GeoMVT: Rethinking Multi-View Point Tracking through Geometric Reasoning

Anonymous authors
Paper under double-blind review

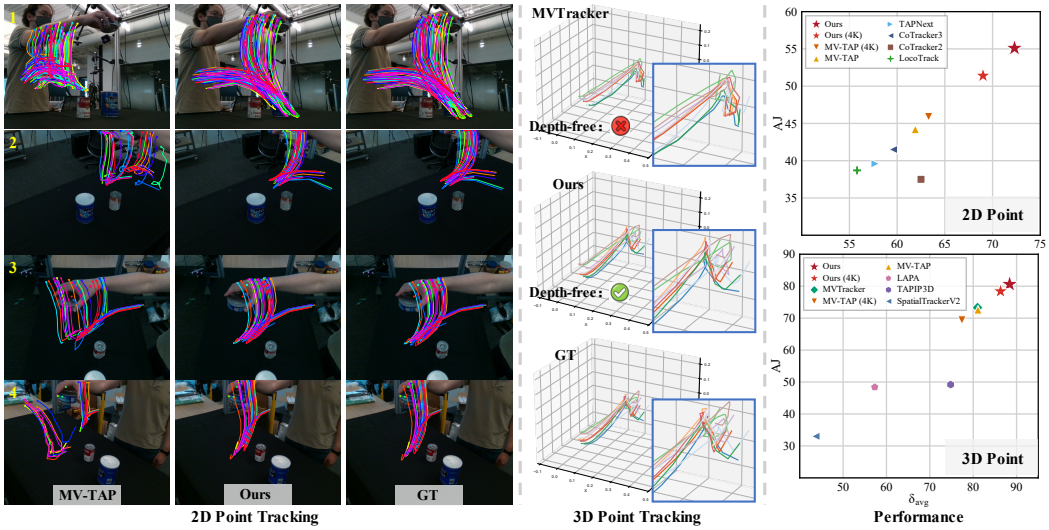


Figure 1: We propose **GeoMVT**, the first depth-free multi-view point tracker for accurate 2D and 3D point tracking. Our method leverages calibrated 3D camera geometry to bridge image-space 2D tracking and 3D motion recovery. We visualize comparisons against MV-TAP (Koo et al., 2026) for 2D tracking and MVTracker (Rajič et al., 2025) for 3D on DexYCB (Chao et al., 2021). Without depth input, GeoMVT substantially outperforms the baselines in both 2D and 3D point tracking.

Abstract

We introduce GeoMVT, a depth-free multi-view point tracker for accurate 2D tracking and reliable 3D motion recovery. Our key idea is to leverage calibrated 3D camera geometry to bridge multi-view 2D tracking and accurate 3D motion recovery by guiding cross-view interactions during trajectory estimation. To this end, we propose a geometry-guided cross-view reasoning framework with two complementary stages: correlation aggregation and point-state updates. We first design epipolar-guided geometric transform attention (EpiGTA) to aggregate local correlation features through relative geometric alignment and adaptive epipolar guidance. Building on these geometry-enhanced features, a geometry-guided transformer combines spatio-temporal context with geometrically aligned cross-view interactions. Together, the two stages form a recurrent loop, where geometry-enhanced correlations guide refinement and updated trajectories determine subsequent correlation sampling and geometric compatibility. We construct a synthetic multi-view dataset of 12K sequences using Kubric for training. Experiments on four standard benchmarks demonstrate substantial improvements over existing multi-view 2D trackers and effective use of additional views across different camera configurations. Moreover, simple triangulation yields competitive 3D tracking, surpassing dedicated 3D trackers on multiple real-world benchmarks without depth input.

1 Introduction

Tracking arbitrary points provides long-range correspondences for robot interaction (Vecerik et al., 2024; Bharadhwaj et al., 2024) and dynamic scene reconstruction (Wang et al., 2024; Feng et al., 2025). Recent 2D trackers have improved the accuracy and temporal consistency of image-space trajectories (Cho et al., 2024; Karaev et al., 2025; Qu et al., 2026). Yet some applications require

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.

spatial motion that cannot be fully determined from individual image projections (Zhang et al., 2025). Recent 3D trackers address this ambiguity by lifting observations with depth (Xiao et al., 2024; Ngo et al., 2025; Zhang et al., 2025) or jointly estimating scene geometry and motion (Xiao et al., 2025). However, these approaches exploit spatial structure but remain tied to the quality of provided or estimated scene geometry, motivating depth-free alternatives for accurate 3D motion recovery.

Calibrated multi-view videos offer such an alternative by observing the same motion from different viewpoints. Reliable 3D recovery therefore requires 2D trajectories that are both accurate within each view and geometrically consistent across views (Hartley & Zisserman, 2003). Existing methods mainly exploit multi-view information by constructing depth-lifted 3D representations (Rajič et al., 2025; Chang et al., 2026) or fusing independently estimated per-view tracks for 3D reconstruction (Galoaa et al., 2026). MV-TAP (Koo et al., 2026) takes a further step by introducing cross-view interaction into synchronized multi-view 2D tracking. Nevertheless, calibrated 3D camera geometry mainly enters as an auxiliary representation cue, rather than directly guiding cross-view interaction throughout iterative trajectory estimation. As a result, appearance variations across wide baselines and asymmetric occlusions can still hinder the effective exchange of complementary observations across views. Consequently, the resulting 2D trajectories may lack sufficient localization accuracy and cross-view consistency for reliable 3D recovery. This points to a more direct role for camera geometry: structuring cross-view interaction during 2D trajectory estimation. In this paper, we therefore ask: *Can calibrated 3D camera geometry directly guide cross-view interactions to bridge accurate multi-view 2D tracking and reliable 3D motion recovery without depth input?*

To this end, we introduce **GeoMVT**, a **Geo**metry-guided cross-view reasoning framework for **M**ulti-**V**iew point **T**racking. Our framework leverages calibrated 3D camera geometry to directly guide cross-view information exchange while retaining trajectory estimation in image space. Specifically, this geometric guidance spans two complementary stages: cross-view correlation aggregation and iterative point-state updates. First, we design epipolar-guided geometric transform attention (EpiGTA) to aggregate encoded local correlation features across cameras. By embedding relative geometric transformations into attention, EpiGTA aligns feature comparison and aggregation across camera frames. Meanwhile, adaptive epipolar guidance modulates these interactions according to the compatibility of current track hypotheses, downweighting geometrically inconsistent support. Building on the resulting geometry-enhanced correlation features, a geometry-guided transformer extends this alignment principle to synchronized point-state interactions. It integrates the aggregated features into point states and combines geometrically guided cross-view exchange with temporal and spatial context for trajectory refinement. In turn, refined positions determine correlation sampling and geometric compatibility in the next iteration, recurrently coupling the two stages. This coupling encourages both accurate per-view localization and cross-view geometric consistency, enabling 3D trajectory recovery through simple triangulation (Hartley & Zisserman, 2003) without depth input.

To support large-scale training, we generate 12K synthetic multi-view training sequences using Kubric (Greff et al., 2022). We evaluate both 2D and 3D tracking on DexYCB (Chao et al., 2021), Panoptic Studio (Joo et al., 2015), Multi-View Kubric (Greff et al., 2022), and Harmony4D (Khirodkar et al., 2024). Across all four benchmarks, GeoMVT substantially outperforms existing methods in multi-view 2D tracking. Moreover, experiments with different camera numbers and layouts show that geometric guidance enables more effective use of complementary observations. Crucially, the improvements extend beyond image-space accuracy. Simple triangulation of the predicted 2D trajectories yields competitive 3D tracking, surpassing dedicated 3D trackers on multiple real-world benchmarks without depth input. These results demonstrate that calibrated geometry can connect accurate image-space tracking with reliable 3D motion recovery. We will release our code, models, and training data to facilitate future research on multi-view point tracking.

2 Related Work

2D Point Tracking. Recent 2D trackers estimate long-range point trajectories beyond pairwise optical flow (Harley et al., 2022; Doersch et al., 2023; Li et al., 2024b;a; Qu et al., 2026; Zholus et al., 2025; Aydemir et al., 2026; Jung et al., 2026). PIPs (Harley et al., 2022) formulates tracking as iterative trajectory refinement driven by correlation features, while TAPIR (Doersch et al., 2023) combines per-frame matching with temporal refinement. Subsequent methods improve spatial context (Karaev et al., 2024), local matching (Cho et al., 2024), and training with real videos (Karaev et al., 2025; Kim et al., 2026). However, these methods primarily target monocular videos. MV-TAP (Koo et al.,

2026) introduces cross-view interaction into multi-view 2D point tracking, but calibrated geometry remains an auxiliary cue rather than directly guiding correlation exchange.

3D Point Tracking. 3D point tracking directly models spatial motion. SpatialTracker (Xiao et al., 2024) incorporates depth through a triplane representation for 3D tracking. TAPIP3D (Zhang et al., 2025) tracks points in persistent 3D feature clouds, and DELTA (Ngo et al., 2025) predicts dense trajectories in UVD space. Other methods jointly estimate geometry and motion (Xiao et al., 2025) or derive dense trajectories from reconstructed pointmaps (Nam et al., 2026). For multi-view 3D tracking, Dynamic 3D Gaussians (Luiten et al., 2024) builds on persistent reconstruction, whereas MVTracker (Rajič et al., 2025) operates in a fused 3D feature point cloud. However, these methods are generally sensitive to the quality of depth or reconstructed 3D geometry. LAPA (Galoaa et al., 2026) avoids external depth by volumetrically fusing independently estimated per-view tracks, but confines multi-view reasoning to 3D reconstruction rather than refining the underlying 2D trajectories.

Geometric Cross-View Reasoning. Camera geometry structures feature interaction and correspondence selection across views. For 3D detection, PETR (Liu et al., 2022) embeds 3D coordinates into image features, and DETR3D (Wang et al., 2022) projects 3D queries to retrieve relevant visual features. GTA (Miyato et al., 2024) uses relative geometric transformations to align cross-view feature interaction, while recent camera encodings (Li et al., 2025; Xie et al., 2026; Wu et al., 2026) model relative and projective geometry to reduce dependence on arbitrary camera coordinates. Complementarily, epipolar constraints reduce correspondence ambiguity in human pose estimation (He et al., 2020), object pose estimation (Chen et al., 2025), and novel-view synthesis (Suhail et al., 2022a;b). Despite these advances, how geometry can evolve from a representational cue into dynamic guidance for cross-view interactions throughout trajectory estimation remains largely unexplored in multi-view point tracking.

3 METHOD

Problem formulation. Given synchronized RGB videos $\mathcal{I} = \{I_{v,t} \in \mathbb{R}^{H\times W\times 3}\}_{v=1,t=1}^{V,T}$ from V calibrated cameras, we track N scene points in the image plane of each camera. Following MV-TAP (Koo et al., 2026), we specify point n in each view v by a query $q_n = (\tau_n^q, x_n^q, y_n^q)$, with τ_n^q denoting the query frame index. Queries sharing index n across views refer to the same physical point but may occur at different frames. We predict 2D trajectories $P \in \mathbb{R}^{V\times T\times N\times 2}$, together with visibility probabilities $\widehat{V}$ and confidence scores $\widehat{C}$, both of size $V \times T \times N$. Camera calibration provides the intrinsics K_v and extrinsics $T = (R_v, t_v)$.

GeoMVT overview. Fig. 2 presents our geometry-guided tracking framework, GeoMVT. Given synchronized videos, calibrated cameras, and query points, we first extract image features with a shared CNN encoder. At each iteration, local correlations at current track estimates are encoded into view-local correlation features. EpiGTA aggregates these features through geometric alignment and epipolar guidance (Sec. 3.1). Then, a geometry-guided transformer combines them with tracking states to update positions, visibility, and confidence (Sec. 3.2). In turn, updated positions determine correlation sampling and geometric relations in the next iteration, coupling cross-view correlation aggregation with point-state updates. The final 2D trajectories are triangulated into 3D (Sec. 3.3).

3.1 CROSS-VIEW CORRELATION AGGREGATION

We compute per-frame feature maps using a shared CNN encoder with stride $k = 4$ and construct an S-level feature pyramid $F_{v,t}^s$ through successive average pooling (Karaev et al., 2025), where $S = 4$. These feature maps are computed once and reused throughout iterative refinement to construct view-local correlation features for geometry-guided cross-view aggregation.

Local 4D correlation features. For point n, we bilinearly sample a square query neighbourhood f_q^s around (x_n^q, y_n^q) from $F_{\tau_n^q}^s$. At each iteration m, we sample its track counterpart f_t^s from F_t^s around $P_{t,n}^{(m)} = (x_{t,n}, y_{t,n})$. We construct a local 4D correlation tensor (Cho et al., 2024) to capture pairwise similarities between these neighbourhoods. We then use a shared two-layer MLP ϕ_{corr} to encode this tensor into a view-local correlation feature for subsequent cross-view aggregation:

$$C_{t,n}^s = \left\{ \left\langle f_q^s(i), f_t^s(j) \right\rangle \;\middle|\; i \in \Omega_q,\ j \in \Omega_p \right\}, \qquad e_{t,n}^s = \phi_{\text{corr}}(C_{t,n}^s) \in \mathbb{R}^d. \tag{1}$$

Here Ω_q and Ω_p index the square query and track neighbourhoods. We repeat this computation independently for each view v and scale s to obtain view-local correlation features $e_{v,t,n}^s$.

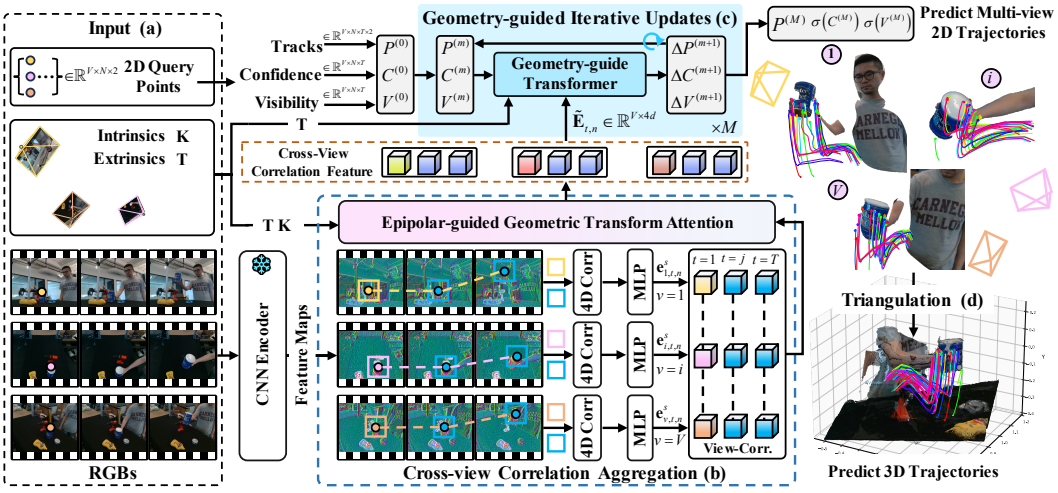


Figure 2: **GeoMVT pipeline.** (a) Given synchronized multi-view RGB videos and camera calibration, we first extract per-view feature maps using a shared CNN encoder. (b) We encode local correlations as view-local correlation features and aggregate them using EpiGTA with geometric alignment and epipolar guidance (Sec. 3.1). (c) The cross-view correlation features are combined with tracking states into point-state tokens, which guide iterative updates of positions, visibility, and confidence through a geometry-guided transformer (Sec. 3.2). (d) At inference, we triangulate the final 2D trajectories into 3D trajectories using weights derived from predicted visibility and confidence (Sec. 3.3).

Epipolar-guided geometric transform attention. View-local correlation features $e^s_{v,t,n}$ preserve within-camera matching without direct cross-view appearance comparisons, but provide limited evidence when a point is occluded or poorly observed. We therefore introduce epipolar-guided geometric transform attention (EpiGTA), which leverages calibrated 3D camera geometry to aggregate complementary correlation features through relative geometric alignment and adaptive epipolar guidance. As illustrated in Fig. 3, given a target-view correlation feature $e^s_{v,t,n}$ and source-view features $e^s_{u,t,n}$, we first apply layer normalization and project them into query, key, and value features $q^s_v, k^s_u, v^s_u \in \mathbb{R}^{d_h}$ for each attention head. Let ρ_v encode the calibrated camera pose and current 2D track location in view v. The relative transformation $\rho_v \rho_u^{-1}$ expresses source-view attention features in the target-view representation. The projected features are transformed as $\bar{q}^s_v = \rho_v^\top q^s_v$, $\bar{k}^s_u = \rho_u^{-1} k^s_u$, and $\bar{v}^s_u = \rho_u^{-1} v^s_u$, enabling cross-view attention over geometrically aligned representations. Following GTA (Miyato et al., 2024), ρ incorporates camera extrinsics and image coordinates through $\mathrm{SE}(3)$, $\mathrm{SO}(3)$, and $\mathrm{SO}(2)$ representations.

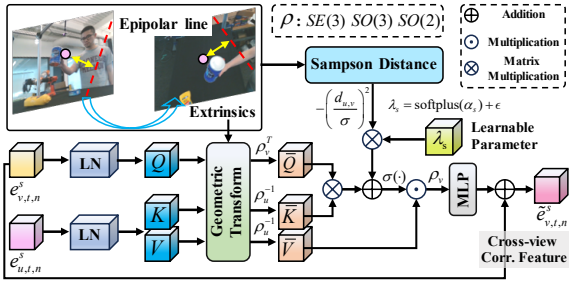


Figure 3: **Epipolar-guided geometric transform attention.** EpiGTA geometrically aligns cross-view features and uses an epipolar attention bias to suppress inconsistent correspondence evidence.

While relative geometric alignment relates feature representations across camera frames, it does not establish the projective compatibility of the current 2D hypotheses. We therefore leverage epipolar compatibility to modulate cross-view attention. For synchronized track centers in views v and u, we convert the Sampson distance (Sampson, 1982; Hartley & Zisserman, 2003) d_{vu} (see Appendix D) into a nonpositive bias and apply it to cross-view attention:

$$B^s_{vu} = -\frac{d_{vu}}{\sigma^2}, \qquad A^s_{vu} = \mathrm{Softmax}_u\left(\frac{(\bar{q}^s_v)^\top \bar{k}^s_u}{\sqrt{d_h}} + \lambda_s B^s_{vu}\right). \tag{2}$$

Here σ is a normalization scale, and $\lambda_s = \mathrm{softplus}(\alpha_s) + \epsilon$ is a positive learnable scale that adaptively balances epipolar guidance against feature similarity at pyramid level s. Thus, cross-view aggregation is governed jointly by feature affinity in aligned representations and the epipolar compatibility of

the current 2D hypotheses. The aligned values are aggregated and mapped back to the target-view representation, then fused with the local feature through an MLP and residual connection:

$$z_v^s = \rho_v \sum_u A_{vu}^s \bar{v}_u^s, \qquad \widetilde{e}_{v,t,n}^s = e_{v,t,n}^s + \mathrm{MLP}(z_v^s). \tag{3}$$

3.2 Geometry-Guided Iterative Updates

Building on the geometry-enhanced correlation features, we extend calibrated geometric guidance to 2D point-state updates over a sliding window of T' frames. At iteration m, we combine cross-view correlation features from four scales into $\widetilde{E}_{v,t,n}^{(m)} \in \mathbb{R}^{4d}$ for each point n in view v. Each point-state token is $G_{v,t,n}^{(m)} = [C_{v,t,n}^{(m)}, V_{v,t,n}^{(m)}, \widetilde{E}_{v,t,n}^{(m)}, \eta_{t\to t+1}^{(m)}, \eta_{t\to t-1}^{(m)}]$. Here $C^{(m)}$ and $V^{(m)}$ are confidence and visibility logits. The Fourier features η encode inter-frame displacements (Tancik et al., 2020). The transformer Ψ processes the token grid $G^{(m)} = \{G_{v,t,n}^{(m)}\}$ with added Fourier time embeddings.

Geometry-guided transformer. To connect point states across time, views, and tracked points, the transformer interleaves temporal, geometry-guided cross-view, and point-space attention. Temporal attention exchanges information along each trajectory, while cross-view attention operates on synchronized states of the same point. Following CoTracker (Karaev et al., 2024), point-space attention uses a small set of learned virtual tracks to capture dependencies among tracked points.

Rather than treating camera geometry (Plücker, 1828) as auxiliary token context (Koo et al., 2026), our cross-view attention incorporates transformations determined by relative camera poses and current image locations directly into state interaction. For each attention head, we project each point's synchronized view tokens into query, key, and value features $q_v, k_u, v_u \in \mathbb{R}^{d_h}$. Following the geometric transformation scheme in EpiGTA, we obtain aligned state-attention features $\bar{q}_v$, $\bar{k}_u$, and $\bar{v}_u$. We then compute cross-view attention and map the aggregated features back to each target view:

$$A_{vu} = \mathrm{Softmax}_u\left(\frac{\bar{q}_v^\top \bar{k}_u}{\sqrt{d_h}}\right), \qquad z_v = \rho_v \sum_u A_{vu}\bar{v}_u. \tag{4}$$

The resulting message is integrated into the target-view point state for trajectory refinement. Unlike correlation aggregation, this interaction also incorporates motion and temporal context to correct the current hypotheses. We therefore retain geometric alignment without the additional epipolar bias, which may downweight useful cross-view information when current position estimates are inaccurate.

Recurrent updates. For the first window, we initialize $P^{(0)}$ by repeating the query coordinates across frames and set the confidence and visibility logits to $C^{(0)} = V^{(0)} = 0$. At iteration m, the transformer predicts residuals $(\Delta P, \Delta C, \Delta V)^{(m+1)} = \Psi(G^{(m)})$. We update the trajectories as $P^{(m+1)} = P^{(m)} + \Delta P^{(m+1)}$, and update confidence and visibility logits analogously. Each position update refreshes the local correlation sampling centers, the image-coordinate components of ρ_v, and the epipolar residuals for the next iteration. The two stages are thus coupled through the evolving 2D hypotheses: aggregated correlation features guide state updates, while revised hypotheses determine the features and geometric compatibility used in subsequent aggregation. After M iterations, $P = P^{(M)}$ gives the predicted 2D trajectories, while $\widehat{C} = \mathrm{sigmoid}(C^{(M)})$ and $\widehat{V} = \mathrm{sigmoid}(V^{(M)})$ give the confidence and visibility predictions. This recurrent geometric guidance promotes accurate and cross-view-consistent 2D trajectories.

3.3 Training and Inference

Training. Following the unrolled window strategy of CoTracker (Karaev et al., 2024), we divide each sequence into J overlapping windows of length T', shifted by $T'/2$ frames, and propagate predictions between consecutive windows. All recurrent predictions are supervised using only 2D point annotations with:

$$\mathcal{L} = \lambda_{\mathrm{track}}\mathcal{L}_{\mathrm{track}} + \mathcal{L}_{\mathrm{vis}} + \mathcal{L}_{\mathrm{conf}}. \tag{5}$$

where $\mathcal{L}_{\mathrm{track}}$ is a Huber loss on the 2D trajectories, while $\mathcal{L}_{\mathrm{vis}}$ and $\mathcal{L}_{\mathrm{conf}}$ are binary cross-entropy losses for visibility and confidence. The complete loss definitions are provided in Appendix D.

Inference. At inference, we iteratively estimate multi-view 2D trajectories and recover 3D trajectories through calibrated, weighted triangulation (Hartley & Zisserman, 2003). Each 2D observation is weighted by the product of its predicted visibility $\widehat{V}_{v,t,n}$ and confidence $\widehat{C}_{v,t,n}$.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. To support large-scale training, we construct a synthetic multi-view dataset of 12K video sequences using Kubric (Greff et al., 2022). For evaluation, we use the DexYCB (Chao et al., 2021), Panoptic Studio (Joo et al., 2015), and Multi-View Kubric tracking benchmarks constructed by MVTracker (Rajič et al., 2025). We additionally evaluate on Harmony4D (Khirodkar et al., 2024) using the tracking annotations released by MV-TAP (Koo et al., 2026).

Evaluation metrics. For consistent comparisons across datasets, we select 8 views per sequence across all datasets using farthest point sampling (FPS). For 2D tracking, we use TAP-Vid (Doersch et al., 2022) metrics: Average Jaccard (AJ) for joint localization and visibility, average location accuracy (δ_{avg}) across distance thresholds, and Occlusion Accuracy (OA) evaluates visibility prediction. For 3D tracking, we report their 3D counterparts and Median Trajectory Error (MTE, cm) over visible points, following MVTracker (Rajič et al., 2025). All metrics are computed per track, averaged within a scene, and finally aggregated across all dataset scenes. See Appendix E for details.

Implementation details. We implement GeoMVT in PyTorch, initialize its feature encoder and transformer from MV-TAP (Koo et al., 2026), and freeze only the encoder during training. We train on our synthetic dataset for 150K steps using a single PRO 6000 GPU with a batch size of 1. We use AdamW (Loshchilov & Hutter, 2017) with a learning rate and weight decay of 10^{-4}, a cosine learning rate schedule, and 5% warm-up. Training uses a 384×512 input resolution, 384 sampled trajectories per sequence, and a 16-frame sliding window. Due to limited computational resources, we train ablation models on 4K synthetic sequences for 50K steps. See Appendix E for further details.

Baselines. We compare our method against recent baselines in three categories. (i) *Single-view 2D tracking.* We include TAPIR (Doersch et al., 2023), CoTracker2 (Karaev et al., 2024), CoTracker3 (Karaev et al., 2025), LocoTrack (Cho et al., 2024), and TAPNext (Zholus et al., 2025). (ii) *Single-view 3D tracking and reconstruction.* We include SpatialTracker (Xiao et al., 2024), SceneTracker (Wang et al., 2025a), DELTA (Ngo et al., 2025), SpatialTrackerV2 (Xiao et al., 2025), TAPIP3D (Zhang et al., 2025), and TrackCraft3R (Nam et al., 2026). We also evaluate TrajVG (Miao et al., 2026), a feed-forward 3D reconstruction model with a 3D tracking branch. (iii) *Multi-view tracking.* We compare with MVTracker (Rajič et al., 2025), MV-TAP (Koo et al., 2026), LAPA (Galoaa et al., 2026). For 2D evaluation, we average per-view scores, running single-view models independently and projecting 3D tracks into each image plane. For 3D evaluation, single-view baselines track each query from the view with the best visibility according to projected depth, following MVTracker (Rajič et al., 2025). For methods requiring depth, we use ground-truth depth for DexYCB (Chao et al., 2021), optimization-based ground-truth depth (Luiten et al., 2024) for Panoptic Studio, simulated depth for Multi-View Kubric (Greff et al., 2022), and depth estimated by VGGT-Ω (Wang et al., 2026) for Harmony4D. See Appendix E for implementation details.

4.2 COMPARISON WITH THE STATE OF THE ART

2D point tracking. As shown in Tab. 1, GeoMVT consistently outperforms all baselines across all 2D metrics on the four benchmarks. At the same 4K training scale as MV-TAP (Koo et al., 2026), our model improves AJ from 45.9 to **51.4** on DexYCB and from 43.9 to **50.9** on Panoptic Studio, together with clear gains in localization and occlusion accuracy. Scaling the training data further improves performance, with the full model reaching **55.1**, **54.1**, **66.4**, and **46.9** AJ on DexYCB, Panoptic Studio, Multi-View Kubric, and Harmony4D, respectively. These improvements are also visually apparent in Fig. 4. These consistent gains demonstrate that geometry-guided cross-view reasoning effectively exploits complementary observations to improve multi-view 2D tracking.

3D point tracking. As shown in Tab. 2, directly triangulating the 2D predictions allows GeoMVT to consistently outperform MV-TAP (Koo et al., 2026) in AJ and MTE across all four datasets. This shows that the gains of geometry-guided tracking extend beyond image-space accuracy and translate into more reliable 3D recovery. Without depth input, GeoMVT further outperforms the dedicated 3D tracker MVTracker (Rajič et al., 2025) on DexYCB and Harmony4D, increasing AJ from 73.3 to **80.6** and from 60.8 to **68.1**, while reducing MTE from 1.7 to **1.1** cm and from 11.9 to **7.8** cm, respectively. On Panoptic Studio, our method achieves comparable AJ (90.4 vs. 91.4) with lower MTE (3.0 vs. 3.2 cm). Although MVTracker remains stronger on Multi-View Kubric, GeoMVT substantially improves over MV-TAP, raising AJ from 30.7 to **56.6** and reducing MTE from 91.1 to **19.6** cm. Fig. 4 provides qualitative comparisons, and Appendix A further provides results using

Table 1: **Quantitative comparison in 2D point tracking.** We report results on DexYCB (Chao et al., 2021), Panoptic Studio (Joo et al., 2015), Multi-View Kubric (Greff et al., 2022), and Harmony4D (Khirodkar et al., 2024). "Target" denotes the dimensionality of the predicted trajectory, while "Space" specifies whether predictions use camera-centric or world coordinates. "Depth" indicates whether depth input is required. "4K" denotes models trained on 4K video sequences. Our method outperforms existing methods by clear margins across all metrics on all four datasets.

Method	Target	Space	Depth	DexYCB			Panoptic Studio			Multi-View Kubric			Harmony4D		
				AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
Single-view input															
TAPIR (Doersch et al., 2023)	2D	Camera	✗	29.6	43.9	66.4	22.1	39.3	60.0	39.6	57.3	78.0	32.6	61.2	60.7
CoTracker2 (Karaev et al., 2024)	2D	Camera	✗	37.5	62.5	69.4	36.3	58.9	71.1	38.8	60.1	78.1	34.8	65.8	63.0
LocoTrack (Cho et al., 2024)	2D	Camera	✗	38.7	55.8	74.1	34.9	56.1	67.5	40.6	60.1	76.3	35.5	65.8	62.0
CoTracker3 (Karaev et al., 2025)	2D	Camera	✗	41.5	59.6	76.4	39.9	62.5	71.9	40.0	58.8	75.9	36.6	67.2	62.0
TAPNext (Zholus et al., 2025)	2D	Camera	✗	39.6	57.7	71.9	36.2	60.1	66.8	48.1	69.0	82.4	37.1	68.0	67.1
SpatialTracker (Xiao et al., 2024)	3D	Camera	✓	23.2	43.3	61.8	19.7	40.5	59.6	34.7	56.1	75.2	34.7	66.5	62.9
TAPIP3D (Zhang et al., 2025)	3D	World	✓	29.1	43.4	66.1	41.9	62.2	78.9	45.3	67.9	79.4	34.2	65.7	61.5
Multi-view input															
MVTracker (Rajič et al., 2025)	3D	World	✓	–	48.4	–	–	62.4	–	–	64.1	–	–	52.5	–
MV-TAP (Koo et al., 2026)	2D	Camera	✗	44.2	61.9	78.3	40.3	62.8	73.1	45.7	64.6	78.7	36.9	67.8	63.0
MV-TAP (4K) (Koo et al., 2026)	2D	Camera	✗	45.9	63.3	77.7	43.9	67.4	74.8	50.9	69.3	82.3	38.5	71.0	69.0
GeoMVT (Ours) (4K)	2D	Camera	✗	51.4	69.0	84.4	50.9	72.4	81.5	59.5	77.6	87.4	44.7	75.1	72.2
GeoMVT (Ours)	2D	Camera	✗	**55.1**	**72.3**	**86.9**	**54.1**	**75.2**	**82.9**	**66.4**	**82.5**	**90.5**	**46.9**	**77.4**	**72.6**

Table 2: **Quantitative comparison in 3D point tracking.** We report results on DexYCB (Chao et al., 2021), Panoptic Studio (Joo et al., 2015), Multi-View Kubric (Greff et al., 2022), and Harmony4D (Khirodkar et al., 2024). For a fair comparison, MV-TAP and our method use the same triangulation procedure to recover 3D trajectories. Our method outperforms existing methods on most datasets without depth input, highlighting its tracking accuracy and cross-view consistency.

Method	Depth	DexYCB				Panoptic Studio				Multi-View Kubric				Harmony4D			
		AJ↑	δ_{avg} ↑	OA↑	MTE↓	AJ↑	δ_{avg} ↑	OA↑	MTE↓	AJ↑	δ_{avg} ↑	OA↑	MTE↓	AJ↑	δ_{avg} ↑	OA↑	MTE↓
Single-view input																	
SpatialTracker (Xiao et al., 2024)	✓	13.5	21.4	60.8	91.6	41.1	52.7	78.8	20.4	14.9	23.4	51.0	268.9	3.4	5.8	63.8	307.6
SceneTracker (Wang et al., 2025a)	✓	–	44.0	–	72.6	–	80.4	–	8.0	–	18.6	–	305.7	–	15.6	–	243.9
DELTA (Ngo et al., 2025)	✓	30.1	45.0	55.7	98.3	67.8	84.5	76.7	6.1	25.7	38.3	54.7	219.2	13.6	19.6	44.8	244.5
SpatialTrackerV2 (Xiao et al., 2025)	✓	33.0	43.7	77.2	53.5	64.4	79.0	78.4	18.4	22.7	33.1	53.5	247.9	8.5	14.8	56.5	248.4
TAPIP3D (Zhang et al., 2025)	✓	49.2	74.8	65.2	3.4	79.9	92.8	85.5	3.6	42.6	71.1	54.7	21.7	40.7	54.9	79.6	17.6
TrajVG (Miao et al., 2026)	✗	17.5	22.9	92.8	19.4	23.3	26.5	97.7	33.5	4.4	6.3	87.5	182.5	18.4	23.2	95.5	46.3
TrackCraft3R (Nam et al., 2026)	✓	18.1	31.7	54.7	16.5	42.0	66.3	63.3	11.0	12.0	19.8	51.5	96.4	20.6	33.5	60.5	28.5
Multi-view input																	
MVTracker (Rajič et al., 2025)	✓	73.3	81.0	**93.0**	1.7	**91.4**	**94.6**	**98.3**	3.2	**68.9**	**80.7**	**90.7**	**10.1**	60.8	68.9	**95.7**	11.9
MV-TAP (Koo et al., 2026)	✗	72.6	81.1	92.9	1.7	80.1	85.7	97.6	5.4	30.7	39.1	89.2	91.1	52.0	60.6	**95.7**	19.7
LAPA (Galoaa et al., 2026)	✗	48.4	57.3	80.1	8.8	75.2	81.1	91.7	9.0	16.1	19.1	61.5	172.3	26.0	33.8	70.7	57.3
MV-TAP (4K) (Koo et al., 2026)	✗	69.4	77.4	92.8	2.0	81.6	86.8	97.0	5.1	29.4	37.2	88.9	82.1	55.1	63.8	**95.7**	12.8
GeoMVT (Ours) (4K)	✗	78.4	86.3	92.9	1.2	90.1	93.8	97.9	3.1	48.4	57.8	89.1	28.7	63.6	72.0	**95.7**	9.2
GeoMVT (Ours)	✗	**80.6**	**88.4**	92.9	**1.1**	90.4	94.1	97.8	**3.0**	56.6	66.1	90.2	19.6	**68.1**	**76.3**	**95.7**	**7.8**

depth predicted by VGGT-Ω (Wang et al., 2026). The inference speed analysis in Appendix C shows that GeoMVT achieves 21.6 FPS with eight views, highlighting its potential for near-real-time and large-scale applications.

4.3 Ablation Studies

All ablation models are trained on 4K synthetic sequences for 50K steps. Unless otherwise stated, all ablations report 2D tracking results. We report ablation results on DexYCB (Chao et al., 2021) and Panoptic Studio (Joo et al., 2015), with results on Multi-View Kubric (Greff et al., 2022) and Harmony4D (Khirodkar et al., 2024) and further ablations provided in Appendix B.

Main component ablation. Tab. 3 evaluates the two main components of our framework. CCA alone improves AJ from 44.0 to 46.2 on Panoptic Studio and from 46.4 to 49.0 on DexYCB, showing that geometry-guided aggregation provides more reliable cross-view correlation features for subsequent refinement. GIU yields larger standalone gains, reaching 48.7 and 49.7 AJ, respectively, which highlights the importance of using relative camera geometry to guide cross-view point-state interaction during iterative refinement. Combining CCA and GIU further improves AJ to **50.9** and

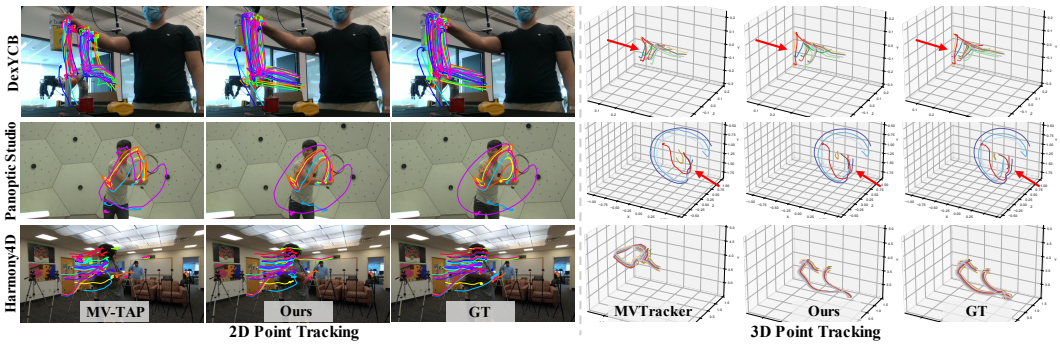


Figure 4: **Qualitative comparison of 2D and 3D point tracking.** Compared with the baselines, GeoMVT produces 2D and 3D trajectories that more closely follow the ground truth (GT). Red arrows highlight regions for comparing 3D trajectories. These examples correspond to the results in Tabs. 1 and 2. Please see Appendix F and the supplementary video for additional visualizations.

Table 3: **Ablation of the main components.** CCA denotes cross-view correlation aggregation, and GIU denotes geometry-guided iterative updates. Coupling the two components improves both track localization and visibility prediction.

CCA	GIU	Panoptic Studio			DexYCB		
		AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
✗	✗	44.0	67.1	75.2	46.4	63.6	78.5
✓	✗	46.2	69.2	76.8	49.0	66.9	81.4
✗	✓	48.7	70.5	80.0	49.7	67.5	83.6
✓	✓	**50.9**	**72.4**	**81.5**	**51.4**	**69.0**	**84.4**

Table 4: **Ablation of EpiGTA in CCA.** GTA denotes geometric transform attention, EPI denotes epipolar guidance, and CVA denotes standard cross-view attention. The baseline (R1) constructs only view-local evidence.

Row	EpiGTA		CVA	Panoptic Studio			DexYCB		
	GTA	EPI		AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
1	✗	✗	✗	44.0	67.1	75.2	46.4	63.6	78.5
2	✗	✗	✓	44.5	67.5	75.9	46.4	63.5	78.8
3	✓	✗	✗	44.8	67.6	76.5	47.6	65.5	80.6
4	✗	✓	✓	**46.2**	**69.6**	76.7	48.6	**66.9**	81.1
5	✓	✓	✗	**46.2**	69.2	**76.8**	**49.0**	**66.9**	**81.4**

51.4, while achieving the best results across all metrics. These results demonstrate that geometric reasoning at the correlation and point-state levels plays complementary roles in multi-view tracking.

Coupled iterative refinement. As shown in Fig. 5, coupling CCA and GIU achieves the highest AJ at every iteration on DexYCB and greater overall refinement gains than either component alone. These results support the complementarity of CCA and GIU during iterative refinement. Results on other datasets are provided in Appendix B.3.

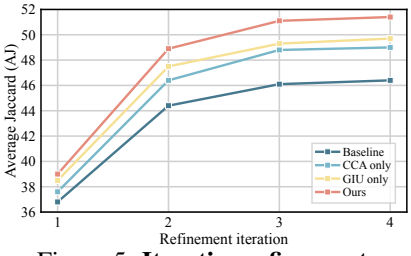


Figure 5: **Iterative refinement.**

Epipolar-guided geometric transform attention. We disable geometry-guided interactions in GIU to isolate the role of EpiGTA within CCA. As shown in Tab. 4, standard cross-view attention yields little improvement over view-local correlation features, whereas geometric transformations improve tracking, particularly on DexYCB. Epipolar guidance further benefits both attention variants, with EpiGTA reaching an AJ of 46.2 on Panoptic Studio and 49.0 on DexYCB. These results highlight the roles of geometric transformations and suppressing geometrically inconsistent observations in effective cross-view correlation aggregation.

Selective epipolar guidance. We further examine the placement of epipolar guidance. As shown in Tab. 5, removing it from CCA or extending it to GIU reduces AJ on both datasets. In CCA, epipolar compatibility filters geometrically inconsistent correspondence evidence. In GIU, however, an additional epipolar bias may suppress cross-view information useful for correcting inaccurate trajectory estimates. These results favor selective epipolar guidance in CCA, with geometric transform attention alone used for state refinement.

Geometry-guided reasoning versus model capacity. We widen the baseline cross-view attention to match or exceed the parameters and FLOPs of our full model. Despite the larger capacity, it reaches only 44.1 and 46.3 AJ on Panoptic Studio and DexYCB, compared with 50.9 and 51.4 for our model. These results show that increasing attention capacity alone does not reproduce the gains of our geometry-guided cross-view reasoning.

Table 5: **Ablation of epipolar guidance placement and model capacity.** "w/o Epi" and "w/ Epi" indicate whether epipolar guidance is disabled or enabled in the specified module. R1 refers to the baseline in Row 1 of Table 4. Wider CVA increases only the width of standard cross-view attention in the iterative transformer.

Model variant	Panoptic Studio			DexYCB			Params (M)	TFLOPs
	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑		
Full model (a)	**50.9**	**72.4**	**81.5**	**51.4**	**69.0**	84.4	**32.28**	16.83
a → CCA w/o Epi	49.3	70.9	80.6	50.0	67.8	83.7	**32.28**	**16.82**
a → GIU w/ Epi	49.4	71.1	80.9	50.9	68.5	**85.0**	**32.28**	16.84
R1 w/ wider CVA	44.1	67.4	75.0	46.3	63.3	78.4	32.48	16.89

Table 6: **Effect of camera layout on Panoptic Studio.** We compare with MV-TAP (Koo et al., 2026). "w/o Geo" corresponds to Row 2 of Table 4, using standard cross-view attention for evidence aggregation without geometric guidance. We report AJ scores.

Camera layout	Mean baseline	Median angle	MV-TAP	w/o Geo	w/ Geo
Compact	1.9 m	42.1°	**45.9**	**46.6**	50.6
Medium	2.9 m	59.9°	43.0	44.9	50.5
Wide	3.8 m	81.6°	43.9	44.5	**50.9**

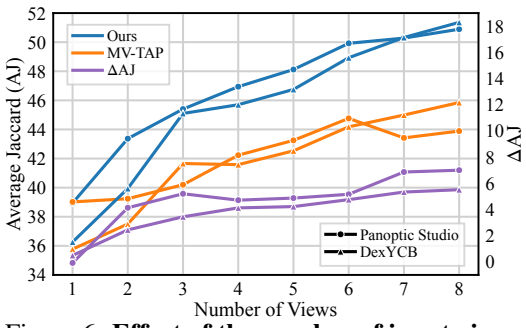


Figure 6: **Effect of the number of input views.** We compare our method with MV-TAP.

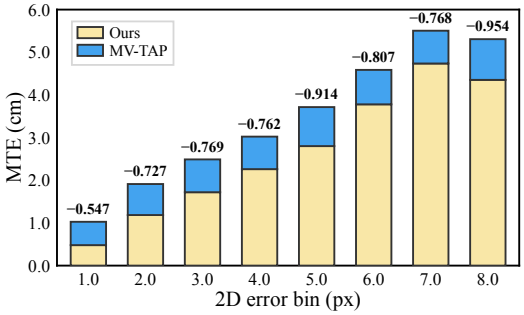


Figure 7: **3D tracking error across 2D error bins on Panoptic Studio.**

Impact of camera layout. In Tab. 6, standard cross-view correlation aggregation improves AJ over MV-TAP (Koo et al., 2026) by 0.6–1.9 across all layouts, suggesting better use of complementary observations. However, its AJ drops from 46.6 to 44.5 as camera baselines and viewing angles increase, indicating that aggregation without geometric guidance remains sensitive to viewpoint differences. Our method maintains **50.5–50.9** AJ, with the advantage over w/o Geo widening from **4.0** to **6.4**, supporting the role of geometric reasoning in exploiting complementary observations across larger viewpoint differences. Limited view combinations in the remaining datasets restrict our additional evaluation to Harmony4D, where we observe similar trends (Appendix B.7).

Effect of the number of input views. Fig. 6 examines how effectively the model uses additional observations on Panoptic Studio and DexYCB. With one view, our method and MV-TAP (Koo et al., 2026) perform similarly. With eight views, our AJ reaches **50.9** on Panoptic Studio and **51.4** on DexYCB, exceeding MV-TAP by **7.0** and **5.5**, respectively. Our advantage over MV-TAP generally grows as more views are added, suggesting that geometry-guided interactions make more effective use of the additional complementary observations. Similar trends are observed on Multi-View Kubric and Harmony4D (see Appendix B.8).

Cross-view consistency for 3D trajectory recovery. We group tracking results on Panoptic Studio into eight 2D error bins of width 1 px, spanning [0, 8) px. With the same triangulation procedure, our method reconstructs more accurate 3D trajectories than MV-TAP (Koo et al., 2026) within every bin, reducing MTE by approximately 0.55–0.95 cm, as shown in Fig. 7. The lower 3D error within the same 2D error ranges suggests that our geometric reasoning framework improves cross-view consistency, enabling multi-view tracks to jointly support more accurate 3D reconstruction. Similar trends are observed on DexYCB, Multi-View Kubric, and Harmony4D (see Appendix B.9).

5 Conclusion

We introduced GeoMVT, a geometry-guided cross-view reasoning framework for multi-view point tracking. GeoMVT couples geometry-guided cross-view correlation aggregation with iterative point-state refinement to exploit complementary multi-view observations. This design suppresses geometrically inconsistent support and produces accurate, cross-view-consistent 2D trajectories. The 2D predictions can be directly triangulated into accurate 3D trajectories without depth input. Extensive experiments demonstrate state-of-the-art multi-view 2D tracking across all four benchmarks and competitive or superior 3D tracking compared with dedicated 3D trackers.

AI USE STATEMENT

In this work, we used generative AI tools to polish the paper's language and assist in checking our research ideas for potential plagiarism or overlap with prior work. We also used these tools to help organize parts of the experimental code. We reviewed all AI-assisted work, and three authors verified and tested the AI-assisted code 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

We are committed to ensuring the reproducibility of our work. Our model design is described in Sec. 3 and Appendix D, with implementation details provided in Sec. 4.1 and Appendix E. The comparative evaluation protocols are documented in Sec. 4.1 and Appendix E, and baseline implementation details are provided in Appendix E. We will publicly release our code, model weights, and synthetic training dataset upon acceptance.

REFERENCES

Görkay Aydemir, Weidi Xie, and Fatma Güney. Track-on2: Enhancing online point tracking with memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pp. 306–324. Springer, 2024.

Yiqian Chang, Qinghong Ye, Haoran Xu, Jianing Li, Dongyang Ma, Xuan Wang, Wei Zhang, Yonghong Tian, and Peixi Peng. Mer-tracker: Towards high-speed 3d point tracking via multi-view event-rgb hybrid cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37032–37042, 2026.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9044–9053, 2021.

Jiahong Chen, Jinghao Wang, Zi Wang, Ziwen Wang, Banglei Guan, and Qifeng Yu. Dkpmv: Dense keypoints fusion from multi-view rgb frames for 6d pose estimation of textureless objects. *arXiv preprint arXiv:2510.10933*, 2025.

Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *European Conference on Computer Vision*, pp. 306–325. Springer, 2024.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.

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*, pp. 10061–10072, 2023.

Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. *arXiv preprint arxiv:2504.13152*, 2025.

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 *2026 International Conference on 3D Vision*, pp. 1–17, 2026. doi: 10.1109/3DV69130.2026.00008.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761, 2022.

Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pp. 59–75. Springer, 2022.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge university press Cambridge, 2003.

Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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*, pp. 3334–3342, 2015.

Sebastian Jung, Artem Zholus, Martin Sundermeyer, Carl Doersch, Ross Goroshin, David Joseph Tan, Sarath Chandar, Rudolph Triebel, and Federico Tombari. Tapnext++: What's next for tracking any point (tap)? *arXiv preprint arXiv:2604.10582*, 2026.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pp. 18–35. Springer, 2024.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6013–6022, October 2025.

Rawal Khirodkar, Jyun-Ting Song, Jinkun Cao, Zhengyi Luo, and Kris Kitani. Harmony4d: A video dataset for in-the-wild close human interactions. *Advances in Neural Information Processing Systems*, 37:107270–107285, 2024.

Inès Hyeonsu Kim, Seokju Cho, Jahyeok Koo, Junghyun Park, Jiahui Huang, Honglak Lee, Joon-Young Lee, and Seungryong Kim. Anthrotap: Learning point tracking with real-world motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42516–42526, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20932–20941, 2026.

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.

Skanda Koppula, Frano Rajic, Abdullah Faiz Ur Rahman, Yi Yang, Ignacio Rocco, Jeet Thakwani, Rishabh Kabra, Andrew Zisserman, Joao Carreira, Siyu Tang, et al. Tapvid-mv: A benchmark for tracking any point in 3d across multiple views. *arXiv preprint arXiv:2609.01899*, 2026.

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

Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Feng Li, Tianhe Ren, Bohan Li, and Lei Zhang. Taptrv2: Attention-based position update improves tracking any point. *Advances in Neural Information Processing Systems*, 37:101074–101095, 2024a.

Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. Taptr: Tracking any point with transformers as detection. In *European Conference on Computer Vision*, pp. 57–75. Springer, 2024b.

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*, volume 38, pp. 15984–16009, 2025. doi: 10.52202/085713-0540.

Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European conference on computer vision*, pp. 531–548. Springer, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.

Xingyu Miao, Weiguang Zhao, Tao Lu, Linning Xu, Mulin Yu, Yang Long, Jiangmiao Pang, and Junting Dong. Trajvg: 3d trajectory-coupled visual geometry learning. *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, 2026.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uJVHygNeSZ.

Jisu Nam, Jahyeok Koo, Soowon Son, Jaewoo Jung, Honggyu An, Junhwa Hur, and Seungryong Kim. TrackCraft3R: Repurposing video diffusion transformers for dense 3d tracking. *arXiv preprint arXiv:2605.12587*, 2026. doi: 10.48550/arXiv.2605.12587.

Tuan Duc Ngo, Peiye Zhuang, Evangelos Kalogerakis, Chuang Gan, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. DELTA: Dense efficient long-range 3d tracking for any video. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d9iHI1eimo.

Julius Plücker. *Analytisch-geometrische Entwicklungen*, volume 1. GD Baedeker, 1828.

Jinyuan Qu, Hongyang Li, Shilong Liu, Tianhe Ren, Zhaoyang Zeng, and Lei Zhang. Taptrv3: Spatial and temporal context foster robust tracking of any point in long video. In *International Conference on Learning Representations*, volume 2026, pp. 24310–24337, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 59–68, 2025.

Paul D. Sampson. Fitting conic sections to "very scattered" data: An iterative refinement of the bookstein algorithm. *Computer Graphics and Image Processing*, 18(1):97–108, 1982.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8269–8279, June 2022a.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pp. 156–174. Springer, 2022b.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547, 2020.

Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5397–5403. IEEE, 2024.

Bo Wang, Jian Li, Yang Yu, Li Liu, Zhenping Sun, and Dewen Hu. Scenetracker: Long-term scene flow estimation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(9): 8305–8312, 2025a. doi: 10.1109/TPAMI.2025.3572489.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025b.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024.

Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on robot learning*, pp. 180–191. PmLR, 2022.

Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. Rayrope: Projective ray positional encoding for multi-view attention. *arXiv preprint arXiv:2601.15275*, 2026.

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20406–20417, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6726–6737, 2025.

Yichen Xie, Depu Meng, Chensheng Peng, Yihan Hu, Quentin Herau, Masayoshi Tomizuka, and Wei Zhan. Urope: Universal relative position embedding across geometric spaces. *arXiv preprint arXiv:2604.18747*, 2026.

Bowei Zhang, Lei Ke, Adam Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. *Advances in Neural Information Processing Systems*, 38:135284–135303, 2025.

Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi SM Sajjadi, Sarath Chandar, and Ross Goroshin. Tapnext: Tracking any point (tap) as next token prediction. *arXiv preprint arXiv:2504.05579*, 2025.

APPENDIX CONTENTS

A 3D Tracking Comparison with Predicted Depth 14
B Additional Ablation Studies 15
B.1 Performance under Camera Calibration Errors 15
B.2 Component Effectiveness on Additional Datasets 16
B.3 Coupled Iterative Refinement 16
B.4 Further Exploration of Geometric Designs 17
B.5 Epipolar Guidance in EpiGTA 19
B.6 Cross-View Attention Ablation 19
B.7 Effect of Camera Layout 21
B.8 Effect of the Number of Views 22
B.9 Cross-View Consistency for 3D Trajectory Recovery 22
B.10 Scaling Experiments 22
C Inference Speed Comparison 23
D Additional Method Details 24
E Additional Experimental Details 24
F Additional Qualitative Results 26
F.1 2D Point Tracking 26
F.2 3D Point Tracking 27
G Limitations and Future Work 31

A 3D TRACKING COMPARISON WITH PREDICTED DEPTH

Tab. A.1 evaluates depth-dependent trackers using VGGT-Ω (Wang et al., 2026) estimates on DexYCB (Chao et al., 2021), Panoptic Studio (Joo et al., 2015), and Multi-View Kubric (Rajič et al., 2025). Compared with the reference-depth results in Tab. 2, MVTracker's AJ drops from **91.4** to **79.2** on Panoptic Studio and from **68.9** to **52.7** on MV-Kubric, while TAPIP3D's MTE increases from **3.6** to **9.5** cm and from **21.7** to **42.7** cm, respectively. These changes illustrate sensitivity to depth input, although the effect varies across methods and metrics. Without depth input, GeoMVT achieves the highest AJ across all three datasets, reaching **90.4** versus MVTracker's 79.2 on Panoptic Studio while reducing MTE from 6.2 to **3.0** cm. This highlights the advantage of recovering 3D trajectories through geometry-guided multi-view tracking without relying on estimated depth.

Table A.1: **3D tracking comparison with estimated depth.** Depth-dependent methods use VGGT-Ω (Wang et al., 2026) estimates instead of the reference depth used in Table 2. Depth-free MV-TAP and GeoMVT results are reproduced from that table. MTE is reported in centimeters; best and second-best results are bold and underlined, respectively.

Method	Depth	DexYCB				Panoptic Studio				MV-Kubric			
		AJ↑	δ_{avg} ↑	OA↑	MTE↓	AJ↑	δ_{avg} ↑	OA↑	MTE↓	AJ↑	δ_{avg} ↑	OA↑	MTE↓
Single-view input													
SpatialTracker (Xiao et al., 2024)	✓	24.1	37.2	61.9	22.0	32.8	44.8	71.3	27.3	7.8	13.1	50.9	318.0
DELTA (Ngo et al., 2025)	✓	29.1	47.5	56.1	18.8	42.5	58.0	68.4	19.6	19.3	30.5	52.7	180.4
SpatialTrackerV2 (Xiao et al., 2025)	✓	29.4	42.0	70.6	20.1	43.4	58.3	78.0	17.1	16.6	26.4	52.4	180.2
TAPIP3D (Zhang et al., 2025)	✓	40.9	68.1	60.9	4.4	51.3	70.1	76.8	9.5	22.0	45.2	47.0	42.7
Multi-view input													
MVTracker (Rajič et al., 2025)	✓	69.3	77.4	**93.0**	2.1	79.2	84.0	**98.3**	6.2	<u>52.7</u>	<u>65.4</u>	87.7	**17.9**
MV-TAP (Koo et al., 2026)	✗	72.6	81.1	<u>92.9</u>	1.7	80.1	85.7	97.6	5.4	30.7	39.1	<u>89.2</u>	91.1
MV-TAP (4K) (Koo et al., 2026)	✗	69.4	77.4	92.8	2.0	81.6	86.8	97.0	5.1	29.4	37.2	88.9	82.1
GeoMVT (Ours) (4K)	✗	<u>78.4</u>	<u>86.3</u>	<u>92.9</u>	<u>1.2</u>	<u>90.1</u>	<u>93.8</u>	<u>97.9</u>	<u>3.1</u>	48.4	57.8	89.1	28.7
GeoMVT (Ours)	✗	**80.6**	**88.4**	<u>92.9</u>	**1.1**	**90.4**	**94.1**	97.8	**3.0**	**56.6**	**66.1**	**90.2**	<u>19.6</u>

B Additional Ablation Studies

We first evaluate performance under camera calibration errors to examine how the models use calibrated geometry (Sec. B.1). We then extend the component and iterative-refinement analyses to additional datasets (Secs. B.2–B.3) and examine geometric designs, epipolar guidance, and cross-view attention variants (Secs. B.4–B.6). Finally, we investigate camera layout and view count (Secs. B.7–B.8), cross-view consistency for 3D recovery (Sec. B.9), and synthetic training-data scaling (Sec. B.10).

B.1 Performance under Camera Calibration Errors

We evaluate performance under camera calibration errors on Panoptic Studio (Joo et al., 2015), DexYCB (Chao et al., 2021), MV-Kubric (Rajič et al., 2025), and Harmony4D (Khirodkar et al., 2024). We separately perturb camera rotations with fixed camera centers and camera positions with fixed rotations. Noise magnitudes are measured by their RMS across eight cameras, with position errors normalized by the median inter-camera baseline. As shown in Figs. B.1 and B.2, our AJ decreases as calibration errors increase, providing evidence that calibrated 3D camera geometry meaningfully influences our tracking predictions. Nevertheless, our method outperforms MV-TAP (Koo et al., 2026) on all four datasets at rotation errors up to 1° and throughout the tested position-error range up to 2%. In contrast, MV-TAP remains nearly unchanged, with occasional slight increases under larger perturbations, suggesting limited awareness of calibrated 3D camera relations, which can constrain the exploitation of complementary observations across views. These observations support the effectiveness of explicitly embedding calibrated geometry into cross-view reasoning, with performance gains retained under moderate calibration errors.

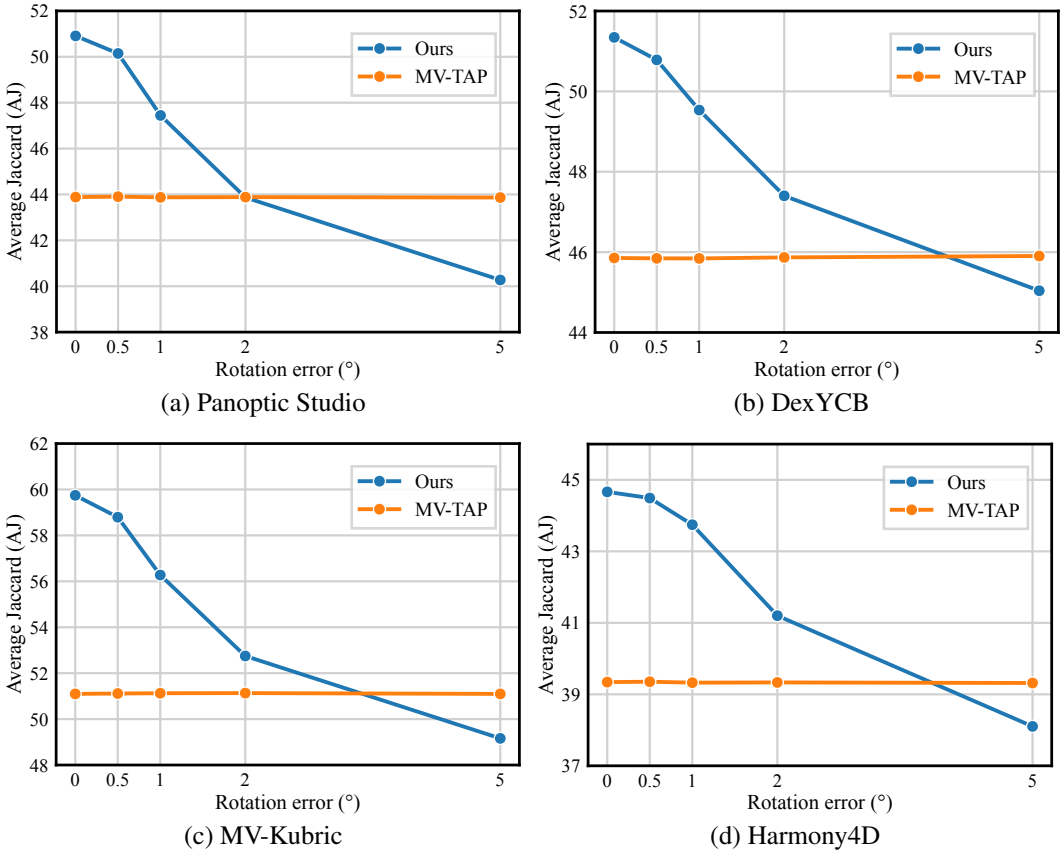
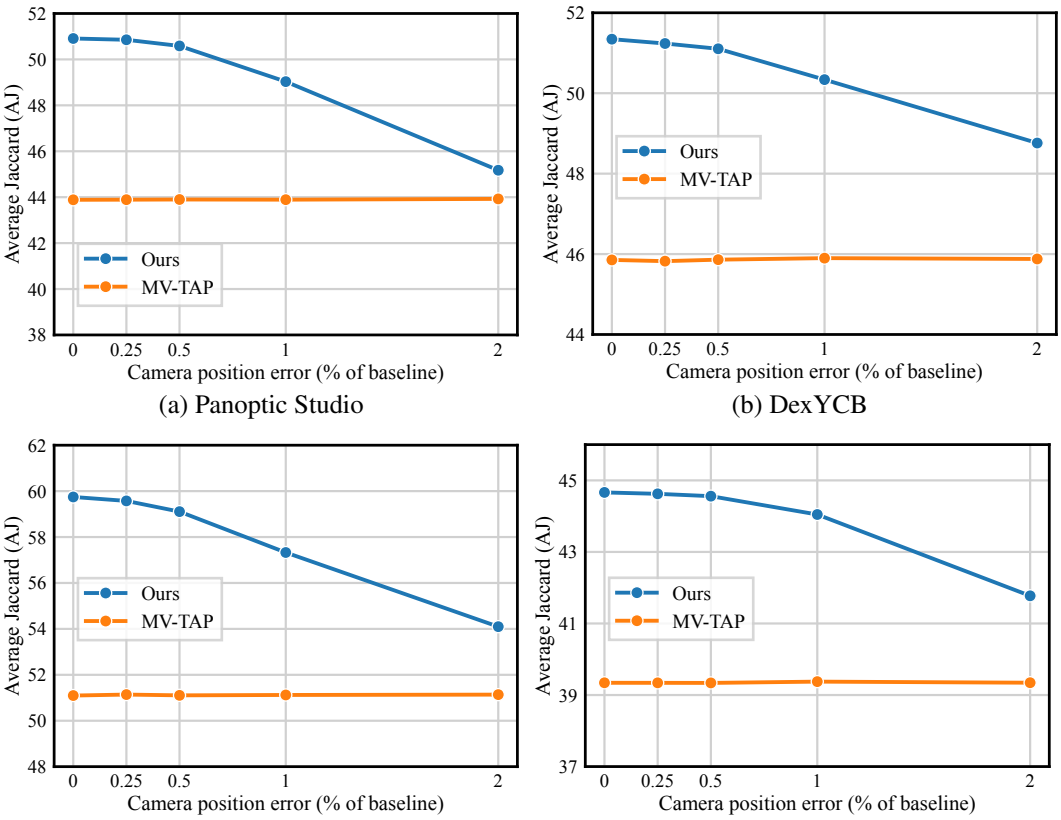


Figure B.1: **Performance under camera rotation errors.** We compare GeoMVT with MV-TAP (Koo et al., 2026) on Panoptic Studio (Joo et al., 2015), DexYCB (Chao et al., 2021), MV-Kubric (Rajič et al., 2025), and Harmony4D (Khirodkar et al., 2024). The horizontal axis gives the root mean square (RMS) rotation perturbation across the eight cameras. Nonzero-noise results are averaged over three noise seeds.



(a) Panoptic Studio (b) DexYCB

(c) MV-Kubric (d) Harmony4D

Figure B.2: **Performance under camera position errors.** We compare GeoMVT with MV-TAP (Koo et al., 2026) on the same four datasets as Figure B.1. Position error is the RMS camera-center displacement across the eight cameras, expressed as a percentage of the sequence's median inter-camera baseline. Nonzero-noise results are averaged over three noise seeds.

B.2 Component Effectiveness on Additional Datasets

Tabs. B.1 and B.2 extend the component and geometric-design ablations to Multi-View Kubric (Rajič et al., 2025) and Harmony4D (Khirodkar et al., 2024).

Coupled geometry-guided cross-view reasoning. In Tab. B.1, CCA alone increases AJ from 51.5 to 54.1 on Multi-View Kubric and from 39.1 to 41.5 on Harmony4D, while GIU alone reaches 57.3 and 43.1, respectively. Combining them further raises AJ to **59.5** and **44.7**, with the best or tied-best results across all metrics. The same trend across synthetic and real scenes supports the complementary roles of geometry-guided correlation aggregation and point-state refinement.

Selective epipolar guidance. Tab. B.2 shows that removing epipolar guidance from CCA reduces AJ to 57.9 and 43.8, while additionally applying it to GIU yields 58.2 and 43.9. Both variants also lower localization accuracy on both datasets, supporting the use of epipolar constraints for correlation aggregation while preserving cross-view corrective interactions during state refinement.

Geometry-guided reasoning versus model capacity. Widening the baseline cross-view attention achieves only 51.0 and 40.5 AJ, compared with **59.5** and **44.7** for the full model in Tab. B.2. This gap further supports geometry-guided cross-view reasoning, rather than increased attention capacity, as the source of the improvement.

B.3 Coupled Iterative Refinement

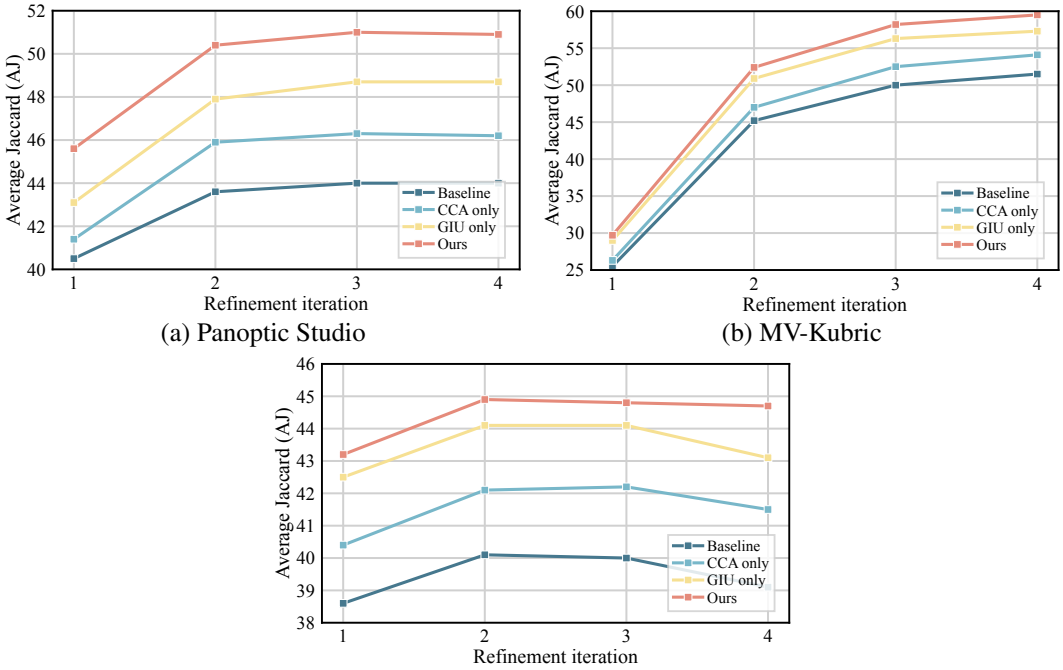
Fig. B.3 extends the DexYCB (Chao et al., 2021) analysis in Fig. 5 to Panoptic Studio (Joo et al., 2015), Multi-View Kubric (Rajič et al., 2025), and Harmony4D (Khirodkar et al., 2024), with all four

Table B.1: **Main-component ablation on additional datasets.** CCA and GIU follow the definitions in Table 3.

CCA	GIU	MV-Kubric			Harmony4D		
		AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
✗	✗	51.5	70.0	82.7	39.1	71.4	68.7
✓	✗	54.1	72.5	84.4	41.5	73.0	70.2
✗	✓	57.3	75.5	86.5	43.1	73.9	**72.2**
✓	✓	**59.5**	**77.6**	**87.4**	**44.7**	**75.1**	**72.2**

Table B.2: **Epipolar guidance placement and model capacity on additional datasets.** Variants follow Table 5.

Model variant	MV-Kubric			Harmony4D		
	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
Full model (a)	**59.5**	**77.6**	**87.4**	**44.7**	**75.1**	72.2
a → CCA w/o Epi	57.9	76.2	86.6	43.8	74.5	**72.8**
a → GIU w/ Epi	58.2	76.1	**87.4**	43.9	74.7	71.5
R1 w/ wider CVA	51.0	69.6	82.5	40.5	72.2	69.3



(a) Panoptic Studio (b) MV-Kubric

(c) Harmony4D

Figure B.3: **Coupled iterative refinement on additional datasets.** We compare the baseline, CCA alone, GIU alone, and their combination on (a) Panoptic Studio (Joo et al., 2015), (b) MV-Kubric (Rajič et al., 2025), and (c) Harmony4D (Khirodkar et al., 2024). DexYCB results are shown in Figure 5.

datasets reported in Tab. B.3. Coupling CCA and GIU achieves the highest AJ at every iteration across all four datasets and greater overall gains from the first to the fourth iteration than either component alone. Its advantage over GIU alone also grows between the first and final iterations, indicating that correlation aggregation benefits subsequent refinement beyond the initial update. These results support the iterative complementarity of our two stages: aggregated correlation features guide state updates, while updated trajectories determine the feature sampling and geometric relations for the next round of aggregation. Even as later iterations approach saturation, the coupled model retains its accuracy advantage.

B.4 Further Exploration of Geometric Designs

We use the CCA configuration as the baseline in Tab. B.4, keeping geometry-guided interactions in GIU disabled to study alternative geometric designs.

Camera intrinsics in geometric transformations. We additionally incorporate camera intrinsics into the cross-view feature transformation using the projective encoding of PRoPE (Li et al., 2025). This raises AJ from 54.1 to 55.2 on Multi-View Kubric but reduces it by 0.6, 1.1, and 0.3 on Panoptic

Table B.3: **AJ across refinement iterations.** Variants follow Tables 3 and B.1, whose results are used for the fourth iteration.

Dataset	Iteration	Baseline	CCA only	GIU only	Ours
Panoptic Studio (Joo et al., 2015)	1	40.5	41.4	43.1	**45.6**
	2	43.6	45.9	47.9	**50.4**
	3	44.0	46.3	48.7	**51.0**
	4	44.0	46.2	48.7	**50.9**
DexYCB (Chao et al., 2021)	1	36.8	37.6	38.5	**39.0**
	2	44.4	46.4	47.5	**48.9**
	3	46.1	48.8	49.3	**51.1**
	4	46.4	49.0	49.7	**51.4**
MV-Kubric (Rajič et al., 2025)	1	25.5	26.3	29.0	**29.7**
	2	45.2	47.0	50.9	**52.4**
	3	50.0	52.5	56.3	**58.2**
	4	51.5	54.1	57.3	**59.5**
Harmony4D (Khirodkar et al., 2024)	1	38.6	40.4	42.5	**43.2**
	2	40.1	42.1	44.1	**44.9**
	3	40.0	42.2	44.1	**44.8**
	4	39.1	41.5	43.1	**44.7**

Studio, DexYCB, and Harmony4D, respectively. CCA aggregates local correlation features that encode query-to-candidate matching relationships, rather than raw image tokens. Applying an intrinsic-dependent transformation to these features may entangle matching quality with camera-specific imaging parameters, reducing their comparability across views and transferability to real scenes. Meanwhile, the epipolar bias already uses intrinsics through the fundamental matrix to assess the compatibility of each current point pair. Additional intrinsic conditioning of feature similarity may therefore downweight useful cross-view information even for geometrically compatible hypotheses, without necessarily improving correspondence discrimination. The observed results favor separating correlation-feature alignment from point-specific epipolar compatibility in this setting.

Auxiliary camera encoding. Following MV-TAP (Koo et al., 2026), we encode each point's Plücker-ray coordinates with an MLP and add the resulting embedding to its point-state token. This provides no consistent improvement over CCA, with AJ on Multi-View Kubric dropping from 54.1 to 53.7. Such additive conditioning leaves the network to infer cross-view geometric relations from individual ray embeddings rather than explicitly structuring their interactions (Miyato et al., 2024). Moreover, the ray coordinates depend on the chosen world frame: changing its origin or orientation changes the encoding even when images and relative camera geometry remain unchanged (Li et al., 2025). Adding these embeddings may therefore introduce coordinate-system dependence into point-state representations, increasing the burden of learning transferable cross-view relations. These results support geometry-guided cross-view interactions without additional coordinate-dependent token embeddings.

Auxiliary depth supervision. Adding a depth loss for each tracked point in each view lowers all three tracking metrics on all four datasets. This auxiliary objective requires shared features to encode scene distance and scale alongside matching relationships, potentially introducing competition between depth regression and precise correspondence estimation. When regressing absolute depth, differences in scene depth ranges can also affect the balance between depth and tracking losses, making optimization sensitive to depth normalization and loss weighting. The results favor geometry-guided cross-view interactions with supervision focused on 2D tracking.

Discussion. Together, these ablations show that incorporating more camera information or additional geometric supervision does not consistently improve tracking. Instead, the results favor assigning geometric information to operations that directly support cross-view correspondence reasoning. Our final design uses relative camera transformations to align correlation features, while camera intrinsics enter the epipolar constraint to assess the compatibility of current point hypotheses. This separates feature alignment from correspondence reliability, allowing complementary observations to be

Table B.4: **Alternative geometric designs in CCA.** CCA is the reference configuration, with geometry-guided interactions in GIU disabled. PRoPE replaces the GTA-based feature transformation with projective camera encoding. Cam retains auxiliary Plücker-ray (Plücker, 1828) encoding used in MV-TAP (Koo et al., 2026), and Depth adds an auxiliary depth loss for each tracked point in each view.

Model variant	Panoptic Studio			DexYCB			Multi-View Kubric			Harmony4D		
	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
CCA	**46.2**	69.2	76.8	**49.0**	**66.9**	**81.4**	54.1	72.5	84.4	**41.5**	**73.0**	70.2
CCA w/ PRoPE	45.6	68.2	**76.9**	47.9	65.9	80.3	**55.2**	**73.2**	**85.7**	41.2	72.5	69.7
CCA + Cam	**46.2**	**69.5**	76.7	**49.0**	**66.9**	81.2	53.7	72.2	83.7	41.1	72.6	**70.9**
CCA + Depth	45.1	68.1	76.1	48.6	66.7	80.7	53.6	72.2	83.6	40.5	72.5	69.9

aggregated while suppressing geometrically inconsistent support. We omit auxiliary ray embeddings and depth supervision, avoiding additional coordinate-dependent token conditioning and keeping the learning objective focused on 2D tracking. The resulting design embeds geometry directly into cross-view interactions, with 3D trajectories subsequently recovered from the multi-view 2D tracks by triangulation.

B.5 Epipolar Guidance in EpiGTA

Sensitivity to the normalization scale. Tab. B.5 evaluates EpiGTA with $\sigma \in \{3.0, 5.0, 7.0\}$. For a fixed learned weight, smaller σ penalizes epipolar deviations more strongly, whereas larger σ relaxes the geometric constraint. Despite this change, AJ varies by at most 0.9 within each dataset, with localization and visibility accuracy showing similarly small variations. No single value performs best across all datasets and metrics, indicating stable tracking over the tested range rather than a sharply tuned optimum. We use $\sigma = 5.0$ as the default, while the learned weights adapt the guidance strength at each feature level.

Table B.5: **Sensitivity to the epipolar normalization scale in EpiGTA.** We vary σ in the epipolar bias $B^s_{vu} = -(d_{vu}/\sigma)^2$ and report 2D tracking results on Panoptic Studio (Joo et al., 2015), DexYCB (Chao et al., 2021), Multi-View Kubric (Greff et al., 2022), and Harmony4D (Khirodkar et al., 2024). Our default setting is $\sigma = 5.0$.

σ	Panoptic Studio			DexYCB			Multi-View Kubric			Harmony4D		
	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
3.0	50.0	71.9	81.2	**51.8**	**69.5**	85.2	**59.9**	**77.8**	**87.8**	44.1	75.0	**72.3**
5.0	**50.9**	**72.4**	**81.5**	51.4	69.0	84.4	59.5	77.6	87.4	**44.7**	**75.1**	72.2
7.0	50.2	71.7	**81.5**	51.5	69.0	**85.3**	59.5	77.4	87.7	43.9	74.7	72.1

Learned scale-dependent guidance. Tab. B.6 reports the learned parameters and positive weights for feature maps downsampled by $4\times$, $8\times$, $16\times$, and $32\times$ relative to the input image. The finest ($4\times$) level receives the largest weight, $\lambda_1 = 0.168599$, approximately 11.8 times the weight at the coarsest ($32\times$) level, $\lambda_4 = 0.014311$. Although the weights are not monotonic across the $8\times$ and $16\times$ levels, all coarser levels use substantially weaker guidance than the $4\times$ level. At coarse scales, weaker penalties retain cross-view information over a wider spatial context for broad matching. At fine scales, stronger penalties suppress geometrically inconsistent support amid local appearance ambiguity, aiding precise localization. The learned weights thus allocate epipolar guidance unevenly across scales rather than imposing a uniform geometric penalty.

B.6 Cross-View Attention Ablation

To isolate how cross-view information flows during iterative refinement, we keep correlation aggregation identical across variants, using standard cross-view attention with epipolar bias but no geometric feature transformations. Fig. B.4 illustrates four transformer variants, all beginning with

Table B.6: **Learned epipolar guidance weights in EpiGTA.** Levels 0–3 correspond to downsampling factors of $4\times$, $8\times$, $16\times$, and $32\times$ relative to the input image, respectively ($s = 1, \ldots, 4$). The raw parameter α_s determines the positive weight $\lambda_s = \mathrm{softplus}(\alpha_s) + \epsilon$.

Feature level	Downsampling	Raw parameter α_s	Positive weight λ_s
Level 0 (finest)	$4\times$	-1.694757	0.168599
Level 1	$8\times$	-3.878388	0.020474
Level 2	$16\times$	-3.400748	0.032805
Level 3 (coarsest)	$32\times$	-4.239618	0.014311

temporal attention. (i) *No view attention* processes each view independently during refinement. (ii) *Global virtual attention* gathers all views into shared virtual tokens, applies self-attention, and returns information to each view. (iii) *Virtual-view attention* first aggregates points into per-view virtual tokens, exchanges information jointly across these tokens, and returns it to the respective views. (iv) *Point-view attention* directly exchanges information between tokens of the same scene point across views before per-view spatial attention. Tab. B.7 reports their tracking results.

Global virtual attention lowers AJ on all four datasets, including a drop from 49.6 to 46.6 on Multi-View Kubric. The shared bank of virtual tokens pools observations across both views and point identities, coupling cross-view correspondence reasoning with spatial aggregation. Information from another view must first enter this shared representation and then be retrieved by the target point, rather than passing directly between observations of the same point. The network must therefore learn both which observations belong together and how to redistribute their information to individual trajectories. This indirect path may dilute a reliable observation with unrelated point states, weakening the targeted corrections needed for a drifting or occluded track. The lower AJ indicates that broader information exchange alone does not ensure effective refinement of individual points.

Virtual-view attention achieves higher AJ than the global variant on every dataset, suggesting a benefit from retaining separate summaries for each view before cross-view exchange. Each bank of virtual tokens first gathers information within its own view and later returns the exchanged information to that view, preserving a more structured input and output path. However, these learned summaries are not explicitly tied to individual scene points, and virtual tokens with the same index across views need not represent the same points. Joint attention over the virtual tokens must still infer which summaries contain relevant support and route it back to the appropriate trajectories. Its inconsistent gains over the variant without view attention suggest that organizing information by view alone is insufficient without an explicit path connecting corresponding points.

Point-view attention directly connects observations of the same scene point at each frame, followed by per-view spatial attention to propagate information among points. This separates cross-view correspondence interaction from spatial context aggregation, raising AJ from 43.6 to **46.2** on Panoptic Studio and from 49.6 to **54.6** on Multi-View Kubric. On Harmony4D, it achieves the highest localization accuracy, although virtual-view attention yields a slightly higher AJ. These results favor information flow that preserves point identity and motivate the direct cross-view interaction structure used in our geometry-guided iterative updates.

Table B.7: **Ablation of cross-view attention in the iterative transformer.** All variants use ordinary cross-view evidence attention with epipolar bias and no geometric feature transformations. We vary the attention structure in the iterative transformer, as illustrated in Figure B.4, and report 2D tracking results.

Attention variant	Panoptic Studio			DexYCB			Multi-View Kubric			Harmony4D		
	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
No view attention	43.6	66.5	74.6	47.6	65.7	79.6	49.6	68.2	80.8	38.9	69.9	68.7
Global virtual attention	42.4	65.9	73.3	46.2	64.3	77.8	46.6	65.5	78.3	38.7	69.6	69.0
Virtual-view attention	43.6	66.7	74.4	46.7	64.9	78.0	49.2	68.0	80.4	**39.3**	70.2	**69.3**
Point-view attention	**46.2**	**69.6**	**76.7**	**48.6**	**66.9**	**81.1**	**54.6**	**73.2**	**84.6**	38.9	**71.3**	**69.3**

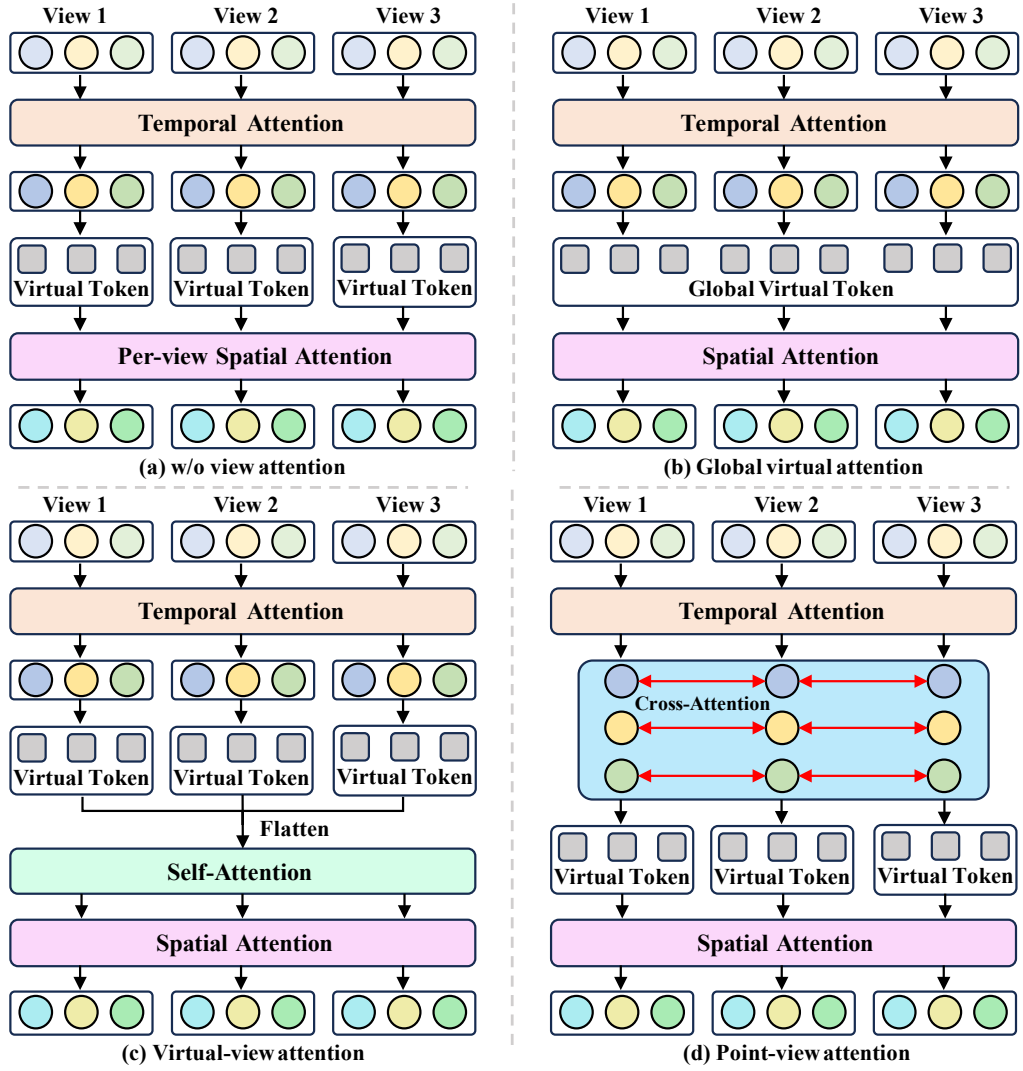


Figure B.4: **Cross-view interaction structures in the iterative transformer.** All variants begin with temporal attention. (a) No cross-view attention is used within the iterative transformer. (b) Shared global virtual tokens gather information from all views, apply self-attention, and pass information back to each view. (c) Per-view virtual tokens gather local information, interact across views through joint self-attention, and return information to their respective views. (d) Tokens of the same scene point directly interact across views before per-view spatial attention.

B.7 Effect of Camera Layout

We sample eight views from the original Harmony4D camera set to form compact, medium, and wide layouts, extending the Panoptic Studio analysis in Tab. 6. In Tab. B.8, standard cross-view correlation aggregation improves AJ over MV-TAP (Koo et al., 2026) by 0.6–2.2 across all layouts, indicating better use of complementary observations. However, its AJ drops from 44.5 to 40.7 as the layout widens, suggesting that aggregation without geometric guidance remains sensitive to viewpoint differences. Our model decreases more moderately from 46.4 to 44.7. The advantage grows from 1.9 AJ in the compact layout to 4.0 in both the medium and wide layouts. This trend is consistent with the Panoptic Studio results in Tab. 6: geometry-guided interactions help exploit complementary observations as larger viewpoint differences make appearance-based association more difficult. We omit the remaining datasets from this comparison because fewer available views limit the choice of camera layouts.

Table B.8: **Effect of camera layout on Harmony4D.** We report AJ scores using eight views on Harmony4D (Khirodkar et al., 2024). We include MV-TAP (Koo et al., 2026) for comparison. "w/o Geo" uses standard cross-view attention for evidence aggregation (Row 2 in Table 4), and "w/ Geo" denotes the full model. ΔAJ denotes the improvement over w/o Geo. Mean baselines are in calibration units, and angles are measured between co-visible rays.

Camera layout	Mean baseline	Ray angle	MV-TAP	w/o Geo	w/ Geo	ΔAJ
Compact	2.8	32.1°	**43.9**	**44.5**	**46.4**	1.9
Medium	4.8	56.8°	39.3	40.3	44.3	**4.0**
Wide	5.2	66.6°	38.5	40.7	44.7	**4.0**

B.8 Effect of the Number of Views

Tab. B.9 extends the analysis in Fig. 6 to all four datasets, varying the number of input views from one to eight. With one view, our AJ differs from MV-TAP (Koo et al., 2026) by at most 0.5 on every dataset. As more views are added, our AJ increases consistently, while MV-TAP largely plateaus beyond six views on Panoptic Studio, Multi-View Kubric, and Harmony4D. On Multi-View Kubric, our AJ rises from 45.9 to **59.5**, compared with 46.1 to 50.9 for MV-TAP, yielding an **8.6** AJ advantage with eight views. The corresponding advantages on DexYCB and Harmony4D reach **5.5** and **6.2**, respectively. These trends support the role of geometry-guided interactions in translating complementary observations from additional views into more accurate tracking.

Table B.9: **Tracking performance with different numbers of views.** We report AJ scores on Panoptic Studio (Joo et al., 2015), DexYCB (Chao et al., 2021), Multi-View Kubric (Greff et al., 2022), and Harmony4D (Khirodkar et al., 2024). ΔAJ denotes Ours minus MV-TAP (Koo et al., 2026).

Views	Panoptic Studio			DexYCB			Multi-View Kubric			Harmony4D		
	MV-TAP	Ours	ΔAJ	MV-TAP	Ours	ΔAJ	MV-TAP	Ours	ΔAJ	MV-TAP	Ours	ΔAJ
1	39.0	38.9	-0.1	35.8	36.3	0.5	46.1	45.9	-0.2	32.0	32.5	0.5
2	39.2	43.4	4.1	37.5	39.9	2.4	48.2	51.5	3.2	34.7	37.2	2.4
3	40.2	45.4	5.2	41.7	45.1	3.4	49.3	54.1	4.7	34.6	37.9	3.3
4	42.2	46.9	4.7	41.6	45.7	4.1	49.2	55.3	6.1	35.7	39.9	4.1
5	43.3	48.1	4.9	42.5	46.8	4.2	50.6	57.4	6.8	37.5	41.9	4.4
6	**44.8**	49.9	5.2	44.2	48.9	4.8	**51.3**	58.7	7.4	**39.9**	43.8	3.9
7	43.4	50.3	6.9	45.0	50.3	5.3	51.1	59.1	8.0	38.9	44.0	5.1
8	43.9	**50.9**	**7.0**	**45.9**	**51.4**	**5.5**	50.9	**59.5**	**8.6**	38.5	**44.7**	**6.2**

B.9 Cross-View Consistency for 3D Trajectory Recovery

Tab. B.10 extends the analysis in Fig. 7 to all four datasets using eight 2D error bins of width 1 px. Each comparison uses the same physical tracks whose errors fall within the same bin for both methods, with an identical triangulation procedure. Our method achieves lower MTE than MV-TAP (Koo et al., 2026) in every available bin across all four datasets. For example, MTE decreases from 1.53 to **0.76** cm in the [4,5) bin on DexYCB and from 20.93 to **18.35** cm in the [7,8) bin on Harmony4D. Triangulation depends on how well observations from different views agree on a common 3D point, not just their individual image-plane errors. The lower reconstruction errors within comparable 2D error ranges therefore support the role of our geometric reasoning framework in producing more consistent multi-view trajectories.

B.10 Scaling Experiments

We scale training iterations in direct proportion to the number of synthetic sequences. Interestingly, Tab. B.11 shows that scaling from 4K to 12K sequences raises AJ from 50.9 to **54.1** on Panoptic Studio and from 51.4 to **55.1** on DexYCB. AJ also increases from 59.5 to **66.4** on MV-Kubric and

Table B.10: **3D trajectory errors within matched 2D error bins.** We report MTE in centimeters on Panoptic Studio (Joo et al., 2015), DexYCB (Chao et al., 2021), Multi-View Kubric (Greff et al., 2022), and Harmony4D (Khirodkar et al., 2024). Each bin contains the same physical tracks whose 2D errors fall within that interval for both MV-TAP (Koo et al., 2026) and our method. ΔMTE denotes Ours minus MV-TAP. Negative values indicate lower error. "–" denotes unavailable results.

2D error	Panoptic Studio			DexYCB			Multi-View Kubric			Harmony4D		
	MV-TAP	Ours	Δ	MV-TAP	Ours	Δ	MV-TAP	Ours	Δ	MV-TAP	Ours	Δ
[0,1)	1.03	0.48	-0.55	0.10	0.06	-0.04	22.46	11.91	-10.55	–	–	–
[1,2)	1.91	1.19	-0.73	0.31	0.21	-0.09	23.59	13.03	-10.56	3.53	3.26	-0.26
[2,3)	2.49	1.72	-0.77	0.50	0.38	-0.12	35.55	11.28	-24.27	5.57	4.98	-0.59
[3,4)	3.02	2.26	-0.76	0.76	0.53	-0.23	42.26	17.21	-25.05	7.87	6.84	-1.02
[4,5)	3.71	2.80	-0.91	1.53	0.76	-0.77	28.83	18.19	-10.64	9.82	8.09	-1.73
[5,6)	4.59	3.78	-0.81	1.23	0.96	-0.28	40.80	20.13	-20.68	10.60	9.70	-0.90
[6,7)	5.51	4.74	-0.77	1.70	1.39	-0.31	26.85	21.53	-5.32	15.06	13.01	-2.04
[7,8)	5.31	4.35	-0.95	1.57	1.35	-0.23	29.79	24.34	-5.45	20.93	18.35	-2.58

from 44.7 to **46.9** on Harmony4D. Despite minor fluctuations, 12K performs best across all metrics on all four datasets. These results suggest that gains from larger synthetic training sets can transfer to real scenes.

Table B.11: **Scaling the synthetic training data.** We report 2D tracking results on Panoptic Studio (Joo et al., 2015), DexYCB (Chao et al., 2021), MV-Kubric (Rajič et al., 2025), and Harmony4D (Khirodkar et al., 2024), scaling training iterations in proportion to the number of sequences. Best results are bold.

Data	Panoptic Studio			DexYCB			MV-Kubric			Harmony4D		
	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑	AJ↑	δ_{avg} ↑	OA↑
4K	50.9	72.4	81.5	51.4	69.0	84.4	59.5	77.6	87.4	44.7	75.1	72.2
5K	50.7	72.2	81.5	52.3	70.0	85.4	60.9	78.5	88.4	44.4	75.3	71.9
8K	52.7	74.2	82.2	53.3	71.0	85.6	63.6	80.8	89.3	45.8	76.8	72.1
12K	**54.1**	**75.2**	**82.9**	**55.1**	**72.3**	**86.9**	**66.4**	**82.5**	**90.5**	**46.9**	**77.4**	**72.6**

C Inference Speed Comparison

We evaluate inference speed on the `basketball` sequence with eight synchronized views and 150 frames. Each method runs in a separate process, with one warm-up run followed by five timed runs whose mean is reported in Tab. C.1. Tracking times cover the full sequence and include triangulation for MV-TAP (Koo et al., 2026) and GeoMVT. For DELTA (Ngo et al., 2025), TAPIP3D (Zhang et al., 2025), and MVTracker (Rajič et al., 2025), we additionally include depth prediction using VGGT-Ω (Wang et al., 2026). Its runtime is estimated as 34.6 s by linearly extrapolating the measured time per synchronized frame to the full sequence, excluding depth-map disk writes. Total times are obtained by summing the separately measured tracking and estimated depth prediction times. We compute FPS as $150/(t_{\text{track}} + t_{\text{depth}})$, counting synchronized multi-view frames rather than individual images. Memory denotes peak usage during tracking.

Depth prediction substantially reduces the overall throughput of methods that require it. For example, MVTracker completes tracking in 5.1 s, faster than our 6.9 s, but its estimated total runtime rises to 39.8 s once depth prediction is included, yielding 3.8 FPS. Our method reaches **21.6 FPS**, including triangulation, approximately **5.7 times** this throughput. In this setting, depth prediction dominates MVTracker's total runtime, whereas our formulation recovers 3D trajectories directly from multi-view 2D tracks without this preprocessing cost.

Compared with MV-TAP, our geometric correlation aggregation and iterative cross-view interactions increase runtime from 5.5 to 6.9 s, reducing throughput from 27.5 to 21.6 FPS. Peak tracking memory remains similar at 25.7 GiB for our model versus 25.6 GiB for MV-TAP. This additional computation

supports the higher 2D tracking accuracy and more accurate 3D trajectory recovery reported in Tabs. 1 and 2. Our framework thus trades some tracking speed for improved accuracy while retaining the overall efficiency advantage of avoiding depth prediction.

Table C.1: **Inference speed comparison.** Runtime on the 150-frame, eight-view `basketball` sequence. Total time and FPS include estimated VGGT-Ω depth computation where required. Memory denotes the measured tracking-stage peak.

Method	Tracking (s)	Depth (s)	Total (s)	FPS	Memory (GiB)
DELTA (Ngo et al., 2025)	21.1	34.6	55.7	2.7	13.0
TAPIP3D (Zhang et al., 2025)	77.0	34.6	111.6	1.3	4.5
MVTracker (Rajič et al., 2025)	5.1	34.6	39.8	3.8	14.8
MV-TAP (Koo et al., 2026)	5.5	–	5.5	27.5	25.6
GeoMVT (Ours)	6.9	–	6.9	21.6	25.7

D Additional Method Details

Sampson distance. Let $\widetilde{p}_{v,t,n}$ and $\widetilde{p}_{u,t,n}$ denote the homogeneous pixel coordinates of the current point estimates in target view v and source view u, respectively. Given the fundamental matrix F_{vu} derived from the calibrated cameras, we compute the Sampson distance (Sampson, 1982; Hartley & Zisserman, 2003) as

$$d_{vu} = \frac{\left(\widetilde{p}_{v,t,n}^{\top} F_{vu} \widetilde{p}_{u,t,n}\right)^2}{(F_{vu}\widetilde{p}_{u,t,n})_1^2 + (F_{vu}\widetilde{p}_{u,t,n})_2^2 + (F_{vu}^{\top}\widetilde{p}_{v,t,n})_1^2 + (F_{vu}^{\top}\widetilde{p}_{v,t,n})_2^2}. \tag{6}$$

The denominator normalizes the epipolar residual using line coefficients from both views, making d_{vu} a bidirectional measure of epipolar inconsistency. Since d_{vu} is already a squared-distance quantity, we form the nonpositive attention bias as $B_{vu}^{s} = -d_{vu}/\sigma^2$, using the same distance normalization scale σ at all feature map scales.

Training objective. Let $P_{v,t,n}^{(m,j)}$ denote the predicted 2D position at refinement step m in window j, with ground truth $P_{v,t,n}^{\star}$. The trajectory loss supervises all recurrent predictions using

$$\mathcal{L}_{\text{track}} = \sum_{j,m,v,t,n} \frac{\gamma^{M-m}}{JMVNT} \operatorname{Huber}\left(P_{v,t,n}^{(m,j)} - P_{v,t,n}^{\star}\right). \tag{7}$$

where j, m, v, t, and n index windows, refinements, views, frames, and points, respectively, and γ emphasizes later refinements. Let $V_{v,t,n}^{\star}$ and $C_{v,t,n}^{\star}$ denote the visibility label and confidence target derived from 2D localization accuracy. We obtain predictions $\widehat{V}^{(m,j)} = \operatorname{sigmoid}(V^{(m,j)})$ and $\widehat{C}^{(m,j)} = \operatorname{sigmoid}(C^{(m,j)})$ from the corresponding logits and compute:

$$\mathcal{L}_{\text{vis}} = \sum_{j,m,v,t,n} \frac{1}{JMVNT} \operatorname{BCE}\left(\widehat{V}_{v,t,n}^{(m,j)}, V_{v,t,n}^{\star}\right). \tag{8}$$

$$\mathcal{L}_{\text{conf}} = \sum_{j,m,v,t,n} \frac{1}{JMVNT} \operatorname{BCE}\left(\widehat{C}_{v,t,n}^{(m,j)}, C_{v,t,n}^{\star}\right). \tag{9}$$

The final objective follows Equation 5 and contains no depth or 3D supervision.

E Additional Experimental Details

Training data generation. We generate synthetic multi-view training sequences using Kubric (Greff et al., 2022), with four synchronized views and 60 frames per view at 30 FPS. Camera positions are sampled on hemispheres with radii ranging from 10 to 12 simulation units, with angular separations of 10°–45° between adjacent viewpoints.

Datasets. Multi-View Kubric provides simulated scenes with synchronized camera views and ground-truth point trajectories (Rajič et al., 2025). For DexYCB, MVTracker derives tracking annotations from ground-truth hand and object poses (Rajič et al., 2025). Its Panoptic Studio benchmark merges per-view tracking annotations from TAPVid-3D (Koppula et al., 2024) into multi-view trajectories (Rajič et al., 2025). Harmony4D captures close human interactions (Khirodkar et al., 2024), for which MV-TAP provides mesh-derived tracking annotations (Koo et al., 2026). The existing Harmony4D 3D annotations have a scale mismatch with the calibrated world coordinate system. We estimate a $\mathrm{Sim}(3)$ transformation from annotated multi-view 2D–3D correspondences to align the canonical label coordinates with the calibrated world frame, yielding metric-scale 3D ground truth.

2D evaluation. We use first-visible queries in each view and evaluate subsequent frames. Following TAP-Vid (Doersch et al., 2022), predicted and ground-truth coordinates are rescaled to 256×256 for metric computation, with distance thresholds of $\{1, 2, 4, 8, 16\}$ pixels. At each threshold, location accuracy is the fraction of ground-truth-visible points localized within that distance, while Jaccard additionally accounts for predicted visibility. We average these scores across thresholds to obtain δ_{avg} and AJ, respectively. OA is the fraction of correctly classified visibility states. We sample 300 query points using the same random seed across all datasets.

3D evaluation. We follow MVTracker (Rajič et al., 2025) for 3D query construction. In our evaluation, each sequence contains 384 queries for Panoptic Studio, DexYCB, and Harmony4D, and 512 for Multi-View Kubric; only dynamic points contribute to the reported metrics. Following MVTracker (Rajič et al., 2025), localization is evaluated using Euclidean distances in 3D, and a point is considered visible when observed in at least one camera. MTE is computed as the median localization error over visible frames for each trajectory, then averaged across trajectories and scenes. For MV-TAP and GeoMVT, we recover 3D trajectories using the same triangulation procedure. Following MVTracker (Rajič et al., 2025), we compute AJ and δ_{avg} using distance thresholds of $\{1, 2, 5, 10, 20\}$ cm for DexYCB, $\{0.65, 1.3, 2.6, 5.2, 10.4\}$ simulation-scale-adjusted centimeters for MV-Kubric, and $\{5, 10, 20, 40\}$ cm for Panoptic Studio. Before simulation-scale adjustment, one Kubric unit corresponds to approximately 13 cm (Rajič et al., 2025). For Harmony4D, we use $\{5, 10, 20, 40\}$ cm, matching Panoptic Studio since both datasets capture human motion.

Implementation details. We set $\sigma = 5.0$ and $\lambda_{\text{track}} = 0.05$. We use $M = 4$ refinement iterations and set the sampling radii of the track and query patches for local 4D correlation to $r_p = r_q = 3$. For each training sample, we randomly select 1 to 4 input views and apply consistent flips to the views and their corresponding camera poses. With probability 0.2, we randomly retain 50% of the frames, using the same temporal sampling across views to simulate different motion speeds.

Baseline implementation details. For a fair comparison on 4K training sequences, we initialize MV-TAP (Koo et al., 2026) from its officially released weights and train it on our synthetic dataset using the same training strategy and number of steps as GeoMVT.

Following MVTracker (Rajič et al., 2025), we run monocular models separately on each camera using queries deemed best visible from that view based on their projected depth. For TrajVG (Miao et al., 2026), we recover scale using the median ratio between reference and predicted depths, estimated separately for each source view and query-time suffix. For TrackCraft3R (Nam et al., 2026), we provide depth and camera calibration, process long sequences as interleaved 12-frame clips sharing the first frame, and retrieve a fixed dense trajectory for each query by 3D nearest-neighbor matching at its query time. Its trajectories are expressed in the first-camera frame and transformed to world coordinates using the known calibration. SceneTracker (Wang et al., 2025a) does not provide usable visibility predictions through its public inference interface, so we omit AJ and OA.

For the three-view LAPA model (Galoaa et al., 2026), we enumerate three-camera combinations from the eight available views and assign each query to the three views deemed best visible under the same depth-based criterion.

To obtain VGGT-Ω depth (Wang et al., 2026), we jointly process all eight views at each synchronized time step. We recover the calibrated scale by estimating a similarity transformation between predicted and calibrated camera centers, without using ground-truth depth or trajectories for scale fitting. The resulting depth maps replace the reference depth while retaining each baseline's weights, query protocol, and depth preprocessing.

F Additional Qualitative Results

To further demonstrate the effectiveness of GeoMVT in multi-view 2D tracking and 3D trajectory recovery, we provide additional qualitative comparisons and corresponding supplementary videos on DexYCB (Chao et al., 2021), Panoptic Studio (Joo et al., 2015), Multi-View Kubric (Greff et al., 2022; Rajič et al., 2025), and Harmony4D (Khirodkar et al., 2024). The videos visualize the trajectories over time and complement the examples shown below.

F.1 2D Point Tracking

Figs. F.1–F.4 compare GeoMVT with MV-TAP (Koo et al., 2026) and ground truth (GT) across four datasets. These examples illustrate reduced trajectory drift under object motion and human interactions.

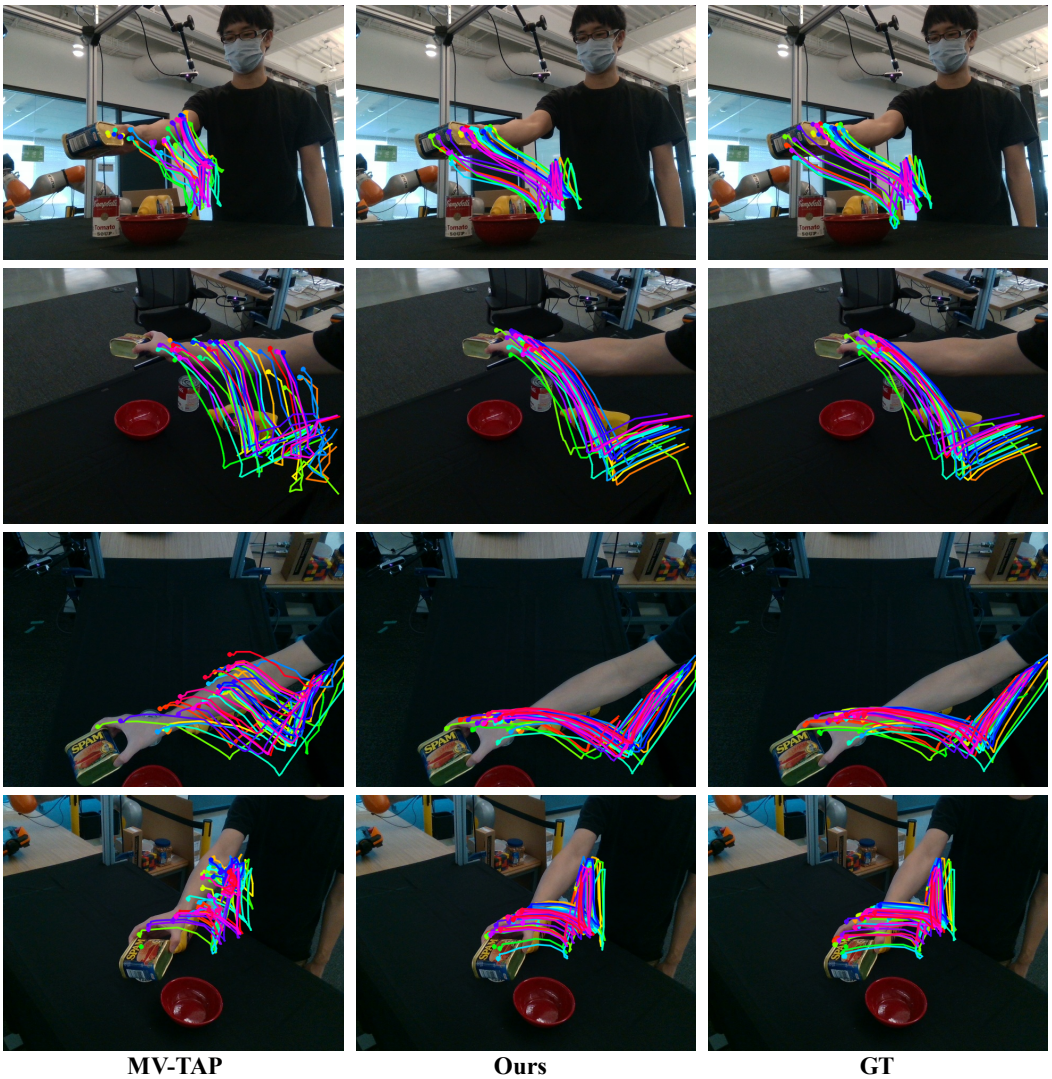


Figure F.1: **2D point tracking on DexYCB** (Chao et al., 2021). Columns show MV-TAP (Koo et al., 2026), GeoMVT, and GT, from left to right. The corresponding video is provided in the supplementary material as `dexycb_2d_output.mp4`.

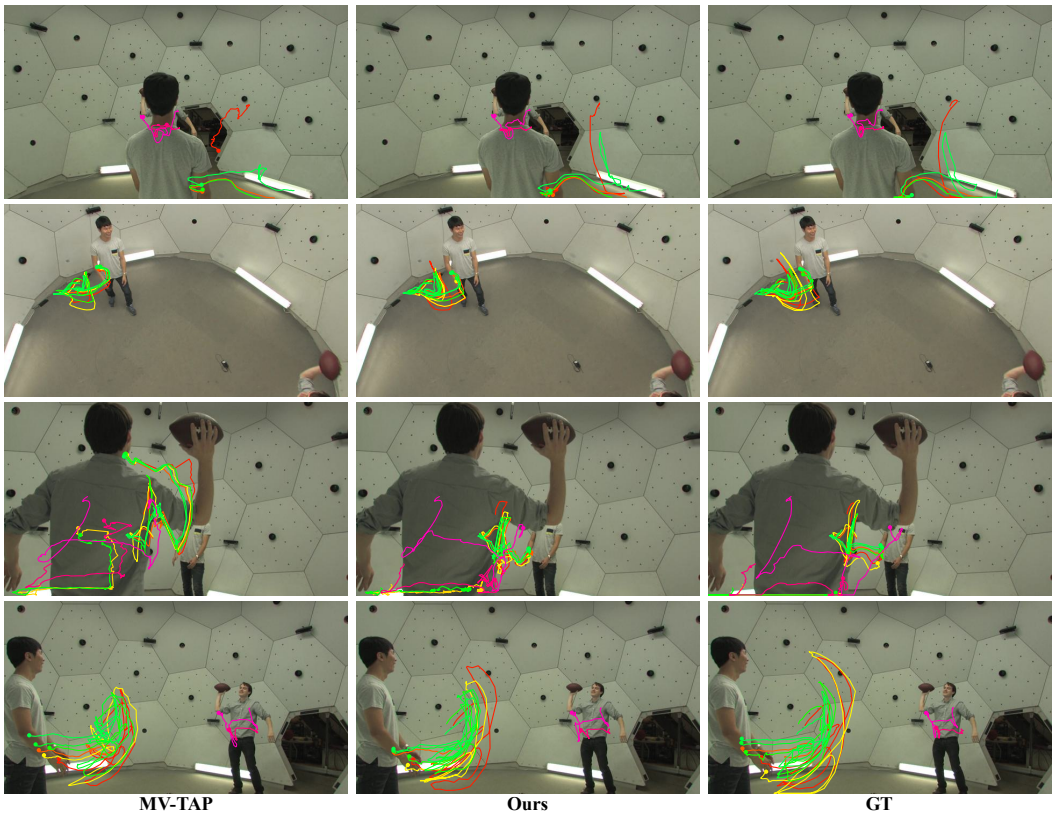


Figure F.2: **2D point tracking on Panoptic Studio** (Joo et al., 2015). Columns show MV-TAP (Koo et al., 2026), GeoMVT, and GT, from left to right. The corresponding video is provided in the supplementary material as `panoptic_2d_output.mp4`.

F.2 3D Point Tracking

Figs. F.5–F.8 compare reconstructed 3D trajectories with MVTracker (Rajič et al., 2025) and GT. Our trajectories are recovered by triangulating multi-view 2D tracks.

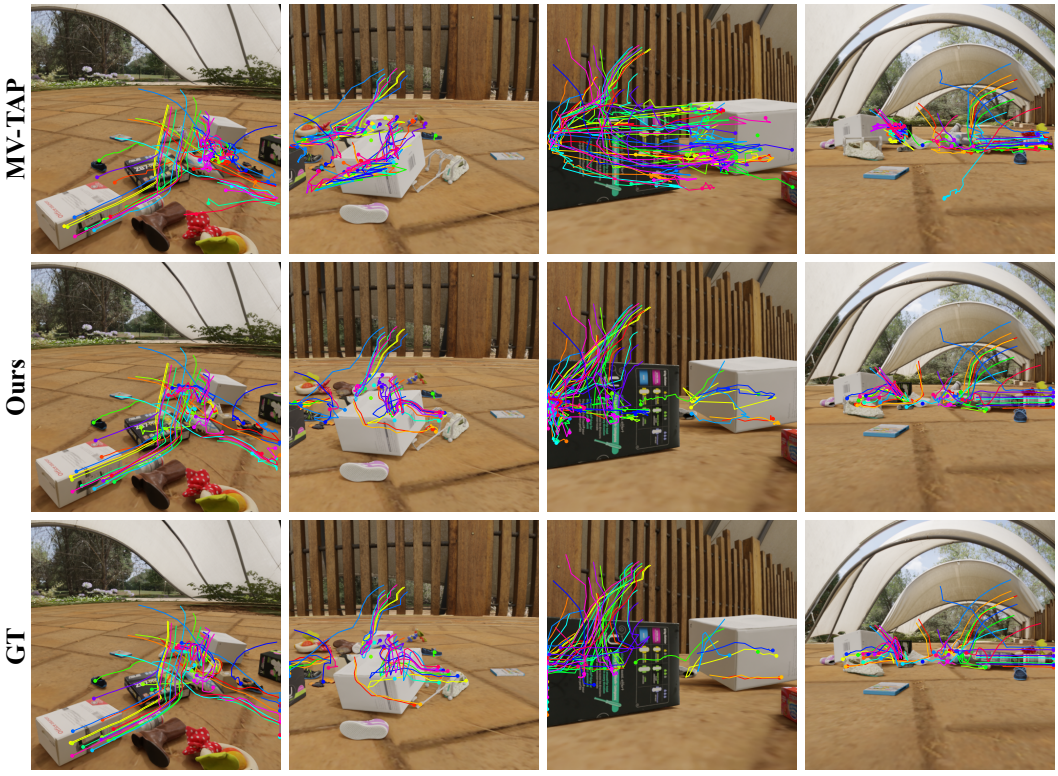


Figure F.3: **2D point tracking on Multi-View Kubric** (Greff et al., 2022; Rajič et al., 2025). Rows show MV-TAP (Koo et al., 2026), GeoMVT, and GT, from top to bottom. The corresponding video is provided in the supplementary material as `kubric_2d_output.mp4`.

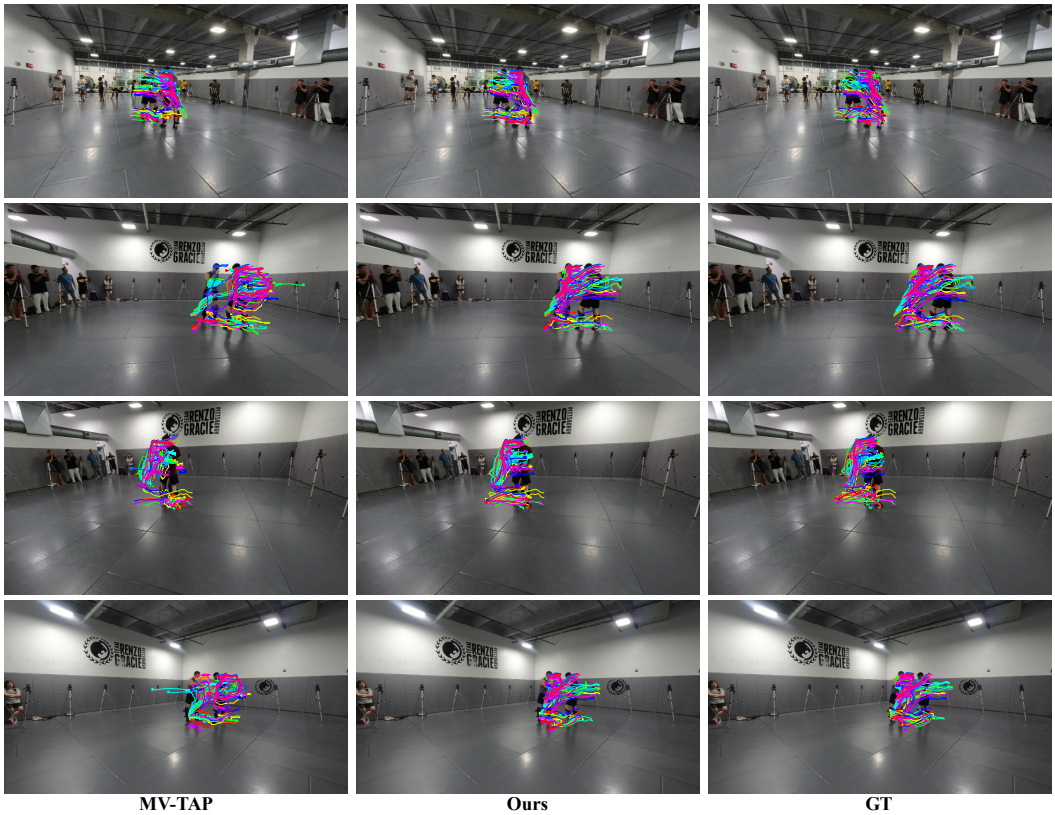


Figure F.4: **2D point tracking on Harmony4D** (Khirodkar et al., 2024). Columns show MV-TAP (Koo et al., 2026), GeoMVT, and GT, from left to right. The corresponding video is provided in the supplementary material as `harmony_2d_output.mp4`.

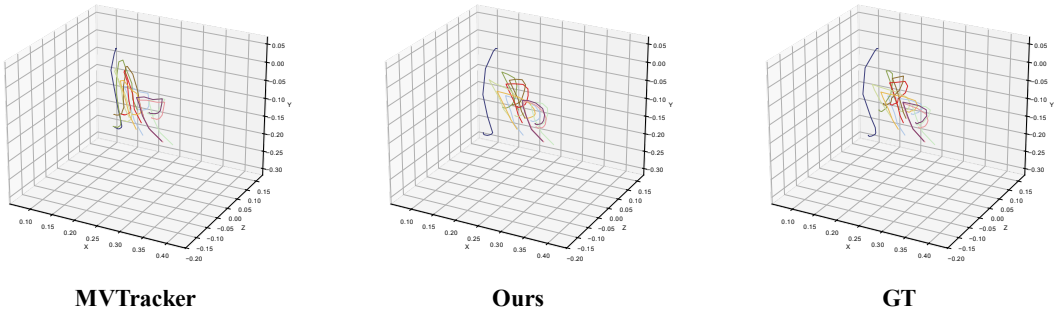
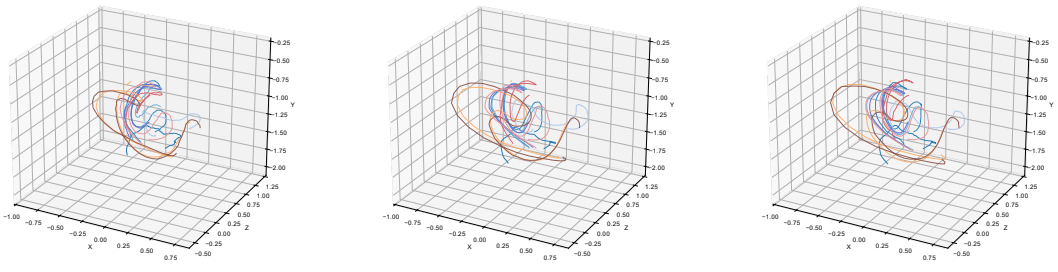
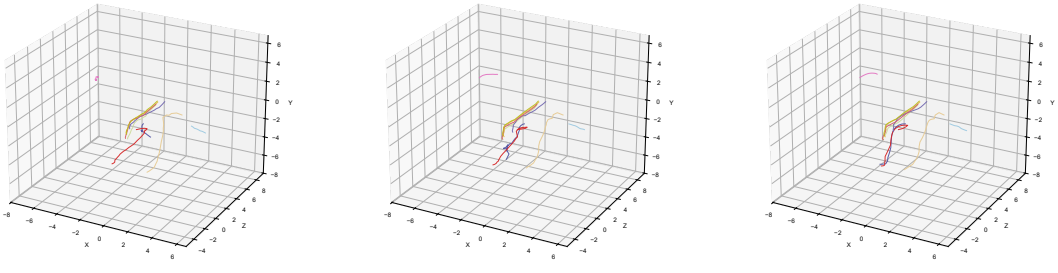


Figure F.5: **3D point tracking on DexYCB** (Chao et al., 2021). Columns show MVTracker (Rajič et al., 2025), GeoMVT, and GT, from left to right. The corresponding video is provided in the supplementary material as `dexycb_3d_output.mp4`.



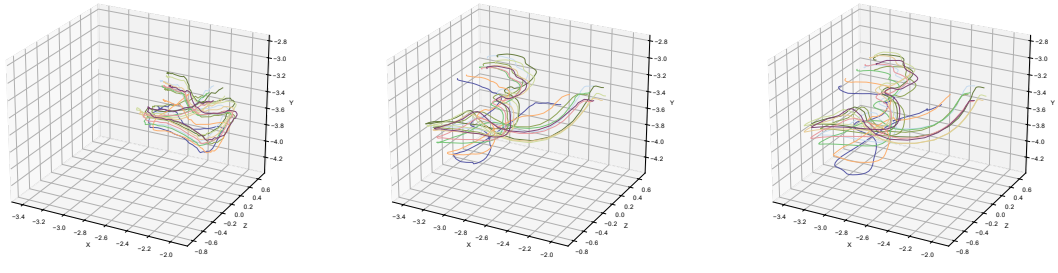
MVTracker **Ours** **GT**

Figure F.6: **3D point tracking on Panoptic Studio** (Joo et al., 2015). Columns show MVTracker (Rajič et al., 2025), GeoMVT, and GT, from left to right. The corresponding video is provided in the supplementary material as `panoptic_3d_output.mp4`.



MVTracker **Ours** **GT**

Figure F.7: **3D point tracking on Multi-View Kubric** (Greff et al., 2022; Rajič et al., 2025). Columns show MVTracker (Rajič et al., 2025), GeoMVT, and GT, from left to right. The corresponding video is provided in the supplementary material as `kubric_3d_output.mp4`.



MVTracker **Ours** **GT**

Figure F.8: **3D point tracking on Harmony4D** (Khirodkar et al., 2024). Columns show MVTracker (Rajič et al., 2025), GeoMVT, and GT, from left to right. The corresponding video is provided in the supplementary material as `harmony_3d_output.mp4`.

G Limitations and Future Work

Scaling to dense multi-view tracking. Our framework constructs local correlation features for each query point and aggregates them through geometry-guided cross-view interactions. Extending this formulation to dense tracking increases computation and memory requirements, as correlations are repeatedly constructed across points, views, and refinement iterations. CoWTracker (Lai et al., 2026) offers an alternative by replacing explicit correlation volumes with feature warping. Inspired by this direction and the joint image feature processing in VGGT (Wang et al., 2025b), future work could explore shared multi-view representations that reduce repeated correlation computation across dense queries. Integrating our geometric reasoning into such representations would provide a path toward dense multi-view tracking while retaining explicit geometric guidance.

Real-world data and generalization. Our model is trained on synthetic sequences, and the real-world benchmarks considered here cover a limited range of scenes and camera configurations. Obtaining reliable trajectories that remain consistent across time and views is costly, requiring accurate calibration and extensive annotation validation, as highlighted by TAPVid-MV (Koppula et al., 2026). Building larger and more diverse real-world multi-view datasets is therefore an important direction. Such data would support both learning from real observations and systematically evaluating generalization to unseen scenes, motions, and camera layouts beyond the synthetic scaling experiments presented here.