Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Hidden or Gone? Measuring How Much Visibility and Persistence Beliefs Can Improve Occlusion Handling in Online Multi-Object Tracking

Srilaxmi Dasari

srilaxmi_d@vnrvjiet.in

Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology

A Hariprasad Reddy

Geethanjali College of Engineering and Technology

A. Naresh

Sri Sai Institute of Technology and Science Autonomous

Anupam Vyas

Bundelkhand University

Neeraj Chandnani

Koneru Lakshmaiah Education Foundation

Pragati Dwivedi

Dr. Ambedkar Institute of Technology for Divyangjan

Research Article

Keywords: multi-object tracking, occlusion, uncertainty estimation, track management, oracle analysis, tracking-by-detection

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11472618/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Hidden or Gone? Measuring How Much Visibility and Persistence Beliefs Can Improve Occlusion Handling in Online Multi-Object Tracking

Srilaxmi Dasari[1*], A Hariprasad Reddy[2], A. Naresh[3], Anupam Vyas[4], Neeraj Chandnani[5], Pragati Dwivedi[6]

[1*]Department of CSE-(AI ML, IoT & RAI), VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India.
[2]Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, Hyderabad, India.
[3]Department of Computer Science and Engineering, Sri Sai Institute of Technology and Science - Autonomous, Rayachoti, India.
[4]Department of Electronics and Communication Engineering, IET, Bundelkhand University, Jhansi, India.
[5]Department of Electronics and Communication Engineering, Koneru Lakshmaiah (KL) College of Engineering, Koneru Lakshmaiah Education Foundation (KLEF), Deemed to be University, Vaddeswaram, Guntur, India.
[6]Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology for Divyangjan (AITD), Kanpur, India.

*Corresponding author(s). E-mail(s): srilaxmi_d@vnrvjiet.in;
Contributing authors: drhariprasadreddy.cse@gcet.edu.in; cse.naresh@ssits.ac.in; dr.anupam@bujhansi.ac.in; neerajchandnani@kluniversity.in; pragati.csjm@gmail.com;

Abstract

Occlusion causes most identity errors in online multi-object tracking, and a natural remedy is an explicit per-track belief about visibility. We measure how much such beliefs can help. We separate the two decisions through which occlusion enters tracking-by-detection, the motion update from a partly occluded detection and the fate of a lost track, learn a belief for each from observable cues, insert both into ByteTrack and OC-SORT, and compare them with oracles that know the true outcome. On MOT17 and MOT20, tuning on first halves and reporting on second halves of each sequence, both beliefs are predictable: heavy occlusion of a detected box at AUC 0.956 and 0.787, and whether a lost person will be re-detected at AUC 0.869–0.890. Few decisions change. Heavily occluded boxes are 2.3% of MOT17 detections, and acting on them leaves HOTA unchanged on both datasets, although on crowded MOT20 it cuts ByteTrack's identity switches by 12%. Lost tracks overwhelmingly belong to people who return, so a tuned memory length captures most of the attainable gain; oracle deletion adds at most 0.25 HOTA points and falls below it on MOT20. The remaining headroom is in re-association: buffered re-association raises ByteTrack's identity retention across detector gaps on MOT17 from

40.8% to 50.4%, beyond the tuned buffer, whereas OC-SORT, which already recovers lost tracks, barely responds. We also show how calibration and risk metrics can make a near-constant belief look good, and release code that regenerates every number.

Keywords: multi-object tracking, occlusion, uncertainty estimation, track management, oracle analysis, tracking-by-detection

1 Introduction

Occlusion is the dominant failure mode of online multi-object tracking. A person walking behind another person stops being detected, the track that followed them goes unmatched, and when the person reappears the tracker must decide whether the new detection continues the old trajectory or starts a new identity. Identity switches and fragmented trajectories concentrate around these moments. Recent trackers attack the problem from many directions: low-confidence detections [1], observation-centric motion recovery [2], pseudo-depth ordering [3, 4], occlusion-aware appearance [5, 6] and learned object permanence [7].

A natural idea, and the one behind the first version of this work, is to give each track an explicit, calibrated belief about whether its object is visible and to let that belief drive the occlusion decisions. The idea is attractive because it is interpretable and cheap. But a belief only helps if two things are true: it must be predictable from what the tracker can observe, and acting on it must change the decisions that cause the errors. Neither is usually measured. Occlusion-aware trackers report benchmark gains of the complete system. Visibility predictors report accuracy or calibration in isolation, and on a strongly imbalanced target a near-constant predictor can look excellent on both.

This paper measures both conditions directly, with the data-analysis discipline the question deserves. We separate the two decisions in which occlusion enters a tracking-by-detection pipeline. The first is whether a *detected* box is heavily occluded, which matters for the motion update. The second is whether a *lost* track's object is still present, which matters for track deletion and re-association. For each we learn a belief from observable cues, test how predictable it is, insert it into the official ByteTrack and OC-SORT code, and compare it with an oracle that knows the true answer. The oracle turns an open question ("would a better occlusion model help?") into a number: the best that any predictor of the same outcome could achieve under the same decision rule and association geometry.

The results, on the MOT17 and MOT20 training sets with a strict first-half/second-half protocol, overturn the premise of the original design and locate where the leverage actually is:

1. **Detection-level occlusion is predictable, but acting on it changes little.** The four box statistics of the submitted design predict heavy occlusion at AUC 0.571 on MOT17, barely above chance and with nearly constant output. A richer set of geometry, occluder-ordering and confidence cues reaches AUC 0.956 (MOT17) and 0.787 (MOT20). Heavily occluded boxes are only 2.3% of MOT17 detections, because the detector rarely fires on them, and down-weighting them in the Kalman update leaves HOTA unchanged on both datasets; on the crowded MOT20, where they are 25.7% of detections, it does cut ByteTrack's identity switches by 12%.
2. **Lost tracks are mostly people who come back.** Only 10.6% (ByteTrack) and 4.9% (OC-SORT) of loss episodes on MOT17, and 2.0% and 1.4% on MOT20, belong to people who leave the scene. Most belong to people the detector misses for a while.
3. **Whether a lost track will return is also predictable.** The persistence belief reaches AUC 0.869 (ByteTrack) and 0.870 (OC-SORT) on MOT17 and 0.890 and 0.890 on MOT20. Time lost and the geometry and motion of the predicted box carry most of the signal; occluder cues add little.
4. **Memory length is the first lever; deletion intelligence has little headroom.** Most of what the occlusion-related variants gain over the default trackers comes from memory length alone (up to +1.53 HOTA for ByteTrack on

MOT20). Even oracle deletion adds at most 0.25 HOTA over the best tuned fixed buffer, and on MOT20 it falls below it.

5. **The remaining headroom is in re-association, and only where the tracker lacks it.** Buffered re-association of lost tracks is the best non-oracle variant for ByteTrack. On MOT17 it beats the tuned buffer in identity retention across detector gaps (+5.6 points, $p = 0.020$), and oracle re-association (evaluated on MOT17 only) shows further headroom; on MOT20 its margin over the buffer is small. The belief gate itself contributes little: the purely geometric version performs almost identically. OC-SORT, which already re-associates lost tracks with their last observation, barely responds to any hook (Section 5.4).

Beyond tracking, the study is a case study in how a plausible model can pass a weak evaluation. Calibration error is uninformative against a near-constant predictor, and a risk proxy of the form $(1 - p) \cdot c$ rewards predicting $p \to 1$. Both flaws were present in the submitted version, and both are avoided here by reporting prevalence-only references, threshold-free metrics and oracle bounds.

The paper is organised as follows. Section 2 reviews occlusion handling in online tracking. Section 3 formulates the two decisions and the VIS-Belief hooks. Section 4 gives the protocol. Section 5 reports the results, and Section 6 discusses implications and limitations.

2 Related work

Online tracking-by-detection.

Most online multi-object trackers follow the tracking-by-detection recipe introduced by SORT: a Kalman filter predicts each track [8], detections are assigned to predictions by a bipartite matching such as the Hungarian algorithm [9], and a track that stays unmatched for a fixed number of frames is deleted [10]. DeepSORT adds appearance embeddings to that cost [11]. ByteTrack keeps low-confidence detections, many of which belong to partly occluded people, and associates them in a second stage [1]. OC-SORT re-estimates the motion state from observations after a gap and re-associates a lost track with its last observation (observation-centric recovery), targeting precisely the drift that accumulates while an object is occluded [2]. BoT-SORT, StrongSORT, Hybrid-SORT and BoostTrack strengthen the association cost with camera-motion compensation, appearance, weak geometric cues or detection-confidence boosting [12–15], and buffered IoU enlarges the matching region to tolerate irregular motion [16]. In all of these systems the lifetime of a lost track is a fixed hyper-parameter (typically 30 frames), and whether the object is occluded or has left the scene is never represented explicitly. We build on the official ByteTrack and OC-SORT implementations and change only the occlusion-related decisions, so that every difference we measure can be attributed.

Occlusion-aware association.

A growing number of trackers state occlusion as their main target. SparseTrack and PD-SORT order objects by pseudo-depth derived from the bottom edge of the box and associate them depth-layer by depth-layer [3, 4]; the same ground-plane ordering underlies the occluder cues we use. OccluSort, OA-SORT and occlusion-aware embedding learning adapt appearance features to the visible part of a person [5, 6, 17]; SMILEtrack, PSMOT and occlusion-calibrated re-identification networks change the detector or the embedding [18–20]; and recent association schemes add occlusion-dependent costs, motion predictors or distributional observations [21–23]. These works report benchmark-level gains of the complete system. They do not separate how much of the gain comes from knowing that an object is occluded, nor how much a perfect occlusion signal could deliver at most. That separation is the question this paper answers.

Object permanence and amodal tracking.

Tokmakov et al. learn object permanence end to end, so that a tracker can keep localising an object it cannot see [7], and TAO-Amodal and OVIS provide benchmarks with heavily occluded objects [24, 25]. End-to-end transformer trackers carry identity through learned queries [26, 27]. These approaches change the detector or the whole model; our scope is the lighter case in which detections are given and only the track-management layer may change, which is how most deployed online trackers operate.

Evaluation and probability quality.
CLEAR-MOT, IDF1 and HOTA measure tracking from complementary angles [28–30]; we report all three through TrackEval [31] on the MOTChallenge benchmarks [32–34]. For the belief models we follow the probability-forecasting literature: proper scoring rules such as the Brier score [35], calibration analysis [36, 37] and, because occlusion is a rare class among detections, precision–recall rather than accuracy [38]. A calibration error is uninformative when the predictor is nearly constant: a model that always outputs the base rate is perfectly calibrated and useless. We therefore report every probability metric next to the corresponding prevalence-only reference.

3 Problem formulation and method

3.1 Two occlusion decisions in tracking-by-detection

An online tracker processes frame t with detections $\mathcal{D}_t = \{d_{t,j}\}$ (boxes $b_{t,j}$, confidences $s_{t,j}$) and a set of tracks. Occlusion enters the tracker through two different decisions.

Detection-level. A detection that is matched to a track updates the track's motion state. If the person is partly hidden, the box is truncated and the update can corrupt position and velocity. The relevant quantity is the probability that the detected box is heavily occluded,

$$q_{t,j} = \Pr\big(v_{t,j} < 0.5 \mid \boldsymbol{\phi}_{t,j}\big)\,, \tag{1}$$

where $v_{t,j} \in [0,1]$ is the annotated visible fraction of the matched person and $\boldsymbol{\phi}_{t,j}$ are cues observable in frame t.

Track-level. A track i that receives no detection becomes *lost*. The tracker must decide every frame whether to keep it and whether a new detection may re-activate it. Both decisions depend on whether the person is still present but undetected, or gone. We define the *persistence belief*

$$\begin{aligned} \pi_{i,t} &= \Pr\big(R_{i,t} = 1 \mid \boldsymbol{\psi}_{i,t}\big)\,, \\ R_{i,t} &= \mathbb{1}[\,i \text{ re-detected in } (t,\ t + T_{\max} - g_{i,t}]\,]\,, \end{aligned} \tag{2}$$

where $g_{i,t}$ is the number of frames the track has been lost, $T_{\max}$ is the hard memory cap, and $\boldsymbol{\psi}_{i,t}$ are the lost-state cues at frame t. $\pi_{i,t}$ is thus the probability that keeping the track until the cap can still pay off. It is re-evaluated every frame from the current cues, so it propagates through the occlusion interval without a hand-set decay rate: it stays high while a closer person covers the predicted position and falls once the predicted box leaves the image or nothing explains the absence.

The submitted version of this work estimated a single per-track visibility probability from four box statistics and used it to lengthen the deletion age by up to 20 frames. Section 5.1 shows why that design could not work; the formulation above separates the two decisions that design conflated.

3.2 Cues

All cues are computed from the current frame's detections, the tracker's own state and label-free sequence statistics; none uses ground truth.

Detection cues $\boldsymbol{\phi}$ fall into four families (Table 1). Geometry describes the box itself. The scene cue compares the box height with the height expected at its ground-contact row, from a robust linear fit of $\log h$ on the bottom-edge coordinate over the confident detections of the sequence's training half. Occluder cues use the ground-plane ordering exploited by SparseTrack and PD-SORT [3, 4]: a detection whose bottom edge is lower in the image is assumed to be closer to the camera, and the fraction of the box covered by such detections measures front occlusion. The last family is the detector confidence. The submitted manuscript used four statistics only (log area, aspect ratio, frame density and maximum IoU); we keep them as the *legacy* cue set.

Lost-state cues $\boldsymbol{\psi}$ describe time lost g/f_s (with frame rate f_s); the track history (number of hits, mean and last confidence, number of previous losses, q of the last detection); the predicted box (fraction inside the image, signed distance to the nearest border); motion (velocity towards the nearest border in body heights per second, and the implied time to exit); and occluders at frame t (coverage of the predicted box by closer tracked boxes and by closer detections, coverage by boxes behind it, and the number of tracks within two body heights).

Table 1 Observable cues. Detection cues feed q (Eq. 1); lost-state cues feed π (Eq. 2).

Model	Family	Cues
q	geometry	$\log(h/H)$, aspect w/h, bottom row y_b/H, border distance, truncation flag
	scene	perspective residual $\log(h/H) - (\hat{a} + \hat{b}\, y_b/H)$
	occluder	max. and summed coverage by closer detections, coverage by farther detections, number of overlapping detections, max. IoU, $\log(1 + \lvert\mathcal{D}_t\rvert)$
	score	detector confidence s
	legacy	log area, aspect, $\log(1 + \lvert\mathcal{D}_t\rvert)$, max. IoU (submitted version)
π	time	g/f_s
	history	$\log(1 + \text{hits})$, mean and last confidence, previous losses, last q
	geometry	fraction of predicted box in image, border distance, $\log(h/H)$
	motion	outward speed (heights/s), time to exit
	occluder	coverage by closer tracks / detections, coverage by farther tracks, tracks nearby

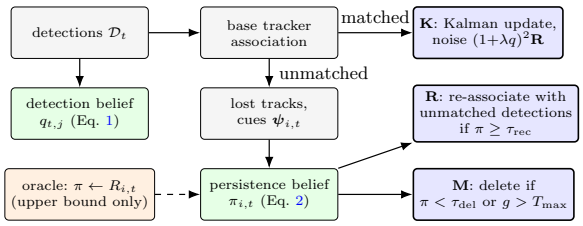


Fig. 1 Where VIS-Belief enters a tracking-by-detection loop. Blue boxes are the switchable hooks and green boxes the learned beliefs. K uses q of the matched detection. The oracle replaces π by the true outcome to bound what any persistence belief can achieve

3.3 Estimating the beliefs

Both q and π are gradient-boosted tree classifiers (LightGBM [39]; 300–400 trees, 31 leaves, learning rate 0.05) followed by Platt scaling [40]. To keep the calibration honest, the Platt map is fitted on out-of-fold scores, with folds defined by sequence. Labels for q come from matching public detections to annotated boxes (Hungarian, IoU ≥ 0.5). Labels for π come from running the base tracker with a 10 s buffer, assigning each track the ground-truth identity it covers most often over its whole life, and checking Eq. 2 inside the same data half. A negative whose look-ahead window extends past the end of the half is right-censored, because the tracker cannot observe the return either, and is excluded from fitting and testing. Positives are always kept.

3.4 VIS-Belief: three hooks into an online tracker

VIS-Belief modifies only the occlusion-related decisions of a base tracker (Fig. 1). Each hook can be switched on independently.

M — belief memory. A lost track is removed when

$$\pi_{i,t} < \tau_{\text{del}} \quad \text{or} \quad g_{i,t} > T_{\max}, \tag{3}$$

replacing the fixed buffer of the base tracker.

R — belief-gated re-association. After the base tracker's own association stages, still-unmatched confident detections may re-activate a lost track with $\pi_{i,t} \geq \tau_{\text{rec}}$. The match score is a buffered IoU [16], taken as the larger of the scores against the predicted box and against the last observed box, with a margin $m(g) = \mu \min(1 + g/f_s, 3)$ that grows with the time lost. Pairs whose height ratio falls outside $[0.67, 1.5]$ are rejected, and the remaining pairs are assigned by the Hungarian algorithm with acceptance threshold θ_{rec}. Setting $\tau_{\text{rec}} = 0$ gives the purely geometric version (R_{geom}), which isolates the contribution of the belief gate.

K — occlusion-aware update. When a detection j updates a track, the measurement noise of the Kalman filter is inflated to $(1 + \lambda q_{t,j})^2\mathbf{R}$, so a probably-occluded box moves the state less.

Oracle variants. Replacing π by the true $R_{i,t}$ gives oracle memory and oracle re-association. Because the oracle must act online, it identifies the person behind a lost track by a majority vote over the track's last ten observed boxes (IoU ≥ 0.5 with the annotation), rather than over the track's whole life as for the training labels, and then looks ahead in the ground truth. Oracles are not methods. They give the best result any predictor of $R_{i,t}$ could achieve with the same decision rule and association geometry, which is the ceiling for every persistence belief of this kind. They are not a ceiling for all possible policies: a track whose person is never again matched by a public detection at IoU ≥ 0.5 has $R_{i,t} = 0$, yet keeping it can still help when a looser detection re-acquires it.

Legacy rule. For comparison we implement the submitted manuscript's rule on top of each base tracker: the deletion age is $A_0 + \mathrm{round}((1 - \bar{p})B)$ with $A_0 = 30$ frames, $B \in \{20, 60\}$ and $\bar{p}$ an exponentially smoothed detection-level visibility ($\bar{p} \leftarrow 0.62(1 - q) + 0.38\bar{p}$).

Cost.

Per frame, the hooks evaluate one tree ensemble on the lost tracks (and one on the detections) and one buffered-IoU matrix between lost tracks and unmatched detections. This is small next to detection and leaves the tracker online.

4 Experimental protocol

Data.

We use the seven MOT17 training sequences with the public Faster R-CNN detections [32, 41] and the four MOT20 training sequences with their public detections [33]. Both datasets annotate a visible fraction for every box, which is what makes occlusion measurable. They differ sharply in density: 112,297 valid pedestrian boxes over 5,316 frames in MOT17 against 1,134,614 over 8,931 frames in MOT20. The MOT20 public detection files carry a 0/1 column instead of a confidence; boxes marked 0 match a ground-truth person as often as boxes marked 1, so every MOT20 detection is treated as confidence 1. This disables ByteTrack's low-score stage and the confidence cues on MOT20, and we interpret MOT20 results with that in mind.

Splits.

Each sequence is split in time into a first and a second half, the convention used by ByteTrack and OC-SORT for validation on MOTChallenge training data [1, 2]. Every model, threshold and hyper-parameter is chosen on the first halves; every number reported as a result comes from the second halves, evaluated once. To make that possible without leaking labels into tuning, the belief models used while tuning on the first halves are cross-fitted: the model applied to sequence k is trained on the first halves of the other sequences. The detection-level model is additionally evaluated leave-one-sequence-out on complete sequences.

Trackers and variants.

The base trackers are the official ByteTrack and OC-SORT code with repository defaults (ByteTrack: confidence threshold 0.6, buffer 30 frames at 30 fps, match threshold 0.8, the value of its demo configuration; the evaluation script defaults to 0.9; OC-SORT: detection threshold 0.6, maximum age 30 frames, IoU threshold 0.3, $\Delta t = 3$, inertia 0.2, IoU cost). Because they run on the public Faster R-CNN detections rather than on the trackers' own YOLOX detections, absolute scores are not comparable with published leaderboard numbers; only differences between variants are interpreted. Both share the same output filter (box area > 10 pixels, $w/h \leq 1.6$). On each base tracker we compare eleven variants: the default; the best fixed buffer from $\{30, \ldots, 300\}$ frames, which controls for memory length alone; the legacy rule; K; M; $\mathrm{R}_{\mathrm{geom}}$; R; M+R; M+R+K; oracle M; and oracle M+R. Every variant has its own small grid (Appendix A), and the grid point with the highest first-half HOTA is the one evaluated on the second halves. The grids differ in size (1 to 24 points); the best fixed buffer, the main control, is searched over eight values from 1 to 20 s, and no selected buffer lies at the edge of its grid.

Metrics.

Belief models are scored with ROC-AUC, average precision (AUPRC), Brier skill score against the prevalence-only forecast, and expected calibration error (15 bins); 95% intervals come from a cluster bootstrap over identities or tracks [42]. Tracking is scored by TrackEval [31] (HOTA, DetA, AssA, IDF1, MOTA, ID switches, fragmentations).

Occlusion events.

Aggregate metrics dilute what happens during occlusion, so we also score the events that matter. A *detector-gap event* is a ground-truth identity that is detected, is then annotated but not covered by any public detection for at least two consecutive frames, and is detected again. We stratify events by gap length (< 0.5 s, 0.5–2 s, ≥ 2 s). An event counts as *retained* when the track matched to the identity just before the gap is also the first track matched to it within 15 frames after re-detection. For each base tracker, all variants are scored on the same events, those whose pre-gap frame every variant matches, and compared

with an exact McNemar test [43] and a cluster bootstrap over identities. McNemar p-values against the default are Holm-corrected over all comparisons in a dataset; we also report the paired difference to the best fixed buffer. Per-sequence HOTA differences are reported as wins and losses over the sequences.

Reproducibility.

All runs are deterministic CPU runs on a 12-core desktop. The code, the exact commits of the vendored ByteTrack, OC-SORT and TrackEval sources, every configuration and every result file are released with the paper (see Declarations). All numbers in this manuscript are generated from those result files.

5 Results

5.1 Detection-level occlusion is predictable; how common it is depends on the scene

Table 2 and Fig. 2 answer the first question. The unit is a second-half detection that matches an annotated person (88.9% of MOT17 detections do). The legacy cues of the submitted version, with the same logistic model and no class weights, predict heavy occlusion on MOT17 at AUC 0.571 and AUPRC 0.030, against a prevalence of 2.3%. After Platt recalibration, which every model in Table 2 receives, its predicted probability barely moves ($\sigma_q = 0.008$) and its calibration error (ECE 0.011) is that of the prevalence-only forecast (0.012). Without recalibration, as in the submitted version, the output varies somewhat more ($\sigma_q = 0.040$) but its Brier skill is negative (-0.064): it forecasts worse than the base rate. Either way, a small calibration error on this target says nothing about discrimination, which is how the submitted manuscript could report near-perfect calibration for a near-useless predictor.

With the full cue set and gradient boosting, heavy occlusion is predicted at AUC 0.956 (95% CI 0.943–0.968) and AUPRC 0.488, a Brier skill of 0.303. It holds on sequences never seen in training (leave-one-sequence-out AUC 0.766–0.971). On MOT20, where people stand much closer together, heavy occlusion is common among detections (25.7%) and the full model reaches AUC 0.787 and AUPRC 0.638. Which cues matter differs between the datasets. On MOT17 the detector confidence is indispensable: without it AUC drops to 0.905 and Brier skill to 0.154. On MOT20, where the public detections carry no confidence, the occluder-ordering cues are: without them AUC drops to 0.651 and AUPRC to 0.388. The scene (perspective) cue helps on neither dataset.

On MOT17, however, only 2.3% of matched detections are heavily occluded, because the public detector rarely fires on a person who is mostly hidden. Occlusion therefore reaches a MOT17 tracker mainly as *missing* detections, not as degraded ones. On MOT20 a quarter of detections are heavily occluded, and Section 5.4 shows that this is where acting on q (hook K) has an effect.

5.2 Lost tracks are mostly people who come back

Table 3 and Fig. 3 classify every loss episode of the base trackers (run on both halves with a 10 s buffer) by what the annotated person does next. On MOT17, 65.0% of ByteTrack losses are people detected again within 3 s and a further 5.1% later; 13.4% stay in the scene but are never detected again. Only 10.6% (ByteTrack) and 4.9% (OC-SORT) belong to people who leave the scene. False tracks account for 5.9% and 26.9%; the share is larger for OC-SORT because it starts tentative tracks more readily. On MOT20 exits are rarer still (2.0% and 1.4%), and 87.9% of ByteTrack losses are re-detected within 3 s. Most gaps are short (median 0.2 s on MOT17 and 0.1 s on MOT20 for ByteTrack), but the 90th percentile of the return gap (2.2 s and 1.4 s) exceeds the default ByteTrack buffer of about 1 s. A default buffer therefore deletes a tail of tracks whose person is still in the scene. This explains why simply keeping tracks longer helps (Section 5.4), and it suggests that the useful question is less "has the person left?" than "will the returning person be recognised as the same identity?".

5.3 Persistence is predictable too

Table 4 reports how well the lost-state cues predict whether the person will be re-detected before the 10 s cap. On MOT17 the full belief reaches AUC 0.869 for ByteTrack and 0.870 for OC-SORT, with

Table 2 Detection-level occlusion ($v < 0.5$) on the second halves. AUPRC of the prevalence-only row is the prevalence. BSS: Brier skill score vs. the second-half prevalence; the prevalence-only row forecasts the first-half prevalence, hence its BSS is slightly negative. All learned rows are Platt-recalibrated; σ_q: standard deviation of the predicted probability. Units are second-half detections that match an annotated person.

Data	Model	AUC	AUPRC	BSS	ECE	σ_q
MOT17	Prevalence only	–	0.023	−0.006	0.012	0.000
	Legacy cues, logistic (submitted)	0.571	0.030	−0.004	0.011	0.008
	Legacy cues, logistic, balanced	0.622	0.033	−0.003	0.010	0.013
	Legacy cues, GBM	0.825	0.113	0.042	0.012	0.033
	All cues, logistic	0.879	0.329	0.185	0.014	0.068
	All cues, GBM (used)	0.956	0.488	0.303	0.007	0.079
	without geometry	0.942	0.387	0.188	0.017	0.043
	without scene	0.958	0.517	0.327	0.008	0.084
	without occluder	0.951	0.380	0.236	0.007	0.064
	without score	0.905	0.424	0.154	0.015	0.034
MOT20	Prevalence only	–	0.257	−0.001	0.014	0.000
	Legacy cues, logistic (submitted)	0.681	0.409	0.011	0.081	0.009
	Legacy cues, logistic, balanced	0.683	0.411	0.026	0.050	0.026
	Legacy cues, GBM	0.689	0.432	0.055	0.057	0.055
	All cues, logistic	0.777	0.621	0.213	0.065	0.189
	All cues, GBM (used)	0.787	0.638	0.224	0.070	0.177
	without geometry	0.762	0.618	0.207	0.060	0.175
	without scene	0.787	0.637	0.212	0.085	0.175
	without occluder	0.651	0.388	0.040	0.033	0.061
	without score	0.787	0.638	0.224	0.070	0.177

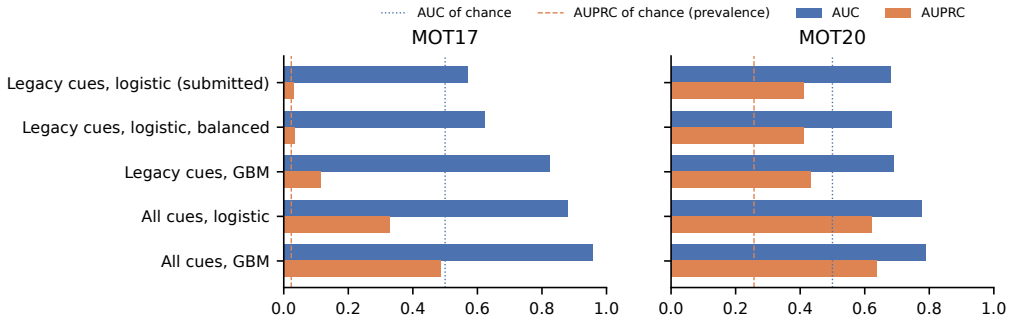


Fig. 2 Detection-level occlusion prediction on the second halves. The legacy cues of the submitted version sit near chance on MOT17; the full cue set is strongly discriminative on both datasets. Dashed lines mark the chance level of each metric

Brier skill 0.324 and 0.359; on MOT20 it reaches 0.890 and 0.890. Time lost alone already gives 0.759 and 0.810 on MOT17. Adding the geometry and motion of the predicted box (whether it is leaving the image, and how fast) raises this to 0.812 and 0.823, and the track history adds a further step (0.861 and 0.860 without occluder cues). The occluder-coverage cues add little on top (0.869, 0.870). Two features of the ablation call for caution. Track history on its own, added to time lost, *lowers* AUC on all four tracker–dataset pairs, although it helps in combination with geometry; and the richer models are less well calibrated than the time-only model. Both point to some over-fitting or shift between halves, and we read the ablation as ranking cue families rather than as precise attributions. The labels exclude right-censored negatives, i.e. lost states whose look-ahead window ends after the half: 34% of MOT17 lost states and 6% of MOT20 lost states. Because positives are always kept, this raises prevalence and can inflate AUC on the short MOT17 halves; the MOT20 AUCs, with little censoring, are similar, which suggests the effect is small. Section 5.4 shows that, as for detection-level occlusion, being predictable is not the same as being useful.

Table 3 What happens to the person behind a lost track (loss episodes of the base tracker with a 10 s buffer, both halves). Shares in %; return gaps in seconds for identities that are detected again.

Data	Tracker	Episodes	Back ≤3 s	Back >3 s	Never re-det.	Exited	False track	Median gap	90th pct.
MOT17	ByteTrack	649	65.0	5.1	13.4	10.6	5.9	0.2	2.2
MOT17	OC-SORT	751	52.9	6.1	9.2	4.9	26.9	0.1	3.0
MOT20	ByteTrack	2951	87.9	5.1	1.9	2.0	3.2	0.1	1.4
MOT20	OC-SORT	2891	81.0	8.4	1.2	1.4	8.0	0.1	2.6

Table 4 Predictability of the persistence belief π (Eq. 2) on second-half lost states. AUC with 95% cluster-bootstrap interval over tracks; BSS: Brier skill vs. prevalence.

Data	Tracker	Cues	AUC [95% CI]	AUPRC	Prev.	BSS	ECE
MOT17	ByteTrack	time lost only	0.759 [0.729, 0.790]	0.733	0.471	0.198	0.032
		time + history	0.715 [0.659, 0.762]	0.693	0.471	0.136	0.035
		time + geometry/motion	0.812 [0.761, 0.858]	0.742	0.471	0.254	0.096
		all but occluder	0.861 [0.821, 0.896]	0.825	0.471	0.315	0.131
		all cues	0.869 [0.829, 0.901]	0.836	0.471	0.324	0.135
	OC-SORT	time lost only	0.810 [0.785, 0.840]	0.679	0.318	0.253	0.053
		time + history	0.773 [0.714, 0.824]	0.615	0.318	0.184	0.061
		time + geometry/motion	0.823 [0.779, 0.859]	0.622	0.318	0.213	0.106
		all but occluder	0.860 [0.817, 0.896]	0.743	0.318	0.335	0.062
		all cues	0.870 [0.826, 0.907]	0.758	0.318	0.359	0.065
MOT20	ByteTrack	time lost only	0.797 [0.786, 0.808]	0.818	0.550	0.260	0.026
		time + history	0.762 [0.748, 0.778]	0.788	0.550	0.172	0.077
		time + geometry/motion	0.871 [0.857, 0.885]	0.876	0.550	0.381	0.103
		all but occluder	0.881 [0.869, 0.894]	0.890	0.550	0.413	0.083
		all cues	0.890 [0.879, 0.902]	0.901	0.550	0.429	0.086
	OC-SORT	time lost only	0.823 [0.815, 0.831]	0.788	0.478	0.313	0.015
		time + history	0.794 [0.782, 0.806]	0.738	0.478	0.245	0.061
		time + geometry/motion	0.876 [0.865, 0.888]	0.840	0.478	0.403	0.081
		all but occluder	0.882 [0.871, 0.893]	0.852	0.478	0.420	0.072
		all cues	0.890 [0.879, 0.900]	0.865	0.478	0.424	0.081

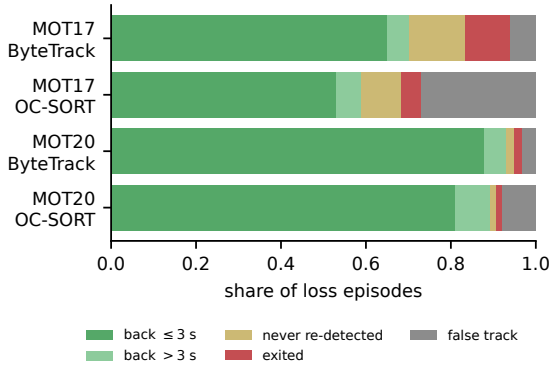


Fig. 3 Fate of the person behind each loss episode (both halves). Exits are rare; most lost tracks belong to people who are detected again, the large majority within 3 s

5.4 Tracking: memory length explains most of the gain

Table 5 gives the tracking results and Fig. 4 shows each variant's HOTA change against the default tracker. The first, second and fourth observations below hold for both base trackers; the third holds for ByteTrack only.

First, memory length alone is the largest single effect. The best fixed buffer, tuned on the first halves, adds +0.62 HOTA to ByteTrack and +0.59 to OC-SORT on MOT17, consistent with the anatomy above.

Second, the learned deletion belief does not beat a well-tuned buffer. M reaches 50.20 and 50.18 HOTA, i.e. −0.07 and −0.19 relative to the best buffer. Even oracle deletion adds only +0.22 and +0.25. Deletion policy has little headroom once the buffer is long enough.

Third, for ByteTrack, buffered re-association adds to the buffer. Belief-gated re-association (R) is the best non-oracle ByteTrack variant: 50.61 HOTA (+0.96 over the default, +0.34 over the best buffer), AssA 60.46 against 58.22, and 195 identity switches against 230 (−15%). The purely geometric R_{geom} reaches 50.50: almost all of the gain comes from the buffered matching itself, and the belief gate adds about a tenth of a HOTA point. Oracle M+R reaches 51.18, the highest

value in the table; this is the only setting in which a better persistence signal is worth more than half a HOTA point. OC-SORT already re-associates lost tracks with their last observation (observation-centric recovery), and on it R adds nothing beyond the buffer (−0.16); its oracle headroom is also small (+0.32).

Fourth, the submitted rule and the Kalman hook leave HOTA essentially unchanged. The legacy rule changes HOTA by +0.39 and +0.21, less than a tuned buffer, and K by +0.05 and −0.12. Single-run HOTA differences of a few hundredths of a point are within the variation between sequences (Table 6, W/L column), and we do not interpret them.

MOT20. The crowded MOT20 scenes repeat the first two observations and sharpen them (Table 5, lower part; reduced grid without M+R and oracle M+R). For ByteTrack, memory length is worth even more: the best buffer adds +1.53 HOTA, and re-association adds only +0.08 on top of it. The learned and the oracle deletion beliefs both fall below the tuned buffer (−0.59 and −0.41), so even perfect knowledge of $R_{i,t}$ does not beat simply keeping tracks longer. In dense crowds, some tracks whose person is never re-matched at IoU ≥ 0.5 still pay off when a looser detection re-acquires them, which an oracle of $R_{i,t}$ cannot know. OC-SORT is almost insensitive to every hook on MOT20: the best buffer changes HOTA by +0.08, re-association by −0.02, and the learned memory by −0.58. K is the one hook that behaves differently here. It leaves HOTA unchanged (−0.04 and −0.08), but with a quarter of the detections heavily occluded it reduces ByteTrack's identity switches from 3301 to 2897 (−12%), the largest reduction of any variant, at a small cost in MOTA.

5.5 What happens at the occlusion events

Aggregate HOTA dilutes the effect of occlusion handling, because most frames contain no occlusion event. Table 6 and Fig. 5 score detector-gap events directly. The MOT17 second halves contain 381 such events; for each base tracker we keep those whose pre-gap frame is matched by every variant (250 for ByteTrack, 134 for OC-SORT), so all variants of a tracker are compared on identical events. With ByteTrack the default keeps identity in 40.8% of these events, and almost never across gaps of 2 s or more. The best fixed buffer gives 44.8%. Buffered re-association raises retention to 50.4% (+9.6 points, 95% CI +4.4 to +15.1; 32 events gained, 8 lost; McNemar $p = 1.8 \times 10^{-4}$, Holm-adjusted 0.004). It also beats the best buffer (+5.6 points, $p = 0.020$). The belief gate changes no event outcome: R and $\mathrm{R}_{\mathrm{geom}}$ retain identity in exactly the same events. With OC-SORT no comparison is significant after correction (R: +5.2 points, $p = 0.092$), again because its own recovery already covers part of the problem. Of the twenty comparisons with the default on MOT17, only R, $\mathrm{R}_{\mathrm{geom}}$ and oracle M+R for ByteTrack survive Holm correction.

On MOT20 the event analysis has far more power: 16,004 detector-gap events, of which 13,537 (ByteTrack) and 8,007 (OC-SORT) form the common sets. For ByteTrack the default retains identity in 79.6% of events, the tuned buffer in 82.3%, buffered re-association in 82.5% and M+R+K in 82.7%. All of these gains over the default survive Holm correction, but almost all of them come from the buffer: R adds +0.2 points beyond it ($p = 0.001$), a small difference that is detectable only because the sample is large. K raises retention slightly (+0.6 points for ByteTrack and +0.5 for OC-SORT; Holm-adjusted $p = 0.056$ and 0.064). For OC-SORT, retention is essentially unchanged by the buffer and by re-association (65.1%, 65.0% and 65.0%), and the learned memory lowers it slightly (−0.7 points, Holm-adjusted $p = 0.038$).

6 Discussion

Where occlusion handling should spend its effort.

Making the tracker *know* that an object is occluded is achievable: occlusion of detected boxes and the persistence of a lost track are both predictable from observable cues (AUC 0.787–0.956 and 0.869–0.890). But knowing it changes few decisions. Detected boxes are rarely heavily occluded on MOT17, so reweighting their updates does nothing there; in the dense MOT20 crowds it reduces identity switches without moving HOTA. Deletion has little headroom once the buffer is long, because exits are rare and false tracks are cheap to keep. The decision with

Table 5 Tracking on the second halves (TrackEval). Each variant uses the grid point with the best first-half HOTA. Δ: HOTA change vs. the default tracker (points). Oracle rows replace π by the true outcome $R_{i,t}$; they bound persistence beliefs of this kind and are not methods. MOT20 uses the reduced grid (no M+R and oracle M+R runs).

Data	Tracker	Variant	HOTA	Δ	DetA	AssA	IDF1	MOTA	IDSW	Frag
MOT17	ByteTrack	Default	49.65	+0.00	42.55	58.22	57.06	47.41	230	345
		Best fixed buffer	50.27	+0.62	42.56	59.64	57.46	47.46	232	351
		Legacy rule (submitted)	50.05	+0.39	42.57	59.11	57.67	47.42	228	349
		K	49.71	+0.05	42.52	58.39	57.12	47.44	232	334
		M	50.20	+0.54	42.58	59.45	57.15	47.45	237	348
		R_{geom}	50.50	+0.84	42.56	60.17	57.62	47.64	197	350
		R	50.61	+0.96	42.55	60.46	57.80	47.62	195	350
		M+R	50.17	+0.52	42.59	59.36	57.12	47.46	233	348
		M+R+K	50.11	+0.46	42.55	59.29	57.27	47.57	215	335
		Oracle M	50.49	+0.84	42.57	60.14	57.95	47.42	225	344
		Oracle M+R	51.18	+1.52	42.58	61.76	59.48	47.60	195	345
	OC-SORT	Default	49.78	+0.00	41.88	59.32	56.25	45.99	156	297
		Best fixed buffer	50.37	+0.59	41.86	60.77	57.02	46.10	137	292
		Legacy rule (submitted)	50.00	+0.21	41.88	59.84	56.62	46.01	157	296
		K	49.66	−0.12	41.87	59.04	56.09	45.97	158	298
		M	50.18	+0.40	41.88	60.28	56.48	46.09	149	295
		R_{geom}	50.34	+0.56	41.96	60.54	56.71	46.23	146	311
		R	50.21	+0.43	41.91	60.31	56.55	46.17	152	318
		M+R	50.29	+0.51	41.90	60.52	56.67	46.15	150	317
		M+R+K	50.34	+0.56	41.88	60.68	56.74	46.13	147	313
		Oracle M	50.63	+0.84	41.85	61.39	57.73	46.06	134	295
		Oracle M+R	50.69	+0.91	41.96	61.40	57.75	46.24	135	300
MOT20	ByteTrack	Default	40.40	+0.00	42.41	38.62	48.01	49.14	3301	20858
		Best fixed buffer	41.93	+1.53	42.54	41.47	50.62	49.30	3106	21418
		Legacy rule (submitted)	41.09	+0.69	42.44	39.91	49.18	49.20	3175	21011
		K	40.36	−0.04	41.90	39.02	48.80	48.76	2897	20264
		M	41.34	+0.94	42.60	40.25	49.81	49.30	3261	21587
		R_{geom}	41.96	+1.57	42.56	41.52	50.70	49.32	3090	21442
		R	42.01	+1.61	42.56	41.60	50.78	49.31	3088	21437
		M+R+K	41.16	+0.76	41.92	40.56	50.90	48.83	2882	21191
		Oracle M	41.52	+1.13	42.57	40.64	49.93	49.30	3147	21291
	OC-SORT	Default	40.34	+0.00	39.11	41.71	48.03	44.25	2502	12108
		Best fixed buffer	40.42	+0.08	38.94	42.07	48.48	44.16	2377	12058
		Legacy rule (submitted)	40.37	+0.02	39.07	41.81	48.07	44.26	2482	12102
		K	40.26	−0.08	39.11	41.55	47.90	44.24	2420	12125
		M	39.76	−0.58	38.88	40.76	47.58	44.12	2489	12076
		R_{geom}	40.30	−0.04	38.94	41.82	48.37	44.18	2390	12072
		R	40.32	−0.02	38.94	41.86	48.35	44.17	2397	12067
		M+R+K	39.97	−0.37	38.89	41.18	47.93	44.16	2332	12095
		Oracle M	40.13	−0.21	39.01	41.40	48.02	44.24	2380	12114

headroom is re-association: recognising a returning person as the same identity. That is an appearance and geometry problem more than a visibility problem, and in our experiments the geometric buffered matching, not the belief gate, carries the gain. It explains why the strongest occlusion-oriented trackers invest in observation-centric recovery, depth-ordered association and occlusion-robust appearance [2, 3, 5], rather than in modelling visibility itself. For a practitioner the actionable conclusions are: (i) tune the lost-track buffer before adding any occlusion module; (ii) for a tracker without observation-centric recovery, add a buffered re-association stage; (iii) in dense crowds, an occlusion-aware motion update is a cheap way to reduce identity switches; and (iv) judge any proposed occlusion signal against an oracle of the same decision before building on it.

Lessons for evaluating probabilistic components.

The submitted version of this work reported an expected calibration error of 0.032 and a large reduction of an association-risk proxy. Both numbers were real, and both were uninformative. On a target with a strong majority class, a constant forecast is calibrated: our prevalence-only reference reaches the same ECE as the legacy model. A risk proxy of the form $(1-p)\,c$ decreases monotonically as $p \to 1$ and is therefore minimised by always predicting "visible". The protocol used here guards against both failures with threshold-free discrimination metrics, a Brier *skill* score

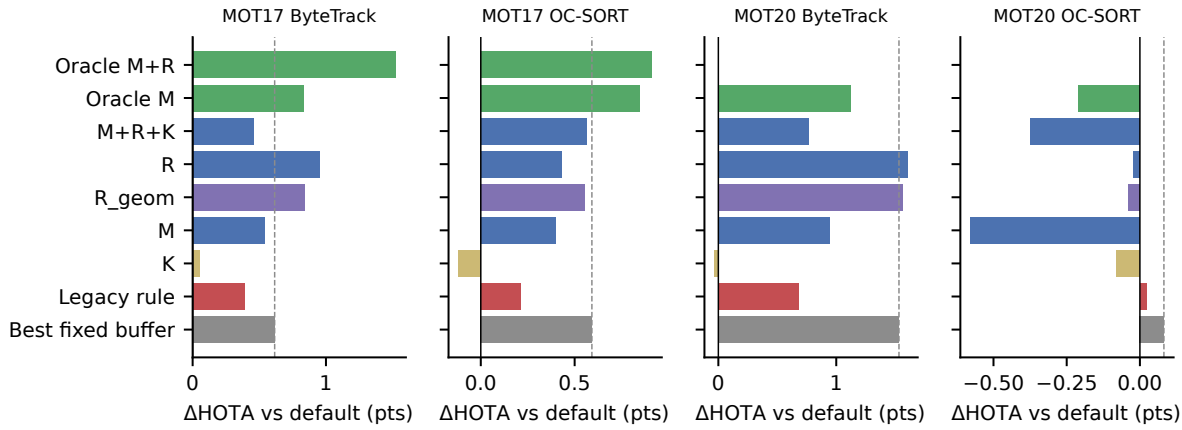


Fig. 4 HOTA change of each variant against the default tracker on the second halves. The dashed line marks the best fixed buffer, the memory-length control. Green bars are oracle persistence beliefs: a ceiling for any predictor of $R_{i,t}$ under the same rule, not for every possible policy

Table 6 Identity retention across detector-gap events on the second halves (%; events in parentheses). All variants of a tracker are scored on the same events: those whose pre-gap frame every variant matches. Δ vs. default with 95% cluster-bootstrap interval over identities; p: exact McNemar test / Holm-adjusted over all comparisons with the default in that dataset; vs. buffer: paired difference to the best fixed buffer (McNemar p); W/L: sequences with higher/lower HOTA than the default.

Data	Tracker	Variant	All	<0.5 s	0.5–2 s	≥2 s	Δ all [95% CI]	p / Holm	vs. buffer	W/L
MOT17	ByteTrack	Default	40.8	54.0 (139)	31.4 (86)	0.0 (25)	–	–	–	–
		Best fixed buffer	44.8	52.5 (139)	44.2 (86)	4.0 (25)	+4.0 [+0.7, +7.4]	0.031 / 0.43	–	5/2
		Legacy rule (submitted)	42.4	53.2 (139)	36.0 (86)	4.0 (25)	+1.6 [−0.8, +4.2]	0.34 / 1.00	−2.4 (0.070)	3/2
		K	40.4	53.2 (139)	31.4 (86)	0.0 (25)	−0.4 [−2.1, +1.4]	1.00 / 1.00	−4.4 (0.035)	4/3
		M	44.8	51.1 (139)	43.0 (86)	16.0 (25)	+4.0 [+0.0, +8.2]	0.064 / 0.83	+0.0 (1.00)	4/3
		R_{geom}	50.4	55.4 (139)	52.3 (86)	16.0 (25)	+9.6 [+4.4, +15.1]	1.8×10^{-4} / 0.004	+5.6 (0.020)	4/3
		R	50.4	55.4 (139)	52.3 (86)	16.0 (25)	+9.6 [+4.4, +15.1]	1.8×10^{-4} / 0.004	+5.6 (0.020)	5/2
		M+R	46.8	51.8 (139)	47.7 (86)	16.0 (25)	+6.0 [+1.6, +10.4]	0.006 / 0.095	+2.0 (0.23)	4/3
		M+R+K	47.6	51.8 (139)	47.7 (86)	24.0 (25)	+6.8 [+2.1, +11.9]	0.003 / 0.057	+2.8 (0.19)	4/3
		Oracle M	46.0	53.2 (139)	43.0 (86)	16.0 (25)	+5.2 [+1.3, +9.1]	0.015 / 0.22	+1.2 (0.55)	5/2
		Oracle M+R	50.8	54.0 (139)	51.2 (86)	32.0 (25)	+10.0 [+4.4, +16.1]	3.5×10^{-4} / 0.006	+6.0 (0.028)	7/0
	OC-SORT	Default	44.0	54.5 (88)	26.3 (38)	12.5 (8)	–	–	–	–
		Best fixed buffer	45.5	53.4 (88)	31.6 (38)	25.0 (8)	+1.5 [−2.2, +6.2]	0.75 / 1.00	–	5/2
		Legacy rule (submitted)	43.3	52.3 (88)	31.6 (38)	0.0 (8)	−0.7 [−3.9, +2.5]	1.00 / 1.00	−2.2 (0.38)	4/0
		K	44.0	54.5 (88)	26.3 (38)	12.5 (8)	+0.0 [+0.0, +0.0]	1.00 / 1.00	−1.5 (0.75)	2/2
		M	41.0	48.9 (88)	26.3 (38)	25.0 (8)	−3.0 [−7.1, +0.9]	0.39 / 1.00	−4.5 (0.11)	3/4
		R_{geom}	49.3	58.0 (88)	34.2 (38)	25.0 (8)	+5.2 [+0.8, +9.7]	0.092 / 1.00	+3.7 (0.18)	4/3
		R	49.3	58.0 (88)	34.2 (38)	25.0 (8)	+5.2 [+0.8, +9.7]	0.092 / 1.00	+3.7 (0.18)	4/3
		M+R	49.3	59.1 (88)	34.2 (38)	12.5 (8)	+5.2 [+0.8, +9.8]	0.092 / 1.00	+3.7 (0.18)	6/1
		M+R+K	48.5	58.0 (88)	34.2 (38)	12.5 (8)	+4.5 [−0.8, +9.6]	0.18 / 1.00	+3.0 (0.34)	6/1
		Oracle M	46.3	54.5 (88)	31.6 (38)	25.0 (8)	+2.2 [−1.7, +7.0]	0.51 / 1.00	+0.7 (1.00)	6/1
		Oracle M+R	49.3	59.1 (88)	31.6 (38)	25.0 (8)	+5.2 [+0.8, +9.7]	0.092 / 1.00	+3.7 (0.30)	6/1
MOT20	ByteTrack	Default	79.6	89.6 (10646)	50.3 (2396)	5.9 (495)	–	–	–	–
		Best fixed buffer	82.3	89.9 (10646)	61.3 (2396)	19.4 (495)	+2.7 [+2.2, +3.1]	1.0×10^{-38} / 1.4×10^{-37}	–	4/0
		Legacy rule (submitted)	80.7	89.8 (10646)	55.8 (2396)	6.3 (495)	+1.1 [+0.9, +1.4]	7.3×10^{-24} / 8.7×10^{-23}	-1.5 (4.2×10^{-18})	4/0
		K	80.2	90.1 (10646)	51.5 (2396)	4.4 (495)	+0.6 [+0.1, +1.0]	0.007 / 0.056	-2.1 (7.1×10^{-15})	2/2
		M	81.8	89.5 (10646)	59.8 (2396)	22.6 (495)	+2.2 [+1.7, +2.8]	1.7×10^{-21} / 1.7×10^{-20}	−0.4 (0.004)	4/0
		R_{geom}	82.5	90.1 (10646)	61.9 (2396)	19.2 (495)	+2.9 [+2.4, +3.4]	8.2×10^{-44} / 1.2×10^{-42}	+0.2 (0.001)	4/0
		R	82.5	90.1 (10646)	61.8 (2396)	19.2 (495)	+2.9 [+2.4, +3.4]	6.4×10^{-44} / 1.0×10^{-42}	+0.2 (0.001)	4/0
		M+R+K	82.7	90.1 (10646)	62.4 (2396)	22.4 (495)	+3.1 [+2.5, +3.7]	5.5×10^{-27} / 7.1×10^{-26}	+0.4 (0.097)	3/1
		Oracle M	81.9	89.4 (10646)	60.6 (2396)	24.4 (495)	+2.3 [+1.8, +2.8]	1.9×10^{-22} / 2.1×10^{-21}	−0.4 (0.033)	4/0
	OC-SORT	Default	65.1	72.7 (6318)	42.8 (1389)	8.7 (300)	–	–	–	–
		Best fixed buffer	65.0	71.5 (6318)	44.8 (1389)	22.3 (300)	−0.1 [−0.6, +0.3]	0.64 / 1.00	–	3/1
		Legacy rule (submitted)	65.2	72.7 (6318)	42.9 (1389)	9.3 (300)	+0.1 [−0.1, +0.3]	0.46 / 1.00	+0.2 (0.41)	3/1
		K	65.7	73.4 (6318)	42.7 (1389)	8.7 (300)	+0.5 [+0.1, +1.0]	0.009 / 0.064	+0.6 (0.020)	3/1
		M	64.4	71.0 (6318)	43.6 (1389)	21.7 (300)	−0.7 [−1.3, −0.2]	0.004 / 0.038	-0.6 (5.0×10^{-4})	0/4
		R_{geom}	65.1	71.5 (6318)	45.1 (1389)	22.7 (300)	−0.0 [−0.5, +0.4]	0.87 / 1.00	+0.1 (0.60)	2/2
		R	65.0	71.4 (6318)	44.9 (1389)	22.7 (300)	−0.1 [−0.6, +0.3]	0.58 / 1.00	−0.0 (0.89)	2/2
		M+R+K	65.7	72.1 (6318)	45.6 (1389)	22.7 (300)	+0.6 [−0.0, +1.1]	0.061 / 0.37	+0.7 (0.013)	0/4
		Oracle M	65.2	71.7 (6318)	44.6 (1389)	23.0 (300)	+0.1 [−0.4, +0.6]	0.76 / 1.00	+0.2 (0.35)	1/3

against the prevalence forecast, the spread of the predicted probability, and oracle bounds on the downstream decision. We regard this as the more general contribution: a model component should be evaluated by the decision it changes, not by the quality of its probabilities in isolation.

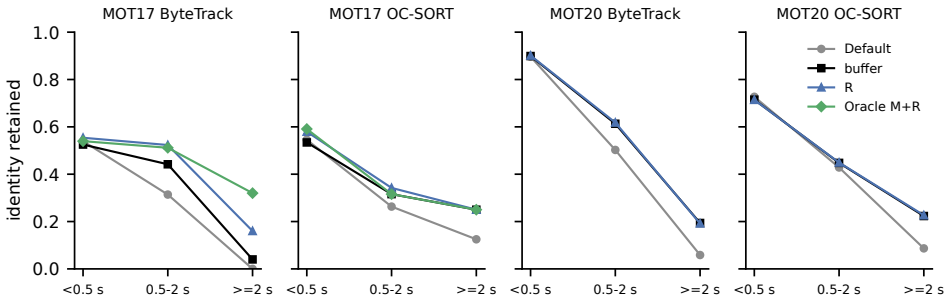


Fig. 5 Identity retention across detector-gap events by gap length (second halves; common event set per tracker)

Limitations.

The study uses the public detections of MOT17 and MOT20 and the training sequences of both benchmarks, evaluated on their second halves; it does not include a test-server submission. The two halves of a sequence share scenes and identities, and the belief models, perspective fits and thresholds are fitted on the first halves of the same sequences that are evaluated. The leave-one-sequence-out result for q suggests that the detection-level belief transfers across scenes, but we did not run leave-one-sequence-out tracking. Each tracking configuration is a single deterministic run, so HOTA differences of a few hundredths of a point carry no evidential weight; the event-level tests are the paper's statistical evidence. On MOT20 the public detections carry no usable confidence, so ByteTrack's low-score stage and the confidence cues are disabled there, and the oracle re-association variant was evaluated on MOT17 only. The persistence belief and the re-association stage are geometric; an appearance model would very likely raise the re-association headroom that the oracle reveals, and is the natural next step. The halves are short (9–66 s), which limits the longest occlusions that can be observed and makes right-censoring frequent on MOT17. Finally, the oracle bounds hold for the decision rules and association geometry used here; a different matching cost or policy could have a different ceiling.

7 Conclusion

We asked how much an explicit visibility belief can improve occlusion handling in online tracking-by-detection, and answered it with learned beliefs, controlled hooks into ByteTrack and OC-SORT, and oracle bounds. Detection-level occlusion is predictable; it is rare among detected boxes on MOT17 (2.3%) but common on MOT20 (25.7%), where acting on it reduces identity switches without changing HOTA. Lost tracks are overwhelmingly people who return, which makes memory length, not deletion intelligence, the first lever. The remaining headroom lies in re-associating returning people, where a buffered re-association stage helps a tracker that lacks observation-centric recovery and an oracle shows more is available on MOT17. The original VIS-Belief design, a per-track visibility probability that lengthens the deletion age, sits on the part of this landscape with the least headroom. We release the code, configurations and every result file so that the bounds can be reused to vet future occlusion modules.

Acknowledgements. We thank the MOTChallenge organisers for the benchmarks and the authors of ByteTrack, OC-SORT and TrackEval for releasing their code.

Statements and Declarations

- **Funding.** No funding was received for conducting this study.
- **Competing interests.** The author has no competing interests to declare that are relevant to the content of this article.
- **Ethics approval and consent to participate.** Not applicable. The study uses only the publicly released MOT17 and MOT20 benchmark data under their terms of use; no new data involving human participants were collected.
- **Consent for publication.** Not applicable.
- **Data availability.** MOT17 and MOT20 are available from https://motchallenge.net. All

derived data (per-detection occlusion probabilities, lost-state tables, tracker outputs, TrackEval results and event tables) are released with the code.

- **Code availability.** The code, the pinned third-party sources, all configurations and the scripts that regenerate every number, table and figure of this article are available from the corresponding author during review and will be released in a public repository on publication.
- **Author contribution.** S.D., H.R., A.N., A.V., N.C., and P.D. contributed equally to this work, including the conception of the study, data collection, formal analysis, and writing of the manuscript.
- **Use of AI tools.** An AI assistant (Claude, Anthropic) was used to help implement the experiment code, run the experiments and draft the text. The author designed and verified the study, checked every result against the released result files, and takes full responsibility for the content.

Appendix A Tuning grids

Every variant was tuned on the first halves over the grid below; the selected point was evaluated once on the second halves. For MOT20 a reduced grid was used because a single MOT20 pass is about ten times longer than a MOT17 pass: buffer $\{30, 90, 150, 300\}$, legacy $B = 20$, K $\lambda \in \{1, 4\}$, M $\tau_{\text{del}} \in \{0.1, 0.2, 0.3, 0.5\}$, R_{geom} and R with $b \in \{90, 300\}$ and $\theta_{\text{rec}} \in \{0.2, 0.3\}$, M+R+K, and oracle M; M+R and oracle M+R were not run on MOT20. The buffer is in frames at 30 fps (ByteTrack rescales it by the frame rate; OC-SORT does not).

References

[1] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: ByteTrack: Multi-object tracking by associating every detection box. In: European Conference on Computer Vision (ECCV), pp. 1–21 (2022). https://doi.org/10.1007/978-3-031-20047-2_1

[2] Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric SORT: Rethinking SORT for robust multi-object tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9686–9696 (2023). https://doi.org/10.1109/CVPR52729.2023.00934

[3] Liu, Z., Wang, X., Wang, C., Liu, W., Bai, X.: SparseTrack: Multi-object tracking by performing scene decomposition based on pseudo-depth. IEEE Transactions on Circuits and Systems for Video Technology (2025) https://doi.org/10.1109/TCSVT.2024.3524670

[4] Wang, Y.-C., Zhang, D.-W., Li, R., Zheng, Z., Li, M.: PD-SORT: Occlusion-robust multi-object tracking using pseudo-depth cues. IEEE Transactions on Consumer Electronics (2025) https://doi.org/10.1109/TCE.2025.3541839

[5] Wang, S.-H., Shi, X.-C., Li, Y.-C., Sun, K.-C., Liu, M.-C.: OccluSort: Pedestrian multi-object tracking in occlusion scenarios based on dynamic occlusion-aware re-identification. Measurement Science and Technology (2025) https://doi.org/10.1088/1361-6501/ae2351

[6] Li, Q., Yang, J.: OA-SORT: Improving multi-object tracking performance with occlusion aware ViT and adaptive appearance update. In: International Joint Conference on Neural Networks (IJCNN) (2025). https://doi.org/10.1109/IJCNN64981.2025.11228170

[7] Tokmakov, P., Li, J., Burgard, W., Gaidon, A.: Learning to track with object permanence. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2021). https://doi.org/10.1109/ICCV48922.2021.01068

[8] Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering **82**(1), 35–45 (1960) https://doi.org/10.1115/1.3662552

[9] Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1–2), 83–97 (1955) https://doi.org/10.1002/nav.3800020109

[10] Bewley, A., Ge, Z., Ott, L., Ramos, F.,

Table A1 Grids per variant (MOT17). b: fixed buffer; τ_{del}, τ_{rec}: belief thresholds; θ_{rec}: buffered-IoU acceptance; μ: margin; λ: noise inflation; $T_{\max} = 10\,\text{s}$ for all memory variants.

Variant	Grid
buffer	$b \in \{30, 60, 90, 150, 240, 300, 450, 600\}$
legacy	$A_0 = 30$, $B \in \{20, 60\}$
K	$\lambda \in \{1, 2, 4\}$
M	$\tau_{\text{del}} \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$
R_{geom}	$b \in \{60, 150, 300\}$, $\theta_{\text{rec}} \in \{0.2, 0.3, 0.4\}$, $\mu \in \{0.2, 0.4\}$, $\tau_{\text{rec}} = 0$
R	$b \in \{60, 150, 300\}$, $\tau_{\text{rec}} \in \{0.3, 0.5\}$, $\theta_{\text{rec}} \in \{0.2, 0.3\}$, $\mu \in \{0.2, 0.4\}$
M+R	$\tau_{\text{del}} \in \{0.1, 0.2\}$, $\tau_{\text{rec}} \in \{0.3, 0.5\}$, $\theta_{\text{rec}} \in \{0.2, 0.3\}$, $\mu = 0.3$
M+R+K	as M+R, with $\lambda \in \{1, 2\}$
oracle M	$\tau_{\text{del}} = 0.5$
oracle M+R	$\tau_{\text{rec}} = 0.5$, $\theta_{\text{rec}} \in \{0.2, 0.3\}$, $\mu \in \{0.2, 0.4\}$

Upcroft, B.: Simple online and realtime tracking. In: IEEE International Conference on Image Processing (ICIP), pp. 3464–3468 (2016). https://doi.org/10.1109/ICIP.2016.7533003

[11] Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: IEEE International Conference on Image Processing (ICIP), pp. 3645–3649 (2017). https://doi.org/10.1109/ICIP.2017.8296962

[12] Aharon, N., Orfaig, R., Bobrovsky, B.-Z.: BoT-SORT: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)

[13] Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H.: StrongSORT: Make DeepSORT great again. IEEE Transactions on Multimedia **25**, 8725–8737 (2023) https://doi.org/10.1109/TMM.2023.3240881

[14] Yang, M., Han, G., Yan, B., Zhang, W., Qi, J., Lu, H., Wang, D.: Hybrid-SORT: Weak cues matter for online multi-object tracking. In: AAAI Conference on Artificial Intelligence, vol. 38, pp. 6504–6512 (2024). https://doi.org/10.1609/aaai.v38i7.28471

[15] Stanojević, V.D., Todorović, B.T.: BoostTrack: Boosting the similarity measure and detection confidence for improved multiple object tracking. Machine Vision and Applications **35**(3), 53 (2024) https://doi.org/10.1007/s00138-024-01531-5

[16] Yang, F., Odashima, S., Masui, S., Jiang, S.: Hard to track objects with irregular motions and similar appearances? Make it easier by buffering the matching space. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2023)

[17] Hu, Y., Niu, A., Zhu, Y., Yan, Q., Sun, J., Zhang, Y.: Multiple object tracking based on occlusion-aware embedding consistency learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2024). https://doi.org/10.1109/ICASSP48485.2024.10446647

[18] Wang, Y.-H., Hsieh, J.-W., Chen, P.-Y., Chang, M.-C., So, H.-H., Li, X.: SMILEtrack: SiMIlarity LEarning for occlusion-aware multiple object tracking. In: AAAI Conference on Artificial Intelligence, vol. 38, pp. 5740–5748 (2024). https://doi.org/10.1609/aaai.v38i6.28386

[19] Zhao, R., Zhang, X., Zhang, J.: PSMOT: Online occlusion-aware multi-object tracking exploiting position sensitivity. Sensors **24**(4), 1199 (2024) https://doi.org/10.3390/s24041199

[20] Su, Y., Sun, R., Shu, X., Zhang, Y., Wu, Q.: Occlusion-aware detection and re-ID calibrated network for multi-object tracking. arXiv preprint arXiv:2308.15795 (2023)

[21] Zhao, Y., Sun, F., Wang, C., Chen, J., Pan, X.: Robust association method based on occlusion-aware in multi-object tracking. International Journal of Machine Learning and Cybernetics (2026) https://doi.org/10.1007/s13042-025-02909-1

[22] Kim, V., Jung, G., Lee, S.-W.: AM-SORT:

Adaptable motion predictor with historical trajectory embedding for multi-object tracking. arXiv preprint arXiv:2401.13950 (2024)

[23] Wang, H.: DOA-SORT: Directional occlusion-aware multi-object tracking with distributional observations. arXiv preprint arXiv:2609.22706 (2026)

[24] Hsieh, C.-Y., Chen, K., Dave, A., Khurana, T., Ramanan, D.: TAO-Amodal: A benchmark for tracking any object amodally. arXiv preprint arXiv:2312.12433 (2023)

[25] Qi, J., Gao, Y., Hu, Y., Wang, X., Liu, X., Bai, X., Belongie, S., Yuille, A., Torr, P.H.S., Bai, S.: Occluded video instance segmentation: A benchmark. International Journal of Computer Vision **130**, 2022–2039 (2022) https://doi.org/10.1007/s11263-022-01629-1

[26] Meinhardt, T., Kirillov, A., Leal-Taixé, L., Feichtenhofer, C.: TrackFormer: Multi-object tracking with transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8844–8854 (2022). https://doi.org/10.1109/CVPR52688.2022.00864

[27] Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: MOTR: End-to-end multiple-object tracking with transformer. In: European Conference on Computer Vision (ECCV), pp. 659–675 (2022). https://doi.org/10.1007/978-3-031-19812-0_38

[28] Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The CLEAR MOT metrics. EURASIP Journal on Image and Video Processing **2008**, 246309 (2008) https://doi.org/10.1155/2008/246309

[29] Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision Workshops (ECCVW), pp. 17–35 (2016). https://doi.org/10.1007/978-3-319-48881-3_2

[30] Luiten, J., Ošep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: HOTA: A higher order metric for evaluating multi-object tracking. International Journal of Computer Vision **129**, 548–578 (2021) https://doi.org/10.1007/s11263-020-01375-2

[31] Luiten, J., Hoffhues, A.: TrackEval. https://github.com/JonathonLuiten/TrackEval (2020)

[32] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)

[33] Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: MOT20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 (2020)

[34] Dendorfer, P., Ošep, A., Milan, A., Schindler, K., Cremers, D., Reid, I., Roth, S., Leal-Taixé, L.: MOTChallenge: A benchmark for single-camera multiple target tracking. International Journal of Computer Vision **129**, 845–881 (2021) https://doi.org/10.1007/s11263-020-01393-0

[35] Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly Weather Review **78**(1), 1–3 (1950) https://doi.org/10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2

[36] Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: International Conference on Machine Learning (ICML), pp. 625–632 (2005). https://doi.org/10.1145/1102351.1102430

[37] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning (ICML), pp. 1321–1330 (2017)

[38] Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE **10**(3), 0118432 (2015) https://doi.org/10.

1371/journal.pone.0118432

[39] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 3146–3154 (2017)

[40] Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A.J., Bartlett, P., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers, pp. 61–74. MIT Press, ??? (1999)

[41] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 91–99 (2015)

[42] Field, C.A., Welsh, A.H.: Bootstrapping clustered data. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **69**(3), 369–390 (2007) https://doi.org/10.1111/j.1467-9868.2007.00593.x

[43] McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika **12**(2), 153–157 (1947) https://doi.org/10.1007/BF02295996