RoboTrack
Point Tracking for Real-World Robotic Manipulation

Vincent Shao[1,2] **Max Argus**[1] **Winson Han**[1] **Eli VanderBilt**[1] **Carissa Schoenick**[1]

Ali Farhadi[2] **Ranjay Krishna**[2] **Zhongzheng Ren**[1,3] **Jae Sung Park**[1]

Allen Institute for AI[1], University of Washington[2], University of North Carolina at Chapel Hill[3]

🤗 **Models:** RoboTracker RoboTracker-Warp

Data: RoboTrack-Bench RoboTrack-15M

Code: https://github.com/allenai/robotrack

Paper: allenai.org/papers/robotrack

Abstract

Ai2

Point trackers turn video into motion trajectories that robotic systems can rely on to learn from demonstrations and guide their manipulation. However, we find that off-the-shelf trackers with strong performance on established benchmarks struggle with the varied interactions and viewpoints of real-world manipulation. To address this gap, we introduce **RoboTrack**, a benchmark and training suite for point tracking in robotic manipulation. Our benchmark, **RoboTrack-Bench**, contains 307 real-world clips with 2,522 human-annotated tracks covering four robot embodiments and human manipulation from exocentric and egocentric views. Evaluation shows that many existing trackers achieve lower scores on RoboTrack-Bench than on prior robotics benchmarks, with even the strongest baseline achieving only 63.9 Average Jaccard (AJ), compared with 85.5 on RGB-Stacking and 70.5 on RoboTAP. To improve tracking in robotic manipulation, we extend the MolmoSpaces simulator to generate **RoboTrack-15M**, a synthetic training set of manipulation-aligned point tracks. We find that fine-tuning on this data improves RoboTrack-Bench by up to 2.1 AJ without hurting general-video tracking and yields state-of-the-art results on all robotics benchmarks, reaching 66.0 AJ on RoboTrack-Bench. Lastly, we demonstrate how point tracking can be applied to a realistic robotics task in a wet lab setting. Using point tracks from our fine-tuned tracker, a robot can transfer liquid from a tube into a vial in 17 of 20 trials despite occlusions, compared with at most 8 for flow- and mask-based alternatives.

1 Introduction

Point tracking estimates the motion of arbitrary image points through a video, across changes in viewpoint, deformation, and occlusion [21, 12]. Modern point trackers are trained with synthetic data, where ground-truth trajectory annotations can be generated automatically from simulated scenes [19, 63]. Despite relying heavily on synthetic supervision, these models generalize well to real-world videos [8, 29, 30, 14]. Point tracking is particularly attractive for robotic manipulation, where tracks of the motion of physical surfaces provide a compact representation for grasping, pushing, placing, and other object interactions [4, 56, 57, 52, 25, 61, 60].

However, manipulation in particular presents difficult tracking conditions: robot grippers occlude objects during interaction, objects move and rotate with the robot, and previously hidden points must be recovered when

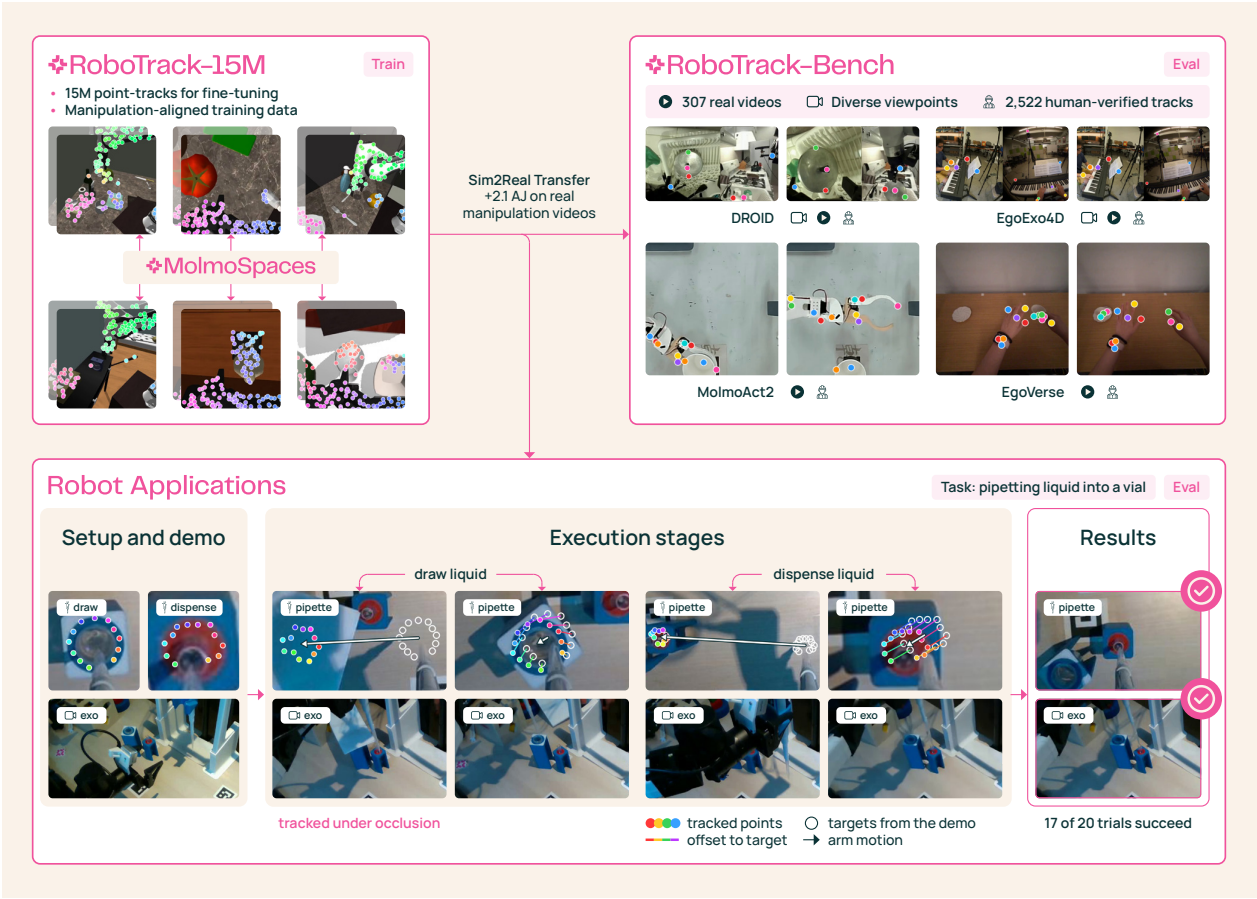


Figure 1 RoboTrack is a data suite for point tracking in robotic manipulation. **RoboTrack-15M** (top left) provides manipulation-aligned training data from MolmoSpaces that improves tracking on real robot videos. **RoboTrack-Bench** (top right) provides human-annotated point tracks on diverse in-the-wild manipulation videos. Lastly, we apply tracking to a real wet lab automation robotics task, where points from a single demo are tracked in the live camera and the arm is servoed until the points align with their targets in the demo.

they reappear. Despite strong results on existing robotic tracking benchmarks, including RGB-Stacking [12] and RoboTAP [52], we find that current trackers frequently fail under these conditions (Figure 2).

To evaluate this gap, we introduce **RoboTrack-Bench**, a benchmark containing 307 real manipulation videos and 2,522 human-verified point trajectories from DROID [32], Ego-Exo4D [18], EgoVerse [45], and MolmoAct2 [16], spanning diverse robots, objects, environments, and ego- and exocentric viewpoints. On RoboTrack-Bench, we find that every leading tracker performs worse than on RoboTAP [52], with drops of 3.0–6.6 Average Jaccard (AJ) points. The gaps between trackers also narrow, as CoWTracker [37], the strongest baseline, leads CoTracker3 (online) [30] by 4.7 AJ on RoboTAP but by only 2.1 AJ on RoboTrack-Bench. Existing benchmarks thus overstate both how ready current trackers are for in-the-wild manipulation and how much they differ from one another.

We next investigate whether supervision with synthetic manipulation videos can narrow this gap. Existing synthetic datasets capture general object motion, rigid-body dynamics, and articulated characters [21, 19, 63], but do not specifically target robot-object interactions. We extend MolmoSpaces [33] to annotate simulated robotic manipulation with ground-truth point trajectories, producing **RoboTrack-15M**.

Our results show that RoboTrack-15M transfers beyond the simulated environment. Starting from a Kubric-pretrained AllTracker, we replace its second training stage with just 10k steps on a mixture of Kubric and RoboTrack-15M. Compared with training on Kubric alone, this improves tracking on all five evaluated benchmarks, including gains of 1.5 AJ on RoboTrack-Bench, 3.0 AJ on RGB-Stacking, 1.2 AJ on DAVIS, 0.7



Figure 2 Existing point tracking models in real manipulation videos. Hollow points are predicted occluded. The state-of-the-art trackers lose correspondence with manipulated objects, such as the tip of a knife, following brief occlusions by the gripper.

AJ on RoboTAP, and 0.6 AJ on Kinetics. The fine tuned model also matches or outperforms the released AllTracker, whose second stage runs for 400K steps, on all benchmarks except DAVIS. We also train using our data starting from the final AllTracker to produce a stronger model, which we call **RoboTracker**. The gains carry over to a different architecture as well; CoWTracker fine-tuned in the same way, which we call **RoboTracker-Warp**, improves RoboTrack-Bench by 2.1 AJ to 66.0 and achieves **state-of-the-art results on all robotics benchmarks**.

Lastly, we show an example application of point tracking to manipulation with a real robot in a wet lab setting in which a robot has to pipette liquid into a small vial, similar to RoboTAP [52]. We find point tracks from the tracker fine-tuned on RoboTrack-15M reliably guide the robot's policy, even when the view of the task is partly occluded. Together, RoboTrack-Bench and RoboTrack-15M provide an evaluation and training suite for studying how synthetic point trajectories transfer to real-world manipulation.

2 Related Work

Track any point. Visual tracking has been studied at several levels of granularity, from bounding boxes in driving, pedestrian, and surgical video [54, 58, 42], to masks for video segmentation [46, 7], keypoints for human and animal pose [48, 39, 43], and individual points [17, 1, 13, 12, 29, 30, 23]. The point formulation, introduced as Particle Video [47], has received renewed attention through datasets and models for tracking arbitrary queried points [22, 12]. While closely related to optical flow [24, 15, 50], point-tracking additionally requires long-range association and visibility prediction. Recent trackers build on architectures developed for flow and correspondence [50, 55]. Several use cost volumes with iterative refinement [22, 13, 12, 8], CoTracker [29, 30] tracks many points jointly, TAPNext [64] recasts tracking as next-token prediction, AllTracker [23] estimates dense long-range correspondence for every pixel, and TAPIP3D [59] tracks points in 3D.

Training data for point tracking. Because point tracks are hard to obtain on real video, trackers mostly train on synthetic data, such as FlyingThings++ [21, 41], Kubric [19], which renders objects falling under camera motion, and PointOdyssey [63], which adds long human and animal motion. Despite this, recent trackers transfer well to real video, and further gains come from pseudo-labeling real footage [14, 30, 51], dense flow supervision [23], or stronger augmentation [28]. Tracks on the robot can be derived from joint states and

Table 1 Point tracking benchmarks for robotic manipulation. RoboTrack-Bench covers the most videos, embodiments, and occlusions on real manipulation with the least static points (move less than one pixel at 256-pixel height).

	RGB-Stacking	RoboTAP	RoboTrack-Bench
Source	Synthetic	Real	Real
Videos	50	265	307
Frames / video	250	272	223
Points / video	30	43.7	8.2
Resolution	256^2	296×222	≥ 640×360
Camera views	exo	exo, ego	exo, ego
Embodiments	1	1	**4 + human**
Static points	37%	20%	**2.2%**
Occluded at least once	48%	56%	**57%**

calibrated cameras [35], but this covers only the robot and can mislabel occlusions caused by other objects.

Benchmarks for point tracking. TAP-Vid [12] established the standard evaluation protocol on real videos from DAVIS [44] and Kinetics [31]. PointOdyssey [63] broadens synthetic evaluation to diverse domains for long-term point tracking, while SynthVerse [62] includes embodied manipulation. Domain-specific benchmarks cover animals [6, 40], surveillance [49], driving [3], and egocentric video [9], while ITTO [11] emphasizes complex motion and re-identification after occlusion. TAPVid-3D [34] and WorldTrack [17] extend evaluation to 3D trajectories. For robotics, TAP-Vid [12] includes synthetic annotations on the RGB-Stacking dataset [38], while RoboTAP [52] provides human-annotated tracks for real tabletop manipulation videos. These latter two benchmarks emphasize relatively controlled tabletop settings, with limited scene clutter, camera variation, and object diversity. Recent work such as TAPVid-MV [35] evaluates multi-view tracking on DROID [32], but the trajectories are derived only from the robot and static background. Table 1 contrasts existing robotics benchmarks with RoboTrack-Bench, highlighting differences in camera viewpoints, manipulation embodiments, and trajectory characteristics.

Point tracking in robotics. Point tracks provide a useful representation of object motion for learning manipulation from demonstration videos [56, 5, 20, 57] and visual feedback for guiding execution. Track2Act [4] uses point tracks to predict object trajectories, while RoboTAP [52] uses keypoint re-detection to transfer behavior from demonstrations to live execution. These uses extend to 3D, where tracks serve as manipulation plans [27], targets for dexterous control [36], and supervision for dynamics prediction [26]. Such applications depend on off-the-shelf trackers like CoTracker [30] to obtain the point tracks. RoboTrack-Bench exposes where these trackers fail on real manipulation videos, while we introduce RoboTrack-15M as training data to address these failures.

3 RoboTrack: Benchmark and Training Data for Robotic Manipulation

In this section, we detail the benchmark statistics and dataset creation framework for RoboTrack.

3.1 RoboTrack-Bench: Real Manipulation Videos

Video sources RoboTrack-Bench contains 307 human-annotated clips with 2,522 point tracks, drawn from DROID [32], MolmoAct2 [16], EgoVerse [45], and Ego-Exo4D [18] (Figure 3). The clips span four robot embodiments and human manipulation, selected for diversity in scenes, actions, and camera viewpoints. For subsets of DROID and Ego-Exo4D, we include synchronized egocentric and exocentric recordings of the same interaction, yielding 33 video pairs that support studying point correspondence across views. Resolutions and per-source statistics are in Table 2.

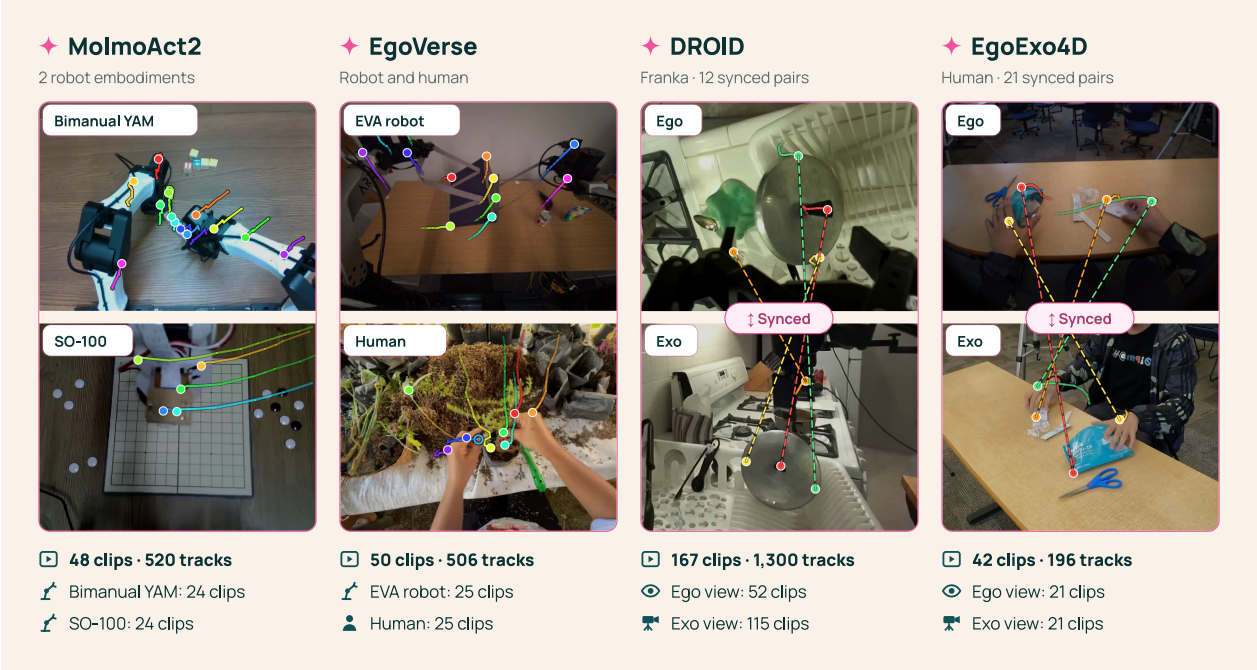


Figure 3 RoboTrack-Bench at a glance. Examples from each source dataset (307 clips and 2,522 human-verified tracks in total). MolmoAct2 and EgoVerse show one embodiment per panel. DROID and Ego-Exo4D show synchronized ego (top) and exo (bottom) views.

Table 2 Composition of RoboTrack-Bench. Exo and Ego count clips by camera view; Ego includes wrist-mounted cameras. 24 clips from DROID (synchronized pairs) and 42 clips from Ego-Exo4D form 33 synchronized exo–ego pairs.

Source	Embodiment	Resolution	Exo	Ego	Clips	Tracks
DROID (single camera)	Franka	1280 × 720	103	40	143	1,236
DROID (synchronized pairs)	Franka	1280 × 720, 672 × 376	12	12	24	64
MolmoAct2	bimanual YAM	640 × 360	23	1	24	263
MolmoAct2	SO-100	640 × 480	14	10	24	257
EgoVerse	EVA	640 × 480	25	0	25	265
EgoVerse	Human	640 × 360 to 640 × 480	0	25	25	241
Ego-Exo4D	Human	1920 × 1080, 1408 × 1408	21	21	42	196
Total			**198**	**109**	**307**	**2,522**

Human annotation pipeline. We follow TAP-Vid [12] in the design of our human annotation pipeline. Annotators are instructed to select any moving point in the video, with no restriction on object categories, and are encouraged to select points that have interesting motion or occlusions.

Step 1: Point selection. Annotators first watch the entire video and choose five points to annotate. Since exocentric robotics videos tend to contain many static points, we instruct annotators to only choose points which move at some point in the video. This eliminates background points and static objects. Each point is selected on the first frame in which it appears, which becomes the query point and frame in RoboTrack-Bench. After selecting a point, annotators label it in one of three categories based on the object that the point lies on: Embodiment (Robot or Human), Manipulated Object, Static Object (Camera Motion).

Step 2: Track refinement. Similar to TAP-Vid [12], we provide a track assist algorithm that uses SEA-RAFT [55] to convert sparsely chosen points into dense tracks that follow the motion of the point. For each of the five points, annotators review the initial track produced by the tracking assistant and add refinement points to fix its drift, making further corrections as needed. Annotators also mark segments in which the point is occluded; they are not required to track the point during occluded sections. See Appendix B.1 for

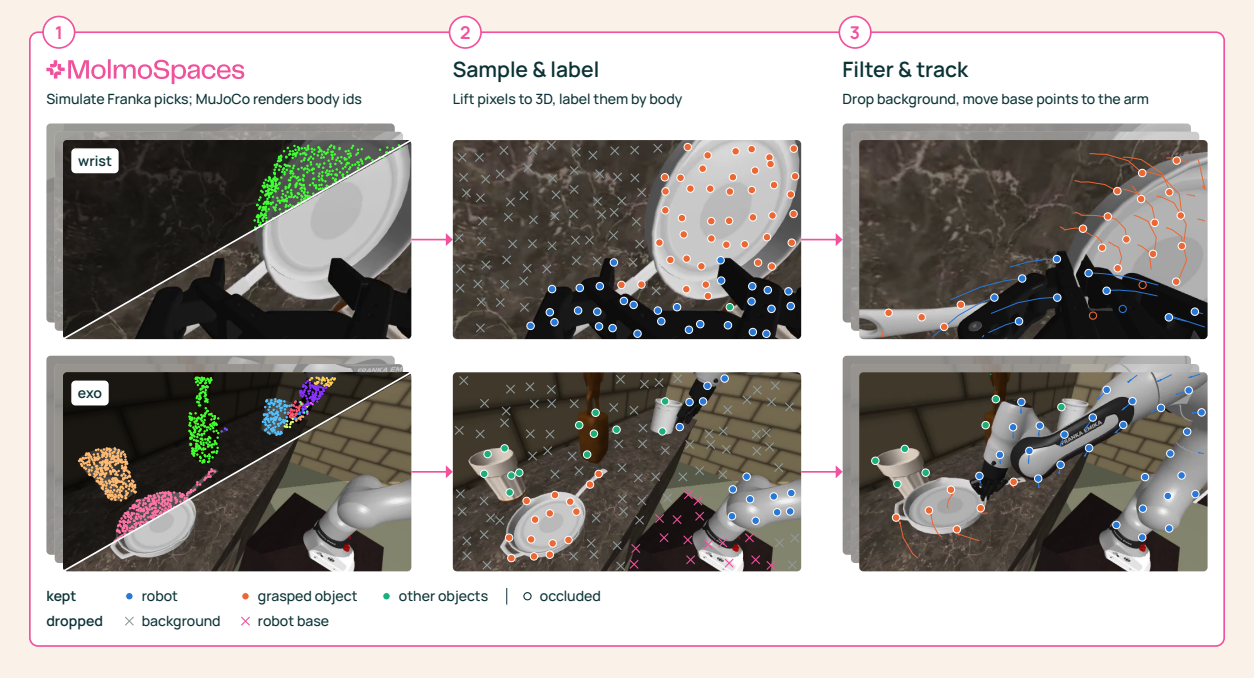


Figure 4 Data generation pipeline of RoboTrack-15M. Objects, indoor scenes, and robot grasps are obtained from MolmoSpaces and rendered in MuJoCo. Point tracks are obtained by unprojecting sampled pixels to 3D and projecting them into every frame via camera intrinsics. Background and robot-base points are filtered out, and visibility is determined by a depth test.

further details on the tracking assistant.

Step 3: Verification. Finally, in a separate verification pass, annotators check every frame for localization and visibility errors, making minor corrections where needed and accepting only fully correct tracks.

3.2 RoboTrack-15M: Synthetic Simulation Data

MolmoSpaces [33] is a robotics simulation ecosystem for generating training and testing data with diverse scenes, objects, and actions. We augment the MolmoSpaces data generation pipeline to produce 2D point tracks for the Tracking Any Point task (Figure 4).

Scene generation. We use the MolmoSpaces scene generation as is, making no changes to how the robot picks up objects or to which indoor environments and objects are chosen. The houses are selected from ProcTHOR [10], each with differing layouts of walls, floors, furniture, and lighting. The robot, a Franka arm, is placed in a valid location, and objects are spawned near the robot for manipulation. We use MuJoCo Classic for rendering and physics simulation. Two cameras capture different perspectives: a wrist camera on the gripper, and an exocentric camera that sees the robot arm and the objects being manipulated.

Obtaining point tracks. One straightforward way to obtain point tracks from simulation is to use mesh vertices, following PointOdyssey [63]. However, we found that this produces ambiguous points near object edges, which appear to be off the object but are meant to be on it. Additionally, many of our objects are low-poly, so vertex sampling severely limits where points can be placed. Instead, we use an approach closer to Kubric [19]: we sample points from visible object pixels, unproject them to 3D using rendered depth, and store their positions in each body's local coordinate frame. This allows us to sample from every object pixel in the video, vastly increasing the diversity of our points.

RoboTrack-15M. The released training set covers 300 ProcTHOR houses with 5 pick episodes each, recorded from two cameras, giving 3,000 videos with 5,000 tracks each (15M tracks in total). Candidate pixels are drawn every 20 frames, uniformly over the visible pixels of movable objects and robot links, and none on walls, floors, or fixed furniture. See Table 6 in the Appendix for dataset statistics.

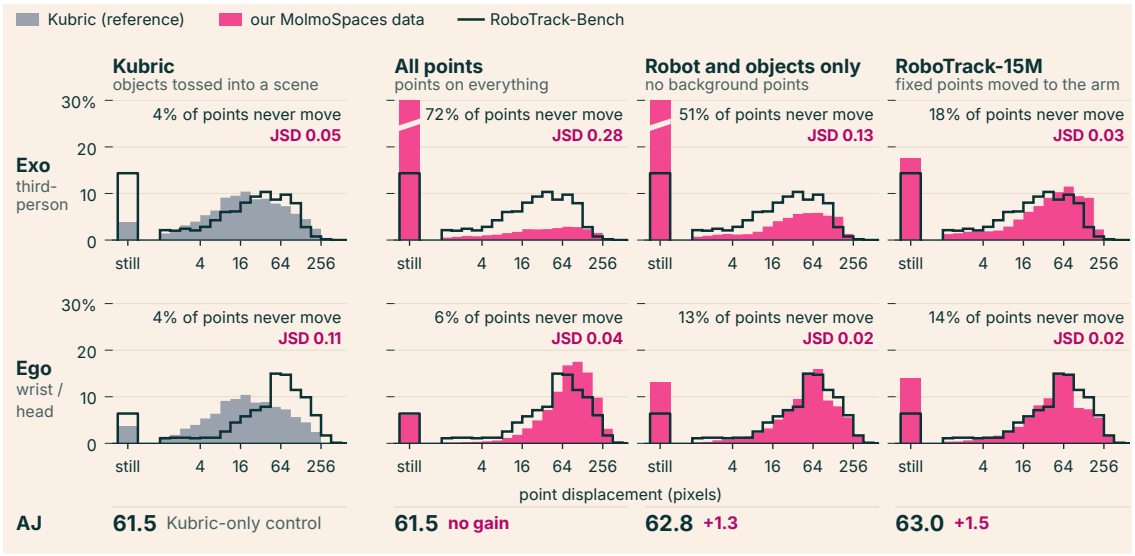


Figure 5 Effect of point selection on point motion in our training data, compared with RoboTrack-Bench. Each histogram shows how far points move on exo (top) and ego (bottom) cameras. The x-axis is the largest distance each point moves from its first visible position throughout the clip, in pixels at 256-pixel image height, and the y-axis is the percentage of points. Similarity to RoboTrack-Bench is measured with the Jensen-Shannon divergence (JSD). While Kubric closely matches exo videos (JSD 0.05), it shows low resemblance with ego videos (JSD 0.11). AJ is measured on RoboTrack-Bench after fine-tuning AllTracker on each dataset mixed with Kubric (Section 4.2).

Selecting supervised points. We study how the selection of supervised points affects sim-to-real transfer by varying the proportion of tracks on robot surfaces, manipulated objects, and backgrounds. Our first attempt sampled points across the scene, but fine-tuning AllTracker on a mix of Kubric and this data (Section 4.2) brought no improvement over fine-tuning on Kubric alone. To diagnose this, we compare the point motion distribution of our data with that of RoboTrack-Bench. For each point, we measure its maximum displacement from its first visible position over 56 frames at 15 fps, normalized to an image height of 256 pixels, and compute the Jensen–Shannon divergence (JSD) between the resulting distributions in Figure 5. Our first attempt shows a large divergence from RoboTrack-Bench on exo videos (JSD 0.28), as most sampled points lie on static background. To close this sim-to-real gap, we first restrict points to the robot and manipulated objects, which aligns our ego videos with RoboTrack-Bench. We then move points on the fixed robot base onto the moving arm, which aligns the exo videos as well (JSD 0.03). The resulting RoboTrack-15M achieves the highest AJ on our benchmark. Note that Kubric closely matches the exo videos (JSD 0.05) but diverges on the ego videos (JSD 0.11). Its objects move on their own in front of a static camera, whereas in robot manipulation the camera often moves with the robot, which may make Kubric less suited for manipulation tasks.

4 Experiments

We first evaluate baseline tracking models on our benchmark alongside existing ones. Then we ablate different training data compositions and compare performance on robotics benchmarks.

Baseline models. We evaluate a range of state-of-the-art point trackers. For CoTracker3 [30], we include the Kubric-only model and a variant fine-tuned on 15k pseudo-labeled real videos. TAPNext and BootsTAPNext [64] share an architecture but are trained on Kubric alone and additionally fine-tuned on 15M real videos, respectively. AllTracker [23] is a dense tracker with similar architecture to CoTracker3, trained on Kubric and then on a mixture of optical flow and point tracking datasets. CoWTracker [37] replaces cost volumes with feature warping on top of a pretrained VGGT [53] backbone and is trained on Kubric.

Evaluation protocol. We use the standard TAP-Vid [12] metrics: Occlusion Accuracy (OA), δ_{vis}, and Average Jaccard (AJ). All metrics are computed after rescaling predictions and ground truth to 256×256. We evaluate

Table 3 Point tracking results on robotic manipulation and general videos. Higher is better for all metrics. Best results are in bold and second best are underlined, and our models are shaded. Among the off-the-shelf models, CoWTracker [37] performs best on average across the robotics benchmarks, followed by AllTracker [23]. RoboTracker and RoboTracker-Warp are AllTracker and CoWTracker fine-tuned with RoboTrack-15M, improving AJ over their base models on all benchmarks except RoboTAP for RoboTracker and DAVIS for RoboTracker-Warp. For training data, 15k and 15M are the real videos pseudo-labeled by CoTracker3 [30] and BootsTAPNext [14], and mix is AllTracker's mix of point tracking and optical flow datasets in stage 2 training.

			Robotic manipulation												General					
			RoboTrack-Bench			RGB-Stack			RoboTAP			Avg.			Kinetics			DAVIS		
Model	Train. Data	Params	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA
CoTracker3 (offline)	Kubric	25M	59.8	73.2	84.6	76.6	86.4	90.8	59.2	73.7	85.9	65.2	77.8	87.1	53.4	66.4	86.1	63.6	76.7	88.2
CoTracker3 (offline)	Kubric+15k	25M	60.2	72.6	85.8	74.4	85.0	91.6	64.7	78.0	89.4	66.5	78.5	88.9	54.8	67.8	87.6	64.8	76.9	91.7
CoTracker3 (online)	Kubric+15k	25M	61.8	74.9	87.3	70.7	83.3	91.4	65.8	78.9	90.6	66.1	79.0	89.8	55.1	68.1	88.3	64.4	76.8	90.6
TAPNext	Kubric	194M	55.1	69.7	87.2	75.0	85.4	**95.2**	58.7	72.2	87.8	62.9	75.8	90.1	48.8	63.3	85.6	62.8	77.0	90.5
BootsTAPNext	Kubric+15M	194M	60.2	73.5	87.1	65.5	78.2	86.8	63.2	74.5	88.4	62.9	75.4	87.5	52.4	65.9	86.3	**66.6**	**79.5**	**92.1**
AllTracker	Kubric	17M	61.4	74.5	83.9	80.7	90.1	91.7	67.5	79.3	90.8	69.9	81.3	88.8	55.8	67.9	86.9	62.8	76.2	87.2
AllTracker	Kubric+mix	17M	62.6	75.7	86.9	81.4	90.1	92.9	68.5	80.8	91.7	70.8	82.2	90.5	56.3	69.0	88.8	64.3	77.1	89.5
CoWTracker	Kubric	~1B	63.9	76.1	88.5	85.5	93.0	94.8	70.5	81.2	93.9	73.3	83.4	92.4	57.6	69.9	**90.0**	66.1	78.4	91.2
RoboTracker	Kubric+mix+RoboTrack-15M	17M	63.4	76.1	86.2	83.2	90.9	93.5	68.4	80.6	91.3	71.7	82.5	90.3	56.6	69.0	87.6	64.6	77.4	89.1
RoboTracker-Warp	Kubric+RoboTrack-15M	~1B	**66.0**	**77.9**	**89.0**	**85.9**	**93.2**	94.5	**71.6**	**81.9**	**94.2**	**74.5**	**84.4**	**92.6**	**58.0**	**70.3**	89.7	65.9	78.9	89.9

every tracker ourselves, each at its own input resolution: CoTracker3, TAPNext, and BootsTAPNext receive frames resized to 256×256, AllTracker receives 384×512 frames, and CoWTracker 336×560 frames.

Following TAP-Vid [12] and CoTracker3 [30], we evaluate in the "first query" mode, where query points are sampled from the first frame in which they become visible. On RoboTAP and Kinetics we evaluate every annotated point over the full length of every video, unlike AllTracker's protocol [23], which trims videos to at most 600 frames and tracks only the points visible at the start frame. For each model, we follow the evaluation protocols outlined in their papers.

4.1 Benchmark Results

Performance on existing robotics benchmarks does not transfer to in-the-wild manipulation. Table 3 compares state-of-the-art point trackers on RoboTrack-Bench and four established benchmarks. Every evaluated model except CoTracker3 (offline, Kubric) scores lower on RoboTrack-Bench than on RoboTAP, with the strongest baseline, CoWTracker, dropping from 70.5 on RoboTAP to 63.9 AJ on RoboTrack-Bench. This suggests that existing robotics benchmarks do not fully capture the challenges of diverse real-world manipulation that RoboTrack-Bench covers with a broader range of objects, environments, and embodiments. Interestingly, AllTracker outperforms CoTracker3 (online) by 2.7 AJ on RoboTAP, but its lead shrinks to just 0.8 AJ on RoboTrack-Bench, and Kubric-only AllTracker falls behind it (61.4 vs. 61.8 AJ). Rankings can even reverse on robotics benchmarks, as TAPNext outperforms BootsTAPNext by 9.5 AJ on the synthetic RGB-Stacking but trails it by 5.1 AJ on RoboTrack-Bench.

Strong general-video tracking performance does not guarantee robust manipulation tracking. BootsTAPNext ranks first on DAVIS, yet falls significantly behind on RoboTrack-Bench, 3.7 AJ behind CoWTracker. It even falls behind the Kubric-only AllTracker on RoboTrack-Bench (60.2 vs. 61.4 AJ), despite outperforming it on DAVIS and being additionally trained on 15M pseudo-labeled real videos. This suggests that gains from large-scale real-video supervision alone do not necessarily carry over to manipulation, and we hypothesize that general web videos rarely contain the close-range hand–object interactions and occlusions common in manipulation.

Fine-tuning on RoboTrack-15M improves performance across benchmarks. To test whether simulated manipulation data can close this gap, we fine-tune AllTracker and CoWTracker on a mix of Kubric and RoboTrack-15M, which we call RoboTracker and RoboTracker-Warp respectively (bottom of Table 3). RoboTracker-Warp improves on all robotics benchmarks, while RoboTracker improves on RoboTrack-Bench and RGB-Stacking and matches AllTracker on RoboTAP. The largest gain appears on RoboTrack-Bench, where RoboTracker-Warp reaches 66.0 AJ (+2.1) and is **the state-of-the-art tracker on every robotics benchmark**. These gains come

Table 4 Data ablations with AllTracker. "+" indicates the additional dataset that was added to Kubric. "mix" is AllTracker's own second-stage mix. Avg. is the mean over the five benchmarks. Bold marks the best value within each block.

	RoboTrack-Bench			RGB-Stack			RoboTAP			Kinetics			DAVIS			Avg.		
Training data	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA
From AllTracker (Kubric)																		
Released checkpoint, no fine-tuning	61.4	74.5	83.9	80.7	90.1	91.7	67.5	79.3	90.8	55.8	67.9	86.9	62.8	76.2	87.2	65.6	77.6	88.1
Kubric only (control)	61.5	74.6	83.9	80.6	89.7	92.1	67.5	79.3	90.7	55.9	68.1	86.9	62.4	75.9	86.7	65.6	77.5	88.1
+ PointOdyssey	61.2	**75.8**	82.3	76.4	89.8	87.2	64.7	79.9	87.2	54.4	**68.9**	83.9	63.0	**77.0**	86.6	64.0	78.3	85.4
+ RoboTrack-15M	63.0	**75.8**	85.3	83.6	91.2	93.5	68.2	80.2	91.2	56.5	68.5	87.5	63.6	76.7	87.7	67.0	78.5	89.0
+ mix + RoboTrack-15M	**63.2**	75.7	**86.1**	**84.2**	**91.6**	**94.1**	**68.9**	**80.6**	**92.0**	**57.0**	**68.9**	**88.4**	**63.9**	76.9	**88.1**	**67.5**	**78.7**	**89.8**
From AllTracker (Kubric+mix), the fully trained model																		
Released checkpoint, no fine-tuning	62.6	75.7	**86.9**	81.4	90.1	92.9	68.5	**80.8**	91.7	56.3	**69.0**	**88.8**	64.3	77.1	**89.5**	66.6	78.5	**90.0**
Kubric + RoboTrack-15M	**63.4**	**76.1**	86.2	**83.2**	**90.9**	**93.5**	68.4	80.6	91.3	56.6	**69.0**	87.6	**64.6**	**77.4**	89.1	**67.3**	**78.8**	89.5
Kubric + mix + RoboTrack-15M	63.0	75.9	86.3	82.7	90.8	93.2	**68.9**	**80.8**	**91.9**	**56.8**	**69.0**	88.3	64.1	76.9	88.8	67.1	78.7	89.7

without sacrificing general video performance, which is maintained or slightly improved on both DAVIS and Kinetics.

Gains from RoboTrack-15M transfer to diverse embodiments. Although RoboTrack-15M contains only a simulated Franka arm, its gains on RoboTrack-Bench are not limited to Franka footage. RoboTracker-Warp improves over CoWTracker on every video source, including other robot embodiments and human hands. RoboTracker's gains over AllTracker are smaller and concentrated on robot footage, clearest on DROID and the EgoVerse EVA robot, while it is slightly lower on the human EgoVerse clips. The per-source breakdown and exact numbers are in Appendix E (Figure 13 and Table 8).

4.2 Synthetic Data Ablations

Experimental setup. We study the effect of synthetic training data through ablations on AllTracker [23], which at 17M parameters is about 60× smaller than CoWTracker and makes repeated fine-tuning runs practical. AllTracker is trained in two stages, first on Kubric and then for 400K steps on a mixture of optical-flow and point-tracking datasets (Kubric+mix). Starting from the released Kubric checkpoint, we replace the second stage with 10k fine-tuning steps at a learning rate of 10^{-5}, mixing each candidate dataset equally with Kubric as replay data to preserve general tracking. We compare against a Kubric-only continuation (Table 4) and study the mixing ratio separately below.

Manipulation-focused synthetic data improves tracking on both robotic and general videos. We first ask whether synthetic data aligned with manipulation helps at all, comparing RoboTrack-15M with PointOdyssey, a general-purpose synthetic dataset, and with the Kubric-only control under the same fine-tuning stage (Table 4). Fine-tuning with RoboTrack-15M improves AJ over the control on all five benchmarks, by 1.5 on RoboTrack-Bench, 3.0 on RGB-Stacking, 1.2 on DAVIS, 0.7 on RoboTAP, and 0.6 on Kinetics. These gains are notable because RoboTrack-15M contains only synthetic interactions from a single robot and task, yet transfers to diverse real-world manipulation and general videos. In contrast, fine-tuning with PointOdyssey does not yield comparable improvements, suggesting that the types of motions represented in synthetic training data matter for transfer.

A short fine-tune on RoboTrack-15M can match substantially longer training on existing tracking datasets. We next ask how this short stage compares with AllTracker's full second stage on its mixture of existing tracking datasets. Despite using only 10k second-stage training steps, versus AllTracker's original 400K, our model outperforms the released fully trained AllTracker on RoboTrack-Bench (63.0 vs. 62.6 AJ) and RGB-Stacking (83.6 vs. 81.4), is comparable on Kinetics (56.5 vs. 56.3) and RoboTAP (68.2 vs. 68.5), and is slightly lower on DAVIS (63.6 vs. 64.3). The datasets are also complementary: applying the same fine-tune to the fully trained AllTracker improves RoboTrack-Bench and RGB-Stacking and is comparable on the other three benchmarks, reaching 63.4 AJ on RoboTrack-Bench.

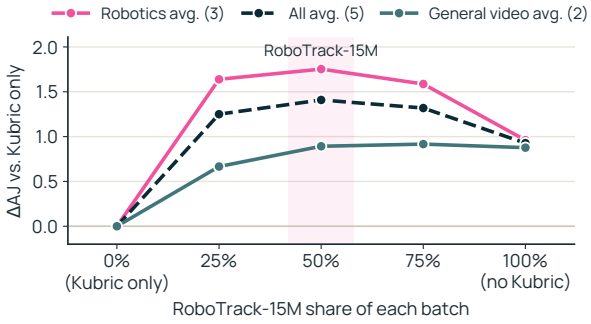


Figure 6 Effect of the RoboTrack-15M share in each batch. ΔAJ is relative to the Kubric-only second stage. We use the 50% mix (shaded), which gives the best average AJ across all benchmarks (67.0, vs. 65.6 for Kubric only).

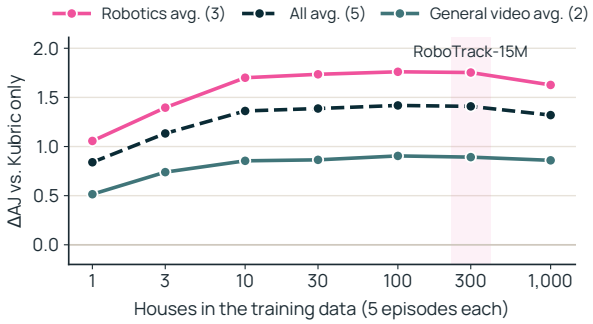


Figure 7 Effect of the number of houses in the training data. ΔAJ is relative to the Kubric-only second stage. The 50% recipe trained on nested subsets of 1 to 1,000 houses (5 episodes each); gains level off at about 10 houses.

A balanced mixture of manipulation data and Kubric gives the best overall performance. We vary the proportion of RoboTrack-15M in each training batch from 0% to 100%, with Kubric making up the remainder (Figure 6). Most of the improvement on robotic benchmarks occurs with the first 25% of manipulation data, increasing average AJ by 1.6 points before performance levels off. Interestingly, training on manipulation data alone reduces the robotics gain to 1.0 AJ, suggesting that mixing with diverse sources such as Kubric is crucial. The 50% mixture achieves the highest average AJ across all five benchmarks (67.0 vs. 65.6 for Kubric alone), which we use to train RoboTracker.

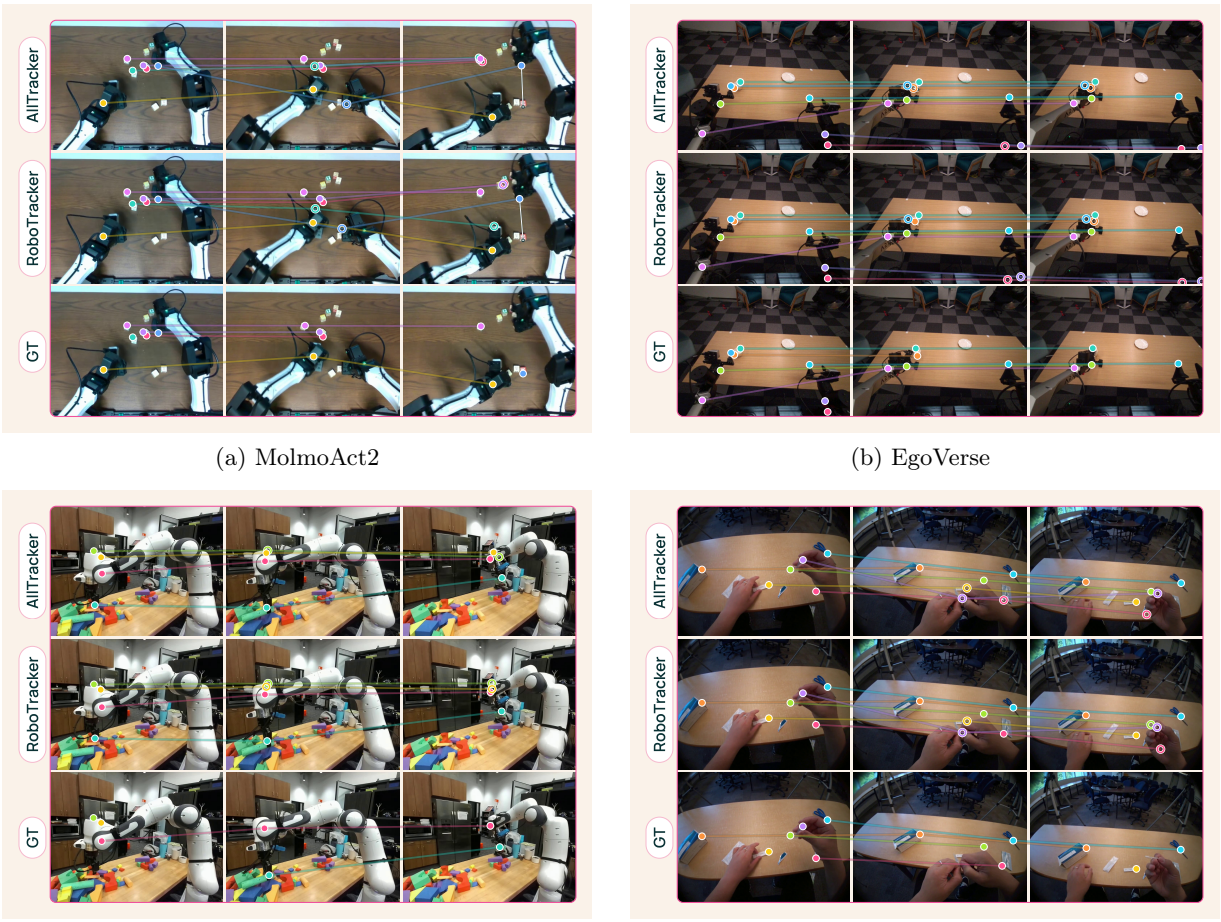
Scaling simulated scenes improves performance with diminishing returns. We next ask how performance changes with the number of scenes in RoboTrack-15M, training the same 50% mixture on nested subsets of 1 to 1,000 houses with five episodes each (Figure 7). Gains grow as houses are added, reaching +1.7 AJ on robotics benchmarks at 10 houses and +1.8 at both 100 and 300 houses, beyond which additional houses have a degrading effect (+1.6 at 1,000). We hypothesize that, because every house contributes the same pick-up task, further gains will come from adding tasks and embodiments rather than more scenes. We therefore use the 300-house set as RoboTrack-15M.

4.3 Qualitative Results

Figure 8 compares AllTracker [23] with RoboTracker on one RoboTrack-Bench clip per source. The benefit of fine-tuning on RoboTrack-15M is clearly visible across all four sources. On MolmoAct2 (a), RoboTracker keeps the points on the correct blocks, including blocks that become occluded. On EgoVerse (b), it places the point on the manipulated cup more precisely. On DROID (c), it correctly predicts which points are occluded, while AllTracker still draws hidden points. On Ego-Exo4D (d), it better localizes the occluded point behind the right hand, drawn in pink.

Figure 14 shows the same comparison between CoWTracker and RoboTracker-Warp. Overall, RoboTracker-Warp stays on the points and predicts their visibility more accurately, especially on the moving robot and hands. On MolmoAct2 (a), it correctly predicts which points on the moving gripper are visible, while CoWTracker marks a visible point on the gripper as occluded and draws a hidden point next to it as visible. On EgoVerse (b), it follows the two points on the blanket as the person pulls it to the edge of the table, while CoWTracker first marks them as occluded and then jumps to the person's other hand. On DROID (c), it stays on the point on the arm as the arm turns and marks the point near the gripper as occluded once it is hidden, while CoWTracker drifts off the first point and keeps drawing the second after it disappears. On Ego-Exo4D (d), both models follow the background points as the camera moves, but RoboTracker-Warp places the point on the object in the person's hands more precisely.

We show failure cases of our models in Appendix F.2.



(a) MolmoAct2 (b) EgoVerse

(c) DROID (d) Ego-Exo4D

Figure 8 AllTracker vs. RoboTracker on RoboTrack-Bench. Hollow points indicate that the model predicts the point is occluded in that frame. Overall, RoboTracker keeps points on the correct objects during manipulation and handles occlusion better.

5 Robotics Application

As a real-world application, we consider wet lab automation, which we solve by using correspondences to align with a demonstration sequence, building on FlowControl [2] and similar to RoboTAP [52]. We chose the common task of pipetting a liquid into a vial; this task is a multi-step sequence that requires high-precision control, in which the robot grasps the pipette, draws liquid from the first vial, dispenses it into the second, and places the pipette back. Figure 9 illustrates our pipeline: at each key frame of the demonstration, the robot tracks points on the target and moves until they match their demonstrated positions, then replays the demonstrated motion.

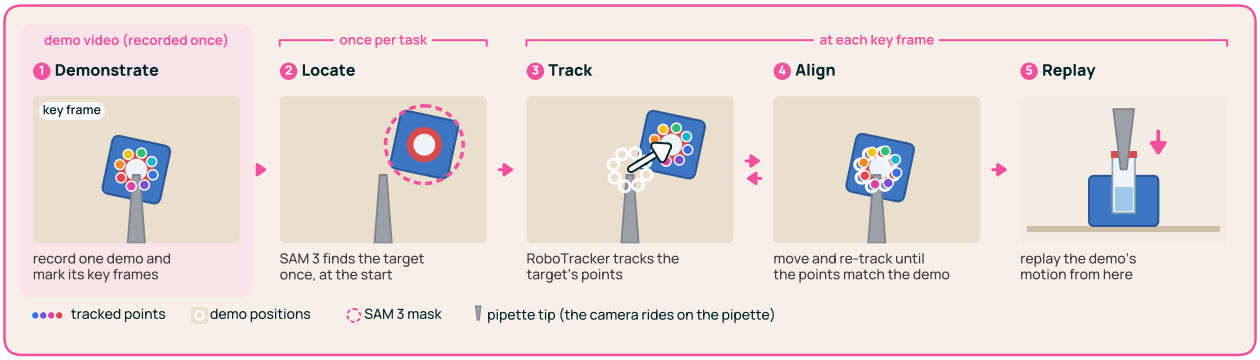


Figure 9 Guiding the robot with point tracks from a single demonstration. We record one demonstration and mark its key frames (1). At test time, SAM3 locates the target once (2). At each key frame, RoboTracker tracks the target's points (3), the robot moves and re-tracks until the points match their demonstrated positions (4), and then replays the demonstrated motion (5). The camera is mounted on the pipette.

Table 5 Real-world pipette transfer. Successes per step over 20 total layouts, one trial per method per layout, with vial positions changed between layouts. Every method grasped and placed the pipette in every trial successfully, so we report only the draw and dispense steps. *Success* counts trials where both succeeded.

	No occlusion			Occlusion		
Method	Draw	Dispense	Success	Draw	Dispense	Success
RAFT (flow)	5/10	5/10	3/10	4/10	2/10	2/10
SAM3 (centroid)	**10/10**	7/10	7/10	4/10	5/10	1/10
RoboTracker (track)	**10/10**	**10/10**	**10/10**	**9/10**	**8/10**	**7/10**

We compare three ways of obtaining the correspondences for this alignment: *RAFT*, dense optical flow [50]. *SAM3 mask centroid*, which segments the target with SAM3 [7] in the live image and aligns centroids. *RoboTracker*, which aligns using point tracks, similar to RoboTAP. We evaluate our results quantitatively in Table 5, and find that our approach is more robust to large displacements compared to optical flow-based correspondences and more robust to occlusion compared to centroid-based alignment. Additional details are provided in Appendix G.

6 Conclusion

We introduce RoboTrack, pairing the real-world benchmark RoboTrack-Bench with RoboTrack-15M, synthetic training data generated in MolmoSpaces. Strong results on RGB-Stacking or RoboTAP do not necessarily transfer to RoboTrack-Bench, suggesting that current evaluations overestimate real-world readiness. Fine-tuning on RoboTrack-15M improves every robotics benchmark for RoboTracker-Warp and RoboTrack-Bench for both models. On a real robot, RoboTracker successfully guides pipette handling in a wet lab. Future work includes better occlusion handling and richer simulated interactions.

References

[1] V. Alumootil, T.-A. Vu, and M. K. Jawed. Dept3r: Joint dense point tracking and 3d reconstruction of dynamic scenes in a single forward pass. *ArXiv*, abs/2512.13122, 2025. URL https://api.semanticscholar.org/CorpusID:283897111.

[2] M. Argus, L. Hermann, J. Long, and T. Brox. Flowcontrol: Optical flow based visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[3] A. Balasingam, J. Chandler, C. Li, Z. Zhang, and H. Balakrishnan. 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*, pages 22488–22497, 2024.

[4] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *ECCV*, 2024.

[5] H. Bharadhwaj, D. Dwibedi, A. Gupta, S. Tulsiani, C. Doersch, T. Xiao, D. Shah, F. Xia, D. Sadigh, and S. Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *CoRL Workshop on X-Embodiment*, 2025.

[6] B. Biggs, T. Roddick, A. Fitzgibbon, and R. Cipolla. Creatures great and smal: Recovering the shape and motion of animals from video. In *Asian Conference on Computer Vision*, pages 3–19. Springer, 2018.

[7] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

[8] S. Cho, J. Huang, J. Nam, H. An, S. Kim, and J.-Y. Lee. Local all-pair correspondence for point tracking. *arXiv preprint arXiv:2407.15420*, 2024.

[9] A. Darkhalil, R. Guerrier, A. W. Harley, and D. Damen. Egopoints: Advancing point tracking for egocentric videos. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8556–8565. IEEE, 2025.

[10] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

[11] I. Demler, S. Chauhan, and G. Gkioxari. Is this tracker on? a benchmark protocol for dynamic tracking. *arXiv preprint arXiv:2510.19819*, 2025. URL https://arxiv.org/abs/2510.19819.

[12] C. Doersch, A. Gupta, L. Markeeva, A. Recasens, L. Smaira, Y. Aytar, J. Carreira, A. Zisserman, and Y. Yang. TAP-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35: 13610–13626, 2022.

[13] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. 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.

[14] C. Doersch, P. Luc, Y. Yang, D. Gokay, S. Koppula, A. Gupta, J. Heyward, I. Rocco, R. Goroshin, J. Carreira, and A. Zisserman. BootsTAP: Bootstrapped training for tracking any point. *arXiv*, 2024.

[15] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golber, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[16] H. Fang, J. Duan, D. Clay, S. Wang, S. Liu, W. Huang, X. Fan, W.-C. Tsai, S. Chen, Y. R. Wang, S. Xing, J. Cho, J. S. Park, A. Eftekhar, P. Sushko, K. Farley, A. Wadhwa, C. Harrison, W. Han, Y.-C. Lee, E. VanderBilt, R. Hendrix, S. Ellawela, L. Ngoo, J. Chai, Z. Ren, A. Farhadi, D. Fox, and R. Krishna. MolmoAct2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026.

[17] H. Feng*, J. Zhang*, Q. Wang, Y. Ye, P. Yu, M. J. Black, T. Darrell, and A. Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[18] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[19] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. 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.

[20] S. Haldar and L. Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. *arXiv preprint arXiv:2502.20391*, 2025.

[21] A. W. Harley, Z. Fang, and K. Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, 2022.

[22] A. W. Harley, Z. Fang, and K. Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, 2022.

[23] A. W. Harley, Y. You, X. Sun, Y. Zheng, N. Raghuraman, Y. Gu, S. Liang, W.-H. Chu, A. Dave, S. You, et al. Alltracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5253–5262, 2025.

[24] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[25] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

[26] W. Huang, Y.-W. Chao, A. Mousavian, M.-Y. Liu, D. Fox, K. Mo, and L. Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026.

[27] A. Hung, B. P. Duisterhof, and J. Ichnowski. 3PoinTr: From human videos to robot policies with 3D point-track plans. *arXiv preprint arXiv:2603.08485*, 2026. URL https://arxiv.org/abs/2603.08485.

[28] S. Jung, A. Zholus, M. Sundermeyer, C. Doersch, R. Goroshin, D. J. Tan, S. Chandar, R. Triebel, and F. Tombari. Tapnext++: What's next for tracking any point (tap)? *arXiv preprint arXiv:2604.10582*, 2026.

[29] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. CoTracker: It is better to track together. 2023.

[30] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

[31] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[32] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, V. Guizilini, D. A. Herrera, M. Heo, K. Hsu, J. Hu, M. Z. Irshad, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[33] Y. Kim, W. Pumacay, O. Rayyan, M. Argus, W. Han, E. VanderBilt, J. Salvador, A. Deshpande, R. Hendrix, S. Jauhri, et al. Molmospaces: A large-scale open ecosystem for robot navigation and manipulation. *arXiv preprint arXiv:2602.11337*, 2026.

[34] S. Koppula, I. Rocco, Y. Yang, J. Heyward, J. Carreira, A. Zisserman, G. Brostow, and C. Doersch. TAPVid-3D: A benchmark for tracking any point in 3D. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2407.05921.

[35] S. Koppula, F. Rajič, A. F. U. Rahman, Y. Yang, I. Rocco, J. Thakwani, R. Kabra, A. Zisserman, J. Carreira, S. Tang, C. Doersch, and G. Brostow. TAPVid-MV: A benchmark for tracking any point in 3D across multiple views. *arXiv preprint arXiv:2609.01899*, 2026. URL https://arxiv.org/abs/2609.01899.

[36] Kuang, Park, K. Fragkiadaki, and S. Tulsiani. Dex4D: Task-agnostic point track policy for sim-to-real dexterous manipulation. *arXiv preprint arXiv:2602.15828*, 2026.

[37] Z. Lai, E. Insafutdinov, E. Sucar, and A. Vedaldi. Cowtracker: Tracking by warping instead of correlation. *arXiv preprint arXiv:2602.04877*, 2026.

[38] A. X. Lee, C. M. Devin, Y. Zhou, T. Lampe, K. Bousmalis, J. T. Springenberg, A. Byravan, A. Abdolmaleki, N. Gileadi, D. Khosid, et al. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In *5th Annual Conference on Robot Learning*, 2021.

[39] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21:1281 – 1289, 2018. URL https://api.semanticscholar.org/CorpusID:4748395.

[40] A. Mathis, T. Biasi, S. Schneider, M. Yuksekgonul, B. Rogers, M. Bethge, and M. W. Mathis. Pretraining boosts out-of-domain robustness for pose estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1859–1868, 2021.

[41] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. 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.

[42] C. I. Nwoye, K. Elgohary, A. Srinivas, F. Zaid, J. L. Lavanchy, and N. Padoy. Cholectrack20: A multi-perspective tracking dataset for surgical tools. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025.

[43] T. D. Pereira, N. Tabris, A. Matsliah, D. M. Turner, J. Li, S. Ravindranath, E. S. Papadoyannis, E. Normand, D. S. Deutsch, Z. Y. Wang, G. C. McKenzie-Smith, C. C. Mitelut, M. D. Castro, J. D'Uva, M. Kislin, D. H. Sanes, S. D. Kocher, S. S-H, A. L. Falkner, J. W. Shaevitz, and M. Murthy. Sleap: A deep learning system for multi-animal pose tracking. *Nature Methods*, 19(4), 2022.

[44] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[45] R. Punamiya, S. Kareer, Z. Liu, J. Citron, R.-Z. Qiu, X. Cai, A. Gavryushin, J. Chen, D. Liconti, L. Y. Zhu, P. Aphiwetsa, B. Li, A. Cheluva, P. Kuppili, Y. Liu, D. Patel, A. Gao, H.-Y. Chung, R. Co, R. Zbizika, J. Liu, X. Xu, H. Xiong, G. Chen, S. Oliani, W. Xuan, C. Yang, X. Wang, J. Fort, R. Newcombe, J. Gao, J. Chong, G. Matsuda, A. Doriwala, M. Pollefeys, R. Katzschmann, X. Wang, S. Song, J. Hoffman, and D. Xu. EgoVerse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

[46] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[47] P. Sand and S. J. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80:72–91, 2006. URL https://api.semanticscholar.org/CorpusID:7697706.

[48] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. *ArXiv*, abs/1610.04889, 2016. URL https://api.semanticscholar.org/CorpusID:16440200.

[49] R. Sundararaman, C. De Almeida Braga, E. Marchand, and J. Pettre. Tracking pedestrian heads in dense crowd. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3865–3875, 2021.

[50] Z. Teed and J. 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*, 2020.

[51] TODO. Real-world point tracking with verifier-guided pseudo-labeling. *arXiv preprint arXiv:2603.12217*, 2026.

[52] M. Vecerik, C. Doersch, Y. Yang, T. Davchev, Y. Aytar, G. Zhou, R. Hadsell, L. Agapito, and J. Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5397–5403. IEEE, 2024.

[53] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[54] X. Wang, S. Qi, J. Zhao, H. Zhou, S. Zhang, G. Wang, K. Tu, S. Guo, J. Zhao, J. Li, and M. Yang. Mctrack: A unified 3d multi-object tracking framework for autonomous driving. *2025 IEEE/RSJ International Conference*

on Intelligent Robots and Systems (IROS), pages 4551–4558, 2024. URL https://api.semanticscholar.org/CorpusID:272832346.

[55] Y. Wang, L. Lipson, and J. Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. *arXiv preprint arXiv:2405.14793*, 2024.

[56] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

[57] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. Flow as the cross-domain manipulation interface. In *Conference on Robot Learning*, 2024.

[58] T. Yamane, R. Masumura, S. Suzuki, and S. Orihashi. Mvtrajecter: Multi-view pedestrian tracking with trajectory motion cost and trajectory appearance cost. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13270–13280, 2025. URL https://api.semanticscholar.org/CorpusID:281081091.

[59] B. Zhang, L. Ke, A. W. Harley, and K. Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. *arXiv preprint arXiv:2504.14717*, 2025.

[60] G. Zhang, W. Huang, S. Shailesh, Y. Peng, J. Duan, and R. Krishna. Grounded action model: 3d grounding as a foundation for robotics. *arXiv preprint arXiv:2609.23863*, 2026.

[61] J. Zhang, C. Zheng, Y. Yang, M. Argus, R. Soraki, W. Han, T. Anderson, C.-L. Li, S. Liu, J. Duan, et al. Molmomotion: Forecasting point trajectories in 3d with language instruction. *arXiv preprint arXiv:2606.18558*, 2026.

[62] W. Zhao, H. Xu, X. Miao, Q. Zhao, R. Zhang, K. Huang, N. Gao, P. Cao, M. Sun, M. Yu, T. Lu, L. Xu, J. Dong, and J. Pang. SynthVerse: A large-scale diverse synthetic dataset for point tracking. In *ACM SIGGRAPH 2026 Conference Papers*, 2026. doi: 10.1145/3799902.3811183. URL https://doi.org/10.1145/3799902.3811183.

[63] Y. Zheng, A. W. Harley, B. Shen, G. Wetzstein, and L. 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.

[64] A. Zholus, C. Doersch, Y. Yang, S. Koppula, V. Patraucean, X. O. He, I. Rocco, M. S. Sajjadi, S. Chandar, and R. Goroshin. Tapnext: Tracking any point (tap) as next token prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9693–9703, 2025.

Appendix

The appendix includes the following sections:

- §A - RoboTrack-Bench Statistics
- §B - RoboTrack-Bench Annotation Details
- §C - RoboTrack-15M details
- §D - Training CoWTracker
- §E - Results by Video Source
- §F - Qualitative Examples
- §G - Real-World Robotics Experiments
- §H - Limitations
- §I - Broader Impact
- §J - Licenses

A RoboTrack-Bench Statistics

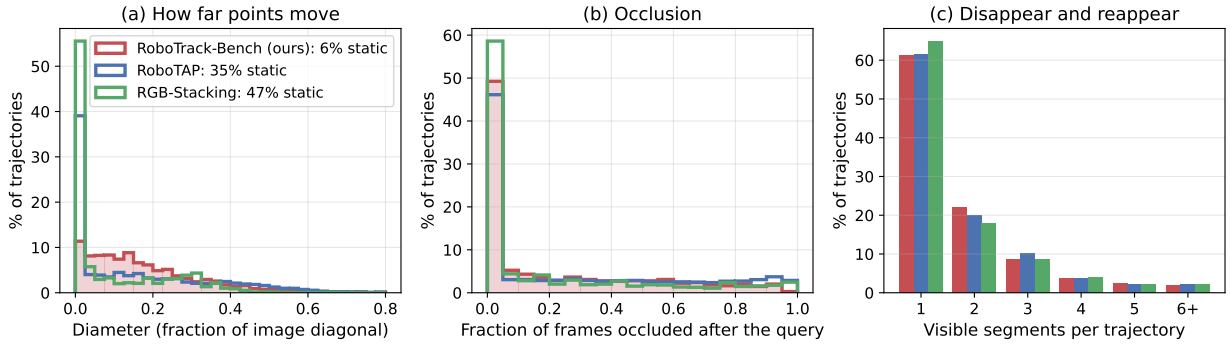
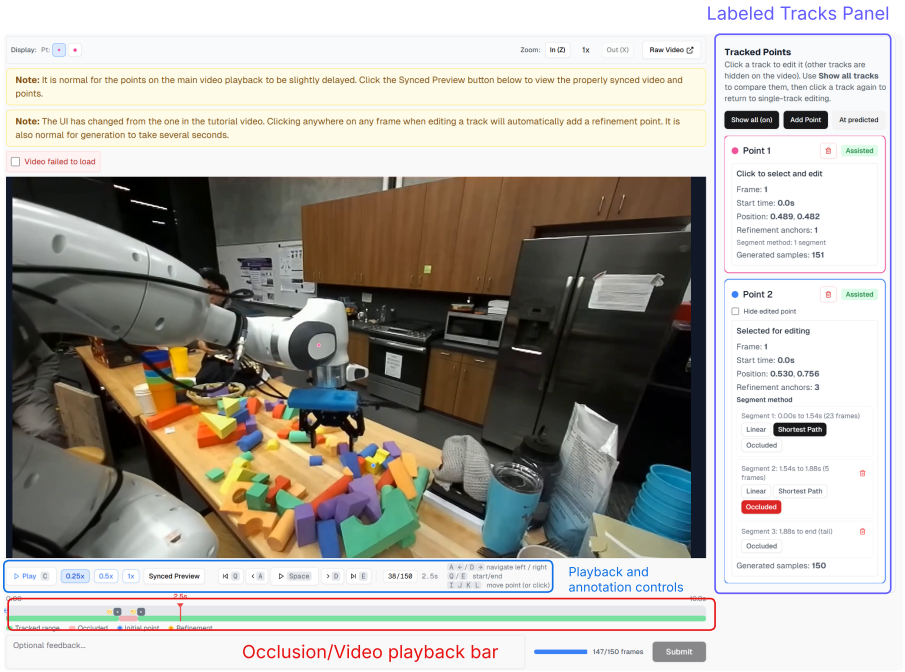


Figure 10 Track statistics of RoboTrack-Bench compared with RoboTAP and RGB-Stacking. (a) Trajectory diameter, the maximum distance between the positions of a point over time, as a fraction of the image diagonal: only 6% of points in RoboTrack-Bench stay static (moving less than 1% of the diagonal), compared with 35% in RoboTAP and 47% in RGB-Stacking. (b) Fraction of frames in which a point is occluded after its query frame. (c) Number of visible segments per trajectory, i.e., contiguous visible sections separated by occlusions; 39% of points in RoboTrack-Bench disappear and reappear at least once. Each benchmark is normalized by its own number of trajectories.

Figure 10 compares the track statistics of RoboTrack-Bench with robotics tracking benchmarks such as RoboTAP and RGB-Stacking. Clips in RoboTrack-Bench average 223 frames (26 to 785) and 8.2 tracks. 79% of tracks are queried on the first frame, with the rest queried when the point first becomes visible. 43% of tracks stay visible for their whole length, a track is occluded for 22% of its frames on average, and 39% of tracks disappear and reappear at least once. The median trajectory diameter is 14% of the image diagonal, in line with our aim of capturing intricate arm movements rather than large camera sweeps. Figure 3 shows qualitative examples from RoboTrack-Bench. Overall, we observe that point tracks are accurate and consistent across videos. We find that annotations with fewer refinement points are often smoother, indicating that the optical flow tracking assistant is very helpful for obtaining high-quality point tracks.



Annotation Instructions

1. Watch the video to understand what happens and which objects move.
2. **Select a point** that you want to track, beginning with the first frame it appears
3. Wait for the tracking assistant to generate a track
4. Use A/D or the arrow keys to **review the generated track frame by frame**.
5. **Refine the track** by adding new points whenever the generated track drifts or snaps to the wrong location.
6. If the point **becomes occluded**, **mark the occluded section**.
7. **Review the generated track carefully** before moving on.
8. Repeat this process until you have created **5 different point tracks**.

Figure 11 Point annotation interface and annotation instructions for RoboTrack-Bench. The interface consists of two main sections: the video on the left, with frame-by-frame seeking, and the labeled tracks on the right.

B RoboTrack-Bench Annotation Details

B.1 Track assist algorithm.

Our track assist algorithm closely follows TAP-Vid [12], with the main difference being the optical flow model. For every video, we compute optical flow using SEA-RAFT [55], which is used to generate assisted tracks for the annotator. The assisted tracks are usually accurate over a few frames, but cannot handle occlusions and tend to drift over time; annotators are responsible for marking occlusions and adding refinement points when the track drifts.

When an annotator adds a point p_s on frame s, we use optical flow to propagate that point from one frame to the next for the entire video. When the annotator chooses a second point p_t on frame t, we use Dijkstra's algorithm to find the shortest path through the optical flow from p_s to p_t. Annotators are also always given the option to linearly interpolate between the two points instead, which we found helpful for certain points.

B.2 Annotation Interface

Figure 11 shows the annotation interface and the instructions given to annotators for labeling RoboTrack-Bench (Section 3.1). The left side is the main video display, where annotators click on the video to add points and use the playback controls to seek through the video. The right side lists the currently tracked points and relevant information for each. Each point's track is divided into time segments by the refinement points the annotator selected; for each segment, the annotator can choose an interpolation method or mark the segment as occluded.

C RoboTrack-15M Details

We provide additional statistics of RoboTrack-15M and describe the strategies we use to create the final point tracks used for training.

Dataset Statistics Table 6 summarizes the composition of RoboTrack-15M. Figure 12 further reports the distribution of track-level statistics in RoboTrack-15M, log scaled.

Obtaining point tracks. We now describe in more detail how point tracks are obtained from the MolmoSpaces simulation (Section 3.2). Given a pixel (u, v) in the query frame with rendered depth d, we unproject it to world coordinates:

$$\mathbf{p}^{\text{world}} = \mathbf{R}_{\text{cam}} \begin{pmatrix} (u - c_x)\, d/f \\ (v - c_y)\, d/f \\ d \end{pmatrix} + \mathbf{t}_{\text{cam}}, \tag{1}$$

where f is the focal length, (c_x, c_y) is the principal point (image center), and $\mathbf{R}_{\text{cam}}$ and $\mathbf{t}_{\text{cam}}$ are the camera's rotation and position in world coordinates. We then express the point in the local frame of its parent body, with rotation $\mathbf{R}_{\text{body}}$ and position $\mathbf{t}_{\text{body}}$:

$$\mathbf{p}^{\text{local}} = \mathbf{R}_{\text{body}}^{\top} \left(\mathbf{p}^{\text{world}} - \mathbf{t}_{\text{body}} \right). \tag{2}$$

At each subsequent frame t, the world position is recovered from the body pose as $\mathbf{p}_t^{\text{world}} = \mathbf{R}_{\text{body},t}\, \mathbf{p}^{\text{local}} + \mathbf{t}_{\text{body},t}$, transformed into the camera frame as $\mathbf{p}_t^{\text{cam}} = \mathbf{R}_{\text{cam},t}^{\top} \left(\mathbf{p}_t^{\text{world}} - \mathbf{t}_{\text{cam},t} \right)$, and projected with the pinhole model:

$$u_t = f \frac{p_{t,x}^{\text{cam}}}{p_{t,z}^{\text{cam}}} + c_x, \qquad v_t = f \frac{p_{t,y}^{\text{cam}}}{p_{t,z}^{\text{cam}}} + c_y. \tag{3}$$

Visibility is determined by comparing the projected depth against the rendered depth buffer D_t:

$$\text{visible}_t = \mathbf{1}\left[\, p_{t,z}^{\text{cam}} - D_t(u_t, v_t) < \epsilon \,\right], \tag{4}$$

where $\epsilon = 0.03\,\text{m}$.

Table 6 Statistics of RoboTrack-15M. Every episode is recorded by an exo and a wrist camera. Static tracks move less than one pixel (at 256-pixel image height) from their first visible position over the whole clip. A point counts as not visible when it is occluded or outside the frame.

Houses / episodes	300 / 1,500
Videos	3,000 (1,500 per camera)
Tracks	15M (5,000 per video)
Frames / video	70 on average (42–177)
Resolution	624×352, 15 fps

	Exo	Wrist	All
Tracks on the robot	82.1%	69.6%	75.8%
Tracks on objects	17.9%	30.4%	24.2%
Queried on the first frame	61.8%	77.6%	69.7%
Static tracks	16.7%	12.6%	14.7%
Not visible at least once	13.5%	16.5%	15.0%

D Training CoWTracker

RoboTracker-Warp is CoWTracker [37] fine-tuned for 10k steps at learning rate 10^{-5} on an equal mix of Kubric and RoboTrack-15M. CoWTracker's training code wasn't released, so we trained using our own re-implementation. To separate the effect of RoboTrack-15M from the effect of the additional training, we fine-tune the released CoWTracker with the same recipe on Kubric alone (Table 7). Compared with this control, RoboTrack-15M adds 1.2 AJ on RoboTrack-Bench and 0.5 AJ on RoboTAP and on the robotics average, with small gains on Kinetics (+0.1) and DAVIS (+0.3) and a small loss on RGB-Stacking (−0.3). The remaining 0.9 AJ of the 2.1 AJ gain over the released model on RoboTrack-Bench comes from the additional Kubric training. The lower occlusion accuracy on DAVIS relative to the released model also appears in the control, so it stems from the fine-tuning recipe rather than from RoboTrack-15M.

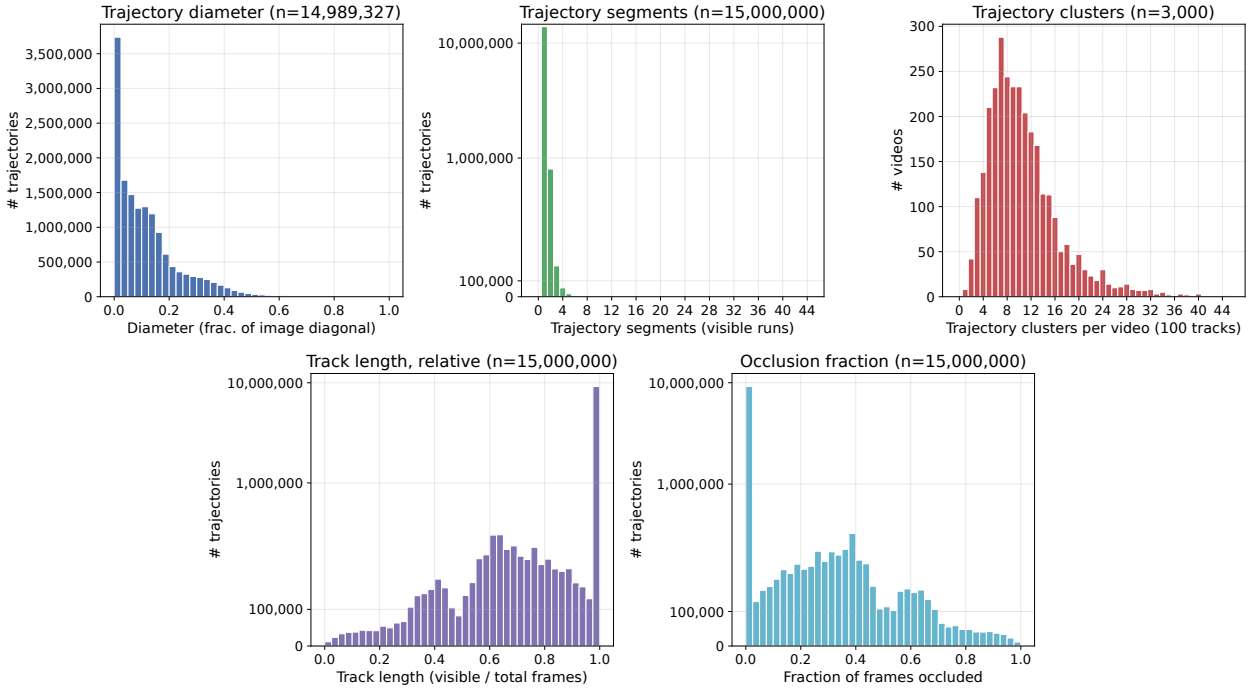


Figure 12 Statistics of tracks in RoboTrack-15M. "Diameter" refers to the maximum distance between the positions of a point over time. Trajectory segments refer to the number of contiguous visible sections of a trajectory, with breaks due to occlusion. Trajectory clusters count the number of motion-distinct groups per video; higher values mean more diverse motion.

Table 7 Kubric-only control for RoboTracker-Warp. All three are models using the CoWTracker architecture. Avg. is the mean over the three robotics benchmarks. Evaluation as in table 3. Bold marks the best value in each column.

	RoboTrack-Bench			RGB-Stack			RoboTAP			Avg.			Kinetics			DAVIS		
Training data	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA
Released CoWTracker (Kubric)	63.9	76.1	88.5	85.5	93.0	**94.8**	70.5	81.2	93.9	73.3	83.4	92.4	57.6	69.9	**90.0**	**66.1**	78.4	**91.2**
+ Kubric only (control)	64.8	77.1	88.1	**86.2**	93.0	94.6	71.1	81.6	93.9	74.0	83.9	92.2	57.9	70.2	89.4	65.6	78.6	89.8
+ Kubric + RoboTrack-15M (RoboTracker-Warp)	**66.0**	**77.9**	**89.0**	85.9	**93.2**	94.5	**71.6**	**81.9**	**94.2**	**74.5**	**84.4**	**92.6**	**58.0**	**70.3**	89.7	65.9	**78.9**	89.9

E Results by Video Source

Table 8 breaks down RoboTrack-Bench results by video source, splitting MolmoAct2 into its YAM and SO-100 clips and EgoVerse into its robot and human clips. Human manipulation is the hardest setting: every model scores its lowest AJ on either Ego-Exo4D or the human EgoVerse clips. The gains from RoboTrack-15M, which contains only robot manipulation, are concentrated on robot videos. RoboTracker-Warp improves over CoWTracker on every source (1.0 to 3.0 AJ) and is the best model on all sources except the human EgoVerse clips, where AllTracker (Kubric+mix) leads. RoboTracker improves over its starting model, AllTracker (Kubric+mix), on DROID, the YAM clips, and the EgoVerse robot clips (+0.8 to +1.2 AJ) and on Ego-Exo4D (+0.7), is comparable on the SO-100 clips, and is 0.5 AJ lower on the human EgoVerse clips. Rankings among off-the-shelf models also change between sources: BootsTAPNext slightly outperforms CoTracker3 (online) on DROID but falls 11.9 AJ behind AllTracker (Kubric) on the human EgoVerse clips. Figure 13 shows the same gains as mean per-clip changes with 95% confidence intervals. Because every subset other than DROID contains only 24 to 42 clips, small per-source differences should be interpreted with caution.

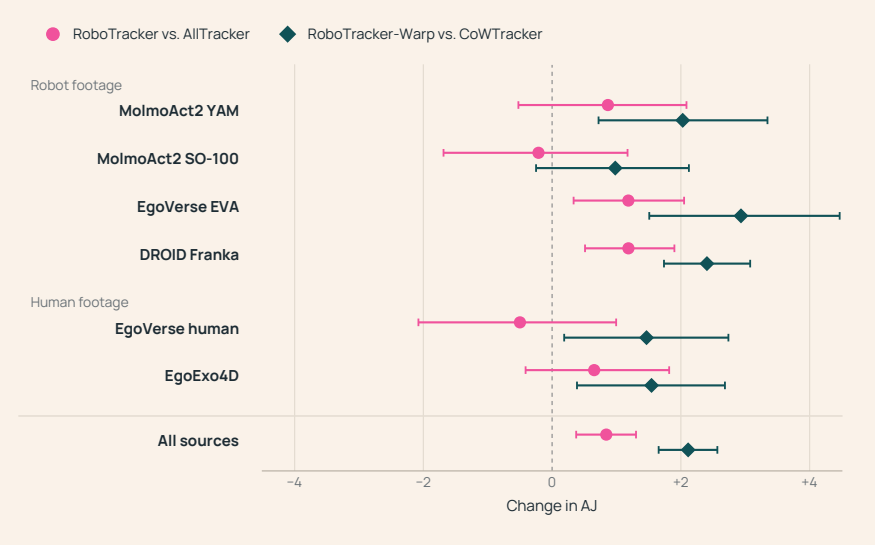


Figure 13 Gains on RoboTrack-Bench by video source. Mean per-clip change in AJ of each of our models over the model it was fine-tuned from, with 95% confidence intervals (paired bootstrap over clips). Per-source scores for all models are in Table 8.

Table 8 Results on RoboTrack-Bench by video source. Each source's score is the mean over its clips, the same aggregation as the overall score (All); clip counts in parentheses. Evaluation as in Table 3. Bold marks the best and underline the second-best value in each column; our models are shaded.

		DROID			MolmoAct2						EgoVerse						Ego-Exo4D			All		
		Franka (167)			YAM (24)			SO-100 (24)			EVA (25)			Human (25)			Human (42)			All sources (307)		
Model	Train. Data	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA	AJ	δ_{vis}	OA
CoTracker3 (offline)	Kubric	60.0	73.6	84.4	68.3	80.0	89.6	58.1	73.2	85.6	61.8	77.2	86.0	58.3	72.4	82.4	54.7	66.1	82.5	59.8	73.2	84.6
CoTracker3 (offline)	Kubric+15k	60.1	72.6	86.0	66.8	77.8	88.5	57.8	72.3	86.1	63.1	75.6	88.6	61.0	74.0	85.3	56.2	67.0	82.2	60.2	72.6	85.8
CoTracker3 (online)	Kubric+15k	61.8	74.8	86.9	68.3	81.0	90.3	59.2	74.8	87.4	64.5	79.1	87.6	62.7	**75.6**	**89.2**	57.8	68.9	85.9	61.8	74.9	87.3
TAPNext	Kubric	56.6	71.8	87.7	59.1	73.2	90.6	53.9	69.2	88.6	55.1	71.0	87.5	48.0	62.1	82.3	51.3	63.7	84.9	55.1	69.7	87.2
BootsTAPNext	Kubric+15M	62.3	75.9	88.7	64.0	76.6	89.1	59.0	72.5	89.0	59.4	74.3	86.5	50.9	64.3	80.4	56.0	67.6	83.3	60.2	73.5	87.1
AllTracker	Kubric	60.2	73.9	82.4	68.7	79.9	89.0	61.1	74.2	86.0	66.7	79.5	88.2	62.8	75.3	87.1	58.1	70.5	81.2	61.4	74.5	83.9
AllTracker	Kubric+mix	62.0	75.6	86.0	68.9	80.4	91.1	61.4	76.0	87.2	66.2	79.2	89.3	**62.9**	**75.6**	88.8	59.6	71.1	84.9	62.6	75.7	86.9
CoWTracker	Kubric	64.4	77.0	88.2	69.2	80.2	92.1	63.2	75.2	**89.4**	66.4	79.2	**90.8**	60.9	73.7	87.7	59.2	70.6	**86.0**	63.9	76.1	88.5
RoboTracker	Kubric+mix+RoboTrack-15M	63.2	76.3	85.8	69.7	80.4	91.1	61.2	75.3	85.8	67.4	79.9	89.0	62.4	75.4	86.8	60.3	71.8	83.2	63.4	76.1	86.2
RoboTracker-Warp	Kubric+RoboTrack-15M	**66.8**	**78.8**	**89.7**	**71.3**	**81.9**	**92.3**	**64.2**	**76.7**	89.0	**69.4**	**82.3**	90.3	62.4	74.9	87.1	**60.8**	**72.1**	84.8	**66.0**	**77.9**	**89.0**

F Qualitative Examples

F.1 RoboTracker-Warp Predictions

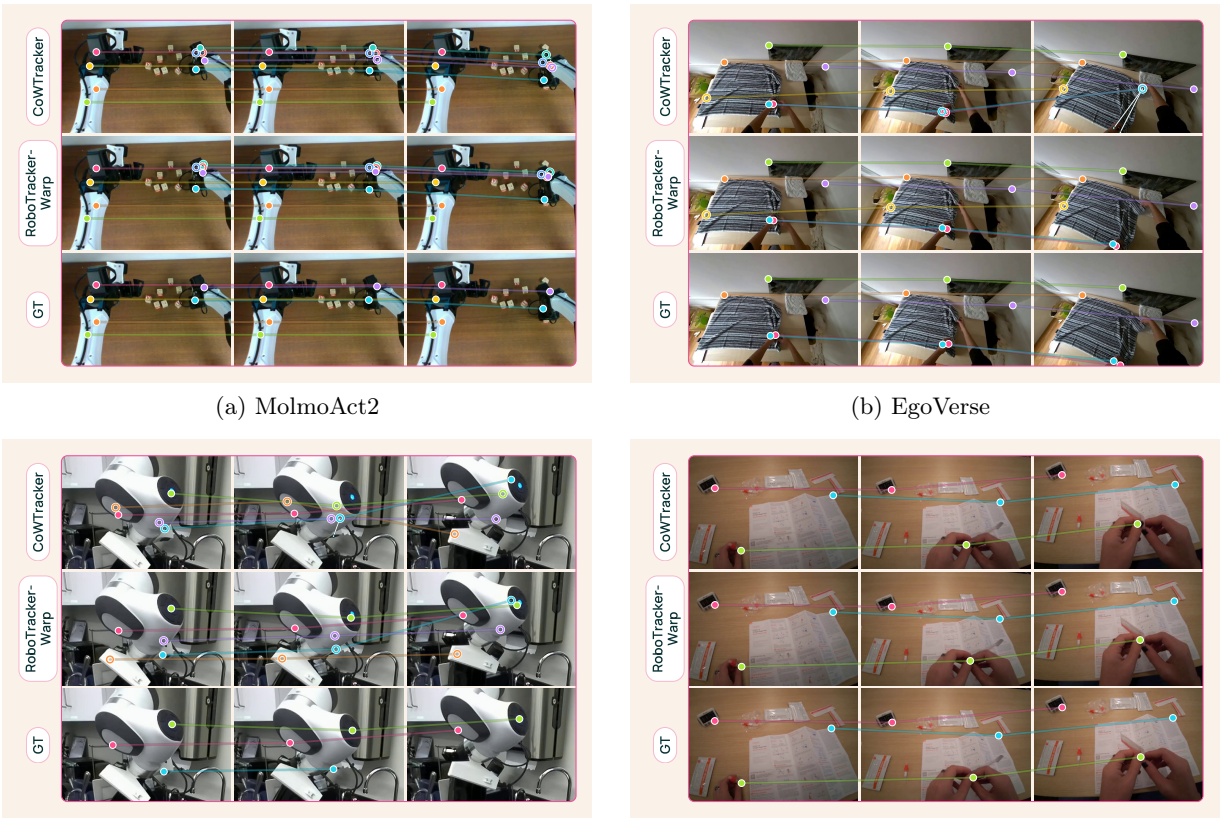
Figure 14 compares RoboTracker-Warp with the released CoWTracker on RoboTrack-Bench clips where fine-tuning on RoboTrack-15M gives a large gain, one from each source.

F.2 Failure Cases

Figure 15 shows clips where both RoboTracker and RoboTracker-Warp fail, one from each source. In each, a point both models track correctly is lost once the object moves quickly, turns, or is covered by a hand or gripper.

G Real-World Robotics Experiments

Setup. We use one arm of a bimanual I2RT YAM robot with a parallel gripper. Two Intel RealSense D405 RGB-D cameras provide observations at 848 × 480 and 15 fps: one on the wrist, and one mounted on the pipette looking down along its tip. The pipette is an electronic pipette that the robot picks up from a holder



(a) MolmoAct2 (b) EgoVerse

(c) DROID (d) Ego-Exo4D

Figure 14 CoWTracker vs. RoboTracker-Warp on RoboTrack-Bench. Each panel shows CoWTracker, RoboTracker-Warp, and the ground truth (GT) in rows, and three frames in temporal order in columns. Filled dots are visible points and hollow points are predicted to be occluded; each color is one track. CoWTracker loses points as the robot, hands, and objects move, while RoboTracker-Warp stays on them.

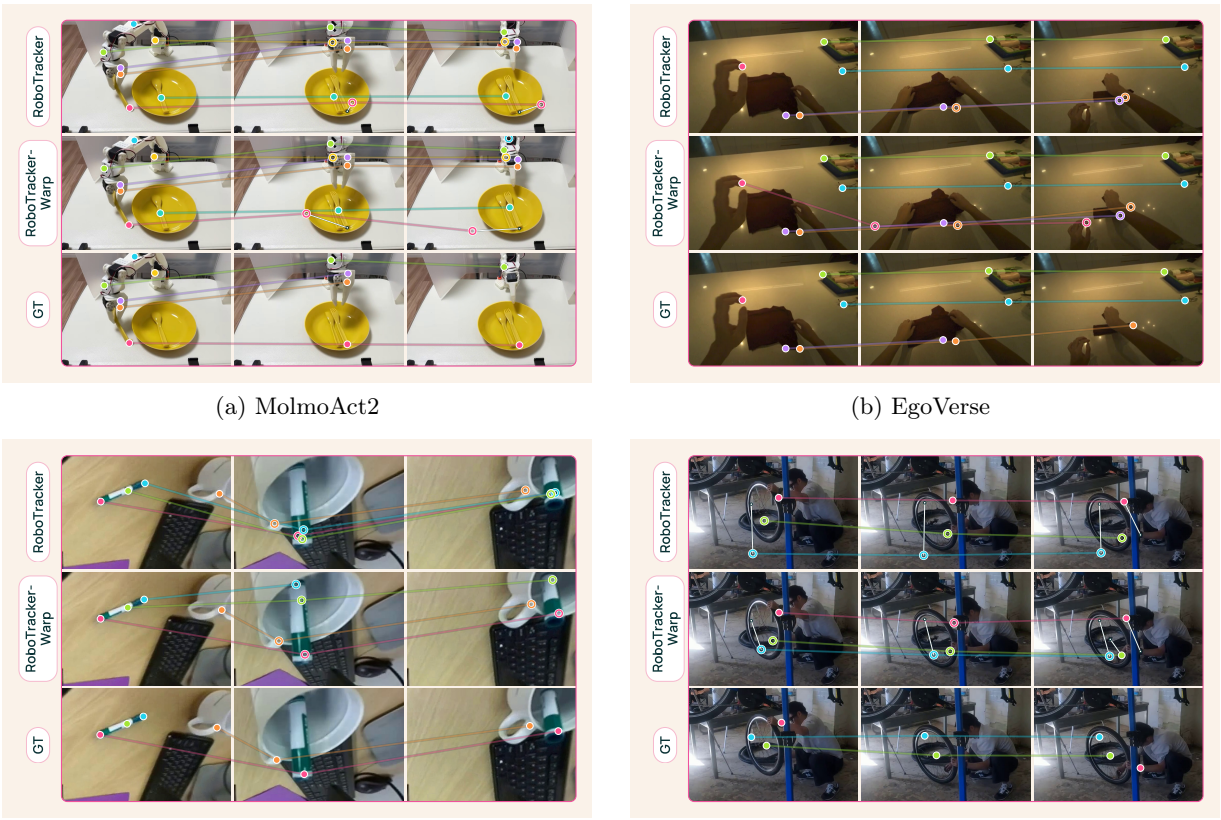
and that is triggered over the network.

Task. The robot picks up the pipette, transfers liquid between two vials, and places the pipette back on the holder. The positions of the pipette and vials change between trials. The task has four vision-guided steps: grasp the pipette, move to vial 1 and aspirate, move to vial 2 and dispense, and return the pipette to its holder. Steps 1 and 4 are guided by the wrist camera while steps 2 and 3 are guided by the pipette camera. A trial counts as a task success only if all four steps succeed.

One-shot imitation by alignment. Similar to RoboTAP [52], the robot repeats a single demonstration. The demonstration is recorded once by teleoperation and divided into steps, each with a keyframe image and a mask of the object that the step acts on (the pipette holder, or the colored ring around a vial). At test time, for each step the robot moves to the keyframe pose, estimates how far the object has moved relative to the demonstration by comparing the keyframe with the live image, shifts the end-effector by that offset, and then replays the demonstrated motion for the step. Since the replay is open-loop, the accuracy of the estimated offset determines whether the pipette tip enters the vial.

Alignment methods. We compare three ways of estimating the offset. All share the same demonstration, controller, and replay, and all are given the same budget of one coarse move followed by three corrections.

- **RAFT (flow)**: optical flow [50] from the keyframe to the live image is read at the object's mask pixels, lifted to 3D with depth, and a rigid motion is fit with RANSAC.



(a) MolmoAct2 (b) EgoVerse

(c) DROID (d) Ego-Exo4D

Figure 15 Failure cases shared by RoboTracker and RoboTracker-Warp. Each panel shows RoboTracker, RoboTracker-Warp, and the ground truth (GT) in rows, and three frames in temporal order in columns: before, when, and after both models lose a point. Filled dots are visible points and hollow points are predicted to be occluded; each color is one track. Both models lose points on small or fast-moving objects and on hands, such as the cutlery lifted by the SO-100 arm (a), a cloth handled by both hands (b), a marker placed into a cup (c), and a turning bicycle wheel (d).

- **SAM3 (centroid)**: the object is segmented in the live image from a text prompt with SAM3 [7], and the end-effector moves so that the centroid of the mask matches the centroid in the demonstration. Every correction uses a new detection.
- **RoboTracker (track)**: the same fit as for flow, with correspondences from our tracker. Each vial is segmented once with SAM3 when the pipette is picked up, and the tracker carries the points forward through the rest of the task. At each correction, the tracker receives the reference frame, up to 14 frames of the video since then, and the live frame, and matches the points of the demonstration mask to the live image. Points that the tracker predicts as occluded are kept in the fit, so the tracker can align when the target is partially hidden.

Protocol. We evaluate each method on 20 layouts, 10 with occlusion and 10 without (Table 5). The locations of vials were randomly shifted between layouts. Occlusions occurred in the middle of the task to test each method's ability to track under intermittent occlusion. Every method completed the grasp and place steps in all trials, so the comparison focuses on the two vial alignment steps.

Results and failure analysis. The three approaches fail in different ways, and the differences show what tracking provides.

Optical flow estimates correspondences directly from the demonstration keyframe to the live image. This works well when the vials are close to their positions in the demonstration, but fails when they are further away. As a result, flow only succeeds on half of the vial alignments even with no occlusions.

Centroid alignment using SAM3 [7] masks is more robust towards displacements, able to guide the robot through both large and small motions. However, since every look depends on a fresh detection with no previous knowledge, SAM3 might become confused by a duplicate or similar object, which is what caused the failures under the no occlusion case. Under occlusion, SAM3 might not find the object, or even if it does, the difference in the mask's shape causes centroid alignment to fail. These factors together cause SAM3 to get a score of 7 under no occlusion and 1 with occlusions.

Tracking identifies the target once and tracks it throughout the video. This allows the method to handle large displacements and track the target accurately under occlusion, mitigating the weaknesses of both previous methods. This lets tracking achieve strong performance in both no occlusion and occlusion, scoring 10 and 7 respectively. Tracking still fails under extended periods of occlusion or when the target is completely occluded because tracking accuracy decreases under occlusion.

H Limitations

RoboTrack-Bench contains 307 clips with 2,522 annotated tracks, which is modest in scale. Acquiring more annotations is costly because each track requires careful human verification, and ensuring trajectory correctness through occlusions remains inherently ambiguous even for annotators. Although the clips span several sources and embodiments, they do not cover all manipulation skills; for example, tasks involving deformable objects or tool use are underrepresented. We fine-tune two trackers, AllTracker and CoWTracker. Although these are the two strongest architecture families, training on more model types would let us be more confident in RoboTrack-15M's benefit. Additionally, CoWTracker did not release its training code, so we re-implemented our version and cannot say for certain that behavior is the exact same. Other recent trackers do not release training pipelines or use bootstrapped training procedures that are difficult to replicate, so we cannot measure how much RoboTrack-15M would benefit those models.

Despite improvements from RoboTrack-15M, important challenges remain. On the simulation side, RoboTrack-15M improves sim-to-real transfer but does not close the gap entirely, suggesting that scene diversity, rendering realism, or motion distribution in our synthetic data still diverge from real manipulation in ways we have not fully characterized. Our qualitative results show that models still struggle with long occlusions, reappearance after occlusion, and small objects that can be confused with the background. These limitations suggest promising directions for future work, including improved occlusion reasoning, higher-resolution tracking for small objects, and richer simulation data with more complex object interactions. We hope that RoboTrack-Bench will support future progress toward reliable point tracking for real-world robotic manipulation.

I Broader Impact

Point tracking is a general-purpose motion estimation primitive with applications well beyond robotic manipulation. On the positive side, more reliable point trackers can improve assistive and household robotics, where accurate motion estimation is needed for safe interaction with people and objects. The same capability is useful in ecological research, for instance tracking animal movement in wildlife footage, and in medical settings such as surgical video analysis, where instrument and tissue motion must be followed precisely.

On the negative side, improvements in point tracking could be repurposed for surveillance, enabling more robust tracking of individuals across camera views or in crowded environments without their consent. More capable tracking models could also be integrated into autonomous weapons systems, where precise motion estimation of targets is operationally relevant. We note that our contribution is a benchmark and training data rather than a deployed tracking system, which limits its direct potential for misuse, but acknowledge that downstream improvements enabled by this work could be applied in such contexts.

J Licenses

DROID [32] is released under CC-BY 4.0. Ego-Exo4D [18] is released under the Ego-Exo4D license agreement. EgoVerse is released under CC-BY-SA 4.0. AllTracker is released under MIT. Kubric [19] is released under

Apache 2.0. MolmoSpaces [33] data is released under CC-BY 4.0 (with Objaverse subsets under ODC-BY 1.0) and code under Apache 2.0. CoTracker3 [30] is released under CC-BY-NC 4.0. CoWTracker weights are released under CC-BY-NC 4.0. TAP-Vid [12] annotations, RGB-Stacking videos, and RoboTAP [52] videos are released under CC-BY. The PointOdyssey [63] dataset is released under CC-BY-NC-SA 4.0. We respect all license terms in our use of these assets.