# StrongSORT: Make DeepSORT Great Again

Yunhao Du, Zhicheng Zhao, *Member, IEEE*, Yang Song, Yanyun Zhao, Fei Su, *Member, IEEE*, Tao Gong, and Hongying Meng, *Senior Member, IEEE*

***Abstract*—Recently, Multi-Object Tracking (MOT) has attracted rising attention, and accordingly, remarkable progresses have been achieved. However, the existing methods tend to use various basic models (e.g, detector and embedding model), and different training or inference tricks, etc. 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 for 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 computation 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 a Gaussian-smoothed interpolation (GSI) based on Gaussian process regression to relieve the 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 is dominated by the tracking-by-detection (TBD) paradigm [3], [4], [32], [55], [60], which performs detection per frame and formulates the MOT problem as a data association task. The TBD methods tend to extract appearance and/or motion embeddings first, then perform a bipartite graph matching. Benefiting from high-performing object detection models, TBD methods have gained favor due to their excellent performance.

Manuscript received 14 September 2022; revised 16 December 2022 and 19 January 2023; accepted 22 January 2023. Date of publication 31 January 2023; date of current version 12 December 2023. This work was supported in part by Chinese National Natural Science Foundation under Grants 62076033 and U1931202, and in part by BUPT Excellent Ph.D. Students Foundation under Grant CX2022145. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. W. Li. *(Corresponding author: Zhicheng Zhao.)*

Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, and Fei Su are with the Beijing Key Laboratory of Network System and Network Culture, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100088, China (e-mail: dyh_bupt@bupt.edu.cn; zhaozc@bupt.edu.cn; sy12138@bupt.edu.cn; zyy@bupt.edu.cn; sufei@bupt.edu.cn).

Tao Gong was with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230052, China. He is now with the Shanghai AI Laboratory, Shanghai, China (e-mail: gongtao@pjlab.org.cn).

Hongying Meng is with the College of Engineering, Design, and Physical Sciences, Brunel University London, UB8 3PH Uxbridge, U.K. (e-mail: hongying.meng@brunel.ac.uk).

Digital Object Identifier 10.1109/TMM.2023.3240881

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 performances [18], [39], which makes it difficult to construct a fair comparison between them. Another problem for a fair comparison is the usage of various external datasets for training [63], [64]. 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 deep learning model to the MOT task. We choose DeepSORT because of its simplicity, expansibility and effectiveness. It's 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 strong detector [18] following [63] and embedding model [30]. Then, we collect some inference tricks from recent works to further improve the performance. It's shown that by simply equipping DeepSORT with these advanced components, resulting in the proposed *StrongSORT*, it can achieve SOTA results on 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 one 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 works in academia and industry.

There exist two "missing" problems in MOT task, i.e., missing association and missing detection. Missing association means the same object is spread in more than one tracklets. 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

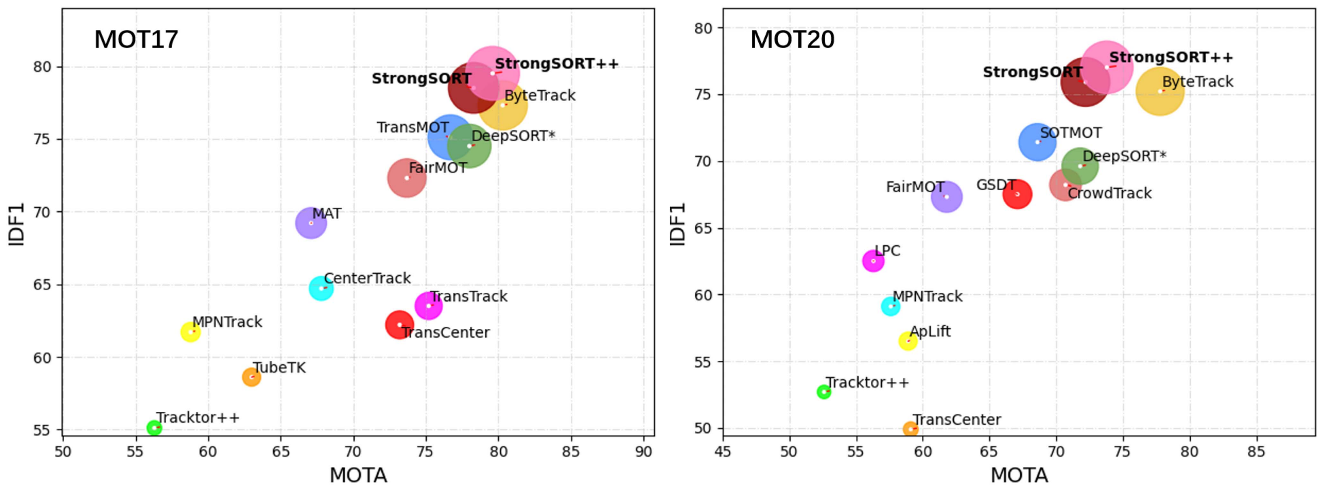

Fig. 1. IDF1-MOTA-HOTA comparisons of state-of-the-art trackers with our proposed StrongSORT and StrongSORT++ on MOT17 and MOT20 test sets. The horizontal axis is MOTA, the vertival axis is IDF1, and the radius of the circle is HOTA. "*" represents our reproduced version. Our StrongSORT++ achieves the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

background, which is usually caused by occlusion and low resolutions.

Firstly, for the missing association problem, several methods propose to associate short tracklets into trajectories using a global link model [11], [35], [47], [50], [58]. They usually firstly generate accurate but incomplete tracklets, 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. By contrast, we propose an appearance-free link model (AFLink), which only utilizes spatio-temporal 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.

Secondly, linear interpolation is widely used to compensate for missing detections [11], [22], [33], [36], [37], [63]. 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, which 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 achieve notable improvements on StrongSORT and other state-of-the-art trackers, e.g., CenterTrack [66], TransTrack [45] and FairMOT [64], with a running speed of 1.7 ms and 7.1 ms per image, respectively, on MOT17. Particularly, 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]. Fig. 1 presents the IDF1-MOTA-HOTA comparisons of state-of-the-art trackers with our proposed StrongSORT and StrongSORT++ on 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 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.
- Extended 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 [3], [4], [21], [32], [55], [60] follow the tracking-by-detection paradigm, which localizes targets first, then associates them with information on appearance, motion, etc. Benefiting from the rapid development of object detection [18], [38], [39], separate trackers have been widely applied in MOT task. Recently, several joint tracking methods [28], [51], [57], [59] have been proposed to jointly train detection and other components such as motion, embedding and association models. The main advantages of these trackers are the low computational cost and comparable performance.

Meanwhile, several recent studies [7], [42], [43], [63] 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 [9], [31]. 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 task. To exploit rich global information, several methods refine the tracking results with a global link model [11], [35], [47], [50], [58]. They firstly generate accurate but incomplete tracklets using spatio-temporal and/or appearance information. Then, these tracklets are linked by exploring global information in an offline manner. TNT [50] is designed with a multi-scale TrackletNet to measure the connectivity between two tracklets. It encodes motion and appearance information in a unified network using multi-scale 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, 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. Differently, the proposed AFLink model 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.

AFLink shares similar motivations with LGMTracker [48], which also associate tracklets with motion information. LGMTracker is designed with an interesting but complex reconstruct-to-embed strategy to perform tracklets association based on GCN and TGC module, 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 just takes 10+ seconds for training, 10 seconds for testing on MOT17.

## C. Interpolation in MOT

Linear interpolation is widely used to fill the gaps of recovered trajectories for missing detections [11], [22], [33], [36], [37], [63]. 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 while missing detection occurs. MAT [20] smooths the linearly interpolated trajectories nonlinearly by adopting a cyclic pseudo-observation trajectory filling strategy. An extra camera motion compensation (CMC) model [13] and 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., 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 [54]. 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 [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. 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 [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, *appearance branch* and *motion branch*, as shown in the top half of Fig. 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 state 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:

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

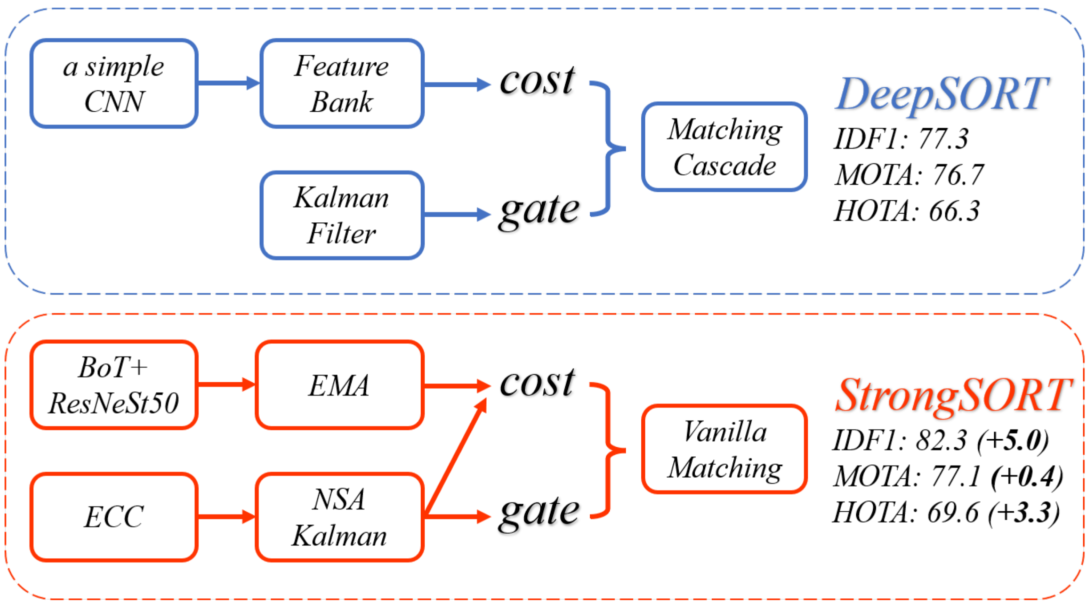

Fig. 2. Framework and performance comparison between DeepSORT and StrongSORT. Performance is evaluated on the MOT17 validation set based on detections predicted by YOLOX [18].

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

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

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

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 spatio-temporal dissimilarity 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 Fig. 2.

*Advanced modules:* DeepSORT uses the optimized Faster R-CNN [39] presented in [60] as the detector, and train a simple CNN as the embedding model. Instead, we replace the detector with YOLOX-X [18] following [63], which is not presented in Fig. 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:* Though the feature bank mechanism in DeepSORT can preserve the long-term information, it is sensitive to detection noises [11]. To solve this problem, we replace the feature bank mechanism with the feature updating strategy proposed in [52], which updates appearance state $e_i^t$ for the $i$-th tracklet at frame $t$ in an exponential moving average (EMA) manner as follows:

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

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 noises. Experiments show that it not only enhances the matching quality, but also reduces the time consumption.

*ECC:* There exist camera movements in multiple benchmarks [19], [31], [44]. Similar to [20], [21], [25], [43], we adopt the Enhanced Correlation Coefficient Maximization (ECC) [13] model for camera motion compensation. It is a technique for parametric image alignment, which can estimate the global rotation and translation between adjacent frames. To be specific, it's based on the following criterion to quatify 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 parameters, 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 iverse 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

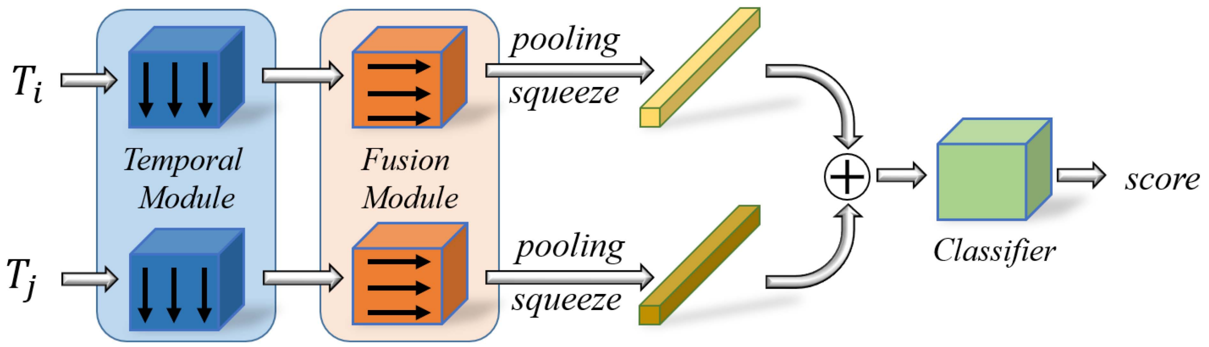

Fig. 3. Framework of the two-branch AFLink model. It adopts two tracklets $T_i$ and $T_j$ as input, where $T_* = \{f_k^*, x_k^*, y_k^*\}_{k=k^*}^{k^*+N-1}$ consists of the frames id $f_k^*$ and positions $(x_k^*, y_k^*)$ of the recent $N = 30$ frames. Then, the temporal module extracts features along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

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 the formula 4-6, a lower $\widetilde{R}_k$ means the detection will have a higher weight in the state update step, and vice versa. It can help improve the accuracy of updated states.

*Motion Cost:* DeepSORT only employs the appearance feature distance as 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]. 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 weight factor $\lambda$ 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 would limit the matching accuracy. We solve this problem by simply replacing 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 the 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 tracklets 3D features and performs association with additional spatial and temporal distances. It has six hyperparameters to be setted, i.e, three thresholds and three weight factors which incurs heavy tuning experiments and poor robustness. Moreover, over-reliance 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 spatio-temporal information.

Fig. 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 frames id $f_k^*$ 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 $7 \times 1$ kernels, which consists of four "Conv-BN-ReLU" layers. Then, the fusion module, which is a single $1 \times 3$ 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 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 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:

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

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 spatio-temporal constraints. Then, the global link is solved

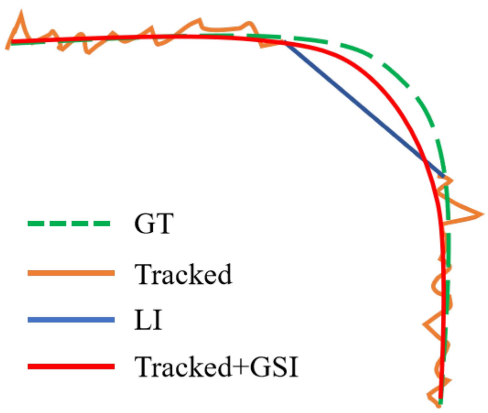

Fig. 4. Illustration of the difference between linear interpolation (LI) and the proposed Gaussian-smoothed interpolation (GSI).

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 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 [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 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:

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

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 new frame set $F^*$, its smoothed positions $P^*$ 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 $\lambda$ 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:

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

where $\tau$ is set to 10 based on the grid search experiment.

Fig. 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 MOT17 [31] and MOT20 [9] 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 [63], [66]. 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 [45], [56], [61], [63], [66]. We add Cityperson [62] and ETHZ [12] for testing as in [28], [52], [63], [64].

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 encorages 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 appearance but diverse motion features.

*Metrics:* We use the metrics MOTA, IDs, IDF1, HOTA, AssA, DetA and FPS to evaluate tracking performance [2], [29], [40]. 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) like 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 $\alpha$ in EMA is 0.9 and the weight factor for appearance cost $\lambda$ is 0.98. For GSI, the maximum gap allowed for interpolation is 20 frames, and hyperparameter $\tau$ is 10.

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)

| 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 |

Best in bold.

For AFLink, the temporal module consists of four convolution layers with $7 \times 1$ 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 $1 \times 3$ 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 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 [26], 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 it enhances positioning accuracy.
4) EMA: The EMA feature updating mechanism brings not only superior association (+0.4 IDF1), but also 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. 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.

*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 benefits DeepSORT greatly. However, with the gradual enhancement of the baseline tracker, it has smaller and smaller advantages, and be even harmful to tracking accuracy. Specifically, for StrongSORTv5, it can bring a gain of 1.4 on IDF1 by replacing matching cascade with vanilla matching. This leads us to the following interesting conclusion: *Though the priori assumption in 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 constructing Intelligent Transportation System (ITS) [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 Fig. 5. Specifically, six trajectories from DeepSORT on MOT17 validation set are sampled. The x-coordinate and y-coordinate represent frame id and normalized velocity respectively. It is shown that the velocity of trajectories with LI jitters wildly (in red), mainly caused by 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, which can produce more accurate and stable localizations.* This feature makes it beneficial to speed estimation and other related tasks.

TABLE II
RESULTS OF APPLYING AFLINK AND GSI TO VARIOUS MOT METHODS

| 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) |

All experiments are performed on the MOT17 validation set with a single GPU. (best in bold).

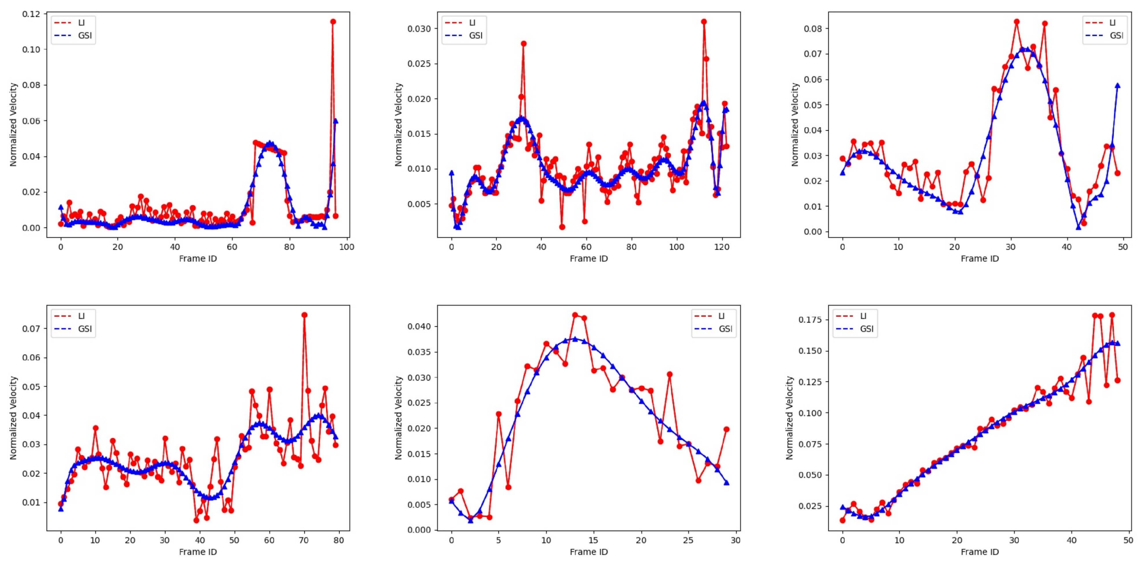

Fig. 5. Comparison of normalized velocity between the trajectories after applying linear interpolation (LI, in red) and Gaussian-smoothed interpolation (GSI, in blue). The x-coordinate represents the frame id, and the y-coordinate is the normalized velocity.

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, DetA, and ranks second for MOTA, IDs. In particular, it yields an accurate association and outperforms

TABLE IV
COMPARISON WITH STATE-OF-THE-ART MOT METHODS ON THE MOT17 TEST SET

| 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 |

"*" 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.

TABLE V
COMPARISON WITH STATE-OF-THE-ART MOT METHODS ON THE MOT20 TEST SET

| 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 |

"*" 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.

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 like in [63]. 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 YOLOX 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 other trackers. Note that we use exactly the same hyperparameters as in MOT17,

TABLE VI
COMPARISON WITH STATE-OF-THE-ART MOT METHODS ON THE DANCETRACK TEST SET

| **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** |

The two best results for each metric are bolded and highlighted in red and blue.

TABLE VII
COMPARISON WITH STATE-OF-THE-ART MOT METHODS ON THE KITTI TEST SET

| | | 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** |

The two best results for each metric are bolded and highlighted in red and blue.

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 beacuse 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 DanceTrack benchmark for all metrics. Because this dataset encorages 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 reuslts as PermaTrack [46] and OC-SORT [7] for fair comparison. Results show that StrongSORT++ achieves comparable results for car and superior performance for pedestrian 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.

Fig. 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 seperate trackers. This problem is mainly caused by the DeepSORT-like paradigm which needs extra detector and appearance model, and the proposed AFLink and GSI are both lightweight algorithms. Moreover, although our method performs well in metrics IDF1 and HOTA, it has a slightly lower MOTA on MOT17 and MOT20, which is mainly caused by many missing detections due to the high threshold of 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.

## VI. CONCLUSION

In this paper, we revisit the classic tracker DeepSORT and upgrade it with new modules and several inference tricks. The

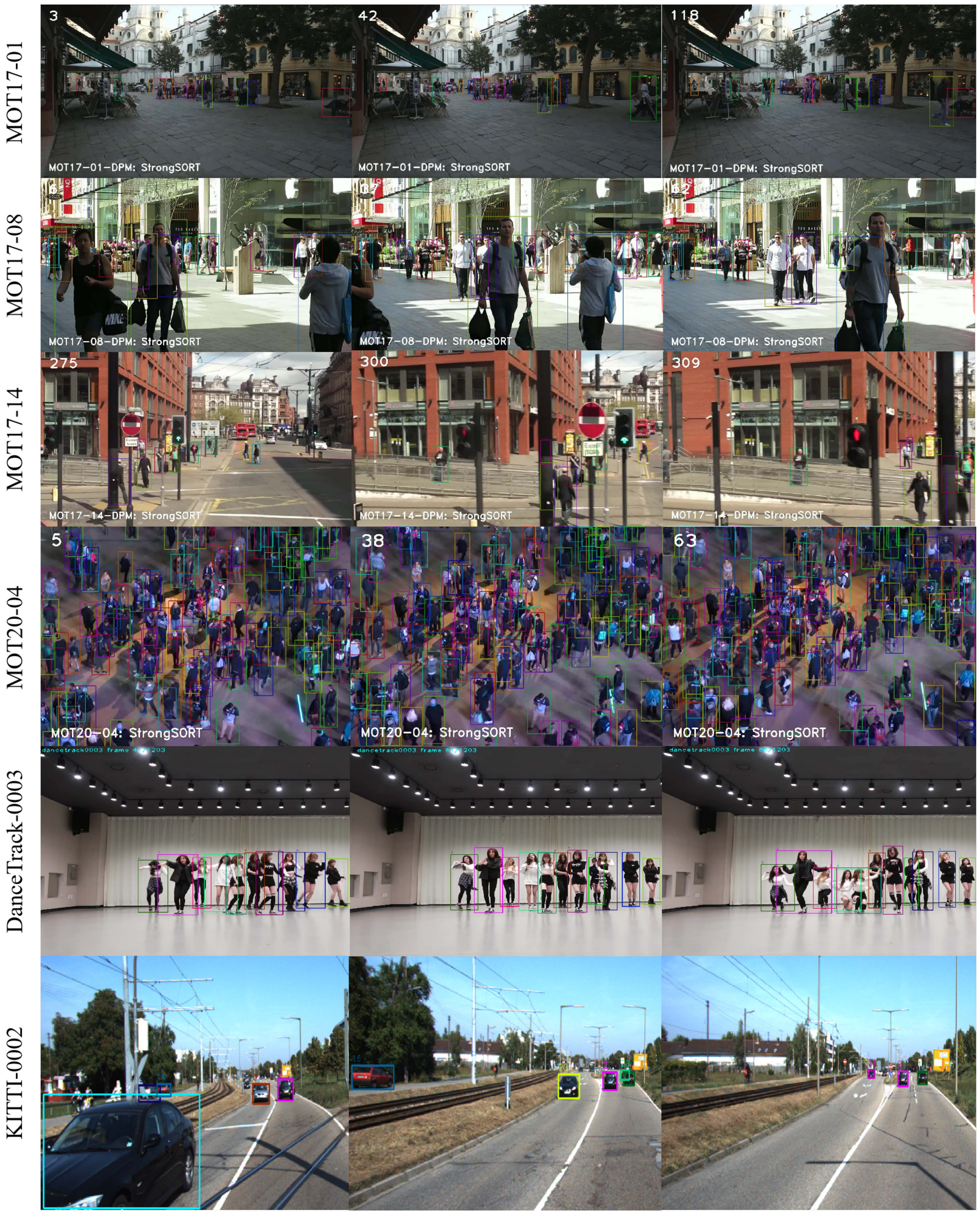

Fig. 6. Sample tracking results visualization of StrongSORT++ on the test sets of MOT17, MOT20, DanceTrack and KITTI. The same box color represents the same ID.

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.

## References

[1] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 941–951.

[2] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The clear mot metrics,” in *Proc. EURASIP J. Image Video Process.*, 2008, pp. 1–10.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 3464–3468.

[4] E. Bochinski, V. Eiselein, and T. Sikora, “High-speed tracking-by-detection without using image information,” in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2017, pp. 1–6.

[5] E. Bochinski, T. Senst, and T. Sikora, “Extending IOU based multi-object tracking by visual information,” in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2018, pp. 1–6.

[6] G. Brasó and L. Leal-Taixé, “Learning a neural solver for multiple object tracking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6247–6257.

[7] J. Cao, X. Weng, R. Khirodkar, J. Pang, and K. Kitani, “Observation-centric sort: Rethinking sort for robust multi-object tracking, 2022, *arXiv:2203.14360*.

[8] P. Dai, X. Wang, W. Zhang, and J. Chen, “Instance segmentation enabled hybrid data association and discriminative hashing for online multi-object tracking,” *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 1709–1723, Jul. 2019.

[9] P. Dendorfer et al., “Mot20: A benchmark for multi object tracking in crowded scenes, 2020, *arXiv:2003.09003*.

[10] Y. Du, Z. Tong, J. Wan, B. Zhang, and Y. Zhao, “PAMI-AD: An activity detector exploiting part-attention and motion information in surveillance videos, in *Proc. IEEE Int. Conf. Multimedia Expo. Workshops*, 2022, pp. 1–6.

[11] Y. Du et al., “Giaotracker: A comprehensive framework for MCMOT with global information and optimizing strategies in visdrone 2021,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2809–2819.

[12] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, “A mobile vision system for robust multi-person tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[13] G. D. Evangelidis and E. Z. Psarakis, “Parametric image alignment using enhanced correlation coefficient maximization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30 no. 10, pp. 1858–1865, Oct. 2008.

[14] D. Fernández Llorca, A. Hernández Martínez, and I. García Daza, “Vision-based vehicle speed estimation: A survey,” *IET Intell. Transport Syst.*, vol. 15, no. 8, pp. 987–1005, 2021.

[15] Z. Fu, F. Angelini, J. Chambers, and S.M. Naqvi, “Multi-level cooperative fusion of GM-PHD filters for online multiple human tracking,” *IEEE Trans. Multimedia*, vol. 21, pp. 2277–2291, 2019.

[16] J. Gao and R. Nevatia, “Revisiting temporal modeling for video-based person reid,” 2018, *arXiv:1805.02104*.

[17] T. Gao, H. Pan, Z. Wang, and H. Gao, “A CRF-based framework for tracklet inactivation in online multi-object tracking,” *IEEE Trans. Multimedia*, vol. 24, pp. 995–1007, 2022.

[18] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “Yolox: Exceeding yolo series in 2021,” 2021, *arXiv:2107.08430*.

[19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[20] S. Han et al., “Mat: Motion-aware multi-object tracking,” *Neurocomputing*, vol. 476, pp. 75–86, 2022.

[21] J. He, Z. Huang, N. Wang, and Z. Zhang, “Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5299–5309.

[22] M. Hofmann, M. Haag, and G. Rigoll, “Unified hierarchical multi-object tracking using global data association,” in *Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveill.*, 2013, pp. 22–28.

[23] H. N. Hu et al., “Monocular quasi-dense 3D object tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1992–2008, Feb. 2023.

[24] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Eng.*, vol. 82D, pp. 35–45, 1960.

[25] T. Khurana, A. Dave, and D. Ramanan, “Detecting invisible people,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3174–3184.

[26] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2014.

[27] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Nav. Res. Logistics Quart.*, vol. 2, no. 1/2, pp. 83–97, 1955.

[28] C. Liang et al., “Rethinking the competition between detection and ReID in multiobject tracking,” *IEEE Trans. Image Process.*, vol. 31, pp. 3182–3196, 2022.

[29] J. Luiten et al., “Hota: A higher order metric for evaluating multi-object tracking,” *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 548–578, 2021.

[30] H. Luo et al., “A strong baseline and batch normalization neck for deep person re-identification,” *IEEE Trans. Multimedia*, vol. 22, no. 10, pp. 2597–2609, Oct. 2020.

[31] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, “Mot16: A benchmark for multi-object tracking,” 2016, *arXiv:1603.00831*.

[32] M. A. Naiel, M. O. Ahmad, M. Swamy, J. Lim, and M. H. Yang, “Online multi-object tracking via robust collaborative model and sample selection,” *Comput. Vis. Image Understanding*, vol. 154, pp. 94–107, 2017.

[33] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, “Tubetk: Adopting tubes to track multi-object in a one-step training model,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6308–6318.

[34] J. Pang et al., “Quasi-dense similarity learning for multiple object tracking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 164–173.

[35] J. Peng et al., “TPM: Multiple object tracking with tracklet-plane matching,” *Pattern Recognit.*, vol. 107, 2020 Art. no. 107480.

[36] A. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, “Multi-object tracking through simultaneous long occlusions and split-merge conditions,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2006, vol. 1, pp. 666–673.

[37] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof, “Occlusion geodesics for online multi-object tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1306–1313.

[38] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” 2018, *arXiv:1804.02767*.

[39] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[40] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 17–35.

[41] S. Shao et al., “Crowdhuman: A benchmark for detecting human in a crowd,” 2018, *arXiv:1805.00123*.

[42] D. Stadler and J. Beyerer, “On the performance of crowd-specific detectors in multi-pedestrian tracking,” in *Proc. 17th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2021, pp. 1–12.

[43] D. Stadler and J. Beyerer, “Modelling ambiguous assignments for multi-person tracking in crowds,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 133–142.

[44] P. Sun et al., “Dancetrack: Multi-object tracking in uniform appearance and diverse motion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 20993–21002.

[45] P. Sun et al., “Transtrack: Multiple object tracking with transformer,” 2020, *arXiv:2012.15460*.

[46] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon, “Learning to track with object permanence,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10860–10869.

[47] B. Wang, G. Wang, K. L. Chan, and L. Wang, “Tracklet association by online target-specific metric learning and coherent dynamics estimation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 589–602, Mar. 2017.

[48] G. Wang et al., “Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9876–9886.

[49] G. Wang, Y. Wang, R. Gu, W. Hu, and J. N. Hwang, “Split and connect: A universal tracklet booster for multi-object tracking,” *IEEE Trans. Multimedia*, early access, Jan. 06, 2022, doi: 10.1109/TMM.2022.3140919.

[50] G. Wang, Y. Wang, H. Zhang, R. Gu, and J. N. Hwang, "Exploit the connectivity: Multi-object tracking with trackletnet," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 482–490.
[51] Q. Wang, Y. Zheng, P. Pan, and Y. Xu, "Multiple object tracking with correlation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3876–3886.
[52] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 107–122.
[53] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10359–10366.
[54] C. Williams and C. Rasmussen, "Gaussian processes for regression," *Adv. Neural Inf. Process. Syst.*, vol. 8, 1995.
[55] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3645–3649.
[56] J. Wu et al., "Track to detect and segment: An online multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12352–12361.
[57] Y. Xu et al., "How to train your deep multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6787–6796.
[58] F. Yang, X. Chang, S. Sakti, Y. Wu, and S. Nakamura, "ReMOT: A model-agnostic refinement for multiple object tracking," *Image Vis. Comput.*, vol. 106, 2021, Art. no. 104091.
[59] E. Yu, Z. Li, S. Han, and H. Wang, "Relationtrack: Relation-aware multiple object tracking with decoupled representation," *IEEE Trans. Multimedia*, early access, Feb. 10, 2022, doi: 10.1109/TMM.2022.3150169.
[60] F. Yu et al., "POI: Multiple object tracking with high performance detection and appearance feature," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 36–42.
[61] F. Zeng, B. Dong, T. Wang, X. Zhang, Y. Wei, and Y. Zhang, "Motr: End-to-end multiple-object tracking with transformer," in *Proc. Comput. Vis.–ECCV 2022: 17th Eur. Conf.*, Israel: Tel Aviv, Oct. 2022, pp. 659–675.
[62] S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3213–3221.
[63] Y. Zhang et al., "Bytetrack: Multi-object tracking by associating every detection box," in *Proc. Comput. Vis.–ECCV 2022: 17th Eur. Conf.*, Israel: Tel Aviv, Oct. 2022, pp. 1–21.
[64] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, 2021.
[65] L. Zheng et al., "Mars: A video benchmark for large-scale person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 868–884.
[66] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 474–490.
[67] Y. Zhu, K. Zhou, M. Wang, Y. Zhao, and Z. Zhao, "A comprehensive solution for detecting events in complex surveillance videos," *Multimedia Tools Appl.*, vol. 78, no. 1, pp. 817–838, 2019.

**Yunhao Du** is currently working toward the Ph.D. degree with the Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include multiple object tracking, object re-identification, and action detection.

**Zhicheng Zhao** (Member, IEEE) received the Ph.D. degree in communication and information systems from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. From 2015 to 2016, he was a Visiting Scholar with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Professor with the Beijing University of Posts and Telecommunications. He has authored or coauthored more than 90 journal articles and conference papers. His research interests include computer vision, and image and video semantic understanding and retrieval.

**Yang Song** is currently working toward the master's degree with the Media Communications and Pattern Recognition Laboratory, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include multi-object tracking, video understanding, and pedestrian re-identification tasks.

**Yanyun Zhao** received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009. She is currently an Associate Professor of the School of Artificial Intelligence, Beijing University of Posts and Telecommunications. She has authored or coauthored more than 60 journal articles and conference papers and some textbooks. Her research interests include pattern recognition and image and video processing.

**Fei Su** (Member, IEEE) received the Ph.D. degree in communication and electrical systems from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2000. From 2008 to 2009, she was a Visiting Scholar with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. She is currently a Professor with Multimedia Communication and Pattern Recognition Laboratory, BUPT. She has authored and coauthored more than 90 journal articles and conference papers and some textbooks. Her interests include pattern recognition, image and video processing, and biometrics.

**Tao Gong** received the Ph.D. degree with School of Cyber Science and Technology, University of Science and Technology of China, Hefei, China, in 2021. He is currently a Young Researcher with Shanghai AI Laboratory. His research interests include computer vision and deep learning.

**Hongying Meng** (Senior Member, IEEE) received the Ph.D. degree in communication and electronic systems from Xi'an Jiaotong University, Xi'an, China. He was a Lecturer with Electronic Engineering Department, Tsinghua University, Beijing, China. He is currently a Reader with the Department of Electronic and Electrical Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge, U.K. Before that, he held research positions in several UK universities, including University College London, London, U.K., University of York, York, U.K., University of Southampton, Southampton, U.K., University of Lincoln, U.K., and University of Dundee, Dundee, U.K. He is a Member of Engineering Professors' Council, and a Fellow of The Higher Education Academy in U.K. He is an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, and the General Chair of 16th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2020).