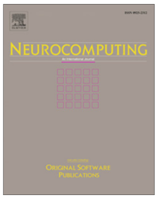
Neurocomputing 483 (2022) 333–347

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Online multi-object tracking with unsupervised re-identification learning and occlusion estimation

Qiankun Liu [a,1], Dongdong Chen [a,2], Qi Chu [a,*], Lu Yuan [b], Bin Liu [a], Lei Zhang [b], Nenghai Yu [a]

[a] *University of Science and Technology of China, Hefei, China*
[b] *Microsoft Cloud AI, Seattle, USA*

ARTICLE INFO

Article history:
Received 10 August 2021
Revised 22 November 2021
Accepted 2 January 2022
Available online 6 January 2022
Communicated by Zidong Wang

Keywords:
Muti-object tracking
Occlusion
Unsupervised learning
Re-identification

ABSTRACT

Occlusion between different objects is a typical challenge in Multi-Object Tracking (MOT), which often leads to inferior tracking results due to the missing detected objects. The common practice in multi-object tracking is re-identifying the missed objects after their reappearance. Though tracking performance can be boosted by the re-identification, the annotation of identity is required to train the model. In addition, such practice of re-identification still can not track those highly occluded objects when they are missed by the detector. In this paper, we focus on online multi-object tracking and design two novel modules, the unsupervised re-identification learning module and the occlusion estimation module, to handle these problems. Specifically, the proposed unsupervised re-identification learning module does not require any (pseudo) identity information nor suffer from the scalability issue. The proposed occlusion estimation module tries to predict the locations where occlusions happen, which are used to estimate the positions of missed objects by the detector. Our study shows that, when applied to state-of-the-art MOT methods, the proposed unsupervised re-identification learning is comparable to supervised re-identification learning, and the tracking performance is further improved by the proposed occlusion estimation module.

1. Introduction

Multi-Object Tracking (MOT) is a fundamental computer vision task with a wide range of applications, including autonomous driving, robot navigation and video analysis. Benefiting from the advance of object detection [16,33,23,57], the tracking-by-detection paradigm has become popular for MOT in the past decade. Though great performance has been achieved recently [53,56,38,54,37], occlusion between objects still remains challenging for MOT.

In MOT scenarios, an object may be missed by the detector due to heavy occlusion, and then reappear after a short while. In order to identify such reappeared objects, re-identification (Re-ID) is often used to associate these reappeared objects with existing tracklets. Most existing MOT works [46,27,2,29] adopt an independent Re-ID model to learn discriminative representations for objects, which introduces extra high computational cost since each object needs to be cropped out and fed into the pre-trained Re-ID model. To achieve real-time tracking, some works try to share the Re-ID feature computation with the backbone in anchor-based detector [28,45] or point-based detector [53] by introducing an extra Re-ID branch that is parallel to the detection branch. Thanks to the sharing of feature maps between different branches, such methods can enable tracking multiple objects in a real-time way.

However, these methods [58,46,45,53] still suffer from the scalability issue in the Re-ID representation learning. For example, [58,45,53] combine several existing tracking and human detection (or Re-ID) datasets together and then learn the Re-ID representation by classifying each identity appeared in the combined dataset as one class (pseudo identity label). Such classification methods may work well for small datasets, but will encounter the learning difficulty when the identity number is huge, because the dimension of the sibling classification layer (fully connected layer) is linearly proportional to the identity number. More importantly, such supervised Re-ID module learning requires the annotation of identities, which is highly expensive and unscalable.

To address this problem, we first propose a new Re-ID module learning mechanism. It adopts an unsupervised matching based

[*] Corresponding author.
E-mail addresses: liuqk3@mail.ustc.edu.cn (Q. Liu), cddlyf@gmail.com (D. Chen), qchu@ustc.edu.cn (Q. Chu), luyuan@microsoft.com (L. Yuan), flowice@ustc.edu.cn (B. Liu), leizhang@microsoft.com (L. Zhang), ynh@ustc.edu.cn (N. Yu).
[1] Equal contribution
[2] Equal contribution

https://doi.org/10.1016/j.neucom.2022.01.008
0925-2312/© 2022 Elsevier B.V. All rights reserved.

loss between two frames (images) rather than the supervised classification loss used in [58,45,53]. This is based on the observation that objects with the same identity in adjacent frames share the similar appearance and objects in different scenes (or within the same image) have different identities and appearances, which is shown in Fig. 1 (a). Compared to the aforementioned methods, this newly proposed unsupervised Re-ID learning mechanism has two merits: 1) it does not need any (pseudo) identity annotation; 2) The matching based loss is irrelevant to the number of identities, thus can be directly trained on massive video-based data that with large number of identities. In addition, the image-based data can also be used for training if we treat two augmentations of one image as the adjacent frames.

Though the Re-ID module can re-identify the reappeared objects after their short-term disappearance, how to proactively track the objects with highly occlusion is still challenging. This is because the severely occluded objects are easily missed by the detector, as shown in Fig. 1 (b). For example, in anchor-based detectors [33,18], the Non-Maximum Suppression (NMS) module will remove highly overlapped boxes. In point-based detectors [23,57], as the object centers are invisible for occluded objects, it is also difficult to learn reliable center point-based features. Recently, how to address the missing detection issue caused by occlusion has attracted lots of attention. Some initial attempts [58,6,10,59] emerge, including detecting visible parts of an object [58,6], using one proposal for multi-prediction [10], and using paired anchors for one detection [59]. .

Different from existing methods, we propose a novel occlusion estimation module to predict whether two objects are occluded. Specifically, an occlusion map which shows all possible occlusion locations in the current frame is predicted. By further combing the status of existing tracklets, we finally design a lost object refinding mechanism to find the occluded objects back.

To evaluate the effectiveness of the above two modules, we conduct extensive experiments by integrating them with different existing state-of-the-art MOT methods. For example, by replacing the supervised classification based Re-ID module in FairMOT [53], the unsupervised Re-ID learning module can still achieve comparable results on the MOT Challenge datasets[31,11] but neither needs any identity annotation nor suffers from any scalability issue. By integrating the occlusion estimation module, both FairMOT [53] and CenterTrack [56] can handle the occlusion better and achieve the performance gain.

To summarize, our contributions are threefold as below:

- We propose a novel unsupervised Re-ID learning module without using any identity information. It can be trained on video-/ image-based data, and also has better scalability to datasets that with massive identities.
- We propose a new occlusion estimation module, which can effectively recognize and track occluded objects when they are missed by the detector by estimating the occlusion location.
- Both the unsupervised Re-ID learning and occlusion module can be applied to existing MOT methods in a natural way. Experimental results demonstrate the effectiveness of the proposed method.

The rest of this paper is organized as follows. In Section 2, we review some related works in terms of MOT, person re-identification, and occlusion handling. Then in Section 3, we elaborate the details of the two newly proposed modules, and the designed lost object refinding mechanism. To demonstrate the effectiveness, extensive experiment and ablation analysis are conducted in Section 4. Finally, we conclude our work in Section 5.

2. Related work

In this section, we first provide a brief overview about the popular tracking-by-detection paradigm for MOT, and then introduce the re-identification for data association in MOT as well as existing occlusion handling mechanisms in object detection and tracking.

2.1. Tracking-by-detection

Most existing MOT frameworks follow a tracking-by-detection paradigm thanks to the advances of object detectors [33,18,23,57]. Specifically, an object detector is used to detect objects in each frame, then a subsequent tracker is utilized to associate the objects across different frames. In terms of temporal information usage, existing MOT methods can be categorized into online [46,2,27,56,53] and offline methods [5,19]. Online methods process video sequences frame-by-frame and track objects by only using information up to the current frame. By contrast, offline methods process video sequences in a batch and can even utilize the whole video information. From the network structure perspective, they can be further categorized into separate modeling

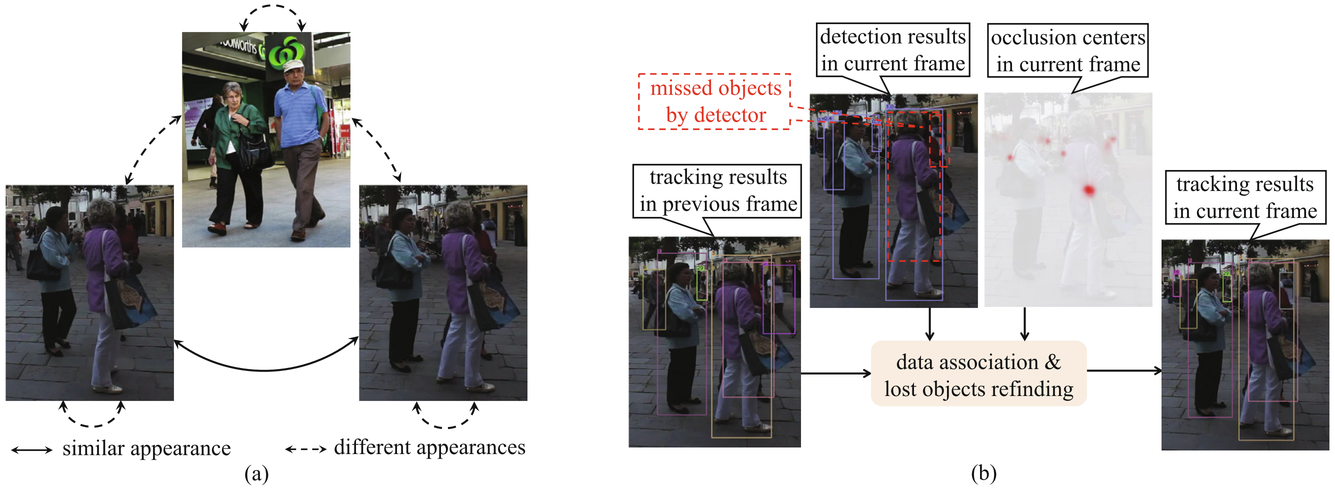

similar appearance
different appearances
(a)
missed objects by detector
detection results in current frame
occlusion centers in current frame
tracking results in previous frame
tracking results in current frame
data association & lost objects refinding
(b)

Fig. 1. (a) The left and right are two adjacent frames from a video while the middle is an image from another scene. An apparent observation is that objects with the same identity in adjacent frames share the similar appearance and objects in different scenes (or within the same image) have different identities and appearances. (b): The left and right are tracking results in previous and current frames, and the middle are the detected objects and occlusion centers in current frame. Some lost objects that are missed by detector can be tracked with the help of predicted occlusion centers. Please refer to Section 3.2.2 for more details about lost ob.jects refinding.

[46,2,27,5,19] and joint modeling methods [56,53,45]. In separate modeling methods, the tracker is independently trained and assumes the detection results are available in advance. In joint modeling methods, the tracker is jointly trained with the detector by sharing the same feature extractor backbone. Therefore, they are often more efficient than the separate modeling methods. Both the newly proposed Re-ID module and occlusion estimation module can be naturally integrated into the online tracking-by-detection MOT system and jointly learned with the detector.

2.2. Re-identification for association

Learning discriminative representations for objects is crucial to identity association in tracking. The representation can be used to re-identify lost objects that reappear after disappearing for a while. Early methods [46,2,27,1] crop the image patch of a detected object, resize and feed it into a separate Re-ID model. It is inevitably time-consuming since the feature representation of different objects has to be computed independently. To reduce the computation, some works attempt to share the Re-ID feature computation with the backbone in anchor-based detector [28,45,42,32] or point-based detector [53] by introducing an extra Re-ID branch that is parallel to detection branch. The common practice in MOT to train the Re-ID module is to classify each identity into one class [58,20,53,45]. There are two fundamental weaknesses of such methods: 1) the Re-ID module is less scalable especially when the amount of identities is huge, because the classifier takes up a lot of memory. For example, FairMOT [53] performs about 339 K classification task to train the Re-ID module. 2) the training of Re-ID module needs to be supervised by identity information. For example, several datasets dedicated for Re-ID are adopted in [53,45]. However, the acquisition of well annotated data costs a lot.

Despite the advance in supervised Re-ID learning [25,39,26], some works for unsupervised Re-ID learning have been proposed [44,20,53,47,15]. These works can be divided into two categories: pseudo identity based [20,53,47,15] and identity free methods [44]. The proposed method is also an identity free method. For the former category, pseudo identities can be obtained by clustering [47,15] or tracking [20]. However, the errors may accumulate and it is challenging to estimate the number of pseudo identities while clustering, and a trajectory of an object breaks into several short trajectories easily while tracking. For the latter category, the correspondence between adjacent frames is used [44]. However, the birth and death of objects are not handled and the relation between objects within one frame is also not exploited. Inspired by these works, we propose to learn Re-ID representation in an unsupervised and matching based loss without using any (pseudo) identity information. It is built upon the observation that objects with the same identity in adjacent frames share the similar appearance and objects in different scenes (or within the same image) have different identities and appearances. Different from the works in [44], our method 1) introduces a placeholder to handle the birth and death of objects and 2) makes usage of the information that objects in different scenes (or within the same image) have different identities. Besides, since our matching based loss is irrelevant to the number of identities, it does not suffer from the scalability issue and can be directly trained on massive data.

2.3. Occlusions between objects

Handling highly occluded objects is a challenging task for both detection [10,59,6] and tracking [8,58,7,9,50,27]. Some anchor-based methods are proposed to handle occlusion [10,59,6] in detection, including predicting multiple instances for one proposal [10] or detecting one pedestrian with a pair of anchors [59] (one is for head and another is for full body). However, both works require a carefully designed NMS algorithm for post processing. In [6], a mask-guided module is proposed to force the detector to pay attention to the more visible head part and thus detect the whole body of a pedestrian. Different from these methods, we propose a new occlusion estimation mechanism upon key-point based detection, which detects the locations where occlusions happen and finds the missed objects caused by detector by combining the tracking status of existing tracklets.

Instead of handling occlusions in the detection stage, there also exist some methods [8,58,7,9,50,27] attempting to handle occlusions in the tracking stage for MOT task. The works in [8,9,58,7] utilize the single object tracking (SOT) method for MOT. In details, a SOT tracker is created and maintained for each object. Once an object is heavily occluded and missed by the detector, the position of it could be estimated by the corresponding SOT tracker. Such practice indeed is an extra detection stage with dedicated detectors (i.e., SOT trackers). The topology between different objects is also exploited to handle occlusions for MOT [50,27]. The hypothesis is that the topology between different objects in adjacent frames is invariant, which is positive to the association of objects, especially when some objects are partially occluded. In addition, the position of a lost object is estimated using the positions of its tracked neighbors [27] based on the topology among them in previous frame. Different from these works, our method detects the locations of occlusion in a frame, and utilizes them to refind the missed objects while tracking online. More detail, if an object is missed by the detector, then it is likely to be heavily occluded by other objects. The detected occlusion locations could be used as the prior information to find it back.

Occlusion is also one critical issue in SOT task [41,13]. Trying to find the visible region of the single object is the main focus in SOT. However, our method detects the overlapped region between different objects.

3. Method

As mentioned above, our paper proposes two key modules for existing multiple object tracking systems. One is the unsupervised Re-ID module learning mechanism, which is competitive to existing supervised counterparts and has better scalability. Another is the occlusion estimation module, which predicts occlusion map to find the occluded objects back. In this section, we first elaborate the details of these two modules, and then show how to naturally integrate them with existing MOT systems.

3.1. Unsupervised Re-ID learning

The proposed unsupervised Re-ID learning can be trained on both video-based and image-based data. Note that it is only used in the training stage. Once the training procedure is finished, the trained models can be directly used to extract discriminative features for different objects, which is the same as existing supervised counterparts. For better understanding, we start with the learning from video-based data, then illustrate learning from image-based data.

3.1.1. Learning from video-based data

Let $I^t \in \mathbb{R}^{W\times H\times 3}$ be the t-th frame from one video and $\mathbf{b}_i^t = (x_{i_l}^t, y_{i_t}^t, x_{i_r}^t, y_{i_b}^t)$ be the groundtruth bounding box of object i in frame I^t. W, H are the width and height of the frame, and $(x_{i_l}^t, y_{i_t}^t), (x_{i_r}^t, y_{i_b}^t)$ are the coordinates of the top-left and bottom-right corners respectively. For Re-ID representation learning, we denote the appearance feature for object i in frame I^t as $f_i^t \in \mathbb{R}^D$, where D is the dimension of the appearance feature vector. Our unsupervised Re-ID representation learning mechanism is general in how f_i^t is calculated as long as it is differentiable. Possible solutions include cropping the image patch based on the given bounding box and feeding the cropped image patch into an extra Re-ID network like [44,2,27], extracting the ROI based appearance features by sharing the same backbone as the detector like [45,42], and extracting the center point based appearance features like [53]. The usage of annotated bounding boxes is equivalent to traditional Re-ID task in which the well cropped image patches are provided. Even though, no identity information is used in the proposed method. In the following, the superscript t of appearance feature is omitted for simplicity.

Given two adjacent frames I^{t-1} and I^t, let $i \in \{0, \ldots, N^{t-1}-1, \ldots, N^{t-1}+N^t-1\}$ be the index of all objects in both two frames, where N^t is the number of objects in frame I^t. The first N^{t-1} objects are from frame I^{t-1} and the rest are from frame I^t. We have two observations: 1) objects in the same frame have different identities; 2) an object is likely to appear in both adjacent frames. Accordingly, as shown in Fig. 2, if we want to assign an object to another object, two types of supervision signals can be exploited: 1) objects within the same frame should not be matched with each other, which is regarded as strong supervision signal; 2) objects in one frame are likely to be matched with objects in another adjacent frame, which is weak supervision signal. In order to learn the Re-ID representation with such supervision signals, we first define a similarity matrix $S \in \mathbb{R}^{(N^{t-1}+N^t)\times(N^{t-1}+N^t)}$ that measures the similarity between each pair of objects, where:

$$S_{i,j} = \begin{cases} \frac{f_i \cdot f_j}{\|f_i\|_2 \|f_j\|_2} & \text{if } i \neq j, \\ -\infty & \text{otherwise}. \end{cases} \tag{1}$$

Obviously, $S_{i,j} = S_{j,i}$. The values in the diagonal of S are set to negative infinity to avoid assigning an object to itself (Eq. (2)). In general, if objects i and j share the same identity, $S_{i,j} > 0$, otherwise, $S_{i,j} < 0$. The assignment matrix $M \in \mathbb{R}^{(N^{t-1}+N^t)\times(N^{t-1}+N^t)}$ can be obtained by applying row-wise softmax function to S as:

$$M_{i,j} = \frac{e^{S_{i,j}T}}{\sum_j e^{S_{i,j}T}}, \tag{2}$$

where T is the temperature of the softmax function. Consider the fact that the number of objects in adjacent frames (i.e., the size of S) could be various, we follow the works in CycAs [44] and set $T = 2\log(C+1)$, where $C = N^{t-1} + N^t$ is the number of columns in S. With this adaptive temperature, the maximum values in each row are almost equally highlighted/maximized even the size of S varies. Since objects in the same frame have different identities, we can supervise the values in the top-left and right-bottom part of M by a intra-frame loss:

$$L_{id}^{intra} = \sum_{0 \leqslant i,j < N^{t-1}} M_{i,j} + \sum_{N^{t-1} \leqslant i,j < N^{t-1}+N^t} M_{i,j}. \tag{3}$$

This corresponds to the aforementioned strong supervision signal. To leverage the weak supervision signal, we first consider the ideal case where all the objects appear in both frames for better understanding. In this case, all the objects in the frame I^{t-1} should be matched to the objects in the frame I^t in a one-to-one manner. Then for each row in M, we encourage each object to be matched to another object with a high confidence by using the below inter-frame margin loss:

$$\begin{aligned} L_{id}^{inter} &= \sum_i \max\{\max_{j', j' \neq j^*} M_{i,j'} + m - M_{i,j^*}, 0\}, \\ &\text{where} \quad j^* = \operatorname*{argmax}_j M_{i,j}. \end{aligned} \tag{4}$$

This shares a similar spirit as the popular triple loss, i.e., the maximum matching probability M_{i,j^*} is larger than the sub-maximum value by a pre-defined margin m (0.5 by default).

Besides the above margin loss, we further add another cycle constraint loss L_{id}^{cycle} for M, which means the forward and backward assignment should be consistent with each other. In details, if an object i in frame I^{t-1} is matched with object j in frame I^t, then the object j in frame I^t must be matched with object i in frame I^{t-1}:

$$L_{id}^{cycle} = \sum_{N^{t-1} \leqslant i < N^{t-1}+N^t, 0 \leqslant j < N^{t-1}} |M_{i,j} - M_{j,i}|. \tag{5}$$

Since two adjacent frames in video-based data often share some objects with the same identities, we call such two adjacent frames a positive sample for the Re-ID module training. The total loss for unsupervised Re-ID learning on such positive samples is:

$$L_{id}^{pos} = \frac{1}{N^{t-1}+N^t}(L_{id}^{intra} + L_{id}^{inter} + L_{id}^{cycle}). \tag{6}$$

Unlike the above ideal case, an object in frame I^{t-1} may disappear in frame I^t (death of objects) and an object may appear in

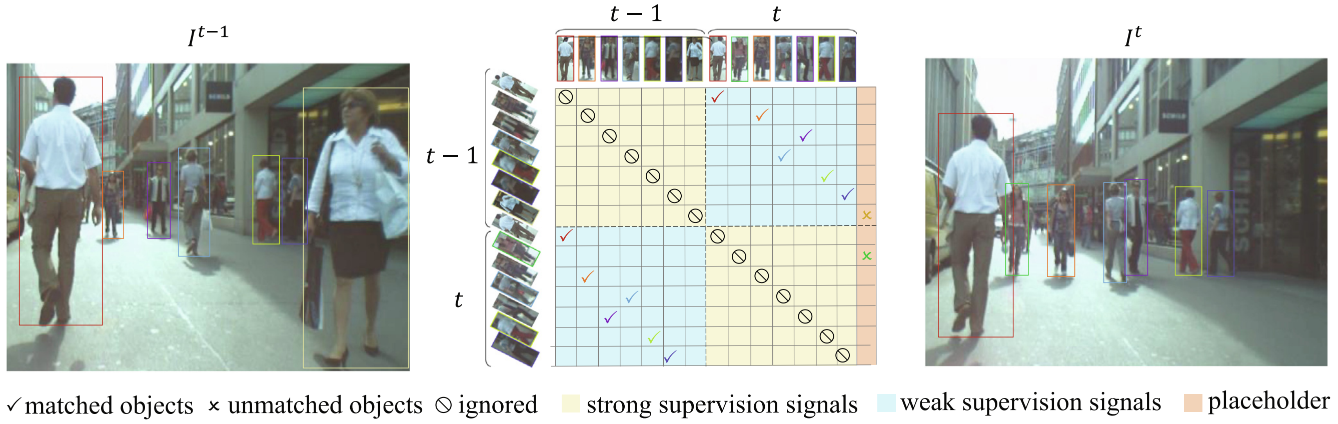

I^{t-1}
$t-1$
t
$t-1$
t
I^t
✓ matched objects × unmatched objects ⦸ ignored
strong supervision signals
weak supervision signals
placeholder

Fig. 2. The proposed un-supervised Re-ID learning method. Left and right are the two adjacent frames and the objects. Middle is the desired assignment results. For a better viewing, the identities of objects are encoded by color. However, the identity information is unused in our method. Two types of supervision signals are exploited. 1) Strong supervision signals: objects within the same frame should not be matched with each other. 2) Weak supervision signals: objects in one frame are likely to be matched with objects in another frame.

frame I^t for the first time but invisible in frame I^{t-1} (birth of objects) in a general case. However, for each row in assignment matrix M, the inter-frame margin loss L_{id}^{inter} will force the maximum value to be larger than the other values by a margin m, which is unsuitable when the corresponding object is disappeared or newly appeared since it does not share the same identity with any one of the other objects. To handle this issue, a new similarity matrix $S' \in \mathbb{R}^{(N^{t-1}+N^t)\times(N^{t-1}+N^t+1)}$ is obtained by padding a placeholder column to S. All values in the padded placeholder column are the same, which is denoted as p. The detailed discussion on p is presented in Experiments Section 4.3.1. With the existence of placeholder column, the similarity scores between disappeared/newly appeared objects and other objects are encouraged to be learned smaller than p. Let $M' \in \mathbb{R}^{(N^{t-1}+N^t)\times(N^{t-1}+N^t+1)}$ be the assignment matrix by applying row-wise softmax function to S' [3]. Then we replace M with M' in Eq. (3), Eq. (4) and Eq. (5) for this general case. In our implementation, the loss for this general case is adopted.

3.1.2. Learning from image-based data

To train the proposed Re-ID module on image-based data, a straightforward way is to get two augmentations of one image and treat these two augmentations as adjacent frames like the video-based data. However, we find only using the above positive sample based loss does not perform very well, since objects in the two augmentations have very similar appearance, thus not strong enough in learning discriminative Re-ID features. Considering the fact that objects in two different static images usually have different identities, we further introduce a negative sample based loss L_{id}^{neg} by treating two different static images from different scenes as a negative sample pair:

$$L_{id}^{neg} = \sum_{0 \leqslant i,j < N^{t-1}+N^t} M'_{i,j}. \tag{7}$$

Similarly, in this formulation, we introduce the extra placeholder p and encourage the cosine distance between the objects in the negative pair to be less than p, which also means that all objects should be assigned to the placeholder. Note that the design of L_{id}^{neg} shares the same spirit with the intra-frame loss L_{id}^{intra}, while the inter-frame margin loss L_{id}^{inter} and cycle constraint loss L_{id}^{cycle} are not used for negative sample pairs.

Therefore, the overall unsupervised Re-ID learning loss for the image based data is:

$$L_{id} = \frac{N^{pos}}{N^{pos}+N^{neg}} L_{id}^{pos} + \frac{N^{neg}}{N^{pos}+N^{neg}} L_{id}^{neg}, \tag{8}$$

where N^{pos} and N^{neg} are the number of positive and negative samples in a batch. In our default setting, $\frac{N^{neg}}{N^{pos}}$ is set to 0.25.

Although the Re-ID module can help re-identify reappeared objects after their short-term disappearance, it is inherently unable to track the occluded objects if they are not detected by the detector. To mitigate the issue caused by the missed detection, we propose an occlusion estimation module to predict whether any occlusion occurs and find lost objects back by combining the predicted occlusions and the tracking status of existing tracklets.

3.2. Occlusion estimation module

3.2.1. Occlusion detection

Inspired by the work of key-point estimation [23,57], the locations of occlusion are treated as key-points and detected by key-point estimation.

Different from the above Re-ID module, the learning of the occlusion estimation module is designed in a supervised way. We automatically generate occlusion annotation based on the bounding boxes of objects, which are available in existing tracking datasets like MOT16 and MOT17 [31].

First, we need to define when an occlusion occurs. Given the bounding box coordinates of two objects i and j within one frame, their overlapped region is defined as $\mathbf{o}_{ij} = \mathcal{O}(\mathbf{b}_i, \mathbf{b}_j) = (x_{ij_l}, y_{ij_t}, x_{ij_r}, y_{ij_b})$. Considering two typical occlusion examples as shown in Fig. 3, we define an indicator function $\mathcal{H}(\cdot)$ that indicates whether an occlusion is valid or not. Only when the overlapped region occupies a large portion of object i or j, the occlusion $\mathbf{o}_{ij}$ is valid. Specifically:

$$\mathcal{H}(\mathbf{o}_{ij}) = \begin{cases} 1 & if \ \frac{\mathcal{A}(\mathbf{o}_{ij})}{\min(\mathcal{A}(\mathbf{b}_i), \mathcal{A}(\mathbf{b}_j))} > \tau, \\ 0 & else, \end{cases} \tag{9}$$

where $\mathcal{A}(\cdot)$ is the function computing the area of a box, and τ is a hyper-parameter which is set as 0.7 by default. In order to refind an occluded object back (Section 3.2.2), we define the center point of $\mathbf{o}_{ij}$ as the occlusion location of two overlapped objects. The groundtruth occlusion map Y is rendered by a 2D Gaussian kernel function based on all the valid occlusions defined in Eq. (9) as:

$$Y_{x,y} = \max_{ij} \mathcal{G}(\mathbf{o}_{ij}, (x,y)), \ \text{subject to} \mathcal{H}(\mathbf{o}_{ij}) = 1, \tag{10}$$

where $\mathcal{G}(\mathbf{o}_{ij}, (x,y)) = \exp(-\frac{((x,y)-\lfloor \frac{\mathbf{p}_{ij}}{R} \rfloor)^2}{2\sigma^2_{\mathbf{o}_{ij}}})$ is the Gaussian kernel function, and $\mathbf{p}_{ij} = (\frac{x_{ij_l}+x_{ij_r}}{2}, \frac{y_{ij_t}+y_{ij_b}}{2})$ is the center point of occlusion $\mathbf{o}_{ij}$. The standard deviation $\sigma_{\mathbf{o}_{ij}}$ of the Gaussian kernel is set to be relative to the size of $\mathbf{o}_{ij}$ following the definition in [23]. In our implementation, we introduce an extra CNN head to obtain the predicted occlusion center heatmap $\hat{Y} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R}}$. It is parallel to the detection head and shares the same backbone network. R is the downsampling factor of the backbone network. Intuitively, the value $\hat{Y}_{x,y} \in [0,1]$ denotes the probability of an occlusion center that locates in (x,y) and is supervised by:

$$L_{occ}^{cen} = \sum_{x,y} \mathcal{L}(Y_{x,y}, \hat{Y}_{x,y}), \tag{11}$$

where $\mathcal{L}(\cdot,\cdot)$ is a variant of focal loss function used in [23] with two hyper-parameters α, β (default values are 2 and 4 respectively):

$$\mathcal{L}(y, \hat{y}) = \begin{cases} -(1-\hat{y})^\alpha \log(\hat{y}) & if \ y = 1, \\ -(1-y)^\beta (\hat{y})^\alpha \log(1-\hat{y}) & else. \end{cases} \tag{12}$$

Considering that R is often larger than 1, we take the inspiration from [57] and add another CNN head to produce an offset heatmap $\hat{\Lambda} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$, which can help compensate the quantization error in generating the occlusion center heatmap Y. The simple L1 loss is used to regress the center offset:

$$L_{occ}^{off} = \sum_{ij} |\hat{\Lambda}_{\lfloor \frac{\mathbf{p}_{ij}}{R} \rfloor} - (\frac{\mathbf{p}_{ij}}{R} - \lfloor \frac{\mathbf{p}_{ij}}{R} \rfloor)|. \tag{13}$$

Fig. 3. Typical occlusion cases. Translucent blue areas denote the positions where occlusions happen and red circles are the occlusion centers. Left: A small box covered by a larger box. Right: Two boxes overlapped with each other.

[3] In this case, $C = N^{t-1} + N^t + 1$ for the calculation of temperature T.

Need to note that the offset supervision is only given at the center locations. The overall occlusion estimation loss is:

$$L_{occ} = \frac{1}{\sum_{i,j}\mathscr{H}(\mathbf{o}_{ij})}(L^{cen}_{occ} + L^{off}_{occ}). \tag{14}$$

3.2.2. Lost object refinding

While tracking online, the occlusion estimation module is used to detect the possible occlusion locations, i.e., the center points of overlapped regions between different objects in a frame. For severely occluded objects, they are easily missed by the detector (thus lost by the tracker). In such case, the corresponding occlusion locations can be used as the prior information to refind them. Specifically, given the set of existing tracklets in frame $t-1$ and the set of newly detected objects in frame t, we match the newly detected objects with existing tracklets. If some tracklets cannot match with any newly detected objects, we treat them as potential lost tracklets/objects and try to find them back. The detailed tracking logic is elaborated in Algorithm 1. Through the refinding of lost objects, the number of false negative objects could be reduced, leading to a higher tracking performance.

Once there exist some potential lost objects, we propose to find the lost objects back by using the predicted occlusion locations and the motion information of the corresponding tracklets, which can be estimated by Kalman filter. In details, suppose we want to refind the lost object i in I^t, its bounding box in I^{t-1} is denoted as $\mathbf{b}_i^{t-1} = (x^{t-1}_{i_l}, y^{t-1}_{i_t}, x^{t-1}_{i_r}, y^{t-1}_{i_b})$. We first predict its location at I^t via Kalman filter and denote the location as $\tilde{\mathbf{b}}_i^t = (\tilde{x}^t_{i_l}, \tilde{y}^t_{i_t}, \tilde{x}^t_{i_r}, \tilde{y}^t_{i_b})$. Then we search all the detected objects that possibly occlude i by considering the estimated occlusion centers close to $\tilde{\mathbf{b}}_i^t$. The detailed search process is illustrated in Fig. 4. Specifically, for each box $\mathbf{b}_j^t$ that possibly overlapped with $\tilde{\mathbf{b}}_i^t$, we first calculate the overlapped region as $\tilde{\mathbf{o}}_{ij} = \mathscr{O}(\tilde{\mathbf{b}}_i^t, \mathbf{b}_j^t)$. Then we get a score between $\tilde{\mathbf{o}}_{ij}$ and one of the predicted occlusion centers $\hat{\mathbf{p}}^t_{ik} = (\hat{x}^t_{ik}, \hat{y}^t_{ik})$ that locates within $\tilde{\mathbf{b}}_i^t$ using the aforementioned Gaussian kernel function. Finally, we choose the best matched pair by $(j', k') = \mathrm{argmax}_{j,k}\mathscr{G}(\tilde{\mathbf{o}}^t_{ij}, \hat{\mathbf{p}}^t_{ik})$. If $\mathscr{G}(\tilde{\mathbf{o}}^t_{ij'}, \hat{\mathbf{p}}^t_{ik'}) > \tau_o$ ($\tau_o = 0.7$ by default), then object i is likely to be occluded by object j', leading to missing detection. Suppose that the box $\mathbf{b}^t_{j'}$ and occlusion center $\hat{\mathbf{p}}^t_{ik'}$ are all correctly estimated and the size of object i in adjacent frames keeps unchanged, the estimated box $\mathbf{b}_i^t$ for object i can be calculated as:

$$\begin{cases} x^t_{i_l} = \mathscr{F}(\tilde{x}^t_{i_l}, \tilde{x}^t_{i_r}, x^t_{j'_l}, x^t_{j'_r}, \hat{x}^t_{ik'}), \\ y^t_{i_t} = \mathscr{F}(\tilde{y}^t_{i_t}, \tilde{y}^t_{i_b}, y^t_{j'_t}, y^t_{j'_b}, \hat{y}^t_{ik'}), \\ x^t_{i_r} = x^t_{i_l} + \tilde{x}^t_{i_r} - \tilde{x}^t_{i_l}, \\ y^t_{i_b} = y^t_{i_t} + \tilde{y}^t_{i_b} - \tilde{y}^t_{i_t}, \end{cases} \tag{15}$$

where $\mathscr{F}(a_1, a_2, b_1, b_2, z)$=

$$\begin{cases} 2z - b_1 - (a_2 - a_1) & \textit{if } a_1 \leqslant b_1 \textit{ and } a_2 \leqslant b_2, \\ z - (a_2 - a_1)/2 & \textit{if } a_1 > b_1 \textit{ and } a_2 \leqslant b_2, \\ 2z - b_2 & \textit{if } a_1 > b_1 \textit{ and } a_2 > b_2, \\ a_1 & \textit{else}. \end{cases} \tag{16}$$

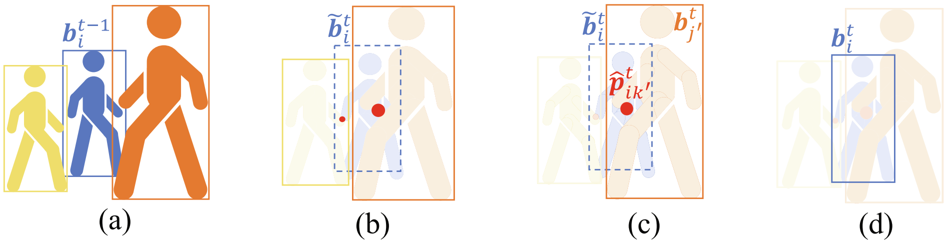


Fig. 4. Illustration of lost object refinding. (a): Tracking results in the previous frame. (b): Predicted box $\tilde{\mathbf{b}}_i^t$ from $\mathbf{b}_i^{t-1}$ via motion, the overlapped boxes with $\tilde{\mathbf{b}}_i^t$, and the predicted occlusion center points that locate within $\tilde{\mathbf{b}}_i^t$. (c): The chosen occlusion center $\hat{\mathbf{p}}^t_{ik'}$ and box $\mathbf{b}^t_{j'}$ used for recovering $\mathbf{b}_i^t$. (d): Recovered box $\mathbf{b}_i^t$ for the lost object i.

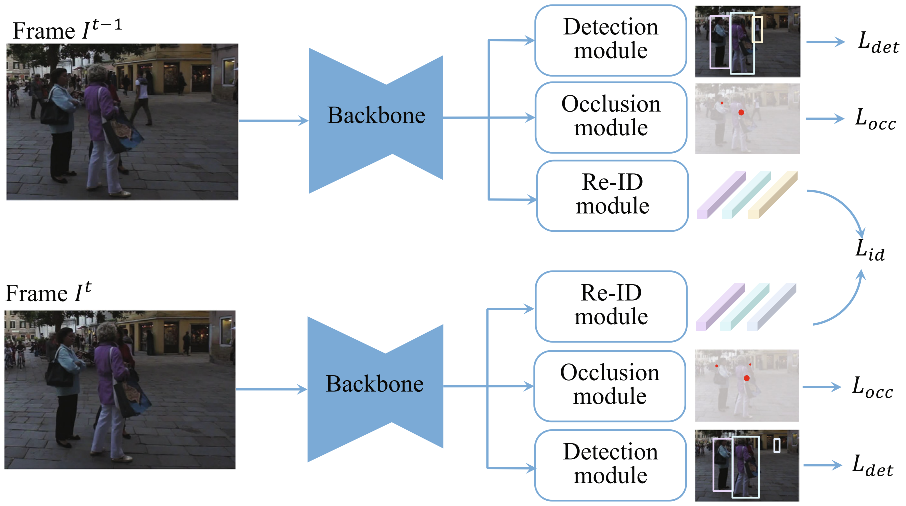


Fig. 5. Illustration of applying of our unsupervised Re-ID module and occlusion module to FairMOT [53]. While integrating, the occlusion module is added to be parallel with detection module which remains unchanged. Re-ID features from two frames are needed to train t.he Re-ID module.

Algorithm 1: Tracking logic between Two Consecutive Frames

```
Input  : $T^{t-1} = \{(\boldsymbol{b}_i^{t-1}, id_i, w_i, \boldsymbol{p}_i^{t-1})\}_{i=1}^{N^{t-1}}$: cached trackltes in frame $t-1$ with
         box $\boldsymbol{b}_i^{t-1}$, identity $id_i$, number of consecutive frames $w_i$ being lost, center
         point $\boldsymbol{p}_i^{t-1}$.
         $D^t = \{(\boldsymbol{b}_j^t, \boldsymbol{p}_j^t)\}_{j=1}^{N^t}$: detected objects in frame $t$ with box $\boldsymbol{b}_j^t$, center point
         $\boldsymbol{p}_j^t$.
         $\hat{P}^t = \{\hat{\boldsymbol{p}}_k^t\}_{k=1}^{N_{occ}^t}$: predicted occlusion center points in frame $t$
Output: $T^t$: cached tracklets in frame $t$.
         $B^t$: tracking results in frame $t$.
Step1: Initialize as empty set
    $T^t \leftarrow \emptyset$, $B^t \leftarrow \emptyset$
Step2: Assign objects to tracklets, and get the assigned index pairs, lost tracklet
  and unassigned object indices
  $\{(i_a, j_a)\}_{a=1}^{A}, \{i_l\}_{l=1}^{L}, \{j_u\}_{u=1}^{U} \leftarrow \textsc{Assign}(T^{t-1}, D^t)$
Step3: Update tracklets with assigned objects
  for $(i_a, j_a) \in \{(i_a, j_a)\}_{a=1}^{A}$ do
    $w_{j_a} \leftarrow 0$, $id_{j_a} \leftarrow id_{i_a}$
    $T^t \leftarrow T^t \cup \{(\boldsymbol{b}_{j_a}^t, id_{j_a}, w_{j_a}, \boldsymbol{p}_{j_a}^t)\}$
    $B^t \leftarrow B^t \cup \{\boldsymbol{b}_{j_a}^t, id_{j_a}\}$
Step4: Initialize new tracklets with unassigned objects
  for $j_u \in \{j_u\}_{u=1}^{U}$ do
    $w_{j_u} \leftarrow 0$, $id_{j_u} \leftarrow \textsc{NewID}()$
    $T^t \leftarrow T^t \cup \{(\boldsymbol{b}_{j_u}^t, id_{j_u}, w_{j_u}, \boldsymbol{p}_{j_u}^t)\}$
    $B^t \leftarrow B^t \cup \{\boldsymbol{b}_{j_u}^t, id_{j_u}\}$
Step5: Handle lost tracklets
  for $i_l \in \{i_l\}_{l=1}^{L}$ do
    if $w_{i_l} < \tau_w$ then /*$\tau_w$ is the time window threshold*/
      /* select the occlusion point and box in current frame to recover the box for
        lost tracklet */
      $j', k' = \text{argmax}_{j,k} \mathcal{G}(\mathcal{O}(\tilde{\boldsymbol{b}}_{i_l}^t, \boldsymbol{b}_j^t), \hat{\boldsymbol{p}}_k^t)$
      if $\mathcal{G}(\mathcal{O}(\tilde{\boldsymbol{b}}_{i_l}^t, \boldsymbol{b}_{j'}^t), \hat{\boldsymbol{p}}_{k'}^t) > \tau_o$ then
        $\boldsymbol{b}_{i_l}^t \leftarrow$ get occluded object box based on Eq. (15)
        $T^t \leftarrow T^t \cup \{(\boldsymbol{b}_{i_l}^t, id_{i_l}, w_{i_l}, \boldsymbol{p}_{i_l}^t)\}$
        $B^t \leftarrow B^t \cup \{\boldsymbol{b}_{i_l}^t, id_{i_l}\}$
      else
        $w_{i_l} \leftarrow w_{i_l} + 1$
        $T^t \leftarrow T^t \cup \{(\tilde{\boldsymbol{b}}_{i_l}^t, id_{i_l}, w_{i_l}, \boldsymbol{p}_{i_l}^t)\}$
Return $T^t$, $B^t$.
```

3.3. Integration into existing methods

The above two modules can be naturally integrated into existing state-of-the-art MOT systems, such as [45,53,56,27]. In details, as long as the original MOT system has or is able to add the differentiable Re-ID feature learning part, the proposed Re-ID learning mechanism can be applied into it and allows large scale unsupervised Re-ID learning. For the occlusion estimation module, it is compatible with MOT systems that are equipped with the modern CNN detector. We can simply implement it by adding the occlusion estimation module as a parallel head to the original detection head and sharing the same CNN backbone.

In Fig. 5, we take the popular tracking framework FairMOT [53] as an example and show the integrated framework. The original FairMOT has one point based detection module, and one supervised Re-ID module that is learnt by classifying each identity as one independent class, which needs costly Re-ID annotation and suffers from the aforementioned dimension explosion problem for huge identity number. We integrate the two proposed modules by changing its Re-ID learning mechanism and adding the occlusion estimation module as described above. In the following experiments, besides FairMOT, we also try to integrate our modules into CenterTrack [56]. Since CenterTrack does not have the Re-ID fea-

Table 1
Tracking results on the MOT17 validation set with respect to different Re-ID learning methods. FairMOT^2 and FairMOT$_{w/o}$ mean the results with or without the original supervised Re-ID method used in FairMOT. ↑ means the larger the better and ↓ means the smaller the better. Best results are shown in **bold** and highlighted with underline.

Trackers	MOTA↑	IDF1 ↑	MT↑	ML↓	FP↓	FN ↓	IDS↓
FairMOT$_{w/o}$	65.8%	61.0%	133	62	**2620**	14735	1098
FairMOT^2 [53]	67.5%	70.2%	134	**55**	2814	**14263**	**492**
FairMOT + CysAs [44]	67.0%	70.8%	134	60	2631	14681	503
FairMOT + UTrack	**67.6%**	**71.8%**	**137**	63	2621	14388	503

ture learning module, we only incorporate the occlusion estimation module into it.

4. Experiments

4.1. Implementation details

The state-of-the-art methods FairMOT [53] and CenterTrack [56] are both implemented based on the key-point based detector CenterNet [57]. We integrate the proposed modules into them to demonstrate the effectiveness. The occlusion loss L_{occ} is added to the detection loss of FairMOT and CenterTrack with the weight of 0.5. The estimation branch for occlusion centers and offsets in the occlusion estimation module both consists of one 3×3 convolutional layer whose output is a 256-channel feature map and one 1×1 convolutional layer that produces the task-specific heatmap. Between these two layers, a ReLU activation function is adopted. For the occlusion center branch, the output heatmap $\hat{Y}$ is activated by the sigmoid function, while for the occlusion offset heatmap $\hat{\Lambda}$, no activation function is adopted. When replacing the supervised Re-ID learning in FairMOT [53] with our unsupervised Re-ID learning, we directly substitute the original Re-ID loss in FairMOT with the loss L_{id} in Eq. (8) while keeping other unchanged. The dimension D of Re-ID feature is set to 256.

By default, the Adam optimizer [22] with the initial learning rate $1e-4$ is used. The models in FairMOT and CenterTrack are trained for 30 and 70 epochs respectively. For the positive samples of unsupervised Re-ID learning from video-based data, the adjacent frames are randomly sampled from consecutive 20 frames.

4.2. Datasets

The proposed method is evaluated on the standard MOTChallenge datasets, including MOT16, MOT17 [31] and MOT20 [11]. There are 7 training and other 7 testing videos in MOT16. MOT17 contains the same videos as MOT16 but with different annotations. MOT20 contains 4 training videos and 4 testing videos. The videos in MOT20 are captured in crowd scenes, which are quite different from those in MOT16 and MOT17. External dataset CrowdHuman [36] is adopted for pre-training. Note that pre-training on external dataset is a common practice in previous works [34,53,45,43,54]. Besides the bounding box annotation for detection, identity information is also provided in MOT16, MOT17 and MOT20. However, the identity information is not used in our training process.

We adopt the standard metrics of MOTChallenge for evaluation, including: Multi-Object Tracking Accuracy (MOTA) [3], Multi-object Tracking Precision (MOTP) [3], ID F1 Score (IDF1), Mostly Tracked objects (MT), Mostly Lost objects (ML), Number of False Positives (FP), Number of False Negatives (FN), Number of Identity Switches (IDS) [24] and number of Fragments (Frag). Some other metrics, including R1 and mAP, are also introduced for the evaluation of different Re-ID methods.

4.3. Ablation studies

Without losing generality, we do ablation study on the MOT17 dataset for simplicity, following the work in FairMOT [53] and CenterTrack [56]. Since no validation data is provided in the MOTChallenge, the common practice is to split each video in the training set into two half videos, the first part is for training and the second part is for validation [53,56,35,43,48]. No external dataset is used if not specified.

4.3.1. Unsupervised Re-ID learning

In this sub-section, we conduct the ablation study for the unsupervised Re-ID learning module by integrating it into the MOT system FairMOT [53].

Comparison with other Re-ID learning methods: We first compare different learning methods for the Re-ID feature module, including the proposed method (UTrack), the unsupervised method CycAs [44] and the supervised method in FairMOT [53]. CycAs utilizes the cycle assignment consistence to learn the Re-ID module, which is the latest identity free method for Re-ID learning. FairMOT treats each identity as a class and the Re-ID module is trained in a supervised classification manner. The detailed comparison results are shown in Table 1. In order to demonstrate the effectiveness of the Re-ID module, the tracking results without Re-ID are also presented in the first row and denoted as FairMOT$_{w/o}$. Note that, except the training method of the Re-ID module, all the other parts remain the same.

Comparing the first row with the remaining three rows, we can find that MOTA, IDF1 and IDS are all greatly improved with the Re-ID module. For example, our UTrack improves MOTA and IDF1 by 1.8% and 10.8% respectively when the tracker is equipped with the Re-ID module trained by the proposed unsupervised method. Compared to the supervised Re-ID used in FairMOT[4], our UTrack can achieve almost the same MOTA even without using any Re-ID supervision. Though FairMOT possesses a slightly lower IDS, our method UTrack performs better in IDF1 by 1.6%, demonstrating the effectiveness of the proposed unsupervised Re-ID learning method. More importantly, our method does not suffer from the dimension explosion issue and is more friendly to the real large-scale MOT systems. Comparing our method UTrack with CysAs [44], both of whom are unsupervised methods, we find that both trackers achieve the same IDS, but our method performs better in terms of MOTA and IDF1. We attribute this to the introduction of the placeholder in the similarity matrix and the strong supervision signal L_{id}^{intra} within the same frame (Eq. (3)).

In Fig. 6, we visualize the learned Re-ID features for different methods by t-SNE [30]. As we can see, compared with the supervised method originally used in FairMOT [53] and the unsupervised method CycAs [44], the features for the same identities are better grouped by the proposed unsupervised method. We further present the assignment between two frames in Fig. 7. As we can see, the object pair with the same identity achieves a much higher similarity score than the counterpart with different identities.

Analysis of Re-ID loss: We then do some ablation analysis on the Re-ID loss L_{id}^{pos} in Eq. (6). It consists of three components. 1) L_{id}^{intra}: the loss used to avoid assigning an object to another one that locates in the same frame. 2) L_{id}^{inter}: the loss that makes sure an

[4] This tracker is trained by ourselves using its official code since the model is not available, which achieves the same MOTA, higher IDF1 and lower IDS.

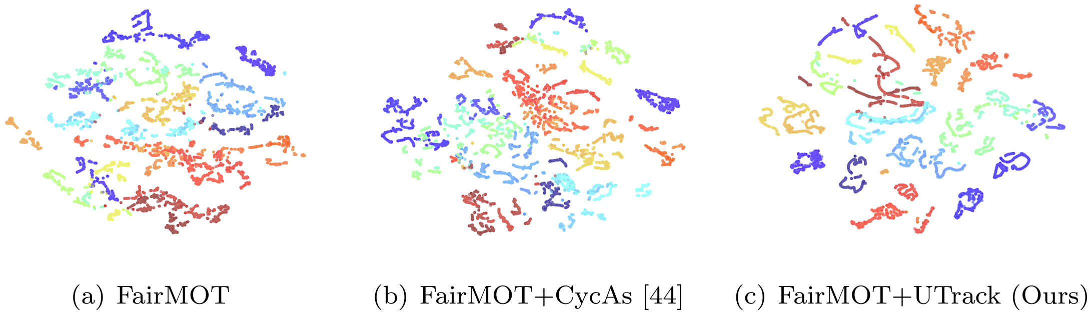

(a) FairMOT (b) FairMOT+CycAs [44] (c) FairMOT+UTrack (Ours)

Fig. 6. Visualized Re-ID features for identities in MOT17 validation set using t-SNE [30]. From left to right are the features learnt by (a): the supervised method in FairMOT [53]. (b): the unsupervised method CycAs [44]. (c): the proposed unsupervised method. Note that only the first 30 identities are presented here. Different colors indicate diffe. rent identities.

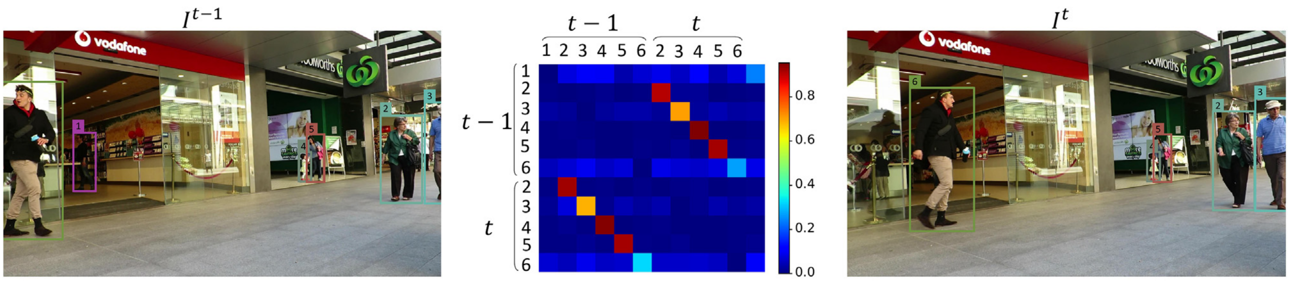


Fig. 7. Illustration of assignment between two frames. Left and right are the detection results, in which the color of boxes and the numbers attached to the boxes indicate the identities. Middle is the assignment matrix M'.

object can be successfully matched with another object that locates in different frames or the placeholder. 3) L_{id}^{cycle}: the loss that constraints that the forward and backward assignment should be consistent with each other. Results are shown in Table 2. It can be seen that, the performance gain from different Re-ID losses in terms of MOTA is not that significant since MOTA is highly affected by the detection performance, i.e., FN and FP. But both IDS and IDF1 are improved by introducing more constraints on the Re-ID module, which demonstrates the effectiveness of each loss in Eq. (6).

Discussion on the temperature T**:** In our default settings, T is adaptive to the number of objects. To show its superiority, we train several models with different fixed temperatures, the tracking results are shown in Table 3. As we can see, provided with larger temperatures ($T \geqslant 4$), the trackers achieve better IDF1 score but degraded MOTA score compared with those trackers equipped with smaller temperatures ($T \leqslant 3$). With the help of adaptive temperature, the tracker obtains a better balance between IDF1 and MOTA scores.

Discussion on the placeholder: Finally, we have a discussion on the value of placeholder p. As mentioned before, the placeholder p is introduced to handle the birth and death of objects, i.e., the newly appeared objects in I^t and the objects appearing in I^{t-1} but disappearing in I^t should be assigned to the placeholder. Let $S_{i,j}$ be the cosine similarity between the Re-ID feature of objects i and j. For sensible matching, $S_{i,j}$ should be greater than p if object i and j have the same identity, otherwise $S_{i,j} < p$.

Taking into intuitive consideration that the cosine similarity between the Re-ID feature of two objects should be positive if they

Table 2
Analysis of Re-ID loss on validation set.

L_{id}^{inter}	L_{id}^{intra}	L_{id}^{cycle}	MOTA↑	IDF1 ↑	MT↑	ML↓	FP↓	FN ↓	IDS↓
✓			67.3%	69.9%	134	**58**	**2422**	14669	592
✓	✓		67.5%	70.6%	133	63	2492	14566	531
✓	✓	✓	**67.6%**	**71.8%**	**137**	63	2621	**14388**	**503**

Table 3
The impact of temperature T on tracking performance. Evaluated on MOT17 validation set. C is the number of columns in similarity matrix.

Tracker	T	MOTA↑	IDF1 ↑	MT↑	ML↓	FP↓	FN ↓	IDS↓
FairMOT + UTrack	1.0	67.5%	63.2%	145	**57**	2778	13978	788
	2.0	**67.9%**	63.5%	**147**	58	2717	13840	789
	3.0	67.6%	65.0%	147	**57**	2894	**13791**	838
	4.0	65.5%	69.1%	128	62	3186	14874	565
	5.0	65.4%	70.1%	129	59	3130	15059	**502**
	$2\log(C+1)$	67.6%	**71.8%**	137	63	**2621**	14388	503

Table 4
Tracking results on the MOT17 validation set with respect to different settings for placeholder.

placeholder	MOTA↑	IDF1 ↑	MT↑	ML↓	FP↓	FN ↓	IDS↓
w/o	67.4%	71.1%	**142**	62	**2457**	14588	557
zero	66.9%	70.7%	138	**60**	2685	14735	**494**
mean	**67.6%**	**71.8%**	137	63	2621	**14388**	503

Table 5
Tracking results on the MOT17 validation set with the occlusion estimation module (OccE) integrated in CenterTrack [56] and FairMOT [53] to show its effectiveness respectively.

trackers	MOTA↑	IDF1 ↑	MT↑	ML↓	FP↓	FN ↓	IDS↓
FairMOT + UTrack	67.6%	71.8%	137	63	**2621**	14388	503
FairMOT + UTrack + GSM [27]	68.1%	71.8%	**164**	**49**	5095	**11763**	**366**
FairMOT + UTrack + OccE	**68.5%**	**72.0%**	142	57	2840	13797	396
CenterTrack	60.7%	62.7%	112	76	**2179**	18447	564
CenterTrack + GSM [27]	61.5%	63.9%	**131**	**63**	4508	**15943**	**254**
CenterTrack + OccE	**62.1%**	**64.6%**	127	68	3372	16583	440

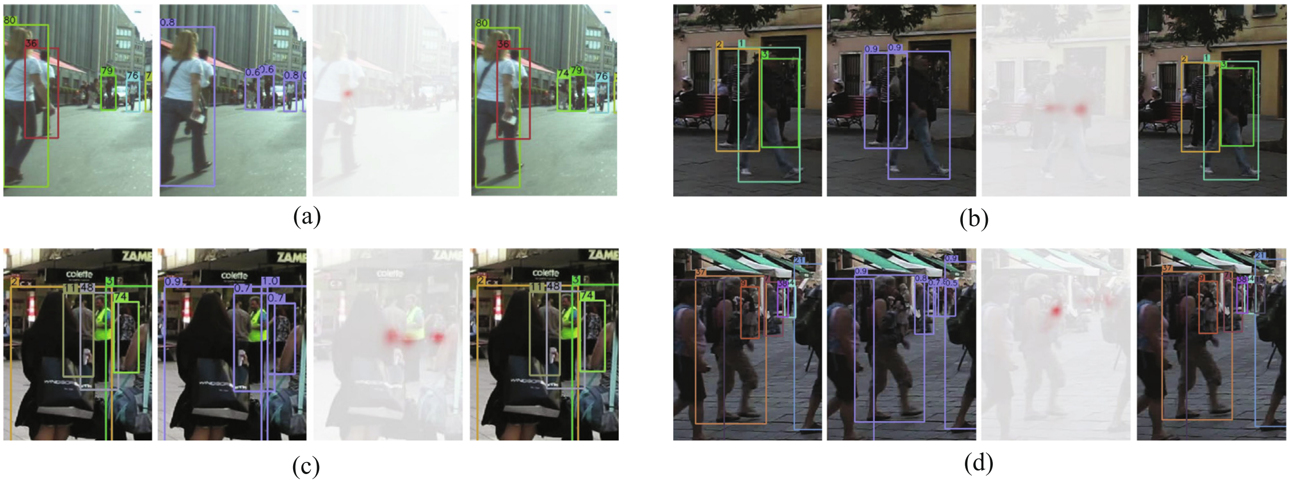


Fig. 8. Some cases where lost objects are re-found by the proposed occlusion estimation module. For each case, from left to right are tracking results in the previous frame, detection results, predicted occlusions and tracking results in the current frame respectively. Specifically, objects 36, 3, 11 and 9 are found back for cases (a), (b), (c) and (d), respectively. Note that the images here are cropped from original images for better viewing.

have the same identity, otherwise negative, it is straightforward to set $p = 0$. However, at the early training stage, we observe that the variance of the values in S is small (about 0.015) and the cosine similarity between any pair of objects is around 0.75. So it is hard for the model to handle the birth and death of objects well at the beginning if $p = 0$. Therefore, we set p as the dynamic mean of the values in S except the diagonal values by default. Interestingly, we observe that the mean of the values in S is about 0 after convergence when trained with this strategy.

The results of three different placeholder settings are shown in Table 4: without placeholder $p, p = 0$, and p as the dynamic mean. As we can see that the tracker achieves the best results in terms of MOTA, IDF1 and FN when the placeholder is set to the dynamic mean of similarity values. Compared to the setting without the placeholder, using placeholder can achieve much lower IDS, which demonstrates the effectiveness of the placeholder.

4.3.2. Occlusion estimation module

To demonstrate the effectiveness of the proposed occlusion estimation module, we apply it to both FairMOT [53] and CenterTrack [56]. Another work of lost object refinding, GSM [27], is also evaluated. As shown in Table 5, with the help of the occlusion estimation module (OccE), many lost objects can be found back, leading to lower FN. Though the FP is slightly increased, the main tracking metric MOTA is still improved. In addition, more objects can be mostly tracked (MT), and fewer objects are mostly lost (ML). Besides, IDS are greatly reduced. Compared with OccE, GSM introduces more FP, resulting a slightly lower MOTA. Though lower IDS is achieved by GSM, the construction and matching of graphs are time consuming.

In Fig. 8, some typical cases where occlusion happens are presented. For each case, the left to the right columns are the tracking results in the previous frame, the detection results and the predicted occlusions, as well as the tracking results in the current frame respectively. As we can see, some occluded objects are missed by the detector and are challenging for existing MOT systems to track successfully. Without such refinding mechanism, they can only handle the detected objects and keep undetected boxes untracked. By integrating the proposed occlusion estimation module and the accompanying object refinding algorithm, we can refind the missed objects back and link them with existing tracklets.

Table 6
The impact of τ on tracking performance. Evaluated on MOT17 validation set.

trackers	τ	MOTA↑	IDF1 ↑	MT↑	ML↓	FP↓	FN ↓	IDS↓
FairMOT + UTrack + OccE	0.3	68.3%	70.4%	**145**	57	3228	**13501**	412
	0.5	68.3%	**72.0%**	143	**55**	3114	13609	426
	0.7	**68.5%**	**72.0%**	142	57	2840	13797	**396**
	0.9	68.4%	71.8%	141	58	**2758**	13928	397

Discussion on the threshold τ: The hyper-parameter τ in Eq. (9) controls the valid number of occlusions in a frame while training. We evaluate the impact of τ on tracking performance by applying it to FairMOT [53]. Results are shown in Table 6. Specifically, the tracker achieves more FN but less FP with a larger τ. This is because fewer occlusions could be detected if the model is trained with a larger τ, thus fewer lost objects could be found back. However, the overall tracking performance (MOTA) is not sensitive to the value of τ and we set it to 0.7 by default.

4.3.3. Pre-training on image-based data

The proposed method can benefit from pre-training on image-based data. To demonstrate it, we use the CrowdHuman dataset [36] for pre-training. Need to note that, the original FairMOT [53] also pre-trains their model on CrowdHuman and its Re-ID module is trained with pseudo identity labels, i.e., a unique identity is assigned to each annotated box, in a classification manner. There are about 339 K boxes in CrowdHuman, so the pseudo identity number is massive, causing the number of parameters in the classifier to be even larger than the total number of parameters in the other modules (54.0 M vs. 19.4 M). By contrast, there are no extra parameters introduced in the proposed unsupervised Re-ID learning method. While training, two augmentations of an image are treated as a positive sample pair, and two different static images are sampled as a negative sample pair.

Table 7
Comparison of re-identification capability of different methods on the MOT17 train split by directly applying the pre-trained models on CrowdHuman [36] without fine-tuning.

Trackers	R1↑	mAP ↑
FairMOT [53]	42.9%	25.4%
FairMOT + CysAs [44]	54.8%	32.9%
FairMOT + UTrack	**56.4%**	**34.1%**

Table 8
The impact of the ratio between the number of negative and positive samples on re-identification capability. These models are trained on CrowdHuman [36] and evaluated on MOT17 train split without fine-tuning.

Tracker	N^{neg}/N^{pos}	R1↑	mAP ↑
FairMOT + UTrack	1/9	54.6%	32.8%
	2/8	56.4%	**34.1%**
	3/7	**58.1%**	31.8%
	4/6	57.9%	31.3%
	5/5	not converged	
	6/4	not converged	

We first compare the re-identification capability of the proposed unsupervised Re-ID learning method UTrack, pseudo identity based method FairMOT [53], and the latest identity free method CysAs [44] in Table 7. While evaluation, each tracklet in MOT17 train split is divided into two half parts. The first part is used as query and the rest part is used as gallery. As we can see, both CysAs and the proposed UTrack achieve much better results than FairMOT. Compared with CysAs, UTrack obtains 1.6% higher R1 and 1.2% higher mAP. We argue this to the introduction of placeholder and the strong supervision signal.

We then evaluate the impact of $\frac{N^{neg}}{N^{pos}}$ on re-identification capability while training on image-based data in Table 8. The model achieves the best R1 score when $\frac{N^{neg}}{N^{pos}} = \frac{3}{7}$, but achieves the best mAP score when $\frac{N^{neg}}{N^{pos}} = \frac{2}{8}$. Interestingly, we find that the Re-ID module cannot converge if $\frac{N^{neg}}{N^{pos}} \geqslant 1$. We set $\frac{N^{neg}}{N^{pos}}$ to $\frac{2}{8}$ for the balance between R1 and mAP scores.

In Table 9, we show the tracking results of trackers on MOT17 dataset by directly applying the CrowdHuman pre-trained model without fine-tuning. Compared with the supervised Re-ID learning in FairMOT and unsupervised Re-ID learning CysAs [44], our UTrack performs much better in terms of MOTA, IDF1, MT, FN and IDS, demonstrating the superiority of the proposed unsupervised Re-ID learning method.

We further show the results in Table 10 by fine-tuning the CrowdHuman pre-trained models (marked by ☆) on the MOT17 dataset. For reference, we also provide the results without pre-training. From the results, we can observe that pre-training on the image-based data can generally boost the overall tracking performance. Compared to the supervised Re-ID learning method used in FairMOT, our unsupervised tracker UTrack can achieve very comparable tracking performance without using any ID supervision. By contrast, pre-training on image-based data using CysAs cannot improve the IDF1 performance.

4.4. Results on MOTChallenge

Though the main focus of this paper is not to achieve the state-of-the-art performance, we still show the tracking results on the standard MOTchallenge benchmark by integrating the proposed modules into FairMOT [53] and CenterTrack [56] respectively. For notation simplicity, we denote the variant of FairMOT that integrates our unsupervised Re-ID learning and occlusion estimation module as ”OUTrack$_{\mathrm{fm}}$”, and the variant of CenterTrack that integrates our occlusion estimation module as ”OTrack$_{\mathrm{ct}}$”. We conduct

Table 9
Tracking results on the MOT17 validation set by directly applying the pre-trained trackers on CrowdHuman [36] without fine-tuning.

Trackers	MOTA↑	IDF1 ↑	MT↑	ML↓	FP↓	FN ↓	IDS↓
FairMOT [53]	64.0%	64.6%	138	63	**2130**	16806	501
FairMOT + CysAs [44]	63.9%	64.9%	137	**62**	2781	16105	594
FairMOT + UTrack	**64.8%**	**69.2%**	**143**	64	2390	**16203**	**418**

Table 10
Tracking results on the MOT17 validation set by fine-tuning the CrowdHuman pre-trained trackers on the MOT17 dataset. Trackers marked with/without * correspond pre-training on CrowdHuman or not.

trackers	MOTA↑	IDF1 ↑	MT↑	ML↓	FP↓	FN ↓	IDS↓
			impact on Re-ID module				
FairMOT [53]	67.5%	70.2%	134	55	2814	14263	492
FairMOT* [53]	70.7%	**74.7%**	**172**	48	3255	12171	**431**
FairMOT + CysAs [44]	67.0%	70.8%	134	60	2631	14681	503
FairMOT + CysAs [44]*	69.4%	70.8%	158	46	3412	12515	592
FairMOT + UTrack	67.6%	71.8%	137	63	**2621**	14388	503
FairMOT + UTrack*	**70.8%**	73.8%	165	**45**	3222	**12052**	524
			impact on occlusion estimation module				
FairMOT + UTrack + OccE	68.5%	72.0%	142	57	2840	13797	**396**
FairMOT + UTrack + OccE*	**72.0%**	**73.1%**	**168**	**46**	3565	**11119**	417
CenterTrack[56]	60.7%	62.7%	112	76	**2179**	18447	564
CenterTrack*[56]	66.1%	64.2%	140	72	2442	15286	588
CenterTrack + OccE	62.1%	64.6%	127	68	3372	16583	440
CenterTrack + OccE*	67.4%	67.6%	158	63	4086	13107	414

Table 11
Benchmark results on MOTChallenge. Trackers marked with † track objects in an offline manner. FairMOT§ is pre-trained on CrowdHuman without five extra datasets[4]. Best results are shown in **bold** and highlighted with underline.

benchmark		methods	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	Recall↑	IDS↓	Frag↓	Hz↑
MOT16	Public	Tracktor++ [2]	56.2%	54.9%	20.7%	35.8%	**2394**	76844	57.9%	617	1068	1.6
		GSM$_{Tracktor}$ [27]	57.0%	58.2%	22.0%	34.5%	4332	73573	59.6%	475	859	7.6
		MPNTrack† [5]	58.6%	61.7%	27.3%	34.0%	4949	70252	61.5%	**354**	**684**	6.5
		Lif_T† [19]	61.3%	64.7%	27.0%	34.0%	4844	65401	64.1%	389	1034	0.5
		TMOH [38]	63.2%	63.5%	27.0%	31.0%	3122	63376	65.2%	635	1486	0.7
		OTrack$_{ct}$ (ours)	65.3%	62.7%	26.1%	34.9%	5179	57484	68.5%	628	1616	17.2
		OUTrack$_{fm}$ (ours)	**69.3%**	**67.5%**	**37.3%**	**19.1%**	10657	**44059**	**75.8%**	1284	2677	**24.5**
	Private	JDE[45]	64.4%	55.8%	35.4%	20.0%	–	–	–	1544	–	22.0
		LM_CNN† [1]	67.4%	61.2%	38.2%	19.2%	10109	48435	73.4%	931	1034	1.7
		LMP† [40]	71.0%	**80.2%**	**46.9%**	21.9%	**7880**	44564	75.6%	**434**	**587**	0.5
		SOTMOT [54]	72.1%	72.3%	44.0%	13.2%	14344	34784	–	1681	–	16
		FairMOT[53]	**74.9%**	72.8%	44.7%	15.9%	10163	34484	81.1%	1074	2567	25.4
		FairMOT§[53]	72.7%	74.0%	42.0%	17.8%	12930	35804	80.4%	1121	2732	**25.4**
		OTrack$_{ct}$ (ours)	73.3%	70.3%	41.3%	15.9%	30057	115944	79.5%	4440	8742	17.0
		OUTrack$_{fm}$ (ours)	74.2%	71.1%	44.8%	**14.0%**	13214	**32581**	**82.1%**	1324	2413	24.8
MOT17	Public	Tracktor++ [2]	56.3%	55.1%	21.1%	35.3%	**8866**	235449	58.3%	1987	3763	1.5
		GSM$_{Tracktor}$ [27]	56.4%	57.8%	22.2%	34.5%	14379	230174	59.2%	1485	2763	8.7
		MPNTrack† [5]	58.8%	61.7%	28.8%	33.5%	17413	213594	62.1%	**1185**	**2265**	6.5
		Lif_T† [19]	60.5%	65.6%	27.0%	33.6%	14966	206619	63.4%	1189	3476	0.5
		CenterTrack [56]	61.5%	59.6%	26.4%	31.9%	14076	200672	64.4%	2583	4965	17.5
		TMOH [38]	62.1%	62.8%	26.9%	31.4%	10951	201195	64.3%	1897	4622	0.7
		SiamMOT [37]	65.9%	63.3%	34.6%	23.9%	18098	170955	–	–	–	17
		OTrack$_{ct}$ (ours)	63.9%	62.3%	25.7%	35.5%	14903	186878	66.9%	1949	4952	17.2
		OUTrack$_{fm}$ (ours)	**69.0%**	**66.8%**	**37.6%**	**19.7%**	28855	**141587**	**74.9%**	4449	8733	**24.8**
	Private	CenterTrack [56]	67.8%	64.7%	34.6%	24.6%	**18498**	160332	71.6%	3039	6102	17.5
		SOTMOT [54]	71.0%	71.9%	42.7%	15.3%	39537	118983	–	5184	–	16
		FairMOT [53]	**73.7%**	72.3%	43.2%	17.3%	27507	117477	79.2%	3303	8073	**25.9**
		FairMOT§ [53]	71.8%	**73.1%**	40.9%	19.0%	34764	120909	78.6%	3534	8724	**25.9**
		OTrack$_{ct}$ (ours)	69.0%	67.8 %	35.4%	21.1%	39159	133143	76.4%	**2643**	6261	16.9
		OUTrack$_{fm}$ (ours)	73.5%	70.2%	**43.3%**	**15.2%**	34764	**110577**	**80.4%**	4110	7506	25.4
MOT20	public	SORT [4]	42.7%	45.1%	16.7%	26.2%	27521	264694	48.8%	4470	17798	57.3
		MLT[52]	48.9%	54.6%	30.9%	22.1%	45660	216803	58.1%	2187	3067	3.7
		Tracktor++[2]	52.6%	52.7%	30.3%	25.0%	**6439**	36680	55.4%	**1648**	4374	1.2
		TMOH [38]	60.1%	61.2%	46.7%	17.8%	38043	165899	67.9%	2342	4320	0.6
		OTrack$_{ct}$ (ours)	60.8%	60.0%	**56.4%**	14.3%	70156	129703	**74.9%**	2783	**2984**	9.4
		OUTrack$_{fm}$ (ours)	**65.3%**	**65.0%**	49.4%	**13.3%**	38709	**13799**	73.4%	2832	7212	**10.7**
	private	FairMOT [53]	61.8%	67.3%	**68.8%**	**7.6%**	103440	**88901**	**80.6%**	5243	7874	**13.2**
		FairMOT§ [53]	68.1%	71.1%	53.3%	12.9%	**30503**	131380	74.6%	3019	10509	**13.2**
		SOTMOT [54]	**68.6%**	**71.4%**	64.9%	9.7%	57064	101154	–	4209	–	8.5
		OTrack$_{ct}$ (ours)	65.8%	61.8%	58.9%	13.1%	48947	125152	73.8%	2897	**3064**	10.3
		OUTrack$_{fm}$ (ours)	68.5%	69.4%	57.9 %	12.2%	37431	123197	76.2%	**2147**	5683	12.4

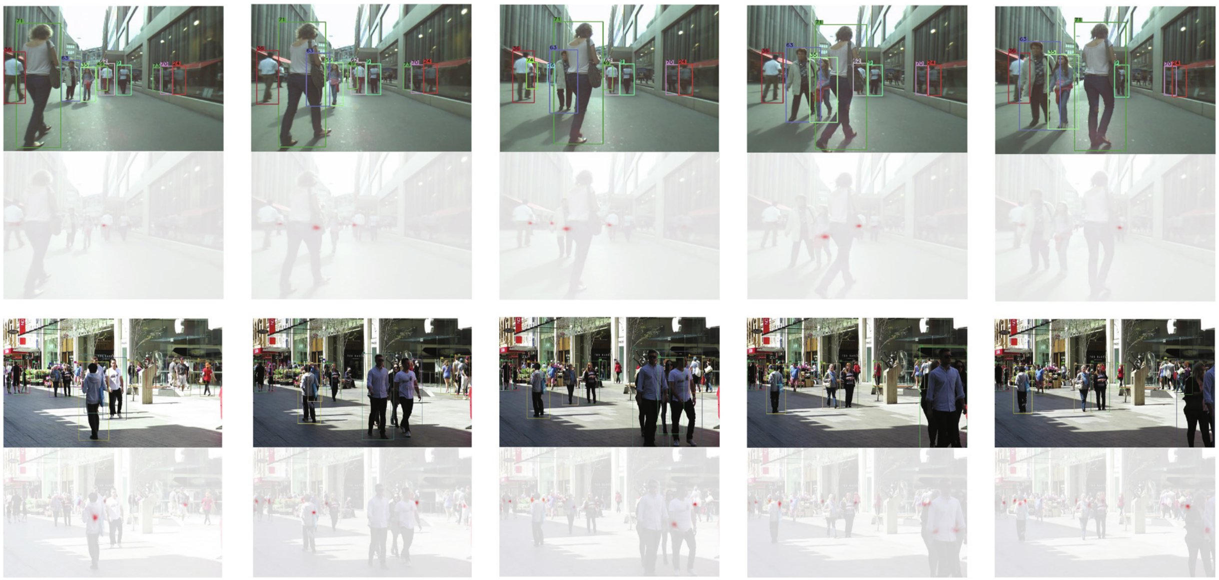


Fig. 9. The tracking results and predicted occlusion heatmaps of OUTrack$_{fm}$.

the evaluation on both public and private detection. For the public detection evaluation, we follow the works in [2,27,56,17,21,35,43,48] to refine the public detections and keep the bounding boxes that are close to the tracked objects. Note that only the provided training sequences are used to train the model for public detection evaluation. For private detection, we follow CenterTrack and FairMOT to pre-train our tracker with the CrowdHuman dataset. However, the original FairMOT also involves a mixture dataset that consists of five extra datasets[5]. For fair comparison, we train FairMOT on CrowdHuman and MOTChallenge without these five extra datasets, which is denoted as FairMOT§.

Results are shown in Table 11. Overall, our FairMOT based tracker OUTrack$_{fm}$ achieves state-of-the-art performance on all datasets and our CenterTrack based tracker OTrack$_{ct}$ outperfoms CenterTrack by a large margin. Compared to offline methods, such as Lif_T [19] and MPNTrack [5], both OUTrack$_{fm}$ and OTrack$_{ct}$ have a higher Frag, the reason is that the short broken tracklets can be linked to a long trajectory by a post process, which is not allowed in online trackers.

Compared with FairMOT, OUTrack$_{fm}$ achieves very comparable performance on MOT17 with much less pretraining data and better performance with the same pretraining data in terms of MOTA. As for IDF1, FairMOT performs better than OUTrack$_{fm}$ by 2.1%. The main reason is that FairMOT supervises Re-ID module with identity information, while the Re-ID module in OUTrack$_{fm}$ is trained in a totally unsupervised manner. It is interesting that OUTrack$_{fm}$ performs much better than FairMOT on MOT20 with private detection. The main reasons may be in two folds: 1) FairMOT fine-tunes the models on MOT20 after pre-training on the mixture dataset, while the mixed datasets used in FairMOT are different from MOT20 that is captured in crowd scenes. However, we fine-tune the model on MOT20 after pre-training on CrowdHuman and the images in CrowdHuman are all collected from crowd scenes. 2) A higher detection confidence (0.5) is used in OUTrack$_{fm}$, since FairMOT (0.3) has a much higher FP. When training FairMOT on the same dataset as OUTrack$_{fm}$ and increasing the detection confidence to 0.5, the main metric MOTA is greatly improved. But OUTrack$_{fm}$ still performs better.

Compared to CenterTrack, OTrack$_{ct}$ performs much better on MOT17 for both the private and public detection settings. For example, our OTrack$_{ct}$ surpasses CenterTrack on MOT17 in terms of MOTA by 2.4% and 1.2% with public and private detection, respectively. Through finding the lost objects based on the predicted occlusions, a higher Recall is also achieved. Though more FP are involved when finding the missed objects, the FN is greatly reduced.

In Fig. 9, some tracking results and the predicted occlusion heatmaps of OUTrack$_{fm}$ are presented. For the first case, we can see that some objects (63 and 35) can be re-identified when they reappeared after a short-term disappearance, demonstrating the effectiveness of our unsupervised Re-ID learning method. For both cases, occlusions between different objects can be effectively detected by occlusion estimation module and those highly occluded objects still can be tracked, indicating the effectiveness of our occlusion estimation module.

5. Conclusion

In this paper, we present a new occlusion-aware multi-object tracking framework. It involves two key modules: unsupervised Re-ID learning and occlusion estimation module. The unsupervised Re-ID learning adopts an unsupervised matching based loss between adjacent frames, whose motivation is that objects with the same identity in adjacent frames share similar appearance and objects in two images that from different scenes (or within the same image) have different identities and appearances. Compared to the supervised classification based Re-ID learning, it does not suffer from the dimension explosion issue for a large identity number and is more friendly to real large-scale applications. The occlusion estimation module can alleviate the tracking lost issue caused by missing detection. It can find the occluded objects back by estimating the occlusion map that shows all possible occlusion locations. The two proposed modules can be applied to existing MOT systems in a natural way and demonstrate their effectiveness.

CRediT authorship contribution statement

Qiankun Liu: Conceptualization, Methodology, Software, Writing – original draft, Visualization. **Dongdong Chen:** Conceptualization, Validation, Writing – original draft, Visualization. **Qi Chu:**

[5] These datasets are ETH [14], CityPerson [51], CalTech [12], CUHK-SYSU [49] and PRW [55]. Besides the box annotations, the identity information is also provided in the last three datasets.

Visualization, Supervision, Writing – review & editing. **Lu Yuan:** Writing – review & editing. **Bin Liu:** Writing – review & editing. **Lei Zhang:** Writing – review & editing. **Nenghai Yu:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. U20B2047, No. 62002336).

References

[1] M. Babaee, Z. Li, G. Rigoll, A dual cnn–rnn for multiple people tracking, Neurocomputing 368 (2019) 69–83.

[2] P. Bergmann, T. Meinhardt, L. Leal-Taixe, Tracking without bells and whistles, in: IEEE International Conference on Computer Vision, 2019, pp. 941–951.

[3] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the clear mot metrics, J. Image Video Process. (2008) 1.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: International Conference on Image Processing, IEEE, 2016, pp. 3464–3468.

[5] G. Brasó, L. Leal-Taixé, Learning a neural solver for multiple object tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 6247–6257.

[6] C. Chi, S. Zhang, J. Xing, Z. Lei, S.Z. Li, X. Zou, et al., Pedhunter: Occlusion robust pedestrian detector in crowded scenes, in: AAAI Conference on Artificial Intelligence, 2020, pp. 10639–10646.

[7] P. Chu, H. Fan, C.C. Tan, H. Ling, Online multi-object tracking with instance-aware tracker and dynamic model refreshment, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 161–170.

[8] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, N. Yu, Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism, in: IEEE International Conference on Computer Vision, 2017, pp. 4836–4845.

[9] Q. Chu, W. Ouyang, B. Liu, F. Zhu, N. Yu, Dasot: A unified framework integrating data association and single object tracking for online multi-object tracking, in: AAAI Conference on Artificial Intelligence, 2020, pp. 10672–10679.

[10] X. Chu, A. Zheng, X. Zhang, J. Sun, Detection in crowded scenes: One proposal, multiple predictions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 12214–12223.

[11] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, L. Leal-Taixé, Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003, 2020..

[12] P. Dollár, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: A benchmark, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 304–311.

[13] X. Dong, J. Shen, L. Shao, F. Porikli, Clnet: A compact latent network for fast adjusting siamese trackers, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16. Springer, 2020, pp. 378–395..

[14] A. Ess, B. Leibe, K. Schindler, L. Van Gool, A mobile vision system for robust multi-person tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[15] H. Fan, L. Zheng, C. Yan, Y. Yang, Unsupervised person re-identification: Clustering and fine-tuning, ACM Trans. Multimedia Comput. Commun. Appl. 14 (4) (2018) 1–18.

[16] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2009) 1627–1645.

[17] J. He, Z. Huang, N. Wang, Z. Zhang, Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5299–5309.

[18] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[19] A. Hornakova, R. Henschel, B. Rosenhahn, P. Swoboda, Lifted disjoint paths with application in multiple object tracking, 2020..

[20] S. Karthik, A. Prabhu, V. Gandhi, Simple unsupervised multi-object tracking. arXiv preprint arXiv:2006.02609, 2020..

[21] C. Kim, L. Fuxin, M. Alotaibi, J.M. Rehg, Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9553–9562.

[22] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization. International Conference for Learning Representations, 2015..

[23] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Europeon Conference on Computer Vision, 2018, pp. 734–750.

[24] Y. Li, C. Huang, R. Nevatia, Learning to associate: Hybridboosted multi-target tracker for crowded scene, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 2953–2960.

[25] Y. Li, G. Yin, C. Liu, X. Yang, Z. Wang, Triplet online instance matching loss for person re-identification, Neurocomputing 433 (2021) 10–18.

[26] H. Liu, Z. Xiao, B. Fan, H. Zeng, Y. Zhang, G. Jiang, Prgcn: Probability prediction with graph convolutional network for person re-identification, Neurocomputing 423 (2021) 57–70.

[27] Q. Liu, Q. Chu, B. Liu, N. Yu, Gsm: Graph similarity model for multi-object tracking, International Joint Conferences on Artificial Intelligence Organization 7 (2020) 530–536.

[28] Q. Liu, B. Liu, Y. Wu, W. Li, N. Yu, Real-time online multi-object tracking in compressed domain, IEEE Access 7 (2019) 76489–76499.

[29] Y. Liu, X. Li, T. Bai, K. Wang, F.-Y. Wang, Multi-object tracking with hard-soft attention network and group-based cost minimization, Neurocomputing 447 (2021) 80–91.

[30] L.v.d. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learni. Res. 9 (2008) 2579–2605..

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

[32] L. Porzi, M. Hofinger, I. Ruiz, J. Serrat, S.R. Bulo, P. Kontschieder, Learning multi-object tracking and segmentation from automatic annotations, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 6846–6855.

[33] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1137–1149.

[34] A. Sadeghian, A. Alahi, S. Savarese, Tracking the untrackable: Learning to track multiple cues with long-term dependencies, in: IEEE International Conference on Computer Vision, 2017, pp. 300–311.

[35] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, S. Gould, Probabilistic tracklet scoring and inpainting for multiple object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14329–14339.

[36] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, J. Sun, Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123, 2018..

[37] B. Shuai, A. Berneshawi, X. Li, D. Modolo, J. Tighe, Siammot: Siamese multi-object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12372–12382.

[38] D. Stadler, J. Beyerer, Improving multiple pedestrian tracking by track management and occlusion handling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10958–10967.

[39] J. Sun, Y. Li, H. Chen, Y. Peng, X. Zhu, J. Zhu, Visible-infrared cross-modality person re-identification based on whole-individual training, Neurocomputing 440 (2021) 1–11.

[40] S. Tang, M. Andriluka, B. Andres, B. Schiele, Multiple people tracking by lifted multicut and person re-identification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3539–3548.

[41] Z. Tu, A. Zhou, C. Gan, B. Jiang, A. Hussain, B. Luo, A novel domain activation mapping-guided network (da-gnt) for visual tracking, Neurocomputing 449 (2021) 443–454.

[42] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B.B.G. Sekar, A. Geiger, B. Leibe, Mots: Multi-object tracking and segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7942–7951.

[43] Q. Wang, Y. Zheng, P. Pan, Y. Xu, Multiple object tracking with correlation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3876–3886.

[44] Z. Wang, J. Zhang, L. Zheng, Y. Liu, Y. Sun, Y. Li, S. Wang, Cycas: Self-supervised cycle association for learning re-identifiable descriptions, European Conference on Computer Vision (2020) 72–88.

[45] Z. Wang, L. Zheng, Y. Liu, S. Wang, Towards real-time multi-object tracking, in: Europeon Conference on Computer Vision, 2019.

[46] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: International Conference on Image Processing, IEEE, 2017, pp. 3645–3649.

[47] G. Wu, X. Zhu, S. Gong, Tracklet self-supervised learning for unsupervised person re-identification, Proceedings of the AAAI Conference on Artificial Intelligence. 34 (2020) 12362–12369.

[48] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, J. Yuan, Track to detect and segment: An online multi-object tracker, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12352–12361.

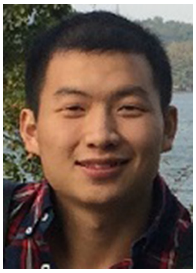
[49] T. Xiao, S. Li, B. Wang, L. Lin, X. Wang, Joint detection and identification feature learning for person search, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3415–3424.

[50] J. Xu, Y. Cao, Z. Zhang, H. Hu, Spatial-temporal relation networks for multi-object tracking, in: IEEE International Conference on Computer Vision, 2019, pp. 3988–3998.

[51] S. Zhang, R. Benenson, B. Schiele, Citypersons: A diverse dataset for pedestrian detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3213–3221.

[52] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke, Z. Xiong, Multiplex labeling graph for near-online tracking in crowded scenes, IEEE Internet Things J. 7 (9) (2020) 7892–7902.
[53] Y. Zhang, C. Wang, X. Wang, W. Zeng, W. Liu, Fairmot: On the fairness of detection and re-identification in multiple object tracking. arXiv: 2004.01888, 2020..
[54] L. Zheng, M. Tang, Y. Chen, G. Zhu, J. Wang, H. Lu, Improving multiple object tracking with single object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2453–2462.
[55] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, Q. Tian, Person re-identification in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1367–1376.
[56] X. Zhou, V. Koltun, P. Krähenbühl, Tracking objects as points, in: European Conference on Computer Vision, Springer, 2020, pp. 474–490.
[57] X. Zhou, D. Wang, P. Krähenbühl, Objects as points. arXiv preprint arXiv:1904.07850, 2019..
[58] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, M.-H. Yang, Online multi-object tracking with dual matching attention networks, in: Europeon Conference on Computer Vision, 2018, pp. 366–382.
[59] J. Zhu, Z. Yuan, C. Zhang, W. Chi, Y. Ling, et al., Crowded human detection via an anchor-pair network, in: The IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 1391–1399.

Qiankun Liu received his BS degree in 2017 from Xidian University. He is currently pursuing the PhD degree in Electrical Engineering in University of Science and Technology of China. His research interests detection, person re-identification and object tracking.

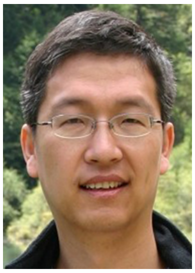


Dongdong Chen is a senior researcher from Microsoft Research. He received his PhD degree under the joint phd program between University of Science and Technology of China and MSRA. His research interests mainly include style transfer, image generation, image restoration, low-level image processing, and general representation learning.

Qi Chu received the B.S. degree in electronic engineering and the Ph.D. degree in information and communication engineering from University of Science and Technology of China in 2014 and 2019, respectively. Currently, he is an Associate Research Fellow at School of Cyber Science and Technology, University of Science and Technology of China. His research interests include deep learning, computer vision and artificial intelligence security.

Lu Yuan received his PhD degree from the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology in 2009. Before that, he received his MS degree at TsingHua University. Now he is a Senior Research Manager in Microsoft Redmond. His research interests include computer vision, applied machine learning and computational photography.

Bin Liu received the B.S. and M.S. degrees, both in electrical engineering, from the University of Science and Technology of China, Hefei, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical engineering from Syracuse University, Syracuse, NY, USA, in 2006. He is currently an Associate Professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include computer vision and internet of things.



Lei Zhang (Fellow, IEEE) received his Ph.D. degree in computer science from Tsinghua University in 2001. He is a principal researcher and research manager in Microsoft, working on computer vision and machine learning. Prior to his current post, he was a senior researcher at Microsoft Research Asia. He is interested in image understanding, visual pattern recognition, and machine learning, and holds 50 U.S. patents in these fields. He has also served as program area chairs or committee members for many related conferences.

Nenghai Yu is a full Professor at University of Science and Technology of China. He is also the director of Information Processing Center of USTC,deputy director of academic committee of School of Information Science and Technology. He received the Ph.D. degree from USTC in 2004. His research focuses on image processing and video analysis, multimedia communication, media content security, Internet information retrieval, data mining and content filtering, network communication and security.