

MHA-Track: Multi-Horizon Association for Vehicle Tracking in Low-Frame-Rate Aerial Imagery

Manuel Mühlhaus^{*1,2}, Reza Bahmanyar¹, Franz Kurz¹, and Friedrich Fraundorfer^{1,2}

¹ Remote Sensing Technology Institute, German Aerospace Center (DLR),
Münchener Str. 20, 82234 Weßling, Germany

manuel.muehlhaus@dlr.de

<https://www.dlr.de/eoc/en>

² Institute of Visual Computing, Graz University of Technology (TU Graz),
Inffeldgasse 16, 8010 Graz, Austria

<http://www.ivc.tugraz.at/>

Abstract. Vehicle tracking in aerial image sequences is promising for traffic monitoring, offering wide coverage and high resolution, but low framerates and large camera motions break standard tracking assumptions. Small, similar vehicles and frequent unmatched cases make adjacent-frame appearance-based association unreliable. We introduce **MHA-Track**, a multi-horizon GNN-based tracker designed for low frame-rate aerial imagery. For each vehicle and anchor frame, **MHA-Track** scores candidate matches across six temporal horizons within overlapping windows of eight camera-motion-aligned frames, competing matches against learned continue and terminate states to handle disappearances. A Gaussian motion model predicts per-horizon displacements, and a two-pass inference refines associations using trajectory speed estimates. Aggregating evidence across horizons and windows enables robust tracking under occlusion and large displacements. On the VETRA dataset, **MHA-Track** outperforms the best baseline by +1.94 HOTA, +1.65 MOTA, +2.97 IDF1, and halves identity switches. **MHA-Track** excels at sharp turns, large camera motions, recovery after occlusion, and tracking similar vehicles in close proximity. Code is publicly available at <https://github.com/ManuelM95/MHA-Track>.

Keywords: Tracking · Vehicles · Attention · Remote Sensing

1 Introduction

Aerial traffic monitoring is an important tool for traffic-state analysis, travel-time estimation, queue analysis, and trajectory extraction [3,6]. Compared to fixed cameras, airborne platforms can observe large road networks and many vehicles from a single moving sensor. Large-format aerial mapping systems provide very

* Corresponding Author

high spatial resolution, but commonly operate at sparse, low-single-digit frame rates [22,26,14]. This low framerate regime is still attractive: each frame covers a large area, and forward platform motion observes new road sections and additional traffic participants. VETRA [12] captures this setting with real-world airplane and helicopter image sequences at 1–2 frames per second, making it a suitable benchmark for low-frame-rate aerial vehicle tracking. However, this setting violates many assumptions of standard dense-video multi-object tracking benchmarks such as MOT Challenge [8], DanceTrack [25], and SportsMOT [7]. Consecutive frames exhibit large camera motion and large vehicle displacement, while vehicles are small, visually similar, and often only weakly distinguishable by appearance. Limited frame overlap causes frequent enter, exit, and no-match cases. In addition, bridges, tunnels, trees, and border truncation create temporary disappearances that require refinding rather than simple adjacent-frame matching.

We therefore propose a method suitable for low framerate aerial vehicle tracking. Our main contributions are:

- We introduce **MHA-Track**, a multi-horizon GNN-based tracker that scores candidates over six temporal offsets per detection, using explicit continue and terminate no-match states for robust association at low framerates and frequent occlusions.
- We propose a Gaussian motion model that predicts per-horizon displacements from image context and neighborhood cues, offering uncertainty-aware scoring and in-frame/out-of-frame signals for no-match decisions.
- We introduce a two-pass inference strategy that injects trajectory speed estimates to refine associations and aggregates evidence across temporal offsets for robust track construction.
- On VETRA, **MHA-Track** outperforms the best method by +1.94 HOTA, +1.65 MOTA, and +2.97 IDF1, cutting identity switches from 792 to 392.

2 Related Work

The MOT Challenge series [8] is the main evaluation protocol for *Multiple Object Tracking* (MOT). The DanceTrack [25] and SportsMOT [7] datasets shift the focus from detection to association by testing trackers on visually similar objects with non-linear, high-speed motion, where appearance-based *re-Identification* (re-ID) fails. Recent advances include CoNo-Link [10], which treats tracklets as graph nodes in a *Graph Neural Network* (GNN) for longer-term reasoning, and NOUGAT [20], which unifies online and offline tracking using a hierarchical GNN and an Autoregressive Long-term Tracking layer, achieving *state-of-the-art* (SOTA) association. CoTracker [13] excels under occlusion and camera motion by jointly tracking up to 70k point across frames, but requires high-frame-rate video and cannot assign detection-level identities. MOTR [30] replaced matching heuristics with learned track queries in a transformer decoder. MOTRv2 [32] and MeMOTR [9] improve detection and long-term re-ID. All these benchmarks and

methods assume dense, high-frequency videos with large, clear objects that do not match real-world aerial traffic monitoring conditions.

Aerial and satellite MOT datasets partially fill this gap, but each targets a distinct scenario. VisDrone [34] provides large-scale UAV footage, but its high frame rate and mostly static platform result in smooth motion, unlike the low-frequency, moving-camera conditions in traffic monitoring. VISO [28] and SatVideoDT [29] use satellite video, where coarse *Ground Sampling Distance* (GSD) and orbital motion result in very different dynamics. M3OT [21] provides dual-UAV multimodal tracking, but focuses on high frame rates and sensor fusion. VETRA [12], designed for real-world traffic monitoring, contains 52 airplane and helicopter sequences at 1–2 Hz, with varying GSD, six-DoF camera motion, and 55–100% frame overlap. It challenges detection, motion modeling, and association in complex urban and sub-urban scenes. Thus, we use VETRA as our main benchmark.

Existing methods target different challenges than VETRA. Even Deep SR-SORT [12], the VETRA baseline that incorporates BRISK-based [15] camera motion compensation and adaptive search radius, still faces significant difficulties. The fundamental limitation among these approaches is the absence of robust multi-horizon association and explicit no-match states. Without mechanisms to defer or reject assignments across variable temporal gaps, identity switches accumulate rapidly in scenarios with low frame rate and large displacement.

3 Method

Given low framerate aerial imagery, large camera motion, and frequent object disappearances, we approach vehicle tracking as *explicit multi-horizon association* rather than adjacent-frame matching. For each detection, the model scores candidate matches over multiple time offsets and compares them with no-match states to decide track continuation or termination when objects disappear.

3.1 Multi-Horizon Input

Each sequence is processed in overlapping (stride 1) windows of 8 consecutive frames. Within each window, each frame is taken in turn as the anchor at $t = 0$. For each anchor, six temporal horizons $\Delta h \in \{-3, -2, -1, +1, +2, +3\}$ reference the three preceding and three following frames. Horizons falling outside the window are masked, ensuring uniform handling of boundaries. Detections in $t = 0$ are treated as queries; detections in horizon frames are match candidates. This results in an association tensor of size $N_H \times N_Q \times (N_C + 2)$, where N_Q is the number of queries in the anchor, $N_H = 6$ is the number of horizons, N_C is the number of candidates per query per horizon, and $+2$ represents continue and terminate no-match states.

Camera-motion alignment. Platform movement between frames causes raw pixel displacements that mix camera and vehicle motion. For each horizon Δh , we project candidates from the frame at $t = \Delta h$ into the anchor’s coordinates

using chained BRISK affine transforms [15]. After alignment, the remaining displacement between query and candidate reflects only vehicle motion, which the model learns to predict.

Candidate and neighbor selection. For each query, we select the N_C nearest projected detections from each horizon as candidate matches. $N_C = 24$ was sufficient to capture the correct matches in nearly all VETRA sequences. We also take the $N_N = 24$ nearest detections in the anchor frame as same-frame neighbors, providing local scene context, such as whether a query is in dense traffic or free-flow conditions.

Sample representation. Each training sample encodes an anchor-centered association problem. It stores the query neighborhood, candidate sets for all valid horizons, and the window’s *Average Car Diagonal* (ACD), the mean bounding box diagonal across all detections in the window, for spatial normalization across sequences. Appearance features are included as ResNet-50 re-ID embeddings [33,11] extracted from image crops defined by the query and candidate bounding boxes. Each sample also provides per-horizon position and assignment labels.

3.2 Multi-Horizon Tracking Overview

Figure 1 summarizes our multi-horizon tracking framework. First, generated inputs and sequence data (e.g., frame rate) are sent to the *Query and Scene Context Encoder*. For each query, the object motion branch predicts displacement distributions at each horizon using dense context sampled from ConvNeXt stage-1 feature maps [16] around the query, guiding the model to likely vehicle locations (e.g., on the road). For assignment, we create a hypothesis token for every query-candidate pair, plus *continue* and *terminate* hypotheses $[T_1, T_2, \dots, T_{N_C}, T_{\text{cont}}, T_{\text{term}}]$. Each hypothesis integrates the enhanced query state, candidate box geometry, re-ID features, and motion alignment from the predicted distributions. *Self-attention* allows candidates and no-match states to share information before scoring. We then evaluate the predicted motion at each candidate location for assignment. A second inference pass reuses speeds from the first-pass associations to refine trajectories. Finally, multi-horizon association scores are aggregated across overlapping windows to produce complete vehicle tracks.

3.3 Query and Scene Context Encoding

Figure 2 illustrates the construction of the query-context representation s_q . This representation encodes the query vehicle, the local traffic context, and the sequence-specific scale and timing. For each query, we create tokens for the query and its nearest neighbors, describing each detection relative to the query with normalized position, distance, heading, and box size. The tokens are normalized by ACD and processed with neighborhood self-attention to capture local interactions. The output token is merged with the sequence conditioning (ACD, heading, frame rate f , and a ConvNeXt feature at the query location). An MLP

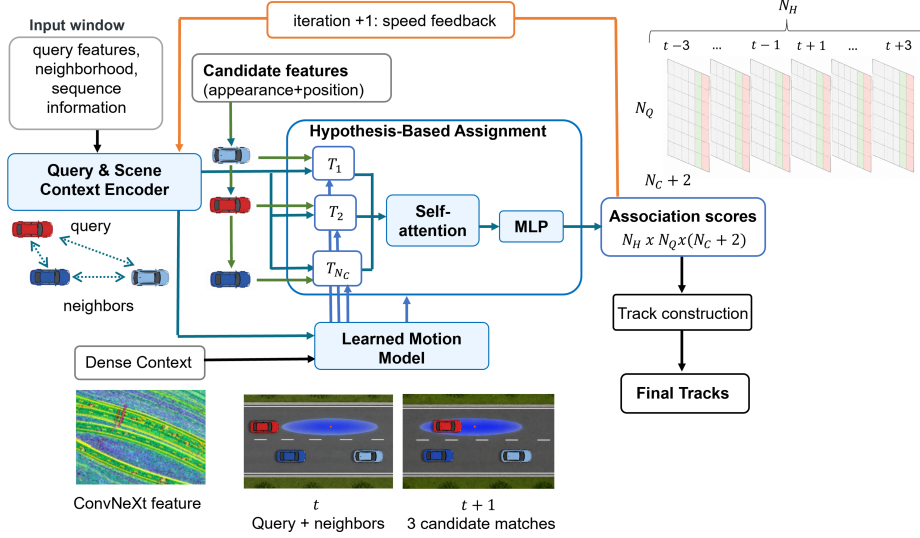


Fig. 1. Overview of **MHA-Track**. The model integrates query context, motion prediction, candidate features, and hypothesis-based assignment to generate association scores, refined by speed feedback in a second pass to produce final tracks.

outputs the final $s_q \in \mathbb{R}^{256}$. In the second inference pass, speeds from first-pass associations for the query and its neighbors are added, giving the encoder short-term trajectory context while keeping the overall representation consistent for motion and assignment modules.

3.4 Learned Motion Model

Figure 3 illustrates our learned motion model for predicting a query vehicle’s movement at each horizon. For every horizon Δh , the model predicts a Gaussian distribution $(\mu_{\Delta h}, \Sigma_{\Delta h})$ over the displacement from the query. We combine s_q with a learned horizon embedding $e_{\Delta h}$ to generate a horizon-specific state $q_{\Delta h}$, enabling distinct motion patterns for past and future horizons using a shared decoder. To add image context, ConvNeXt features are sampled at multiple points around the query, scaled by ACD and aligned with the vehicle’s heading. Each probe forms a token with the feature and its relative geometry; a self-attention block lets probes share information. $q_{\Delta h}$ then attends to these probes, summarizing the image context into a dense motion vector $c_{\Delta h}$. The model also uses a compact displacement prior—a library of learned maneuvers and covariance biases—providing flexible preferences for common vehicle motions. The final Gaussian decoder integrates $q_{\Delta h}$, $c_{\Delta h}$, the motion prior, and optional speed feedback from the second inference pass. Separate MLP heads predict the mean and covariance of the displacement distribution:

$$p_{\Delta h}(d^{uv}) = \mathcal{N}(d^{uv}; \mu_{\Delta h}, \Sigma_{\Delta h}). \quad (1)$$

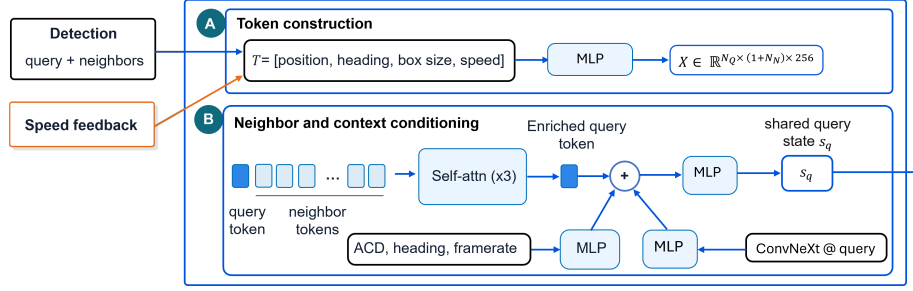


Fig. 2. Query and scene context encoder in Figure 1. The query, same-frame neighbors, sequence information, and a local ConvNeXt feature are combined into a shared query-context representation for motion prediction and assignment.

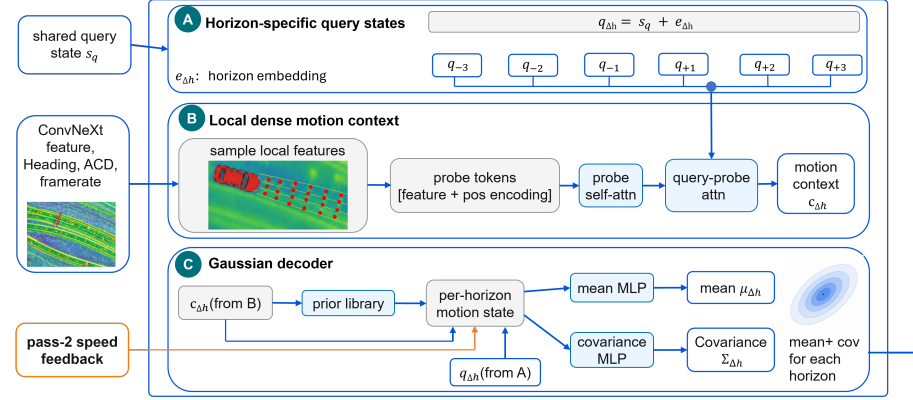


Fig. 3. Learned motion model combining query context, dense image context, displacement priors, and optional speed feedback to predict one Gaussian displacement distribution per horizon, providing motion likelihoods and border-based state cues.

Here, d^{uv} denotes displacement in the query-centered, heading-aligned coordinate system uv . The position branch is trained with the negative log-likelihood of *Ground Truth* (GT) displacement under the predicted Gaussian, encouraging both accuracy and reliable uncertainty estimates. The predicted distribution serves two purposes: (1) candidate locations are scored by motion likelihood for assignment, and (2) the mean position is compared to image borders to generate *in-frame*, *out-of-frame* (OoF), and *keep* cues for downstream assignment.

3.5 Hypothesis-Based Assignment

Figure 4 illustrates the assignment module. For each query q and horizon Δh , the model predicts a flat assignment row over all candidates and two no-match states: *continue* (track continues) and *terminate* (track ends). Each assignment is a hypothesis token, combining s_q with candidate-specific evidence: re-ID appearance, bounding box geometry, relative position, and motion features. Invalid

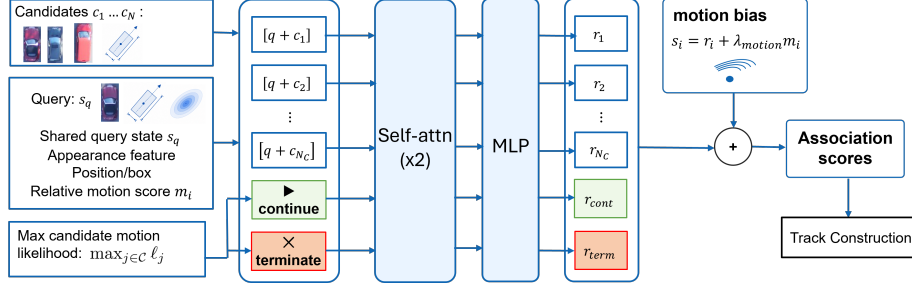


Fig. 4. Hypothesis-based assignment. Candidate, continue, and terminate hypotheses are scored jointly with self-attention. Candidate tokens encode re-ID appearance, oriented box geometry, and motion evidence. The association scores are passed to linking.

candidates are masked. *Continue* and *terminate* have their own learned hypothesis types and use state cues (e.g., *in-frame*, *OoF*). All hypotheses are processed together with self-attention, allowing for direct comparison between candidates and no-match states. A shared linear head outputs raw scores for all hypotheses (r_i, r_{cont}, r_{term}).

The model also evaluates the predicted Gaussian motion distribution at each candidate location. For candidate i , this yields the motion log-likelihood $\ell_i = \log p_{\Delta h}(d_i^{uv})$, where d_i^{uv} is the candidate displacement in the query-centered, heading-aligned coordinate system. We then compute a row-normalized relative motion score $m_i = \ell_i - \max_{j \in \mathcal{C}} \ell_j$, where $\mathcal{C}$ is the candidate set. The best motion-compatible candidate gets $m_i = 0$; others get non-positive scores. This score is used as input for the candidate hypothesis token and as a learned bias on the final logit:

$$s_i = r_i + \lambda_{motion} m_i, \quad i \in \{1, \dots, N_C\}. \quad (2)$$

Here, λ_{motion} is a learned scale controlling how strongly the Gaussian motion likelihood influences candidate assignment. The final assignment is a *softmax* over all candidates, *continue*, and *terminate* scores, so a selects one candidate or a no-match state:

$$p(a | q, \Delta h) = \text{softmax}([s_1, \dots, s_{N_C}, s_{cont}, s_{term}]). \quad (3)$$

The resulting association scores are supervised with *cross-entropy* loss and passed to the final linking stage.

3.6 Two-Pass Inference and Track Construction

At inference, we run the model twice: the first pass estimates query and neighbor speeds, and the second generates association scores for track construction. We aggregate and score link evidence across horizons and windows. This makes linking robust to short occlusions, missed detections, and ambiguous single-frame matches. For each query and horizon, we retain candidate links that are preferred over *continue* and *terminate*. Because windows overlap, the same link can

be proposed multiple times—positive horizons give forward evidence, negatives give reverse confirmation. The final tracks are built in two steps: (1) select reliable $\Delta h = +1$ links to form short tracklets; (2) use stronger $\Delta h = +2$ and $\Delta h = +3$ links to connect tracklets across gaps if they have better support than local alternatives. The final solver ensures that each detection has at most one incoming and one outgoing edge, and that the selected links are chained into MOT-style tracks.

4 Results and Discussion

Experimental setups. We evaluate on the VETRA test split with both sequence-wise and combined results, and additionally report combined validation results; sequence-wise validation results are provided in the supplementary material. All metrics are computed with TrackEval [17], focusing on HOTA [18], MOTA and ID switches [5], and IDF1 [23]. IDF1 measures identity preservation as the F1 score over identity-matched detections,

$$\text{IDF1} = \frac{2 \text{IDTP}}{2 \text{IDTP} + \text{IDFP} + \text{IDFN}}. \quad (4)$$

MOTA equally counts false negatives, false positives, and identity switches relative to the total number of ground-truth detections,

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_t \text{GT}_t}, \quad (5)$$

where GT_t denotes the number of ground-truth detections in frame t , whereas HOTA balances detection and association accuracy across localization thresholds. All methods are evaluated in the detector-based setting using the same detections provided by VETRA [12]. We further report an oracle setting with ground-truth detections to isolate association and track-construction errors. All variants of our method are trained on the VETRA training split for 100 epochs, after which performance stabilizes. Our main baseline is Deep SR-SORT [12], which uses re-ID appearance features from detection-box crops, compensates camera motion before projecting detections into the next frame, filters candidates with a handcrafted speed-based search radius, and applies a SORT-style appearance cost matrix after candidate filtering.

4.1 Overall Performance

Table 1 highlights key findings across three categories.

General-purpose trackers underperform on VETRA. NOOUGAT [20] reports state-of-the-art results on several standard MOT benchmarks and is also the strongest general-purpose tracker in our VETRA comparison, outperforming classical online baselines such as ByteTrack [31], DeepSort [27], Deep-OC-Sort [19] and BoT-SORT [1]. However, it still reaches only 68.0 HOTA and 73.4 IDF1

Table 1. Tracking performance on the VETRA test and val sets. *GT* means using GT detections. All metrics except IDSW are in percent. Best results are bold. All previous method results except for NOOUGAT are from [12].

Method	Split	HOTA $\uparrow$	MOTA $\uparrow$	IDSW $\downarrow$	IDF1 $\uparrow$
ByteTrack [31]	test	36.4	17.8	17,328	13.6
BoT-SORT [1]	test	50.8	48.2	6,886	18.5
Deep SORT [27]	test	58.3	58.1	3,077	76.4
Deep OC-SORT [19]	test	46.8	31.2	10,334	44.7
NOOUGAT [20]	test	68.0	61.5	1,855	73.4
Deep SR-SORT [12]	test	82.2	88.5	792	90.3
MHA-Track (our)	test	84.1	90.1	392	93.3
Deep SR-SORT	val	76.2	78.2	389	78.9
MHA-Track (our)	val	78.2	80.6	243	82.3
Deep SR-SORT _{GT}	test	94.74	93.76	791	95.17
MHA-Track_{GT} (our)	test	97.7	97.9	468	96.9

on VETRA. This shows that methods tuned for dense, high-fps videos struggle with the low-frequency, moving-camera demands of aerial traffic monitoring.

VETRA-specific designs are essential. Deep SR-SORT, with camera motion compensation and adaptive search radius, improves HOTA to 82.2 and IDF1 to 90.3, outperforming NOOUGAT by wide margins. This demonstrates that domain-tailored motion modeling and appearance training are essential for this context. Our **MHA-Track** surpasses Deep SR-SORT by +1.9 HOTA, +1.6 MOTA, and +3.0 IDF1 on the test set. The most significant gain is in IDSW, cut by 50% (from 792 to 392), showing that multi-horizon association with explicit no-match states addresses the main failure mode left by Deep SR-SORT. Similar improvements on the validation split (HOTA +2.0, IDF1 +3.4) confirm these gains generalize. **Detection error is a major constraint.** With GT detections, our method achieves 97.7 HOTA and 97.9 MOTA, compared to 84.1 and 90.1 with predicted detections, a gap of 13.6 and 7.8 percent, indicating that some association errors originate from detection errors, not association logic. Improving aerial vehicle detection remains an important direction for future work.

Runtime. Although this was not the focus of this work and no dedicated real-time optimization was performed, **MHA-Track** remains practical for offline processing. As shown in Table 2, the full post-detection pipeline processes the VETRA test set in 3209.3s on a single NVIDIA TITAN RTX, corresponding to 21.83s per frame on average. Runtime also depends on the number of overlapping windows, so sequences with more detections can still be faster when fewer windows are processed. The maximum observed CUDA memory reserved by PyTorch was 17.1GB. Across all test sequences, ConvNeXt feature extraction required 2.62s per image on average, while BRISK affine estimation required 2.16s per adjacent image pair. Training was performed on a single NVIDIA A100 and completed in approximately 10 hours.

4.2 Sequence-wise Performance

Table 3 shows that gains are consistent but vary by sequence, reflecting VETRA’s diverse challenges. Sequences with moderate framerates, high overlap, and dense

Table 2. Runtime of **MHA-Track** on the VETRA test sequences on an NVIDIA TITAN RTX. Runtime denotes the full post-detection pipeline from detector outputs and images to the final MOT result file, including BRISK affine estimation, ConvNeXt feature extraction, ReID/window preprocessing, association inference, graph linking, and result export. Detector runtime and TrackEval runtime are excluded.

Seq.	Frames	Dets./frame	Resolution	Runtime (Seq.) [s]	Time/frame [s]
18	8	177.1	5616×3744	79.1	9.88
19	6	587.0	5616×3744	199.5	33.26
20	10	489.2	5616×3744	550.5	55.05
21	20	46.8	5184×3456	148.5	7.43
22	17	99.0	5616×3744	202.2	11.89
23	54	6.5	5472×3648	381.5	7.06
24	32	289.9	5472×3648	1648.0	51.50
All	147	150.2	–	3209.3	21.83

Table 3. Performance of **MHA-Track** on the sequences of the VETRA test set. For each sequence, its statistics (GSD, framerate, overlap, per-image average instance number) are also given. *GT* means using GT detections. All metrics, except IDSW, are reported as percentages. Δ means improvement over the baseline Deep SR-SORT. Positive Δ IDSW denotes fewer ID switches.

ID	GSD	FPS	Ov.	Inst.	HOTA	MOTA	IDSW	IDF1	Δ HOTA	Δ MOTA	Δ IDSW	Δ IDF1	HOTA	MOTA	IDSW	IDF1
[cm]	[Hz]	[%]	/img		(GT)	(GT)	(GT)	(GT)					(GT)	(GT)	(GT)	(GT)
18	10	0.42	70	189	81.79	83.60	92	87.53	+1.27	-0.34	-1	+0.76	94.36	92.84	107	91.77
19	14	1.11	88	623	80.29	88.62	42	93.19	+1.55	+2.40	+89	+2.19	98.75	98.62	50	98.29
20	8	1.67	89	498	88.16	95.71	23	97.59	+1.38	+1.12	+88	+1.66	99.37	99.26	37	99.21
21	10	0.76	90	51	80.64	90.68	29	90.04	+5.57	+4.72	+52	+9.95	94.00	96.25	35	91.43
22	16	1.00	86	120	83.82	87.68	18	93.25	+0.00	-0.19	-2	-0.03	99.28	99.15	14	99.09
23	3	2.39	80	6	74.87	63.19	3	79.70	-0.72	+2.10	+4	+0.42	99.39	99.67	1	99.02
24	9	0.98	88	322	84.42	89.98	185	92.78	+2.60	+1.93	+170	+4.23	96.91	97.65	224	95.98

objects (**Seq 19, 20, 24**) have the largest reductions in identity switches. In contrast, **Seq 18** (ultra-low framerate) and **Seq 23** (high-resolution, sparse) reveal the method’s limitations. **Seq 21** illustrates the value of multi-horizon association (+5.57 HOTA, +9.95 IDF1, 52 fewer IDSW), with two bridge occlusions. Confusion plots in Figure 5 (a,b) show the baseline fragments identities after occlusion, creating new tracks instead of recovering the originals. Our method leverages longer-horizon evidence: while the 75% confidence ellipses in Figure 6 (b) are not perfectly centered, they capture the likely trajectory through the bridge, acting as a motion prior that reduces wrong associations and aids identity recovery after occlusion.

Seq 24 (+2.60 HOTA, +4.23 IDF1, 170 fewer IDSW) improves with dense neighborhood context and pass-2 query-speed feedback, offering strong trajectory evidence through sharp turns. As Figure 6 (c,e) shows, the model handles directional ambiguity by inflating confidence ellipses, generating uncertainty-aware predictions that still include the vehicle’s next position and maintain robust association under sudden heading changes. **Seq 19** (+89 fewer IDSW) features many identical vehicles on a freight train in close proximity (Figure 5 (e,f)), making appearance-based re-ID ineffective. Here, motion cues and neighborhood context enable identity distinction and reduce IDSW. **Seq 20** presents a mostly static

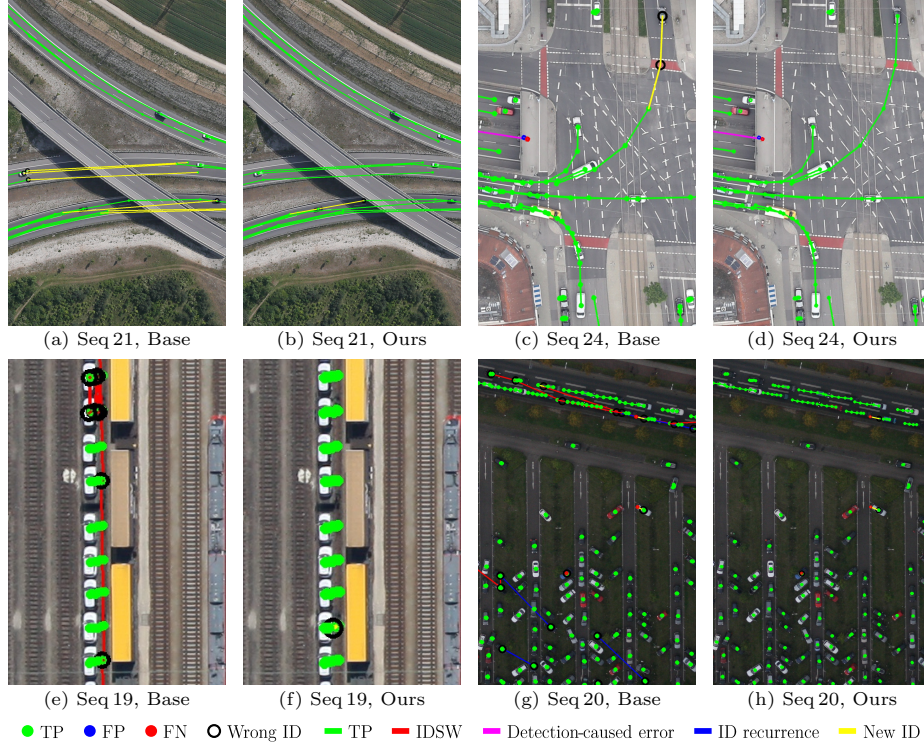


Fig. 5. Comparison of confusion plots between Deep SR-SORT baseline and our **MHA-Track**. Green dots and lines indicate correct detections and associations. Non-green elements indicate different errors.

scene where the model predicts minimal offsets with small covariance ellipses (Figure 6(d)). Query history and neighborhood context prevent unnecessary identity changes, as confirmed by the confusion plots in Figure 5(g,h).

Using GT boxes instead of predicted detections significantly improves all sequences, confirming that detection errors account for most residual mistakes. Importantly, our method clearly outperforms Deep SR-SORT in this ideal setting, showing that association and track construction add value beyond detection accuracy. **Seq 23** best demonstrates the detection bottleneck: it is among the hardest with predicted detections (MOTA 63.19), but becomes one of the easiest when using GT (MOTA 99.67). Its high GSD of 3 cm puts it in a scale poorly covered during detector training, which mainly used lower-resolution EAGLE dataset [2]. As shown in Figure 6(a), using ground-truth boxes as input leads to reliable position and OoF predictions by the motion model. This supports the interpretation that detection quality is the main bottleneck for sequence 23 in the detector-based setting. **Seq 18** is difficult even with GT boxes because its low frame overlap and ultra-low framerate (0.42 Hz) allow only one input window, with no overlapping support, mirroring the performance drop for larger

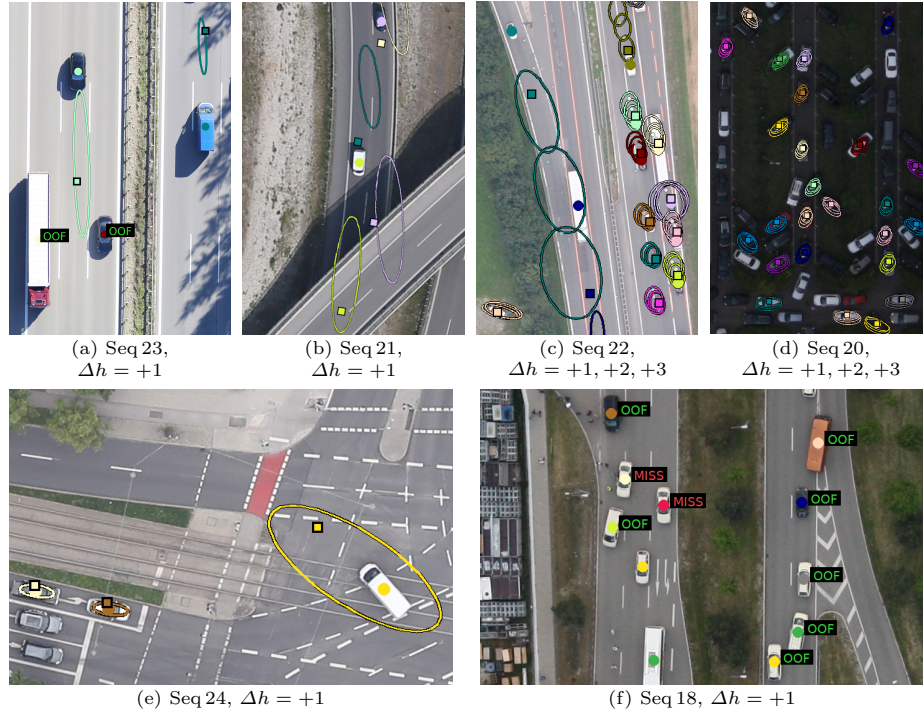


Fig. 6. Examples of predicted 75% Gaussian confidence ellipses and out-of-frame (OoF) estimates using ground-truth boxes as input. Green and red OoFs indicate true and false positive OoF predictions, and MISS means false negative OoF prediction. Square markers show GT position at Δh . (a,b,e,f) show single-horizon and (c,d) show multi-horizon predictions.

degrees in Table 4. As shown in Figure 6 (f), most vehicles are correctly predicted as OoF, though motion can be overestimated due to past-speed bias. **Seq 24** is still challenging with perfect detections, as tunnel disappearances exceed 20 seconds, beyond our model’s short-horizon gap-refinding scope. **Seq 22** is lower resolution, but our model performs well with both GT and predicted detections. Overall, **MHA-Track** excels at short-term association and gap recovery, while long disappearances and detection errors remain key failures.

4.3 Ablation Study

Table 4 highlights that Gaussian motion likelihood is crucial for assignment, removing it drops IDF1 from 93.29 to 87.86. Using only the mean, not the full Gaussian, undermines robust association. Hypothesis self-attention further boosts performance by jointly comparing all hypotheses. Adding more Gaussian components provides no benefit, indicating that VETRA vehicle motion is mostly unimodal. Speed feedback and dense context yield consistent but smaller gains; removing all pass-2 speed feedback has a larger impact, confirming that

Table 4. Ablation results on the VETRA test set. Model, dense-context, and speed-feedback ablations modify the prediction stage, while linking ablations keep the trained model fixed and change only deterministic track construction. All metrics except IDSW are reported in percent.

Ablation	Variant	HOTA $\uparrow$	MOTA $\uparrow$	IDSW $\downarrow$	IDF1 $\uparrow$
Model / inference	w/o hypothesis self-attention	83.67	89.82	454	92.69
	w/o motion likelihood	80.73	85.88	1341	87.86
	w/o border priors in hypothesis tokens	84.04	90.03	406	93.13
	w/o query-speed conditioning	83.77	89.96	423	92.72
	MoG K=2	84.03	89.96	423	93.20
	MoG K=3	84.00	90.00	414	93.09
Dense-context	w/o dense branches	83.67	89.78	464	92.57
	zero dense features	84.00	89.87	442	93.09
Inference only speed removal	w/o pass-2 speed feedback	83.62	89.68	485	92.54
	w/o neighbor speed feedback	83.89	89.88	441	92.92
	w/o query speed feedback	83.88	89.92	431	92.94
Linking	max-confidence linking	83.61	88.76	693	92.57
	+1 only linking	82.86	89.10	615	91.03
	every 2nd window	83.79	89.69	401	93.04
	every 4th window	79.37	83.50	400	89.58
Full method		84.13	90.10	392	93.29

query and neighbor speeds are complementary. Dense context helps moderately: zeroing features is better than removing the branch, so branch structure matters even if features are weak. Among the tested dense features, ConvNeXt outperformed DINOv3-SAT493M [24] and SatlasPretrain [4] features when used with probe self-attention. Linking ablations show that support-based track construction is key. Max-confidence and +1-only linking reduce IDF1, confirming the benefit of aggregating multi-horizon evidence and using +2/ +3 links for gap recovery. Stride 2 remains robust, while stride 4 drops performance, highlighting the need for temporal support, especially in short sequences such as [Seq. 18](#).

5 Conclusions

We proposed **MHA-Track** for vehicle tracking in high-resolution low framerate aerial image sequences. **MHA-Track** performs multi-horizon association within windows of camera-motion-aligned frames, combining learned Gaussian motion estimation, assignment with explicit no-match states, and evidence aggregation across overlapping temporal offsets to robustly recover vehicle identity under occlusion and large displacements. On the VETRA benchmark, **MHA-Track** improves over baseline by +1.94 HOTA, +1.65 MOTA, and +2.97 IDF1, and reduces identity switches from 792 to 392. Gains are strongest in short-term occlusions, such as bridge passages, where multi-horizon evidence recovers lost vehicles, and in scenes with many similar vehicles or sharp turns, where uncertainty-aware motion prediction and neighborhood context reduce false associations. Most remaining errors stem from detection failures, especially at underrepresented spatial resolution, or from disappearances beyond the three-frame association horizon. Therefore, detector robustness and long-term re-identification are key directions for future work.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: Bot-sort: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)
2. Azimi, S.M., Bahmanyar, R., Henry, C., Kurz, F.: Eagle: Large-scale vehicle detection dataset in real-world scenarios using aerial imagery. In: ICPR. pp. 6920–6927 (2021). <https://doi.org/10.1109/ICPR48806.2021.9412353>
3. Barmpounakis, E.N., Geroliminis, N.: On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment. *Transportation Research Part C: Emerging Technologies* **111**, 50–71 (2020). <https://doi.org/10.1016/j.trc.2019.11.023>
4. Bastani, F., Wolters, P., Gupta, R., Ferdinando, J., Kembhavi, A.: SatlasPretrain: A large-scale dataset for remote sensing image understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16772–16782 (2023)
5. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing* **2008**, 1–10 (2008)
6. Butilă, E.V., Boboc, R.G.: Urban traffic monitoring and analysis using unmanned aerial vehicles (uavs): A systematic literature review. *Remote Sensing* **14**(3), 620 (2022). <https://doi.org/10.3390/rs14030620>
7. Cui, Y., Zeng, C., Zhao, X., Yang, Y., Wu, G., Wang, L.: Sportsmot: A large multi-object tracking dataset in multiple sports scenes. arXiv preprint arXiv:2304.05170 (2023)
8. 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. *IJCV* **129**(4), 845–881 (2021). <https://doi.org/10.1007/s11263-020-01393-0>, <https://doi.org/10.1007/s11263-020-01393-0>
9. Gao, R., Wang, L.: MeMOTR: Long-term memory-augmented transformer for multi-object tracking. In: ICCV. pp. 9901–9910 (October 2023)
10. Gao, Y., Xu, H., Li, J., Wang, N., Gao, X.: Multi-scene generalized trajectory global graph solver with composite nodes for multiple object tracking. *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(3), 1842–1850 (Mar 2024). <https://doi.org/10.1609/aaai.v38i3.27953>
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
12. Hellekes, J., Mühlhaus, M., Bahmanyar, R., Azimi, S.M., Kurz, F.: Vetra: A dataset for vehicle tracking in aerial imagery – new challenges for multi-object tracking. In: ECCV. p. 52–70. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-73013-9_4, https://doi.org/10.1007/978-3-031-73013-9_4
13. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together (2024), <https://arxiv.org/abs/2307.07635>
14. Leica Geosystems: Leica DMC III: Digital mapping camera datasheet. https://leica-geosystems.com/-/media/files/leicageosystems/products/datasheets/leica_dmcciii_ds.ashx (2016), accessed: 2026-05-28
15. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: ICCV. pp. 2548–2555 (2011). <https://doi.org/10.1109/ICCV.2011.6126542>

16. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986 (2022)
17. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: Trackeval: Hota and clear mot evaluation code. <https://github.com/JonathonLuiten/TrackEval> (2020), accessed: 2026-05-27
18. Luiten, J., Osep, 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**(2), 548–578 (2021)
19. Maggolino, G., Ahmad, A., Cao, J., Kitani, K.: Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. *ICIP* pp. 3025–3029 (2023), <https://api.semanticscholar.org/CorpusID:257102808>
20. Missaoui, B., Cetintas, O., Brasó, G., et al.: Noougat: Towards unified online and offline multi-object tracking. *International Journal of Computer Vision* **134**(1), 272 (2026). <https://doi.org/10.1007/s11263-026-02858-4>, <https://doi.org/10.1007/s11263-026-02858-4>
21. Nie, Z., Xue, L., Fang, Z., Tang, C., Wang, X., Zhang, H.: M3ot: A multi-drone multi-modality dataset for multi-object tracking. *Scientific Data* **12**(1), 1927 (2025). <https://doi.org/10.1038/s41597-025-06204-0>, <https://doi.org/10.1038/s41597-025-06204-0>
22. Phase One: iXM-RS150F: The 150mp aerial camera solution. <https://www.phaseone.com/solutions/geospatial-solutions/components/ixm-rs150f/> (2024), accessed: 2026-05-28
23. Ristani, E., Solera, F., Zou, R.S., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: *European Conference on Computer Vision Workshops*. pp. 17–35. Springer (2016)
24. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3. *arXiv preprint arXiv:2508.10104* (2025)
25. Sun, P., Cao, J., Jiang, Y., Yuan, Z., Bai, S., Kitani, K., Luo, P.: Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In: *CVPR* (2022)
26. Vexcel Imaging: UltraCam Eagle 4.1: Push the limits of your aerial missions. https://www.vexcel-imaging.com/brochures/UC_Eagle.4.1-en.pdf (2022), accessed: 2026-05-28
27. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: *ICIP*. pp. 3645–3649. IEEE (2017). <https://doi.org/10.1109/ICIP.2017.8296962>
28. Yin, Q., Hu, Q., Liu, H., Zhang, F., Wang, Y., Lin, Z., An, W., Guo, Y.: Detecting and tracking small and dense moving objects in satellite videos: A benchmark. *IEEE Transactions on Geoscience and Remote Sensing* (2021)
29. Yin, Q., Hu, Q., Liu, H., Zhang, F., Wang, Y., Lin, Z., An, W., Guo, Y.: Detecting and tracking small and dense moving objects in satellite videos: A benchmark. *IEEE Transactions on Geoscience and Remote Sensing* (2021)
30. Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: Motr: End-to-end multiple-object tracking with transformer. In: *ECCV* (2022)
31. 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*. pp. 1–21. Springer (2022)

32. Zhang, Y., Wang, T., Zhang, X.: MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors . In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22056–22065. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2023). <https://doi.org/10.1109/CVPR52729.2023.02112>, <https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.02112>
33. Zhou, K., Xiang, T.: Torchreid: A library for deep learning person re-identification in pytorch. arXiv preprint arXiv:1910.10093 (2019)
34. Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., Ling, H.: Detection and tracking meet drones challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(11), 7380–7399 (2021)