This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3140919, IEEE Transactions on Multimedia

Split and Connect: A Universal Tracklet Booster for Multi-Object Tracking

Gaoang Wang, *Member, IEEE,* Yizhou Wang, *Student Member, IEEE,* Renshu Gu, *Member, IEEE,* Weijie Hu, and Jenq-Neng Hwang, *Fellow, IEEE*

***Abstract*—Multi-object tracking (MOT) is an essential task in the computer vision field. With the fast development of deep learning technology in recent years, MOT has achieved great improvement. However, some challenges still remain, such as sensitiveness to occlusion, instability under different lighting conditions, and non-robustness to deformable objects, causing incorrect temporal associations. To address such common challenges in most of the existing trackers, in this paper, a tracklet booster (TBooster) algorithm is proposed to correct the association errors resulting from existing trackers. The correction of the association error from TBooster has two folds: split tracklets on potential ID-change positions and then connect multiple tracklets into one if they are from the same object. To achieve this goal, the TBooster consists of two components, *i.e.*, Splitter and Connector. In Splitter, an architecture with stacked temporal dilated convolution blocks is employed for the splitting position prediction via label smoothing strategy with adaptive Gaussian kernels. In Connector, a multi-head self-attention-based encoder is exploited for the tracklet embedding, which is further used to connect tracklets into full tracks. We conduct sufficient experiments on MOT17 and MOT20 benchmark datasets and achieve promising results. Combined with the proposed tracklet booster, existing trackers can achieve large improvements on the IDF1 score, which shows the effectiveness of the proposed TBooster.**

***Index Terms*—multi-object tracking, embedding, attention**

I. Introduction

MULTI-OBJECT tracking (MOT) has drawn great attention in recent years. This technique is critically needed in many tasks, such as traffic flow analysis [1]–[4], human behavior prediction and pose estimation [5]–[10], autonomous driving assistance [11], [12], and even for underwater animal abundance estimation [13]–[15]. Recent years have seen the emergence of a variety of tracking algorithms, from graph clustering methods [1], [16]–[18] to graph neural networks [19]–[22] that aggregate information across frames and objects; from tracking-by-detection paradigm [23]–[25] to joint detection and tracking [19], [26]–[30] to improve the detection performance with multiple frames; from Kalman filtering [31] to recurrent neural networks (RNN) [32] and long-short term memory (LSTM) [33] to boost association performance with the motion clue. However, due to the noisy visual object detection and occlusion, tracking multiple objects over a long time is yet very challenging.

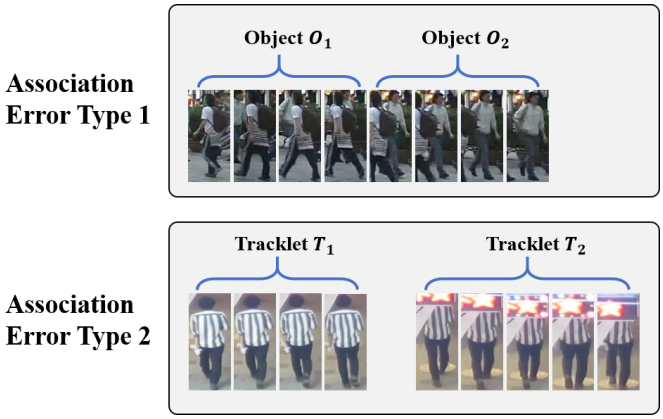


Fig. 1. Demonstration of association errors. The first error type shows multiple objects are associated with the same tracklet. The second error shows the same object are assigned to different tracklets.

Typically, the tracking error comes from two parts: object detection and temporal association. To measure the performance of trackers, three main evaluation metrics, *i.e.*, multi-object tracking accuracy (MOTA) [34], IDF1 score [35], and higher-order tracking accuracy (HOTA) [36], are widely used in the MOT field. As demonstrated in [36], MOTA emphasizes detection accuracy while IDF1 focuses more on association measurement. As shown in Fig. 1, generally, the association errors can be concluded into two categories: 1) different objects are associated with the same tracklet, 2) tracklets from the same object are assigned to different IDs. A good tracker should reduce these two types of errors as much as possible in the association task. Due to missing detection, change of lighting condition, camera movement, occlusion, and object deformation, how to achieve accurate association is one of the major challenges of almost all existing trackers. Some works [37]–[39] focus on addressing association errors or designing more comprehensive metrics to measure association performance. Specifically, [37] presents a changing point detection with the forward and backward validation. It is simple but not learnable and does not merge among small tracklets. [38] refines the detection with a designed spatial conflict graph and applies a switcher-aware association between two frames. However, it fails to take account of long temporal dependencies.

Gaoang Wang is with the Zhejiang University - University of Illinois Urbana-Champaign Institute (ZJU-UIUC), and College of Computer Science and Technology, Zhejiang University, China. E-mail: gaoangwang@intl.zju.edu.cn.

Yizhou Wang and Jenq-Neng Hwang are with the University of Washington, United States.

Renshu Gu is with Hangzhou Dianzi University, China.

Weijie Hu is with Guangdong University of Petrochemical Technology, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3140919, IEEE Transactions on Multimedia

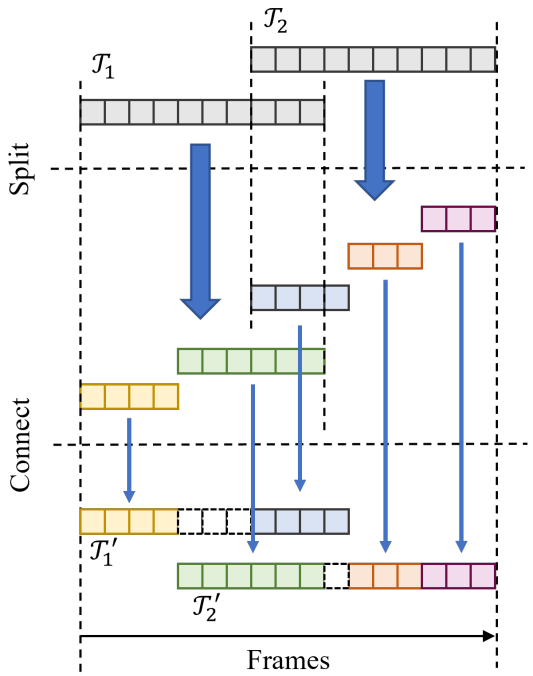


Fig. 2. Demonstration of the proposed tracklet booster. In this example, two tracklets $\mathcal{T}_1$ and $\mathcal{T}_2$ are first split into five small pieces of tracklets. Then they are re-connected into two new tracklets $\mathcal{T}_1^{'}$ and $\mathcal{T}_2^{'}$. The horizontal axis represents the aligned frames.

In this paper, we propose a novel method, tracklet booster (TBooster), that directly targets the two types of association error to boost the performance on IDF1 score, without tackling much for the detection error. Also, it can be efficiently plugged into any existing trackers. TBooster has two main modules, Splitter and Connector. Splitter aims at finding potential ID-change positions of a tracklet and split the tracklet into small pieces at the detected ID-change positions. Splitter is designed with stacked dilated temporal convolution blocks to measure the temporal consistency of a tracklet. An adaptive label smoothing strategy with Gaussian kernels is proposed to improve the stability of the model training. With Splitter, the first type of error that multiple objects are assigned with the same ID can be largely reduced. On the other hand, Connector is introduced to address the second type of association error that the same identity is assigned to multiple tracklets. Connector aims at distinguishing different objects and merging multiple tracklets into one if they are from the same object. Specifically, Connector is modeled as a tracklet embedding network. Tracklets with small embedding distances are grouped and assigned with the same tracking ID. Inspired by the transformer [40], we use the multi-head self-attention mechanism to learn the tracklet embedding. A demonstration of the tracklet booster is shown in Fig. 2 and the details of the framework are shown in Fig. 3.

We summarize our contributions as follows:

- We propose a novel tracklet boosting model, consisting of Splitter and Connector, to directly address the temporal association errors that exist in almost all trackers in the MOT field. Besides, the proposed TBooster can be integrated with any existing trackers to significantly improve their tracking performance.
- A novel adaptive label smoothing strategy with Gaussian kernels is proposed in Splitter to predict the potential ID-change positions within the given tracklet.
- A multi-head self-attention based encoder is employed for tracklet embedding, which serves as the main module of the proposed Connector.
- We conduct experiments on MOT17 and MOT20 benchmark datasets and prove the generality, effectiveness, and robustness of our tracklet boosting method.

The outline of the paper is as follows: In Section II, we review the state-of-the-art (SOTA) related works in MOT. Section III describes the details of the proposed Splitter and Connector, including architecture, loss function, and inference algorithms. The experiments and discussions are provided in Section IV, followed by the conclusion and future works in Section V.

II. Related Work

A. Joint Learning of Multiple Cues

Multiple cues, such as appearance, motion, and interaction, can provide important information for MOT [41]–[44]. For example, the interaction cues among different objects are considered in the relationship among neighboring targets in a crowd or a group [41]. [42] employs a spatial-temporal relation module that jointly learns appearance and topology in a unified network. [43] presents a single network that unifies object motion and affinity model in a multi-task learning framework. Several methods combine other cues for refinement. For example, [44] applies articulation detection for detection refinement to improve the performance of tracking.

B. Tracking with Graph Models

Graph models [1], [16]–[18], [45]–[54] are widely used in MOT for the temporal association. The association is traditionally solved by optimizing the total cost or energy function. For example, [17] formulates MOT as a minimization of a continuous energy function. [16] proposes a novel graph-based formulation that links and clusters person hypotheses over time by solving an instance of a minimum cost lifted multi-cut problem. [49] proposes an end-to-end framework for learning parameters of min-cost flow MOT problem with quadratic trajectory interactions including suppression of overlapping tracks and contextual cues about the co-occurrence of different objects. [53] proposes an extension to the disjoint paths problem in which additional lifted edges are introduced to provide path connectivity priors. [50] proposes an efficient online min-cost flow tracking algorithm with bounded memory and computation. In addition to the single-view tracking, graph-based methods are also explored in multi-view tracking tasks [55]–[57], where the multi-view tracking is formulated as a graph clustering problem. Usually, detections or tracklets are adopted as graph nodes. Then, the similarities among nodes are measured on the connected edges. For detection-based graphs, the temporal information is not well utilized

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3140919, IEEE Transactions on Multimedia

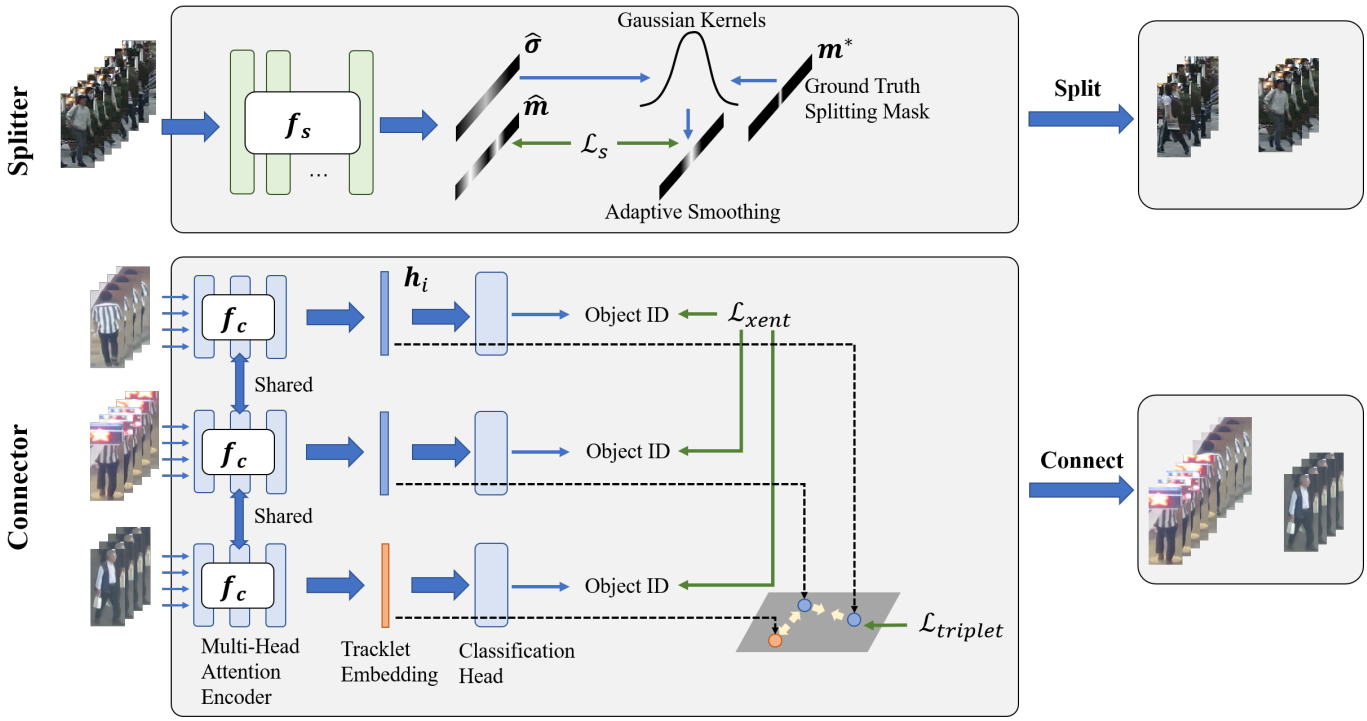


Fig. 3. The flowchart of the proposed tracklet booster. The top part and bottom part show the process of Splitter and Connector, respectively. f_s and f_c represent the dilated temporal convolution model and multi-head self-attention based encoder used in Splitter and Connector, respectively. $\hat{\boldsymbol{\sigma}}$ is the predicted standard deviation for adaptive smoothing; $\hat{\boldsymbol{m}}$ is the predicted splitting position mask; $\boldsymbol{m}^*$ is the ground truth splitting mask; $\mathcal{L}_s$, $\mathcal{L}_{xent}$ and $\mathcal{L}_{triplet}$ are mean squared error, cross-entropy loss and triplet loss, respectively.

and usually comes with a remarkably high dimensional affinity matrix with a heavy computational cost. The tracklet-based graph, on the other hand, is much more efficient and also incorporates global temporal information. However, the conventional graph models, based on optimization, usually suffer from the empirically setting of hand-crafted features. Moreover, representative embeddings are not well-explored for the temporal association.

Since the graph neural networks (GNN) show great power recently, many approaches [19]–[22], [58] adopt GNN for the association, rather than using conventional graph models based on optimization. Specifically, [20] exploits the classical network flow formulation of MOT to define a fully differentiable framework based on message passing networks. [21] presents an adaptive graph neural network to fuse locations, appearance, and historical information for MOT. [22] proposes a novel feature interaction mechanism based on the GNN to learn the interaction among objects. [58] presents a novel learnable graph matching method to address association issues in MOT. [59] proposes a novel proposal-based learnable framework, which models MOT as a proposal generation, proposal scoring, and trajectory inference paradigm on an affinity graph. The existing methods show the effectiveness of employing GNN to MOT. Moreover, GNN also shows power in other related vision tasks, such as human action recognition [60], visual question answering [61], and single object tracking [62]. However, GNN-based methods often suffer from over-smooth issues when a deep architecture is employed.

C. Joint Detection and Tracking

More recently, joint detection and tracking-based methods have drawn great attention [19], [26]–[30], [63], [64]. Usually, such trackers take sequential adjacent frames as input. Features are aggregated in different frames, and bounding box regression is conducted with temporal information. For example, Tracktor [29] applies the bounding box regression to refine the box position in the next frame to form the track; JDE [30] incorporates the appearance embedding model into a single-shot detector that can simultaneously output detections and the corresponding embeddings; while CenterTrack [26] applies a detection model to a pair of images and detections from the prior frame. Besides that, CTracker [27] constructs tracklets by chaining paired boxes in every two frames. TubeTK [28] directly predicts a box tube as a tracklet in an offline manner. [63] proposes a quasi-dense similarity learning approach that densely samples hundreds of region proposals on a pair of images for contrastive learning. [64] learns a unified network for both detection and Re-ID via a multi-task learning framework. However, due to the heavy computational cost, the networks can only take a very limited number of frames as input. However, such methods usually suffer long-time occlusions, which further results in temporal association error.

D. Tracking with Visual Transformers

Recently, due to the non-local attention mechanism, transformers [40] show great success in many visual tasks, such

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3140919, IEEE Transactions on Multimedia

TABLE I
NOTATIONS USED IN TBOOSTER

Symbol	Description
f_s	Splitter model.
f_c	Connector model.
$\boldsymbol{m}$	ID-change indicator mask.
$\boldsymbol{\sigma}$	Standard deviation of Gaussian density function.
$\boldsymbol{W}$	Kernel weights in deep networks.
$\boldsymbol{h}$	Tracklet embedding.
$\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}$	Query, Key and Value of multi-head attention layers.
$\boldsymbol{A}$	Attention maps.
α	Margin for triplet loss.
δ_s	Threshold of Splitter model.
δ_c	Threshold of Connector model.
δ_t	Threshold of temporal gap between tracklets.
T	Size of the temporal sliding window.
D	Channel dimension.
$\mathcal{T}$	Tracklet.
$\mathcal{S}_{\mathcal{T}}$	Tracklet set.
$\mathcal{G}(\mathcal{V}, \mathcal{E})$	Tracklet graph for Connector, where $\mathcal{V}$ and $\mathcal{E}$ are vertex set and edge set, respectively.
SA	Self-attention.
MSA	Multi-head self-attention.
GAP	Global average pooling.
norm_{l_2}	l_2 normalization.

as image classification [65], object detection [66], [67], 3D human pose estimation [68], and low-level image processing [69]. Moreover, several methods with visual transformers [70]–[72] are explored in the MOT field. For example, [70] proposes a baseline tracker via a transformer, which takes advantage of the query-key mechanism and introduces a set of learned object queries into the pipeline to enable detecting new-coming objects. [71] extends the DETR object detector [66] and achieves a seamless data association between frames in a new tracking-by-attention paradigm by encoder-decoder and self-attention mechanisms. [72] leverages graph transformers to efficiently model the spatial and temporal interactions among the objects. However, transformer-based trackers usually require pre-training on large-scale datasets.

III. METHOD

A. Overview

The motivation of the TBooster is straightforward. We take the tracklets from any tracklet generators [13], [48], [73] or preliminary tracking results as the input. Due to matching errors, a tracklet may contain multiple object IDs. To clean the IDs within a tracklet, Splitter is proposed to split tracklets into small pieces on the potential ID-change positions to ensure split tracklets have purer IDs as much as possible. Next, the split tracklets are sent into Connector to learn representative embeddings. Finally, the tracklets with similar embeddings are grouped to form clusters, *i.e.*, to generate the final entire tracks. The framework of the TBooster is shown in Fig. 3. We summarize the notations used in this paper in Table I. The details of Splitter and Connector are demonstrated in the following sub-sections.

B. Splitter

In this sub-section, we demonstrate the proposed Splitter, which is designed to predict the potential ID-change positions and split the tracklets. Denote the input tracklets as $\boldsymbol{x}$ with the dimension $K \times T$, where K is the feature dimension, and T is the temporal length of a tracklet. Denote the ID-change position mask as $\boldsymbol{m}$ with the dimension $T-1$, where $\boldsymbol{m}$ is set as follows,

$$\boldsymbol{m}_t = \begin{cases} 1, & \text{if } \text{ID}_t \neq \text{ID}_{t+1}; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The goal of Splitter is to estimate the ID-change position mask given the input tracklets $\boldsymbol{x}$, *i.e.*,

$$\hat{\boldsymbol{m}} = f_s(\boldsymbol{x}), \tag{2}$$

where $f_s(\cdot)$ is the proposed Splitter.

To learn Splitter, a loss function, *i.e.*, $\mathcal{L}_s = l(\hat{\boldsymbol{m}}, \boldsymbol{m}^*)$, needs to be defined between the predicted ID-change position mask $\hat{\boldsymbol{m}}$ and the ground truth ID-change position mask $\boldsymbol{m}^*$. As shown in Fig. 4, the object embeddings in adjacent frames usually change gradually rather than changing abruptly, even in the ID-change positions. However, the ground truth mask is hard labeled with zeros and ones. Such inconsistency between hard labels and features will harm the stability of the model training with the commonly used mean squared error (MSE), *i.e.*, $\mathcal{L}_s = ||\hat{\boldsymbol{m}} - \boldsymbol{m}^*||^2$. As a result, it would be more appropriate to use soft labels rather than hard labels in the loss function design.

To incorporate the soft label, we adopt Gaussian smoothing in the mean squared error as the loss function as follows,

$$\mathcal{L}_s = \sum_t \left(\hat{\boldsymbol{m}}_t - \min\left(\sum_\tau \boldsymbol{m}^*_\tau \exp\left(-\frac{(\tau - t)^2}{\sigma^2} \right), 1 \right) \right)^2, \tag{3}$$

where t and τ are the frame index for the predicted mask $\hat{\boldsymbol{m}}$ and ground truth mask $\boldsymbol{m}^*$, respectively; σ is the standard deviation that controls the smoothness of the label. The ground truth mask $\boldsymbol{m}^*$ is used as an indicator for the summation of Gaussian kernels. We also use $\min(\cdot, 1)$ to constrain the smoothed mask labels in the range $[0, 1]$.

In different scenarios, the gradual change of embeddings when the ID change happens can last from a few frames to tens of frames. In other words, σ should be dependent on the duration of the ID-change transition. Setting a fixed σ in the model would derive a sub-optimal solution. To deal with such an issue, we propose an adaptive Gaussian smoothing strategy. Along with the prediction of ID-change position mask $\hat{\boldsymbol{m}}$, we also predict a time and tracklet dependent $\hat{\boldsymbol{\sigma}}$ simultaneously via Splitter as follows,

$$[\hat{\boldsymbol{m}}, \hat{\boldsymbol{\sigma}}] = f_s(\boldsymbol{x}), \tag{4}$$

where $\hat{\boldsymbol{\sigma}}$ has the same dimension as $\hat{\boldsymbol{m}}$. Based on the time dependent $\hat{\boldsymbol{\sigma}}$, we modify the loss function as follows,

$$\mathcal{L}_s = \sum_t \left(\hat{\boldsymbol{m}}_t - \min\left(\sum_\tau \boldsymbol{m}^*_\tau \exp\left(-\frac{(\tau - t)^2}{\tilde{\boldsymbol{\sigma}}^2_\tau} \right), 1 \right) \right)^2. \tag{5}$$

where we set $\tilde{\boldsymbol{\sigma}} = \max(\min(\hat{\boldsymbol{\sigma}}, 10), \epsilon)$ to avoid irregular predictions. Typically, ϵ is set to be a small positive scalar, *e.g.*, 0.001. With adaptive soft labels, Splitter model can predict ID-change positions with different durations of transitions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3140919, IEEE Transactions on Multimedia

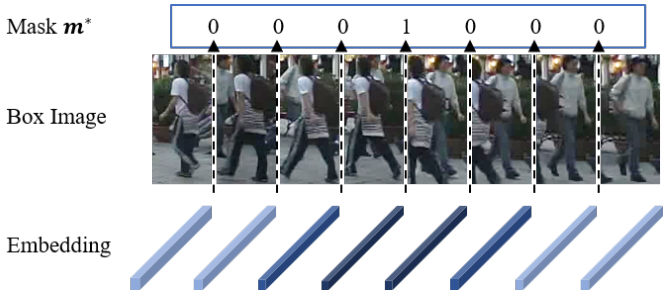


Fig. 4. An example of the transition of the ID change. Usually, the embeddings around the ID change positions change gradually. Thus, the hard label of the ground truth splitting position mask $\boldsymbol{m}^*$ can cause instability in the training.

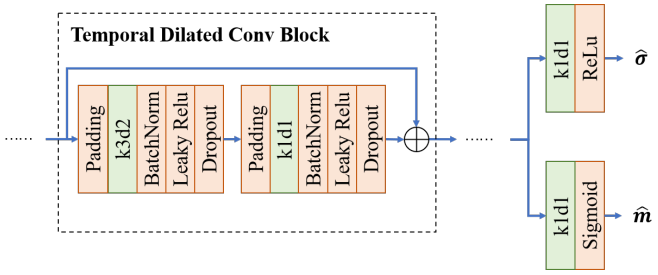


Fig. 5. The architecture of the Splitter model. The box with dashed lines represents each temporal dilated convolution block. "k3d2" means the temporal convolution with kernel size 3 and dilation rate 2. "$\oplus$" represents the summation operation.

The architecture of Splitter is designed as follows. Commonly used stacked dilated temporal convolution and point-wise convolution blocks are employed as the backbone for the feature extraction. Then two fully connected layers are used for ID-change position mask and standard deviation prediction, respectively. We also add a skip connection for each intermediate block. In total, we stack 24 blocks. As a result, the receptive field is large enough to capture the long-term temporal patterns to detect the splitting positions. The architecture of Splitter is shown in Fig. 5.

C. Connector

In this sub-section, we demonstrate the proposed Connector, which is designed to connect multiple tracklets if they are from the same object. We formulate the tracklet connection as a tracklet embedding problem as follows,

$$\boldsymbol{h} = f_c(\boldsymbol{x}), \tag{6}$$

where f_c is the proposed Connector model, $\boldsymbol{h}$ is the tracklet embedding with $\boldsymbol{h} \in \mathbb{R}^D$. After embedding is learned, tracklets from the same object should have smaller embedding distances, while the tracklets from distinct objects should have larger distances.

Due to the sensitivity of occlusion, the difference of lighting condition, deformation in object pose, features from tracklets may have much difference even though they are from the same object. To address such issues, self-attention is an appropriate strategy for tracklet embedding. Inspired by the transformer [40], a multi-head self-attention mechanism is employed for tracklet embedding in the proposed Connector model. The embedding framework is shown in Fig. 6.

Specifically, given the tracklet appearance features $\boldsymbol{x} \in \mathbb{R}^{K\times T}$, the self-attention (SA) is defined as follows,

$$\begin{aligned} [\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}] &= \boldsymbol{x}^T \boldsymbol{W}, \\ \boldsymbol{A} &= \text{softmax}\left(\boldsymbol{q}\boldsymbol{k}^T / \sqrt{D_h}\right), \\ \text{SA}(\boldsymbol{x}) &= \boldsymbol{A}\boldsymbol{v}, \end{aligned} \tag{7}$$

where $\boldsymbol{W} \in \mathbb{R}^{D\times 3D_h}$ is the transformation that converts the input tracklet to query, key and value; $\boldsymbol{A} \in \mathbb{R}^{T\times T}$ stores the attention weights; $\text{SA}(\boldsymbol{x})$ represents the self-attention given the input $\boldsymbol{x}$. For the first input layer, we set $D = K$. Then we define the multi-head self-attention (MSA) with the concatenation of k SAs as follows,

$$\text{MSA}(\boldsymbol{x}) = [\text{SA}_1(\boldsymbol{x}); \text{SA}_2(\boldsymbol{x}); ...; \text{SA}_k(\boldsymbol{x})]\boldsymbol{W}^O, \tag{8}$$

where $\boldsymbol{W}^O \in \mathbb{R}^{kD_h\times D}$ aggregates the information from multiple heads and transforms it back to the original dimension D and D_h is the number of channels for each head. After the multi-head self-attention layer, a feed-forward layer is further applied for encoding.

After the final layer of the encoder, we also stack a classification head for learning the embedding. The classification head includes a global average pooling (GAP) layer, l_2 normalization, and one fully connected layer, *i.e.*,

$$\begin{aligned} \boldsymbol{h} &= \text{norm}_{l_2}\left(\text{GAP}(\boldsymbol{z}_L)\right), \\ \boldsymbol{p} &= \boldsymbol{W}_c \boldsymbol{h}, \end{aligned} \tag{9}$$

where $\boldsymbol{z}_L$ is the output of the last encoding layer, $\boldsymbol{h}$ is the final tracklet embedding, $\boldsymbol{W}_c \in \mathbb{R}^{D\times C}$ is the weight of the fully connected layer, $\boldsymbol{p}$ is the predicted logits for C object IDs. The purpose of GAP is to pool the features along the temporal dimension from $D \times T$ to D. Cross-entropy loss $\mathcal{L}_{xent}$ and triplet loss $\mathcal{L}_{triplet}$ are used in the training as follows,

$$\begin{aligned} \mathcal{L}_{xent} &= \sum_{i=1}^{N} -\log\left(\frac{\exp\left(\boldsymbol{p}_i^c\right)}{\sum_{j=1}^{C}\exp\left(\boldsymbol{p}_i^j\right)}\right), \\ \mathcal{L}_{triplet} &= \sum_{i=1}^{N}[||\boldsymbol{h}_i^a - \boldsymbol{h}_i^p||_2 - ||\boldsymbol{h}_i^a - \boldsymbol{h}_i^n||_2 + \alpha]_+, \\ \mathcal{L}_{total} &= \mathcal{L}_{xent} + \lambda\mathcal{L}_{triplet}, \end{aligned} \tag{10}$$

where $\boldsymbol{h}_i^a$, $\boldsymbol{h}_i^p$ and $\boldsymbol{h}_i^n$ are embeddings of an anchor tracklet, a positive tracklet, and a negative tracklet in a mini-batch, α is a pre-defined distance margin, and λ is the trade-off between two losses.

For the architecture of Connector, we stack 6 encoding layers. 4-head self-attention is employed in each encoding layer. We set $D = 512$ as the intermediate channel dimension.

D. Tracking Inference

In the inference stage, each of the initial tracklets is first sent to the Splitter model. Based on the predicted ID-change position mask $\hat{\boldsymbol{m}}$, local maximum peaks are picked and the peak values are compared with a pre-defined splitting threshold

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3140919, IEEE Transactions on Multimedia

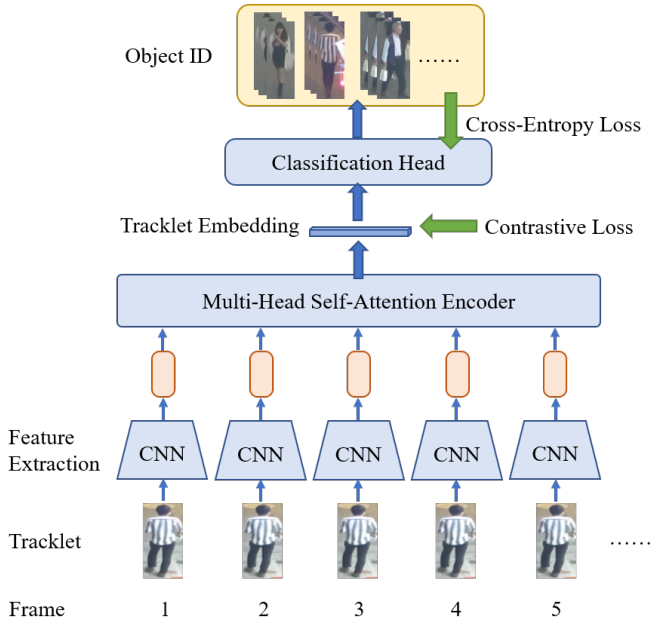


Fig. 6. Connector model. Input tracklets are first sent to a pre-trained CNN for feature extraction. A multi-head self-attention based encoder is employed with the combination of cross-entropy loss and contrastive loss for tracklet embedding.

δ_s. The positions with peak values greater than δ_s are treated as splitting positions. Then tracklets are split at the predicted splitting positions.

Specifically, denote the initial tracklet set as $\mathcal{S}_\mathcal{T} = \{\mathcal{T}_k | k = 1, 2, ..., M\}$. Since some tracklets have long time durations, we adopt a temporal window approach that conducts split prediction based on each sliding window. To avoid boundary effects, the adjacent sliding windows overlap with each other. For overlapping positions that have multiple estimations, we take the average as the final splitting prediction as follows.

$$\hat{\boldsymbol{p}}_s^k = \frac{\sum_t f_s(\mathcal{T}_k[t-T:t])}{\sum_t \mathbf{1}[t-T:t]}, \tag{11}$$

where T is the temporal window size, $\mathcal{T}_k[t-T:t]$ is the tracklet cut based on the temporal window, $\mathbf{1}[t-T:t]$ is the all-one vector within the sliding window, and $\hat{\boldsymbol{p}}_s^k$ is the averaged final prediction. Then we take the local maximums and split tracklets according to δ_s. After splitting all tracklets, we update the tracklet set $\mathcal{S}_\mathcal{T}$. The details of the inference stage of Splitter are summarized in Algorithm 1.

To group tracklets into full tracks with the trained Connector, a tracklet graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is built based on the split tracklets, where $\mathcal{V}$ and $\mathcal{E}$ are the vertex set and edge set, respectively. Specifically, $\mathcal{V} = \mathcal{S}_\mathcal{T}$, while $\mathcal{E}$ is a finite set in which every element $e \in \mathcal{E}$ represents an edge between a pair of two tracklets $\mathcal{T}_u, \mathcal{T}_w \in \mathcal{V}$ that 1) are not far away in the time domain, yet 2) have no temporal overlap with each other. The two conditions are as follows,

$$\begin{aligned} &1)\ \min_{t_u \in T_f(u), t_w \in T_f(w)} |t_u - t_w| \leq \delta_t, \\ &2)\ T_f(u) \cap T_f(w) = \emptyset, \end{aligned} \tag{12}$$

where $T_f(u)$ is the set of frame indices of the tracklet $\mathcal{T}_u$. Based on the built tracklet graph, each tracklet is embedded via the encoder of the proposed Connector. A pairwise Euclidean distance on each element $e \in \mathcal{E}$ is measured. With a sufficiently good embedding framework, a simple bottom-up hierarchical clustering algorithm is employed to group tracklets if two tracklets satisfy

$$||\boldsymbol{h}_i - \boldsymbol{h}_j||_2 < \delta_c, \tag{13}$$

where δ_c is a pre-defined connection threshold. The setting of the hyper-parameters is described in Section IV-C. The details of the inference stage of Connector are summarized in Algorithm 2.

The time complexity for Splitter and Connector is analyzed as follows. Denote the number of tracklets as M, and the temporal window size as T. Since Splitter is conducted in a temporal window manner, the time complexity in inference is $O(MT)$. For Connector, the tracklet embedding takes about $O(MT)$, while the graph clustering with the greedy method takes about $O(M^2T)$. As a result, the complexity of the entire model is rough $O(M^2T)$.

Algorithm 1: Inference for Splitter

Input: Tracklet set $\mathcal{S}_\mathcal{T}$, Splitter model f_s.
Initialization: Set the temporal window T and Splitter threshold δ_s.
for $k = 1, 2, ..., M$ **do**
  Get the k-th tracklet $\mathcal{T}_k$.
  Set the count mask of $\mathcal{T}_k$ as $\boldsymbol{M}_k \leftarrow \mathbf{0}$.
  Set Splitter prediction as $\boldsymbol{m}_k \leftarrow \mathbf{0}$.
  for $t = 1, 2, ..., t_{max}$ **do**
    Set the starting frame of the temporal window as $\tau \leftarrow \max(t - T, 0)$.
    Update the count mask as $\boldsymbol{M}_k[\tau : t] \leftarrow \boldsymbol{M}_k[\tau : t] + 1$.
    Update Splitter prediction as $\boldsymbol{m}_k[\tau : t] \leftarrow \boldsymbol{m}_k[\tau : t] + f_s(\mathcal{T}_i[\tau : t])$.
  Get the final Splitter prediction as $\boldsymbol{m}_k \leftarrow \boldsymbol{m}_k / \boldsymbol{M}_k$.
  Get the set of local maximums of $\boldsymbol{m}_k$ as $\{p_k\}$.
  Remove p_k from $\{p_k\}$ if $p_k < \delta_s$.
  Split $\mathcal{T}_k$ into a new tracklet set $\mathcal{S}_{\mathcal{T}_k}$ given splitting positions $\{p_k\}$.
  Update the tracklet set $\mathcal{S}_T \leftarrow \mathcal{S}_T / \{\mathcal{T}_k\} \cup \mathcal{S}_{\mathcal{T}_k}$.
Output: Updated tracklet set $\mathcal{S}_T$.

IV. Experiments

A. Datasets

To evaluate the proposed TBooster, we conduct experiments on two widely used pedestrian tracking benchmark datasets, *i.e.*, MOT17 [34] and MOT20 [78]. The details of the datasets are described as follows.

Algorithm 2: Inference for Connector

Input: Tracklet set $\mathcal{S}_{\mathcal{T}}$, Connector model f_c.
Initialization: Set the temporal window T and Connector threshold δ_c.
Build the tracklet graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ based on the conditions in Eq. (12).
for $k = 1, 2, ..., M$ **do**
    Get tracklet embedding $\boldsymbol{h}_k = f_c(\mathcal{T}_k)$ based on the encoder of Connector.
for $e \in \mathcal{E}$ **do**
    Calculate embedding distance $d_e = ||\boldsymbol{h}_u - \boldsymbol{h}_w||_2$, where u and w are the nodes associated with the edge e.
Sort $\{d_e | e \in \mathcal{E}\}$ in ascending order.
while *True* **do**
    Select d_{min} with minimum distance in $\{d_e | e \in \mathcal{E}\}$.
    if $d_{min} > \delta_c$ **then**
        Break;
    Select the edge e, tracklets $\mathcal{T}_u$ and $\mathcal{T}_w$ associated with d_{min}.
    Merge $\mathcal{T}_u$ and $\mathcal{T}_w$ to $\mathcal{T}_{new}$.
    Remove edges that connect with $\mathcal{T}_u$ or $\mathcal{T}_w$ and do not satisfy the conditions in Eq. (12). Update edge set $\mathcal{E}$ after removal.
    Update the tracklet set $\mathcal{S}_T \leftarrow \mathcal{S}_T \cup \{\mathcal{T}_{new}\}/\{\mathcal{T}_u, \mathcal{T}_w\}$.
Assign a unique tracking ID to each of the tracklets in $\mathcal{S}_{\mathcal{T}}$.
Output: Tracking IDs for tracklets in $\mathcal{S}_T$.

*1) **MOT17 dataset**:* MOT17 is a widely used pedestrian tracking benchmark dataset. In total, there are 7 training video sequences and 7 testing video sequences. The benchmark also provides public deformable part models (DPM) [79], Faster-RCNN [80] and scale-dependent pooling (SDP) [81] detections for both training and testing data. The number of tracks is 1,331 and the number of total frames is 11,235.

*2) **MOT20 dataset**:* MOT20 is a recently released pedestrian tracking dataset for crowded scenarios with plenty of occlusions. The average density is over 150. In total, there are 4 training video sequences and 4 testing video sequences. The dataset also provides public Faster R-CNN detections with ResNet101 [82] as the backbone. The number of trajectories in the training data is 3,833 and the total number of frames is 13,410.

B. Training Details

The training details for the proposed TBooster are described as follows.

*1) **Splitter**:* We apply the simple and fast IOU tracker [74] to the randomly jittered and horizontally flipped ground truth bounding boxes with variant overlapping thresholds to generate tracklets on the fly. There are two reasons that we use the IOU tracker to generate training tracklets. First, the IOU tracker is the one with the highest computation speed without any complex models, so that it can be efficiently used to generate augmented tracklets on the fly with different IOU thresholds as the training data. On the contrary, it is very expensive to adopt other model-based trackers to generate augmented training tracklets. Second, compared with other trackers, the IOU tracker can generate more association errors with the extreme simple association strategy. We can augment different types of association errors when varying the IOU threshold. Based on the augmented tracklets, each tracklet may contain multiple ground truth object IDs since there are association errors. We store the ID-change positions in the mask $\boldsymbol{m}^*$, which is treated as the ground truth splitting positions. The temporal window T is set to be 65 frames. For tracklets that last shorter than 65 frames, padding with the starting and ending frames is used. A baseline Re-ID method [83] is adopted for extracting the input appearance feature with dimension 2048. Normalized bounding box parameters (x, y, w, h) are used as motion features. Thus, the input tracklet has a size of 2052×65. We use Adam optimizer with an initial learning rate of 0.001. The cosine annealing learning rate scheduler is adopted with a maximum iteration of 60,000.

*2) **Connector**:* For training Connector, we follow the similar data preparation as Splitter. Slightly different from the tracklets generated for training Splitter, we ensure that each tracklet only contains one object ID. Adam optimizer with initial learning rate 0.0001 is adopted. We use cosine annealing learning rate scheduler with maximum iteration 120,000. For hyper-parameters in the loss function, we set $\lambda = 0.5$ and $\alpha = 0.2$ in Eq. (10).

C. Inference Details

In the inference stage, we use tracklets generated from existing trackers as input. We process the tracklets with the temporal sliding window procedure with 50% overlapping frames. Splitter is firstly conducted to predict the potential splitting positions. As defined in Section III-D, we set the splitting threshold $\delta_s = 0.5$ for all experiments. Connector is also conducted in a sliding window manner. We set $\delta_t = 64$ and $\delta_c = 0.9$ for all following experiments. Currently, the proposed method is conducted offline. Since it is processed in a temporal sliding window manner, it can be modified to an online method, which will be implemented in future work.

D. Evaluation Metrics

We employ ID F1 measure (IDF1) [35], multiple object tracking accuracy (MOTA), mostly tracked targets (MT), mostly lost targets (ML), fragments (FM) and identity switches (IDS) [34] as the evaluation metrics of the tracking performance, which are widely used in MOT.

E. Main Results

We evaluate the performance of the proposed tracklet booster on MOT17 and MOT20 with SOTA methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3140919, IEEE Transactions on Multimedia

TABLE II
COMPARISON WITH SOTA METHODS ON MOT17 AND MOT20 TESTING SET

Method	IDF1 (%) ↑	MOTA (%) ↑	MT (%) ↑	ML (%) ↓	IDS ↓	FRAG ↓	FPS ↑
MOT17							
IOU [74]	39.4	45.5	369	953	5,988	7,404	1,522.9
TBooster+IOU	**45.1** (+5.7)	45.8	369	953	4,189	7,430	7.7
Tracktor_v2 [29]	55.1	56.3	498	831	1,987	3,763	1.5
TBooster+Tracktor_v2	**59.2** (+4.1)	56.4	498	831	1,785	3,750	1.3
MPNTrack [20]	61.7	58.8	679	788	1,185	2,265	6.5
TBooster+MPNTrack	**62.3** (+0.6)	58.9	682	790	1,198	2,209	4.2
CenterTrack [26]	59.6	61.5	621	752	2,583	4,965	17.0
TBooster+CenterTrack	**63.3** (+3.7)	61.5	622	754	2,470	5,079	6.9
FairMOT [64]	72.3	73.7	1,017	408	3,303	8,073	25.9
TBooster+FairMOT	**74.0** (+1.7)	74.1	1,122	363	2,454	4,116	5.7
MOT20							
Tracktor [29]	52.7	52.6	365	331	1,648	4,374	1.2
TBooster+Tracktor	**53.3** (+0.6)	52.6	365	329	1,734	4,389	0.6
UnsupTrack [75]	50.6	53.6	376	311	2,178	4,335	1.3
TBooster+UnsupTrack	**54.3** (+3.7)	53.7	374	313	1,771	4,322	0.6
MOT20_TBC [76]	50.1	54.5	415	245	2,449	2,580	5.6
TBooster+MOT20_TBC	**54.3** (+4.2)	54.6	416	247	1,771	2,679	0.8
GNNMatch [77]	49.0	54.5	407	317	2,038	2,456	0.1
TBooster+GNNMatch	**53.4** (+4.4)	54.6	407	317	1.674	2,455	0.1

*For each pair of comparisons, the first row shows the original SOTA method and the second row shows the corresponding method combined with the proposed TBooster. The number in "()" besides the IDF1 score is the improvement after TBooster applied.

*1) **MOT17**:* We test five baseline and SOTA methods, *i.e.*, IOU [74], Tracktor [29], MPNTrack [20], CenterTrack [26], and FairMOT [64] on MOT17, combined with the proposed TBooster method. Specifically, the IOU tracker uses intersection over the union between bounding boxes across frames as the main clue for data association; Tracktor exploits the bounding box regression of an object detector to predict the position of an object in the next frame, given sequential input frames; MPNTrack exploits the classical network flow formulation of MOT to define a fully differentiable framework based on message passing networks; CenterTrack applies detection to a pair of frames and associates the objects to the previous frame; while FairMOT learns a unified network for both detection and Re-ID via a multi-task learning framework. The results are reported in the top part of Table II. For each pair of comparisons, the original method is shown in the first row, while the method with the proposed TBooster is shown in the second row. As expected, there are significant improvements on the IDF1 score, ranging from +0.6% to +5.7%. Since we do not tackle detection algorithms, the MOTA metrics roughly remain the same with TBooster added.

*2) **MOT20**:* We test four SOTA methods, *i.e.*, Tracktor [29], UnsupTrack [75], MOT20_TBC [76], and GCNNMatch [77] on MOT20, combined with the proposed TBooster method. Specifically, Tracktor is the same tracker as used in MOT17; UnsupTrack trains a Re-ID network to predict the generated labels using cross-entropy loss; MOT20_TBC jointly models detection, counting, and tracking of multiple targets as a network flow program, which simultaneously finds the global optimal detections and trajectories of multiple targets over the whole video; GCNNMatch uses graph convolutional neural network-based feature extraction and end-to-end feature matching for object association. The results are reported in the bottom part of Table II. Similarly, there are also large improvements on the IDF1 compared with the original methods. This demonstrates the effectiveness of the proposed method on the association task in MOT.

F. Qualitative Results

To better visualize the improvement with TBooster, we show some qualitative comparison results between CenterTrack and TBooster in Fig. 7. The frames are sampled from sequences MOT17-01, MOT17-03, and MOT17-06, respectively. Each color of the bounding box represents a distinct predicted object ID. For each sequence, the first row shows the result from CenterTrack, and the second row is from TBooster. Frame index is shown on the bottom-right of each image. The objects pointed with red arrows are used for comparison between CenterTrack and TBooster. In sequence MOT17-01, the person with the pointed arrow changes ID after it reappears from occlusion at frame 160. The same situation happens for CenterTrack in MOT17-03. However, with the Connector module in TBooster, the association error is fixed, as shown in the second row of the first two examples. For the example in MOT17-06, CenterTrack assigns the same ID to three different pedestrians pointed with the red arrows. With the effectiveness of Splitter in TBooster, the association error is fixed for this situation. These three examples verify how the proposed Splitter and Connector correct the two types of association errors mentioned in the Introduction section.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3140919, IEEE Transactions on Multimedia

Fig. 7. Qualitative comparison results between CenterTrack and TBooster. The frames are sampled from sequences MOT17-01, MOT17-03, and MOT17-06, respectively. Each color of the bounding box represents a distinct predicted object ID. For each sequence, the first row shows the result from CenterTrack, and the second row is from TBooster. Frame index is shown on the bottom-right of each image. The objects pointed with red arrows are used for comparison between CenterTrack and TBooster. These three examples verify how the proposed Splitter and Connector correct the two types of association errors mentioned in the Introduction section.

TABLE III
EFFECTIVENESS OF SPLITTER AND CONNECTOR

Method	IDF1 (%)	MOTA (%)
Original	48.5	55.1
w/. Connector	52.2 (+3.7)	55.4
w/. Splitter	47.1 (-1.4)	54.4
w/. Splitter & Connector	**54.6** (+6.1)	55.2

TABLE IV
EFFECT OF DIFFERENT COMBINATIONS OF THRESHOLDS FOR SPLITTER AND CONNECTOR

Method	IDF1 (%)	MOTA (%)
Original baseline	48.5	55.1
$\delta_s = 0.5, \delta_c = 0.5$	52.9 (+4.4)	54.9
$\delta_s = 0.5, \delta_c = 0.7$	53.8 (+5.3)	55.1
$\delta_s = 0.5, \delta_c = 0.9$	**54.6** (+6.1)	55.2
$\delta_s = 0.5, \delta_c = 1.1$	53.2 (+4.7)	55.2
$\delta_s = 0.7, \delta_c = 0.5$	51.6 (+3.1)	55.2
$\delta_s = 0.7, \delta_c = 0.7$	52.3 (+3.8)	55.3
$\delta_s = 0.7, \delta_c = 0.9$	52.2 (+3.7)	55.3
$\delta_s = 0.7, \delta_c = 1.1$	52.1 (+3.6)	55.3
$\delta_s = 0.9, \delta_c = 0.5$	51.8 (+3.3)	55.3
$\delta_s = 0.9, \delta_c = 0.7$	52.4 (+3.9)	55.4
$\delta_s = 0.9, \delta_c = 0.9$	52.2 (+3.7)	55.4
$\delta_s = 0.9, \delta_c = 1.1$	52.1 (+3.6)	55.4

G. Ablation Study

For the ablation study, we use the MOT17-09 sequence as the validation set and the rest sequences as the training set. We adopt the IOU tracker as the baseline original tracking method.

*1) **Study of the Effectiveness of the Splitter and Connector**:* To test the effectiveness of Splitter and Connector modules, we compare three different settings, *i.e.*, combining Connector without Splitter, combining Splitter without Connector, and combining both Splitter and Connector. The comparison results are shown in Table III. As shown in the second row of the table, with a standalone Connector module, the IDF1 is boosted by 3.7%. However, as shown in the third row, the IDF1 is decreased by 1.4% with the standalone Splitter module. This means some tracks are divided into pieces, which negatively affects the IDF1 score. This is a common phenomenon, especially when occlusion happens. There is a high chance for tracklets to get split since the appearance feature changes rapidly. As shown in the last row of the table, with the combination of both Splitter and Connector, IDF1 is boosted by 6.1%, which is a further boost of 2.4% compared with the standalone Connector module in the second row. This is because, after the Splitter module, tracklets can get much purer IDs, which further helps Connector to increase the grouping accuracy. This demonstrates the effectiveness of both Splitter and Connector modules.

*2) **Effect of Different Inference Thresholds**:* We also conduct experiments related to thresholds δ_s and δ_c for Splitter and Connector in the inference stage, respectively. We vary δ_s from 0.5 to 0.9 and δ_c from 0.5 to 1.1. We report the results in Table IV. From the results, there is an over 3.0% consistent improvement on the IDF1 score against the original tracking performance. This demonstrates the robustness of both Splitter and Connector modules.

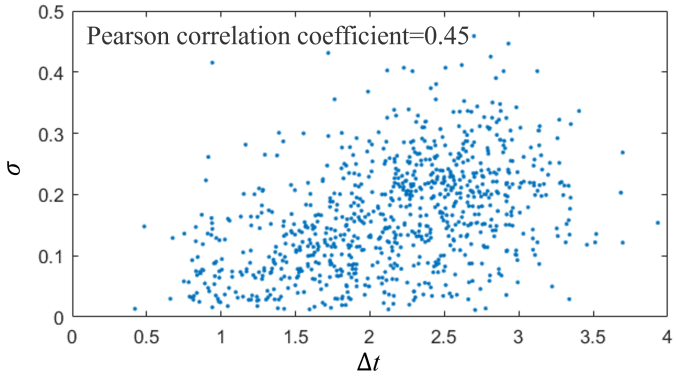


Fig. 8. Pearson correlation coefficient between the estimated time duration of ID-change and the predicted standard deviation from Splitter.

*3) **Effect of Adaptive Smoothing Strategy**:* To validate the effectiveness of the adaptive smoothing strategy in Splitter, we use the baseline method without adaptive Gaussian smoothing in Splitter, where the ground truth splitting position $\boldsymbol{m}^*$ is directly employed to guide the prediction $\hat{\boldsymbol{m}}$ using mean squared error, *i.e.*, the loss is set as $\mathcal{L}_s = ||\hat{\boldsymbol{m}} - \boldsymbol{m}^*||^2$. In the evaluation, for both the baseline method and the proposed Splitter with adaptive Gaussian smoothing, we select the local peaks from the predicted mask $\hat{\boldsymbol{m}}$ as the detected splitting positions and use corresponding peak values as the predicted confidences. Average precision (AP) is adopted as the evaluation metric for measuring the splitting performance. We show the results in Table V. Compared with the baseline method, there is a 9.4% improvement on AP for the proposed approach, which shows the effectiveness of the adaptive Gaussian smoothing strategy.

To verify our motivation that the standard deviation of the adaptive Gaussian kernel for the label smoothing should have a correlation with the duration of ID-change, we further measure their relationship based on the Pearson correlation coefficient (PCC), between the expected ID-change duration Δt and the predicted standard deviation σ as follow,

$$\mathrm{PCC}(\Delta t, \sigma) = \frac{\sum_i (\Delta t_i - \bar{\Delta t})(\sigma_i - \bar{\sigma})}{\sqrt{\sum_i (\Delta t_i - \bar{\Delta t})^2 \sum_i (\sigma_i - \bar{\sigma})^2}}. \quad (14)$$

Since we do not have ground truth ID-change duration, we use the expected ID-change duration instead. The expected duration Δt at each ID-change position τ_0 is defined as follows,

$$\Delta t(\tau_0) = \frac{\sum_{\tau \in \mathcal{N}(\tau_0)} \boldsymbol{m}_\tau |\tau - \tau_0|}{\sum_{\tau \in \mathcal{N}(\tau_0)} \boldsymbol{m}_\tau}, \quad (15)$$

where $\boldsymbol{m}_\tau$ is the prediction of the ID-change probability at position τ, $\mathcal{N}(\tau_0)$ is the set of timestamps in the neighborhood of τ_0 (we set $\mathcal{N}(\tau_0) = \{\tau_0 - 5, \tau_0 - 4, \ldots, \tau_0 + 5\}$). We randomly selected 800 pairs of $(\Delta t, \sigma)$, as shown in Fig. 8. Based on the above two definitions, the calculated PCC is

TABLE V
COMPARISON BETWEEN ADAPTIVE SMOOTHING AND THE BASELINE METHOD IN SPLITTER

	Baseline	Adaptive Smoothing
Average Precision (%)	49.0	**58.4**

TABLE VI
EFFECT OF VARYING NUMBER OF HEADS IN THE SELF-ATTENTION

#Heads	IDF1 (%)	MOTA (%)
1	53.5	55.1
2	53.9	55.2
4	**54.6**	55.2
8	54.4	55.2

0.45, indicating there is a positive correlation between the ID-change time duration and the standard deviation of the adaptive Gaussian kernels. This verifies our intuition that a longer duration ID-change duration should be softened more in the loss function design.

*4) **Effect of Multi-Head Self-Attention**:* To illustrate the influence of the multi-head self-attention mechanism, we conduct experiments with a variant number of attention heads in Connector. Specifically, we train Connector with four settings, *i.e.*, with 1, 2, 4, and 8 heads, respectively. We assign $512/k$ channels to each head to ensure the total number of channels is fixed for all settings. We adopt the same training scheduler and keep other parts unchanged. The results are shown in Table VI. The best result is achieved when 4-head attention is employed and no further improvement with more attention heads used. Typically, more heads have the capability to deal with more complex situations. Meanwhile, the dimension of each head is reduced when we add more heads if the channel number is fixed. A trade-off should be made for both of the effects. Thus, the result also conforms to our expectations.

H. Limitations

Although we achieve comparable SOTA performance on MOT17 and MOT20, there are some limitations of the proposed method. First, it is not an online method. It can be used for refining the association errors and is suitable for offline analysis, like surveillance monitoring, object counting, and scene analysis. However, it may be not suitable for autonomous driving scenarios that require real-time feedback. Besides that, the improvement becomes limited if the original trackers perform very well. This is reasonable since fewer association errors can be rectified from original trackers.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a simple yet effective tracklet boosting (TBooster) approach, that can be easily combined with the existing trackers and boost the association performance. The tracklet booster has two main modules, *i.e.*, Splitter and Connector. Splitter estimates the ID-change positions and splits tracklets into small parts, while Connector merges multiple tracklets into clusters if they are from the same object. To stabilize the training of Splitter, a novel adaptive Gaussian smoothing strategy is proposed. To learn the discriminative tracklet embeddings, a multi-head self-attention mechanism is employed in Connector. We validate TBooster on two widely used benchmark datasets, *i.e.*, MOT17 and MOT20, and achieve significant improvement against SOTA methods. Moreover, we also conduct sufficient experiments on several aspects of tracklet booster in the ablation study, which further proves the effectiveness of each module in the proposed method.

We will work on two directions in our future work. Firstly, we are working on a real-time implementation that adopts the temporal sliding window strategy simultaneously along with the processing of original trackers. Secondly, we will work on combining the graph neural networks with a multi-head self-attention encoder for tracklet embeddings that takes the relationships among different tracklets into consideration. We believe that the future directions will also benefit the MOT field.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (62106219), the Fundamental Research Funds for the Central Universities (2021QN81017), Guangdong Provincial Special Funding projects for Introducing Innovation Team and Industry University Research Cooperation (2019C002001), Guangdong Maoming Science and Technology Special Industry-University Research Integration Project (2020576).

REFERENCES

[1] Z. Tang, G. Wang, H. Xiao, A. Zheng, and J.-N. Hwang, "Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features," in *CVPR Workshop (CVPRW) on the AI City Challenge*, 2018.

[2] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. Anastasiu, and J.-N. Hwang, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8797–8806.

[3] G. Wang, X. Yuan, A. Zhang, H.-M. Hsu, and J.-N. Hwang, "Anomaly candidate identification and starting time estimation of vehicles from traffic videos," in *AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California*, 2019.

[4] H.-M. Hsu, T.-W. Huang, G. Wang, J. Cai, Z. Lei, and J.-N. Hwang, "Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models," in *AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California*, 2019.

[5] Y. Wu, D. Kong, S. Wang, J. Li, and B. Yin, "An unsupervised real-time framework of human pose tracking from range image sequences," *IEEE Transactions on Multimedia*, vol. 22, no. 8, pp. 2177–2190, 2020.

[6] R. Gu, G. Wang, and J.-N. Hwang, "Efficient multi-person hierarchical 3d pose estimation for autonomous driving," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2019, pp. 163–168.

[7] R. Gu, G. Wang, Z. Jiang, and J.-N. Hwang, "Multi-person hierarchical 3d pose estimation in natural videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4245–4257, 2019.

[8] R. Gu, G. Wang, and J.-N. Hwang, "Exploring severe occlusion: Multi-person 3d pose estimation with gated convolution," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 8243–8250.

[9] B. Jagadeesh and C. M. Patil, "Human motion tracking for human behavior analysis using gaussian mixture model and kalman filtering," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 2637–2644, 2018.

[10] A. Jalal, M. Mahmood, and A. S. Hasan, "Multi-features descriptors for human activity tracking and recognition in indoor-outdoor environments," in *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. IEEE, 2019, pp. 371–376.

[11] M. Chaabane, P. Zhang, J. R. Beveridge, and S. O'Hara, "Deft: Detection embeddings for tracking," *arXiv preprint arXiv:2102.02267*, 2021.

[12] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krahenbuhl, T. Darrell, and F. Yu, "Joint monocular 3d vehicle detection and tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5390–5399.

[13] G. Wang, J.-N. Hwang, K. Williams, and G. Cutter, "Closed-loop tracking-by-detection for rov-based multiple fish tracking," in *Computer Vision for Analysis of Underwater Imagery (CVAUI), 2016 ICPR 2nd Workshop on*. IEEE, 2016, pp. 7–12.

[14] M.-C. Chuang, J.-N. Hwang, J.-H. Ye, S.-C. Huang, and K. Williams, "Underwater fish tracking for moving cameras based on deformable multiple kernels," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 9, pp. 2467–2477, 2016.

[15] M. Dawkins, L. Sherrill, K. Fieldhouse, A. Hoogs, B. Richards, D. Zhang, L. Prasad, K. Williams, N. Lauffenburger, and G. Wang, "An open-source platform for underwater image and video analytics," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 898–906.

[16] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person reidentification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3539–3548.

[17] A. Milan, K. Schindler, and S. Roth, "Multi-target tracking by discrete-continuous energy minimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2054–2068, 2016.

[18] R. Kumar, G. Charpiat, and M. Thonnat, "Multiple object tracking by efficient graph partitioning," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 445–460.

[19] Y. Wang, K. Kitani, and X. Weng, "Joint Object Detection and Multi-Object Tracking with Graph Neural Networks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[20] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6247–6257.

[21] C. Shan, C. Wei, B. Deng, J. Huang, X.-S. Hua, X. Cheng, and K. Liang, "Fgagt: Flow-guided adaptive graph tracking," *arXiv preprint arXiv:2010.09015*, 2020.

[22] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6499–6508.

[23] Q. Zhou, B. Zhong, Y. Zhang, J. Li, and Y. Fu, "Deep alignment network based multi-person tracking with occlusion and motion reasoning," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1183–1194, 2019.

[24] Q. Bao, W. Liu, Y. Cheng, B. Zhou, and T. Mei, "Pose-guided tracking-by-detection: Robust multi-person pose tracking," *IEEE Transactions on Multimedia*, vol. 23, pp. 161–175, 2021.

[25] J. Shen, D. Yu, L. Deng, and X. Dong, "Fast online tracking with detection refinement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 162–173, 2017.

[26] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," 2020.

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

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

[29] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[30] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," in *European Conference on Computer Vision 2020*. Springer International Publishing, 2020, pp. 107–122.

[31] D. Y. Kim and M. Jeon, "Data fusion of radar and image measurements for multi-object tracking via kalman filtering," *Information Sciences*, vol. 278, pp. 641–652, 2014.

[32] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks." in *AAAI*, vol. 2, 2017, p. 4.

[33] Y. Lu, C. Lu, and C.-K. Tang, "Online video object detection using association lstm," in *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy*, 2017, pp. 22–29.

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

[35] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 17–35.

[36] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision*, pp. 1–31, 2020.

[37] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee, "Multi-class multi-object tracking using changing point detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 68–83.

[38] W. Feng, Z. Hu, B. Li, W. Gan, W. Wu, and W. Ouyang, "Samot: Switcher-aware multi-object tracking and still another mot measure," *arXiv preprint arXiv:2009.10338*, 2020.

[39] J. Valmadre, A. Bewley, J. Huang, C. Sun, C. Sminchisescu, and C. Schmid, "Local metrics for multi-object tracking," *arXiv preprint arXiv:2104.02631*, 2021.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[41] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," *arXiv preprint arXiv:1701.01909*, vol. 4, no. 5, p. 6, 2017.

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

[43] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "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*, 2020, pp. 6768–6777.

[44] Y. Liu, J. Yin, D. Yu, S. Zhao, and J. Shen, "Multiple people tracking with articulation detection and stitching strategy," *Neurocomputing*, vol. 386, pp. 18–29, 2020.

[45] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3029–3037.

[46] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Subgraph decomposition for multi-target tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5033–5041.

[47] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li, "Multiple target tracking based on undirected hierarchical relation hypergraph," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1282–1289.

[48] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with trackletnet," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 482–490.

[49] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *International journal of computer vision*, vol. 122, no. 3, pp. 484–501, 2017.

[50] P. Lenz, A. Geiger, and R. Urtasun, "Followme: Efficient online min-cost flow tracking with bounded memory and computation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4364–4372.

[51] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, 2013.

[52] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1926–1933.

[53] A. Hornakova, R. Henschel, B. Rosenhahn, and P. Swoboda, "Lifted disjoint paths with application in multiple object tracking," in *The 37th International Conference on Machine Learning (ICML)*, July 2020.

[54] G. Wang, R. Gu, Z. Liu, W. Hu, M. Song, and J.-N. Hwang, "Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking," in *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, 2021, pp. 9876–9886.

[55] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu, "Multi-view people tracking via hierarchical trajectory composition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4256–4265.

[56] Y. Xu, X. Liu, L. Qin, and S.-C. Zhu, "Cross-view people tracking by scene-centered spatio-temporal parsing." in *AAAI*, 2017, pp. 4299–4305.

[57] R. Han, W. Feng, J. Zhao, Z. Niu, Y. Zhang, L. Wan, and S. Wang, "Complementary-view multiple human tracking," in *AAAI Conference on Artificial Intelligence*, 2020.

[58] J. He, Z. Huang, N. Wang, and 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.

[59] P. Dai, R. Weng, W. Choi, C. Zhang, Z. He, and W. Ding, "Learning a proposal classifier for multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[60] M. Guo, E. Chou, D.-A. Huang, S. Song, S. Yeung, and L. Fei-Fei, "Neural graph matching networks for fewshot 3d action recognition," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 653–669.

[61] M. Narasimhan, S. Lazebnik, and A. G. Schwing, "Out of the box: Reasoning with graph convolution nets for factual visual question answering," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[62] J. Gao, T. Zhang, and C. Xu, "Graph convolutional tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4649–4659.

[63] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[64] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, pp. 1–19, 2021.

[65] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[66] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[67] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable {detr}: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2021.

[68] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3d human pose estimation with spatial and temporal transformers," *arXiv preprint arXiv:2103.10455*, 2021.

[69] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[70] P. Sun, Y. Jiang, R. Zhang, E. Xie, J. Cao, X. Hu, T. Kong, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple-object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[71] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," *arXiv preprint arXiv:2101.02702*, 2021.

[72] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "Spatial-temporal graph transformer for multiple object tracking," *arXiv preprint arXiv:2104.00194*, 2021.

[73] Z. Zhang, J. Wu, X. Zhang, and C. Zhang, "Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project," *arXiv preprint arXiv:1712.09531*, 2017.

[74] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017*, Lecce, Italy, Aug. 2017.

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

[76] W. Ren, X. Wang, J. Tian, Y. Tang, and A. B. Chan, "Tracking-by-counting: Using network flows on crowd density maps for tracking multiple targets," *IEEE Transactions on Image Processing*, vol. 30, pp. 1439–1452, 2020.

[77] I. Papakis, A. Sarkar, and A. Karpatne, "Gcnnmatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization," *arXiv preprint arXiv:2010.00067*, 2020.

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

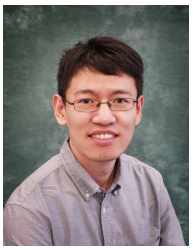
[79] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[80] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

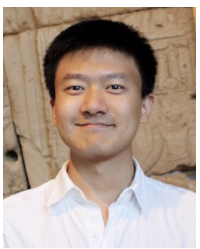
[81] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2129–2137.

[82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[83] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, "Bag of tricks and a strong baseline for deep person re-identification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.



Dr. Gaoang Wang joined the international campus of Zhejiang University as an Assistant Professor in September 2020. He is also an Adjunct Assistant Professor at UIUC. Gaoang Wang received a B.S. degree at Fudan University in 2013, a M.S. degree at the University of Wisconsin-Madison in 2015, and a Ph.D. degree from the Information Processing Laboratory of the Electrical and Computer Engineering department at the University of Washington in 2019. After that, he joined Megvii US office in July 2019 as a research scientist working on multi-frame fusion. He then joined Wyze Labs in November 2019 working on deep neural network design for edge-cloud collaboration. His research interests are computer vision, machine learning, artificial intelligence, including multi-object tracking, representation learning, and active learning. Gaoang Wang published papers in many renowned journals and conferences, including IEEE T-IP, IEEE T-CSVT, IEEE T-VT, ICCV, ACM MM, CVPR, etc.



Yizhou Wang is a Ph.D. candidate in Electrical and Computer Engineering at the University of Washington, advised by Prof. Jenq-Neng Hwang. He received his M.S. degree in Electrical Engineering in 2018 from Columbia University, advised by Prof. Shih-Fu Chang. He received the B.Eng. degree in Automation at Northwestern Polytechnical University in 2016. His research interests include autonomous driving, computer vision, deep learning, and cross-modal learning.

Dr. Renshu Gu joined Hangzhou Dianzi University in 2020. She received the bachelor's degree in electrical engineering from Nanjing University in 2011. She received the Ph.D. degree from the Department of Electrical and Computer Engineering at the University of Washington in 2020. Her passions include image/video analytics, computer vision, and multimedia.

Dr. Weijie Hu received a bachelor's degree from Tianjin University, a master's degree from the Graduate School of the Chinese Academy of Sciences, and a Ph.D. degree from the University of the Chinese Academy of Sciences. Weijie Hu has engaged in physical chemistry, new chemical materials, chemical park management (smart park construction and safety direction), science popularization system construction, and other basic scientific research and industry-university-research applied research work for nearly 20 years. Weijie Hu participated in or presided over more than 40 scientific and technological projects, published more than 20 domestic and foreign research papers, applied for more than 30 domestic patents (12 authorized), and the main drafter of the national standard "GB 2019 Smart Chemical Park Construction Guide".

Dr. Jenq-Neng Hwang received the BS and MS degrees, both in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1981 and 1983 separately. He then received his Ph.D. degree from the University of Southern California. In the summer of 1989, Dr. Hwang joined the Department of Electrical and Computer Engineering (ECE) of the University of Washington in Seattle, where he has been promoted to Full Professor since 1999. He is the Director of the Information Processing Lab. (IPL), which has won several AI City Challenges and BMTT Tracking awards in the past years. Dr. Hwang served as associate editors for IEEE T-SP, T-NN and T-CSVT, T-IP and Signal Processing Magazine (SPM). He was the General Co-Chair of 2021 IEEE World AI IoT Congress, as well as the program Co-Chairs of IEEE ICME 2016, ICASSP 1998 and ISCAS 2009. Dr. Hwang is a fellow of IEEE since 2001.