StrongSORT: Make DeepSORT Great Again

Yunhao Du, Zhicheng Zhao, Yang Song
Yanyun Zhao, Fei Su, Tao Gong, Hongying Meng

***Abstract*—Recently, multi-object tracking (MOT) has attracted increasing attention, and accordingly, remarkable progress has been achieved. However, the existing methods tend to use various basic models (e.g., detector and embedding models) and different training or inference tricks. As a result, the construction of a good baseline for a fair comparison is essential. In this paper, a classic tracker, i.e., DeepSORT, is first revisited, and then is significantly improved from multiple perspectives such as object detection, feature embedding, and trajectory association. The proposed tracker, named StrongSORT, contributes a strong and fair baseline to the MOT community. Moreover, two lightweight and plug-and-play algorithms are proposed to address two inherent "missing" problems of MOT: missing association and missing detection. Specifically, unlike most methods, which associate short tracklets into complete trajectories at high computational complexity, we propose an appearance-free link model (AFLink) to perform global association without appearance information, and achieve a good balance between speed and accuracy. Furthermore, we propose Gaussian-smoothed interpolation (GSI) based on Gaussian process regression to relieve missing detection. AFLink and GSI can be easily plugged into various trackers with a negligible extra computational cost (1.7 ms and 7.1 ms per image, respectively, on MOT17). Finally, by fusing StrongSORT with AFLink and GSI, the final tracker (StrongSORT++) achieves state-of-the-art results on multiple public benchmarks, i.e., MOT17, MOT20, DanceTrack and KITTI. Codes are available at https://github.com/dyhBUPT/StrongSORT and https://github.com/open-mmlab/mmtracking.**

***Index Terms*—Multi-Object Tracking, Baseline, AFLink, GSI.**

I. Introduction

MULTI-OBJECT TRACKING (MOT) aims to detect and track all specific classes of objects frame by frame, which plays an essential role in video understanding. In the past few years, the MOT task has been dominated by the tracking-by-detection (TBD) paradigm [60, 3, 55, 4, 32], which performs per frame detection and formulates the MOT problem as a data association task. TBD methods tend to extract appearance and/or motion embeddings first and then perform bipartite graph matching. Benefiting from high-performing object detection models, TBD methods have gained favour due to their excellent performance.

Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao and Fei Su are with Beijing Key Laboratory of Network System and Network Culture, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. (e-mail:{dyh_bupt, zhaozc, sy12138, zyy, sufei}@bupt.edu.cn)

Tao Gong is currently a Young Researcher at Shanghai AI Laboratory. He received the Ph.D. degree in School of Cyber Science and Technology from the University of Science and Technology of China in 2021. His research interests span computer vision and deep learning. (e-mail:gongtao@pjlab.org.cn)

Hongying Meng is with the College of Engineering, Design, and Physical Sciences, Brunel University London, Uxbridge, United Kingdom. (e-mail:hongying.meng@brunel.ac.uk)

As MOT is a downstream task corresponding to object detection and object re-identification (ReID), recent works tend to use various detectors and ReID models to increase MOT performance [18, 39], which makes it difficult to construct a fair comparison between them. Another problem preventing fair comparison is the usage of various external datasets for training [64, 63]. Moreover, some training and inference tricks are also used to improve the tracking performance.

To solve the above problems, this paper presents a simple but effective MOT baseline called StrongSORT. We revisit the classic TBD tracker DeepSORT [55], which is among the earliest methods that apply a deep learning model to the MOT task. We choose DeepSORT because of its simplicity, expansibility and effectiveness. It is claimed that DeepSORT underperforms compared with state-of-the-art methods because of its outdated techniques, rather than its tracking paradigm. To be specific, we first equip DeepSORT with a strong detector [18] following [63] and embedding model [30]. Then, we collect some inference tricks from recent works to further improve its performance. Simply equipping DeepSORT with these advanced components results in the proposed *StrongSORT*, and it is shown that it can achieve SOTA results on the popular benchmarks MOT17 [31] and MOT20 [9].

The motivations of StrongSORT can be summarized as follows:

- It can serve as a baseline for fair comparison between different tracking methods, especially for tracking-by-detection trackers.
- Compared to weak baselines, a stronger baseline can better demonstrate the effectiveness of methods.
- The elaborately collected inference tricks can be applied on other trackers without the need to retrain the model. This can benefit some tasks in academia and industry.

There are two "missing" problems in the MOT task, i.e., missing association and missing detection. Missing association means the same object is spread in more than one tracklet. This problem is particularly common in online trackers because they lack global information in association. Missing detection, also known as false negatives, refers to recognizing the object as background, which is usually caused by occlusion and low resolutions.

First, for the missing association problem, several methods propose to associate short tracklets into trajectories using a global link model [11, 47, 50, 35, 58]. They usually first generate accurate but incomplete tracklets and then associate them with global information in an offline manner. Although these methods improve tracking performance significantly, they rely on computation-intensive models, especially appearance embeddings. In contrast, we propose an appearance-free link model (AFLink), which only utilizes spatiotemporal

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2023.3240881

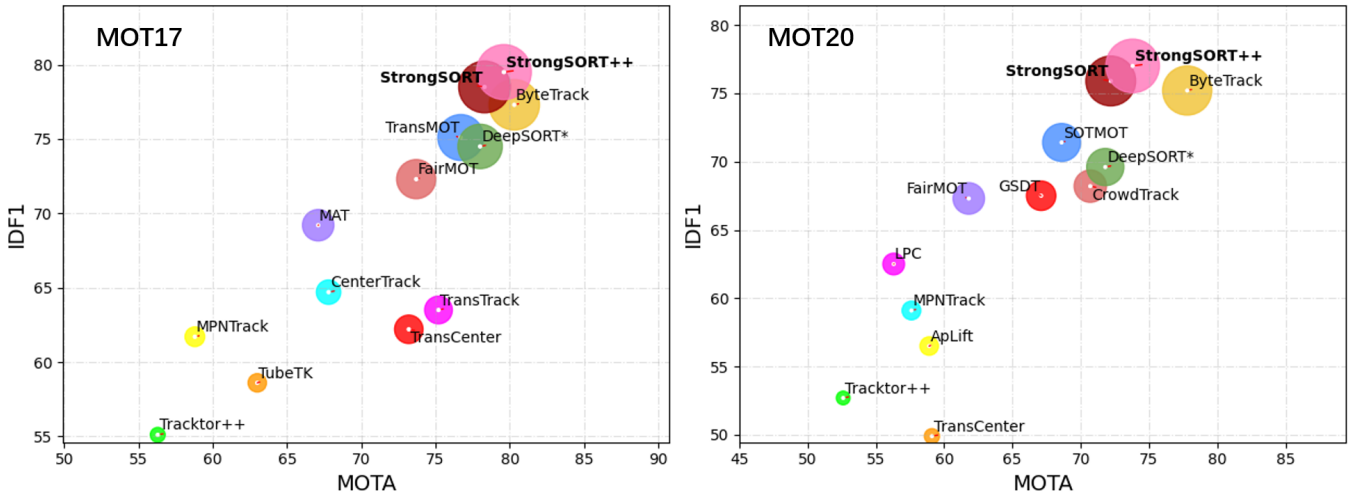

Fig. 1: IDF1-MOTA-HOTA comparisons of state-of-the-art trackers with our proposed StrongSORT and StrongSORT++ on MOT17 and MOT20 test sets. The horizontal axis is MOTA, the vertical axis is IDF1, and the radius of the circle is HOTA. "*" represents our reproduced version. Our StrongSORT++ achieves the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

information to predict whether the two input tracklets belong to the same ID. Without the appearance model, AFLink achieves a better trade-off between speed and accuracy.

Second, linear interpolation is widely used to compensate for missing detections [36, 22, 33, 37, 63, 11]. However, it ignores motion information during interpolation, which limits the accuracy of the interpolated positions. To solve this problem, we propose the Gaussian-smoothed interpolation algorithm (GSI), which fixes the interpolated bounding boxes using the Gaussian process regression algorithm [54]. GSI is also a kind of detection noise filter that can produce more accurate and stable localizations.

AFLink and GSI are both lightweight, plug-and-play, model-independent and appearance-free models, which are beneficial and suitable for this study. Extensive experiments demonstrate that they can create notable improvements in StrongSORT and other state-of-the-art trackers, e.g., CenterTrack [66], TransTrack [45] and FairMOT [64], with running speeds of 1.7 ms and 7.1 ms per image, respectively, on MOT17. In particular, by applying AFLink and GSI to StrongSORT, we obtain a stronger tracker called StrongSORT++. It achieves SOTA results on various benchmarks, i.e., MOT17, MOT20, DanceTrack [44] and KITTI [19]. Figure 1 presents the IDF1-MOTA-HOTA comparisons of state-of-the-art trackers with our proposed StrongSORT and StrongSORT++ on the MOT17 and MOT20 test sets.

The contributions of our work are summarized as follows:

- We propose StrongSORT, which equips DeepSORT with advanced modules (i.e., detector and embedding model) and some inference tricks. It can serve as a strong and fair baseline for other MOT methods, which is valuable to both academia and industry.
- We propose two novel and lightweight algorithms, AFLink and GSI, which can be plugged into various trackers to improve their performance with a negligible computational cost.
- Extensive experiments are designed to demonstrate the effectiveness of the proposed methods. Furthermore, the proposed StrongSORT and StrongSORT++ achieve SOTA performance on multiple benchmarks, including MOT17, MOT20, DanceTrack and KITTI.

II. Related Work

A. Separate and Joint Trackers

MOT methods can be classified into separate and joint trackers. Separate trackers [60, 3, 55, 4, 32, 21] follow the tracking-by-detection paradigm, which localizes targets first and then associates them with information on appearance, motion, etc. Benefiting from the rapid development of object detection [39, 38, 18], separate trackers have been widely applied in MOT tasks. Recently, several joint tracking methods [57, 59, 28, 51] have been proposed to jointly train detection and other components, such as motion, embedding and association models. The main advantages of these trackers are low computational cost and comparable performance.

Meanwhile, several recent studies [42, 43, 63, 7] 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 [31, 9]. However, abandoning appearance features would lead to poor robustness in more complex scenes. In this paper, we adopt the DeepSORT-like [55] paradigm and equip it with advanced techniques from various aspects to confirm the effectiveness of this classic framework.

B. Global Link in MOT

Missing association is an essential problem in MOT tasks. To exploit rich global information, several methods refine the tracking results with a global link model [11, 47, 50, 35, 58]. They first generate accurate but incomplete tracklets using

spatiotemporal and/or appearance information. Then, these tracklets are linked by exploring global information in an offline manner. TNT [50] is designed with a multiscale TrackletNet to measure the connectivity between two tracklets. It encodes motion and appearance information in a unified network using multiscale convolution kernels. TPM [35] is presented with 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 [58] splits imperfect trajectories into tracklets and then merges them with appearance features. GIAOTracker [11] proposes a complex global link algorithm that encodes tracklet appearance features using an improved ResNet50-TP model [16] and associates tracklets together with spatial and temporal costs. Although these methods yield notable improvements, they rely on appearance features, which bring high computational cost. In contrast, the proposed AFLink model exploits only 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.

AFLink shares similar motivations with LGMTracker [48], which also associates tracklets with motion information. LGMTracker is designed with an interesting but complex reconstruct-to-embed strategy to perform tracklet association based on GCN and TGC modules, which aims to solve the problem of latent space dissimilarity. However, AFLink shows that by carefully designing the framework and training strategy, a much simpler and more lightweight module can still work well. Particularly, AFlink takes only 10+ seconds for training and 10 seconds for testing on MOT17.

C. Interpolation in MOT

Linear interpolation is widely used to fill the gaps in recovered trajectories for missing detections [36, 22, 33, 37, 63, 11]. 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 spatiotemporal information effectively. V-IOUTracker [5] extends IOUTracker [4] by falling back to single-object tracking while missing detection occurs. MAT [20] smooths linearly interpolated trajectories nonlinearly by adopting a cyclic pseudo-observation trajectory filling strategy. An extra camera motion compensation (CMC) model [13] and a Kalman filter [24] are needed to predict missing positions. MAATrack [43] simplifies it by applying only the CMC model. All these methods apply extra models, i.e., a single-object tracker, CMC, and a Kalman filter, in exchange for performance gains. Instead, we propose modeling nonlinear motion on the basis of the Gaussian process regression (GPR) algorithm [54]. Without additional time-consuming components, our proposed GSI algorithm achieves a good trade-off between accuracy and efficiency.

The most similar work to our GSI is [67], 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. In contrast, we study the MOT task and adopt GPR to refine the interpolated localizations. Moreover, we present an adaptive smoothness factor instead of presetting a hyperparameter as done in [67].

III. StrongSORT

In this section, we present various approaches to upgrade DeepSORT [55] to StrongSORT. Specifically, we review DeepSORT in Section A and introduce StrongSORT in Section B. 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 present a strong MOT baseline.

A. Review of DeepSORT

We briefly summarize DeepSORT as a two-branch framework, that is, with an *appearance branch* and a *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 [65], 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 B_i of the i-th tracklet and the feature f_j of the j-th detection is computed as

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

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

In the motion branch, the Kalman filter algorithm [24] accounts for predicting the positions of tracklets in the current frame. It works by a two-phase process, i.e., state prediction and state update. In the state prediction step, it predicts the current state as:

$$\hat{x}'_k = F_k \hat{x}_{k-1}, \tag{2}$$

$$P'_k = F_k P_{k-1} F_k^T + Q_k, \tag{3}$$

where $\hat{x}_{k-1}$ and P_{k-1} are the mean and covariance of the state at time step $k-1$, $\hat{x}'_k$ and P'_k are the estimated states at time step k, F_k is the state transition model, and Q_k is the covariance of the process noise. In the state update step, the Kalman gain is calculated based on the covariance of the estimated state P'_k and the observation noise R_k as:

$$K = P'_k H_k^T (H_k P'_k H_k^T + R_k)^{-1}, \tag{4}$$

where H_k^T is the observation model, which maps the state from the estimation space to the observation space. Then, the Kalman gain K is used to update the final state:

$$x_k = \hat{x}'_k + K(z_k - H_k \hat{x}'_k), \tag{5}$$

$$P_k = (I - K H_k) P'_k, \tag{6}$$

where z_k is the measurement at time step k. Given the motion state of tracklets and new-coming detections, Mahalanobis distance is used to measure the spatiotemporal dissimilarity

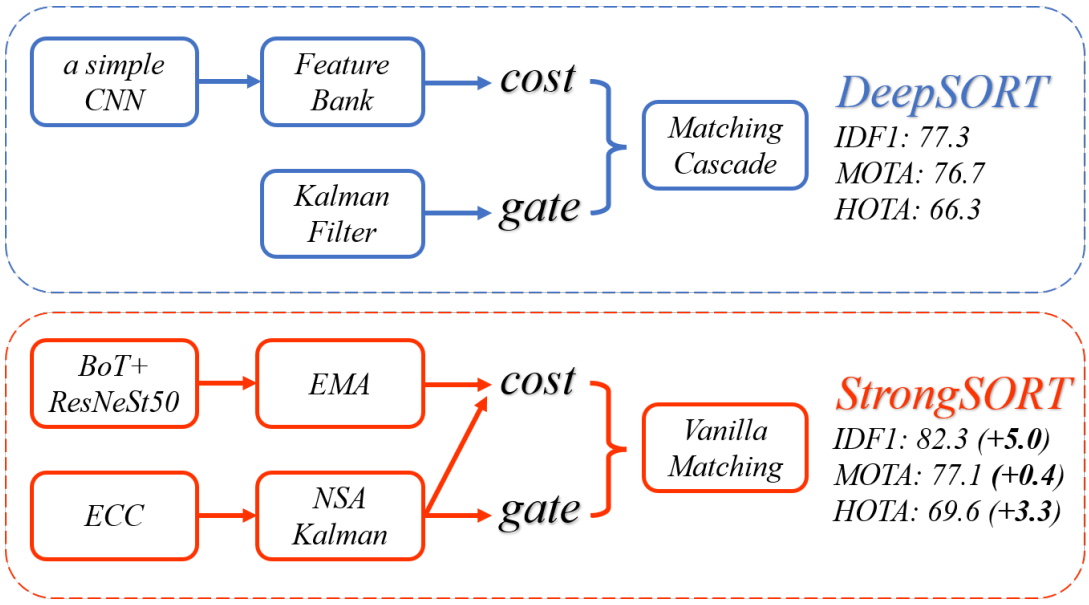

Fig. 2: Framework and performance comparison between DeepSORT and StrongSORT. Performance is evaluated on the MOT17 validation set based on detections predicted by YOLOX [18].

between them. 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 [27].

B. StrongSORT

Our improvements over DeepSORT include advanced modules and some inference tricks, as shown in the bottom half of Figure 2.

Advanced modules. DeepSORT uses the optimized Faster R-CNN [39] presented in [60] as the detector and trains a simple CNN as the embedding model. Instead, we replace the detector with YOLOX-X [18] following [63], which is not presented in Figure 2 for clarity. In addition, a stronger appearance feature extractor, BoT [30], is applied to replace the original simple CNN, which can extract much more discriminative features.

EMA. Although the feature bank mechanism in DeepSORT can preserve long-term information, it is sensitive to detection noise [11]. To solve this problem, we replace the feature bank mechanism with the feature updating strategy proposed in [52], which updates the appearance state e_i^t for the i-th tracklet at frame t in an exponential moving average (EMA) manner as follows:

$$e_i^t = \alpha e_i^{t-1} + (1-\alpha) f_i^t, \tag{7}$$

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 leverages the information of inter-frame feature changes and can depress detection noise. Experiments show that it not only enhances the matching quality but also reduces the time consumption.

ECC. Camera movements exist in multiple benchmarks [31, 44, 19]. Similar to [20, 43, 25, 21], we adopt the enhanced correlation coefficient maximization (ECC) [13] model for camera motion compensation. It is a technique for parametric image alignment that can estimate the global rotation and translation between adjacent frames. Specifically, it is based on the following criterion to quantify the performance of the warping transformation:

$$E_{ECC}(\mathbf{p}) = \left\| \frac{\bar{\mathbf{i}}_\mathbf{r}}{\|\bar{\mathbf{i}}_\mathbf{r}\|} - \frac{\bar{\mathbf{i}}_\mathbf{w}(\mathbf{p})}{\|\bar{\mathbf{i}}_\mathbf{w}(\mathbf{p})\|} \right\|^2, \tag{8}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\mathbf{p}$ is the warping parameter, and $\bar{\mathbf{i}}_\mathbf{r}$ and $\bar{\mathbf{i}}_\mathbf{w}(\mathbf{p})$ are the zero-mean versions of the reference (template) image $\mathbf{i}_\mathbf{r}$ and warped image $\mathbf{i}_\mathbf{w}(\mathbf{p})$. Then, the image alignment problem is solved by minimizing $E_{ECC}(\mathbf{p})$, with the proposed forward additive iterative algorithm or inverse compositional iterative algorithm. Due to its efficiency and effectiveness, ECC is widely used to compensate for the motion noise caused by camera movement in MOT tasks.

NSA Kalman. The vanilla Kalman filter is vulnerable w.r.t. low-quality detections [43] and ignores the information on scales of detection noise [11]. To solve this problem, we borrow the NSA Kalman algorithm from GIAOTracker [11], which proposes a formula to adaptively calculate the noise covariance $\widetilde{R}_k$:

$$\widetilde{R}_k = (1-c_k) R_k, \tag{9}$$

where R_k is the preset constant measurement noise covariance and c_k is the detection confidence score at state k. Intuitively, the detection has a higher score c_k when it has less noise, which results in a low $\widetilde{R}_k$. According to formulas 4-6, a lower $\widetilde{R}_k$ means that the detection will have a higher weight in the state update step, and vice versa. This can help improve the accuracy of updated states.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2023.3240881

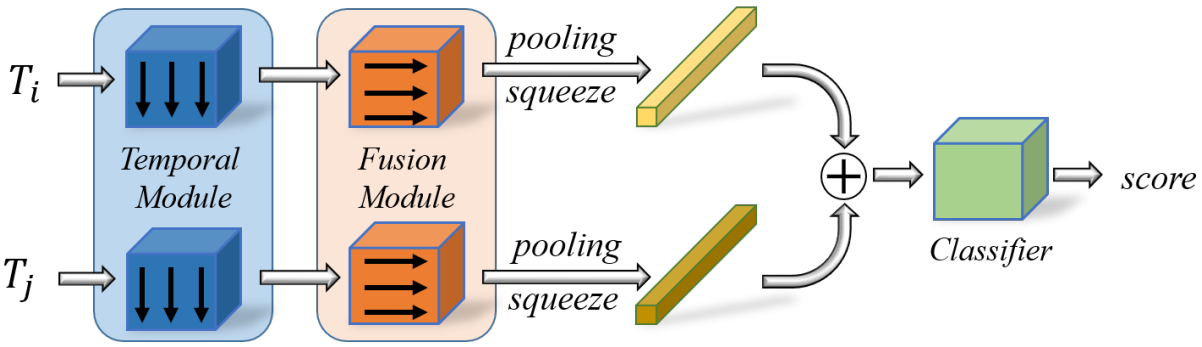

Fig. 3: Framework of the two-branch AFLink model. It adopts two tracklets T_i and T_j as input, where $T_* = \{f_k^*, x_k^*, y_k^*\}_{k=k^*}^{k^*+N-1}$ consists of the frame id f_k^* and positions (x_k^*, y_k^*) of the recent $N = 30$ frames. Then, the temporal module extracts features along the temporal dimension with 7×1 convolutions and the fusion module integrates information along the feature dimension with 1×3 convolutions. These two tracklet features are pooled, squeezed and concatenated, and then input into a classifier to predict the association score.

Motion Cost. DeepSORT only employs the appearance feature distance as a matching cost during the first association stage, in which the motion distance is only used as the gate. Instead, we solve the assignment problem with both appearance and motion information, similar to [52, 64]. The cost matrix C is a weighted sum of appearance cost A_a and motion cost A_m as follows:

$$C = \lambda A_a + (1 - \lambda) A_m, \tag{10}$$

where the weight factor λ is set to 0.98, as in [52, 64].

Vanilla Matching. An 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 confusing associations. Therefore, additional prior constraints limit the matching accuracy. We solve this problem by simply replacing the matching cascade with vanilla global linear assignment.

IV. StrongSORT++

We present a strong baseline in Section III. In this section, we introduce two lightweight, plug-and-play, model-independent, appearance-free algorithms, namely, AFLink and GSI, to further solve the problems of missing association and missing detection. We call the final method StrongSORT++, which integrates StrongSORT with these two algorithms.

A. 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 have numerous hyperparameters to fine-tune. For example, the link algorithm in GIAOTracker [11] utilizes an improved ResNet50-TP [16] to extract tracklet 3D features and performs association with additional spatial and temporal distances. It has six hyperparameters to be set, i.e., three thresholds and three weight factors, which incurs heavy tuning experiments and poor robustness. Moreover, overreliance on appearance features can be vulnerable to occlusion. Motivated by this, we design an appearance-free model, AFLink, to predict the connectivity between two tracklets by relying only on spatiotemporal 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 $T_* = \{f_k^*, x_k^*, y_k^*\}_{k=k^*}^{k^*+N-1}$ consists of the frame id f_k^* and positions (x_k^*, y_k^*) of the most recent $N = 30$ frames. Zero padding is used for tracklets that is shorter than 30 frames. A temporal module is applied to extract features by convolving along the temporal dimension with 7×1 kernels, which consists of four "Conv-BN-ReLU" layers. Then, the fusion module, which is a single 1×3 convolution layer with BN and ReLU, is used 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 and then concatenated, which includes rich spatiotemporal information. Finally, an MLP is used to predict a confidence score for association. Note that the weights of the two branches in the temporal and fusion modules are not shared.

During training, the association procedure is formulated as a binary classification task. Then, it is optimized with the binary cross-entropy loss as follows:

$$\begin{aligned} L_n^{BCE} = -(y_n log(\frac{e^{x_n}}{e^{x_n} + e^{1-x_n}}) + \\ (1 - y_n) log(1 - \frac{e^{1-x_n}}{e^{x_n} + e^{1-x_n}})), \end{aligned} \tag{11}$$

where $x_n \in [0, 1]$ is the predicted probability of association for sample pair n, and $y_n \in \{0, 1\}$ is the ground truth.

During association, we filter out unreasonable tracklet pairs with spatiotemporal constraints. Then, the global link is solved as a linear assignment task [27] with the predicted connectivity score.

B. GSI

Interpolation is widely used to fill the gaps in trajectories caused by missing detections. Linear interpolation is popular due to its simplicity; however, its accuracy is limited because it does not use motion information. Although several strategies

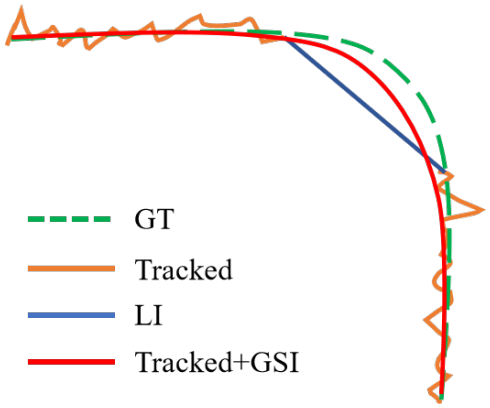

Fig. 4: Illustration of the difference between linear interpolation (LI) and the proposed Gaussian-smoothed interpolation (GSI).

have been proposed to solve this problem, they generally introduce additional time-consuming modules, e.g., a single-object tracker, a Kalman filter, and ECC. In contrast, we present a lightweight interpolation algorithm that employs Gaussian process regression [54] to model nonlinear motion.

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

$$p_t = f^{(i)}(t) + \epsilon, \tag{12}$$

where $t \in F$ is the frame id, $p_t \in P$ is the position coordinate variable 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:

$$f^{(i)} \in GP(0, k(\cdot, \cdot)), \tag{13}$$

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 a new frame set F^*, its smoothed position P^* is predicted by

$$P^* = K(F^*, F)(K(F, F) + \sigma^2 I)^{-1} P, \tag{14}$$

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 to its length. We simply design it as a function adaptive to length l as follows:

$$\lambda = \tau * log(\tau^3 / l), \tag{15}$$

where τ is set to 10 based on the ablation experiment.

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) solves both problems simultaneously by smoothing the entire trajectory with an adaptive smoothness factor.

V. Experiments

A. Setting

Datasets. We conduct experiments on the MOT17 [31] and MOT20 [9] datasets under the "private detection" protocol. MOT17 is a popular dataset for MOT, which consists of 7 sequences and 5,316 frames for training and 7 sequences and 5919 frames for testing. MOT20 is a dataset of highly crowded challenging scenes, with 4 sequences and 8,931 frames for training and 4 sequences and 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 [66, 63]. We use DukeMTMC [40] to pretrain our appearance feature extractor. We train the detector on the CrowdHuman dataset [41] and MOT17 half training set for ablation following [66, 63, 45, 56, 61]. We add Cityperson [62] and ETHZ [12] for testing as in [63, 52, 64, 28].

We also test StrongSORT++ on KITTI [19] and DacneTrack [44]. KITTI is a popular dataset related to autonomous driving tasks. It can be used for pedestrian and car tracking, which consists of 21 training sequences and 29 test sequences with a relatively low frame rate of 10 FPS. DanceTrack is a recently proposed dataset for multi-human tracking, which encourages more MOT algorithms that rely less on visual discrimination and depend more on motion analysis. It consists of 100 group dancing videos, where humans have similar appearances but diverse motion features.

Metrics. We use the metrics MOTA, IDs, IDF1, HOTA, AssA, DetA and FPS to evaluate tracking performance [2, 40, 29]. 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. 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), such as in MOTA and IDF1, and better takes localization accuracy into account.

Implementation Details. We present the default implementation details in this section. For detection, we adopt YOLOX-X [18] as our detector for an improved time-accuracy trade-off. The training schedule is similar to that in [63]. 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 matching distance threshold is 0.45, the warp mode for ECC is *MOTION EUCLIDEAN*, 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 and a ReLU activation layer. The fusion module includes a 1×3 convolution, a BN and a ReLU. It does not 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

TABLE I: 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(↑)	MOTA(↑)	HOTA(↑)	FPS(↑)
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

TABLE II: Results of applying AFLink and GSI to various MOT methods. All experiments are performed on the MOT17 validation set with a single GPU. (best in bold)

Method	AFLink	GSI	IDF1(↑)	MOTA(↑)	HOTA(↑)	FPS(↑)
StrongSORTv1	-	-	79.5	76.8	67.8	**8.3**
	✓		80.0	76.8	68.1	8.2
	✓	✓	**80.4(+0.9)**	**78.2(+1.4)**	**68.9(+1.1)**	7.8 (-0.5)
StrongSORTv3	-	-	79.7	77.1	68.3	**6.2**
	✓		80.5	77.1	68.6	6.1
	✓	✓	**80.9(+1.2)**	**78.7(+1.6)**	**69.5(+1.2)**	5.9 (-0.3)
StrongSORTv6	-	-	82.3	77.1	69.6	**7.5**
	✓		82.5	77.1	69.6	7.4
	✓	✓	**83.3(+1.0)**	**78.7(+1.6)**	**70.8(+1.2)**	7.0 (-0.5)
CenterTrack [66]	-	-	64.6	66.8	55.3	**14.4**
	✓		68.3	66.9	57.2	14.1
	✓	✓	**68.4(+3.8)**	**66.9(+0.1)**	**57.6(+2.3)**	12.8 (-1.6)
TransTrack [45]	-	-	68.6	67.7	58.1	**5.8**
	✓		69.1	67.7	58.3	5.8
	✓	✓	**69.9(+1.3)**	**69.6(1.9)**	**59.4(+1.3)**	5.6 (-0.2)
FairMOT [64]	-	-	72.7	69.1	57.3	**12.0**
	✓		73.2	69.2	57.6	11.8
	✓	✓	**74.2(+1.5)**	**71.1(+2.0)**	**59.0(+1.7)**	10.9 (-1.1)

are generated by cutting annotated trajectories into tracklets with random spatiotemporal noise at a 1:3 ratio of positive to negative samples. We use Adam as the optimizer [26] and 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.

B. Ablation Studies

Ablation study for StrongSORT. Table I summarizes the path from DeepSORT to StrongSORT:

1) BoT: Replacing the original feature extractor with BoT leads to a significant improvement for IDF1 (+2.2), indicating that association quality benefits from more discriminative appearance features.

2) ECC: The CMC model results in a slight increase in IDF1 (+0.2) and MOTA (+0.3), implying that it helps extract more precise motion information.

3) NSA: The NSA Kalman filter improves HOTA (+0.4) but not MOTA and IDF1. This means that it enhances positioning accuracy.

4) EMA: The EMA feature updating mechanism brings not only superior association (+0.4 IDF1) but also a faster speed (+1.2 FPS).

5) MC: Matching with both appearance and motion cost aids association (+0.8 IDF1).

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 (+1.4).

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 [66], TransTrack [45] and FairMOT [64]). Their results are shown in Table II. 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. In 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

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2023.3240881

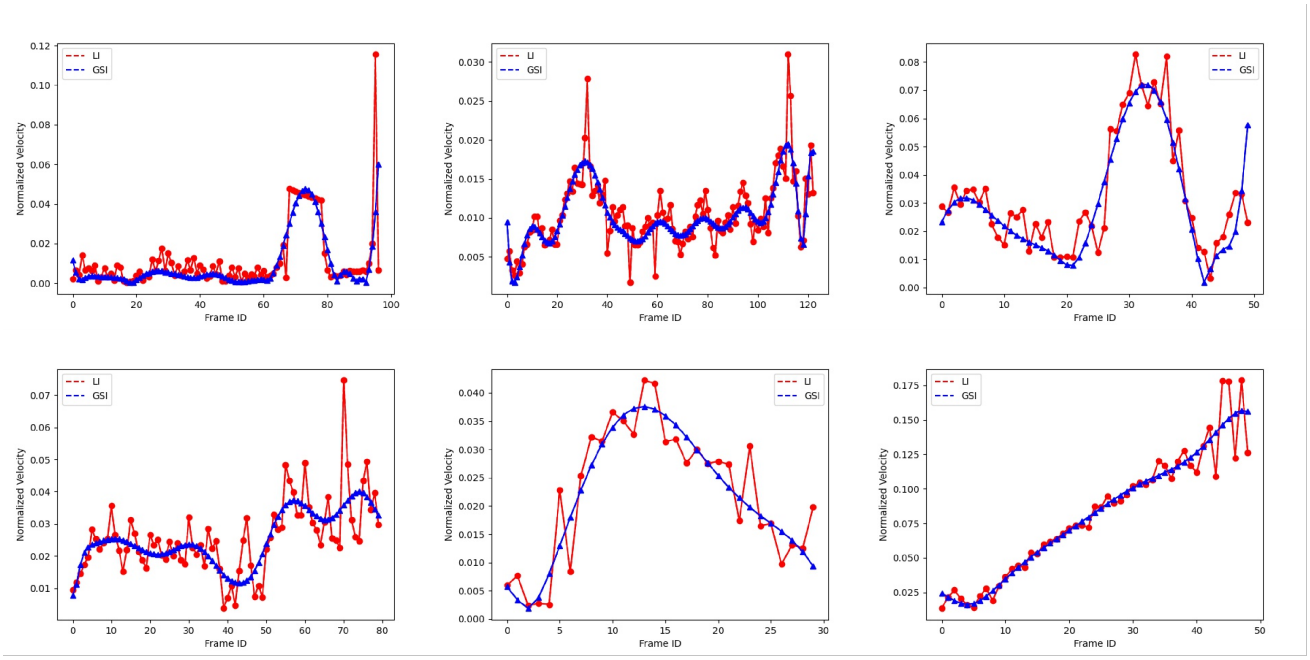

Fig. 5: Comparison of normalized velocity between the trajectories after applying linear interpolation (LI, in red) and Gaussian-smoothed interpolation (GSI, in blue). The x-coordinate represents the frame id, and the y-coordinate is the normalized velocity.

works better on stronger trackers, but it can be confused by the large amount of false association in poor trackers.

Ablation study for vanilla matching. We present the comparison between the matching cascade algorithm and vanilla matching on different baselines in Table III. It is shown that the matching cascade algorithm greatly benefits DeepSORT. However, with the gradual enhancement of the baseline tracker, it has increasingly smaller advantages and is even harmful to tracking accuracy. Specifically, for StrongSORTv5, it can bring a gain of 1.4 on IDF1 by replacing the matching cascade with vanilla matching. This leads us to the following interesting conclusion: *Although the priori assumption in the matching cascade can reduce confusing associations in poor trackers, this additional constraint will limit the performance of stronger trackers instead.*

Additional analysis of GSI. Speed estimation is essential for some downstream tasks, e.g., action analysis [10] and benefits the construction of intelligent transportation systems (ITSs) [14]. To measure the performance of different interpolation algorithms on the speed estimation task, we compare the normalized velocity between trajectories after applying linear interpolation (LI) and Gaussian-smoothed interpolation (GSI) in Figure 5. Specifically, six trajectories from DeepSORT on the MOT17 validation set are sampled. The x-coordinate and y-coordinate represent the frame id and normalized velocity, respectively. It is shown that the velocity of trajectories with LI jitters wildly (in red), mainly due to detection noise. Instead, trajectories with GSI have more stable velocity (in blue). This gives us another perspective to understand GSI: *GSI is a kind of detection noise filter that can produce more accurate and stable localizations.* This feature is beneficial to speed estimation and other related tasks.

TABLE III: Ablation study on the MOT17 validation set for the matching cascade algorithm and vanilla matching.

Method	Matching	IDF1(↑)	MOTA(↑)
DeepSORT	Cascade	77.3	76.7
	Vanilla	76.2 (**-1.1**)	76.7 (**-0.0**)
StrongSORTv1	Cascade	79.5	76.8
	Vanilla	79.6 (**+0.1**)	76.7 (**-0.1**)
StrongSORTv2	Cascade	79.7	77.1
	Vanilla	79.7 (**+0.0**)	77.1 (**+0.0**)
StrongSORTv3	Cascade	79.7	77.1
	Vanilla	79.9 (**+0.2**)	77.1 (**+0.0**)
StrongSORTv4	Cascade	80.1	77.0
	Vanilla	81.9 (**+1.8**)	76.9 (**-0.1**)
StrongSORTv5	Cascade	80.9	77.0
	Vanilla	82.3 (**+1.4**)	77.1 (**+0.1**)

C. Main Results

We compare StrongSORT, StrongSORT+ (StrongSORT + AFLink) and StrongSORT++ (StrongSORT + AFLink + GSI) with state-of-the-art trackers on the test sets of MOT17, MOT20, DanceTrack and KITTI, as shown in Tables IV, V, VI and VII, respectively. Notably, comparing FPS fairly 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 on MOT17 for metrics HOTA, IDF1, AssA, and DetA and ranks second for MOTA and IDs. In particular, it yields an accurate association and outperforms the second-performance tracker by a large margin (i.e., +2.1 IDF1 and +2.1 AssA). We use the same hyperparameters as in the ablation study and do not carefully tune them for each sequence as in [63]. The steady improvements on the test

TABLE IV: Comparison with state-of-the-art MOT methods on the MOT17 test set. "*" represents our reproduced version. "(w/o LI)" means abandoning the offline linear interpolation procedure. The two best results for each metric are bolded and highlighted in red and blue.

mode	Method	Ref.	HOTA(↑)	IDF1(↑)	MOTA(↑)	AssA(↑)	DetA(↑)	IDs(↓)	FPS(↑)
online	SORT [3]	ICIP2016	34.0	39.8	43.1	31.8	37.0	4,852	**143.3**
	MTDF [15]	TMM2019	37.7	45.2	49.6	34.5	42.0	5,567	1.2
	DeepMOT [57]	CVPR2020	42.4	53.8	53.7	42.7	42.5	1,947	4.9
	ISEHDADH [8]	TMM2019	-	-	54.5	-	-	3,010	3.6
	Tracktor++ [1]	ICCV2019	44.8	55.1	56.3	45.1	44.9	1,987	1.5
	TubeTK [33]	CVPR2020	48.0	58.6	63.0	45.1	51.4	4,137	3.0
	CRF-MOT [17]	TMM2022	-	60.4	58.9	-	-	2,544	-
	CenterTrack [66]	ECCV2020	52.2	64.7	67.8	51.0	53.8	3,039	3.8
	TransTrack [45]	arxiv2020	54.1	63.5	75.2	47.9	61.6	3,603	59.2
	PermaTrack [46]	ICCV2021	55.5	68.9	73.8	53.1	58.5	3,699	11.9
	CSTrack [28]	TIP2022	59.3	72.6	74.9	57.9	61.1	3,567	15.8
	FairMOT [64]	IJCV2021	59.3	72.3	73.7	58.0	60.9	3,303	25.9
	CrowdTrack [42]	AVSS2021	60.3	73.6	75.6	59.3	61.5	2,544	**140.8**
	CorrTracker [51]	CVPR2021	60.7	73.6	76.5	58.9	62.9	3,369	15.6
	RelationTrack [59]	TMM2022	61.0	74.7	73.8	61.5	60.6	**1,374**	8.5
	OC-SORT* (w/o LI) [7]	arxiv2022	61.7	76.2	76.0	62.0	61.6	2,199	29.0
	ByteTrack* (w/o LI) [63]	ECCV2022	**62.8**	**77.2**	**78.9**	**62.2**	**63.8**	2,310	29.6
	DeepSORT* [55]	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
offline	TPM [35]	PR2020	41.5	52.6	54.2	40.9	42.5	1,824	0.8
	MPNTrack [6]	CVPR2020	49.0	61.7	58.8	51.1	47.3	**1,185**	6.5
	TBooster [49]	TMM2022	50.5	63.3	61.5	52.0	49.2	2,478	6.9
	MAT [20]	NC2022	56.0	69.2	67.1	57.2	55.1	1,279	11.5
	ReMOT [58]	IVC2021	59.7	72.0	77.0	57.1	62.8	2,853	1.8
	MAATrack [43]	WACVw2022	62.0	75.9	79.4	60.2	64.2	1,452	**189.1**
	OC-SORT [7]	arxiv2022	63.2	77.5	78.0	63.4	63.2	1,950	29.0
	ByteTrack* [63]	ECCV2022	63.2	77.4	**79.7**	62.3	**64.4**	2,253	**29.6**
	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 V: Comparison with state-of-the-art MOT methods on the MOT20 test set. "*" represents our reproduced version. "(w/o LI)" means abandoning the offline linear interpolation procedure. The two best results for each metric are bolded and highlighted in red and blue.

mode	Method	Ref.	HOTA(↑)	IDF1(↑)	MOTA(↑)	AssA(↑)	DetA(↑)	IDs(↓)	FPS(↑)
online	SORT [3]	ICIP2016	36.1	45.1	42.7	35.9	36.7	4,470	**57.3**
	Tracktor++ [1]	ICCV2019	42.1	52.7	52.6	42.0	42.3	1,648	1.2
	CSTrack [28]	TIP2022	54.0	68.6	66.6	54.0	54.2	3,196	4.5
	FairMOT [64]	IJCV2021	54.6	67.3	61.8	54.7	54.7	5,243	13.2
	CrowdTrack [42]	AVSS2021	55.0	68.2	70.7	52.6	57.7	3,198	9.5
	RelationTrack [59]	TMM2022	56.5	70.5	67.2	56.4	56.8	4,243	4.3
	OC-SORT* (w/o LI) [7]	arxiv2022	60.5	74.4	**73.1**	**60.8**	**60.5**	**1,307**	-
	ByteTrack* (w/o LI) [63]	ECCV2022	**60.9**	**74.9**	**75.7**	59.9	**62.0**	1,347	**17.5**
	DeepSORT* [55]	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
offline	TBooster [49]	TMM2022	42.5	53.4	54.6	41.4	43.8	1,674	0.1
	MPNTrack [6]	CVPR2020	46.8	59.1	57.6	47.3	46.6	1,210	6.5
	MAATrack [43]	WACVw2022	57.3	71.2	73.9	55.1	59.7	1,331	**14.7**
	ReMOT [58]	IVC2021	61.2	73.1	**77.4**	58.7	**63.9**	1,789	0.4
	OC-SORT [7]	arxiv2022	**62.1**	75.9	75.5	-	-	**913**	-
	ByteTrack* [63]	ECCV2022	61.2	75.1	**76.5**	60.0	**62.6**	1,120	**17.5**
	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

TABLE VI: Comparison with state-of-the-art MOT methods on the DanceTrack test set. The two best results for each metric are bolded and highlighted in red and blue.

Method	Ref.	HOTA(↑)	IDF1(↑)	MOTA(↑)	AssA(↑)	DetA(↑)
CenterTrack [66]	ECCV2020	41.8	35.7	86.8	22.6	78.1
FairMOT [64]	IJCV2021	39.7	40.8	82.2	23.8	66.7
TransTrack [45]	arxiv2020	45.5	45.2	88.4	27.5	75.9
TraDes [56]	CVPR2021	43.3	41.2	86.2	25.4	74.5
ByteTrack [63]	ECCV2022	47.7	53.9	**89.6**	32.1	71.0
MOTR [61]	ECCV2022	54.2	51.5	79.7	**40.2**	73.5
OC-SORT [7]	arxiv2022	**55.1**	**54.2**	89.4	38.0	**80.3**
StrongSORT++	ours	**55.6**	**55.2**	**91.1**	**38.6**	**80.7**

TABLE VII: Comparison with state-of-the-art MOT methods on the KITTI test set. The two best results for each metric are bolded and highlighted in red and blue.

		Car				Pedestrian			
Method	Ref.	HOTA(↑)	MOTA(↑)	AssA(↑)	IDs(↓)	HOTA(↑)	MOTA(↑)	AssA(↑)	IDs(↓)
AB3D [53]	IROS2020	69.99	83.61	69.33	**113**	37.81	38.13	44.33	**181**
MPNTrack [6]	CVPR2020	-	-	-	-	45.26	46.23	47.28	397
CenterTrack [66]	ECCV2020	73.02	88.83	71.20	254	40.35	53.84	36.93	425
QD-3DT [23]	TPAMI2022	72.77	85.94	72.19	**206**	41.08	51.77	38.82	717
QDTrack [34]	CVPR2021	68.45	84.93	65.49	313	41.12	55.55	38.10	487
LGMTracker [48]	ICCV2021	73.14	87.60	72.31	448	-	-	-	-
PermaTrack [46]	ICCV2021	**77.42**	**90.85**	**77.66**	275	47.43	65.05	43.66	483
OC-SORT [7]	arxiv2022	76.54	90.28	76.39	250	**54.69**	**65.14**	**59.08**	204
StrongSORT++	ours	**77.75**	**90.35**	**78.20**	440	**54.48**	**67.38**	**57.31**	**178**

set prove the robustness of our methods. It is worth noting that our reproduced version of DeepSORT (with a stronger detector YOLOX and several tuned hyperparameters) also performs well on the benchmark, which demonstrates the effectiveness of the DeepSORT-like tracking paradigm.

MOT20. The data in MOT20 is taken from more crowded scenarios. High occlusion means a high risk of missing detections and associations. StrongSORT++ still ranks first for the metrics HOTA, IDF1 and AssA. It achieves significantly fewer IDs than 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 to 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 [63] is only 87,594.

DanceTrack. Our StrongSORT++ also achieves the best results on the DanceTrack benchmark for most metrics. Because this dataset focuses less attention on appearance features, we abandon the appearance-related optimizations here, i.e., BoT and EMA. The NMS threshold is set as 0.7, the matching distance is 0.3, the AFLink prediction threshold is 0.9, and the GSI interpolation threshold is 5 frames. For fair comparison, we use the same detections with ByteTrack [63] and achieve much better results, which demonstrates the superiority of our method.

KITTI. On the KITTI dataset, we use the same detection results as PermaTrack [46] and OC-SORT [7] for fair comparison. The results show that StrongSORT++ achieves comparable results for cars and superior performance for pedestrians compared to PermaTrack. For simplicity, we only apply two tricks (i.e., ECC and NSA Kalman) and two proposed algorithms (i.e., AFLink and GSI) here.

D. Qualitative Results.

Figure 6 visualizes several tracking results of StrongSORT++ on the test sets of MOT17, MOT20, DanceTrack and KITTI. 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 show that our method can work well while the camera is moving. Moreover, the results of MOT20-04 show the excellent performance of StrongSORT++ in scenarios with severe occlusion. The results of DanceTrack and KITTI demonstrate the effectiveness of StrongSORT++ while facing the problems of complex motion patterns and low frame rates.

E. Limitations

StrongSORT and StrongSORT++ still have several limitations. One concern is their relatively low running speed compared to joint trackers and several appearance-free separate trackers. This problem is mainly caused by the DeepSORT-like paradigm, which requires an extra detector and appearance model, and the proposed AFLink and GSI are both lightweight algorithms. Moreover, although our method performs well on the IDF1 and HOTA metrics, it has a slightly lower MOTA on MOT17 and MOT20, 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. For AFLink, although it performs

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2023.3240881

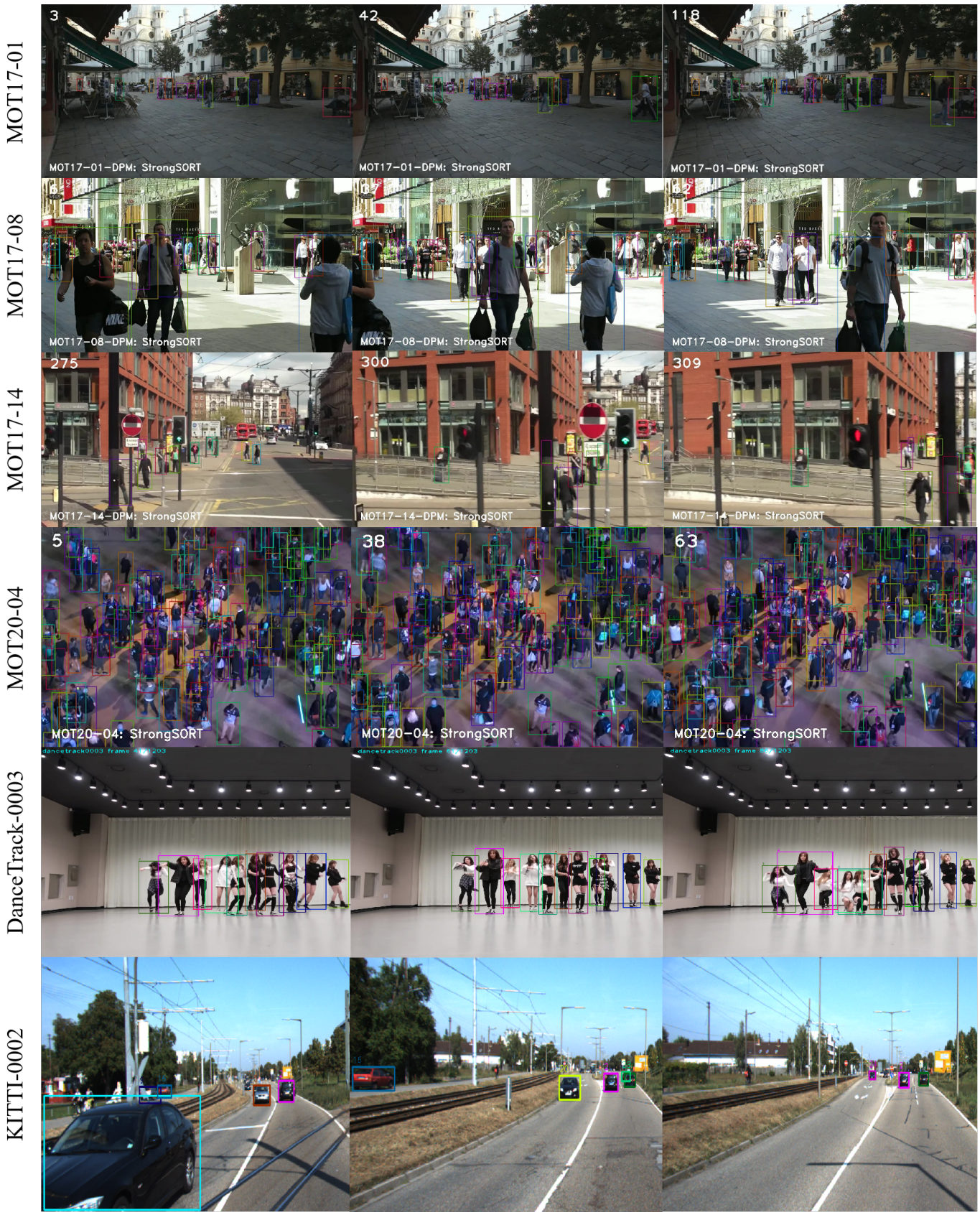

Fig. 6: Sample tracking results visualization of StrongSORT++ on the test sets of MOT17, MOT20, DanceTrack and KITTI. The box color corresponds to the ID.

well in restoring missing associations, it is helpless against false association problems. Specifically, AFLink cannot split mixed-up ID trajectories into accurate tracklets. Future work is needed to develop stronger and more flexible global link strategies.

VI. Conclusion

In this paper, we revisit the classic tracker DeepSORT and upgrade it with new modules and several inference tricks. The resulting new tracker, StrongSORT, can serve as a new strong baseline for the MOT task.

We also propose two lightweight and appearance-free algorithms, AFLink and GSI, to solve the missing association and missing detection problems. Experiments show that they can be applied to and benefit various state-of-the-art trackers with a negligible extra computational cost.

By integrating StrongSORT with AFLink and GSI, the resulting tracker StrongSORT++ achieves state-of-the-art results on multiple benchmarks, i.e., MOT17, MOT20, DanceTrack and KITTI.

Acknowledgments

This work is supported by Chinese National Natural Science Foundation under Grants (62076033, U1931202) and BUPT Excellent Ph.D. Students Foundation (CX2022145).

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) 9

[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) 6

[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) 1, 2, 9

[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) 1, 2, 3

[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) 3

[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) 9, 10

[7] 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) 2, 9, 10

[8] Dai, P., Wang, X., Zhang, W., Chen, J.: Instance segmentation enabled hybrid data association and discriminative hashing for online multi-object tracking. IEEE Transactions on Multimedia **21**(7), 1709–1723 (2018) 9

[9] 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) 1, 2, 6

[10] Du, Y., Tong, Z., Wan, J., Zhang, B., Zhao, Y.: Pamiad: An activity detector exploiting part-attention and motion information in surveillance videos. arXiv preprint arXiv:2203.03796 (2022) 8

[11] Du, Y., Wan, J., Zhao, Y., Zhang, B., Tong, Z., Dong, J.: Giaotracker: 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) 1, 2, 3, 4, 5

[12] 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) 6

[13] 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) 3, 4

[14] Fernández Llorca, D., Hernández Martínez, A., García Daza, I.: Vision-based vehicle speed estimation: A survey. IET Intelligent Transport Systems **15**(8), 987–1005 (2021) 8

[15] Fu, Z., Angelini, F., Chambers, J., Naqvi, S.M.: Multi-level cooperative fusion of gm-phd filters for online multiple human tracking. IEEE Transactions on Multimedia **21**(9), 2277–2291 (2019) 9

[16] Gao, J., Nevatia, R.: Revisiting temporal modeling for video-based person reid. arXiv preprint arXiv:1805.02104 (2018) 3, 5

[17] Gao, T., Pan, H., Wang, Z., Gao, H.: A crf-based framework for tracklet inactivation in online multi-object tracking. IEEE Transactions on Multimedia **24**, 995–1007 (2021) 9

[18] Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021) 1, 2, 4, 6

[19] Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013) 2, 4, 6

[20] Han, S., Huang, P., Wang, H., Yu, E., Liu, D., Pan, X.: Mat: Motion-aware multi-object tracking. Neurocomputing (2022) 3, 4, 9

[21] He, J., Huang, Z., Wang, N., Zhang, Z.: Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5299–5309 (2021) 2, 4

[22] Hofmann, M., Haag, M., Rigoll, G.: Unified hierarchi-

cal 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) 2, 3

[23] Hu, H.N., Yang, Y.H., Fischer, T., Darrell, T., Yu, F., Sun, M.: Monocular quasi-dense 3d object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022) 10

[24] Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering **82D**, 35–45 (1960) 3

[25] Khurana, T., Dave, A., Ramanan, D.: Detecting invisible people. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3174–3184 (2021) 4

[26] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 7

[27] Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955) 4, 5

[28] Liang, C., Zhang, Z., Zhou, X., Li, B., Zhu, S., Hu, W.: Rethinking the competition between detection and reid in multiobject tracking. IEEE Transactions on Image Processing **31**, 3182–3196 (2022) 2, 6, 9

[29] 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) 6

[30] 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) 1, 4

[31] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016) 1, 2, 4, 6

[32] 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) 1, 2

[33] Pang, B., Li, Y., Zhang, Y., Li, M., Lu, C.: Tubetk: 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) 2, 3, 9

[34] 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) 10

[35] 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) 1, 2, 3, 9

[36] 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) 2, 3

[37] 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) 2, 3

[38] Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018) 2

[39] 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) 1, 2, 4

[40] 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) 6

[41] 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) 6

[42] 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) 2, 9

[43] 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) 2, 3, 4, 9

[44] 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) 2, 4, 6

[45] 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) 2, 6, 7, 9, 10

[46] 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) 9, 10

[47] 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) 1, 2

[48] Wang, G., Gu, R., Liu, Z., Hu, W., Song, M., Hwang, J.N.: Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9876–9886 (2021) 3, 10

[49] Wang, G., Wang, Y., Gu, R., Hu, W., Hwang, J.N.: Split and connect: A universal tracklet booster for multi-object tracking. IEEE Transactions on Multimedia (2022) 9

[50] 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) 1, 2, 3
[51] 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) 2, 9
[52] 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) 4, 5, 6
[53] Weng, X., Wang, J., Held, D., Kitani, K.: 3d multi-object tracking: A baseline and new evaluation metrics. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10359–10366. IEEE (2020) 10
[54] Williams, C., Rasmussen, C.: Gaussian processes for regression. Advances in neural information processing systems **8** (1995) 2, 3, 6
[55] 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) 1, 2, 3, 9
[56] 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) 6, 10
[57] 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) 2, 9
[58] 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) 1, 2, 3, 9
[59] Yu, E., Li, Z., Han, S., Wang, H.: Relationtrack: Relation-aware multiple object tracking with decoupled representation. IEEE Transactions on Multimedia (2022) 2, 9
[60] 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) 1, 2, 4
[61] 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) 6, 10
[62] 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) 6
[63] 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) 1, 2, 3, 4, 6, 8, 9, 10
[64] 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) 1, 2, 5, 6, 7, 9, 10
[65] 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) 3
[66] Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: European Conference on Computer Vision. pp. 474–490. Springer (2020) 2, 6, 7, 9, 10
[67] 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) 3