arXiv:submit/8425325 [cs.CV] 25 Nov 2026

Metric 4D Modeling for Embodied Perception

Di Liu[∞], **Xuweiyi Chen**[∞,♆], **Cheng Zhang**[∞], **Adam Harley**[∞], **Zhao Dong**[∞], **Tanner Schmidt**[∞], **David Caruso**[∞], **Pierre Moulon**[∞], **Shugao Ma**[∞], **Zezhou Cheng**[♆], **Richard Newcombe**[∞], **Jakob Julian Engel**[∞]

[∞]Reality Labs Research, Meta, [♆]University of Virginia

Project page Code Benchmark

Metric 3D geometry and motion are essential for embodied reasoning. Yet existing feedforward 4D models underuse synchronized views and calibrated cameras available to embodied systems. We introduce SCALE4D, a unified feedforward model that predicts metric 3D geometry and trajectories from single- or multiview videos, with optional camera conditioning. A shared point-query representation integrates spatiotemporal attention and cross-view fusion, while metric camera positional encoding incorporates calibrated geometry without compromising physical scale. Geometry-preserving scale augmentation encourages the model to use camera baselines for metric prediction. Beyond supervised learning, SCALE4D's flexible camera conditioning enables asymmetric self-training, where stronger camera-conditioned predictions supervise RGB-only inference on unlabeled videos. Across egocentric and robotic benchmarks, SCALE4D achieves strong metric tracking and depth without ground-truth scale alignment. We further demonstrate zero-shot generalization to larger numbers of views, with performance continuing to improve beyond the maximum of three views seen during training.

Date: November 25, 2026

Meta

1 Introduction

Embodied systems, from wearable devices to robots, should understand not only what happens around them, but also where objects are and how they move in metric space [84]. Metric 3D geometry and motion are essential for embodied reasoning, enabling physically meaningful estimates of distances, velocities, and accelerations needed to understand and interact with the environment. Recent feedforward 4D models enable direct prediction of scene geometry and trajectories from video [12, 41, 59, 74, 80], yet recovering accurate metric geometry remains challenging. Unlike general video, embodied systems often capture synchronized observations from multiple cameras, together with calibrated camera poses [2, 11, 31]. These measurements provide complementary geometric evidence and physical scale references that could improve 4D perception [22, 30]. However, existing approaches have limited flexibility in leveraging such measurements across different sensing configurations.

We introduce SCALE4D, a unified framework for metric 4D perception that adapts to the geometric information available at inference (Fig. 1). Our model supports both single-view and multiview videos, with or without calibrated camera information, using a shared representation and a single set of weights. These configurations provide different levels of geometric evidence: additional views offer complementary observations that help resolve spatial ambiguity, while calibrated camera poses provide a physical reference for recovering metric scale. Rather than treating these settings as separate tasks, SCALE4D integrates visual and geometric evidence within a unified feedforward architecture, enabling metric geometry and motion prediction across diverse embodied scenarios.

A central challenge is reliably incorporating calibrated camera geometry into metric 4D prediction. Directly using camera translations in physical units can destabilize geometric attention during training, while normalizing them risks losing metric scale information needed for accurate prediction. We address this challenge through metric camera positional encoding (MCPE), which regulates camera translation magnitudes while explicitly retaining metric scale cues. We further introduce geometry-preserving scale augmentation to encourage the model to use calibrated camera baselines rather than relying solely on appearance-based scale priors.

Beyond its strong metric predictions, SCALE4D's flexible input interface also provides a natural foundation for self-training. In particular, calibrated cameras enable more accurate metric predictions that can supervise RGB-only learning on unlabeled videos. We

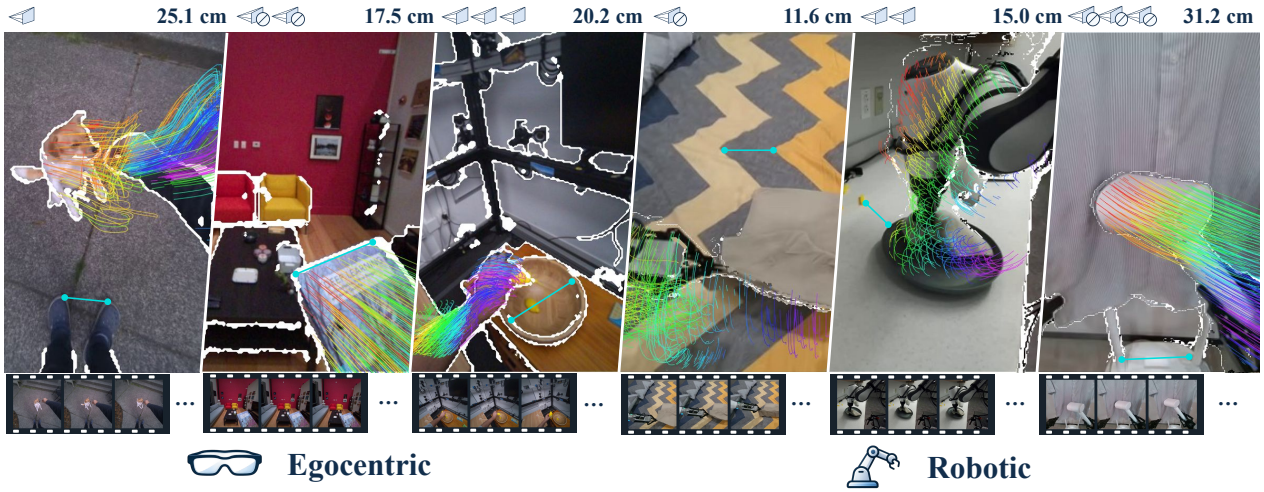

Figure 1 Scale4D predicts dense metric 3D trajectories across embodied perception scenarios, supporting varying view counts and optional camera conditioning. Cones (◁) show supported view counts; slashed circles (⊘) mark disabled camera conditioning. Numbers indicate predicted 3D distances (cm) between marked points. Filmstrips present frames from the query view.

explore this capability through camera-asymmetric self-training, where a camera-conditioned teacher supervises an RGB-only student on unlabeled videos. We show that this asymmetry improves camera-free inference over symmetric self-training, illustrating how geometric measurements can benefit learning even when they are not available at inference.

Our contributions are: (1) a unified feedforward framework for metric 4D perception that supports flexible view counts and camera configurations, generalizing zero-shot from at most three training views to ten; (2) MCPE and geometry-preserving scale augmentation for stable and accurate use of metric camera information; and (3) a demonstration of Scale4D's potential for self-training through camera-asymmetric supervision, improving RGB-only metric prediction using unlabeled calibrated videos.

2 Related Work

Metric Space Modeling. Recovering geometry in physical units remains challenging due to the inherent scale ambiguity of monocular observations. Monocular approaches such as Metric3D [78], UniDepth [51, 52], and MoGe [32, 66] learn metric geometry from RGB images, while Depth Anything 3 [40] extends geometric prediction to flexible view configurations. Alternatively, calibrated stereo provides a direct metric reference for depth estimation [72], and methods such as Pow3R [22] and MapAnything [30] incorporate additional geometric measurements. Extending metric prediction to dynamic scenes is more challenging, as both scene geometry and motion must be recovered in a consistent metric coordinate system. Any4D [29] addresses this problem through a factorized representation, predicting scale-normalized geometry and motion together with a separate global metric scale factor. In contrast, Scale4D directly predicts metric 3D positions and trajectories through a unified point–time query representation, incorporating calibrated camera geometry into attention when available.

Unified 4D Reconstruction and Tracking. Feedforward reconstruction provides an alternative to classical structure-from-motion and multiview stereo [56, 57]. DUSt3R [67] and MASt3R [37] regress point maps from image pairs, while VGGT [63] and related methods [71, 76] extend reconstruction to multiple images. Recent approaches recover dynamic geometry from video [6, 39, 42, 65, 68, 82], integrate geometric observations across time [17, 59, 60], or optimize scene-specific dynamic representations [36, 64]. Point trackers such as PIPs [18], TAPIR [9], CoTracker [27], and AllTracker [19] predict image-space trajectories, while SpatialTracker [73, 74], DELTA [46], and TAPIP3D [79] recover 3D trajectories using depth information. Other methods directly predict 3D motion [12, 23, 43, 44], while multiview tracking approaches [13, 24, 33, 53] exploit synchronized observations. More recently, D4RT [80], L4P [1], and TraceAnything [41] unify geometry and motion prediction within feedforward 4D frameworks. Scale4D extends unified 4D perception to metric space with flexible view counts and optional camera conditioning, without requiring an external depth estimator.

Camera Conditioning and Geometric Attention. Per-

pixel Plücker embeddings encode viewing rays [20, 58], although their absolute coordinates depend on the chosen world frame. In geometric attention, GTA [45] applies group actions to attention features, PRoPE [38] introduces projective relative positional encoding, and UCPE [81] uses local ray frames to represent relative camera geometry for video generation. While these methods encode camera relationships, their application to metric 4D prediction requires handling camera translations in physical units without losing absolute scale information. SCALE4D addresses this challenge through scale-conditioned geometric attention, enabling stable metric camera conditioning alongside RGB-only inference.

Self-supervised geometry learning and distillation. Learning from unlabeled visual data has been explored through geometric consistency and self-training. Self-supervised depth methods exploit photometric and geometric constraints to learn scene structure without dense annotations [3, 15]. More recent approaches leverage pseudo-labels for monocular depth estimation [77] and point tracking [10]. Selfi [8] improves 3D reconstruction through geometric feature alignment, while SelfEvo [21] introduces online self-distillation with spatiotemporal context asymmetry for 4D perception. SCALE4D instead exploits geometric observation asymmetry, where camera conditioning enable a teacher to supervise metric trajectory prediction from reduced inputs. Supervised replay preserves a metric reference as self-labeling expands to additional calibrated videos.

3 Metric 4D Modeling

SCALE4D extends D4RT's feedforward encoder–decoder architecture [80] to predict metric scene geometry and motion from varying numbers of synchronized views, with optional camera conditioning (Fig. 2). Our design exploits calibrated camera geometry when available while supporting prediction from reduced inputs. A central challenge is preserving metric scale: jointly scaling scene geometry and camera translations leaves image observations unchanged, while calibrated camera baselines provide an explicit metric reference that can resolve this ambiguity. We therefore develop a multiview architecture with geometry-aware attention, together with metric learning strategies that preserve and exploit physical scale.

3.1 Multiview Architecture

Multiview Spatiotemporal Encoder. Given V synchronized videos with T frames $\mathcal{I} = \{I_{v,t}\}_{v=1,t=1}^{V,T}$, a shared embedding layer forms spatial tokens spanning two consecutive frames. With weights shared across views, the encoder $\mathcal{E}_\theta$ alternates spatiotemporal *video self-attention* within each view and *same-time view attention* across views. With one view, the latter reduces to spatial attention. The encoder produces multiview features $\{\mathbf{F}_v\}_{v=1}^{V}$.

Camera Conditioning. To incorporate camera geometry, we introduce a parallel geometry-aware attention branch following UCPE [81], termed metric camera positional encoding (MCPE). Given calibration $\mathcal{C} = \{\mathbf{K}_{v,t}, \mathbf{T}^{w}_{c,v,t}\}$ with camera intrinsics and metric camera-to-world poses, MCPE associates video tokens with viewing rays and operates alongside RGB attention in both encoder stages. Omitting MLPs, each stage updates token features $\mathbf{z} = \{\mathbf{z}_a\}$, where $\mathbf{z}_a \in \mathbb{R}^D$ denotes the a-th token feature, as

$$\mathbf{z}'_a = \mathbf{z}_a + \mathcal{A}_{\mathrm{RGB}}(\mathbf{z})_a + b_{\mathrm{cam}}\,\mathrm{MCPE}(\mathbf{z};\mathcal{C})_a, \quad (1)$$

where $\mathcal{A}_{\mathrm{RGB}}$ denotes standard attention over video token features, and $b_{\mathrm{cam}} \in \{0, 1\}$ enables or disables camera conditioning. We jointly train both settings, allowing the same model weights to operate with or without camera information. Sec. 3.2 details MCPE.

Metric 4D Query Decoder. Following D4RT [80], we represent 4D geometry through point–time queries. A query $q_{i,t} = (\mathbf{u}_i, v_i, s_i, t, t_c)$ identifies point i by its source pixel $\mathbf{u}_i \in \mathbb{R}^2$, source view v_i, and source time s_i, and requests its 3D position at target time t in the reference camera frame (v_i, t_c). To construct shared decoder memory $\mathbf{F}$, we normalize and concatenate encoder features across layers for each view, project them into tokens, and aggregate tokens across views. A query decoder $\mathcal{D}_\theta$ predicts $\hat{\mathbf{x}}_{i,t}^{v_i,t_c} = \mathcal{D}_\theta(q_{i,t}; \mathbf{F}) \in \mathbb{R}^3$. The decoder uses cross-attention to retrieve scene features, together with a local RGB patch encoding the source point's appearance. Fixing $(\mathbf{u}_i, v_i, s_i, t_c)$ and varying t yields a metric 3D trajectory. Querying all pixels at $t = s_i = t_c$ produces a depth map, while fixing $t = s_i$ and a shared reference frame yields a 3D point cloud. Auxiliary heads predict image positions, visibility, 3D motion, normals, and confidence. Each clip is encoded once, and all queries reuse its features.

3.2 Metric Camera Positional Encoding

We build on UCPE [81], which incorporates camera geometry into attention through relative ray transformations. For each image token a, the camera origin, viewing direction, and camera's downward axis define a local ray frame with ray-to-world transformation $\mathbf{T}_a^{\mathrm{wr}}$, where w and r denote the world and ray coordinate frames, respectively. The relative ray transformation between tokens a and b is $\mathbf{G}_{ab} = (\mathbf{T}_a^{\mathrm{wr}})^{-1}\mathbf{T}_b^{\mathrm{wr}}$.

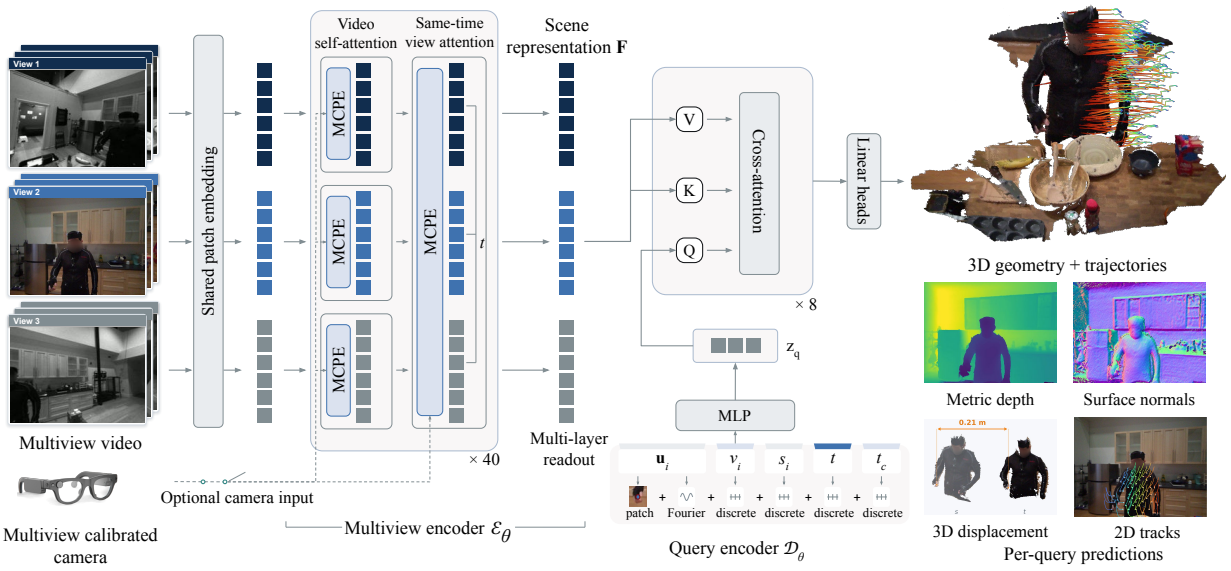

Figure 2 Scale4D architecture. Video self-attention within each view and same-time attention across views encode synchronized videos, with optional MCPE camera conditioning. A multi-layer readout forms the scene representation $\mathbf{F}$. Point–time queries decode metric geometry, trajectories, and auxiliary outputs without re-encoding the video.

UCPE constructs token-dependent operators $\mathbf{D}_a = \mathrm{blkdiag}(\mathbf{D}_a^{\mathrm{ray}}, \mathbf{D}_a^{\mathrm{RoPE}})$, where $\mathbf{D}_a^{\mathrm{ray}}$ applies the world-to-ray transformation $(\mathbf{T}_a^{\mathrm{wr}})^{-1}$ to four-channel geometric feature groups, while $\mathbf{D}_a^{\mathrm{RoPE}}$ encodes spatial positions. The geometric components satisfy

$$\mathbf{D}_a^{\mathrm{ray}}(\mathbf{D}_b^{\mathrm{ray}})^{-1} = \mathbf{I} \otimes \mathbf{G}_{ab}, \tag{2}$$

where $\mathbf{I}$ is an identity matrix over geometric feature groups. The resulting attention is

$$\begin{aligned} A_{ab} &= \mathrm{softmax}_b\left(\frac{\mathbf{q}_a^\top \mathbf{D}_a \mathbf{D}_b^{-1} \mathbf{k}_b}{\sqrt{d}}\right), \\ \mathbf{o}_a &= \mathbf{D}_a \sum_b A_{ab} \mathbf{D}_b^{-1} \mathbf{v}_b, \end{aligned} \tag{3}$$

where A_{ab} is the attention weight from query token a to key token b, $\mathbf{q}_a$, $\mathbf{k}_b$, and $\mathbf{v}_b$ are the query, key, and value of head dimension d, and $\mathbf{o}_a$ is the attention output for token a.

Applying this formulation to metric 4D prediction presents two challenges. First, camera translations in physical units can produce large attention logits, making geometric conditioning numerically unstable. Second, each video token spans two consecutive frames with potentially different camera poses, whereas UCPE associates each token with a single ray frame. We address these challenges through scale-conditioned and dual-frame ray attention (Fig. 3).

Scale-conditioned Attention. As shown in Eq. (2), each four-channel geometric feature group is transformed by the relative ray transformation $\mathbf{G}_{ab} = (\mathbf{R}_{ab}, \mathbf{t}_{ab})$, where $\mathbf{R}_{ab}$ and $\mathbf{t}_{ab}$ denote relative rotation and translation. For a query–key pair $\mathbf{q} = (\mathbf{q}_{xyz}, q_h)$ and $\mathbf{k} = (\mathbf{k}_{xyz}, k_h)$, the contribution to the attention logit in Eq. 3 expands as

$$\begin{aligned} \mathbf{q}^\top \mathbf{G}_{ab} \mathbf{k} = \mathbf{q}_{xyz}^\top \mathbf{R}_{ab} \mathbf{k}_{xyz} \\ + k_h \mathbf{q}_{xyz}^\top \mathbf{t}_{ab} + q_h k_h. \end{aligned}$$

The translation-dependent term $k_h \mathbf{q}_{xyz}^\top \mathbf{t}_{ab}$ scales with camera separation, potentially producing large attention logits when using metric camera poses.

To regulate this dependence while retaining metric scale information, we normalize camera translations and separately encode their original scale. Let $\Delta = \max_{j,k} \|\mathbf{c}_j - \mathbf{c}_k\|_2$ denote the maximum camera separation across all views and times. We define

$$s = f(\Delta / C), \qquad \bar{\mathbf{c}}_j = (\mathbf{c}_j - \mathbf{c}_{\mathrm{ref}})/s, \tag{4}$$

where $f(x) = \max(1, x)$, $C = 20\,\mathrm{m}$, and $\mathbf{c}_{\mathrm{ref}}$ is the first camera center. Camera spans up to C thus keep their metric translations ($s = 1$), while larger spans are scaled down to C. We construct the ray transformations from the normalized camera centers while keeping rotations and intrinsics unchanged.

To retain the original metric scale, we condition the MCPE attention features on the normalization factor through a learned bias-free projection $\Phi : \mathbb{R} \to \mathbb{R}^D$, yielding $\mathbf{h} = \mathrm{LN}_{\mathrm{cam}}(\mathbf{z}) + \Phi(\log s)$. The scale embedding is shared across tokens, and $\mathbf{h}$ is projected into the queries, keys, and values of the MCPE attention

branch. Thus, normalized ray geometry regulates translation-dependent attention logits, while scale-conditioned features retain the original metric scale of the camera layout.

Dual-frame Ray Attention. Each video token jointly represents two consecutive frames, which may have different camera poses. Associating the token with only one ray frame would discard the geometric information of the other frame. We therefore construct two key–value entries from the same token representation, each associated with its corresponding frame's ray geometry. The entries share projection weights, while queries use the first frame's ray coordinate system.

We extend Eq. (3) to two temporal ray frames per token. For token a, let $\mathbf{D}_{a,0}$ and $\mathbf{D}_{a,1}$ denote the geometric operators of its two frames, constructed from normalized camera centers. Both key–value entries share the same scale-conditioned feature $\mathbf{h}_a$ and projection weights, with frame-specific Lat-Up biases added when gravity information is available. Queries use the first frame's ray representation, while attention is jointly normalized over tokens and frames:

$$A_{abr} = \mathrm{softmax}_{b,r}\left(\frac{\mathbf{q}_a^\top \mathbf{D}_{a,0}\mathbf{D}_{b,r}^{-1}\mathbf{k}_{b,r}}{\sqrt{d}}\right),$$

$$\mathbf{o}_a = \mathbf{D}_{a,0}\sum_b\sum_{r=0}^{1} A_{abr}\mathbf{D}_{b,r}^{-1}\mathbf{v}_{b,r},$$

where $r \in \{0,1\}$ indexes the two frames. Restricting attention to $r=0$ recovers the single-frame formulation of Eq. (3).

3.3 Learning Metric Scale and Training

Scale Augmentation. Metric supervision alone does not ensure that the model uses calibrated camera geometry rather than appearance-based scale priors. To encourage geometric scale reasoning, we jointly rescale camera translations and scene geometry while keeping RGB observations unchanged. For a sampled factor $\alpha > 0$, camera centers $\mathbf{c}_j$ and world-space target points $\mathbf{X}$ are scaled about the reference camera center $\mathbf{o} = \mathbf{c}_{\mathrm{ref}}$: $\mathbf{c}_j^\alpha = \mathbf{o} + \alpha(\mathbf{c}_j - \mathbf{o})$ and $\mathbf{X}^\alpha = \mathbf{o} + \alpha(\mathbf{X} - \mathbf{o})$. Camera rotations and intrinsics remain unchanged, preserving image projections while scaling metric 3D positions and displacements by α. Identical RGB observations thus correspond to different metric targets, forcing the model to rely on calibrated camera separations to infer scale. We apply scale augmentation only to camera-conditioned examples with sufficient camera displacement, before camera-span normalization.

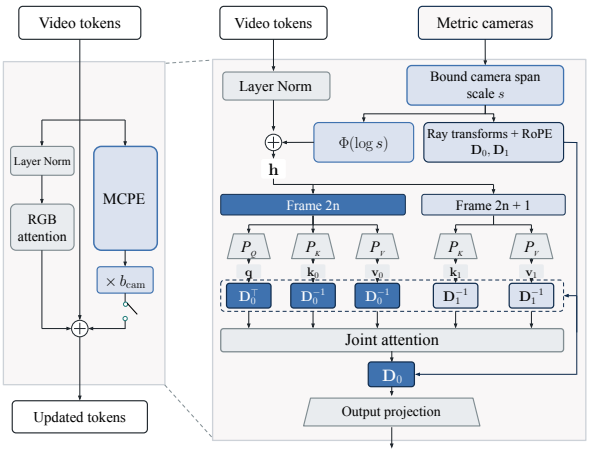

Figure 3 Overview of MCPE integration. MCPE runs beside RGB attention as a residual gated by b_{cam}. Each two-frame token gives one attention query $\mathbf{q}$ and a key–value entry $(\mathbf{k}_r, \mathbf{v}_r)$ per frame r, with shared projections and frame ray transforms $\mathbf{D}_r$, and both entries share one softmax.

Training Objectives. We replace D4RT's depth-normalized position objective with $\mathcal{L}_{\mathrm{3D}}^{\mathrm{metric}}$, a confidence-weighted L_1 loss on metric 3D positions. Unlike depth-normalized objectives, it directly penalizes errors in meters. The total objective combines this loss with auxiliary supervision for 2D positions, visibility, motion, normals, and confidence, as well as multiview reprojection:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\mathrm{3D}}^{\mathrm{metric}} &+ \lambda_{\mathrm{2D}}\mathcal{L}_{\mathrm{2D}} + \lambda_{\mathrm{vis}}\mathcal{L}_{\mathrm{vis}} \\ &+ \lambda_{\mathrm{motion}}\mathcal{L}_{\mathrm{motion}} + \lambda_{\mathrm{normal}}\mathcal{L}_{\mathrm{normal}} \\ &+ \lambda_{\mathrm{conf}}\mathcal{L}_{\mathrm{conf}} + \lambda_{\mathrm{reproj}}\mathcal{L}_{\mathrm{reproj}}.\end{aligned} \tag{5}$$

Sec. 4.2 lists the loss weights. To enforce geometric consistency across calibrated views, we express predicted 3D positions in world coordinates $\hat{\mathbf{X}}_{i,t}^w$, transform them into the target camera coordinate system, and project them into view v at time t:

$$\ell_{\mathrm{reproj}}(i,t,v) = \left\| \Pi_{v,t}\left(\mathbf{R}_{v,t}^{cw}\hat{\mathbf{X}}_{i,t}^w + \mathbf{t}_{v,t}^{cw}\right) - \mathbf{u}_{i,t,v}^* \right\|_1, \tag{6}$$

where $\mathbf{R}_{v,t}^{cw}$ and $\mathbf{t}_{v,t}^{cw}$ define the world-to-camera transformation, $\Pi_{v,t}$ projects camera-space points into image coordinates using the camera intrinsics, and $\mathbf{u}_{i,t,v}^*$ denotes the observed correspondence. Both projected and observed image coordinates are normalized by image size. We average valid reprojection errors separately over the source and other calibrated views, summing the two contributions. Only visible, in-frame correspondences with positive predicted depth are considered, and cross-view errors are clipped to reduce the influence of outliers. Importantly, supervision views need not be encoder

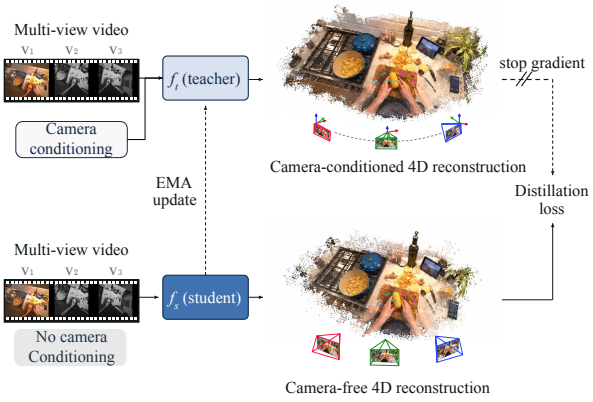

Figure 4 Camera-asymmetry self-training. A camera-conditioned teacher supervises a student without camera conditioning using the same input views. The teacher generates metric trajectory targets for the student, with its weights updated through an exponential moving average (EMA) of the student.

inputs, allowing additional geometric observations to constrain metric predictions during training.

3.4 Camera-Asymmetric Self-Training

The strong metric predictions and flexible camera conditioning of SCALE4D make it a natural foundation for self-training. In particular, camera-conditioned predictions can provide geometric supervision for RGB-only inference on unlabeled calibrated videos. We exploit this capability through an asymmetric teacher–student setup, where the teacher receives camera information while the student operates on the same RGB observations without cameras(Fig. 4).

The teacher and student share the same architecture, pretrained initialization, and RGB observations, but only the teacher receives calibrated camera information. This allows the teacher to provide calibration-informed metric supervision for RGB-only prediction. For shared point–time queries, the student learns from detached teacher predictions of 3D positions, motion, and auxiliary geometric quantities in the same metric coordinate frame, without scale or rigid alignment. Teacher confidence weights valid pseudo-targets in the distillation objective $\mathcal{L}_{\text{distill}}$.

To stabilize continued learning, we interleave distillation on unlabeled videos with supervised replay from labeled data, sampling the two streams with probabilities 0.1 and 0.9, respectively. We update the teacher as an exponential moving average (EMA) of the student: $\theta_{\text{T}} \leftarrow \tau\theta_{\text{T}} + (1-\tau)\theta_{\text{S}}$, where τ is the EMA decay. This enables continued learning from calibrated videos without requiring additional ground-truth 3D trajectories. Sec. 4.6 compares it with symmetric self-training on ADT and HOT3D, evaluating the student with cameras both off and on.

4 Experiments

4.1 Training Data

Our training mixture contains public and internal datasets including Dynamic Replica [26], PointOdyssey [83], FallingThings [62], Kubric [16], TartanAir [69], TartanGround [50], CO3Dv2 [54], GigaHands [14], and Waymo Open [61]. All clips are standardized into synchronized RGB, calibrated cameras, metric 3D trajectories, and validity masks. Appendix A lists the number of samples and frames of each source (Table 5).

Metric scale filtering. Metric supervision is only as reliable as the scale of its labels. For sources with uncertain scale, we rescale each clip using the median ratio between its depth and frozen DA3 metric depth [40]. We measure these ratios at visible points and retain only clips with consistent ratios in log space, filtering clips with unreliable scale estimates out of the metric supervision.

Stereo trajectory annotation. For internal calibrated stereo data, we construct metric 3D trajectories from stereo depth estimated by FoundationStereo [72] and 2D tracks from AllTracker [19]. Stereo provides metric depth but no temporal correspondences, while 2D trackers can drift. Our labeling pipeline therefore retains tracks that agree across multiple anchors and refines their depth estimates.

AllTracker runs forward and backward from evenly spaced anchors. When a track reaches another anchor, tracking restarts there and is checked over the overlapping interval. We retain the longest consistent segment, merge compatible segments across anchors, and record the number of supporting anchors per track.

Static tracks are fitted to a single world point. For dynamic tracks, we consider depth candidates around each tracked pixel to handle foreground- background ambiguity near depth edges, selecting a sequence based on image-space proximity and robust 3D motion smoothness. Finally, we fill missing depths from neighboring frames, discard tracks with insufficient reliable 3D observations or abrupt jumps, and combine tracker visibility with depth-based occlusion checks.

4.2 Training Setup

We train a single model with one to three input views and optional camera conditioning. Each example has

Method	Depth model	ADT [49]			HOT3D [2]			DexYCB [5]			DROID [31]		
		AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑
MVTracker [53]	DA3-Metric [40]	27.20	39.30	89.79	25.90	44.36	75.41	62.19	79.08	84.85	18.67	45.47	57.41
	DA3-Nested [40]	42.13	52.99	92.20	19.69	33.23	80.27	62.79	77.91	86.78	16.24	42.24	56.99
	MoGe-3 [32]	34.84	47.09	90.93	35.74	58.00	73.31	63.94	82.40	83.63	23.80	58.91	58.49
	UniDepth-v2 [52]	39.10	50.92	89.73	34.50	53.24	78.01	60.36	76.60	86.03	16.91	40.96	56.75
OmniX [24]	DA3-Metric	23.27	34.34	82.50	8.41	14.76	81.46	53.77	71.23	82.44	13.08	32.82	49.50
	DA3-Nested	33.37	46.32	82.50	6.85	12.42	81.46	51.38	67.69	82.44	12.78	32.88	49.50
	MoGe-3	27.94	40.42	82.50	7.61	13.40	81.46	56.04	73.83	82.44	14.89	37.59	49.50
	UniDepth-v2	29.77	41.84	82.50	6.69	12.12	81.46	55.14	72.98	82.44	10.43	26.66	49.50
ARROW [13]	DA3-Metric	26.47	36.51	88.89	9.33	15.67	86.74	68.92	81.39	89.16	19.85	41.19	57.67
	DA3-Nested	27.24	35.84	88.89	7.66	13.16	86.74	68.70	79.99	89.16	18.01	38.01	57.67
	MoGe-3	28.70	39.58	88.89	8.71	14.59	86.74	71.39	84.09	89.16	22.21	46.45	57.67
	UniDepth-v2	27.87	36.84	88.89	7.78	13.25	86.74	64.88	75.92	89.16	14.62	30.70	57.67
Ours	—	54.45	66.92	94.94	52.18	70.39	88.15	73.77	83.72	92.92	65.17	98.77	84.12

Table 1 Multi-view metric 3D tracking. All methods use three synchronized views; no ground-truth scale alignment is applied. MVTracker consumes known cameras and the listed metric depth. OmniX and ARROW recover scale from that depth. Our camera-conditioned model uses known cameras and requires no external depth estimator.

Method	ADT [49]			HOT3D [2]			DexYCB [5]			DROID [31]		
	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑
DELTA† [46]	42.62	52.88	93.81	22.09	34.57	87.54	63.81	74.51	90.08	19.51	44.37	68.78
TAPIP3D (RGBD)† [79]	44.39	54.69	93.18	22.22	34.79	86.71	62.92	74.57	89.75	21.44	48.62	69.87
Point4D‡ [23]	38.58	50.40	88.09	7.80	13.58	85.48	62.03	74.93	88.66	20.92	47.37	59.08
AllTracker † [19]	44.51	54.17	95.02	23.05	34.53	89.42	61.61	74.21	90.38	23.68	47.40	67.09
CoTracker3 † [28]	44.13	52.99	95.16	22.37	32.20	88.97	63.45	73.11	91.59	22.74	34.84	75.63
Track4World§ [43]	33.28	43.79	95.14	9.72	16.79	88.65	63.17	74.77	91.49	23.82	49.28	70.94
4RC‡ [44]	29.37	42.73	79.04	16.79	28.10	79.70	58.62	76.35	82.49	11.29	28.19	40.02
OmniX‡ [24]	34.89	48.44	82.91	6.02	10.63	83.28	47.34	63.60	83.12	17.88	43.66	49.20
SpaTracker-v2‡* [74]	43.44	52.75	94.86	24.04	35.64	88.80	60.38	72.90	89.33	23.74	49.81	72.48
Any4D‡ [29]	27.83	38.03	89.03	22.80	34.47	83.60	53.35	68.75	84.62	9.07	24.86	45.70
π^4 [70]	40.87	51.59	93.76	27.97	40.03	88.39	34.80	43.63	89.28	27.43	57.25	62.21
OpenD4RT‡ [47, 80]	22.79	32.49	90.81	6.20	13.18	77.21	57.35	69.42	90.00	19.86	42.49	67.19
Ours	52.21	64.89	94.67	51.74	69.87	89.03	65.72	76.10	92.19	62.53	97.04	83.34

Table 2 Single-view metric 3D tracking. No ground-truth scale alignment. Ours uses known intrinsics and metric poses without external depth. DA3-Nested [40] supplies external depth: † input/lifting; ‡ scale recovery. §: embedded DA3; ∗: test-time optimization.

48 frames per view at 224×224 resolution. Camera conditioning is enabled with probability 0.5, independently of the view count, using a shared gate across all MCPE branches. During training, one source view is selected per example, and the camera reference time t_c equals the target time t with probability 0.4. At evaluation, we use $t_c = t$.

We apply geometry-preserving scale augmentation when MCPE is enabled and at least one encoded camera center is at least 5 cm from the first. In these cases, we sample $\alpha \sim \mathcal{U}(0.8, 1.2)$; otherwise, $\alpha = 1$. We scale camera translations and metric 3D supervision while preserving RGB observations and the 2D image correspondences. The same RGB frames thus appear with metric targets that are up to 20% larger or smaller.

We train Scale4D for 500k steps using 16 H100 GPU nodes, with a batch size of 4 per GPU. The learning rate peaks at 10^{-4} for both encoder and decoder after 1k warmup steps, followed by cosine decay to 10^{-6}. For the training objective in Eq. 5, we set $\lambda_{2D} = \lambda_{vis} = \lambda_{motion} = \lambda_{reproj} = 0.1$, $\lambda_{conf} = 0.2$, and $\lambda_{normal} = 0.5$. The resulting checkpoint is used for all main evaluations and supports every input configuration with a single set of weights, without any additional test-time optimization.

4.3 3D Tracking Evaluation

Evaluation protocol. We evaluate metric 3D tracking on ADT [49], HOT3D [2], DexYCB [5], and DROID [31], with 50 clips each from ADT, HOT3D, and DROID. Following TAPVid-3D [35], we report 3D Average Jaccard (AJ), Average Percentage of Points within Delta (APD_{3D}), and Occlusion Accuracy (OA). AJ and APD_{3D} average over fixed metric thresholds of 5, 10, 20, and 40 cm, while OA measures visibility prediction accuracy. Unless otherwise stated, predictions are evaluated directly in metric space without ground-truth scale alignment.

ADT follows the benchmark query times and target-view pixels [34]. DexYCB uses the published query times and target-camera-visible anchors. Its 24–25-frame clips are repeat-padded to 48 frames

Method	ADT [49]			HOT3D [2]			DexYCB [5]			DROID [31]		
	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑
DELTA† [46]	47.93	63.54	92.23	40.36	56.34	87.14	75.09	88.71	88.22	30.66	63.37	68.37
TAPIP3D (RGBD)† [79]	49.41	64.33	93.61	37.79	54.33	86.17	72.59	87.73	87.47	32.99	68.26	68.82
Point4D‡ [23]	59.66	77.56	88.09	9.53	16.25	85.48	78.08	92.68	88.66	31.94	70.78	59.08
AllTracker † [19]	49.35	63.80	95.02	38.61	54.29	89.42	73.29	87.73	90.38	35.41	68.71	67.09
CoTracker3 † [28]	49.27	62.45	95.16	38.24	51.42	88.97	76.01	86.45	91.59	34.46	53.30	75.63
Track4World§ [43]	55.78	69.51	95.14	9.39	16.07	88.65	76.02	88.53	91.49	32.25	66.31	70.94
4RC‡ [44]	46.90	67.14	79.04	25.83	41.04	79.70	73.42	93.32	82.49	20.26	53.84	40.02
OmniX‡ [24]	53.46	74.75	82.91	7.42	12.93	83.28	62.25	82.34	83.12	27.58	68.06	49.20
SpaTracker-v2‡* [74]	67.12	80.40	94.86	47.38	64.84	88.80	73.79	87.91	89.33	36.03	73.52	72.48
Any4D‡ [29]	43.74	59.88	89.03	40.47	57.92	83.60	68.30	87.62	84.62	18.35	50.30	45.70
OpenD4RT [47, 80]	31.04	44.21	90.81	7.98	16.86	77.21	70.51	84.97	90.00	28.43	58.91	67.19
Ours (cam off)	65.74	80.03	94.57	58.01	74.89	89.22	81.66	92.63	92.42	63.63	97.73	83.43
Ours (cam on)	67.08	81.18	94.67	52.72	70.86	89.03	83.33	93.98	92.19	63.44	97.86	83.34

Table 3 Scale-aligned single-view 3D tracking. One median scale is fitted to ground truth per clip; visibility is unchanged. Camera off disables MCPE while retaining intrinsics for UV-depth lifting. Symbols and colors follow Table 2.

for methods requiring fixed-length inputs, with padded frames excluded from evaluation. HOT3D and DROID use fixed, visually diverse 48-frame subsets. HOT3D trajectories track annotated hand and object mesh vertices, with visibility determined by mesh occlusion, calibrated image bounds, and arm/hand segmentation masks in rectified pinhole views. DROID trajectories track sampled robot surfaces using recorded forward kinematics, with visibility determined by mesh occlusion and SAM3 [4] foreground masks. Queries begin at the first valid observation in the reference camera (HOT3D RGB and DROID exterior cam1), with at most 512 queries per DROID clip. Both datasets use identical frozen anchor queries across single- and multi-view evaluation. Ground-truth annotations define query locations and evaluation trajectories, but ground-truth depth is never provided as a model input. APD_{3D} is evaluated on ground-truth-visible points, and clips are weighted equally within each dataset.

Our model receives camera information only through MCPE. For evaluation, intrinsics are used to lift predicted image positions and depth into 3D, while benchmark camera poses transform predictions into a common coordinate frame, including when MCPE is disabled. These transformations are applied only during evaluation and do not provide camera information to the model when MCPE is disabled.

Baselines requiring metric depth use DA3-Metric, DA3-Nested [40], MoGe-3 [32], or UniDepth-v2 [52], all given known intrinsics. Depth-assisted baselines use DA3-Nested, which additionally receives metric poses for moving-camera sequences, unless noted. The symbols in Tables 1 and 2 indicate the use of external depth, while Table 6 summarizes the inputs available to each depth provider. Table 7 in the appendix repeats the single-view comparison with each of the four depth providers. TAPIP3D RGBD additionally receives known intrinsics, poses, and estimated depth. DELTA and the AllTracker and CoTracker3 lifting pipelines use known intrinsics, while Point4D and OmniX estimate camera geometry. On HOT3D, our evaluation adapter supplies known intrinsics and poses to the DA3 component of Track4World, although these inputs are not supported by its released interface. Tables 2 and 3 also include OpenD4RT [47], an unofficial implementation of D4RT [80], evaluated using its public 48-frame checkpoint `OpenD4RT_48CLIP_9Mix_NoCropAUG`.

Multi-view metric results. With three calibrated views and no external depth, SCALE4D achieves the highest metric AJ on all four benchmarks against every tracker and depth provider (Table 1), with the largest margins on HOT3D (+16.4 AJ) and DROID (+41.4 AJ) over the strongest baseline.

Single-view metric results. With the same weights and a single calibrated video, SCALE4D also achieves the highest AJ on all four benchmarks (Table 2), dropping less than 3 AJ points from the three-view setting on HOT3D and DROID. The margin is smallest on DexYCB (1.9 AJ), whose static cameras provide no motion cue for scale.

Scale-aligned results. Metric tracking performance reflects both global scale accuracy and residual 3D reconstruction and tracking errors. Table 3 evaluates performance after fitting a single median scale factor per clip. This corrects global scale differences but does not eliminate local geometric distortions, trajectory errors, or differences in method inputs. With or without camera conditioning, SCALE4D outperforms all baselines in AJ on HOT3D, DexYCB, and DROID. On ADT, it matches SpaTracker-v2, which uses test-time optimization, in AJ. Camera-conditioned SCALE4D achieves the highest APD_{3D} across all four benchmarks, indicating that its advantages extend beyond metric scale estimation to relative 3D geometry and motion.

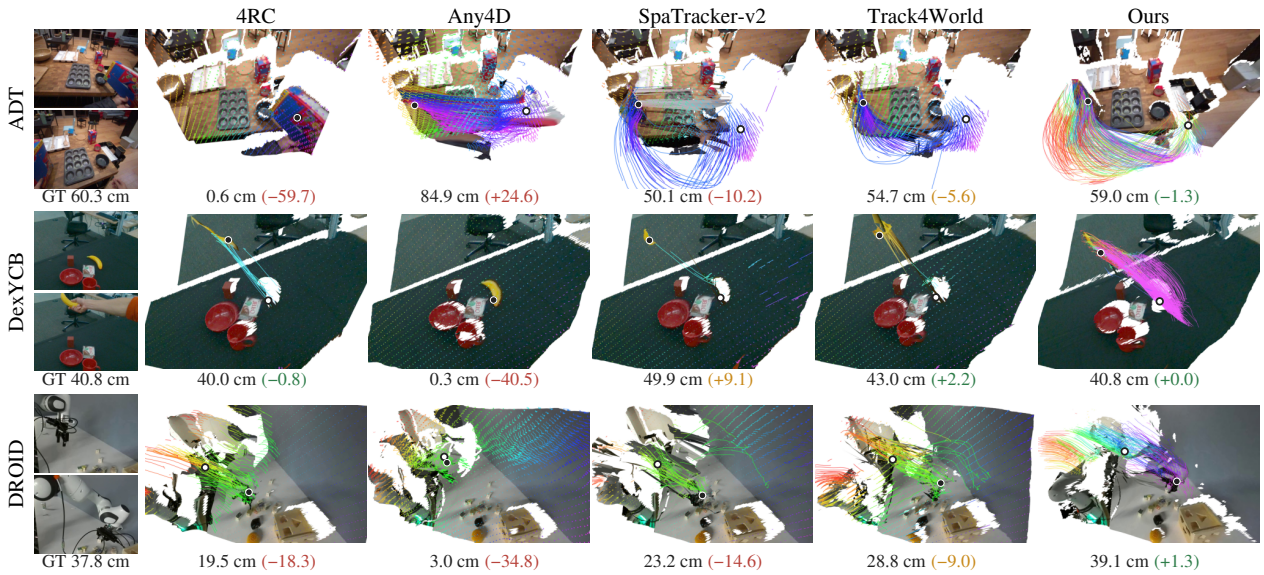

Figure 5 Metric 4D tracking results across methods. Each row shows the first and last input frames of a clip and its last frame from one viewpoint near the input camera, shared by all methods. Below each panel: the moved object's travel from the first to the last frame and its error in cm, with ground truth (GT) under the input frames. 4RC, Any4D and SpaTracker-v2 take their metric scale from DA3-Metric [40]; Track4World uses its embedded DA3.

Method	Cam.	ScanNet [7]			Bonn [48]		
		AbsRel ↓	$\delta_{1.25}$ ↑	TAE ↓	AbsRel ↓	$\delta_{1.25}$ ↑	TAE ↓
DA3NESTED [40]	off	0.062	0.970	1.00	0.137	0.902	2.43
MapAnything [30]	off	0.317	0.450	1.14	0.218	0.760	2.62
Any4D [29]	off	0.340	0.233	1.24	0.445	0.018	2.65
Ours	off	0.083	0.945	0.68	0.133	0.895	1.76
Ours	on	0.041	0.979	0.62	0.107	0.967	1.80

Table 4 Metric video depth. Camera on uses known intrinsics, metric poses, and world up; off uses RGB only as input. No GT scale alignment. TAE is reported in percent.

4.4 Dense Metric Depth

Beyond 3D tracking, SCALE4D directly predicts dense metric depth without task-specific fine-tuning. We evaluate on ScanNet [7] and Bonn [48] using AbsRel, $\delta_{1.25}$, and temporal alignment error (TAE) [75], which measures depth disagreement between adjacent frames after pose-based warping. All metrics are computed on the first 48 frames of each sequence without ground-truth scale or shift alignment.

Camera inputs for each method are summarized in Table 4. We exclude two ScanNet sequences due to inaccurate camera poses. DA3-Nested uses its unconditioned video-depth interface. TAE uses ground-truth depth only to define validity masks. For Bonn, both of our configurations use the published marker-to-optical-camera transformation.

Results. As shown in Table 4, RGB-only inference outperforms MapAnything and Any4D in both AbsRel and $\delta_{1.25}$. Camera conditioning further improves depth accuracy, achieving the best AbsRel and $\delta_{1.25}$ on both datasets. Both configurations also outperform all baselines in TAE, showing strong temporal consistency of the predicted depth.

4.5 Qualitative Results

Fig. 5 compares metric 4D tracking across methods from shared novel viewpoints. Baselines exhibit static-point drift and ray-like depth errors, while SCALE4D maintains coherent geometry and motion, with object displacement errors within 1.5 cm across all three examples. Fig. 6 and 7 further demonstrate SCALE4D on diverse real-world scenarios, including dynamic outdoor scenes, human-object interactions, and robotic manipulation. Across these settings, SCALE4D reconstructs coherent 3D geometry and tracks fast and complex motion. Fig. 20 in Appendix C shows eight more in-the-wild recordings.

4.6 Camera-Asymmetric Self-Improvement

We investigate whether SCALE4D can leverage its stronger camera-conditioned predictions to improve RGB-only inference through self-training (Fig. 8). We evaluate on the egocentric ADT and HOT3D benchmarks, starting all runs from a shared checkpoint.

Self-training setup. As a control, symmetric self-training uses the same recipe, including initialization,

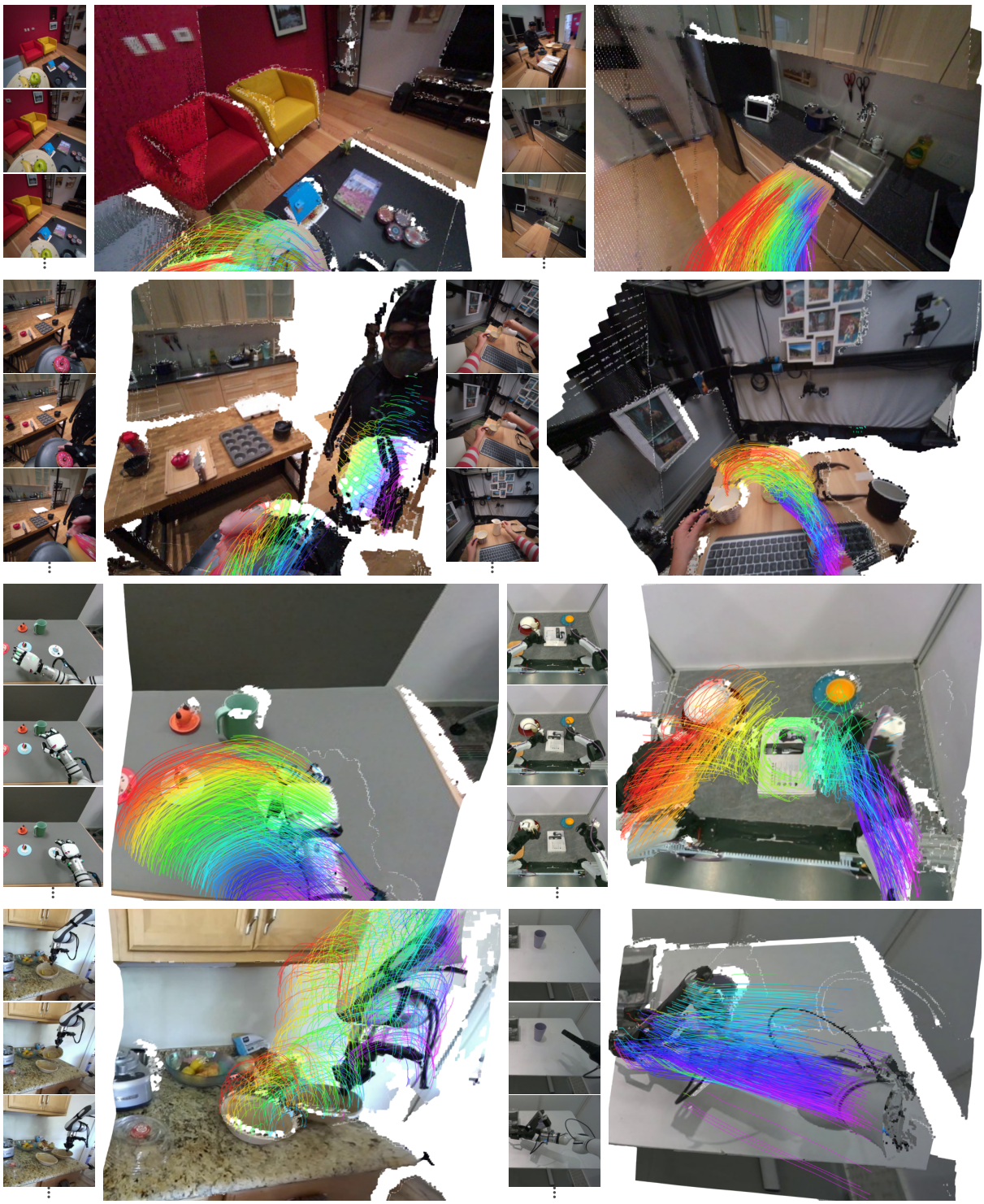

Figure 6 Qualitative results on egocentric and robot videos. Each example shows the first, middle, and last input frames and our prediction at the last frame: the reconstructed surface and the trajectories of moving points, colored by pixel position, seen slightly off the input camera. Rows 1–2: egocentric ADT and HOT3D videos with known cameras (one view; three for the right ADT example in row 1). Rows 3–4: robot manipulation; VITRA, ABC, and RealSource use one RGB view without camera input, and DROID uses two calibrated views.

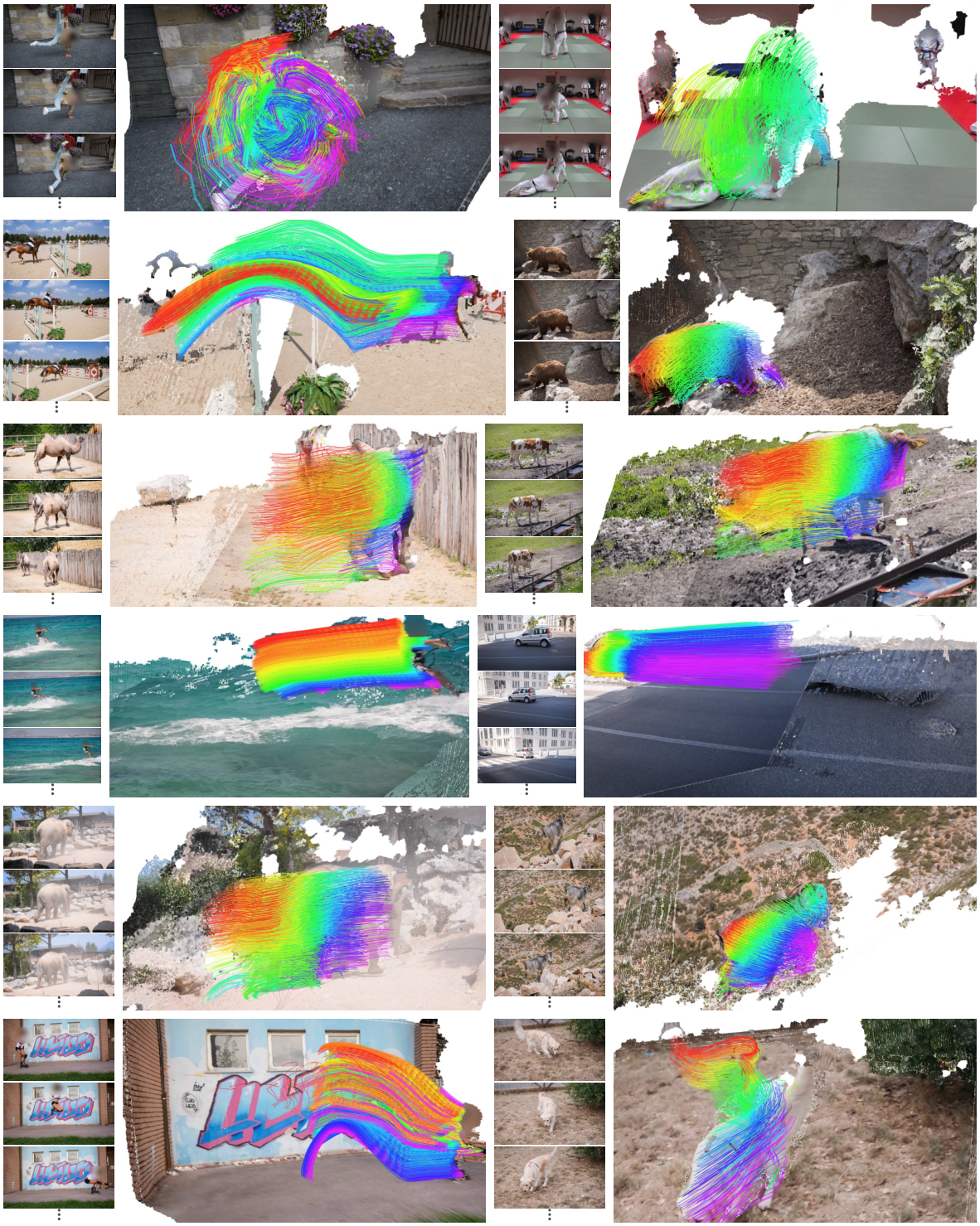

Figure 7 Results on DAVIS. One RGB video with no camera input; SCALE4D estimates the focal length and camera motion. Each example shows the first, middle, and last input frames and our prediction, rendered as in Figure 6. Faces are blurred.

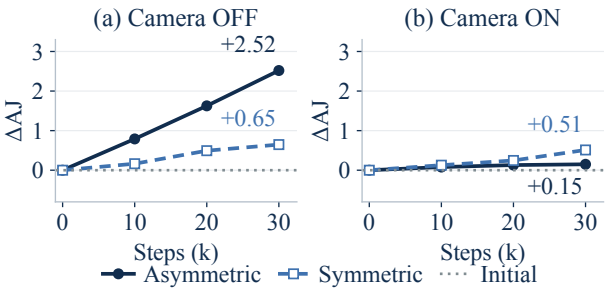

Figure 8 Camera-Asymmetric Self-Improvement. AJ gain over initialization (ΔAJ, no scale alignment), averaged over ADT and HOT3D with three views and EMA weights, under (a) camera-off and (b) camera-on inference. The teacher is camera-conditioned (asymmetric) or camera-off (symmetric).

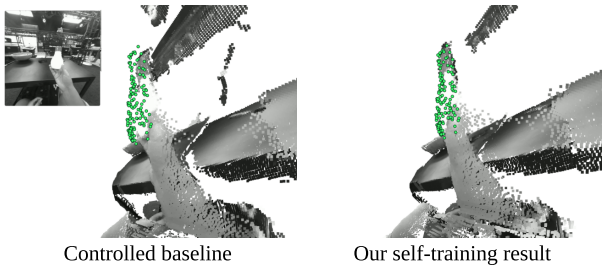

Figure 9 Camera-asymmetric self-training visualization. Left: symmetric controlled baseline; right: our camera-asymmetric self-training result. Green points mark ground truth.

unlabeled videos, replay, and EMA teacher, but disables camera conditioning in the teacher, so teacher and student receive identical inputs. This comparison isolates the effect of camera conditioning in the teacher, the only difference between the two runs. Appendix B gives the fine-tuning details.

Results. A camera-conditioned teacher provides more effective supervision for a camera-free student than symmetric camera-off self-training. With camera-off inference, the advantage grows during training (Fig. 8a), yielding a mean AJ improvement of 2.52 points over initialization after 30k steps, compared with 0.65 points for symmetric self-training. Between 20k and 30k steps, symmetric self-training adds only 0.2 AJ, while the asymmetric student adds 0.9 AJ. Improvements on both ADT (+1.29 AJ) and HOT3D (+3.75 AJ) demonstrate the benefit of transferring calibration-informed geometric knowledge to RGB-only prediction. These gains do not come at the expense of camera-conditioned performance, which stays within 0.15 AJ of the initial model (Fig. 8b). Fig. 9 shows a qualitative example on a Show3D clip, where both students run on RGB only and see the same frames; green points mark the ground-truth

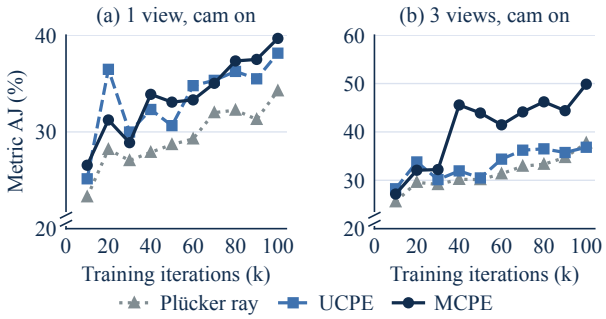

Figure 10 Camera encoding during training. Panels (a,b) use one and three input views, respectively. Cameras are on at evaluation; all runs train with 50% camera conditioning.

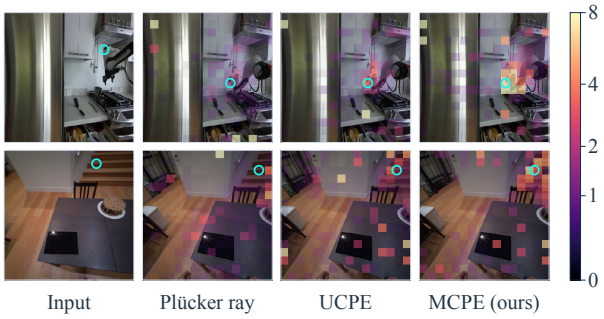

Figure 11 Attention under different camera encodings. Block-40 attention maps for two queries using Plücker-ray embeddings, UCPE, and MCPE, with cameras enabled. Cyan circles mark queries and ground-truth correspondences. Attention is averaged over 16 heads and normalized by uniform attention (logarithmic color scale).

positions of the object.

5 Design Ablations

5.1 Geometric Conditioning and Metric Scale

Encoding approaches. We compare additive Plücker-ray embeddings with geometric attention (UCPE and MCPE) across different view counts and camera-conditioning settings. Fig. 10 shows that all three approaches achieve comparable performance without camera conditioning, while geometric attention, particularly MCPE, provides substantially higher metric AJ when cameras are available. The advantage is especially pronounced with three views, highlighting MCPE's ability to exploit calibrated geometry without compromising RGB-only inference. This motivates our design of separate RGB and camera attention branches for flexible camera conditioning. The attention maps in Fig. 11 further illustrate how the different encodings attend to

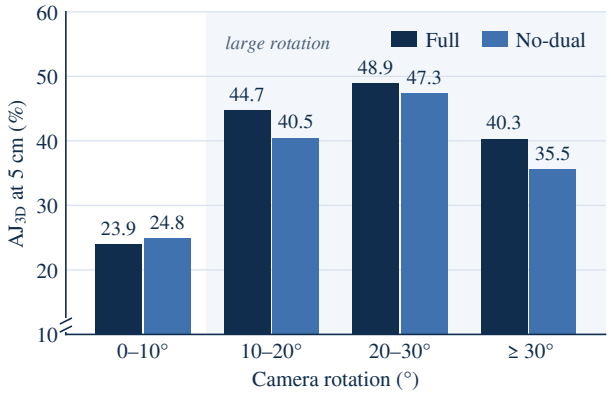

Figure 12 Dual-frame rays under camera rotation. Whole-clip AJ at 5 cm, grouped by the maximum within-pair camera rotation across all views. Full outperforms No-dual on clips with maximum rotations of at least 10°, with the largest gain at ≥ 30° (+4.8 AJ). The shaded region indicates rotations of at least 10°.

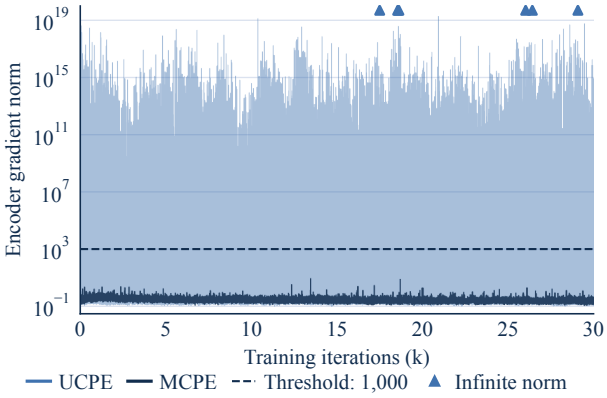

Figure 13 Encoder gradient norms during training. Raw pre-clipping norms on a logarithmic axis. The dashed line marks 10^3, triangles denote infinite UCPE norms.

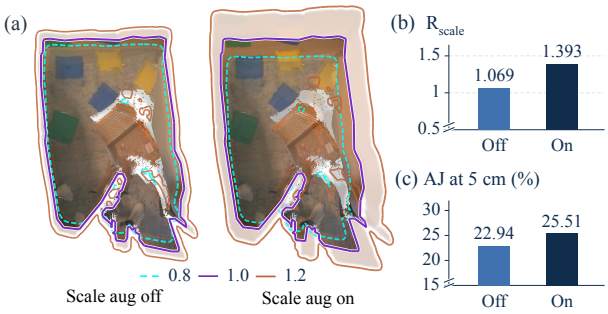

Figure 14 Learning Metric Scale from Cameras. (a) Predicted geometry under camera-baseline scaling ($\alpha = 0.8, 1.0, 1.2$) with fixed RGB. (b) Mean depth response ratio R_{scale}. (c) AJ at 5 cm.

ground-truth correspondences.

Dual-frame rays under camera rotation. We compare the Full and No-dual models after 50k fine-tuning updates from a shared initialization, using three views with camera conditioning. All clips are grouped by the maximum camera rotation within the original encoder frame pairs $(0, 1), (2, 3), \ldots$, across all three views. We preserve the original frame sampling and queries and evaluate whole-clip AJ at a 5 cm threshold, weighting clips equally within each bin. As shown in Fig. 12, Full improves tracking on clips with within-pair rotations of at least 10°, with the largest gain at ≥ 30° (40.30 vs. 35.54 AJ). This suggests that representing both temporal ray frames is particularly beneficial when camera rotation produces substantial changes in viewing direction within an encoder token.

Training stability. Fig. 13 compares raw encoder gradient norms over aligned 30k-iteration windows. Uncapped UCPE exhibits frequent large gradient spikes and occasional infinite norms, whereas MCPE maintains substantially smaller and more stable gradients. This comparison illustrates the improved numerical stability of span normalization and scale conditioning.

Learning metric scale from cameras. Fig. 14 examines whether SCALE4D uses camera baselines as a metric scale cue. With RGB observations and queries fixed, we rescale camera translations by $\alpha \in \{0.8, 1.0, 1.2\}$ and examine the resulting geometry (Fig. 14a). We quantify scale sensitivity by the ratio R_{scale} of mean predicted depths at $\alpha = 1.2$ and $\alpha = 0.8$. A proportional response yields $R_{\text{scale}} = 1.5$, whereas scale invariance yields 1. Averaged equally across four benchmarks, scale augmentation increases R_{scale} from 1.069 to 1.393, bringing the response closer to the expected proportional scaling (Fig. 14b). Scale augmentation also improves AJ at a 5 cm threshold from 22.94 to 25.51 (Fig. 14c).

Geometric Measurements. Fig. 15 evaluates the benefits of additional measurements at inference using identical model weights. Increasing input views from one to three improves metric AJ by 9.4 points and scale-aligned AJ by 2.4 points, indicating improved 3D localization beyond scale correction. Enabling camera conditioning improves metric AJ by 20.1 points but scale-aligned AJ by only 2.8 points, highlighting the role of calibrated cameras in recovering metric scale. Qualitative results further illustrate improved geometry and trajectory accuracy with richer geometric inputs.

5.2 Zero-Shot Inference Beyond Three Views

Although SCALE4D is trained with at most three views, its architecture supports a variable number of inputs. We evaluate zero-shot generalization to up

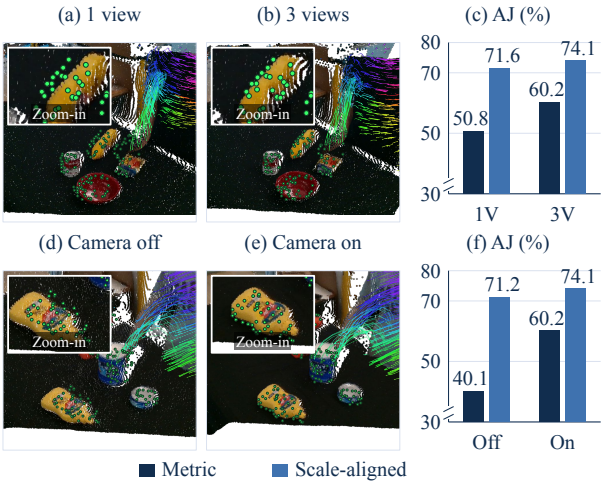

Figure 15 Benefits of Geometric Measurements. Additional views (a-c) and camera conditioning (d-f) improve metric 3D tracking. Green markers indicate ground truth.

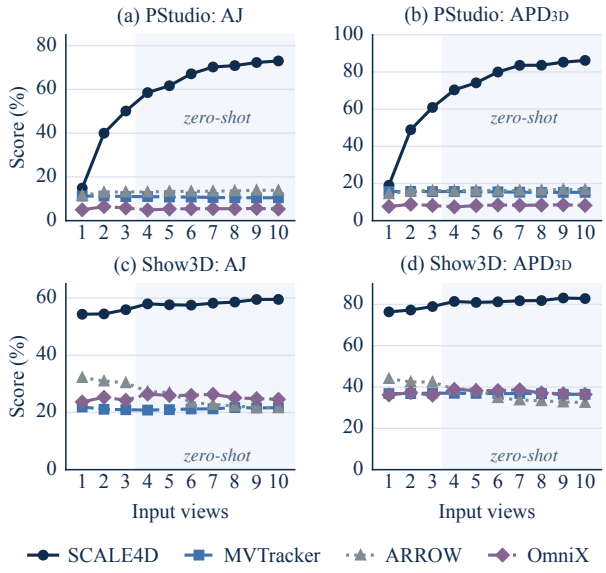

Figure 16 Zero-shot generalization to additional views. Metric AJ and APD_{3D} on PStudio and Show3D using identical clips and queries. SCALE4D is trained with at most three views; shaded views 4–10 are unseen during training. SCALE4D continues to improve beyond three views, while baselines show limited gains. DA3-Nested provides depth to MVTracker and metric scale to ARROW and OmniX.

to ten views using the frozen checkpoint with camera conditioning on PStudio [25] and Show3D [55].

Inference extension. The encoder processes a variable number of views without learned view-index embeddings. For additional views, we extend the runtime view-count limits and reuse the decoder's learned view embeddings, keeping the source-view embedding fixed and alternating the two auxiliary-view embeddings. All selected views are jointly encoded, and the decoder attends to their combined features without changing model weights or performing any additional test-time optimization.

Evaluation protocol. We evaluate 20 clips each from PStudio and Show3D, using the same frozen clips and queries across methods and view counts. The target camera remains in slot zero. Each setting uses the first N cameras from a fixed ten-camera list: the target followed by its nine nearest camera centers for PStudio, and cam1–cam10 for Show3D. For SCALE4D, we use 48 frames at 224×224 resolution and identical target-camera queries at every view count from one to ten.

Multi-view baselines. We compare against released checkpoints of MVTracker [53], ARROW [13], and OmniX [24], using identical clips, camera subsets, and queries at one through ten views. MVTracker receives RGB, calibration, and DA3-Nested metric depth [40]. ARROW and OmniX estimate camera geometry from RGB and recover metric scale using DA3-Nested target-view depth. All methods retain their native image preprocessing. For OmniX, we extend the view-count limit without changing model weights and restrict motion queries to target-camera source frames and output times, while retaining all camera features for cross-attention. This enables 48-frame inference within 80 GB without reducing image resolution. A three-view, four-frame check reproduces the original predictions exactly. Depth estimates are reused across view counts, and no method uses ground-truth depth or trajectories for scale fitting.

Results. Fig. 16 shows that SCALE4D improves consistently from one to three views and continues to benefit from additional views beyond its training limit. From three to ten views, PStudio gains 22.91 AJ and 25.29 APD_{3D} points, while Show3D gains 3.60 AJ and 3.86 APD_{3D} points. At ten views, SCALE4D outperforms all three baselines on both datasets in metric AJ and APD_{3D}. The baselines show limited improvement with additional views on PStudio; on Show3D, ARROW degrades while MVTracker and OmniX remain nearly unchanged. These results demonstrate that SCALE4D can exploit additional calibrated observations beyond the view counts encountered during training, without updating its weights. Fig. 17 shows two qualitative examples.

5.3 Training Recipe Ablations

Pseudo-labeled training data. Our training mixture includes egocentric stereo pseudo-labeled clips (Sec. 4.1). Fig. 18 compares the full mixture with

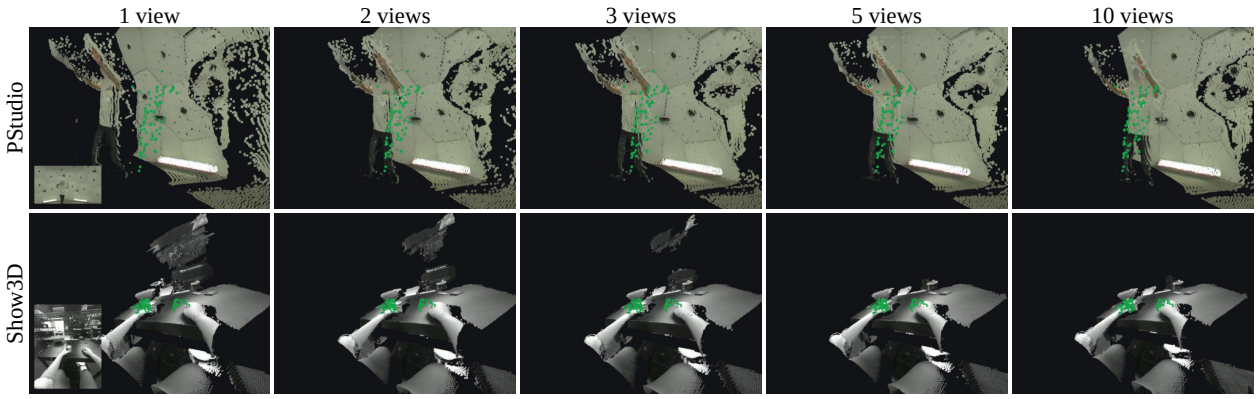

Figure 17 Qualitative results beyond three views. Predictions from 1 to 10 input views; counts above three are unseen in training. Green dots are ground truth. More views move the PStudio subject onto the ground truth and remove the floating background in Show3D. Examples chosen for visualization.

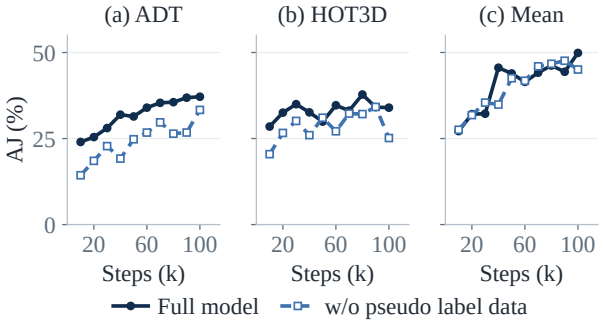

Figure 18 Ablation on pseudo-labeled training data. Three-view, camera-conditioned metric AJ with and without pseudo-labeled training data. (a) ADT, (b) HOT3D, and (c) the equally weighted mean across ADT, HOT3D, DexYCB, and DROID.

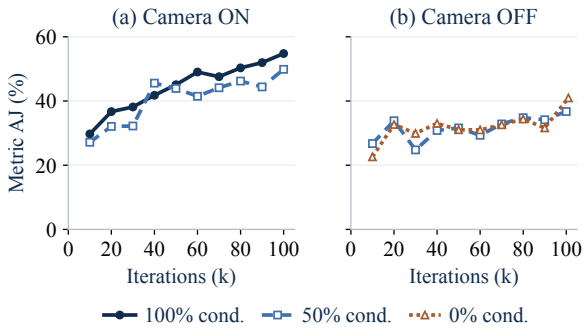

Figure 19 Ablation on camera-conditioning frequency. Three-view metric AJ averaged over all test clips. (a) Camera-on inference for models trained with 50% or 100% camera conditioning. (b) Camera-off inference for models trained with 0%, 50%, or 100% camera conditioning.

a variant excluding these clips, while keeping the remaining training recipe unchanged. At 100k iterations, the full mixture improves metric AJ from 33.32 to 37.14 on ADT and from 25.15 to 33.97 on HOT3D, while the four-dataset mean remains broadly comparable across checkpoints. These results suggest that egocentric pseudo-labeled data particularly benefits benchmarks with similar visual domains, without substantially changing overall performance.

Camera-conditioning frequency. Fig. 19 compares models trained with camera conditioning enabled for 0%, 50%, or 100% of examples. With cameras enabled at inference, 100% conditioning achieves the highest AJ, while 50% conditioning remains competitive. With cameras disabled, 50% conditioning achieves 36.73 AJ at 100k iterations, comparable to the 0%-conditioned model. These results show that camera dropout enables a single model to support both inference settings, balancing camera-conditioned accuracy with camera-free performance. We use 50% conditioning for the final model (Sec. 4.2).

6 Conclusion

We introduce SCALE4D, a unified feedforward model for metric 4D perception across diverse sensing configurations. By effectively leveraging geometric measurements, SCALE4D delivers accurate metric geometry and motion predictions with flexible camera conditioning. Despite being trained with at most three views, SCALE4D generalizes zero-shot to larger numbers of views and continues to benefit from additional observations at inference. We further demonstrate that its flexible camera conditioning enables asymmetric self-training, where camera-conditioned predictions improve RGB-only inference on unlabeled videos. Together, these results establish SCALE4D as a scalable and flexible foundation for metric 4D perception and future self-training research on unlabeled calibrated videos.

References

[1] Abhishek Badki, Hang Su, Bowen Wen, and Orazio Gallo. L4P: Towards unified low-level 4D vision perception. In *International Conference on 3D Vision (3DV)*, 2026.

[2] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. Introducing HOT3D: An Egocentric Dataset for 3D Hand and Object Tracking. *arXiv preprint arXiv:2406.09598*, 2024.

[3] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32, 2019.

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, et al. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations (ICLR)*, 2026.

[5] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[6] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3R: Estimating Disentangled Motion from DUSt3R Without Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9158–9168, 2025.

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

[8] Youming Deng, Songyou Peng, Junyi Zhang, Kathryn Heal, Tiancheng Sun, John Flynn, Steve Marschner, and Lucy Chai. Selfi: Self-improving reconstruction engine via 3d geometric feature alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7351–7361, 2026.

[9] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10061–10072, 2023.

[10] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, Joao Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Asian Conference on Computer Vision*, pages 483–500. Springer, 2024.

[11] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, et al. Project aria: A new tool for egocentric multi-modal AI research. *arXiv preprint arXiv:2308.13561*, 2023.

[12] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8503–8513, 2025.

[13] Ilya Fradlin, Christian Schmidt, Jens Piekenbrinck, Karim Knaebel, Gonzalo Martin Garcia, and Bastian Leibe. ARROW: Arbitrary reconstruction and tracking of 4d observations in the wild. *arXiv preprint arXiv:2610.01314*, 2026.

[14] Rao Fu, Dingxi Zhang, Alex Jiang, Wanjia Fu, Austin Funk, Daniel Ritchie, and Srinath Sridhar. GigaHands: A massive annotated dataset of bimanual hand activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17461–17474, 2025.

[15] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019.

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[17] Jisang Han, Honggyu An, Jaewoo Jung, Takuya Narihira, Junyoung Seo, Kazumi Fukuda, Chaehyun Kim, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. Enhancing 3D reconstruction for dynamic scenes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[18] Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories. *arXiv preprint arXiv:2204.04153*, 2022.

[19] Adam W. Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, Rares Ambrus, Katerina Fragkiadaki, and Leonidas Guibas. AllTracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5253–5262, 2025.

[20] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

[21] Nan Huang, Pengcheng Yu, Weijia Zeng, James M Rehg, Angjoo Kanazawa, Haiwen Feng, and Qianqian Wang. Self-improving 4d perception via self-distillation. *arXiv preprint arXiv:2604.08532*, 2026.

[22] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3R: Empowering unconstrained 3D reconstruction with camera and scene priors. In *Proc. CVPR*, pages 1071–1081, 2025.

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

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

[25] 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 *Proceedings of the IEEE international conference on computer vision*, pages 3334–3342, 2015.

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

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

[28] 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 *Proc. ICCV*, 2025.

[29] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4D: Unified feed-forward metric 4D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14578–14589, 2026.

[30] 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 *International Conference on 3D Vision*, 2026.

[31] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *RSS*, 2024.

[32] Lingyu Kong, Ruicheng Li, Ruicheng Wang, Sicheng Xu, Chengtang Yao, Jianfeng Xiang, and Jiaolong Yang. MoGe-3: Fine-Detail Monocular Geometry Estimation with Self-Guided Sparse Volumetric Refinement. *arXiv preprint arXiv:2607.17967*, 2026.

[33] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20932–20941, 2026.

[34] Skanda Koppula, Ignacio Rocco, Yi Yang, Joe Heyward, Joao Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. Tapvid-3d: A benchmark for tracking any point in 3d. *Advances in Neural Information Processing Systems*, 37:82149–82165, 2024.

[35] 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 *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

[36] Jiahui Lei, Yijia Weng, Adam W. Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic Gaussian Fusion from Casual Videos via 4D Motion Scaffolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6165–6177, 2025.

[37] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding Image Matching in 3D with MASt3R. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[38] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems*, 2025.

[39] Zhengqi Li, Richard Tucker, Forrester Cole, Qian-

qian 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10486–10496, 2025.

[40] 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. *arXiv preprint arXiv:2511.10647*, 2025.

[41] Xinhang Liu, Yuxi Xiao, Donny Y. Chen, Jiashi Feng, Yu-Wing Tai, Chi-Keung Tang, and Bingyi Kang. Trace Anything: Representing Any Video in 4D via Trajectory Fields. In *International Conference on Learning Representations (ICLR)*, 2026.

[42] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3R: Aligned Monocular Depth Estimation for Dynamic Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22820–22830, 2025.

[43] Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4World: Feedforward world-centric dense 3D tracking of all pixels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

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

[45] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations*, 2024.

[46] 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 *International Conference on Learning Representations (ICLR)*, 2025.

[47] OpenD4RT contributors. OpenD4RT: An unofficial PyTorch/GPU implementation of D4RT. https://github.com/Lijiaxin0111/Open-d4rt, 2026.

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

[49] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20133–20143, 2023.

[50] Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. TartanGround: A large-scale dataset for ground robot perception and navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[51] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal Monocular Metric Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10106–10116, 2024.

[52] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal Monocular Metric Depth Estimation Made Simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3): 2354–2367, 2026.

[53] 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.

[54] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[55] Patrick Rim, Kevin Harris, Braden Copple, Shangchen Han, Xu Xie, Ivan Shugurov, Sizhe An, He Wen, Alex Wong, Tomas Hodan, et al. Show3d: Capturing scenes of 3d hands and objects in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7111–7120, 2026.

[56] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[57] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision*, 2016.

[58] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Frédo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Advances in Neural Information Processing Systems*, 2021.

[59] Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic Point Maps: A versatile

representation for dynamic 3d reconstruction. In *Proc. ICCV*, 2025.

[60] Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-DPM: 4D Video Reconstruction with Dynamic Point Maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14502–14511, 2026.

[61] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurélien 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[62] Jonathan Tremblay, Thang To, and Stan Birchfield. Falling things: A synthetic dataset for 3D object detection and pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2038–2041, 2018.

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

[64] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9660–9672, 2025.

[65] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10510–10522, 2025.

[66] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details. *arXiv preprint arXiv:2507.02546*, 2025.

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

[68] Shizun Wang, Zhenxiang Jiang, Xingyi Yang, and Xinchao Wang. C4D: 4D Made from 3D through Dual Correspondences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7570–7580, 2025.

[69] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[70] Yifan Wang, Weiguang Zhao, Ni Yang, Haoyu Guo, Junting Dong, Weinan Zhang, and Chunhua Shen. π^4: A foundation model for 4D scene geometry and motion. *arXiv preprint arXiv:2611.23144*, 2026.

[71] 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 *Proc. ICLR*, 2026.

[72] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. FoundationStereo: Zero-shot stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5249–5260, 2025.

[73] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. SpatialTracker: Tracking any 2D pixels in 3D space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20406–20417, 2024.

[74] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6726–6737, 2025.

[75] Honghui Yang, Di Huang, Wei Yin, Chunhua Shen, Haifeng Liu, Xiaofei He, Binbin Lin, Wanli Ouyang, and Tong He. Depth Any Video with Scalable Synthetic Data. In *International Conference on Learning Representations (ICLR)*, 2025.

[76] 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. *Proc. CVPR*, 2025.

[77] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10371–10381. IEEE, 2024.

[78] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3D: Towards Zero-shot Metric 3D Prediction from A Single Image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9043–9053, 2023.

[79] Bowei Zhang, Lei Ke, Adam W. Harley, and Ka-

terina Fragkiadaki. TAPIP3D: Tracking any point in persistent 3D geometry. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[80] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[81] Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Cruz Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[82] 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 *Proc. ICLR*, 2025.

[83] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[84] Fengjun Zhong, Congjia Chen, Zhaoxu Liu, Jinyang Du, Yuchen Gong, Enqi Mao, Ruihao Gong, ShuJie Wang, Xianglong Liu, and Zhongliang Qiao. Beyond relative geometry: Metric-aware geometry perception for robotics. *arXiv preprint arXiv:2608.27497*, 2026.

Appendix

A Training Data

Table 5 summarizes the public training sources of Sec. 4.1: the number of training samples and video frames contributed by each dataset. Sources with uncertain scale are rescaled and filtered as described there. Our internal stereo clips, labeled with the pipeline of Sec. 4.1, are not listed.

Data source	Training samples	Video frames
Aggregated datasets with annotations		
Dynamic Replica [26]	483	144,900
PointOdyssey [83]	8,191	1,048,448
FallingThings [62]	1,013	14,120
Kubric [16]	13,862	782,640
CO3Dv2 [54]	5,905	1,182,963
TartanAir [69]	1,098	275,887
GigaHands [14]	2,818	589,399
TartanGround [50]	1,467	397,961
Waymo Open [61]	732	145,012

Table 5 Training data statistics. Breakdown of the number of training samples and video frames across public datasets.

B Experimental Details

Camera-asymmetric self-training (Sec. 4.6). A camera-conditioned teacher supervises a camera-free student using unlabeled calibrated videos with labeled replay. Fine-tuning uses 2 H100 GPU nodes with a global batch size of 64. We use cosine learning-rate schedules from 10^{-6} to 2×10^{-7} for the encoder and from 10^{-5} to 2×10^{-6} for other parameters.

Multi-view baselines (Sec. 5.2). MVTracker uses RGB, calibration, and DA3-Nested metric depth [40]. ARROW and OmniX estimate camera geometry from RGB, with metric scale recovered using DA3-Nested target-view depth. All methods retain their native image preprocessing. For OmniX, we extend the view-count limit without changing model weights and restrict motion queries to target-camera source frames and output times, while retaining all camera features for cross-attention. This enables 48-frame inference within 80 GB without reducing image resolution. A three-view, four-frame check reproduces the original predictions exactly. Depth estimates are cached and reused across view counts, and no method uses ground-truth depth or trajectories for scale fitting.

Robustness to depth providers. To assess sensitivity to external depth estimates, Table 7 compares DA3-Nested with DA3-Metric, MoGe-3, and UniDepth-v2 for each applicable baseline, while our model remains unchanged. π^4 (native) uses its predicted scale without depth-provider rescaling. SCALE4D achieves the highest AJ on ADT, HOT3D, and DROID across all tested depth providers. On DexYCB, the strongest baseline configuration uses MoGe-3 depth, indicating greater sensitivity to the choice of depth provider. Table 6 lists the camera inputs that each depth provider receives.

Depth provider	Known intrinsics	Known metric poses
DA3-Metric [40]	Focal conversion to meters	Not supplied
DA3-Nested [40]	Network input	Supplied for translating camera streams
MoGe-3 [32]	Horizontal field of view	Not supplied
UniDepth-v2 [52]	Intrinsics matrix input	Not supplied

Table 6 Camera inputs to the tracking depth providers, in addition to each tracker's own camera interface. DA3-Nested omits poses for static cameras such as DexYCB's.

C Additional Qualitative Results

Fig. 20 shows eight more in-the-wild recordings of people and robots, complementing Figs. 6 and 7. Each example shows input frames of the query view and our reconstruction with dense 3D trajectories, rendered from a novel viewpoint.

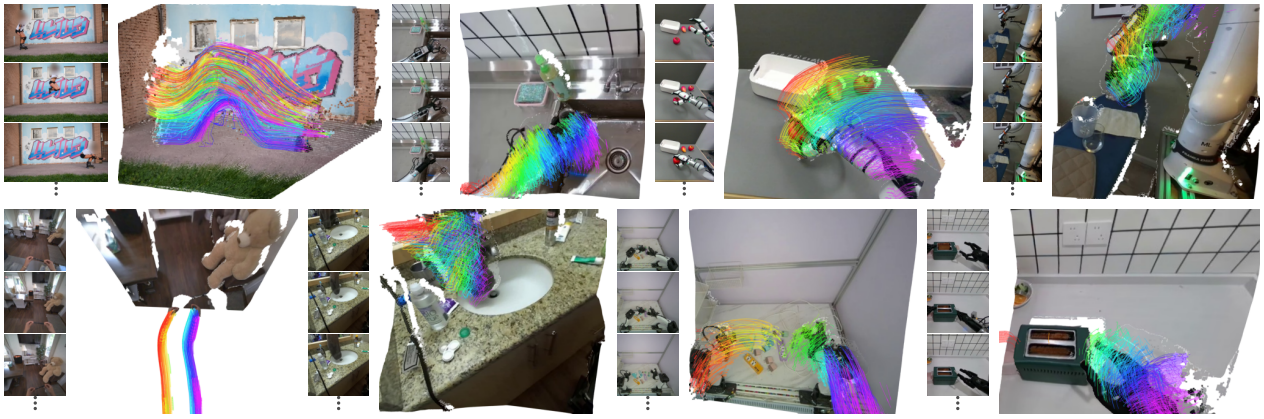

Figure 20 In-the-wild 4D metric perception. Eight more recordings of people and robots, seen from novel viewpoints.

Method	Depth model	ADT [49]			HOT3D [2]			DexYCB [5]			DROID [31]		
		AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑
DELTA† [46]	DA3-Metric [40]	28.55	39.97	92.23	32.00	48.38	87.14	64.81	78.60	88.22	23.92	50.28	68.37
	DA3-Nested [40]	42.62	52.88	93.81	22.09	34.57	87.54	63.81	74.51	90.08	19.51	44.37	68.78
	MoGe-3 [32]	36.73	48.23	92.87	45.72	62.18	87.23	67.89	80.80	90.09	33.79	67.14	70.83
	UniDepth-v2 [52]	41.44	52.57	91.70	40.42	56.51	87.11	66.89	79.36	88.69	20.26	41.66	68.74
TAPIP3D (RGBD)† [79]	DA3-Metric	29.14	40.41	93.61	30.36	47.04	86.17	63.53	78.68	87.47	26.62	55.95	68.82
	DA3-Nested	44.39	54.69	93.18	22.22	34.79	86.71	62.92	74.57	89.75	21.44	48.62	69.87
	MoGe-3	37.49	48.46	94.10	44.23	61.41	85.93	67.45	81.54	90.05	37.22	71.23	72.93
	UniDepth-v2	42.46	53.75	93.04	39.59	56.41	85.96	61.49	75.35	88.42	21.76	44.49	70.61
Point4D‡ [23]	DA3-Metric	29.46	39.86	88.09	9.54	16.35	85.48	65.50	78.93	88.66	22.99	51.75	59.08
	DA3-Nested	38.58	50.40	88.09	7.80	13.58	85.48	62.03	74.93	88.66	20.92	47.37	59.08
	MoGe-3	33.45	45.37	88.09	8.86	15.11	85.48	67.43	81.05	88.66	26.77	58.74	59.08
	UniDepth-v2	36.38	48.38	88.09	8.21	14.23	85.48	59.67	72.11	88.66	16.57	37.25	59.08
AllTracker† [19]	DA3-Metric	29.43	40.30	95.02	31.66	46.87	89.42	64.61	78.95	90.38	28.39	55.62	67.09
	DA3-Nested	44.51	54.17	95.02	23.05	34.53	89.42	61.61	74.21	90.38	23.68	47.40	67.09
	MoGe-3	37.41	48.05	95.02	45.67	61.23	89.42	67.11	81.19	90.38	35.05	67.71	67.09
	UniDepth-v2	42.93	53.51	95.02	40.89	56.38	89.42	61.05	75.11	90.38	22.11	43.89	67.09
CoTracker3† [28]	DA3-Metric	29.39	39.52	95.16	30.67	43.72	88.97	66.53	77.70	91.59	27.17	40.71	75.63
	DA3-Nested	44.13	52.99	95.16	22.37	32.20	88.97	63.45	73.11	91.59	22.74	34.84	75.63
	MoGe-3	37.36	47.05	95.16	44.83	57.87	88.97	68.60	79.47	91.59	34.55	50.69	75.63
	UniDepth-v2	42.58	52.46	95.16	40.31	53.27	88.97	63.17	74.22	91.59	21.52	32.78	75.63
4RC‡ [44]	DA3-Metric	24.96	37.80	79.04	21.49	35.54	79.70	62.37	80.88	82.49	12.01	30.70	40.02
	DA3-Nested	29.37	42.73	79.04	16.79	28.10	79.70	58.62	76.35	82.49	11.29	28.19	40.02
	MoGe-3	28.62	43.02	79.04	21.89	35.04	79.70	64.50	83.47	82.49	13.85	33.75	40.02
	UniDepth-v2	27.76	40.35	79.04	18.85	30.29	79.70	59.08	76.66	82.49	8.09	20.15	40.02
OmniX‡ [24]	DA3-Metric	24.77	36.28	82.91	7.44	12.90	83.28	50.90	67.74	83.12	18.53	45.54	49.20
	DA3-Nested	34.89	48.44	82.91	6.02	10.63	83.28	47.34	63.60	83.12	17.88	43.66	49.20
	MoGe-3	29.28	42.74	82.91	6.98	12.03	83.28	54.79	73.26	83.12	22.15	53.67	49.20
	UniDepth-v2	30.21	42.56	82.91	6.08	10.66	83.28	53.72	71.61	83.12	14.00	34.96	49.20
SpaTracker-v2‡* [74]	DA3-Metric	32.58	42.12	94.86	36.77	53.10	88.80	64.32	77.61	89.33	26.33	55.17	72.48
	DA3-Nested	43.44	52.75	94.86	24.04	35.64	88.80	60.38	72.90	89.33	23.74	49.81	72.48
	MoGe-3	36.29	47.20	94.86	43.56	60.71	88.80	66.96	81.31	89.33	31.33	64.90	72.48
	UniDepth-v2	41.55	51.17	94.86	38.95	54.58	88.80	65.22	79.03	89.33	20.09	42.84	72.48
Any4D‡ [29]	DA3-Metric	25.08	35.75	89.03	30.77	46.49	83.60	56.71	73.38	84.62	9.63	25.98	45.70
	DA3-Nested	27.83	38.03	89.03	22.80	34.47	83.60	53.35	68.75	84.62	9.07	24.86	45.70
	MoGe-3	26.87	38.64	89.03	32.32	46.86	83.60	60.66	78.60	84.62	11.12	29.27	45.70
	UniDepth-v2	30.08	40.52	89.03	29.12	42.17	83.60	60.23	78.33	84.62	6.32	16.79	45.70
$\pi^{4\ddagger}$ [70]	Native	40.87	51.59	93.76	27.97	40.03	88.39	34.80	43.63	89.28	27.43	57.25	62.21
	DA3-Metric	36.13	45.78	93.76	37.17	52.98	88.39	61.91	74.73	89.28	26.94	57.69	62.21
	DA3-Nested	43.91	54.46	93.76	25.00	36.39	88.39	58.50	70.73	89.28	23.15	50.09	62.21
	MoGe-3	39.27	50.14	93.76	42.88	59.54	88.39	67.50	82.10	89.28	31.11	65.49	62.21
	UniDepth-v2	40.73	51.13	93.76	33.63	49.10	88.39	63.20	77.86	89.28	20.44	44.85	62.21
OpenD4RT‡ [47, 80]	DA3-Metric	19.76	29.12	90.81	7.62	15.61	77.21	61.10	73.81	90.00	22.38	46.98	67.19
	DA3-Nested	22.79	32.49	90.81	6.20	13.18	77.21	57.35	69.42	90.00	19.86	42.49	67.19
	MoGe-3	22.73	33.02	90.81	7.59	15.62	77.21	62.01	75.50	90.00	25.02	52.88	67.19
	UniDepth-v2	21.84	31.59	90.81	6.88	14.56	77.21	59.16	72.05	90.00	18.35	39.41	67.19
Ours (cam on)	None	52.21	64.89	94.67	51.74	69.87	89.03	65.72	76.10	92.19	62.53	97.04	83.34

Table 7 Additional depth providers for single-view metric tracking. Pi4 (native) retains its predicted scale without external depth. DA3-Nested reference results from Table 2 are repeated for comparison; the embedded-DA3 reference is reported there. Our camera-on model receives known intrinsics and metric poses through MCPE and uses no external depth. Symbols follow Table 2. Camera inputs are listed in Table 6. Shading ranks the best, second, and third configurations shown here, with ties at displayed precision sharing a rank.