

StrongSORT: Make DeepSORT Great Again

Yunhao Du¹, Yang Song¹, Bo Yang³, and Yanyun Zhao^{1,2}

¹Beijing University of Posts and Telecommunications

²Beijing Key Laboratory of Network System and Network Culture, China

{dyh_bupt, sy12138, zyy}@bupt.edu.cn

³Xidian University

byang1@stu.xidian.edu.cn

Abstract. Existing Multi-Object Tracking (MOT) methods can be roughly classified as tracking-by-detection and joint-detection-association paradigms. Although the latter has elicited more attention and demonstrates comparable performance relative to the former, we claim that the tracking-by-detection paradigm is still the optimal solution in terms of tracking accuracy. In this paper, we revisit the classic tracker DeepSORT and upgrade it from various aspects, i.e., detection, embedding and association. The resulting tracker, called **StrongSORT**, sets new HOTA and IDF1 records on MOT17 and MOT20. We also present two lightweight and plug-and-play algorithms to further refine the tracking results. Firstly, an appearance-free link model (AFLink) is proposed to associate short tracklets into complete trajectories. To the best of our knowledge, this is the first global link model without appearance information. Secondly, we propose Gaussian-smoothed interpolation (GSI) to compensate for missing detections. Instead of ignoring motion information like linear interpolation, GSI is based on the Gaussian process regression algorithm and can achieve more accurate localizations. Moreover, AFLink and GSI can be plugged into various trackers with a negligible extra computational cost (591.9 and 140.9 Hz, respectively, on MOT17). By integrating StrongSORT with the two algorithms, the final tracker **Strong-SORT++** ranks first on MOT17 and MOT20 in terms of HOTA and IDF1 metrics and surpasses the second-place one by 1.3 - 2.2. Code will be released soon.

Keywords: Multi-Object Tracking, Tracking-by-detection, Lightweight

1 Introduction

Multi-Object Tracking (MOT) plays an essential role in video understanding. It aims to detect and track all specific classes of objects frame by frame. In the past few years, the tracking-by-detection paradigm [3, 4, 36, 62, 69] dominated the MOT task. It performs detection per frame and formulates the MOT problem as a data association task. Benefiting from high-performing object detection models, tracking-by-detection methods have gained favor due to their excellent

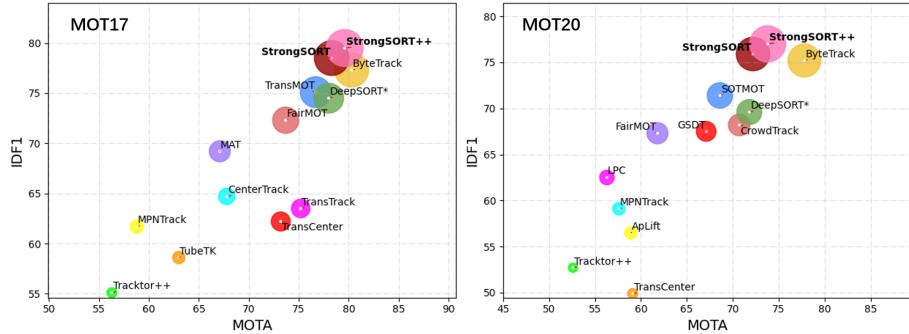


Fig. 1. IDF1-MOTA-HOTA comparisons of state-of-the-art trackers with our proposed StrongSORT and StrongSORT++ on MOT17 and MOT20 test sets. The horizontal axis is MOTA, the vertical axis is IDF1, and the radius of the circle is HOTA. “*” represents our reproduced version. Our StrongSORT++ achieves the best IDF1 and HOTA and comparable MOTA performance.

performance. However, these methods generally require multiple computationally expensive components, such as a detector and an embedding model. To solve this problem, several recent methods [1, 60, 74] integrate the detector and embedding model into a unified framework. Moreover, joint detection and embedding training appears to produce better results compared with the separate one [47]. Thus, these methods (*joint trackers*) achieve comparable or even better tracking accuracy compared with tracking-by-detection ones (*separate trackers*).

The success of joint trackers has motivated researchers to design unified tracking frameworks for various components, e.g., detection, motion, embedding, and association models [30, 32, 38, 57, 59, 65, 68]. However, we argue that two problems exist in these joint frameworks: (1) the competition between different components and (2) limited data for training these components jointly. Although several strategies have been proposed to solve them, these problems still lower the upper bound of tracking accuracy. On the contrary, the potential of separate trackers seems to be underestimated.

In this paper, we revisit the classic separate tracker DeepSORT [62], which is among the earliest methods that apply the deep learning model to the MOT task. It's claimed that DeepSORT underperforms compared with state-of-the-art methods because of its outdated techniques, rather than its tracking paradigm. We show that by simply equipping DeepSORT with advanced components in various aspects, resulting in the proposed *StrongSORT*, it can achieve new SOTA on popular benchmarks MOT17 [35] and MOT20 [11].

Two lightweight, plug-and-play, model-independent, appearance-free algorithms are also proposed to refine the tracking results. Firstly, to better exploit the global information, several methods propose to associate short tracklets into trajectories by using a global link model [12, 39, 55, 56, 67]. They usually generate accurate but incomplete tracklets and associate them with global information in

an offline manner. Although these methods improve tracking performance significantly, they all rely on computation-intensive models, especially appearance embeddings. By contrast, we propose an appearance-free link model (AFLink) that only utilizes spatio-temporal information to predict whether the two input tracklets belong to the same ID.

Secondly, linear interpolation is widely used to compensate for missing detections [12, 21, 37, 40, 41, 73]. However, it ignores motion information, which limits the accuracy of the interpolated positions. To solve this problem, we propose the Gaussian-smoothed interpolation algorithm (GSI), which enhances the interpolation by using the Gaussian process regression algorithm [61].

Extensive experiments prove that the two proposed algorithms achieve notable improvements on StrongSORT and other state-of-the-art trackers, e.g., CenterTrack [77], TransTrack [50] and FairMOT [74]. Particularly, by applying AFLink and GSI to StrongSORT, we obtain a stronger tracker called *StrongSORT++*. It achieves 64.4 HOTA, 79.5 IDF1 and 79.6 MOTA (7.1 Hz) on the MOT17 test set and 62.6 HOTA, 77.0 IDF1 and 73.8 MOTA (1.4 Hz) on the MOT20 test set. Figure 1 compares our StrongSORT and StrongSORT++ with state-of-the-art trackers on MOT17 and MOT20 test sets. Our method achieves the best IDF1 and HOTA and a comparable MOTA performance. Furthermore, AFLink and GSI respectively run at 591.9 and 140.9 Hz on MOT17, 224.0 and 17.6 Hz on MOT20, resulting in a negligible computational cost.

The contributions of our work are summarized as follows:

- 1) We revisit the classic separate tracker DeepSORT and improve it from various aspects, resulting in StrongSORT, which sets new HOTA and IDF1 records on MOT17 and MOT20 datasets.
- 2) We propose two lightweight and appearance-free algorithms, AFLink and GSI, which can be plugged into various trackers to improve their performance by a large margin.
- 3) By integrating StrongSORT with AFLink and GSI, our StrongSORT++ ranks first on MOT17 and MOT20 in terms of widely used HOTA and IDF1 metrics and surpasses the second-place one [73] by 1.3 - 2.2.

2 Related Work

2.1 Separate and Joint Trackers

MOT methods can be classified as separate and joint trackers. Separate trackers [3, 4, 7, 8, 15, 36, 62, 69] follow the tracking-by-detection paradigm and localize targets first and then associate them with information on appearance, motion, etc. Benefiting from the rapid development of object detection [17, 42, 43, 52, 53, 78], separate trackers have dominated the MOT task for years. Recently, several joint trackers [30, 32, 38, 57, 59, 65, 68] have been proposed to train detection and some other components jointly, e.g., motion, embedding and association models. The main benefit of these trackers is their low computational cost and comparable performance. However, we claim that joint trackers face two major problems:

competition between different components and limited data for training the components jointly. The two problems limit the upper bound of tracking accuracy. Therefore, we argue that the tracking-by-detection paradigm is still the optimal solution for tracking performance.

Meanwhile, several recent studies [48, 49, 73] have abandoned appearance information and relied only on high-performance detectors and motion information, which achieve high running speed and state-of-the-art performance on MOTChallenge benchmarks [11, 35]. However, we argue that it's partly due to the general simplicity of motion patterns in these datasets. Abandoning appearance features would lead to poor robustness in more complex scenes. In this paper, we adopt the DeepSORT-like [62] paradigm and equip it with advanced techniques from various aspects to confirm the effectiveness of this classic framework.

2.2 Global Link in MOT

To exploit rich global information, several methods refine the tracking results with a global link model [12, 39, 55, 56, 67]. They tend to generate accurate but incomplete tracklets by using spatio-temporal and/or appearance information first. Then, these tracklets are linked by exploring global information in an offline manner. TNT [56] designs a multi-scale TrackletNet to measure the connectivity between two tracklets. It encodes motion and appearance information in a unified network by using multi-scale convolution kernels. TPM [39] presents a tracklet-plane matching process to push easily confusable tracklets into different tracklet-planes, which helps reduce the confusion in the tracklet matching step. ReMOT [67] is improved from ReMOTS [66]. Given any tracking results, ReMOT splits imperfect trajectories into tracklets and then merges them with appearance features. GIAOTracker [12] proposes a complex global link algorithm that encodes tracklet appearance features by using an improved ResNet50-TP model [16] and associates tracklets together with spatial and temporal costs. Although these methods yield notable improvements, they all rely on appearance features, which bring high computational cost. Differently, we propose the AFLink model that only exploits motion information to predict the link confidence between two tracklets. By designing an appropriate model framework and training process, AFLink benefits various state-of-the-art trackers with a negligible extra cost. To the best of our knowledge, this is the first appearance-free and lightweight global link model for the MOT task.

2.3 Interpolation in MOT

Linear interpolation is widely used to fill the gaps of recovered trajectories for missing detections [12, 21, 37, 40, 41, 73]. Despite its simplicity and effectiveness, linear interpolation ignores motion information, which limits the accuracy of the restored bounding boxes. To solve this problem, several strategies have been proposed to utilize spatio-temporal information effectively. V-IOUTracker [5] extends IOUTracker [4] by falling back to single-object tracking [20, 25] while missing detection occurs. MAT [19] smooths the linearly interpolated trajectories

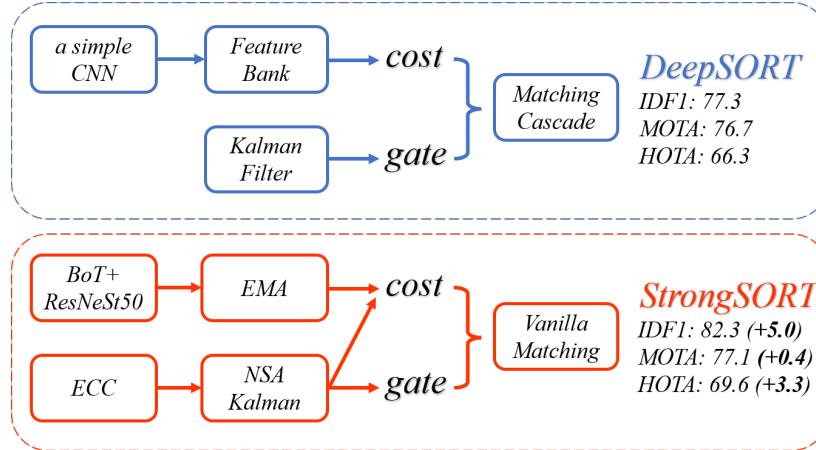


Fig. 2. Framework and performance comparison between DeepSORT and StrongSORT. Performance is evaluated on the MOT17 validation set based on detections predicted by YOLOX [17].

nonlinearly by adopting a cyclic pseudo-observation trajectory filling strategy. An extra camera motion compensation (CMC) model [14] and Kalman filter [26] are needed to predict missing positions. MAATrack [49] simplifies it by applying only the CMC model. All these methods apply extra models, i.e., single-object tracker, CMC, Kalman filter, in exchange for performance gains. Instead, we propose to model nonlinear motion on the basis of the Gaussian process regression (GPR) algorithm [61]. Without additional time-consuming components, our proposed GSI algorithm achieves a good trade-off between accuracy and efficiency.

The most similar work with our GSI is [79], which uses the GPR algorithm to smooth the uninterpolated tracklets for accurate velocity predictions. However, it works for the event detection task in surveillance videos. Differently, we study on the MOT task and adopt GPR to refine the interpolated localizations. Moreover, we present an adaptive smoothness factor, instead of presetting a hyperparameter like [79].

3 StrongSORT

In this section, we present various approaches to improve the classic tracker DeepSORT [62]. Specifically, we review DeepSORT in Section 3.1 and introduce StrongSORT in Section 3.2. Notably, we do not claim any algorithmic novelty in this section. Instead, our contributions here lie in giving a clear understanding of DeepSORT and equipping it with various advanced techniques to prove the effectiveness of its paradigm.

3.1 Review of DeepSORT

We briefly summarize DeepSORT as a two-branch framework, that is, *appearance branch* and *motion branch*, as shown in the top half of Figure 2.

In the appearance branch, given detections in each frame, the deep appearance descriptor (a simple CNN), which is pretrained on the person re-identification dataset MARS [75], is applied to extract their appearance features. It utilizes a feature bank mechanism to store the features of the last 100 frames for each tracklet. As new detections come, the smallest cosine distance between the feature bank R_i of the i -th tracklet and the feature f_j of the j -th detection is computed as

$$d(i, j) = \min\{1 - f_j^T f_k^{(i)} \mid f_k^{(i)} \in R_i\}. \quad (1)$$

The distance is used as the matching cost during the association procedure.

In the motion branch, the Kalman filter algorithm [26] accounts for predicting the positions of tracklets in the current frame. Then, Mahalanobis distance is used to measure the spatio-temporal dissimilarity between tracklets and detections. DeepSORT takes this motion distance as a gate to filter out unlikely associations.

Afterwards, the matching cascade algorithm is proposed to solve the association task as a series of subproblems instead of a global assignment problem. The core idea is to give greater matching priority to more frequently seen objects. Each association subproblem is solved using the Hungarian algorithm [29].

3.2 Stronger DeepSORT

Our improvements over DeepSORT lie mainly in the two branches, as shown in the bottom half of Figure 2. For the appearance branch, a stronger appearance feature extractor, BoT [34], is applied to replace the original simple CNN. By taking ResNeSt50 [71] as the backbone and pretraining on the DukeMTMC-reID [44] dataset, it can extract much more discriminative features. In addition, we replace the feature bank with the feature updating strategy proposed in [60], which updates appearance state e_i^t for the i -th tracklet at frame t in an exponential moving average (EMA) manner as follows:

$$e_i^t = \alpha e_i^{t-1} + (1 - \alpha) f_i^t, \quad (2)$$

where f_i^t is the appearance embedding of the current matched detection and $\alpha = 0.9$ is a momentum term. The EMA updating strategy not only enhances the matching quality, but also reduces the time consumption.

For the motion branch, similar to [19, 27, 49], we adopt ECC [14] for camera motion compensation. Furthermore, the vanilla Kalman filter is vulnerable w.r.t. low-quality detections [49] and ignores the information on the scales of detection noise. To solve this problem, we borrow the NSA Kalman algorithm from [12] that proposes a formula to adaptively calculate the noise covariance \tilde{R}_k :

$$\tilde{R}_k = (1 - c_k) R_k, \quad (3)$$

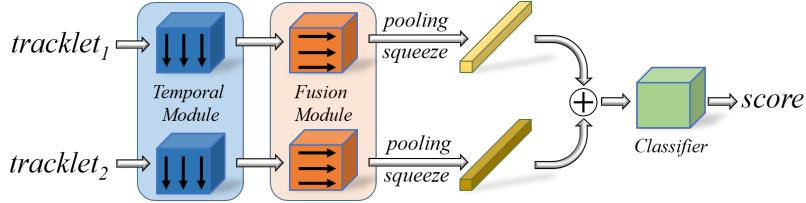


Fig. 3. Framework of the AFLink model. It adopts the spatio-temporal information of two tracklets as the input and then predicts their connectivity.

where R_k is the preset constant measurement noise covariance and c_k is the detection confidence score at state k .

Furthermore, instead of employing only the appearance feature distance during matching, we solve the assignment problem with both appearance and motion information, similar to [60]. Cost matrix C is a weighted sum of appearance cost A_a and motion cost A_m as follows:

$$C = \lambda A_a + (1 - \lambda) A_m, \quad (4)$$

where weight factor λ is set to 0.98. Another interesting finding is that although the matching cascade algorithm is not trivial in DeepSORT, it limits the performance as the tracker becomes more powerful. The reason is that as the tracker becomes stronger, it becomes more robust to confusable associations. Therefore, additional prior constraints would limit the matching accuracy. We replace matching cascade with vanilla global linear assignment.

4 StrongSORT++

We presented a strong tracker in Section 3. In this section, we introduce two lightweight, plug-and-play, model-independent, appearance-free algorithms, namely AFLink and GSI, to further refine the tracking results. We call the final method StrongSORT++, which integrates StrongSORT with the two algorithms.

4.1 AFLink

The global link for tracklets is used in several works to pursue highly accurate associations. However, they generally rely on computationally expensive components and numerous hyperparameters to fine-tune. For example, the link algorithm in GIAOTracker [12] utilizes an improved ResNet50-TP [16] to extract tracklets 3D features and performs association with additional spatial and temporal distances. This means 6 hyperparameters (3 thresholds and 3 weight factors) are to be fine-tuned, which incurs additional tuning experiments and poor robustness. Moreover, we find that over-reliance on appearance features is vulnerable to noise. Motivated by this, we design an appearance-free model,

AFLink, to predict the connectivity between two tracklets by relying only on spatio-temporal information.

Figure 3 shows the two-branch framework of the AFLink model. It adopts two tracklets T_i and T_j as the input, where $T_* = \{f_k, x_k, y_k\}_{k=1}^N$ consists of the frames f_k and positions (x_k, y_k) of the recent $N = 30$ frames. Zero padding is used for those shorter than 30 frames. A temporal module is applied to extract features by convolving along the temporal dimension with 7×1 kernels. Then, a fusion module performs 1×3 convolutions to integrate the information from different feature dimensions, namely f , x and y . The two resulting feature maps are pooled and squeezed to feature vectors respectively, and then concatenated, which includes rich spatio-temporal information. Finally, an MLP is used to predict a confidence score for association. Note that the temporal module and fusion module of the two branches are not tied.

During association, we filter out unreasonable tracklet pairs with spatio-temporal constraints. Then, the global link is solved as a linear assignment task [29] with the predicted connectivity score.

4.2 GSI

Interpolation is widely used to fill the gaps in trajectories caused by missing detections. Linear interpolation is highly popular due to its simplicity. However, its accuracy is limited because it does not use motion information. Although several strategies have been proposed to solve this problem, they generally introduce additional time-consuming modules, e.g., single-object tracker, Kalman filter, ECC. Differently, we present a lightweight interpolation algorithm that employs Gaussian process regression [61] to model nonlinear motion.

We formulate the GSI model for the i -th trajectory as follows:

$$p_t = f^{(i)}(t) + \epsilon, \quad (5)$$

where $t \in F$ is the frame, $p_t \in P$ is the position coordinate variate at frame t (i.e., x, y, w, h) and $\epsilon \sim N(0, \sigma^2)$ is Gaussian noise. Given tracked and linearly interpolated trajectories $S^{(i)} = \{t^{(i)}, p_t^{(i)}\}_{t=1}^L$ with length L , the task of nonlinear motion modeling is solved by fitting the function $f^{(i)}$. We assume that it obeys a Gaussian process $f^{(i)} \in GP(0, k(\cdot, \cdot))$, where $k(x, x') = \exp(-\frac{\|x-x'\|^2}{2\lambda^2})$ is a radial basis function kernel. On the basis of the properties of the Gaussian process, given new frame set F^* , its smoothed positions P^* is predicted by

$$P^* = K(F^*, F)(K(F, F) + \sigma^2 I)^{-1}P, \quad (6)$$

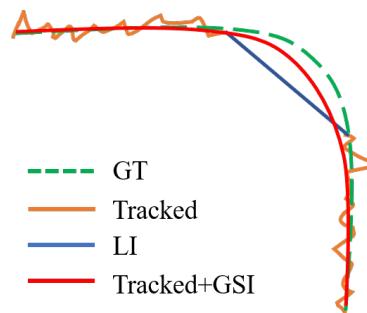


Fig. 4. Illustration of the difference between linear interpolation (LI) and the proposed Gaussian-smoothed interpolation (GSI).

where $K(\cdot, \cdot)$ is a covariance function based on $k(\cdot, \cdot)$. Moreover, hyperparameter λ controls the smoothness of the trajectory, which should be related with its length. We simply design it as a function adaptive to length l as follows:

$$\lambda = \tau * \log(\tau^3 / l), \quad (7)$$

where τ is set to 10.

Figure 4 illustrates an example of the difference between GSI and linear interpolation (LI). The raw tracked results (in orange) generally include noisy jitter, and LI (in blue) ignores motion information. Our GSI (in red) solve both problems simultaneously by smoothing the entire trajectory with an adaptive smoothness factor.

5 Experiments

5.1 Datasets and Evaluation Metrics

Datasets. We conduct experiments on MOT17 [35] and MOT20 [11] datasets under the "private detection" protocol. MOT17 is a popular dataset for MOT, which consists of 7 sequences, 5,316 frames for training and 7 sequences, 5919 frames for testing. MOT20 is set for highly crowded challenging scenes, with 4 sequences, 8,931 frames for training and 4 sequences, 4,479 frames for testing. For ablation studies, we take the first half of each sequence in the MOT17 training set for training and the last half for validation following [73, 77]. We use DukeMTMC [44] to pretrain our appearance feature extractor. We train the detector on the CrowdHuman dataset [46] and MOT17 half training set for ablation following [50, 63, 70, 73, 77]. We add Cityperson [72] and ETHZ [13] for testing as in [30, 60, 73, 74].

Metrics. We use the metrics MOTA, IDs, IDF1, HOTA, AssA, DetA and FPS to evaluate tracking performance [2, 33, 44]. MOTA is computed based on FP, FN and IDs, and focuses more on detection performance. By comparison, IDF1 better measures the consistency of ID matching [23]. HOTA is an explicit combination of detection score DetA and association score AssA, which balances the effects of performing accurate detection and association into a single unified metric. Moreover, it evaluates at a number of different distinct detection similarity values (0.05 to 0.95 in 0.05 intervals) between predicted and GT bounding boxes, instead of setting a single value (i.e., 0.5) like MOTA and IDF1, and better takes localization accuracy into account.

5.2 Implementation Details

For detection, we adopt YOLOX-X [17] pretrained on COCO [31] as our detector for an improved time-accuracy trade-off. The training schedule is similar to that in [73]. In inference, a threshold of 0.8 is set for non-maximum suppression (NMS) and a threshold of 0.6 for detection confidence. For StrongSORT, the feature distance threshold is 0.45, the warp mode for ECC is *MOTION EUCLIDEAN*,

Table 1. Ablation study on the MOT17 validation set for basic strategies, i.e., stronger feature extractor (BoT), camera motion compensation (ECC), NSA Kalman filter (NSA), EMA feature updating mechanism (EMA), matching with motion cost (MC) and abandoning matching cascade (woC). (best in bold)

Method	BoT	ECC	NSA	EMA	MC	woC	IDF1(\uparrow)	MOTA(\uparrow)	HOTA(\uparrow)	FPS(\uparrow)
Baseline	-	-	-	-	-	-	77.3	76.7	66.3	13.8
StrongSORTv1	✓						79.5	76.8	67.8	8.3
StrongSORTv2	✓	✓					79.7	77.1	67.9	6.3
StrongSORTv3	✓	✓	✓				79.7	77.1	68.3	6.2
StrongSORTv4	✓	✓	✓	✓			80.1	77.0	68.2	7.4
StrongSORTv5	✓	✓	✓	✓	✓		80.9	77.0	68.9	7.4
StrongSORTv6	✓	✓	✓	✓	✓	✓	82.3	77.1	69.6	7.5

the momentum term α in EMA is 0.9 and the weight factor for appearance cost λ is 0.98. For GSI, the maximum gap allowed for interpolation is 20 frames, and hyperparameter τ is 10.

For AFLink, the temporal module consists of four convolution layers with 7×1 kernels and $\{32, 64, 128, 256\}$ output channels. Each convolution is followed by a BN layer [24] and a ReLU activation layer [18]. The fusion module includes a 1×3 convolution, a BN and a ReLU. It doesn't change the number of channels. The classifier is an MLP with two fully connected layers and a ReLU layer inserted in between. The training data are generated by cutting annotated trajectories into tracklets with random spatio-temporal noise at a 1:3 ratio of positive and negative samples. We use Adam as the optimizer [28], cross-entropy loss as the objective function and train it for 20 epochs with a cosine annealing learning rate schedule. The overall training process takes just over 10 seconds. In inference, a temporal distance threshold of 30 frames and a spatial distance threshold of 75 pixels are used to filter out unreasonable association pairs. Finally, the association is considered if its prediction score is larger than 0.95.

All experiments are conducted on a server machine with a single V100.

5.3 Ablation Studies

Albation study for StrongSORT. Table 1 summarizes the path from Deep-SORT to StrongSORT:

- 1) BoT: Replacing the original feature extractor with BoT leads to a significant improvement for IDF1, indicating that association quality benefits from more discriminative appearance features.
- 2) ECC: The CMC model results in a slight increase in IDF1 and MOTA, implying that it helps extract more precise motion information.
- 3) NSA: The NSA Kalman filter improves HOTA but not MOTA and IDF1. This means it enhances positioning accuracy.
- 4) EMA: The EMA feature updating mechanism brings not only superior association, but also faster speed.
- 5) MC: Matching with both appearance and motion cost aids association.

Table 2. Results of applying AFLink and GSI to various MOT methods. All experiments are performed on the MOT17 validation set. (best in bold)

Method	AFLink	GSI	IDF1(↑)	MOTA(↑)	HOTA(↑)
StrongSORTv1	-	-	79.5	76.8	67.8
	✓	-	80.0	76.8	68.1
	✓	✓	80.4(+0.9)	78.2(+1.4)	68.9(+1.1)
StrongSORTv3	-	-	79.7	77.1	68.3
	✓	-	80.5	77.1	68.6
	✓	✓	80.9(+1.2)	78.7(+1.6)	69.5(+1.2)
StrongSORTv6	-	-	82.3	77.1	69.6
	✓	-	82.5	77.1	69.6
	✓	✓	83.3(+1.0)	78.7(+1.6)	70.8(+1.2)
CenterTrack [77]	-	-	64.6	66.8	55.3
	✓	-	68.3	66.9	57.2
	✓	✓	68.4(+3.8)	66.9(+0.1)	57.6(+2.3)
TransTrack [50]	-	-	68.6	67.7	58.1
	✓	-	69.1	67.7	58.3
	✓	✓	69.9(+1.3)	69.6(1.9)	59.4(+1.3)
FairMOT [74]	-	-	72.7	69.1	57.3
	✓	-	73.2	69.2	57.6
	✓	✓	74.2(+1.5)	71.1(+2.0)	59.0(+1.7)

Table 3. Comparison of linear interpolation (LI) and our proposed Gaussian-smoothed interpolation (GSI). We take StrongSORT+AFLink as the baseline and experiment on the MOT17 validation set. (best in bold)

Method	IDF1(↑)	MOTA(↑)	HOTA(↑)	FPS(↑)
StrongSORT+AFLink	82.5	77.1	69.6	7.5
StrongSORT+AFLink+LI	83.2	78.5	70.6	7.4
StrongSORT+AFLink+GSI	83.3	78.7	70.8	7.1

6) woC: For the stronger tracker, the matching cascade algorithm with redundant prior information limits the tracking accuracy. By simply employing a vanilla matching method, IDF1 is improved by a large margin.

Ablation study for AFLink and GSI. We apply AFLink and GSI on six different trackers, i.e., three versions of StrongSORT and three state-of-the-art trackers (CenterTrack [77], TransTrack [50] and FairMOT [74]). Their results are shown in Table 2. The first line of the results for each tracker is the original performance. The application of AFLink (the second line) brings different levels of improvement for the different trackers. Specifically, poorer trackers tend to benefit more from AFLink due to more missing associations. Particularly, the IDF1 of CenterTrack is improved by 3.7. The third line of the results for each tracker proves the effectiveness of GSI for both detection and association. Different from AFLink, GSI works better on stronger trackers. It would be confused by the large amount of false association in poor trackers. Table 3 compares our GSI with LI. The results show that GSI yields better performance with a little extra computational cost.

Table 4. Comparison with state-of-the-art MOT methods on the MOT17 test set. “*” represents our reproduced version. The two best results for each metric are bolded and highlighted in red and blue.

Method	Ref.	HOTA(↑)	IDF1(↑)	MOTA(↑)	AssA(↑)	DetA(↑)	IDs(↓)	FPS(↑)
SORT [3]	ICIP2016	34.0	39.8	43.1	31.8	37.0	4,852	143.3
DAN [51]	TPAMI2019	39.3	49.5	52.4	36.3	43.1	8,431	6.3
TPM [39]	PR2020	41.5	52.6	54.2	40.9	42.5	1,824	0.8
DeepMOT [65]	CVPR2020	42.4	53.8	53.7	42.7	42.5	1,947	4.9
Tracktor++ [1]	ICCV2019	44.8	55.1	56.3	45.1	44.9	1,987	1.5
TubeTK [37]	CVPR2020	48.0	58.6	63.0	45.1	51.4	4,137	3.0
ArTIST [45]	CVPR2021	48.9	59.7	62.3	48.3	50.0	2,062	4.5
MPNTrack [6]	CVPR2020	49.0	61.7	58.8	51.1	47.3	1,185	6.5
CenterTrack [77]	ECCV2020	52.2	64.7	67.8	51.0	53.8	3,039	3.8
TransTrack [50]	arxiv2021	54.1	63.5	75.2	47.9	61.6	3,603	59.2
TransCenter [64]	arxiv2021	54.5	62.2	73.2	49.7	60.1	4,614	1.0
GSDT [59]	ICRA2021	55.5	68.7	66.2	54.8	56.4	3,318	4.9
PermaTrack [54]	ICCV2021	55.5	68.9	73.8	53.1	58.5	3,699	11.9
MAT [19]	NC2022	56.0	69.2	67.1	57.2	55.1	1,279	11.5
CSTrack [30]	arxiv2020	59.3	72.6	74.9	57.9	61.1	3,567	15.8
FairMOT [74]	IJCV2021	59.3	72.3	73.7	58.0	60.9	3,303	25.9
ReMOT [67]	IVC2021	59.7	72.0	77.0	57.1	62.8	2,853	1.8
CrowdTrack [48]	AVSS2021	60.3	73.6	75.6	59.3	61.5	2,544	140.8
CorrTracker [57]	CVPR2021	60.7	73.6	76.5	58.9	62.9	3,369	15.6
RelationTrack [68]	arxiv2021	61.0	74.7	73.8	61.5	60.6	1,374	8.5
TransMOT [9]	arxiv2021	61.7	75.1	76.7	59.9	63.7	2,346	1.1
GRTU [58]	ICCV2021	62.0	75.0	74.9	62.1	62.1	1,812	3.6
MAATrack [49]	WACVw2022	62.0	75.9	79.4	60.2	64.2	1,452	189.1
ByteTrack [73]	arxiv2021	63.1	77.3	80.3	62.0	64.5	2,196	29.6
DeepSORT* [62]	ICIP2017	61.2	74.5	78.0	59.7	63.1	1,821	13.8
StrongSORT	ours	63.5	78.5	78.3	63.7	63.6	1,446	7.5
StrongSORT+	ours	63.7	79.0	78.3	64.1	63.6	1,401	7.4
StrongSORT++	ours	64.4	79.5	79.6	64.4	64.6	1,194	7.1

Table 5. Comparison with state-of-the-art MOT methods on the MOT20 test set. “*” represents our reproduced version. The two best results for each metric are bolded and highlighted in red and blue.

Method	Ref.	HOTA(↑)	IDF1(↑)	MOTA(↑)	AssA(↑)	DetA(↑)	IDs(↓)	FPS(↑)
SORT [3]	ICIP2016	36.1	45.1	42.7	35.9	36.7	4,470	57.3
ArTIST [45]	CVPR2021	41.6	51.0	53.6	40.2	43.3	1,531	1.0
Tracktor++ [1]	ICCV2019	42.1	52.7	52.6	42.0	42.3	1,648	1.2
TransCenter [64]	arxiv2021	43.6	49.9	59.1	37.0	51.8	4,597	1.0
ApLift [22]	ICCV2021	46.6	56.5	58.9	45.2	48.2	2,241	0.4
MPNTrack [6]	CVPR2020	46.8	59.1	57.6	47.3	46.6	1,210	6.5
LPC [10]	CVPR2021	49.0	62.5	56.3	52.4	45.8	1,562	0.7
GSDT [59]	ICRA2021	53.6	67.5	67.1	52.7	54.7	3,131	0.9
CSTrack [30]	arxiv2020	54.0	68.6	66.6	54.0	54.2	3,196	4.5
FairMOT [74]	IJCV2021	54.6	67.3	61.8	54.7	54.7	5,243	13.2
CrowdTrack [48]	AVSS2021	55.0	68.2	70.7	52.6	57.7	3,198	9.5
RelationTrack [68]	arxiv2021	56.5	70.5	67.2	56.4	56.8	4,243	4.3
MAATrack [49]	WACVw2022	57.3	71.2	73.9	55.1	59.7	1,331	14.7
SOTMOT [76]	CVPR2021	57.4	71.4	68.6	57.3	57.7	4,209	224.0
ReMOT [67]	IVC2021	61.2	73.1	77.4	58.7	63.9	1,789	0.4
ByteTrack [73]	arxiv2021	61.3	75.2	77.8	59.6	63.4	1,223	17.5
DeepSORT* [62]	ICIP2017	57.1	69.6	71.8	55.5	59.0	1,418	3.2
StrongSORT	ours	61.5	75.9	72.2	63.2	59.9	1,066	1.5
StrongSORT+	ours	61.6	76.3	72.2	63.6	59.9	1,045	1.5
StrongSORT++	ours	62.6	77.0	73.8	64.0	61.3	770	1.4



Fig. 5. Sample tracking results visualization of StrongSORT++ on the test sets of MOT17 and MOT20. The same box color represents the same ID.

5.4 MOTChallenge Results

We compare StrongSORT, StrongSORT+ (StrongSORT+AFLink) and StrongSORT++ (StrongSORT+AFLink+GSI) with state-of-the-art trackers on the test sets of MOT17 and MOT20, as shown in Tables 4 and 5, respectively. Notably, comparing FPS with absolute fairness is difficult because the speed claimed by each method depends on the devices where they are implemented, and the time spent on detections is generally excluded for tracking-by-detection trackers.

MOT17. StrongSORT++ ranks first among all published methods on MOT17 for metrics HOTA, IDF1, AssA, DetA, and ranks second for MOTA, IDs. In particular, it yields an accurate association and outperforms the second-performance tracker by a large margin (i.e., +2.2 IDF1 and +2.4 AssA). We use the same hyperparameters as in the ablation study and do not carefully tune them for each sequence like in [73]. The steady improvements on the test set prove the robustness of our methods. It is worth noting that, our reproduced version of DeepSORT (with a stronger detector and several tuned hyperparameters) also performs well on the benchmark, which demonstrates the effectiveness of the DeepSORT-like tracking paradigm.

MOT20. MOT20 is from more crowded scenarios. High occlusion means a high risk of missing detections and associations. StrongSORT++ still ranks first for metrics HOTA, IDF1 and AssA. It achieves significantly less IDs than the other trackers. Note that we use exactly the same hyperparameters as in MOT17, which implies the generalization capability of our method. Its detection performance (MOTA and Deta) is slightly poor compared with that of several trackers. We think this is because we use the same detection score threshold as in MOT17, which results in many missing detections. Specifically, the metric FN (number of false negatives) of our StrongSORT++ is 117,920, whereas that of ByteTrack [73] is only 87,594.

Qualitative Results. Figure 5 visualizes several tracking results of Strong-SORT++ on the test sets of MOT17 and MOT20. The results of MOT17-01 show the effectiveness of our method in normal scenarios. From the results of MOT17-08, we can see correct associations after occlusion. The results of MOT17-14 prove that our method can work well while the camera is moving. Moreover, the results of MOT20-04 show the excellent performance of Strong-SORT++ in scenarios with severe occlusion.

5.5 Limitations

StrongSORT and StrongSORT++ still have several limitations. The main concern is their relatively low running speed compared with joint trackers and several appearance-free separate trackers. Further research on improving computational efficiency is necessary. Moreover, although our method ranks first in metrics IDF1 and HOTA, it has a slightly lower MOTA, which is mainly caused by many missing detections due to the high threshold of the detection score. We believe an elaborate threshold strategy or association algorithm would help. As for AFLink, although it performs well in restoring missing associations, it is helpless against false association problems. Specifically, AFLink cannot split ID mixed-up trajectories into accurate tracklets. Future work is needed to develop stronger and more flexible global link strategies.

6 Conclusion

In this paper, we revisit the classic tracker DeepSORT and improve it in various aspects. The resulting StrongSORT achieves new SOTA on MOT17 and MOT20 benchmarks and demonstrates the effectiveness of the DeepSORT-like paradigm. We also propose two lightweight and appearance-free algorithms to further refine the tracking results. Experiments show that they can be applied to and benefit various state-of-the-art trackers with a negligible extra computational cost. Our final method, StrongSORT++, ranks first on MOT17 and MOT20 in terms of HOTA and IDF1 metrics and surpasses the second-place one by 1.3 - 2.2. Notably, our method runs relatively slow compared with joint trackers. In the future, we will investigate further for an improved time-accuracy trade-off.

References

1. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 941–951 (2019)
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing **2008**, 1–10 (2008)
3. 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)
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. IEEE (2017)
5. Bochinski, E., Senst, T., Sikora, T.: Extending iou based multi-object tracking by visual information. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6. IEEE (2018)
6. Brasó, G., Leal-Taixé, L.: Learning a neural solver for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6247–6257 (2020)
7. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE international conference on multimedia and expo (ICME). pp. 1–6. IEEE (2018)
8. Chu, P., Ling, H.: Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6172–6181 (2019)
9. Chu, P., Wang, J., You, Q., Ling, H., Liu, Z.: Transmot: Spatial-temporal graph transformer for multiple object tracking. arXiv preprint arXiv:2104.00194 (2021)
10. Dai, P., Weng, R., Choi, W., Zhang, C., He, Z., Ding, W.: Learning a proposal classifier for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2443–2452 (2021)
11. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 (2020)
12. Du, Y., Wan, J., Zhao, Y., Zhang, B., Tong, Z., Dong, J.: Giaotacker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2809–2819 (2021)
13. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: A mobile vision system for robust multi-person tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
14. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE transactions on pattern analysis and machine intelligence **30**(10), 1858–1865 (2008)
15. Feng, W., Hu, Z., Wu, W., Yan, J., Ouyang, W.: Multi-object tracking with multiple cues and switcher-aware classification. arXiv preprint arXiv:1901.06129 (2019)
16. Gao, J., Nevatia, R.: Revisiting temporal modeling for video-based person reid. arXiv preprint arXiv:1805.02104 (2018)
17. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)

18. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 315–323. JMLR Workshop and Conference Proceedings (2011)
19. Han, S., Huang, P., Wang, H., Yu, E., Liu, D., Pan, X.: Mat: Motion-aware multi-object tracking. Neurocomputing (2022)
20. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE transactions on pattern analysis and machine intelligence **37**(3), 583–596 (2014)
21. Hofmann, M., Haag, M., Rigoll, G.: Unified hierarchical multi-object tracking using global data association. In: 2013 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS). pp. 22–28. IEEE (2013)
22. Hornakova, A., Kaiser, T., Swoboda, P., Rolinek, M., Rosenhahn, B., Henschel, R.: Making higher order mot scalable: An efficient approximate solver for lifted disjoint paths. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6330–6340 (2021)
23. Huang, Y., Zhu, F., Zeng, Z., Qiu, X., Shen, Y., Wu, J.: Sqe: a self quality evaluation metric for parameters optimization in multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8306–8314 (2020)
24. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
25. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: Automatic detection of tracking failures. In: 2010 20th international conference on pattern recognition. pp. 2756–2759. IEEE (2010)
26. Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering **82D**, 35–45 (1960)
27. Khurana, T., Dave, A., Ramanan, D.: Detecting invisible people. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3174–3184 (2021)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
30. Liang, C., Zhang, Z., Lu, Y., Zhou, X., Li, B., Ye, X., Zou, J.: Rethinking the competition between detection and reid in multi-object tracking. arXiv preprint arXiv:2010.12138 (2020)
31. 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: European conference on computer vision. pp. 740–755. Springer (2014)
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**(2), 548–578 (2021)
34. Luo, H., Jiang, W., Gu, Y., Liu, F., Liao, X., Lai, S., Gu, J.: A strong baseline and batch normalization neck for deep person re-identification. IEEE Transactions on Multimedia **22**(10), 2597–2609 (2019)
35. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)

36. Naiel, M.A., Ahmad, M.O., Swamy, M., Lim, J., Yang, M.H.: Online multi-object tracking via robust collaborative model and sample selection. *Computer Vision and Image Understanding* **154**, 94–107 (2017)
37. Pang, B., Li, Y., Zhang, Y., Li, M., Lu, C.: Tubek: Adopting tubes to track multi-object in a one-step training model. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6308–6318 (2020)
38. Peng, J., Wang, C., Wan, F., Wu, Y., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In: European conference on computer vision. pp. 145–161. Springer (2020)
39. Peng, J., Wang, T., Lin, W., Wang, J., See, J., Wen, S., Ding, E.: Tpm: Multiple object tracking with tracklet-plane matching. *Pattern Recognition* **107**, 107480 (2020)
40. Perera, A.A., Srinivas, C., Hoogs, A., Brooksby, G., Hu, W.: Multi-object tracking through simultaneous long occlusions and split-merge conditions. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06). vol. 1, pp. 666–673. IEEE (2006)
41. Possegger, H., Mauthner, T., Roth, P.M., Bischof, H.: Occlusion geodesics for online multi-object tracking. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1306–1313 (2014)
42. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
43. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **28** (2015)
44. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European conference on computer vision. pp. 17–35. Springer (2016)
45. Saleh, F., Aliakbarian, S., Rezatofighi, H., Salzmann, M., Gould, S.: Probabilistic tracklet scoring and inpainting for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14329–14339 (2021)
46. 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)
47. Shuai, B., Berneshawi, A., Li, X., Modolo, D., Tighe, J.: Siammot: Siamese multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12372–12382 (2021)
48. Stadler, D., Beyerer, J.: On the performance of crowd-specific detectors in multi-pedestrian tracking. In: 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–12. IEEE (2021)
49. Stadler, D., Beyerer, J.: Modelling ambiguous assignments for multi-person tracking in crowds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 133–142 (2022)
50. 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)
51. Sun, S., Akhtar, N., Song, H., Mian, A., Shah, M.: Deep affinity network for multiple object tracking. *IEEE transactions on pattern analysis and machine intelligence* **43**(1), 104–119 (2019)

52. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
53. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)
54. Tokmakov, P., Li, J., Burgard, W., Gaidon, A.: Learning to track with object permanence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10860–10869 (2021)
55. Wang, B., Wang, G., Chan, K.L., Wang, L.: Tracklet association by online target-specific metric learning and coherent dynamics estimation. *IEEE transactions on pattern analysis and machine intelligence* **39**(3), 589–602 (2016)
56. Wang, G., Wang, Y., Zhang, H., Gu, R., Hwang, J.N.: Exploit the connectivity: Multi-object tracking with trackletnet. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 482–490 (2019)
57. Wang, Q., Zheng, Y., Pan, P., Xu, Y.: Multiple object tracking with correlation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3876–3886 (2021)
58. Wang, S., Sheng, H., Zhang, Y., Wu, Y., Xiong, Z.: A general recurrent tracking framework without real data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13219–13228 (2021)
59. Wang, Y., Kitani, K., Weng, X.: Joint object detection and multi-object tracking with graph neural networks. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13708–13715. IEEE (2021)
60. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: European Conference on Computer Vision. pp. 107–122. Springer (2020)
61. Williams, C., Rasmussen, C.: Gaussian processes for regression. *Advances in neural information processing systems* **8** (1995)
62. 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)
63. Wu, J., Cao, J., Song, L., Wang, Y., Yang, M., Yuan, J.: Track to detect and segment: An online multi-object tracker. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12352–12361 (2021)
64. Xu, Y., Ban, Y., Delorme, G., Gan, C., Rus, D., Alameda-Pineda, X.: Transcenter: Transformers with dense queries for multiple-object tracking. arXiv preprint arXiv:2103.15145 (2021)
65. Xu, Y., Osep, A., Ban, Y., Horaud, R., Leal-Taixé, L., Alameda-Pineda, X.: How to train your deep multi-object tracker. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6787–6796 (2020)
66. 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)
67. Yang, F., Chang, X., Sakti, S., Wu, Y., Nakamura, S.: Remot: A model-agnostic refinement for multiple object tracking. *Image and Vision Computing* **106**, 104091 (2021)
68. Yu, E., Li, Z., Han, S., Wang, H.: Relationtrack: Relation-aware multiple object tracking with decoupled representation. arXiv preprint arXiv:2105.04322 (2021)

69. Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J.: Poi: Multiple object tracking with high performance detection and appearance feature. In: European Conference on Computer Vision. pp. 36–42. Springer (2016)
70. Zeng, F., Dong, B., Wang, T., Zhang, X., Wei, Y.: Motr: End-to-end multiple-object tracking with transformer. arXiv preprint arXiv:2105.03247 (2021)
71. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 (2020)
72. Zhang, S., Benenson, R., Schiele, B.: Citypersons: A diverse dataset for pedestrian detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3221 (2017)
73. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864 (2021)
74. 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)
75. 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: European conference on computer vision. pp. 868–884. Springer (2016)
76. Zheng, L., Tang, M., Chen, Y., Zhu, G., Wang, J., Lu, H.: Improving multiple object tracking with single object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2453–2462 (2021)
77. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: European Conference on Computer Vision. pp. 474–490. Springer (2020)
78. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
79. Zhu, Y., Zhou, K., Wang, M., Zhao, Y., Zhao, Z.: A comprehensive solution for detecting events in complex surveillance videos. Multimedia Tools and Applications **78**(1), 817–838 (2019)