# NUSCTRACK: A SURROUND-VIEW BENCHMARK FOR 3D TRACK-ANY-POINT IN AUTONOMOUS DRIVING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Track-Any-Point (TAP) follows a queried pixel through a video with occlusion labels. On a surround-view driving rig, the same physical point often leaves the query camera's field of view and must be recovered in another camera, even when it lies on unlabeled background that has no 3D box. Existing autonomous driving Tracking benchmarks do not measure that setting: they use one video stream or indoor cameras with large overlap, and driving TAP has been mostly monocular and instance-centric. We introduce **NuscTrack**, a surround-view benchmark for cross-view 3D TAP on nuScenes: a queried pixel in one camera maps to a metric ego-frame trajectory and per-camera visibility. Instance-surface points follow the rigid motion of their 3D boxes, static background points remain fixed in the world frame, and visibility is determined using completed LiDAR depth. The processed pool has $15.5$M / $3.0$M unique 3D tracks (train / val) after length, visibility, and range filters; $78\%$ are visible in at least two cameras, covering both instance surfaces and unlabeled background. We evaluate all methods on the same image queries, reporting 3D-AJ, APD, MTE, and OA split by All / instance / background, without fitting scale to ground truth. Models trained on monocular TAP data or on indoor multi-view data do not transfer to this setting: they remain at single-digit 3D-AJ with multi-metre error. Training the same multi-view model on NuscTrack improves metric localization, while background queries stay harder than instance queries. On the same queries, six-camera input improves instance 3D-AJ over a single camera while background 3D-AJ barely changes. We hope NuscTrack can serve as a shared benchmark for metric 3D TAP on surround-view driving video, measuring whether a tracker keeps an identity across cameras on both instance surfaces and unlabeled background.

## 1 INTRODUCTION

Surround-view driving systems already reason in a shared metric frame attached to the vehicle: 3D boxes, occupancy, and bird's-eye-view features all live in that frame (Caesar et al., 2020; Huang et al., 2021). Those representations still do not say whether a queried image location on a car, a lane marking, or a facade is the same 3D point a few frames later, or which of the six cameras currently see it. Track-Any-Point (TAP) is the interface for that question: follow an arbitrary pixel through time and mark when it is not visible (Doersch et al., 2022). On a surround rig the same physical point often leaves the camera that received the query and must be recovered in another view, including static background that never receives a 3D box. Figure 1 shows the measurement on a nuScenes clip: vehicle LiDAR samples handed from `CAM_FRONT_RIGHT` into `CAM_FRONT`, together with road and sidewalk points that still have a metric ego-frame trajectory and per-camera visibility.

Existing TAP resources miss that measurement. DriveTrack (Balasingam et al., 2024) scales TAP annotation on driving video by propagating LiDAR points with object poses, but the released tracks are monocular 2D trajectories on object surfaces. TAPVid-3D (Koppula et al., 2024) evaluates metric 3D TAP, including a DriveTrack split, yet the driving queries remain instance-centric and the protocol is still one camera per clip; published 2D-AJ on that split is far higher than 3D-AJ, which already shows that depth, not 2D correspondence, is the harder outdoor term. MVTracker (Rajič et al., 2025) tracks points in 3D from several cameras, but in bounded indoor capture with substantial overlap, and its original protocol can initialize from a 3D location rather than from an



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

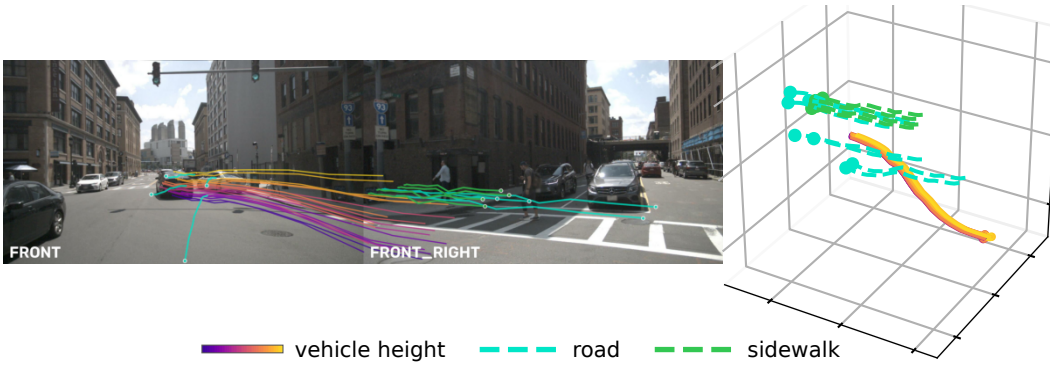


Figure 1: Example of a NuscTrack label on one nuScenes clip, with the same queried points shown on the surround images and in the ego frame.

image pixel on a wide-baseline rig. They therefore do not measure wide-baseline cross-camera tracking, or TAP on unlabeled background rather than only boxed objects.

These gaps suggest two questions. RQ1: can we construct scalable, reproducible labels for surround-view 3D TAP, covering instance surfaces and unlabeled background, with per-camera visibility? RQ2: does a shared protocol on this task reveal where current TAP methods succeed or fail, in ways that monocular and indoor multi-view evaluations do not?

A benchmark that answers RQ1 and RQ2 should satisfy four design goals. G1 Coverage: queries live on a six-camera outdoor rig, include background, and carry per-camera visibility. G2 Diagnostics: scores split All / instance / background, and the main 3D metrics stay in meters without a test-time scale fit to ground truth. G3 Reproducibility: every method reads the same 2D query list, with a fixed evaluation seed. G4 Fair query: the test-time query is an image point $(cam, t, x, y)$; ground-truth 3D is not provided as input.

We construct **NuscTrack** from nuScenes (Caesar et al., 2020) under those goals. Instance points follow DriveTrack-style rigidity inside 3D boxes; background points from static lidarseg classes stay in the world frame and move in the ego frame only through ego pose. Visibility uses depth completed by OMNI-DC (Zuo et al., 2025) rather than nearest-neighbor interpolation. We specify a shared-query protocol whose main scores are 3D Average Jaccard (3D-AJ), average points within threshold (APD), median trajectory error (MTE), and any-view occlusion accuracy (OA). Baselines include TAPNet-BEV and CoTracker-BEV (TAP-Net and CoTracker heads on the same BEV backbone), monocular TAP lifted with UniDepth, and MVTracker under the same pixel queries. MVTracker with Kubric pretraining and finetuning has extra motion pretraining that the other rows do not.

Our contributions are:

- **Task.** Synchronized surround-view video and queries $(cam, t, x, y)$ map to ego-frame 3D trajectories and per-camera, per-timestamp visibility. Test-time queries are 2D (Sec. 3).
- **Dataset.** NuscTrack on nuScenes: rigid instance surfaces, world-frame static background, and OMNI-DC visibility (out of frame, or completed depth smaller than 0.9 times the point-to-camera range) (RQ1; Sec. 4).
- **Protocol.** We use a shared query set; sample All / instance / background from the same frame; use any-view OA as the primary visibility metric, with camera-IoU and per-view OA where available (Sec. 6.1); and apply no ground-truth scale alignment. We evaluate TAPNet-BEV, CoTracker-BEV, monocular TAP with metric depth, and MVTracker trained from scratch or with unmatched motion pretraining (Sec. 3, 5).
- **Findings.** Our protocol reveals a gap overlooked by existing TAP benchmarks: methods strong in single-camera or indoor multi-view settings still struggle to maintain metric 3D identity on surround-view rigs, especially for static background regions (RQ2; Sec. 6.1).

Table 1: Comparison of NuscTrack with other driving-scene datasets. Columns are surround-view evaluation, metric 3D labels, long-term identity, and pixel- or point-level motion.

| Dataset | Surr. | Met. 3D | Long-term | Pix./pt |
|---|---|---|---|---|
| KITTI scene flow | × | ✓ | × | ✓ |
| Waymo, Argoverse 2 | ✓ | ✓ | ✓ | × |
| DriveTrack | × | × | ✓ | ✓ |
| TAPVid-3D | × | ✓ | ✓ | ✓ |
| NuscTrack (ours) | ✓ | ✓ | ✓ | ✓ |

Table 2: Comparison of what box MOT, occupancy prediction, and TAP each provide.

| Capability | Box MOT | Occ. pred. | TAP |
|---|---|---|---|
| Object class and size | ✓ | × | × |
| Occupied space | × | ✓ | × |
| Long-term identity | ✓ | × | ✓ |
| Structure with no box | × | ✓ | ✓ |
| Pixel-level correspondence | × | × | ✓ |

## 2 Related Work

**Box MOT, occupancy prediction, and TAP.** Multi-camera 3D detection already places boxes, with class and size, in a shared ego frame (Huang et al., 2021; Li et al., 2024). Occupancy prediction fills that frame with occupied space, including structures that have no box (Wei et al., 2023). Waymo and Argoverse 2 keep long-term instance identity for those boxes on a surround rig, with metric 3D labels, but release no pixel- or point-level motion (Sun et al., 2020; Wilson et al., 2021). None of the three outputs says whether a queried image location is the same 3D point a few frames later, or which cameras see it. Table 2 summarizes these distinctions, and Table 1 records the two datasets.

**2D TAP datasets and trackers.** TAP-Vid (Doersch et al., 2022) defined long-range 2D point tracking with occlusion on DAVIS and Kinetics and released TAP-Net. TAPIR (Doersch et al., 2023) combines that per-frame matcher with the iterative temporal window of PIPs (Harley et al., 2022); later trackers include CoTracker (Karaev et al., 2024; 2025) and LocoTrack (Cho et al., 2024). PointOdyssey (Zheng et al., 2023) supplies the long synthetic trajectories used to train TAP models, and BootsTAP (Doersch et al., 2024) then bootstraps TAPIR on unlabeled web video. DriveTrack (Balasingam et al., 2024) showed that driving video can supply TAP labels at a scale those sources do not cover, by moving LiDAR points with vehicle poses on the Waymo Open Dataset (Sun et al., 2020). EgoPoints (Darkhalil et al., 2025) stresses points that leave the frame and must be re-identified on return, still inside one egocentric camera. Those resources remain one video stream: they do not ask where the same point sits in the other cameras of a surround rig, and DriveTrack does not store a metric ego-frame trajectory for every query.

**3D TAP in driving.** KITTI scene flow releases metric 3D motion of pixels over two frames, not a long-term track and not on a surround rig (Menze & Geiger, 2015). TAPVid-3D (Koppula et al., 2024) lifts 2D tracks with depth and reports 3D-AJ, APD, and OA, with a DriveTrack split that is still monocular and instance-centric. Monocular 3D trackers such as SpatialTracker (Xiao et al., 2024), TAPIP3D (Zhang et al., 2025), SpatialTrackerV2 (Xiao et al., 2025), and Delta (Ngo et al., 2024) operate in that regime. SpatialTracker lifts pixels with monocular depth and tracks them in 3D; SpatialTrackerV2 is the later model we evaluate. On TAPVid-3D, 2D-AJ on DriveTrack stays in a range associated with strong 2D TAP, while 3D-AJ drops once metric depth is required (Koppula et al., 2024). That split diagnoses depth under one camera; it does not diagnose camera handover or background points without boxes. SceneTracker (Wang et al., 2024) comes closer on content: its LSFDriving set has long-term 3D tracks on static background, rigid vehicles, and non-rigid pedestrians in real driving video, but the input is monocular RGB-D, so it still does not score a handover across a surround rig.

**Indoor multi-view TAP.** MV-TAP (Koo et al., 2025) studies multi-view 2D correspondence. MV-Tracker (Rajič et al., 2025) predicts 3D trajectories from several views on indoor sets such as Panoptic Studio (Joo et al., 2015) and DexYCB (Chao et al., 2021). LAPA (Galoaa et al., 2025) is a feed-forward multi-camera model that maps synchronized views to 3D trajectories, evaluated on multi-camera versions of TAPVid-3D and PointOdyssey rather than on a driving surround rig. Indoor rigs have large overlap and a bounded workspace, unlike a surround driving rig, on which monocular TAP and indoor multi-view trackers fail, as shown in Fig. 2.

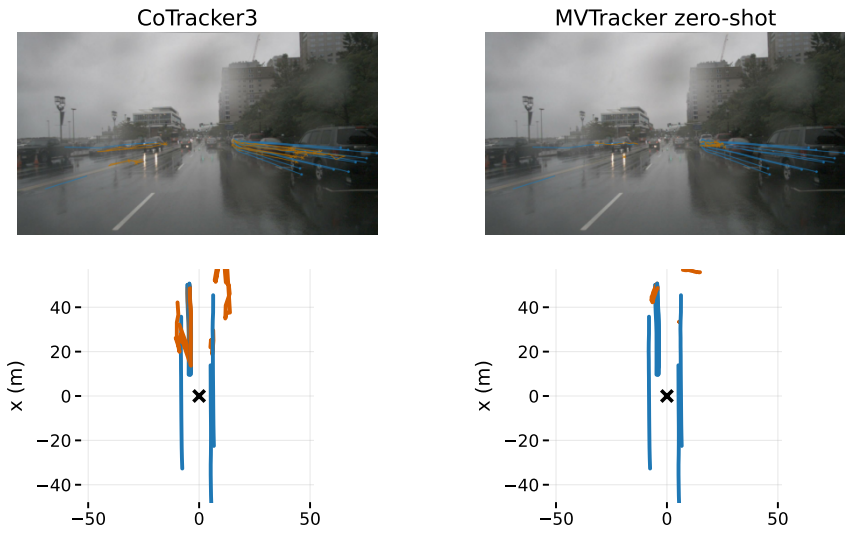


Figure 2: Zero-shot tracks on CAM_FRONT (top) and in ego-BEV (bottom), with ground truth in blue, predictions in yellow, and the ego vehicle marked by $\times$. Point-level tracks for these weights come from Kubric, not from a surround-view driving rig (Table 1). CoTracker3 is trained on monocular Kubric 2D tracks and then on real-video pseudo-labels. MVTracker is trained on synthetic multi-view Kubric with 3D tracks, still not on a surround-view rig.

## 3 Task and Protocol

**Task.** Each example is a clip with $N_{\text{camera}} = 6$ synchronized surround views, known intrinsics, and camera-to-ego extrinsics. A query is $(cam, t, x, y)$: pixel $(x, y)$ in camera $cam$ at time $t$. The model predicts the 3D trajectory in the ego-vehicle frame and visibility $v_{t,k} \in \{0, 1\}$ for every camera $k$ and timestamp $t$. A point that is occluded in the query camera may still be visible in another view; association is therefore 3D and cross-view. This is G1 (coverage) and G4 (fair query) in operational form.

**Shared queries.** All methods use the same exported query list: same clip, same tracks, same $(cam, t, x, y)$ (G3). We follow TAP-Vid *first*: the first timestamp at which the point is visible in any camera and the remaining horizon is long enough; if several cameras see it at that frame, each camera yields a query. Evaluation uses frames after the query with $\text{valid}{=}\text{true}$. External code must read the exported list rather than resample. When a method needs a 3D initialization, it back-projects with estimated depth; the depth source is given in Sec. 5. Oracle ground-truth depth at the query is an upper bound only and is not a main-table row.

**Splits (All / instance / background).** All, instance, and background scores are diagnostics on the same forward pass (G2): instance versus background is a cut of one query set, not two runs. Clips with no instance (or no background) tracks are kept; the empty subset is omitted from that split's mean. Dataset scores are micro-averages over trajectories.

**Metrics.** We report 3D-AJ, APD, MTE, and any-view OA as in TAPVid-3D (Koppula et al., 2024) and MVTracker (Rajič et al., 2025), with fixed-metre thresholds $\{0.01, 0.04, 0.16, 0.64, 2.56\}$ and no scale alignment to ground truth. For methods that emit six-camera visibility we additionally report camera-IoU and per-view OA on the same tracks (Sec. 6.1). Formulae are in Appendix A.

## 4 The NuscTrack Dataset

NuscTrack is built from nuScenes (Caesar et al., 2020): 1000 scenes, six surround cameras, roof LiDAR, and 3D boxes. We use the official train (700) and val (150) scenes; the test split has no

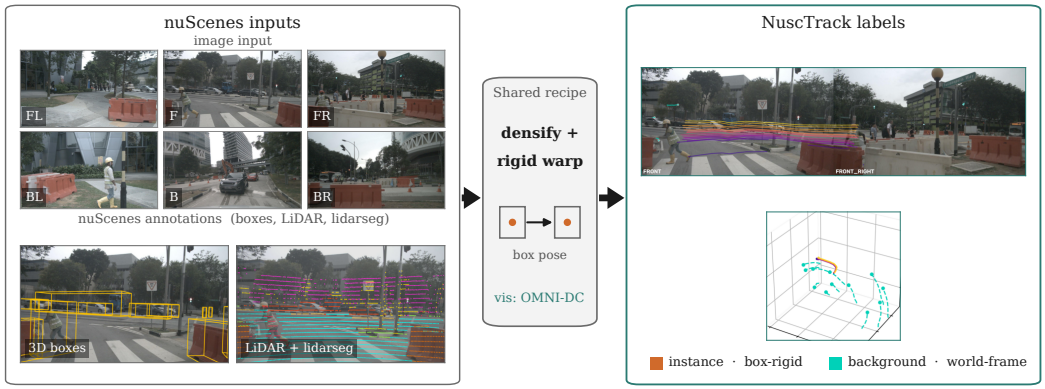


Figure 3: Pipeline for building NuscTrack from nuScenes.

public boxes. Figure 3 records what is inherited from DriveTrack and what is added (RQ1): densify-and-warp is the same recipe; the output is one ego-frame 3D track with six-camera visibility, plus world-frame background from lidarseg.

### 4.1 Construction

As in DriveTrack (Balasingam et al., 2024), densify-and-warp is the shared recipe (Fig. 3); the three steps below state the surround-view and background additions.

**Densification.** We densify each LiDAR keyframe by aggregating nearby keyframes and two sweeps, then downsample with stride 10. Static points remain fixed in world coordinates; points inside boxes are warped by per-frame object poses. Ego poses are interpolated to reduce LiDAR–camera time misalignment.

**Two motion models.** Instance trajectories keep a surface point in the object's box frame and transform it by box poses; at most 5000 tracks are kept per instance. Background trajectories are points outside every 3D box, drawn from static lidarseg classes (road, walkable surfaces, manmade structure, vegetation), voxel-filtered at 1 m, capped at 5000 per scene, and held fixed in the world frame; later frames apply only ego-pose changes. The two models are not mixed: a background point is never rotated by a box. Figure 4 shows both on adjacent cameras: a boxed instance on the left of each row, and background tracks with no box on the right.

**Visibility.** We complete per-camera depth with OMNI-DC (Zuo et al., 2025) from densified LiDAR, replacing nearest-neighbor interpolation used in an earlier export. Each 3D point is projected to all six cameras. It is marked invisible if it falls outside the image or is occluded, i.e., when completed depth at the pixel is smaller than $0.9$ times the point-to-camera distance.

### 4.2 Filtering and Quality Gates

Each scene is split into two 24-frame clips with opposite temporal order. We keep trajectories that span at least 10 frames, are visible in at least 4 frames from any camera, and come within 20 m of the ego vehicle at least once. Clips with no remaining trajectories are dropped. These automatic filters are the quality gates. There is no exhaustive human re-annotation, so inter-annotator agreement on the full corpus does not apply. A human audit of 96 unique 3D tracks from the val evaluation queries inspects stored 2D projections and per-camera visibility (Appendix F). Training and evaluation then sample at most $N{=}384$ tracks per clip, stratified with background ratio $0.5$ (about 1:1 instance versus background; the other class fills if one is short). Training may reshuffle each epoch; evaluation uses a fixed seed per scene and clip direction. That stratified sample is All in Table 4; instance and background scores cut it after one forward pass as in Sec. 3.

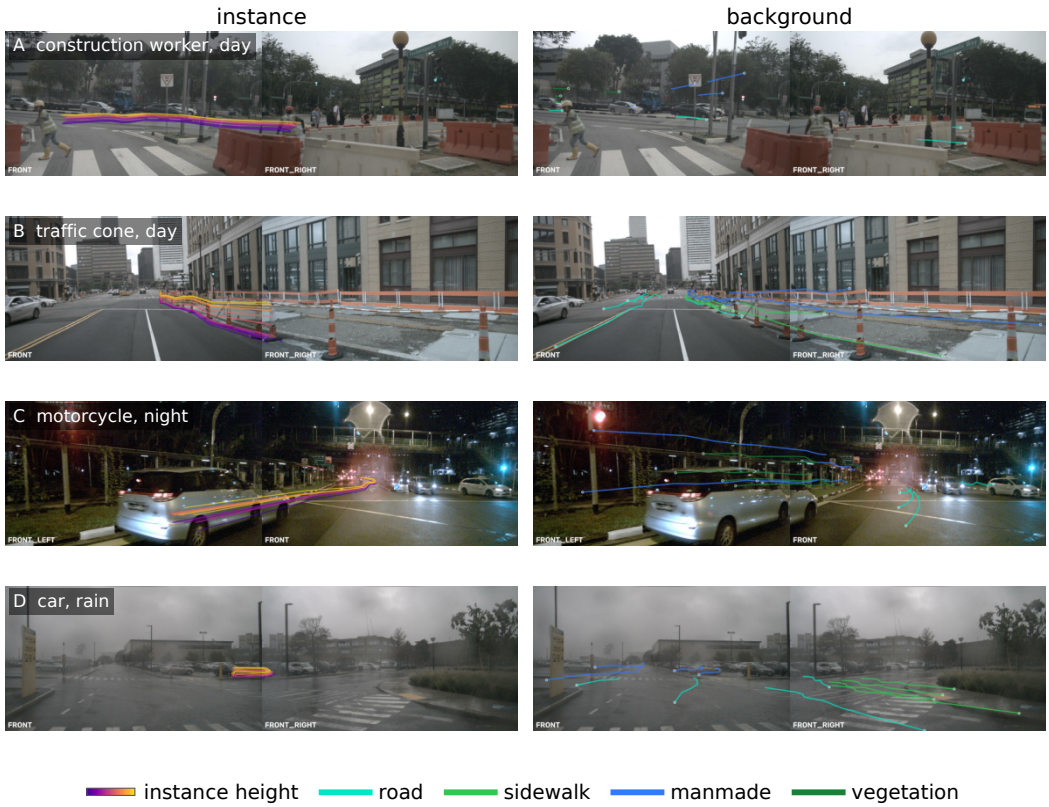


Figure 4: Cross-view NuscTrack tracks on four clips. Each row is one timestamp on a pair of adjacent cameras. Left: instance surface points inside a 3D box (color encodes height). Right: background tracks from static lidarseg classes that have no box.

### 4.3 Statistics

Table 3 is the processed pool after clip filtering, counted as unique 3D tracks (a point seen by several cameras at the TAP-Vid first frame writes several query records; 1.12 queries per unique track). Train/val cover the official 700/150 scenes as 1400/300 bidirectional clips. Figure 5 shows the mix: vehicles are 71% of unique 3D tracks, while each background lidarseg group is 0.5–0.9 M, comparable to human rather than to vehicle. Overall, 78.1% of unique tracks are visible in at least two cameras (instance 78.5%, background 75.5%); mean length is 19.35 frames. This pool is *not* the evaluation mix: All is the stratified 384-point sample in Sec. 4.2, not the natural 86%/14% instance/background split. Leaf counts, histograms, and generation hyperparameters are in Appendix E; a human audit of automatic labels in Appendix F; license, PII, and the split relative to nuScenes detection pretraining in Appendix G; label caveats in Appendix C.

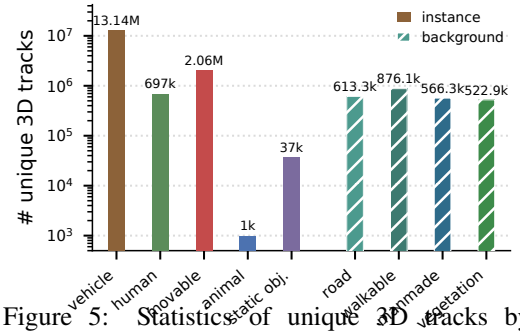


Figure 5: Statistics of unique 3D tracks by nuScenes semantic category in NuscTrack (log-$y$ scale). Solid bars denote instance superclasses and hatched bars background tracks. Vehicles dominate, while each background group is comparable to humans (∼0.5–0.9,M).

Table 3: Scale of the NuscTrack pool after clip filtering, counted as unique 3D tracks. Unique 3D, Inst., and BG are in millions of tracks. Query tracks total 17.37 M / 3.37 M on train / val.

| | Scenes/clips | Unique 3D | Inst. | BG | Len. | $\geq$2 cam | Q/clip |
|---|---|---|---|---|---|---|---|
| Train | 700/1400 | 15.49 | 13.37 | 2.12 | 19.30 | 79.0% | 12,408 |
| Val | 150/300 | 3.03 | 2.57 | 0.46 | 19.60 | 73.4% | 11,237 |

## 5 BASELINES

We evaluate several methods under Sec. 3. Monocular TAP and every MVTracker row use UniDepthV2; TAPNet-BEV and CoTracker-BEV use a learned 50-bin head. Initialization is in the paragraphs below and in Sec. 6. Implementation details are in Appendix B.

**Monocular TAP with metric depth.** SpatialTrackerV2 (Xiao et al., 2025), CoTracker3 (Karaev et al., 2025), and TAPIP3D (Zhang et al., 2025) run with official weights on the query camera only. 2D tracks are lifted with UniDepthV2 (Piccinelli et al., 2025) when the method needs metric depth. They have no per-camera visibility over the other five views.

**TAPNet-BEV.** We adapt TAP-Net (Doersch et al., 2022) by replacing image features with the same BEV backbone as below, regressing 3D coordinates, and predicting six-camera visibility. Query depth uses a 50-bin MLP. The main-table row loads the same BEVDet initialization as CoTracker-BEV so the comparison is the tracking head, not detector pretraining.

**MVTracker.** MVTracker (Rajič et al., 2025) is evaluated on the same 2D queries (no ground-truth 3D query), with UniDepthV2 as the metric depth. Scratch training on NuscTrack and MV-Kubric pretraining plus finetuning are both in Table 4; the latter has a pretraining budget that is not matched to the other rows.

**CoTracker-BEV.** We adapt CoTracker (Karaev et al., 2024) by replacing image features with BEV features from a BEVDet image backbone and view transformer (Huang et al., 2021), lifting the shared 2D query with a 50-bin depth head, and running sliding-window trajectory updates in ego 3D with six-camera visibility. It is a surround-view baseline, not a method claim. Unless noted, BEV weights come from a nuScenes BEVDet checkpoint so the TAPNet-BEV row isolates the tracking head. That checkpoint sees nuScenes detection images on the train split, not NuscTrack TAP labels. The default schedule is in Appendix B.

## 6 EXPERIMENTS

Experiments are organized by the research questions in Sec. 1. RQ1 is answered by the construction in Sec. 4.1. RQ2 is Table 4 (including its camera-IoU column), Table 5, and Table 6.

**Setup.** Every row uses the same exported queries (Sec. 3): surround-view methods take $(cam, t, x, y)$; monocular TAP runs each camera as its own video, so the tracker sees $(t, x, y)$ and the camera index only selects the video, then lifts to ego 3D with the video calibration. No row receives ground-truth 3D or ground-truth query depth. External depth is UniDepthV2; TAPNet-BEV and CoTracker-BEV use a learned 50-bin head, share the BEVDet initialization and training schedule, and differ in the trajectory head. MVTracker zero-shot, scratch, and Kubric+FT share the same six-camera inputs; Kubric+FT's extra pretraining is not matched to the other rows. Resolutions, checkpoints, and the remaining hyperparameters are in Appendix B.

### 6.1 MAIN BENCHMARK

**Point-Tracking.** Table 4 is the primary comparison on NuscTrack val. Each cell is All / instance / background on the same 115,200 / 56,264 / 58,936 queries. The row that combines NuscTrack training with Kubric initialization has extra motion pretraining and is not a matched comparison.

Table 4: Main point-tracking results on NuscTrack val. 3D-AJ, APD, and OA are percentages, MTE is in metres, and cam-IoU is in $[0, 1]$; $^*$ is not trained on NuscTrack, $^\dagger$ is trained on the train split, $^\ddagger$ is an external initialization, $--$ is a method without six-camera visibility, and (per-cam) adds that head to MVTracker trained from scratch.

| Method | 3D-AJ | APD | OA | MTE | cam-IoU |
|---|---|---|---|---|---|
| SpatialTrackerV2$^*$ | 6.91 / 8.29 / 5.59 | 9.23 / 11.02 / 7.51 | 61.04 / 61.12 / 60.97 | 11.21 / 10.21 / 12.17 | – |
| CoTracker3$^*$ | 7.52 / 9.90 / 5.25 | 9.67 / 12.62 / 6.85 | 56.99 / 56.01 / 57.92 | 16.96 / 15.54 / 18.30 | – |
| TAPIP3D$^*$ | 7.83 / 10.44 / 5.34 | 10.62 / 14.03 / 7.36 | 63.44 / 63.63 / 63.26 | 12.23 / 9.70 / 14.65 | – |
| TAPNet-BEV$^{\dagger\ddagger}$ | 2.78 / 4.43 / 1.21 | 5.26 / 7.54 / 3.07 | 63.79 / 69.09 / 58.73 | 9.93 / 9.00 / 10.83 | 0.394 / 0.451 / 0.339 |
| MVTracker$^{*\ddagger}$ | 5.07 / 6.67 / 3.54 | 6.53 / 8.58 / 4.58 | 60.15 / 59.06 / 61.18 | 21.25 / 20.08 / 22.37 | – |
| MVTracker$^\dagger$ | 10.11 / 12.20 / 8.12 | 14.30 / 16.34 / 12.35 | 76.36 / 80.08 / 72.81 | 4.24 / 4.45 / 4.04 | – |
| MVTracker$^\dagger$ (per-cam) | 10.99 / 12.80 / 9.26 | 14.62 / 16.48 / 12.84 | 83.79 / 88.11 / 79.65 | 4.12 / 4.46 / 3.79 | 0.853 / 0.847 / 0.860 |
| CoTracker-BEV$^\dagger$ | 2.53 / 2.53 / 2.53 | 4.01 / 3.92 / 4.10 | 72.27 / 75.70 / 69.01 | 8.86 / 10.73 / 7.09 | 0.644 / 0.622 / 0.665 |
| CoTracker-BEV$^{\dagger\ddagger}$ | 8.54 / 12.11 / 5.13 | 11.55 / 15.71 / 7.59 | 77.35 / 82.45 / 72.47 | 4.90 / 4.04 / 5.72 | 0.741 / 0.789 / 0.694 |
| MVTracker$^{\dagger\ddagger}$ | 12.65 / 17.05 / 8.45 | 17.36 / 22.02 / 12.92 | 77.06 / 82.94 / 71.44 | 3.81 / 3.16 / 4.44 | – |

Table 5: Per-camera occlusion accuracy on the NuscTrack val queries from Table 4, in surround-view order with matching markers.

| Method | FL | F | FR | BL | B | BR |
|---|---|---|---|---|---|---|
| TAPNet-BEV$^{\dagger\ddagger}$ | 87.3 | 74.0 | 86.3 | 93.2 | 78.4 | 92.8 |
| MVTracker$^\dagger$ (per-cam) | 94.2 | 91.4 | 93.9 | 94.9 | 87.1 | 94.8 |
| CoTracker-BEV$^\dagger$ | 92.8 | 86.2 | 92.6 | 94.8 | 82.1 | 95.1 |
| CoTracker-BEV$^{\dagger\ddagger}$ | 94.8 | 92.0 | 94.3 | 95.6 | 86.7 | 95.7 |

Table 6: 3D-AJ and MTE metric comparison of single-view and surround-view CoTracker-BEV on the NuscTrack valuation set.

| Setting | 3D-AJ | MTE |
|---|---|---|
| 1-view (FRONT) | 4.6 / 5.8 / 3.4 | 7.36 |
| surround (FRONT) | 7.8 / 12.1 / 3.5 | 5.98 |
| surround (all) | 8.54 / 12.11 / 5.13 | 4.90 |

Zero-shot rows, trained on monocular TAP data or on indoor multi-view data, stay together: All 3D-AJ is about 5–8 and MTE is above 11 m. The controlled comparison is one multi-view model with and without NuscTrack training: Kubric weights alone give 5.07 3D-AJ and 21.25 m MTE, while training on NuscTrack raises this to 10.11 / 12.20 / 8.12 and 4.24 m. BEV adaptations of existing TAP methods, also trained on this split, sit with that trained setting rather than with the zero-shot rows. In both cases, background 3D-AJ remains below instance 3D-AJ.

Models trained on monocular TAP data or indoor multi-view data do not transfer directly to NuscTrack in the zero-shot setting, whereas training on NuscTrack improves metric localization. Background queries, however, remain harder than instance queries.

**Per-camera visibility.** Any-view OA only checks visibility in at least one camera. Camera-IoU, the last column of Table 4, scores whether the predicted camera set matches; the formula is in Appendix A. The cell is filled only for a six-camera visibility head. Monocular TAP never sees the other views, and the other MVTracker rows report any-view visibility, so those entries stay blank.

On the filled rows, camera-IoU separates the heads more sharply than any-view OA. TAPNet-BEV's any-view OA stays with the monocular band while its camera-IoU does not, so a correct decision that the point is visible somewhere is not the same as naming the cameras. BEVDet initialization raises CoTracker-BEV's camera-IoU together with its localization. The instance–background gap is not stable across these heads: it favors instances with BEVDet, reverses without that initialization, and is nearly flat for the per-camera MVTracker head.

Table 5 does not replace camera-IoU. Per-view OA remains high across cameras, with `CAM_BACK` usually the weakest, while camera-IoU can be much lower. Correctly predicting invisibility for a camera the point never enters improves per-view OA without improving camera-set matching.

### 6.2 Is surround view necessary?

Table 6 compares a single-view model with a surround-view model. The first two rows use the same `CAM_FRONT` queries ($n$=112,807), ensuring that any improvement is not due to evaluating additional points that start in other cameras. The last row follows the full protocol of Table 4 ($n$=115,200). On the matched query set, surround-view training improves instance 3D localization and lowers MTE, showing that the gain comes from multi-view training rather than broader query coverage. In contrast, background 3D-AJ changes only marginally, from 3.4 to 3.5.

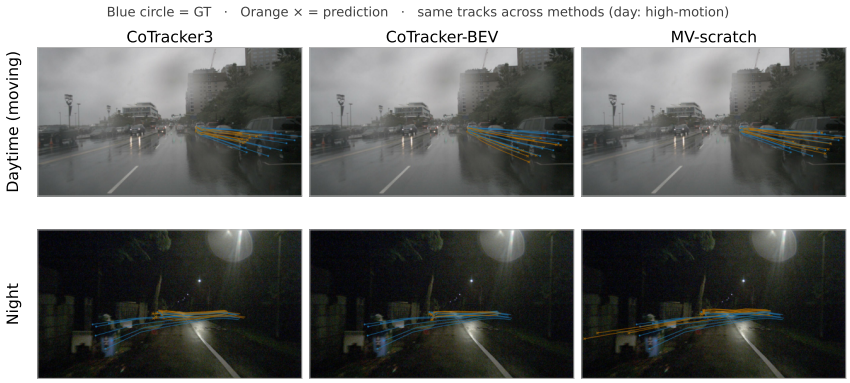


Figure 6: Qualitative comparison of predicted tracks and ground truth.

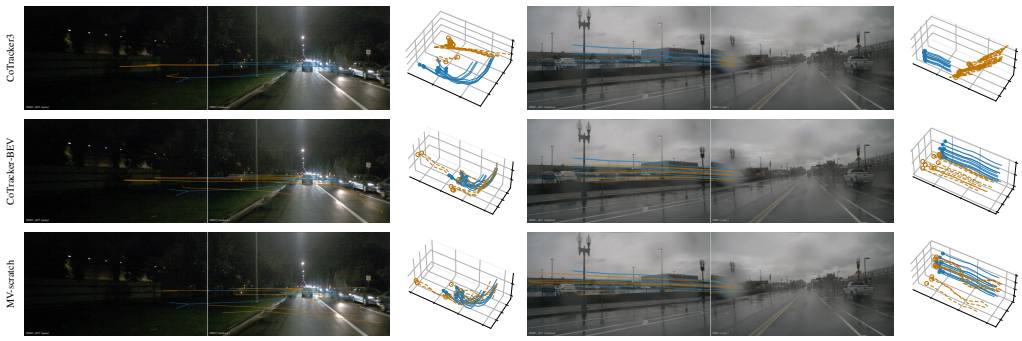


Figure 7: Qualitative examples of cross-camera handover.

### 6.3 Qualitative examples

Figure 6 gives a compact `CAM_FRONT` view of two regimes under the shared query protocol. The daytime row uses a *moving* ego clip (`scene-0906_former`, about 98 m travel) rather than a high-AJ parked scene: road and roadside trails make motion readable. Scratch MVTracker stays closer to the ground-truth flow while CoTracker-BEV shows more wobble; CoTracker3+UniDepth is weaker on this clip. The night row (`scene-1060_later`) is harder for CoTracker3+UniDepth and CoTracker-BEV; scratch MVTracker is the least broken of the three but remains far from solved. Kubric-pretrained MVTracker is omitted; that row has extra Kubric pretraining (Table 4). Appendix D adds ego-BEV panels and a roadside early-drop case.

Figure 7 is the case those `CAM_FRONT` panels do not show: the point leaves the query camera and continues in an adjacent view. On the rainy daytime handover (`scene-0912_former`), scratch MVTracker reaches 13.2 3D-AJ against 1.7 for CoTracker3+UniDepth. A query-camera-only lift cannot continue across that seam. Screening details and two further fragments are in Appendix I.

## 7 Conclusion

NuscTrack evaluates cross-view 3D track-any-point on a six-camera driving rig, mapping an image query to a metric ego-frame trajectory for both instance and background points and to per-camera visibility from completed depth, under one shared query set that scores All, instance, and background from a single sample without fitting scale to ground truth. Models trained on monocular TAP data or indoor multi-view data do not transfer directly to NuscTrack in the zero-shot setting, whereas training on NuscTrack improves metric localization, and background queries remain harder than instance queries; surround-view training improves instance 3D localization over a single-view model on the same queries, so the gain is not from scoring additional cameras, whereas background 3D-AJ barely changes. NuscTrack is a testbed for these results.

### AI USE STATEMENT

(This section is required and does not count toward the page limit.)

We used large language models to aid and polish the writing of this paper. We have reviewed the text and take responsibility for the final content.

### REPRODUCIBILITY STATEMENT

Annotation code, converter filters, evaluation code, the fixed query list, and training configs will be released with a public URL upon acceptance. NuscTrack tracks inherit the nuScenes license; intended use, personally identifiable information, and the split relative to detection pretraining are in Appendix G. Sections 3–5 specify queries, splits, and metrics; Sec. 6.1 records camera-IoU and per-view OA; the appendix records a query-camera 2D diagnostic where native 2D tracks are stored, and implementation details. A fixed evaluation seed is used for track sampling.

## REFERENCES

Arjun Balasingam, Joseph Chandler, Chenning Li, Zhoutong Zhang, and Hari 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*, pp. 22488–22497, 2024.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

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

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

Ahmad Darkhalil, Rhodri Guerrier, Adam W Harley, and Dima Damen. Egopoints: Advancing point tracking for egocentric videos. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 8556–8565. IEEE, 2025.

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

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10061–10072, 2023.

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped training for tracking-any-point. In *Asian Conference on Computer Vision*, 2024.

Bishoy Galoaa, Xiangyu Bai, Shayda Moezzi, Utsav Nandi, Sai Siddhartha Vivek Dhir Rangoju, Somaieh Amraee, and Sarah Ostadabbas. Look around and pay attention: Multi-camera point tracking reimagined with transformers. *arXiv preprint arXiv:2512.04213*, 2025.

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

Junjie Huang, Guan Huang, Zheng Zhu, Ye Yun, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pp. 3334–3342, 2015.

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

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6013–6022, 2025.

Jahyeok Koo, Inès Hyeonsu Kim, Mungyeom Kim, Junghyun Park, Seohyun Park, Jaeyeong Kim, Jung Yi, Seokju Cho, and Seungryong Kim. Mv-tap: Tracking any point in multi-view videos. *arXiv preprint arXiv:2512.02006*, 2025.

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

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):2020–2036, 2024.

Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3061–3070, 2015.

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.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Frano Rajič, Haofei Xu, Marko Mihajlovic, Siyuan Li, Irem Demir, Emircan Gündoğdu, Lei Ke, Sergey Prokudin, Marc Pollefeys, and Siyu Tang. Multi-view 3d point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

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.

Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21729–21740, 2023.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

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

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6726–6737, 2025.

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

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

Yiming Zuo, Willow Yang, Zeyu Ma, and Jia Deng. OMNI-DC: Highly robust depth completion with multiresolution depth integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9287–9297, 2025.

## A METRICS

Let $P_t$ and $\hat{P}_t$ be ground-truth and predicted ego 3D locations at time $t$ after the query, on frames with $\mathrm{valid}{=}\mathrm{true}$ and finite ground truth. APD averages, over the five thresholds $\tau \in \{0.01, 0.04, 0.16, 0.64, 2.56\}$ m, the fraction of visible valid frames with $\|\hat{P}_t - P_t\|_2 < \tau$. 3D-AJ is the TAP Jaccard that jointly counts localization within $\tau$ and visibility; we average the five thresholds. MTE is the median over time of $\|\hat{P}_t - P_t\|_2$ on finite errors, then averaged over trajectories (mean of medians). Non-finite predictions are misses for AJ and APD; trajectories with no finite predicted error are omitted from MTE and counted in $\mathrm{pct_nonfinite_pred}$. Dataset scores are unweighted means over trajectories.

Let $v_{t,k}, \hat{v}_{t,k} \in \{0, 1\}$ be ground-truth and predicted visibility in camera $k$ on those frames (predicted scores thresholded at $0.5$). Any-view flags are $v_t = \bigvee_k v_{t,k}$ and $\hat{v}_t = \bigvee_k \hat{v}_{t,k}$. Any-view OA is the fraction of frames where $\hat{v}_t = v_t$, then averaged over trajectories.

**Per-view OA and camera-IoU.** These two scores are defined only for methods that emit six-camera visibility. Per-view OA for camera $k$ is the fraction of the same frames where $\hat{v}_{t,k} = v_{t,k}$, then averaged over trajectories. Table 5 reports All; a correct invisible decision on a camera the point never enters still counts.

Camera-IoU is the set IoU of the visible-camera sets $S_t = \{k : v_{t,k} = 1\}$ and $\hat{S}_t = \{k : \hat{v}_{t,k} = 1\}$. It is computed only on frames where the point is visible in at least one ground-truth camera ($|S_t| > 0$):

$$\mathrm{IoU}_t = |S_t \cap \hat{S}_t| / |S_t \cup \hat{S}_t|. \tag{1}$$

The trajectory score is the mean of $\mathrm{IoU}_t$ over those frames; the dataset score is the mean over trajectories that have at least one such frame (All / instance / background in Table 4). Methods that emit only any-view visibility are left blank; we do not impute camera-IoU by reprojecting 3D tracks.

Query-camera 2D TAP-Vid uses only the query camera, with coordinates mapped to $256 \times 256$ and pixel thresholds $\{1, 2, 4, 8, 16\}$. It is reported when native 2D tracks are stored (Table 12). Per-view 2D scores a reprojection on every pair $(t, k)$ where ground truth is visible in camera $k$. The two 2D columns are not merged into one number. Per-view 2D is defined here and is not tabulated.

## B BASELINE DETAILS

Shared inputs, the ban on ground-truth query depth and ground-truth 3D queries, and the controls that match TAPNet-BEV to CoTracker-BEV and the three MVTracker rows to each other are in Sec. 6. This section records checkpoints and hyperparameters that are not shared.

**Monocular TAP with UniDepthV2.** SpatialTrackerV2 uses the official offline Hugging Face weights (`Yuxihenry/SpatialTrackerV2-Offline`) (Xiao et al., 2025). CoTracker3 uses the official offline checkpoint `scaled_offline.pth` (Karaev et al., 2025). TAPIP3D uses the

official RGB-D checkpoint `tapip3d_final.pth` (Zhang et al., 2025). None of the three is finetuned on NuscTrack. SpatialTrackerV2 and TAPIP3D are evaluated at $384 \times 512$, matching TAPIP3D's published eval resolution; CoTracker3 uses the native query-camera frames. They do not predict per-camera visibility on the other five views.

**TAPNet-BEV.** The trajectory head is TAP-Net on BEV features rather than the CoTracker-style sliding window used by CoTracker-BEV (Doersch et al., 2022; Huang et al., 2021). TAPNet-BEV and CoTracker-BEV share the BEVDet backbone and view transformer, image size $432 \times 768$, the nuScenes BEVDet initialization, and one schedule: 8 GPUs, batch size 1, AdamW at $2 \times 10^{-4}$, EMA weights after 64 epochs.

**MVTracker.** All three rows use six-camera RGB-D at native $432 \times 768$ (no Kubric $384 \times 512$ resize) and 4 update iterations (Rajič et al., 2025). Zero-shot loads the official Kubric checkpoint `mvtracker_200000_june2025.pth` and is not trained on NuscTrack. Scratch trains from random weights on NuscTrack train (8 GPUs, batch size 1, AdamW at $5 \times 10^{-4}$, 7744 steps). Kubric+FT loads the same official Kubric checkpoint and finetunes on NuscTrack train (7 GPUs, batch size 1, AdamW at $5 \times 10^{-5}$, 7000 steps). That extra pretraining is not matched to the other rows in Table 4.

**CoTracker-BEV.** The query pixel is lifted to ego 3D with the 50-bin head, copied across $T$ frames, and updated for $M$ steps from local BEV correlation features, jointly with six-camera visibility and a confidence head. Training uses Huber loss on 3D tracks, confidence cross-entropy, per-camera visibility cross-entropy, and Huber on the query lift. Training uses that shared schedule with window $S{=}12$, $M{=}4$ updates, correlation radius $r{=}3$, and a BEV grid of $0.8$ m. The visibility training loss is per-camera cross-entropy; any-view OA is evaluation-only. The scratch row uses that schedule with no BEVDet initialization. The BEVDet checkpoint is trained for nuScenes detection on the official train scenes; it does not see NuscTrack TAP labels. Oracle query depth (`qp_manner=use_gt`) is excluded from the main table.

## C Limitations

The labels inherit rigidity from 3D boxes, so human surface tracks are an approximation and articulated parts (a sliding door, a raised arm) are not followed. The audit sample in Appendix F includes human and movable tracks for that reason. Keyframe clips at 2 Hz are shorter and more temporally sparse than many indoor TAP videos. Visibility is only as good as completed depth and calibration. Grazing cases of the $0.9$ completed-depth rule are recorded as Uncertain in that audit, not as visibility errors. NuscTrack cannot be reproduced on KITTI-style video that lacks comparable 3D boxes and lidarseg without a new annotation path. Kubric-pretrained MVTracker without NuscTrack training matches the weak monocular band; the same architecture after NuscTrack finetuning is the strongest row we report, but its Kubric motion pretraining is not matched to the other rows, so we do not treat it as a like-for-like competitor.

## D Additional qualitative examples

Figure 8 expands Figure 6 with ego-BEV views and a roadside early-drop case. All three columns use the same subsampled queries so blue GT curves are comparable across methods. In each ego-BEV panel, $x$ is forward and $y$ is left–right in the ego frame; the black cross marks the ego origin. The rows are diagnostics, not a ranking: Kubric+FT is omitted, as in the main figure. Unless noted, 3D-AJ, MTE, and OA below are *per-clip* scores on that named 24-frame fragment only; they are not the val aggregates in Table 4.

**Row A: `scene-0906_former` (moving daytime).** On this fragment alone, `CAM_FRONT` trails are long because the ego vehicle travels about 98 m; the same queries in ego-BEV should stay near the blue GT. Per-clip scores on `scene-0906_former`: CoTracker3+UniDepth reaches only 6.13 3D-AJ with 14.38 m MTE, while CoTracker-BEV and scratch MVTracker are 16.81 / 2.50 m and 13.83 / 2.42 m. The ego-BEV gap matches that split: the monocular lift often leaves the orange trail beside the GT lane rather than on it, which is the usual cost of a strong 2D tracker plus an

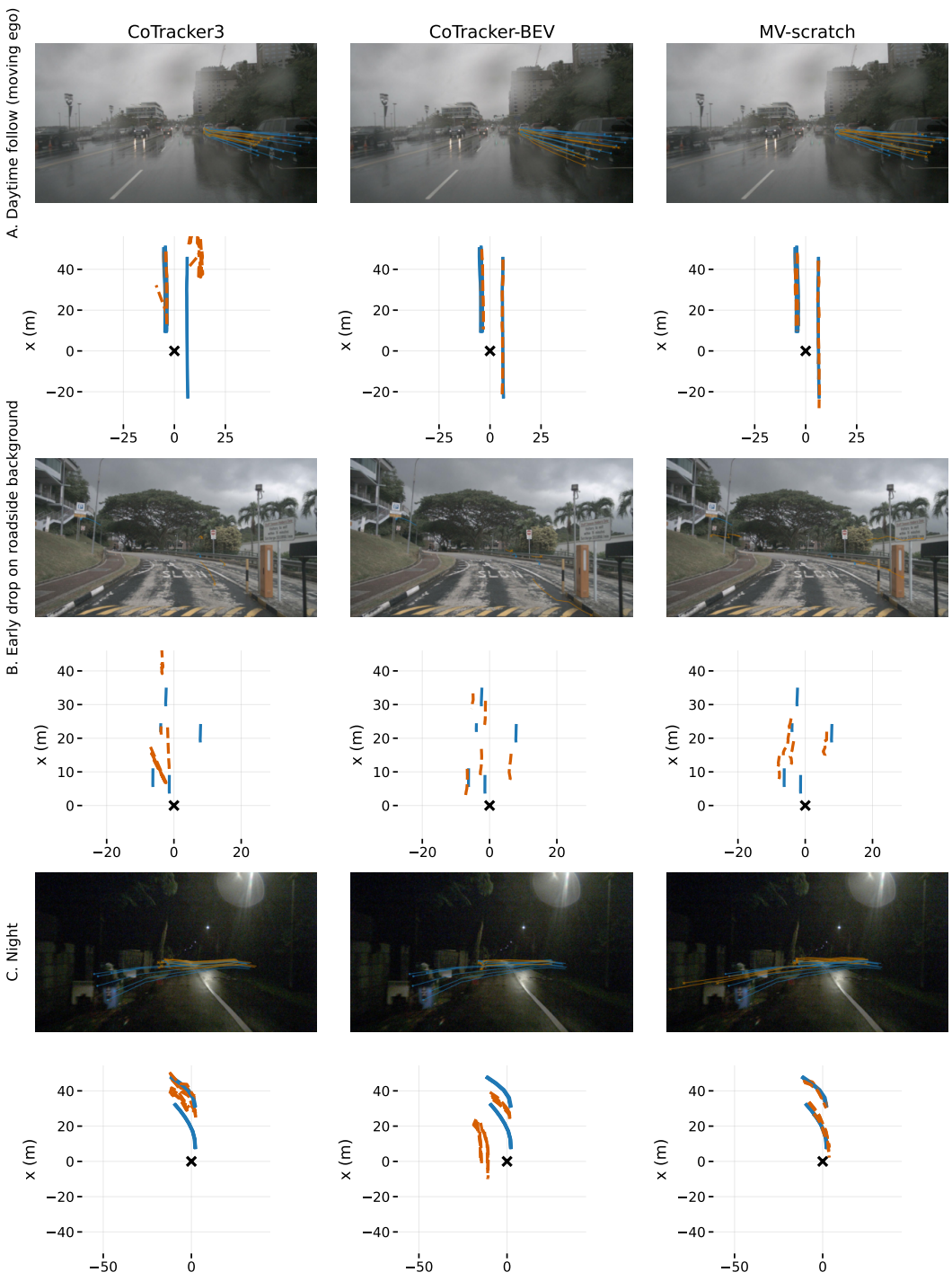


Figure 8: Additional qualitative comparison of predicted tracks and ground truth, with the ego-frame view under each image.

independent metric depth head under large ego motion. CoTracker-BEV stays closer in range but oscillates around the GT curve (BEV quantization and sliding-window updates). Scratch MVTracker is smoother in BEV, yet background points remain the weak side of this fragment (per-clip BG 3D-AJ 7.62 vs. instance 20.05).

**Row B: `scene-0929_later` (early drop on roadside background).** This fragment is the OA-versus-3D pattern in Finding 1. Per-clip on `scene-0929_later`, CoTracker-BEV any-view OA is still 81.6, but 3D-AJ collapses to 2.37 (MTE 6.88 m); scratch MVTracker is only slightly higher (3.37 3D-AJ, 5.50 m MTE), and CoTracker3+UniDepth is 2.17 / 11.02 m with OA already down to 45.4. In the figure, orange trails on roadside structure often terminate or peel away while the blue GT continues along the curb—visibility can remain partially correct while metric identity on static background is already wrong. Background queries have no box motion to lean on, so small depth or correspondence errors integrate into large ego-frame drift; the 2D overlay alone does not reveal that failure.

**Row C: `scene-1060_later` (night).** Low light on this night fragment hurts both appearance matching and depth. Per-clip on `scene-1060_later`, CoTracker3+UniDepth and CoTracker-BEV fall to 1.26 and 0.34 3D-AJ with MTE 20.64 m and 12.75 m; scratch MVTracker is the least broken (6.82 3D-AJ, 4.10 m MTE) but still far from the daytime fragment above. Ego-BEV makes the gap obvious: orange curves bend away from GT or shorten under headlights and underexposed roadside, even when a few `CAM_FRONT` markers still look plausible. Cross-view handover is in Figure 7; two further fragments are in Appendix I.

## E DATASET POOL STATISTICS

Counts below are the processed 24-frame pool (unique 3D tracks unless noted), not the 384-point stratified sample used in training and Table 4. A query record is written per camera when several cameras see the point at the TAP-Vid first frame.

**Generation.** Raw export uses seed 42, OMNI-DC depth, occlusion ratio 0.9, at most 5000 tracks per instance and 5000 background tracks per scene after 1 m voxel filtering. The converter then applies the clip filters in Sec. 4.2. Raw background points before clip filtering sum to 4,247,279 across 5136 background files. Figure 3 in the main text compares the recipe to DriveTrack.

**Composition.** Table 7 repeats the groups of Fig. 5 with query-track totals. Figure 9 and Table 8 give the 29 leaf labels. Mean any-view visible-frame ratio on valid frames is 0.848 (instance 0.863, background 0.755). Mean ego distance on valid frames is 20.44 m (instance 19.91, background 23.71). Among scenes in the processed clips, official nuScenes captions mention night in 99 scenes and rain in 165.

Table 7: Counts of query tracks and unique 3D tracks by coarse nuScenes category in the NuscTrack pool.

| Group | Query tracks | Unique 3D |
|---|---|---|
| vehicle | 14,760,971 | 13,140,700 |
| human | 806,297 | 697,014 |
| movable_object | 2,340,052 | 2,060,760 |
| animal | 1,158 | 1,000 |
| static_object | 42,266 | 37,017 |
| background (road) | 656,558 | 613,313 |
| background (walkable) | 949,683 | 876,149 |
| background (manmade) | 619,905 | 566,296 |
| background (vegetation) | 564,870 | 522,853 |

**Offset from the box center.** A 3D box encodes pose and size; it does not encode where a LiDAR return sits on the surface. For a rigid instance the Euclidean offset from the box center is constant over the clip, so we compute it once: at the first annotated frame we take all stored instance points, then keep the *median* radius per instance. Each of 30,919 instances with at least one finite track is weighted equally, including the 3.5% of vehicles that hit the raw-export cap of 5000 tracks. Background tracks have no box and are omitted. Figure 13 reports the normalized radius $\tilde{r}=2r/\sqrt{w^2+l^2+h^2}$, so 0 is the box center and 1 is a corner. Vehicle instances (median $r=1.78$ m, $\tilde{r}=0.64$) and other movable/static objects (median $r=0.54$ m, $\tilde{r}=0.52$) sit well away from the center; humans are closer in metres because the box is smaller (median $r=0.38$ m) but still not concentrated at $\tilde{r}=0$ (median $\tilde{r}=0.37$). Larger $r$ on vehicles is box size and surface sampling, not higher label

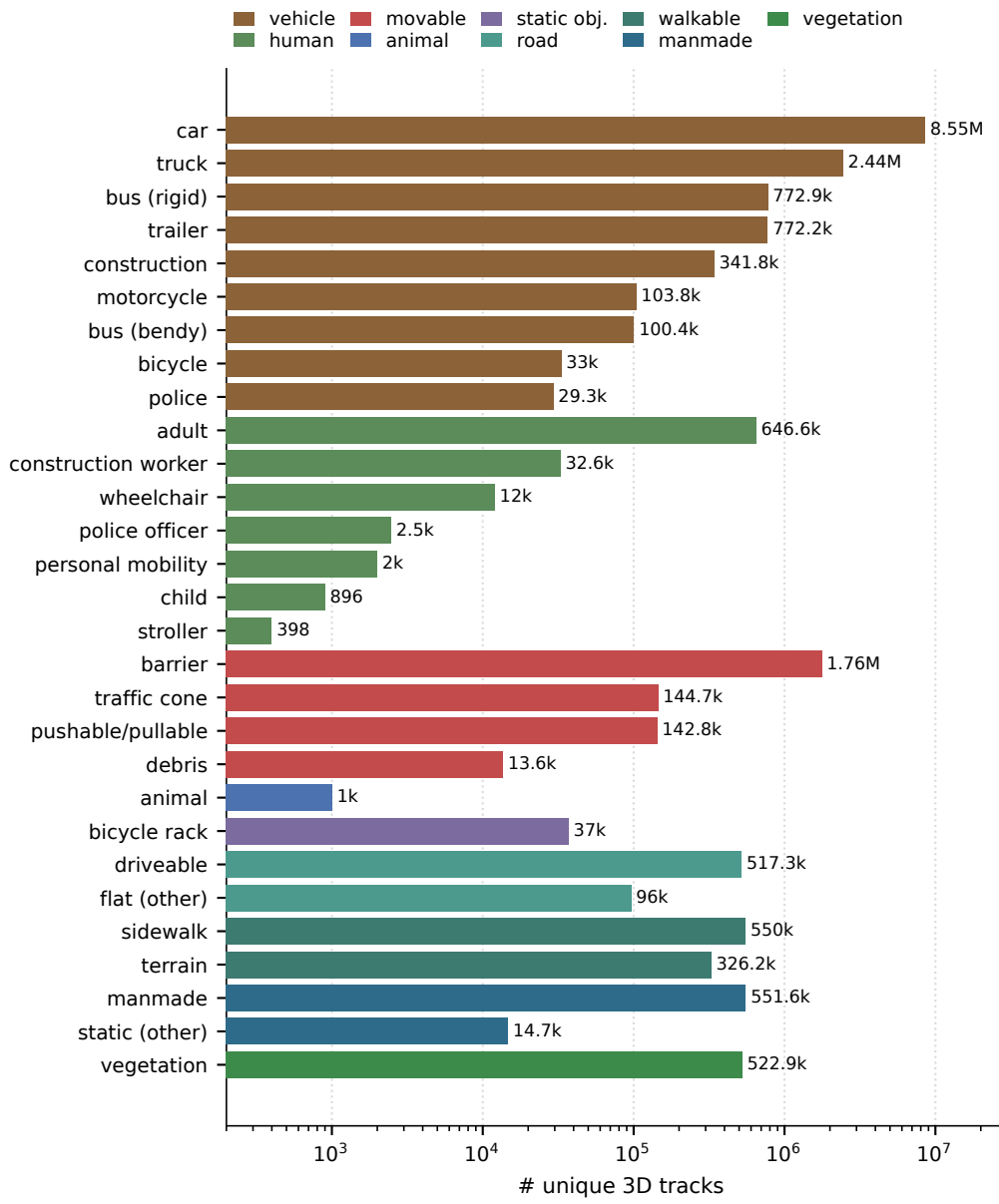


Figure 9: Statistics of unique 3D tracks by fine-grained nuScenes semantic class in NuscTrack, on a log-$x$ axis.

noise; human $\tilde{r}$ is not a rigidity residual. The statistic still treats each instance as rigid (one radius for the clip), including humans. This is a geometric description of where the TAP samples sit, not a planning, occupancy, or annotation-error evaluation.

Table 8: Counts of query tracks and unique 3D tracks by fine-grained nuScenes class in the NuscTrack pool.

| Category | Query tracks | Unique 3D |
|---|---|---|
| vehicle.car | 9,547,329 | 8,547,795 |
| vehicle.truck | 2,772,092 | 2,439,553 |
| vehicle.bus.rigid | 850,445 | 772,887 |
| vehicle.trailer | 899,430 | 772,212 |
| vehicle.construction | 389,865 | 341,784 |
| vehicle.motorcycle | 117,924 | 103,758 |
| vehicle.bus.bendy | 114,287 | 100,383 |
| vehicle.bicycle | 36,858 | 33,024 |
| vehicle.emergency.police | 32,741 | 29,304 |
| human.pedestrian.adult | 750,086 | 646,650 |
| human.pedestrian.construction_worker | 36,171 | 32,620 |
| human.pedestrian.wheelchair | 13,286 | 11,992 |
| human.pedestrian.police_officer | 2,685 | 2,469 |
| human.pedestrian.personal_mobility | 2,344 | 1,989 |
| human.pedestrian.child | 1,257 | 896 |
| human.pedestrian.stroller | 468 | 398 |
| movable_object.barrier | 2,004,246 | 1,759,699 |
| movable_object.trafficcone | 163,244 | 144,710 |
| movable_object.pushable_pullable | 157,755 | 142,787 |
| movable_object.debris | 14,807 | 13,564 |
| animal | 1,158 | 1,000 |
| static_object.bicycle_rack | 42,266 | 37,017 |
| flat.driveable_surface | 554,857 | 517,290 |
| flat.other | 101,701 | 96,023 |
| flat.sidewalk | 595,356 | 549,998 |
| flat.terrain | 354,327 | 326,151 |
| static.manmade | 604,087 | 551,609 |
| static.other | 15,818 | 14,687 |
| static.vegetation | 564,870 | 522,853 |

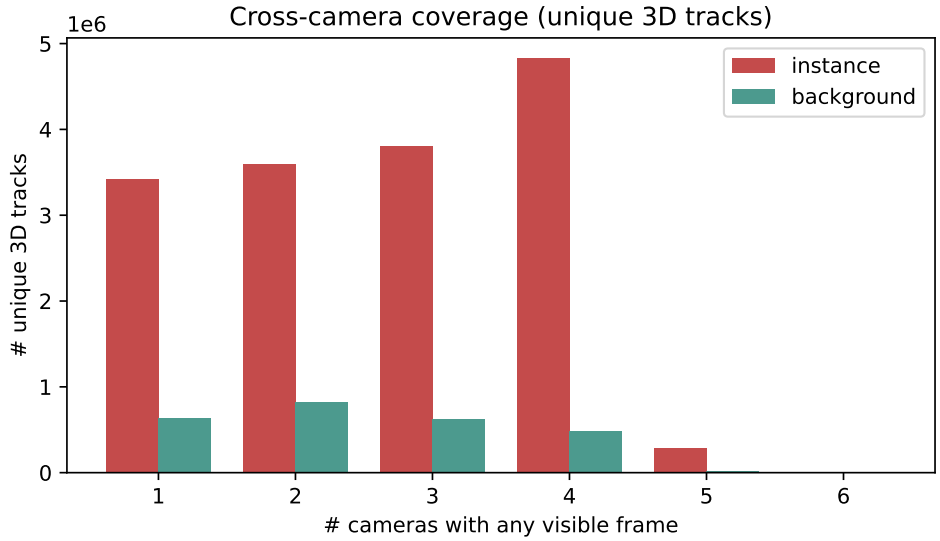


Figure 10: Statistics of how many surround cameras see each unique 3D track in NuscTrack, where the mode is four cameras and 78.1% of tracks are visible in at least two views.

Table 9: Summary, by object group, of where NuscTrack instance points sit inside the 3D box.

| Group | #instances | median $r$ (m) | median $\tilde{r}$ | share at cap 5000 |
|---|---|---|---|---|
| vehicle | 16,222 | 1.78 | 0.64 | 3.5% |
| human | 5,511 | 0.38 | 0.37 | $< 0.1\%$ |
| other | 9,186 | 0.54 | 0.52 | 0.2% |

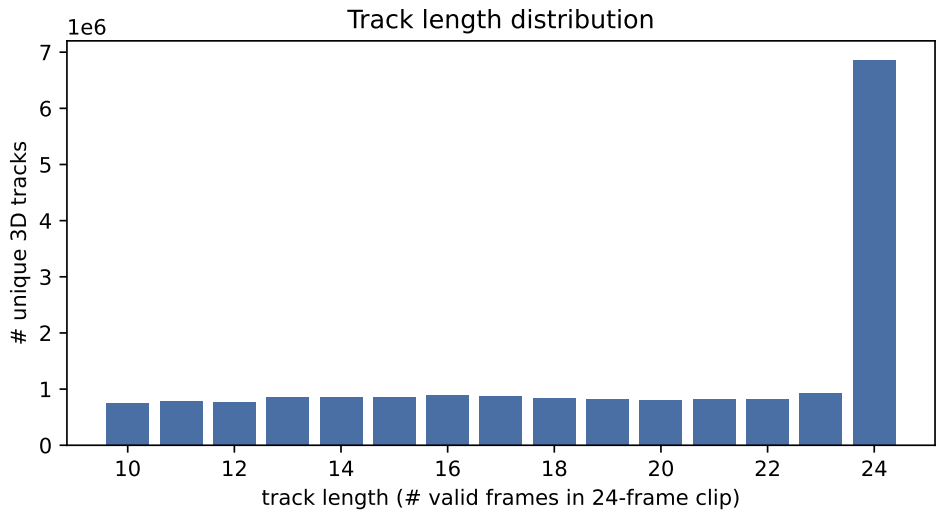


Figure 11: Statistics of track length for unique 3D tracks in NuscTrack, where tracks shorter than 10 frames are removed and the mean length is 19.35 frames.

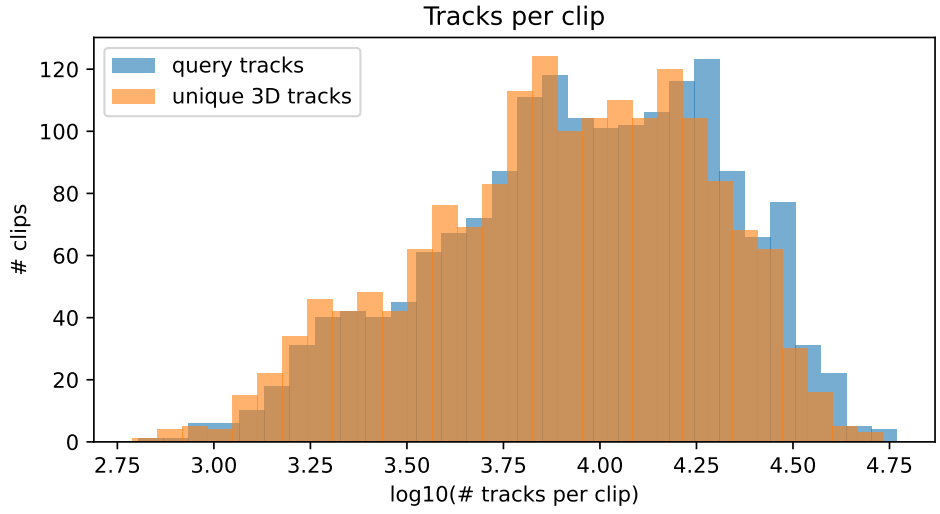


Figure 12: Statistics of how many tracks each clip contains in the NuscTrack pool, with means of 12,201 query tracks and 10,891 unique 3D tracks.

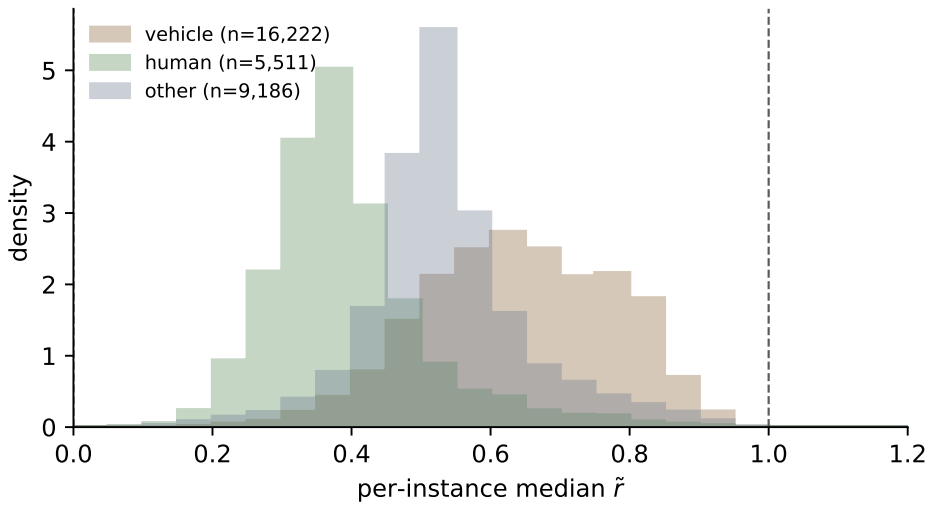


Figure 13: Statistics of where NuscTrack instance points sit inside the 3D box.

# F HUMAN AUDIT OF AUTOMATIC LABELS

The corpus is produced by the automatic recipe in Sec. 4.1. This appendix is a spot check of those labels, not a second annotation pass and not an inter-annotator study of the full corpus. Tracks come from the val 384-point evaluation sample in Sec. 4.2, counted as unique 3D points: several TAP-Vid-first queries that share one 3D track are not double-counted.

**Sample.** We inspect 16 val clips: 10 daytime-clear, 3 rain, and 3 night, using the same caption tags already counted in Appendix E. Each clip contributes 6 unique 3D tracks (3 instance, 3 background), 96 tracks in total (Table 10). Instance tracks oversample humans and movable objects relative to the vehicle-heavy pool. Background tracks split equally across road, walkable, manmade, and vegetation lidarseg groups.

Table 10: Composition of the human-audit sample.

| Split | Tracks |
|---|---|
| All | 96 |
| Instance | 48 |
| vehicle | 24 |
| human | 16 |
| movable | 8 |
| Background | 48 |
| road | 12 |
| walkable | 12 |
| manmade | 12 |
| vegetation | 12 |

**What is shown.** Inspection uses the stored 2D tracks and per-camera visibility flags, not a second geometric projection. For each track we open the TAP-Vid first query $(cam, t, x, y)$ and two later timestamps: the midpoint of the remaining valid span and the last valid frame. At each timestamp a camera is drawn only when the point has positive depth and a pixel inside the image. Visibility true draws a circle; visibility false draws a cross. Occluded points still project: a cross on the occluder is the intended mark, not a coordinate error.

**Decisions.** On the query frame the point must sit on a supported surface, not in the sky. On frames with visibility true, the circle must remain on the same physical point as the query; a deviation larger than 12 pixels is a location failure. At the mean ego distance in Appendix E that threshold is on the order of the $0.16$ m APD bin. Visibility is scored Agree, Disagree, or Uncertain. Disagree is a clearly visible point marked invisible, or a clearly occluded or out-of-image point marked visible. Grazing occlusions, glass, thin poles, and the $0.9$ completed-depth boundary are Uncertain and are not counted as failures. A track fails if any inspected in-image frame is a location failure or a visibility Disagree. Uncertain marks are counted separately and are not track failures.

**Outcomes.** Of the 96 tracks, 8 had a location failure and 13 had a visibility Disagree (Table 11). The two counts may overlap on the same track and are not added.

Table 11: Human-audit outcomes on 96 unique 3D tracks (track-level). A track may appear in both rows.

| Outcome | Tracks |
|---|---|
| Location failure | 8 |
| Visibility Disagree | 13 |

# G DATASET CARD

**Motivation and intended use.** NuscTrack is a research benchmark for cross-view 3D Track-Any-Point on a six-camera driving rig. It is not a planning, collision, or onboard-detection dataset, and it is not a substitute for 3D boxes or occupancy.

**Source.** Images, LiDAR, 3D boxes, lidarseg, and official train/val scene lists come from nuScenes (Caesar et al., 2020). This release adds densified 3D tracks, per-camera visibility, clip filters, and a fixed evaluation query list. A human audit of val evaluation queries is in Appendix F. nuScenes test scenes are unused because they have no public boxes.

**License.** NuscTrack tracks and queries inherit the nuScenes terms (CC BY-NC-SA). Users must also comply with the nuScenes Terms of Use for the underlying sensor data. Annotation, conversion, and evaluation code will be released upon acceptance; a public URL will replace this placeholder.

**Collection and PII.** There is no new capture. Street-view faces and license plates, if present, are those already in nuScenes; this annotation path does not add identity labels or extra cameras.

**Splits and leakage.** Train and val follow the official nuScenes 700/150 scene split. TAP supervision for methods trained on NuscTrack uses only the train split; evaluation reads the exported query list rather than resampling. BEVDet initialization for TAPNet-BEV and CoTracker-BEV is a nuScenes *detection* checkpoint on the same train scenes: it has seen those images and boxes, not NuscTrack TAP labels.

**Maintenance.** Code, the query list, and this card will be updated if a later export changes the filters or the visibility rule; older exports should not be mixed with Table 4.

## H Additional evaluation tables

The query counts for Table 4 are 115,200 / 56,264 / 58,936 (All / instance / background). $\mathrm{pct_nonfinite_pred}$ is 0 on every filled main-table row.

## I Cross-view camera-handover failures

Figure 7 in Sec. 6.3 shows two handover fragments under a shared-track protocol. The `CAM_FRONT` panels in Figure 6 and Appendix D do not show whether a tracker keeps the same ego-3D identity when a point leaves the query camera and continues in an adjacent view—the setting that monocular DriveTrack-style TAP cannot score. This section screens val clips for strong ground-truth camera handover (high fraction of frames after the query where the point is no longer visible in the query camera but is visible in an adjacent camera), then draws the hardest fragments: the same $N{=}10$ handover-rich queries for CoTracker3+UniDepthV2, CoTracker-BEV, and MVTracker scratch, matched one-to-one across dumps by query-frame ego XYZ (maximum distance 0.5 m). Each row is one method. The left panel stitches two adjacent cameras in surround order (right-camera $u$ shifted by image width) so a trail can cross the seam continuously; blue is GT and orange is prediction. The right panel is the same tracks in the nuScenes ego frame ($x$ forward, $y$ left, $z$ up). Unless noted, 3D-AJ, MTE, OA, and camera-IoU below are *per-clip* scores on that named 24-frame fragment (full stratified sample), not the $N{=}10$ overlay alone and not Table 4.

**`scene-1069_former` (night handover failure).** This night fragment is among the strongest GT handovers in the screen (99.5% of valid post-query frames leave the query camera; mean adjacent-handover fraction 0.32) and among the weakest BEV 3D scores. Per-clip on `scene-1069_former`: CoTracker-BEV 3.6 3D-AJ / 11.33 m MTE with OA 65.7 and camera-IoU 0.51; scratch MVTracker 5.2 3D-AJ; CoTracker3+UniDepth 1.1 3D-AJ. In Figure 7 (left), GT trails continue from `FRONT_LEFT` into `FRONT` across the seam, while orange predictions often stop near the boundary or peel away in ego-3D—visibility can remain partially correct while the metric identity after the camera switch is already wrong. Low light compounds both appearance matching and depth, so the 2D stitch alone understates the ego-frame break.

**`scene-0912_former` (monocular gap on handover).** The same screen flags this rainy daytime clip for a large scratch-MV versus CoTracker3 gap under strong handover (99.2% leave-query; adjacent-handover fraction 0.22). Per-clip: scratch MVTracker 13.2 3D-AJ vs. CoTracker3+UniDepth 1.7, with CoTracker-BEV in between at 8.2 3D-AJ / 3.53 m MTE (OA 80.3,

Table 12: 2D tracking accuracy on the query camera for NuscTrack val, with each cell reported as All / instance / background.

| Method | 2D-AJ | 2D-APD | 2D-OA |
|---|---|---|---|
| CoTracker3 | 37.38 / 45.13 / 29.94 | 50.89 / 59.07 / 42.73 | 83.18 / 86.13 / 80.37 |

camera-IoU $0.81$). Figure 7 (right) matches that split: CoTracker3 trails that look plausible near the seam collapse or cluster in ego-3D, whereas multi-view methods keep longer orange paths closer to the blue GT. A query-camera-only lift cannot be asked to invent the adjacent view; the gap is exactly what xTAP3D is meant to expose.

**`scene-0632_former` (which-camera mismatch).** Per-clip OA on this fragment is still high for CoTracker-BEV ($82.6$), yet 3D-AJ is only $4.8$ (MTE $5.29$ m) and camera-IoU ($0.75$) lags the OA—the model can be roughly right about "visible somewhere" while wrong about which surround camera carries the point after the switch. In Figure 14 (top), GT crosses `FRONT`→`FRONT_RIGHT`; predictions that remain in the wrong panel or drift in ego-3D illustrate that any-view OA alone is an incomplete visibility diagnosis for xTAP3D (Finding 1).

**`scene-0906_former` (handover reference).** The same daytime follow used in Figure 6 / Appendix D also has substantial GT handover ($95.8\%$ leave-query). Per-clip CoTracker-BEV reaches $16.8$ 3D-AJ / $2.50$ m MTE with camera-IoU $0.85$—among the stronger handover clips in the screen—while CoTracker3+UniDepth remains at $6.1$ 3D-AJ. Figure 14 (bottom) is a reference row, not a solved case: multi-view orange trails follow GT across `FRONT_LEFT`|`FRONT` more often than on the night failure above, but background points and long ego travel still leave clear ego-3D residuals. Together, the four clips argue that surround evaluation must inspect adjacent-camera continuity and ego-3D jointly; a clean single-camera overlay is not sufficient.

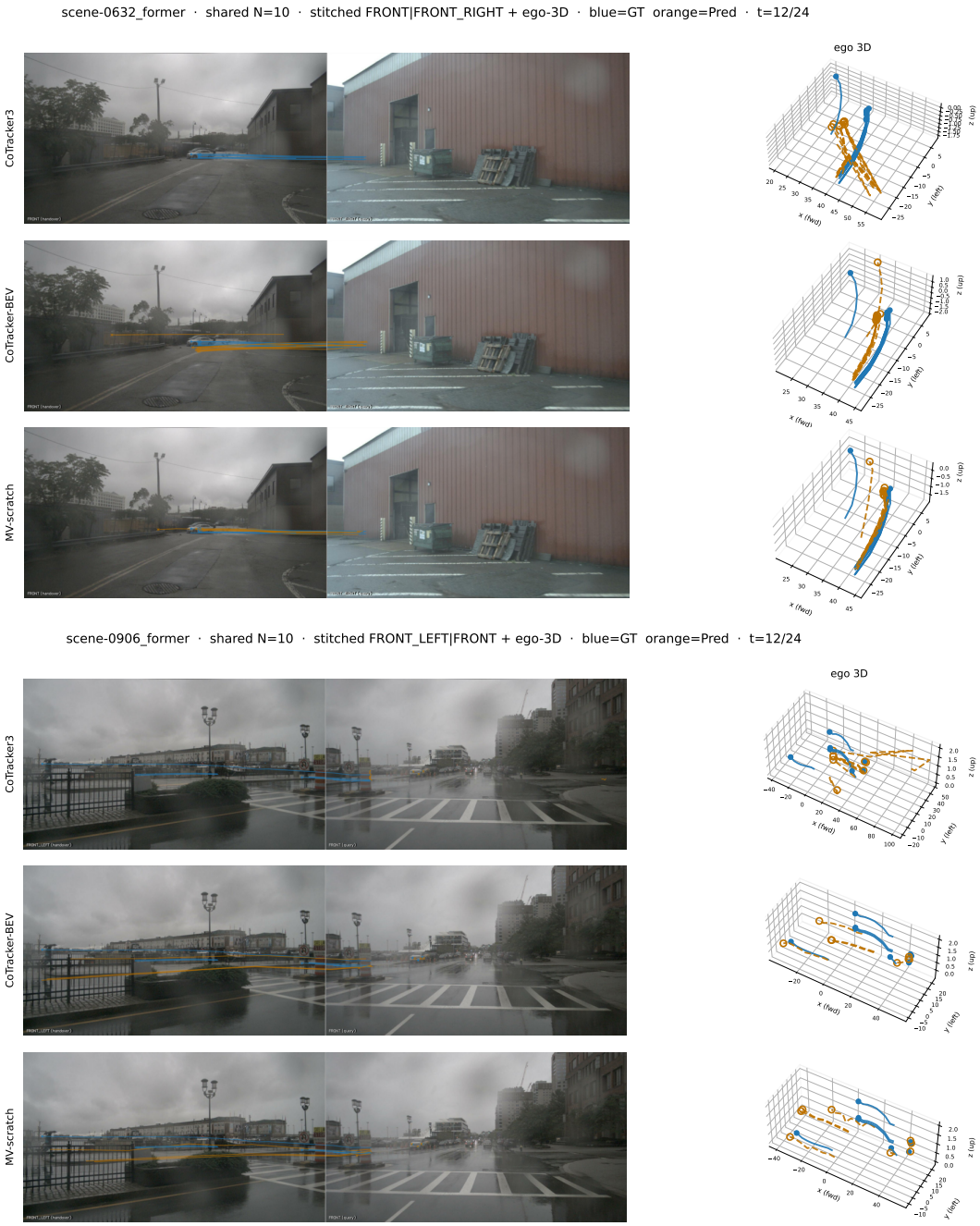


Figure 14: Further qualitative examples of cross-camera handover.