Chained-Tracker: Chaining Paired Attentive Regression Results for End-to-End Joint Multiple-Object Detection and Tracking

Jinlong Peng[1] *, Changan Wang[1] *, Fangbin Wan[2], Yang Wu[3] **, Yabiao Wang[1], Ying Tai[1], Chengjie Wang[1], Jilin Li[1], Feiyue Huang[1], and Yanwei Fu[2]

[1] Tencent Youtu Lab {jeromepeng, changanwang, caseywang, yingtai, jasoncjwang, jerolinli, garyhuang}@tencent.com
[2] Fudan University {fbwan18, yanweifu}@fudan.edu.cn
[3] Nara Institute of Science and Technology yangwu@rsc.naist.jp

Abstract. Existing Multiple-Object Tracking (MOT) methods either follow the tracking-by-detection paradigm to conduct object detection, feature extraction and data association separately, or have two of the three subtasks integrated to form a partially end-to-end solution. Going beyond these sub-optimal frameworks, we propose a simple online model named Chained-Tracker (CTracker), which naturally integrates all the three subtasks into an end-to-end solution (the first as far as we know). It chains paired bounding boxes regression results estimated from overlapping nodes, of which each node covers two adjacent frames. The paired regression is made attentive by object-attention (brought by a detection module) and identity-attention (ensured by an ID verification module). The two major novelties: chained structure and paired attentive regression, make CTracker simple, fast and effective, setting new MOTA records on MOT16 and MOT17 challenge datasets (67.6 and 66.6, respectively), without relying on any extra training data. The source code of CTracker can be found at: github.com/pjl1995/CTracker.

Keywords: Multiple-Object Tracking, Chained-Tracker, End-to-end solution, Joint detection and tracking

1 Introduction

Video-based scene understanding and human behavior analysis are important high-level tasks in computer vision with many valuable real applications. They rely on many other tasks, within which *Multiple-Object Tracking (MOT)* is a significant one. However, MOT remains challenging due to occlusions, object trajectory overlap, challenging background, *etc.*, especially for crowded scenes.

Despite the great efforts and encouraging progress in the past years, there are two major problems of existing MOT solutions. One is that most methods

* *Equal contribution.*
** *Corresponding author: Yang Wu (wuyang0321@gmail.com)*

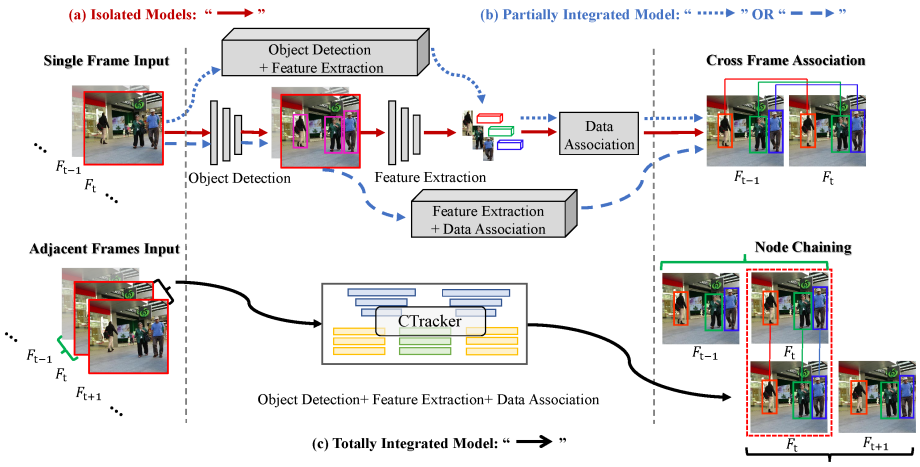

Fig. 1. Comparison of our CTracker (Bottom) with other typical MOT methods (Top), which are either isolated models or partially integrated models. Our CTracker significantly differs from other methods in two aspects: 1) It is a totally end-to-end model using adjacent frame pair as input and generating the box pair representing the same target. 2) We convert the challenging cross-frame association problem into pair-wise object detection problem.

are based on the tracking-by-detection paradigm [1], which is plausible but sub-optimal due to the infeasibility of global (end-to-end) optimization. It usually contains three sequential subtasks: object detection, feature extraction and data association. However, splitting the whole task into isolated subtasks may lead to local optima and more computation cost than end-to-end solutions. Moreover, data association heavily relies on the quality of object detection, which by itself is hard to generate reliable and stable results across frames as it discards the temporal relationships of adjacent frames.

The other problem is that recent MOT methods get more and more complex as they try to gain better performances. *Re-identification* and *attention* are two major points found to be helpful for improving the performance of MOT. Re-identification (or ID verification) is used to extract more robust features for data association. Attention helps the model to be more focused, avoiding the distraction by irrelevant yet confusing information (e.g. the complex background). Despite their effectiveness, the involvement of them in existing solutions greatly increases the model complexity and computational cost.

In order to solve the above problems, we propose a novel online tracking method named *Chained-Tracker* (CTracker), which unifies object detection, feature extraction and data association into a single end-to-end model. As can be seen in Fig. 1, our novel CTracker model is cleaner and simpler than the classical tracking-by-detection or partially end-to-end MOT methods. It takes adjacent frame pairs as input to perform joint detection and tracking in a single regres-

sion model that simultaneously regress the paired bounding boxes for the targets that appear in both of the two adjacent frames.

Furthermore, we introduce a joint attention module using predicted confidence maps to further improve the performance of our CTracker. It guides the paired boxes regression branch to focus on informative spatial regions with two other branches. One is the object classification branch, which predicts the confidence scores for the first box in the detected box pairs, and such scores are used to guide the regression branch to focus on the foreground regions. The other one is the ID verification branch whose prediction facilitates the regression branch to focus on regions corresponding to the same target. Finally, the bounding box pairs are filtered according to the classification confidence. Then, the generated box pairs belonging to the adjacent frame pairs could be associated using simple methods like IoU (Intersection over Union) matching [2] according to their boxes in the common frame. In this way, the tracking process could be achieved by chaining all the adjacent frame pairs (*i.e.* chain nodes) sequentially.

Benefiting from the end-to-end optimization of joint detection and tracking network, our model shows significant superiority over strong competitors while remaining simple. With the temporal information of the combined features from adjacent frames, the detector becomes more robust, which in turn makes data association easier, and finally results in better tracking performance.

The contribution of this paper can be summarized into the following aspects:

1. We propose an end-to-end online Multiple-Object Tracking model, to optimize object detection, feature extraction and data association simultaneously. Our proposed CTracker is the first method that converts the challenging data association problem to a pair-wise object detection problem.

2. We design a joint attention module to highlight informative regions for box pair regression and the performance of our CTracker is further improved.

3. Our online CTracker achieves state-of-the-art performance on the tracking result list with private detection of MOT16 and MOT17.

2 Related Work

2.1 Detection-based MOT Methods

Yu *et. al* [3] proposed the POI algorithm, which conducted a high-performance detector based on Faster R-CNN [4] by adding several extra pedestrian detection datasets. Chen *et. al* [5] incorporated an enhanced detection model by simultaneously modeling the detection-scene relation and detection-detection relation, called EDMT. Furthermore, Henschel *et. al* [6] added a head detection model to support MOT in addition to original pedestrian detection, which also needed extra training data and annotations. Bergmann *et. al* [7] proposed the Tracktor by exploiting the bounding box regression to predict the position of the pedestrian in the next frame, which was equal to modifying the detection box. However, the detection model and the tracking model in these detection-based methods are completely **independent**, which is complex and time-consuming. While our

CTracker algorithm only needs one **integrated** model to perform detection and tracking, which is simple and efficient.

2.2 Partially End-to-end MOT Methods

Lu *et. al* [8] proposed RetinaTrack, which combined detection and feature extraction in the network and used greedy bipartite matching for data association. Sun *et. al* [9] harnessed the power of deep learning for data association in tracking by jointly modeling object appearances and their affinities between different frames. Similarly, Chu *et. al* [10] designed the FAMNet to jointly optimize the feature extraction, affinity estimation and multi-dimensional assignment. Li *et. al* [11] proposed TrackNet by using frame tubes as input to do joint detection and tracking, however the links among tubes are not modeled which limits the trajectory lengths. Moreover, the model is designed and tested only for rigid object (vehicle) tracking, leaving its generalization ability questionable. Despite their differences, all these methods are just **partially** end-to-end MOT methods, because they just integrated some parts of the whole model, *i.e.* [8] combined the detection and feature extraction module in a network, [9, 10] combined the feature extraction and data association module. Differently, our CTracker is a **totally** end-to-end joint detection and tracking methods, unifying the object detection, feature extraction and data association in a single model.

2.3 Attention-assistant MOT Methods

Chu *et. al* [12] introduced a Spatial-Temporal Attention Mechanism (STAM) to handle the tracking drift caused by the occlusion and interaction among targets. Similarly, Zhu *et. al* [13] proposed a Dual Matching Attention Networks (DMAN) with both spatial and temporal attention mechanisms to perform the tracklet data association. Gao *et. al* [14] also utilized an attention-based appearance model to solve the inter-object occlusion. All these attention-assistant MOT methods used a complex attention model to optimize data association in the **local** bounding box level. While our CTracker can improve both the detection and tracking performance through the simple object-attention and identity-attention in the **global** image level, which is more efficient.

3 Methodology

3.1 Problem Settings

Given an image sequence $\{F_t\}_{t=1}^{N}$ with totally N frames, Multiple-Object Tracking task aims to output all the bounding boxes $\{\mathcal{G}_t\}_{t=1}^{N}$ and identity labels $\{\mathcal{Y}_t^{GT}\}_{t=1}^{N}$ for all the objects of interest in all the frames where they appear. $F_t \in \mathbb{R}^{c\times w\times h}$ indicates the t-th frame, $\mathcal{G}_t \subset \mathbb{R}^4$ represents the ground-truth bounding boxes of the K_t number of targets in t-th frame and $\mathcal{Y}_t^{GT} \subset \mathbb{Z}$ denotes their identities. Most of the recent MOT algorithms divide the MOT task

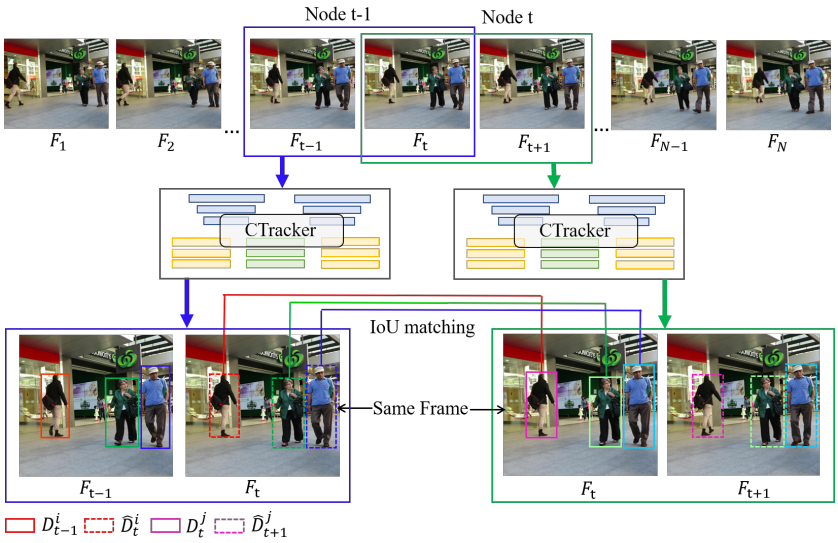

Fig. 2. Illustration of the node chaining. After generating bounding box pairs $\{\mathcal{D}_{t-1}, \hat{\mathcal{D}}_t\}$ by CTracker for two arbitrary adjacent nodes (F_{t-1}, F_t) and (F_t, F_{t+1}), we chain these two nodes by doing IoU matching on the shared common frame. Such a chaining is done sequentially over all adjacent nodes to generate long trajectories for the whole video sequence. More detailed can be found in the main text.

into three components, which are object detection, feature extraction and data association. However, many researches and experiments demonstrate that the association's effectiveness relies heavily on the performance of detection. Therefore, in order to better utilize their correlation, in this paper, we propose a novel Chained-Tracker (abbr. CTracker), which uses a single network to simultaneously achieve object detection, feature extraction and data association. We introduce the pipeline of our CTracker in the subsection 3.2. The details of the network and loss design are described separately in the subsection 3.3 and 3.4.

3.2 Chained-Tracker Pipeline

Framework. Different from other MOT models that only takes *a single frame* as input, our CTracker model requires *two adjacent frames* as input, which is called a chain node. The first chain node is (F_1, F_2) and the last (*i.e.,* the N-th) is (F_N, F_{N+1}). Note that F_N is the last frame, so we just take the copy version of F_N as F_{N+1}. Given the node (F_{t-1}, F_t) as input, CTracker can generate bounding box pairs $\{(D^i_{t-1}, \hat{D}^i_t)\}_{i=1}^{n_{t-1}}$ of the same targets appearing in both frames, where n_{t-1} is the total pair number, $D^i_{t-1} \in \mathcal{D}_{t-1} \subset \mathbb{R}^4$ and $\hat{D}^i_t \in \mathcal{D}_t \subset \mathbb{R}^4$ denote the two bounding boxes of the same target. Similarly, we can also get the box pairs $\{(D^j_t, \hat{D}^j_{t+1})\}_{j=1}^{n_t}$ in the next node (F_t, F_{t+1}). As can be seen in Fig. 2, assume that $\hat{D}^i_t$ and D^j_t represent detected boxes of the same target located in the

common frame of the adjacent nodes, there shall be only slight difference between the two boxes. We can further use an extremely simple matching strategy (as detailed below) to chain the two boxes, instead of using complicated appearance features as in canonical MOT methods. By chaining nodes sequentially over the given sequence, we can obtain long trajectories of all the detected targets.

Node chaining. We use $\{\mathcal{D}_{t-1}, \hat{\mathcal{D}}_t\}$ to represent $\{(D^i_{t-1}, \hat{D}^i_t)\}^{n_{t-1}}_{i=1}$ for convenience. The node chaining is done as follows. Firstly, in the node, every detected bounding box $D^i_1 \in \mathcal{D}_1$ is initialized as a tracklet with a randomly assigned identity. Secondly, for any another node t, we chain the adjacent nodes (F_{t-1}, F_t) and (F_t, F_{t+1}) by calculating the IoU (Intersection over Union) between the boxes in $\hat{\mathcal{D}}_t$ and $\mathcal{D}_t$ as shown in Fig. 2, where $\hat{\mathcal{D}}_t$ is the last boxes set of $\{\mathcal{D}_{t-1}, \hat{\mathcal{D}}_t\}$ and $\mathcal{D}_t$ is the former boxes set of $\{\mathcal{D}_t, \hat{\mathcal{D}}_{t+1}\}$. Getting the IoU affinity, the detected boxes in $\hat{\mathcal{D}}_t$ and $\mathcal{D}_t$ are matched by applying the Kuhn-Munkres (KM) algorithm [15]. For each matched box pair $\hat{D}^i_t$ and D^j_t, the tracklet that $\hat{D}^i_t$ belongs to is updated by appending D^j_t. Any unmatched box D^k_t is initialized as a new tracklet with a new identity. The chaining is done sequentially over all adjacent nodes and it builds long trajectories for individual targets.

Robustness enhancement (esp. against occlusions). To enhance the model's robustness to serious occlusions (which can make detection fail in certain frames) and short-term disappearing (followed by quick reappearing), we retain the terminated tracklets and their identities for up to σ frames and continue finding matches for them in these frames, with the simple constant velocity prediction model [16, 17] for motion estimation. In greater details, suppose target $(D^l_{t-1}, \hat{D}^l_t)$ cannot find its match is node t, we apply the constant velocity model to predict its bounding box $P^l_{t+\tau}$ in frame $t+\tau$ $(1 <= \tau <= \sigma)$ according to D^l_{t-1} (not the less reliable $\hat{D}^l_t$). When we chain node $t+\tau-1$ and node $t+\tau$ with $\{\mathcal{D}_{t+\tau-1}, \hat{\mathcal{D}}_{t+\tau}\}$ and $\{\mathcal{D}_{t+\tau}, \hat{\mathcal{D}}_{t+\tau+1}\}$, the current set of all the predicted bounding boxes of retained targets denoted by $\mathcal{P}_{t+\tau}$, is appended to $\hat{\mathcal{D}}_{t+\tau}$ for matching with $\mathcal{D}_{t+\tau}$. If $P^i_{t+\tau}$ gets a match, its tracklet will be extended by linking to the new bounding boxes.

Effectiveness and limitations. Our model is effective for handling the cases when targets appear or disappear (*i.e.*, enter or leave camera view), which are quite common for MOT. When a target is not in frame $t-1$ but appears in frame t, it is likely that no bounding box pair for it gets generated in the chain node (F_{t-1}, F_t). However, as long as this target continues to appear in frame $t+1$, it will be detected in the next chain node (F_t, F_{t+1}) and get a new tracklet and identity there. Similarly, if a target is in the frame $t-1$ but disappears from frame t, it will not be detected in node (F_t, F_{t+1}), resulting the termination of its tracklet in node $t-1$ or even $t-2$. Note that the chaining operation itself cannot be fully parameterized and therefore it cannot be optimized together with the regressions. Since the regression model (as detailed below) does the major work and there is no need to get feedback for it from the chaining operation, we still use the "end-to-end" property to describe CTracker. A pure end-to-end trainable model requires a differentiable replacement to the current IoU matching based chaining strategy.

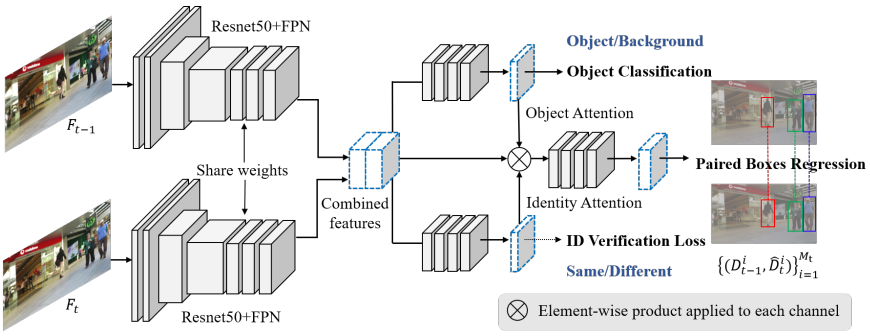

Fig. 3. Network architecture of CTracker. Given two adjacent frames, we firstly use two backbone branches with tied weights to extract the features for each frame separately. Then we concatenate features of the two frames on channel level and the combined features are used to predict the paired boxes. To highlight local informative regions for paired boxes regression, the combined features are multiplied with the attention maps from the object classification branch and the ID verification branch.

3.3 Network architecture

Overview. Our proposed CTracker network uses two adjacent frames as input and regresses the bounding box pair of the same target. To do this, we adopt ResNet-50 [18] as the backbone to extract high-level semantic features. It then integrates Feature Pyramid Networks (FPN) to generate multi-scale feature representation for subsequent prediction. In order to associate targets in adjacent frames, the scale-level feature maps from individual frames are firstly concatenated together, and then fed into the prediction network to regress bounding box pairs. As can be seen in Fig. 3, the paired boxes regression branch generates a box pair for each target, and the object classification branch predicts a score for each pair indicating the confidence of being foreground. To help the paired boxes regression branch to avoid the distraction by irrelevant yet confusing information, the object classification branch and the extra ID verification branch are used for attention guidance.

Paired Boxes Regression. Inspired by predicting the offsets relative to predefined (default) anchor boxes in object detection, we propose Chained-Anchors for the paired boxes regression branch to regress two boxes simultaneously. As a novel natural derivative of the anchors used in most object detection methods, Chained-Anchors are densely arranged on a spatial grid, each of them allows predicting two bounding boxes of the same object instance in two adjacent frames. In order to handle the large scale variation in real scenes, the K-means clustering as used in [19] is conducted on all ground-truth bounding boxes in the dataset for getting the scales of chained-anchors. And each cluster is assigned to the corresponding level of FPN for later scale specific predictions. The detected bounding box pairs are firstly post-processed with soft-NMS [20] according to the IoU of the first box in each pair, and then filtered based on the confidence

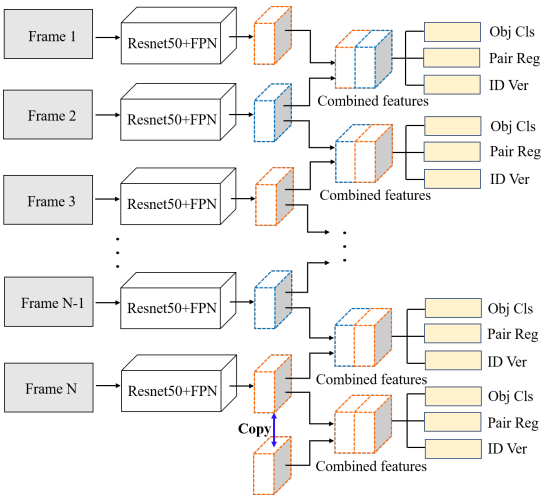

Fig. 4. Memory sharing mechanism in our CTracker. The extracted features of each frame (except the first one) are firstly used in the current chain node, and then can be saved and reused in the next chain node. Note that when making inference for the last node, the features of the last frame N is also reused as the features of the hypothetical frame $N+1$ to avoid the repeated computation for frame N.

scores from the classification branch. Finally, the remaining box pairs are chained into the whole tracking trajectories using the method described in Sec. 3.2. To keep our model simple, both the paired boxes regression branch and the classification branch only stack four consecutive 3×3 Conv layers interleaved with ReLU activations before the final convolution layer.

Joint Attention Module. We design an attention mechanism based component called Joint Attention Module (JAM) to highlight local informative regions in the combined features before the regression branch. As shown from the right of Fig. 3, the ID verification branch is introduced to get confidence scores, indicating whether the two boxes in the detected pair belong to the same target. Then both the predicted confidence map of ID verification branch and object classification branch are used as attention maps. Note that the guidance from the two branches is complementary, the confidence maps from the classification branch focuses on foreground regions while the prediction from the ID verification branch is used to highlight the features of the same target.

Feature Reuse. Since the input of the network contains two adjacent frames, the common frame of two adjacent nodes has to be used twice in the tracking process. To avoid the nearly double cost of computation and memory in inference, we propose a Memory Sharing Mechanism (MSM) to temporarily save the extracted features of the current frame and reuse them until the next node is processed, as shown in Fig. 4. Besides, in order to make inference for the last node, we make a copy of frame N as the hypothetical frame $N+1$. To further

avoid the repeated computation for the frame $N+1$, we also apply the trick of feature resue to frame N, and the feature of frame N is copied as the feature of the hypothetical frame $N+1$. We demonstrate that the proposed MSM can reduce almost half of the overall computation and time cost.

3.4 Label Assignment and Loss Design

For an arbitrary chain node (F_t, F_{t+1}), let $A_t^i = (x_a^{t,i}, y_a^{t,i}, w_a^{t,i}, h_a^{t,i})$ denote its i-th chained-anchor (where $x_a^{t,i}$ and $y_a^{t,i}$ are the box center coordinates; $w_a^{t,i}$ and $h_a^{t,i}$ are the width and height, respectively), we adopt a ground-truth bounding box matching strategy similar to that of SSD [21]. We use a matrix M to denote the result of such a matching. If G_t^j is the corresponding ground-truth bounding box in F_t for A_t^i, which is judged by the IoU ratio (higher than a threshold T_p), then we have $M_{ij} = 1$. If the IoU ratio is lower than another smaller threshold T_n, then $M_{ij} = 0$. Based on M, we can assign the ground-truth label c_{cls}^i to CTracker's classification branch for A_t^i as:

$$c_{\text{cls}}^i = \begin{cases} 1, \text{ if } \Sigma_{j=1}^{K_t} M_{ij} = 1, \\ 0, \text{ if } \Sigma_{j=1}^{K_t} M_{ij} = 0, \end{cases} \tag{1}$$

where K_t is the total number of ground-truth bounding boxes for frame F_t.

With A_t^i, suppose the predicted pair of bounding boxes are $(D_t^i, \hat{D}_{t+1}^i)$ and the corresponding ground-truth bounding boxes are (G_t^j, G_{t+1}^k) when they exist, the ID verification branch of CTracker shall get its ground-truth label as:

$$c_{\text{id}}^i = \begin{cases} 1, \text{ if } c_{\text{cls}}^i = 1 \textbf{ and } \mathcal{I}[G_t^j] = \mathcal{I}[G_{t+1}^k], \\ 0, \qquad\qquad \text{otherwise}, \end{cases} \tag{2}$$

where $\mathcal{I}[\cdot]$ represents the identity of the target in the bounding box.

We follow Faster R-CNN [22] to regress offsets of $(D_t^i, \hat{D}_{t+1}^i)$ w.r.t. A_t^i, where $D_t^i = (x_d^{t,i}, y_d^{t,i}, w_d^{t,i}, h_d^{t,i})$. Let $(\Delta_d^{t,i}, \Delta_{\hat{d}}^{t+1,i})$ denote these offsets and $(\Delta_g^{t,j}, \Delta_g^{t+1,k})$ be the offsets for the ground-truths, we list the details of $\Delta_d^{t,i} = (\Delta_{d,x}^{t,i}, \Delta_{d,y}^{t,i}, \Delta_{d,w}^{t,i}, \Delta_{d,h}^{t,i})$ as an example (the others are similar):

$$\begin{aligned} \Delta_{d,x}^{t,i} = (x_d^{t,i} - x_a^{t,i})/w_a^{t,i}, \quad \Delta_{d,y}^{t,i} = (y_d^{t,i} - y_a^{t,i})/h_a^{t,i}, \\ \Delta_{d,w}^{t,i} = \log(w_d^{t,i}/w_a^{t,i}), \quad \Delta_{d,h}^{t,i} = \log(h_d^{t,i}/h_a^{t,i}). \end{aligned} \tag{3}$$

The loss for the paired boxes regression branch is defined as follows:

$$\begin{aligned} & L_{reg}(\Delta_d^{t,i}, \Delta_{\hat{d}}^{t+1,i}, \Delta_g^{t,j}, \Delta_g^{t+1,k}) \\ & = \sum_{l \in \{x,y,w,h\}} \left[\text{smooth}_{L_1}(\Delta_{d,l}^{t,i} - \Delta_{g,l}^{t,j}) + \text{smooth}_{L_1}(\Delta_{\hat{d},l}^{t+1,i} - \Delta_{g,l}^{t+1,k}) \right] /8, \end{aligned} \tag{4}$$

where smooth_{L_1} is the smooth L_1 loss.

The total loss of CTracker is

$$L_{all} = \sum_{t,i} \left[L_{reg}(\Delta_d^{t,i}, \Delta_{\hat{d}}^{t+1,i}, \Delta_g^{t,j}, \Delta_g^{t+1,k}) + \alpha \mathcal{F}(p_{cls}^i, c_{cls}^i) + \beta \mathcal{F}(p_{id}^i, c_{id}^i) \right], \quad (5)$$

where $\mathcal{F}(p_{cls}^i, c_{cls}^i)$ and $\mathcal{F}(p_{id}^i, c_{id}^i)$ are the focal losses [23] for the classification branch and the ID verification branch (for mitigating the sample imbalance problem), respectively, with p_{cls}^i and p_{id}^i denoting their predictions (confidence scores); α and β are the weighting factors.

4 Experiment

4.1 Datasets and Evaluation Metrics

We conduct the experiments on two public datasets: MOT16 [24] and MOT17. which contain the same image sequences including 7 training sequences and 7 test sequences. However, MOT16 and MOT17 contain different detection input, and different ground-truth labels (bounding boxes and identities), which would influence the training of CTracker. In public detection, MOT16 includes DPM [25] detector while MOT17 includes DPM, Faster R-CNN [4] and SDP [26] detectors. For a fair comparison with other methods, we trained two models separately using the training data from MOT16 and MOT17, and separately applied the two models on the MOT16 test set and MOT17 test set.

In the MOTChallenge benchmark, tracking performance is measured by the widely used CLEAR MOT Metrics [27], including Multiple-Object Tracking Accuracy (MOTA), Multiple-Object Tracking Precision (MOTP), the total number of False Negatives (FN), False Positives (FP), Identity Switches (IDS), and the percentage of Mostly Tracked Trajectories (MT), Mostly Lost Trajectories (ML). ID F1 Score (IDF1) is also used to measure the trajectory identity accuracy. Among these metrics, MOTA is the primary metric to measure the overall detection and tracking performance. In addition, we use Tracker Speed in Frames Per Seconds (Hz) to measure the tracking speed of all methods.

4.2 Implementation Details

All the experiments are done with PyTorch. During training, the ground-truth boxes with a visible score above 0.1 are selected for training. To avoid overfitting, we use several data augmentation strategies such as photometric distortions, random flip and random crop. The same augmentation operation is guaranteed to apply for each image in the same training pair. Then the augmented image pair are resized or padded to the half of their original images' shorter side. We also add a novel data augmentation strategy in the temporal dimension to form chain nodes: instead of always choosing two adjacent frames, we sample two frames close to each other with a random temporal gap (1 to 3 frames).

As a speed-accuracy trade-off, we use the Resnet50 [18] network as the backbone in all the following experiments. All trainable weights except the BN parameters in Resnet50 are trained end-to-end using the Adam optimizer. We

Table 1. Ablation study on MOT17 test dataset.

Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓
Baseline	64.4	51.6	78.2	28.5%	28.0%	**16089**	178704	6336
Baseline+ObjAtten	66.0	55.7	**78.8**	31.3%	24.5%	17724	168522	5595
Baseline+ObjAtten+IDVer	65.6	55.2	78.3	**32.6%**	24.7%	25815	162489	5769
Baseline+JointAtten	**66.6**	**57.4**	78.2	32.2%	**24.2%**	22284	**160491**	**5529**

initialize the parameters for all the newly added convolutional layers with the Kaiming initialization method in [28] and set the initial learning rate to $5 \times e^{-5}$. The model training process takes 100 epochs with the batch size of 8 (4 training pairs). The weighting factors α and β in the loss function are both set to 1. In the anchor matching stage, we use 0.5 for the positive threshold and 0.4 for the negative threshold. For paired boxes post-processing, we use a threshold of 0.7 for the soft-nms, and then further filter remaining pairs with the confidence threshold of 0.4. In the chaining stage, the IoU matching threshold is 0.5, and the retention threshold of σ is 10.

4.3 Ablation Study

Performance analysis. We compare the following models on MOT17 dataset to show the effectiveness of CTracker's parts:
(1) *Baseline*. It only covers the classification branch and the paired boxes regression branch, without guidance from any attention map. This is the simplest implementation of our CTracker.
(2) *Baseline+ObjAtten*. In addition to the Baseline, the predicted confidence map of the object classification branch is used as an attention map, which is multiplied to the combined features before the paired boxes regression branch.
(3) *Baseline+ObjAtten+IDVer*. Except for the object classification branch with attention map and the paired boxes regression branch, we add the ID verification branch but do not use it as attention guidance.
(4) *Baseline+JointAtten (CTracker)*. This is the full version of our approach.

Results presented in Table 1 show that:

(1) *Baseline+ObjAtten* performs significantly better than *Baseline*, which proves the effectiveness of the object attention operation. By applying the object classification branch as the attention map of the paired boxes regression branch, we can get more accurate bounding boxes. There is a significant improvement of MOTA, which increases from 64.4 to 66.0 and MOTP also increases from 78.2 to 78.8. The more accurate bounding boxes also result in better performance of data association, with IDF1 increasing from 51.6 to 55.7.

(2) *Baseline+ObjAtten+IDVer* performs slightly worse than *Baseline+ObjAtten*. Simply adding the independent ID verification branch is weak due to the lack of bounding boxes information. Reliable identification needs good bounding boxes.

(3) *Baseline+JointAtten* further outperforms *Baseline+ObjAtten*, indicating that the ID attention operation is also beneficial. By adding the ID verification branch and using it as another guidance of the paired boxes regression branch,

Fig. 5. Qualitative results of our CTracker on MOT17 test dataset. MOT17-03 sequence is captured by a static camera and MOT17-07 sequence is captured by a moving camera. The detected bounding boxes and the tracking trajectory with the same identity are displayed by the same color.

Table 2. Time cost analysis of CTracker.

Methods	Time cost (ms)			
	Backbone	Prediction	Chaining	Total
CTracker-Det	80.27	38.78	-	119.05
CTracker w/o MSM	154.53	66.93	2.10	223.56
CTracker	80.29	65.71	2.10	148.10

the association of the regressed bounding boxes is more accurate. Though MOTA is only improved by 0.6, the IDF1 is improved by 1.7, and IDF1 can better reflect the accuracy of data association more clearly. On the other hand, by adding the ID attention, the model pays more attention to the data association and sacrifices slightly of the regression bounding box precision, thus the MOTP is decreased from 78.8 to 78.2. Qualitative results of CTracker are illustrated in Fig. 5.

Time cost analysis. We analyze the inference speed for each module in CTracker, displayed in Table 2. The time cost is measured for 1080×1920 images using single Tesla P40 and cuDNN v7 with Intel Xeon E5-2699v4@2.20GHz. In Table 2, CTracker-Det only predicts boxes for a single frame, which is the initial detection network of CTracker. Since nearly 70% of the forward time is spent on the backbone network, our original CTracker costs about double-time to perform joint detection and tracking compared with the initial detection network, the time increasing from 119.05 ms to 223.56 ms. With the help of the proposed Memory Sharing Mechanism (MSM) in Sec. 3.3, we achieve a faster joint detection and tracking model with only 29.05 ms extra cost compared with the detection network. There is just a small increase of time from 119.05 ms to 148.10 ms. To some extent, 29.05 ms per frame means the tracking module runs at 34.4 FPS, demonstrating the efficiency of our online approach.

Table 3. Comparisons of tracking results on MOT16 test dataset.

Public Detection										
Process	Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Hz↑
Offline	MHT-bLSTM [29]	42.1	47.8	75.9	14.9%	44.4%	11637	93172	753	**1.8**
	Quad-CNN [30]	44.1	38.3	76.4	14.6%	44.9%	**6388**	94775	745	**1.8**
	EDMT [5]	45.3	47.9	75.9	17.0%	**39.9%**	11122	87890	639	**1.8**
	LMP [31]	**48.8**	**51.3**	**79.0**	**18.2%**	40.1%	6654	**86245**	**481**	0.5
Online	CDA-DDAL [32]	43.9	45.1	74.7	10.7%	44.4%	6450	95175	676	-
	STAM [12]	46.0	50.0	74.9	14.6%	43.6%	6895	91117	**473**	-
	DMAN [13]	46.1	**54.8**	73.8	17.4%	42.7%	7909	89874	532	-
	MOTDT [33]	47.6	50.9	74.8	15.2%	38.3%	9253	85431	792	**20.6**
	Tracktor [7]	**54.4**	52.5	**78.2**	**19.0%**	**36.9%**	**3280**	**79149**	682	-
Private Detection										
Process	Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Hz↑
Offline	NOMT [34]	62.2	**62.6**	**79.6**	32.5%	31.1%	**5119**	63352	**406**	11.5
	MCMOT-HDM [35]	62.4	51.6	78.3	31.5%	24.2%	9855	57257	1394	**34.9**
	KDNT [3]	**68.2**	60.0	79.4	**41.0%**	**19.0%**	11479	**45605**	933	0.7
Online	EAMTT [36]	52.5	53.3	78.8	19.0%	34.9%	**4407**	81223	910	12.0
	DeepSORT [16]	61.4	62.2	79.1	32.8%	**18.2%**	12852	56668	**781**	20.0
	CNNMTT [37]	65.2	62.2	78.4	32.4%	21.3%	6578	55896	946	11.2
	POI [3]	66.1	**65.1**	**79.5**	**34.0%**	20.8%	5061	55914	805	9.9
	CTracker (Ours)	**67.6**	57.2	78.4	32.9%	23.1%	8934	**48305**	1897	**34.4**

Table 4. Comparisons of tracking results on MOT17 test dataset.

Public Detection										
Process	Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Hz↑
Offline	MHT-bLSTM [29]	47.5	51.9	**77.5**	18.2%	41.7%	25981	268042	2069	**1.8**
	EDMT [5]	50.0	51.3	77.3	**21.6%**	36.3%	32279	**247297**	2264	**1.8**
	JCC [38]	51.2	**54.5**	75.9	20.9%	37.0%	25937	247822	**1802**	-
	FWT [6]	**51.3**	47.6	77.0	21.4%	**35.2%**	**24101**	247921	2648	-
Online	DMAN [13]	48.2	**55.7**	75.9	19.3%	38.3%	26218	263608	2194	-
	MOTDT [33]	50.9	52.7	76.6	17.5%	**35.7%**	24069	250768	2474	**20.6**
	Tracktor [7]	**53.5**	52.3	**78.0**	**19.5%**	36.6%	**12201**	**248047**	**2072**	-
Private Detection										
Process	Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Hz↑
Online	Tracktor+CTdet [7]	54.4	56.1	78.1	25.7%	29.8%	44109	210774	2574	-
	DeepSORT [16]	60.3	**61.2**	**79.1**	31.5%	**20.3%**	36111	185301	**2442**	20.0
	CTracker (Ours)	**66.6**	57.4	78.2	**32.2%**	24.2%	**22284**	**160491**	5529	**34.4**

4.4 Benchmark Evaluation

We compare our CTracker approach with other MOT methods on both MOT16 and MOT17 test datasets. For comparison, we trained our model separately using the MOT16 training data and MOT17 training data. Table 3 and Table 4 compare the tracking results of all the methods separately on MOT16 and MOT17 test dataset. From Table 3 and Table 4 we can find that:

(1) In the private detection part of both MOT16 and MOT17, our CTracker significantly outperforms existing online MOT methods in terms of MOTA. In MOT16, the MOTA of our approach is only 0.6 lower than the best offline method KDNT [3], while it is 1.5 higher than its online version POI [3]. In addi-

tion, KDNT and POI use many extra training data, including ETHZ pedestrian dataset [39], Caltech pedestrian dataset [40] and their own collected surveillance dataset [3]. While we only use the training data of MOT16. MOTA is the primary metric reflecting the overall detection and tracking performance, which proves the effectiveness of our approach.

(2) In the public detection part, Tracktor [7] performs the best in terms of MOTA. To have a comparison with Tracktor using the same detection result, we reproduce Tracktor using its code. Tracktor+CTdet in Table 4 is the tracking result of Tracktor using the detection result of our CTracker. Compared with the results of public detection, the MOTA of Tracktor+CTdet increases from 53.5 to 54.4 and IDF1 increases from 52.3 to 56.1, which indicates that the performance of our detection is better than the public detection. Besides, our CTracker outperforms Tracktor+CTdet in terms of all the metrics except IDS, which further proves the superior tracking performance of our CTracker.

(3) On the other hand, to keep the simplicity and efficiency of our CTracker, we abandon using the patch-level ReID features of the detected boxes like other MOT methods to enhance cross-frame data association. Thus, the IDF1 and IDS of our CTracker approach are lower than several methods. We conduct an extra experiment by adding features, introduced in the supplementary. To further prove the efficiency of our approach, we compare the time cost of CTracker with other state-of-the-art MOT methods on the MOT16 and MOT17 benchmark, as shown in the Hz column of Tabel 3 and Tabel 4. From Tabel 3 and Tabel 4 we can find that CTracker achieves the best tracking speed among all online MOT methods, although the fastest offline method runs at a similar tracking speed as our CTracker, but has a much lower MOTA than our CTracker, demonstrating the effectiveness and efficiency of our approach.

5 Conclusion

We designed a novel joint multiple-object detection and tracking framework named Chained-Tracker in this paper, which is the first totally end-to-end solution as far as we are aware. Different from existing methods, we use two adjacent frames as the input of our network, which is called a chain node. The network regresses a pair of bounding boxes for the same target in the two adjacent frames, guided by a simple yet novel joint attention module: an interplay of detection-driven object attention and ID verification-injected identity attention. Using the simple IoU information, two adjacent and overlapping nodes can be chained by their boxes in the common frame. The tracking trajectories can be generated by alternately applying the paired boxes regression and node chaining. Extensive experiments on widely used MOT benchmarks demonstrate the superiority of our approach in terms of both effectiveness and efficiency.

Acknowledgement

This work was supported by a MSRA Collaborative Research 2019 Grant.

References

1. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV. (2009)
2. Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: AVSS. (2017)
3. Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J.: Poi: multiple object tracking with high performance detection and appearance feature. In: ECCV. (2016)
4. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
5. Chen, J., Sheng, H., Zhang, Y., Xiong, Z.: Enhancing detection model for multiple hypothesis tracking. In: CVPRW. (2017)
6. Henschel, R., Leal-Taixé, L., Cremers, D., Rosenhahn, B.: Fusion of head and full-body detectors for multi-object tracking. In: CVPRW. (2018)
7. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: ICCV. (2019)
8. Lu, Z., Rathod, V., Votel, R., Huang, J.: Retinatrack: Online single stage joint detection and tracking. In: CVPR. (2020)
9. Sun, S., Akhtar, N., Song, H., Mian, A.S., Shah, M.: Deep affinity network for multiple object tracking. TPAMI (2019)
10. Chu, P., Ling, H.: Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In: ICCV. (2019)
11. Li, C., Dobler, G., Feng, X., Wang, Y.: Tracknet: Simultaneous object detection and tracking and its application in traffic video analysis. arXiv preprint arXiv:1902.01466 (2019)
12. Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., Yu, N.: Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In: ICCV. (2017)
13. Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., Yang, M.H.: Online multi-object tracking with dual matching attention networks. In: ECCV. (2018)
14. Gao, X., Jiang, T.: Osmo: Online specific models for occlusion in multiple object tracking under surveillance scene. In: ACMMM. (2018)
15. Kuhn, H.W.: The hungarian method for the assignment problem. NRL (1955)
16. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP. (2017)
17. Peng, J., Wang, T., Lin, W., Wang, J., See, J., Wen, S., Ding, E.: Tpm: Multiple object tracking with tracklet-plane matching. PR (2020)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
19. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: CVPR. (2017)
20. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms – improving object detection with one line of code. In: ICCV. (2017)
21. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. (2016)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
23. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: CVPR. (2017)
24. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)

25. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI (2010)
26. Yang, F., Choi, W., Lin, Y.: Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In: CVPR. (2016)
27. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. JIVP (2008)
28. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV. (2015)
29. Kim, C., Li, F., Rehg, J.M.: Multi-object tracking with neural gating using bilinear lstm. In: ECCV. (2018)
30. Son, J., Baek, M., Cho, M., Han, B.: Multi-object tracking with quadruplet convolutional neural networks. In: CVPR. (2017)
31. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person re-identification. In: CVPR. (2017)
32. Bae, S.H., Yoon, K.J.: Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. TPAMI (2018)
33. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: ICME. (2018)
34. Choi, W.: Near-online multi-target tracking with aggregated local flow descriptor. In: ICCV. (2015)
35. Lee, B., Erdenee, E., Jin, S., Nam, M.Y., Jung, Y.G., Rhee, P.K.: Multi-class multi-object tracking using changing point detection. In: ECCV. (2016)
36. Sanchez-Matilla, R., Poiesi, F., Cavallaro, A.: Online multi-target tracking with strong and weak detections. In: ECCV. (2016)
37. Mahmoudi, N., Ahadi, S.M., Rahmati, M.: Multi-target tracking using cnn-based features: Cnnmtt. MTAP (2019)
38. Keuper, M., Tang, S., Andres, B., Brox, T., Schiele, B.: Motion segmentation & multiple object tracking by correlation co-clustering. TPAMI (2018)
39. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: A mobile vision system for robust multi-person tracking. In: CVPR. (2008)
40. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR. (2009)