac{\|\boldsymbol{d}\|_2}{\epsilon}\right), \qquad \epsilon = 10^{-4}, \tag{9}$$

with $\rho(\mathbf{0}) = \mathbf{0}$. We average $\|\hat{\boldsymbol{d}}_i - \rho(\boldsymbol{d}_i)\|_1$ over valid displacement targets to obtain $\mathcal{L}_{\text{disp}}$.

B.2 Training Data

The mixture contains approximately 78K scenes and 22.8M camera frames. We draw a dataset according to its mixture weight, then a scene uniformly within its training index. The weights balance annotation coverage and scene diversity, allocating 78% to dynamic data and 22% to static data.

Table 8: **Dataset-specific sampling.** We sample frames based on temporal locality, covisibility, or randomly from all available observations. "Mixed" indicates sampling from multiple camera streams. Where 3D tracks are available, we use them to construct a large percentage of all queries. The remaining query budget is divided into queries constructed from motion boundaries (B), dynamic regions (D), or based on geometric clues (G). Geometry queries are based on depth and normal discontinuities or uniformly sampled over the image plane.

Dataset	Image sampling		Query sampling				
	Policy	Local window	Track (%)	B (%)	D (%)	G (%)	Geometry
Syn4D	Mixed	$\times 10$	80	25	40	35	Edges
Dynamic Replica	Local	$\times 10$	80	25	40	35	Edges
Stereo4D	Local	$\times 4$	100	—	—	—	—
PointOdyssey, standard	Local	$\times 10$	70	25	40	35	Edges
flat background	Local	$\times 10$	80	35	55	10	Uniform
moving background	Local	$\times 10$	80	35	55	10	Uniform
Kubric-4D	Mixed	$\times 5$	80	35	55	10	Uniform
CoTracker3-Kubric	Local	$\times 4$	80	35	55	10	Uniform
ParallelDomain-4D	Mixed	$\times 5$	60	0	0	100	Edges
Waymo DriveTrack	Single camera	$\times 3$	70	0	0	100	Uniform
Kubric MOVi-F	Random	—	80	35	55	10	Uniform
nuScenes	Single camera	$\times 3$	0	0	0	100	Edges
MidAir	Covisibility	—	0	0	0	100	Edges
GTA-SfM	Covisibility	—	0	0	0	100	Edges
MVS-Synth	Covisibility	—	0	0	0	100	Edges
WildRGB-D	Random	—	0	0	0	100	Edges
Hypersim	Covisibility	—	0	0	0	100	Edges
ScanNet++	Covisibility	—	0	0	0	100	Edges
BlendedMVS	Covisibility	—	0	0	0	100	Edges
Structured3D	Covisibility	—	0	0	0	100	Edges
Aria Synthetic Environments	Local	$\times 2$	0	0	0	100	Edges

B.3 Sampling Strategy

The relational query interface separates the observations that provide context from the locations and annotations used for supervision. We first select an image set, then sample pixels or track identities, assign the roles in Equation (1), and construct the available targets. This lets static multi-view geometry, dynamic trajectories, and scenes without tracks contribute to one training objective.

Image selection. For a set of $S \in \{2, \ldots, 24\}$ images, local sampling selects observations without replacement from a contiguous candidate window of up to kS frames. Table 8 reports the maximum local-window multiplier: $\times 10$, for example, draws integer k uniformly from 1 to 10, subject to sequence length. Multi-view datasets mix single-camera sets with observations drawn across cameras in a temporal window; the resulting sets need not contain every camera at every timestep. Static datasets use covisibility where available, with minimum overlap 0.2, or local/random sampling as specified in the table.

Depth and normal edges. We combine relative depth jumps above 5% in a 3×3 neighborhood with normal changes above $25°$. Sampling normals are estimated from depth on a normalized view plane, independently of the normal annotations used by the loss. The normal detector compares four-connected neighbors, rejects grazing-angle estimates above $80°$, and requires changes to persist over a three-pixel radius. For the combined edge set $\mathcal{P}_{\text{edge}}$ and eligible pixel locations $\boldsymbol{p}$, we construct distance-based weights

$$w(\boldsymbol{p}) = 1 + 10 \exp(-d(\boldsymbol{p}, \mathcal{P}_{\text{edge}})/2)\,, \tag{10}$$

where d is measured in pixels. Up to 30% of the geometry budget is sampled uniformly within three pixels of an edge; the remainder is sampled without replacement from the unselected pixels with probability proportional to w. The positive base weight preserves coverage away from boundaries, and sampling becomes uniform when no edges are found.

Normal edges expose changes in surface orientation that need not produce a large depth jump. Qualitatively, omitting them led to more rounded reconstructions at sharp transitions, consistent with

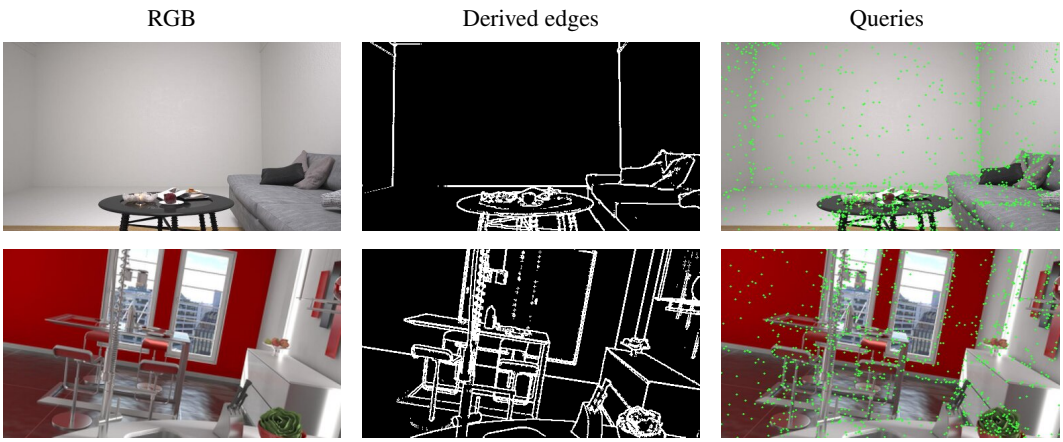


Figure 6: **Geometric query sampling.** On static scenes, we oversample queries near depth and normal discontinuities. From left to right: the RGB of a sample frame; the detected edges; the sampled source (u, v) coordinates. The fraction of queries constructed this way varies per dataset, see Table 8.

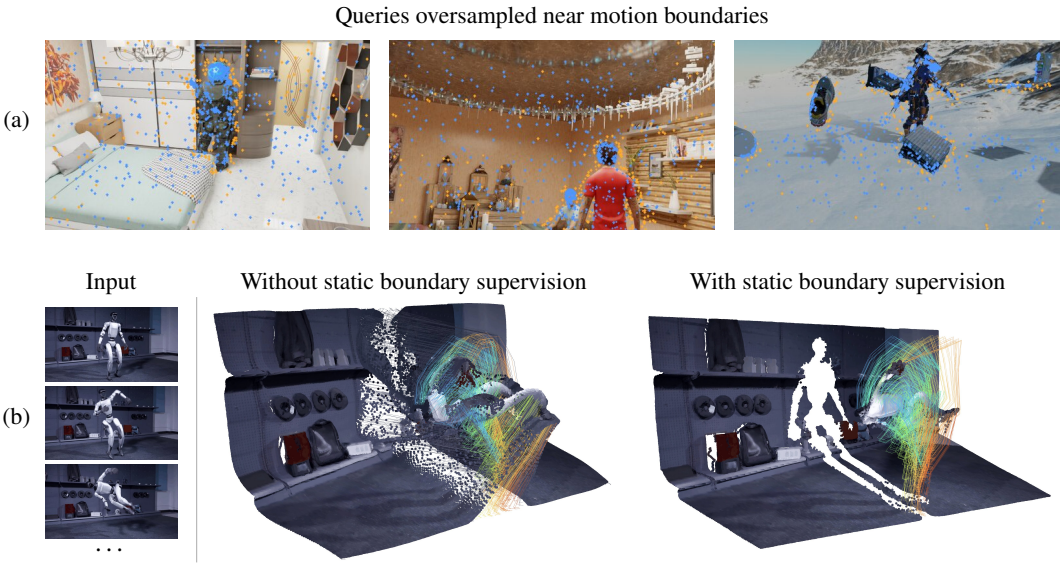


Figure 7: **Motion boundary oversampling.** (a) Oversampling near motion boundaries encourages the model to distinguish static and moving points. Blue marks indicate queries sampled from 3D tracks, and orange points indicate queries oversampled near motion boundaries. (b) Qualitative comparison of the impact of oversampled motion boundaries. Left: RGB input; Middle: tracks without motion boundary oversampling; Right: tracks with motion boundary oversampling. The fraction of queries constructed this way varies per dataset, see Table 8.

insufficient supervision of the abrupt surface change. We visualize an example of detected edges and sampled queries in Figure 6.

Motion and boundary allocation. After reserving a dataset-specific track share, we divide ordinary pixel queries among static-side motion boundaries, dynamic regions, and background geometry. The boundary partition covers a nine-pixel band, with Gaussian weights centered 4.5 pixels into the static side and standard deviation 2 pixels. Dynamic sampling covers the object interior with additional silhouette weight. Background geometry uses the edge distribution above or uniform sampling for flat-background profiles. Sampling the static side is motivated by a qualitative failure mode: supervising predominantly the moving side can allow the nearby static background to deform with it. See Figure 7 for a visualization of boundary sampling and the described failure mode.

Frame roles and valid supervision. Static scenes allocate 40% of queries to $i_{\mathrm{src}} = i_{\mathrm{tgt}} = i_{\mathrm{ref}}$; the rest sample a target observation and a different reference observation where available. Dynamic-scene queries use $i_{\mathrm{ref}} = i_{\mathrm{tgt}}$ for 40% of queries and another valid reference for the rest. This coordinate choice is independent of whether the point changes state.

Track queries use valid, visible source states and valid target states, including occluded ones. Ordinary dynamic-region queries remain same-time, $i_{\mathrm{tgt}} = i_{\mathrm{src}}$. Background queries can connect observations when their static status is trusted. All ordinary queries remain same-time for nuScenes, ParallelDomain-4D, and PointOdyssey's moving-background profile; their available tracks retain correspondence-based targets. Waymo additionally requests cross-time queries on semantic classes identified as static. Table 8 reports requested allocations: candidate shortages can increase the geometry share, and remaining non-track shortfalls can use unselected dynamic pixels.

C Evaluation Details

C.1 Evaluation Protocols

WorldTrack benchmark details. Following WorldTrack (Feng et al., 2025), we initialize queries at visible positions in the first frame and evaluate trajectories over the first 64 frames. We use the global median scale alignment protocol. Ground-truth trajectories are expressed in the first camera's coordinate system, and predictions are multiplied by

$$s = \frac{\mathrm{median}_{i \in \mathcal{V}} \, \|\boldsymbol{x}_i^{\mathrm{gt}}\|_2}{\mathrm{median}_{i \in \mathcal{V}} \, \|\hat{\boldsymbol{x}}_i\|_2}, \tag{11}$$

where $\mathcal{V}$ indexes valid track–time correspondences with finite ground truth and predictions. This fits a single scale without adjusting rotation or translation. APD averages the percentage of valid positions with Euclidean error at most $\{0.1, 0.3, 0.5, 1.0\}$ meters; EPE is the mean Euclidean error. Dynamic-only evaluation retains tracks whose cumulative displacement over consecutive valid ground-truth positions exceeds 1 cm, and estimates scale separately on this subset. Results are averaged across sequences.

Multi-view benchmark adaptation. We provide the sampling and scoring details for the RGB-only adaptation in Section 4. For Multi-View Kubric and DexYCB we retain the 24 timesteps and the released query subsets. For Panoptic Studio, we subsample every second frame from the first 48 frames $(0, 2, \ldots, 46)$ for each of the camera-rig configurations. To maintain sufficient query coverage in this shorter clip, we reapply MVTracker's query-sampling procedure within the retained window. The source view is sampled uniformly among views in which the query is visible; ground-truth calibration is used only to project the query into that image and construct evaluation targets.

Multi-view coordinates and scoring. Each query is predicted at all retained timesteps, both before and after its source time. We score one 3D position per physical timestep in the first input camera's coordinate system, taking the first camera's target slot. Ground-truth visibility is the union over rig cameras, using physical visibility rather than the causal mask used for query selection; predicted visibility likewise uses the maximum over views. APD and EPE evaluate valid, ground-truth-visible positions, whereas OA measures visibility agreement over all valid positions, including occluded states.

We fit one scale per scene and rig using Equation (11) over valid, visible correspondences with finite predictions. For Table 3, Kubric EPE is converted to meters using approximately 0.13 m per native unit. We pool track–time pairs within each scene, average scenes and then Panoptic Studio rigs equally, and weight the three datasets equally in the overall average. These protocol changes, including the different input assumptions, preclude direct numerical comparison with the official MVTracker results.

C.2 Baseline adaptation for multi-view tracking

Trackers with external geometry. For the multi-view comparison, we replace ground-truth depth and calibration with predicted geometry and lift pixel queries using the predicted source-view

Algorithm 1 Derivation of camera pose $i \to j$.

Require: Encoded context $\boldsymbol{F}$, images I_i, I_j.

 # Query construction and decoding

$\mathcal{G}_i \leftarrow \mathrm{UniformGrid}(I_i, 64 \times 64)$ ▷ *Grid for source frame i*

$\mathcal{Q}_i \leftarrow \{\boldsymbol{q}_{i,k}\}_{k=1}^{M}, \quad \boldsymbol{q}_{i,k} = (u_k, v_k, i, i, i)$ ▷ *Coordinates in camera i*

$\mathcal{Q}_j \leftarrow \{\boldsymbol{q}_{j,k}\}_{k=1}^{M}, \quad \boldsymbol{q}_{j,k} = (u_k, v_k, i, i, j)$ ▷ *Coordinates in camera j*

$(\boldsymbol{X}^{(s)}, \boldsymbol{c}^{(s)}, \boldsymbol{m}^{(s)}) \leftarrow \mathrm{Decode}(\mathcal{Q}_s, \boldsymbol{F}), \quad s \in \{i, j\}$

 # Confidence-based selection and weighting

$\boldsymbol{c} \leftarrow \boldsymbol{c}^{(i)} \odot \boldsymbol{c}^{(j)}$ ▷ *Joint confidence*

$\mathcal{K} \leftarrow \{k \in \{1, \dots, M\} \mid m_k^{(i)} = m_k^{(j)} = 1\}$ ▷ *Valid in both frames*

$\mathcal{K} \leftarrow \{k \in \mathcal{K} \mid c_k \geq \mathrm{median}(\boldsymbol{c}_{\mathcal{K}})\}$ ▷ *Keep the most confident half*

$w_k \leftarrow c_k^2 / \sum_{\ell \in \mathcal{K}} c_\ell^2, \quad k \in \mathcal{K}$ ▷ *Normalize squared confidence*

 # Solve via SVD: align camera i to camera j

$(\boldsymbol{R}_{i\to j}, \boldsymbol{t}_{i\to j}) \leftarrow \arg\min_{\boldsymbol{R}\in \mathrm{SO}(3),\ \boldsymbol{t}\in\mathbb{R}^3} \sum_{k\in\mathcal{K}} w_k \left\| \boldsymbol{x}_k^{(j)} - \left(\boldsymbol{R}\boldsymbol{x}_k^{(i)} + \boldsymbol{t} \right) \right\|_2^2$ ▷ *Solve by SVD*

return $(\boldsymbol{R}_{i\to j}, \boldsymbol{t}_{i\to j})$ ▷ $\boldsymbol{x}^{(j)} = \boldsymbol{R}_{i\to j}\boldsymbol{x}^{(i)} + \boldsymbol{t}_{i\to j}$

depth. MVTracker uses VGGT or DA3 reconstruction independently at each timestep, or VGGT-Ω reconstruction jointly over the clip. TAPIP3D (Zhang et al., 2025a) tracks each query in its source-view video using joint VGGT-Ω geometry, while LAPA (Galoaa et al., 2026) uses this geometry with CoTracker3 observations. MVTracker and LAPA combine forward and reversed-time passes at each query's source time; TAPIP3D uses its bidirectional inference mode. MVTracker uses a 4×4 global support grid per view at the start of each pass, and TAPIP3D uses a 40×40 support grid. The VGGT-Ω comparisons use the 416-resolution checkpoint released in August 2026.

Joint geometry and tracking baselines. For 4RC, we place the reference image first, sample queries from the dense track field of their source image, and restore the original input order. OmniX (Jiang et al., 2026a) receives camera-grouped videos with synchronized timestamps. Given OmniX's high memory requirements, we batch trajectory-head and deformable-attention computations to avoid GPU memory exhaustion while preserving full temporal context. Both methods are scored using the same adapted protocol; AJ and OA are omitted because neither provides a native visibility prediction.

C.3 Camera Pose Recovery

Following D4RT (Zhang et al., 2026), Algorithm 1 recovers the pose of each camera i relative to reference camera j. The point matrices $\boldsymbol{X}^{(i)}$ and $\boldsymbol{X}^{(j)}$, with columns $\boldsymbol{x}_k^{(i)}$ and $\boldsymbol{x}_k^{(j)}$, describe the same physical points at source state i, expressed in cameras i and j; moving objects therefore also provide valid correspondences. Both query sets use the same enumeration $\mathcal{G}_i = \{(u_k, v_k)\}_{k=1}^{M}$ of the source grid; decoding preserves k in the point-matrix columns and confidence and mask entries. We choose the middle input as reference and assign it the identity pose. The decoding step constructs validity masks $\boldsymbol{m}^{(i)}, \boldsymbol{m}^{(j)}$ that retain finite 3D predictions with positive finite confidence, in-bounds predicted UV, and positive visibility logits. The final SVD solve fits a proper rotation and translation without rescaling. We omit prediction hats in this algorithm; $\boldsymbol{R}_{i\to j}$ is a rotation matrix and $\boldsymbol{t}_{i\to j}$ is a translation vector.

D Additional Results

D.1 Further Ablations

Design ablations. We evaluate design alternatives on 64-frame WorldTrack sequences using the final configuration in Table 6 as the baseline. Each variant follows the same ablation training protocol, except for the indicated change. Table 9 reports absolute baseline scores and differences for each variant, for both all-point and dynamic-only tracking. Freezing the encoder and training only the decoder reduces mean APD by 8.71 percentage points overall. This illustrates that while the encoder

Table 9: **Design-choice ablations on WorldTrack.** All rows use the ablation training setup of Table 6. Entries report APD (%) and EPE (m) over all points. S5 reports absolute scores, other rows report signed differences from S5. The first three variants are trained separately with the indicated change. The EMA row evaluates exponential-moving-average weights from the S5 training run.

Configuration	Aria Digital Twin		Dynamic Replica		PointOdyssey		Panoptic Studio		Avg.	
	APD ↑	EPE ↓	APD ↑	EPE ↓	APD ↑	EPE ↓	APD ↑	EPE ↓	APD ↑	EPE ↓
Frozen encoder	−6.85	+0.0612	−8.20	+0.0706	−4.39	+0.0851	−15.41	+0.1487	−8.71	+0.0914
Multi-level context	+0.33	−0.0013	−0.33	**−0.0018**	−0.21	+0.0042	**+1.32**	**−0.0088**	+0.28	**−0.0019**
Patch-feature query initialization	−0.59	+0.0072	−2.59	+0.0311	−0.33	+0.0270	−2.06	+0.0166	−1.39	+0.0205
Stage S5 (Table 6)	87.52	0.1495	**81.01**	0.2026	74.36	0.2519	77.13	0.1894	80.01	0.1983
EMA weights (final ARROW configuration)	**+1.37**	**−0.0110**	−0.23	+0.0046	**+0.26**	**−0.0049**	−0.09	+0.0060	**+0.33**	−0.0013

Table 10: **Short-context ablations on WorldTrack.** Here we use 24-frame full-context inference during all stages. Stages follow Table 6. 🔧: order-dependent querying (D4RT); 🪄: order-invariant querying (ARROW). Entries report APD (%) and EPE (m) over all points. S5 reports absolute scores, other rows report signed differences from S5.

Stage	Query Approach	PE	Sequential Sampling	Data	Eval	Aria Digital Twin		Dynamic Replica		PointOdyssey		Panoptic Studio		Avg.	
						APD ↑	EPE ↓	APD ↑	EPE ↓	APD ↑	EPE ↓	APD ↑	EPE ↓	APD ↑	EPE ↓
S1	🔧	✓	✓	video	full	−9.09	+0.0795	+0.04	−0.0001	−0.19	+0.0043	−2.50	+0.0178	−2.94	+0.0254
S2	🪄	✓	✓	video	full	−4.29	+0.0351	**+1.31**	**−0.0113**	**+1.27**	−0.0078	−5.51	+0.0469	−1.81	+0.0157
S3	🪄	✗	✓	video	full	−2.30	+0.0174	+0.03	+0.0035	+0.95	**−0.0096**	−5.71	+0.0455	−1.75	+0.0142
S4	🪄	✗	✗	video	full	−3.46	+0.0272	−0.61	+0.0069	−2.24	+0.0163	−3.65	+0.0316	−2.49	+0.0205
S5	🪄	✗	✗	all	full	**87.19**	**0.1560**	79.14	0.2211	73.68	0.2599	**80.10**	**0.1668**	**80.03**	**0.2010**

representation can provide a good baseline for tracking, fine-tuning the encoder is important for optimal performance.

Multi-level decoding uses DA3-Large layers $11, 15, 19, 23$, matching the feature levels of its DPT head. Successive decoder layers attend to these levels and then reuse the deepest level. Gains are small on average and vary across datasets; we therefore retain a shared final-layer context for simplicity and memory efficiency. We also evaluate a common alternative for query initialization: bilinearly sampling encoder patch features at the queried UV coordinates, replacing the RGB crop and Fourier-coordinate encoding (Chen et al., 2026; Huang et al., 2026; Yu et al., 2026). Although this variant converges faster initially, it yields lower final tracking accuracy and predictions with a more patch-like spatial structure. These observations favor initializing the query's appearance and spatial components independently of encoder patch features. Finally, we maintain an exponential moving average (EMA) of the model parameters. EMA slightly improves average APD and EPE for both all-point and dynamic-only tracking, so we retain it for the final model.

Short-context tracking. To assess whether our more general formulation incurs a cost in short-context tracking accuracy, we evaluate all cumulative stages on 24-frame WorldTrack sequences, keeping within the 2–24-image training range (Table 10). All variants therefore use full-context inference. Stage S5 achieves the best average APD and EPE for both all-point and dynamic-only tracking, although earlier stages lead on some individual datasets. These results demonstrate that the final design's broader input support comes without an average accuracy penalty at context lengths seen during training.

D.2 Long-Range Tracking

Figure 8 compares full-context inference with 48-frame windows and 8-frame overlap on Aria Digital Twin and Panoptic Studio. Each point evaluates an entire sequence prefix using WorldTrack's APD thresholds and one global median-distance scale. Windowed inference transfers predicted 2D query locations between windows. For ARROW, full context yields higher APD at every evaluated prefix longer than one window, with a widening gap as the sequence grows. This comparison supports retaining full context, although it does not isolate the contributions of handoff drift and occlusion recovery.

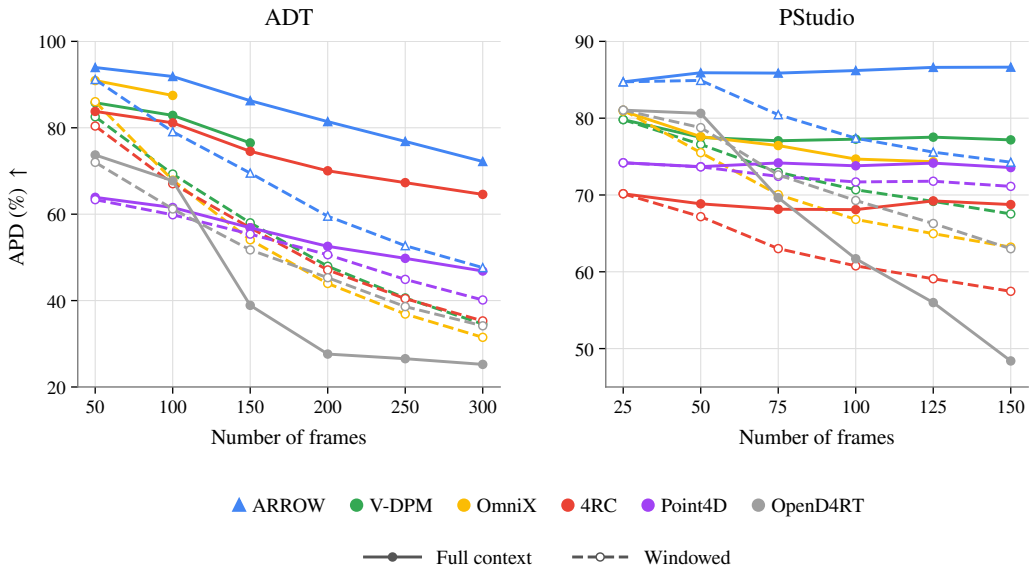


Figure 8: **Long tracking performance.** All-point APD over the indicated sequence prefix with full-context inference (solid) or 48-frame windows with 8-frame overlap (dashed). Full-context inference performs better than windowed inference for all methods except OpenD4RT. ARROW outperforms other methods under matched scene size and inference method. Evaluations run on NVIDIA GH200 GPUs. V-DPM evaluation on 200 frames did not finish within 80 GPU-hours. Other missing points ran out of memory.

E Limitations

While ARROW shows strong tracking performance as reported in the main paper, removing explicit time encodings may limit the models ability to interpolate between points in time or to extrapolate to future points in time. To be more precise, ARROW only supports ouputs at those points in time that correspond to the observations that it received as input.

ARROW can produce dense, high-resolution outputs; however, its computational requirements grow with both the number of context images and the number of decoded queries, making such outputs computationally expensive.

Besides, while queries share image context during decoding, they do not interact directly. We hypothesize that this lack of interaction might lead to weaker performance in far-away/low-confidence regions.

Furthermore, in multi-view tasks the model performs poorly for out-of-distribution training data (e.g., ego-centric, side-view). We see tackling such domain gaps as an exciting future work direction.

Finally, the pose and (u, v) estimations are globally accurate but lack local precision. Future work may improve the precision of such estimates, for example via feature matching or local optimization.

F Data Preprocessing

PointOdyssey. Moving backgrounds, flattened background geometry and noisy annotations complicate geometric and motion supervision in PointOdyssey, as illustrated in Figure 9. Instance labels include static environment objects and do not directly specify motion. We therefore review scenes in our training index using an interactive viewer, label instances as static, dynamic, or unreliable, and record background geometry and annotation issues. Mask refinement repairs local label dropouts, treats larger depth-valid unlabeled regions separately, and combines manual decisions with track-based checks for unresolved regions. Unreliable regions receive a dedicated ignore label. The resulting scene metadata determines the sampling profile: standard, flat-background, or moving-background. The flat-background profile emphasizes tracks and moving instances, while the moving-background

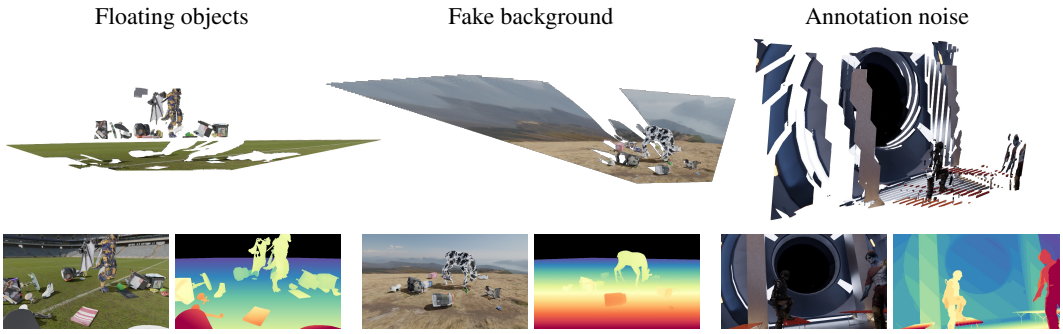


Figure 9: **Geometry and annotation issues in PointOdyssey.** From left to right: objects floating above the ground instead of resting on it, a background image mapped onto a flat plane rather than modeled scene geometry, and depth annotation artifacts that produce fragmented geometry. Each example shows a 3D visualization and the corresponding RGB image (bottom left) and depth map (bottom right). These corrupted cases motivate our manual scene review, scene-specific sampling rules, and exclusion of unreliable supervision.

profile restricts ordinary queries to the source point state. We preserve per-frame intrinsics for zoom sequences and transform the released normals into our camera coordinate convention.

Hypersim. Some supplied camera orientation matrices do not have determinant $+1$. We correct these using SVD, flipping the singular-vector direction associated with the smallest singular value when necessary to obtain a proper rotation. This avoids treating a reflection as a camera rotation. We also reconcile normal-map axes and orient valid normals toward the camera.

Track and visibility corrections. We decode Syn4D's compact surface correspondences into shared world-space trajectories and map Dynamic Replica's frame-local instance labels to persistent identities. ParallelDomain-4D trajectories are derived from flow and depth and validated using forward–backward, cross-view depth, flow, and instance-consistency checks.

Other data preparation. We remove simulated vignetting and rectify ASE cameras, filter unreliable BlendedMVS geometry, and retain WildRGB-D depth only in reliable foreground regions. For each scene, we construct a covisibility graph from pairwise overlap estimated through depth reprojection consistency, which guides image sampling during training. Dynamic identities remain defined by the dataset's track correspondences.

G Additional Qualitative Results

Figure 10 compares depth predictions for a selected video frame. Figure 11 shows UV tracks and visibility for queries from different source frames, with correct occlusion predictions, consistent tracks through articulated motion, and static background points remaining attached to the same scene locations despite camera motion. Figures 12 and 13 compare reconstruction and multi-view 3D tracking, respectively, while Figure 14 shows further qualitative examples.

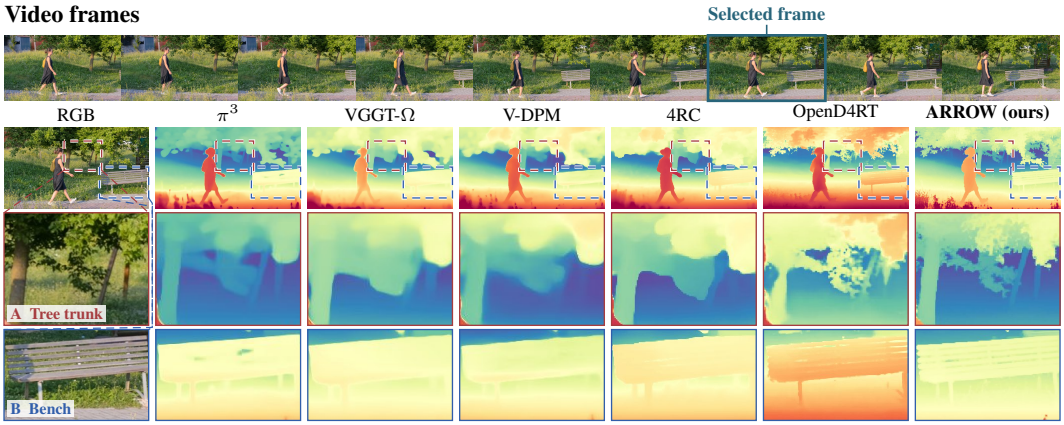


Figure 10: **Video depth comparison.** Top: RGB frames in temporal order, with the selected frame outlined. Middle: the selected RGB frame and depth predictions. Bottom: matching crops of the tree trunk (A) and the bench (B), marked by dashed boxes in the full views. ARROW preserves fine structural details in both highlighted regions.

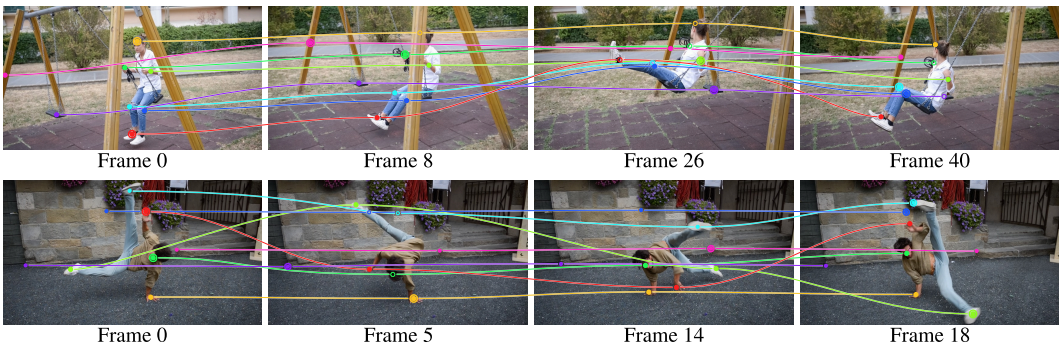


Figure 11: **Qualitative UV tracking.** ARROW predicts image coordinates $(\hat{u}, \hat{v})$ and visibility for the swing (top) and breakdance (bottom) sequences. A circled dot ($\odot$) marks each source query in its source frame; all other markers are predictions. Filled dots ($\bullet$) indicate visible points and hollow circles ($\circ$) indicate occluded points. Colors identify corresponding points, and curves connect their locations across the displayed frames. These examples illustrate occlusion handling, tracking through challenging articulated motion, and consistent tracking of static scene points under camera motion.

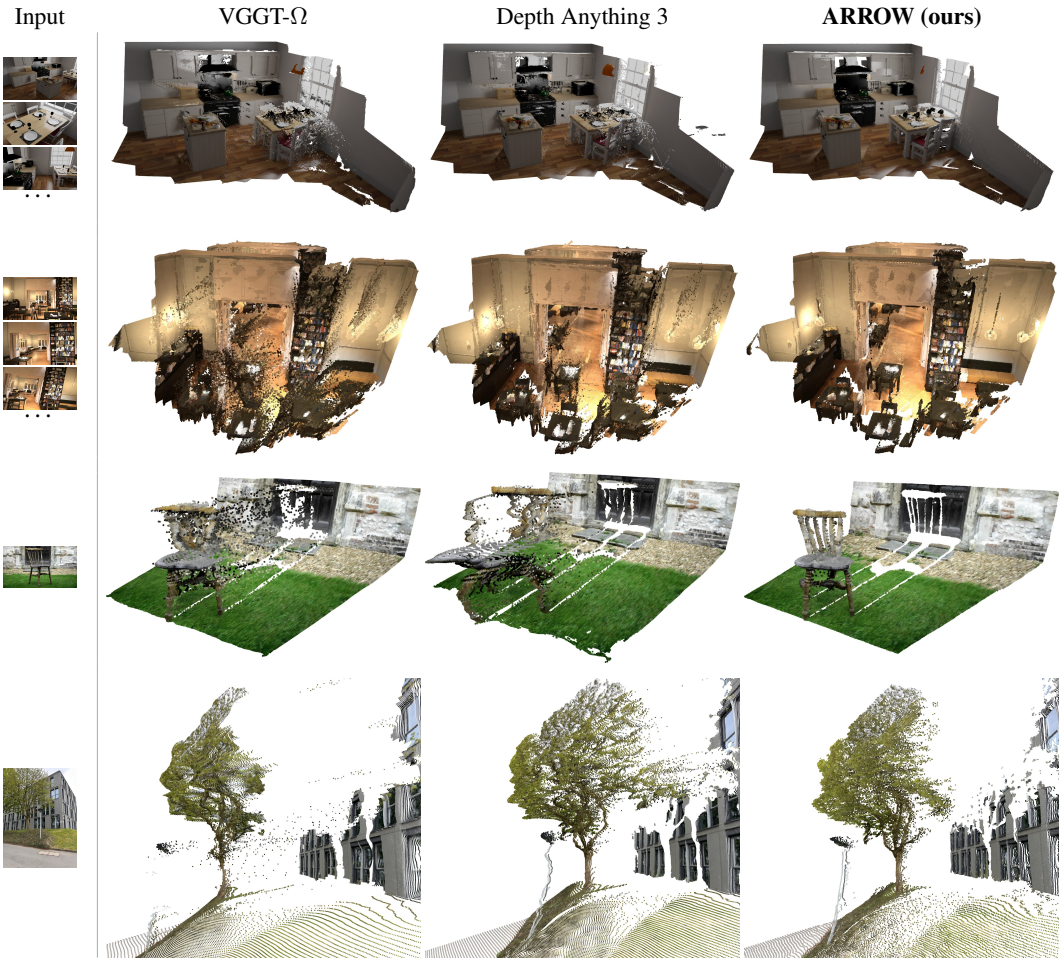


Figure 12: **3D reconstruction.** Multi-view (top two rows) and monocular (bottom two rows) 3D reconstruction results compared with state-of-the-art methods.

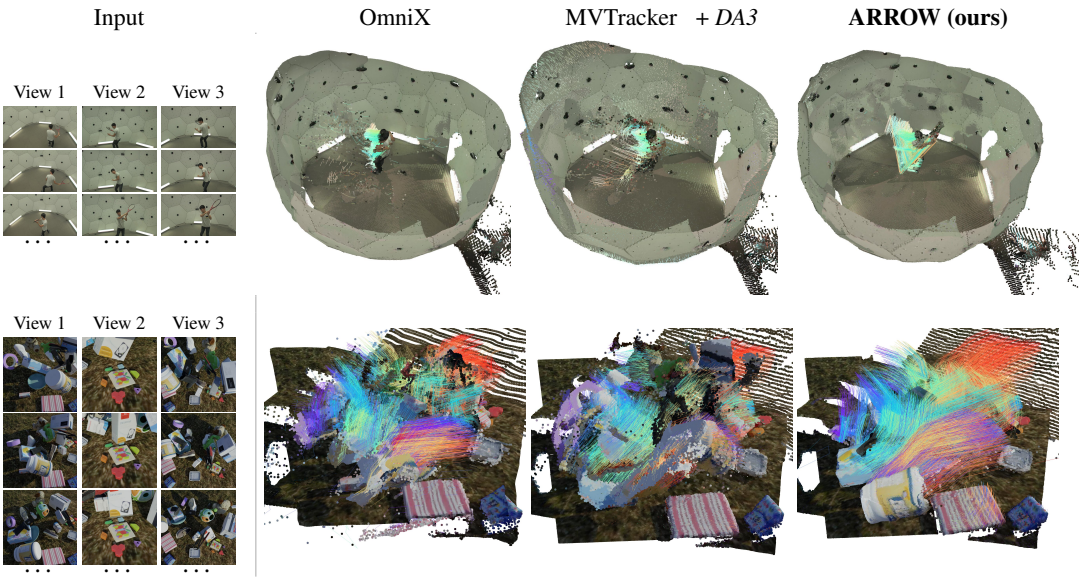


Figure 13: **Multi-view 3D tracking.** Examples from Panoptic Studio (top) and Multi-View Kubric (bottom). The three input columns on the left show successive frames from three camera views.

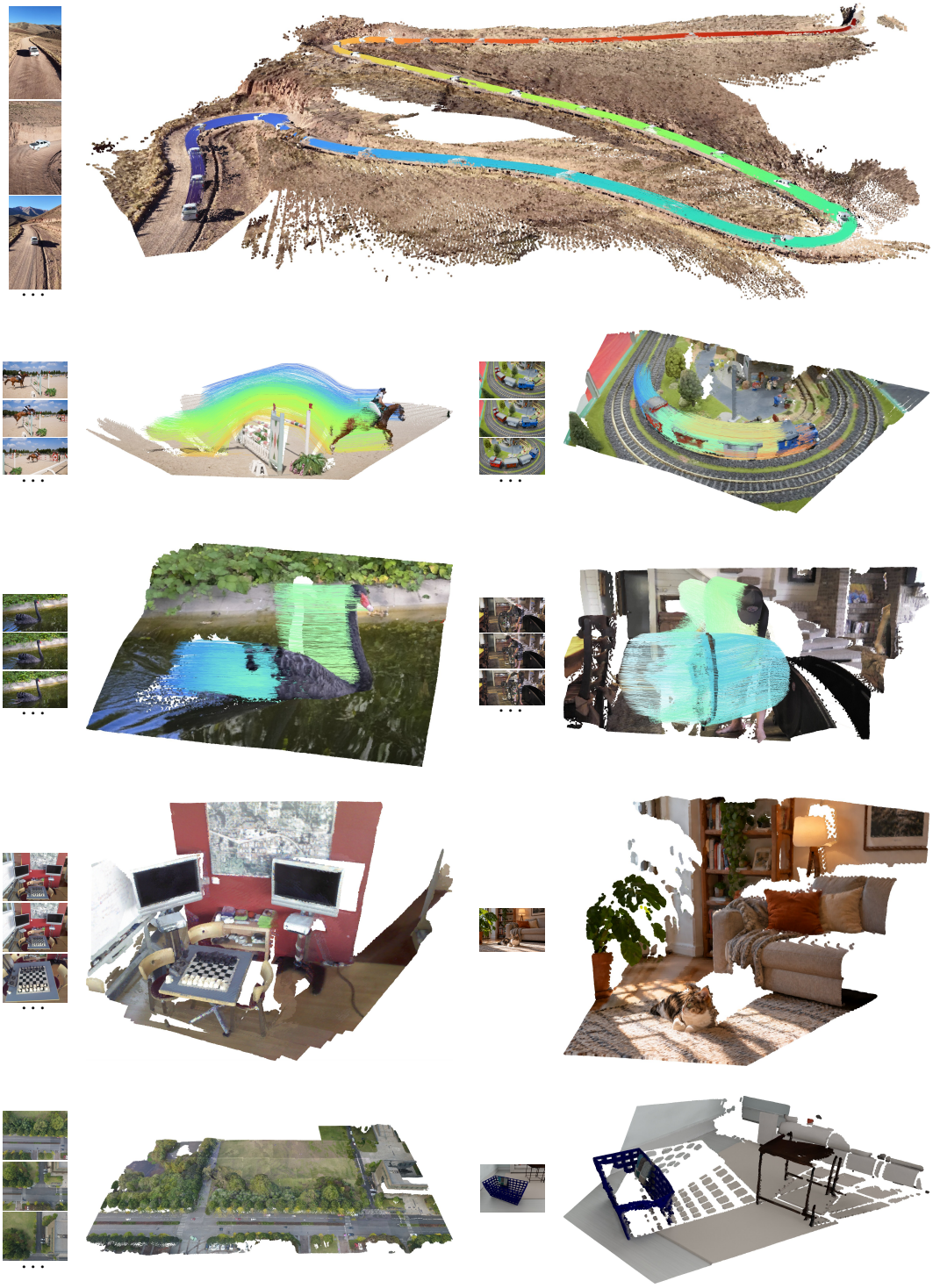


Figure 14: **Additional qualitative results.** ARROW works well across dynamic scenes (top three rows), static scenes (bottom two left) and monocular inputs (bottom two right).