Decode-MOT: How Can We Hurdle Frames to Go Beyond Tracking-by-Detection?

Seong-Ho Lee, Dae-Hyeon Park, and Seung-Hwan Bae, *Member, IEEE*

***Abstract*— The speed of tracking-by-detection (TBD) greatly depends on the number of running a detector because the detection is the most expensive operation in TBD. In many practical cases, multi-object tracking (MOT) can be, however, achieved based tracking-by-motion (TBM) only. This is a possible solution without much loss of MOT accuracy when the variations of object cardinality and motions are not much within consecutive frames. Therefore, the MOT problem can be transformed to find the best TBD and TBM mechanism. To achieve it, we propose a novel decision coordinator for MOT (Decode-MOT) which can determine the best TBD/TBM mechanism according to scene and tracking contexts. In specific, our Decode-MOT learns tracking and scene contextual similarities between frames. Because the contextual similarities can vary significantly according to the used trackers and tracking scenes, we learn the Decode-MOT via self-supervision. The evaluation results on MOT challenge datasets prove that our method can boost the tracking speed greatly while keeping the state-of-the-art MOT accuracy. Our code will be available at https://github.com/reussite-cv/Decode-MOT.**

***Index Terms*— Multi-object Tracking, tracking-by-detection, tracking-by-motion, scene and tracking contextual learning, hierarchical association.**

I. Introduction

TRACKING-BY-DETECTION [1], [2], [3], [4] is still a dominant paradigm in multi-object tracking (MOT). Basically, it builds object tracks by using temporal local and global associations between detections. Therefore, exploiting outputs of an accurate detector improves the MOT accuracy significantly due to the reduction of the uncertainty for possible object locations. The recent advance of deep convolutional detectors [5], [6], [7], [8], [9], [10] also contributes to boosting the accuracy of the modern MOT methods [2], [3], [11], [12], [13]. However, employing high-accurate detectors degrades the overall tracking speed (including the detection latency) in return.

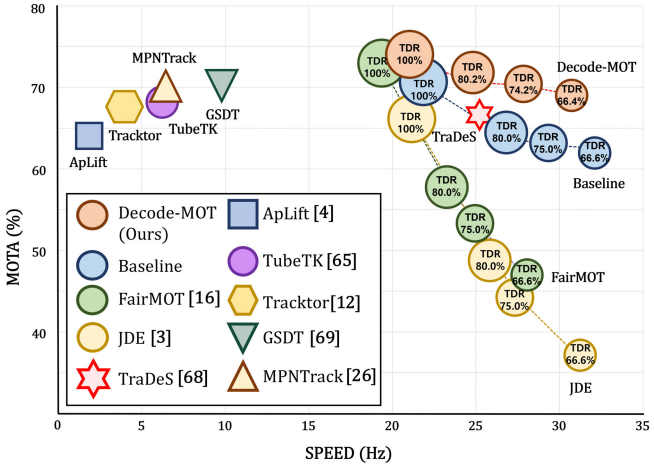

Fig. 1. Accuracy and speed of the recent methods on the MOTChallenge dataset.

For addressing this, detectors [14], [15] with light ConvNets can be used for MOT [2], [3], [16]. However, the computational cost for detection is much costly compared to the association. This means that the total MOT speed is affected the most by a detector in the tracking-by-detection approach.

However, we argue that it is not necessary to run the costly detector over whole frames. Tracking at a frame could be achieved well by tracking-by-motion (TBM) which predicts tracked motions at the current frame. In addition, we find that the frame to be possible of the detection-skipping has similar contexts with nearby frames in general. In other words, the variations of tracking contexts and scene context are not much. Therefore, we can localize object tracks at this frame by performing TBM approach. As a result, the overall tracking speed can be enhanced more as the usage of a detector becomes less. To this end, we can adopt the simple approach which alternates between TBD and TBM mechanism with a certain interval (*i.e.* uniform sampling on frame domain) [17], [18]. However, finding the optimal value of the interval is tricky since this interval can be different even within a sequence. For instance, the interval should be lower when the motion variations of objects increase more or the tracking quality becomes lower. Therefore, our work aims at

Manuscript received 1 August 2022; revised 30 May 2023; accepted 7 July 2023. Date of publication 28 July 2023; date of current version 3 August 2023. This work was supported in part by the National Research Foundation of Korea (NRF) under Grant NRF-2022R1C1C1009208, in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) (Deep Total Recall, 10%) under Grant 2022-0-00448, in part by IITP (Artificial Intelligence Convergence Innovation Human Resources Development: Inha University) under Grant RS-2022-00155915, and in part by the Basic Science Research Program through the NRF under Grant 2022R1A6A1A03051705. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xu-Yao Zhang. *(Seong-Ho Lee and Dae-Hyeon Park contributed equally to this work.) (Corresponding author: Seung-Hwan Bae.)*

Seong-Ho Lee was with the Vision and Learning Laboratory, Inha University, Incheon 22212, Republic of Korea. He is now with SK Hynix, Icheon, Gyeonggi-do 17336, Republic of Korea (e-mail: sho6368@gmail.com).

Dae-Hyeon Park and Seung-Hwan Bae are with the Vision and Learning Laboratory, Inha University, Incheon 22212, Republic of Korea (e-mail: saintPalite2221@inha.edu; shbae@inha.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2023.3298538, provided by the authors.

Digital Object Identifier 10.1109/TIP.2023.3298538

seeking the best sweet spot of a tracker for real-time MOT, whereas minimizing the reduction of MOT accuracy.

For achieving this, some recent works [19], [20] compare the tracking boxes between previous and current frames, and perform the TBD when the mismatch between boxes increases. However, we assert that the object cardinality should be considered when evaluating the tracking contextual similarity since the track initialization and termination cannot be achieved without TBD. In addition, contextual variation of a whole scene should be evaluated for capturing the variations of object appearances or other occluders. Therefore, comparing tracking boxes or RoI features within object regions [19], [20], [21] is not sufficient to determine TBD and TBM mechanism.

To resolve this, our core idea is to measure the dissimilarity of Conv feature maps between the current and the key (*i.e.* the most recent frame using TBD) frames. For learning scene contextual similarity between frames, we propose a novel self-attention learning with the online updated Conv features. Additionally, we present tracking contextual measures that can evaluate the motion and cardinality similarities with tracking results inferred at key and current frames.

Based on the scene and tracking context learning, we design a decision coordinator for MOT (Decode-MOT). More concretely, we feed the scene attention features to a decision coordinator as inputs. However, the supervision for training our Decode-MOT is not practical because the GT is not available. In some senses, making GT could be meaningless since it relies on the performance of a tracker and a detector. We, therefore, train our Decode-MOT via self-supervision. We present pseudo labeling that represents TBD or TBM actions. Then, we define a new decision loss with the motion and cardinality similarities for TBD and TBM results. By minimizing this loss, we can train the decision coordinator to make it predict the best decision for TBM or TBD at each frame. The main benefit of our self-supervised learning, this mechanism can be determined adaptively according to the performance of the used detector and tracker. It indicates that our self-supervised learning can be applied to other TBD methods [1], [3], [16], [22].

For robust Decode-MOT, we present a hierarchical confidence association between detections and tracks. In this hierarchical association, we consider confidences of tracks and detections and associate them hierarchically by reducing the ambiguity of possible matching combinations gradually.

To sum up, the main contributions of this paper are (i) proposition of Decode-MOT that can determine the best TBD or TBM mechanism for real-time and high-accurate tracking; (ii) proposition of a new contextual learning in order to measure scene and tracking contextual similarities between different frames; (iii) proposition of a self-supervision method based on the scene and tracking contexts; (iv) proposition of a hierarchical confidence association which can reduce association ambiguity gradually. By applying our proposed methods, we achieve about 1.5× faster speed while reducing MOTA by 4.3%. We have also provided extensive ablation studies and comparisons over the state-of-the-art MOT methods on the MOT benchmark dataset [23]. Our Decode-MOT achieves state-of-the-arts 73.2% MOTA and 21.6Hz speed on the MOT17 test set by using a single Titan Xp. As shown in Fig. 1, recent trackers focus on improving the MOT speed by modifying its architectures [3], [16]. Thus, the overall model complexity can be reduced. However, our method aims to boost the MOT speed by determining the optimal tracking mechanism (*i.e.* TBM or TBD). It implies that the algorithm complexity of the MOT system can be downsized since the total number of operations of running a detector gets decreased. In our experiment in Fig. 1 and Table VI, we prove that our Decode-MOT can provide the higher gains for both accuracy and speed than recent MOT methods. As shown in Fig. 1, we observe that applying the simple approach, which alternates TBD and TBM with a fixed interval, for other SOTA trackers [3], [16] can decrease MOT accuracies easily because the optimal key intervals can be different for the performance of a tracker. In addition, the performance degradation of our Decode-MOT is much lower than other methods as shown in Fig. 1 and Table I as TDR decreases.

II. Related Work

We discuss previous works on multi-object tracking, efficient object tracking, and self-supervised learning which are related to our work.

A. Multi-Object Tracking

The goal of object tracking is to track multiple objects and build its (*or* their) trajectories. The multi-object tracking approaches can be categorized into tracking-by-detection and tracking-by-motion according to the use or not of a detector. In this section, we discuss the details of both methods.

1) Tracking-by-Detection: The tracking-by-detection approach [1], [2], [3], [13], [22], [24], [25], [26], [27] first determines possible object locations within an image by applying a detector, and then associates detections between consecutive frames to build a track with distinguishable identities (IDs). The TBD approach can be categorized into separate detection and embedding (SDE) [1], [5], [11] and joint detection and embedding (JDE) [3], [13], [16], [25]. The main difference between both methods is whether detection and association tasks are handled in a common network (*or* backbone). For solving both tasks, the SDE methods use different networks trained independently. On the other hand, the JDE methods attach detection and association heads to a shared single network, and train them using joint learning. Therefore, the JDE improves tracking speed by sharing low-level features. However, both methods leverage a detector for whole frames. Thus, the detection process largely affects the overall MOT complexity still.

On the other hand, as an effort to improve the accuracy, the self-attention mechanism [28], [29] are used. In specific, [22], [30], [31], [32] use the attention for improving association. We also use the attention, but we learn the attention features online by comparing features from different frames.

2) Tracking-by-Motion: The tracking-by-motion approach estimates the object states at the current frame based on the prediction of its tracked motions up to a certain previous frame. Therefore, this is much faster than TBD due to the

absence of a costly detection process. However, it is not possible for applying TBM over the whole frames since new track initialization or track recovery due to occlusions can not be achieved without detections. Therefore, it is crucial to determine a key frame needed to run a detector. To this end, [19], [20], [21], [33] choose the key frame by considering object regions and region features. Specifically, [21] determines the key frame by computing an object size and motion differences between consecutive frames. References [20] and [33] formulate the task into a reinforcement learning problem and find the key frame based on the dissimilarity of object localization. Unfortunately, these methods do not consider the cardinality variation between frames. Therefore, we evaluate tracking contextual similarity between frames in consideration of motion and cardinality variations together. As a result, our method can determine the best key frame by using the integrated contexts. To show the effects of using both contexts, we provide the ablation study of our Decode-MOT by using those differently.

On the other hand, in the respect of precise predictions of object motions during TBM, many recent visual object-tracking (VOT) methods [34], [35], [36], [37], [38], [39] focused on learning stronger and discriminative object features. For example, [39] formulates the task into a target matching problem within a correlation feature map of the Siamese network which is generated by convolving the search region feature map with the object region of interest (RoI) feature. Reference [38] exploits self-attention mechanism [28] in order to learn discriminative features. Reference [37] introduces the quadruplet network [40] into visual object tracking for the same purpose. Albeit we use the discriminative feature learning with the attention methods between consecutive feature maps as done in these VOT methods, our approach is more focused on improving total tracking speed rather than accuracy by using the discriminative feature for determining the MOT mechanism.

B. Efficient Object Tracking

In recent years, various efficient tracking methods have been proposed for embedding trackers on real-time applications. Reference [41] shows the possibility of constructing a lightweight tracker by using neural architecture search. Shen et al. [42] exploits the knowledge distillation [43] in order to learn the lightweight tracker from the more accurate but heavier tracker. Wang et al. [3] integrates a detection model and an appearance embedding model by sharing the same set of low-level features for avoiding the costly feature re-extraction. Reference [16] learns low dimensional Re-ID features to improve a model inference speed. Note that these methods can improve the tracking speed by reducing the model complexity of trackers. On the other hand, our method rather focuses on reducing the MOT algorithm complexity at a system level by minimizing the operations of running TBD as shown in Fig. 1. Therefore, the distinct benefit is that the conventional methods of reducing model complexity can be also compatible with our method.

C. Self-Supervised Learning

Self-supervised learning (SSL) [44], [45] generates pseudo labels for learning a model with unlabeled data. There are some studies to improve the generalization ability of MOT models using SSL. Reference [16] improves the re-identification (Re-ID) generalization with human detection datasets [46]. Reference [22] exploits a self-supervised loss in order to apply constraints of the spatial correlation learning [47]. Reference [48] generates pseudo labels using SORT [49] for training an unsupervised Re-ID model. We also leverage SSL learning for MOT. However, our learning method aims at improving the tracking speed as well as the model generalization.

III. Decision Coordinator for MOT (Decode-MOT)

Figure 2 and Figure 4 show an overview of our decision coordinator for MOT (Decode-MOT). It consists mainly of the decision coordinator, the scene and tracking contextual learning, and the hierarchical confidence association. We first discuss TBD and TBM, and then present details of each method.

A. Tracking-by-Detection (TBD) and Tracking-by-Motion (TBM)

Given a sequence with N frames, we apply a detector $\mathbf{D}$ for the image I_t to generate a set of detection boxes $\mathbf{D}(I_t) = \mathcal{D}_t = \{\mathbf{d}_t^i, y_t^i\}_{i=1}^{|\mathcal{D}_t|}$, where $\mathbf{d}_t^i$ is a bounding box for an object i, and $|\mathcal{D}_t|$ is the number of detected objects at frame t. We then formulate a tracking-by-detection problem to estimate a set of tracks $\mathcal{T}_t = \{\hat{\mathbf{d}}_t^j, \hat{y}_t^j\}_{j=1}^{|\mathcal{T}_t|}$ at the current frame t as $\mathcal{T}_t^{(TD)} = \mathbf{T}(\mathcal{T}_{1:t-1}, \mathcal{D}_t)$, where $\hat{\mathbf{d}}_t^j$ and $\hat{y}_t^j$ are a refined bounding box by tracking and track identity label for a track j. $\mathbf{T}$ and $|\mathcal{T}_t|$ are a tracker and the number of tracks at frame t.

When predicting $\mathcal{T}_t$ based on the previous knowledge $\mathcal{T}_{1:t-1}$ up to frame $t-1$ only, we consider this as the TBM problem and formulate it as $\mathcal{T}_t^{(TM)} = \mathbf{T}(\mathcal{T}_{1:t-1})$. In general, this problem can be solved by the motion prediction based on tracking results $\mathcal{T}_{1:t-1}$ up to frame $t-1$. In our case, we use simple Kalman filtering [50]. In most cases, the quality of $\mathcal{T}_t^{(TD)}$ is better than $\mathcal{T}_t^{(TM)}$ because of the more accurate association. On the other hand, the tracking complexity for $\mathcal{T}_t^{(TD)}$ is higher than that of $\mathcal{T}_t^{(TM)}$. Therefore, we need to find the TBD and TBM mechanism for reducing tracking complexity while keeping the accuracy. As mentioned, to achieve this, our core idea is to measure the tracking and scene contextual similarity extracted at current image frame t and previous frames $1:t-1$.

To find optimal the mechanism for whole frames, we assume that $\mathcal{T}^{(TM)}$ at frame t is possible if $\mathcal{T}_t^{(TD)} \approx \mathcal{T}_t^{(TM)}$. In many cases, this assumption is feasible when the tracking (*i.e.* object motions and cardinality) and scene (*i.e.* image feature) contextual information are similar between consecutive frames. Therefore, we present methodologies to learn and measure the contextual similarities. Based on those contextual similarities, we learn a decision coordinator and determine TBD and TBM operation per frame using the learned coordinator.

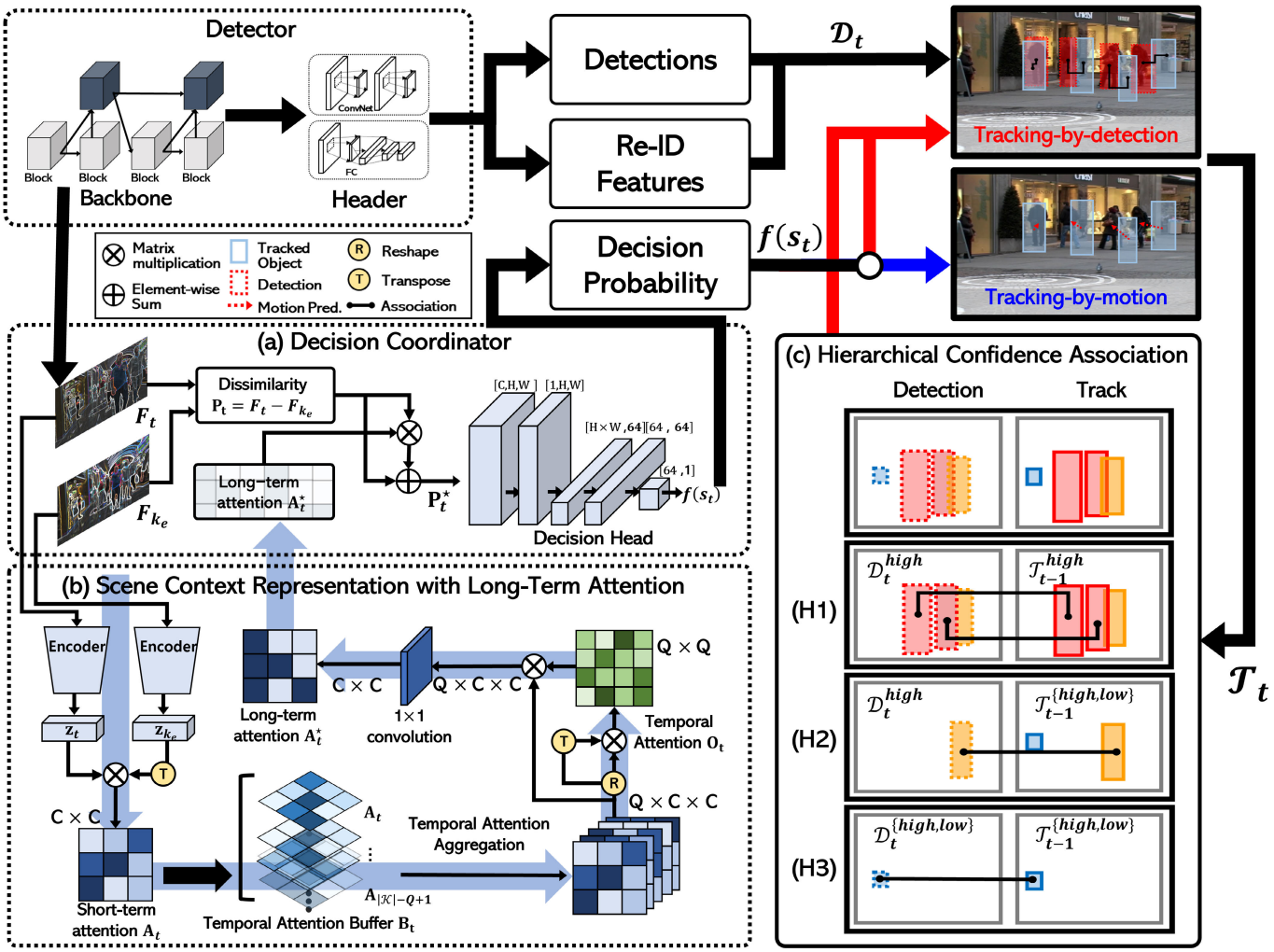

Fig. 2. The overall architecture of our Decode-MOT. It consists of (a) a decision coordinator of predicting the probability of TBM, (b) a scene context representation module of evaluating the long-term attention between different frames, and (c) a hierarchical association of linking between detections and tracks progressively.

B. Decision Coordinator

In the decision coordinator shown in Fig. 2 (a), we first calculate the dissimilarity of feature maps $P_t \in \mathbb{R}^{C\times H\times W}$ by subtracting current and key frame features $P_t = F_t - F_{k_e}$. Let denote $r(t) = 1$ when a detector runs at frame t. Otherwise, $r(t) = 0$. We define $\mathcal{K} = \{k_j | 1 \le k_s \le k_j \le k_e \le t, r(k_j) = 1\}$ as a set of time stamps of running the detector up to frame t. k_s and k_e are the start- and end-stamp of running the detector. Then, we perform a matrix multiplication between P_t and an attention of the scene context $\mathbf{A}_t^\star$ for emphasizing the dissimilarity further which is discussed in Sec. III-C. After multiplying them, we perform element-wise summation between the multiplied feature and P_t to obtain the refined feature $P_t^\star$. Subsequently, the decision head outputs the decision coordination probability $f(s_t)$ with the refined $P_t^\star$ as its input. To train the coordinator, we present a decision loss in consideration of tracking contexts as in Sec. IV. Because the coordinator determines TBD or TBM based on scene and tracking contextual similarities, our Decode-MOT is likely to operate TBM as $f(s_t)$ becomes higher.

In detail, for the decision head, we feed the refined feature $P_t^\star$ to 3×3, and then 1×1 ConvBlocks. Here, each block contains a convolution, batch normalization, and Leaky ReLU activation ($\alpha = 0.01$) layers. The output channel sizes of 3×3 and 1×1 convolution layers are C and 1, respectively. After consecutive convolutions, we flatten the feature map into a feature vector with $H \times W$ dimension and then apply 3 fully connected layers of 64, 64, and 1 neurons in order to output s_t. Finally, we can calculate the decision probability $f(s_t)$ with a sigmoid function $f(\cdot)$.

C. Scene Context Representation With Long-Term Attention

The scene context learning is basically achieved by learning discrepancy between F_t and F_{k_e} at current t and key k_e (*i.e.* the most recent frame using a detector). For representing the feature discrepancy more, we present an online attention learning during MOT. The overall attention learning process includes short-term and long-term attention learning as depicted in Fig. 2 (b). In short-term attention learning, we extract the embedding features $\mathbf{z}_t$ and $\mathbf{z}_{k_e}$ by feeding F_t and F_{k_e} frames to a shared encoder, and then correlate them to learn the channel

attention $\mathbf{A}_t$. Thus, we can capture cardinality and motion variations from this short-term attention learning. In addition, we aim at learning a long-term temporal dependency for the variations of channel attentions. To this end, we aggregate and update short-term channel attentions up to current frame t, and self-correlate the aggregated attentions to generate the stronger attention $\mathbf{A}_t^\star$. We use $\mathbf{A}_t^\star$ to make P_t more meaningful semantic representation.

1) Short-Term Attention: We feed F_t into an encoder in order to extract a latent vector $\mathbf{z}_t \in \mathbb{R}^C$. We implement the encoder with two 3×3 ConvBlocks, an average pooling layer, and a max pooling layer. Each ConvBlock contains a 3×3 convolution layer, a ReLU activation layer, and a 2×2 max-pooling layer to reduce the spatial resolution of the feature map for the efficient attention learning. Then, we feed the reduced feature map into an average and a max pooling layer independently in order to aggregate spatial information. Subsequently, the encoder can output the latent vector $\mathbf{z}_t$ by performing element-wise summation between two aggregated vectors. Similarly, we extract the key frame latent vector $\mathbf{z}_{k_e}$ by propagating F_{k_e} to the same shared encoder. By multiplying $\mathbf{z}_t$ with the transposed $\mathbf{z}_{k_e}$, we can learn a short-term channel attention $\mathbf{A}_t \in \mathbb{R}^{C \times C}$.

2) Long-Term Attention: From the short-term attention network, we can generate a channel attention map $\mathbf{A}_t$ per frame. For learning long-term variation of the attention features up to the current frame, we present long-term attention learning. When $r(t) = 1$, we can update the set $\mathcal{K}$ (with $|\mathcal{K}|$ time stamps for detection) by adding t into this ($k_e = t$). We then update a temporal attention buffer $\mathbf{B}_t = \{\mathbf{A}_k | k \in \{k_{|\mathcal{K}|-Q+1}, \ldots, t\}\}$ having Q attentions by adding the new $\mathbf{A}_t$ and deleting the oldest one within $\mathbf{B}_t$. Therefore, $\mathbf{B}_t$ contains all the short-term attentions from the $k_{|\mathcal{K}|-Q+1}$ to current t frames. In addition, we can represent $\mathbf{B}_t$ as $\mathbf{B}_t \in \mathbb{R}^{Q \times C \times C}$ by concatenating each attention $\mathbf{A}_k$ along the temporal dimension. Then, we can reshape $\mathbf{B}_t$ to make its dimension as $Q \times C^2$. To obtain a temporal attention $\mathbf{O}_t \in \mathbb{R}^{Q \times Q}$, we perform a matrix multiplication between $\mathbf{B}_t$ and the transpose of $\mathbf{B}_t$. We then apply a softmax function for the $\mathbf{O}_t$ along the channel axis for normalization. Subsequently, we perform a matrix multiplication between $\mathbf{B}_t$, $\mathbf{O}_t$, and apply a 1×1 convolution layer and a softmax function in order for temporal feature aggregation. As a result, we can generate a long-term attention $\mathbf{A}_t^\star \in \mathbb{R}^{C \times C}$. By using $\mathbf{A}_t^\star$, we can learn a channel weighted feature $P_t^\star$ as follows:

$$P_t^\star = P_t + \gamma(\mathbf{A}_t^\star \otimes P_t), \tag{1}$$

where $\otimes$ is a matrix multiplication. γ is a learnable weight parameter, which is initialized to 0. Note that the temporal buffer $\mathbf{B}_t$ is updated in online since a temporal window of the buffer also moves for the next frame tracking. Therefore, the long-term attention $\mathbf{A}_t^\star$ can be updated in online.

D. Hierarchical Confidence Association

For associating tracks $\mathcal{T}_{1:t}$ with detections $\mathcal{D}_{1:t}$ in DecodeMOT, we present the hierarchical association using confidences of tracks and detections shown in Fig. 2 (d). In our case, we consider a detection confidence as a classification probability of the detection box from a detector and track confidence as the confidence of the associated detection. When a track is not associated, its confidence gradually decreases with a certain scaling factor φ (We set 0.9 in our experiment.). Therefore, a track confidence $conf(\mathcal{T}_t^i)$ can be defined as follows:

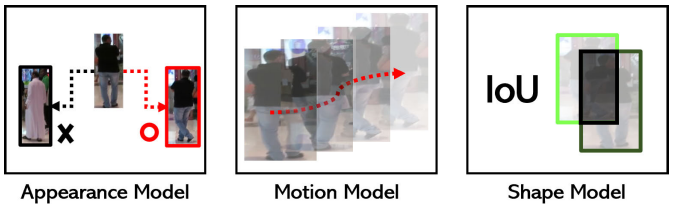

Fig. 3. An illustration of affinity models consisting of the appearance, motion and shape model. We exploit them when associating between tracks and detections.

$$conf(\mathcal{T}_t^i) = \begin{cases} conf(\mathcal{D}_t^{i^*}), & s.t. \mathcal{T}_t^i \in \mathcal{T}_t^m \\ \varphi \cdot conf(\mathcal{T}_{t-1}^i), & s.t.\ \mathcal{T}_t^i \in \mathcal{T}_t^u \end{cases}, \tag{2}$$

where $conf(\mathcal{D}_t^{i^*})$ is a detection confidence associated with the track $\mathcal{T}_{t-1}^i$ from a used detector. $\mathcal{T}_t^m$ and $\mathcal{T}_t^u$ are matched and unmatched track sets at frame t, respectively.

In order to compute the similarity between tracks $\mathcal{T}_{t-1}$ and detections $\mathcal{D}_t$, we use appearance, motion, and shape models denoted as $\mathcal{A}^{app}$, $\mathcal{A}^{mot}$, and $\mathcal{A}^{shape}$, respectively, as shown in Fig. 3. As an appearance affinity model, we use 128-dimensional embedding vectors for each RoI region from an appearance feature network consisting of a 3×3 and an 1×1 convolution and a ReLU activation layers. To evaluate appearance affinity, we use the cosine distance. To evaluate motion affinity, we compute the Mahalanobis distance between the center coordinate of a detection box and refined position of a track. For the shape affinity, we calculate IoU distance between bounding boxes. Subsequently, we apply the weighted summation for all affinity models in order to calculate the total affinity between a track and detection $\mathcal{A}^{tot}(\mathcal{T}_{t-1}^i, \mathcal{D}_t^j)$ as follows:

$$\mathcal{A}^{tot} = \kappa^{app} \cdot \mathcal{A}^{app} + \kappa^{mot} \cdot \mathcal{A}^{mot} + \kappa^{shape} \cdot \mathcal{A}^{shape}, \tag{3}$$

where weight parameters for appearance, motion, and shape affinity models are denoted as κ^{app}, κ^{mot}, and κ^{shape}, respectively. We set these hyper parameter values experimentally as described in Sec. V-A. Note that we apply the same values for all our experiments. $\mathcal{T}_{t-1}^i$ and $\mathcal{D}_t^j$ are a track and detection, respectively. i, j are indexes of tracks and detections. In Eq. (3), $(\mathcal{T}_{t-1}^i, \mathcal{D}_t^j)$ is omitted for the simplicity.

For hierarchical association, we first categorize tracks $\mathcal{T}_{t-1}$ and detections $\mathcal{D}_t$ in terms of their confidences $conf(\cdot)$. If their confidences are above a certain threshold θ_{conf}, we regard them as high confidence tracks $\mathcal{T}_{t-1}^{high}$ and detections $\mathcal{D}_t^{high}$. Otherwise, low confidence tracks $\mathcal{T}_{t-1}^{low}$ and detections $\mathcal{D}_t^{low}$. Here, in case of detections, we discard detections whose confidence is less than $\psi \cdot \theta_{conf}$ in order to reduce the number of false detections. We set ψ to 0.8 in our experiments.

After then, we perform sequential associations with those affinity models as follows:

Algorithm 1 Hierarchical Confidence Association

Input : Tracks $\mathcal{T}_{t-1}$, detections $\mathcal{D}_t$
Output: Updated tracks $\mathcal{T}_t$

// **Categorize $\mathcal{T}_{t-1}$ and $\mathcal{D}_t$ in terms of their confidences**
$\mathcal{T}_{t-1}^{high}, \mathcal{T}_{t-1}^{low} \leftarrow$ Categorize $\mathcal{T}_{t-1}$ by θ_{conf}
$\mathcal{D}_t^{high}, \mathcal{D}_t^{low} \leftarrow$ Categorize $\mathcal{D}_t$ by θ_{conf} and $\psi \cdot \theta_{conf}$
// **(H1) association**
Compute $\mathcal{A}^{tot}(\mathcal{T}_{t-1}^{high}, \mathcal{D}_t^{high})$ by Eq. (3)
Determine optimal pairs of $\mathcal{T}_{t-1}^{high}$ and $\mathcal{D}_t^{high}$ using Eq. (4)
Generate $\mathcal{T}_{t-1}^{m_1}, \mathcal{D}_t^{m_1}, \mathcal{T}_{t-1}^{u_1}$, and $\mathcal{D}_t^{u_1}$
// **(H2) association**
$\mathcal{T}_{t-1}^{u_1} \leftarrow Merge(\mathcal{T}_{t-1}^{u_1}, \mathcal{T}_{t-1}^{low})$
Compute $\mathcal{A}^{shape}(\mathcal{T}_{t-1}^{u_1}, \mathcal{D}_t^{u_1})$
Determine optimal pairs of $\mathcal{T}_{t-1}^{u_1}$ and $\mathcal{D}_t^{high}$ using Eq. (5)
Generate $\mathcal{T}_{t-1}^{m_2}, \mathcal{D}_t^{m_2}, \mathcal{T}_{t-1}^{u_2}$, and $\mathcal{D}_t^{u_2}$
// **(H3) association**
$\mathcal{D}_t^{u_2} \leftarrow Merge(\mathcal{D}_t^{u_2}, \mathcal{D}_t^{low})$
Compute $\mathcal{A}^{shape}(\mathcal{T}_{t-1}^{u_2}, \mathcal{D}_t^{u_2})$
Determine optimal pairs of $\mathcal{T}_{t-1}^{u_2}$ and $\mathcal{D}_t^{u_2}$ using Eq. (6)
Generate $\mathcal{T}_{t-1}^{m_3}, \mathcal{D}_t^{m_3}, \mathcal{T}_{t-1}^{u_3}$, and $\mathcal{D}_t^{u_3}$
// **Track update**
Update $\mathcal{T}_t$ by matching pairs $(T_{t-1}^{m_1}, D_t^{m_1})$, $(T_{t-1}^{m_2}, D_t^{m_2})$, $(T_{t-1}^{m_3}, D_t^{m_3})$
Update $\mathcal{T}_t$ by generating new tracks using $\mathcal{D}_t^{u_3} \backslash \mathcal{D}_t^{low}$
Update $\mathcal{T}_t$ by unmatched tracks $\mathcal{T}_{t-1}^{u_3}$

(H1) association: We first associate between tracks $\mathcal{T}_{t-1}^{high}$ and detections $\mathcal{D}_t^{high}$ with high confidences using $\mathcal{A}^{tot}$ because these are reliable matching pairs. To this end, we compute an affinity matrix $\mathcal{M}_t$ whose elements are the total affinity scores between tracks and detections as follows:

$$\mathcal{M}_t = [-\mathcal{A}_{ij}]_{|\mathcal{T}_{t-1}^{high}| \times |\mathcal{D}_t^{high}|}, \tag{4}$$

where $|\mathcal{T}_{t-1}^{high}|$ and $|\mathcal{D}_t^{high}|$ are object cardinality of $\mathcal{T}_{t-1}^{high}$ and $\mathcal{D}_t^{high}$, respectively. Then, we determine optimal matching pairs in $\mathcal{M}_t$ using the Hungarian algorithm [51] such that the total affinity score is maximized. By exploiting this association, we can reduce the association ambiguity since reliable objects are associated without the interference of unreliable objects. Subsequently, we can output $\mathcal{T}_{t-1}^{m_1}, \mathcal{D}_t^{m_1}, \mathcal{T}_{t-1}^{u_1}$ and $\mathcal{D}_t^{u_1}$ which are matched tracks and detections, and unmatched tracks and detections in (H1) association, respectively. $\mathcal{T}_{t-1}^{u_1}$ and $\mathcal{D}_t^{u_1}$ are reused in (H2) association.

(H2) association: After reducing some association ambiguity in (H1), we associate the remaining tracks $T_{t-1}^{u_1}$ with detections having high confidence $D_t^{u_1}$. Here, we merge $\mathcal{T}_{t-1}^{u_1}$ with $\mathcal{T}_{t-1}^{low}$ in order to consider low confidence tracks together. Different from (H1), we only use the shape affinity $\mathcal{A}^{shape}$ since tracks with low confidences usually have contaminated appearance and motions. For finding optimal pairs, we generate $\mathcal{M}_t$ as follows:

$$\mathcal{M}_t = [-\mathcal{A}_{ij}^{shape}]_{|\mathcal{T}_{t-1}^{u_1}| \times |\mathcal{D}_t^{u_1}|} \tag{5}$$

After finding optimal pairs, we output $\mathcal{T}_{t-1}^{m_2}, \mathcal{D}_t^{m_2}, \mathcal{T}_{t-1}^{u_2}$ and $\mathcal{D}_t^{u_2}$ in (H2) association. $\mathcal{T}_{t-1}^{u_2}$ and $\mathcal{D}_t^{u_2}$ are reused in (H3) association.

(H3) association: we then associate between all tracks and detections in order to associate low confidence detections D_t^{low}. To this end, we combine $\mathcal{D}_t^{u_2}$ with $\mathcal{D}_t^{low}$. We also find optimal pairs after calculating $\mathcal{M}_t$ whose elements are calculated using the shape model only as follows:

$$\mathcal{M}_t = [-\mathcal{A}_{ij}^{shape}]_{|\mathcal{T}_{t-1}^{u_2}| \times |\mathcal{D}_t^{u_2}|} \tag{6}$$

Then, we output $\mathcal{T}_{t-1}^{m_3}, \mathcal{D}_t^{m_3}, \mathcal{T}_{t-1}^{u_3}$ and $\mathcal{D}_t^{u_3}$ in (H3) association. After these associations, we can update $\mathcal{T}_t$ using matched tracks/detections $(T_{t-1}^{m_1}, D_t^{m_1})$, $(T_{t-1}^{m_2}, D_t^{m_2})$, $(T_{t-1}^{m_3}, D_t^{m_3})$. We also initialize new tracks using unmatched high-confidence detections $\mathcal{D}_t^{u_3} \backslash \mathcal{D}_t^{low}$ as mentioned in Sec. V-A. We summarize our proposed association method as shown in Algorithm 1. Refer to our supplementary material for a more detailed association algorithm table.

IV. Decode-MOT Training via Self-Supervision

In order to train the Decode-MOT, we propose to learn the Decode-MOT via self-supervision due to following reasons: (1) There is a no available public GT or any guideline for TBD and TBM scheduling. (2) Generating the GT is also challenging due to the strong performance dependency between a tracker and a detector. (3) It is tricky to generate GT fitting a certain performance point because the accuracy and speed are always the trade-off relationship.

Therefore, we generate pseudo labels to represent actions of running TBD or TBM. Our idea of generating the pseudo labels is to exploit the tracking contextual similarity between previous key and current frames. In particular, as shown in Fig. 4, we measure the contextual similarities for cardinality and motion between them. As a result, our self-supervision method can generate pseudo labels adaptively depending on the tracking contextual similarities.

A. Tracking Contextual Similarity

1) Cardinality Similarity: We first measure the cardinality similarity between the tracking results from $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$. The reason of evaluating this similarity is that the new track initialization and track termination cannot be achieved by the tracking-by-motion. Therefore, we compare the track cardinality between them, and can define the cardinality similarity S_{card} as follows:

$$S_{card}(\mathcal{T}_t^{(TD)}, \mathcal{T}_t^{(TM)}) = \min\left(-\tfrac{1}{e} \cdot \ln\left(1 - R(\mathcal{T}_t^{(TD)}, \mathcal{T}_t^{(TM)})\right), 1\right), \tag{7}$$

where $R(\mathcal{T}_t^{(TD)}, \mathcal{T}_t^{(TM)})$ is the object cardinality ratio between $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$. For normalizing the ratio within the interval $[0, 1]$, we divide each cardinality with the other cardinality, and find the minimum score as $R(\mathcal{T}_t^{(TD)}, \mathcal{T}_t^{(TM)}) = \min\left(\frac{|\mathcal{T}_t^{(TD)}|}{|\mathcal{T}_t^{(TM)}|}, \frac{|\mathcal{T}_t^{(TM)}|}{|\mathcal{T}_t^{(TD)}|}\right)$ In Eq. (7), $S_{card} = 1$ means that they produce the same number of tracks. This indicates that track initialization and termination events are not likely to occur at frame t. Thus, the TM is encouraged as S_{card} increases. Since the cardinality similarity does not consider track IDs to identify the new track initialization and termination, it seems to be difficult to address some cases (*e.g.* when one pedestrian disappears and another one appears at the same frame). To address this, we exploit the motion similarity together.

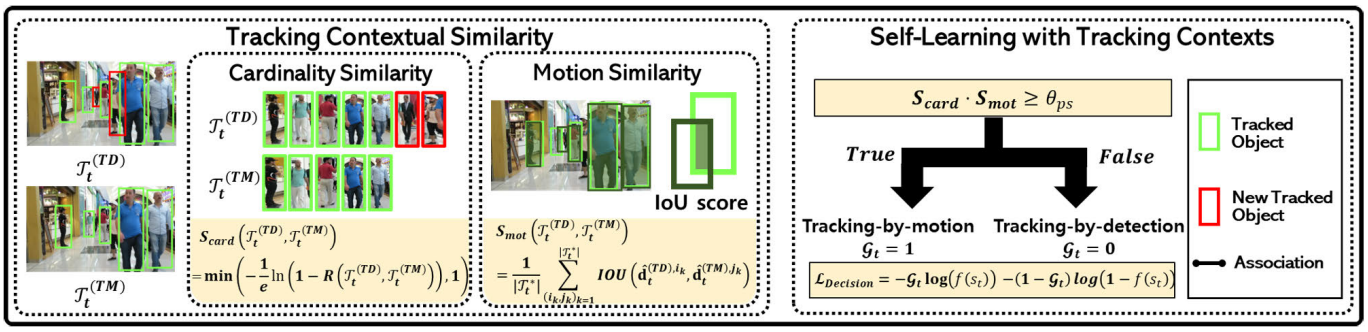

Fig. 4. The proposed tracking contextual learning is shown. We exploit the tracking contextual similarity measures between $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ for learning our Decode-MOT via self-supervision. Using the cardinality similarity we can capture the variation of the track cardinality, but using the motion similarity evaluate the difference of their the localization qualities. We assume that TBD is needed at a frame when these similarities become low.

2) Motion Similarity: In order to evaluate mismatches between refined bounding boxes $\hat{\mathbf{d}}_t$ between $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$, we define the motion similarity. Once corresponding box pairs between them are given, this similarity can be evaluated by using the IoU (intersection over union) measure. Therefore, we can define the motion similarity S_{mot} as follows:

$$S_{mot}(\mathcal{T}_t^{(TD)}, \mathcal{T}_t^{(TM)}) = \frac{1}{|\mathcal{T}_t^*|} \sum_{(i_k, j_k)_{k=1}}^{|\mathcal{T}_t^*|} IOU\left(\hat{\mathbf{d}}_t^{(TD),i_k}, \hat{\mathbf{d}}_t^{(TM),j_k}\right), \quad (8)$$

where $|\mathcal{T}_t^*|$ is the cardinality of matched pairs between $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$. To determine the correspondence (i_k, j_k), we define a bigraph whose bounding boxes of $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ configures two disjoint and independent sets, and each node for $\mathcal{T}_t^{(TD)}$ are connected with every nodes for $\mathcal{T}_t^{(TM)}$. The edge weight is evaluated by using the IoU score between connected nodes. The maximum-weight matching pairs of this graph can be determined optimally by Hungarian algorithm [51]. Then. we evaluate S_{mot} by computing the averaged IoU scores of the matched pairs. We discard a matched one whose IoU score is less than 0.5. Note that leveraging our motion similarity together could address the track ID issue when the regions of track initialization and termination regions are inconsistent. This is because the motion similarity is evaluated for the matched tracks $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ only (*i.e.* considered as the same track IDs). Here, for determining the optimal matching pair, we use the bipartite matching algorithm [51].

B. Self-Learning With Tracking Contexts

For training a decision coordinator via self-supervision, we generate pseudo GT labels $\mathcal{G} = \{\mathcal{G}_t\}_{t=1}^N$. Given a sequence with N frames as a training set, we generate pseudo labels at each frame by using our online tracker $\mathbf{T}$ and detector $\mathbf{D}$. Then, we can generate $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ per frame. Each label $\mathcal{G}_t$ at frame t is a binary label, and represents tracking-by-motion (TM=1) or tracking-by-detection (TD=0) actions. We then evaluate the similarity scores S_{card} Eq. (7) and S_{mot} Eq. (8) by comparing $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$. We can define $\mathcal{G}_t$ as follows:

$$\mathcal{G}_t = \begin{cases} 1, & s.t.\ S_{card} \cdot S_{mot} \geq \theta_{ps} \\ 0, & s.t.\ S_{card} \cdot S_{mot} < \theta_{ps} \end{cases}, \quad (9)$$

where, θ_{ps} is a pseudo labeling threshold. We set it experimentally based on Fig. 5. The meaning behind of each constraint is: when $S_{card} \cdot S_{mot} \geq \theta_{ps}$, the tracking contexts of both trackers $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ are similar. Therefore, we encourage $\mathcal{T}_t^{(TM)}$ at this frame. On the other hand, we encourage $\mathcal{T}_t^{(TD)}$ since the tracking contextual similarity becomes lower due to motion and/or cardinality mismatches.

With the pseudo labeled $\mathcal{G}_t$, we present a decision loss $\mathcal{L}_{decision}$ using the binary cross entropy for learning our decision coordinator as follows:

$$\mathcal{L}_{decision} = -\mathcal{G}_t \log(f(s_t)) - (1 - \mathcal{G}_t) \log(1 - f(s_t)). \quad (10)$$

V. Experiments

In this section, we conduct extensive ablation studies and comparisons over the state-of-the-art (SOTA) methods in order to prove the effects of our method.

Dataset: We use MOTChallenge dataset [23] which contains 7 sequences captured from dynamic or static cameras with 14∼30 Hz frame rates. We exploit the MOT17 training set for training Decode-MOT. For comparison with SOTA methods, we train our Decode-MOT on the pedestrian sets[1] and evaluate our tracker on MOT16, MOT17, and MOT20 test sets using the challenge server. For ablation studies, we use the MOT17 training sets only, and evaluate our methods on the MOT15 training set. Here, we do not evaluate our tracker on the overlapped sequences within the MOT17 training set.[2]

Evaluation Metrics: We use the common MOT metrics [57], [58] as also used in the MOTChallenge: multiple object tracking accuracy (MOTA ↑), higher order tracking accuracy (HOTA ↑), ID F1 score (IDF1 ↑), the number of false positives (FP ↓), the number of false negatives (FN ↓), the number of identity switches (IDs ↓), the ratio of mostly tracked trajectories (MT ↑), the ratio of mostly lost trajectories (ML ↓), the number of track fragment (FG ↓), and multi-object tracking speed (Hz ↑). Here, ↑ and ↓ denote that higher and lower scores are better MOT results, respectively. In addition, we evaluate a tracking-by-detection ratio (TDR) by diving the number of TBD operations with the number of total frames. As described in Sec. III-A, TDR affects the accuracy and speed significantly. As TDR increases, the accuracy gets higher, but the speed slower.

[1]We use CalTech [52], CityPersons [53], CUHK-SYSU [54], PRW [55], ETH [56], and MOT17 [23] training sets.

[2]Venice-2 (=MOT17-02), ADL-Rundle-8 (=MOT17-10), ADL-Rundle-6 (=MOT17-09), ETH-Pedcross2 (=MOT17-05).

TABLE I
COMPARISON AMONG OUR DECODE-MOT, THE BASELINE WITH THE HIERARCHICAL ASSOCIATION, AND THE BASELINE TRACKER WITH DIFFERENT TDRS ON MOT15 DATASET. THE PERCENTAGE IN [·] SHOWS THE SPEED GAIN AND ACCURACY REDUCTION RATES OF EACH TRACKER AS TDR DECREASES

Name	TDR	MOTA↑	FP↓	FN↓	IDs↓	Hz↑
Baseline	100%	71.0%	1,558	3,206	291	21.4
	80.0%	63.8% **[7.2%↓]**	2,301	3,394	610	26.9 **[25.7%↑]**
	75.0%	63.2% **[7.8%↓]**	2,384	3,430	599	29.2 **[36.4%↑]**
	66.6%	62.5% **[8.5%↓]**	2,384	3,551	595	32.1 **[50.0%↑]**
Baseline with Hier. Asso.	100%	73.4%	2,045	2,481	114	21.0
	80.0%	68.2% **[5.2%↓]**	2,529	2,859	145	24.9 **[18.5%↑]**
	75.0%	68.0% **[5.4%↓]**	2,512	2,934	134	26.5 **[26.1%↑]**
	66.6%	67.1% **[6.3%↓]**	2,367	3,219	152	29.7 **[41.4%↑]**
Decode-MOT (Ours)	100%	73.4%	2,045	2,481	114	21.0
	80.2%	70.6% **[2.8%↓]**	2,140	2,852	126	24.9 **[18.5%↑]**
	74.2%	70.3% **[3.1%↓]**	2,018	3,042	125	27.7 **[31.9%↑]**
	66.4%	69.1% **[4.3%↓]**	1,932	3,316	140	30.7 **[46.1%↑]**

A. Implementation Details

We employ DLA-34 [59] as the backbone network in Decode-MOT. We then mount the anchor-free detection [25], appearance feature, decision coordinator head networks on the shared backbone. The proposed scene contextual learning network is embedded into the decision coordinator. as shown in Fig. 2. The networks except for the decision coordinator are trained in advance. Subsequently, we train our decision coordinator by minimizing Eq. (10) while freezing the pre-trained networks such as the backbone, detection, and appearance feature heads. For the scene contextual learning network, we set the buffer size Q to 4. For hierarchical confidence association, κ^{app}, κ^{mot}, and κ^{shape} are set to 0.735, 0.015, and 0.25, respectively. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We train our networks for 30 epochs with a mini-batch including 6 images from different sequences. We set a learning rate to 5e-5, and decay it by a factor of 0.1 at 20 epochs. During inference, we determine tracking-by-detection (TD) or tracking-by-motion (TM) actions per frame by inferring the output $f(s_t)$ of the decision coordinator. When $f(s_t) < \theta_{det}$, we use TD. Otherwise, we use TM. Here, θ_{det} is a threshold to determine a TD or TM action. In a TD action, we localize objects with bounding boxes using a detector and then associate them with existing object tracks. In a TM action, we only localize tracks by predicting their motions with Kalman Filtering [50]. To boost total tracking speed further, we predict a decision score at next frame $t+1$ by degrading the $f(s_t)$ with a factor ξ(=0.83 in our experiment) without running the coordinator when $f(s_t) \geq \theta_{det}$ as follows:

$$f(s_{t+1}) = \begin{cases} \xi \cdot f(s_t), \ s.t. \ f(s_t) \geq \theta_{det} \\ \mathrm{DM}(F_{k_e}, F_t), \ s.t. \ f(s_t) < \theta_{det} \end{cases}, \quad (11)$$

In other words, we assume $\mathcal{T}_t^{(TD)} \approx \mathcal{T}_t^{(TM)}$ when $f(s_t) \geq \theta_{det}$ as mentioned in Sec. III-A. For track initialization, given non-associated high confidence detections $\mathcal{D}_t^{u_3} \backslash \mathcal{D}_t^{low}$ with any existing tracks each frame, we initialize new tracks when the detections at frame t are associated with ones at frame $t-1$ by evaluating IoU scores. Note that we do not use non-associated low-confidence detections for track initialization since these are highly possible to be false detections. For track termination, we degrade track confidences when they are not associated with detections. Then, we eliminate a track when its confidence becomes less than θ_{term}(=0.5 in our experiments) in order to reduce false tracks.

We test our Decode-MOT on a PC with i7-8700K CPU (3.70GHz) and a single Titan Xp (12GiB memory).

B. Ablation Study

1) Decision Coordinator: We compare our Decode-MOT with the baseline. As a baseline, we eliminate the proposed decision coordinator and association method from our Decode-MOT. Instead, we apply a simple method which determines TBD or TBM using an uniform interval, and the association method used in CenterTrack [25]. For the uniform interval, we apply 3 different intervals: #4TD-#1TM (TDR=80%), #3TD-#1TM (TDR=75%), and #2TD-#1TM (TDR=66.6%). Here, #TD and #TM are the number of frames applying the TBD and TBM operations in a series. For our Decode-MOT, we tune TDRs of the Decode-MOT by changing θ_{det} to evaluate both trackers with the almost same TDR.

Table I shows detailed comparison results between our Decode-MOT and the baseline. We observed that the accuracy gap between Decode-MOT and the baseline becomes larger as the TDR decreases. Specifically, when comparing results with TDR=100% and TDR=66% of the Decode-MOT, the speed is improved by 46.4% while reducing MOTA by 4.3%. However, in the baseline tracker, its MOTA score is greatly degraded by 8.5%. When comparing both trackers with other TDRs, our Decode-MOT shows much better accuracy while maintaining similar speeds. Additionally, we have conducted additional comparison with the baseline tracker with the hierarchal association in Table I for showing the effect of our decision coordinator. Compared to the baseline, our association method achieves a MOTA gain by 2.4% when TDR=100%.

TABLE II

EFFECTS OF THE SCENE CONTEXTUAL LEARNING WHEN TDR=52.5%

Name	Short-term Atten.	Long-term Atten.	MOTA	Hz
A1			54.6%	38.7
A2	✓		56.2%	38.8
A3 (Ours)	✓	✓	61.3%	38.0

TABLE III

EFFECTS OF THE TRACKING CONTEXTUAL LEARNING WHEN TDR=52.5% AND TDR=70.0%

TDR	Name	Cardinality Similarity	Motion Similarity	Motion Similarity w. GT	MT	FN	MOTA	Hz
52.5%	B1	✓			37.4%	5,269	59.6%	37.1
	B2		✓		27.3%	8,098	46.3%	37.4
	B3 (Ours)	✓	✓		43.3%	4,801	61.3%	38.0
	B4	✓		✓	9.7%	5,325	58.7%	38.6
70.0%	B1	✓			60.6%	3,338	67.9%	25.5
	B2		✓		57.4%	3,542	68.2%	25.9
	B3 (Ours)	✓	✓		63.0%	3,194	69.1%	24.6
	B4	✓		✓	60.6%	3,142	69.0%	26.3

TABLE IV

EFFECTS OF THE HIERARCHICAL CONFIDENCE ASSOCIATION

Name	Decision Coordinator	Hier. Conf. Asso.	TDR	MOTA	Hz
C1			100%	71.0%	21.4
C2	✓		90.2%	68.8%	23.9
C3 (Ours)	✓	✓	100%	73.4%	21.0
			90.2%	72.0%	23.8

Remarkably, we observed that our association method provides better MOTA score gains by 4.4%, 4.8%, and 4.6% when using an uniform interval for TDR=80.0%, TDR=75.0%, and TDR=66.6%, respectively. These results mean that our hierarchical confidence association is effective when operating tracking-by-motion. When comparing the baseline with our association method and Decode-MOT, we observed that our Decode-MOT improves MOTA scores consistently by about 2.2% on average when TBM operates (*i.e.* TDR=80.0%, TDR=75.0%, and TDR=66.6%). Furthermore, we observe that our Decode-MOT shows better MOT speeds despite the additional computational cost of the decision coordinator when TDR=75.0% and TDR=66.6%. This is because inaccurate decision can lead to false positives (*e.g.* false tracks and track fragments) which increase the computational cost for associating tracks and detections. From these results, we verify that our method can improve both the MOT speed and the MOTA score when alternating TBD and TBM compared to the baseline added our association method. Figure 1 shows more comparison results among recent methods. As shown, Decode-MOT shows 56.3% and 52.6% MOTA scores when TDR=49.8% and TDR=46.8%, respectively. Furthermore, Decode-MOT achieves the much better accuracy scores compared to recent trackers [3], [16]. From these results, we verify that our method is very effective for boosting the speed while minimizing MOTA reduction.

2) Scene Contextual Learning: We train and evaluate the Decode-MOT with different attention methods in Table II: (A1) does not exploit attentions; (A2) uses short-term attention only; (A3) uses both short-term and long-term attentions.

TABLE V

COMPARISONS OF DIFFERENT TBM METHODS ON MOT15 TRAINING SET. THE PERCENTAGE IN [·] SHOWS THE GAIN AS TDR DECREASES

TDR	Kalman filter (Ours)		Linear motion	
	MOTA↑	Hz↑	MOTA↑	Hz↑
100%	73.4%	21.0	73.4%	21.1
90.2%	72.0% **[1.4%↓]**	23.8 **[13.3%↑]**	70.1% **[3.3%↓]**	24.3 **[15.2%↑]**
80.2%	70.6% **[2.8%↓]**	24.9 **[18.8%↑]**	66.0% **[7.4%↓]**	24.5 **[16.1%↑]**
74.2%	70.3% **[3.1%↓]**	27.7 **[31.9%↑]**	63.0% **[10.4%↓]**	25.6 **[21.3%↑]**
66.4%	69.1% **[4.3%↓]**	30.7 **[46.4%↑]**	53.2% **[20.2%↓]**	30.7 **[45.5%↑]**

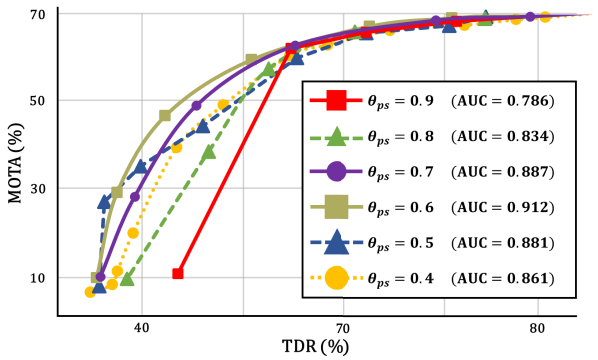

Fig. 5. Comparison of our Decode-MOT with different θ_{ps}..

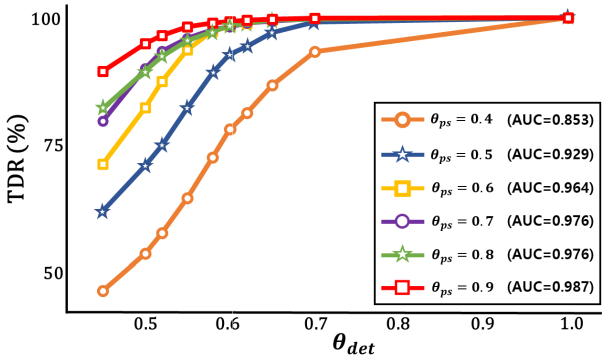

Fig. 6. Sensitivity analysis for θ_{ps} in terms of TDR and θ_{det}.

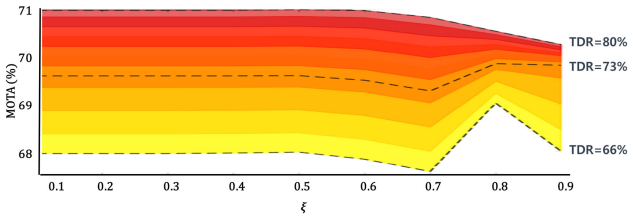

Fig. 7. Sensitivity analysis for ξ according to different TDRs. All other parameter values are fixed.

As shown in Table II, using both attentions (A3) shows better MOTA scores compared to (A1) and (A2). Also, we find that the long-term attention is very effective to increase MOTA with little complexity.

3) Tracking Contextual Learning: In Table III, we perform the comparisons of our contextual learning methods. (B1) exploits the cardinality similarity only; (B2) uses the motion similarity, which is similar to [20] and [21] because its *track* action is determined by IOU scores between GT boxes and tracked boxes; (B3) uses both contextual similarities for track-

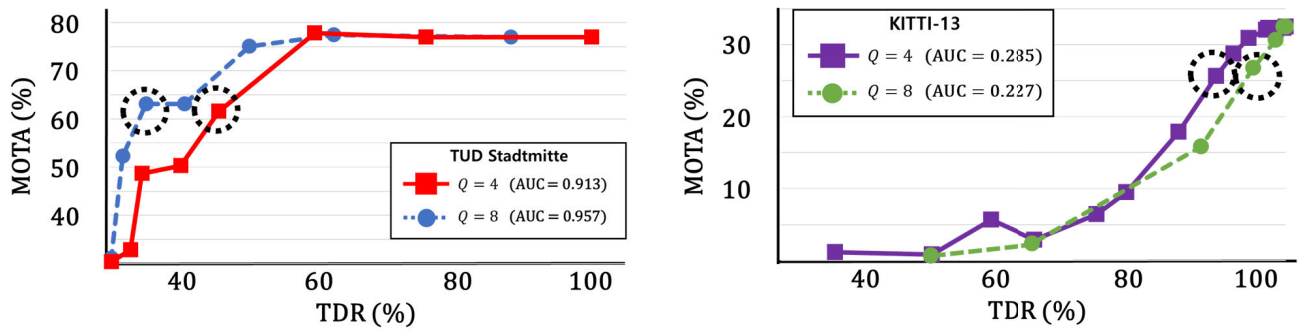

Fig. 8. Comparison of the Decode-MOT with different Q values on static (TUD-Stadtmitte) and dynamic (KITTI-13) sequences.

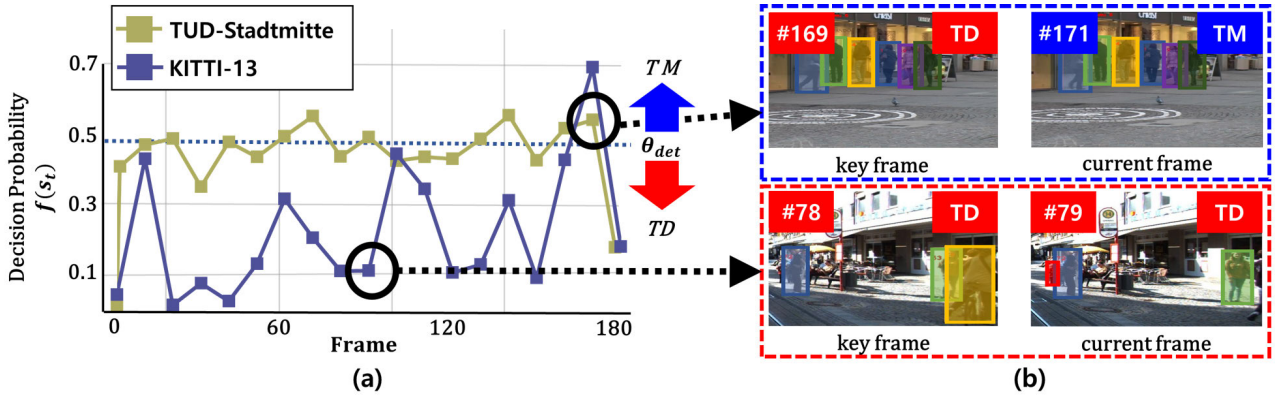

Fig. 9. Qualitative comparison on static (TUD-Stadtmitte) and dynamic (KITTI-13) sequences: (a) The variation of the decision probability $f(s_t)$. (b) The selected key and current frames at some $f(s_t)$ values.

Fig. 10. A failure case: some inaccurate tracking results of our Decode MOT on MOT16-12 (a) and MOT16-14 (b) sequences. True positives and false negatives are marked with orange and red boxes, respectively.

ing contextual learning; (B4) uses both contextual similarity too. However, we develop the motion similarity to use the object IDs of the ground truth (GT) for the motion similarity computation. We assign IDs of the GT objects $\mathcal{T}_t^{(GT)}$ to $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ by associating $\mathcal{T}_t^{(GT)}$ with $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ using the Hungarian method with IoU measure. Then, we evaluate the motion similarity scores for the tracks with the same IDs of $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ using Eq. (8).

As shown, (B1) and (B3) using the cardinality similarity show the better MOTA compared to (B2) using the motion similarity only when TDR=52.5%. It reflects that capturing the cardinality variation is more crucial for determining TBD and TBM mechanism. As a result, the FN (*i.e.* missed tracks) score dramatically increases ($4,801 \rightarrow 8,098$) without considering the cardinality. This verifies that the cardinality is also needed to be considered when determining key frames (*c.f.* [19], [20], [21], [33]). Although (B2) achieves slightly better scores compared to (B1) when TDR=70.0%, the FN score of (B2) is still worse than one of (B1). On the other hand, we note that (B3) using both similarities consistently achieves the best MOTA score whenever TDR is changed. Although the direct comparison with the tracker [20], [21] is not made,[3] this ablation study indicates that our Decode-MOT with both contextual similarities could achieve the better scores than those using the motion similarity only. Also, when comparing of (B3) and (B4) scores at TDR=70.0%, both trackers achieve the similar MOTA score. However, for the speed (B4) shows the better than (B3). It is because that in (B4) the ratio of mostly tracked trajectories is decreased (63.0% $\rightarrow$ 60.6%). We expect that the reduction of the MT score is because the decision of (B4) is somewhat biased toward the accurate tracks which can be associated with the GT.

In addition, (B4) shows the lower MOTA score than (B3) at TDR=52.5%. Furthermore, (B4) shows the lowest ratio of mostly tracked trajectories among (B1)-(B4). We expect that the performance degradation of (B4) is due to the overfitting of the coordinator by too precise association. More concretely,

[3]Due to the different down-streaming task (*i.e.* visual object tracking) and not opened codes to the public.

TABLE VI

COMPARISON WITH THE SOTA TRACKERS ON MOTCHALLENGE 16/17/20 TEST SETS. ALL TRACKERS ARE ONLINE-TRACKING METHODS AND USE PRIVATE DETECTORS

Tracker	Publication	Year	MOTA↑	HOTA↑	IDF1↑	FP↓	FN↓	MT↑	ML↓	FG↓	IDs↓	Hz↑	GPU
MOT16 Test Set													
POI [11]	ECCV	2016	66.1%	-	65.1%	5,061	55,914	34.0%	20.8%	3,093	805	9.9	GTX 970
EAMTT [60]	ECCV	2016	52.5%	41.9%	53.3%	4,407	81,223	19.0%	34.9%	1,321	910	12.2	-
DeepSORT [1]	ICIP	2017	61.4%	50.1%	62.2%	12,852	56,668	24.9%	13.8%	2,008	781	17.4	M6000
VMaxx [61]	ICIP	2018	62.6%	41.7%	49.2%	10,604	56,182	32.7%	21.1%	1,534	1,389	6.5	GTX 1080
RAN [62]	WACV	2018	63.0%	52.1%	63.8%	13,663	53,248	39.9%	22.1%	1,251	482	1.6	Titan X
HOGM [63]	ICPR	2018	64.8%	-	73.5%	13,470	49,927	40.6%	22.0%	1,050	794	18.2	-
LM_CNN [64]	Neurocomputing	2019	67.4%	-	61.2%	10,109	48,435	38.2%	19.2%	1,034	981	1.7	-
Tube_TK [65]	CVPR	2020	64.0%	48.7%	59.4%	10,962	53,626	33.5%	19.4%	1,366	1,117	1.0	Titan Xp
CenterTrack [25]	ECCV	2020	69.6%	-	60.7%	10,458	42,805	-	-	-	2,124	17.5	Titan Xp
CTracker [66]	ECCV	2020	67.6%	48.8%	57.2%	8,934	48,305	32.9%	23.1%	3,112	1,897	6.8	Tesla P40
QDTrack [67]	CVPR	2021	69.8%	54.5%	67.1%	9,861	44,050	41.6%	19.8%	2,653	1,097	20.3	GTX 1080ti
TraDeS [68]	CVPR	2021	70.1%	53.2%	64.7%	8,091	45,210	37.3%	20.0%	1,575	1,144	22.3	RTX 2080ti
GSDT [69]	ICRA	2021	74.5%	56.6%	68.1%	8,913	36,428	41.2%	17.3%	2,670	1,229	1.6	Titan Xp
CorrTracker [22]	CVPR	2021	76.6%	61.0%	74.3%	10,860	30,756	47.8%	13.3%	1,709	979	15.9	Tesla V100
HTA [70]	PRL	2021	62.4%	51.8%	64.2%	19,071	47,839	37.5%	12.1%	2,529	1,619	19.7	RTX 2080ti
CSTrack [71]	IEEE TIP	2022	75.6%	-	73.3%	9,646	33,777	42.8%	16.5%	-	1,121	16.4	RTX 2080ti
MeMOT [72]	CVPR	2022	72.6%	57.4%	69.7%	14,595	34,595	44.9%	16.6%	845	-	-	-
MTrack [73]	CVPR	2022	72.9%	-	74.3%	19,236	29,554	50.6%	15.7%	-	642	-	-
Decode-MOT (Ours)			74.7%	60.2%	73.0%	9,590	35,507	43.7%	17.0%	1,954	1,094	21.6	Titan Xp
MOT17 Test Set													
CenterTrack [25]	ECCV	2020	67.8%	-	64.7%	18,498	160,332	26.4%	31.9%	-	3,039	17.5	Titan Xp
CTracker [66]	ECCV	2020	66.6%	49.0%	57.4%	22,284	160,491	32.2%	24.2%	9,114	5,529	6.8	Tesla P40
QDTrack [67]	CVPR	2021	68.7%	53.9%	66.3%	26,589	146,643	40.6%	21.9%	8,091	3,378	20.3	GTX 1080ti
CrowdTrack [74]	AVSS	2021	75.6%	60.3%	73.6%	25,950	109,101	46.5%	12.2%	-	2,544	-	-
TraDeS [68]	CVPR	2021	69.1%	52.7%	63.9%	20,892	150,060	36.4%	21.5%	4,833	3,555	17.5	RTX 2080ti
Tube_TK [65]	CVPR	2021	63.0%	48.0%	58.6%	27,060	177,483	31.2%	19.9%	5,727	4,137	3.0	Titan Xp
GSDT [69]	ICRA	2021	73.2%	55.2%	66.5%	26,397	120,666	41.7%	17.5%	8,604	3,891	4.9	Titan Xp
CorrTracker [22]	CVPR	2021	76.5%	60.7%	73.6%	29,808	99,510	47.6%	12.7%	6,063	3,369	15.6	Tesla V100
TransCenter [75]	arXiv	2021	73.2%	54.5%	62.2%	23,112	123,738	40.8%	18.5%	9,519	4,614	1.0	Titan RTX
MOTR [76]	arXiv	2021	73.4%	57.8%	68.6%	-	-	-	-	-	2,439	7.5	Tesla V100
CSTrack [71]	IEEE TIP	2022	74.9%	-	72.3%	23,847	114,303	41.5%	17.5%	-	3,567	16.4	RTX 2080ti
MeMOT [72]	CVPR	2022	72.5%	56.9%	69.0%	37,221	115,248	43.8%	18.0%	2,724	-	-	-
MTrack [73]	CVPR	2022	72.1%	-	73.5%	53,361	101,844	49.0%	16.8%	-	2,028	-	-
TrackFormer [77]	CVPR	2022	74.1%	57.3%	68.0%	34,602	108,777	47.3%	10.4%	4,221	2,829	5.7	-
Decode-MOT (Ours)			73.2%	59.6%	72.0%	26,484	121,293	42.4%	18.5%	6,051	3,363	21.6	Titan Xp
MOT20 Test Set													
GSDT [69]	ICRA	2021	67.1%	53.6%	67.5%	31,507	135,395	53.1%	13.2%	9,878	3,230	1.5	Titan Xp
CorrTracker [22]	CVPR	2021	65.2%	-	69.1%	79,429	95,855	66.4%	8.9%	-	5,183	8.5	Tesla V100
FairMOT [16]	IJCV	2021	61.8%	54.6%	67.3%	103,440	88,901	68.8%	7.6%	7,874	5,243	13.2	RTX 2080ti
CrowdTrack [74]	AVSS	2021	70.7%	55.0%	68.2%	21,928	126,533	54.9%	12.1%	-	3,198	-	-
TransCenter [75]	arXiv	2021	58.5%	43.5%	49.6%	64,217	146,019	48.6%	14.9%	9,581	4,695	1.0	Titan RTX
CSTrack [71]	IEEE TIP	2022	66.6%	-	68.6%	25,404	144,358	50.4%	15.5%	-	3,196	4.5	RTX 2080ti
MeMOT [72]	CVPR	2022	63.7%	54.1%	66.1%	47,882	137,983	57.5%	14.3%	1,938	-	-	RTX 2080ti
MTrack [73]	CVPR	2022	63.5%	-	69.2%	96,123	86,946	68.8%	7.5%	-	6,031	-	-
TrackFormer [77]	CVPR	2022	68.6%	54.7%	65.7%	20,348	140,373	53.6%	14.6%	2,474	1,532	5.7	-
Decode-MOT (Ours)			67.2%	54.5%	69.0%	35,217	131,502	57.1%	13.6%	7,084	2,805	12.2	Titan Xp

the motion similarity becomes higher since the more false or inaccurate tracks of $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ can be excluded by the preceding association $\mathcal{T}_t^{(TD)}$ and $\mathcal{T}_t^{(TM)}$ with $\mathcal{T}_t^{(GT)}$. Therefore, our decision coordinator tends to be biased more toward the motion context than the cardinality one as TDR decreases. However, when comparing the results of (B1) and (B2) in Table III, the cardinality similarity should much contribute to the decision coordinator at the lower TDR. Eventually, this comparison indicates that our self-supervision reflecting the performance of an applied detector is key for decision coordinator learning.

4) Hierarchical Confidence Association: We compare the effect of our association method as shown in Table IV. We observe that (C3) using our association improves the MOTA score by 3.2% compared to (C2) without the association. (C3) with TDR=90.2% achieves the better MOTA and speed compared to the baseline with TDR=100%. We also find that the cost of using our association is negligible when compared to (C2) and (C3). It shows that our association method is indeed an effective method to boost MOT accuracy.

5) Tracking-by-Motion: To compare different TBM methods, we implement a linear motion model. We estimate the linear motion for each object by computing the difference between center positions at the two recent previous frames from TBD. As shown in Table V, the TBM using Kalman filtering provides the better performance than using the linear motion. It also indicates that the quality of TBM can affect the overall performance more as TDR decreases.

6) Sensitivity Analysis for θ_{ps}: To analyze the sensitivity our Decode-MOT against θ_{ps}, we train different Decode-MOT by changing θ_{ps} and compare their MOTA scores for different TDRs. As shown in Fig. 5, the MOTA difference is marginal when TDR $\geq$ 60%. This means that θ_{ps} is not a sensitive parameter when TBD operates frequently. On the other hand, when TDR $<$ 60%, a high $\theta_{ps} = 0.9$ shows the steep MOTA reduction. This is because the decision coordinator is rather over-fitted since the pseudo labels of the GT $\mathcal{G}$ is highly biased to TD. Also, the decision of our coordinator could be biased to TM when $\theta_{ps} = 0.4$. Furthermore, we conduct a sensitivity

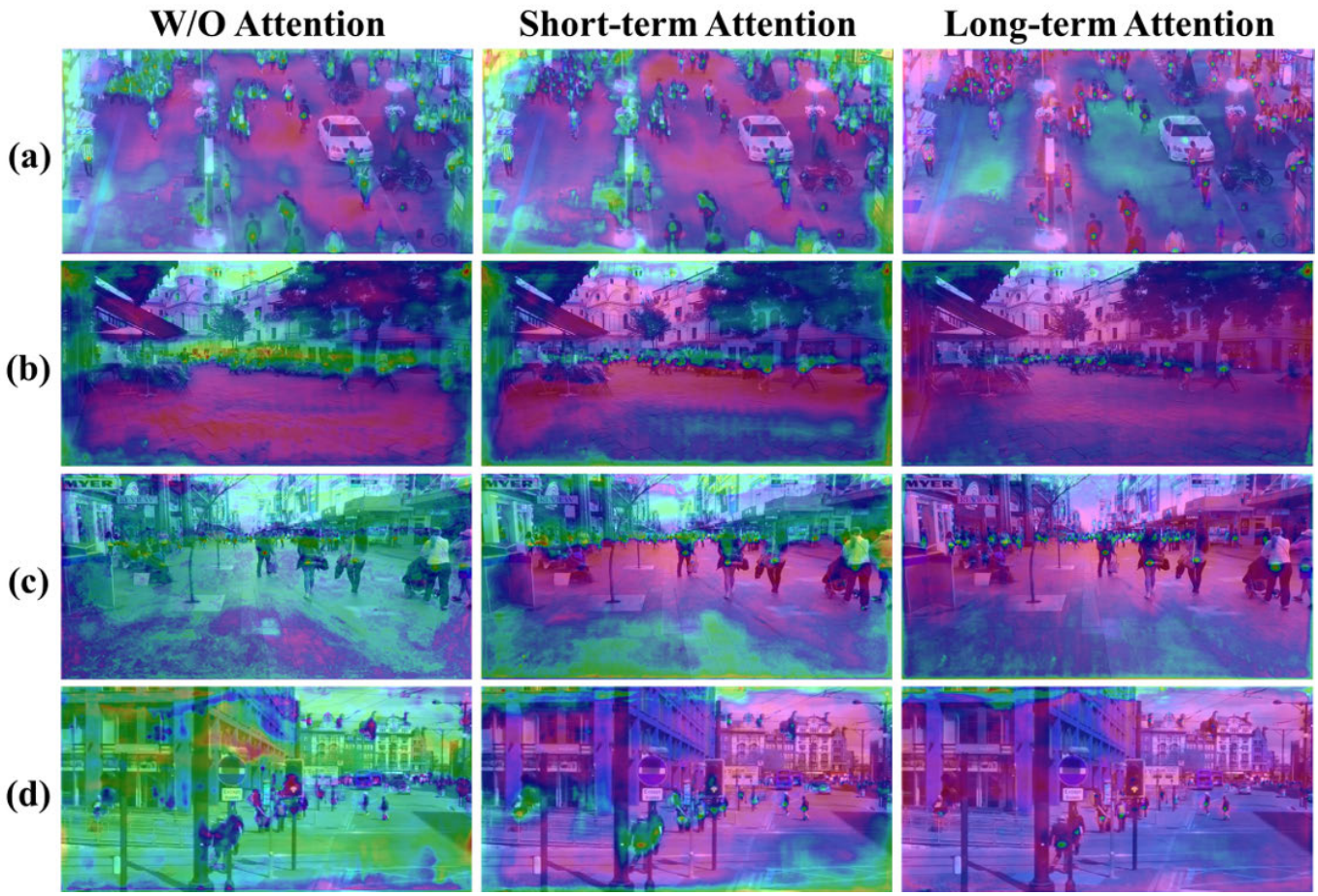

Fig. 11. Qualitative comparisons of the scene contextual learning. We visualize activation maps by applying different attention methods. All images are from MOTChallenge sets [23]: (a) MOT16-03, (b) MOT16-01, (c) MOT16-07, (d) MOT16-14.

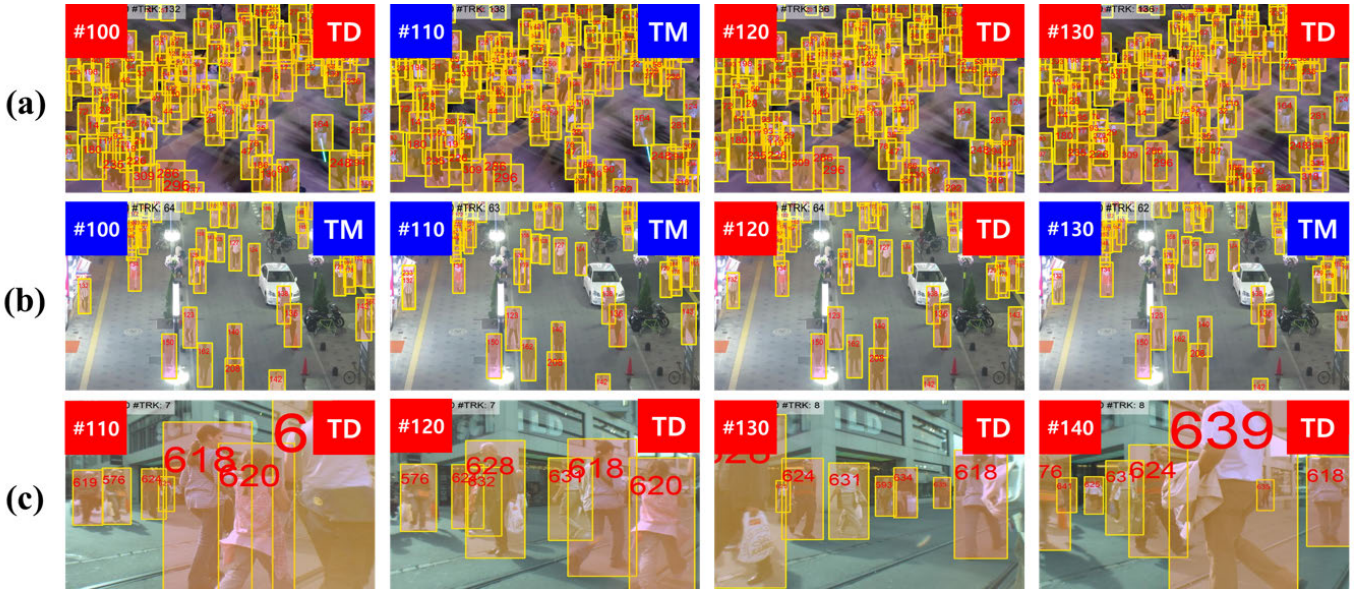

Fig. 12. Qualitative tracking results for our Decode-MOT. All images are from MOTChallenge sets [23]: (a)MOT20-04, (b)MOT16-03, (c)MOT16-06.

analysis for θ_{ps} in terms of TDR and θ_{det} as shown in Fig. 6. We observe that the TBD difference is marginal when θ_{det} is high. However, the TBD difference becomes larger as θ_{det} decreases. In other word, the low $\theta_{ps} = 0.4$ shows the lower TDR when $\theta_{det} \leq 0.7$ compared to the high $\theta_{ps} = 0.9$. It implies that θ_{ps} can affect TDR and θ_{det}. Therefore, setting the proper θ_{ps} is important to avoid the biased results in terms of TDR and θ_{det}.

7) Sensitivity Analysis for ξ: We conduct a sensitivity analysis for the degradation factor ξ. As shown in Fig. 7, the MOTA variation for different ξ is so marginal. This proves that our method is not sensitive to ξ.

8) Buffer Size Q: To investigate the effects of Q used in the scene contextual learning, we evaluate our trackers with different Q values. For more evaluation, we categorize MOT sequences into dynamic and static sequences in consideration of the amount of camera and object movement, and the similarity between consecutive frames for object cardinality and motions. As a result, we find out KITTI-13 and TUD-Stadtmitte are the most dynamic and static ones, respectively.

Figure 8 shows MOTA comparisons for different Q values on each sequence. For the dynamic sequence, we observe that using $Q = 4$ shows the higher AUC score than $Q = 8$. On the other hand, for the static sequence, $Q = 8$ achieves the higher AUC than $Q = 4$. It implies that the temporal buffer size Q relies on the similarity between the consecutive frames.

9) TBD and TBM Mechanism on Static and Dynamic Scenes: As shown in Fig. 9 (a), we show the variation of the decision probability $f(s_t)$ of the decision coordinator per frame. We observe that our coordinator tends to produce low $f(s_t)$ more frequently on the dynamic sequence. In Fig. 9 (b), we show the selected key and current frames at some $f(s_t)$ values. Both frames have the similar object cardinality and motions when $f(s_t)$ is high. In Fig. 8, we also mark points achieving 80% MOTA score over the score with TDR=100% using a black dotted circle. Those accuracies can be achieved with the much lower TDR on the static scene. Therefore, the overall tracking speed can be enhanced much more with low MOTA loss in the static sequence.

C. Qualitative Comparisons

In order to verify the effects of the scene contextual learning as described in Sec. III-C, we show the qualitative comparisons of activation maps with different attention methods as shown in Fig. 11. To this end, we present the activation maps without attentions, using the short-term attention only, and using both the long-term and short-term attentions. For visualizing them, we apply GradCAM [78] to the feature maps P_t and the channel weighted feature maps $P_t^{\star}$, and overlay them on their current images I_t. We observe that using both attentions can capture objects well than other methods. In specific, in dynamic sequences such as Fig. 11 (c) and (d), using both attentions can capture objects finer for the rapid scene context variations compared to using the short-term attention only. It indicates that our proposed scene contextual learning is beneficial to learning scene contexts. In some cases, inaccurate tracking results of our tracker occur as shown in Fig. 10. We found out that the false negative tracks are made by TBM for the small objects mostly. Since the small objects have very ambiguous appearance features, our decision coordinator would make an inaccurate decision with the low discriminative features in those scenes. We know that this is still one of the challenging issues for MOT. This problem could be resolved by designing the cost-sensitive loss for small objects or enhancing feature discriminativeness.

D. Comparison With State-of-the-Arts Methods

For comparing the recent trackers, we evaluate our Decode-MOT on MOT16, MOT17 and MOT20 test sets. For a fair comparison, we compare ours with the online tracking methods using private detectors. We set θ_{det} to 0.5 in the most ablation study and MOT16 and MOT17 comparisons with SOTA trackers, but we only set it to 0.05 for MOT20. The reason of using different the θ_{det} is that the mean crowd density of MOT20 (170.9) is about 5.5 times higher than ones of MOT16 (30.8) and MOT17 (31.8) sequences as shown in the related paper [23] and MOT benchmark websites (https://motchallenge.net/data/MOT20/, accessed on 20 May 2023). It means that many pedestrians appear and disappear frequently in MOT20 sequences. Therefore, cardinality similarities between consecutive frames in MOT20 are lower than ones in other sequences (MOT16 and MOT17). It makes our decision coordinator output a lower decision probability $f(s_t)$. Therefore, we set θ_{det} to 0.05 in order to encourage our decision coordinator to perform TBM more frequently and boost the MOT speed further.

As shown in Table VI, our Decode-MOT achieves 73.2% MOTA and 21.6Hz on the MOT17 test set, which are the remarkable speeds while achieving the higher MOTA and HOTA than other methods [25], [65], [67], [68], [69], [75]. In addition, our Decode-MOT shows comparable accuracies with recent tracking methods [71], [76], [77] while showing better speed. In MOT16 test sets, we achieve remarkable accuracies and speed. Only two trackers [22], [71] show higher accuracies but lower speed compared to ours. In MOT20 test sets, our Decode-MOT shows 67.2% MOTA and 12.2Hz, which are competitive scores compared to recent tracking methods. In particular, our tracker generates much more gains for both speed and accuracy on all the MOT sets over the recent trackers evaluated with the same Titan Xp GPU. Figure 12 shows our Decode-MOT tracking results on MOTChallenge dataset. Our system tracks the most objects even though the objects are frequently occluded while reducing the number of detector operation for boosting speed. For showing more results, we provide the supplementary videos.

VI. Conclusion

For real-time and high accurate MOT, we propose a novel Decode-MOT which can determine the best tracking-by-detection (TBD) or tracking-by-motion (TBM) mechanism during onilne MOT. To this end, we present the scene contextual learning using long-term attention for generating more discriminative features between consecutive frames. Because the TBD/TBM mechanism can be different for the nature of MOT methods, we propose the self-supervised learning based on tracking contextual similarities in terms of cardinality and motion. For the more robust association, we present a hierarchical confidence association which can reduce the association ambiguity step-by-step. From the extensive ablation studies and comparisons with the recent methods, we verify that our method is beneficial to boost the overall speed and accuracy together. We believe that our work could be an important guideline for future real-time MOT methods. Since our

Decode-MOT reduces the complexity of the MOT algorithm at the system level, our method could be compatible with tracking methods aiming at reducing the model complexity. Therefore, the knowledge distillation [42] and atrous spatial pyramid pooling [79] could be incorporated into our method, and we expect great synergy can be made by combining our method with such lightweight models. In addition, enhancing the decision coordinator itself is critical for boosting tracking accuracy and speed more. Since capturing the context difference of different frames is an important cue for our coordinator, we can strengthen the long-term attention learning with multi-head attention modules [28].

REFERENCES

[1] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[2] P. Voigtlaender et al., "MOTS: Multi-object tracking and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7934–7943.

[3] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 107–122.

[4] A. Hornakova, T. Kaiser, P. Swoboda, M. Rolinek, B. Rosenhahn, and R. Henschel, "Making higher order MOT scalable: An efficient approximate solver for lifted disjoint paths," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6310–6320.

[5] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[9] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6402–6411.

[10] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9626–9635.

[11] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: Multiple object tracking with high performance detection and appearance feature," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 36–42.

[12] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.

[13] Z. Lu, V. Rathod, R. Votel, and J. Huang, "RetinaTrack: Online single stage joint detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14656–14666.

[14] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[15] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13903–13912.

[16] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 3069–3087, Sep. 2021.

[17] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3057–3065.

[18] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4141–4150.

[19] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7210–7218.

[20] H. Luo, W. Xie, X. Wang, and W. Zeng, "Detect or track: Towards cost-effective video object detection/tracking," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 8803–8810.

[21] K. Chen et al., "Optimizing video object detection via a scale-time lattice," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7814–7823.

[22] Q. Wang, Y. Zheng, P. Pan, and Y. Xu, "Multiple object tracking with correlation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3875–3885.

[23] P. Dendorfer et al., "MOTChallenge: A benchmark for single-camera multiple target tracking," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 845–881, Apr. 2021.

[24] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial–temporal relation networks for multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3987–3997.

[25] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 474–490.

[26] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6246–6256.

[27] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6767–6776.

[28] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[29] O. Wiles, S. Ehrhardt, and A. Zisserman, "Co-attention for conditioned image matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15915–15924.

[30] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 366–382.

[31] S. Guo, J. Wang, X. Wang, and D. Tao, "Online multiple object tracking with cross-task synergy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8132–8141.

[32] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial–temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4846–4855.

[33] C.-H. Yao, C. Fang, X. Shen, Y. Wan, and M.-H. Yang, "Video object detection via object-level temporal aggregation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 160–177.

[34] Y. Cui, C. Jiang, L. Wang, and G. Wu, "MixFormer: End-to-end tracking with iterative mixed attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13598–13608.

[35] L. Lin, H. Fan, Z. Zhang, Y. Xu, and H. Ling, "SwinTrack: A simple and strong baseline for transformer tracking," in *Proc. 36th Conf. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 16743–16754.

[36] Z. Liang and J. Shen, "Local semantic Siamese networks for fast tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 3351–3364, 2020.

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

[38] J. Shen, X. Tang, X. Dong, and L. Shao, "Visual object tracking by hierarchical attention Siamese network," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3068–3080, Jul. 2020.

[39] M. Cen and C. Jung, "Fully convolutional Siamese fusion networks for object tracking," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*. Athens, Greece: Springer, Oct. 2018, pp. 3718–3722.

[40] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1320–1329.

[41] B. Yan, H. Peng, K. Wu, D. Wang, J. Fu, and H. Lu, "LightTrack: Finding lightweight neural networks for object tracking via one-shot architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15175–15184.

[42] J. Shen, Y. Liu, X. Dong, X. Lu, F. S. Khan, and S. Hoi, "Distilled Siamese networks for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8896–8909, Dec. 2022.

[43] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[44] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, "Boosting self-supervised learning via knowledge transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9359–9367.

[45] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-supervised GANs via auxiliary rotation loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12146–12155.

[46] S. Shao et al., "CrowdHuman: A benchmark for detecting human in a crowd," 2018, *arXiv:1805.00123*.

[47] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[48] S. Karthik, A. Prabhu, and V. Gandhi, "Simple unsupervised multi-object tracking," 2020, *arXiv:2006.02609*.

[49] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[50] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[51] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

[52] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 304–311.

[53] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4457–4465.

[54] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3376–3385.

[55] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, "Person re-identification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3346–3355.

[56] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[57] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, May 2008, Art. no. 246309.

[58] J. Luiten et al., "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 548–578, Feb. 2021.

[59] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2403–2412.

[60] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 84–99.

[61] X. Wan, J. Wang, Z. Kong, Q. Zhao, and S. Deng, "Multi-object tracking using online metric learning with long short-term memory," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 788–792.

[62] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 466–475.

[63] Z. Zhou, J. Xing, M. Zhang, and W. Hu, "Online multi-target tracking with tensor-based high-order graph matching," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1809–1814.

[64] M. Babaee, Z. Li, and G. Rigoll, "A dual CNN–RNN for multiple people tracking," *Neurocomputing*, vol. 368, pp. 69–83, Nov. 2019.

[65] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "TubeTK: Adopting tubes to track multi-object in a one-step training model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6307–6317.

[66] J. Peng et al., "Chained-Tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 145–161.

[67] J. Pang et al., "Quasi-dense similarity learning for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 164–173.

[68] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12347–12356.

[69] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13708–13715.

[70] X. Lin, C.-T. Li, V. Sanchez, and C. Maple, "On the detection-to-track association for online multi-object tracking," *Pattern Recognit. Lett.*, vol. 146, pp. 200–207, Jun. 2021.

[71] C. Liang, Z. Zhang, X. Zhou, B. Li, S. Zhu, and W. Hu, "Rethinking the competition between detection and ReID in multiobject tracking," *IEEE Trans. Image Process.*, vol. 31, pp. 3182–3196, 2022.

[72] J. Cai et al., "MeMOT: Multi-object tracking with memory," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8080–8090.

[73] E. Yu, Z. Li, and S. Han, "Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8824–8833.

[74] D. Stadler and J. Beyerer, "On the performance of crowd-specific detectors in multi-pedestrian tracking," in *Proc. 17th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2021, pp. 1–12.

[75] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "TransCenter: Transformers with dense representations for multiple-object tracking," 2021, *arXiv:2103.15145*.

[76] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "MOTR: End-to-end multiple-object tracking with transformer," 2021, *arXiv:2105.03247*.

[77] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8834–8844.

[78] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[79] Z. Zhao, S. Zhao, and J. Shen, "Real-time and light-weighted unsupervised video object segmentation network," *Pattern Recognit.*, vol. 120, Dec. 2021, Art. no. 108120.

Seong-Ho Lee received the B.S. degree in computer science and engineering from Incheon National University in 2019, and the M.S. degree from the Department of Electronic Computer Engineering, Inha University, South Korea. He was a full-time Researcher at Inha University in 2022. He is currently working at SK Hynix Inc. His current research interests are multi-object tracking, object detection, self-supervised learning, generative adversarial networks, and multi-scale representation.

Dae-Hyeon Park received the B.S. degree in computer engineering from Inha University in 2020, where he is currently pursuing the M.S. degree with the Department of Electronic Computer Engineering. His current research interests include single/multi-object tracking, multi-modal learning, real-time systems, and self-attention mechanism.

Seung-Hwan Bae (Member, IEEE) received the B.S. degree in information and communication engineering from Chungbuk National University in 2009 and the M.S. and Ph.D. degrees in information and communications from the Gwangju Institute of Science and Technology (GIST) in 2010 and 2015, respectively. He was a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), South Korea, from 2015 to 2017. He was an Assistant Professor with the Department of Computer Science and Engineering, Incheon National University, South Korea, from 2017 to 2020. He is currently an Associate Professor with the Department of Computer Engineering, Inha University. His research interests include object tracking, object detection, generative model learning, continual learning, and on-device ML.