Learning Discriminative Motion Models for Multiple Object Tracking

Yi-Fan Li, Hong-Bing Ji, *Senior Member, IEEE,* Wen-Bo Zhang, and Yu-Kun Lai, *Member, IEEE*

***Abstract*—Motion models are vital for solving multiple object tracking (MOT), which makes instance-level position predictions of targets to handle occlusions and noisy detections. Recent methods have proposed the use of Single Object Tracking (SOT) techniques to build motion models and unify the SOT tracker with the object detector into a single network for high-efficiency MOT. However, three feature incompatibility issues in the required features of this paradigm are ignored, leading to inferior performance. First, the object detector requires class-specific features to localize objects of pre-defined classes. Contrarily, target-specific features are required in SOT to track the target of interest with an unknown category. Second, MOT relies on intra-class differences to associate targets of the same identity (ID). On the other hand, the SOT trackers focus on inter-class differences to distinguish the tracking target from the background. Third, classification confidence is used to determine the existence of targets, which is obtained with category-related features and cannot accurately reveal the existence of targets in tracking scenes. To address these issues, we propose a novel Task-specific Feature Encoding Network (TFEN) to extract task-driven features for different sub-networks. Besides, we propose a novel Quadruplet State Sampling (QSS) strategy to form the training samples of the motion model and guide the SOT trackers to capture identity-discriminative features in position predictions. Finally, we propose an Existence Aware Tracking (EAT) algorithm by estimating the existence confidence of targets and re-considering low-scored predictions to recover missed targets. Experimental results indicate that the proposed Discriminative Motion Model-based tracker (DMMTracker) can effectively address these issues when employing SOT trackers as motion models, leading to highly competitive results on MOT benchmarks.**

***Index Terms*—Multiple object tracking, Motion models, Single object tracking.**

I. Introduction

MULTIPLE object tracking (MOT) is the task of estimating the locations of objects of interest on each frame and temporally linking identical objects with consistent identities (IDs) to form trajectories. As a fundamental task in computer vision, MOT has a wide range of applications, including autonomous driving, video surveillance, and human-computer interaction. Although significant improvements have been achieved with the help of advanced object detection techniques, MOT is still challenging in crowded scenes with occlusions, distractors, and camera motion.

This work was supported by the National Natural Science Foundation of China (No. 62276204). *(Corresponding author: Hong-Bing Ji.)*

Y.-F. Li, H.-B. Ji and W.-B Zhang are with the School of Electronic Engineering, Xidian University, Xi'an, China, 710071. (E-mail: leon63963@163.com; hbji@xidian.edu.cn; wbzhang@xidian.edu.cn.)

Y.-K. Lai is with the School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, U.K. (E-mail: LaiY4@cardiff.ac.uk)

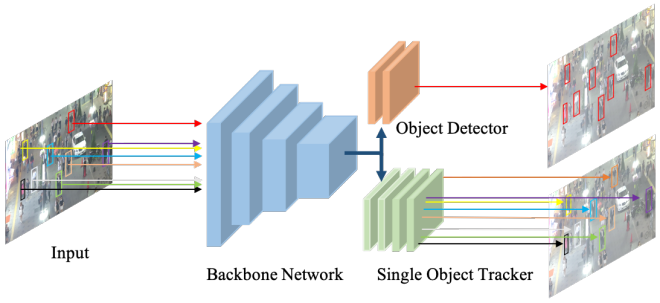

Fig. 1. Illustration of the unified framework that combines the object detector and SOT tracker into a single network with multi-task learning. The object detections and SOT-based position predictions are performed simultaneously for all targets by sharing backbone features.

Most previous methods in MOT adopt the Tracking-by-Detection (TBD) paradigm [1]–[3], in which off-the-shelf detectors provide framewise detections. Then, the MOT is treated as a data association problem that links the detected objects temporally with existing tracks. The Joint-Detection-and-Tracking (JDT) [4]–[6] framework is proposed to eliminate the separation issue in TBD by integrating the object detector and the tracking-related networks into a unified network. Appearance features [1], [3], [7] are widely adopted for data association. However, these methods are less reliable under occlusions and camera motions since the discrimination and consistency of features are impaired in these scenarios.

Motion models are widely adopted that provide instance-level position predictions to compensate for noisy detections, recover missing targets, and provide motion features where reliable appearance features are absent. Among different motion models, linear motion models [1], [5], displacement regression networks [8], [9], and complex learnable networks [10], [11] are commonly used for position predictions. However, linear models and displacement regression networks are insufficient for modeling complex motion patterns and lack identity information, and learnable networks are heavily dependent on training data. Thus, these motion models are vulnerable to occlusions and distractors, degrading the tracking performance in crowded scenes.

Single Object Tracking (SOT) techniques are employed to build motion models to predict target positions and provide tracking candidates in recent methods [12]–[17]. SOT models are more discriminative and face similar challenges to MOT, including occlusions, distractors, target drift, and motion blur. Unlike MOT, a specific object of interest with an unknown category is tracked in SOT [18]–[25]. It is intuitive to think that MOT can be solved by applying SOT multiple times on

a per-object basis for each frame. However, the computational cost dramatically increases in crowded scenes since the visual features of targets are extracted separately and repetitively. Moreover, SOT-based predictions are prone to drift in MOT scenes where intra-class occlusions and distractors are commonly observed. To improve the tracking accuracy and reduce the computational burden, as shown in Fig. 1, some methods integrate the SOT techniques and the object detectors into a single network [14]–[17] to avoid repetitive feature extraction and accelerate the tracking speed. By doing so, all targets are tracked in a single forward propagation with shared backbone features, significantly reducing the computational complexity and inference time.

Despite the effectiveness, three feature incompatibility issues are ignored when employing SOT trackers as motion models and unifying them with the detectors in a single network. The first is the incompatibility of the required features between the object detection and the SOT. The detector is trained to localize objects of pre-defined classes, which requires *class-specific* features. In contrast, the SOT tracker relies on *target-specific* features since only the target of interest is tracked. Therefore, this difference will lead to incompatibility in required features between the detector and the SOT tracker, resulting in an inferior performance for both tasks. The second incompatibility is the required features between SOT and MOT. Only the specific target of interest with an unknown class is tracked in SOT. In contrast, an arbitrary number of targets of interest categories are tracked in MOT. Therefore, SOT trackers are sensitive to *inter-class* differences and learn to discriminate tracking targets from backgrounds, while MOT trackers concentrate on *intra-class* differences and capture identity-discriminative features to temporally associate the same target. This difference will confuse the SOT-based motion models and degrade the accuracy of position predictions in MOT scenarios. The third incompatibility is the required features between classification and the existence of targets. The classification confidence is used to determine the existence of targets while tracking. However, it is obtained with category-related features instead of existence cognition, only revealing only the *probability of category*. Thus, the classification confidence cannot accurately reflect the *existence of targets*, and classification-based judgment will lead to missing targets and track fragmentation.

To address the above issues, we propose a novel Discriminative Motion Model (DMM) by building an enhanced region-based Siamese tracker as the motion model. We unify the proposed motion model with a region-based detector [26] to form the DMMTracker. In detail, we propose a Task-specific Feature Encoding Network (TFEN) to decouple shared backbone features and encode task-driven features for different tasks. Class-specific and target-specific features are captured respectively for the detector and SOT tracker with TFEN, alleviating the incompatibility between the two tasks. Then, we propose a novel Quadruple State Sampling (QSS) strategy to form training samples of the motion model by incorporating four MOT tracking states while training the SOT tracker. The QSS enables the SOT tracker to be sensitive to the intra-class differences and extract identity-discriminative features to distinguish distractors of the same class, thus enhancing the identity consistency of predicted tracks and mitigating the differences in required features between SOT and MOT. Finally, to deal with the feature incompatibility between the classification and the existence, we augment the SOT tracker by estimating the existence confidence of the targets. Then, we propose an Existence Aware Tracking (EAT) algorithm for identity management, which enables the missed targets to be recovered, increasing the accuracy of existence judgments. Incorporating all the proposed modules, the resulting DMMTracker significantly improves the SOT-based methods and achieves highly competitive results in MOT benchmarks.

The contributions of our method are summarized as follows:

- We propose a novel Task-specific Feature Encoding Network (TFEN) to alleviate the incompatibility in required features between the detection and SOT by extracting class-specific features and target-specific features on top of the shared backbone features.
- We propose a new Quadruplet State Sampling (QSS) strategy to eliminate the difference in required features between SOT and MOT by sampling training triplets of the SOT trackers, which enables the motion models to be sensitive to the intra-class differences and capture identity-discriminative features in position predictions.
- We propose to enhance the SOT tracker by estimating target existence confidence and associating targets with an Existence Aware Tracking (EAT) algorithm, which can recover missed targets and bridge the gap between the classification and the existence of targets.

Extensive ablation experiments and benchmark datasets results demonstrate the effectiveness of our proposed DMMTracker. The remainder of this paper is organized as follows: Sec. II reviews related works. Sec. III illustrates the proposed method. The experiments and discussion are presented in Sec. IV. Finally, Sec. V concludes this paper.

II. Related Work

Motion models are commonly employed in MOT for target position predictions, which can compensate for occlusions and noisy detections and provide geometric information and candidate bounding boxes. This section reviews motion models used in deep learning-based MOT methods, including linear models, regression-based models, learnable network-based models, and SOT-based models.

A. Linear Motion Models

Appearance embeddings are discriminative in identifying targets and are widely utilized as representative features for data association. Most appearance-based methods apply a simple linear motion model to compensate for contaminated appearance embeddings under the assumption that objects move linearly. The Kalman Filter (KF) is widely used in these methods by treating detections as observations, and it provides geometrical constraints for similarity measurements, such as in the two-stage methods DeepSORT [1] and ByteTrack [3] and in the one-shot methods JDE [4] and FairMOT [5].

An improved KF-based motion model is proposed in OC-SORT [27] to handle the accumulation of errors caused by traditional linear state estimation. A simpler model with a constant velocity assumption is used in TMOH [28].

Despite their simplicity, real-world motions are much more complicated and are difficult to predict. Thus, linear motion models cannot handle long-term occlusions and fast camera motion, resulting in missing targets, target drifts, and ID switches. In contrast, our DMMTracker takes advantage of advanced SOT techniques, which can model complex motion patterns and accurately predict target positions by capturing discriminative visual features.

B. Regression-based Motion Models

Some motion models make position predictions by regressing displacements of bounding boxes and center points. The regression-based method Tracktor [8] is developed from Faster R-CNN [26] by reusing its regression head, and the target locations are obtained by regressing the displacements of the bounding box between consecutive frames. Attention networks are proposed in TADAM [29] to enhance the regressor by guiding the network to focus more on targets and less on distractors. Similarly, CenterTrack [9] regresses the offsets of the target centers between adjacent frames and links each updated center to the nearest target to assign identities.

However, displacement regressions are predicted based on regional visual features, which are unaware of the identities of targets and predict in an identity-agnostic way. Thus, these methods suffer heavily from intra-class occlusion and distractors of similar appearance, yielding ID switches and missing targets. On the contrary, our DMMTracker can capture identity-discriminative features and predict positions in an identity-aware fashion. Moreover, DMMTracker can handle noisy detections and missing targets by prioritizing SOT predictions in forming trajectories.

C. Learnable Network-based Motion Models

Some methods construct complex deep networks to build learnable motion models to learn motion dynamics and predict positions. The sequence learning and memory ability of RNN (Recurrent Neural Networks) enable it to be a motion model. The RNN-based probabilistic autoregressive motion model is proposed in [10] by learning the multi-modal distribution of natural tracks. The RNN-based trajectory estimator is introduced in [30] as the motion model based on mixture density networks. In addition, learnable short-term and long-term motion models are proposed in MotionTrack [11] to associate trajectories from short to long ranges. A Transformer-based motion model is proposed in [31] by considering the motion of the target and the camera simultaneously. Moreover, learnable motion predictors are also used in vehicle tracking [32], [33].

However, the parameters of learnable motion models are tuned on the assumptions of specific distribution and built with complex networks, thus making it difficult to generalize to new scenes and other methods. Conversely, our DMMTracker learns a matching function with pixel-level supervision on top of visual features without motion priors. Therefore, it can accurately localize targets in new scenes, and more advanced SOT techniques can be implemented for further enhancement.

D. SOT-based Motion Models

The computational costs of directly applying SOT in MOT dramatically increase when there are a large number of targets and repetitive feature extractions. Therefore, the unified network paradigm is proposed by adapting the SOT trackers to sub-networks to facilitate tracking efficiency and reduce training complexity. STAM [12] and DMAN [34] assign each target a SOT tracker for position estimation to generate tracklets and handle occlusions with spatial-temporal attention networks. End-to-end trainable methods that unify the SOT trackers and data association modules have been proposed in recent works. For example, the SOT tracker is integrated with data association networks by cross-correlation operations in FAMNet [13] to recover missed targets. DASOT [14] integrates the SOT tracker and data association network into a single network by sharing backbone features. Similarly, the SOT tracker and metric learning module are integrated into a unified triplet network in UMA [15].

Unlike previous methods, recent trackers prioritize SOT predictions in extending trajectories. SOTMOT [17] proposes to extend CenterNet [35] by adding a SOT branch to benefit from the discriminative power of the SOT, and trajectories are formed based on SOT predictions. The region-based SOT tracker [18] is adopted and combined with Faster R-CNN to form the region-based MOT tracker SiamMOT [16]. Object detection and position predictions are performed simultaneously in these trackers, and trajectory extension and identity assignments are performed based on SOT predictions. DMMTracker is built on top of SiamMOT, and we enhance the tracker by solving the incompatibility issues that arise when employing SOT techniques as motion models.

III. Methodology

In this section, we first review the preliminaries of the Siamese SOT tracker in Sec. III-A and provide an overview of our method in Sec. III-B. Afterward, Sec. III-C introduces the proposed Task-specific Feature Encoding Network (TFEN). Sec. III-D presents the Quadruplet State Sampling (QSS) strategy. Sec. III-E describes the Existence Aware Tracking (EAT) algorithms. Finally, we illustrate the details of training and inference in Sec. III-F.

A. Preliminaries

The region-based Siamese SOT tracker [18], [19] is employed as the motion model in our DMMTracker. This method transfers the SOT into a template matching problem that finds the most similar patch in the search region, and it shows superior performance and high efficiency in SOT. Specifically, a Siamese network is utilized to extract features $\varphi(Z)$ for template Z and $\varphi(X)$ for search region X. The template patch is the bounding box of the existing track, and the search region is obtained by expanding the template patch by a predefined factor. A channel-wise correlation layer is employed to

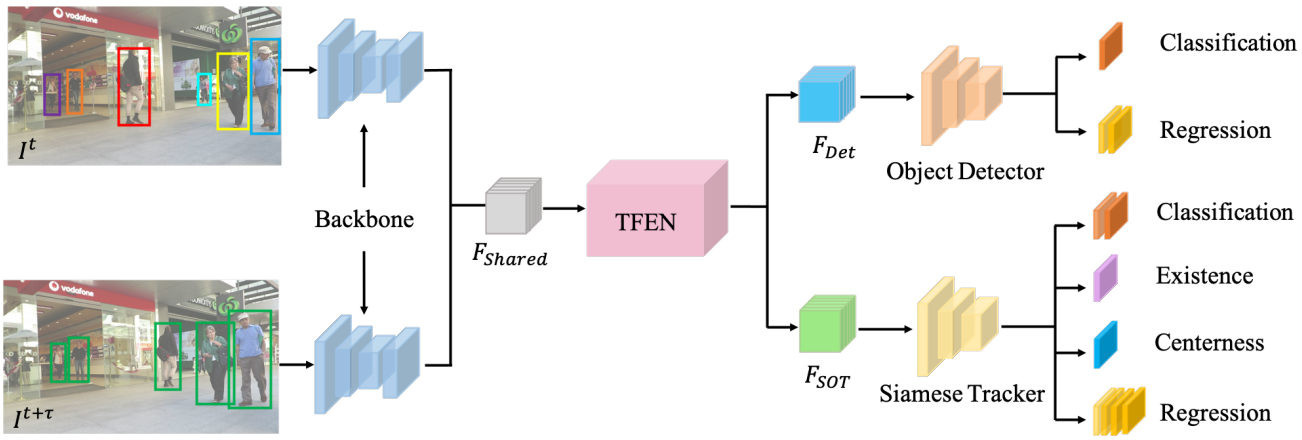

Fig. 2. Overview of the proposed tracker DMMTracker. The existing tracks on the frame I^t and detections of the frame $I^{t+\tau}$ are used as input. The backbone network extracts shared features F_{shared}. The proposed Target-specific Feature Encoding Network (TFEN) encodes task-specific features F_{Det} and F_{SOT}, which are used for the object detector and SOT tracker, respectively. The Siamese tracker is trained with training triplets sampled by the proposed Quadruplet State Sampling (QSS) strategy. The target existence is estimated by building an existence prediction network on the Siamese tracker. The final MOT results are obtained by merging the SOT predictions and the object detections with the proposed Existence Aware Tracking (EAT) algorithm.

produce a pixel-level response map f, which contains massive amounts of semantic information about instance-level motion and is calculated based on $\varphi(Z)$ and $\varphi(X)$ as follows:

$$f(Z, X) = \varphi(Z) * \varphi(X), \tag{1}$$

where $*$ represents the channel-wise cross-correlation operation. Based on this dense response map, three branches are built with convolutional layers: a classification branch, a regression branch, and a centerness branch. The classification branch estimates the foreground-background probabilities. The regression branch predicts the locations of the bounding boxes. The centerness branch determines the closeness of each location relative to the center of the target. The centerness predictions help to remove outliers since the locations far away from the center are prone to producing low-quality predictions. Moreover, cosine window and scale change penalty are employed [18] to discourage dramatic changes in position and shape to promote smooth trajectories.

The Siamese tracker is built based on the Region Proposal Network (RPN) [26] and transfers the SOT as a one-shot detection task. Hence, the Siamese tracker can be adapted as a sub-network and unified with the region-based object detector [26] to form an MOT tracker [16], where shared backbone features are shared and fed into the object detector and the SOT tracker simultaneously. More specifically, given the existing tracks of the frame I^t and detections of the frame $I^{t+\tau}$ with random τ frame gaps, each existing target is assigned a SOT tracker and is used as the template patch to find the most similar target patch in the frame $I^{t+\tau}$ by the assigned SOT tracker. All targets are predicted simultaneously in one forward propagation, significantly reducing the computational complexity and running time. The identities of tracks are transferred with the position prediction process, eliminating the necessity of complex data association methods. Therefore, this tracking paradigm solves MOT by multiple template matching, and the final MOT results are obtained by merging SOT-based predictions with object detections.

B. Overview

The proposed DMMTracker is built based on the region-based method SiamMOT by enhancing its motion models. The overview of DMMTracker is shown in Fig. 2, where a region-based Siamese tracker is employed as the motion model for position predictions and is unified with a region-based object detector [26] in a single network.

To alleviate the incompatibility regarding the required features between the SOT and detection, we propose a Target-specific Feature Encoding Network (TFEN), which is used to encode task-driven features for each sub-task, i.e., class-specific features for the object detector and target-specific features for the Siamese tracker. In addition, the training samples of the Siamese tracker are constructed with the proposed Quadruplet State Sampling (QSS) strategy, which aims to guide the Siamese tracker to be sensitive to intra-class differences and capture identity-discriminative features in template matching, eliminating the issue of feature incompatibility between SOT and MOT. Moreover, as shown in Fig. 2, the existence confidences of targets are estimated by building a prediction network on the Siamese tracker, which can capture existence-related features and recover the missed targets that are wrongly suppressed in traditional classification-based identity management. The final trajectories are obtained with the proposed Existence Aware Tracking (EAT) algorithm.

C. Task-specific Feature Encoding Network

The superior performance demonstrates the effectiveness of integrating detectors and SOT trackers into a unified network. This one-shot framework can balance the accuracy and speed of the tracker. However, the difference in required features between the detection and SOT methods is deleterious for feature extraction, degrading the tracking performance. In detail, the object detector is trained to predict the scale and location of all targets that belong to pre-defined classes. The semantic features of these categories are needed. Therefore, class-specific features are essential for detectors. In contrast,

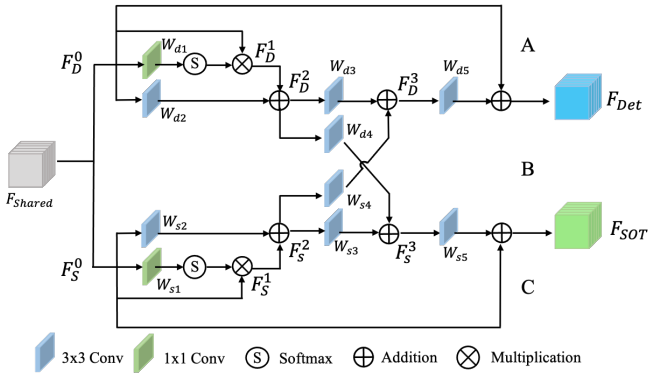

Fig. 3. Overview of the proposed Target-specific Feature Encoding Network (TFEN), which encodes class-specific features for the object detector and target-specific features for the Siamese tracker, alleviates the incompatibility regarding the required features between the detection and SOT.

the SOT trackers are trained to distinguish the specific object of an unknown class from the surrounding background. Thus, the target-specific features are imperative for SOT trackers. Since the backbone features are shared and fed into two sub-networks, neither network can obtain task-dependent features, which inevitably deteriorates the MOT performance.

To resolve the incompatibility issue between the object detector and the SOT tracker, we propose the Target-specific Feature Encoding Network (TFEN) to capture task-dependent features for the two sub-networks. The structure of the TFEN is shown in Fig. 3, which comprises three parts (A, B, and C) obtained by stacking convolutional layers. The TFEN encodes task-related features in networks A and C for different tasks, captures mutual information between two tasks in network B, and finally yields the class-specific features F_{Det} for the detector and target-specific F_{SOT} for the SOT tracker.

In detail, we first pass shared features F_{shared} through a 1x1Conv layer and Softmax, which are used as attention mechanisms to re-weight the channels of F_{shared} to obtain F_D^1 and F_S^1 since different channels of F_{shared} focus on different feature aspects. To aggregate local features, we use other 3x3Conv-GN-ReLU layers in A and C for feature extraction on top of F_{shared}. Then, we fuse them with F_D^1 and F_S^1 by element-wise addition to obtain self-related features F_D^2 and F_S^2 for different branches before mutual information is passed on. The 3x3Conv and 1x1Conv represent convolution layers with 3×3 and 1×1 kernels, respectively. The GN denotes the Group Normalization [36]. These processes are produced as follows:

$$F_D^1 = F_D^0 * Softmax(W_{d1} F_D^0), \tag{2}$$
$$F_D^2 = ReLU(GN(W_{d2} F_D^0)) + F_D^1, \tag{3}$$
$$F_S^1 = F_S^0 * Softmax(W_{s1} F_S^0), \tag{4}$$
$$F_S^2 = ReLU(GN(W_{s2} F_S^0)) + F_S^1, \tag{5}$$

where W_{d1}, W_{d2}, W_{s1}, and W_{s2} denote the learnable weights of the four projection layers shown in Fig. 3, implemented as convolution layers in A and C. The fused F_D^2 and F_S^2 contain different self-related local features from F_{shared} for the object detector and Siamese tracker, respectively.

As shown in Fig. 3, other 3x3Conv-GN-ReLU layers are employed in B to extract and pass mutual features for each sub-task, which is then fused with features F_D^2 and F_S^2 to obtain the more compact middle features F_D^3 and F_S^3. These processes are shown as follows:

$$F_D^3 = ReLU(GN(W_{d3} F_D^2)) + ReLU(GN(W_{s4} F_S^2)), \tag{6}$$
$$F_S^3 = ReLU(GN(W_{s3} F_S^2)) + ReLU(GN(W_{d4} F_D^2)), \tag{7}$$

where W_{d3}, W_{d4}, W_{s3}, and W_{s4} denote the learnable weights for the four linear projections shown in Fig. 3, implemented as convolution layers. F_D^3 and F_S^3 are high-level features incorporating massive task-driven local features enhanced by mutually beneficial features of different tasks.

Finally, the other two 3x3Conv-GN-ReLU layers are utilized for further feature aggregation on top of F_D^3 and F_S^3, and the residual links from the input features F_D^0 and F_S^0 are employed to incorporate low-level features to form the desired task-specific features F_{Det} and F_{SOT}, respectively. These processes are implemented as follows:

$$F_{Det} = ReLU(GN(W_{d5} F_D^3)) + F_D^0, \tag{8}$$
$$F_{SOT} = ReLU(GN(W_{d5} F_S^3)) + F_S^0, \tag{9}$$

where W_{d5} and W_{s5} are two learnable weight matrices, as shown in Fig. 3. The fused features F_{Det} and F_{SOT} are separately fed into the corresponding branch for further processing. The detection-required features F_{Det} capture class-specific (e.g., pedestrians for MOT) information, which is beneficial for estimating target locations and scales. Meanwhile, the SOT branch can extract target-specific features for each assigned target, enhancing the accuracy of template matching and boosting the identity consistency of tracks. Therefore, the proposed TFEN can relieve the incompatibility in features between the detection and SOT.

Although the proposed TFEN is built in a symmetric structure, the outputs F_{Det} and F_{SOT} are used as inputs for different tasks and are supervised by different loss functions. Thus, task-driven parameters are learned independently from different training samples for different branches of the TFEN. The effectiveness of TFEN is proven by the enhanced object detection and position prediction performance in experiments.

D. Quadruplet State Sampling

The SOT technique is deployed as the motion model for position predictions, and the balance between accuracy and speed is achieved by integrating the SOT tracker and object detector into a single network with multi-task learning. Despite the effectiveness, the incompatibility of the required features between the SOT and MOT degrades the tracking performance. More specifically, the specific target of interest with an unknown category is tracked in SOT. Therefore, the SOT tracker is sensitive to inter-class differences since it learns to distinguish targets from backgrounds. On the contrary, multiple targets belonging to the same class are tracked in MOT, emphasizing intra-class differences to maintain identity consistency across frames. Hence, employing SOT trackers as motion models in MOT needs to alleviate this feature

incompatibility issue and guide the motion model to capture identity-discriminative features during position predictions.

Since network feature extraction is highly related and heavily relies on the training data [23], [37], we propose a novel Quadruplet State Sampling (QSS) strategy to construct the training sample of the motion model to address the above incompatibility issue. The QSS guides the Siamese tracker to focus more on the intra-class differences and extract identity-discriminative features through network training to better handle the challenging scenarios in MOT. Specifically, the region-based Siamese tracker learns a matching function in the training process with the dedicated sampled training triplet $\Theta_i^{t,t+\tau} = \left(R_i^t, S_i^{t+\tau}, R_i^{t+\tau}\right)$ from the selected image pairs, where R_i^t denotes the template patch, $S_i^{t+\tau}$ is the corresponding search region on frame $I^{t+\tau}$ obtained by expanding R_i^t, and $R_i^{t+\tau}$ is the tracking target of R_i^t in the frame $I^{t+\tau}$. The template patch R_i^t is provided by RPN outputs and selected with IoU-based matching following [18] while training. We also use proposals of the RPN to form the tracking targets $R_i^{t+\tau}$ via IoU-based selection, which can provide more contextual information on tracking scenes during training.

The proposed QSS provides four tracking states of the MOT by forming corresponding tracking scenarios in the training triplets. Four types of training triplets are composed of positive triplets (P), hard positive triplets (HP), negative triplets (N), and hard negative triplets (HN). In detail, for the ith training triplet $\Theta_i^{t,t+\tau}$, if the tracking target $R_i^{t+\tau}$ of the template R_i^t exists in the search region $S_i^{t+\tau}$, this makes a continuous tracking scenario, and the states of the template, search region, and tracking target are all marked as 1. Thus, the positive triplet (P) represents the continuous trajectory of the R_i^t, and we denote this positive triplet as $P = (1, 1, 1)$. The target $R_i^{t+\tau}$ is selected from the proposals of the RPN whose assigned ID is the same as the ground truth of template R_i^t.

However, suppose that the search region $S_i^{t+\tau}$ does not include the tracking target of the template R_i^t, i.e., the search region includes targets of different identities or has no valid targets. This case may be caused by occlusions or the target leaving the scene. Thus, there is no valid tracking target of the same ID as the template R_i^t in this training triplet. This scenario makes the hard positive (HP) triplet, denoted as $HP = (1, 0, 0)$. We employ a dummy bounding box (whose coordinates and ID are set to -1) as the tracking target $R_i^{t+\tau}$ to form this hard positive training triplet.

Conversely, suppose there is no valid object in the template R_i^t (i.e., dummy template), and the search region $S_i^{t+\tau}$ has not been assigned with any valid IDs. In this case, the tracking target $R_i^{t+\tau}$ is a dummy bounding box, and this scenario results in a negative triplet (N), denoted as $N = (0, 0, 0)$. However, if the search region $S_i^{t+\tau}$ contains targets with valid IDs while the template R_i^t does not, the target $R_i^{t+\tau}$ will still be a dummy bounding box. This training triplet makes the hard negative (HN) case and is marked as $HN = (0, 1, 0)$. We equally sample these four types of training triplets to construct the training samples of the SOT tracker.

Among the training triplets formed with the QSS strategy, the positive triplets can guide the SOT tracker to learn what to track, and the hard positive triplets provide examples of track terminations. The negative triplets provide the scenarios of no tracking targets, which can guide the SOT tracker to recognize the background, and the hard negative triplets help the tracker to identify distractors to reduce false predictions and ID switches. Therefore, the QSS can help the SOT tracker better recognize intra-class differences, capture identity-discriminative features to distinguish targets, and make correct decisions when encountering state transitions.

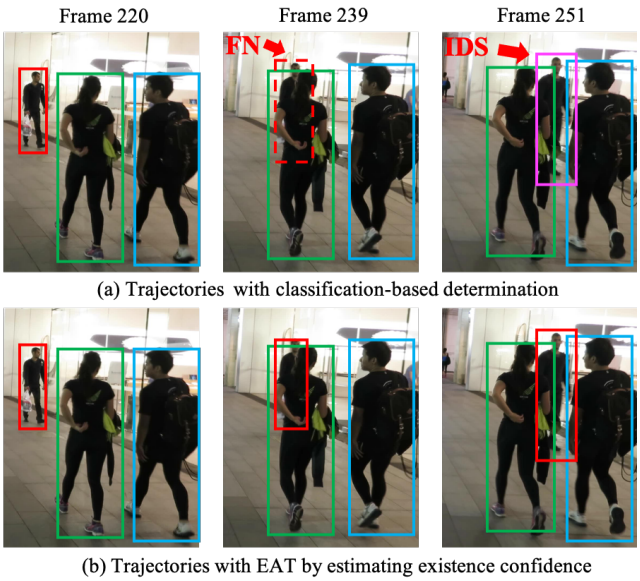

Fig. 4. Examples of the EAT algorithm. (a) shows the trajectories obtained by determining the existence of targets based on classification confidence. The correct prediction (shown in the red dashed box) is suppressed, resulting in FN and IDS. (b) shows the trajectories produced by the proposed EAT, where the occluded targets are continuously tracked with consistent ID.

E. Existence Aware Tracking

The classification confidence is widely used to suppress redundant bounding boxes in MOT. If the overlap between two bounding boxes is higher than a pre-defined threshold, they are regarded as representing the same target [38]. Those with lower classification scores are suppressed, and the surviving ones remain for identity assignments. However, the classification score reveals the confidence of an object belonging to a specific category. Thus, it cannot accurately reflect the existence of predicted targets in tracking scenes. We argue that this classification-based procedure is not optimal for determining the existence of targets since classification is estimated based on classification-related features instead of existence-related features. Some methods have attempted to mitigate this problematic decision-making process by referring to the historical information of trajectories [39] and re-considering the lower-scored detections [3]. Although reasonably effective, the difference in required features between the classification and the existence of targets remains.

As shown in Fig. 4 (a), low-scored target predictions, which are caused by contaminated features under occlusions, are not equivalent to false positives. Motivated by this, we propose to re-consider low-scored predictions to recover missed targets

and boost the accuracy of the existence judgment. We enhance the Siamese tracker by estimating the existence confidence of targets and bridging the gap between existence and classification by the Existence Aware Tracking (EAT) algorithm. To train this added existence prediction network, we first construct the instance-level existence labels for all targets by transforming the target visibility scores. The visibility scores are provided by MOT datasets, which reveal the visible proportion for each target. We assume that a target exists in tracking scenes if its visibility score is higher than 0.1, and we set the corresponding existence label $\hat{\mathcal{E}}$ to 1. Conversely, a target with visibility lower than 0.1 is regarded as leaving the scene or completely occluded, and its existence label $\hat{\mathcal{E}}$ is set to 0. Thus, the existence label $\hat{\mathcal{E}}$ is obtained as follows:

$$\hat{\mathcal{E}} = \begin{cases} 1 & visibility \geq 0.1 \\ 0 & otherwise. \end{cases} \tag{10}$$

We add a 3×3 convolutional layer and Softmax on top of the response map to estimate the instance-level existence confidence, which is built in parallel with the classification and centerness branches on the Siamese tracker, as shown in Fig. 2. The predicted existence confidence ranges from 0 to 1, showing the probability that a target exists in the tracking scenes, irrespective of its location and classification score. The existence-related features are captured in this prediction network. Thus, the existence estimation is less affected by occlusions and distractors than in the classification branch. We take the predicted existence confidence into consideration when deciding the survival of predicted targets.

We propose to tackle the incompatibility issue between existence and classification with the Existence Aware Tracking (EAT) algorithm by referring to the existence confidence in identity management. Specifically, for pixel (i, j) on the response map f, assume that the predicted classification confidence is $S^{i,j}_{cls}$ and that the centerness score is $S^{i,j}_{cen}$. The original existence score of the location (i, j) can be calculated as $S^{i,j} = S^{i,j}_{cls} \cdot S^{i,j}_{cen}$, reflecting the confidence of the location (i, j) being the center of the tracking target, and $S^{i,j}$ is used for the existence management.

If $S^{i,j}$ is higher than the threshold η_1, this prediction can form a high-scored trajectory. The corresponding predicted target is deemed to exist in the scene and added to the tracking set $\mathcal{T}_1$, where the successfully tracked targets lie. For targets whose $S^{i,j}$ is lower than η_1, instead of removing them as in previous methods, we further consider the potential of this target existing in the scene. A low-scored target is a potential true positive if its predicted existence confidence $S^{i,j}_{exi}$ is higher than a threshold η_2. The high-scored $S^{i,j}_{exi}$ indicates that this target may still exist in the scene and is partially occluded. Thus, we collected this target and added it to the set $\mathcal{T}_2$ for further consideration. The remaining tracks with low existence confidences are no longer considered.

In the proposed Existence Aware Tracking (EAT) algorithm, we first merge the predicted targets in $\mathcal{T}_1$ with the detections $\mathcal{D}$ via NMS with a threshold of 0.5 by prioritizing the SOT predictions to remove redundant bounding boxes. Then, we process the targets in $\mathcal{T}_2$ in the second round. The unmatched detections in the first association are denoted as $\mathcal{D}^1_{un}$, and we calculate the bounding box overlaps between the predicted targets in $\mathcal{T}_2$ and detections in $\mathcal{D}^1_{un}$ to obtain the spatial similarities between them. We associate two sets of targets with the Hungarian algorithm, i.e., we spatially match predictions with detections to verify the existence of targets in $\mathcal{T}_2$. The remaining unmatched detections are initialized as new tracks.

The advantages of the proposed EAT algorithm are twofold. For one thing, all tracks are simultaneously predicted by their own assigned SOT trackers. The identities are kept and transferred with SOT predictions, turning the MOT into a template-matching problem and eliminating the complexity of data association. For another thing, with the predicted existence confidence, partially occluded and missed targets are recovered, as shown in Fig 4 (b), reducing FNs and ID switches and mitigating the issue in the difference of required features between classification and existence of targets.

F. Training and Inference

Unifying the object detector and SOT tracker into a single network enables DMMTracker to be trained end-to-end, and the training loss comprises three parts: $\mathcal{L}_{RPN}$ for the RPN, $\mathcal{L}_{Det}$ for the object detector, and $\mathcal{L}_{SOT}$ for the Siamese tracker. The training of the RPN and detector follows [26] by adding up the losses of classifications and regressions. The proposed QSS strategy samples the training triplets for the Siamese tracker, where four types of training triplets are equally distributed. As shown in Fig. 2, the prediction heads of the Siamese tracker have four outputs, including classification, regression, centerness, and existence. The losses of classification $\mathcal{L}_{cls}$ and regression $\mathcal{L}_{reg}$ in the Siamese tracker are obtained with Focal Loss and IoU Loss, respectively.

The centerness of the location (x, y) is defined as:

$$\mathcal{C}_{x,y} = \sqrt{\frac{\min\left(x - x_0, x_1 - x\right)}{\max\left(x - x_0, x_1 - x\right)} \cdot \frac{\min\left(y - y_0, y_1 - y\right)}{\max\left(y - y_0, y_1 - y\right)}}, \tag{11}$$

where (x_0, y_0) and (x_1, y_1) denote the top-left and bottom-right coordinates of the ground truth bounding box. Obviously, the centerness $\mathcal{C}_{x,y}$ contrasts with the distance between the location (x, y) and the corresponding target center, and $\mathcal{C}_{x,y}$ is set to 0 if (x, y) is located in the background area. The loss of the centerness branch is calculated as follows:

$$\mathcal{L}_{cen} = \frac{-1}{N} \sum_{x,y} \mathcal{C}_{x,y} \cdot \log \mathcal{A}^{cen}_{x,y} + (1 - \mathcal{C}_{x,y}) \cdot \log\left(1 - \mathcal{A}^{cen}_{x,y}\right), \tag{12}$$

where $\mathcal{A}^{cen}$ denotes the predicted centerness heatmap, and N represents the total number of valid points inside the target.

As illustrated in Sec. III-E, the target existence confidence is predicted based on its ground truths $\hat{\mathcal{E}}$ in Eq. 10, and the training of the existence estimation network is supervised by Cross Entropy Loss as follows:

$$\mathcal{L}_{exi} = CrossEntropy\left(\mathcal{E}, \hat{\mathcal{E}}\right), \tag{13}$$

where $\mathcal{E}$ is the predicted target existence. Therefore, the loss

of the Siamese tracker $\mathcal{L}_{SOT}$ can be formed as follows:

$$\mathcal{L}_{SOT} = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \mathcal{L}_{cen} + \mathcal{L}_{exi}. \tag{14}$$

The overall loss $\mathcal{L}$ of the proposed DMMTracker is:

$$\mathcal{L} = \mathcal{L}_{RPN} + \mathcal{L}_{Det} + \mathcal{L}_{SOT}. \tag{15}$$

At the inference stage, we adopt the scale change penalty and the cosine window function [18] to refine the classification score. The final tracking results are obtained by merging detections and SOT predictions following the proposed EAT algorithm. The targets that fail in predictions by the SOT tracker are kept inactive for 30 frames instead of being terminated, and the SOT tracker continuously searches for targets in future frames along with active tracks to handle occlusions and target re-appearance. Moreover, our DMMTracker follows the public tracking paradigm, i.e., we do not initialize new tracks unless provided by the benchmark detections.

IV. Experiments

In this section, we first present the evaluation datasets, metrics, and implementation details. Then, we demonstrate the effectiveness of the proposed Task-specific Feature Encoding Network (TFEN), Quadruplet State Sampling (QSS) strategy, and Existence Aware Tracking (EAT) algorithm by ablation studies. Finally, we compare our method with those of previous works to demonstrate the superior performance of DMMTracker and discuss its limitations.

A. Datasets and Metrics

We evaluate our tracker on MOTChallenge Benchmark datasets, including MOT16, MOT17 [40], and MOT20 [41]. The MOT16 and MOT17 contain the same 7 sequences for training with publicly available ground truths and 7 sequences for testing. However, the frame-wise detections of MOT16 are provided by DPM [42], while MOT17 provides more accurate annotations from detectors DPM, Faster R-CNN, and SDP [43]. The MOT20 contains 4 training and 4 testing sequences with detections provided by Faster R-CNN under extremely crowded scenes. To make fair comparisons, we conducted all experiments with the public detection regime to avoid the discrepancy introduced by detectors.

We follow the common practices of adopting the CLEAR MOT metric [44] for evaluation. Specifically, metrics including Multi-Object Tracking Accuracy (MOTA), ID F1 Scores (IDF1) [45], False Positives (FP), False Negatives (FN), ID switches (IDS), Most Tracked (MT), Most Lost (ML) and Frames Per Second (FPS) are assessed. The MOTA primarily evaluates tracking convergence, and IDF1 describes identity consistency. FPS indicates the inference speed. In addition, the newly released metric Higher Order Tracking Accuracy (HOTA) [46], which balances the effect of detection, association, and localization into a single metric, is also employed for comparison.

B. Implementation Details

We utilized DLA-34 [47] with FPN [48] as the backbone network. The training strategy mostly follows the baseline method [16]. Specifically, our network is pre-trained on CrowdHuman [49] with a learning rate of 5×10^{-3} and batch size of 16 under the image-based training strategy and then fine-tuned on MOT17/MOT20 with a learning rate of 1.5×10^{-3} and batch size of 8 under the video-based training strategy. Image pairs are used as the input for our tracker, and the outputs of the RPN provide the training candidates by sampling 256 image regions. We resize the image pair to ensure that its shorter side has 800 pixels.

The parameter settings follow the baseline to eliminate the influence of fine-tuning and demonstrate the effectiveness of the proposed method. Specifically, the detection confidence threshold for starting a new track is 0.6, and the threshold η_1 is set to 0.4 to determine the survival of the SOT-based predictions. The threshold η_2 is set to 0.3, which is experimentally selected and shown to be optimal for deciding whether to re-consider low-scored predictions in the EAT algorithm. In addition, we expand the template patch by a factor of 2 for both the width and height to obtain the search region of each target while predicting positions. All the experiments are performed on two NVIDIA GeForce RTX 2080 Ti GPUs, unless stated otherwise.

C. Effectiveness of the Proposed Modules

To verify the effectiveness of each component in DMMTracker, we conduct experiments by ablating each proposed module of this paper. The results are obtained by testing on the MOT17 test dataset under the public detection protocol to avoid the influence of overfitting, and the results are evaluated via the MOTChallenge online submission. The results are shown in Table I, where the baseline utilizes a vanilla Siamese tracker [18] as the motion model for instance-level position prediction. We assembled the proposed TFEN, QSS, and EAT on top of the baseline for testing.

As shown in Model ① in Table I, compared with the baseline, the proposed TFEN can reduce FN and ML and increase MT, showing that more targets are successfully recovered and tracked. Consequently, the MOTA improved by 0.6, and IDF1 increased by 0.5. Similar improvements can be observed by comparing Models ⑤ and ⑥ with Models ② and ③. Therefore, task-driven features are captured with the help of TFEN, as demonstrated by the improved performance of the detection-related metric MOTA and the identity-related metric IDF1, and the incompatibility in required features between the detection and SOT is alleviated at the same time.

Compared with the baseline, the increased IDF1 and MT verify the enhanced identity consistency of Model ②, which is attributed to the proposed QSS in guiding the SOT tracker to capture identity-discriminative features and intra-class differences during position predictions. Similar improvements can be observed by comparing Model ⑤ with Model ①. Since using QSS alone enriches training triplets compared with the baseline, such as hard positives and hard negatives, the improvements of Model ② are less prominent than those of Model ① when evaluated on the test dataset. However, significant improvements are achieved in Model ⑤ when using TFEN and QSS together, where the task-specific features

TABLE I
ABLATION STUDIES ON EACH COMPONENT OF THE DMMTRACKER. THE RESULTS ARE OBTAINED ON THE MOT17 TEST DATASET WITH PUBLIC DETECTION PROTOCOL. THE ARROWS IN THE TABLE INDICATE THE OPTIMAL TREND OF EACH METRIC.

Model	TFEN	QSS	EAT	MOTA↑	IDF1↑	HOTA↑	MT↑	ML↓	FP↓	FN↓	IDS↓
Baseline				65.9	63.3	-	34.6	23.9	18098	170955	3040
①	✓			66.5	63.8	51.9	39.2	19.8	23656	161786	3411
②		✓		66.1	63.4	52.0	39.1	19.7	23955	163584	3811
③			✓	66.2	63.4	51.5	39.0	20.4	23542	163719	3304
④		✓	✓	66.2	63.8	51.6	39.1	20.0	25460	161806	3613
⑤	✓	✓		66.7	64.2	51.9	38.9	19.9	23343	160920	3424
⑥	✓		✓	66.9	64.3	52.0	39.1	19.7	23583	159865	3227
⑦	✓	✓	✓	67.1	64.3	52.1	39.7	19.7	25728	156791	3135

TABLE II
ABLATION STUDIES ON THE TFEN. THE A, B, AND C IN THE TABLE ARE THREE PARTS OF TFEN.

Model	A	B	C	MOTA↑	IDF1↑	FP↓	FN↓	IDS↓
Baseline				64.1	63.6	12151	107325	1460
①	✓			64.3	64.2	11521	106887	1878
②			✓	64.3	64.5	11580	106594	2070
③	✓		✓	64.4	64.6	11516	105838	2080
TFEN	✓	✓	✓	64.5	65.0	11448	105949	2041

TABLE III
ABLATION STUDIES OF THE EFFECTIVENESS OF THE PROPOSED QSS STRATEGY AND EAT ALGORITHM.

Model	QSS	EAT	MOTA↑	IDF1↑	FP↓	FN↓	IDS↓
SiamMOT			64.1	63.6	12151	107325	1460
	✓		64.4	64.1	10891	107232	1943
		✓	64.2	64.0	12287	107039	1929
	✓	✓	64.6	64.6	12255	105221	1729
DMMTracker			64.7	64.9	11677	105110	2035
	✓		64.7	65.0	12202	105014	1868

are captured on top, increasing the baseline by 0.8 in MOTA and 0.9 in IDF1, showing the effectiveness of task-specific features in facilitating the discrimination of identity features

Reduced FN and ML and increased MOTA in Model ③ over the baseline demonstrate that the existence estimation and EAT algorithm can help to recover missed targets and form longer trajectories. The effectiveness of EAT can also be observed from the boosted performance in Model ⑥ compared to Model ①. Thus, predicting the existence of targets and managing identity with the EAT can help to recover more targets, boost the accuracy of existence judgment, and eliminate the difference in required features between classification and existence. Moreover, the comparison of the MOTA and HOTA between Model ⑥ and Model ③ shows that the target existence judgments can also benefit from the target-specific features since existence estimation is a target-dependent task.

D. Ablation Experiment of TFEN

The TFEN is designed to capture task-driven features from shared backbone features. As shown in Fig. 3, the independent features for each task are encoded in sub-networks A and C, and mutual information is aggregated by sub-network B to contribute to the final task-specific features. To investigate the effectiveness of each part, we ablate A, B, and C of TFEN, and the results on the MOT17 training set are shown in Table II. The baseline in Table II feeds the shared backbone features into the object detector and SOT tracker without processing.

According to Models ① and ② of Table II, feature encoding networks A and C can independently improve the tracking performance. Specifically, network A encodes class-specific features for the detector, reducing FP and FN and improving the MOTA and IDF1, as shown in Model ①. More noticeable improvements are achieved when equipped with network C in Model ② , which enhances the performance of the SOT tracker by capturing target-specific features. The reason for this improvement difference is that the base network of our method is Faster R-CNN, which is pre-trained on a larger detection dataset before being fine-tuned on the tracking dataset. Thus, the SOT tracker is biased regarding the feature extraction after fine-tuning, leaving more room for improvement. This phenomenon also proves the existence of differences in the desired features between detection and SOT, which leads to this imbalance feature extraction issue. It is no surprise that networks A and C can further boost the performance when used together in Model ③ since they are designed to extract different task-specific features

The best results are achieved by adding network B on top of A and C, demonstrating that beneficial mutual information between the detector and the SOT tracker exists, which can enrich task-specific features and boost tracking performance.

E. Ablation Experiment of QSS

The core motivation for employing QSS to sample training triplets of the Siamese tracker lies in the fact that training samples can guide the feature extraction and affect the prediction accuracy. To understand the impact of QSS on promoting identity consistency, we conducted ablation studies on the MOT17 training set by applying the QSS strategy to SiamMOT and DMMTracker. The results are shown in Table III. Models without QSS in the table are trained with the vanilla sampling strategy as in the baseline method.

Compared with models without QSS, we can see that QSS can help improve identity consistency for both SiamMOT and DMMTracker. Although IDS increases in both trackers, the increased IDF1 (0.5 for SiamMOT and 0.3 for DMMTracker) verifies that the identity consistency is enhanced. Therefore, by training with four types of triplets sampled by the QSS strategy, the Siamese tracker is more sensitive to intra-class

differences and captures identity-discriminative features to distinguish tracking targets from distractors, leading to improved identity assignment accuracy. As a result, longer trajectories are formed with enhanced template-matching accuracy, reducing the number of FPs and FNs and relieving the issue of feature incompatibility between the SOT and MOT.

F. Ablation Experiment of EAT

The incompatibility of the required features between the classification and existence is a long-standing issue in video understanding tasks. DMMTracker handles this issue by estimating target existence confidences and associating targets with EAT algorithms. We conduct ablation studies on the MOT17 training set to verify its effectiveness by applying target existence estimation and EAT algorithm to the baseline SiamMOT and DMMTracker. The models without EAT use traditional classification-based determinations.

It can be observed from Table III that when managing identities with the EAT algorithm, both SiamMOT and DMMTracker can recover missed targets to reduce FN and outperform their counterparts in Table III. The reason for the increased FP of SiamMOT when applying EAT is that some false positives are mistakenly recovered. However, the improved MOTA and IDF1 verify that more true positive targets are recovered and some false identity assignments are revised, leading to the formation of high-quality trajectories.

In addition, we investigate the effectiveness of the IoU-based detection matching verification step in the EAT algorithm. Although MOTA increases slightly (from 64.5 to 64.6) by recovering high existence confidence targets without detection matching verification, the FP increases sharply (from 11448 to 11861), leading to decreased IDF1 (from 65.0 to 64.5). Therefore, we only recover the low-scored predictions, which are matched and verified via detection.

The threshold η_1 in EAT determines the survival of predictions of the SOT tracker, and η_2 is the threshold for determining whether a low-scored prediction is re-considered. An ablation experiment is conducted on the MOT17 training set to investigate the parameter settings, and the results are shown in Table IV. As expected, a higher threshold η_1 increases FN since more predictions are suppressed. In contrast, a lower η_1 would inevitably increase FP and IDS. We set η_1 to 0.4, which follows the baseline and avoids overfitting. We set η_2 to 0.3, which achieves good performance among different settings.

It can be found that FN decreases as η_2 increases in Table IV, which may be counterintuitive. We conjecture the reason is that the number of false predictions increases when a lower threshold η_2 is used first. However, these falsely recovered targets are difficult to track in later frames since there are no matching targets in future frames, leading to increased FN. In contrast, a larger η_2 will increase the proportion of recovered true positives and reduce the incorrectly survived ones. Therefore, most recovered targets can be continuously tracked in the later frames, leading to a reduced FN.

G. Benchmark Comparison

We extensively evaluate our DMMTracker by comparing it with the state-of-the-art (SOTA) methods on the MOTChallenge benchmarks, including the MOT16, MOT17, and MOT20 datasets. We follow the public detection paradigm and submit our results to the MOTChallenge online testing service. The results are illustrated in Table V. We only consider methods of public detection paradigms that are comparable to our tracker.

TABLE IV
SENSITIVITY ANALYSIS OF THE THRESHOLD η_1 AND THE THRESHOLD η_2 ON THE MOT17 TRAINING DATASET.

η_1	η_2	MOTA↑	IDF1↑	FP↓	FN↓	IDS↓
0.3	0.1	63.9	64.4	12958	106527	2049
0.3	0.2	63.9	64.4	13108	106406	2054
0.3	0.3	63.9	64.3	13272	106329	2065
0.3	0.4	63.9	64.3	13352	106270	2064
0.4	0.1	64.6	64.9	10677	106510	2035
0.4	0.2	64.5	64.9	11236	106244	2039
0.4	0.3	64.5	65.0	11448	105949	2041
0.4	0.4	64.5	65.0	11583	105920	2036
0.5	0.1	65.0	64.6	7235	108839	1932
0.5	0.2	64.9	64.8	7931	108275	1920
0.5	0.3	65.0	65.1	8082	107804	1919
0.5	0.4	65.0	64.9	8342	107568	1923

The proposed DMMTracker achieves very competitive results with public detections in all datasets. In particular, DMMTracker outperforms the baseline SiamMOT in MOTA and IDF1 in MOT17, demonstrating the effectiveness of the proposed discriminative motion model in addressing incompatibility issues and boosting tracking performance. Besides, DMMTracker achieves the best MT, ML, and FN in both MOT16 and MOT17, showing that more targets are recovered and longer trajectories are formed. The decreased ML and FN are attributed to the TFEN and EAT, which enhance target position prediction and existence management accuracy. The TFEN and QSS help to promote identity consistency by capturing identity-specific features and leading to increased MT and IDF1. Therefore, the superior performance of DMMTracker confirms the effectiveness of our method in enhancing the accuracy of position predictions and identity assignments.

Compared with other SOT-based trackers, including DMAN [34], STAM [12], KCF [50], DASOT [14], FAMNet [12], UMA [15], and SOTMOT [17], our DMMTracker achieves the best performances on MOTA, IDF1, and HOTA. The previous SOTA method, SOTMOT, is developed from CenterNet [35] and tracks by exploiting center-based SOT predictions. The superior performance of DMMTracker demonstrates the advantage of the region-based features in target position predictions. Moreover, DMMTracker significantly outperforms trackers that employ regression-based motion models [8], [55], confirming the superiority of SOT trackers in making instance-level position predictions in MOT.

Compared with the SOTA method ByteTrack [3], which is a two-stage tracker that relies on an advanced detector [56]. Our DMMTracker is a one-stage tracker that can detect and track all targets in a single forward and benefit from the reciprocal information between detection and position prediction, significantly reducing the complexity of data association. Moreover, our DMMTracker outperforms the recently proposed SOTA methods OC-SORT [27] and UTM [52] in MOT17 under the public detection paradigm.

TABLE V
COMPARISONS WITH THE STATE-OF-THE-ART METHODS ON MOT16, MOT17, AND MOT20 DATASETS WITH PUBLIC DETECTION PROTOCOL. THE SECOND COLUMN INDICATES WHETHER THE TRACKER IS A SOT-BASED METHOD. THE BEST RESULTS OF EACH DATASET ARE HIGHLIGHTED IN BOLD, AND THE SECOND BEST ARE UNDERLINED.

Dataset	Method	SOT	MOTA↑	IDF1↑	HOTA↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FPS↑
MOT16	STAM [12]	✓	46.0	50.0	37.9	14.6	43.6	6895	91117	473	0.2
	DASOT [14]	✓	46.1	49.4	37.3	14.6	41.6	8222	89204	802	9.0
	DMAN [34]	✓	46.1	54.8	40.3	17.4	42.7	7909	89874	532	0.3
	KCF [50]	✓	48.8	47.2	37.2	15.8	38.1	5875	86567	906	0.1
	Tracktor [8]		56.2	54.9	44.6	20.7	35.8	**2394**	76844	617	1.6
	IQHAT [51]		58.6	62.4	49.3	26.1	36.5	4074	71026	636	8.1
	TMOH [28]		63.2	62.5	50.7	27.0	31.0	3122	63376	**428**	0.7
	UTM [52]		63.8	**67.1**	**53.1**	33.3	28.2	8328	57269	635	13.1
	DMMTracker	✓	**67.0**	64.7	52.0	**37.4**	**21.3**	8532	**50827**	884	**16.2**
MOT17	DMAN [34]	✓	48.2	57.8	42.5	19.3	38.3	26218	263608	2194	0.3
	DASOT [14]	✓	49.5	51.8	41.5	20.4	34.6	33640	247370	4142	9.1
	FAMNet [12]	✓	52.0	48.7	-	19.1	33.4	14138	253613	3072	-
	UMA [15]	✓	53.1	54.4	-	21.5	31.8	22893	239534	2251	-
	Tracktor [8]		56.3	55.1	44.8	21.1	35.3	8866	235449	1987	1.5
	OC-SORT [27]		58.2	65.1	52.4	18.3	44.8	**4379**	230449	**784**	**28.6**
	TADAM [29]		59.7	58.7	-	-	-	9676	216029	1930	-
	CenterTrack [9]		61.5	59.6	48.2	26.4	31.9	14076	200672	2583	17.0
	TMOH [28]		62.1	62.8	50.4	26.9	31.4	10951	201195	1897	0.7
	ArTIST [10]		62.3	59.7	48.9	29.1	34.0	19611	191207	2062	4.5
	SOTMOT [17]	✓	62.8	67.4	-	24.4	33.0	6556	201319	2017	16.0
	UTM [52]		63.5	65.1	52.5	37.4	27.0	33683	170352	1686	13.1
	SiamMOT [16]	✓	65.9	63.3	-	34.6	23.9	18098	170955	3040	-
	ByteTrack [3]		**67.4**	**70.0**	**56.1**	31.0	31.2	9939	172636	2387	27.4
	DMMTracker	✓	67.1	64.3	52.1	**39.7**	**19.6**	25728	**156791**	3135	16.1
MOT20	Tracktor [8]		52.6	52.7	42.1	29.4	26.7	6930	236680	1648	1.2
	TBooster [53]		54.6	53.4	42.5	32.8	25.5	9486	223607	2038	1.4
	MTracker [54]		55.6	65.0	-	35.7	31.2	12297	216986	**480**	-
	TADAM [55]		56.6	51.6	-	-	-	39407	182520	2690	-
	IQHAT [51]		57.1	57.7	45.7	40.8	20.0	32247	187937	1875	7.5
	OC-SORT [27]		59.9	**67.0**	54.3	38.5	26.6	**4434**	202502	554	**27.6**
	TMOH [28]		60.1	61.2	48.9	46.7	17.8	38043	165899	2342	0.6
	UTM [52]		**64.4**	65.9	**53.3**	**65.0**	**13.3**	82726	**98974**	2592	6.2
	DMMTracker	✓	62.5	60.5	48.7	49.6	16.4	33795	158057	2043	9.7

H. Qualitative Results and Discussion

We show the qualitative results of our DMMTracker in crowded scenes of the MOT17 and MOT20 test sequences in Fig. 5. We can observe that DMMTracker can accurately track targets under frequent occlusions and maintain consistent identity across frames. The results of MOT17-03 and MOT20-04 show that DMMTracker can successfully localize and track small targets under different lighting conditions. Moreover, the visualization of MOT17-08 and MOT20-08 shows the generalization of DMMTracker.

Despite the superior performance, there is still plenty of room for improvement. We show failure cases of DMMTracker in Fig. 6. The target indicated by the red arrows in Fig. 6 (a) intersects with others and reappears after occlusion, and the ID switch occurs twice. The temporal consistency of target region features is contaminated under continuous occlusion, degrading the accuracy of template matching and identity assignment. This issue can be resolved by referring to global features [57] and dynamically updating the cached features [11] to reduce the reliance on recent tracking results.

Fig. 6 (b) shows the ID switches caused by large target scale changes and long-term occlusions. The reasons for these ID Switches are twofold. First, the tracker lacks long-term occlusion solutions, i.e., the cached features of unmatched targets are removed from the memory after long-term occlusions. In addition, the appearance changes dramatically, breaking the temporal consistency of features and reducing the accuracy of visual similarity. This issue can be resolved by using dedicated memory modules [58] and dynamic target feature update schemes [59].

TABLE VI
COMPARISON IN TRACKING SPEED BETWEEN DMMTRACKER AND ITS LITE TRACKER DMM-LITE.

Model	MOTA↑	IDF1↑	FP↓	FN↓	IDS↓	FPS↑
DMMTracker	64.5	65.0	11448	105949	2041	18.2
DMM-lite	63.1	62.4	12276	110213	1839	25.4

Although the proposed DMMTracker achieves superior performance, the tracking speed is not very fast, it cannot meet the demands of real-time applications, and it becomes worse as the number of targets increases. To satisfy the requirements of real-time tracking, we propose a lite version by simplifying the structure of DMMTracker in two ways. First, we remove the FPN network from the backbone to reduce the computation burden. Second, we build a simplified TFEN by removing network B and simplifying networks A and C since favorable results can be achieved with these sub-networks. We test the tracking speed of this lite tracker (denoted as DMM-lite) on a 3090 Ti GPU and compare the performances of DMMTracker and DMM-lite in the MOT17 under the same experimental settings. The results are shown in Table VI.

From the table, we can see that the DMM-lite can track faster, which can achieve real-time tracking speed. However, the MOTA and IDF1 decrease by 1.4 and 2.6, respectively, and the numbers of false predictions and missing targets increase

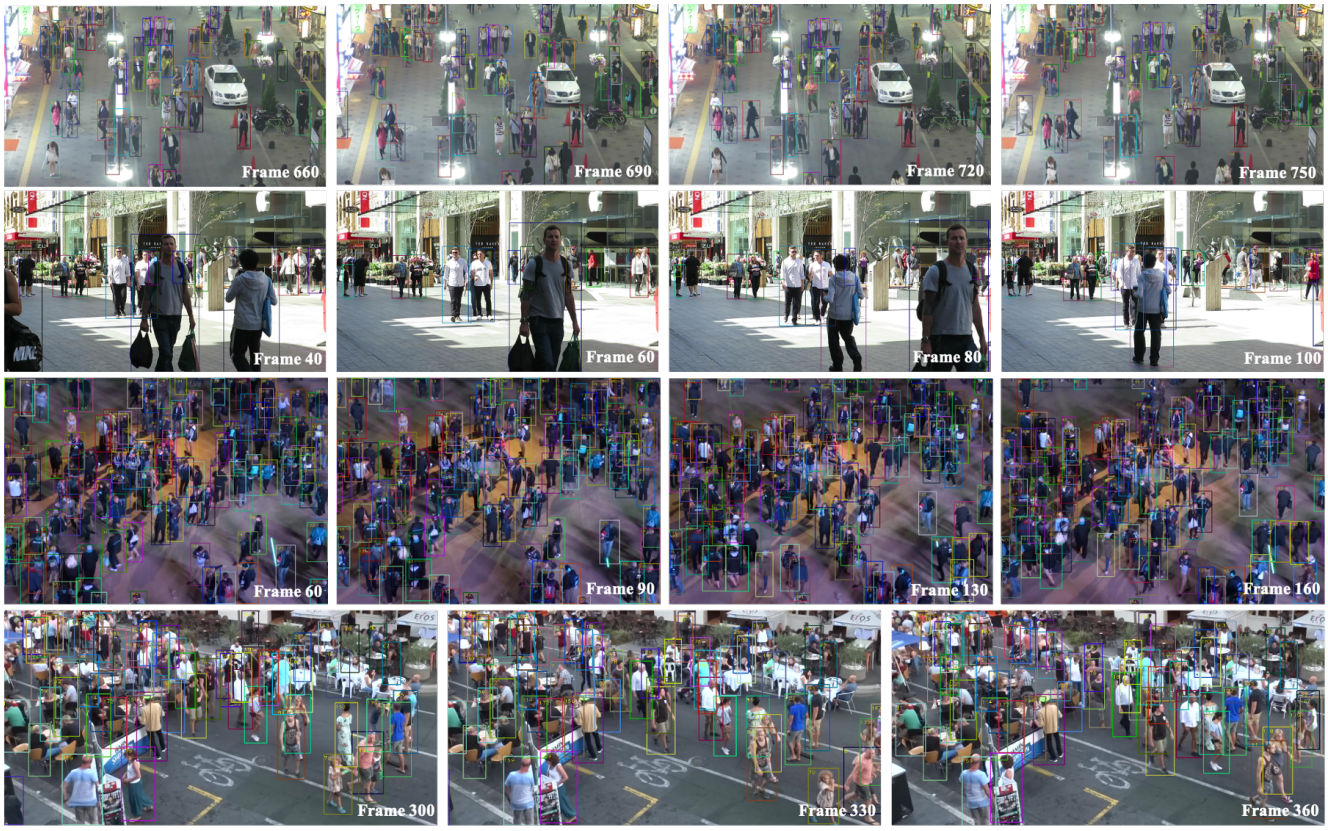

Fig. 5. Qualitative results of the proposed tracker DMMTracker in crowded scenes. From top to bottom: MOT17-03, MOT17-08, MOT20-04, and MOT20-08. Four video sequences are shown with crowded scenes containing frequent target occlusions and intersections.

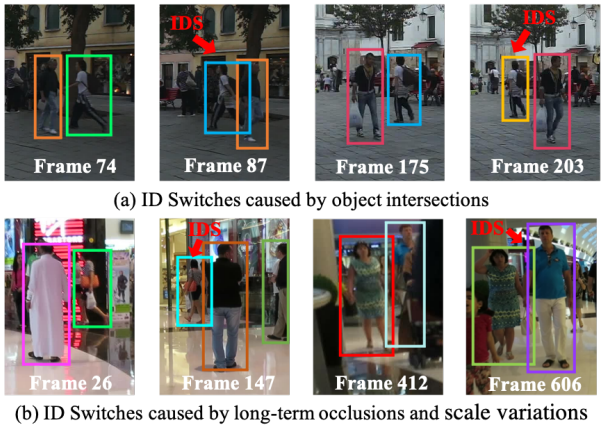

Fig. 6. Examples of failure cases. (a) shows the ID switches caused by frequent object intersections. (b) illustrates the ID switches caused by long-term occlusions and large-scale variations.

simultaneously. Thus, the tracking speed can be expedited by pruning the network structure and sacrificing some tracking robustness. In our future work, we will focus on facilitating tracking speed by applying more advanced SOT methods [60], [61] and referring to knowledge distillation techniques [22] to build the lite tracker, whereby the tracking speed can be boosted without harming tracking performance.

V. Conclusion

In this paper, we discussed the issue of feature incompatibility when employing the SOT trackers as motion models for position predictions in MOT. We resolved these issues by proposing a region-based discriminative motion model. A novel Task-specific Feature Encoding Network (TFEN) was proposed to alleviate the incompatibility between object detection and SOT by extracting task-dependent features for each sub-network. Then, we proposed a novel Quadruplet State Sampling (QSS) strategy to construct the training sample of the SOT tracker, which can guide the motion model to capture identity-discriminative features in position predictions and mitigate the incompatibility between the SOT and MOT. Finally, we proposed an Existence Aware Tracking (EAT) algorithm to eliminate the incompatibility between the classification and the existence of targets by estimating the target existence confidence and re-considering the low-scored predictions for identity assignment. The experiments demonstrate the effectiveness of the proposed DMMTracker in alleviating the incompatibility issues of applying the SOT tracker as the motion model in MOT, and very competitive results are achieved in MOT benchmarks.

References

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

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[3] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *European Conference on Computer Vision*. Springer, 2022, pp. 1–21.

[4] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *European conference on computer vision*. Springer, 2020, pp. 107–122.

[5] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021.

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

[7] E. Yu, Z. Li, S. Han, and H. Wang, "Relationtrack: Relation-aware multiple object tracking with decoupled representation," *IEEE Transactions on Multimedia*, 2022.

[8] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 941–951.

[9] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European Conference on Computer Vision*. Springer, 2020, pp. 474–490.

[10] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould, "Probabilistic tracklet scoring and inpainting for multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 329–14 339.

[11] Z. Qin, S. Zhou, L. Wang, J. Duan, G. Hua, and W. Tang, "Motiontrack: Learning robust short-term and long-term motions for multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 939–17 948.

[12] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4836–4845.

[13] P. Chu and H. Ling, "Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6172–6181.

[14] Q. Chu, W. Ouyang, B. Liu, F. Zhu, and N. Yu, "Dasot: A unified framework integrating data association and single object tracking for online multi-object tracking," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 10 672–10 679.

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

[16] B. Shuai, A. Berneshawi, X. Li, D. Modolo, and J. Tighe, "Siammot: Siamese multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 372–12 382.

[17] L. Zheng, M. Tang, Y. Chen, G. Zhu, J. Wang, and H. Lu, "Improving multiple object tracking with single object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2453–2462.

[18] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8971–8980.

[19] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[20] Z. Liang and J. Shen, "Local semantic siamese networks for fast tracking," *IEEE Transactions on Image Processing*, vol. 29, pp. 3351–3364, 2019.

[21] J. Shen, X. Tang, X. Dong, and L. Shao, "Visual object tracking by hierarchical attention siamese network," *IEEE transactions on cybernetics*, vol. 50, no. 7, pp. 3068–3080, 2019.

[22] J. Shen, Y. Liu, X. Dong, X. Lu, F. S. Khan, and S. Hoi, "Distilled siamese networks for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8896–8909, 2021.

[23] X. Dong, J. Shen, L. Shao, and F. Porikli, "Clnet: A compact latent network for fast adjusting siamese trackers," in *European Conference on Computer Vision*. Springer, 2020, pp. 378–395.

[24] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 976–985, 2020.

[25] D. Yuan, X. Chang, Q. Liu, Y. Yang, D. Wang, M. Shu, Z. He, and G. Shi, "Active learning for deep visual tracking," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[27] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9686–9696.

[28] D. Stadler and J. Beyerer, "Improving multiple pedestrian tracking by track management and occlusion handling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 958–10 967.

[29] S. Guo, J. Wang, X. Wang, and D. Tao, "Online multiple object tracking with cross-task synergy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8136–8145.

[30] A. Girbau, X. Giró-i Nieto, I. Rius, and F. Marqués, "Multiple object tracking with mixture density networks for trajectory estimation," *arXiv preprint arXiv:2106.10950*, 2021.

[31] J. Yang, H. Ge, S. Su, and G. Liu, "Transformer-based two-source motion model for multi-object tracking," *Applied Intelligence*, pp. 1–13, 2022.

[32] S. Sun, N. Akhtar, X. Song, H. Song, A. Mian, and M. Shah, "Simultaneous detection and tracking with motion modelling for multiple object tracking," in *European Conference on Computer Vision 2020*. Springer, 2020, pp. 626–643.

[33] D. Wu, W. Han, T. Wang, X. Dong, X. Zhang, and J. Shen, "Referring multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 633–14 642.

[34] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 366–382.

[35] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[36] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[37] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3516–3527, 2019.

[38] J. Nie, Z. He, Y. Yang, M. Gao, and Z. Dong, "Learning localization-aware target confidence for siamese visual tracking," *IEEE Transactions on Multimedia*, pp. 1–13, 2022.

[39] Y.-f. Li, H.-b. Ji, X. Chen, Y.-k. Lai, and Y.-l. Yang, "Multi-object tracking with robust object regression and association," *Computer Vision and Image Understanding*, p. 103586, 2022.

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

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

[42] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[43] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2129–2137.

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

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

[46] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, no. 2, pp. 548–578, 2021.
[47] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 2403–2412.
[48] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
[49] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *arXiv preprint arXiv:1805.00123*, 2018.
[50] P. Chu, H. Fan, C. C. Tan, and H. Ling, "Online multi-object tracking with instance-aware tracker and dynamic model refreshment," in *2019 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2019, pp. 161–170.
[51] Y. He, X. Wei, X. Hong, W. Ke, and Y. Gong, "Identity-quantity harmonic multi-object tracking," *IEEE Transactions on Image Processing*, vol. 31, pp. 2201–2215, 2022.
[52] S. You, H. Yao, B.-K. Bao, and C. Xu, "Utm: A unified multiple object tracking model with identity-aware feature enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 876–21 886.
[53] G. Wang, Y. Wang, R. Gu, W. Hu, and J.-N. Hwang, "Split and connect: A universal tracklet booster for multi-object tracking," *IEEE Transactions on Multimedia*, 2022.
[54] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Robust multi-object tracking by marginal inference," in *European Conference on Computer Vision*. Springer, 2022, pp. 22–40.
[55] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "Siamcar: Siamese fully convolutional classification and regression for visual tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6269–6277.
[56] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.
[57] L. Han and Z. Yin, "Global memory and local continuity for video object detection," *IEEE Transactions on Multimedia*, 2022.
[58] M. Xi, W. Zhou, N. Wang, and H. Li, "Learning temporal-correlated and channel-decorrelated siamese networks for visual tracking," *IEEE Transactions on Multimedia*, 2021.
[59] J. Shen, Z. Liang, J. Liu, H. Sun, L. Shao, and D. Tao, "Multiobject tracking by submodular optimization," *IEEE transactions on cybernetics*, vol. 49, no. 6, pp. 1990–2001, 2018.
[60] Z. Zhao, S. Zhao, and J. Shen, "Real-time and light-weighted unsupervised video object segmentation network," *Pattern Recognition*, vol. 120, p. 108120, 2021.
[61] X. Dong, J. Shen, F. Porikli, J. Luo, and L. Shao, "Adaptive siamese tracking with a compact latent network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.