

Track4World: Feedforward World-centric Dense 3D Tracking of All Pixels

Jiahao Lu¹, Jiayi Xu¹, Wenbo Hu^{†2}, Ruijie Zhu³, Chengfeng Zhao¹,
Sai-Kit Yeung¹, Ying Shan², and Yuan Liu^{†1}

¹ The Hong Kong University of Science and Technology, Hong Kong, China

² ARC Lab, Tencent IEG, China

³ Nanyang Technological University, Singapore

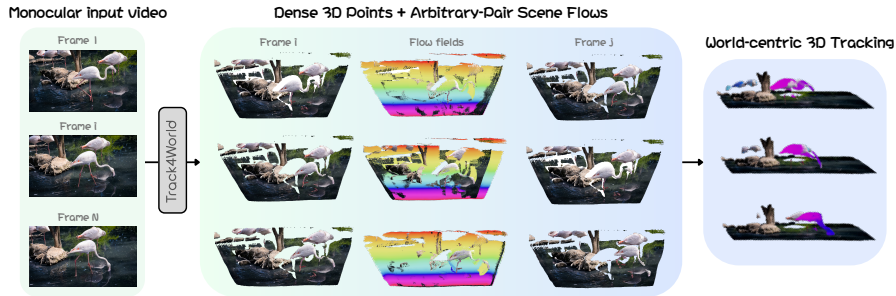


Fig. 1: Track4World estimates dense 3D scene flow of every pixel between arbitrary frame pairs from a monocular video in a global feedforward manner, enabling efficient and dense 3D tracking of every pixel in the world-centric coordinate system.

Abstract. Estimating the 3D trajectory of every pixel from a monocular video is crucial and promising for a comprehensive understanding of the 3D dynamics of videos. Recent monocular 3D tracking works demonstrate impressive performance, but are limited to either tracking sparse points on the first frame or a slow optimization-based framework for dense tracking. In this paper, we propose a feedforward model, called **Track4World**, enabling an efficient holistic 3D tracking of every pixel in the world-centric coordinate system. Built on the global 3D scene representation encoded by a VGGT-style ViT, Track4World applies a novel 3D correlation scheme to simultaneously estimate the pixel-wise 2D and 3D dense flow between arbitrary frame pairs. The estimated scene flow, along with the reconstructed 3D geometry, enables subsequent efficient 3D tracking of every pixel of this video. Extensive experiments on multiple benchmarks demonstrate that our approach consistently outperforms existing methods in 2D/3D flow estimation and 3D tracking, highlighting its robustness and scalability for real-world 4D reconstruction tasks. Project page: <https://jiah-cloud.github.io/Track4World.github.io/>.

Keywords: Feedforward Dense 3D Tracking · 4D Reconstruction

[†] Corresponding authors.

1 Introduction

Reconstructing 4D dynamics of videos [5, 15, 40, 43, 45, 48, 53, 69, 75, 78, 84] is a task of great significance, with far-reaching applications [3, 20, 24, 80] in robotics, animation production, physical law inference, and other fields. To grasp the holistic dynamics of videos, it is essential to predict the dynamic motion of all pixels in 3D space from monocular video inputs. However, this still remains an extremely challenging problem: monocular geometric reconstruction is inherently ill-posed to recover 3D from single-view observations, and further tracking 3D points at different time steps adds an extra layer of complexity.

While recent advances have addressed isolated aspects of this problem, robust and holistic 3D tracking for *every pixel* is still an unsolved problem. Some recent tracking-based methods (e.g., St4RTrack [15], SpatialTrackerV2 (STV2) [68], DELTA [48]), extends feedforward 3D geometry estimation [61, 65] with temporal association for 3D tracking. Though achieving impressive performance and accuracy, these methods are restricted to 3D tracking of the points on the first frame, failing to capture motions for new pixels in subsequent frames. Another work called TrackingWorld [45] enables dense 3D tracking of all frames by fusing multiple modalities, such as 2D flow [28], masks [74], and depth priors [51]. While being effective, this pipeline is computationally expensive and limited by the inability to learn joint spatiotemporal priors, often resulting in temporally inconsistent or suboptimal solutions. Following recent trends in feedforward 3D reconstruction methods, such as DUST3R [65], VGGT [61], and Pi3 [66], designing an efficient feedforward model for holistic 3D tracking of every pixel in a monocular video becomes a promising direction.

In this work, we propose **Track4World**, a feedforward framework designed to estimate 3D tracking for every pixel of every frame in a monocular video in the world-centric coordinate system. Following the feedforward design trends, Track4World is built upon the VGGT-like ViT-style 3D geometry reconstruction framework [41, 61, 66] and associates different frames to estimate dense 3D tracking for every pixel in the world-centric coordinate system.

Designing such an association scheme for the 3D tracking of every pixel is computationally expensive. A naive solution is to generalize the 3D tracking methods, Spatial Tracker V2 (STV2) [68] or DELTA [48], to track every pixel. However, explicitly predicting trajectories for every pixel on all frames is computationally prohibitive. The large number of pixels leads to unaffordable memory and computation consumption during both training and inference, with huge redundant 3D trajectories across different frames.

Instead of directly predicting 3D trajectories of all pixels, Track4World resorts to predicting the two-frame scene flows as the representation of the holistic 3D video dynamics. We draw inspiration from efficient pair-wise motion estimation methods, such as St4RTrack [15] and ZeroMSF [40], to estimate the dense scene flow between arbitrary frame pairs. Then, the estimated pair-wise motion can be easily utilized to construct the 3D tracking of arbitrary pixels in a video. By decomposing the continuous tracking problem into pair-wise scene flow

estimations, we significantly reduce the computational redundancy, transforming this into a more manageable task.

However, effectively adapting a VGGT-like ViT-style 3D reconstruction framework for dense scene flow estimation remains an open problem. A straightforward solution might involve appending a motion decoding head to regress 3D scene flow directly [15, 40]. Yet, such implicit regression approaches are typically data-hungry, requiring massive 3D training datasets and heavily parameterized networks. Moreover, they often struggle to capture fine-grained motion details, leading to suboptimal tracking accuracy. Conversely, state-of-the-art 3D tracking methods such as STV2 [68], and 2D optical flow methods like RAFT [58], have demonstrated that constructing correlation volumes and iteratively updating the motion field yield more precise results. Building on this insight, we take a step back to the effective and accurate correlation-based estimation scheme to incorporate a novel correlation-based iterative 3D scene flow estimation strategy tailored for dense pixel-level prediction within our feedforward framework.

Our proposed iterative correlation scheme differs from all existing correlation-based methods to address the specific challenges of dense 3D tracking.

(1) 2D-to-3D Correlation. The core of our framework is a novel hybrid correlation mechanism that replaces the expensive 3D spatial correlation (which typically requires k -nearest neighbor searches in 3D space followed by cross-attention operations) used in prior works [68, 75]. Instead, we efficiently fuse the 3D geometric feature embeddings from the ViT backbone with 2D pixel-wise correlations, allowing us to compute 3D flow updates rapidly without explicitly constructing heavy 3D spatial correlation. To further make all-pixel estimation tractable, we instantiate this mechanism in a sparse-to-dense manner: rather than running iterative correlation updates at full resolution, we operate on a set of sparse anchor points and recover the dense motion via learned upsampling. While both choices reduce computation, they act along different axes—what the correlations are (image-plane rather than 3D spatial) versus where they are computed (sparse anchors rather than all pixels)—and the former is the architectural core of our design.

(2) 2D-3D Joint Supervision. Our 2D-to-3D correlation mechanism inherently supports the dual prediction of 2D and 3D flows. This structural alignment enables a joint supervision strategy that leverages abundant 2D flow datasets to provide auxiliary training signals for the 3D scene flow task. Consequently, we circumvent the severe scarcity of 3D ground-truth annotations, utilizing 2D flow training to enhance the generalization capability of our model.

(3) Global Scene Flow. Unlike traditional scene flow estimation methods that are limited to adjacent frames, our framework is designed to estimate flows between arbitrary frame pairs within the video sequence, not limiting neighboring frames. By processing the entire video sequence simultaneously, our network leverages global temporal context to resolve local ambiguities, compensating for estimation errors that might occur in isolated frame pairs.

After estimating the accurate pairwise 2D-3D flows for the whole video, Track4World fuses the 3D trajectories in the global coordinate system and then

constructs a holistic 3D tracking of every pixel in the world-centric coordinate system. Extensive experiments show that Track4World provides accurate, flexible, and comprehensive 3D motion estimation, enabling robust dense 3D tracking in the world coordinate system and consistently outperforming existing baselines [30, 40, 56, 68, 78].

2 Related Work

2.1 Video Geometry Estimation

Video geometry estimation aims to generate temporally consistent point maps from videos. Early works focus on optimization-based refinement, e.g., RobustCVD [32] and CasualSAM [79], which optimize depth and camera parameters using geometric constraints. Recent differentiable SLAM methods, such as MegaSaM [39], Uni4D [74], and ViPE [23], incorporate depth priors and correspondence maps for video-level point map prediction. A second line of research explores data-driven approaches [8, 9, 25, 35, 37, 44, 61, 62, 65, 70, 73, 77, 86], leveraging world-centric geometry for robust, consistent estimation. However, VGGT [61] shows that camera-centric depth and pose often outperform world-centric representations. Consequently, recent methods, including GeometryCrafter [71], Pi3 [66], and DA3 [41], focus on camera-centric predictions. Our approach follows this trend by employing a camera-centric representation (camera-centric point clouds and camera poses), from which world-centric reconstruction can be directly derived, while jointly estimating geometry and motion.

2.2 Joint Geometry and Motion Estimation

Joint geometry and motion estimation aims to predict both 3D point clouds and motion trajectories from videos. This is inherently more challenging than geometry estimation alone due to the need for consistent spatiotemporal reasoning. Existing approaches generally follow three directions. First, inspired by 2D point tracking [14, 21, 28, 29, 36, 52], early methods such as SpatialTracker [69] and DELTA [48] represent points in (u, v, d) coordinates, which require known camera intrinsics and may cause geometric inaccuracies. Later works, e.g., TAPIP3D [75] and SpatialTrackerV2 [68], track directly in (x, y, z) space for improved stability, but rely on explicit 3D spatial correlations that are computationally costly and thus restricted to sparse points. Second, methods like St4RTrack [15], POMATO [78], Stereo4D [26], and ZeroMSF [40] reconstruct motion by estimating pairwise point maps or scene flow [54, 59, 72] between two frames via implicit regression. Third, optimization-based methods such as TrackingWorld [45] combine multiple cues to recover dense 3D tracking for all pixels.

Motivated by these methods, we propose a feed-forward, holistic framework for the dense 3D tracking of every pixel. By leveraging a novel 3D correlation scheme, we simultaneously estimate pixel-wise 2D and 3D dense flow between arbitrary frame pairs. The estimated motion, along with the reconstructed 3D geometry,

enables the subsequent efficient computation of continuous 3D trajectories for every pixel within the global coordinate system.

Concurrent works. Several concurrent works have explored similar directions in joint geometry and motion estimation [1, 30, 43, 46, 56, 76, 85]. Among these, TraceAnything [43], Any4D [30], MotionCrafter [85], 4RC [46], DePT3R [1], D4RT [76] and V-DPM [56] also predict motion between timesteps. Track4World introduces two key design choices that distinguish it from these methods. First, we utilize explicit feature correlations to enhance motion prediction accuracy. Second, our novel 2D-lifted correlation architecture alleviates the computational bottleneck inherent in explicit 3D correlation, improving efficiency while enabling the 3D tracking module to benefit from abundant 2D training data.

3 Method

3.1 Overview

Given a video sequence of T frames, $\{\mathbf{I}_i \in \mathbb{R}^{H \times W \times 3} \mid i = 1, \dots, T\}$, Track4World aims to construct a holistic 3D track for every pixel within a world-centric coordinate system (as illustrated in Fig. 2). To achieve this, our framework operates in three stages. First, we extract global scene representations, including geometric features, camera-centric point clouds, and camera poses, using a finetuned geometry encoder initialized from recent state-of-the-art feedforward 3D reconstruction models [41, 63, 66], such as Pi3 or Depth Anything v3 (DA3), (see supplementary material for details). Second, built upon these global representations, our core scene flow decoder predicts dense 3D scene flow between arbitrary pairs of source and target timesteps (i, j) . To reduce computational overhead, this decoder operates in a **sparse-to-dense** manner. Crucially, it introduces a novel **2D-to-3D correlation** module that completely avoids expensive 3D spatial correlation, which in turn enables a **2D-3D joint supervision** strategy to circumvent the severe scarcity of 3D ground-truth annotations. Finally, Track4World fuses these pairwise 3D scene flows to formulate the ultimate holistic 3D tracks.

3.2 Scene Flow Decoder

Building upon the global scene representations (camera-centric pixel-aligned point clouds $\mathbf{P}_i$, camera poses $\mathbf{T}_i$, and geometry features $\mathbf{F}_i$) extracted by the ViT backbone [41, 63, 66], the scene flow decoder predicts dense 3D scene flow between arbitrary frame pairs (i, j) for the whole video, allowing subsequent 3D tracking in the world-centric coordinate system.

Anchor Point Preparation. To manage the computational overhead of all-pixel flow estimation, we avoid iterative correlation updates at the original resolution and instead operate on a set of sparse anchor points. We augment the geometry tokens $\mathbf{F}_i$ with temporal context via global self-attention to obtain refined features $\hat{\mathbf{F}}_i$, while a lightweight context encoder extracts semantic features $\bar{\mathbf{F}}_i$ from the input images. We then downsample the point clouds $\mathbf{P}_i$ and both feature maps

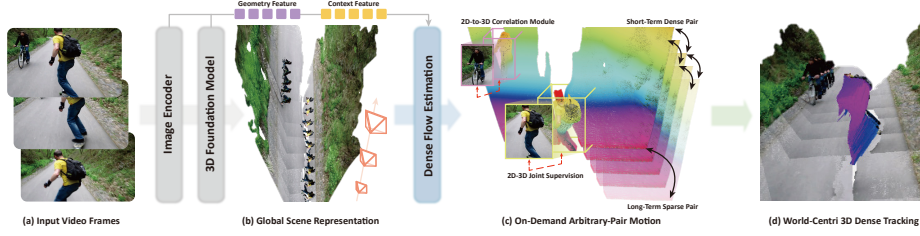


Fig. 2: Overview. Given (a) the input video frames, Track4World first extracts (b) global scene representations (geometric embeddings, point clouds, and camera poses). (c) A sparse-to-dense scene flow decoder then predicts 2D-3D joint flows between arbitrary timesteps, which applies a novel 2D-to-3D correlation scheme to improve efficiency and allows 2D-3D joint supervision. (d) The pairwise flows are ultimately fused to establish holistic world-centric 3D tracking.

$(\hat{\mathbf{F}}_i, \tilde{\mathbf{F}}_i)$ to 1/8 resolution, which are upsampled back to the full resolution at the end.

2D-to-3D Correlation for Flow Estimation. This section aims to estimate the 3D scene flow on the anchor points using correlation. Previous methods, like STV2 [68] or TAPIP3D [75], demonstrate accurate 3D tracking by the correlation scheme but consume lots of computation in finding k -nearest point in 3D space for correlation as shown in Fig. 3 (top), making these methods limited to tracking only sparse 3D points. To scale this for dense 3D flow estimation, we introduce an efficient 2D-to-3D correlation module that elegantly fuses geometric features with image-plane correlations and lifts it for 3D flow estimation.

In the proposed 2D-to-3D correlation module, both 2D and 3D flows are estimated with several update iterations to compute the correlations and refine the target positions. In each iteration, the 2D flow and 3D flow are updated sequentially in a coupled manner, as shown in Fig. 3 (bottom). For a given source-target image pair (i, j) , the 2D flow is updated first, serving as a reliable spatial basis for the subsequent

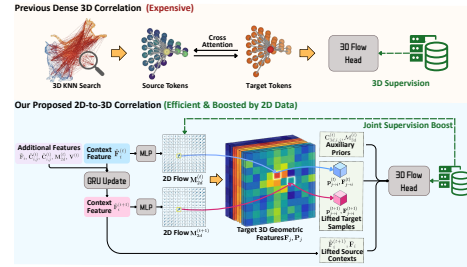


Fig. 3: Comparison of correlation mechanisms. Prior methods (top) rely on explicit k -nearest neighbor searches and cross-attention in 3D space, leading to high computational costs. In contrast, our proposed method (bottom) anchors 3D updates directly to intermediate image-plane correlations. We annotate the construction of the RAFT-style 2D correlation from context and geometric features, and how the 3D features are assembled within the 3D flow update process. This design significantly improves computational efficiency and allows the 3D tracking module to be effectively boosted by abundant 2D training data.

update of the 3D flow. Both 2D flows $\mathbf{M}_{2d}^{(0)}$ and 3D flows $\mathbf{M}_{3d}^{(0)}$ are initialized to 0 everywhere and updated with the following scheme.

2D Iterative Correlation. We first construct geometric feature correlation volumes $\hat{\mathbf{C}}_{i,j}^{(t)}$ and semantic feature correlation volumes $\tilde{\mathbf{C}}_{i,j}^{(t)}$ between the source position on the source image and the neighborhood positions on the target image using the current 2D flow $\mathbf{M}_{2d}^{(t)}$. At the t -th iteration, a GRU-based operator updates the hidden context feature from $\hat{\mathbf{F}}_i^{(t)}$ to $\hat{\mathbf{F}}_i^{(t+1)}$, which in turn drives the update of the 2D flow $\mathbf{M}_{2d}^{(t)}$ and the visibility confidence $\mathbf{V}^{(t)}$ (whose two channels encode the visibility and the tracking confidence, respectively):

$$\hat{\mathbf{F}}_i^{(t+1)} = \text{GRU}(\hat{\mathbf{F}}_i^{(t)}, \tilde{\mathbf{F}}_i, \hat{\mathbf{C}}_{i,j}^{(t)}, \tilde{\mathbf{C}}_{i,j}^{(t)}, \mathbf{M}_{2d}^{(t)}, \mathbf{V}^{(t)}), \quad (1)$$

$$\mathbf{M}_{2d}^{(t+1)} = \mathbf{M}_{2d}^{(t)} + \text{MLP}(\hat{\mathbf{F}}_i^{(t+1)}), \quad (2)$$

$$\mathbf{V}^{(t+1)} = \mathbf{V}^{(t)} + \text{MLP}(\hat{\mathbf{F}}_i^{(t+1)}). \quad (3)$$

This 2D branch follows the GRU-based recurrent update scheme of RAFT [58], but differs in two ways: it jointly exploits geometric ($\hat{\mathbf{C}}_{i,j}^{(t)}$) and semantic ($\tilde{\mathbf{C}}_{i,j}^{(t)}$) correlations, and it adopts a temporal aggregator (Sec. 3.3) to incorporate long-range temporal context beyond two frames. Nevertheless, the 2D branch mainly provides a spatial basis; our core contribution lies in the 3D flow module below, which brings 3D awareness and in turn benefits both 2D and 3D estimation. Then, the 2D flow update is used as a basis for 3D flow update as follows.

Lifting for 3D Flow Iterative Estimation. With the 2D flow $\mathbf{M}_{2d}^{(t)}$ and $\mathbf{M}_{2d}^{(t+1)}$, we can retrieve the corresponding 2D target positions on iterations t and $t+1$. Then, by interpolating on the extracted global point maps $\mathbf{P}_j$, we can warp the target-frame geometry to the source frame i and compute 3D coordinates $\mathbf{p}_{j \rightarrow i}^{(t)}$ and $\mathbf{p}_{j \rightarrow i}^{(t+1)}$ for these two target positions correspondingly. Then, $\mathbf{p}_{j \rightarrow i}^{(t+1)} - \mathbf{p}_{j \rightarrow i}^{(t)}$ can be an initial estimation of the 3D flow update $\Delta \mathbf{M}_{3d}^{(t)}$. However, this only lifts the 2D flow update to 3D without considering any additional 3D geometric correlation. We further fuse all 3D information together using a 3D flow head $\mathcal{H}_{3d}$ to predict the 3D flow update $\Delta \mathbf{M}_{3d}^{(t)}$ by:

$$\Delta \mathbf{M}_{3d}^{(t)} = \mathcal{H}_{3d} \left(\underbrace{\mathbf{p}_{j \rightarrow i}^{(t)}, \mathbf{p}_{j \rightarrow i}^{(t+1)}, \mathbf{F}_{j \rightarrow i}^{(t)}, \mathbf{F}_{j \rightarrow i}^{(t+1)}}_{\text{Lifted Target Samples}}, \underbrace{\hat{\mathbf{F}}_i^{(t+1)}, \mathbf{F}_i}_{\text{Lifted Source Contexts}}, \underbrace{\mathbf{C}_{3d,i,j}^{(t)}, \mathcal{M}_{3d}^{(t)}}_{\text{Auxiliary Priors}} \right), \quad (4)$$

where $\mathbf{F}_{j \rightarrow i}^{(t)}$ and $\mathbf{F}_{j \rightarrow i}^{(t+1)}$ denote the geometric features of $\mathbf{F}_j$ interpolated at the predicted target positions in two consecutive iterations and warped to the source frame i , providing the 3D shape information of these two predicted target positions, while $\hat{\mathbf{F}}_i^{(t+1)}, \mathbf{F}_i$ are the source geometric features on the anchor points. Here, $\mathbf{F}_i$ is the geometric feature extracted by the backbone and $\hat{\mathbf{F}}_i$ is its temporally enhanced version via global attention, with the superscript t denoting

the iteration index (i.e., $\hat{\mathbf{F}}_i^{(t+1)}$ is the next-iteration update of $\hat{\mathbf{F}}_i^{(t)}$). We further compute the 3D spatial similarity through a correlation $\mathbf{C}_{3d,i,j}^{(t)}$ that is built in a RAFT-style manner but operates on *3D point coordinates* instead of learned feature embeddings. The **source** is the transformed point map $(\mathbf{P}_i + \mathbf{M}_{3d}^{(t)})$ and the **target** is the point map $\mathbf{P}_j$, both at $1/8$ resolution. We construct a multi-scale 3D pyramid correlation volume from pixel-wise geometric distances and apply a RAFT-style lookup that samples a $(2r+1)^2$ neighborhood centered at the current 2D flow prediction $\mathbf{M}_{2d}^{(t)}$ across L pyramid levels, yielding $L(2r+1)^2$ channels per pixel. Unlike the 2D branch that correlates high-dimensional appearance features for visual similarity, $\mathbf{C}_{3d,i,j}^{(t)}$ explicitly models geometric consistency in 3D coordinate space, providing informative cues for refining the 3D flow. Additionally, we leverage the historical 3D flow $\mathbf{M}_{3d}^{(t)}$ to construct a trajectory prior $\mathcal{M}_{3d}^{(t)}$, which constrains the smoothness of the flow update. Then, the 3D flows are updated with $\mathbf{M}_{3d}^{(t+1)} \leftarrow \mathbf{M}_{3d}^{(t)} + \Delta\mathbf{M}_{3d}^{(t)}$. This 2D-to-3D correlation scheme allows us to accurately and iteratively update both 2D and 3D flows simultaneously while keeping the computation complexity manageable even for all anchor points from all frames.

Dense Scene Flow Recovery. In the final step, we recover the dense scene flow for the entire image. The low-resolution ($1/8$) flow maps are upsampled to full resolution via a learned convex upsampling, where decoded contextual weights form each high-resolution flow vector as a convex combination of its coarse neighbors (detailed in the supplementary material). To maximize the utility of our 2D-lifted architecture, we introduce a hybrid unprojection scheme. Because the intermediate 2D flow captures image-plane dynamics with high precision, we directly combine it with the Z-axis displacement from the 3D scene flow. Using camera intrinsics derived from the predicted point clouds, we project this combined motion into (x, y, z) space to produce the final refined 3D scene flow. The benefits of this hybrid design are analyzed in Tab. 7.

3.3 Discussion

Computational Efficiency. Prior works [68, 75] heavily rely on explicit 3D spatial correlations, which typically require a computationally expensive k -nearest neighbor (k -NN) search in 3D space for each query point, followed by a cross-attention mechanism with the retrieved target points. In contrast, our 3D flow estimation is fundamentally driven by the proposed 2D-lifted correlation module. This design completely bypasses both the 3D k -NN search and the heavy cross-attention operations, significantly alleviating the computational burden. Specifically, while traditional 3D point-based matching incurs a theoretical complexity of at least $\mathcal{O}(N \log N + N \cdot k)$ (or $\mathcal{O}(N^2)$ for dense global attention) for N points, our warp-sampling mechanism operates with a strict $\mathcal{O}(N)$ complexity via direct coordinate-based lookups from the 2D flow.

2D-3D Joint Supervision. This structural alignment naturally supports dual supervision. Unlike prior arts [68, 75] that define correlations strictly in 3D space, our architecture anchors the 3D updates directly to image-plane correlations.

This allows us to apply auxiliary training signals from abundant 2D datasets directly to the intermediate 2D motion priors. By leveraging massive 2D data to guide the lifting process, we effectively circumvent the severe scarcity of 3D ground-truth annotations and significantly enhance the generalization capability of our model.

Global Trajectory Inference. Leveraging the predicted arbitrary-pair dense flows, our framework seamlessly generalizes to global 3D tracking. For long-range point tracking, we infer flows from a reference frame to all subsequent frames. This process is refined by a *temporal aggregator* that applies attention over concatenated flow features to enforce temporal consistency. In contrast, for dense tracking of every pixel, we compute and chain consecutive frame-to-frame flows, yielding continuous 3D trajectories in the global coordinate system.

Short-to-Long-Term Supervision. During training, we implement a variable-stride sampling strategy: the model is simultaneously supervised by dense flow ground truth over varying temporal intervals and sparse trajectory annotations across the entire sequence. By optimizing these short- and long-term objectives jointly, the network learns to reconcile local motion precision with global trajectory consistency, constructing a holistic 3D track for every pixel.

4 Experiments

Implementation details. Notably, we employ tailored training strategies to fine-tune the base geometric model, leading to refined point map and camera pose estimations as well. Unless specified, all reported results use DA3 [41] as the default backbone initialization. Implementation details about specific fine-tuning strategies are provided in the supplementary materials.

Experiment overview. To comprehensively evaluate our framework, we conduct experiments across a variety of tasks, including 2D/3D flow estimation, 2D/3D tracking, point map prediction, and camera pose estimation. Furthermore, we present qualitative results on diverse in-the-wild videos and conduct extensive ablation studies to validate the effectiveness of each proposed module.

4.1 Scene and Optical Flow Estimation

Evaluation datasets. We evaluate our model on one in-domain dataset, Kubric-3D val [18], and two out-of-domain datasets, KITTI [16] and BlinkVision [38]. All three datasets provide ground-truth optical flow and scene flow annotations. For the Kubric-3D dataset, we sample two evaluation settings: a *short-range* setting where the source and reference frames are 4 frames apart, and a *long-range* setting where they are 16 frames apart, to assess flow estimation performance under different temporal gaps. For the BlinkVision [38] dataset, the source–target frame pair is randomly sampled.

Evaluation metrics. For scene flow estimation, we follow previous works [19, 40, 42, 59] and report the standard metrics: End Point Error (EPE3D), Accuracy Strict (AccS), and Accuracy Relax (AccR). Additionally, we evaluate the quality

Method	In domain											
	Kubric-3D val [18] (short)						Kubric-3D val (long)					
	Abs Rel $\downarrow$ $\delta < 1.25$ $\uparrow$	EPE3D $\downarrow$	AccS $\uparrow$	AccR $\uparrow$	EPE2D $\downarrow$	AccS _{2D} $\uparrow$	Abs Rel $\downarrow$ $\delta < 1.25$ $\uparrow$	EPE3D $\downarrow$	AccS $\uparrow$	AccR $\uparrow$	EPE2D $\downarrow$	AccS _{2D} $\uparrow$
RAFT [58]	/	/	/	/	/	6.7974	/	/	/	/	/	/
GMFlowNet [81]	/	/	/	/	/	7.0390	/	/	/	/	/	/
SEA-RAFT [67]	/	/	/	/	/	9.5794	/	/	/	/	/	/
RAFT-3D [59]	0.0649	0.9344	0.6170	0.0015	0.0078	40.4480	0.1245	0.8422	1.5652	0.0001	0.0010	51.9034
OpticalExpansion [72]	0.2170	0.6266	0.2093	0.2890	0.4760	19.6471	0.2177	0.6316	0.7037	0.1062	0.1903	51.9049
POMATO [78]	0.1525	0.8329	0.9672	0.0565	0.1696	/	0.1761	0.7760	1.6925	0.0148	0.0564	58.7148
ZeroMSF [40]	0.0860	0.9196	0.3528	0.1867	0.3413	/	0.1208	0.8609	1.2182	0.0475	0.0895	83.6966
Any4D [30]	0.0585	0.9547	0.3908	0.1610	0.2893	/	0.1017	0.8770	1.2442	0.0429	0.0855	0.0004
V-DPM [56]	0.0716	0.9010	0.4087	0.1442	0.2491	/	0.1155	0.8205	1.2620	0.0407	0.0803	0.0255
Ours	0.0344	0.9719	0.1537	0.5494	0.7460	1.8685	0.0472	0.9371	0.4808	0.3247	0.5491	15.0906
Method	Out of domain											
	KITTI [16]						BlinkVision [38]					
	Abs Rel $\downarrow$ $\delta < 1.25$ $\uparrow$	EPE3D $\downarrow$	AccS $\uparrow$	AccR $\uparrow$	EPE2D $\downarrow$	AccS _{2D} $\uparrow$	Abs Rel $\downarrow$ $\delta < 1.25$ $\uparrow$	EPE3D $\downarrow$	AccS $\uparrow$	AccR $\uparrow$	EPE2D $\downarrow$	AccS _{2D} $\uparrow$
RAFT [58]	/	/	/	/	/	5.4150	/	/	/	/	/	/
GMFlowNet [81]	/	/	/	/	/	4.6977	/	/	/	/	/	/
SEA-RAFT [67]	/	/	/	/	/	4.8863	/	/	/	/	/	/
RAFT-3D [59]	0.1619	0.8413	0.3837	0.0118	0.0678	54.0938	0.1426	0.8455	0.6690	0.0454	0.1280	14.1255
OpticalExpansion [72]	0.2764	0.4302	0.2410	0.1353	0.2612	8.8808	0.3372	0.4099	0.4406	0.2091	0.3116	12.0176
POMATO [78]	0.2752	0.4359	0.2602	0.1127	0.2156	/	0.2089	0.6569	0.4038	0.1522	0.2870	20.2384
ZeroMSF [40]	0.2064	0.5913	0.1823	0.1695	0.3481	/	0.1934	0.6620	0.3937	0.1913	0.2991	5.5697
Any4D [30]	0.2398	0.4974	0.1856	0.1429	0.2931	/	0.2218	0.6125	0.9238	0.1242	0.1818	85.4975
V-DPM [56]	0.1469	0.7981	0.4462	0.1180	0.1608	/	0.2117	0.6449	1.1476	0.1079	0.1547	0.0018
Ours	0.0707	0.9570	0.0742	0.6929	0.8238	2.5722	0.0371	0.9768	0.1135	0.5091	0.7144	7.5632

Table 1: Evaluation on scene and optical flow estimation. We evaluate our model on one in-domain dataset: Kubirc-3D val [18] and two out-of-domain datasets: KITTI [16], BlinkVision [38]. **Best** results are highlighted in darker blue, and **second best** in lighter blue.

of the previous-frame point cloud and the transformed point cloud obtained by applying the predicted scene flow. We adopt the absolute relative error (Abs Rel) and the percentage of inlier points ($\delta < 1.25$) for this assessment. To ensure a fair comparison, both predicted point clouds are aligned to the ground truth by optimizing a shared scale factor and translation. This scale factor is further applied to align the predicted scene flow before computing EPE3D, AccS, and AccR. This alignment procedure follows the method described in MoGe [63]. For optical flow estimation, we follow [58, 67, 81] to report 3 metrics, EPE2D, AccS_{2D}, AccR_{2D}.

Comparison with existing methods. We compare our method against several representative approaches: three optical flow methods (RAFT [58], GMFlowNet [81], and SEA-RAFT [67]), two scene flow methods (RAFT-3D [59] and OpticalExpansion [72]), two joint geometry and scene flow methods (POMATO [78] and ZeroMSF [40]), as well as concurrent works Any4D [30] and V-DPM [56]. For RAFT-3D, we provide the depth and camera parameters predicted by VGGT [61] as input, while OpticalExpansion uses the point clouds predicted by POMATO. Tab. 1 presents a quantitative comparison, demonstrating that our method consistently outperforms these prior works across all four datasets. This validates the effectiveness of the proposed 2D-3D correlation for the flow estimation task.

4.2 3D Tracking Estimation

Evaluation datasets. Following prior works [48, 68, 69, 78], we evaluate our method on ADT [50], PStudio [27], and DriveTrack [2] benchmarks from the TAPVid-3D [33] dataset, as well as the validation set of the PointOdyssey [83] dataset. Consistent with POMATO [78], we reformulate the datasets by projecting

Method	PointOdyssey [83]		ADT [50]		PStudio [27]		DriveTrack [2]		Avg.	
	L-16	L-50	L-16	L-50	L-16	L-50	L-16	L-50	L-16	L-50
<i>Camera coordinate 3D tracking</i>										
SpatialTracker* [69]	0.3116	0.2977	0.4962	0.4692	0.5390	0.4991	0.2529	0.2502	0.3999	0.3791
DELTA* [48]	0.3529	0.3412	0.5116	0.4952	0.5922	0.5533	0.2704	0.2701	0.4317	0.4150
STV2 [†] [68]	0.1864	0.1785	0.2400	0.2330	0.3784	0.3690	0.1711	0.1725	0.2400	0.2383
MASt3R [35]	0.3546	0.3253	0.3368	0.3029	0.3293	0.2956	0.2767	0.2559	0.3244	0.2949
MonST3R [77]	0.3912	0.3860	0.3694	0.3429	0.3511	0.3381	0.3056	0.2787	0.3543	0.3364
POMATO [78]	0.4816	0.4623	0.5338	0.5299	0.5163	0.4726	0.4237	0.4329	0.4888	0.4744
ZeroMSF [40]	0.4214	0.3887	0.5382	0.4635	0.5083	0.4524	0.4448	0.4513	0.4782	0.4390
Ours	0.5397	0.5268	0.6501	0.6091	0.5948	0.5423	0.5003	0.5092	0.5712	0.5469
<i>World coordinate 3D tracking</i>										
STV2 [†] [68]	0.1925	0.1763	0.2456	0.2163	0.3790	0.3689	0.1711	0.1725	0.2470	0.2335
POMATO [‡] [78]	0.4425	0.3905	0.3611	0.3548	0.5166	0.4713	0.4227	0.4210	0.4357	0.4094
ZeroMSF [‡] [40]	0.4053	0.3505	0.4530	0.3563	0.4828	0.4386	0.4474	0.4382	0.4471	0.3959
Any4D [30]	0.4769	0.4174	0.4460	0.3717	0.5707	0.5066	0.5235	0.5079	0.5043	0.4509
V-DPM [56]	0.4848	0.4233	0.4783	0.3759	0.6084	0.5795	0.4854	0.4817	0.5142	0.4668
Ours	0.5345	0.5162	0.6250	0.5622	0.5946	0.5422	0.5003	0.5087	0.5636	0.5323

Table 2: 3D tracking estimation. We report the APD metric to evaluate 3D point tracking on the PointOdyssey [83], ADT [50], PStudio [27], and DriveTrack [2] datasets. L-16 and L-50 indicate tracking within the temporal length of 16 and 50 frames, respectively. * indicates that ground-truth camera intrinsics are used. † indicates that bundle adjustment is not used. ‡ denotes methods using camera poses estimated by VGGT [61].

all query points within a temporal window to the first frame. The window sizes are 16 and 50 in the evaluation.

Evaluation metrics. We use the Average Percent Deviation (APD) metric, which measures the percentage of points whose predicted positions fall within a specified threshold relative to the ground-truth depth. APD provides a direct and quantitative assessment of the tracking accuracy.

Comparison with existing methods. We compare our method with representative 3D tracking and geometry/scene flow approaches. The baselines include 3D trackers SpatialTracker [69] and DELTA [48] (both using ground-truth intrinsics), as well as STV2 [68]. We further compare against geometry and scene flow methods MASt3R [35], MonST3R [77], POMATO [78], and ZeroMSF [40], along with concurrent works Any4D [30] and V-DPM [56]. We evaluate 3D tracking results in Tab. 2, following the approach described in Sec. 3.3 to achieve dense tracking and perform sampling based on the UV coordinates of the query points. The results demonstrate that our method consistently outperforms most existing 3D tracking and pairwise scene flow approaches across multiple datasets, in both camera-centric and world-centric coordinate systems, highlighting its effectiveness and generalization.

4.3 2D Tracking Estimation

Evaluation datasets. Following prior works [13, 28, 29], we evaluate our method on three datasets: Kinetics [6], RoboTAP [60], and RGB-Stacking [34]. Kinetics comprises 1,144 YouTube videos from the Kinetics-700–2020 validation set, featuring complex camera motion and cluttered backgrounds, with an average of 26 tracks per video. RoboTAP consists of 265 real-world robotic manipulation videos, with an average duration of 272 frames per video. RGB-Stacking is a synthetic dataset of robotic videos, characterized by numerous texture-less regions that make tracking particularly challenging.

Evaluation metrics. We evaluate tracking performance using the TAP-Vid metrics [12]. Specifically, we report Occlusion Accuracy (OA), measuring occlusion prediction as a binary classification; δ_{vis}^{avg} , the fraction of visible points tracked within 1, 2, 4, 8, and 16 pixels, averaged across thresholds; and Average Jaccard (AJ), which jointly assesses geometric and occlusion prediction accuracy.

Comparison with existing methods. We compare our method with representative 2D tracking approaches, including CoTracker3 [28], LocoTrack [10], the dedicated dense 2D tracker AllTracker [21], and other state-of-the-art methods. In Tab. 3, we present the 2D tracking results from our 2D branch. Leveraging valuable geometric cues and joint training on multi-modal, multi-source data, our method achieves performance comparable to these state-of-the-art 2D tracking approaches. Notably, despite our joint 3D optimization, our 2D branch even remains on par with AllTracker, indicating that incorporating 3D supervision does not degrade 2D tracking accuracy.

Method	Kinetics [6]			RoboTAP [60]			RGB-S [34]		
	AJ $\uparrow$	$\delta_{vis}^{avg}\uparrow$	OA $\uparrow$	AJ $\uparrow$	$\delta_{vis}^{avg}\uparrow$	OA $\uparrow$	AJ $\uparrow$	$\delta_{vis}^{avg}\uparrow$	OA $\uparrow$
PIPs++ [83]	/	63.5	/	/	63.0	/	/	58.5	/
TAPIR [14]	49.6	64.2	85.0	59.6	73.4	87.0	55.5	69.7	88.0
CoTracker [29]	49.6	64.3	83.3	58.6	70.6	87.0	67.4	78.9	85.2
TAPTR [36]	49.0	64.4	85.2	60.1	75.3	86.9	60.8	76.2	87.0
LocoTrack [10]	52.9	66.8	85.3	62.3	76.2	87.1	69.7	83.2	89.5
BootsTAPIR [13]	54.6	68.4	86.5	64.9	80.1	86.3	70.8	83.0	89.9
CoTracker3 [28]	55.8	68.5	88.3	66.4	78.8	90.8	71.7	83.6	91.1
AllTracker [21]	59.2	72.2	90.4	71.1	82.7	92.9	79.5	89.6	91.8
Ours	59.1	71.3	90.6	70.9	81.8	93.3	78.2	88.5	92.3

Table 3: 2D tracking estimation. We evaluate our model on three datasets: Kinetics [6], RoboTAP [60], and RGB-Stacking [34].

4.4 Geometry and Camera Pose Estimation

Method	Params	GMU Kitchen [17]		Monkaa [47]		Sintel [4]		ScanNet test [11]		Kubric-3D val [18]		KITTI [16]		Tum [55]		Avg.	
		Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$	Abs Rel $\downarrow$ $\delta < 1.25 \uparrow$				
MoGe [63]	314M	0.1728	0.6725	0.2069	0.6317	0.2181	0.6615	0.1194	0.8447	0.0852	0.9294	0.0801	0.9374	0.1563	0.7663	0.1484	0.7776
VGIT [61]	1.26B	0.2530	0.4454	0.2009	0.6678	0.2004	0.7303	0.0763	0.9242	0.0316	0.9733	0.1245	0.8517	0.1278	0.8236	0.1449	0.7738
MoGe-2 [64]	331M	0.0654	0.9391	0.1904	0.6797	0.2058	0.6903	0.0626	0.9673	0.1184	0.8592	0.0776	0.9627	0.1156	0.8694	0.1194	0.8525
MapAnything [31]	563M	0.1090	0.9279	0.2012	0.7379	0.2079	0.7141	0.0463	0.9833	0.0684	0.9475	0.1016	0.9144	0.1132	0.9458	0.1208	0.8816
P3 [66]	1.29B	0.0458	0.9538	0.0774	0.9256	0.1489	0.7899	0.0750	0.9532	0.0337	0.9817	0.0866	0.8877	0.0403	0.9680	0.0738	0.9200
DA3 [41]	1.36B	0.0551	0.9343	0.1131	0.8761	0.1575	0.8064	0.0222	0.9917	0.0431	0.9599	0.0404	0.9861	0.0886	0.9515	0.0743	0.9294
Ours	1.38B	0.0431	0.9341	0.0853	0.9123	0.1261	0.8291	0.0288	0.9872	0.0191	0.9939	0.0268	0.9877	0.0572	0.9638	0.0552	0.9440

Table 4: Evaluation on point map estimation. Results are aligned with the ground truth by optimizing a shared scale factor and shift across the entire video.

Point Map Estimation. We evaluate geometry estimation across seven diverse datasets: indoor (GMU Kitchen [17], ScanNet [11], TUM [55]), outdoor (KITTI [16]), and synthetic (Monkaa [47], Sintel [4], Kubric-3D [18]). Following established practices [22, 63], predictions are aligned with ground truth via a shared scale and shift before computing the absolute relative error (Abs Rel) and inlier percentage ($\delta < 1.25$). Compared to state-of-the-art baselines (e.g.,

MoGe [63], Pi3 [66], DA3 [41]), our method achieves highly competitive performance (Tab. 4), providing a robust geometric foundation for downstream 3D flow estimation.

Camera Pose Estimation. We benchmark camera pose on Bonn [49] and Sintel [4]. For Sintel, we strictly follow prior protocols [7, 82] by filtering out static or straight-motion sequences, leaving 14 dynamic sequences for evaluation. We report Absolute Translation Error (ATE), Relative Translation Error (RTE), and Relative Rotation Error (RRE). As shown in Tab. 5, our approach demonstrates strong effectiveness compared to both joint depth-pose estimators (e.g., DA3 [41]) and advanced joint depth-pose-motion estimators (e.g., STV2 [68]).

Method	Sintel [4]			Bonn [49]		
	ATE↓	RTE↓	RRE↓	ATE↓	RTE↓	RRE↓
Align3R [44]	0.128	0.042	0.432	0.023	0.007	0.620
CUT3R [62]	0.217	0.070	0.636	0.035	0.014	1.212
VGGT [61]	0.167	0.062	0.490	0.051	0.011	1.038
MapAnything [31]	0.227	0.111	2.047	0.026	0.014	0.668
Pi3 [66]	0.088	0.043	0.299	0.012	0.011	0.612
DA3 [41]	0.124	0.061	0.331	0.010	0.011	0.638
POMATO [57]	0.209	0.064	0.694	0.041	0.017	0.832
STV2 [68]	0.133	0.057	0.641	0.019	0.015	0.701
Ours	0.119	0.054	0.309	0.009	0.009	0.604

Table 5: Camera pose estimation. We evaluate our model on two datasets: Sintel [4] and Bonn [49]. ‡ indicates using ground-truth camera intrinsics as input.

4.5 Ablation Study

Method	Point Map		Scene Flow			3D Tracking	
	Abs Rel ↓	$\delta < 1.25 \uparrow$	EPE3D ↓	Abs Rel ↓	$\delta < 1.25 \uparrow$	L-16 ↑	L-50 ↑
Ours (MoGe [63])	0.0973	0.8921	0.3180	0.0680	0.9479	0.5447	0.5135
Ours (Pi3 [66])	0.0492	0.9537	0.2569	0.0548	0.9645	0.5734	0.5424
Ours (DA3 [41])	0.0552	0.9440	0.2056	0.0474	0.9607	0.5712	0.5469

Table 6: Effect of different 3D foundation models.

Effect of different 3D backbone ViT models. Tab. 6 reports the average results obtained using different 3D backbone ViT models as the global geometry encoder initialization. The results demonstrate the flexibility of our framework and show that it remains effective across different backbone choices.

Ablation study on the scene flow decoder. Tab. 7 validates our key design choices, reporting results averaged over the same four datasets used for flow estimation in Tab. 1. **(1) 2D Supervision:** Training solely on 3D datasets (“w/o 2D Supervision”) causes a severe performance collapse (EPE3D 0.6511), underscoring its necessity for guiding 3D flow. **(2) Target Lifting:** Removing lifted target samples (“w/o Target lifting”) prevents the network from dynamically querying the target space via 2D displacements. The resulting performance drop confirms that mapping 2D matches to 3D residuals is essential for spatial refinement. **(3) Iterative Updates:** Replacing the iterative refinement mechanism with a direct, single-step regression (“w/o iterations”) leads to a performance decline. This highlights that iteratively updating the flow field is crucial for achieving highly precise results. **(4) Auxiliary Priors:** Ablating the auxiliary priors (“w/o $\mathbf{C}_{3d,i,j}^{(t)}$ & $\mathcal{M}_{3d}^{(t)}$ ”) decreases accuracy, verifying their indispensability for global spatial alignment and kinematic smoothness across iterations. **(5) Hybrid**

Formulation (introduced in Sec. 3.2, Dense Scene Flow Recovery): Our full model (EPE3D 0.2056) significantly outperforms both heuristically lifting 2D flow with depth (“2D flow + d ”, 0.8210) and purely regressing 3D flow (“Pure 3D flow”, 0.2854). This proves that explicitly intertwining 2D flow with Z-axis displacement is superior to relying on isolated domains.

Setting	Abs Rel ↓	$\delta < 1.25 \uparrow$	EPE3D ↓
w/o 2D Supervision	0.1021	0.9199	0.6511
w/o Target lifting	0.0534	0.9543	0.3017
w/o iterations	0.0506	0.9559	0.2356
w/o $\mathbf{C}_{3d,i,j}^{(t)}$ & $\mathcal{M}_{3d}^{(t)}$	0.0482	0.9591	0.2201
2D flow + d	0.1304	0.8911	0.8210
Pure 3D flow	0.0552	0.9570	0.2815
Full (Ours)	0.0474	0.9607	0.2056

Table 7: Ablation study on the scene flow decoder.

Method	Time (s) ↓	Mem. (GB) ↓	Parm. (M) ↓
POMATO [78] (Dense)	4.8	16	133.64
ZeroMSF [40] (Dense)	8.2	10	153.84
STV2 [68] (Sparse)	5.8	19	65.99
STV2 [68] (Dense)	OOM	OOM	65.99
Ours w/o 2D-to-3D (Dense)	OOM	OOM	56.90
Ours (Dense)	3.4	14	26.06

Table 8: Efficiency comparison on 16-frame ADT [50] sequences. Gray rows indicate Out-of-Memory (OOM) failures under dense tracking settings.

Efficiency of scene flow decoder. Tab. 8 compares the inference time (Time), memory consumption (Mem.), and parameter count (Parm.) on 16-frame sequences from the ADT [50] dataset. First, pairwise scene flow approaches, such as POMATO [78] and ZeroMSF [40], natively support dense tracking by computing flow independently for every pixel. However, unlike our efficient sparse-to-dense formulation, their per-pixel prediction approach incurs significantly higher computational cost, resulting in increased latency without providing a clear advantage in memory efficiency. On the other hand, STV2 [68] relies on a traditional 3D spatial correlation mechanism for iterative trajectory estimation. Its high computational complexity and large parameter count impose a severe bottleneck. Consequently, STV2 is restricted to sparse tracking and suffers from Out-of-Memory (OOM) errors when extended to dense tracking scenarios. To explicitly validate that this bottleneck lies in the traditional 3D computation, we replace our proposed 2D-to-3D correlation module with the aforementioned traditional 3D mechanism (**Ours w/o 2D-to-3D**). As shown in the gray rows of Tab. 8, this variant similarly crashes with an OOM error under dense tracking settings. Overall, by bypassing the prohibitive cost of explicit 3D correlations, our method achieves superior efficiency in runtime, memory footprint, and model size, all while natively supporting dense tracking and maintaining state-of-the-art performance.

4.6 Qualitative Visualization

To demonstrate our method’s effectiveness, Fig. 4 visualizes the tracking and flow results across five aspects. The first four illustrate **camera-centric** perspectives: (1) **1st-frame 2D tracking**, where consistent colors denote temporal pixel correspondences. (2) **Scene flow**, contrasting the original RGB point cloud with the flow-transformed geometry (rainbow). (3) **1st-frame 3D tracking**, featuring uniformly downsampled camera-centric trajectories for rendering efficiency. (4) **All-frame 3D tracking**, capturing dense camera-centric trajectories of both existing and newly emerging objects. Distinct from the above, (5) **World-centric**

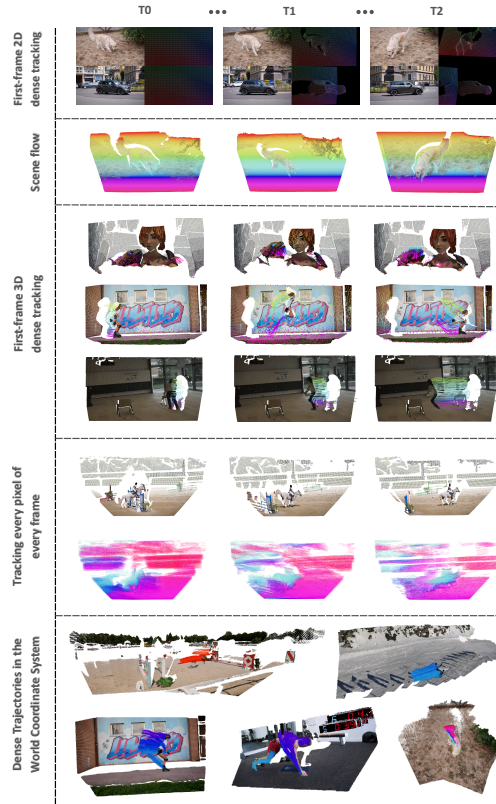


Fig. 4: Qualitative results on diverse in-the-wild videos.

tracking maps these trajectories to a global coordinate system. This effectively decouples camera ego-motion from object dynamics, yielding spatially stable backgrounds and physically coherent absolute motions for dynamic entities. For dynamic visualizations, please refer to the supplementary video.

5 Conclusion

In this paper, we present **Track4World**, a feedforward foundation model designed for efficient, dense 3D tracking within a world-centric coordinate framework. Leveraging a global 3D scene representation parameterized by a VGGT-style Vision Transformer, our method introduces a novel 3D correlation mechanism to simultaneously estimate dense 2D and 3D scene flow between arbitrary pairs of frames. By tightly coupling this estimated scene flow with the reconstructed 3D geometry, Track4World facilitates highly efficient, pixel-wise 3D trajectory extraction across the entire video. Extensive evaluations across multiple benchmarks demonstrate the exceptional robustness, scalability, and generalization capabilities of our approach, marking a significant step toward holistic 4D reconstruction from in-the-wild monocular videos.

Acknowledgement. Yuan Liu was supported by the HKUST ‘30 for 30’ Talent Acquisition Campaign, with funding provided by Kerry Group Limited.

References

1. Alumootil, V., Vu, T.A., Jawed, M.K.: Dept3r: Joint dense point tracking and 3d reconstruction of dynamic scenes in a single forward pass. arXiv preprint arXiv:2512.13122 (2025)
2. Balasingam, A., Chandler, J., Li, C., Zhang, Z., Balakrishnan, H.: Drivetrack: A benchmark for long-range point tracking in real-world videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22488–22497 (2024)
3. Burgert, R., Xu, Y., Xian, W., Pilarski, O., Clausen, P., He, M., Ma, L., Deng, Y., Li, L., Mousavi, M., Ryoo, M., Debevec, P., Yu, N.: Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In: CVPR (2025)
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part VI 12. pp. 611–625. Springer (2012)
5. Cao, Y., Lu, J., Huang, Z., Shen, Z., Zhao, C., Hong, F., Chen, Z., Li, X., Wang, W., Liu, Y., et al.: Reconstructing 4d spatial intelligence: A survey. arXiv preprint arXiv:2507.21045 (2025)
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
7. Chen, W., Chen, L., Wang, R., Pollefeys, M.: Leap-vo: Long-term effective any point tracking for visual odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19844–19853 (2024)
8. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Easi3r: Estimating disentangled motion from dust3r without training. arXiv preprint arXiv:2503.24391 (2025)
9. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Ttt3r: 3d reconstruction as test-time training. arXiv preprint arXiv:2509.26645 (2025)
10. Cho, S., Huang, J., Nam, J., An, H., Kim, S., Lee, J.Y.: Local all-pair correspondence for point tracking. In: European Conference on Computer Vision. pp. 306–325. Springer (2024)
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
12. Doersch, C., Gupta, A., Markeeva, L., Recasens, A., Smaira, L., Aytar, Y., Carreira, J., Zisserman, A., Yang, Y.: Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems* **35**, 13610–13626 (2022)
13. Doersch, C., Luc, P., Yang, Y., Gokay, D., Koppula, S., Gupta, A., Heyward, J., Rocco, I., Goroshin, R., Carreira, J., et al.: Bootstap: Bootstrapped training for tracking-any-point. In: Proceedings of the Asian Conference on Computer Vision. pp. 3257–3274 (2024)
14. Doersch, C., Yang, Y., Vecerik, M., Gokay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: 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)

15. Feng, H., Zhang, J., Wang, Q., Ye, Y., Yu, P., Black, M.J., Darrell, T., Kanazawa, A.: St4rtrack: Simultaneous 4d reconstruction and tracking in the world. arXiv preprint arXiv:2504.13152 (2025)
16. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. *The international journal of robotics research* **32**(11), 1231–1237 (2013)
17. Georgakis, G., Reza, M.A., Mousavian, A., Le, P.H., Košecká, J.: Multiview rgb-d dataset for object instance detection. In: 2016 Fourth international conference on 3D vision (3DV). pp. 426–434. IEEE (2016)
18. Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D.J., Gnanapragasam, D., Golemo, F., Herrmann, C., et al.: Kubric: A scalable dataset generator. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3749–3761 (2022)
19. Gu, X., Wang, Y., Wu, C., Lee, Y.J., Wang, P.: Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3254–3263 (2019)
20. Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., et al.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. arXiv preprint arXiv:2501.03847 (2025)
21. Harley, A.W., You, Y., Sun, X., Zheng, Y., Raghuraman, N., Gu, Y., Liang, S., Chu, W.H., Dave, A., You, S., et al.: Alltracker: Efficient dense point tracking at high resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5253–5262 (2025)
22. Hu, W., Gao, X., Li, X., Zhao, S., Cun, X., Zhang, Y., Quan, L., Shan, Y.: Depthcrafter: Generating consistent long depth sequences for open-world videos. arXiv preprint arXiv:2409.02095 (2024)
23. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., et al.: Vipe: Video pose engine for 3d geometric perception. arXiv preprint arXiv:2508.10934 (2025)
24. Huang, W., Chao, Y.W., Mousavian, A., Liu, M.Y., Fox, D., Mo, K., Fei-Fei, L.: Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. arXiv preprint arXiv:2601.03782 (2026)
25. Jiang, Z., Zheng, C., Laina, I., Larlus, D., Vedaldi, A.: Geo4d: Leveraging video generators for geometric 4d scene reconstruction. arXiv preprint arXiv:2504.07961 (2025)
26. Jin, L., Tucker, R., Li, Z., Fouhey, D., Snavely, N., Holynski, A.: Stereo4d: Learning how things move in 3d from internet stereo videos. CVPR (2025)
27. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: Proceedings of the IEEE international conference on computer vision. pp. 3334–3342 (2015)
28. Karaev, N., Makarov, I., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. arXiv preprint arXiv:2410.11831 (2024)
29. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. In: European Conference on Computer Vision. pp. 18–35. Springer (2024)
30. Karhade, J., Keetha, N., Zhang, Y., Gupta, T., Sharma, A., Scherer, S., Ramanan, D.: Any4d: Unified feed-forward metric 4d reconstruction. arXiv preprint arXiv:2512.10935 (2025)

31. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414* (2025)
32. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 1611–1621 (2021)
33. Koppula, S., Rocco, I., Yang, Y., Heyward, J., Carreira, J., Zisserman, A., Brostow, G., Doersch, C.: Tapvid-3d: A benchmark for tracking any point in 3d. *Advances in Neural Information Processing Systems* **37**, 82149–82165 (2024)
34. Lee, A.X., Devin, C.M., Zhou, Y., Lampe, T., Bousmalis, K., Springenberg, J.T., Byravan, A., Abdolmaleki, A., Gileadi, N., Khosid, D., et al.: Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In: *5th Annual Conference on Robot Learning* (2021)
35. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: *European Conference on Computer Vision*. pp. 71–91. Springer (2024)
36. Li, H., Zhang, H., Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, L.: Taptr: Tracking any point with transformers as detection. In: *European Conference on Computer Vision*. pp. 57–75. Springer (2024)
37. Li, M., Li, P., Zhang, Z., Lu, J., Zhao, C., Xue, W., Liu, Q., Peng, S., Zhang, W., Luo, W., et al.: Unish: Unifying scene and human reconstruction in a feed-forward pass. *arXiv preprint arXiv:2601.01222* (2026)
38. Li, Y., Shen, Y., Huang, Z., Chen, S., Bian, W., Shi, X., Wang, F.Y., Sun, K., Bao, H., Cui, Z., et al.: Blinkvision: A benchmark for optical flow, scene flow and point tracking estimation using rgb frames and events. In: *European conference on computer vision*. pp. 19–36. Springer (2024)
39. Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., Snavely, N.: Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. *arXiv preprint arXiv:2412.04463* (2024)
40. Liang, Y., Badki, A., Su, H., Tompkin, J., Gallo, O.: Zero-shot monocular scene flow estimation in the wild. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 21031–21044 (2025)
41. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647* (2025)
42. Liu, X., Qi, C.R., Guibas, L.J.: Flownet3d: Learning scene flow in 3d point clouds. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 529–537 (2019)
43. Liu, X., Xiao, Y., Chen, D.Y., Feng, J., Tai, Y.W., Tang, C.K., Kang, B.: Trace anything: Representing any video in 4d via trajectory fields. *arXiv preprint arXiv:2510.13802* (2025)
44. Lu, J., Huang, T., Li, P., Dou, Z., Lin, C., Cui, Z., Dong, Z., Yeung, S.K., Wang, W., Liu, Y.: Align3r: Aligned monocular depth estimation for dynamic videos. *arXiv preprint arXiv:2412.03079* (2024)
45. Lu, J., Xiong, W., Deng, J., Li, P., Huang, T., Dou, Z., Lin, C., Yeung, S.K., Liu, Y.: Trackingworld: World-centric monocular 3d tracking of almost all pixels. *arXiv preprint arXiv:2512.08358* (2025)
46. Luo, Y., Zhou, S., Lan, Y., Pan, X., Loy, C.C.: 4rc: 4d reconstruction via conditional querying anytime and anywhere. *arXiv preprint arXiv:2602.10094* (2026)
47. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene

- flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016)
48. Ngo, T.D., Zhuang, P., Gan, C., Kalogerakis, E., Tulyakov, S., Lee, H.Y., Wang, C.: Delta: Dense efficient long-range 3d tracking for any video. arXiv preprint arXiv:2410.24211 (2024)
 49. Palazzolo, E., Behley, J., Lottes, P., Giguere, P., Stachniss, C.: Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7855–7862. IEEE (2019)
 50. Pan, X., Charron, N., Yang, Y., Peters, S., Whelan, T., Kong, C., Parkhi, O., Newcombe, R., Ren, Y.C.: Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20133–20143 (2023)
 51. Piccinelli, L., Yang, Y.H., Sakaridis, C., Segu, M., Li, S., Van Gool, L., Yu, F.: Unidepth: Universal monocular metric depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10106–10116 (2024)
 52. Qu, J., Li, H., Liu, S., Ren, T., Zeng, Z., Zhang, L.: Taptrv3: Spatial and temporal context foster robust tracking of any point in long video. arXiv preprint arXiv:2411.18671 (2024)
 53. Ren, Y., Zhao, C., He, Y., Cong, P., Liang, H., Yu, J., Xu, L., Ma, Y.: Lidar-aid inertial poser: Large-scale human motion capture by sparse inertial and lidar sensors. IEEE Transactions on Visualization and Computer Graphics **29**(5), 2337–2347 (2023)
 54. Schmid, J., Jahedi, A., Senn, N.B., Bruhn, A.: Ms-raft-3d: A multi-scale architecture for recurrent image-based scene flow. In: 2025 IEEE International Conference on Image Processing (ICIP). pp. 1570–1575. IEEE (2025)
 55. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. pp. 573–580. IEEE (2012)
 56. Sucar, E., Insafutdinov, E., Lai, Z., Vedaldi, A.: V-dpm: 4d video reconstruction with dynamic point maps. arXiv preprint arXiv:2601.09499 (2026)
 57. Team, A., Zhu, H., Wang, Y., Zhou, J., Chang, W., Zhou, Y., Li, Z., Chen, J., Shen, C., Pang, J., et al.: Aether: Geometric-aware unified world modeling. arXiv preprint arXiv:2503.18945 (2025)
 58. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 402–419. Springer (2020)
 59. Teed, Z., Deng, J.: Raft-3d: Scene flow using rigid-motion embeddings. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8375–8384 (2021)
 60. Vecerik, M., Doersch, C., Yang, Y., Davchev, T., Aytar, Y., Zhou, G., Hadsell, R., Agapito, L., Scholz, J.: Robotap: Tracking arbitrary points for few-shot visual imitation. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 5397–5403. IEEE (2024)
 61. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupperecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. arXiv preprint arXiv:2503.11651 (2025)
 62. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. arXiv preprint arXiv:2501.12387 (2025)

63. Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 5261–5271 (2025)
64. Wang, R., Xu, S., Dong, Y., Deng, Y., Xiang, J., Lv, Z., Sun, G., Tong, X., Yang, J.: Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546* (2025)
65. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 20697–20709 (2024)
66. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347* (2025)
67. Wang, Y., Lipson, L., Deng, J.: Sea-raft: Simple, efficient, accurate raft for optical flow. In: *European Conference on Computer Vision*. pp. 36–54. Springer (2024)
68. Xiao, Y., Wang, J., Xue, N., Karaev, N., Makarov, Y., Kang, B., Zhu, X., Bao, H., Shen, Y., Zhou, X.: Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 6726–6737 (2025)
69. Xiao, Y., Wang, Q., Zhang, S., Xue, N., Peng, S., Shen, Y., Zhou, X.: 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)
70. Xiong, W., Yuan, Z., Lu, J., Zhao, C., Li, P., Liu, Y.: Human3r: Incorporating human priors for better 3d dynamic reconstruction from monocular videos. *arXiv preprint arXiv:2512.06368* (2025)
71. Xu, T.X., Gao, X., Hu, W., Li, X., Zhang, S.H., Shan, Y.: Geometrycrafter: Consistent geometry estimation for open-world videos with diffusion priors. *arXiv preprint arXiv:2504.01016* (2025)
72. Yang, G., Ramanan, D.: Upgrading optical flow to 3d scene flow through optical expansion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 1334–1343 (2020)
73. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 21924–21935 (2025)
74. Yao, D.Y., Zhai, A.J., Wang, S.: Uni4d: Unifying visual foundation models for 4d modeling from a single video. *arXiv preprint arXiv:2503.21761* (2025)
75. Zhang, B., Ke, L., Harley, A.W., Fragkiadaki, K.: Tapip3d: Tracking any point in persistent 3d geometry. *arXiv preprint arXiv:2504.14717* (2025)
76. Zhang, C., Moing, G.L., Koppula, S., Rocco, I., Momeni, L., Xie, J., Sun, S., Sukthankar, R., Barral, J.K., Hadsell, R., et al.: Efficiently reconstructing dynamic scenes one d4rt at a time. *arXiv preprint arXiv:2512.08924* (2025)
77. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825* (2024)
78. Zhang, S., Ge, Y., Tian, J., Xu, G., Chen, H., Lv, C., Shen, C.: Pomato: Marrying pointmap matching with temporal motion for dynamic 3d reconstruction. *arXiv preprint arXiv:2504.05692* (2025)
79. Zhang, Z., Cole, F., Li, Z., Rubinstein, M., Snavely, N., Freeman, W.T.: Structure and motion from casual videos. In: *European Conference on Computer Vision*. pp. 20–37. Springer (2022)

80. Zhao, C., Zhang, J., Du, J., Shan, Z., Wang, J., Yu, J., Wang, J., Xu, L.: I'm hoi: Inertia-aware monocular capture of 3d human-object interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 729–741 (2024)
81. Zhao, S., Zhao, L., Zhang, Z., Zhou, E., Metaxas, D.: Global matching with overlapping attention for optical flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17592–17601 (2022)
82. Zhao, W., Liu, S., Guo, H., Wang, W., Liu, Y.J.: Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In: European Conference on Computer Vision. pp. 523–542. Springer (2022)
83. Zheng, Y., Harley, A.W., Shen, B., Wetzstein, G., Guibas, L.J.: Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19855–19865 (2023)
84. Zhu, R., Liang, Y., Chang, H., Deng, J., Lu, J., Yang, W., Zhang, T., Zhang, Y.: Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. arXiv preprint arXiv:2410.07707 (2024)
85. Zhu, R., Lu, J., Hu, W., Han, X., Cai, J., Shan, Y., Zheng, C.: Motioncrafter: Dense geometry and motion reconstruction with a 4d vae. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 40303–40315 (2026)
86. Zhuo, D., Zheng, W., Guo, J., Wu, Y., Zhou, J., Lu, J.: Streaming 4d visual geometry transformer. arXiv preprint arXiv:2507.11539 (2025)