Appendix

We here provide additional details on the method, mathematical proofs, implementation details and experimental results. In Sec. A we provide a mathematical derivation of the latent transition probability on a random walk. Refer to Sec. B for details on Walker's training. Refer to Sec. C and Sec. D for an in-depth explanation of the inference tracking algorithms of Walker and QD-Walker respectively. Sec. E reports implementation details. In Sec. F, we motivate the use of a sparse setting and stress the usefulness of self-supervised trackers leveraging the temporal information to make full use of such label-efficient settings. Finally, we report additional quantitative and qualitative (Sec. G) results.

A Latent Transition Probability Derivation

We prove Eq. (5) (Sec. 3.4), which represents the probability of transitioning on a given latent node $\mathbf{q}^j_{t+k}$ given that a walk on the appearance graph $\mathcal{G}$ (Sec. 3.2) starts from $\mathbf{q}^i_{t^+}$ and ends in $\mathbf{q}^l_t$.

Let $\mathcal{G} : \mathbf{Q}^+_t \rightarrow \mathbf{Q}_{t+k} \rightarrow \mathbf{Q}_t$ be a cyclic temporal graph connecting the nodes $\mathbf{Q}^+_t$ in the frame I_t to $\mathbf{Q}_{t+k}$ in the frame I_{t+k} and back to $\mathbf{Q}_t$ in I_t. $\mathcal{G}$ is a Markov chain described by the forward and backward transitions $A^{t+k}_{t^+}$ and A^t_{t+k}, whose chained transition $\bar{A}^t_{t^+}$ describes the cycle correspondence as a multi-step walk along the appearance graph $\mathcal{G}$. Let X_t be the state of a walker at time t, and $p_{X_t}(i)$ the probability of being at node i at time t.

Theorem 1. *The probability of transitioning on a latent node $\boldsymbol{q}^j_{t+k}$ on the reference image I_{t+k} when starting from $\boldsymbol{q}^i_{t^+}$ in I_t and ending on $\boldsymbol{q}^l_t$ in I_t along the cycle walk $\mathcal{G}$ is:*

$$p^{\mathcal{G}}_{X_{t+k}|X_t,X^+_t}(j|l,i) = p^{\mathcal{G}}_{X_t|X_{t+k}}(l|j)p^{\mathcal{G}}X_{t+k}|X^+_t(j|i)/C \tag{14}$$

$$= A^{t+k}_{t^+}(i,j)A^t_{t+k}(j,l)/C \tag{15}$$

where $C = \sum_{\boldsymbol{q}^m_{t+k} \in \boldsymbol{Q}_{t+k}} p^{\mathcal{G}}_{X_t|X_{t+k}}(l|m)p^{\mathcal{G}}_{X_{t+k}|X^+_t}(m|i)$ is a normalizing constant.

Proof (Proof of Theorem 1).

$$\begin{aligned}
p^{\mathcal{G}}_{X_{t+k}|X_t,X^+_t}(j|l,i) &\overset{(1)}{=} \frac{p^{\mathcal{G}}_{X_t,X_{t+k},X^+_t}(l,j,i)}{p^{\mathcal{G}}_{X_t,X^+_t}(l,i)} \\
&\overset{(2)}{=} \frac{p^{\mathcal{G}}_{X_t|X_{t+k}}(l|j)p^{\mathcal{G}}_{X_{t+k}|X^+_t}(j|i)}{p^{\mathcal{G}}X_t|X^+_t(l|i)p^{\mathcal{G}}X^+_t(i)} \\
&\overset{(3)}{=} \frac{p^{\mathcal{G}}_{X_t|X_{t+k}}(l|j)p^{\mathcal{G}}_{X_{t+k}|X^+_t}(j|i)}{\sum_{\mathbf{q}^m_{t+k} \in \mathbf{Q}_{t+k}} p^{\mathcal{G}}_{X_t|X_{t+k}}(l|m)p^{\mathcal{G}}_{X_{t+k}|X_t}(m|l)p^{\mathcal{G}}_{X^+_t}(i)} \\
&\overset{(4)}{=} p^{\mathcal{G}}_{X_t|X_{t+k}}(l|j)p^{\mathcal{G}}_{X_{t+k}|X^+_t}(j|i) \;/\; C \\
&\overset{(5)}{=} A^{t+k}_{t^+}(i,j)A^t_{t+k}(j,l) \;/\; C
\end{aligned}$$

We here motivate the steps in the proof:

(1) by the definition of conditional probability.

(2) since a walk on the appearance graph $\mathcal{G}$ defined in Sec. 3.2 is a first-order Markov chain, each transition only depends on the previous state, *i.e.* $p^{\mathcal{G}}_{X_t, X_{t+k}, X_t^+}(l, j, i) = p^{\mathcal{G}}_{X_t|X_{t+k}}(l|j) p^{\mathcal{G}}_{X_{t+k}|X_t^+}(j|i)$.

(3) since a walk on the appearance graph $\mathcal{G}$ defined in Sec. 3.2 is a first-order Markov chain, each transition only depends on the previous state, *i.e.* $p^{\mathcal{G}}_{X_t|X_t^+}(l|i) = \sum_{\mathbf{q}^m_{t+k} \in \mathbf{Q}_{t+k}} p^{\mathcal{G}}_{X_t|X_{t+k}}(l|m) p^{\mathcal{G}}_{X_{t+k}|X_t}(m|l)$. Moreover, we marginalize over all possible transition states $\mathbf{q}^m_{t+k} \in \mathbf{Q}_{t+k}$.

(4) for a chosen starting node $\mathbf{q}^i_t$ and ending node $\mathbf{q}^l_t$,
$C = \sum_{\mathbf{q}^m_{t+k} \in \mathbf{Q}_{t+k}} p^{\mathcal{G}}_{X_t|X_{t+k}}(l|m) p^{\mathcal{G}}_{X_{t+k}|X_t}(m|l) p^{\mathcal{G}}_{X_t^+}(i)$ is a normalization constant.

(5) according to our definition of the transition probability matrices for a random walk on an appearance graph $\mathcal{G}$ (Sec. 3.3).

B Training a Walker

We here provide additional details on Walker's training, which we introduced in Sec. 3.3, Sec. 3.4, Sec. 3.5. In particular, we discussed our multi-positive contrastive random walk in Sec. 3.3, our cluster-wise forward assignment and optimization in Sec. 3.4, and the total loss in Sec. 3.5. To make the understanding of our training pipeline easier, we provide pseudo-code in Alg. 1.

Node Embedding. During one training iteration, we are given the detections $\mathcal{D}_t$ on I_t and $\mathcal{D}_{t+k}$ on I_{t+k}, and the ground-truth detections $\hat{\mathcal{D}}_t$ on I_t and $\hat{\mathcal{D}}_{t+k}$ on I_{t+k}. We first embed the detections to obtain their embeddings, *i.e.* $\mathbf{q}_t$ and $\mathbf{q}_{t+k}$ respectively.

Node Selection. Depending on the setting - *i.e. dense* or *sparse* (see Sec. 4.1) - we use different policies for selecting the positive and negative nodes in each frame. Note that we defined in Sec. 3.2 the positive nodes as the ones with high IoU with a set of reference bounding boxes $\bar{\mathcal{D}}_t$. In the *sparse* setting, we cannot assume the detection annotations to be available for both key and reference frame. Thus, we use the high-confidence detections $\mathcal{D}^{\text{high}}_t$ as set of reference bounding boxes $\bar{\mathcal{D}}_t = \mathcal{D}^{\text{high}}_t$. In the *dense* setting, detection annotations are available for all frames. We can thus reliably identify good nodes over which performing our contrastive random walk as the nodes overlapping with the ground truth bounding boxes $\hat{\mathcal{D}}_t$, *i.e.* $\bar{\mathcal{D}}_t = \hat{\mathcal{D}}_t$. Given the reference bounding boxes $\bar{\mathcal{D}}_t$, we sample positive and negative nodes with a rate of 1/3.

Cluster Assignment. We then compute the forward $A^{t+k}_{t^+}$, backward A^t_{t+k}, and cycle $\bar{A}^t_{t^+}$ transition probabilities (Sec. 3.3). We obtain the set of unique clusters $\mathcal{C}_t$ in the key frame I_t, and sort and filter them by their cluster cycle probability, ensuring that it must be higher than a threshold β_{cycle}. Finally, we incrementally match key clusters to reference clusters $\mathcal{Z}^i_{t+k}$ based on their max-likelihood transition state, as introduced in Eq. (7).

Total Loss. The pseudo-assignments identified with the algorithm described above are then optimized with the forward loss $\mathcal{L}_{\text{forward}}$, applied jointly with the cycle loss $\mathcal{L}_{\text{cycle}}$.

C Tracking with Walker

We introduced Walker's tracking scheme in Sec. 3.6. To make the understanding of our matching pipeline with fused motion and appearance easier, we provide in Alg. 2 the matching pseudo-code for a whole video `V`.

Inspired by BYTE [56], Walker adopts a two-stage matching scheme. Let $\mathcal{T}$ be the tracklets of the video up to time $t-1$. Let `Det` be the object detector. Let I_t be the incoming frame at time t. $\mathcal{D}_t = \texttt{Det}(I_t)$ is the set of detections predicted by the object detector on I_t. We define the set of high-confidence detections $\mathcal{D}_t^{\text{high}} = \{d_t^i \in \mathcal{D}_t \mid \text{conf}(d_t^i) \geq \beta_{\text{high}}\}$ as those with confidence greater than a threshold β_{high}, and the set of low-confidence detections $\mathcal{D}_t^{\text{low}} = \{d_t^i \in \mathcal{D}_t \mid \beta_{\text{low}} \leq \text{conf}(d_t^i) < \beta_{\text{high}}\}$ as those with confidence between thresholds β_{low} and β_{high}.

In the *first association stage*, Walker matches high-confidence detections $\mathcal{D}_t^{\text{high}}$ to tracklets $\mathcal{T}$ based on our cost matrix W (defined in Eq. (13)) that fuses motion and appearance costs. In the *second association stage*, low confidence detections $\mathcal{D}_t^{\text{low}}$ are assigned to the remaining tracklets $\mathcal{T}_{\text{remain}}$ based on their IoU. Unmatched tracklets $\mathcal{T}_{\text{unmatched}}$ are deleted, and new tracklets are initialized from the remaining high-confidence detections $\mathcal{D}_t^{\text{remain}}$.

Track rebirth [49, 59] is not shown in the algorithm for simplicity. For additional details on the track management scheme, refer to BYTE [56].

D Tracking with QD-Walker

We briefly introduced QD-Walker's appearance-only tracking scheme in Sec. 4.1. The goal was to provide an appearance-only tracking baseline that could be used to directly compare to QDTrack-S and other self-supervised Re-ID baselines to establish which self-supervised appearance learning schemes translates to the better tracker.

To make the understanding of our appearance-only matching pipeline easier, we provide pseudo-code for one matching step (Alg. 3). Inspired by QDTrack [14, 35], Walker matches detections to tracklets based on their appearance. However, as opposed to QDTrack's bisoftmax, Walker uses the biwalk similarity metric $s_{i,j}^{biwalk}$ introduced in Eq. (11) between the embeddings of the i-th detection and the j-th tracklet.

We borrow from QDTrack the track management scheme to keep track of inactive and currently active tracks and to handle the matching of objects. Active tracks are tracks that have a matching detection in the previous frame, otherwise they become inactive. Tracks that are inactive for K frames will be removed and not be considered for matching. In particular, Walker first removes duplicate detections with inter-class NMS with confidence threshold `Det. Conf. Thr.` and IoU threshold `Det. NMS IoU. Thr.`. Detections are only considered for matching to existing tracks if the detection confidence is above a threshold β_{obj}. A match is determined if the biwalk similarity $s_{i,j}^{biwalk}$ is higher than a threshold β_{match}. For unmatched objects that have a detection confidence higher than a threshold

β_{new}, we initialize a new track instead. We keep the unmatched objects as backdrops for L frames and use them as matching candidates. Detections that are matched to backdrops will thus not be matched to existing tracks. The tracklet embeddings are updated with an exponential moving average with momentum m. For additional details on the track management scheme, refer to the original QDTrack paper [35].

E Implementation Details

We report the training and inference hyperparameters for Walker in Tab. 5, identified by parameter-search on the validation set of each dataset. Since our inference algorithm builds on top of BYTE and QDTrack, we take their hyperparameters directly unless differently specified. Notice that Walker shares the same trained model and parameters with QD-Walker, only the inference scheme differs. Results reported for other trackers are directly taken from their papers or re-run following the hyperparameters introduced in the respective paper.

Training Hyperparameters The key frame is sampled from the set of frames with bounding box annotations, *i.e.* in the sparse setting we assume that one frame every k is labeled starting from the first frame in the video sequence. We sample the reference frame from a neighborhood of the key frame, where the neighborhood width is $\hat{k}$. For data augmentation, we utilize mosaic augmentation on key and reference frame, followed by consistent photometric augmentations as in [56]. We then apply non-consistent multi-scale resizing augmentations on key and reference frame, with a scale range (0.5, 1.5) around the basic image size 1440 x 800.

Inference Hyperparameters We report the inference hyperparameters for Walker following the naming convention established throughout the paper, and re-iterated in Sec. B and Sec. C.

F Datasets and Annotations

The minimal annotation frame rate found across tracking datasets is 1 FPS [9]. Under this cut-off value, annotating tracking is often not possible due to the limited living span of objects in a video. For this reason, the TAO dataset [9] was originally annotated at 1 Hz. NuScenes [5] is annotated only at 2 Hz due to the difficulty in calibration and syncronization of multiple sensors. However, the large differences in appearance across the sparsely annotated frames in such datasets makes it difficult to learn supervised trackers. For this reason, the TAO dataset [9] was later refined to 6 FPS [2]. By not requiring instance labels, a good self-supervised tracker would achieve good tracking performance even when trained under a sparse annotation regimen, as it could make use of the unlabeled frames.

For this reason, we choose to evaluate self-supervised trackers trained with detection annotations at 0.1 FPS (Sec. 4.1), a value sensitively below the common annotation rate and often sparser than the average object living time in

Table 5: Hyper-parameters used in each benchmark. We include both training and inference parameters of Walker across all datasets.

	Parameter	MOT17	DanceTrack	BDD100K
Training	λ_1	1.0	1.0	0.5
	λ_2	2.0	2.0	1.0
	$\hat{k}$	10	10	3
	α_1	0.7	0.7	0.7
	α_2	0.3	0.3	0.3
	β_{obj}	0.3	0.3	0.3
	β_{cycle}	0.8	0.8	0.8
	τ	0.05	0.05	0.05
Inference	`Det. Conf. Thr.`	0.1	0.1	0.1
	`Det. NMS IoU Thr.`	0.7	0.6	0.65
	β_{new}	0.75	0.8	0.5
	β_{high}	0.3	0.6	0.35
	β_{low}	0.1	0.1	0.1
	$\beta^{\text{high}}_{\text{match}}$	0.1	0.1	0.1
	β_{biwalk}	0.2	0.2	0.2
	β_{IoU}	0.5	0.5	0.5
	λ_{biwalk}	2.0	2.0	2.0
	$\beta^{\text{low}}_{\text{match}}$	0.5	0.5	0.5
	β_{cycle}	0.1	0.1	0.1
	τ	0.07	0.07	0.07
	K	30	20	10
	m	0.5	0.8	0.8

a video. Note that on MOT17 we only validate the dense protocol due to the very small size of its half-train set (only 7 videos totalling 2658 frames). Self-supervised tracking methods leveraging temporal self-supervision can make full use of the video stream, even in correspondence of the unlabeled frames, overcoming the limitations of training supervised trackers on sparsely annotated data. Moreover, by learning from such a low annotation frame rate, self-supervised multiple object tracking algorithms such as Walker allow to significantly reduce the annotation cost for video datasets. Finally, Walker can be in principle extended to fully unlabeled videos. Given a pre-trained object detector, Walker can be used to train the embedding head on the unlabeled videos while keeping the detector frozen or finetuning it with knowledge distillation techniques. We leave this interesting application to future work.

G Additional Results

G.1 Additional Self-supervised Re-ID baselines

We compare to additional self-supervised Re-ID baselines [3, 20, 47]. Since such methods do not provide an official implementation, or they cannot be easily extended to an appearance-only setting, we compare Walker against their published results.

In Tab. 6, we compare on MOT17's public Faster R-CNN detections against Bastani *et al.* [3] and Ho *et al.* [20]. Walker greatly outperforms both approaches, showing the superiority of our self-supervised appearance representations.

Table 6: Comparison to baselines on public detections. We compare to existing baselines which report results on the public detection set of MOT17. For a fair comparison, we use Faster R-CNN and train only on MOT17, without using Crowdhuman.

Method	MOTA	IDF1	MOTP
Bastani et al. [3]	56.8	58.3	-
Ho et al. [20]	48.1	-	76.7
Walker	**68.0**	**64.5**	**78.4**

Table 7: Comparison to CRW as Re-ID. We compare Walker on the MOT17 validation set against the CRW used as a Re-ID module in a JDE [48] tracker as in [47]. Both methods combine appearance with motion.

Method	HOTA	IDF1
JDE-CRW [47]	61.7	73.0
Walker (Ours)	**63.6**	**77.4**

In Tab. 7, we compare against a straightforward extension of the CRW to Re-ID by directly using the CRW module trained for point correspondence as an object-level Re-ID module. Although their performance is satisfying (albeit greatly supported by the two-stage pipeline and motion-based heuristics of the JDE's algorithm), the appearance representations learned by the original CRW algorithm are not object-specific and do not enforce mutual exclusivity. By addressing both limitations, Walker achieves higher performance.

In Tab. 14, we report the performance compared on *all* and *top-5 hardest* sequences from DanceTrack val. We compare Walker to ByteTrack [56] and ByteTrack + [22]. We choose ByteTrack as a representative motion-only tracker, and naively extend it with a pre-trained [22] as Re-ID head. Since [22] was trained on the DAVIS dataset, ByteTrack + [22] drastically fails to cope with DanceTrack's similar object appearances, worsening ByteTrack's association (Tab. 14).

G.2 Ablation on Method Details

We here ablate on the method components that leverage the quasi-dense nature of our temporal object appearance graph. In particular, we ablate on (i) the effectiveness of the proposed method components, (ii) the effect of different appearance-based match metrics, (iii) the use of a single-positive vs. a multi-positive contrastive cycle consistency objective, and (iv) the importance of enforcing mutually-exclusive assignments.

Method Components. We ablate on the effectiveness of each proposed component (Tab. 8) on top of the naive quasi-dense contrastive random walk baseline (QD-CRW). We incrementally add our multi-positive contrastive objective (+ multi-positive), enforce mutually-exclusive connectivity (+ mutually-exclusive), replace the bisoftmax similarity with our biwalk match metric in QDTrack's appearance-only inference (+ biwalk), and add motion constraints to reject unlikely appearance-based associations (+ motion). While all rows in Tab. 8 learn

from our proposed TOAG, our contributions clearly promote an optimal graph topology for MOT (5.1 vs. 38.6 AssA).

Table 8: Ablation on our individual method components on top of the naive quasi-dense CRW (QD-CRW) on DanceTrack val.

Method	HOTA	AssA	DetA
QD-CRW	19.2	5.1	**74.1**
+ multi-positive	46.3	30.2	71.8
+ mutually-exclusive	47.3	31.3	71.7
+ biwalk (= QD-Walker)	49.0	32.8	73.6
+ motion (= Walker)	**53.0**	**38.6**	73.1

Table 9: Ablation on match metrics for appearance-only tracking (QD-Walker) on DanceTrack val.

Metric	HOTA	AssA	DetA
Cosine	47.3	31.3	71.7
Bisoftmax [14]	46.8	30.6	71.7
Biwalk	**49.0**	**32.8**	**73.6**

Appearance-based Match Metrics. We ablate on the effect of different appearance-based similarity metrics in appearance-only MOT with QD-Walker (Tab. 9). Our proposed biwalk improves the overall tracking performance.

Multi-positive Cyclic Contrastive Objective. In Tab. 10, we ablate on different formulations of our cycle consistency formulation introduced in Sec. 3.3. We report the results for cycle walks optimized wrt. a single target (a), and multiple targets (b). We find that our proposed multi-positive formulation is remarkably more effective than the naive single positive baseline. We argue that the single-positive baseline treats as negatives for the contrastive loss also all the other nodes expect for the self node that represent detections which are highly overlapping with the target node, and likely to represent the same instance. Consequently, a significant amount of noise is injected in the training, making it more difficult for the embedding head to discriminate instances. We solve this problem with our multi-positive formulation, which enables multiple positive target for each contrastive random walk.

Table 10: Ablation on the selection policy for the cycle walk targets. We ablate on the DanceTrack validation set on different options of the target nodes to optimize for a cycle walk $\mathcal{G}_i$ starting from a node $\mathbf{q}^i_{t+}$ in I_t and ending on $\mathbf{q}^i_{t+}$ itself. The forward loss is not applied here. Optimizing cycle transitions only with respect to the destination node $\mathbf{q}^i_{t+}$ itself (a) considers as negatives also the highly overlapping nodes which are likely to represent the same instance, creating a conflicting self-supervisory signal. This problem is solved by considering as positives all the nodes Y^+_i highly overlapping with $\mathbf{q}^i_{t+}$.

Selection Policy	Cycle Prob.	HOTA	AssA	DetA	MOTA	IDF1
a) Single-positive	$p^{\mathcal{G}}_{X_t \mid X^+_t}(i \mid i)$	39.6	22.8	69.4	79.1	37.4
b) Multi-positive	$p^{\mathcal{G}}_{X_t \mid X^+_t}(Y^+_i \mid i)$	**48.7**	**31.1**	**77.1**	**88.9**	**48.0**

Mutually-exclusive Forward Assignments. In Tab. 11, we ablate on different policies to identify and optimize the forward assignments according to the formulation introduced in Sec. 3.4. We report the results with cluster-wise mutually-exclusive assignments (c) and assignments that are not mutually-exclusive (a,

Table 11: Ablation on the selection policy for the match pseudo-labels. We ablate on the DanceTrack validation set on different formulations of the max-likelihood transition state for a cycle walk $\mathcal{G}_i$ starting from a node $\mathbf{q}^i_{t^+}$ in I_t and ending on $\mathbf{q}^i_{t^+}$ itself in I_t after transitioning on I_{t+k}. *Single-positive* consists in identifying the max-likelihood transition state on the cycle walk starting from a node $\mathbf{q}^i_{t^+}$ and ending on the node $\mathbf{q}^i_{t^+}$ itself; *Multi-positive* averages over the multi-positive target nodes Y_i^+ for a cycle transition starting in $\mathbf{q}^i_{t^+}$; *Cluster-wise Multi-positive* further averages over the nodes in the starting cluster $\mathcal{C}^i_t = Y_i^+$, and enforces cluster-wise mutually-exclusive assignments with the algorithm described in Sec. 3.4.

Selection Policy	Latent Transition Prob.	HOTA	AssA	DetA	MOTA	IDF1
Single-positive	$p^{\mathcal{G}}_{X_{t+k}\|X_t,X_t^+}(j\|i,i)$	46.2	29.1	76.8	87.0	45.9
Multi-positive	$p^{\mathcal{G}}_{X_{t+k}\|X_t,X_t^+}(j\|Y_i^+,i)$	46.5	30.0	76.8	86.9	46.2
Cluster-wise Multi-positive	$p^{\mathcal{G}}_{X_{t+k}\|X_t,X_t^+}(j\|Y_i^+,Y_i^+)$	**49.8**	**32.2**	**77.3**	**89.4**	**49.3**

Table 12: Ablation on the loss components. We ablate on the DanceTrack validation set on the importance of each proposed loss components.

$\mathcal{L}_{\text{cycle}}$	$\mathcal{L}_{\text{forward}}$	HOTA	AssA	DetA	MOTA	IDF1
✓	-	48.7	31.1	77.1	88.9	48.0
✓	✓	**49.8**	**32.2**	**77.3**	**89.4**	**49.3**

b). In particular, (a) uses a single-node to single-node cycle walk formulation to independently identify the max-likelihood latent transition state in the reference frame that matches each node in the key frame. (b) further refines it by averaging the latent transition probabilities over the set of possible targets for the cycle walks departing from a node in the key frame. However, both (a) and (b) consider that each starting node can get independent assignments that are not mutually-exclusive, meaning that nodes in I_t from different instances may be assigned to nodes in I_{t+k} from a same instance, causing conflicts in the optimization. This problem is elegantly addressed by our mutually-exclusive cluster-wise assignment and optimization strategy introduced in Sec. 3.4, which (i) prevents nodes from a same cluster in the key frame to be assigned to nodes in different clusters in the reference frame, and (ii) prevents nodes from different clusters in the key frame to be assigned to a same cluster in the reference frame.

G.3 Ablation on the Impact of the Hyperparameters

Current tracking-by-detection (TbD) methods, including Walker, are hyperparameter-heavy. However, Walker's 14 inference hyperparameters are comparable to state-of-the-art tracking-by-detection methods combining motion and appearance, *e.g.* BoT-SORT and StrongSORT have 13 according to their official code. As mentioned in Sec. E, our inference algorithm builds on QDTrack and BYTE. When not explicitly mentioned, we keep all hyperparameters as in their original works.

Table 13: Ablation on Walker's sensitive inference parameters on DanceTrack val.

β_{high}	β_{match}^{high}	λ_{biwalk}	HOTA	DetA	AssA	MOTA	IDF1
0.5	0.1	1.0	52.6	73.1	38.1	86.9	56.4
0.5	0.1	2.0	53.2	73.1	38.9	87.0	**57.2**
0.5	0.2	2.0	52.6	73.5	37.9	86.6	55.0
0.6	0.1	2.0	**53.4**	**73.6**	**39.0**	**87.2**	56.3

Table 14: Comparison to ByteTrack + [22] on DanceTrack dense val. Performance compared on *all* and *top-5 hardest* sequences. FLOPs are computed using an input size of 3x640x640 to the YOLOX-X detector, of 3x256x128 to [20]'s ResNet-18 Re-ID branch and of 320x7x7 (RoI size in YOLOX-X) to our 4conv-1fc emb. head.

	All			*Hard (top-5)*				
Method	HOTA	AssA	DetA	HOTA	AssA	DetA	Det. FLOPS (G)	Re-ID FLOPS (G)
ByteTrack [56]	48.9	33.1	72.4	35.6	18.8	68.0	281.9	-
ByteTrack + [22]	24.6	15.1	72.1	17.4	7.3	68.3	281.9	1.19×10^{-3}
Walker	**53.0**	**38.6**	**73.1**	**41.6**	**25.4**	**69.5**	**281.9**	$\mathbf{0.14 \times 10^{-3}}$

For the remaining hyperparameters, we conducted a grid search. We here report an analysis of the impact of Walker's most-sensitive inference parameters in Tab. 13.

G.4 Ablation on Loss Components

In Tab. 12, we ablate on the importance of the cycle and forward losses towards our total loss introduced in Sec. 3.5. We find that applying the forward loss on top of the cycle loss results in a considerable improvement in performance, highlighting the importance of identifying and optimizing max-likelihood latent transition states in a mutually-exclusive fashion according to our proposal in Sec. 3.4. In particular, the performance improvements originates from (i) the quality of the forward assignments refined by averaging over all the walks starting from all the nodes in a given cluster and ending on the multi-positive targets for the corresponding starting node, and (ii) the cluster-wise mutual-exclusivity property enforced as described in Sec. 3.4.

G.5 Ablation on Model Complexity

In Tab. 14, we ablate on the FLOPS requirements of different methods on DanceTrack val. We compare Walker to ByteTrack [56] and ByteTrack + [22]. FLOPs are computed using an input size of 3x640x640 to the YOLOX-X detector, of 3x256x128 to [20]'s ResNet-18 Re-ID branch and of 320x7x7 (RoI size in YOLOX-X) to our 4conv-1fc emb. head. ByteTrack + [22] requires a separate R-18 Re-ID head which is $\sim$9$\times$ more computationally expensive (Re-ID FLOPS, Tab. 14) than our tiny embedding head, which operates on small-size RoIs and is computationally negligible wrt. the detector.

G.6 Qualitative Results

We report a qualitative comparison on DanceTrack of the existing self-supervised tracking methods, *i.e.* QDTrack-S [14], QD-Walker (ours), and Walker (ours).

Figs. 6, 8 and 10 show the tracking results for each method, where the same color is used through time to represent the same ID. Figs. 7, 9 and 10 show the ID switches (blue) and correctly tracked bounding boxes (green). The qualitative results remark the superiority of Walker over QDTrack-S. By sharing the inference algorithm with QDTrack-S, QD-Walker demonstrates the superiority of our self-supervised appearance-learning algorithm, showing significantly less ID switches under complex occlusions. This is made possible by our temporal self-supervision in videos, which makes our learned appearance descriptors more robust to the sudden appearance and pose changes in highly dynamic videos as the ones in DanceTrack. Moreover, the improved tracking algorithm of the full Walker further boosts our tracking performance performance. By taking into account the motion information, Walker notably reduces the number of ID switches in uniform appearance settings such as DanceTrack by constraining matches to only happen near likely future positions of an object. Notably, Fig. 11 shows a case of rapid object motion and sudden pose changes. For ease of visualization, we crop all frames around an area of interest, *i.e.* where the dancers are thrown in the air. The dynamic evolutions that the dancers are performing make tracking extremely difficult for a self-supervised tracker trained on static images such as QDTrack-S. This can be noticed by the high amount of ID switches (blue boxes). Instead, our trackers trained on the temporal video stream learn appearance representations robust to the temporal pose changes of the dancers, as it can be seen by the significantly better results and reduced ID switches. It is worth noticing that our motion-constrained tracker (Walker) prevents the ID switch at time $t = \hat{t}$ that still occurs in the unconstrained QD-Walker. Finally, in Fig. 9 we identify a case where both QD-Walker and Walker cannot remedy an ID switch. Due to the sudden change in appearance and pose of the dancer, our trackers initiate a new tracklet for an already existing object in $t = \hat{t} - k$.

Algorithm 1 Training pipeline of Walker for identifying and optimizing pseudo-assignments.

Input: detections $\mathcal{D}_t$ at time t and detections $\mathcal{D}_{t+k}$ at time $t+k$, ground-truth detections $\hat{\mathcal{D}}_t$ at time t and ground-truth detections $\hat{\mathcal{D}}_{t+k}$ at time $t+k$, setting `setting` (*dense* or *sparse*)

embed detections
$\mathbf{Q}_t$ = `embed`($\mathcal{D}_t$)
$\mathbf{Q}_{t+k}$ = `embed`($\mathcal{D}_{t+k}$)
select reference nodes for walk based on setting
if `setting` == *dense*
    $\bar{\mathcal{D}}_t = \hat{\mathcal{D}}_t$
    $\bar{\mathcal{D}}_{t+k} = \hat{\mathcal{D}}_{t+k}$
else if `setting` == *sparse*
    # filter detections by confidence
    $\bar{\mathcal{D}}_t$ = `filterByConf`($\mathcal{D}_t$, β_{obj})
    $\bar{\mathcal{D}}_{t+k}$ = `filterByConf`($\mathcal{D}_{t+k}$, β_{obj})
end if
# negative-positive balance for walk nodes
$\mathbf{Q}_t = (\mathbf{Q}_t^+, \mathbf{Q}_t^-)$ = `negPosBalance`($(\mathbf{Q}_t, \mathcal{D}_t)$, `gt`=$\bar{\mathcal{D}}_t$, `neg_pos_rate`=3)
$\mathbf{Q}_{t+k} = (\mathbf{Q}_{t+k}^+, \mathbf{Q}_{t+k}^-)$ = `negPosBalance`($(\mathbf{Q}_{t+k}, \mathcal{D}_{t+k})$, `gt`=$\bar{\mathcal{D}}_{t+k}$, `neg_pos_rate`=3)
compute cycle probabilities
A_{t+}^{t+k} = `computeTransition`($\mathbf{Q}_t^+, \mathbf{Q}_{t+k}$)
A_{t+k}^{t} = `computeTransition`($\mathbf{Q}_{t+k}, \mathbf{Q}_t$)
$\bar{A}_{t+}^{t}$ = `concatTransitions`(A_t^{t+k}, A_{t+k}^t)
get valid clusters
$\mathcal{C}_t$ = `getClusters`($\mathbf{Q}_t^+$)
$\mathcal{C}_t$ = `set`($\mathcal{C}_t^{\text{high}}$) # keep only unique clusters
$\mathcal{C}_t$ = `sorted`($\mathcal{C}_t$, `key`=$\bar{A}_{t+}^t$)
$\mathcal{C}_t^{\text{valid}}$ = `filterByConf`($\mathcal{C}_t$, $\bar{A}_{t+}^t$, β_{cycle})
find pseudo-assignments
$\mathcal{Z}_{t+k}^{\text{assigned}} = [\,]$ # set of assigned clusters
for $\mathcal{C}_t^i$ `in` $\mathcal{C}_t^{\text{valid}}$
    # find match not in $\mathcal{Z}_{t+k}^{\text{assigned}}$
    $\mathcal{Z}_{t+k}^i$ = `findMatch`(A_{t+}^{t+k}, $\mathcal{C}_t^i$, $\mathcal{Z}_{t+k}^{\text{assigned}}$)
    $\mathcal{Z}_{t+k}^{\text{assigned}}$.`append`($\mathcal{Z}_{t+k}^i$)
end for
# compute losses
$\mathcal{L}_{\text{cycle}}$ = `cycleLoss`($\bar{A}_{t+}^t$, $\mathcal{C}_t^{\text{high}}$)
$\mathcal{L}_{\text{forward}}$ = `forwardLoss`(A_{t+}^{t+k}, $\mathcal{C}_t^{\text{valid}}$, $\mathcal{Z}_{t+k}^{\text{assigned}}$)
$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cycle}} + \mathcal{L}_{\text{forward}}$

Algorithm 2 Inference pipeline of Walker for associating objects across a video sequence.

Input: A video sequence `V`; object detector `Det`
Output: Tracks $\mathcal{T}$ of the video

Initialization: $\mathcal{T} \leftarrow \emptyset$
for frame $I_t \in$ `V`
    # predict detection boxes & scores
    $\mathcal{D}_t \leftarrow$ `Det`(I_t)
    $\mathcal{D}_t^{\text{high}} \leftarrow \emptyset$
    $\mathcal{D}_t^{\text{low}} \leftarrow \emptyset$
    for $d_t^i \in \mathcal{D}_t$
        if $\text{conf}(d_t^i) \geq \beta_{\text{high}}$
            $\mathcal{D}_t^{\text{high}} \leftarrow \mathcal{D}_t^{\text{high}} \cup \{d_t^i\}$
        else if $\beta_{\text{low}} \leq \text{conf}(d_t^i) < \beta_{\text{high}}$
            $\mathcal{D}_t^{\text{low}} \leftarrow \mathcal{D}_t^{\text{low}} \cup \{d\}$
        end if
    end for
    # predict new locations of tracks
    for $t \in \mathcal{T}$
        $t \leftarrow$ `KalmanFilter`(t)
    end for
    # first association
    Associate $\mathcal{T}$ and $\mathcal{D}_t^{\text{high}}$ using W^{++} (Eq. (13)) and match threshold $\beta_{\text{match}}^{\text{high}}$
    $\mathcal{D}_t^{\text{remain}} \leftarrow$ remaining object boxes from $\mathcal{D}_t^{\text{high}}$
    $\mathcal{T}_{\text{remain}} \leftarrow$ remaining tracks from $\mathcal{T}$
    # second association
    Associate $\mathcal{T}_{\text{remain}}$ and $\mathcal{D}_t^{\text{low}}$ using IoU distance and match threshold $\beta_{\text{match}}^{\text{low}}$
    $\mathcal{T}_{\text{unmatched}} \leftarrow$ remaining tracks from $\mathcal{T}_{\text{remain}}$
    # delete unmatched tracks $\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_{\text{unmatched}}$
    # initialize new tracks
    for $d_t^j \in \mathcal{D}_t^{\text{remain}}$
        $\mathcal{T} \leftarrow \mathcal{T} \cup \{d_t^j\}$
    end for
end for
return $\mathcal{T}$

Algorithm 3 Inference pipeline of QD-Walker for associating objects across a video sequence.

Input: frame index t, detections $\mathbf{b}_i$, scores s_i, detection embeddings $\mathbf{n}_i$ for $i = 1 \dots N$, and track embeddings $\mathbf{m}_j$ for $j = 1 \dots M$

```
# compute matching scores
DuplicateRemoval(b_i)
for i = 1...N, j = 1...M
    f(i, j) = biwalk(n_i, m_j)
end for
# track management
for i = 1...N
    c = max(f(i))                                   # match confidence
    j_match = argmax(f(i))                          # matched track ID
    # object match found
    if c > β_match and s_i > β_obj
    and isNotBackdrop(j_match)
        # update track
        updateTrack(j_match, b_i, n_i, t)
    else if s_i > β_new
        # create new track
        createTrack(b_i, n_i, t)
    else
        # add new backdrop
        addBackdrop(b_i, n_i, t)
    end if
end for
```

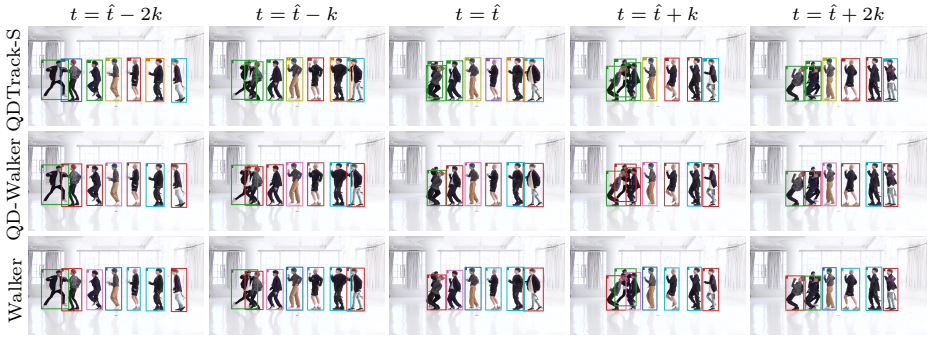

Fig. 6: Tracking results on the sequence *0058* of the DanceTrack validation set. We analyze 5 frames centered around the frame #128 at time $\hat{t}$ and spaced by $k{=}4/30$ seconds. We compare the self-supervised trackers QDTrack-S [14], QD-Walker (ours), and Walker (ours). On each row, boxes of the same color correspond to the same tracking ID.

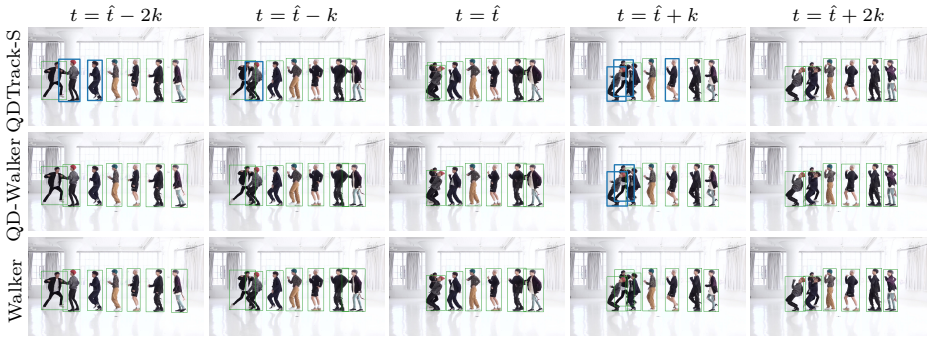

Fig. 7: ID switches on the sequence *0058* of the DanceTrack validation set. We analyze 5 frames centered around the frame #128 at time $\hat{t}$ and spaced by $k{=}4/30$ seconds. We compare the self-supervised trackers QDTrack-S [14], QD-Walker (ours), and Walker (ours). On each row, boxes colored in green are correctly tracked, while blue ones represent ID switches.

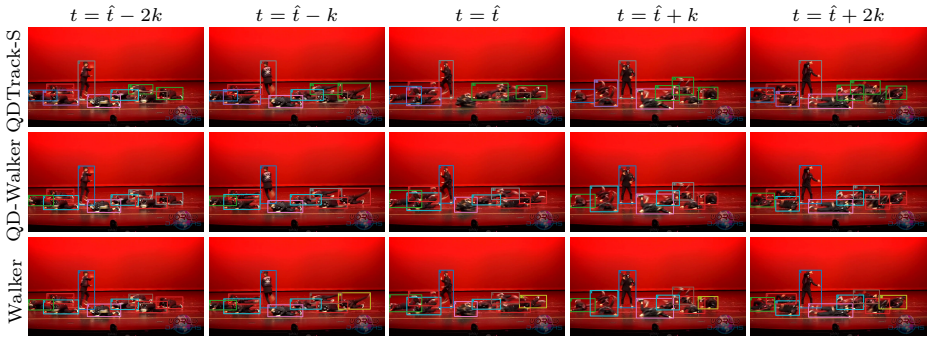

Fig. 8: Tracking results on the sequence *0077* of the DanceTrack validation set. We analyze 5 frames centered around the frame #222 at time $\hat{t}$ and spaced by k=5/30 seconds. We compare the self-supervised trackers QDTrack-S [14], QD-Walker (ours), and Walker (ours). On each row, boxes of the same color correspond to the same tracking ID.

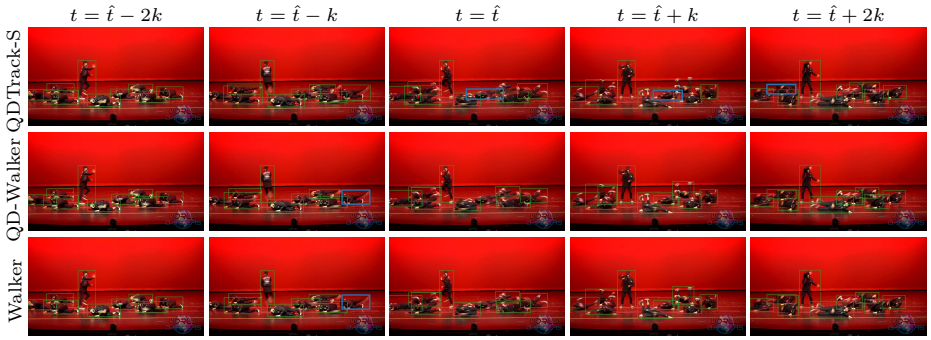

Fig. 9: ID switches on the sequence *0077* of the DanceTrack validation set. We analyze 5 frames centered around the frame #222 at time $\hat{t}$ and spaced by k=5/30 seconds. We compare the self-supervised trackers QDTrack-S [14], QD-Walker (ours), and Walker (ours). On each row, boxes colored in green are correctly tracked, while blue ones represent ID switches.

Fig. 10: Tracking results on the sequence *0081* of the DanceTrack validation set. We analyze 5 frames centered around the frame #40 at time $\hat{t}$ and spaced by k=3/30 seconds. We compare the self-supervised trackers QDTrack-S [14], QD-Walker (ours), and Walker (ours). On each row, boxes of the same color correspond to the same tracking ID. For ease of visualization, we crop all frames around an area of interest.

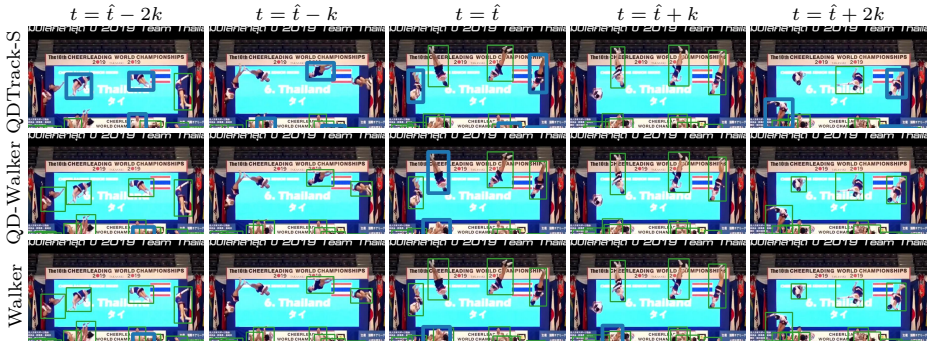

Fig. 11: ID switches on the sequence *0081* of the DanceTrack validation set. We analyze 5 frames centered around the frame #40 at time $\hat{t}$ and spaced by k=3/30 seconds. We compare the self-supervised trackers QDTrack-S [14], QD-Walker (ours), and Walker (ours). On each row, boxes colored in green are correctly tracked, while blue ones represent ID switches. For ease of visualization, we crop around an area of interest.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: Bot-sort: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)
2. Athar, A., Luiten, J., Voigtlaender, P., Khurana, T., Dave, A., Leibe, B., Ramanan, D.: Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1674–1683 (2023)
3. Bastani, F., He, S., Madden, S.: Self-supervised multi-object tracking with cross-input consistency. Advances in Neural Information Processing Systems **34**, 13695–13706 (2021)
4. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). pp. 3464–3468. IEEE (2016)
5. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
6. Cao, J., Weng, X., Khirodkar, R., Pang, J., Kitani, K.: Observation-centric sort: Rethinking sort for robust multi-object tracking. arXiv preprint arXiv:2203.14360 (2022)
7. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
8. Collicott, B., Sarvaiya, M., Weston, B.: Self-supervised feature learning for online multi-object tracking
9. Dave, A., Khurana, T., Tokmakov, P., Schmid, C., Ramanan, D.: Tao: A large-scale benchmark for tracking any object. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. pp. 436–454. Springer (2020)
10. Dendorfer, P., Osep, 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)
11. Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H.: Strongsort: Make deepsort great again. IEEE Transactions on Multimedia (2023)
12. Elhoseny, M.: Multi-object detection and tracking (modt) machine learning model for real-time video surveillance systems. Circuits, Systems, and Signal Processing **39**(2), 611–630 (2020)
13. Ess, A., Schindler, K., Leibe, B., Van Gool, L.: Object detection and tracking for autonomous navigation in dynamic environments. The International Journal of Robotics Research **29**(14), 1707–1725 (2010)
14. Fischer, T., Pang, J., Huang, T.E., Qiu, L., Chen, H., Darrell, T., Yu, F.: Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking. arXiv preprint arXiv:2210.06984 (2022)
15. Gan, Y., Han, R., Yin, L., Feng, W., Wang, S.: Self-supervised multi-view multi-human association and tracking. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 282–290 (2021)
16. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)

17. Gupta, A., Wu, J., Deng, J., Fei-Fei, L.: Siamese masked autoencoders. arXiv preprint arXiv:2305.14344 (2023)
18. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
19. Heigold, G., Minderer, M., Gritsenko, A., Bewley, A., Keysers, D., Lučić, M., Yu, F., Kipf, T.: Video owl-vit: Temporally-consistent open-world localization in video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13802–13811 (2023)
20. Ho, K., Kardoost, A., Pfreundt, F.J., Keuper, J., Keuper, M.: A two-stage minimum cost multicut approach to self-supervised multiple person tracking. In: Proceedings of the Asian Conference on Computer Vision (2020)
21. Huang, K., Lertniphonphan, K., Chen, F., Li, J., Wang, Z.: Multi-object tracking by self-supervised learning appearance model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3162–3168 (2023)
22. Jabri, A., Owens, A., Efros, A.: Space-time correspondence as a contrastive random walk. Advances in neural information processing systems **33**, 19545–19560 (2020)
23. Kalman, R.E.: A new approach to linear filtering and prediction problems (1960)
24. Karthik, S., Prabhu, A., Gandhi, V.: Simple unsupervised multi-object tracking. arXiv preprint arXiv:2006.02609 (2020)
25. Kim, S., Lee, J., Ko, B.C.: Ssl-mot: self-supervised learning based multi-object tracking. Applied Intelligence **53**(1), 930–940 (2023)
26. Li, S., Danelljan, M., Ding, H., Huang, T.E., Yu, F.: Tracking every thing in the wild. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII. pp. 498–515. Springer (2022)
27. Li, S., Fischer, T., Ke, L., Ding, H., Danelljan, M., Yu, F.: Ovtrack: Open-vocabulary multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5567–5577 (2023)
28. Li, S., Ke, L., Danelljan, M., Piccinelli, L., Segu, M., Van Gool, L., Yu, F.: Matching anything by segmenting anything. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18963–18973 (2024)
29. Li, S., Ke, L., Yang, Y.H., Piccinelli, L., Segu, M., Danelljan, M., Van Gool, L.: Slack: Semantic, location and appearance aware open-vocabulary tracking. In: Computer Vision–ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings. Springer (2024)
30. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
31. Liu, Z., Segu, M., Yu, F.: Cooler: Class-incremental learning for appearance-based multiple object tracking. arXiv preprint arXiv:2310.03006 (2023)
32. Lu, Z., Rathod, V., Votel, R., Huang, J.: Retinatrack: Online single stage joint detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14668–14678 (2020)
33. 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**, 548–578 (2021)
34. Meinhardt, T., Kirillov, A., Leal-Taixe, 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)
35. 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)

36. Park, Y., Lepetit, V., Woo, W.: Multiple 3d object tracking for augmented reality. In: 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. pp. 117–120. IEEE (2008)
37. Reid, D.: An algorithm for tracking multiple targets. IEEE transactions on Automatic Control **24**(6), 843–854 (1979)
38. Segu, M., Piccinelli, L., Li, S., Yang, Y.H., Schiele, B., Van Gool, L.: Samba: Synchronized set-of-sequences modeling for end-to-end multiple object tracking. arXiv preprint (2024)
39. Segu, M., Schiele, B., Yu, F.: Darth: Holistic test-time adaptation for multiple object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9717–9727 (2023)
40. 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)
41. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
42. 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)
43. 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)
44. Sun, T., Segu, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., Yu, F.: Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21371–21382 (2022)
45. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2566–2576 (2019)
46. Wang, Y.H.: Smiletrack: Similarity learning for multiple object tracking. arXiv preprint arXiv:2211.08824 (2022)
47. Wang, Z., Zhao, H., Li, Y.L., Wang, S., Torr, P., Bertinetto, L.: Do different tracking tasks require different appearance models? Advances in Neural Information Processing Systems **34**, 726–738 (2021)
48. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 107–122. Springer (2020)
49. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)
50. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
51. Yang, F., Chang, X., Dang, C., Zheng, Z., Sakti, S., Nakamura, S., Wu, Y.: Remots: Self-supervised refining multi-object tracking and segmentation. arXiv preprint arXiv:2007.03200 (2020)
52. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos. arXiv preprint arXiv:2304.11968 (2023)

53. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5188–5197 (2019)
54. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2636–2645 (2020)
55. Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: Motr: End-to-end multiple-object tracking with transformer. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII. pp. 659–675. Springer (2022)
56. 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: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII. pp. 1–21. Springer (2022)
57. 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**, 3069–3087 (2021)
58. Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., Tian, Q.: Mars: A video benchmark for large-scale person re-identification. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14. pp. 868–884. Springer (2016)
59. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV. pp. 474–490. Springer (2020)