DeconfuseTrack: Dealing with Confusion for Multi-Object Tracking

Cheng Huang[*], Shoudong Han[*,†], Mengyu He, Wenbo Zheng, Yuhao Wei
National Key Laboratory of Multispectral Information Intelligent Processing Technology,
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
{chenghuang7, shoudonghan}@hust.edu.cn

Abstract

Accurate data association is crucial in reducing confusion, such as ID switches and assignment errors, in multi-object tracking (MOT). However, existing advanced methods often overlook the diversity among trajectories and the ambiguity and conflicts present in motion and appearance cues, leading to confusion among detections, trajectories, and associations when performing simple global data association. To address this issue, we propose a simple, versatile, and highly interpretable data association approach called Decomposed Data Association (DDA). DDA decomposes the traditional association problem into multiple sub-problems using a series of non-learning-based modules and selectively addresses the confusion in each sub-problem by incorporating targeted exploitation of new cues. Additionally, we introduce Occlusion-aware Non-Maximum Suppression (ONMS) to retain more occluded detections, thereby increasing opportunities for association with trajectories and indirectly reducing the confusion caused by missed detections. Finally, based on DDA and ONMS, we design a powerful multi-object tracker named DeconfuseTrack, specifically focused on resolving confusion in MOT. Extensive experiments conducted on the MOT17 and MOT20 datasets demonstrate that our proposed DDA and ONMS significantly enhance the performance of several popular trackers. Moreover, DeconfuseTrack achieves state-of-the-art performance on the MOT17 and MOT20 test sets, significantly outperforms the baseline tracker ByteTrack in metrics such as HOTA, IDF1, AssA. This validates that our tracking design effectively reduces confusion caused by simple global association.

1. Introduction

Multi-object tracking (MOT) is a crucial task in the field of computer vision with extensive applications, including video surveillance [42], autonomous driving [33], and human-computer interaction [13]. The goal of MOT is to track multiple objects of interest simultaneously in a video sequence. Despite significant advancements in this field, MOT still faces several challenges, such as occlusion, appearance variations, and complex interactions between objects.

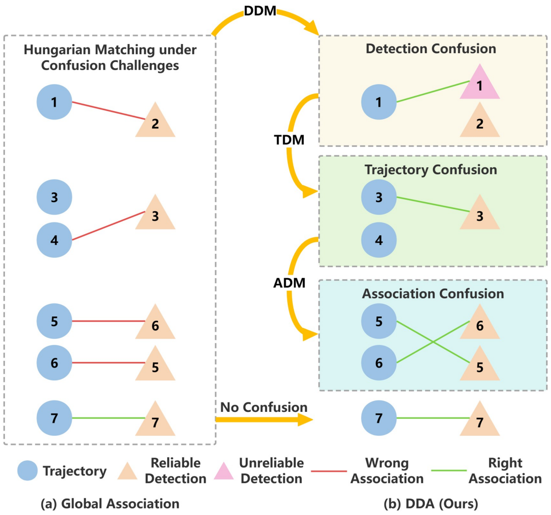

Figure 1. Comparing different data association methods. (a) Global Association. (b) Decomposed Data Association (Ours)

In recent years, most MOT methods [5, 7, 12, 50, 58] adopt the tracking-by-detection paradigm. In this paradigm, data association plays a crucial role in MOT by establishing correspondences between tracking trajectories and detection results. To improve the accuracy of data association, many methods introduce additional cues to complement motion cues, including appearance features [50, 57], motion direction [7, 28], confidence scores [23, 58], depth information [11, 29], and natural language cues [55]. These additional cues are shown to alleviate issues caused by the ambiguity of motion cues or motion estimation errors to some extent. Furthermore, some methods [26, 31, 58] divide the data association process into multiple stages, assigning priorities to trajectories and detection results through incremental matching, thereby reducing confusion during association.

[*] Equal contribution
[†] Corresponding author

However, current MOT methods still have some limitations in terms of data association, primarily in two aspects. Firstly, many state-of-the-art methods [7, 12, 57] treat data association as a global optimization problem, considering the assignment between all tracking trajectories and detection results as a single optimization task. However, such holistic association approaches may retain numerous confusions, leading to a degradation in tracking performance. As shown in Fig. 1(a), global association often leads to numerous confusions. For instance, Trajectory 1 is incorrectly associated with Detection 2, which should be initialized as a new track. In another case, Trajectory 4 is mistakenly matched with Detection 3. Additionally, Trajectory 5 and 6 experience ID switches due to their close proximity. Secondly, although many methods consider using multiple cues to complement motion information, these methods [12, 57] either simply linearly weight the multiple cues or utilize heuristic rules [1, 28] for fusion. This approach, on one hand, makes the tracker highly sensitive to fusion hyperparameters, and on the other hand, may introduce uncertainties from the new cues that could potentially interfere with the accuracy of motion cues.

To address these issues, we propose a simple yet effective multi-object tracking method called DeconfuseTrack, which aims to tackle confusion in data association. Firstly, we suggest a more detailed consideration of the data association problem in multi-object tracking. We decompose the global association problem into several sub-problems, including the association between a single trajectory and multiple detections, the association between multiple trajectories and a single detection, and the association between multiple trajectories and multiple detections. By considering these sub-problems more thoroughly, we can minimize erroneous association assignments. When utilizing appearance cues, we adopt a decoupling strategy, only supplementing with appearance cues when the discriminative power of motion cues is insufficient in the sub-problems. The use of appearance cues is constrained within a certain range to minimize interference with motion cues. As shown in Fig. 1(b), Through Detection Disambiguation Module (DDM), we identify Detection 1 that, although unreliable, is a better fit for Trajectory 1, freeing up Detection 2. With Trajectory Disambiguation Module (TDM), we make the correct selection between Trajectory 3 and Trajectory 4 for Detection 3. Through Association Disambiguation Module (ADM), we avoid association confusion between Trajectory 5 and Trajectory 6. Secondly, to enhance detection performance and mitigate confusion caused by missed detections, we design Occlusion-aware Non-Maximum Suppression (ONMS) to retain more occluded detection boxes for association. Extensive experimental results demonstrate that our proposed DeconfuseTrack method outperforms state-of-the-art methods on two widely adopted benchmark datasets, MOT17 [32] and MOT20 [10].

Our work has made the following main contributions:

- We design a novel plug-and-play data association method called Decomposed Data Association (DDA). It decomposes the traditional global data association into a series of sub-problems and handles them step by step, reducing the confusion in the assignment stage of MOT matching.
- We propose ONMS, a method that preserves more occluded detections for data association in the post-detection processing stage. This approach has the potential to reduce the occurrence of confusion during association.
- By combining DDA and ONMS, we propose a simple yet powerful multi-object tracker named DeconfuseTrack, to address the challenges of confusion in MOT.

2. Related Work

Tracking-by-Detection. Among the frameworks utilized in MOT, the tracking-by-detection paradigm stands out as the earliest and most widely embraced approach. It aims to detect objects in video frames using an object detector and then connect the objects across frames using data association methods. The performance of detection plays a crucial role in improving tracking performance. Therefore, some methods choose to use better detectors to obtain improved detection results. For example, the MOT17 dataset [32] uses DPM [15], Faster-RCNN [38], and SDP [51]. CenterNet [61] is adopted by many methods [45, 48, 57, 62] due to its simplicity and ease of use. YOLOX [18] has become the choice of most MOT methods [7, 11, 12, 35, 47, 58, 60] due to its powerful detection performance. Another category of methods focus on improving the accuracy of object motion prediction to better associate objects across frames. For instance, many methods [5, 7, 19, 50, 57, 58] employ Kalman filters [22] for motion prediction. Some methods [1, 3, 12] consider using camera motion compensation to assist in object motion prediction. A few methods [11, 35, 40] utilize learnable models for motion prediction. In addition, appearance modeling is crucial for improving object discrimination. Some methods [1, 12, 14, 50] use independent Re-Identification (ReID) models to extract appearance features of the targets. Other methods [34, 37, 49, 57] incorporate the ReID task as a branch of the detector, enabling a single model to simultaneously perform detection and embed target features.

Our approach follows the popular tracking-by-detection paradigm and maintains the same configuration as the popular methods [1, 7, 58] in terms of motion prediction, ReID, and other aspects. However, we observe that the majority of methods simply employ NMS to filter out duplicate detections, resulting in the loss of many detected occluded objects and underutilization of the detector's performance. Therefore, we chose to use a simple ONMS technique to retain as many detections as possible.

Data Association. Data association is a crucial module in multi-object trackers, aiming to accurately assign tracking trajectories to detections. Data association methods in MOT can be traced back to the radar domain, such as JPDA [2, 16] and MHT [36]. However, these algorithms have high computational complexity and require prior assumptions about the number of targets, making them less commonly adopted in modern visual MOT. Many visual MOT data association methods are based on SORT [5], which utilizes a simple Hungarian matching [24] to assign detections to tracks. DeepSORT [50] introduces a cascaded matching approach that categorizes tracks into "confirmed" and "tentative" states. It prioritizes matching the confirmed tracks before considering the tentative ones, reducing identity switches during tracking. ByteTrack [58] proposes a multi-stage association method that first associates tracks with high-scoring detections in the first stage, and then associates the remaining tracks with low-scoring detections in the second stage. By utilizing more detections, it significantly reduces false negatives. Another category of methods achieve implicit data association using learnable models. For example, some methods [9, 20, 25, 53] employ Graph Neural Networks (GNN) to model similarity and data association. Another set of methods [6, 17, 30, 56] utilize the query mechanism of Transformers [44] for association.

Although different trackers employ various data association methods, they typically adopt a global association approach, often overlooking the individual characteristics of trajectories and detections while neglecting the ambiguity in the clues. In contrast, our method utilizes a decomposed data association technique, which tackles the ambiguity from a more granular perspective. Moreover, we don't incorporate any learnable modules, striking a balance between speed and interpretability.

3. Method

3.1. Notation

Our method follows the popular tracking-by-detection paradigm. Firstly, we utilize a detector to obtain detection results for each frame. The detection results for frame t can be represented as $\mathrm{D}^t = \{\mathrm{d}_i^t \mid i \in \{1, 2, \cdots, N\}\}$, where N is the number of detection boxes in the current frame. Each detection $\mathrm{d}_i^t \in \mathbb{R}^5$ can be represented as $\mathrm{d}_i^t = (x, y, w, h, c)$, where (x, y) denotes the center coordinates of the bounding box, w and h represent the width and height of the bounding box, and c is the confidence score of the detection. Trajectories can be represented as $\mathbb{T} = \{\mathcal{T}_j \mid j \in \{1, 2, \cdots, M\}\}$, where M is the total number of trajectories. Each trajectory is defined as $\mathcal{T}_j = \{\mathrm{o}^s \mid s \in \{t_s, t_s + 1, \cdots, t_e\}\}$, where j is the identity of the trajectory, t_s represents the initialization time of the trajectory, t_e represents the termination time of the trajectory, and $\mathrm{o}^s = (x, y, w, h)$ represents the position of the trajectory at time s. We divide D^t into two categories like ByteTrack[58]: reliable and unreliable, denoted as D^t_{fisrt} and D^t_{second} respectively. They are used for the first and second data associations. This will be explained in detail in Sec. 3.3.

3.2. Decomposed Data Association (DDA)

To tackle the assignment problem and alleviate confusion in tracking, we propose the DDA method. For each frame, similar to popular approaches [5, 7, 58], we utilize the Kalman filter [22] to obtain predicted positions $\mathrm{L} = \{\mathrm{l}_j = (x, y, w, h) \mid j \in \mathbb{T}\}$ for each trajectory in $\mathbb{T}$. The positional similarity between d_i and $\mathcal{T}_j$ is defined as the IoU between the detection bounding box and the predicted bounding box of the trajectory:

$$\mathrm{LocSim}(\mathrm{d}_i, \mathcal{T}_j) = \mathrm{IoU}(\mathrm{d}_i, \mathrm{l}_j)\,. \tag{1}$$

Then use D^t_{first} and L to calculate the cost matrix C:

$$\mathrm{C}_{ij} = 1 - \mathrm{LocSim}(\mathrm{d}_i, \mathcal{T}_j)\,,\ \mathrm{d}_i \in \mathrm{D}^t_{first}\,,\ \mathcal{T}_j \in \mathbb{T}\,. \tag{2}$$

Finally, we utilize the Hungarian algorithm [24] to solve C and obtain the allocation result $\mathrm{P} = \{(\mathrm{d}, \mathcal{T}) \mid \mathrm{d} \in \mathrm{D}^t_{matched}\,, \mathcal{T} \in \mathbb{T}_{matched}\}$ where $\mathrm{D}^t_{matched}$ represents matched detections, and $\mathbb{T}_{matched}$ represents matched tracks. Previous methods would directly output the assignment result at this stage. However, this global association approach still retains some errors due to the confusion in the clues. To obtain more accurate assignment results, we aim to refine P and achieve a finer-grained allocation.

Detection Disambiguation Module (DDM). DDM aims to resolve the confusion between multiple detections and a single trajectory. However, during tracking, trajectories are usually more abundant than reliable detections, making it challenging to encounter situations where multiple reliable detections correspond to a single trajectory. If we were to consider unreliable detections as well, the number of detections would far exceed the number of trajectories. Nonetheless, blindly relying on unreliable detections would introduce numerous errors, as their appearance information is generally unreliable. Therefore, in DDM, we choose to solely utilize motion cues for deconfusion, as they are more reliable in this context.

For the j-th assignment pair in P, we identify the set of detection boxes that could potentially cause confusion with this assignment:

$$\begin{aligned}\mathrm{D}^j_{blur} = \{\mathrm{d}_i \mid \mathrm{LocSim}(\mathrm{d}_i, \mathcal{T}_j) - \mathrm{LocSim}(\mathrm{d}_j, \mathcal{T}_j) > \kappa\,,\\ \mathrm{d}_i \in \mathrm{D}^t_{second}, \mathrm{d}_j \in \mathrm{D}^t_{matched}, \mathcal{T}_j \in \mathbb{T}_{matched}\}\,,\end{aligned} \tag{3}$$

where κ is the confusion reduction factor, this process is equivalent to finding potentially more suitable unreliable detections for each matched trajectory. Next, we define the assignment relationship P_{new}, where the assignment pairs

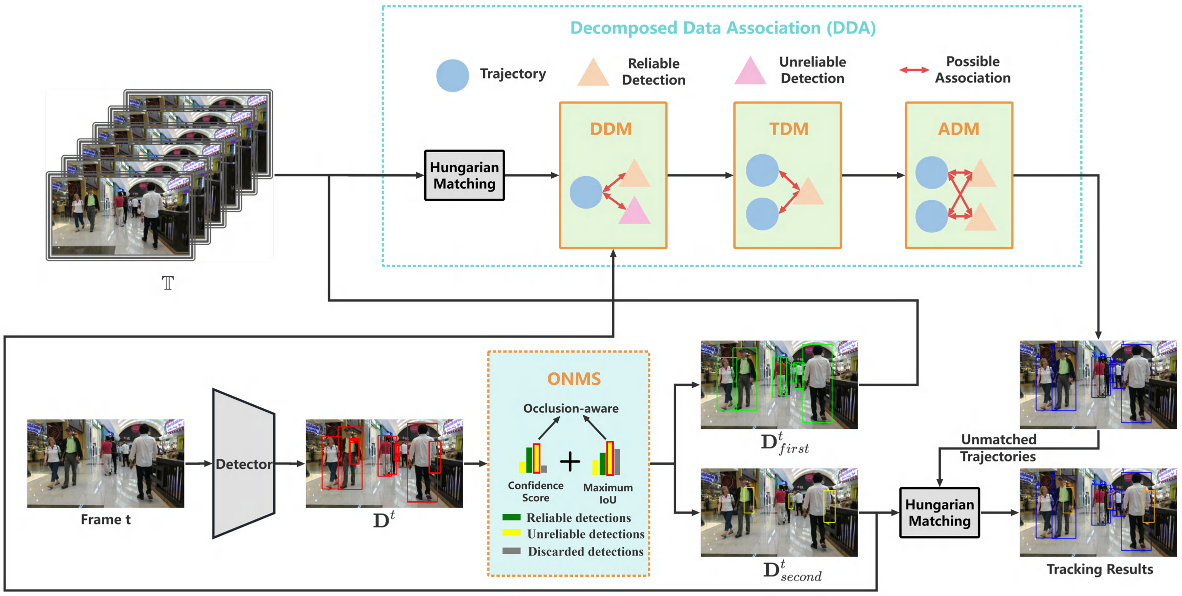

Figure 2. The overall pipeline of DeconfuseTrack. (1) Utilizing a detector to obtain the detection results for the current frame. (2) Employing ONMS to separate the detection results into reliable and unreliable detections. (3) Performing the first association using DDA. (4) Conducting the second association using unreliable detections and unassociated trajectories.

in P_{new} represent the deconfused assignment pairs resulting from our deconfusion process:

$$\begin{aligned} P_{new} = \{(\mathrm{d}, \mathcal{T}_j) \mid \mathrm{d} = \operatorname*{argmax}_{d_i \in \mathrm{D}^j_{blur}} \mathrm{LocSim}(\mathrm{d}_i, \mathcal{T}_j) \\ \text{if } \mathrm{D}^j_{blur} \neq \phi, \mathrm{d}_j \in \mathrm{D}^t_{matched}, \mathcal{T}_j \in \mathbb{T}_{matched}\} . \end{aligned} \tag{4}$$

In the case of conflicts where the same unreliable detection may be selected by multiple trajectories, we retain only the assignment with a higher positional similarity. Finally, we move the unreliable detection boxes matched in P_{new} into the reliable detection boxes:

$$\begin{aligned} \mathrm{D}^{t}_{first}{}' &= \mathrm{D}^t_{first} \cup \{\mathrm{d} \mid (\mathrm{d}, \mathcal{T}) \in \mathrm{P}_{new}\} \\ \mathrm{D}^{t}_{second}{}' &= \mathrm{D}^t_{second} - \{\mathrm{d} \mid (\mathrm{d}, \mathcal{T}) \in \mathrm{P}_{new}\} . \end{aligned} \tag{5}$$

By increasing κ, we can ensure that the trajectories in P_{new} find much more suitable detections compared to those in the original P. After obtaining P_{new}, we potentially free up some reliable detection boxes in P, as they are replaced by more appropriate unreliable detection boxes. However, these mismatched reliable detection boxes still have the possibility of being associated with unmatched trajectories. Therefore, in the final step, while ensuring the validity of the assignment relationship in P_{new}, we perform a reassignment of $\mathbb{T}$ and $\mathrm{D}^{t}_{first}{}'$ to obtain the new assignment relationship P_{ddm} after detection disambiguation.

Trajectory Disambiguation Module (TDM). Targets in the tracking process are prone to fragmentation due to occlusion, rapid motion, and other factors, leading to the formation of multiple trajectories. Additionally, erroneous initialization of false detections can also contribute to the increase in the number of trajectories. As a result, there is a challenge of matching multiple trajectories to a single detection. Furthermore, factors such as camera motion, long-term target absence, and inaccurate detector localization contribute to the ambiguity of the predicted positions L. Relying solely on motion cues can lead to confusion. In light of these challenges, we choose to incorporate appearance cues to alleviate the confusion between trajectories and compensate for the limitations of motion information.

First, we identify all unmatched trajectories $\mathbb{T}_{lost} = \mathbb{T} - \mathbb{T}_{matched}$ in the current frame. These trajectories may have been erroneously rejected due to the ambiguity of motion cues. For the j-th assignment pair $(\mathrm{d}_j, \mathcal{T}_j)$ in P, we then identify the set of trajectories that may cause confusion with this assignment:

$$\begin{aligned} \mathbb{T}^j_{blur} = \{\mathcal{T}_i \mid \mathrm{LocSim}(\mathrm{d}_j, \mathcal{T}_j) - \mathrm{LocSim}(\mathrm{d}_j, \mathcal{T}_i) < \kappa , \\ \mathcal{T}_i \in \mathbb{T}_{lost}, \mathrm{d}_j \in \mathrm{D}^t_{matched}, \mathcal{T}_j \in \mathbb{T}_{matched}\} \cup \mathcal{T}_{\mathrm{j}} . \end{aligned} \tag{6}$$

The parameter κ represents the confusion reduction factor, which is used to adjust the degree of confusion reduction. A larger value of κ indicates a higher level of distrust in the positional cues. Next, we employ an appearance model to obtain more accurate assignments. The appearance embedding of trajectory $\mathcal{T}$ is denoted as $f_{\mathcal{T}}$ and the appearance embedding of detection d is denoted as f_{d}. For each set of confused trajectories $\mathbb{T}^j_{blur}$, we select the trajectory with the closest appearance distance to f_{d_j}:

$$\mathcal{T}_{best}^{j} = \underset{\mathcal{T} \in \mathbb{T}_{blur}^{j}}{\operatorname{argmin}} \operatorname{CosDist}(f_{\mathrm{d}_j}, f_{\mathcal{T}}) \,, \tag{7}$$

where $\operatorname{CosDist}(\cdot)$ represents the calculation of cosine distance between two vectors. $\mathcal{T}_{best}^{j}$ can also refer to $\mathcal{T}_j$ itself. In the case of conflicts where a single trajectory may be selected as $\mathcal{T}_{best}^{j}$ by multiple detections, we only retain the assignment with the smaller cosine distance. Finally, we replace $\mathcal{T}_j$ in the original assignment pair with $\mathcal{T}_{best}^{j}$ to obtain the new assignment relationship after trajectory disambiguation:

$$\mathrm{P}_{tdm} = \{(\mathrm{d}_j, \mathcal{T}_{best}^{j}) \mid \mathrm{d}_j \in \mathrm{D}_{matched}^{t}\}. \tag{8}$$

Association Disambiguation Module (ADM). During the tracking process, there can also be cases of target occlusion, target intersection, and other scenarios where we encounter confusion in associating multiple detections with multiple trajectories. For simplicity, we address the confusion between two detections and two trajectories at a time. Cases involving multiple-to-multiple associations can be decomposed into several two-to-two problems for resolution.

First, for any two distinct assignments in P, we use the coefficient of variation to quantify the confusion between them in terms of positional cues:

$$\begin{aligned}
&\operatorname{Cv}(i,j) = \frac{\operatorname{Std}\left(\{\operatorname{LocSim}\left(\mathrm{d}_{k_1}, \mathcal{T}_{k_2}\right) \mid k_1, k_2 \in \{i,j\}\}\right)}{\operatorname{Mean}\left(\{\operatorname{LocSim}\left(\mathrm{d}_{k_1}, \mathcal{T}_{k_2}\right) \mid k_1, k_2 \in \{i,j\}\}\right)}, \\
&\mathrm{d}_i, \mathrm{d}_j \in \mathrm{D}_{matched}^{t}, \mathcal{T}_i, \mathcal{T}_j \in \mathbb{T}_{matched}, i \neq j \,.
\end{aligned} \tag{9}$$

When the coefficient of variation is small, it indicates that there is little difference in positional cues between the assignment pairs. As mentioned in TDM, positional cues are inherently ambiguous, so a small coefficient of variation implies a strong level of confusion between them. Conversely, a large coefficient of variation indicates a significant difference in positional cues between the assignment pair, suggesting a weak level of confusion. Next, we identify all the assignment pairs that exhibit strong confusion:

$$\mathrm{P}_{blur} = \{\left(\left(\mathrm{d}_i, \mathcal{T}_i\right), \left(\mathrm{d}_j, \mathcal{T}_j\right)\right) \mid \operatorname{Cv}(i,j) < \kappa\} \,, \tag{10}$$

where κ is the confusion reduction factor. Next, similar to TDM, we utilize appearance cues to resolve the positional confusion in P_{blur} and find more suitable assignment relationships. If the sum of the appearance distances of the assignment pairs in P_{blur} is smaller after resolving the cross-association, we consider the post-cross-association assignment pairs to be better and include them in the set P_{new}:

$$\begin{aligned}
\mathrm{P}_{new} = \{((\mathrm{d}_i, \mathcal{T}_j), (\mathrm{d}_j, \mathcal{T}_i)) \mid & \operatorname{CosDist}(f_{\mathrm{d}_i}, f_{\mathcal{T}_j}) \\
& + \operatorname{CosDist}(f_{\mathrm{d}_j}, f_{\mathcal{T}_i}) < \operatorname{CosDist}(f_{\mathrm{d}_i}, f_{\mathcal{T}_i}) \\
& + \operatorname{CosDist}(f_{\mathrm{d}_j}, f_{\mathcal{T}_j}), ((\mathrm{d}_i, \mathcal{T}_i), (\mathrm{d}_j, \mathcal{T}_j)) \in \mathrm{P}_{blur}\} \,.
\end{aligned} \tag{11}$$

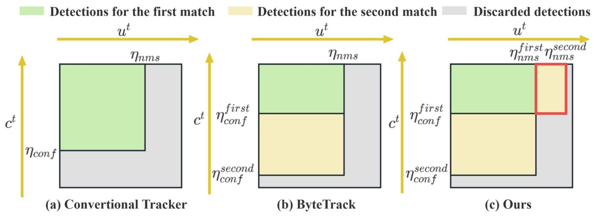

Figure 3. Comparing the post-processing approaches of different trackers: (a) Ordinary trackers use NMS and discard low-scoring detections. (b) ByteTrack also utilizes NMS but retains low-scoring detections. (c) Our method employs ONMS and retains occluded detectons.

During complex matching involving the cross-association of multiple detections and trajectories, conflicts can arise. To resolve these conflicts, we perform the Hungarian matching algorithm again using appearance cues to eliminate the conflicts in P_{new}. Finally, we combine the revised assignment relationship P_{new} with the original set P to obtain the new assignment relationship $\mathrm{P}_{\mathrm{adm}}$ after association disambiguation.

Module Combination. The DDM, TDM, and ADM modules are designed to take the assignment relationship P as input and generate a new assignment relationship $\mathrm{P}^{'}$. Therefore, these three modules can be combined in a serial manner to form the overall DDA. Considering that DDM modifies D_{first}^{t} and D_{second}^{t} to allow for more possibilities in subsequent modules, we prioritize using the DDM module. Since ADM deals with a larger scope than TDM, we place it at the end. All three modules share the confusion reduction factor κ as a hyperparameter for robustness and simplicity. When κ is increased, we consider more confusion cases in TDM and ADM, while fewer unreliable detections are considered in DDM. Thus, a larger value of κ indicates less reliance on positional cues, whereas a smaller value indicates greater reliance on motion cues. The value of κ can be flexibly adjusted based on the motion characteristics of the cameras and tracked objects in the dataset. Additionally, the DDA design does not include any learnable components. The appearance cues used for deconfusion can be obtained from any appearance model, making it easy to integrate the DDA onto other trackers in a flexible and convenient manner.

3.3. Occlusion-aware NMS (ONMS)

Improving the quality of detections can increase the success rate of data association and reduce incorrect associations. By reducing missed detections, more accurate location information can be obtained for the trajectories, reducing confusion caused by inaccurate motion predictions. Therefore, improving detections is crucial for enhancing MOT performance.

We denote the confidence of d^t as c^t and its maximum

IoU with detections having higher confidence as u^t:

$$u_i^t = \max_{\mathrm{d}_j^t \in \{\mathrm{d}^t | c^t > c_i^t, \mathrm{d}^t \in \mathrm{D}^t\}} \mathrm{IoU}(\mathrm{d}_i^t, \mathrm{d}_j^t)\,. \tag{12}$$

As shown in Fig. 3(a), conventional trackers[3, 5, 50, 57, 62] set a confidence threshold η_{conf} and an NMS threshold η_{nms} to retain only detections with c^t higher than η_{conf} and u^t lower than η_{nms} for a single global association. However, this approach mistakenly discards many correct detections. To address this issue, as illustrated in Fig. 3(b), ByteTrack[58] divides detections into two groups by setting two confidence thresholds η_{conf}^{first} and η_{conf}^{second}, and performs two-stage associations to utilize more detections, significantly improving MOT performance. However, we believe that there is still room for improvement. In scenarios with dense target occlusion, we observe that detectors are not incapable of detecting heavily occluded objects. However, previous methods use a single NMS threshold η_{nms} to post-process detection results, striking a balance between missed detections and false positives. Consequently, heavily occluded target boxes are discarded by NMS, even if they have high confidence scores. To address this limitation, as shown in Fig. 3(c), we propose setting two NMS thresholds η_{nms}^{first} and η_{nms}^{second} to retain more detections for the data association stage:

$$\begin{aligned} \mathrm{D}_{first}^t &= \{\mathrm{d}^t \mid \mathrm{d}^t \in \mathrm{D}^t, c^t \geq \eta_{conf}^{first}, u^t \leq \eta_{nms}^{first}\} \qquad (13) \\ \mathrm{D}_{second}^t &= \{\mathrm{d}^t \mid \mathrm{d}^t \in \mathrm{D}^t, \eta_{conf}^{first} > c^t \geq \eta_{conf}^{second}, u^t \leq \eta_{nms}^{first}\} \\ &\cup \{\mathrm{d}^t \mid \mathrm{d}^t \in \mathrm{D}^t, c^t \geq \eta_{conf}^{first}, \eta_{nms}^{first} < u^t \leq \eta_{nms}^{second}\} \end{aligned}$$

3.4. DeconfuseTrack

By combining DDA and ONMS, we propose a tracker called DeconfuseTrack that focuses on addressing confusion in MOT. It adopts the popular tracking-by-detection architecture[7, 12, 58] and utilizes ONMS to enhance the output of the detector, reducing confusion caused by insufficient detection capabilities. Additionally, DDA is employed for more precise data association, reducing confusion arising from ambiguous positional cues. The overall architecture is illustrated in Fig. 2. For the first frame of each tracking video, we initialize $\mathbb{T}$ with D_{first}^t. In the subsequent frames, we update $\mathbb{T}$ using D_{first}^t and D_{second}^t. Unassociated detection boxes in D_{first}^t are added to $\mathbb{T}$ as newborn trajectories, while trajectories in $\mathbb{T}$ that have not been updated within a specified time are removed.

4. Experiments

4.1. Setting

Method	Venue	HOTA↑	IDF1↑	MOTA↑	AssA↑	DetA↑
Learnable Matcher:						
MOTR[56]	ECCV'22	57.8	68.6	73.4	55.7	60,3
MeMOT[6]	CVPR'22	56.9	69.0	72.5	55.2	-
MOTRv2[60]	CVPR'23	62.0	75.0	78.6	60.6	63.8
UTM[53]	CVPR'23	64.0	78.7	**81.8**	-	-
MeMOTR[17]	ICCV'23	58.8	71.5	72.8	58.4	59.6
Non-Learnable Matcher:						
FairMOT[57]	IJCV'21	59.3	72.3	73.7	58.0	60.9
QDTrack[34]	CVPR'21	53.9	66.3	68.7	52.7	55.6
RelationTrack[54]	TMM'22	61.0	74.7	73.8	61.5	60.6
MTracker[59]	ECCV'22	-	75.9	77.3	-	-
ByteTrack[58]	ECCV'22	63.1	77.3	80.3	62.0	64.5
QuoVadis[11]	NeurIPS'22	63.1	77.7	80.3	62.1	64.6
RTU++[46]	TIP'22	63.9	79.1	79.5	63.7	64.5
SAT[47]	ACM MM'22	64.4	79.8	80.0	64.4	64.8
C-BIOU[52]	WACV'23	64.1	79.7	81.1	63.7	64.8
StrongSORT++[12]	TMM'23	64.4	79.5	79.6	64.4	64.6
OC-SORT[7]	CVPR'23	63.2	77.5	78.0	63.2	-
GHOST[41]	CVPR'23	62.8	77.1	78.9	-	-
DeconfuseTrack	-	**64.9**	**80.6**	80.4	**65.1**	**65.0**

Table 1. Comparing with state-of-the-art methods on the MOT17 test set under the private detection protocol. The methods within the pink block utilize YOLOX [18] as the detector. The best results are highlighted in **bold**.

Datasets. We evaluate our DeconfuseTrack using the widely recognized MOT17 [32] and MOT20 [10] benchmarks, following the "Private Detection" protocol. The MOT17 dataset consists of multiple multi-object tracking video sequences captured in natural scenes, providing high-quality annotation information. This dataset includes challenging scenarios such as camera motion and pedestrian occlusion, among others. The MOT20 dataset contains scenes with denser crowds, making it more prone to object confusion. Both the MOT17 and MOT20 datasets provide only training and testing sets. For ablation experiments, we follow the convention proposed in [62], where we use the first half of each video in the MOT17 training set for training and the second half for validation.

Metrics. We utilize widely accepted evaluation metrics, including the CLEAR metrics [4], IDF1 [39], and HOTA [27]. The MOTA, DetA primarily focus on the detection performance, while IDF1, AssA primarily assess the association performance. HOTA provides a balanced evaluation of both detection and tracking performance. DetA primarily reflects detection performance, which fluctuates only slightly around zero points in our experimental process.Therefore, we did not highlight DetA in our experiments.

Implementation Details. We implemented DeconfuseTrack within the MMTracking framework [8] and selected ByteTrack [58] as the baseline. To ensure a fair comparison, we adopted all the hyperparameter settings of ByteTrack and used the same YOLOX [18] detector trained in ByteTrack. For the appearance model, we trained the SBS-50 model from FastReID [21] for 60 epochs on both MOT17 and MOT20 datasets. Regarding DDA, we selected confusion reduction factor κ to 0.3. For ONMS, we set the thresholds η_{nms}^{first} and η_{nms}^{second} to 0.7 and 0.95, respectively.

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

MOT17. Tab. 1 presents the performance of DeconfuseTrack on the MOT17 test dataset. Compared to the baseline

ByteTrack, our method shows significant improvements in association performance, with an increase of 1.8% in HOTA, 3.3% in IDF1, and 3.1% in AssA. These results indicate that our proposed DDA and ONMS methods serve as strong complements to ByteTrack, effectively reducing the confusion caused by simple global data association methods. Moreover, our approach demonstrates a substantial advantage over other trackers [7, 12, 41, 52, 59] that employ unique designs for data association. We achieve the top ranking in HOTA, IDF1, AssA and DetA. These findings suggest that our data association method itself effectively addresses the challenges posed by camera motion and image blurring in the MOT17 dataset, even without the use of complex components such as motion camera compensation.

MOT20. The metrics of The performance metrics of DeconfuseTrack on the MOT20 test dataset are presented in Tab. 2. Our method consistently outperforms ByteTrack in all metrics, with a 2% improvement in HOTA, a 2.4% improvement in IDF1, a 0.3% improvement in MOTA, a 3.1% improvement in AssA, and a 0.7% improvement in DetA. These findings validate the effectiveness of our proposed enhancements in dense object scenarios. Compared to other trackers, our method ranks first in HOTA, IDF1, and DetA, with MOTA only 0.1% lower than the top-performing method [53]. Even when compared to methods that utilize reinforcement learning [47], recurrent neural network [46], graph neural network [53], and Transformer [60] in the data association stage, our approach still exhibits a significant advantage. This suggests that through our decomposed association design, strong performance can be achieved with a simple clue extraction and modeling process.

4.3. Ablation Studies

Analysis of DDA. We conduct ablation experiments to validate the effectiveness of the components in DDA, and the results are shown in Tab. 3. Firstly, when using each deconfusion module separately (rows 2-4), we observe significant improvements. Using TDM alone results in a 0.7% increase in HOTA, 0.6% increase in MOTA, 1.2% increase in IDF1, and 1% increase in AssA. Similarly, using ADM alone leads to a 0.5% increase in both HOTA and MOTA, a 0.6% increase in IDF1, and a 1% increase in AssA. This indicates that the appearance cues effectively alleviate the confusion caused by motion cues. However, when using DDM alone (row 1), the performance gain is relatively low. We speculate that relying solely on low-score detections without utilizing appearance cues limits the ability to deconfuse the motion cues. When combining DDM with TDM (row 5), there is a notable improvement compared to using DDM alone. We believe this is because DDM and TDM synergistically work together, where high-score detections freed by DDM can be further deconfused by TDM after re-association. Finally, when all three sub-modules are used together (row 6), there is an additional 0.4% increase in MOTA, further demonstrating the effectiveness of DDA.

Method	Venue	HOTA↑	IDF1↑	MOTA↑	AssA↑	DetA↑
Learnable Matcher:						
MeMOT[6]	CVPR'22	54.1	66.1	63.7	55.0	-
MOTRv2[60]	CVPR'23	60.3	72.2	76.2	58.1	62.9
UTM[53]	CVPR'23	62.5	76.9	**78.2**	-	-
Non-Learnable Matcher:						
FairMOT[57]	IJCV'21	54.6	67.3	61.8	54.7	54.7
RelationTrack[54]	TMM'22	56.5	70.5	67.2	56.4	56.8
MTracker[59]	ECCV'22	-	67.7	66.3	-	-
ByteTrack[58]	ECCV'22	61.3	75.2	77.8	59.6	63.4
QuoVadis[11]	NeurIPS'22	61.5	75.7	77.8	59.9	63.3
RTU++[46]	TIP'22	62.8	76.8	76.5	62.6	63.1
SAT[47]	ACM MM'22	62.6	76.6	75.0	63.2	62.1
StrongSORT++[12]	TMM'23	62.6	77.0	73.8	**64.0**	61.3
OC-SORT[7]	CVPR'23	62.1	75.9	75.5	62.0	-
GHOST[41]	CVPR'23	61.2	75.2	73.7	-	-
DeconfuseTrack	-	**63.3**	**77.6**	78.1	62.7	**64.1**

Table 2. Comparing with state-of-the-art methods on the MOT20 test set under the private detection protocol. The methods within the pink block utilize YOLOX [18] as the detector. The best results are highlighted in **bold**.

Method	Components			Metrics			
	DDM	TDM	ADM	HOTA↑	MOTA↑	IDF1↑	AssA↑
Baseline				69.0	79.4	80.6	70.6
	√			69.1	79.4	80.7	70.7
		√		69.7	80.0	81.8	71.6
			√	69.5	79.9	81.2	71.6
	√	√		**69.9**	80.0	**82.1**	71.9
Baseline+DDA	√	√	√	**69.9**	**80.4**	**82.1**	**72.0**

Table 3. The ablation study of DDA. (DDM: Detection Disambiguation Module, TDM: Trajectory Disambiguation Module, ADM: Association Disambiguation Module)

Component-wise Analysis. In the ablation experiments, we validated the effectiveness of the components, and the results are shown in Tab. 4. When ONMS was used alone (row 3), there was a 0.6% increase in MOTA, while the overall tracking performance remained largely unchanged. This is consistent with our hypothesis that ONMS primarily aims to improve the detection stage. When combining DDA and ONMS to form DeconfuseTrack (row 4), significant improvements in various association metrics were observed compared to using DDA alone (row 2). This is because the detections recovered by ONMS can be utilized by the DDM component in DDA, enhancing the accuracy of association in cases of severe occlusion. Overall, compared to the baseline (row 1), DeconfuseTrack achieved a 1.7% increase in HOTA, a 1.1% increase in MOTA, a 3.1% increase in IDF1, and a 3% increase in AssA, demonstrating notable improvements in both detection and association aspects.

Robustness to Confusion Reduction Factor. The confusion reduction factor κ is an important hyperparameter in DeconfuseTrack. We adjusted it from 0.1 to 0.5 and compared the tracking metrics. The results are shown in Fig. 4. From the results, we can observe that κ is sensitive and achieves the maximum performance when set to 0.3. Therefore, we selected κ to be 0.3.

Application on Other Trackers. We incorporated

Method	Metrics			
	HOTA↑	MOTA↑	IDF1↑	AssA↑
Baseline	69.0	79.4	80.6	70.6
Baseline+DDA	69.9	80.4	82.1	72.0
Baseline+ONMS	69.1	80.0	80.7	70.6
DeconfuseTrack	**70.7**	**80.5**	**83.7**	**73.6**

Table 4. Ablation study of the components. (DDA: Decomposed Data Association, ONMS: Occlusion-aware NMS).

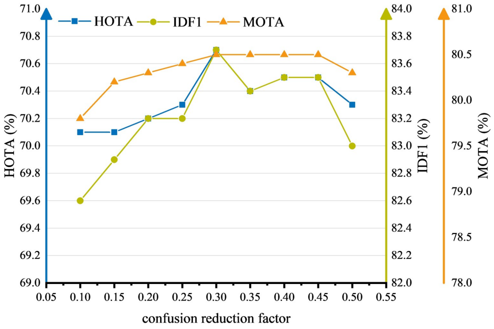

Figure 4. Comparison of the performances of DeconfuseTrack under different detection confusion reduction factor. The results are from the validation set of MOT17.

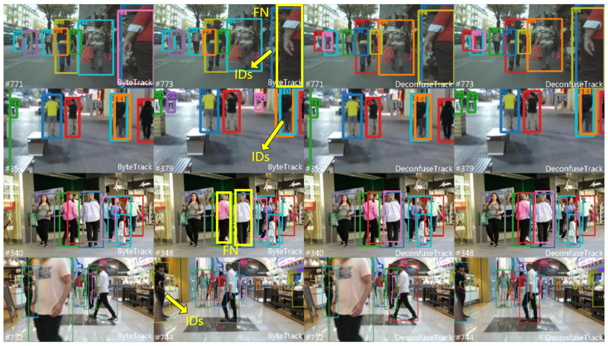

Figure 5. Visualization on the MOT17 validation set.

ONMS and DDA into 5 popular trackers based on the MMTracking[8] framework, and the results are shown in Tab. 5. Our method aims to reduce assignment confusion primarily for trackers that rely on motion cues. As a result, our approach yields good performance on trackers like SORT[5], Tracktor[3], and ByteTrack[58]. However, its impact on DeepSORT[50] is not substantial and may even cause a decrease in IDF1. These results indicate that ONMS and DDA have strong generalization capabilities, allowing for easy integration into advanced trackers and obtaining relatively stable performance gains.

Visualization. We visualized partial results of ByteTrack and our proposed DeconfuseTrack, as shown in Fig. 5. ByteTrack suffers from missed detections and blurry motion clues due to occlusions, image blurriness, camera motion, and other factors. Consequently, it leads to confusion issues such as ID switches, target losses, and localization errors. In contrast, our method incorporates ONMS and DDA, effectively mitigating these problems, which validates the necessity and effectiveness of our deconfusion approach.

Method	Components			Metrics			
	B	O	D	HOTA↑	MOTA↑	IDF1↑	AssA↑
SORT[5]				52.0	62.0	57.8	49.2
	√			52.9	63.0	59.7	50.5
	√	√		53.5(+0.6)	63.8(+0.8)	61.2(+1.5)	51.4(+0.9)
	√	√	√	54.0(+1.1)	65.0(+2.0)	62.5(+2.8)	52.1(+1.6)
DeepSORT[50]				57.3	63.7	69.7	59.6
	√			57.3	65.1	70.2	59.6
	√	√		57.5(+0.2)	65.7(+0.6)	70.0(-0.2)	59.6(+0.0)
	√	√	√	57.6(+0.3)	66.1(+1.0)	70.1(-0.1)	59.8(+0.2)
Tracktor[3]				52.4	61.0	59.8	51.3
	√			53.0	62.0	60.7	52.1
	√	√		53.3(+0.3)	62.5(+0.5)	60.8(+0.1)	52.1(+0.0)
	√	√	√	54.9(+1.9)	63.5(+1.5)	64.8(+4.1)	55.3(+3.2)
Tracktor++[3]				55.7	64.0	66.9	57.7
	√			55.7	64.5	66.8	57.5
	√	√		56.4(+0.7)	65.5(+1.0)	67.6(+0.8)	57.9(+0.4)
	√	√	√	56.4(+0.7)	65.6(+1.1)	67.9(+1.1)	58.0(+0.5)
OC-SORT[7]				67.8	77.4	78.0	69.3
	√			68.8	79.7	79.9	70.1
	√	√		68.9(+0.1)	79.8(+0.1)	80.2(+0.3)	70.1(+0.0)
	√	√	√	68.9(+0.1)	80.0(+0.3)	80.6(+0.7)	70.1(+0.0)

Table 5. Results of applying ONMS and DDA to popular trackers on the MOT17 validation set. In order to highlight our contribution, we provide results compared to each baseline after adding two-stage data association method BYTE[58]. Performance improvements are indicated in green. (B: BYTE, O: ONMS, D: DDA)

5. Conclusion

In this study, we propose a novel plug-and-play data association method called DDA and a simple detection post-processing method called ONMS. Based on these methods, we design a tracker named DeconfuseTrack that focuses on addressing confusion issues in MOT. Extensive experiments demonstrate that a more detailed consideration of data association can significantly improve the performance of existing MOT methods. Our work aims to break the current trend of oversimplified data association steps in most MOT methods and provide insights to researchers, inspiring the development of more effective data association techniques. We believe that by carefully considering data association, future MOT methods can achieve higher accuracy and robustness, driving advancements in the field.

Acknowledgements

This work was supported by the National Key Laboratory Foundation of Multispectral Information Intelligent Processing Technology under Grant No. 6142113220208, and the Science and Technology Plan Project of Jinan under Grant No. 202214002.

References

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

[2] Yaakov Bar-Shalom, Thomas E. Fortmann, and Peter G. Cable. Tracking and Data Association. *The Journal of the Acoustical Society of America*, 87(2):918–919, 1990. 3

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019. 2, 6, 8

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 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. IEEE, 2016. 1, 2, 3, 6, 8

[6] 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. 3, 6, 7

[7] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *CVPR*, pages 9686–9696, 2023. 1, 2, 3, 6, 7, 8

[8] MMTracking Contributors. MMTracking: OpenMMLab video perception toolbox and benchmark. https://github.com/open-mmlab/mmtracking, 2020. 6, 8

[9] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a proposal classifier for multiple object tracking. In *CVPR*, pages 2443–2452, 2021. 3

[10] 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. 2, 6

[11] Patrick Dendorfer, Vladimir Yugay, Aljosa Osep, and Laura Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking? *NeurIPS*, 35:15657–15671, 2022. 1, 2, 6, 7

[12] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE TMM*, 2023. 1, 2, 6, 7

[13] Matteo Dunnhofer, Antonino Furnari, Giovanni Maria Farinella, and Christian Micheloni. Visual object tracking in first person vision. *IJCV*, 131(1):259–283, 2023. 1

[14] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. In *WACV*, pages 466–475. IEEE, 2018. 2

[15] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8. IEEE, 2008. 2

[16] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3): 173–184, 1983. 3

[17] Ruopeng Gao and Limin Wang. Memotr: Long-term memory-augmented transformer for multi-object tracking. In *ICCV*, pages 9901–9910, 2023. 3, 6

[18] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 2, 6, 7

[19] 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

[20] 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. 3

[21] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. In *ACM MM*, pages 9664–9667, 2023. 6

[22] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82 (1):35–45, 1960. 2, 3, 11

[23] Donghwa Kang, Seunghoon Lee, Hoon Sung Chwa, Seung-Hwan Bae, Chang Mook Kang, Jinkyu Lee, and Hyeongboo Baek. Rt-mot: Confidence-aware real-time scheduling framework for multi-object tracking tasks. In *RTSS*, pages 318–330. IEEE, 2022. 1

[24] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 3

[25] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. In *IJCAI*, pages 530–536, 2020. 3

[26] Zelin Liu, Xinggang Wang, Cheng Wang, Wenyu Liu, and Xiang Bai. Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth. *arXiv preprint arXiv:2306.05238*, 2023. 1

[27] 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*, 129:548–578, 2021. 6

[28] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. *arXiv preprint arXiv:2302.11813*, 2023. 1, 2

[29] Gianluca Mancusi, Aniello Panariello, Angelo Porrello, Matteo Fabbri, Simone Calderara, and Rita Cucchiara. Trackflow: Multi-object tracking with normalizing flows. In *ICCV*, pages 9531–9543, 2023. 1

[30] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *CVPR*, pages 8844–8854, 2022. 3

[31] Ting Meng, Chunyun Fu, Mingguang Huang, Xiyang Wang, Jiawei He, Tao Huang, and Wankai Shi. Localization-guided track: A deep association multi-object tracking framework based on localization confidence of detections. *arXiv preprint arXiv:2309.09765*, 2023. 1

[32] 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. 2, 6, 11

[33] Aljoša Osep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. Combined image-and world-space tracking in traffic scenes. In *ICRA*, pages 1988–1995. IEEE, 2017. 1

[34] 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

[35] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In *CVPR*, pages 17939–17948, 2023. 2

[36] Donald Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979. 3

[37] Hao Ren, Shoudong Han, Huilin Ding, Ziwen Zhang, Hongwei Wang, and Faquan Wang. Focus on details: Online multi-object tracking with diverse fine-grained representation. In *CVPR*, pages 11289–11298, 2023. 2

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 2

[39] 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. Springer, 2016. 6

[40] 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

[41] Jenny Seidenschwarz, Guillem Brasó, Víctor Castro Serrano, Ismail Elezi, and Laura Leal-Taixé. Simple cues lead to a strong multi-object tracker. In *CVPR*, pages 13813–13823, 2023. 6, 7

[42] Bing Shuai, Alessandro Bergamo, Uta Buechler, Andrew Berneshawi, Alyssa Boden, and Joseph Tighe. Large scale real-world multi-person tracking. In *ECCV*, pages 504–521. Springer, 2022. 1

[43] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *CVPR*, pages 20993–21002, 2022. 11

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 3

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

[46] Shuai Wang, Hao Sheng, Da Yang, Yang Zhang, Yubin Wu, and Sizhe Wang. Extendable multiple nodes recurrent tracking framework with rtu++. *IEEE TIP*, 31:5257–5271, 2022. 6, 7

[47] Shuai Wang, Da Yang, Yubin Wu, Yang Liu, and Hao Sheng. Tracking game: Self-adaptative agent based multi-object tracking. In *ACM MM*, pages 1964–1972, 2022. 2, 6, 7

[48] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. In *ICRA*, pages 13708–13715. IEEE, 2021. 2

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

[50] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017. 1, 2, 3, 6, 8

[51] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*, pages 2129–2137, 2016. 2

[52] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *WACV*, pages 4799–4808, 2023. 6, 7

[53] Sisi You, Hantao Yao, Bing-Kun Bao, and Changsheng Xu. Utm: A unified multiple object tracking model with identity-aware feature enhancement. In *CVPR*, pages 21876–21886, 2023. 3, 6, 7

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

[55] En Yu, Songtao Liu, Zhuoling Li, Jinrong Yang, Zeming Li, Shoudong Han, and Wenbing Tao. Generalizing multiple object tracking to unseen domains by introducing natural language representation. In *AAAI*, pages 3304–3312, 2023. 1

[56] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *ECCV*, pages 659–675. Springer, 2022. 3, 6

[57] 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*, 129:3069–3087, 2021. 1, 2, 6, 7

[58] 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. Springer, 2022. 1, 2, 3, 6, 7, 8

[59] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Robust multi-object tracking by marginal inference. In *ECCV*, pages 22–40. Springer, 2022. 6, 7

[60] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *CVPR*, pages 22056–22065, 2023. 2, 6, 7

[61] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[62] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, pages 474–490. Springer, 2020. 2, 6