This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

MotionTrack: Learning Robust Short-term and Long-term Motions for Multi-Object Tracking

Zheng Qin[1†] Sanping Zhou[1†] Le Wang[1*] Jinghai Duan[2] Gang Hua[3] Wei Tang[4]
[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]School of Software Engineering, Xi'an Jiaotong University
[3]Wormpex AI Research [4]University of Illinois at Chicago

Abstract

The main challenge of Multi-Object Tracking (MOT) lies in maintaining a continuous trajectory for each target. Existing methods often learn reliable motion patterns to match the same target between adjacent frames and discriminative appearance features to re-identify the lost targets after a long period. However, the reliability of motion prediction and the discriminability of appearances can be easily hurt by dense crowds and extreme occlusions in the tracking process. In this paper, we propose a simple yet effective multi-object tracker, i.e., MotionTrack, which learns robust short-term and long-term motions in a unified framework to associate trajectories from a short to long range. For dense crowds, we design a novel Interaction Module to learn interaction-aware motions from short-term trajectories, which can estimate the complex movement of each target. For extreme occlusions, we build a novel Refind Module to learn reliable long-term motions from the target's history trajectory, which can link the interrupted trajectory with its corresponding detection. Our Interaction Module and Refind Module are embedded in the well-known tracking-by-detection paradigm, which can work in tandem to maintain superior performance. Extensive experimental results on MOT17 and MOT20 datasets demonstrate the superiority of our approach in challenging scenarios, and it achieves state-of-the-art performances at various MOT metrics. Code is available at https://github.com/qwomeng/MotionTrack.

1. Introduction

Multi-Object Tracking (MOT) is a fundamental task in computer vision, which has a wide range of applications, such as autonomous driving [8] and intelligent surveillance [29]. It aims at jointly locating targets through bounding boxes and recognizing their identities throughout a whole video [40]. Though great progress has been made in the past few years, MOT still remains a challenging task due to the dynamic environment, such as dense crowds and extreme occlusions, in the tracking scenario.

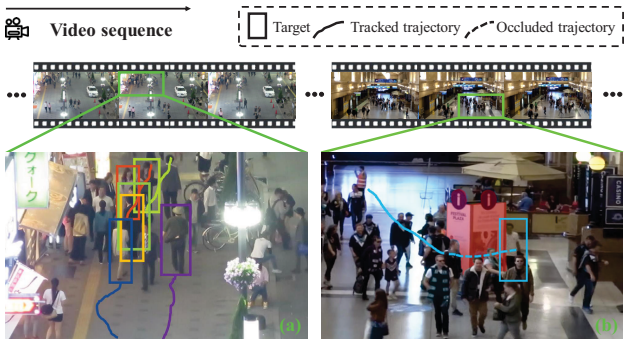

Figure 1. **Illustration of challenging scenarios in different videos**. (a) **Dense crowds.** Pedestrians do not move independently in this situation. They will be affected by their surrounding neighbors to avoid collisions which will make their motion patterns hard to learn in practice. (b) **Extreme occlusion.** Pedestrians are easily occluded by fixed facilities for a long period, such as billboard and sunshade, in which the dynamic environment will make them undergone a large appearance variation.

In general, the existing MOT methods either follow the tracking-by-detection [2] or tracking-by-regression [39, 40, 59], paradigm. The former methods first detect objects in each video frame and then associate detections between adjacent frames to create individual object tracks over time. The latter methods conduct tracking differently: the object detector not only provides frame-wise detections but also replaces the data association with a continuous regression of each tracklet to its new position. Regardless of the

[†]Co-first authors. *Corresponding author.

paradigm, all methods need to address the short-range and long-range association problems, *i.e.*, how to associate the alive tracklets with detections in a short time, and how to re-identify the lost tracklets with detections after a long period.

For the short-range association problem, discriminative motion patterns and appearance features [5, 44] are often learned to conduct data association between adjacent frames. However, as shown in Figure 1 (a), it is tough to learn discriminative representations in the dense crowd scenario. On the one hand, the bounding boxes of detections are too small to be distinguished by their appearances. On the other hand, different targets need to plan suitable paths to avoid collisions, which makes the resulting motions very complex in the tracking process. For the long-range association problem, prior works [24, 43, 44] usually learn discriminative appearance features to re-identify the lost targets after long occlusion [56–58]. As shown in Figure 1 (b), the main bottleneck of these methods is how to keep the robustness of features against different poses, low resolution, and poor illumination for the same target. To alleviate this issue, the memory technology [7, 46] is widely applied to store diverse features for each target to match different targets in a multi-query manner. Moreover, a lot of memories and time will be consumed by the memory module and multi-query regime, unfriendly to real-time tracking.

In this paper, we propose a simple yet effective object tracker, *i.e.*, MotionTrack, to address the short-range and long-range association problems in MOT. In particular, our MotionTrack follows the tracking-by-detection paradigm, in which both interaction-aware and history trajectory-based motions are learned to associate trajectories from a short to long range. To deal with the short-range association problem, we design a novel Interaction Module to model all interactions between targets, which can predict their complex motions to avoid collisions. The Interaction Module uses an asymmetric adjacency matrix to represent the interaction between targets, and obtains the prediction after the information fusion by a graph convolution network. Thanks to the captured target interaction, those short-term occluded targets can be successfully tracked in dense crowds. To deal with the long-range association problem, we design a novel Refind Module based on the history trajectory of each target. It can effectively re-identify the lost targets through two steps: correlation calculation and error compensation. For the lost tracklets and the unmatched detections, the correlation calculation step takes the features of history trajectories and current detections as input, and computes a correlation matrix to represent the possibility that they are associated. Afterward, the error compensation step is further taken to revise the occluded trajectories. Extensive experiments on two benchmark datasets (MOT17 and MOT20) demonstrate that our proposed MotionTrack outperforms the previous state-of-the-art methods.

The main contribution of this work can be highlighted as follows: (1) We propose a simple yet effective multi-object tracker, MotionTrack, to address the short-range and long-range association problems; (2) We design a novel Interaction Module to model the interaction between targets, which can handle complex motions in dense crowds; (3) We design a novel Refind Module to learn discriminative motion patterns, which can re-identify the lost tracklets with current detections.

2. Related Work

Tracking-by-Detection. The recent advancement of object detection brings remarkable improvement to the tracking-by-detection paradigm [2]. In this framework, an existing object detector [14, 32] generates detections in the current frame, then a matching algorithm, *e.g.*, the Hungarian algorithm [28], builds tracklets by associating the detections across different frames. Many efforts have been made in different aspects to improve the effectiveness of this paradigm. For example, SORT [5] adds the Kalman Filter (KF) [19] to approximate the inter-frame displacements, while other models [30, 41, 44, 47] focus on distinguishing appearance features to improve the matching accuracy. Some other works [6, 18, 21] formulate data association as a graph optimization problem by considering each detection as a graph node. For example, ByteTrack [52] enhances tracking performance by fully using detections. We follow this tracking-by-detection paradigm and propose a novel framework to extract interaction-aware and history trajectory-based motions for more accurate short and long range association.

Motion Models. The motion models can be divided into filter-based and model-based methods in MOT. The filter-based methods mainly consider motion prediction as state estimation. For example, the well-known SORT [5] introduces KF [19] into MOT as a linear constant velocity model based on the assumption of independence across the objects and the camera motion, which inspires a series of works [17, 20, 53] to improve the motion prediction in different aspects. Later, many works [1, 3, 12, 36] consider the camera motion compensation of the tracker for robust tracking. Meanwhile, some works [12, 13] using KF variants to further improve prediction accuracy.

Recently, model-based methods combine motion and visual information to provide better predictions based on a data-driven paradigm. For example, Tracktor [3] adopts the regression part from Faster R-CNN [32] to predict the displacement of targets between adjacent frames. FFT [51] further adds optical flow to help regress the displacement. Besides, CenterTrack [59] builds a tracking branch to predict the motion specifically. ArTIST [34] considers motion as a probability distribution and implicitly models all

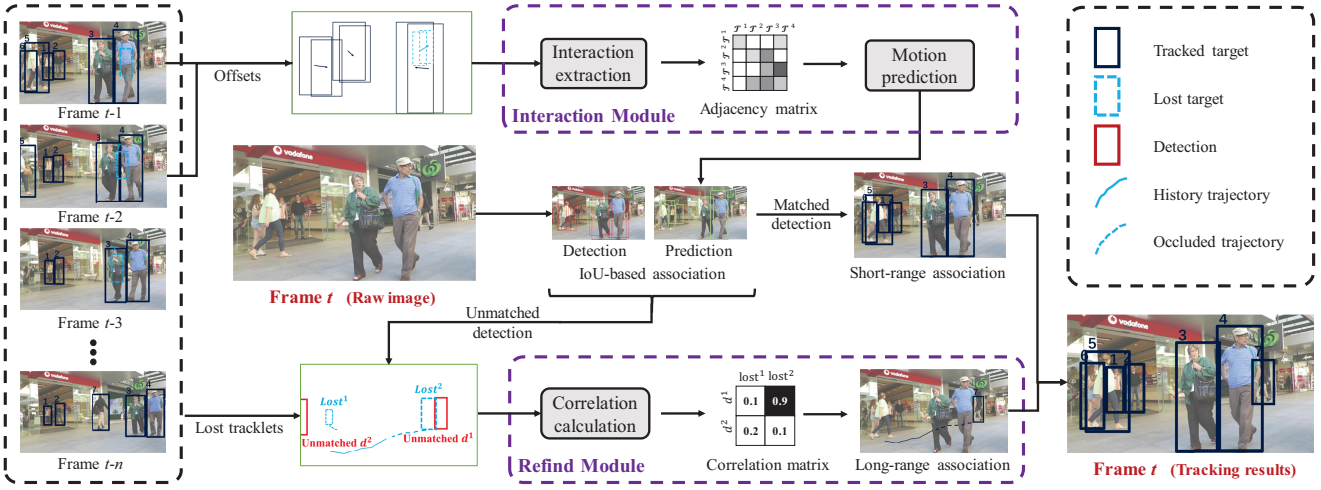

Figure 2. **Overview of our MotionTrack**. The **Interaction Module** captures the directional interaction relationship between tracklets, then fuses the interaction information and predicts the location in the next frame for the short-range association. The **Refind Module** analyzes the correspondence between unmatched lost tracklets and detections by correlation calculation, then the matched pairs are further chosen to complete the long-range association via an additional error compensation. Finally, the short-range and long-range associations are combined to generate complete tracking results.

surrounding interactions with max-pooling to one feature. However, most motion models do not consider explicit interactions between targets, especially in dense crowd scenes. As a result, they cannot accurately estimate the complex movements of nearby targets.

Occlusion. How to deal with occlusions has been a long-standing challenge in MOT. In particular, occlusions can be divided into short-term and long-term occlusions. The short-term occlusion means the target is incomplete in a frame as some other objects occlude it. It prevents the extraction of high-quality detection features for the association. To address this issue, some works try to separate the unoccluded and occluded targets [15, 46].

The long-term occlusion occurs when the tracking target is lost for a long period due to obstacles. To alleviate this issue, DeepSORT [44] proposes a cascaded matching strategy that first matches the detection boxes to the alive tracklets and then to the lost targets based on appearance features. MeMOT [7] builds a memory bank to store the appearance features of the tracklets for retrieval. QuoVadis [10] convert trajectories [16, 35] to a bird's eye view. ByteTrack [52] re-identifies the lost tracklet using the IoU score between the tracklet's iterative prediction and detection. However, these methods will become less reliable as the occlusion time becomes longer. We propose a new representation of the tracklet based on its history trajectory, so that the model can still provide a reliable matching strategy in extremely occluded scenes.

3. Method

3.1. Notation

As shown in Figure 2, our MotionTrack follows a well-known tracking-by-detection paradigm [52]. We first process each frame with YOLOX [14] to obtain the detection results. The detections are denoted as $\mathcal{D}^t = \{\mathbf{d}_i^t\}_{i=1}^N$ containing N detections in frame t. A detection $\mathbf{d}_i^t \in \mathbb{R}^4$ is represented as (x, y, w, h), where (x, y) means the bounding box center, and w and h indicate its width and height, respectively. We denote the set of M tracklets by $\mathbb{T} = \{\mathcal{T}_j\}_{j=1}^M$. $\mathcal{T}_j$ is a tracklet with identity j and is defined as $\mathcal{T}_j = \{\mathbf{l}_j^{t_0}, \mathbf{l}_j^{t_0+1}, ..., \mathbf{l}_j^t\}$, where $\mathbf{l}_j^t$ is the location in frame t, and t_0 is the initialized moment.

When tracking begins, we initialize the set of tracklets $\mathbb{T}$ with $\mathcal{D}^1$. For the subsequent video frames, we assign the new detections to their corresponding tracklets, and update $\mathbb{T}$ at each time step. Throughout the whole video sequence, new tracklets are constantly initialized and incorporated into $\mathbb{T}$. Meanwhile, the existing tracklets may be terminated and removed from $\mathbb{T}$. In the tracking process, the tracklet may be interrupted, for example due to dense crowds and extreme occlusion, therefore we further divide $\mathbb{T}$ into two parts, *i.e.*, $\mathbb{T}^{\text{alive}}$ and $\mathbb{T}^{\text{lost}}$, which denote the set of tracked tracklets and the set of lost but not yet removed tracklets, respectively. A tracklet in $\mathbb{T}^{\text{alive}}$ is represented as $\mathcal{T}_k = \{\mathbf{d}_k^{t_0}, \mathbf{d}_k^{t_0+1}, ..., \mathbf{d}_k^t\}$, and a tracklet in $\mathbb{T}^{\text{lost}}$ is represented as $\mathcal{T}_s = \{\mathbf{d}_s^{t_0}, \mathbf{d}_s^{t_0+1}, ..., \mathbf{p}_s^{t_{\text{lost}}}, \mathbf{p}_s^{t_{\text{lost}}+1}, ..., \mathbf{p}_s^t\}$, where t_{lost} denotes the moment it is lost, and $\mathbf{p}_s^t$ is the prediction during occlusion calculated in Section 3.3 below.

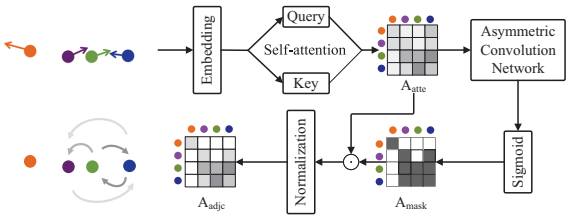

Figure 3. **Illustration of Interaction Extraction**. We capture interactions between tracklets, and the shades of arrows reflect the degree of interaction.

3.2. Overview of MotionTrack

Our MotionTrack mainly executes two steps for each video frame:

- **Step 1: Short-range association.** Modeling the inter-tracklet interaction to obtain more accurate predictions and the short-range tracking results.
- **Step 2: Long-range association.** Re-identifying lost tracklets based on the history trajectory and unmatched detections and then compensating the trajectory during occlusion.

All tracks go through three states from birth to death: alive, lost, and dead. The tracklet being tracked is called alive, and its state is converted to lost upon interruption by accident, such as extreme occlusion. When the interruption occurs for a long time, the tracklet becomes dead, and we remove it from the tracklets set $\mathbb{T}$.

Short-range Association. Given the input video frame t, we first obtain its detections $\mathcal{D}^t$. In addition, we have obtained the set of M tracklets $\mathbb{T}$ up to frame $t-1$, with S lost tracklets $\mathbb{T}^{\text{lost}}$. Then, we construct the directed interactions between tracklets to get the predictions at frame t. As shown in Figure 2, we first calculate the iterative offsets $\mathbf{O}^t \in \mathbb{R}^{M\times 4}$, where each row $\mathbf{o}_j^t = (\Delta x_j^t, \Delta y_j^t, \Delta w_j^t, \Delta h_j^t)$ and Δ denotes the offset from frame $t-2$ to frame $t-1$. The Interaction Module concatenates absolute coordinates and offsets as input, denote as $\mathbf{I}^t \in \mathbb{R}^{M\times 8}$. An asymmetric interaction matrix $\mathbf{A}^{\text{adjc}} \in [0,1]^{M\times M}$ is obtained through interaction extraction, where each element indicates the impact of one tracklet on the other. Afterward, $\mathbf{A}^{\text{adjc}}$ is used to estimate an accurate offset in the motion prediction step. Finally, we follow the association policy in ByteTrack [52] to update the alive tracklets $\mathbb{T}^{\text{alive}}$ with matched detections and record the predictions for lost tracklets $\mathbb{T}^{\text{lost}}$.

Long-range Association. Suppose that there are S lost tracklets and U unmatched detections after the short-range association. Some unmatched detections could share the identities with those lost tracklets, which motivates us to learn history trajectory-based representations for the long-range association. In particular, we first calculate the correlation between the S lost tracklets and the U unmatched detections based on the spatial distribution of trajectories and the velocity-time relationship to obtain the correlation matrix $\mathbf{C}^{\text{corre}} \in [0,1]^{S\times U}$. Then, we retain highly relevant pairs and utilize error compensation to refine the trajectory. Finally, we combine the results of short-range and long-range associations to generate the complete tracking results at frame t.

3.3. Interaction Module

To obtain more accurate tracklets, we capture the directed interactions between tracklets in the interaction extraction step and then use them to estimate the offsets between two consecutive frames in the motion prediction step.

Interaction Extraction. As shown in Figure 3, we first obtain the attention matrix $\mathbf{A}^{\text{atte}} \in \mathbb{R}^{M\times M}$, which measures the interaction magnitude between every pair of tracklets, using the self-attention mechanism [38]:

$$\begin{aligned}
\mathbf{E}^t &= \phi\left(\mathbf{I}^t, \mathbf{W}^{\text{E}}\right), \\
\mathbf{Q}^t &= \phi\left(\mathbf{E}^t, \mathbf{W}^{\text{Q}}\right), \\
\mathbf{K}^t &= \phi\left(\mathbf{E}^t, \mathbf{W}^{\text{K}}\right), \\
\mathbf{A}^{\text{atte}} &= \text{Softmax}\left(\frac{\mathbf{Q}^t {\mathbf{K}^t}^{\text{T}}}{\sqrt{d}}\right),
\end{aligned} \tag{1}$$

where $\phi(\cdot,\cdot)$ denotes linear transformation (multiplying a weight matrix and adding a bias), $\mathbf{E}^t$ is a higher-dimension embedding mapped from $\mathbf{I}^t$. $\mathbf{Q}^t \in \mathbb{R}^{M\times D}$ and $\mathbf{K}^t \in \mathbb{R}^{M\times D}$ are the query and key of the self-attention mechanism. $\mathbf{W}^{\text{E}}$, $\mathbf{W}^{\text{Q}}$, and $\mathbf{W}^{\text{K}}$ are the weights of the linear transformation, and $\sqrt{d} = \sqrt{D}$ is the scaling factor [38].

We use the attention matrix $\mathbf{A}^{\text{atte}}$ to express the asymmetric interaction between different tracklets instead of the undirected spatial distance. The (i,j)-th element in $\mathbf{A}^{\text{atte}}$ represents the influence of tracklet i on tracklet j. To further consider the whole scene, such as group behavior, we model higher levels of interaction via a cascade of asymmetric convolution [11]:

$$\mathbf{A}_l = \delta\left(\text{conv}\left(\mathbf{A}_{l\text{-}1}, \mathbf{K}_{1\times\kappa}\right) + \text{conv}\left(\mathbf{A}_{l\text{-}1}, \mathbf{K}_{\kappa\times 1}\right)\right), \tag{2}$$

where $\mathbf{K}_{1\times\kappa}$ and $\mathbf{K}_{\kappa\times 1}$ are the asymmetric convolution kernels, δ denotes the PReLU, and $\mathbf{A}_0$ is initialized as $\mathbf{A}^{\text{atte}}$. There are L convolution layers. Afterward, to capture significant interactions between tracklets, we retain only high attention values in $\mathbf{A}^{\text{adjc}} \in [0,1]^{M\times M}$:

$$\begin{aligned}
\mathbf{A}^{\text{mask}} &= \text{sgn}\left(\varphi(\mathbf{A}_{\text{L}}) - \xi\right), \\
\mathbf{A}^{\text{adjc}} &= \mathbf{A}^{\text{mask}} \odot \mathbf{A}^{\text{atte}},
\end{aligned} \tag{3}$$

where φ and $\odot$ denote the sigmoid function and element-wise multiplication, respectively, sgn is a sign function, and

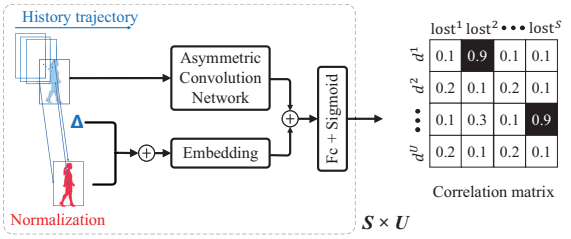

Figure 4. **Illustration of Correlation Calculation**. We compute a correlation matrix to represent the association probabilities between lost tracklets and detections.

$\xi \in [0, 1]$ is a threshold. Finally, we normalize all the non-zero elements in $\mathbf{A}^{\text{adjc}}$.

Motion Prediction. After extracting the interaction matrix between tracklets, we use a graph convolution [22] to fuse the interactions for each tracklet and get the prediction using a multi-layer perceptron (MLP):

$$\mathbf{P}^{\text{offs}} = \text{MLP}\left(\delta\left(\phi(\mathbf{A}^{\text{adjc}} \cdot \mathbf{O}^t, \mathbf{W}^{\text{G}})\right)\right), \tag{4}$$

where $\mathbf{W}^{\text{G}}$ is the weight of the linear transformation. The prediction $\mathbf{P}^{\text{offs}} \in \mathbb{R}^{M\times 4}$ will be used with $\mathcal{D}^t$ for IoU-based association.

3.4. Refind Module

To refind the lost tracklet, we first identify its matched detection in the correlation calculation step and then refine the occluded trajectory in the error compensation step.

Correlation Calculation. After IoU-based association, there are U unmatched detections $\mathbf{D}^{\text{rest}} \in \mathbb{R}^{U\times 5}$, and S lost tracklets $\mathbf{T}^{\text{lost}} \in \mathbb{R}^{S\times 30\times 5}$, in which we record the last thirty alive locations for each lost tracklet. $\mathbf{D}^{\text{rest}}$ and $\mathbf{T}^{\text{lost}}$ include both time and locations, *i.e.*, (t, x, y, w, h), and they are the inputs to the Refind Module. As shown in Figure 4, we first normalize $\mathbf{D}^{\text{rest}}$ and $\mathbf{T}^{\text{lost}}$ in the last dimension and then extract features from them separately. We apply the asymmetric convolution to $\mathbf{T}^{\text{lost}}$ in the second and third dimensions, respectively, and then pool them into feature vectors $\mathbf{F}^{\text{traj}} \in \mathbb{R}^{S\times D}$:

$$\begin{aligned} \mathbf{T}_l &= \delta\left(\text{conv}(\mathbf{T}_{l\text{-}1}, \mathbf{K}_{\kappa\times 1})\right), \\ \mathbf{F}^{\text{traj}} &= \text{pool}\left(\delta(\text{conv}(\mathbf{T}_{\text{L}}, \mathbf{K}_{1\times\kappa}))\right), \end{aligned} \tag{5}$$

where $\mathbf{T}_0$ is initialized as $\mathbf{T}^{\text{lost}}$ and there are L convolution layers. We calculate the difference between the detection and the last alive location, and concatenate it with the detection as $\hat{\mathbf{D}}^{\text{rest}} \in \mathbb{R}^{U\times 10}$. Then, we map it to high-dimensional features $\mathbf{F}^{\text{dete}} \in \mathbb{R}^{U\times D}$ as follows:

$$\mathbf{F}^{\text{dete}} = \phi(\hat{\mathbf{D}}^{\text{rest}}, \mathbf{W}^{\text{D}}), \tag{6}$$

where $\mathbf{W}^{\text{D}}$ is the weight of the linear transformation. Afterward, we combine $\mathbf{F}^{\text{traj}}$ and $\mathbf{F}^{\text{dete}}$ into a feature matrix $\mathbf{F} \in \mathbb{R}^{(S\times U)\times 2D}$, which models the spatial distribution pattern and velocity-time correlation, *etc*. A fully connected layer and a sigmoid function are then applied to yield the correlation score. Here, we obtain the correlation matrix $\mathbf{C}^{\text{corr}} \in \mathbb{R}^{S\times U}$ reflecting the association probabilities between lost tracklets and unmatched detections. Finally, we use the greedy algorithm to pick the matched pairs with high correlation socre and initialize the remaining unmatched detections as new tracklets.

Error Compensation. After re-identifying the lost tracklet with its matched detection, we need to fill the trajectory during the long-time occlusion. Unlike other interpolation methods, we correct the predicted trajectory instead of generating a new one. We use the error between the matched detection $\mathbf{d}^t$ and the prediction $\mathbf{p}^t$ of the lost tracklet to infer the errors during occlusion and refine the prediction:

$$\mathbf{d}^{t_p} = \mathbf{p}^{t_p} + \left(\mathbf{d}^t - \mathbf{p}^t\right)\frac{t_p - t_1}{t_2 - t_1}, \quad t_1 < t_p < t_2, \tag{7}$$

where the tracklet becomes lost after frame t_1 and is refound at frame t_2. When the tracklet is occluded, it still exists and interacts with other tracklets, which can provide some information to support the prediction of the lost one. Related experiments are in the supplementary material.

3.5. Training

The output of the Interaction Module is a set of offsets $\mathbf{P}^{\text{offs}}$. We convert it to location coordinates $\mathbf{P}^{\text{coor}}$ based on location in the previous frame. Each coordinate in $\mathbf{P}^{\text{coor}}$ is composed of four values (x, y, w, h), which are not independent. These four variables affect each other, so they should not be supervised individually but combined to drive the training of the Interaction Module. Inspired by the training process of the detector [14], we employ the IoU loss [49] to supervise the Interaction Module as follows:

$$\mathcal{L}^{\text{INTR}} = 1 - \text{IoU}\left(\mathbf{P}^{\text{coor}}, \mathbf{P}^{\text{gt}}\right)^2, \tag{8}$$

where IoU denotes the Intersection over Union. We take three consecutive frames as a training sample. The offset between the first two frames is taken as input, and the last frame is used as supervision $\mathbf{P}^{\text{gt}}$.

For the training of Refind Module, we extract the data samples and labels from the tracking dataset. We first extract all the trajectories in a complete video, and then randomly couple them in pairs, each being a training set. For each training set, we sample the tracklet and detection, and label positive or negative by whether they are from the same trajectory. Then, we supervise the correlation calculation in Refind Module with a binary cross-entropy loss function:

$$\mathcal{L}^{\text{CORR}} = \frac{1}{n}\sum_{i}^{n} -[y_i \log(c_i) + (1 - y_i)\log(1 - c_i)], \tag{9}$$

Tracker	Venue	IDF1 ↑	MOTA ↑	HOTA ↑	AssA ↑	DetA ↑	FP ↓	FN ↓	IDs ↓	Frag ↓
ReMOT [45]	IVC'21	72.0	77.0	59.7	57.1	62.8	33204	93612	2853	5304
QuasiDense [30]	CVPR'21	66.3	68.7	53.9	52.7	55.6	26589	146643	3378	8091
SOTMOT [55]	CVPR'21	71.9	71.0	-	-	-	39537	118983	5184	-
SiamMOT [23]	CVPR'21	72.3	76.3	-	-	-	-	-	-	-
CorrTracker [42]	CVPR'21	73.6	76.5	60.7	58.5	62.9	29808	99510	3369	6063
PermaTrackPr [37]	ICCV'21	68.9	73.8	55.5	53.1	58.5	28998	115104	3699	6132
FairMOT [53]	IJCV'21	72.3	73.7	59.3	58.0	60.9	27507	117477	3303	8073
CSTrack [24]	TIP'22	72.6	74.9	59.3	57.9	61.1	23847	114303	3567	7668
RelationTrack [48]	TMM'22	74.7	73.8	61.0	61.5	60.6	27999	118623	1374	2166
TrackFormer [26]	CVPR'22	68.0	74.1	-	-	-	34602	108777	2829	-
MeMOT [7]	CVPR'22	69.0	72.5	56.9	55.2	-	37221	115248	2724	-
MTrack [46]	CVPR'22	73.5	72.1	-	-	-	53361	101844	2028	-
MOTR [50]	ECCV'22	68.6	73.4	57.8	55.7	60.3	-	-	2439	-
ByteTrack [52]	ECCV'22	77.3	80.3	63.1	62.0	64.5	25491	83721	2196	2277
P3AFormer(+W&B) [54]	ECCV'22	78.1	**81.2**	-	-	-	**17281**	86861	1893	-
MotionTrack(ours)	-	**80.1**	81.1	**65.1**	**65.1**	**65.4**	23802	**81660**	**1140**	**1605**

Table 1. Comparison with the state-of-the-art methods under the "private detector" protocol on the MOT17 test set. ↑ means higher is better, ↓ means lower is better. The best results for each metric are bolded.

where c_i denotes the predicted correlation score, and y_i indicates the ground truth correlation label, in which 1 and 0 represent the positive and negative correlation, respectively.

4. Experiments

4.1. Setting

Datasets. We evaluate our MotionTrack under the "private detection" protocol on the MOT17 [27] and MOT20 [9] datasets. For fair comparisons, we directly apply the publicly available detector of YOLOX [14], trained by ByteTrack [52] on MOT17 and MOT20. For the training of the Interaction Module and Refind Module, we use only half the training set of MOT17 and MOT20.

Metrics. We employ CLEAR metrics [4] (MOTA, FP, FN, IDs, *etc.*), IDF1 [33], and HOTA [25] to evaluate different aspects of tracking performance. In particular, IDF1 focuses more on association performance, MOTA is computed based on FP, FN, and IDs, which mainly rely on the detection performance because the number of FPs and FNs is larger than IDs. HOTA is a unified metric that balances the effectiveness of detection and association.

Implementation Details. We implemented our MotionTrack in PyTorch [31], and performed all experiments on one NVIDIA GeForce RTX 3090 Ti GPU. For fair comparisons, we directly apply the publicly available detector of YOLOX [14], trained by [52] for MOT17, MOT20. For the Interaction Module, the threshold used by the signal function is set to 0.6. For Refind Module, pairs with correlation scores less than 0.9 were rejected. For the tracking process, we make full use of all detections with double matching, following [52]. The default high and low thresholds are 0.6 and 0.1, respectively. Unless otherwise specified, the new tracklet initialization score is 0.7. In the IoU-based association, we reject the matching if the IoU score is smaller than 0.2, and we use global motion compensation for steadier tracking. For the lost tracklets, we keep 60 frames for MOT17 and 120 frames for MOT20, respectively.

4.2. Comparison with the State-of-the-Art Methods

MOT17. As shown in Table 1, our MotionTrack outperforms the state-of-the-art methods in most key metrics, *i.e.*, ranks first for metrics IDF1, HOTA, AssA, DetA, IDs, Frag and ranks second for MOTA. Our approach focuses on solving dense crowds and extreme occlusion to enable the tracker with a more robust ability for identity preservation over a short to long range. Consequently, it produces more accurate associations and vastly outperforms the second-performance tracker in metrics reflecting the association ability (*i.e.*, +2.0 IDF1 and +3.1 AssA). It should be pointed out that P3AFormer is based on segmentation, while our MotionTrack is based on detection. Even in this situation, we are only 0.1 lower than P3AFormer in MOTA, but IDF1 surpasses it by 2.0. We are the highest on HOTA, demonstrating the robustness and comprehensive tracking capability of our MotionTrack. For the IDs metric, we are 40% less than P3AFormer, which shows the strong ability of our association component.

MOT20. As shown in Table 2, our MotionTrack still achieves state-of-the-art results on the MOT20 dataset [1] . Even though ByteTrack uses duplicate detections as our method, our MotionTrack has made significant progress in several core metrics (*i.e.*, +1.3 IDF1, +0.2 MOTA, and +1.5 HOTA). Besides, we still achieve the highest HOTA in this more challenging scenario. Moreover, our MotionTrack achieves the least IDs, which is 13% less than the second P3AFormer. The underlying reason is that we infer the occluded trajectory in the Refind Module, which is able to

[1] We ranked second among all methods on the official MOT Challenge evaluation server and first among all online methods.

Tracker	Venue	IDF1 ↑	MOTA ↑	HOTA ↑	AssA ↑	DetA ↑	FP ↓	FN ↓	IDs ↓	Frag ↓
FairMOT [53]	IJCV'21	67.3	61.8	54.6	54.7	54.7	103440	88901	5243	7874
CorrTracker [42]	CVPR'21	69.1	65.2	-	-	-	79429	95855	5183	-
SiamMOT [23]	CVPR'21	69.1	67.1	-	-	-	-	-	-	-
SOTMOT [55]	CVPR'21	71.4	68.6	57.4	57.3	57.7	57064	101154	4209	7568
CSTrack [24]	TIP'22	68.6	66.6	54.0	50.0	54.2	**25404**	144358	3196	7632
RelationTrack [48]	TMM'22	70.5	67.2	56.5	56.4	56.8	61134	104597	4243	8236
MeMOT [7]	CVPR'22	66.1	63.7	54.1	55.0	-	47882	137982	1938	-
MTrack [46]	CVPR'22	69.2	63.5	-	-	-	96123	86964	6031	-
ByteTrack [52]	ECCV'22	75.2	77.8	61.3	59.6	63.4	26249	87594	1223	1460
P3AFormer(+W&B) [54]	ECCV'22	76.4	**78.1**	-	-	-	25413	86510	1332	-
MotionTrack(ours)	-	**76.5**	78.0	**62.8**	**61.8**	**64.0**	28629	**84152**	**1165**	**1321**

Table 2. Comparison with the state-of-the-art methods under the "private detector" protocol on the MOT20 test set. ↑ means higher is better, ↓ means lower is better. The best results for each metric are bolded.

Setting	IDF1 ↑	MOTA ↑	HOTA ↑	AssA ↑	DetA ↑	IDs ↓
Baseline	82.6	80.4	70.2	72.4	68.7	402
Baseline+I	83.0	80.5	70.5	72.9	68.8	390
Baseline+I+R	83.7	80.7	70.8	73.5	68.9	378

Table 3. Ablation studies on Interaction Module (I) and Refind Module (R) on the MOT17 validation set.

Setting	#	IDF1 ↑	MOTA ↑	HOTA ↑	AssA ↑	DetA ↑
IoU-based	30	80.9	79.8	69.0	70.6	68.1
	120	80.1	77.6	68.4	70.5	66.9
	Δ	**-0.8**	**-2.2**	**-0.6**	**-0.1**	**-1.2**
ReID-based	30	77.2	77.0	66.4	67.1	66.3
	120	70.4	67.5	60.6	60.3	61.6
	Δ	**-6.8**	**-9.5**	**-5.8**	**-6.8**	**-4.7**
Ours	30	82.6	80.4	70.2	72.4	68.7
	120	83.3	80.7	70.7	73.2	68.8
	Δ	**+0.7**	**+0.3**	**+0.5**	**+0.8**	**+0.1**

Table 4. Comparison with other methods for handling occlusions on the MOT17 validation set. We raise the upper limit of occlusion time from 30 to 120 frames to reflect the ability to deal with long-term occlusion. Increases and decreases in metrics are marked in green and red, respectively.

Setting	⩾ 20	⩾ 40	⩾ 60	⩾ 80	⩾ 100
Baseline	77.2	73.6	75.2	73.3	71.5
Ours	78.3	75.1	76.8	74.1	72.4
Improvement	**+1.1**	**+1.5**	**+1.6**	**+0.8**	**+0.9**

Table 5. Evaluation of MOTA for crowd and occlusion cases on the MOT17 validation set. We set visibility < 0.25 and define the minimum time constants for occlusion or crowds as 20 to 100, respectively. Increases and decreases in metrics are marked in green and red, respectively.

keep a consistent identity for each trajectory.

4.3. Ablation Study

Effect of Each Component. We conduct ablative experiments to verify the contribution of each component of our MotionTrack. For reliable verification, all other settings of baseline are the same, except it utilizes the Kalman Filter (KF) for motion prediction and re-identifies the lost tracklet using the IoU score between the tracklet's iterative prediction and detection. Then, our proposed Interaction Module and Refind Module replace the above two components, respectively. As shown in Table 3, the Interaction Module can improve IDF1, HOTA, AssA, and IDs, which indicates the effectiveness of the introduction of interaction. Our Refind Module can improve all the metrics, indicating that the learned history trajectory-based motions are effective for the long-range data association problem.

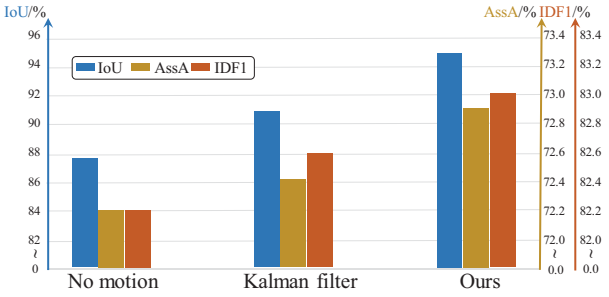

Figure 5. **Results of Motion Models**. The three metrics reflect the motion prediction accuracy (IoU) and the ability of association (IDF1 and AssA), respectively.

Analysis of Interation Module. The IoU-based trackers predict the location of tracklets in the next frame, and then compute the IoU between the detection boxes and the predicted boxes to conduct data association. These methods heavily rely on accurate predictions to maintain a high-quality data association. As shown in Figure 5, we compare the prediction accuracy, in which we take the average IoU of all targets to count the total IoU score. The results show that the introduction of interaction yields more accurate predictions than the traditional Kalman Filter. Meanwhile, steady improvements in IDF1 and AssA indicate that Interaction Module improves prediction accuracy and leads to stronger association capability.

Advantage of Refind Module. In practice, the IoU-based and Re-ID-based approaches are often taken to deal with

the occlusion problem, while our method learns the discriminative motion patterns between the history trajectories and current detection to address the extreme occlusion problem. As shown in Table 4, we compare the classical approach [44, 52] with our method in terms of its ability to handle long-term occlusion. The results show that the first two methods cause dramatic performance decreases due to a large number of incorrect associations, while our approach method achieves a consistent improvement in dealing the long-term occlusion.

Extra Evaluation of Crowd and Occlusion. Extreme occlusion and dense crowds are not always present in the tracking scene. Therefore evaluating the entire dataset cannot reflect our ability to directly solve the two challenging problems. To address this issue, we propose to modify MOTA to crowdMOTA, which separates people in the dataset who are likely to be occluded or crowded, and evaluate the tracking performance of these people. In particular, we select people by visibility label in the dataset, *i.e.*, considering people being crowded or occluded when their visibility is below 0.25. We set the minimum number of frames from 20 to 100 for continuous occlusion or crowding to validate the solution for samples with different difficulty levels. As shown inc Table 5, the improvement in samples of different difficulties illustrates the effectiveness of our method in solving dense crowds and extreme occlusion.

4.4. Visualization

Visualization of Directed Interaction. The directed interaction is visualized in Figure 6, from which we find that our method possesses the ability to capture effective interaction in different scenarios. In particular, (a) shows that the stride forward movement pattern of the pedestrian is affected by the pedestrians coming towards him. In (b), a cyclist walking toward the left will affect two oncoming pedestrians. As shown in (c), even if the tracklet is in the LOST state, he is still influenced by the pedestrians around him. Although he cannot be seen, he is still actually in the crowd. As a result, considering the interaction of nearby people helps describe the movement of the target during occlusion.

Visualization of Refinding Targets. As shown in Figure 7, two cases of refinding targets are given in the red box. When the target is occluded, our Interaction Module still iteratively infers its location. Therefore, our Refind Module can accurately re-identify the lost targets through correlation calculation and refine the predicted trajectory by error compensation.

5. Conclusion

We propose MotionTrack for online MOT, which introduces an Interaction Module and Refind Module to address the short-term and long-term association problems in MOT. Our results on the MOT benchmark datasets have shown the benefits of our method. The application of interactions can lead to more accurate location prediction, yielding more robust data association. Even though the targets have been occluded for a long period, they can still be refound by learning discriminative motion patterns between the history trajectory and current detection. Notably, without using any complex components, such as person Re-ID, our tracker achieves state-of-the-art performance.

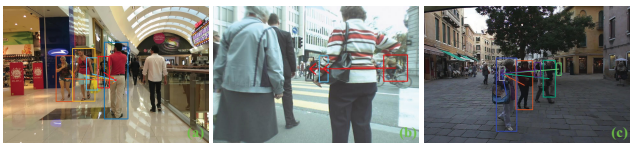

Figure 6. **Visualization of Directed Interaction**. The boxes in different colors represent the bounding boxes of different targets, the end of arrow represents the affecting target, and the head of arrow represents affected target.

Figure 7. **Visualization of Refinding Targets**. The red boxes represent the locations during occlusion, which are obtained based on both the iterative prediction of the Interaction Module and the error compensation of the Refind Module.

Limitation and Future Work. One major limitation of our method is that we only considered the motion patterns and relationships between pedestrians while ignoring the drivable information in interaction, which may weaken its performance in motion prediction. In the future, we plan to explore this prior information in our MotionTrack to further improve the tracking performance. Besides, two modules we proposed can be further combined to support each other for achieving better results.

Acknowledgement

This work was supported in part by National Key R&D Program of China under Grant 2021YFB1714700, NSFC under Grants 62088102 and 62106192, Natural Science Foundation of Shaanxi Province under Grants 2022JC-41, China Postdoctoral Science Foundation under Grant 2022T150518, and Fundamental Research Funds for the Central Universities under Grants XTR042021005 and XTR072022001.

References

[1] Nir Aharo, Roy Orfaig, and Ben-Zion Bobrovsky. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 2

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, pages 1–8, 2008. 1, 2

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019. 2

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *JIVP*, 2008:1–10, 2008. 6

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468, 2016. 2

[6] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, pages 6247–6257, 2020. 2

[7] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. MeMOT: Multi-object tracking with memory. In *CVPR*, pages 8090–8100, 2022. 2, 3, 6, 7

[8] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. DeepDriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, pages 2722–2730, 2015. 1

[9] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 6

[10] Patrick Dendorfer, Vladimir Yugay, Aljoša Ošep, and Laura Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking? 2022. 3

[11] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. ACNet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *ICCV*, pages 1911–1920, 2019. 4

[12] Yunhao Du, Yang Song, Bo Yang, and Yanyun Zhao. StrongSORT: Make deepsort great again. *arXiv preprint arXiv:2202.13514*, 2022. 2

[13] Yunhao Du, Junfeng Wan, Yanyun Zhao, Binyu Zhang, Zhihang Tong, and Junhao Dong. GIAOTracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021. In *ICCV*, pages 2809–2819, 2021. 2

[14] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 2, 3, 5, 6

[15] Song Guo, Jingya Wang, Xinchao Wang, and Dacheng Tao. Online multiple object tracking with cross-task synergy. In *CVPR*, pages 8136–8145, 2021. 3

[16] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, pages 2255–2264, 2018. 3

[17] Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, and Xiaofeng Pan. MAT: Motion-aware multi-object tracking. *Neurocomputing*, 476:75–86, 2022. 2

[18] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In *CVPR*, pages 5299–5309, 2021. 2

[19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 2

[20] Tarasha Khurana, Achal Dave, and Deva Ramanan. Detecting invisible people. In *ICCV*, pages 3174–3184, 2021. 2

[21] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *CVPR*, pages 4696–4704, 2015. 2

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 5

[23] Chao Liang, Zhipeng Zhang, Xue Zhou, Bing Li, and Weiming Hu. One more check: making "fake background" be tracked again. In *AAAI*, pages 1546–1554, 2022. 6, 7

[24] Chao Liang, Zhipeng Zhang, Xue Zhou, Bing Li, Shuyuan Zhu, and Weiming Hu. Rethinking the competition between detection and reid in multiobject tracking. *IEEE T-IP*, 31:3182–3196, 2022. 2, 6, 7

[25] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *IJCV*, 129(2):548–578, 2021. 6

[26] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-object tracking with transformers. In *CVPR*, pages 8844–8854, 2022. 6

[27] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 6

[28] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957. 2

[29] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, pages 3153–3160, 2011. 1

[30] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, pages 164–173, 2021. 2, 6

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019. 6

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 2

[33] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for

multi-target, multi-camera tracking. In *ECCV*, pages 17–35, 2016. 6

[34] Fatemeh Saleh, Sadegh Aliakbarian, Hamid Rezatofighi, Mathieu Salzmann, and Stephen Gould. Probabilistic tracklet scoring and inpainting for multiple object tracking. In *CVPR*, pages 14329–14339, 2021. 2

[35] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *CVPR*, pages 8994–9003, 2021. 3

[36] Daniel Stadler and Jürgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *WACV*, pages 133–142, 2022. 2

[37] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *ICCV*, pages 10860–10869, 2021. 6

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 6000–6010, 2017. 4

[39] Xingyu Wan, Jiakai Cao, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Tracking beyond detection: learning a global response map for end-to-end multi-object tracking. *IEEE T-IP*, 30:8222–8235, 2021. 1

[40] Xingyu Wan, Sanping Zhou, Jinjun Wang, and Rongye Meng. Multiple object tracking by trajectory map regression with temporal priors embedding. In *ACM MM*, pages 1377–1386, 2021. 1

[41] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *CVPR*, pages 3876–3886, 2021. 2

[42] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *CVPR*, pages 3876–3886, 2021. 6, 7

[43] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, pages 107–122, 2020. 2

[44] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649, 2017. 2, 3, 8

[45] Fan Yang, Xin Chang, Sakriani Sakti, Yang Wu, and Satoshi Nakamura. ReMOT: A model-agnostic refinement for multiple object tracking. *Image Vis. Comput.*, 106:104091, 2021. 6

[46] En Yu, Zhuoling Li, and Shoudong Han. Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking. In *CVPR*, pages 8834–8843, 2022. 2, 3, 6, 7

[47] En Yu, Zhuoling Li, and Shoudong Han. Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking. In *CVPR*, pages 8834–8843, 2022. 2

[48] En Yu, Zhuoling Li, Shoudong Han, and Hongwei Wang. RelationTrack: Relation-aware multiple object tracking with decoupled representation. *IEEE T-MM*, 2022. 6, 7

[49] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. UnitBox: An advanced object detection network. In *ACM MM*, pages 516–520, 2016. 5

[50] Fangao Zeng, Bin Dong, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. *arXiv preprint arXiv:2105.03247*, 2021. 6

[51] Jimuyang Zhang, Sanping Zhou, Xin Chang, Fangbin Wan, Jinjun Wang, Yang Wu, and Dong Huang. Multiple object tracking by flowing and fusing. *arXiv preprint arXiv:2001.11180*, 2020. 2

[52] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *ECCV*, pages 1–21, 2022. 2, 3, 4, 6, 7, 8

[53] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 129(11):3069–3087, 2021. 2, 6, 7

[54] Zelin Zhao, Ze Wu, Yueqing Zhuang, Boxun Li, and Jiaya Jia. Tracking objects as pixel-wise distributions. In *ECCV*, pages 76–94, 2022. 6, 7

[55] Linyu Zheng, Ming Tang, Yingying Chen, Guibo Zhu, Jinqiao Wang, and Hanqing Lu. Improving multiple object tracking with single object tracking. In *CVPR*, pages 2453–2462, 2021. 6, 7

[56] Sanping Zhou, Fei Wang, Zeyi Huang, and Jinjun Wang. Discriminative feature learning with consistent attention regularization for person re-identification. In *ICCV*, pages 8040–8049, 2019. 2

[57] Sanping Zhou, Jinjun Wang, Deyu Meng, Yudong Liang, Yihong Gong, and Nanning Zheng. Discriminative feature learning with foreground attention for person re-identification. *IEEE T-IP*, 28:4671–4684, 2019. 2

[58] Sanping Zhou, Jinjun Wang, Jiayun Wang, Yihong Gong, and Nanning Zheng. Point to set similarity based deep feature learning for person re-identification. In *CVPR*, pages 3741–3750, 2017. 2

[59] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, pages 474–490, 2020. 1, 2