One More Check: Making "Fake Background" Be Tracked Again

Chao Liang[1]*, Zhipeng Zhang[2]*, Xue Zhou[1, 3, 4]†, Bing Li[2], Weiming Hu[2]

[1]School of Automation Engineering, University of Electronic Science and Technology of China (UESTC)
[2] NLPR, Institute of Automation, Chinese Academy of Sciences (CASIA)
[3] Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China (UESTC)
[4] Intelligent Terminal Key Laboratory of SiChuan Province
chaoliang1996@gmail.com, zhangzhipeng2017@ia.ac.cn, zhouxue@uestc.edu.cn

Abstract

The one-shot multi-object tracking, which integrates object detection and ID embedding extraction into a unified network, has achieved groundbreaking results in recent years. However, current one-shot trackers solely rely on single-frame detections to predict candidate bounding boxes, which may be unreliable when facing disastrous visual degradation, e.g., motion blur, occlusions. Once a target bounding box is mistakenly classified as background by the detector, the temporal consistency of its corresponding tracklet will be no longer maintained. In this paper, we set out to restore the bounding boxes misclassified as "fake background" by proposing a re-check network. The re-check network innovatively expands the role of ID embedding from data association to motion forecasting by effectively propagating previous tracklets to the current frame with a small overhead. Note that the propagation results are yielded by an independent and efficient embedding search, preventing the model from over-relying on detection results. Eventually, it helps to reload the "fake background" and repair the broken tracklets. Building on a strong baseline CSTrack, we construct a new one-shot tracker and achieve favorable gains by 70.7 $\rightarrow$ 76.4, 70.6 $\rightarrow$ 76.3 MOTA on MOT16 and MOT17, respectively. It also reaches a new state-of-the-art MOTA and IDF1 performance. Code is released at https://github.com/JudasDie/SOTS.

Introduction

Multi-object tracking (MOT), aiming to estimate the trajectory of each target in a video sequence, is one of the most fundamental yet challenging tasks in computer vision (Luo et al. 2020). The related technique underpins significant applications from video surveillance to autonomous driving.

The current MOT methods are categorized into two-step and one-shot frameworks. The two-step framework (Bewley et al. 2016; Wojke, Bewley, and Paulus 2017; Yu et al. 2016; Tang et al. 2017; Xu et al. 2019), following the tracking-by-detection paradigm (or more precisely, tracking-after-detection), disentangles MOT into candidate boxes prediction and tracklet association. Though favored in astonishing performance, they suffer from massive computation cost brought by separately extracting ID (identity) embedding of each candidate box through an isolated ReID (Re-identification) network (Wojke, Bewley, and Paulus 2017; Zheng et al. 2017). Recently, the one-shot methods (Xiao et al. 2017; Wang et al. 2019; Zhang et al. 2020a; Liang et al. 2020), which integrate detection and ID embedding extraction into a unified network, have drawn great attention because of their balanced speed and accuracy. By sharing features and conducting multi-task learning, they are capable of running at quasi real-time speed. We observed that most existing one-shot trackers work under a strong complete detection assumption, in other words, all targets are presumed to be correctly localized by the detector. However, various real-world challenges may break such assumption and cause these approaches to fail. Fig. 1 shows typical failure cases of the one-shot trackers (*e.g.*, CSTrack (Liang et al. 2020)), where targets (red boxes) are considered as the background in some frames due to small foreground probabilities. The missed targets will break temporal consistency of a tracklet.

Figure 1: Illustration of tracklets generated from CSTrack on two clips. Wherein, "fake background" is represented by red box. The blue arrow indicates the motion direction of the target. Best viewed in color and zoom in.

Revisiting the failures of one-shot trackers, we find that the integrated detector solely considers single-frame visual cues. Nevertheless, the challenging scenes in practical tracking, *e.g.*, occlusion, motion blur, background clutter, will cause visual feature degradation, which may eventually mislead the detector to classify the targets as background. Hence, heavily relying on single frame detections is not reliable. In contrast, human vision has a dynamic view of tar-

*Equally Contribute. † Corresponding Author.

gets, not only taking current visual cues into consideration, but also being continuously aware of the temporal consistency of moving targets. This inspires us that exploring temporal cues might be a potential solution for reloading the misclassified targets of the detector.

In this work, motivated by inheriting the merits of one-shot models and mining temporal cues to make missed targets be tracked again, we propose a double-check mechanism to construct a new one-shot tracker. As an auxiliary to initial detections, a re-check network is delicately designed to learn the transduction of previous tracklets to the current frame. Given a target that appeared in previous frames, the propagation results re-check the entire scene in the current frame to make a candidate box prediction. If a groundtruth target box does not exist in the first-check predictions (*i.e.,* the results of object detector), as a potential misclassified target, it has a chance to be restored. Technically tracklet propagation is achieved by ID embedding search across frames, which is inspired by cross-correlation operation from Siamese trackers (Bertinetto et al. 2016; Zhang and Peng 2019; Zhang et al. 2020b). Some previous MOT methods (Chu and Ling 2019; Yin et al. 2020) attempt to introduce a separate Siamese network to learn additional clues of all targets for motion search, which are tedious and complex. Differently, we innovatively expand the role of ID embeddings from data association to motion forecasting. By reusing ID embeddings for propagation, the overhead of modeling temporal cues is minimized. Even with multiple tracklets, our re-check network can still propagate with one forward pass by a simple matrix multiplication.

Finally, we propose our new one-shot tracker, namely **OMC** (the initials of **O**ne **M**ore **C**heck), which is built on a baseline model CSTrack (Liang et al. 2020). It's worth noted that our proposed tracker efficiently integrates detection, embedding extraction and temporal cues mining into a unified framework. We evaluate the proposed OMC on three MOT Challenge [1] benchmarks: MOT16 (Milan et al. 2016), MOT17 (Milan et al. 2016) and MOT20 (Dendorfer et al. 2020). Our method achieves new state-of-the-art MOTA and IDF1 on all three benchmarks. Furthermore, compared with other trackers using temporal cues under the same public detection protocol (Milan et al. 2016), our method still achieves better tracking performance on MOTA and IDF1.

The main contributions of our work are as follows:

- We propose a simple yet effective double-check mechanism to restore the misclassified targets induced by the imperfect detection in MOT task. Our proposed re-check network flexibly expands ID embeddings from data association to motion forecasting, propagating previous tracklets to the current frame with a small overhead.
- Our proposed re-check network is a "plug-and-play" module that can work well with other one-shot trackers. We build it on a strong baseline CSTrack and construct a new one-shot tracker. The experimental results demonstrate that our tracker **OMC** not only outperforms CSTrack largely, but also achieves new state-of-the-art MOTA and IDF1 scores on all three benchmarks.

[1] https://motchallenge.net

Related Work

Detection-based Tracking

Recent MOT trackers can be summarized into two streams, *i.e.*, two-step and one-shot structures. The former one follows the tracking-by-detection paradigm, where object bounding boxes are first predicted by a detector and then linked into tracklets by an association network (Bewley et al. 2016; Wojke, Bewley, and Paulus 2017; Yu et al. 2016; Tang et al. 2017; Xu et al. 2019). These methods mainly focus on improving association accuracy. Though favored in good tracking performance, they suffer from computation cost brought by extracting ID embeddings for all bounding boxes with an additional ReID network (Wojke, Bewley, and Paulus 2017; Zheng et al. 2017). Alternatively, the one-shot paradigm which integrates detection and ID embedding extraction into a unified network, is a new trend in MOT (Xiao et al. 2017; Wang et al. 2019; Zhang et al. 2020a; Liang et al. 2020). Tong *et al*. (Xiao et al. 2017) first propose an end-to-end framework to jointly handle detection and ReID tasks. By adding extra fully connected layers to a two-stage detector (Faster RCNN (Ren et al. 2016)), the model can simultaneously generate detection boxes and the corresponding ID embeddings. Recent proposed JDE (Wang et al. 2019) converts the one-stage detector YOLOv3 (Redmon and Farhadi 2018) to a one-shot tracker by redesigning the prediction head. The follow-up CSTrack (Liang et al. 2020) further eases the competition between detection and ID embeddings learning by applying a cross-attention network to JDE (Wang et al. 2019). However, detection-based methods assume that all the targets can be precisely localized by the detector, which is not valid in practical tracking. When challenging scenes degrade the visual cues, the detector may miss some targets. In this work, we exploit cross-frame temporal cues to alleviate this issue. Below, we briefly review other methods that utilize temporal features to improve MOT trackers and discuss the differences between us.

Temporal Cues Mining

Some previous works attempt to utilize extra information, *e.g.*, motion (Chen et al. 2018; Hornakova et al. 2020), temporal visual features (Chu and Ling 2019), to improve detection performance in MOT task. Early works (Dehghan et al. 2015; Ristani and Tomasi 2018; Chen et al. 2018; Hornakova et al. 2020) consider MOT as a global optimization problem and obtains auxiliary candidates generated by Kalman filter, spatial interpolation, or visual extrapolation. Albeit efficient and straightforward, these methods leverage both past and future frames for batch processing that is not suitable for causal applications. For the aim of online tracking, recent works apply off-the-shelf SOT trackers, *e.g.*, SiamFC (Bertinetto et al. 2016), to estimate target motion. In the literature, there are two major branches of inserting SOT trackers into the MOT system. The first one (Chu and Ling 2019; Yin et al. 2020; Chu et al. 2020) aims to modify SOT networks and integrate them into an isolated association network for joint learning. In this regard, it's essential to equip an extra affinity learning model for handling drift. The other one (Chu et al. 2017; Sadeghian, Alahi, and Savarese 2017;

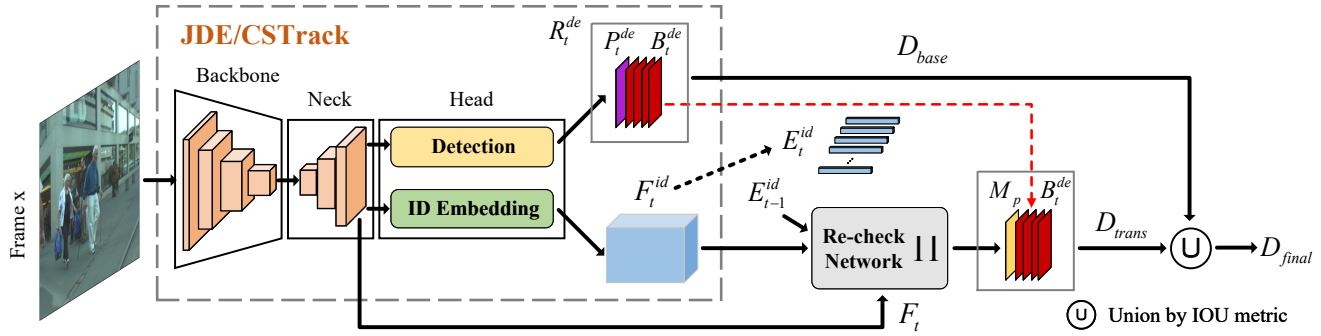

Figure 2: Overview of the proposed OMC. It consists of the baseline CStrack tracker and a re-check network. The CStrack tracker first generates detection result $\boldsymbol{R}_t^{de}$ and candidate embeddings $\boldsymbol{F}_t^{id}$. Then, re-check network improves temporal consistency by reloading the misclassified targets induced by the detector.

Zhu et al. 2018; Zhang et al. 2021) exploits separate SOT trackers to create auxiliary clues for handling complex MOT scenes. Despite the performance gains, they are not suitable for real-time applications because of the massive computation brought by applying a SOT network to learn auxiliary clues for all targets. In our work, instead of assigning an extra and complex SOT network, we expand the role of ID embeddings from data association to motion forecasting by similarity matching. It makes our tracker capable of tracking multiple targets with only a simple forward pass.

Methodology

In this section, we describe the proposed tracking framework, as illustrated in Fig. 2.

Overview

The proposed model is conducted on a recent MOT tracker, namely CSTrack (Liang et al. 2020), which is a variant of the recent JDE framework (Wang et al. 2019). In this section, we firstly describe the reasoning procedure of JDE and CSTrack. Then we elaborate on the details of integrating our proposed model into the baseline tracker.

Baseline Tracker. JDE (Wang et al. 2019) devotes effort to building a real-time one-shot MOT framework by allowing object detection and ID embedding extraction to be learned in a shared model, as shown in Fig. 2. Given a frame $\boldsymbol{x}$, it is firstly processed by a feature extractor Ψ (*e.g., Backbone* and *Neck*), which generates the feature $\boldsymbol{F}_t$,

$$\boldsymbol{F}_t = \Psi(\boldsymbol{x}). \tag{1}$$

Then $\boldsymbol{F}_t$ is fed into the *Head* network Φ to simultaneously predict detection results and ID embeddings,

$$[\boldsymbol{R}_t^{de}, \boldsymbol{F}_t^{id}] = \Phi(\boldsymbol{F}_t), \tag{2}$$

where $\boldsymbol{R}_t^{de}$ is the detection results (including one map $\boldsymbol{P}_t^{de} \in \mathbb{R}^{H\times W\times 1}$ for foreground probabilities and the others $\boldsymbol{B}_t^{de} \in \mathbb{R}^{H\times W\times 4}$ for raw boxes). $\boldsymbol{F}_t^{id} \in \mathbb{R}^{H\times W\times C}$ (C=512) denotes ID embeddings. The detection results $\boldsymbol{R}_t^{de}$ are processed by greedy-NMS (Ren et al. 2016) to generate the **basic detections** $\boldsymbol{D}_{base}$. Each box in $\boldsymbol{D}_{base}$ corresponds to a $1\times1\times C$ embedding in $\boldsymbol{F}_t^{id}$. We denote $\boldsymbol{E}_t^{id}$ as a set that contains embeddings of all boxes in $\boldsymbol{D}_{base}$. Finally, the boxes $\boldsymbol{D}_{base}$ and the ID embeddings $\boldsymbol{E}_t^{id}$ are utilized to associate with the prior tracklets by greedy bipartite matching. The recent CSTrack (Liang et al. 2020) introduces cross-attention to ease the competition between detection and ReID, which significantly improves the JDE with small overhead. Here, we use CSTrack as our baseline tracker.

OMC. In this work, we propose a re-check network to repair the "fake background" induced by the detector in JDE and CSTrack. As shown in Fig. 2, we reuse the ID embeddings from previous targets ($\boldsymbol{E}_{t-1}^{id}$) as temporal cues. The re-check network Π transfers the prior tracklets by measuring the similarity between $\boldsymbol{E}_{t-1}^{id}$ and $\boldsymbol{F}_t^{id}$. Specifically, we modify the cross-correlation layer, that is used by Siamese method in single object tracking (Bertinetto et al. 2016), to make it capable of tracking multiple targets in a single forward pass. We experimentally observed that if one target disappears in the current frame, it tends to introduce a false-positive response in the similarity map. To alleviate this issue, we fuse the visual feature $\boldsymbol{F}_t$ with the similarity map, and then refine them to a finer guidance map. For simplicity, we omit the operation of the re-check network as,

$$\boldsymbol{M}_p = \Pi\left(\boldsymbol{F}_t^{id}, \boldsymbol{E}_{t-1}^{id}, \boldsymbol{F}_t\right), \tag{3}$$

where the final prediction $\boldsymbol{M}_p$ represents the transduction of prior tracklets to the current frame. We consider $\boldsymbol{M}_p$ as the foreground probability, and send it to greedy-NMS with original bounding boxes $\boldsymbol{B}_t^{de}$ (red response maps). The outputs of NMS, namely **transductive detections** $\boldsymbol{D}_{trans}$, are combined with basic detections $\boldsymbol{D}_{base}$ through the proposed IOU vote mechanism to generate the final candidate bounding boxes $\boldsymbol{D}_{final}$. $\boldsymbol{D}_{final}$ and the corresponding ID embeddings extracted from $\boldsymbol{F}_t^{id}$ are used for latter association. When the basic detections mistakenly classify the targets as background, the transductive detections can recheck the "fake background" and restore the missed boxes.

Re-check Network

To improve the temporal consistency broken by "fake background", we propose a lightweight re-check network to re-

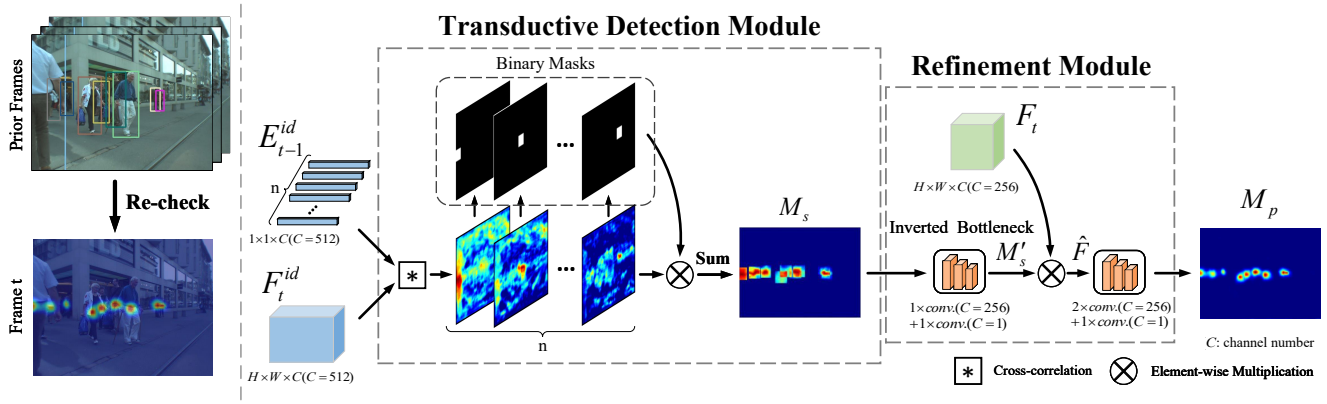

Figure 3: The architecture of the proposed re-check network. Re-check network consists of two major components: the transductive detection module and the refinement module. More details are described in section "Methodology".

Algorithm 1: Modified Cross-correlation, PyTorch-like

```
def D (E_{t-1}^{id}, F_t^{id}):
    E = torch.Tensor(E_{t-1}^{id}).view(n, c)  # convert list to matrix
    F = F_t^{id}.view(c, h * w)  # reshape tensor to matrix
    M = torch.matmul(E, F)  # matrix multiplication
    return M.view(h, w, n)  # reshape matrix to tensor
```

store the missed targets induced by the detector. Precisely, re-check network consists of two modules, *i.e.*, the transductive detection module for tracklets propagation and the refinement module for false positives filtering.

Transductive Detection Module The transductive detection module aims to propagate previous tracklets to current frame, in other words, predict locations of existing targets. Concretely, target locations are predicted by measuring the similarities between previous tracklet embeddings $\boldsymbol{E}_{t-1}^{id} = \{\boldsymbol{e}_{t-1}^{1}, \cdots, \boldsymbol{e}_{t-1}^{n}\}$ and current candidate embeddings $\boldsymbol{F}_{t}^{id}$, where n indicates the number of previous tracklets (include all active tracklets in the inference stage, not just the last frame). We get a location response map m_i for each target through a cross-correlation operator $*$,

$$\boldsymbol{m}_i = (\boldsymbol{e}_{t-1}^{i} * \boldsymbol{F}_{t}^{id})|_{i=1}^{n}. \tag{4}$$

Wherein, location with the maximum value in $\boldsymbol{m}_i$ indicates the predicted state of a previous tracklet. Eq. 4 yields a set of similarity maps $\boldsymbol{M} = \{\boldsymbol{m}_1, \cdots, \boldsymbol{m}_n\}$, in which each denotes the transductive detection result of a previous tracklet. Notably, the modified cross-correlation in our model can be implemented with a simple matrix multiplication. We attach the PyTorch codes in Alg. 1.

We then discretize $\boldsymbol{m}_i$ to a binary mask $\hat{\boldsymbol{m}}_i$ by shrinking the scope of high responses. The underlying reason for this operation is that objects with similar appearance may bring high response. Thus, shrinking the scope of high responses can reduce ambiguous predictions. More formally, the binary mask $\hat{\boldsymbol{m}}_i$ is obtained by,

$$\hat{\boldsymbol{m}}_i^{xy} = \begin{cases} 1 & \text{if } \|x - c_x\| \leq r, \|y - c_y\| \leq r \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\hat{\boldsymbol{m}}_i^{xy}$ denotes the value at (x, y) of $\hat{\boldsymbol{m}}_i$, and c_x, c_y indicate the locations of maximum value in $\boldsymbol{m}_i$. r is the shrinking radius. The region within and outside the square is set to 1 and 0, respectively. Afterwards, we multiply the binary mask $\hat{\boldsymbol{m}}_i$ to the original similarity map $\boldsymbol{m}_i$ to reduce ambiguous responses. Finally, we aggregate the response maps by element-wise summation along channel dimension,

$$\boldsymbol{M}_s = \sum_{i=1}^{n} (\hat{\boldsymbol{m}}_i \cdot \boldsymbol{m}_i). \tag{6}$$

The aggregated similarity map $\boldsymbol{M}_s$ reveals the probability of a location in the current frame that contains a bounding box associated with previous tracklets.

Refinement Module We observed that objects disappearing in the current frame tend to bring false positives during tracklet transduction. To alleviate this issue, we arrange the refinement module to introduce the original visual feature $\boldsymbol{F}_t \in \mathbb{R}^{H \times W \times C}$ (C=256) to provide informative semantics for finer localization. We firstly encode the similarity map $\boldsymbol{M}_s$ with an inverted bottleneck module (Sandler et al. 2018). Concretely, a 3×3 convolution layer maps $\boldsymbol{M}_s$ to high dimensional space, *i.e.*, the channels of 256. Then another 3×3 convolution layer follows to down-sample the channel to 1, as $\boldsymbol{M}_s^{'} \in \mathbb{R}^{H \times W \times 1}$. The refined similarity map $\boldsymbol{M}_s^{'}$ is multiplied by the visual feature $\boldsymbol{F}_t$ to get the enhanced feature $\hat{\boldsymbol{F}} \in \mathbb{R}^{H \times W \times C}$ (C=256),

$$\hat{\boldsymbol{F}} = \boldsymbol{F}_t \cdot \boldsymbol{M}_s^{'}. \tag{7}$$

Later, the enhanced feature $\hat{\boldsymbol{F}}$ passes through several convolution layers to obtain the final prediction $\boldsymbol{M}_p$.

Optimization Besides the loss for the baseline tracker CSTrack (Liang et al. 2020), we introduce a supervised function to train the re-check network. The ground-truth for the similarity map $\boldsymbol{M}_p$ is defined as a combination of multiple Gaussian distributions. Specifically, for each target, its supervised signal is a Gaussian-like mask,

$$\boldsymbol{t}_i = \exp\left(-\frac{(x - c_i^x)^2 + (y - c_i^y)^2}{2\sigma_i^2}\right) \tag{8}$$

where $c_i = (c_i^x, c_i^y)$ denotes the center location of a target and σ_i is the object size-adaptive standard deviation (Law and Deng 2018). Eq. 8 generates a set of ground-truth masks $\boldsymbol{t} = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_n\}$. Then we sum all elements in $\boldsymbol{t}$ along channel dimension to get the supervised signal $\boldsymbol{T}$ for $\boldsymbol{M}_p$. To reduce the overlap between two Gaussian distributions, we set an upper limit of σ_i to 1. We employ the Logistic−MSE Loss (Allen 1971) to train re-check network,

$$\mathcal{L}_g = -\frac{1}{n}\sum_{xy}\begin{cases} \left(1-\boldsymbol{M}_p^{xy}\right)\log\left(\boldsymbol{M}_p^{xy}\right), & \text{if } \boldsymbol{T}^{xy}=1 \\ \left(1-\boldsymbol{T}^{xy}\right)\boldsymbol{M}_p^{xy}\log\left(1-\boldsymbol{M}_p^{xy}\right), & \text{else} \end{cases} \tag{9}$$

where $\boldsymbol{M}^{xy}$ and $\boldsymbol{T}^{xy}$ indicate the value of a location in $\boldsymbol{M}_p$ and $\boldsymbol{T}$, respectively.

Fusing Basic and Transductive Detections

In this section, we detail how to fuse the transductive detections $\boldsymbol{D}_{trans}$ and basic detections $\boldsymbol{D}_{base}$ to get the final candidate boxes $\boldsymbol{D}_{final}$ for association. We firstly calculate the targetness score s for each bounding box $\boldsymbol{b}_i$ in $\boldsymbol{D}_{trans}$ by IOU metric, as

$$s = 1 - \max\left(IOU\left(\boldsymbol{b}_i, \boldsymbol{D}_{base}\right)\right), \tag{10}$$

where a higher s indicates the box $\boldsymbol{b}_i$ does not appear in the basic detections, which is a probable missed bounding box. Then, the boxes with a score above threshold ϵ are retrained as complement of basic detections. We set ϵ to 0.5. When the basic detections miss some targets, the transductive detections can restore them to keep the temporal consistency of tracklets.

Comparisons with Related Works

In this section, we further discuss the differences with other works which share similar spirit with our method.

OMC *vs.* UMA (Yin et al. 2020) and DASOT (Chu et al. 2020). Recent works UMA and DASOT also adopt SOT trackers or mechanism in MOT tracking. However, our method differs from them in two fundamental ways. 1) *Local or Global search.* UMA and DASOT only consider a small neighborhood region when searching a target. However, local search is not effective when fast motion happens. Conversely, in our work, the tracklet transduction is accomplished with global search, which is more robust for fast motion cases. 2) *Unified or Separated Framework.* UMA and DASOT only integrate temporal cues mining and data association into a model, which is separated with the object detector. This is obviously tedious and time-consuming since the raw image input needs to be performed forward inference two or even more times. Differently, in our work, detection-transduction-association are unified in a tracking framework, which obtains all outputs with only one single pass and enjoys easier implementation.

OMC *vs.* Tracktor (Bergmann, Meinhardt, and Leal-Taixe 2019). Both OMC and Tracktor attempt to propagate previous tracklets to the current frame in a simple one-shot framework. Tracktor considers the bounding boxes in the last frame as regions of interest (ROIs) in the current frame, and then extracts features inside the ROIs. The locations of existing targets are predicted by directly regressing the ROI features. However, the tracklet transduction of Tracktor still relies on the single-frame visual cues. Differently, OMC transfers the previous tracklets by measuring ID embedding similarities between the last frame and the current frame. By reusing the object ID embeddings of the last frame as temporal cues, OMC can restore missed targets more effectively.

Experiments

Implementation Details

Baseline Tracker Modification. In the vanilla JDE (Wang et al. 2019) and CSTrack (Liang et al. 2020), the offset between an anchor center $\boldsymbol{a} = (a_x, a_y)$ and the center of corresponding bounding box $\boldsymbol{b} = (b_x, b_y)$ is restricted to $0 \sim 1$ (on the feature map) by the sigmoid function,

$$\boldsymbol{\Delta} = \boldsymbol{b} - \boldsymbol{a} = \mathrm{Sigmoid}(\boldsymbol{r}) \tag{11}$$

where $\boldsymbol{r}$ indicates the network's regression output and $\boldsymbol{\Delta} = (\Delta_x, \Delta_y)$ denotes the predicted offset. However, at the boundary of an image, the offset is often larger than 1. As shown in Fig. 4, the centers of groundtruth boxes (green) are outside the image boundary. However, due to the hard restriction of Sigmoid, the predicted boxes (red) hardly cover the whole objects. When an object appears with only part-body, the incomplete box prediction will be considered as false positive because of the large differences between the ground-truth bounding box and the incomplete box, which eventually degrades tracking performance. To alleviate this issue, we modify the regression mechanism to a boundary-aware regression (BAR) as,

$$\boldsymbol{\Delta} = \boldsymbol{b} - \boldsymbol{a} = \left(\mathrm{Sigmoid}\left(\boldsymbol{r}\right) - 0.5\right) \times h, \tag{12}$$

where h is the learnable scale parameter. The scale parameter allows the network to predict offsets larger than 1. As shown in Fig. 4 (c), BAR is capable of predicting invisible part of the objects based on the visible part.

Training and Testing. We build our tracker by integrating the proposed re-check network into CSTrack (Liang et al. 2020). For the sake of fairness, we use the same training data as CSTrack, including ETH (Ess et al. 2008), CityPerson (Zhang, Benenson, and Schiele 2017), CalTech (Dollár et al. 2009), MOT17 (Milan et al. 2016), CUDK-SYSU (Xiao et al. 2017), PRW (Zheng et al. 2017) and CrowdHuman (Shao et al. 2018). The training procedure consists of two stages, *i.e.*, basic tracker training and re-check network optimization. In the first stage, we equip CSTrack with the Boundary-Aware Regression (basic tracker) and train it following the standard settings of CSTrack. Concretely, the network is trained with a SGD optimizer for 30 epochs. The batch size is 8. The initial learning rate is 5×10^{-4}, and it decays to 5×10^{-5} at the 20^{th} epoch. In the second stage, we train the proposed re-check network while fixing the basic tracker's parameters on MOT17 (Milan et al. 2016) training set. During training, we randomly sample image pairs from adjacent frames in the same video sequence, one for generating exemplar embeddings $\boldsymbol{E}_{t-1}^{id}$ and the other for generating candidate embeddings $\boldsymbol{F}_t^{id}$. Each iteration contains 8 pairs. Other training schedules follow the settings in the first training stage.

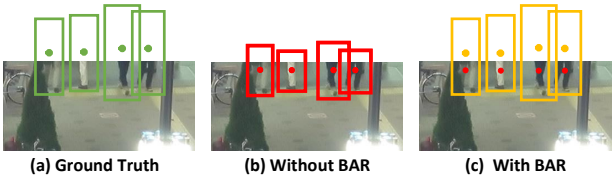

Figure 4: Visualization detection results at the boundary. (a) Ground-truth bounding boxes. (b) The incomplete bounding box prediction. (c) The bounding box prediction with boundary-aware regression (BAR). The red points represent the anchor centers and green/yellow points represent centers of the bounding boxes.

	R	BAR	MOTA↑	IDF1↑	MT↑	FP↓	FN↓	FPS↑	Param
①			70.6	71.6	37.5	24804	137832	**15.8**	74.6M
②	✓		75.5	72.9	42.0	27334	107284	13.3	77.5M
③		✓	73.1	72.4	39.9	**19772**	128184	15.2	74.6M
④	✓	✓	**76.3**	**73.8**	**44.7**	28894	**101022**	12.8	77.5M

Table 1: Component-wise analysis of the proposed model.

We set r in Eq. 5 to 3 and initialize the scale parameter h in Eq. 12 to 10. Other hyperparameters and testing stage follow settings in CSTrack without other specifications.

Our tracker is implemented using Python 3.7 and PyTorch 1.6.0. The experiments are conducted on a single RTX 2080Ti GPU and Xeon Gold 5218 2.30GHz CPU.

Evaluation Datasets and Metrics. We evaluate our tracker on three MOT Challenge benchmarks, *i.e.,* MOT16 (Milan et al. 2016), MOT17 (Milan et al. 2016) and the recent released MOT20 (Dendorfer et al. 2020). Following the common practices in MOT Challenge (Milan et al. 2016), we employ the CLEAR metric (Bernardin and Stiefelhagen 2008), particularly MOTA (the primary metric of MOT) and IDF1 (Ristani et al. 2016) to evaluate the overall performance. We also report other common metrics for evaluation, which include the ratio of Most Tracked targets (MT), the ratio of Most Lost targets (ML) , False Positives (FP), False Negatives (FN) and running speed (FPS).

Analysis of the Proposed Method

Component-wise Analysis. To verify the efficacy of the proposed method, we perform a component-wise analysis on MOT17 testing set, as presented in Tab. 1. When equipping the baseline tracker (①) with the proposed re-check network (R), it significantly decreases FN from 137832 to 107284 (② *vs.* ①), which achieves favorable 4.9 points gains on MOTA and 4.5 points gains on MT. This confirms the effectiveness of the re-check network on restoring "fake background". The introduced boundary-aware regression (BAR) aims to reason the invisible part of objects when they appear at the boundary of image. Tab. 1 shows that the BAR brings gains of 2.5 points on MOTA and 0.8 points on IDF1 (③ *vs.* ①), respectively. Overall (① *vs.* ④), compared with the baseline tracker, our model significantly improves tracking performance, *i.e.*, MOTA +5.7 points, IDF1 +2.2 points and MT +7.2 points, with a small overhead, *i.e.*, 12.8 FPS *vs.* 15.8 FPS and model parameters 77.5M *vs.* 74.6M.

	Method	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓
①	Baseline-BAR	73.1	72.4	39.9	16.4	**19772**	128184
②	+ R *w/o* Global	75.4	73.2	43.4	15.4	31013	103766
③	+ R *w/o* Shrink	73.6	72.6	**47.8**	**11.2**	48915	**95829**
④	+ R *w/o* $\boldsymbol{F}_t$	69.3	70.7	45.2	13.5	67885	100793
⑤	+ R *w/o* IBM	75.7	73.4	45.1	13.5	32005	100590
⑥	+ R	**76.3**	**73.8**	44.7	13.6	28894	101022

Table 2: Analysis of re-check network on MOT17 testing set. ① indicates our baseline (CSTrack) with BAR. While with "w/o" means that the method discards this module.

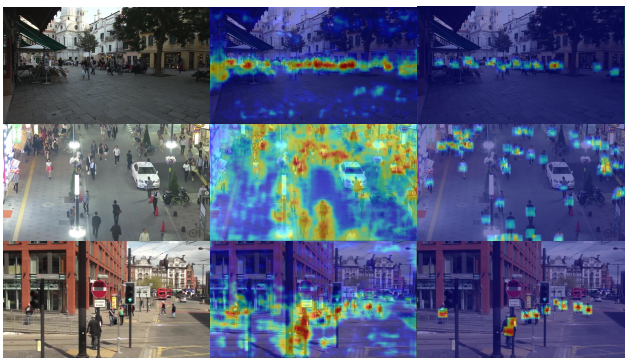

Figure 5: Visualization of the response maps with (c) and without (b) shrinking on MOT17 dataset. To make it clear, we show the corresponding original image in (a).

Understanding the Re-check Network. To understand the impact of the re-check network, we evaluate the tracker (Baseline-BAR) with different variants of the re-check network, as shown in Tab. 2. Firstly, we replace the global search with the standard local search, *i.e.*, considering the neighborhood region of previous targets in the last frame (Chu et al. 2020). Comparing the results of ② and ⑥, we find that with a global view, our tracker can more accurately propagate previous tracklets to current frame, which achieves better tracking performance on FP, FN, MOTA and IDF1 scores. Secondly, we discard the shrinking operation in Eq. 5. As the result shown in ③, without shrinking, the FP number dramatically increases, which eventually causes the decrease of MOTA score. We visualize the shrinking and non-shrinking response maps in Fig. 5, which shows that the shrinking operation can filter most false-positive responses and effectively keep the transductions of previous targets. Furthermore, we conduct two ablation experiments to prove the rationality of the refinement module design, as shown in ④ and ⑤. During calculating the similarity map $\mathbf{M}_p$, we involve visual feature $\mathbf{F}_t$ to mitigate false-positive transduction. The result of ④ verifies that introducing $\boldsymbol{F}_t$ can effectively decrease FP number, *i.e.*, 67885 → 28894. When we discard the inverted bottleneck module (IBM), the MOTA score decreases from 76.3 to 75.7 (⑤ *vs.* ⑥). It confirms that the precoding of similarity map and the semantic information in visual feature can complement each other for better tracking performance.

Method	Published	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	FPS↑
MOT16								
POI (Yu et al. 2016)	ECCV16	66.1	65.1	34.0	21.3	**5061**	55914	<5.2
DeepSORT-2 (Wojke, Bewley, and Paulus 2017)	ICIP17	61.4	62.2	32.8	18.2	12852	56668	<6.7
HOGM (Zhou et al. 2018)	ICPR18	64.8	73.5	40.6	22.0	13470	49927	<8.0
TubeTK (Pang et al. 2020)	CVPR20	64.0	59.4	33.5	19.4	11544	47502	1.0
CTracker (Peng et al. 2020)	ECCV20	67.6	57.2	32.9	23.1	8934	48305	6.8
QDTrack (Pang et al. 2021)	CVPR21	69.8	67.1	41.7	19.8	9861	44050	14∼30
TraDeS (Wu et al. 2021)	CVPR21	70.1	64.7	37.3	20.0	8091	45210	15
FairMOT (Zhang et al. 2020a)	IJCV21	74.9	72.8	44.7	15.9	10163	34484	**18.9**
JDE (Wang et al. 2019)	ECCV20	64.4	55.8	35.4	20.0	10642	52523	18.5
CSTrack (Liang et al. 2020)	Arxiv20	70.7	71.8	38.2	17.8	10286	41974	15.8
OMC	Ours	**76.4**	**74.1**	**46.1**	**13.3**	10821	**31044**	12.8
MOT17								
TubeTK (Pang et al. 2020)	CVPR20	63.0	58.6	31.2	19.9	27060	177483	3.0
CTracker (Peng et al. 2020)	ECCV20	66.6	57.4	32.2	24.2	22284	160491	6.8
CenterTrack (Zhou, Koltun, and Krähenbühl 2020)	ECCV20	67.8	64.7	34.6	24.6	**18498**	160332	**22.0**
QDTrack (Pang et al. 2021)	CVPR21	68.7	66.3	40.6	21.8	26589	146643	14∼30
TraDeS (Wu et al. 2021)	CVPR21	69.1	63.9	36.4	21.5	20892	150060	15
FairMOT (Zhang et al. 2020a)	IJCV21	73.7	72.3	43.2	17.3	27507	117477	18.9
CSTrack (Liang et al. 2020)	Arxiv20	70.6	71.6	37.5	18.7	24804	137832	15.8
OMC	Ours	**76.3**	**73.8**	**44.7**	**13.6**	28894	**101022**	12.8
MOT20								
FairMOT (Zhang et al. 2020a)	IJCV21	61.8	67.3	**68.8**	**7.6**	103440	**88901**	**8.4**
OMC	Ours	**70.7**	**67.8**	56.6	13.3	**22689**	125039	6.7

Table 3: Comparison with the state-of-the-art online MOT systems under private detection protocol. We report the corresponding official metrics. ↑/↓ indicate that higher/lower is better, respectively. For a fair comparison, we obtain FPS of each method under the same experimental conditions. The best scores of methods are marked in **bold**.

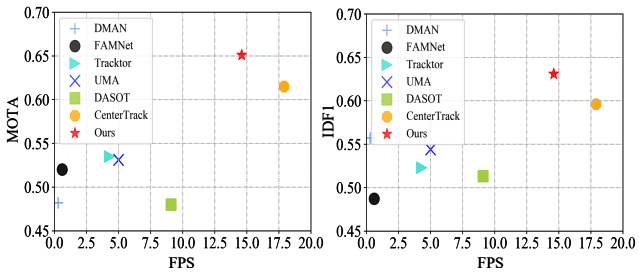

Figure 6: Tracking performance (MOTA and IDF1) and tracking speed (FPS) of the proposed method and other MOT methods on MOT17 benchmark.

Comparison on MOT Benchmarks

State-of-the-art Comparison. We compare OMC with other state-of-the-art MOT methods on the testing sets of MOT16, MOT17 and MOT20. For evaluating on the MOT20 benchmark, we fine-tune it on the training set of MOT20 following the same training procedure. As shown in Tab. 3, our tracker achieves new state-of-the-art MOTA and IDF1 scores on all three benchmarks. Specifically, the proposed OMC outperforms the recent state-of-the-art tracker FairMOT (Zhang et al. 2020a) by 1.5 $\sim$ 8.9 points on MOTA.

Methods with Temporal Cues Mining. To better illustrate the effectiveness of the proposed method, we compare our tracker with the advanced MOT methods using temporal cues under the same public detection protocol (Milan et al. 2016). The compared methods are divided into two categories, *i.e.*, one using SOT trackers which include DMAN (Zhu et al. 2018), FAMNet (Chu and Ling 2019), UMA (Yin et al. 2020) and DASOT (Chu et al. 2020), and the other directly propagating previous tracklets by predicting the offset of bounding boxes in the last frame, *i.e.*, Tracktor (Bergmann, Meinhardt, and Leal-Taixe 2019) and CenterTrack (Zhou, Koltun, and Krähenbühl 2020). As shown in Fig. 6, our method runs faster (14.6 FPS *vs.* 0.3 $\sim$ 9.1 FPS) and achieves better tracking performance compared with other methods using SOT trackers. Moreover, comparing our tracker with methods directly propagating previous tracklets, our method still gains best tracking performance on MOTA and IDF1.

Conclusion

This work has presented a novel double-check approach for MOT, to reload the "fake background" that is caused by the detector's over-reliance on the single-frame visual cues. Unlike prior attempts, we propose a novel re-check network, which can mining temporal cues with a small overhead. Concretely, we expand the role of ID embeddings from data association to motion forecasting and propagate the previous tracklets to the current frame using global embedding search. Based on this, we construct a new one-shot MOT tracker, namely OMC, which integrates detection, embedding extraction and temporal cues mining into a unified framework. The quantitative experimental results have shown that the re-check network can restore the targets missed by the detector more effectively than the prior methods. OMC is simple, efficient, and achieves new state-of-the-art performance on MOT16, MOT17, and MOT20.

Acknowledgments

This work was supported by the Natural Science Foundation of China (No. 61972071, U20A20184), the Sichuan Science and Technology Program (2020YJ0036), the 2019 Fundamental Research Funds for the Central Univeasities, the Research Program of Zhejiang Lab (2019KDAB02), the Open Project Program of the National Laboratory of Pattern Recognition (201900014), and Grant SCITLAB-1005 of Intelligent Terminal Key Laboratory of SiChuan Province.

References

Allen, D. M. 1971. Mean square error of prediction as a criterion for selecting variables. *Technometrics*, 13(3): 469–475.

Bergmann, P.; Meinhardt, T.; and Leal-Taixe, L. 2019. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 941–951.

Bernardin, K.; and Stiefelhagen, R. 2008. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008: 1–10.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, 850–865. Springer.

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. 2016. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, 3464–3468. IEEE.

Chen, L.; Ai, H.; Zhuang, Z.; and Shang, C. 2018. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE.

Chu, P.; and Ling, H. 2019. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6172–6181.

Chu, Q.; Ouyang, W.; Li, H.; Wang, X.; Liu, B.; and Yu, N. 2017. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*, 4836–4845.

Chu, Q.; Ouyang, W.; Liu, B.; Zhu, F.; and Yu, N. 2020. Dasot: A unified framework integrating data association and single object tracking for online multi-object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10672–10679.

Dehghan, A.; Tian, Y.; Torr, P. H.; and Shah, M. 2015. Target identity-aware network flow for online multiple target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1146–1154.

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

Dollár, P.; Wojek, C.; Schiele, B.; and Perona, P. 2009. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 304–311. IEEE.

Ess, A.; Leibe, B.; Schindler, K.; and Van Gool, L. 2008. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE.

Fang, K.; Xiang, Y.; Li, X.; and Savarese, S. 2018. Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 466–475. IEEE.

Hornakova, A.; Henschel, R.; Rosenhahn, B.; and Swoboda, P. 2020. Lifted disjoint paths with application in multiple object tracking. In *International Conference on Machine Learning*, 4364–4375. PMLR.

Law, H.; and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, 734–750.

Liang, C.; Zhang, Z.; Lu, Y.; Zhou, X.; Li, B.; Ye, X.; and Zou, J. 2020. Rethinking the competition between detection and ReID in Multi-Object Tracking. *arXiv preprint arXiv:2010.12138*.

Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; and Kim, T.-K. 2020. Multiple object tracking: A literature review. *Artificial Intelligence*, 103448.

Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; and Schindler, K. 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.

Pang, B.; Li, Y.; Zhang, Y.; Li, M.; and Lu, C. 2020. TubeTK: Adopting tubes to track multi-object in a one-step training model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6308–6318.

Pang, J.; Qiu, L.; Li, X.; Chen, H.; Li, Q.; Darrell, T.; and Yu, F. 2021. Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 164–173.

Peng, J.; Wang, C.; Wan, F.; Wu, Y.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; and Fu, Y. 2020. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *European Conference on Computer Vision*, 145–161. Springer.

Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2016. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6): 1137–1149.

Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, 17–35. Springer.

Ristani, E.; and Tomasi, C. 2018. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6036–6046.

Sadeghian, A.; Alahi, A.; and Savarese, S. 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, 300–311.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

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

Tang, S.; Andriluka, M.; Andres, B.; and Schiele, B. 2017. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3539–3548.

Wang, Z.; Zheng, L.; Liu, Y.; and Wang, S. 2019. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2(3): 4.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, 3645–3649. IEEE.

Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; and Yuan, J. 2021. Track to Detect and Segment: An Online Multi-Object Tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12352–12361.

Xiao, T.; Li, S.; Wang, B.; Lin, L.; and Wang, X. 2017. Joint detection and identification feature learning for person search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3415–3424.

Xu, J.; Cao, Y.; Zhang, Z.; and Hu, H. 2019. Spatial-temporal relation networks for multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3988–3998.

Yin, J.; Wang, W.; Meng, Q.; Yang, R.; and Shen, J. 2020. A unified object motion and affinity model for online multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6768–6777.

Yu, F.; Li, W.; Li, Q.; Liu, Y.; Shi, X.; and Yan, J. 2016. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, 36–42. Springer.

Zhang, S.; Benenson, R.; and Schiele, B. 2017. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3213–3221.

Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; and Liu, W. 2020a. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *arXiv e-prints*, arXiv–2004.

Zhang, Z.; Liu, Y.; Wang, X.; Li, B.; and Hu, W. 2021. Learn to match: Automatic matching network design for visual tracking. *arXiv preprint arXiv:2108.00803*.

Zhang, Z.; and Peng, H. 2019. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 4591–4600.

Zhang, Z.; Peng, H.; Fu, J.; Li, B.; and Hu, W. 2020b. Ocean: Object-aware Anchor-free Tracking. In *ECCV*.

Zheng, L.; Zhang, H.; Sun, S.; Chandraker, M.; Yang, Y.; and Tian, Q. 2017. Person re-identification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1367–1376.

Zhou, X.; Koltun, V.; and Krähenbühl, P. 2020. Tracking objects as points. In *European Conference on Computer Vision*, 474–490. Springer.

Zhou, Z.; Xing, J.; Zhang, M.; and Hu, W. 2018. Online multi-target tracking with tensor-based high-order graph matching. In *2018 24th International Conference on Pattern Recognition (ICPR)*, 1809–1814. IEEE.

Zhu, J.; Yang, H.; Liu, N.; Kim, M.; Zhang, W.; and Yang, M.-H. 2018. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 366–382.