SpatialTrackerV2: 3D Point Tracking Made Easy

Yuxi Xiao[1]* Jianyuan Wang[2] Nan Xue[3] Nikita Karaev[2,4] Yuri Makarov[4] Bingyi Kang[5]
Xing Zhu[3] Hujun Bao[1] Yujun Shen[3] Xiaowei Zhou[1]†
[1]Zhejiang University [2]Oxford [3]Ant Group [4]Pixelwise AI [5]Bytedance Seed

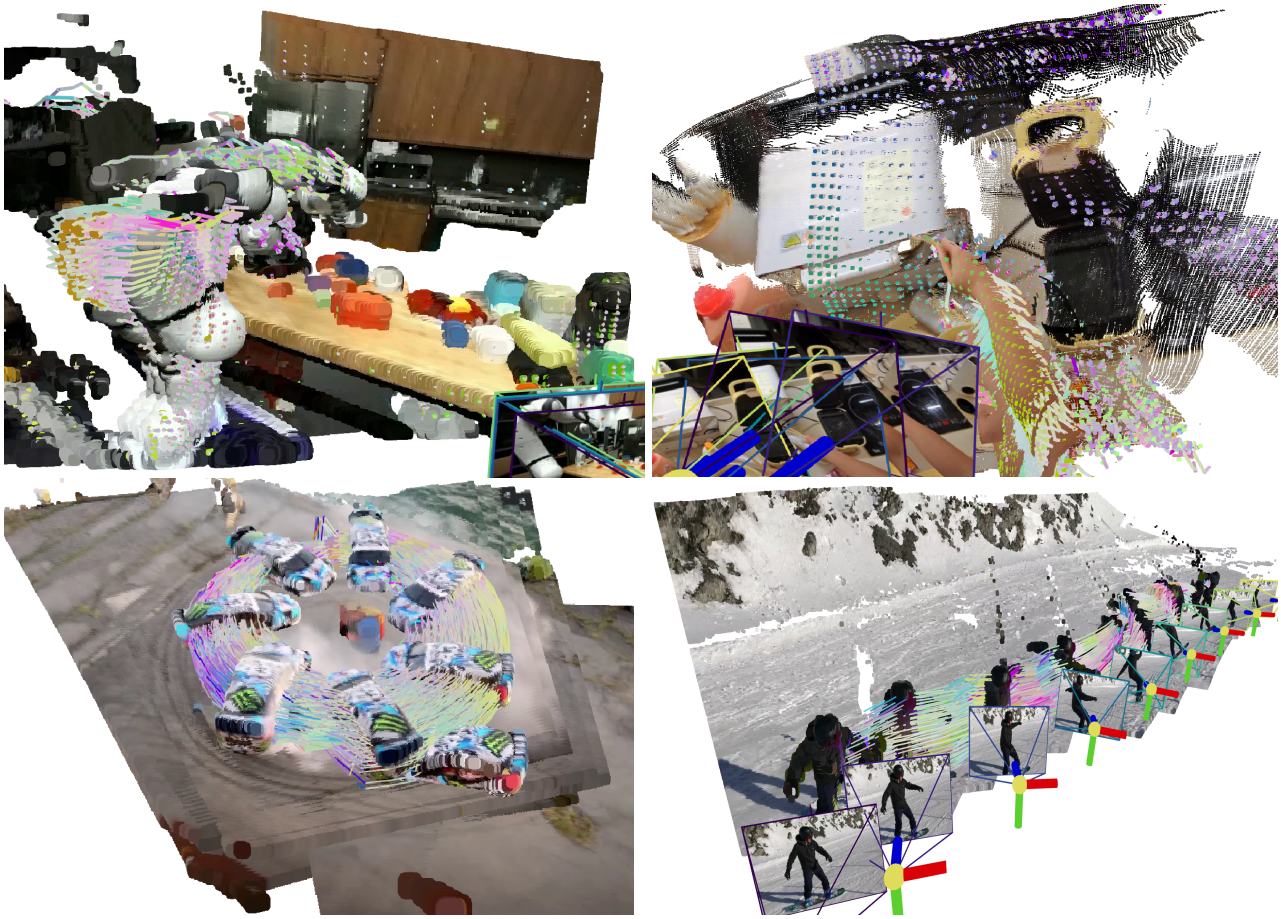


Figure 1. **SpatialTrackerV2** produces consistent 3D scene geometry, camera poses, and 3D point trajectories all at once from monocular videos of arbitrary scenarios, e.g., robotic manipulation, first-person egocentric views, and dynamic sports (drifting and skating) shown in this figure. Try our online demo at https://huggingface.co/spaces/Yuxihenry/SpatialTrackerV2.

Abstract

We present SpatialTrackerV2, a feed-forward 3D point tracking method for monocular videos. Going beyond modular pipelines built on off-the-shelf components for 3D tracking, our approach unifies the intrinsic connections between point tracking, monocular depth, and camera pose estimation into a high-performing and feedforward 3D point tracker. It decomposes world-space 3D motion into scene geometry, camera ego-motion, and pixel-wise object motion, with a fully differentiable and end-to-end architecture, allowing scalable training across a wide range of datasets, including synthetic sequences, posed RGB-D videos, and unlabeled in-the-wild footage. By learning geometry and motion jointly from such heterogeneous data, SpatialTrackerV2 outperforms existing 3D tracking methods by **30%**, *and matches the accuracy of leading dynamic 3D reconstruction approaches while running 50× faster.*

* Partially completed during Ant internship. † Corresponding author.

1. Introduction

3D point tracking aims to recover long-term 3D trajectories of arbitrary points from monocular videos. As a universal dynamic scene representation, it has recently shown strong potential in diverse applications, including robotics [48, 56, 67], video generation [22, 29, 57, 81], and 3D/4D reconstruction [10, 34, 38, 76]. Compared to parametric motion models (e.g., SMPL [47], MANO [63], skeletons, or 3D bounding boxes), it offers greater flexibility and generalization over various real-world scenes, as shown in Fig. 1.

Existing solutions [1, 4, 54, 75, 86] of 3D point tracking extensively explored the well-developed low/mid-level vision models, such as optical flow [45, 68] and monocular depth estimation [27, 87], and took benefits from 2D point tracking models [15, 23, 33]. Among them, optimization-based methods [42, 75, 95] distill the optical flow, monocular depth models and camera motions for each given monocular video with promising results obtained, while being computationally expensive due to their per-scene optimization designs. SpatialTracker [86] moved forward to efficient 3D point tracking with a feed-forward model, and the more recent works [1, 4, 54] explored different architecture designs and rendering constraints to achieve higher-quality 3D tracking. Nevertheless, the feed-forward solutions are limited to training data scalability issues due to the need for ground-truth 3D tracks as supervision, which downgrades the tracking quality in real-world casual captures. Moreover, overlooking the inherent interplay between camera motion, object motion, and scene geometry results in error entanglement and accumulation across modules.

These limitations motivate our core insights: (1) The reliance on ground-truth 3D trajectories constrains the scalability of existing feed-forward models, highlighting the need for designs that can generalize across diverse and weakly-supervised data sources. (2) The absence of joint reasoning over scene geometry, camera motion, and object motion leads to compounded errors and degraded performance, underscoring the importance of disentangling and explicitly modeling these motion components. To address these challenges, we decompose 3D point tracking into three distinct components: video depth, ego (camera) motion, and object motion, and integrate them within a fully differentiable pipeline that supports scalable joint training across heterogeneous data.

In our SpatialTrackerV2, a front-end and back-end architecture is proposed. The front end is a video depth estimator and camera pose initializer, adapted from typical monocular depth prediction frameworks [88] with attention-based temporal information encoding [74]. The predicted video depths and camera poses are then fused through a scale-shift estimation module, which ensures consistency between the depth and motion predictions. The back end consists of a proposed Joint Motion Optimization Module, which takes the video depth and coarse camera trajectories as input and iteratively estimates 2D and 3D trajectories, along with trajectory-wise dynamics and visibility scores. This enables an efficient bundle adjustment process for optimizing camera poses in the loop. At its core lies a novel SyncFormer, which separately models the 2D and 3D correlations in two branches, connected by multiple cross-attention layers. This design mitigates mutual interference between 2D and 3D embeddings and allows the model to update representations in two distinct spaces, namely the image (UV) space and the camera coordinate space. Furthermore, benefiting from this dual-branch design, bundle adjustment can be effectively applied to jointly optimize camera poses as well as the 2D and 3D trajectories.

This unified and differentiable pipeline makes large-scale training on diverse datasets possible. For RGB-D datasets with camera poses, we jointly train 3D tracking using consistency constraints from ground-truth depth and camera poses for static points, while dynamic points seamlessly contribute to the optimization. For video datasets that provide only camera pose annotations and lack depth information, we leverage consistency among camera poses, and 2D and 3D point tracking to drive the model's optimization. Relying on this framework, we successfully scale up training of the entire pipeline across 17 datasets.

Evaluations on the TAPVid-3D benchmark [37] show that our method sets a new state-of-the-art in 3D point tracking, achieving 21.2 AJ and 31.0 APD_{3D}, surpassing DELTA [54] with relative improvements of 61.8% and 50.5%, respectively. Additionally, extensive experiments on dynamic reconstruction show our superior results on consistent video depth and camera poses estimation. Specifically, SpatialTrackerV2 beats the best dynamic reconstruction method, MegaSAM [42], on most video depth datasets and achieves comparable results on various camera pose benchmarks, while its inference speed is 50 $\times$ faster.

2. Related work

This section covers relevant literature on 3D tracking, depth, and camera pose estimation.

2.1. Point tracking

PIPs [23] revisited the 2D point tracking task first introduced in [64] and proposed a deep learning approach to solve it. TAP-Vid [14] redefined the problem and introduced both a benchmark and a simple architecture, TAP-Net. Subsequently, the performance was improved in TAPIR [15] by combining the global matching capabilities of TAP-Net with the local refinement offered by PIPs. CoTracker [33] pioneered tracking through occlusions using a transformer architecture combined with joint attention, followed by TAPTR [40] and LocoTrack [11], which im-

proved efficiency and introduced 4D correlation volumes. Recently, BootsTAPIR [16] and CoTracker3 [32] explored the use of unlabeled data to achieve better performance.

While 2D point tracking has been extensively studied, 3D point tracking remains a relatively new field. The first method to demonstrate 3D point tracking capabilities was the test-time optimization-based OmniMotion [75]. Later, SpatialTracker [86] introduced the first feed-forward 3D point tracker by combining a 2D point tracker [33] with depth priors from a monocular depth estimator [3]. SceneTracker [71] proposed a new architecture for 3D tracking with depth priors, while DELTA [54] improved efficiency and achieved dense 3D tracking. Recently, TAPIP3D [90] improved 3D tracking robustness by lifting image features into the world coordinate space and performing tracking there. All of these 3D tracking models were trained on small synthetic datasets.

Unlike these methods, we present a scalable 3D tracking framework trained on a collection of both real and synthetic datasets, while also explicitly modeling camera motion to improve performance on egocentric videos.

2.2. Depth estimation

Early methods such as Eigen et al. [17] introduced single-view depth estimation using CNNs, but were limited by dataset scale and poor generalization [18, 89]. MiDaS [5] improved this by mixing datasets for broader coverage, and ZoeDepth [3] adapted it for metric depth, though ambiguity from missing camera intrinsics remained. Later, Metric3D [26] and UniDepth [60] addressed this by jointly estimating intrinsics and normalized depth.

With large-scale pretraining (e.g., diffusion [13, 41], DINO [8]), recent models like Marigold [70] and DepthAnything [87, 88] have significantly advanced zero-shot depth. Extensions to video [9, 27] have also emerged.

In this work, we build on DepthAnythingV2 and extend it to video, aiming for not just consistent depth, but a unified framework that also predicts camera poses and tracks, with proper scale alignment—posing new challenges beyond static depth estimation.

2.3. Camera estimation

Traditional camera pose estimation methods [24, 58] rely on image-to-image point correspondences using keypoint detectors (e.g., SIFT [49, 50], SURF [2]) and matching techniques such as nearest neighbors, followed by geometric algorithms like the five-point and eight-point methods [24, 25, 39, 55, 85]. Bundle Adjustment [69] is also commonly employed to further enhance accuracy. Recently, direct regression approaches using neural networks [35, 43, 51, 84, 92] have emerged, aiming to overcome limitations in sparse-view scenarios or when correspondences are unreliable. Diffusion models, such as PoseDiffusion [72] and RayDiffusion [93], have also been explored, offering strong accuracy but suffering from high inference costs. In contrast, the camera head of VGGSfM [73] or VGGT [74] adopts an iterative refinement paradigm similar to RAFT [68], striking a balance between accuracy and inference cost, and enabling estimation of both extrinsic and intrinsic parameters.

3. Method

Given a video with T frames $(\mathcal{I}^t)_{t=1}^T$ and N query points $\mathcal{Q}_i = (x_i, y_i) \in \mathbb{R}^2, i = 1, \ldots, N$, the goal of 3D tracking is to recover pixel-wise 3D trajectories $\mathcal{T} = (\mathcal{T}_i^t)_{i=1,\ldots,N}^{t=1,\ldots,T}$ for each query. In order to account for static and dynamic parts of the scene, we decompose $\mathcal{T}$ into the ego camera motion $\mathcal{T}_{\text{ego}}$ and object motion $\mathcal{T}_{\text{object}}$ as shown in Fig. 2.

3.1. Ego Motion Component

Ego motion, represented by camera trajectories $\mathcal{T}_{\text{ego}}$, is a major contributor to the 3D flow in camera coordinates. To compute the 3D tracks induced by ego motion, we need to estimate scale-aligned camera trajectories and video depth.

Video Depth. Monocular depth estimation models, such as DepthAnything [87], typically follow an encoder-decoder design with a vision encoder like DINO [8] and a DPT-style decoder. Following VGGT [74], we extend the monocular encoder into a temporal encoder using an alternating-attention mechanism. This mechanism alternates between intra-frame self-attention and inter-frame attention over flattened video tokens, effectively balancing performance and efficiency. Furthermore, two learnable tokens, $\mathbf{P}_{\text{tok}}$ and $\mathbf{S}_{\text{tok}}$, are incorporated into the alternating-attention layers to capture high-level semantics for pose and scale regression.

Camera Tracker. Similar to [73, 74], we adopt a differentiable pose head, $\mathcal{H}$, to decode pose, scale and shift directly:

$$\mathcal{P}^t, a, b = \mathcal{H}(\mathbf{P}_{\text{tok}}, \mathbf{S}_{\text{tok}}), \tag{1}$$

where $\mathcal{P}^t \in \mathbb{R}^{T\times 8}$ is the camera encoding parameterized by a quaternion, a translation vector, and the normalized focal length concatenated together; The parameters a and b represent the scale and shift used to align the depth with the estimated camera poses. After that, we can easily obtain the 3D trajectories induced by ego-motion $\mathcal{T}_{\text{ego}}$:

$$\mathcal{T}_{\text{ego}} = \mathcal{W}(\mathcal{P}, a * \mathcal{D}_{\text{norm}} + b), \tag{2}$$

where $\mathcal{D}_{\text{norm}}$ is the raw results after DPT head with activation functions and $\mathcal{W}$ is the camera transformation.

3.2. Joint Motion Optimization.

After ego-motion initialization, we jointly estimate 2D trajectories $\mathcal{T}^{\text{2d}} \in \mathbb{R}^{T\times N\times 2}$ in UV space and their corresponding 3D trajectories $\mathcal{T}^{\text{3d}} \in \mathbb{R}^{T\times N\times 3}$ in the camera coordinate system using an iterative transformer module, referred

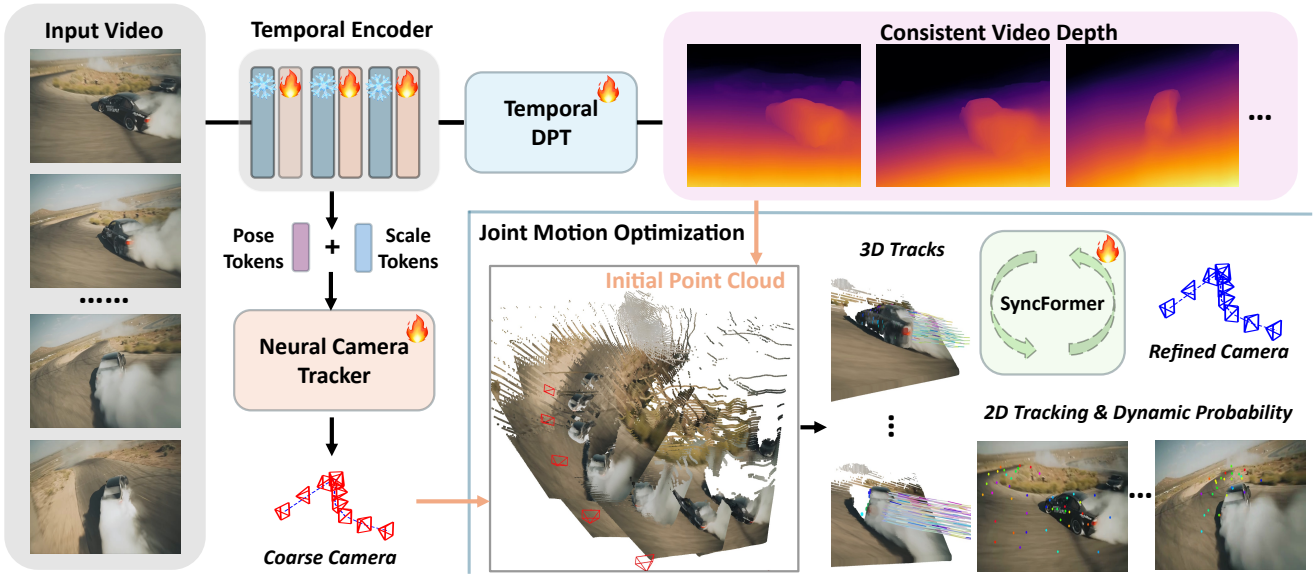


Figure 2. **Pipeline Overview.** Our method adopts a front-end and back-end architecture. The front-end estimates scale-aligned depth and camera poses from the input video, which are used to construct initial static 3D tracks. The back-end then iteratively refines both tracks and poses via joint motion optimization.

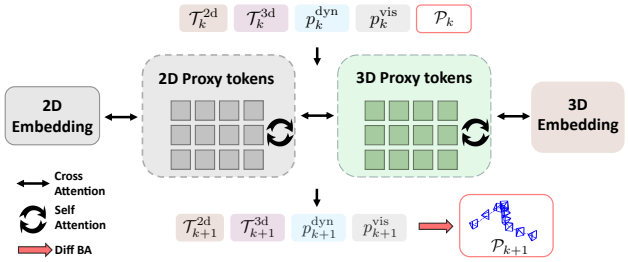


Figure 3. **SyncFormer.** The model takes previous estimates and their corresponding embeddings as input, and updates them iteratively. The 2D and 3D embeddings are processed in separate branches that interact via cross-attention.

to as SyncFormer. In parallel, the SyncFormer also dynamically estimates the visibility probability p^{vis} and dynamic probability p^{dyn} for each trajectory, enabling an efficient bundle adjustment process to refine the camera poses $\mathcal{P}$.

SyncFormer. As illustrated in Fig. 3, in every iteration, SyncFormer takes 2D embeddings, 3D embeddings, and camera poses as input, and updates the 2D trajectories $\mathcal{T}^{\text{2d}}$, 3D trajectories $\mathcal{T}^{\text{3d}}$, dynamic probabilities p^{dyn}, and visibility scores p^{vis}:

$$\mathcal{T}^{\text{2d}}_{k+1}, \mathcal{T}^{\text{3d}}_{k+1}, p^{\text{dyn}}_{k+1}, p^{\text{vis}}_{k+1} = f_{\text{sync}}(\mathcal{T}^{\text{2d}}_{k}, \mathcal{T}^{\text{3d}}_{k}, p^{\text{dyn}}_{k}, p^{\text{vis}}_{k}, \mathcal{P}_k), \quad (3)$$

where f_{sync} denotes the transformer-based update function. To better capture the distinct characteristics of 2D and 3D motion, the 2D and 3D trajectory updaters are modeled using separate attention layers. To reduce computational cost, correlation embeddings are first encoded into a compact set of 2D and 3D proxy tokens using cross-attention. Then, information is exchanged between the 2D and 3D branches via a cross-attention layer between the respective proxy tokens. This design decouples the mutual influence between 2D and 3D tracking, which are updated in two different domains: the UV space for 2D trajectories and the camera coordinate space for 3D trajectories.

2D and 3D Embeddings. As the input for SyncFormer, the 2D and 3D embeddings encode the status of current estimations while recording the neighbourhood information for updating. For 2D embeddings, we keep the same to Cotracker3 [33]. The 3D embeddings $\mathbf{E}^{\text{3d}} = (Corr_{\text{3D}}, \mathbf{e}^{\text{time}}, \mathbf{e}^{\text{Gpos}}, p^{\text{dyn}}, p^{\text{vis}})$ contains the 3D correlation features $Corr_{\text{3D}}$, the time embeddings $\mathbf{e}^{\text{time}}$, global position embeddings $\mathbf{e}^{\text{Gpos}}$ and dynamic-visibility scores. The 3D correlation $Corr_{\text{3D}}$ is the main features for the 3D embedding. Different to 2D correlations, we expect 3D tracking branch updates the 3D position in the camera coordinate space. Therefore, we calculate the 3D correlations on the normalized point maps from front-end instead of depth map. Similar to 2D correlations, we construct multi-resolution point maps and compute the relative translations between each point and its neighbors within radius of 3. We then apply harmonic positional encoding to project these relative translations into high-dimensional feature representations, which are combined with semantic features to compute the final 3D correlations:

$$Corr_{\text{3D}} = [K(\frac{\mathbf{x}}{ks} + \delta, \frac{\mathbf{y}}{ks} + \delta) : \delta \in \mathbb{Z}, \|\delta\|_\infty \leq \Delta], \quad (4)$$

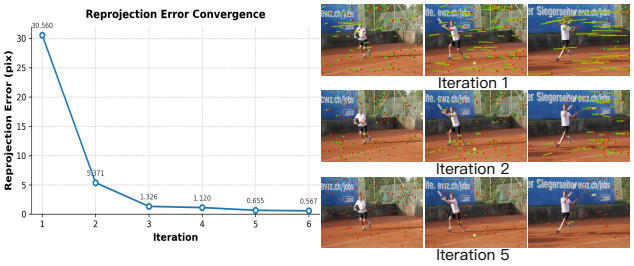


Figure 4. **Iterative Process of SyncFormer.** The left subfigure shows the convergence curve of reprojection error, illustrating rapid reprojection error reduction. The right subfigures visualizes the progressive alignment between 2D tracking results (in green) and projections of 3D (world) tracking transformed by camera poses (in red).

with Δ set to 3 and at multiple scales $s = 1, 2, 3, 4$, K denotes the operator for encoding relative translation features. Besides, in order to encode the camera pose to assist the tracking estimations, we propose $\mathbf{e}^{\text{Gpos}}$ which transforms the query points $\mathcal{Q}_i$ into each frames with the current camera poses, denoted as anchor points $\mathcal{Q}_i^{\text{anc}}$. $\mathbf{e}^{\text{Gpos}}$ is calculated from the relative translation between the current position and $\mathbf{e}^{\text{Gpos}}$, and they are then projected into high dimension with same tricks as above.

Camera Motion Optimization. After each iteration, the updates of $\mathcal{T}^{\text{2d}}_{k+1}, \mathcal{T}^{\text{3d}}_{k+1}, p^{\text{dyn}}_{k+1}, p^{\text{vis}}_{k+1}$ naturally form self-consistency constraints with the camera poses. These intrinsic constraints can be seamlessly incorporated into a bundle adjustment formulation. Specifically, we apply a weighted Procrustes analysis to register $\mathcal{T}^{\text{3d}}_{k+1}$ into the world coordinate frame, where the alignment weights are given by the dynamic scores. This coarse alignment process is differentiable and provides effective supervision for learning the dynamic probability. Then we fuse the aligned $\mathcal{T}^{\text{3d}}_{k+1}$ to world points $\mathbf{P}^{\text{world}}_{k+1} \in \mathbb{R}^{N\times 3}$, where the dynamic points are filtered with the estimated dynamic scores. After this, we apply a direct bundle adjustment to optimize the camera poses with paired $\mathbf{P}^{\text{world}}_{k+1}, \mathcal{T}^{\text{2d}}_{k+1}$ and p^{vis}_{k+1}. In the next iteration, the updated camera poses $\mathcal{P}_{k+1}$ influence the subsequent loop through global position embedding. It is worth noting that the 2D and 3D trajectories are not updated via bundle adjustment; instead, the entire system is primarily driven by SyncFormer. As shown in Fig. 4, the reprojection errors decrease rapidly over iterations, while the 2D trajectories gradually become consistent with the 3D projections transformed by camera motion.

3.3. Training

Training Datasets. Our model is trained on a large collection of 17 datasets, each providing different forms of supervision. The training data can be broadly categorized into three types: (1) Posed RGB-D with tracking annotations, (2) Posed RGB-D, (3) Pose-only or unlabeled data. For (1) containing Kubric [21], PointOdyssey [96] and Dynamic Replica [31], we add the full supervisions of camera poses, video depth, dynamic segmentation and 2D, 3D tracking. Within category (2), including VKITTI [7], TartanAir [80], Spring [53], DL3DV [44], BEDLAM [6], MVS-Synth [28], CO3Dv2 [61], COP3D [65], WildRGBD [83], ScanNet++ [12] and OmniObject3D [82], we leverage depth supervision to improve the video depth estimator. In addition, joint training losses are applied to ensure that the 3D tracking remains consistent with the estimated depth. As for category (3), it includes HOI4D [46], Ego4D [20], and Stereo4D [30]. For this type of data, we apply camera loss and joint training losses here, and follow [9] by using a monocular depth model [78] as a teacher to preserve relative depth accuracy. It is worth noting that Stereo4D [30] provides only very sparse depth on valid 3D tracks. Therefore, we choose not to use its annotations and instead treat it as pose-only data, given that it is sourced from Internet videos with rich scene diversity.

Implementation details. Our training recipe consists of several stages. **Stage 1.** We first train the front end model to jointly estimate video depth and camera poses on category (1) and (2) datasets which sums up to 14 datasets. Training is conducted with mixed BF16 precision, while the DPT and camera tracker modules use full precision to ensure stable optimization. We use the AdamW optimizer with a learning rate of 5×10^{-5} and apply gradient clipping with a threshold of 0.1. Stage 1 training is performed for 200k iterations on 64 H20 GPUs. We shuffle the video length from 1-24 during the training. **Stage 2.** At this stage, we initialize SyncFormer using the category (1) datasets, where ground-truth depth is provided and camera poses are initialized as identity matrices. We load the Cotracker3 [32] checkpoint to initialize the 2D tracking branch. This stage training takes 3 days of 100k iterations on 8 H20 GPUs. The video length is shuffle from 12-48 during the training. **Stage 3.** Finally, we fixed the alternation-attention layers in front end and train the whole pipeline in all datasets for 20 hours to converge.

4. Experiments

We evaluate our model across all sub-tasks, including 3D tracking (Sec. 4.1), and dynamic 3D reconstruction (Sec. 4.2). In addition, we conduct comprehensive ablation studies (Sec. 4.3) to analyze the impact of key design choices and to demonstrate the effectiveness of unified modeling and scaling up training.

4.1. 3D Point Tracking

Dataset. We evaluate our model and compare it with existing baselines on TAPVid-3D [37], a comprehensive benchmark spanning diverse scenarios including *Driving*, *Ego-*

Table 1. **3D tracking results on the TAPVid-3D benchmark.** We report the 3D Average Jaccard (AJ), Average 3D Position Accuracy (APD_{3D}), and Occlusion Accuracy (OA) across the Aria, DriveTrack, and PStudio subsets. *offl*$^+$ and *offl*$^-$ denote our offline model with/without considering the camera motion, respectively. COL, Univ2, and Mega are abbreviations for COLMAP, UnidepthV2, and MegaSAM. The **best** and the second best are highlighted.

Methods	Type	Depth / Cam Pose	Aria			DriveTrack			PStudio			Average		
			AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑	AJ ↑	APD_{3D} ↑	OA ↑
BootsTAPIR	Type I	COL	9.1	14.5	78.6	11.8	18.6	83.8	6.9	11.6	81.8	9.3	14.9	81.4
TAPTR	Type I	Univ2	15.7	24.2	87.8	12.4	19.1	84.8	7.3	13.5	84.3	11.8	18.9	85.6
LocoTrack	Type I	Univ2	15.1	24.0	83.5	13.0	19.8	82.8	7.2	13.1	80.1	11.8	19.0	82.3
CoTracker3	Type I	COL	8.0	12.3	78.6	11.7	19.1	81.7	8.1	13.5	77.2	9.3	15.0	79.1
		Univ2	15.8	24.4	88.9	13.5	19.9	87.1	9.2	13.8	84.2	12.8	19.4	86.7
		Mega	20.4	30.1	89.8	14.1	20.3	88.5	17.4	27.2	85.0	17.3	25.9	87.8
SpatialTracker	Type II	Univ2	13.6	20.9	90.5	8.3	14.5	82.8	8.0	15.0	75.8	10.0	16.8	83.0
		Mega	15.9	23.8	90.1	7.7	13.5	85.2	15.3	25.2	78.1	13.0	20.8	84.5
SceneTracker	Type II	Univ2	-	23.1	-	6.8	-	-	12.7	-	-	-	14.2	-
DELTA	Type II	Univ2	16.6	24.4	86.8	14.6	22.5	85.8	8.2	15.0	76.4	13.1	20.6	83.0
Ours-*offl*$^-$	Type II	Univ2	18.6	26.3	90.8	16.4	24.3	90.2	18.1	27.6	86.7	17.7	26.0	89.2
		Mega	22.3	32.2	93.7	15.8	23.0	90.0	18.2	28.6	87.3	18.7	27.9	90.5
TAPIP3D	Type III	Mega	23.5	32.8	91.2	14.9	21.8	82.6	18.1	27.7	85.5	18.8	27.4	86.4
Ours-*offl*$^+$	Type III	Mega	**24.7**	**35.2**	**93.9**	16.0	23.4	90.1	18.6	28.7	86.1	19.8	29.1	90.0
		Full-ours	24.6	34.7	93.6	**17.6**	**26.1**	**90.8**	**21.9**	**32.1**	**87.4**	**21.2**	**31.0**	**90.6**

Table 2. **Video depth evaluation. Type I** represents the methods specialized in video depth estimation, while **Type II** are neural reconstruction model, jointly recovering the geometry and camera motion from the video. **Type III** denotes the SoTA optimization-based method. The **best** and the second best results are highlighted.

Method / Metrics	Average		KITTI [19]		TUM Dyn [66]		Bonn [59]		Sintel [52]	
	AbsRel (↓)	$\delta_{1.25}$ (↑)	AbsRel (↓)	$\delta_{1.25}$ (↑)	AbsRel (↓)	$\delta_{1.25}$ (↑)	AbsRel (↓)	$\delta_{1.25}$ (↑)	AbsRel (↓)	$\delta_{1.25}$ (↑)
DepthCrafter [27]	0.143	0.857	0.111	0.885	0.123	0.873	0.066	0.979	0.272	0.693
VDA [9]	0.154	0.882	0.080	0.951	0.118	0.920	0.049	0.982	0.370	0.674
DUSt3R [79]	0.240	0.766	0.124	0.849	0.187	0.792	0.174	0.835	0.475	0.591
MonST3R [91]	0.171	0.802	0.083	0.934	0.197	0.726	0.061	0.954	0.343	0.594
CUT3R [77]	0.186	0.814	0.104	0.899	0.108	0.847	0.068	0.950	0.466	0.560
VGGT [74]	0.104	0.881	**0.051**	0.966	0.068	0.939	0.056	0.963	0.242	0.659
MegaSAM [74]	0.093	0.894	0.069	0.916	0.081	0.935	0.037	0.977	**0.185**	**0.746**
Ours	**0.081**	**0.910**	0.052	**0.973**	**0.045**	**0.976**	**0.028**	**0.988**	0.199	0.703

centric, and *Studio*. This benchmark consists of 4,569 evaluation videos, where the video length varies from 25 to 300 frames, and three 3D point tracking metrics are reported. Specifically, Occlusion Accuracy (OA) measures the precision of occlusion predictions; APD_{3D} denotes the average percentage of estimated errors within multiple threshold scales δ; and Average Jaccard (AJ) quantifies the accuracy of both position and occlusion estimation.

Baselines and Settings. The existing baselines can be broadly categorized into three types. **Type I:** 2D trackers followed by depth lifting. We report the current state-of-the-art 2D tracking models, CoTracker3 [32] and BootsTAPIR [16], as representatives of this category. **Type II:** 3D trackers in camera space. We include SpatialTrackerV1 [86] and DELTA [54], evaluated with two state-of-the-art depth estimators, *i.e.*, UnidepthV2 [60] and MegaSAM [42]. Notably, MegaSAM is an optimization-based SLAM system for estimating consistent video depth, which generally performs better than feedforward models such as UnidepthV2. To ensure fair comparisons under the same camera-space 3D tracking setup, our 3D tracker module is also evaluated in combination with these depth estimators. **Type III:** 3D trackers in world space. We compare our fully end-to-end model with the recently released TAPIP3D [90], which requires consistent video depth and camera poses as input, provided by MegaSAM. For a thorough and fair comparison, we also report results where our depth and pose estimations are replaced by those from MegaSAM.

Quantitative Results and Analysis. In the TAPVid-3D benchmark, our method achieves the best results across various settings, consistently outperforming all baselines under comparable conditions as in Tab. 1. From those detailed comparisons, we can draw several conclusions from it.

- **Better depth estimation is crucial for 3D tracking:** For Type I methods, the 3D tracking performance largely re-

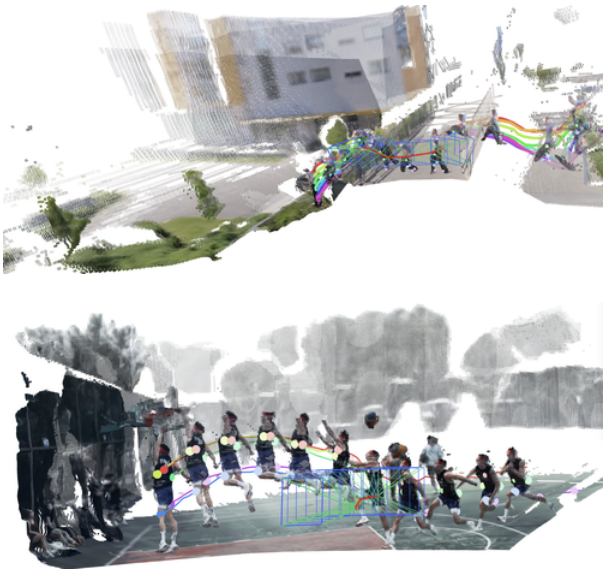


Figure 5. **Fused Point Clouds, Camera Poses, and 3D Point Trajectories.** We visualize the fused point clouds reconstructed from our video depth and camera poses, along with long-term 3D point trajectories in world space.

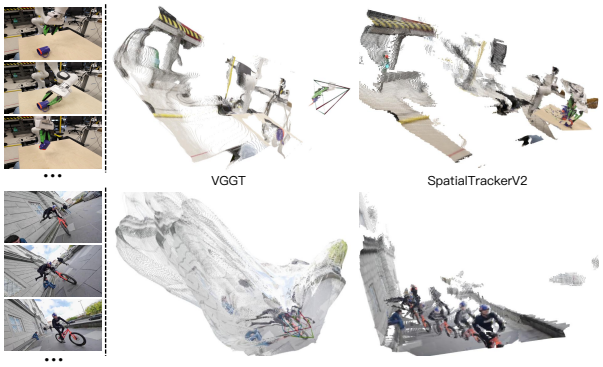


Figure 6. **Qualitative Comparisons on Internet Videos.** To assess generalization, we compare our method with VGGT [74] on challenging Internet videos.

flects the accuracy of the underlying 2D tracks and the associated depth predictions. Intriguingly, CoTracker3 + MegaSAM achieves substantial improvements over CoTracker3 + UniDepthV2, with 17.3 vs. 12.8 in AJ, 25.9 vs. 19.4 in APD_{3D}, and 87.8 vs. 86.7 in OA. This represents the best performance among Type I methods. However, our method Ours-*offl*$^{-}$, which uses the weaker depth estimator UniDepthV2, still outperforms CoTracker3 with MegaSAM, achieving 18.7 vs. 17.3 in AJ and 27.4 vs. 25.9 in APD_{3D}. This is because Type I methods rely purely on back-projection, making them more sensitive to depth inconsistencies. In contrast, our method directly predicts 3D trajectories using a spatial-temporal transformer, which provides inherent temporal smoothness and robustness to noisy inputs.

- **Camera motion decomposition improves 3D tracking:** As shown in the comparison between Ours-*offl*$^{-}$ + MegaSAM and Ours-*offl*$^{+}$ + MegaSAM, incorporating camera pose estimation clearly improves 3D tracking accuracy. For example, on the Aria subset, our method achieves 24.7 vs. 22.3 in AJ and 35.2 vs. 32.2 in APD_{3D}. This improvement is largely due to the design of the TAPVid-3D benchmark, which includes a substantial number of background points for evaluation in Aria. These background points primarily reflect camera motion and are particularly challenging for methods that only track in camera space, especially due to frequent out-of-view scenarios. In DriveTrack and Pstudio, the improvements are less, as DriveTrack includes only dynamic points on moving vehicles, while Pstudio contains static scenes with no camera motion.
- **Conclusion:** Our systematic experiments strongly validate our core insight: decomposing 3D point tracking into video depth and camera pose estimation not only enhances each component individually, but also leads to significantly more accurate and robust 3D tracking as a whole, as shown in Fig. 5.

4.2. Dynamic 3D Reconstruction

4.2.1. Video Depth Evaluation

Datasets. To illustrate the effectiveness and generalization ability of our method, we evaluate our video depth estimation method across four mainstream datasets, *i.e.*, KITTI [19], Sintel [52], Bonn [59], TUM Dynamics [66]. These datasets encompass both indoor and outdoor scenes, featuring video sequences ranging from 50 to 110 frames, providing a comprehensive benchmark for assessing depth estimation consistency across varied environments.

Baselines. We compare our approach against three categories of methods. **Type I** includes SoTA video depth methods, DepthCrafter [27] and Video Depth Anything [87]. **Type II** consists of SoTA deep reconstruction approaches, *i.e.* DUST3R [79], MonST3R [91], CUT3R [77] and VGGT [74] **Type III** denotes the SoTA dynamic Structure-from-Motion (SfM) system, MegaSAM [42]. It leverages an optimization-based paradigm to jointly estimate consistent video depths and camera poses, with constraints enforced by optical flow and monocular depth priors.

Metrics. We evaluate the performance of video depth estimation using geometric accuracy metrics. To ensure fair comparisons with previous works [9, 77, 91], we follow their approach by aligning the predicted video depth to the ground truth using a shared scale and shift, and then compute the Absolute Relative Error (AbsRel) and δ_1 metrics.

Quantitative Results. As shown in Tab. 2, our method clearly outperforms all existing approaches. Specifically,

Table 3. **Evaluation on Camera Pose Estimation** on TUM-dynamic [66], Lightspeed [12], and Sintel [52]. All values are absolute trajectory error (ATE), relative pose error (RPE) for translation and rotation. The **best** and the second best are highlighted.

Method	TUM-dynamic (ATE / RPE_t / RPE_r)	Lightspeed (ATE / RPE_t / RPE_r)	Sintel (ATE / RPE_t / RPE_r)
Particle-SfM [95]	–	0.185 / 0.075 / 2.990	0.129 / 0.031 / 0.535
Robust-CVD [36]	0.153 / 0.026 / 3.528	– / – / –	0.360 / 0.154 / 3.443
CasualSAM [94]	0.071 / 0.010 / 1.712	– / – / –	0.141 / 0.035 / 0.615
DUST3R [79]	0.140 / 0.106 / 3.286	0.412 / 0.177 / 20.100	0.290 / 0.132 / 7.869
CUT3R [77]	0.046 / 0.015 / 0.473	0.274 / 0.067 / 1.561	0.213 / 0.066 / 0.621
MonST3R [91]	0.098 / 0.019 / 0.935	0.149 / 0.046 / 1.210	0.078 / 0.038 / 0.490
VGGT [74]	0.021 / 0.013 / 0.327	0.226 / 0.086 / 1.729	0.082 / 0.043 / 1.253
MegaSAM [42]	0.013 / 0.011 / 0.340	**0.105** / 0.040 / **0.996**	**0.023** / **0.008** / **0.060**
Ours-Front	0.038 / 0.022 / 0.480	0.203 / 0.079 / 1.689	0.075 / 0.045 / 0.805
Ours	**0.012** / **0.010** / **0.305**	0.134 / **0.039** / 1.340	0.054 / 0.027 / 0.288

we surpass our baseline in **Type II**, VGGT [74], by a significant margin. On average, our method achieves an AbsRel of 0.081 compared to 0.104 ($-\mathbf{22.1}\%$), and a $\delta_{1.25}$ of 0.910 compared to 0.881 ($+\mathbf{3.3}\%$). Besides, compared to MegaSAM (**Type III**), our method also maintain clear advantages with 0.081 v.s. 0.093 in AbsRel, and 0.910 v.s. 0.894 in $\delta_{1.25}$. It is important to note that MegaSAM [42] usually needs 5-10 min for a 100 frames video, while our method only takes 5-10 seconds for each which nearly achieves 50 $\times$ faster.

4.2.2. Camera Poses

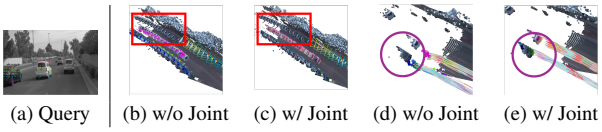
To evaluate the accuracy of camera motion estimation, we conduct the comparisons on Sintel [52], TUM Dynamics [66], and Lightspeed [62]. Sintel and Lightspeed contains numerous challenging dynamic scenes with large ego-motion and object motions, and they are both the synthetic data. TUM Dynamics [66] is a real data, captured by well-calibrated RGBD sensors. We report Absolute Translation Error (ATE), Relative Rotation Error (RPE rot) and Relative Translation Error (RTE) after Sim(3) alignment with the ground truth, as in [77, 91]. Shown in Tab. 3, our method outperforms all regression-based methods, much better than our baseline, VGGT [74] and on par with MegaSAM [42]. Besides, the table illustrates the significance of back end optimization. After joint motion optimization, the pose estimations become nearly twice accurate than before.

4.2.3. Internet Videos.

We present further qualitative comparisons with VGGT [74] on diverse Internet videos. As illustrated in Fig. 6, our method achieves more consistent depth and accurate camera poses, showcasing superior generalization.

4.3. Ablation Analysis

Our ablation study investigates the impact of training data, joint training, and the SyncFormer design on 2D and 3D point tracking. For depth estimation, we analyze different loss functions and training strategies to improve the generalizability of video depth prediction.



(a) Query (b) w/o Joint (c) w/ Joint (d) w/o Joint (e) w/ Joint

Figure 7. The influence of Joint Training.

Table 4. **TAP-Vid DAVIS results.** Cotracker3-3D is a fine-tuned model by adding simple 3D project layer. Mean represents the average of AJ and δ^{vis}_{avg}.

Method	Lift 3D	AJ ↑	δ^{vis}_{avg} ↑	OA ↑	Mean ↑
TAPTR	✗	63.0	76.1	91.1	69.5
LocoTrack	✗	62.9	75.3	87.2	69.1
CoTracker3	✗	64.4	76.9	**91.2**	70.7
DELTA	3D-Dec	62.7	76.7	88.2	69.7
SpatialTracker	Tri-plane	61.1	76.3	89.5	68.7
CoTracker3-3D	3D-Proj	51.6	65.2	85.3	58.4
Ours	SyncFormer	**64.9**	**77.5**	91.0	**71.2**

3D Point Tracking. To illustrate the gains brought by scaling up the 3D point tracking to a wider range of data, we naively train a base model (Base@K in Table 5), and an advanced one, Base@V,K,P, which is trained on VKITTI, Kubric, and PointOdyssey. As the table shows, Base@V,K,P outperforms Base@K by a clear margin. Meanwhile, our final version is clearly better than these two. Besides, different from Base@K, Base@V,K,P is jointly trained on VKITTI, which brings very significant improvements on a similar type of real data: DriveTrack shows 14.7 vs. 7.4 in AJ and 21.9 vs. 13.3 in APD_{3D}. Fig. 7 qualitatively demonstrates the meaning of joint training, i.e., joint training contributes to minimizing the 3D tracking drifts when the model is trained on new patterns of data.

2D Point Tracking. We report a naive alternative to SyncFormer by modifying the final output layer of CoTracker3 [33] to output 2D and 3D tracking. The 2D and 3D embeddings are directly concatenated and projected back to the original dimension using an inserted, learnable linear layer. We name this naive baseline as Cotracker3-3D. As shown in Tab. 4, a simple adaptation of 3D lifting leads to a significant drop in 2D tracking accuracy—AJ drops from 64.4 to 51.6, δ from 76.9 to 65.2, and OA from 91.2 to 85.3. This degradation occurs because the 2D and 3D correlation signals become entangled, disrupting the model's ability to focus on reliable features. Moreover, the update dynamics in 2D and 3D differ substantially, as they occur in separate spaces—UVD space for 2D and camera space for 3D. We also report comparisons with other 3D lifting techniques. These results validate our SyncFormer design, which effectively lifts tracking into 3D while preserving—or even im-

proving—accuracy in 2D.

Depth Estimation. We ablate different training datasets and their coverage to assess their impact on model performance. Additionally, we study the influence of different loss functions applied to real data, *i.e*., the pearson loss and full depth losses. As shown in Tab. 2, if we applied the full losses for depth will cause the obvious performance drop in the synthetic data, *i.e*. Sintel. On the contrary, if we only used the synthetic data to train the model, we found the performance in the real dataset, especially KITTI will heavily drop. Therefore, we leverage the strategy of using different losses for real and synthetic data to offset the influence of domain gap and different errors distribution. As shown in Tab. 2, Ours-Synthetic is our method trained only on the synthetic datasets. Ours-Real-Full is our method using the full depth loss in the real data, where the model was influenced by the noised distributions in the real dataset. The noised of real data has the negative influence on the model's influence on the synthetic data. Meanwhile, the different errors pattern also worsen the model's zero-shot capabilities in the real data.

Table 5. **Ablation of Training Data and Joint Training.** K, V, P represents Kubric [21], VKITTI [7] and PointOdyssey [96] in training. Inference depths are provided by Unidepthv2.

Method	Joint Training	Aria		DriveTrack		PStudio	
		AJ	APD_{3D}	AJ	APD_{3D}	AJ	APD_{3D}
Base@K	No	16.0	24.4	7.4	13.3	12.6	20.3
Base@V, K, P	Yes	15.7	24.1	14.7	21.9	17.2	27.4

5. Conclusion

This work introduces SpatialTrackerV2, a feedforward, scalable, and state-of-the-art approach for 3D point tracking in monocular videos. Built upon a deep exploration of widely-used low- and mid-level representations of motion and scene geometry, our method unifies consistent scene geometry, camera motion, and pixel-wise 3D motion into a fully differentiable end-to-end pipeline. SpatialTrackerV2 accurately reconstructs 3D trajectories from monocular videos, achieving strong quantitative results on public benchmarks and demonstrating robust performance on casually captured Internet videos. We believe SpatialTrackerV2 establishes a solid foundation for real-world motion understanding and brings us a step closer to physical intelligence by exploring large-scale vision data.

Acknowledgment

This work was partially supported by Ant Group Research Intern Program, Zhejiang Provincial Natural Science Foundation of China (No. LR25F020003) and Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

References

[1] Abhishek Badki, Hang Su, Bowen Wen, and Orazio Gallo. L4p: Low-level 4d vision perception unified. *arXiv preprint arXiv:2502.13078*, 2025. 2

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *CVIU*, 110(3), 2008. 3

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 3

[4] Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yijin Li, Fu-Yun Wang, and Hongsheng Li. Gs-dit: Advancing video generation with pseudo 4d gaussian fields through efficient dense 3d point tracking. *CoRR*, abs/2501.02690, 2025. 2

[5] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 - A model zoo for robust monocular relative depth estimation. *CoRR*, abs/2307.14460, 2023. 3

[6] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *CVPR*, pages 8726–8737. IEEE, 2023. 5

[7] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020. 5, 9

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[9] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. *CoRR*, abs/2501.12375, 2025. 3, 5, 6, 7

[10] Zequn Chen, Jiezhi Yang, and Heng Yang. Pref3r: Pose-free feed-forward 3d gaussian splatting from variable-length image sequence. *CoRR*, abs/2411.16877, 2024. 2

[11] 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*, pages 306–325. Springer, 2024. 2

[12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 5, 8

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 3

[14] 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. 2

[15] 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*, pages 10061–10072, 2023. 2

[16] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Proceedings of the Asian Conference on Computer Vision*, pages 3257–3274, 2024. 3, 6

[17] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, pages 2366–2374, 2014. 3

[18] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011. Computer Vision Foundation / IEEE Computer Society, 2018. 3

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 6, 7

[20] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022. 5

[21] 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*, pages 3749–3761, 2022. 5, 9

[22] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. *CoRR*, abs/2501.03847, 2025. 2

[23] Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, 2022. 2

[24] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 3

[25] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997. 3

[26] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE TPAMI*, 46(12):10579–10596, 2024. 3

[27] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *CoRR*, abs/2409.02095, 2024. 2, 3, 6, 7

[28] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, pages 2821–2830. Computer Vision Foundation / IEEE Computer Society, 2018. 5

[29] Hyeonho Jeong, Chun-Hao Paul Huang, Jong Chul Ye, Niloy J. Mitra, and Duygu Ceylan. Track4gen: Teaching video diffusion models to track points improves video generation. *CoRR*, abs/2412.06016, 2024. 2

[30] Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4d: Learning how things move in 3d from internet stereo videos. *arXiv preprint*, 2024. 5

[31] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023. 5

[32] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 3, 5, 6

[33] 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*, pages 18–35. Springer, 2024. 2, 3, 4, 8

[34] Yoni Kasten, Wuyue Lu, and Haggai Maron. Fast encoder-based 3d from casual videos via point track processing. *arXiv preprint arXiv:2404.07097*, 2024. 2

[35] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE international conference on computer vision*, pages 1521–1529, 2017. 3

[36] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1611–1621. Computer Vision Foundation / IEEE, 2021. 8

[37] Skanda Koppula, Ignacio Rocco, Yi Yang, Joseph Heyward, João Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. Tapvid-3d: A benchmark for tracking any point in 3d. In *NeurIPS*, 2024. 2, 5

[38] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024. 2

[39] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition (ICPR'06)*, pages 630–633. IEEE, 2006. 3

[40] 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*, pages 57–75. Springer, 2024. 2

[41] Xiaodi Li, Zongxin Yang, Ruijie Quan, and Yi Yang. Drip: Unleashing diffusion priors for joint foreground and alpha prediction in image matting. *Advances in Neural Information Processing Systems*, 37:79868–79888, 2024. 3

[42] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *CVPR*, 2025. 2, 6, 7, 8

[43] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. *arXiv preprint arXiv:2305.04926*, 2023. 3

[44] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 5

[45] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010. 2

[46] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4d egocentric dataset for category-level human-object interaction. In *CVPR*, pages 20981–20990. IEEE, 2022. 5

[47] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 2

[48] Haozhe Lou, Yurong Liu, Yike Pan, Yiran Geng, Jianteng Chen, Wenlong Ma, Chenglong Li, Lin Wang, Hengzhen Feng, Lu Shi, Liyi Luo, and Yongliang Shi. Robo-gs: A physics consistent spatial-temporal model for robotic arm with hybrid representation. *CoRR*, abs/2408.14873, 2024. 2

[49] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. ICCV*, 1999. 3

[50] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. 3

[51] Wei-Chiu Ma, Anqi Joyce Yang, Shenlong Wang, Raquel Urtasun, and Antonio Torralba. Virtual correspondence: Humans as a cue for extreme-view geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15924–15934, 2022. 3

[52] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 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*, pages 4040–4048, 2016. 6, 7, 8

[53] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*, 2023. 5

[54] Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. *arXiv preprint arXiv:2410.24211*, 2024. 2, 3, 6

[55] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 3

[56] Dantong Niu, Yuvan Sharma, Haoru Xue, Giscard Biamby, Junyi Zhang, Ziteng Ji, Trevor Darrell, and Roei Herzig. Pre-training auto-regressive robotic models with 4d representations. *arXiv preprint arXiv:2502.13142*, 2025. 2

[57] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. Codef: Content deformation fields for temporally consistent video processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8089–8099, 2024. 2

[58] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017. 3

[59] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *IROS*, 2019. 6, 7

[60] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segù, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *CVPR*, pages 10106–10116. IEEE, 2024. 3, 6

[61] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotný. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*. IEEE, 2021. 5

[62] Chris Rockwell, Joseph Tung, Tsung-Yi Lin, Ming-Yu Liu, David F. Fouhey, and Chen-Hsuan Lin. Dynamic camera poses and where to find them. In *CVPR*, 2025. 8

[63] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 2

[64] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International journal of computer vision*, 80:72–91, 2008. 2

[65] Samarth Sinha, Roman Shapovalov, Jeremy Reizenstein, Ignacio Rocco, Natalia Neverova, Andrea Vedaldi, and David Novotný. Common pets in 3d: Dynamic new-view synthesis of real-life deformable categories. In *CVPR*, pages 4881–4891. IEEE, 2023. 5

[66] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, pages 573–580. IEEE, 2012. 6, 7, 8

[67] Tutian Tang, Minghao Liu, Wenqiang Xu, and Cewu Lu. Kalib: Markerless hand-eye calibration with keypoint tracking. *CoRR*, abs/2408.10562, 2024. 2

[68] Zachary Teed and Jia Deng. 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*, pages 402–419. Springer, 2020. 2, 3

[69] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a mod-

ern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 298–372. Springer, 2000. 3

[70] Massimiliano Viola, Kevin Qu, Nando Metzger, Bingxin Ke, Alexander Becker, Konrad Schindler, and Anton Obukhov. Marigold-dc: Zero-shot monocular depth completion with guided diffusion. *CoRR*, abs/2412.13389, 2024. 3

[71] Bo Wang, Jian Li, Yang Yu, Li Liu, Zhenping Sun, and Dewen Hu. Scenetracker: Long-term scene flow estimation network. *arXiv preprint arXiv:2403.19924*, 2024. 3

[72] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023. 3

[73] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024. 3

[74] 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*, pages 5294–5306, 2025. 2, 3, 6, 7, 8

[75] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19795–19806, 2023. 2, 3

[76] 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. 2

[77] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025. 6, 7, 8

[78] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision, 2024. 5

[79] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 6, 7, 8

[80] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 5

[81] Zhouxia Wang, Yushi Lan, Shangchen Zhou, and Chen Change Loy. Objctrl-2.5 d: Training-free object control with camera poses. *arXiv preprint arXiv:2412.07721*, 2024. 2

[82] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, pages 803–814. IEEE, 2023. 5

[83] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. RGBD objects in the wild: Scaling real-world 3d object learning from RGB-D videos. In *CVPR*, pages 22378–22389. IEEE, 2024. 5

[84] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 3

[85] Yuxi Xiao, Nan Xue, Tianfu Wu, and Gui-Song Xia. Level-S^2fM: Structure From Motion on Neural Level Set of Implicit Surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17205–17214, 2023. 3

[86] 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*, pages 20406–20417, 2024. 2, 3, 6

[87] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 2, 3, 7

[88] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2025. 2, 3

[89] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *CVPR*, pages 3906–3915. IEEE, 2022. 3

[90] Bowei Zhang, Lei Ke, Adam W Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. *arXiv preprint arXiv:2504.14717*, 2025. 3, 6

[91] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024. 6, 7, 8

[92] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, pages 592–611. Springer, 2022. 3

[93] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. *arXiv preprint arXiv:2402.14817*, 2024. 3

[94] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T. Freeman. Structure and motion from casual videos. In *ECCV*, pages 20–37. Springer, 2022. 8

[95] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *ECCV*. Springer, 2022. 2, 8

[96] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 5, 9