SMILEtrack: SiMIlarity LEarning for Occlusion-Aware Multiple Object Tracking

Yu-Hsiang Wang[1], Jun-Wei Hsieh[1*], Ping-Yang Chen[2], Ming-Ching Chang[3], Hung-Hin So[4], Xin Li[3]

[1]College of Artificial Intelligence and Green Energy, National Yang Ming Chiao Tung University, Taiwan
[2]Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
[3]Department of Computer Science, University at Albany - SUNY, USA
[4]The Chinese University of Hong Kong, China
j122333221@gmail.com, [jwhsieh, pingyang.cs08]@nycu.edu.tw, mchang2@albany.edu, Xin.Li@mail.wvu.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 tradeoff 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.

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 (Wang, Wang, and Yuille 2013; Luo et al. 2017), smart video analysis, and autonomous driving. Modern MOT systems (Bewley et al. 2016; Wang et al. 2020b) 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.

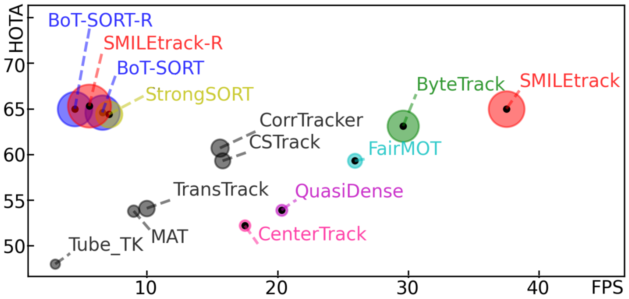

Figure 1: Comparative analysis of HOTA-MOTA-FPS for different trackers on the MOT17 test set. X-axis: FPS (running speed). Y-axis: HOTA. Circle radius: MOTA score. SMILEtrack registers 80.7 MOTA and 65.0 HOTA at 37.5 FPS, exceeding all other trackers (see Table 1 for details).

Despite enormous efforts in MOT investigation, the task remains challenging due to vague objects, occlusion, and complex scenes in real-world applications.

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 (Wang et al. 2020b; Zhang et al. 2021c) 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 (Bewley et al. 2016; Du et al. 2023; Aharon, Orfaig, and Bobrovsky 2021) require a detector and a re-identification model. The detector locates all objects in a single frame via bounding boxes (Ren et al. 2016; Liu et al. 2016; Redmon et al. 2016; Redmon and Farhadi 2017, 2018; Bochkovskiy, Wang, and Liao 2020; Chen et al. 2021). 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 Tracking-by-Attention (TbA) paradigm (Zhang et al. 2021a; Peng et al. 2021; Yang et al. 2021; Li et al. 2021) applies attention to data associations and jointly performs tracking and detection via Transformer (Vaswani et al. 2017b).

*Corresponding author.

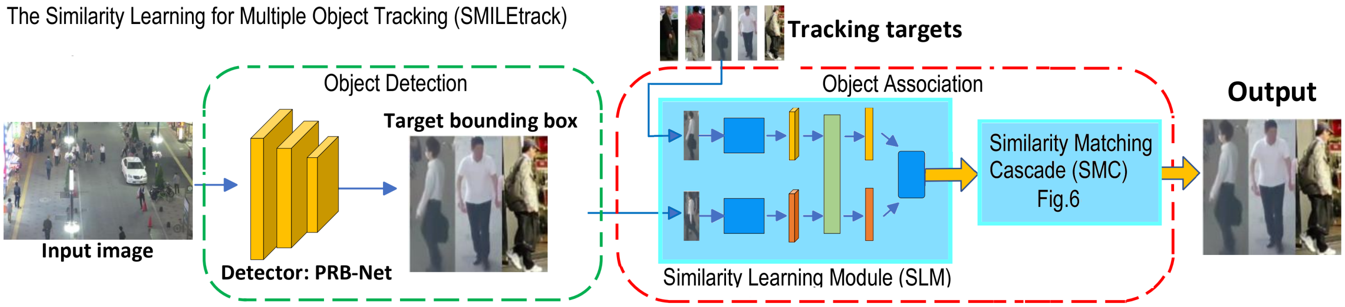
The Similarity Learning for Multiple Object Tracking (SMILEtrack)
Tracking targets
Object Detection
Target bounding box
Input image
Detector: PRB-Net
Object Association
Similarity Learning Module (SLM)
Similarity Matching Cascade (SMC) Fig.6
Output

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.

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 (Zhang et al. 2021b) 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 (Bernardin and Stiefelhagen 2018)).

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 (Dosovitskiy et al. 2020), 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 (Zhang et al. 2021b) 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.

Related Work

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 (Ren et al. 2016) offer high accuracy but at the cost of speed. On the contrary, one-stage methods are faster but less accurate. YOLO object detection models (Redmon et al. 2016; Redmon and Farhadi 2017, 2018; Bochkovskiy, Wang, and Liao 2020) 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 (Zhou, Wang, and Krähenbühl 2019), and YOLOX (Ge et al. 2021) have emerged. Despite their improvements (Zhang et al. 2021b; Aharon, Orfaig, and Bobrovsky 2021), these tracking devices still struggle to accurately detect objects of varying sizes. PRB-Net (Chen et al. 2021) is an effective object detector for MOT tasks, addressing the limitations of anchor-based and anchor-free detectors.

Object Association: SORT (Bewley et al. 2016) 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 (Wojke, Bewley, and Paulus 2017) alleviates occlusion issues by incorporating CNN-based appearance features; however, this compromises execution speed. To address this efficiency issue, FairMOT (Zhang et al. 2021c) employs an anchor-free method based on CenterNet (Zhou, Wang, and Krähenbühl 2019), which significantly improves the MOT performance on the MOT17 dataset. To improve tracking efficiency, numerous MOT methods (Stadler and Beyerer 2021, 2022) ignore the appearance features of objects, instead leveraging high-performance detectors and motion cues. Despite achieving impressive results and fast inference on MOTChallenge (Milan et al. 2016) 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.

Tracking-by-Attention

Trackformer (Meinhardt et al. 2022) 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 (Sun et al. 2020) uses an attention-based query-key mechanism to perform object detection and association in a single shot based on Deformable DETR (Zhu et al. 2021). TransCenter (Xu et al. 2021) 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.

Methodology

We introduce **Similarity Learning for Multiple Object Tracking (SMILEtrack)**, a novel MOT architecture integrating a detector (Chen et al. 2021) 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 YOLOX (Ge et al. 2021). 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.

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 (Wojke, Bewley, and Paulus 2017) 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.

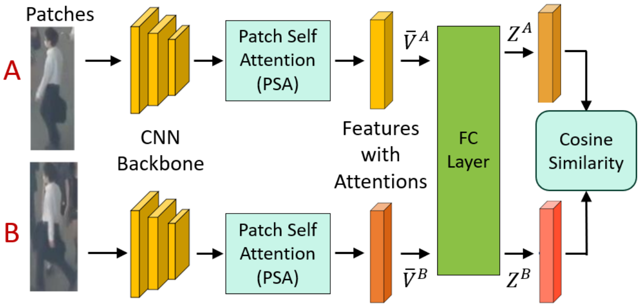

Figure 3: Appearance similarity between low-score detection at the current frame and tracks at the previous frame.

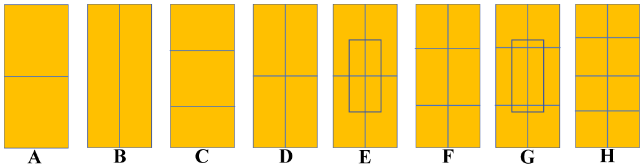

Figure 4: Different types of patch layout: configuration (E) achieves the best performance because it can actively attend to PSA-occluded parts when occlusion occurs.

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).

Patch Self-Attention (PSA) Block To produce a reliable appearance feature, a superior feature representation is essential. Inspired by the Vision Transformer (VIT) (Dosovitskiy et al. 2020), 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

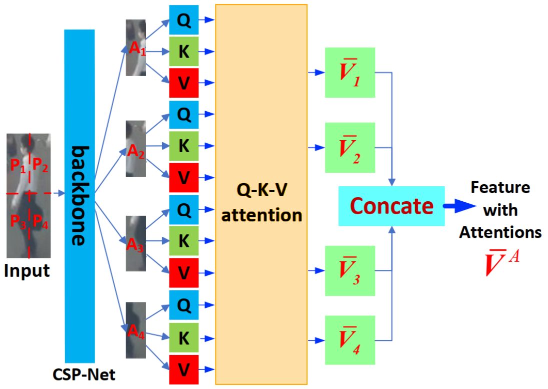

Figure 5: The Patch Self-Attention (PSA) architecture.

detailed architecture of the PSA block is shown in Fig. 5.

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 (Vaswani et al. 2017a). 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 (Wang et al. 2020a) 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 (Wang et al. 2020a).

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:

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

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 A_i, its attention $\alpha_{i,j}$ to A_j can be calculated according to the following equation:

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

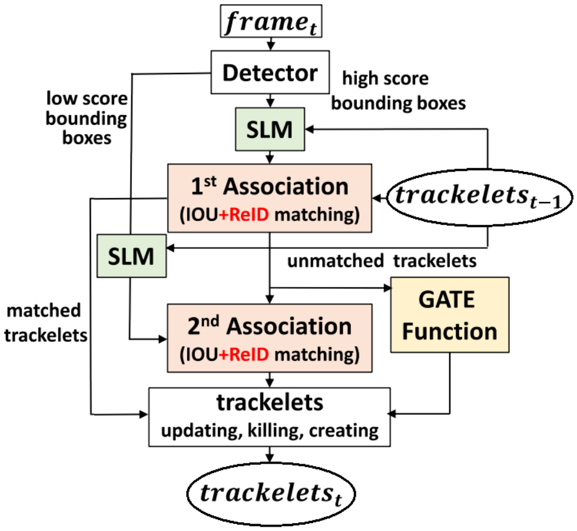
$frame_t$
Detector
high score bounding boxes
low score bounding boxes
SLM
1st Association (IOU+ReID matching)
$trackelets_{t-1}$
SLM
unmatched trackelets
matched trackelets
GATE Function
2nd Association (IOU+ReID matching)
trackelets updating, killing, creating
$trackelets_t$

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

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 $\boldsymbol{Z}^B$.

Similarity Matching Cascade (SMC) for 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 (Luiten et al. 2021). In the literature, ByteTrack (Zhang et al. 2021b) 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

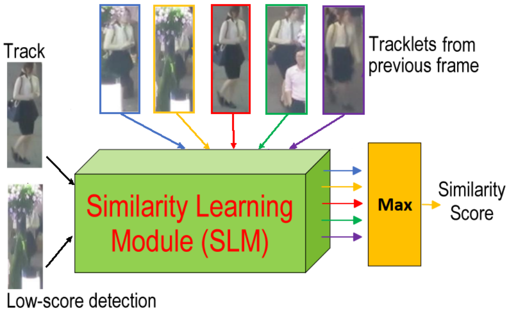
Track
Tracklets from previous frame
Similarity Learning Module (SLM)
Max
Similarity Score
Low-score detection

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.

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. Let $S^L_{iou}(i,j)$ be 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:

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

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 $\mathcal{T}^L_{Remain}$.

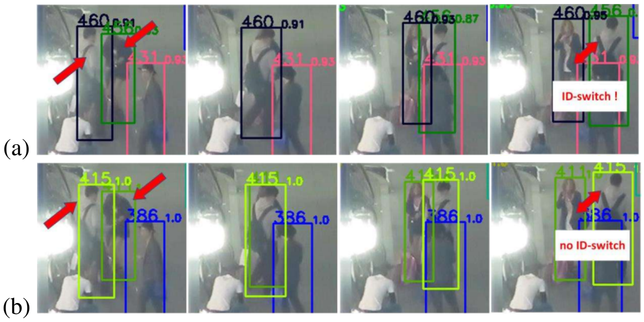

Figure 8: The use of a GATE function can better handle the occlusion and ID-switch problems in MOT. (a) Results of MOT without using the GATE function. When the two targets are getting closer and the IOU score is higher than the appearance score, an ID-switch problem happens. (b) Results of MOT using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

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 method 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. This paper introduces 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 (Zhang et al. 2021b), 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.

Experimental Results

Implementation Details. Our experiments were conducted on MOT17 and MOT20 benchmarks (Milan et al. 2016), with additional training on datasets (Schöps et al. 2017;

Method	MOTA ↑	IDF1 ↑	HOTA ↑	FN ↓	FP ↓	IDs ↓	MT ↑	ML ↓	FPS ↑
Tube_TK (Pang et al. 2020)	63.0	58.6	48.0	177,483	27,060	4,137	31.2%	19.9%	3.0
MOTR (Zeng et al. 2021)	65.1	66.4	-	149,307	45,486	2,049	33.0%	25.2%	-
CTracker (Peng et al. 2020)	66.6	57.4	-	160,491	22,284	5,529	-	-	-
CenterTrack (Zhou, Koltun, and Krähenbühl 2020)	67.8	64.7	52.2	160,332	18,498	3,039	34.6%	24.6%	17.5
QuasiDense (Pang et al. 2021)	68.7	66.3	53.9	146,643	26,589	3,378	40.6%	29.1%	20.3
TraDes (Wu et al. 2021)	69.1	63.9	-	150,060	20,892	3,555	-	-	-
MAT (Li, Tokmakov, and Gaidon 2021)	69.5	63.1	53.8	138,741	30,660	2,844	43.8%	18.9%	9.0
SOTMOT (Zheng et al. 2021)	71.0	71.9	-	118,983	39,537	5,184	42.7%	15.3%	16.0
GSDT (Wang, Kitani, and Weng 2021)	73.2	66.5	-	120,666	26,397	3,891	-	-	-
FairMOT (Zhang et al. 2021c)	73.7	72.3	59.3	117,477	27,507	3,303	43.2%	17.3%	25.9
RelationTrack (Yu et al. 2021)	73.8	74.7	-	118,623	27,999	1,374	-	-	-
PermaTrackPr (Tokmakov et al. 2021)	73.8	68.9	-	115,104	28,998	3,699	-	-	-
CSTrack (Liang et al. 2021)	74.9	72.6	59.3	114,303	23,847	3,567	41.5%	17.5%	15.8
TransTrack (Sun et al. 2020)	75.2	63.5	54.1	86,442	50,157	3,603	55.3%	**10.2**%	10.0
SiamMOT (Wang et al. 2021b)	76.3	72.3	-	-	-	-	-	-	-
TransCenter (Xu et al. 2021)	76.4	65.4	-	89,712	37,005	6,402	51.7%	11.6%	-
CorrTracker (Wang et al. 2021a)	76.5	73.6	60.7	99,510	29,808	3,369	47.6%	12.7%	15.6
TransMOT (Liu et al. 2021)	76.7	75.1	-	93,150	36,231	2,346	-	-	-
ReMOT (Zhang et al. 2020)	77.0	72.0	-	93,612	33,204	2,853	-	-	-
OCSORT (Cao et al. 2022)	78.0	77.5	-	107,055	**15,129**	1,950	-	-	-
MAATrack (Stadler and Beyerer 2022)	79.4	75.9	62.0	77,661	37,320	1,452	-	-	-
StrongSORT++ (Du et al. 2023)	79.6	79.5	64.4	86,205	27,876	**1,194**	53.6%	13.9%	7.1
ByteTrack (Zhang et al. 2021b)	80.3	77.3	63.1	83,721	25,491	2,196	53.2%	14.5%	29.6
BoT-SORT (Aharon, Orfaig, and Bobrovsky 2021)	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 (Aharon, Orfaig, and Bobrovsky 2021)	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

Table 1: Comparison against the SoTA MOT methods on the MOT17 (Milan et al. 2016) test set.

Dollár et al. 2009; Milan et al. 2016; Zhang, Benenson, and Schiele 2017; Shao et al. 2018; Ess et al. 2008; Xiao et al. 2017; Zheng et al. 2017). For re-ID models, datasets providing both bounding box location and identity information, such as CalTech (Dollár et al. 2009), PRW (Zheng et al. 2017), and CUHK-SYSU (Xiao et al. 2017), were used. Evaluation metrics (Milan et al. 2016) included MOTA (Bernardin and Stiefelhagen 2018), IDF1 (Ristani et al. 2016), and HOTA (Luiten et al. 2021), highlighting detection performance and identity matching. Our detector was initialized on the COCO dataset (Lin et al. 2014) 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.

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 (Milan et al. 2016). 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 (Zhang et al. 2021b) shows high efficiency among SoTA methods, but also exhibits higher false positive and false negative rates. On the other hand, StrongSORT++ (Du et al. 2023) 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 is the most efficient among all the SoTA methods but with higher IDs and FN. StrongSORT++ 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 remains the most efficient MOT method, while StrongSORT++ achieves the lowest false positives, but still with a much higher FN.

Ablation Studies

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 (Milan et al. 2016). As shown in Fig. 4, the type-E patch layout outperforms others with respect to the metrics MOTA, IDF1, and IDs. This paper

Method	MOTA ↑	IDF1 ↑	HOTA ↑	FN ↓	FP ↓	IDs ↓	FPS ↑
FairMOT (Zhang et al. 2021c)	61.8	67.3	54.6	103,440	88,901	5,243	13.2
CSTrack (Liang et al. 2021)	66.6	68.6	54.0	144,358	25,404	3,196	4.5
TransTrack (Sun et al. 2020)	65.0	59.4	48.5	150,197	27,197	3,608	7.2
TransCenter (Xu et al. 2021)	61.9	50.4	-	146,347	45,895	4,653	1.0
CorrTracker (Wang et al. 2021a)	65.2	69.1	-	95,855	79,429	5,183	8.5
GSDT (Wang, Kitani, and Weng 2021)	67.1	67.5	53.6	135,409	31,913	3,131	0.9
SiamMOT (Wang et al. 2021b)	67.1	69.1	-	-	-	-	4.3
RelationTrack (Yu et al. 2021)	67.2	70.5	56.5	104,597	61,134	4,243	2.7
SOTMOT (Zheng et al. 2021)	68.6	71.4	-	101,154	57,064	4,209	8.5
MAATrack (Stadler and Beyerer 2022)	73.9	71.2	57.3	108,744	24,942	1,331	14.7
StrongSORT++ (Du et al. 2023)	73.8	77.0	62.6	117,920	**16,632**	**770**	-
OCSORT (Cao et al. 2022)	75.7	76.3	62.4	105,894	19,067	942	-
TransMOT (Liu et al. 2021)	77.5	75.2	-	**80,788**	34,201	1615	-
ByteTrack (Zhang et al. 2021b)	77.8	75.2	61.3	87,594	26,249	1,223	17.5
BoT-SORT (Aharon, Orfaig, and Bobrovsky 2021)	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 (Aharon, Orfaig, and Bobrovsky 2021)	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

Table 2: Comparison against the SoTA methods on the MOT20 (Dendorfer et al. 2020) test set.

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 3: Ablation analysis of SLM, SMC, and GATE Function (GF) on the MOT17 validation set, compared to the leading ByteTrack (Zhang et al. 2021b) that utilizes the YOLOX (Ge et al. 2021) detector. The FPS encompasses detection, NMS, re-identification, and data association, excluding image acquisition and video encoding/decoding processes.

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

adopts the type-E patch layout for all performance evaluation and comparison.

Re-Identification Strategies. Traditional methods primarily rely on IOU to calculate similarity scores (Zhang et al. 2021b). 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 (Chen et al. 2021) outperforms ByteTrack with YOLOX (Ge et al. 2021) 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.

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

Aharon, N.; Orfaig, R.; and Bobrovsky, B.-Z. 2021. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv:2206.14651.*

Bernardin, K.; and Stiefelhagen, R. 2018. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP JIVP*.

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. 2016. Simple online and realtime tracking. *ICIP*.

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-y. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv:2004.10934*.

Cao, J.; Weng, X.; Khirodkar, R.; Pang, J.; and Kitani, K. 2022. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking. *arXiv:2203.14360*.

Chen, P.-Y.; Chang, M.-C.; Hsieh, J.-W.; and Chen, Y.-S. 2021. Parallel residual bi-fusion feature pyramid network for accurate single-shot object detection. *IEEE Transactions on Image Processing*, 30: 9099–9111.

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

Dollár, P.; Wojek, C.; Schiele, B.; and Perona, P. 2009. Pedestrian Detection: A Benchmark. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929*.

Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; and Meng, H. 2023. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*.

Ess, A.; Leibe, B.; Schindler, K.; and Van Gool, L. 2008. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv:2107.08430*.

Li, J.; Tokmakov, P.; and Gaidon, A. 2021. MAT:: Motion-aware multi-object tracking. *Neurocomputing*, 476: 104–114.

Li, Z.; Zhang, J.; Wang, P.; Zhang, L.; Cao, L.; and Li, Y. 2021. Set transformer for multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4374–4383.

Liang, C.; Zhang, Z.; Lu, Y.; Zhou, X.; Li, B.; Ye, X.; and Zou, J. 2021. Rethinking the Competition Between Detection and ReID in Multiobject Tracking. *IEEE Transactions on Image Processing*, 30: 7188–7200.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; and Dollár, P. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. *ECCV*, 21–37.

Liu, Z.; Wang, L.; Wang, Z.; and Siu, W.-C. 2021. TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking. In *ICCV*, 10002–10011.

Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.; Geiger, A.; Leal-Taixé, L.; and Leibe, B. 2021. HOTA: A higher order metric for evaluating multi-object tracking. *IJCV*, 129: 548–578.

Luo, C.; Ma, C.; Wang, C.; and Wang, Y. 2017. Learning discriminative activated simplices for action recognition. In *AAAI*.

Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; and Feichtenhofer, C. 2022. TrackFormer: Multi-Object Tracking with Transformers. In *CVPR*.

Milan, A.; Leal-Taixe, L.; Reid, I.; Roth, S.; and Schindler, K. 2016. MOT16: A Benchmark for Multi-Object Tracking. *arXiv:1603.00831*.

Pang, B.; Li, Y.; Zhang, Y.; Li, M.; and Lu, C. 2020. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *CVPR*, 6308–6318.

Pang, J.; Qiu, L.; Li, X.; Chen, H.; Li, Q.; Darrell, T.; and Yu, F. 2021. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, 164–173.

Peng, J.; Wang, L.; Wan, F.; Wu, Y.; Chen, Y.; and Tai, Y. 2020. Chained-Tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *ECCV*, 144–161. Springer.

Peng, T.; Wang, Y.; Zou, Y.; Cao, Z.; Qiao, Y.; and Yuille, A. 2021. TransTrack: Transformer based Multiple Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14830–14839.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. *arXiv:1506.02640*.

Redmon, J.; and Farhadi, A. 2017. YOLO9000: Better, Faster, Stronger. In *CVPR*, 6517–6525.

Redmon, J.; and Farhadi, A. 2018. YOLOv3: An Incremental Improvement. *arXiv:1804.02767*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497*.

Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C. 2016. Performance Measures and a Data Set for Multi-target, Multi-camera Tracking. In *ECCV Workshops*.

Schöps, T.; Schönberger, J. L.; Galliani, S.; Sattler, T.; Schindler, K.; Pollefeys, M.; and Geiger, A. 2017. A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos. In *CVPR*.

Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; and Sun, J. 2018. CrowdHuman: A Benchmark for Detecting Human in a Crowd. *arXiv:1805.00123*.

Stadler, D.; and Beyerer, J. 2021. On the Performance of Crowd-Specific Detectors in Multi-Pedestrian Tracking. In *AVSS*, 1–12.

Stadler, D.; and Beyerer, J. 2022. Modelling Ambiguous Assignments for Multi-Person Tracking in Crowds. In *WACV*, 133–142.

Sun, P.; Jiang, Y.; Rufeng, Z.; Xie, E.; Cao, J.; Hu, X.; Kong, T.; Yuan, Z.; Wang, C.; and Luo, P. 2020. TransTrack: Multiple-Object Tracking with Transformer. *arXiv:2012.15460*.

Tokmakov, P.; Li, J.; Burgard, W.; and Gaidon, A. 2021. Learning to Track with Object Permanence. In *ICCV*, 10012–10021.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017a. Attention is all you need. *NeurIPS*, 30.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017b. Attention is All you Need. In *NeurIPS*, volume 30.

Wang, C.; Wang, Y.; and Yuille, A. L. 2013. An Approach to Pose-Based Action Recognition. In *CVPR*, 915–922.

Wang, C.-Y.; et al. 2020a. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In *CVPR Workshops*.

Wang, Q.; Zheng, Y.; Pan, P.; and Xu, Y. 2021a. Multiple Object Tracking with Correlation Learning. In *CVPR*, 3876–3886.

Wang, Y.; Kitani, K.; and Weng, X. 2021. Joint Object Detection and Multi-Object Tracking with Graph Neural Networks. In *ICRA*, 10077–10083. IEEE.

Wang, Z.; Li, Z.; Han, S.; and Wang, H. 2021b. One More Check: Making "Fake Background" Be Tracked Again. In *AAAI*, volume 35, 15446–15454.

Wang, Z.; Zheng, L.; Liu, Y.; and Wang, S. 2020b. Towards Real-Time Multi-Object Tracking. In *ECCV*.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *ICIP*, 3645–3649.

Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; and Yuan, J. 2021. Track To Detect and Segment: An Online Multi-Object Tracker. In *CVPR*, 12352–12361.

Xiao, T.; Li, S.; Wang, B.; Lin, L.; and Wang, X. 2017. Joint Detection and Identification Feature Learning for Person Search. *arXiv:1604.01850*.

Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; and Alameda-Pineda, X. 2021. TransCenter: Transformers with Dense Representations for Multiple-Object Tracking. *IEEE PAMI*.

Yang, L.; Wang, P.; Li, X.; Wu, W.; and Luo, P. 2021. MOTSNet: A Unified Framework for Multi-Object Tracking and Segmentation. In *CVPR*, 11686–11695.

Yu, E.; Li, Z.; Han, S.; and Wang, H. 2021. RelationTrack: Relation-Aware Multiple Object Tracking With Decoupled Representation. In *ICIP*, 3004–3008.

Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; and Wei, Y. 2021. MOTR: End-to-end multiple-object tracking with transformer. In *ICCV*, 11076–11085.

Zhang, K.; Wang, X.; Yang, J.; Ma, Q.; and Huang, G. 2021a. TrackFormer: Multi-Object Tracking with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8841–8850.

Zhang, S.; Benenson, R.; and Schiele, B. 2017. CityPersons: A Diverse Dataset for Pedestrian Detection. In *CVPR*, 4457–4465.

Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; and Wang, X. 2021b. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv:2110.06864*.

Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; and Liu, W. 2021c. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 129: 3069–3087.

Zhang, Z.; Wang, L.; Wang, Z.; and Siu, W.-C. 2020. ReMOT: A model-agnostic refinement for multiple object tracking. *Image and Vision Computing*, 105: 104067.

Zheng, L.; Tang, M.; Chen, Y.; Zhu, G.; Wang, J.; and Lu, H. 2021. Improving Multiple Object Tracking With Single Object Tracking. In *CVPR*, 2453–2462.

Zheng, L.; Zhang, H.; Sun, S.; Chandraker, M.; Yang, Y.; and Tian, Q. 2017. Person Re-identification in the Wild. *arXiv:1604.02531*.

Zhou, X.; Koltun, V.; and Krähenbühl, P. 2020. Tracking Objects as Points. In *ECCV*, 474–490. Springer.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as Points. *arXiv:1904.07850*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable DETR: Deformable transformers for end-to-end object detection. In *IICLR*.