SMILEtrack: SiMIlarity LEarning for Occlusion-Aware Multiple Object Tracking

Wang Yu Hsiang[1], *Jun-Wei Hsieh[1], Ping-Yang Chen[2], Ming-Ching Chang[3], Hung Hin So[4], Xin Li[3]
[1]College of AI and Green Energy and [2]Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
[3]Department of Computer Science, University at Albany, State University of New York, USA
[4] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
j122333221.ai09@nycu.edu.tw, *jwhsieh@nycu.edu.tw, pingyang.cs08@nycu.edu.tw, mchang2@albany.edu,
1155157729@link.cuhk.edu.hk, xli48@albany.edu

Abstract

Despite recent progress in Multiple Object Tracking (MOT), several obstacles such as occlusions, similar objects, and complex scenes remain an open challenge. Meanwhile, a systematic study of the cost-performance trade-off for the popular tracking-by-detection paradigm is still lacking. This paper introduces SMILEtrack, an innovative object tracker that effectively addresses these challenges by integrating an efficient object detector with a Siamese network-based Similarity Learning Module (SLM). The technical contributions of SMILETrack are twofold. First, we propose an SLM that calculates the appearance similarity between two objects, overcoming the limitations of feature descriptors in Separate Detection and Embedding (SDE) models. The SLM incorporates a Patch Self-Attention (PSA) block inspired by the vision Transformer, which generates reliable features for accurate similarity matching. Second, we develop a Similarity Matching Cascade (SMC) module with a novel GATE function for robust object matching across consecutive video frames, further enhancing MOT performance. Together, these innovations help SMILETrack achieve an improved trade-off between the cost (e.g., *running speed) and performance* (e.g., *tracking accuracy) over several existing state-of-the-art benchmarks, including the popular BYTETrack method. SMILETrack outperforms BYTETrack by* **0.4-0.8 MOTA** *and* **2.1-2.2 HOTA** *points on MOT17 and MOT20 datasets. Code is available at https://github.com/pingyang1117/SMILEtrack_Official*

1. Introduction

The task of Multiple Object Tracking (MOT) is to estimate the trajectories of each target and associate them between frames in video sequences. MOT has found widespread applications in various fields, including computer interaction [19, 37], smart video analysis, and autonomous driving. Modern MOT systems [3, 42] typically follow the Tracking-By-Detection (TbD) paradigm, which involves two separate steps of detection and tracking. The detection step locates the object of interest in a single video frame, while the tracking step links each detected object to the existing tracks or creates new tracks if none are found. Despite enormous efforts in MOT investigation, the task remains challenging due to vague objects, occlusion, and complex scenes in real-world applications.

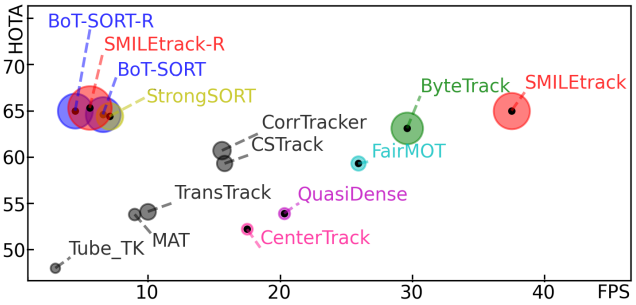

Figure 1. Comparative analysis of HOTA-MOTA-FPS for different trackers on the MOT17 test set. X-axis: FPS (running speed). Y-axis: HOTA. Circle radius: MOTA score. SMILEtrack registers 80.7 MOTA and 65.0 HOTA at 37.5 FPS, exceeding all other trackers (see Table 1 for details).

In the Tracking-By-Detection (TbD) paradigm, two primary strategies prevail, namely Joint Detection and Embedding (JDE) and Separate Detection and Embedding (SDE). JDE methods [42, 51] combine the detector and the embedding model into a single-shot deep network that outputs the detection results and the corresponding appearance embedding features in one inference. Alternatively, SDE methods [1, 3, 10] require at least two function components: a detector and a re-identification model. The detector locates all objects in a single frame via bounding

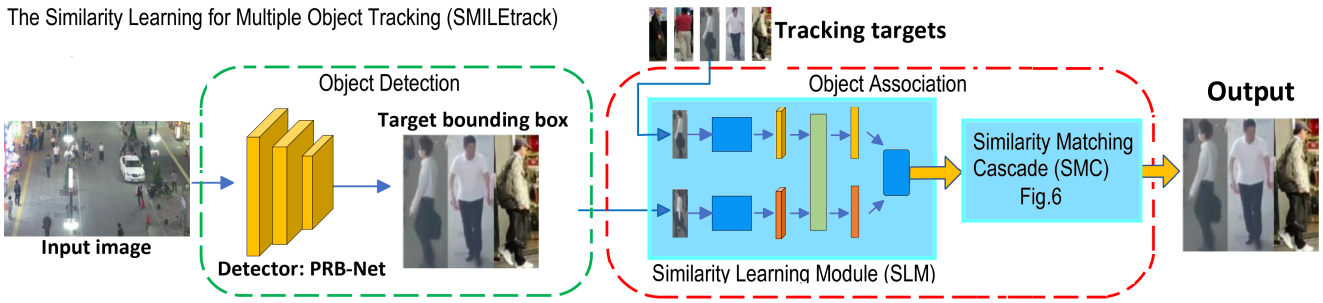

Figure 2. The architecture of the proposed SMILEtracker. SMILEtracker is a Siamese network-like architecture that learns the appearance features of two objects and calculates their similarity score. SMILEtracker consists of two modules: (i) object detection and (ii) object association.

boxes [4, 6, 16, 25–28]. The re-identification model then extracts the embedding features of each object from its bounding box, and these features are used to associate each bounding box with one of the existing trajectories. Despite their flexibility, the efficiency of SDE methods trails behind that of JDE due to the necessity of two separate models.

The motivation behind this work is two-fold. One of the long-standing problems in MOT is occlusion handling, and the other is a principled solution to speed-accuracy trade-off. Although the TbA method has impressive results on feature attention, its exceptional feature attention results in a high time complexity that reduces inference speed. In addition, occlusions can cause tracked objects to pay less attention, resulting in the failure of MOT. Meanwhile, TbD methods such as ByteTrack [50] enjoy computational efficiency, but their accuracy is not optimized. It is highly desirable to develop a class of MOT methods that can strike an improved trade-off between cost (*e.g.*, running speed measured by FPS) and performance (*e.g.* tracking accuracy measured by MOTA [2]).

This paper proposes a novel object tracker, **Similarity Learning for Multiple Object Tracking (SMILEtrack)**, which combines an object detector and a Similarity Learning Module (SLM) to address various challenges in MOT, especially occlusion. Fig. 2 shows the architecture of our SMILEtrack, which provides two major contributions to achieving the State-of-the-Art (SoTA) MOT system: (1) an efficient and lightweight self-attention mechanism that learns the similarity between two candidate bounding boxes. Although the SDE model can achieve high accuracy in object tracking, most feature descriptors used in the model cannot differentiate between objects with similar appearances. To solve this problem, we propose a Siamese network-based Similarity Learning Module (SLM) that can calculate the similarity in appearance between two objects. Inspired by the vision Transformer [9], we introduce a Patch Self-Attention (PSA) block in SLM to produce reliable features for similarity matching. (2) a robust tracker with a novel GATE function that can associate each candidate bounding box from video frames, leading to improved MOT performance. To better handle occlusions, we create a Similarity Matching Cascade (SMC) module that takes SLM results and matches multiple objects robustly across frames. The proposed network achieves SoTA performance on the MOT17 and MOT20 datasets. Contributions of our work are summarized as follows.

- We propose SMILETrack, a separate detection and tracking model, to track multiple objects in frames. SMILETrack can outperform BYTETrack [50] by 0.4-0.8 MOTA points and over 2.0 HOTA points on the MOT17 and MOT20 datasets; see to Fig. 1.
- We introduce a Siamese network-based Similarity Learning Module (SLM) to learn the similarity in appearance between objects for tracking.
- A Patch Self-Attention (PSA) block is proposed that uses a self-attention mechanism to produce reliable features for similarity matching.
- We design a Similarity Matching Cascade (SMC) module to match objects more reliably across frames, which improves performance largely in the presence of occlusions.

2. Related Work

2.1. Tracking-by-Detection

The Tracking-by-Detection (TbD) method has become one of the most popular approaches in the MOT framework. The main tasks of the TbD method can be roughly divided into two parts: object detection and object association.

Object Detection: Mainstream visual object detection models fall into two categories, namely, the two-stage (proposal-driven) and one-stage (direct) detectors. The two-stage methods [28] offer high accuracy but at the cost of speed. On the contrary, one-stage methods are faster but less accurate. YOLO object detection models [4, 25–27] have been widely used in multi-object tracking (MOT) applications due to their speed and accuracy. However, these anchor-based detectors introduce many hyperparameters and consume significant time and memory during training. To mitigate these issues, anchor-free detectors such

as CenterNet [56], and YOLOX [12] have emerged. Despite their improvements [1, 50], these tracking devices still struggle to accurately detect objects of varying sizes. PRB-Net [6] is an effective object detector for MOT tasks, addressing the limitations of anchor-based and anchor-free detectors.

Object Association: SORT [3] is a simple effective tracking algorithm that uses Kalman filtering and Hungarian matching for object association. It struggles with challenges such as occlusions and fast-moving objects. DeepSORT [43] alleviates occlusion issues by incorporating CNN-based appearance features; however, this compromises execution speed. To address this efficiency issue, FairMOT [51] employs an anchor-free method based on CenterNet [56], which significantly improves the MOT performance on the MOT17 dataset. To improve tracking efficiency, numerous MOT methods [32, 33] ignore the appearance features of objects, instead leveraging high-performance detectors and motion cues. Despite achieving impressive results and fast inference on MOTChallenge [21] benchmarks, we posit that their performance is largely dependent on the simplicity of the movement patterns of the dataset. Omitting appearance features may compromise tracking accuracy and robustness in densely populated scenes.

2.2. Tracking-by-Attention

Trackformer [20] extends its success in object detection to MOT by casting the task into a frame-to-frame set prediction problem. Data association between frames is calculated through attention, and a set of track predictions across frames is evolved using the encoder-decoder architecture of Transformer. Similarly, TransTrack [34] uses an attention-based query-key mechanism to perform object detection and association in a single shot based on Deformable DETR [57]. TransCenter [46] is another Transformer-based architecture that uses image-related dense detection queries and sparse tracking queries for MOT. However, all Transformer-based schemes are computationally intensive, and thus not suitable for real-time applications.

3. Methodology

We introduce **Similarity Learning for Multiple Object Tracking (SMILEtrack)**, a novel MOT architecture integrating a detector [6] and a Similarity Learning Module (SLM). SMILEtrack comprises two modules, as shown in Fig. 2: *object detection* and *object association*. The former model was designed primarily to excel in localizing large and small pedestrians, achieving both accuracy and efficiency, making it a superior choice over the traditional YOLOX method [12]. The technical contributions of this work are mainly in the latter module, which consists of: (1) *similarity calculation*, where a novel similarity learning module (SLM) learns the appropriate features and computes an appearance affinity matrix using a Siamese network; and (2) *object association*, where a Similarity Matching Cascade (SMC) module solves the MOT linear assignment problem using the Hungarian algorithm. Details are explained in the following sections.

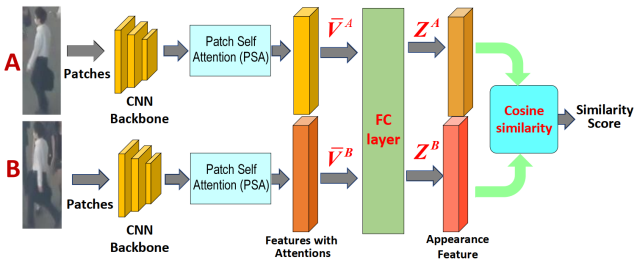

Figure 3. Appearance similarity between low-score detection at the current frame and tracks at the previous frame.

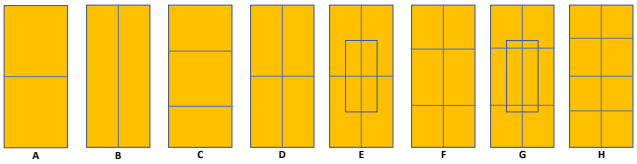

Figure 4. Different types of patch layout: configuration (E) achieves the best performance because it can actively attend to PSA-occluded parts when occlusion occurs.

3.1. Similarity Learning Module (SLM)

Object appearance information is essential for achieving robust tracking quality. Although SORT is a simple association framework that can achieve high-speed inference time, its similarity score does not consider object appearance information and cannot handle long-term occlusion or objects with fast motion. DeepSORT [43] addresses this problem by using a pre-trained CNN to compute bounding-box appearance descriptors. However, this descriptor only considers the similarity between the same objects, without considering the dissimilarity between different objects in different frames. Here, we propose the Similarity Learning Module (SLM) that leverages a Siamese network architecture to learn more discriminative appearance features and accurately track objects across frames.

Fig. 3 shows the SLM architecture. It takes the target and query objects as input in the Siamese network. Both are divided into several patches and then pass through the **Patch Self-Attention (PSA)** block. Note that the height-width ratio of all patches is not fixed (see Fig. 4). Since objects of interest in the MOT17 and MOT20 datasets are assumed to be pedestrians, we have found that configuration (E) achieves the best performance. This can be explained away by observing that layout (E) exploits both prior knowledge about walking pedestrians (i.e., the height-width ratio is approximately 2:1) and translation invariance (i.e., the center box is a shifted version of four surrounding boxes).

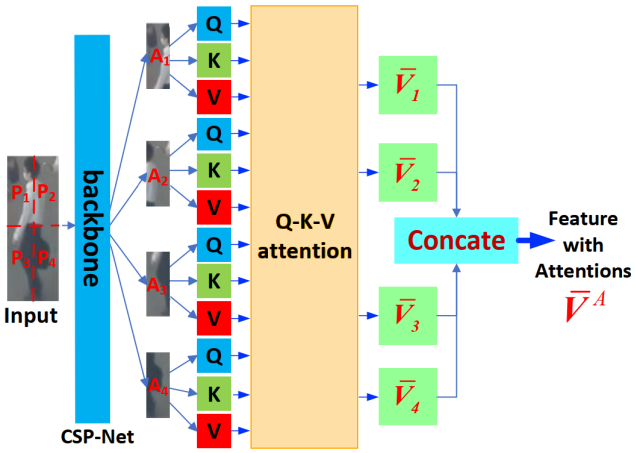

Figure 5. The Patch Self-Attention (PSA) architecture.

3.1.1 Patch Self-Attention (PSA) Block

To produce a reliable appearance feature, a superior feature representation is essential. Inspired by the Vision Transformer (VIT) [9], each SLM input is divided into separate patches. Then, all the patches and their positions are embedded together and fed into a backbone to extract rich feature vectors. Then, three fully connected networks are adopted to convert the deep visual features of all patches to three sets of compact features, *i.e.*, query, key, and value. Based on the features from the query and key sets, various attentions among different combinations can be calculated and used to weight the features from the value set of each patch to form a feature vector to represent an object more accurately. The detailed architecture of the PSA block is shown in Fig. 5.

3.1.2 The Q-K-V Attention

Since input objects are of different sizes, we resize them to a fixed size $W \times H$ where W and H are, respectively, set to 80 and 224 in this paper. Assume that an object $\boldsymbol{A}$ is divided into N_P patches $\{P_i\}_{i=1,...,N_P}$. Each patch P_i has a fixed size $W_P \times H_P$. Then, we use a row-major scanning order to convert each P_i to a column vector. Then, an object can be represented as a sequence of N_p feature vectors: $(P_1, ..., P_i, ..., P_{N_p}), P_i \in R^{D_p}$, where $D_p = W_p \times H_p$. Before feature extraction, the values of pixels in P_i are normalized to $[0, 1]$. Since there are geometrical relations between the patches in $\boldsymbol{A}$, their representation should be modified to preserve position-dependent properties. For the ith patch P_i, its position embedding vector E_i is specified by the standard transformer [36]. It follows that an object $\boldsymbol{A}$ is embedded as $\boldsymbol{A} = (A_1, ..., A_i, ..., A_{N_P})$, where $A_i = P_i + E_i$ and $\boldsymbol{A} \in R^{D_p \times N_P}$. For each A_i, we adopt the CSP-Net framework [38] as the backbone to convert it into a feature matrix $\boldsymbol{F_i}$. $\boldsymbol{F_i}$ includes d_f row vectors and C column vectors; that is, $\boldsymbol{F_i} \in R^{d_f \times C}$, where C is the number of feature channels and d_f is the size of the last layer of the feature pyramid created by CSP-Net [38].

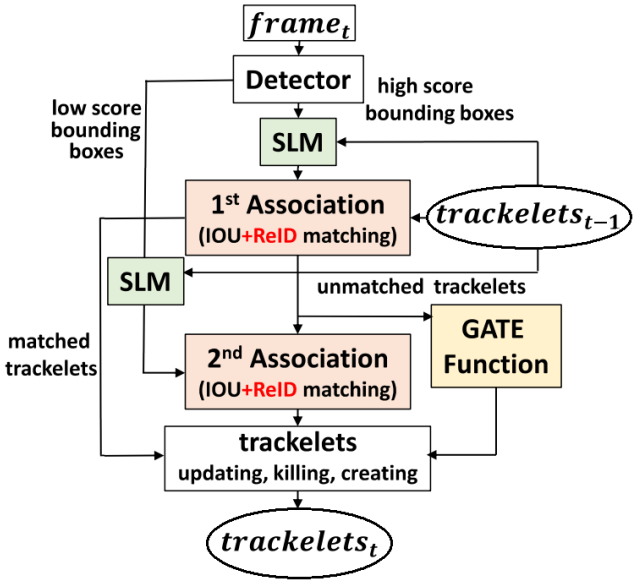

Figure 6. The Similarity Matching Cascade (SMC) pipeline.

Let $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$ be three learned linear transforms that map $\boldsymbol{F_i}$ to the query $\boldsymbol{Q_i}$, the key $\boldsymbol{K_i}$, and the value $\boldsymbol{V_i}$, respectively. Assume that $\boldsymbol{W}_K$ and $\boldsymbol{W}_Q$ have the same number of column vectors, *i.e.*, d_k. Also, there are d_v column vectors in $\boldsymbol{W}_V$. Then $\boldsymbol{W}_Q \in R^{C \times d_k}$, $\boldsymbol{W}_K \in R^{C \times d_k}$, and $\boldsymbol{W}_V \in R^{C \times d_v}$. With $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$, we can obtain $\boldsymbol{Q_i}$, $\boldsymbol{K_i}$, and $\boldsymbol{V_i}$ by the following equations:

$$\boldsymbol{Q_i} = \boldsymbol{F_i}\boldsymbol{W}_Q, \boldsymbol{K_i} = \boldsymbol{F_i}\boldsymbol{W}_K, \boldsymbol{V_i} = \boldsymbol{F_i}\boldsymbol{W}_V, \quad (1)$$

where $\boldsymbol{Q_i} \in R^{d_f \times d_k}$, $\boldsymbol{K_i} \in R^{d_f \times d_k}$, and $\boldsymbol{V_i} \in R^{d_f \times d_v}$. Before matching, the norms of $\boldsymbol{Q_i}$, $\boldsymbol{K_i}$, and $\boldsymbol{V_i}$ will be normalized to be one; that is, $||\boldsymbol{Q_i}||$=1, $||\boldsymbol{K_i}||$=1, and $||\boldsymbol{V_i}||$=1.

Let $\otimes$ denote the Hadamard product and $Sum(\boldsymbol{M})$ be an element-wise sum on a matrix $\boldsymbol{M}$. For each A_i, its attention $\alpha_{i,j}$ to A_j can be calculated according to the following equation:

$$\alpha^{i,j} = \frac{Sum(\boldsymbol{Q}_i \otimes \boldsymbol{K}_j)}{\sum_{j=1}^{N_p} Sum(\boldsymbol{Q}_i \otimes \boldsymbol{K}_j)}. \quad (2)$$

With $\alpha_{i,j}$, A_i is converted to a feature vector $\bar{\boldsymbol{V}}_i$ as follows: $\bar{\boldsymbol{V}}_i = \sum_{j=1}^{N_p} \alpha_{i,j}\boldsymbol{V}_j$. After concatenating all $\bar{\boldsymbol{V}}_i$, a new feature vector $\bar{\boldsymbol{V}}^A$ is created from $\boldsymbol{A}$ for object tracking: $\bar{\boldsymbol{V}}^A = (\bar{\boldsymbol{V}}_1, ..., \bar{\boldsymbol{V}}_i, ..., \bar{\boldsymbol{V}}_{N_P})$. In Fig. 5, after the PSA block, $\bar{\boldsymbol{V}}^A$ is converted to a new feature vector $\boldsymbol{Z}^A$ by using a fully-connected network. Then, given two objects $\boldsymbol{A}$ and $\boldsymbol{B}$, with SLM, their similarity score can be measured by calculating the cosine similarity between $\boldsymbol{Z}^A$ and $\boldsymbol{Z}^B$.

3.2. Similarity Matching Cascade (SMC) for Target Tracking

Object association is the crucial step after similarity calculation for MOT. A well-designed association strategy can have a significant impact on tracking results such as HOTA [18]. In the literature, ByteTrack [50] is a simple yet effective method of association with objects, where detected boxes are classified by their confidence scores from high to low, and the best match in history is found based on the IOU criterion. Although ByteTrack achieves SoTA performance in some MOT evaluations (i.e., simple motion patterns), relying solely on the IOU distance for data association can result in frequent ID switches when visually similar targets approach each other (e.g., one occludes the other). To address this issue, we designed the SMC association method as shown in Fig. 6 that integrates the advantages of ByteTrack to achieve an improved trade-off between speed and accuracy.

Let $\mathbb{O}$ denote the set of objects detected by the PRB-Net from the current frame. All objects O_i in $\mathbb{O}$ are sorted according to their detection scores in descending order (the median detection score is μ). Subsequently, all objects O_i in $\mathbb{O}$ are divided into two sets: $\mathbb{O}^H$ and $\mathbb{O}^L$-based thresholding. Any object in $\mathbb{O}$ with a detection score higher than the threshold μ is placed in $\mathbb{O}^H$. If its detection score is lower than μ but higher than 0.1, it belongs to $\mathbb{O}^L$. We treat an object as background or noise if its detection score is below 0.1. Two different association strategies are employed to match elements in $\mathbb{O}^H$ and $\mathbb{O}^L$, respectively.

Let $\mathcal{T}$ represent the track list stored in the previous frame. Before matching, each track in $\mathcal{T}$ predicts its new position in the current frame using a Kalman filter. Moreover, $\mathcal{T}_i(k)$ denotes the kth fragment or tracklet of the ith track in $\mathcal{T}$, where $\mathcal{T}_i$(last) refers to the last fragment of $\mathcal{T}_i$. Furthermore, we use $S^H_{iou}(i,j)$ and $S^H_{app}(i,j)$ to denote the IOU similarity matrix and the appearance similarity matrix, respectively, between $\mathcal{T}_i$(last) and the jth object O_j in $\mathbb{O}^H$. The value of $S^H_{app}(i,j)$ is obtained using the SLM method as follows: $S^H_{app}(i,j) = SLM(\mathcal{T}_i(last), O_j)$. By integrating $S^H_{iou}(i,j)$ and $S^H_{app}(i,j)$ together, the similarity between $\mathcal{T}_i$ and the jth object O_j in $\mathbb{O}^H$ is calculated as follows:

$$S^H(i,j) = S^H_{iou}(i,j) + S^H_{app}(i,j). \tag{3}$$

Fig. 7 shows an example to calculate appearance similarity by the multi-templated SLM. Furthermore, we denote $S^L_{iou}(i,j)$ as the IOU similarity matrix between $\mathcal{T}_i$(last) and the j-th object O_j in $\mathbb{O}^L$. Similar to Eq. (3), the integrated similarity between $\mathcal{T}_i$ and the j-th object O_j in $\mathbb{O}^L$ is calculated as:

$$S^L(i,j) = S^L_{iou}(i,j) + S^L_{app}(i,j). \tag{4}$$

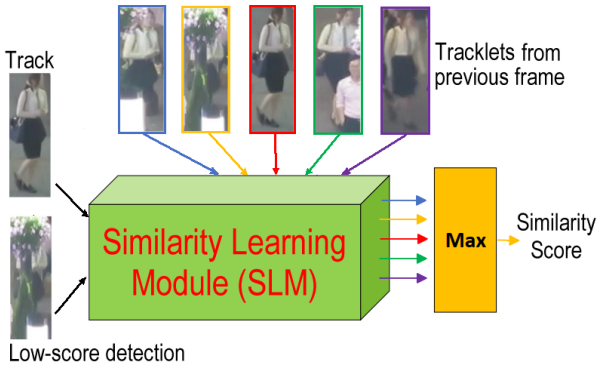

Figure 7. Appearance similarity between low-score detection at the current frame and tracks at the previous frame. Five tracklets compute a similarity score with the low-score detection using SLM. The most similar tracklet is selected, as indicated by the orange arrow in the figure.

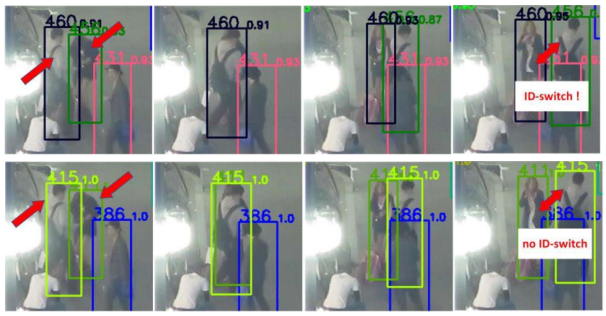

Figure 8. The use of a GATE function can better handle the occlusion and ID-switch problems in MOT. (a) Results of MOT without using the GATE function. When the two targets are getting closer and the IOU score is higher than the appearance score, an ID-switch problem happens. (b) Results of MOT using the GATE function.

Using $S^H(i,j)$ and $S^L(i,j)$, we initially associate the objects in $\mathbb{O}^H$ with tracklets in $\mathcal{T}_i$. However, due to occlusions or blur, some tracklets in $\mathbb{O}^H$ remain unmatched. To address this issue, we subsequently associate the objects in $\mathbb{O}^L$ with these unmatched tracklets, leading to State-of-The-Art (SoTA) MOT performance. The details of the SMC module are described below:

Stage I: During the first stage of association, our focus is on finding matches between $\mathbb{O}^H$ and $\mathcal{T}$. We employ the Hungarian algorithm to perform linear assignment using the similarity matrix $S^H(i,j)$. The unmatched objects of $\mathbb{O}^H$ and the unmatched tracks of $\mathcal{T}$ are then placed in $\mathbb{O}^H_{Remain}$ and $\mathcal{T}^H_{Remain}$, respectively.

Stage II: In the second matching stage, we match the objects in $\mathbb{O}^L$ to the tracklets in $\mathcal{T}^H_{Remain}$. We complete the linear assignment by the Hungarian algorithm with the similarity matrix S^L. The unmatched objects in $\mathbb{O}^L$ and the unmatched tracks in $\mathcal{T}^H_{Remain}$ are placed in $\mathbb{O}^L_{Remain}$ and $\mathcal{T}^L_{Remain}$.

Table 1. Comparison against the SoTA MOT methods on the MOT17 [21] test set.

Method	MOTA ↑	IDF1 ↑	HOTA ↑	FN ↓	FP ↓	IDs ↓	MT ↑	ML ↓	FPS ↑
Tube_TK [22]	63.0	58.6	48.0	177,483	27,060	4,137	31.2%	19.9%	3.0
MOTR [48]	65.1	66.4	-	149,307	45,486	2,049	33.0%	25.2%	-
CTracker [24]	66.6	57.4	-	160,491	22,284	5,529	-	-	-
CenterTrack [55]	67.8	64.7	52.2	160,332	18,498	3,039	34.6%	24.6%	17.5
QuasiDense [23]	68.7	66.3	53.9	146,643	26,589	3,378	40.6%	29.1%	20.3
TraDes [44]	69.1	63.9	-	150,060	20,892	3,555	-	-	-
MAT [13]	69.5	63.1	53.8	138,741	30,660	2,844	43.8%	18.9%	9.0
SOTMOT [53]	71.0	71.9	-	118,983	39,537	5,184	42.7%	15.3%	16.0
GSDT [40]	73.2	66.5	-	120,666	26,397	3,891	-	-	-
FairMOT [51]	73.7	72.3	59.3	117,477	27,507	3,303	43.2%	17.3%	25.9
RelationTrack [47]	73.8	74.7	-	118,623	27,999	1,374	-	-	-
PermaTrackPr [35]	73.8	68.9	-	115,104	28,998	3,699	-	-	-
CSTrack [14]	74.9	72.6	59.3	114,303	23,847	3,567	41.5%	17.5%	15.8
TransTrack [34]	75.2	63.5	54.1	86,442	50,157	3,603	55.3%	**10.2**%	10.0
SiamMOT [41]	76.3	72.3	-	-	-	-	-	-	-
TransCenter [46]	76.4	65.4	-	89,712	37,005	6,402	51.7%	11.6%	-
CorrTracker [39]	76.5	73.6	60.7	99,510	29,808	3,369	47.6%	12.7%	15.6
TransMOT [17]	76.7	75.1	-	93,150	36,231	2,346	-	-	-
ReMOT [52]	77.0	72.0	-	93,612	33,204	2,853	-	-	-
OCSORT [5]	78.0	77.5	-	107,055	**15,129**	1,950	-	-	-
MAATrack [33]	79.4	75.9	62.0	77,661	37,320	1,452	-	-	-
StrongSORT++ [10]	79.6	79.5	64.4	86,205	27,876	**1,194**	53.6%	13.9%	7.1
ByteTrack [50]	80.3	77.3	63.1	83,721	25,491	2,196	53.2%	14.5%	29.6
BoT-SORT [1]	80.6	79.5	64.6	85,398	22,524	1,257	-	-	6.6
SMILEtrack w/o Re-ID (Ours)	80.7	80.1	65.0	81,792	23,187	1,251	54.7%	14.2%	**37.5**
BoT-SORT-R [1]	80.5	80.2	65.0	86,037	22,521	1,212	-	-	4.5
SMILEtrack (Ours)	**81.1**	**80.5**	**65.3**	**79,428**	22,963	1,246	**56.3**%	14.7%	5.6

3.3. The SMC GATE Function

To calculate the similarity score, most MOT methods use a weighted sum to combine the IOU and the appearance information to improve the accuracy of data association. However, this approach can cause problems when the IOU score is significantly higher than the appearance similarity score between two distinct pedestrians, as they may only overlap, but are not the same. To address this problem, we introduce a GATE function in the SMC module to reject a target if its appearance similarity score is low, even when it comes with a high IOU score.

Due to occlusions or lighting changes, objects in $\mathbb{O}^H_{Remain}$ with higher scores may not be matched in the current frames, but their correspondences may potentially be found in future frames. If a target in $\mathbb{O}^H_{Remain}$ passes the GATE function check, the SMC module will generate a new tracklet and add it to $\mathcal{T}$ for further matching. The GATE function uses a threshold τ to select objects from $\mathbb{O}^H_{Remain}$ if their detection scores are higher than τ and include them in $\mathcal{T}$ as new tracks for further association. Objects in $\mathbb{O}^H_{Remain}$ with detection scores lower than τ, as well as those in $\mathbb{O}^L_{Remain}$, are considered background and filtered out. It is important to note that tracks in $\mathcal{T}^L_{Remain}$ are deleted if they remain unmatched for more than 30 frames.

This GATE function is a novel addition not present in ByteTrack [50], and it aims to re-select potential tracks from $\mathbb{O}^H_{Remain}$ to handle challenging scenarios involving severe occlusions. Without this GATE function, ByteTrack cannot determine whether the objects to be matched are seriously occluded or not. Fig. 8 and Table 3 shows the advantage of the GATE function.

4. Experimental Results

Implementation Details. Our experiments were conducted on MOT17 and MOT20 benchmarks [21], with additional training on datasets [8,11,21,30,31,45,49,54]. For re-ID models, datasets providing both bounding box location and identity information, such as CalTech [8], PRW [54], and CUHK-SYSU [45], were used. Evaluation metrics [21] included MOTA [2], IDF1 [29], and HOTA [18], highlighting detection performance and identity matching. Our detector was initialized on the COCO dataset [15] and fine-tuned on MOT datasets, employing data augmentation and an SGD optimizer with cosine annealing. The SMC module introduced a GATE function to manage new tracklets, with key parameters assessed in an ablation study. Additional details regarding the effects of track buffer, template lengths, and patch layout can be found in the supplementary.

Table 2. Comparison against the SoTA methods on the MOT20 [7] test set.

Method	MOTA ↑	IDF1 ↑	HOTA ↑	FN ↓	FP ↓	IDs ↓	FPS ↑
FairMOT [51]	61.8	67.3	54.6	103,440	88,901	5,243	13.2
CSTrack [14]	66.6	68.6	54.0	144,358	25,404	3,196	4.5
TransTrack [34]	65.0	59.4	48.5	150,197	27,197	3,608	7.2
TransCenter [46]	61.9	50.4	-	146,347	45,895	4,653	1.0
CorrTracker [39]	65.2	69.1	-	95,855	79,429	5,183	8.5
GSDT [40]	67.1	67.5	53.6	135,409	31,913	3,131	0.9
SiamMOT [41]	67.1	69.1	-	-	-	-	4.3
RelationTrack [47]	67.2	70.5	56.5	104,597	61,134	4,243	2.7
SOTMOT [53]	68.6	71.4	-	101,154	57,064	4,209	8.5
MAATrack [33]	73.9	71.2	57.3	108,744	24,942	1,331	14.7
StrongSORT++ [10]	73.8	77.0	62.6	117,920	**16,632**	**770**	-
OCSORT [5]	75.7	76.3	62.4	105,894	19,067	942	-
TransMOT [17]	77.5	75.2	-	**80,788**	34,201	1615	-
ByteTrack [50]	77.8	75.2	61.3	87,594	26,249	1,223	17.5
BoT-SORT [1]	77.7	76.3	62.6	86,037	22,521	1,212	6.6
SMILEtrack w/o Re-ID (Ours)	78.0	76.3	63.0	86,112	23,246	1,208	**22.9**
BoT-SORT-R [1]	77.8	77.5	63.3	88,863	24,638	1,257	2.4
SMILEtrack(Ours)	**78.2**	**77.5**	**63.4**	85,548	24,554	1,318	7.2

4.1. Evaluation Results

Table 1 presents the evaluation comparisons of our SMILEtrack with State-of-The-Art (SoTA) tracking models on the MOT17 test set, following the evaluation of the MOTChallenge [21]. All evaluation results were obtained using the official MOTChallenge evaluation website. SMILEtrack outperforms all SoTA methods in several metrics, namely MOTA, IDF1, HOTA, FN, and MT, respectively. Note that the MOT community pays particular attention to the compound metrics MOTA and IDF1. Additionally, in the MOT17 dataset, SMILEtrack is the only method to achieve an IDF1 score higher than 80. ByteTrack [50] shows high efficiency among SoTA methods, but also exhibits higher false positive and false negative rates. On the other hand, StrongSORT++ [10] achieves the lowest false negatives but with significantly higher false positives.

Our SMILEtrack is the only one method that achieves a score higher than 80 in the IDF1 metric on the MOT17. ByteTrack [50] is the most efficient among all the SoTA methods but with higher IDs and FN. StrongSORT++ [10] obtains the lowest IDs but with a much higher FN. Table 2 presents comparisons of our SMILEtrack with the SoTA methods on the MOT20 test set. SMILEtrack surpasses all SoTA methods in the MOTA, IDF1, HOTA, and FN metrics on MOT20. ByteTrack [50] remains the most efficient MOT method, while StrongSORT++ [10] achieves the lowest false positives, but still with a much higher FN.

4.2. Ablation Studies

4.2.1 Effects of Patch Layouts

Different patch arrangements will affect the performance of SLM. Therefore, the first ablation study aims to investigate the effects of different patch layouts on SLM performance improvements. Fig. 4 shows different types of patch layouts. Table 4 shows the effects of different patch layouts on performance improvements evaluated on the MOT17 val set [21]. As shown in Fig. 4, the type-E patch layout outperforms others with respect to the metrics MOTA, IDF1, and IDs. This paper adopts the type-E patch layout for all performance evaluation and comparison.

4.2.2 Re-Identification Strategies.

Traditional methods primarily rely on IOU to calculate similarity scores [50]. These methods often fail to track rapidly moving objects effectively due to the lack of appearance matching, leading to an increase in identity switches. As shown in Table 3, SMILEtrack with PRB-Net [6] outperforms ByteTrack [50] with YOLOX [12] in terms of all metrics and efficiency because ByteTrack encounters issues regarding re-identification. Noteably, when ByteTrack is incorporated with our methods, such as the SLM, SMC, and GATE functions, its accuracy improves substantially to the level comparable to SMILEtrack, which justifies the effectiveness of SLM, SMC, and GATE.

Table 3. Ablation analysis of SLM, SMC, and GATE Function (GF) on the MOT17 validation set, compared to the leading ByteTrack [50] that utilizes the YOLOX [12] detector. The FPS encompasses detection, NMS, re-identification, and data association, excluding image acquisition and video encoding/decoding processes.

Method	Detector	SLM	SMC	GF	MOTA ↑	IDF1 ↑	IDs ↓	FPS ↑
ByteTrack	YOLOX				74.1	77.0	803	**9.7**
SMILEtrack	YOLOX	✓			76.2	78.4	647	8.1
SMILEtrack	YOLOX	✓	✓		76.9	79.1	594	8.0
SMILEtrack	YOLOX	✓	✓	✓	**77.5**	**79.9**	**554**	7.5
SMILEtrack	PRB-Net				75.3	77.5	856	**10.2**
SMILEtrack	PRB-Net	✓			77.6	79.3	601	8.5
SMILEtrack	PRB-Net	✓	✓		78.2	80.2	543	8.2
SMILEtrack	PRB-Net	✓	✓	✓	**78.6**	**80.8**	**509**	7.8

Table 4. Effects of patch layouts on performance improvement evaluated on the MOT17 val set.

Method	A	B	C	D	E	F	G	H
MOTA↑	76.0	76.1	76.2	76.4	**76.4**	76.3	76.4	76.4
IDF1↑	77.4	77.6	77.7	77.9	**78.4**	78.2	78.4	78.3
IDs↓	732	705	681	654	**624**	645	633	630

Table 5. Performance comparisons of similarity scores on the MOT17 validation set.

SMC Stage I	SMC Stage II	MOTA ↑	IDF1 ↑	IDs ↓
IOU	IOU	76.2	74.0	731
SLM + IOU	IOU	**76.5**	78.8	615
IOU	SLM+IOU	76.1	73.7	740
SLM+IOU	SLM+IOU	76.4	78.5	624
SLM+IOU	SLM (Multi)+IOU	**76.5**	**78.9**	**585**

4.2.3 Combination of SLM and IOU for different Stages.

Table 5 presents performance comparisons for different combinations of IOU and appearance features in Stages I and II of the SMC module. Objects in $\mathbb{O}^H$ generally have higher detection scores and fewer occlusions. Using both the IOU and the appearance features in stage I, SMILEtrack achieves improved MOTA, IDF1, and IDs, as in Table 5 rows 2 and 3. Objects in $\mathbb{O}^L$ are more prone to occlusions or motion blurring. Thus, SMC Stage II relying solely on IOU yields the best results, as shown in Table 5 rows 3 and 5. When multiple templates are created in the SLM to address the issue of low detection scores, the best results are obtained by combining SLM (multiple templates) with IOU, as shown in the last row in Table 5.

5. Conclusion

In this paper, we propose SMILEtrack, a Siamese network-like architecture that effectively learns object appearance features for single-camera multiple-object tracking. We introduce the Similarity Matching Cascade (SMC) for bounding box association in each frame, and our experiments demonstrate that SMILEtrack achieves high-performance scores in terms of MOTA, IDF1, IDs, and FPS on the MOT17 and MOT20 datasets.

Future work. As SMILEtrack is a Separate Detection and Embedding (SDE) method, it has a slower runtime compared to Joint Detection and Embedding (JDE) methods. In the future, we plan to explore approaches that can improve the efficiency *versus* accuracy trade-off in MOT tasks.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv:2206.14651*, 2021. 1, 3, 6, 7

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP JIVP*, 2018. 2, 6

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *ICIP*, Sep 2016. 1, 3

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hongyuan Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*, 2020. 2

[5] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv:2203.14360*, 2022. 6, 7

[6] Ping-Yang Chen, Ming-Ching Chang, Jun-Wei Hsieh, and Yong-Sheng Chen. Parallel residual bi-fusion feature pyramid network for accurate single-shot object detection. *IEEE Transactions on Image Processing*, 30:9099–9111, 2021. 2, 3, 7, 11

[7] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 7

[8] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009. 6

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020. 2, 4

[10] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023. 1, 6, 7

[11] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 6

[12] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv:2107.08430*, 2021. 3, 7, 8

[13] Jie Li, Pavel Tokmakov, and Adrien Gaidon. Mat:: Motion-aware multi-object tracking. *Neurocomputing*, 476:104–114, 2021. 6

[14] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multiobject tracking. *IEEE Transactions on Image Processing*, 30:7188–7200, 2021. 6, 7

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 6

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *ECCV*, page 21–37, 2016. 2

[17] Zhenbo Liu, Lijun Wang, Zhihong Wang, and Wan-Chi Siu. TransMOT: Spatial-temporal graph transformer for multiple object tracking. In *ICCV*, page 10002–10011, 2021. 6, 7

[18] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *IJCV*, 129:548–578, 2021. 5, 6

[19] Chenxu Luo, Chang Ma, Chunyu Wang, and Yizhou Wang. Learning discriminative activated simplices for action recognition. In *AAAI*, January 2017. 1

[20] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-object tracking with transformers. In *CVPR*, 2022. 3

[21] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016. 3, 6, 7

[22] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *CVPR*, pages 6308–6318, 2020. 6

[23] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, page 164–173, 2021. 6

[24] Jialian Peng, Liangliang Wang, Fangbin Wan, Yabiao Wu, Yichen Chen, and Ying Tai. Chained-Tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *ECCV*, pages 144–161. Springer, 2020. 6

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *arXiv:1506.02640*, 2016. 2

[26] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *CVPR*, pages 6517–6525, 2017. 2

[27] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv:1804.02767*, 2018. 2

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv:1506.01497*, 2016. 2

[29] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV Workshops*, 2016. 6

[30] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 6

[31] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv:1805.00123*, 2018. 6

[32] Daniel Stadler and Jurgen Beyerer. On the performance of crowd-specific detectors in multi-pedestrian tracking. In *AVSS*, page 1–12, 2021. 3

[33] Daniel Stadler and Jürgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *WACV*, pages 133–142, January 2022. 3, 6, 7

[34] Peize Sun, Yi Jiang, Zhang Rufeng, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv:2012.15460*, 2020. 3, 6, 7

[35] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *ICCV*, page 10012–10021, 2021. 6

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 4

[37] Chunyu Wang, Yizhou Wang, and Alan L. Yuille. An approach to pose-based action recognition. In *CVPR*, pages 915–922, 2013. 1

[38] Chien-Yao Wang et al. CSPNet: A new backbone that can enhance learning capability of cnn. In *CVPR Workshops*, June 2020. 4

[39] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *CVPR*, pages 3876–3886, 2021. 6, 7

[40] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. In *ICRA*, pages 10077–10083. IEEE, 2021. 6, 7

[41] Zhenbo Wang, Zhuoling Li, Shoudong Han, and Hongwei Wang. One more check: Making "fake background" be tracked again. In *AAAI*, volume 35, pages 15446–15454, 2021. 6, 7

[42] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, 2020. 1

[43] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649, 2017. 3

[44] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, pages 12352–12361, June 2021. 6

[45] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. *arXiv:1604.01850*, 2017. 6

[46] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense representations for multiple-object tracking. *IEEE PAMI*, 2021. 3, 6, 7

[47] En Yu, Zhuoling Li, Shoudong Han, and Hongwei Wang. Relationtrack: Relation-aware multiple object tracking with decoupled representation. In *ICIP*, pages 3004–3008, 2021. 6, 7

[48] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-end multiple-object tracking with transformer. In *ICCV*, pages 11076–11085, 2021. 6

[49] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. CityPersons: A diverse dataset for pedestrian detection. In *CVPR*, pages 4457–4465, 2017. 6

[50] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *arXiv:2110.06864*, 2021. 2, 3, 5, 6, 7, 8, 11

[51] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 129:3069–3087, 2021. 1, 3, 6, 7

[52] Zhenbo Zhang, Lijun Wang, Zhihong Wang, and Wan-Chi Siu. Remot: A model-agnostic refinement for multiple object tracking. *Image and Vision Computing*, 105:104067, 2020. 6

[53] Linyu Zheng, Ming Tang, Yingying Chen, Guibo Zhu, Jinqiao Wang, and Hanqing Lu. Improving multiple object tracking with single object tracking. In *CVPR*, pages 2453–2462, June 2021. 6, 7

[54] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. *arXiv:1604.02531*, 2017. 6

[55] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, pages 474–490. Springer, 2020. 6

[56] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv:1904.07850*, 2019. 3

[57] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *IICLR*, 2021. 3

A. Details of SMC pipeline

Fig. 6 shows the Similarity Matching Cascade (SMC) pipeline. The SMC is a hybrid association technique that balances the matching accuracy and the running speed. In comparison, ByteTrack [50] utilizes a simple method that sorts detected boxes by confidence and matches based on IOU. Despite its stellar performance in certain MOT cases, the IOU-only association can falter when similar objects converge.

Objects detected by PRB-Net [6] in the current frame are denoted as $\mathbb{O}$ and sorted by detection scores. They are split into high ($\mathbb{O}^H$) and low ($\mathbb{O}^L$) score sets based on a median score (μ) threshold. Those below a 0.1 score are viewed as noise. Different matching strategies are used for these two sets.

Tracks from the previous frames are stored in $\mathcal{T}$. Each track predicts its next position using a Kalman filter. We compute similarity matrices $S^H_{iou}(i,j)$ and $S^H_{app}(i,j)$ for high-score objects and tracks, and combine them by Eq. 3. For low-score objects, due to possible occlusions, the appearance matrix is computed differently by Eq. 4. The initial associations between $\mathbb{O}^H$ and $\mathcal{T}_i$ use these matrices. Unmatched high-score detections are later paired with low-score objects to maximize MOT performance.

A.1. Stages of SMC Module

- Stage I: Linear assignment matches between $\mathbb{O}^H$ and $\mathcal{T}$ using the Hungarian algorithm.
- Stage II: Objects in $\mathbb{O}^L$ are paired with the remaining tracks in $\mathcal{T}^H_{Remain}$.

A.2. SMC GATE Function

Most MOT techniques merge IOU and appearance data with a fixed weight for easy calculation of the similarity score. This can be misleading when overlapping, distinct objects have high IOU but different appearances. The GATE function in the SMC module filters out such mismatches.

Unmatched objects with high scores might find matches in subsequent frames due to occlusions or lighting changes. If an object passes the GATE function, it starts a new tracklet in $\mathcal{T}$. This GATE function, absent in ByteTrack, reselects tracks from $\mathbb{O}^H_{Remain}$ during severe occlusions. Its benefits are evident in Fig. 6 and ensuing results.

B. Supplementary Experimental Results

B.1. Effects of Track Buffer and Template Lengths

We conducted an investigation into scenarios where an object remains consistently undetected within a video sequence. Our study aimed to determine the optimal duration for retaining such objects within the tracklet list, as well as the ideal number of templates to be stored for effective tracking. The outcomes of this analysis are presented in Table 6, which highlights the impact of track buffer length on performance improvement. Evidently, the most favorable results are achieved with a buffer length of 20. Consequently, this research establishes the buffer length as 20 for subsequent experiments. Furthermore, Table 7 shows the influence of the number of templates on performance enhancement. The adoption of 30 templates per object is observed to yield the highest MOTA, IDF1, and ID metrics. Therefore, this study sets the number of templates at 30 for consistent application.

Track Buffer	MOTA ↑	IDF1 ↑	IDS ↓
10	76.3	78.0	642
20	**76.4**	**78.5**	**624**
30	76.4	78.3	633

Table 6. Effects of object history length on performance improvement.

Number of templates	MOTA ↑	IDF1 ↑	IDS ↓
10	76.2	78.1	665
20	76.3	78.2	645
30	**76.4**	**78.5**	**624**

Table 7. Performance comparisons regarding the number of object templates.

Feature dim	MOTA ↑	IDF1 ↑	IDS ↓
64	76.3	78.1	**621**
128	**76.4**	**78.5**	624
256	76.3	78.1	645

Table 8. Performance comparisons regarding feature dimensions.

B.2. Effects of Feature Dimension

To re-track an object, an similarity learning module (SLM) is created in our SMILE-track. Table 8 tabulates the effects of feature dimension on Re-Identification performance. Clearly, the feature dimension of 64 leads to the best IDs, while the dimension of 128 leads to the best MOTA and IDF1. To maximize the overall MOT performance, the feature dimension 128 is chosen in SLM for representing a tracked target.

B.3. Effects of GATE Function and Multi-template SLM.

The SMC GATE function is introduced to select unmatched boxes with high detection scores from $\mathbb{O}^H_{Remain}$ to form tracklets for further association. To address the issue of unreliable features in low-score detection boxes, we

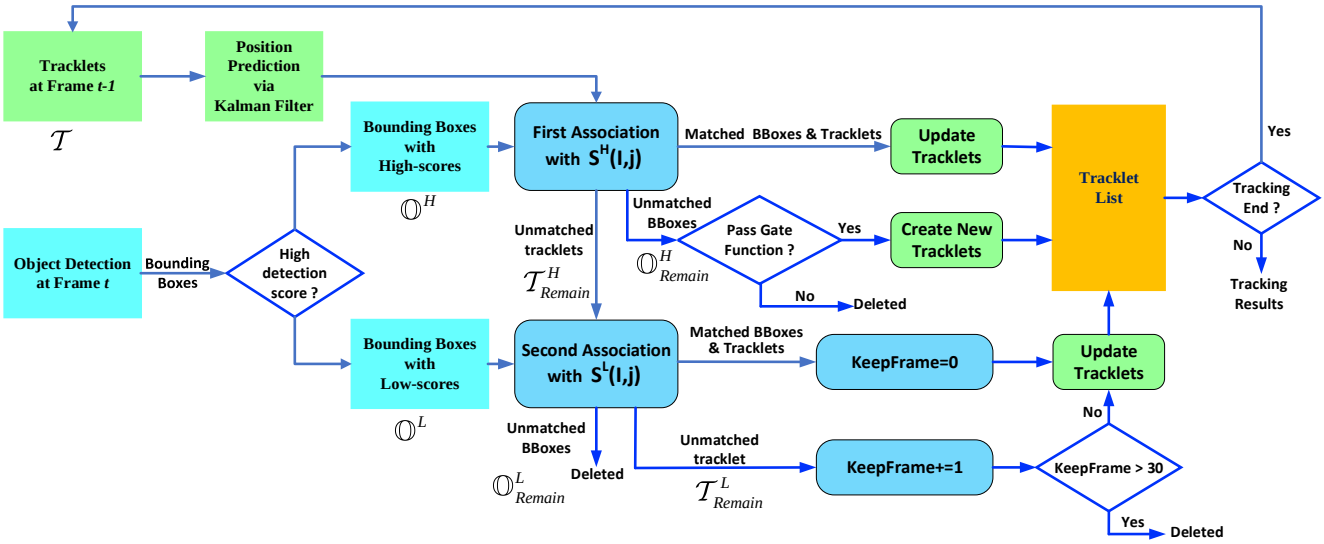

Figure 9. The **Similarity Matching Cascade (SMC)** pipeline.

SMC Stages I and II	Gate Function	SLM-Multiple Templates	MOTA	IDF1	IDS
SLM + IOU			76.4	78.5	624
SLM + IOU	√		76.5	78.7	601
SLM + IOU		√	76.5	78.9	585
SLM + IOU	√	√	**76.6**	**79.2**	**545**

Table 9. Ablation study on the MOT17 validation set for different strategies.

utilize the multi-template SLM mechanism to enhance the robustness of data association. Table 9 presents the ablation studies examining the effects of the GATE function and the multi-template SLM on performance improvements. The best performance is achieved when both the GATE function and the SLM with multiple templates are employed.