A Unified Object Motion and Affinity Model for Online Multi-Object Tracking

Junbo Yin[1], Wenguan Wang[2]*, Qinghao Meng[1], Ruigang Yang[3,4,6], Jianbing Shen[5,1]

[1]Beijing Lab of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, China

[2]ETH Zurich, Switzerland [3] Baidu Research [4] National Engineering Laboratory of Deep Learning Technology and Application, China

[5] Inception Institute of Artificial Intelligence, UAE [6] University of Kentucky, Kentucky, USA

yinjunbo@bit.edu.cn wenguanwang.ai@gmail.com

https://github.com/yinjunbo/UMA-MOT

Abstract

Current popular online multi-object tracking (MOT) solutions apply single object trackers (SOTs) to capture object motions, while often requiring an extra affinity network to associate objects, especially for the occluded ones. This brings extra computational overhead due to repetitive feature extraction for SOT and affinity computation. Meanwhile, the model size of the sophisticated affinity network is usually non-trivial. In this paper, we propose a novel MOT framework that unifies object motion and affinity model into a single network, named UMA, in order to learn a compact feature that is discriminative for both object motion and affinity measure. In particular, UMA integrates single object tracking and metric learning into a unified triplet network by means of multi-task learning. Such design brings advantages of improved computation efficiency, low memory requirement and simplified training procedure. In addition, we equip our model with a task-specific attention module, which is used to boost task-aware feature learning. The proposed UMA can be easily trained end-to-end, and is elegant – requiring only one training stage. Experimental results show that it achieves promising performance on several MOT Challenge benchmarks.

1. Introduction

Online multi-object tracking (MOT) aims to accurately locate trajectory of each target while maintaining their identities with information accumulated up to the current frame. In the last decades, MOT has attracted increasing attentions, as it benefits a wide range of applications, such as video surveillance analyses and autonomous driving [48, 56, 57].

Current MOT solutions typically involve an object motion model and an affinity model. The former one leverages temporal information for object instance localization and tracklet generation, while the latter deals with distractors (*e.g.*, targets with similar appearance) or occlusions by measuring object similarity in data association. Specifically, some online MOT algorithms are based on a tracking-by-detection paradigm [25, 39, 52, 1, 29], *i.e.*, associating detections across frames by computing pairwise affinity. Thus they mainly focus on the design of the affinity model. However, as temporal cues are not explored in the object detection phase, the quality of detections is often limited, further decreasing the MOT performance. MOT scenarios, *e.g.*, the video sequences in MOT Challenge [32, 38], often yield crowded people with rare poses or various sizes. In such cases, even the leading detectors [43] may produce many False Positive (FP) and False Negative (FN) results, causing adverse impact on the subsequent data association stage.

This calls for better leveraging motion cues in MOT. Thus another trend is to apply single object trackers (SOTs) in online MOT [11, 61]. They take advantage of SOTs to address the value of temporal information and recover the missing candidate detections. Such paradigm yields more natural tracklets and usually leads to better tracking results according to the FN metric. However, crowded distractors and their frequent interactions often lead to occlusion situations, which are quite challenging for these solutions. To tackle this issue, follow-up methods [44, 11, 68, 9, 10] integrate SOT based motion model with affinity estimation. In particular, they first recognize the state of the targets according to the confidence of SOTs, then update the tracked targets and maintain the identities of the occluded targets through affinity measure for tracklet-detection pairs in data association phase. Though inspired, they still suffer from several limitations. First, features used for SOTs and affinity measure are extracted from two separate models, which incurs expensive computational overhead. Second, since they do not make use of SOT features in affinity computation, they have to train an extra affinity network (*e.g.*, ResNet50 in [68] and ResNet101 in [10]) to remedy this. This further increases their memory demand, which critically limits their applicability in source-constrained en-

*Corresponding author: *Wenguan Wang*.

vironments. Third, the independent feature extraction of SOTs and affinity models, and the complicated affinity network design, together make the training procedures sophisticated, which often require multiple alternations or cascaded-training strategy. Moreover, they do not explore the relation of the SOTs and the affinity model, *i.e*., the affinity model could help the SOTs to access the identity information and thus learn more discriminative features to better handle occlusions.

To alleviate the above issues, we propose a multi-task learning based online MOT model, UMA, which end-to-end integrates the SOT based motion model and affinity network into a unified framework. The learnt features are promoted to capture more identity-discriminative information, thus simplifying the training and testing procedures. In particular, it unifies a Siamese SOT and a ranking network into a triplet architecture. Two branches of the triplet network, *e.g*., the positive and anchor branches, count for the SOT based motion prediction task, while all the three branches address the target identity-aware ranking task by metric learning. This provides several benefits. First, the metric learning within the ranking task assigns the learnt features identity-discriminative ability, facilitating the SOT model to better locate the targets and handle occlusions. Second, this enables feature sharing between the SOT based tracklet generation stage and the affinity-dependent data association stage, eliminating the requirement of designing an additional affinity network and improving the computation efficiency. Third, it provides a more straightforward, one-step training protocol instead of previous sophisticated, multi-alternation or cascaded training strategy. Furthermore, a task-specific attention (TSA) module is equipped with our UMA model, to address the specific nature of the multiple tasks and boost more task-specific feature learning. It performs context exploitation self-adaptively on the shared features extracted by the multi-task network and is lightweight with budgeted computational cost, meanwhile producing better performance. To summarize, we propose a triplet network, UMA, which unifies the object motion prediction and affinity measure tasks in online MOT. UMA addresses SOT-applicable as well as association-discriminative feature learning with an attentive multi-task learning mechanism. This presents an elegant, effective yet efficient MOT model with lower memory requirement and a simple, end-to-end training protocol. Further with an elaborately designed online tracking pipeline, our lightweight model reaches state-of-the-art performance against most online and even offline algorithms on several MOT Challenge benchmarks.

2. Related Work

MOT. Existing MOT approaches can be categorized into offline and online modes. *Offline* methods [41, 14, 51, 54, 52] can leverage both past and future frames for batch processing. They typically consider MOT as a global optimization problem in various forms such as multi-cut [51, 52], k-partite graph [66, 13] and network flow [67, 14]. Though favored in handling ambiguous tracking results, they are not suitable for causal applications such as autonomous driving.

Online MOT methods can only access the information available up to the current frame, thus easily suffering from target occlusions or noisy detections. The majority of previous approaches [1, 2, 25, 39, 63] adopt a tracking-by-detection pipeline, whose performance is largely limited by the detection results. Some others [68, 44, 11, 9, 10] instead apply SOTs [22, 4, 34, 17, 16] to carry out online MOT and generally gain better results.

Object Motion Models in Online MOT. Basically, object motion model is helpful in dealing with the noisy detections. For instance, Xiang *et al.* [62] employ an optical flow-based SOT, TLD [27], to track individual target. Sadeghian *et al.* [44] further extend this pipeline with a multi-LSTM network for exploiting different long-term cues. After that, Zhu *et al.* [68] equip their framework with a more advanced tracker: ECO [12], and design an attention-based network to handle occlusions. Their promising results demonstrate the advantages of applying SOTs as motion models. However, all these approaches require an additional affinity model to address occlusions, and often learn features for the SOTs and affinity models independently, leading to increased computation cost, non-trivial memory demand and sophisticated training protocols. Though [11] uses a shared backbone to extract features for all the targets, multiple online updating sub-nets are further added to specifically handle each target. In sharp contrast, we attempt to learn a 'universal' feature that preserves enough information for both motion and affinity models, which essentially simplifies the training and testing procedures.

Object Affinity Models in Online MOT. In the data association phase, the object affinity model is usually used to link tracklets or detections cross frames in terms of the pairwise affinity, which is a crucial way to handle occlusions in online MOT. To produce reliable affinity estimations, object appearance cues are indispensable, and Siamese or triplet networks [8, 36, 55] with metric learning provide powerful tools to acquire a discriminative and robust feature embedding. In particular, Leal-Taixé *et al*. [31] apply a Siamese network to estimate the affinity of the provided detections by aggregating targets appearance and optical flow information. Son *et al*. [47] propose a quadruplet loss to stress targets appearance together with their temporal adjacencies. In [52], a Siamese network is used to leverage human pose information for long-range target-relation modeling. Voigtlaender *et al*. [53] extend the Mask R-CNN [20] with 3D convolutional layers and propose an association head to extract embedding vectors for each region proposals by us-

ing the batch hard triplet loss [24]. Bergmann *et al*. [2] also present a short-term re-identification model based on the Siamese network. Xu *et al*. [63] jointly utilize the appearance, location and topology information to compute the affinity by employing the relation networks [58] in both the spatial and temporal domains. Notably, all these methods work on the tracking-by-detection mode. Differently, we in-depth inject metric learning into a SOT model, through a unified triplet network. It learns a discriminative feature for both the object motion prediction and affinity measure sub-tasks, bringing an effective yet efficient solution.

3. Our Algorithm

In this section, we first give a brief review of the Siamese SOT [4] (§3.1), as it is used as the backbone of our model. Then, the details of our UMA model are presented in §3.2. Finally, in §3.3, we elaborate on our whole online MOT pipeline. As UMA utilizes a *single* feature extraction network for both SOT based tracklet generation and object affinity measure, it presents a more efficient online solution with many non-trivial technical improvements.

3.1. Preliminary of Siamese SOT

Our backbone model is a recently proposed deep tracker: SiamFC [4], which is based on a Siamese network and shows promising performance in the single object tracking field. It operates at around 120 fps on one GPU, built upon the lightweight AlexNet [30].

Basically, SiamFC transfers tracking task to patches matching in an embedding space. The Siamese network is learnt as the matching function, which is applied to find the most similar patch in the new frame compared with the initial target patch in the first frame. Specifically, as shown in Fig. 1, the Siamese tracker comprises two parameter-sharing branches, each of which is a 5-layer convolutional network ϕ. One branch takes as input the target detection given at the first frame, called as exemplar. The other one takes as input the instance, *i.e*., the searching region in each subsequent frame including the candidate patches. Given the features embeddings: $\phi(z)$ and $\phi(x)$, of the exemplar z and instance x, a cross correlation layer τ is applied to compare their similarity and obtain a response map v:

$$v = \tau(x, z) = \phi(x) * \phi(z) + b, \tag{1}$$

where '$*$' indicates the convolutional operator and b is the biases term. Then, given a ground-truth map y, a logistic loss is applied on v for training:

$$L_{\text{SOT}} = \sum\nolimits_{p \in \mathcal{P}} \frac{1}{|\mathcal{P}|} \log\left(1 + e^{-v_p y_p}\right), \tag{2}$$

where p indicates a candidate position in the lattice $\mathcal{P}$ of x. For each candidate $x_p \in x$ from the instance input x, v_p is the response value of an exemplar-candidate pair, *i.e*., $v_p = f(x_p, z)$, and $y_p \in \{+1, -1\}$ is the corresponding ground-truth label for v_p.

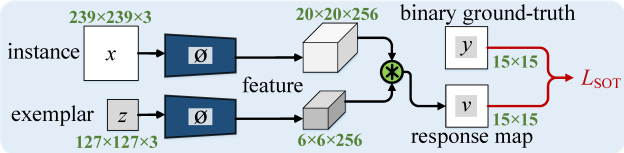

Figure 1: **Illustration of the network architecture of the Siamese SOT** during the training phase.

3.2. Our UMA Model for Online MOT

Main Idea. Previous SOT based online MOT methods typically design an extra network for affinity measure, in addition to the SOT network. In contrast, we try to integrate the object motion and affinity networks into a unified model. This brings several advantages, as mentioned in §1. The core idea is to enforce the network to simultaneously learn the two tasks: single object tracking and affinity prediction, forming a unified multi-task learning framework. Some ones may concern the features obtained from top-performing SOTs are already good enough for affinity measure. Actually, though SOT features are powerful, they are not discriminative enough to estimate a reliable affinity. This is because SOTs rarely access the identity information during training, thus their features typically distinguish targets from the substantial background well, while capture relatively less identity information. From the perspective of data association, SOT features have already encoded some useful information, thus it is more desirable and efficient to make use of these features instead of learning extra 'affinity features' from scratch. These considerations motivate us to learn a unified yet powerful feature that is applicable to both tasks, yielding an elegant online MOT framework.

Triplet-based MOT Framework. To achieve our goal, our UMA model is designed as a triplet network architecture, as shown in Fig. 2, where the triplet network comprises three weight-sharing branches, *i.e*., an exemplar branch, a positive-instance branch and a negative-instance branch. We adopt the exemplar as the *anchor*. The instances from the same targets are used as *positive samples*, while the ones from different targets as *negative*. The integration of the exemplar branch and positive-instance branch can be viewed as a Siamese tracking network, while the whole triplet network yields a unified metric learning framework.

Specifically, for the i^{th} target, given an exemplar z_i, a positive-instance x_i, and a negative-instance x_j sampled from a different target j, we extract their features from the backbone AlexNet: $\boldsymbol{f}_{z_i} = \phi(z_i) \in \mathbb{R}^{6\times6\times256}$, $\boldsymbol{f}_{x_i} = \phi(x_i) \in \mathbb{R}^{20\times20\times256}$, and $\boldsymbol{f}_{x_j} = \phi(x_j) \in \mathbb{R}^{20\times20\times256}$. Then, for the single object tracking task, it can be trained over $(\boldsymbol{f}_{z_i}, \boldsymbol{f}_{x_i})$ using Eq. 2.

For the whole triplet-based model, it is designed to learn

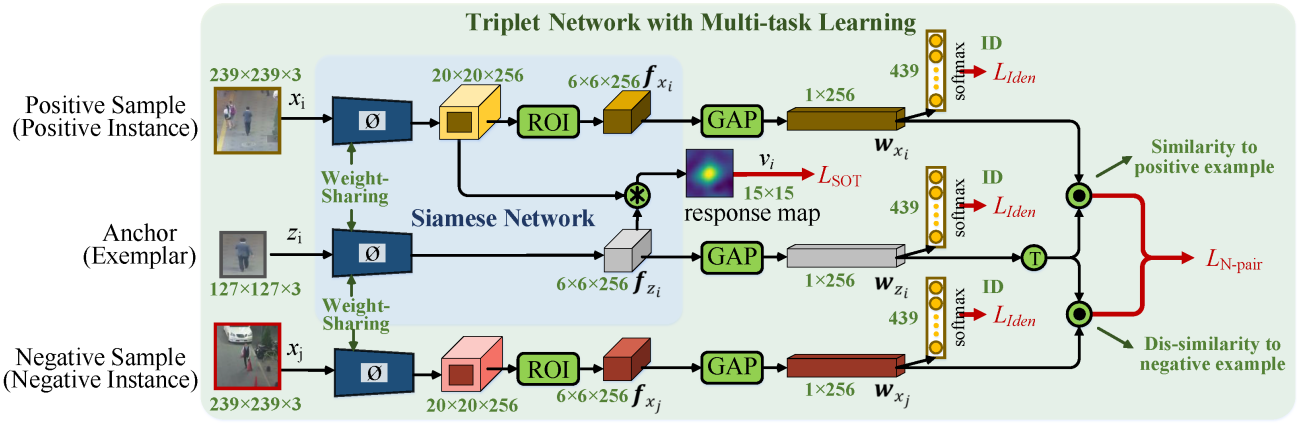

Figure 2: **Illustration of our proposed UMA model**, which is built upon a triplet architecture with multi-task learning. UMA simultaneously learns two tasks: SOT based object motion prediction and affinity-dependent ranking, producing a strong feature that is applicable to both the tracklet generation as well as the affinity measure phases.

the ranking task for affinity estimation. This is achieved by a metric learning paradigm, *i.e.*, enforcing the features of positive examples closer to the anchor than other negative examples. Specifically, we first apply the ROI-Align [20] layer on $\boldsymbol{f}_{x_i}$ and $\boldsymbol{f}_{x_j}$ respectively, to extract two $6\times6\times256$ target features from the centers of x_i and x_j (as the targets are centred at the instance examples during training [4]). Such operation allows the model to specifically focus on learning more identity-discriminative features for affinity measure, suppress the information from cluster background, and produce feature maps with the same resolution to the anchor feature $\boldsymbol{f}_{z_i}$. Then a global average pooling (GAP) is applied to the anchor feature $\boldsymbol{f}_{z_i}$, as well as the aligned features of $\boldsymbol{f}_{x_i}$ and $\boldsymbol{f}_{x_j}$, producing three 256-d features, denoted as $\boldsymbol{w}_{z_i}$, $\boldsymbol{w}_{x_i}$ and $\boldsymbol{w}_{x_j}$, respectively. This enforces the regularization of the network and reduces model size.

Given a mini-batch of N training sample pairs, *e.g.*, $\mathcal{B}=\{(x_i,z_i)\}_{i=1}^{N}$, a standard triplet loss [59] works in the following format:

$$L_{\text{Tri}}=\frac{1}{N}\sum_{i,j}^{N}\max(0,||\boldsymbol{w}_{z_i}-\boldsymbol{w}_{x_i}||_2^2-||\boldsymbol{w}_{z_i}-\boldsymbol{w}_{x_j}||_2^2+m), \quad (3)$$

where m is a margin that is enforced between positive and negative pairs. The objective of this loss is to keep distance between the anchor and positive smaller than the distance between the anchor and negative. However, in our batch construction, the number of the positive samples is significantly smaller than the negative ones, which will restrict the performance of the triplet loss L_{Tri} in hard data mining [24]. To overcome this hurdle, we replace Eq. 3 with a N-pair loss [46]:

$$L_{\text{N-pair}}=\frac{1}{N}\sum_{i=1}^{N}\log\Big(1+\sum_{i\neq j}^{N}\exp\big(\boldsymbol{w}_{z_i}^{\top}\boldsymbol{w}_{x_j}-\boldsymbol{w}_{z_i}^{\top}\boldsymbol{w}_{x_i}\big)\Big). \quad (4)$$

The rationale is that, after looping over all the triplets in $\mathcal{B}$, the final distance metric can be balanced correctly.

Additionally, with the target identity at hand, we can further minimize a cross-entropy based identification loss [9]:

$$L_{\text{Iden}}=-(\frac{1}{N}\sum_{i=1}^{N}\log\hat{p}_{z_i}+\frac{1}{N}\sum_{i=1}^{N}\log\hat{p}_{x_i}), \quad (5)$$

where $\hat{p}_{z_i}\in[0,1]$ is the predicted probability of z_i for the ith identity class. The identity prediction score is obtained by applying two fully connected layers (with dimension 512 and 439) and a *softmax* layer over $\boldsymbol{w}_{z_i}$ or $\boldsymbol{w}_{x_i}$. Please note that there are a total of 439 identities in our training set.

Hence, the final loss is computed as a combination of the SOT loss L_{SOT}, defined over the Siamese network, and the affinity-related losses $L_{\text{N-pair}}$ and L_{Iden}, defined over the whole triplet network:

$$L=L_{\text{SOT}}+(\lambda_1 L_{\text{N-pair}}+\lambda_2 L_{\text{Iden}}), \quad (6)$$

where λs are the coefficients to balance different losses. In this way, both the SOT based motion model and the ranking based affinity model can be trained end-to-end in a unified triplet network, which facilitates the training process.

Furthermore, with our multi-task design, we can derive a reliable affinity from the features extracted from our model:

$$c=\boldsymbol{w}_I^{\top}\boldsymbol{w}_{I'}, \quad (7)$$

where I and I' are two image patch inputs, *e.g.*, an exemplar with an instance region or a detection patch, and c is the affinity. To in-depth analyze the advantage of the features learnt from our model in affinity measure, we use the features extracted from the Siamese SOT model to compute the affinity, where the SOT model does not use either the additional branch or the extra losses (*i.e.*, $L_{\text{N-pair}}$ and L_{Iden}). Fig. 3 gives the performance comparison of the two models with hard cases, *e.g.*, the affinity between negative sample pairs with similar appearance or that between positive sample pairs with changeable appearance. From Fig. 3 (a) we can find that, when only using L_{SOT}, the affinity between negative sample pairs is even larger than the one between

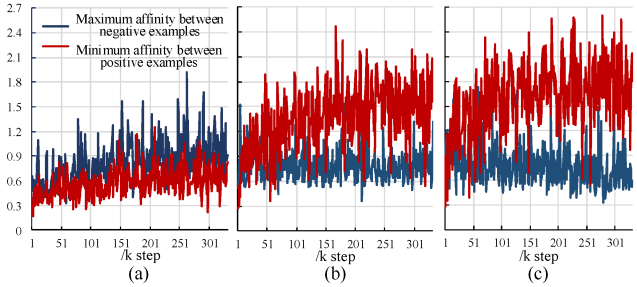

Figure 3: **Affinities measured using the features** (a) $\boldsymbol{f}$ from the Siamese SOT with L_{SOT} loss, (b) $\boldsymbol{w}$ from the triplet network with multi-task learning, and (c) $\boldsymbol{w}^{\text{AFF}}$ from our whole UMA with multi-task learning and TSA module, respectively.

positive sample pairs. This clearly demonstrates the weak discriminability of the SOT features. In Fig. 3 (b) and (c), the affinity between positive sample pairs is substantially larger than that between negative ones even in hard cases, which proves our multi-task features $\boldsymbol{w}$ are highly applicable for affinity measure. More detailed quantitative experiments can be found at §4.

Task-Specific Attention Module. For our triplet-based model described above, an identical feature produced by the backbone AlexNet $\phi(\cdot)$ is used for both the SOT based motion prediction and affinity measure tasks. Potential problems of such design lie in the loss of sensibility to subtle distinctions between the two tasks and the ignorance of their task-specific factors. A meaningful feature for SOT may not best fit affinity measure, vice verse. For example, context information is often stressed in SOT, *e.g.*, auxiliary objects approaching the target may afford correlation information for tracking [65, 46]. However, for affinity measure, local semantic features around the key points are more informative to identify the query target [7, 49], while the auxiliary objects may interfere with the determination. To address this issue, we further equip our model with a task-specific attention (TSA) module for emphasizing task-aware feature learning with very low computation cost.

Our TSA module is designed based on the famous Squeeze-and-Excitation Network (SENet) [26], as it does not rely on extra input and is trivial in runtime, which are essential for online MOT. It re-weights the feature response across channels using the *squeeze* and *excitation* operations. More specifically, the *squeeze* operator acquires global information via aggregating the features across all the spatial locations through channel-wise global average pooling:

$$s_l = \text{GAP}_l(\boldsymbol{f}) = \text{GAP}_l(\phi(\cdot)) \in \mathbb{R}, \tag{8}$$

where GAP_l indicates global average pooling over the feature $\boldsymbol{f}$ in l^{th} channel. In the *excite* step, a gating mechanism is employed on the channel-wise descriptor $\boldsymbol{s} = [s_1, s_2, \ldots, s_{256}] \in \mathbb{R}^{256}$:

$$\boldsymbol{a} = \sigma(\boldsymbol{W}_2\delta(\boldsymbol{W}_1\boldsymbol{s})) = [a_1, a_2, \ldots, a_{256}] \in [0,1]^{256}. \tag{9}$$

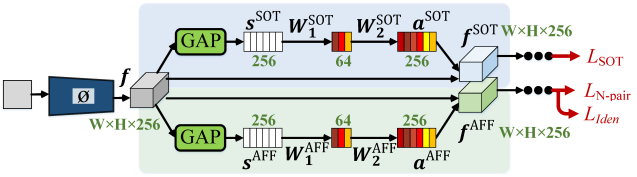

Figure 4: **Illustration of the TSA module**, which enables our model to stress task-specific features.

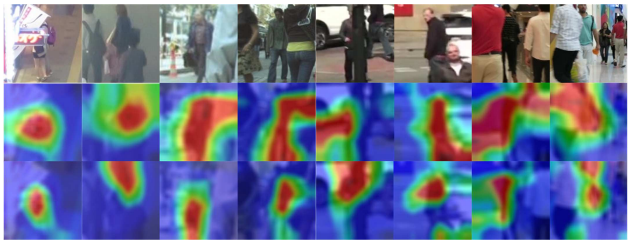

Figure 5: **Visualization of task-specific features** stressed by TSA module. Features extracted for tracking part are shown in the 2^{nd} row, while those for affinity measure part are in the 3^{rd} row.

σ and δ are *sigmoid* and *ReLU* functions, respectively. Through the dimensionality reduction and increasing operations (parameterized by two fully connected layers $\boldsymbol{W}_1 \in \mathbb{R}^{64\times256}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{256\times64}$), the attention vector $\boldsymbol{a}$ encodes non-mutually-exclusive relations among the 256 channels.

Using the SENet framework, our TSA module learns two kinds of attentions: $\boldsymbol{a}^{\text{SOT}}$ and $\boldsymbol{a}^{\text{AFF}}$ for addressing different tasks (see Fig. 4). $\boldsymbol{a}^{\text{SOT}}$ and $\boldsymbol{a}^{\text{AFF}}$ are first applied to re-weight the channels of the 'universal' feature $\boldsymbol{f} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{256}]$ extracted from the backbone AlexNet:

$$\begin{aligned} \boldsymbol{f}^{\text{SOT}} &= [a_1^{\text{SOT}} \cdot \boldsymbol{f}_1, \ldots, a_{256}^{\text{SOT}} \cdot \boldsymbol{f}_{256}], \\ \boldsymbol{f}^{\text{AFF}} &= [a_1^{\text{AFF}} \cdot \boldsymbol{f}_1, \ldots, a_{256}^{\text{AFF}} \cdot \boldsymbol{f}_{256}]. \end{aligned} \tag{10}$$

Then we feed the supervision of L_{SOT} to the SOT-aware feature $\boldsymbol{f}^{\text{SOT}}$, while add $L_{\text{N-pair}}$ and L_{Iden} losses to the affinity-related feature $\boldsymbol{w}^{\text{AFF}}$ (derived from $\boldsymbol{f}^{\text{AFF}}$, as described before). In this way, the TSA module will learn to generate task-specific attentions. Through our lightweight TSA mechanism, our model is able to produce task-specific features while using a same backbone network $\phi(\cdot)$. For the single object tracking task, the SOT-aware attention $\boldsymbol{a}^{\text{SOT}}$ can stress useful context for boosting tracking accuracy. For the affinity measure, the affinity-aware attention $\boldsymbol{a}^{\text{AFF}}$ is employed to capture fine-grained local semantic features. Thus the targets with changeable appearance can be better aligned. From Fig. 3 (c), we can observe further improved affinity estimations using the affinity-specific attention enhanced feature $\boldsymbol{w}^{\text{AFF}}$. Visualization of the attention-enhanced features for each task is presented in Fig. 5. More detailed quantitative analyses can be found in §4.2.

3.3. Our Online MOT Pipeline

We have elaborated on our network architecture. Next we will detail our whole pipeline for online MOT. Basically,

each target is associated with two states, *i.e.*, tracked or occluded, decided by the occlusion detection. We first apply our UMA (working on the SOT mode) to generate tracklets for the tracked targets. Then we perform data association based on the affinity produced by UMA (working on the ranking mode), to recover the occluded targets.

Tracklet Generation and Occlusion Detection: During tracking, our UMA is applied to each target (exemplar z), which is initialized by the provided detections. UMA is able to update the position of each target used as a SOT (relying on the SOT-specific feature $\boldsymbol{f}^{\text{SOT}}$). Simultaneously, it measures the affinity between the exemplar and instances in subsequent frames to detect occlusions (using the affinity estimation-related feature $\boldsymbol{f}^{\text{AFF}}$).

Concretely, we use a search region centred at the previous position of the target as the instance x. Given z and x, we get a response map v via Eq. 1 with SOT-specific feature $\boldsymbol{f}^{\text{SOT}}$. Then the target bounding box (bbox) is obtained according to the position with the maximum score in v [4].

Meanwhile, with the exemplar z and instance x, UMA computes the affinity for detecting occlusions. It works in the ranking mode and uses the affinity-specific features $\boldsymbol{w}_z^{\text{AFF}}$ and $\boldsymbol{w}_x^{\text{AFF}}$ to get the affinity c (Eq. 7). Note that the target may appear in any part of x during the tracking stage, thus we apply ROI-Align [20] on the instance feature ($\boldsymbol{f}_x^{\text{AFF}} \in \mathbb{R}^{22\times22\times256}$ during testing) to obtain an aligned target feature, with the bbox provided by the SOT. Then we get $\boldsymbol{w}_x^{\text{AFF}}$ through GAP and further compute the affinity c. Compared with previous works [62, 44, 68] that use the confidence produced by SOTs to detect the occlusions, our method gives a more robust result, which is illustrated in Fig. 6. Additionally, following [68], we integrate the affinity with the historic average intersection-over-union (IOU) between the tracklet and the nearest detection, for filtering out the FP tracking results and detecting occlusions more reliably. Once the affinity c is below a threshold α or the average IOU is below β, the target is recognized to be occluded; tracked otherwise. We further refine the tracked bboxes by averaging the nearest detection with greedy algorithm. Then the refined bboxes are gathered as the tracklet of the target z. Detections that have IOU below a certain threshold γ with any tracking bboxes will be regarded as candidate detections, *e.g.*, a reappearing occluded target or an entirely new target.

Data Association: During data association, we deal with those candidate detections and address the occluded targets, *i.e.*, recognizing a candidate detection as a reappearing occluded or an entirely new target, and then recovering its identity (if the first case) or assigning a new identity (if the second). Different from prior work designing complicated strategies [52, 13, 5], we use a relatively simple data association method, due to the reliable affinity measured from our UMA. Given the candidate detection set $\mathcal{D}$ and tracklet

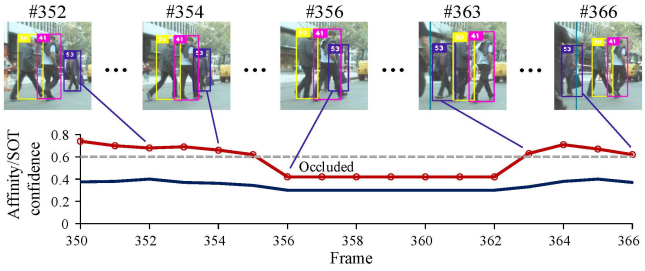

Figure 6: **Illustration of the occlusion handling.** The red line denotes the affinity produced by our UMA model, while the blue line signifies the confidence of the Siamese SOT. Our proposed model is more robust in detecting and addressing the occlusions.

set $\mathcal{T}$ of occluded targets, produced from the previous stage, we build an affinity matrix $C \in \mathbb{R}^{|\mathcal{D}|\times|\mathcal{T}|}$ to obtain the optimal assignment. More specifically, for a tracklet $T \in \mathcal{T}$, we uniformly sample K samples from T, *i.e.*, $\{t_1, t_2, ..., t_K\}$. Then the affinity between T and a candidate detection $d \in \mathcal{D}$ is calculated by:

$$c' = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{w}_d^{\top}\boldsymbol{w}_{t_k}. \tag{11}$$

After computing all the affinity, we construct the cost matrix C (affinity matrix) and obtain the optimal assignment by applying the Hungarian algorithm [40] over C. According to the assignment result, a candidate detection is assigned the identity of an occluded target that links to it. If a candidate detection does not link to any occluded targets, we view it as an entirely new target and assign it a new identity.

Trajectory Management: For trajectory initialization, we adopt method in [35] to alleviate the influence caused by FP detections. Besides, a target will be terminated if it moves out of the view or keeps occluded for over certain frames.

4. Experiments

Datasets: We evaluate our approach on the MOT16 and MOT17 datasets from MOT Challenge [38], which is a standardized benchmark focusing on multiple people tracking. MOT16 dataset contains 14 video sequences (7 for training and 7 for testing) from unconstrained environments filmed with both static and moving cameras. It provides ground-truth annotations for the training set and detections [18] for both sets. MOT17 contains more video sequences than MOT16, and provides accurate annotations and richer detections from different detectors, *i.e.*, DPM [18], SDP [64] and FRCNN [43]. For the evaluation of the two test sets, the results are submitted to the server of the benchmarks.

Evaluation Metrics: For quantitative performance evaluation, we adopt the widely used CLEAR MOT metrics *et al.* [3], *i.e.*, the multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP), false positives (FP), false negatives (FN), identity switches (IDS) and IDF1 score. In addition, the metrics defined in [33] are also used, including the percentage of mostly tracked targets (MT) and

Mode	Method	Publication	Year	MOTA↑	IDF1↑	MOTP ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Hz ↑
Online	STAM [11]	ICCV	2017	46.0	50.0	74.9	14.60%	43.60%	6,895	91,117	**473**	0.2
	AMIR [44]	ICCV	2017	47.2	46.3	75.8	14.00%	41.60%	**2,681**	92,856	774	1.0
	DMAN [68]	ECCV	2018	46.1	**54.8**	73.8	17.40%	42.70%	7,909	89,874	532	0.3
	C-DRL [42]	ECCV	2018	47.3	-	74.6	17.40%	39.90%	6,375	88,543	-	1.0
	KCF16 [9]	WACV	2019	48.8	47.2	75.7	15.80%	38.10%	5,875	86,567	906	0.1
	Tracktor++ [2]	ICCV	2019	**54.4**	52.5	**78.2**	**19.00%**	36.90%	3,280	**79,149**	682	2.0
	UMA (ours)	CVPR	2020	50.5	52.8	74.1	17.80%	**33.70%**	7,587	81,924	685	**5.0**
Offline	QuadMOT [47]	CVPR	2017	44.1	38.3	76.4	14.60%	44.90%	**6,388**	94,775	745	1.8
	FWT [23]	CVPRW	2018	47.8	47.8	**77.0**	19.10%	38.20%	8,886	85,487	852	0.2
	MHTBLSTM [29]	ECCV	2018	42.1	47.8	75.9	14.90%	44.40%	11,637	93,172	753	1.8
	JCC [28]	TPAMI	2018	47.1	52.3	-	**20.40%**	46.90%	6,703	89,368	**370**	1.8
	TLMHT [45]	TCSVT	2018	**48.7**	**55.3**	76.4	15.70%	44.50%	6,632	86,504	413	**4.8**
	LNUH [60]	AAAI	2019	47.5	43.6	-	19.40%	**36.90%**	13,002	**81,762**	1,035	0.8

Table 1: **Quantitative results on MOT16.** The best scores of online and offline MOT methods are marked in **red** and **blue**, respectively.

the percentage of mostly lost targets (ML). MT refers to the ratio of ground-truth trajectories that are covered by any track hypothesis for at least 80% of their respective life span. ML is computed as the ratio of ground-truth trajectories that are covered by any track hypothesis for at most 20% of their respective life span.

Implementation Details: We adopt the sequences in MOT17 for training. The exemplar-positive instance pair $<x_i, z_i>$ is composed of image patches from the same targets in various frames. Patches from different targets are then chosen as the negative instances. During training, the sizes of the exemplar and instance are set as 127×127 and 239×239, respectively. The AlexNet pre-trained model on ImageNet dataset [15] is used to initiate the shared part of our UMA model, while other layers are initialized through He initialization [21]. We use the learning rate configuration in [4]. The coefficient parameters in Eq. 6 are set as $\lambda_1 = \lambda_2 = 0.1$. The total loss is minimized through momentum optimization [50] with a mini-batch of size 8. The thresholds α and β used for detecting occlusions are set to 0.6 and 0.5, respectively. The threshold γ is set to 0.5, which decides whether a detection is selected as candidates for data association. We empirically set the threshold for terminating an occluded target as 30 frames.

4.1. Performance on MOT Benchmark Datasets

Quantitative and Qualitative Performance: We evaluate our approach on the test sets of MOT16 and MOT17 benchmarks. The performance of our algorithm and other recent MOT algorithms are presented in Table 1 and Table 2, where our lightweight UMA model outperforms most online and even offline MOT algorithms according to the MOTA and IDF1 metrics. For instance, as shown in Table 1, we improve 1.7% in MOTA, 5.6% in IDF1 compared with KCF16 [9], which is an online algorithm taking KCF [22] as the motion model. Results from Table 2 give another powerful support to the performance of our approach, on which we simultaneously achieve the better

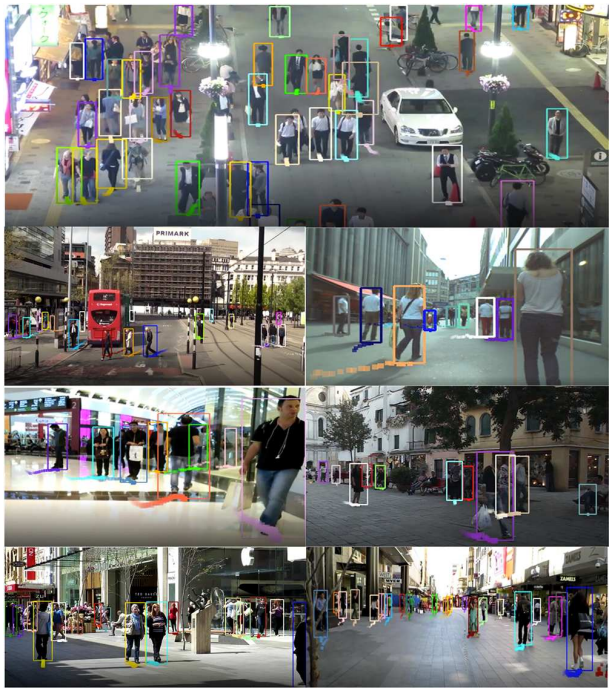

Figure 7: **Qualitative tracking results on the test sequences of MOT17 benchmark**. The color of each bounding box indicates the target identity. The dotted line under each bounding box denotes the recent tracklet of each target.

MOTA, MT, ML and FN against most published online and offline methods. In particular, FAMNet [10] is a recent work that also applies the Siamese SOT, and we surpass it in terms of both the MOTA and IDF1. Additionally, we improve Tracktor++ [2] by 2.1% according to the IDF1 metric, which validates the effectiveness of our unified model in dealing with occlusions and distractors. In a nutshell, our lightweight UMA model achieves state-of-the-art performance, benefiting from the multi-task learning framework. Qualitative results of each sequence on the MOT17 test set are illustrated in Fig 7.

Tracking Speed and Model Size: Our online MOT

Mode	Method	Publication	Year	MOTA↑	IDF1↑	MOTP ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Hz ↑
Online	DMAN [68]	ECCV	2018	48.2	**55.7**	75.5	19.30%	38.30%	26,218	263,608	2,194	0.3
	MTDF [19]	TMM	2019	49.6	45.2	74.5	18.90%	33.10%	37,124	241,768	5,567	1.2
	FAMNet [10]	ICCV	2019	52.0	48.7	76.5	19.10%	33.40%	14,138	253,616	3,072	0.6
	Tracktor++ [2]	ICCV	2019	**53.5**	52.3	**78.0**	19.50%	36.60%	**12,201**	248,047	**2,072**	2.0
	UMA (ours)	CVPR	2020	53.1	54.4	75.5	**21.50%**	**31.80%**	22,893	**239,534**	2,251	**5.0**
Offline	EDMT [6]	CVPRW	2017	50.0	51.3	77.3	**21.60%**	36.30%	32,279	**247,297**	2,264	0.6
	FWT [23]	CVPRW	2018	**51.3**	47.6	77.0	21.40%	**35.20%**	24,101	247,921	2,648	0.2
	MOTBLSTM [29]	ECCV	2018	47.5	51.9	77.5	18.20%	41.70%	25,981	268,042	2,069	1.9
	TLMHT [45]	TCSVT	2018	50.6	56.5	**77.6**	17.60%	43.40%	**22,213**	255,030	**1,407**	2.6
	JCC [28]	TPAMI	2018	51.2	54.5	-	20.90%	37.00%	25,937	247,822	1,802	1.8
	SAS [37]	CVPR	2019	44.2	**57.2**	76.4	16.10%	44.30%	29,473	283,611	1,529	**4.8**

Table 2: **Quantitative results on MOT17.** The best scores of online and offline MOT methods are marked in **red** and **blue**, respectively.

Aspect	Module	validation set: {MOT17-09, MOT17-10}						
		MOTA↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓
Full model	UMA *w.* TSA Loss: L_{SOT} (Eq. 2) + $L_{\text{N-pair}}$ (Eq. 4) + L_{Iden} (Eq. 5)	**53.0**	**61.9**	26.51%	20.48%	969	**7,236**	**56**
Loss (*w/o.* TSA)	L_{SOT} (Eq. 2) + L_{Tri} (Eq. 3)	51.7	50.9	24.10%	**19.28%**	922	7,483	87
	L_{SOT} (Eq. 2) + $L_{\text{N-pair}}$ (Eq. 4)	52.3	53.1	25.30%	20.48%	**851**	7,456	73
	L_{SOT} (Eq. 2) + $L_{\text{N-pair}}$ (Eq. 4) + L_{Iden} (Eq. 5)	52.4	58.6	25.30%	20.48%	853	7,458	58
Structure	SOT *only*	50.4	48.1	**27.71%**	24.10%	1,189	7,675	138

Table 3: **Ablation studies** on two validation sequences of MOT17.

pipeline operates at a speed of around 5.0 fps on the test sequences of MOT17 with a 1080TI GPU, which is more efficient than most previous work, without losing the accuracy. Regarding the model size, for [2] and [68], it is 270M and 300M, respectively. In contrast, the whole size of our UMA is only around 30M, which is more suitable in source-constrained environments.

4.2. Ablation Studies

In order to support the effectiveness of the proposed model, we conduct ablation studies on two sequences of the MOT17 training set, *i.e.*, {MOT17-09, MOT17-10}, and use other sequences for training.

As shown in Table 3, compared with the basic SOT (last row), our full model achieves significantly better MOTA. This verifies the effectiveness of our whole framework (§3.3). As our main motivation is to jointly learn the object motion prediction and affinity measure tasks, we next validate the ability of our model with multi-task learning (*w/o.* the TSA module). We can observe that, compared with the framework trained only with the SOT, our method with the full loss (the next-to-last row) improves 13.8%, 2.3% and 59.4% in terms of the IDF1, MOTA and IDS metrics, respectively. This demonstrates the significance of the integrated metric learning in addressing occlusions. Third, we compare different combinations of losses within our UMA to validate the discriminative ability of the learnt features embedding. Considering results from the 2^{nd} to 4^{th} rows, we can find that jointly applying the N-pair loss and identification loss gives the best results according to the IDF1 and IDS metrics. Finally, we observe that the full model with the TSA module produces the best results, which further improves performance in terms of the MOTA, IDF1 and FN metrics. This indicates that the original shared features are less compatible with the tasks, while the TSA module effectively stresses the task-aware context.

5. Conclusions

This work proposed a novel online MOT model, named UMA, aiming to perform the object motion prediction and affinity measure tasks in a unified network via multi-task learning. This is achieved by integrating the tracking loss and the metric learning losses into a triplet network during the training stage. The learnt feature not only enables the model to effectively measure the affinity in the data association phase, but also helps the SOT to distinguish the distractors during the tracklet generation phase. Compared with previous SOT based MOT approaches that train separate networks for the motion and affinity models, our method provides new insights by effectively improving the computation efficiency and simplifying the training procedure. Additionally, we extended our model with a TSA module to boost task-specific feature learning by emphasizing on different feature context. Extensive experimental results on the MOT16 and MOT17 benchmarks demonstrated the effectiveness of our lightweight model, which achieves competitive performance against many state-of-the-art approaches.

Acknowledgements This work was sponsored by Zhejiang Lab's Open Fund (No. 2019KD0AB04), Zhejiang Lab's International Talent Fund for Young Professionals, CCF-Tencent Open Fund and ARO grant W911NF-18-1-0296.

References

[1] Seung-Hwan Bae and Kuk-Jin Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, 2014. 1, 2

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019. 2, 3, 7, 8

[3] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008. 6

[4] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016. 2, 3, 4, 6, 7

[5] Visesh Chari, Simon Lacoste-Julien, Ivan Laptev, and Josef Sivic. On pairwise costs for network flow multi-object tracking. In *CVPR*, 2015. 6

[6] Jiahui Chen, Hao Sheng, Yang Zhang, and Zhang Xiong. Enhancing detection model for multiple hypothesis tracking. In *CVPR*, 2017. 8

[7] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, 2016. 5

[8] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 2

[9] Peng Chu, Heng Fan, Chiu C Tan, and Haibin Ling. Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In *WACV*, 2019. 1, 2, 4, 7

[10] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *ICCV*, 2019. 1, 2, 7, 8

[11] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *ICCV*, 2017. 1, 2, 7

[12] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017. 2

[13] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *CVPR*, 2015. 2, 6

[14] Afshin Dehghan, Yicong Tian, Philip HS Torr, and Mubarak Shah. Target identity-aware network flow for online multiple target tracking. In *CVPR*, 2015. 2

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 7

[16] Xingping Dong, Jianbing Shen, Wenguan Wang, Ling Shao, Haibin Ling, and Fatih Porikli. Dynamical hyperparameter optimization via deep reinforcement learning in tracking. *TPAMI*, 2019. 2

[17] Xingping Dong, Jianbing Shen, Dajiang Yu, Wenguan Wang, Jianhong Liu, and Hua Huang. Occlusion-aware real-time object tracking. *TMM*, 19(4):763–771, 2016. 2

[18] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010. 6

[19] Zeyu Fu, Federico Angelini, Jonathon Chambers, and Syed Mohsen Naqvi. Multi-level cooperative fusion of gm-phd filters for online multiple human tracking. *TMM*, 21(9):2277–2291, 2019. 8

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 4, 6

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 7

[22] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2014. 2, 7

[23] Roberto Henschel, Laura Leal-Taixe, Daniel Cremers, and Bodo Rosenhahn. Fusion of head and full-body detectors for multi-object tracking. In *CVPRW*, 2018. 7, 8

[24] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 3, 4

[25] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. Online multi-object tracking via structural constraint event aggregation. In *CVPR*, 2016. 1, 2

[26] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 5

[27] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas, et al. Tracking-learning-detection. *TPAMI*, 34(7):1409, 2012. 2

[28] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *TPAMI*, 42(1):140–153, 2018. 7, 8

[29] Chanho Kim, Fuxin Li, and James M Rehg. Multi-object tracking with neural gating using bilinear lstm. In *ECCV*, 2018. 1, 7, 8

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 3

[31] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. In *CVPRW*, 2016. 2

[32] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015. 1

[33] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, 2009. 6

[34] Si Liu, Tianzhu Zhang, Xiaochun Cao, and Changsheng Xu. Structural correlation filter for robust visual tracking. In *CVPR*, 2016. 2

[35] Yuanpei Liu, Junbo Yin, Dajiang Yu, Sanyuan Zhao, and Jianbing Shen. Multiple people tracking with articulation detection and stitching strategy. *Neurocomputing*, 2019. 6

[36] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019. 2

[37] Andrii Maksai and Pascal Fua. Eliminating exposure bias and loss-evaluation mismatch in multiple object tracking. In *CVPR*, 2019. 8

[38] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and

Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 1, 6

[39] Anton Milan, Seyed Hamid Rezatofighi, Anthony R Dick, Ian D Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, 2016. 1, 2

[40] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 6

[41] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 2

[42] Liangliang Ren, Jiwen Lu, Zifeng Wang, Qi Tian, and Jie Zhou. Collaborative deep reinforcement learning for multi-object tracking. In *ECCV*, 2018. 7

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 6

[44] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *ICCV*, 2017. 1, 2, 6, 7

[45] Hao Sheng, Jiahui Chen, Yang Zhang, Wei Ke, Zhang Xiong, and Jingyi Yu. Iterative multiple hypothesis tracking with tracklet-level association. *TPAMI*, 29(12):3660–3672, 2018. 7, 8

[46] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016. 4, 5

[47] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *CVPR*, 2017. 2, 7

[48] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *CVPR*, 2019. 1

[49] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *ICCV*, 2017. 5

[50] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013. 7

[51] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Multi-person tracking by multicut and deep matching. In *ECCV*, 2016. 2

[52] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*, 2017. 1, 2, 6

[53] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019. 2

[54] Bing Wang, Li Wang, Bing Shuai, Zhen Zuo, Ting Liu, Kap Luk Chan, and Gang Wang. Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. In *CVPRW*, 2016. 2

[55] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 2

[56] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *TPAMI*, 41(4):985–998, 2018. 1

[57] Wenguan Wang, Zhijie Zhang, Siyuan Qi, Jianbing Shen, Yanwei Pang, and Ling Shao. Learning compositional neural information fusion for human parsing. In *ICCV*, 2019. 1

[58] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 3

[59] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *NeurIPS*, 2006. 4

[60] Longyin Wen, Dawei Du, Shengkun Li, Xiao Bian, and Siwei Lyu. Learning non-uniform hypergraph for multi-object tracking. In *AAAI*, 2019. 7

[61] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 1

[62] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *ICCV*, 2015. 2, 6

[63] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *ICCV*, 2019. 2, 3

[64] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*, 2016. 6

[65] Ming Yang, Ying Wu, and Gang Hua. Context-aware visual tracking. *TPAMI*, 31(7):1195–1209, 2009. 5

[66] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*. 2012. 2

[67] Li Zhang, Yuan Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 2

[68] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *ECCV*, 2018. 1, 2, 6, 7, 8