SelfMOTR: Revisiting MOTR with Self-Generating Detection Priors

Fabian Gülhan[1,3,4✉⋆], Emil Mededovic[1✉⋆], Yuli Wu[1,2], and Johannes Stegmaier[1,2✉]

[1] Chair of Imaging and Computer Vision, RWTH Aachen University, Germany
[2] Heinrich Heine University Düsseldorf, Faculty of Mathematics and Natural Sciences, Machine Learning for Medical Data, Germany
[3] Institute of Pathology, Technical University of Munich, Germany
[4] Munich Center for Machine Learning (MCML), Munich, Germany
✉ Corresponding authors: fabian.guelhan@tum.de, emil.mededovic@lfb.rwth-aachen.de, johannes.stegmaier@hhu.de

Abstract. End-to-end transformer architectures have driven significant progress in multi-object tracking by unifying detection and association into a single, heuristic-free framework. Despite these benefits, poor detection performance and the inherent conflict between detection and association in a joint architecture remain critical concerns. Recent approaches aim to mitigate these issues by employing advanced denoising or label assignment strategies, or by incorporating detection priors from external object detectors. In this paper, we propose SelfMOTR, a simple yet highly effective detector-free alternative that decouples proposal discovery from association using self-generated internal detection priors. Through extensive analysis and ablation studies, we show that end-to-end transformer trackers with joint detection–association decoding retain substantial hidden detection capacity, and we provide a practical detector-free mechanism for leveraging it. To shed light on these joint decoding dynamics, we draw inspiration from attention sink analyses in large language models, leveraging Track Attention Mass to show that standard generic queries exhibit unbalanced attention, frequently struggling to weigh track context against novel object discovery. SelfMOTR achieves highly competitive performance in complex, dynamic environments, yielding 69.2 HOTA on DanceTrack and leading with 71.1 HOTA on the Bird Flock Tracking (BFT) dataset. Project page: https://medem23.github.io/SM.

Keywords: Multi-object tracking · End-to-end tracking · Detection priors

1 Introduction

Multi-object tracking (MOT) aims to localize all instances of interest in a video and maintain their identities over time [3, 4, 28, 37]. While classical tracking-by-detection pipelines achieve strong performance by pairing high-quality detectors

⋆ Equal contribution

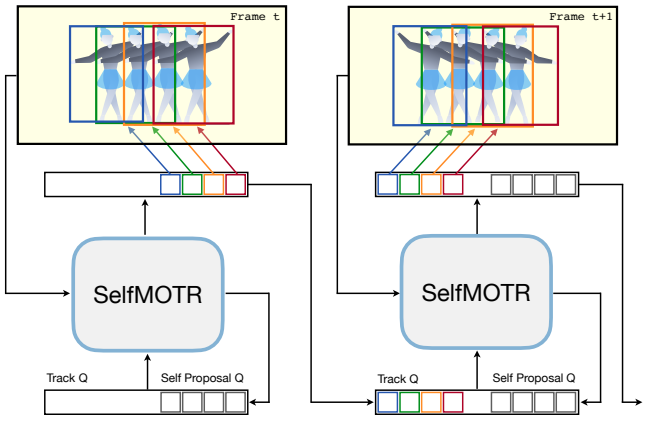


Fig. 1: SelfMOTR. A detection-only pass produces self-proposal queries, which are then fused with propagated track queries in the tracking pass to mitigate detection–association interference without external detectors.

with well-engineered association modules, their reliance on disjointed components often necessitates heavy, task-specific tuning [1, 10, 24, 42, 43]. End-to-end transformer architectures emerged as an elegant alternative. By extending query-based detection models [6, 21, 54] with temporal propagation, frameworks such as MOTR [45] unify discovery and association into a single learnable process, thereby eliminating complex, multi-stage pipelines.

However, subsequent analyses have consistently reported two limitations of end-to-end transformer trackers: (i) their detection accuracy lags behind dedicated detectors on the same data, and (ii) the joint optimization of detection and association creates a supervision imbalance in which track queries receive most of the labels, while detect queries are comparatively under-trained. This detection–association conflict has been explicitly addressed in follow-up work through more generous label assignment [41,44], decoder-level denoising [41], or, most effectively, by injecting external detection priors from pretrained detectors [44,49] to relieve the burden on the transformer decoder. These strategies demonstrate that high-quality priors are a powerful inductive bias for end-to-end MOT, but they also reintroduce an extra model and partly undermine the self-contained character of end-to-end tracking.

In this work, we revisit the source of the detection prior itself. A key empirical observation, also reported in CO-MOT [41], is that joint detection–association transformers can exhibit substantially stronger detection performance when track queries are removed at inference. This confirms that the bottleneck is not a lack of representational capacity, but rather direct interference between detect and track queries during joint decoding. To shed light on this mechanism, we introduce Track Attention Mass, a decoder self-attention diagnostic that measures the fraction of attention mass assigned from detect queries to track queries. Using this lens, we find that in the absence of explicit spatial priors, generic detect

queries often exhibit unbalanced self-attention: a substantial subset becomes overly track-dominant, while others exhibit highly localized interactions within the non-track pool. These behaviors are consistent with reduced effective capacity for novel object discovery under joint decoding. Motivated by these diagnostics, we propose SelfMOTR (see Fig. 1), which uses the model's own detection-only forward mode to generate spatial priors and reuse them as proposal queries in the tracking pass. Unlike approaches that import priors from external detectors, SelfMOTR produces proposals natively aligned with the backbone and decoder parameterization. By supplying natively aligned priors without external detectors, this approach mitigates query interference, leading us to advocate internal prior generation as a default design principle for unified tracking.

2 Related Works

Offline Graph-Based Tracking. Multi-object tracking methods are commonly divided into offline and online approaches. Offline trackers process entire video segments, enabling global association graphs that link detections across multiple frames. GNN-based methods such as SUSHI [7] and CoNo-Link [13] construct global spatio-temporal graphs over full sequences and achieve strong long-range association. Related works extend this paradigm to automated labeling (SPAM [8]) and arbitrary temporal horizons (NOOUGAT [26]). However, because these methods rely on global graph construction and optimization, they cannot operate as strictly online, real-time trackers.

Heuristic-Based Online Tracking. Online trackers have traditionally relied on a detect-then-track heuristic. Tracking-by-detection pipelines, popularized by SORT [3], extract bounding boxes using an external detector, predict motion via Kalman Filters [16], and associate targets using Hungarian matching [18]. Successors like StrongSORT [10] and OC-SORT [5] refine these motion and appearance cues, while GHOST [31] demonstrates the power of domain-adapted re-identification features. To handle occlusions, methods like ByteTrack [47], BoostTrack [34], and TrackTrack [33] actively recover low-confidence or NMS-suppressed boxes to aid association. Alternatively, single-stage joint trackers [38,48] extract boxes and identity embeddings in a single pass for speed, though often at the cost of accuracy. Ultimately, both of these families remain bottlenecked by their reliance on external detectors and hand-crafted association rules.

End-to-End Transformer Tracking. End-to-end transformer trackers [25,36, 45] aim to eliminate the detect-then-track split entirely by propagating queries over time, allowing a single model to produce both detections and identities.

An early step in this direction is TransTrack [36], which adopts a dual-decoder architecture: a detection decoder processes learned object queries to detect new objects, while a propagation decoder uses track queries to maintain existing trajectories. The outputs of the two decoders are then associated through IoU matching [18]. In contrast, TrackFormer [25] processes new object queries and propagated track queries jointly within a single transformer decoder, although it still relies on post-processing and remains sensitive to hyperparam-

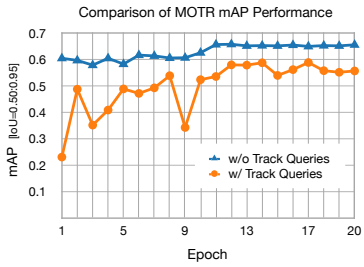


Fig. 2: MOTR is trained on DanceTrack [35] using the standard tracking setup. At each training checkpoint, we compare standard inference with track queries (circles) to a detection-only inference mode where track queries are disabled (triangles).

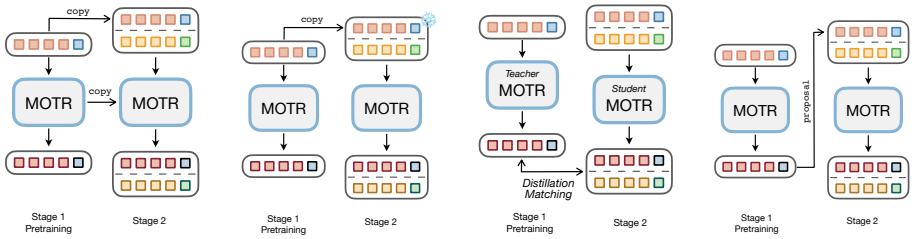
eters. MOTR [45] systematizes this concept with a Query Interaction Module that learns to manage query interactions. Several memory-augmented variants, such as MeMOTR [12] and SambaMOTR [30], introduce supplementary memory modules after the core tracking step to improve long-term identity preservation, though this comes at the cost of extra components. Subsequent works sought to resolve the inherent detection deficits of this pipeline by re-introducing external priors. MOTRv2 [49] augments MOTR by aligning queries to an external detector, ensuring track queries stay tied to reliable image boxes. MOTRv3 [44] takes a further step toward stability by utilizing YOLOX [14] for soft-label distillation during training, alongside additional refinements. CO-MOT [41], in turn, avoids external detectors entirely by introducing shadow queries to keep extra candidate targets alive, although this requires complex query-management logic.

Unlike approaches that rely on external detectors [44, 49] or require additional query-management logic for extra queries [41], SelfMOTR generates its own spatial proposals internally. By leveraging its own encoder features and a lightweight detection-only forward pass, the tracker generates self-proposals that are natively aligned with the decoder's parameterization. This yields a remarkably simple, detector-free, and fully end-to-end framework that preserves robust novel object detection while keeping the tracking pipeline elegantly compact.

3 Method

3.1 Latent Detection and Prior Injection

To study whether MOTR truly contains a usable detector, we first reproduce standard MOTR training on DanceTrack [35] and evaluate the model after each epoch under two inference modes: (i) the original MOTR inference with track queries enabled, and (ii) the same checkpoints evaluated with track queries disabled. The resulting curves (Fig. 2) reveal a consistent and substantial gap: removing track queries at inference yields higher and more stable mAP throughout training. These findings suggest that MOTR learns a strong detection signal early on, but that this signal is partially suppressed once temporal association is



(a) Detection Pretraining **(b)** Query Pretraining **(c)** Distillation **(d)** Anchor Proposal

Fig. 3: Four ways of injecting detection priors into MOTR. (a) *Detection Pretraining:* Train MOTR as a detector on the target dataset with track queries disabled, then fine-tune in the standard tracking setup, transferring the full model. (b) *Query Pretraining:* Initialize tracking using only the detection query embeddings from the detection-pretrained MOTR; train the backbone and decoder from scratch. (c) *Distillation:* Use a frozen detection-pretrained MOTR as a teacher to generate boxes; train a student MOTR with the standard MOT loss plus a Hungarian-matched distillation loss. (d) *Anchor Proposal:* Instead of learning decoder anchors from scratch, bounding boxes from the detection-pretrained MOTR are converted into 4D anchor boxes and refined in a lightweight decoder pass.

introduced in the decoder. This observation immediately raises three questions: (i) is this latent detection capability actually useful for tracking, (ii) can it be injected into the MOTR training process in a controlled manner, and (iii) what is the most effective mechanism for doing so? To answer these questions, we conduct an extensive set of experiments.

Detection Pretraining. As a first step, we explicitly train MOTR as a detector on the target tracking dataset. We disable track queries and optimize only the detect queries together with the Deformable DETR [54] architecture. This produces a MOTR-family detector that is matched to the data domain and exposes the *hidden* detection capability independently of tracking. When we subsequently run the standard MOTR tracking training from this detection-pretrained initialization (Fig. 3a), we observe small but consistent gains over the baseline on DanceTrack (HOTA 51.2 → 51.5, DetA 68.8 → 69.0 in Tab. 1). This shows that reusing a domain-aligned detector in MOTR is beneficial.

Query Pretraining. Detection pretraining transfers the entire detection model, including parameters that are not necessarily aligned with the tracking representation. We therefore consider a targeted variant, in which only the pretrained detect queries are transferred (Fig. 3b) to the tracking training and additionally frozen. This preserves the detection bias in the query space without imposing full feature-space alignment and aims to ease the detection-association conflict by freezing the detection prior. As reported in Tab. 1, this strategy improves association-related metrics noticeably (AssA 38.4 → 41.1, IDF1 49.1 → 51.9), while the detection score (DetA) remains comparable. Hence, initializing the detect queries alone emerges as an effective way to inject a detection-aware prior.

Table 1: Analysis on injecting detection capability into MOTR using the MOTR detection pre-trained model on the DanceTrack [35] validation set. We compare detector pretraining, query pretraining, distillation, and anchor proposal strategy.

Method	**Metrics**				
	HOTA ↑	DetA ↑	AssA ↑	IDF_1 ↑	MOTA ↑
MOTR	51.2	68.8	38.4	49.1	74.4
+ Det. Pret.	$51.5^{\uparrow 0.3}$	$69.0^{\uparrow 0.2}$	$38.7^{\uparrow 0.3}$	$49.4^{\uparrow 0.3}$	$73.8^{\downarrow 0.6}$
+ Query Pret.	$52.7^{\uparrow 1.5}$	$68.0^{\downarrow 0.8}$	$41.1^{\uparrow 2.7}$	$51.9^{\uparrow 2.8}$	$72.5^{\downarrow 1.9}$
+ Distillation	$52.1^{\uparrow 0.9}$	$\mathbf{72.0}^{\uparrow 3.2}$	$37.9^{\downarrow 0.5}$	$50.8^{\uparrow 1.7}$	$\mathbf{80.1}^{\uparrow 5.7}$
+ Anchor Proposal	$\mathbf{58.0}^{\uparrow 6.8}$	$71.2^{\uparrow 2.4}$	$\mathbf{47.5}^{\uparrow 9.1}$	$\mathbf{59.5}^{\uparrow 10.4}$	$79.2^{\uparrow 4.8}$

Distillation. The two strategies above inject a detection prior only before tracking training and do not explicitly preserve detection quality during learning. To enforce the prior throughout training, we add a detection distillation objective. A detection-pretrained MOTR is frozen as a teacher and produces pseudo-boxes; a student MOTR is trained from scratch with the standard collective average loss (CAL) [45] plus a Hungarian-matched distillation term:

$$\mathcal{L} = \mathcal{L}_{\text{MOTR}} + \lambda\, \mathcal{L}_{\text{distill}}, \qquad \mathcal{L}_{\text{distill}} = \sum_{(i,j)\in\pi^*} \ell\big(\hat{y}_i^{\,s},\, \hat{y}_j^{\,t}\big)\,, \tag{1}$$

where π^* is the Hungarian assignment between student predictions and teacher detections (Fig. 3c). This yields a clear boost in detection-oriented metrics (DetA 68.8→72.0, MOTA 74.4→80.1) while maintaining overall tracking performance (Tab. 1), showing that the detection prior can be enforced during training.

Anchor Proposal. Although these techniques improve how the detection prior is injected, detections are still generated during the same tracking decoder pass, which is exactly where we observe query interference. To decouple proposal generation from tracking, we reuse the detection-pretrained MOTR to provide explicit 4D anchor boxes. Instead of learning decoder anchors from scratch, we extract teacher bounding boxes and use them as anchors during both training and inference (Fig. 3d). This yields the largest gain in our ablation: HOTA 51.2 → 58.0 (+6.8), AssA 38.4 → 47.5 (+9.1), and IDF1 49.1 → 59.5 (+10.4) (Tab. 1, + Anchor Proposal). Hence, making the detection prior explicit as 4D anchor proposals is the most effective standalone way to exploit detection priors (Tab. 1).

However, all these variants still rely on an externally trained MOTR detection model that is frozen and reused as a source of bounding boxes. This is conceptually similar to [49].

3.2 SelfMOTR

Motivation. Instead of querying a frozen external detector for 4D anchors, we let the tracking model generate and consume its own anchors at every frame. As

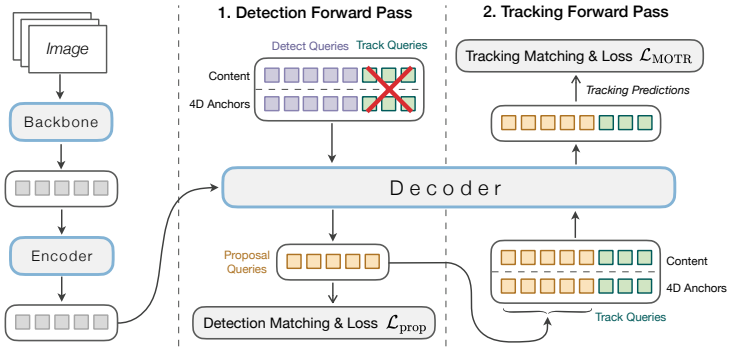


Fig. 4: Overview of **SelfMOTR**. The figure shows how, at each frame, we first use MOTR's detection branch to produce boxes and scores, convert the confident ones into self-proposal queries (4D anchor + confidence-conditioned content), and then concatenate these proposals with the track queries so that the shared decoder can refine and associate them in the tracking pass.

illustrated in Fig. 4, SelfMOTR first runs a detection-only pass that produces self proposals, and then feeds these proposals, together with track queries, into a shared tracking decoder. In this way, proposal generation is fully internal, the pipeline remains end-to-end.

Proposal generation. Given encoder features for the current frame, we run a decoder pass with detect queries and obtain N_{det} detection predictions

$$\mathcal{P} = \{\hat{\mathbf{y}}_k\}_{k=1}^{N_{\text{det}}}, \quad \hat{\mathbf{y}}_k = (\hat{\mathbf{b}}_k^{\text{det}}, \hat{c}_k^{\text{det}}),$$

where $\hat{\mathbf{b}}_k^{\text{det}} = (\hat{b}_k^x, \hat{b}_k^y, \hat{b}_k^w, \hat{b}_k^h)$ is a 4D anchor and $\hat{c}_k^{\text{det}} \in [0,1]$ is the confidence score predicted by the detection head.

We retain only confident detections using a proposal threshold c_{prop},

$$\mathcal{Q} = \left\{(\hat{\mathbf{b}}_k^{\text{det}}, \hat{c}_k^{\text{det}}) : \hat{c}_k^{\text{det}} \geq c_{\text{prop}}\right\}_{k=1}^{N_{\text{prop}}},$$

and convert each element into a proposal query token. Following the notation in Fig. 4, the positional part of the k-th proposal query is simply the predicted 4D anchor

$$\mathbf{z}_{\text{pos},k}^{\text{prop}} = \hat{\mathbf{b}}_k^{\text{det}} \in \mathbb{R}^4,$$

while its content part is a confidence-modulated embedding

$$\mathbf{z}_{\text{content},k}^{\text{prop}} = \mathbf{q}_{\text{s}}^{\text{prop}} + \text{PE}(\hat{c}_k^{\text{det}}),$$

where $\mathbf{q}_{\text{s}}^{\text{prop}} \in \mathbb{R}^d$ is a learned shared proposal query and $\text{PE} : \mathbb{R} \to \mathbb{R}^d$ is a sine–cosine positional encoding of the confidence.

To increase robustness to missed detections, we append a small fixed set of learned proposal anchors ($N_{\text{learned}} = 10$) to the filtered proposals, obtaining the final proposal-query set

$$\mathcal{Q}^{\text{prop}} = \left\{(\mathbf{z}_{\text{pos},k}^{\text{prop}}, \mathbf{z}_{\text{content},k}^{\text{prop}})\right\}_{k=1}^{N_{\text{prop}}+N_{\text{learned}}}. \tag{2}$$

Proposal Supervision. We supervise the detection-only pass directly with the ground-truth boxes of the current frame using the standard set-based matching of Deformable DETR [54]. Let $\mathcal{G} = \{\mathbf{g}_j\}_{j=1}^{N}$ denote the ground-truth objects. We compute a Hungarian matching [18] between $\mathcal{P}$ and $\mathcal{G}$ using the usual combination of classification and box regression costs, and define the proposal loss

$$\mathcal{L}_{\text{prop}} = \sum_{k=1}^{N_{\text{det}}} \big(\lambda_{\text{focal}}\, \mathcal{L}_{\text{focal}}(\hat{c}_k^{\text{det}}, \mathbf{g}_{\pi(k)}) + \lambda_{\ell_1}\, \mathcal{L}_{\ell_1}(\hat{\mathbf{b}}_k^{\text{det}}, \mathbf{g}_{\pi(k)}) + \lambda_{\text{giou}}\, \mathcal{L}_{\text{giou}}(\hat{\mathbf{b}}_k^{\text{det}}, \mathbf{g}_{\pi(k)}) \big), \tag{3}$$

where $\mathcal{L}_{\text{focal}}$ is the focal classification loss, $\mathcal{L}_{\ell_1}$ is the ℓ_1 regression loss on the bounding box parameters, $\mathcal{L}_{\text{giou}}$ is the generalized IoU loss, $\lambda_{\text{focal}}, \lambda_{\ell_1}, \lambda_{\text{giou}}$ balance the terms, and π denotes the optimal matching.

Feeding proposals into tracking. At frame t, we first run the detection-only pass to obtain self-proposals $\mathcal{Q}_t^{\text{prop}}$ as seen in Eq. 2. We then convert them to query embeddings and concatenate them with MOTR's track queries produced by QIM from frame $t-1$. The input query set for the tracking decoder becomes

$$\mathcal{Q}_t = \mathcal{Q}_{t-1}^{\text{track}} \cup \mathcal{Q}_t^{\text{prop}},$$

where $\mathcal{Q}_{t-1}^{\text{track}}$ contains the propagated track queries $(\mathbf{z}_{\text{pos}}^{\text{track}}, \mathbf{z}_{\text{content}}^{\text{track}})$ from the previous frame and $\mathcal{Q}_t^{\text{prop}}$ are the proposal queries. The shared decoder then performs standard decoding on this joint set. Crucially, because the proposals have already been generated in a separate detection-only pass that does not see track queries, the tracking decoder no longer needs to spend capacity on discovering new objects and can focus on refining and associating the combined proposal and track queries.

Training Objective. During training, both passes share parameters and are optimized jointly. The final training loss is the sum of the standard MOTR [45] tracking loss and the proposal loss:

$$\mathcal{L} = \mathcal{L}_{\text{MOTR}} + \lambda_{\text{prop}}\, \mathcal{L}_{\text{prop}}, \tag{4}$$

where $\lambda_{\text{prop}} = 0.5$ controls the relative weight of the detection-only proposal supervision.

4 Experiments

4.1 Datasets & Metrics

DanceTrack. DanceTrack [35] is a large-scale dataset for multi-person tracking in dynamic group dancing scenarios. It is characterized by objects with highly uniform appearance and diverse, non-linear motion patterns, encouraging MOT methods to rely less on visual distinctiveness and more on motion pattern modeling.

Bird Flock Tracking (BFT). BFT [51] dataset includes 106 clips from the

BBC documentary series Earthflight [9]. It is specifically designed to challenge multi-object tracking algorithms through the severe, continuous target deformation and unpredictable aerial maneuvers of fast-moving bird flocks.

AnimalTrack. AnimalTrack [46] is a multi-animal tracking benchmark containing 58 manually labeled sequences that capture diverse objects and environments. Although smaller than datasets like DanceTrack or BFT, it challenges existing tracking algorithms with highly occluded scenes, unpredictable motion, and a dense concentration of objects, averaging 33 targets per sequence.

Metrics. For evaluation, we primarily adopt the Higher Order Tracking Accuracy (HOTA) metric [23], as it provides a balanced measure of detection accuracy (DetA) and association accuracy (AssA), and more generally decomposes performance into complementary components of tracking quality. In addition to HOTA, we report the standard multi-object tracking metrics MOTA [2] and IDF1 [29] to facilitate comparison with prior work, where MOTA aggregates false positives, false negatives, and identity switches into a single score, and IDF1 measures the consistency of identity preservation over time. For detection performance, we follow the COCO evaluation protocol [20] and use the mean Average Precision (mAP) computed over multiple IoU thresholds. As is common in the literature, we refer to this mAP simply as AP.

4.2 Implementation Details

We train with an effective batch size of 8 using gradient accumulation, where each training sample is a video clip of N_{clip} frames. We use the AdamW optimizer [17, 22] with weight decay 1×10^{-4}, an initial learning rate of 2×10^{-4} for all parameters except the backbone (set to 2×10^{-5}), and dropout ratio 0. Following MOTR [45], we adopt the same image augmentations and resize training images so that the shorter side is 800 px while constraining the longer side to at most 1536 px, preserving aspect ratio. At inference, the maximum longer side is set to 1333 px. All models are initialized from Deformable DETR weights [53] pretrained on COCO [20]. Throughout all experiments, we fix the clip length to $N_{\text{clip}} = 5$ and incorporate both the QIMv2 module [49, 50] and the query denoising strategy [19, 49]. We set $\tau_{\text{en}} = \tau_{\text{ex}} = 0.5$ for all experiments to keep the setup simple. For DanceTrack, we jointly train with CrowdHuman [32] by applying random spatial shifts to static images to synthesize short video clips with pseudo-tracks. The DanceTrack model is trained for 10 epochs, with the learning rate decayed by a factor of 10 at epoch 8, and we set $T_{\text{reid}} = 20$. For BFT, we train solely on the target dataset for 20 epochs with decaying the learning rate at epoch 16 and set $T_{\text{reid}} = 15$. Similarly, for AnimalTrack, we train for 20 epochs with the same learning rate schedule as BFT, but set $T_{\text{reid}} = 10$.

4.3 Main Results

Tab. 2 reports results on the DanceTrack test set. Among fully end-to-end trackers, our SelfMOTR attains 69.2 HOTA and 72.5 IDF_1, i.e. it is on par with CO-MOT (69.4 HOTA, 71.9 IDF_1) while achieving the highest association score

(59.3 AssA) in the end-to-end block. Since our method does not make use of an external detector and generates proposals from its own features, this higher AssA is consistent with a better feature-space alignment between detection and track queries. DetA (80.9) is slightly lower than CO-MOT (82.1), which is expected in the detector-free setting, but the IDF_1 gain shows that our internally generated proposals are sufficiently stable for identity preservation.
As detailed in Table 3, our proposed SelfMOTR achieves state-of-the-art performance on the highly dynamic BFT dataset, surpassing all evaluated non-end-to-end and end-to-end methods with a leading HOTA of 71.1. Furthermore, SelfMOTR establishes a new best among end-to-end trackers on the AnimalTrack dataset. This result is particularly remarkable given that AnimalTrack is a relatively small dataset, a scenario where data-hungry end-to-end models traditionally struggle to avoid overfitting and fail to generalize. Notably, compared to other tracking-by-propagation approaches, SelfMOTR exhibits a substantial gain on AnimalTrack (e.g., +14.5 HOTA over TrackFormer), indicating that the proposed formulation generalizes beyond the training regime and transfers more effectively to challenging animal scenarios.

4.4 Diagnosing Query Interference in Joint Decoding

Preliminary. Inspired by recent analyses of attention sinks in large language models [15,39], we use a similar lens to study query interactions in joint-decoding trackers. Rather than assuming a single universal sink behavior, we examine whether detect queries develop imbalanced self-attention, with some becoming overly track-focused while others largely ignore track context, both of which can undermine novel object detection. To diagnose this, we compute two token-level statistics from the decoder's self-attention weights $\alpha \in \mathbb{R}^{Q\times Q}$, focusing on how detect queries attend to the sequence. Because the number of active track keys ($|\mathcal{T}|$) and detection keys ($|\mathcal{D}|$) varies across frames, a raw sum of attention weights is heavily biased by cardinality. To isolate true track preference, we define a cardinality-corrected *Track-Attention Mass* ($\tilde{M}_i$) for each detection query i as the ratio of average attention per track key to average attention per detection key:

$$\tilde{M}_i = \frac{\frac{1}{|\mathcal{T}|}\sum_{j\in\mathcal{T}}\alpha_{ij}}{\frac{1}{|\mathcal{T}|}\sum_{j\in\mathcal{T}}\alpha_{ij} + \frac{1}{|\mathcal{D}|}\sum_{j\in\mathcal{D}}\alpha_{ij}} \in [0,1]$$

To assess whether a given $\tilde{M}_i$ corresponds to concentrated attention (sink-like behavior) or distributed interaction, we compute the *normalized Shannon entropy* of the query's attention row. Let $p_{ij} = \alpha_{ij}/\sum_k \alpha_{ik}$ be the normalized distribution over all Q keys. The normalized entropy is:

$$H_{\text{norm}}(p_i) = -\frac{1}{\log Q}\sum_{j=1}^{Q} p_{ij}\log(p_{ij}).$$

Together, $\tilde{M}_i$ captures *where* attention mass is allocated (track vs. non-track keys), while $H_{\text{norm}}(p_i)$ captures *how* it is distributed, enabling a robust diagno-

Table 2: State-of-the-art comparison on the DanceTrack [35] test set across graph-based, non end-to-end, and end-to-end trackers. The best end-to-end result is in bold, second best is underlined.

Model	**Metrics**				
	HOTA↑	DetA↑	AssA↑	IDF_1↑	MOTA↑
Graph-based					
SUSHI [7]	63.3	80.1	50.1	63.4	88.7
CoNo-Link [13]	63.8	80.2	50.7	64.1	89.7
SPAM [8]	64.0	–	–	63.4	89.2
NOOUGAT [26]	68.4	–	58.7	72.7	88.9
Non End-to-End					
ByteTrack [47]	47.4	71.0	32.1	53.9	89.6
GHOST [31]	56.7	81.1	39.8	57.7	91.3
OC-SORT [5]	55.1	80.3	38.3	54.6	92.0
Hybrid-SORT [42]	65.7	–	52.6	63.0	91.8
TrackTrack [33]	66.5	–	52.9	67.8	93.6
MOTRv2 [49]	69.9	83.0	59.0	71.7	91.9
End-to-End					
TransTrack [36]	45.5	75.9	27.5	45.2	88.4
MOTR [45]	54.2	73.5	40.2	51.5	79.7
DNMOT [11]	53.5	–	–	49.7	89.1
MeMOTR [12]	68.5	80.5	58.4	71.2	89.9
SambaMOTR [30]	67.2	78.8	57.5	70.0	88.1
MOTRv3 [44]	68.3	–	–	70.1	**91.7**
CO-MOT [41]	**69.4**	**82.1**	<u>58.9</u>	<u>71.9</u>	<u>91.2</u>
SelfMOTR (Ours)	<u>69.2</u>	<u>80.9</u>	**59.3**	**72.5**	89.9

sis of attention imbalance in joint decoders.

Layer-wise Attention Dynamics and Contextualization. The layer-wise attention profiles (Fig. 5a) reveal an intermediate *contextualization* phase in the decoder (Layers 2–4), where query–query interactions are highly active. In standard joint-decoding trackers like MOTR, discovery queries allocate a disproportionately high amount of their attention mass to the track pool during this phase. We hypothesize that this strong group-level dominance prematurely biases discovery queries toward existing tracks before new object hypotheses are fully formed. As the decoder reaches the final *synthesis* layer, our token-level analysis (Fig. 5b) shows that MOTR's attention mass remains broadly dispersed across the spectrum and is noticeably skewed toward high track-attention. Crucially, this wide dispersion is accompanied by a distinct drop in normalized entropy at both extremes of the spectrum. Such extreme regimes can severely hinder performance: queries that become heavily skewed toward tracks lose the independence required to discover novel objects, whereas the subset of queries that

Table 3: State-of-the-art comparison on BFT [51] and AnimalTrack [46] test sets across non end-to-end and end-to-end trackers. † denotes tracking-by-propagation methods. The best end-to-end result is in bold, best overall underlined.

Model	BFT			AnimalTrack		
	HOTA↑	IDF_1↑	MOTA↑	HOTA↑	IDF_1↑	MOTA↑
Non End-to-End						
JDE [38]	30.7	37.4	35.4	26.8	31.0	27.3
FairMOT [48]	40.2	41.8	56.0	30.6	38.8	29.0
CenterTrack [52]	65.0	61.0	60.2	9.9	7.0	1.6
SORT [3]	61.2	77.2	75.5	42.8	49.2	55.6
ByteTrack [47]	62.5	82.3	77.2	40.1	51.2	38.5
OC-SORT [5]	66.8	79.3	77.1	–	–	–
QDTrack [27]	–	–	–	<u>47.0</u>	<u>56.3</u>	<u>55.7</u>
End-to-End						
TransCenter [40]	60.0	72.4	74.1	–	–	–
TransTrack [36]	62.1	71.4	71.4	45.4	53.4	48.3
TrackFormer† [25]	63.3	72.4	74.1	31.0	36.5	20.4
SambaMOTR† [30]	69.6	81.9	72.0	–	–	–
SelfMOTR† (Ours)	**<u>71.1</u>**	**<u>82.7</u>**	**<u>77.6</u>**	**45.5**	**53.7**	**49.5**

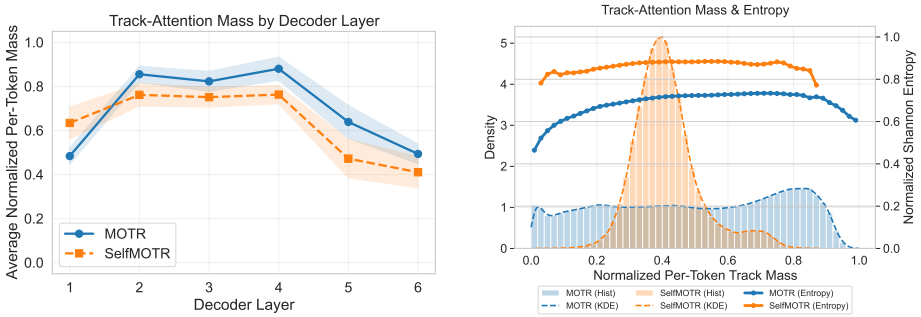
largely ignore tracks lose crucial spatial context and coordination, which can lead to redundant predictions.

In contrast, SelfMOTR's self-generated proposals provide a robust initial candidate structure. This structural prior reduces the reliance on track queries during the intermediate contextualization phase, yielding a more stable layer-wise trajectory (see Fig. 5a). Consequently, SelfMOTR successfully averts attention polarization in the final layer, concentrating query mass into a balanced, healthy regime (centered near 0.4) while preserving strictly high entropy (≈ 0.85 as shown in Fig. 5b). Thus, we hypothesize that a balanced utilization of track context versus discovery cues, directly mitigates the detection–association conflict observed in our experiments.

4.5 Ablations

Detection–Association Conflict. Finally, Tab. 4a compares detection AP with and without track queries. For MOTR, enabling track queries reduces AP from 66.3 to 60.0 (–6.3), confirming that tracking can overwrite detection. For our self-proposal architecture, the drop is only –0.3 (71.2 → 70.9), i.e. the detection–association conflict is practically removed while staying fully transformer-based and detector-free.

Data Scaling. Tab. 4b shows that both the anchor-based MOTR and our self-proposal variant benefit from HSV augmentation and joint CrowdHuman training, but the gains are consistently larger for self proposals (e.g. HOTA 58.2 →



(a) Layer-wise Average Track-Attention Mass **(b)** Track-Attention Mass & Entropy

Fig. 5: Diagnostic analysis of Track Attention Mass on the DanceTrack [35] validation set. **(a)** The evolution of the average Track-Attention Mass across decoder layers. **(b)** The distribution of mass and normalized Shannon entropy at the final decoder layer.

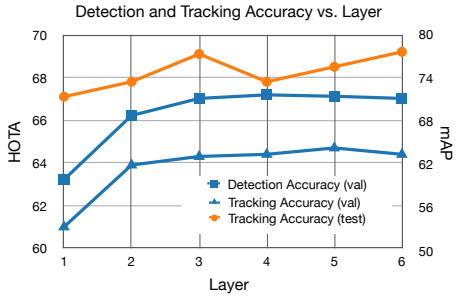


Fig. 6: Effect of decoder depth on accuracy. We vary the number of decoder layers used in the detection-only pass and report tracking accuracy (HOTA) on the validation and test sets, as well as detection accuracy (mAP) on the DanceTrack [35] validation set.

64.3 vs. 55.8 → 60.5). This indicates that, once the detection and tracking parts are decoupled, additional data can be exploited more effectively.

Proposal Threshold. Tab. 4c studies the confidence threshold c_{prop} used to select detections as proposals. A low threshold (0.05) gives the best overall HOTA (58.2) and AssA (47.4), while a higher threshold slightly improves DetA but reduces association. This matches the intuition that keeping more proposals is helpful for identity continuity.

Runtime and Efficiency. Although SelfMOTR runs the decoder twice, it remains computationally efficient. The heavy backbone encoder is executed only once per frame, and the additional cost stems solely from a lightweight second decoder pass. Our decoder-depth analysis for the detection (proposal-generation) pass (Fig. 6) shows that both tracking and detection performance saturate after just a few decoder layers, with deeper decoders offering diminishing returns. This enables a shallow proposal generator while reusing the same decoder for tracking, yielding a compact dual-pass design. In practice, SelfMOTR achieves

Table 4: Comprehensive ablation studies for SelfMOTR on the DanceTrack [35] validation set. We analyze the detection–association conflict, inference speed, data scaling, and proposal thresholds.

Method	**Track Queries**	AP ↑	AP_{50} ↑	AP_{75} ↑	FPS ↑
MOTR		66.3	85.5	70.7	–
MOTR	✓	$60.0^{\downarrow 6.3}$	$78.1^{\downarrow 7.4}$	$63.7^{\downarrow 7.0}$	24.8
SelfMOTR (Ours)		71.2	90.6	76.6	–
SelfMOTR (Ours)	✓	$70.9^{\downarrow 0.3}$	$89.4^{\downarrow 1.2}$	$76.3^{\downarrow 0.3}$	20.7

(a) Detection–association conflict (evaluated with and without track queries). Inference speed measured on Tesla L40S.

Method	HOTA ↑	DetA ↑	AssA ↑	IDF_1 ↑	MOTA ↑
Learnable 4D Anchor	55.8	69.7	44.9	57.2	77.2
+ HSV & CrowdHuman	60.5	74.4	49.4	62.0	83.5
Self Proposal	$58.2^{\uparrow 2.4}$	$71.9^{\uparrow 2.2}$	$47.4^{\uparrow 2.5}$	$59.0^{\uparrow 1.8}$	$80.0^{\uparrow 2.8}$
+ HSV & CrowdHuman	$\mathbf{64.3}^{\uparrow 3.8}$	$\mathbf{77.2}^{\uparrow 2.8}$	$\mathbf{53.7}^{\uparrow 4.3}$	$\mathbf{66.6}^{\uparrow 4.6}$	$\mathbf{86.5}^{\uparrow 3.0}$

(b) Impact of HSV augmentations and joint CrowdHuman pre-training.

Thresh.	HOTA ↑	DetA ↑	AssA ↑	IDF_1 ↑	MOTA ↑
0.05	**58.2**	71.9	**47.4**	**59.0**	**80.0**
0.50	57.7	**72.1**	46.4	58.6	79.9

(c) Proposal confidence threshold.

20.7 FPS on a Tesla L40S (Tab. 4a), with a modest runtime overhead relative to MOTR with learnable 4D anchors (24.8 FPS, Tab. 4a), while consistently improving HOTA and AssA. Overall, Fig. 6 and Tab. 4a suggest that a shallow detection pass suffices to reach the accuracy plateau, allowing practitioners to trade one or two decoder layers for additional speed while retaining most of the dual-pass gains.

Shared Decoder Design. To isolate the effect of decoder weight sharing, we compare SelfMOTR with a dual-decoder variant that uses independent decoder parameters for proposal generation and track refinement. Although the dual-decoder model achieves nearly identical detection accuracy, it performs substantially worse in association, obtaining 54.5 vs. 59.3 AssA and 69.1 vs. 72.5 IDF_1, which results in a lower HOTA score of 66.2 vs. 69.2 (Tab. 5). These results indicate that decoder weight sharing is not only a compact implementation choice, but improves association by aligning proposal generation and track refinement.

Detection Comparison. MOTRv2 benefits from YOLOX, a strong external proposal detector with higher AP than our internal proposal pass, 79.7 vs. 71.2 (Tab. 6), which carries over to full tracking AP, 75.3 vs. 70.9, and

Table 5: Effect of decoder weight sharing on DanceTrack [35] test. Dual Decoder uses separate decoder parameter sets for proposal generation and track refinement, whereas SelfMOTR shares decoder weights across both functions. Higher is better for all metrics; bold indicates the better result.

Model	**Metrics**				
	HOTA↑	DetA↑	AssA↑	IDF_1↑	MOTA↑
Dual Decoder	<u>66.2</u>	<u>80.6</u>	<u>54.5</u>	<u>69.1</u>	**90.0**
SelfMOTR	**69.2**	**80.9**	**59.3**	**72.5**	<u>89.9</u>

Table 6: Proposal-generation and tracking-output AP on DanceTrack [35] val. YOLOX is the separately trained external proposal detector used by MOTRv2.

Method	**Track Queries**	**AP** ↑	**AP_{50}** ↑	**AP_{75}** ↑	**Params**
YOLOX		79.7	95.6	84.6	99 M
SelfMOTR		71.2	90.6	76.6	42 M
MOTRv2	✓	75.3	90.9	81.2	141 M
SelfMOTR	✓	70.9	89.4	76.3	42 M

explains its stronger DetA/MOTA and slightly higher HOTA. However, this requires a separately trained detector and a much larger pipeline: 141M parameters vs. SelfMOTR's 42M parameters. Despite weaker proposal AP, SelfMOTR achieves higher AssA/IDF1 on DanceTrack test, suggesting that self-generated, decoder-compatible proposals better support association while preserving the self-contained character of end-to-end tracking.

5 Conclusion

In this paper, we present SelfMOTR, a simple and effective approach that decouples proposal discovery from association using self-generated detection priors. By aligning priors internally, SelfMOTR yields strong proposals while keeping an end-to-end tracking pipeline. This design substantially reduces detection–association query interference, suggesting internal prior generation as a useful default in unified tracking architectures. Our analyses also indicate that joint-decoding transformers have sufficient capacity for robust detection and tracking, and future work can explore more adaptive ways to leverage this capacity.

Acknowledgements

This work was funded by the German Research Foundation DFG with the grants STE2802/4-1 (EM). Computations were performed with computing resources granted by RWTH Aachen under projects `rwth1905`.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: BoT-SORT: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)
2. Bernardin, K., Stiefelhagen, R.: Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. EURASIP Journal on Image and Video Processing **2008**(1), 246309 (2008)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: IEEE International Conference on Image Processing. pp. 3464–3468 (2016)
4. Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6 (2017)
5. Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric SORT: Rethinking SORT for robust multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9686–9696 (2023)
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision. pp. 213–229. Springer (2020)
7. Cetintas, O., Brasó, G., Leal-Taixé, L.: Unifying short and long-term tracking with graph hierarchies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22877–22887 (2023)
8. Cetintas, O., Meinhardt, T., Brasó, G., Leal-Taixé, L.: SPAMming labels: Efficient annotations for the trackers of tomorrow. In: Proceedings of the European Conference on Computer Vision. pp. 377–395. Springer (2024)
9. Downer, J., Tennant, D.: Earthflight. `https://www.bbc.co.uk/programmes/b018xsc1` (2011), BBC. Accessed: 2026-02-15
10. 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)
11. Fu, T., Wang, X., Yu, H., Niu, K., Li, B., Xue, X.: DeNoising-MOT: Towards Multiple Object Tracking with Severe Occlusions. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 2734–2743 (2023)
12. Gao, R., Wang, L.: MeMOTR: Long-term memory-augmented transformer for multi-object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9901–9910 (2023)
13. Gao, Y., Xu, H., Li, J., Wang, N., Gao, X.: Multi-scene generalized trajectory global graph solver with composite nodes for multiple object tracking. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 1842–1850 (2024)
14. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: Exceeding YOLO Series in 2021. arXiv preprint arXiv:2107.08430 (2021)
15. Gu, X., Pang, T., Du, C., Liu, Q., Zhang, F., Du, C., Wang, Y., Lin, M.: When attention sink emerges in language models: An empirical view. In: International Conference on Learning Representations (2025)
16. Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering **82**(1), 35–45 (1960)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)

18. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
19. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: DN-DETR: Accelerate DETR training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13619–13627 (2022)
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. Lecture Notes in Computer Science, vol. 8693, pp. 740–755 (2014)
21. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: DAB-DETR: Dynamic anchor boxes are better queries for DETR. In: International Conference on Learning Representations (2022)
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
23. 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)
24. Maggiolino, G., Ahmad, A., Cao, J., Kitani, K.: Deep OC-Sort: Multi-pedestrian tracking by adaptive re-identification. In: IEEE International Conference on Image Processing. pp. 3025–3029 (2023)
25. Meinhardt, T., Kirillov, A., Leal-Taixé, L., Feichtenhofer, C.: TrackFormer: Multi-object tracking with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8844–8854 (2022)
26. Missaoui, B., Cetintas, O., Brasó, G., Meinhardt, T., Leal-Taixé, L.: Noougat: Towards unified online and offline multi-object tracking. arXiv preprint arXiv:2509.02111 (2025)
27. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 164–173 (2021)
28. Qin, Z., Wang, L., Zhou, S., Fu, P., Hua, G., Tang, W.: Towards generalizable multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18995–19004 (2024)
29. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: Proceedings of European Conference on Computer Vision. pp. 17–35 (2016)
30. Segu, M., Piccinelli, L., Li, S., Yang, Y.H., Gool, L.V., Schiele, B.: Samba: Synchronized set-of-sequences modeling for multiple object tracking. In: The Thirteenth International Conference on Learning Representations (2025)
31. Seidenschwarz, J., Brasó, G., Serrano, V.C., Elezi, I., Leal-Taixé, L.: Simple cues lead to a strong multi-object tracker. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13813–13823 (2023)
32. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: CrowdHuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018)
33. Shim, K., Ko, K., Yang, Y., Kim, C.: Focusing on tracks for online multi-object tracking. In: Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. pp. 11687–11696 (2025)
34. Stanojevic, V.D., Todorovic, B.T.: BoostTrack: Boosting the similarity measure and detection confidence for improved multiple object tracking. Machine Vision and Applications **35**(3), 53 (2024)

35. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20993–21002 (2022)
36. Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P.: TransTrack: Multiple object tracking with transformer. arXiv preprint arXiv:2012.15460 (2020)
37. Wang, X., Ma, K., Liu, Q., Zou, Y., Fu, Y.: Multi-object tracking in the dark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 382–392 (2024)
38. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: Proceedings of the European Conference on Computer Vision. pp. 107–122. Springer (2020)
39. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. In: International Conference on Learning Representations (2024)
40. Xu, Y., Ban, Y., Delorme, G., Gan, C., Rus, D., Alameda-Pineda, X.: TransCenter: Transformers With Dense Representations for Multiple-Object Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(6), 7820–7835 (2023)
41. Yan, F., Luo, W., Zhong, Y., Gan, Y., Ma, L.: CO-MOT: Boosting end-to-end transformer-based multi-object tracking via coopetition label assignment and shadow sets. In: International Conference on Learning Representations (2025)
42. 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: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 6504–6512 (2024)
43. Yi, K., Luo, K., Luo, X., Huang, J., Wu, H., Hu, R., Hao, W.: UCMCTrack: Multi-Object Tracking with Uniform Camera Motion Compensation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 6702–6710 (2024)
44. Yu, E., Wang, T., Li, Z., Zhang, Y., Zhang, X., Tao, W.: MOTRv3: Release-Fetch Supervision for End-to-End Multi-Object Tracking. arXiv preprint arXiv:2305.14298 (2023)
45. Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: MOTR: End-to-End Multiple-Object Tracking with Transformer. In: Proceedings of the European Conference on Computer Vision. pp. 659–675. Springer (2022)
46. Zhang, L., Gao, J., Xiao, Z., Fan, H.: AnimalTrack: A Benchmark for Multi-Animal Tracking in the Wild. International Journal of Computer Vision **131**(2), 496–513 (2023)
47. 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)
48. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: FairMOT: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision **129**(11), 3069–3087 (2021)
49. Zhang, Y., Wang, T., Zhang, X.: MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22056–22065 (2023)
50. Zhang, Y., Wang, T., Zhang, X.: MOTRv2 (official implementation). `https://github.com/megvii-research/MOTRv2` (2023), Megvii Research. Accessed: 2025-11-13

51. Zheng, G., Lin, S., Zuo, H., Fu, C., Pan, J.: NetTrack: Tracking highly dynamic objects with a net. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19145–19155 (2024)
52. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: European Conference on Computer Vision. pp. 474–490. Springer (2020)
53. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR (official implementation). `https://github.com/fundamentalvision/Deformable-DETR` (2020), fundamentalvision (SenseTime Research). Accessed: 2025-11-05
54. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2021)