LTTrack: Rethinking the Tracking Framework for Long-Term Multi-Object Tracking

Jiaping Lin, Gang Liang, Rongchuan Zhang

***Abstract*—Long-term tracking is a commonly overlooked yet practical scenario in multi-object tracking. Handling occlusion and re-identifying long-lost targets are the main challenges for effective long-term tracking. In occlusion scenarios, both appearance and motion features can be unreliable, leading to association failure. For long-lost targets, predicting their long-term motion suffers from severe error accumulation, making the target re-identification challenging. In this paper, we propose a multi-object tracker called LTTrack for long-term tracking. For occlusion handling, we develop the Position-Based Association (PBA) module, which encodes relative and absolute positions as interaction and motion features for association. With interaction features, PBA can handle occlusion scenes where appearance and motion features are unreliable. For long-lost target re-identification, the Long-Term Motion (LTM) model is devised. By encoding long-term motion trends of targets for long-term motion prediction, LTM alleviates the error accumulation problem. Moreover, to prevent the erroneous deletion of long-lost tracks, we propose the Zombie Track Re-Match (ZTRM) strategy to re-identify long-lost targets so that they will neither be prematurely deleted nor disrupt the association of other tracks. Extensive experiments conducted on MOT17, MOT20, and DanceTrack demonstrate that LTTrack achieves performance comparable to state-of-the-art methods. The code and models are available at https://github.com/Lin-Jiaping/LTTrack.**

***Index Terms*—Multi-object tracking, long-term tracking, tracking-by-detection, motion model, data association.**

I. Introduction

MULTI-object tracking (MOT) aims to track all the objects of interest in the input video, determining the position of each object in every frame of the video. As a fundamental task in computer vision, MOT finds widespread applications in fields such as autonomous driving [1], intelligent surveillance [2], human-computer interaction [3], and more. Existing MOT methods can be categorized into two groups: tracking-by-query (TBQ) [4]–[9] and tracking-by-detection (TBD) [10]–[14]. TBQ methods incorporate the Transformer [15] and utilize the track query to decode the positions of tracked targets in each frame for tracking purposes. TBD methods consist of two main components: object detection and data association. Object detection is responsible for locating all objects in each frame. Data association links detected objects with the same identity across different frames based on appearance, motion, and other cues to form tracks. Comparing TBQ and TBD, the former demonstrates greater potential in scenarios involving complex target motion because of the robust modeling capabilities of Transformer for time-series data. However, TBQ models usually involve numerous attention operations, introducing high computational costs. Conversely, TBD models have a lower computational overhead. Besides, TBQ models necessitate large-scale datasets to train high-performance models, resulting in inferior performance compared to TBD models on small datasets like [16], [17].

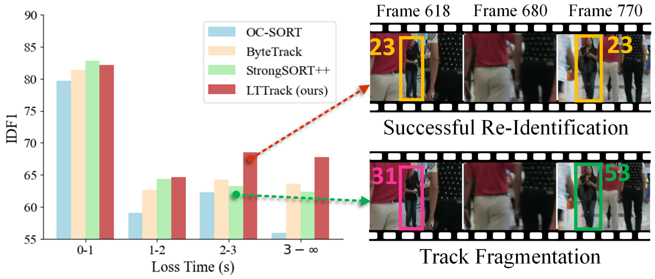


Fig. 1. Performance comparison between state-of-the-art trackers [11], [13], [18] and LTTrack on the MOT17 validation set. The horizontal axis represents loss durations, while the vertical axis denotes the identity assignment performance, i.e., IDF1 [19]. LTTrack excels in tracking long-lost targets.

State-of-the-art (SOTA) MOT algorithms can maintain high tracking performance in simple scenarios, such as short-term occlusion. However, long-term tracking remains a significant challenge. Note that we define long-lost targets as targets lost for over 1 second and long-term tracking as the task of tracking targets for over 1 second, which is similar to [20]. Handling occlusion and re-identifying long-lost targets are the primary difficulties in long-term tracking. Occlusion can lead to a track failing to associate with any detections, i.e. target loss, or matching with an incorrect detection and breaking into two or more tracklets, i.e. track fragmentation. As depicted in Fig. 1, with the increase in loss duration, existing methods exhibit a notable decrease in IDF1 score on the MOT17 dataset [16]. Specifically, when the loss duration exceeds 3 seconds, existing methods show a decrease of more than 15% in IDF1 score. There are two reasons for this phenomenon. First, after long-term loss, the appearance of targets may undergo significant changes and motion models cannot accurately predict target motion due to the lack of detections for correction. This complicates the target re-identification process based on appearance and motion features. Secondly, track management strategies in existing methods are not applicable to long-term

Jiaping Lin, Gang Liang, and Rongchuan Zhang are with the School of Cyber Science and Engineering, Sichuan University, Chengdu 610200, China (e-mail: linjiaping1@stu.scu.edu.cn; lianggang@scu.edu.cn; zhangrongchuan1@stu.scu.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grant 62162057, in part by the Sichuan Science and Technology Program under Grant 2022YFG0182, in part by Luzhou Project of Sichuan University under Grant 2023CDLZ-2, in part by the Local projects of the Ministry of Education under Grant 2020CDZG-18, 2021CDLZ-12, and 2021CDZG-11. (Corresponding author: Gang Liang)

tracking scenarios. Specifically, existing track management strategies [10], [13], [21], [22] assume that tracks which have been lost for more than n frames are terminated and delete these tracks. To prevent terminated tracks from interfering with the association of other tracks, the value of n is set relatively small, which leads to the mistaken deletion of long-lost targets.

To address the above problem, existing methods concentrate on extracting more robust and distinctive features to improve association accuracy. [7], [14], [23] create a memory bank to store the historical appearance features of tracks, thereby creating robust appearance features. However, as the occlusion time increases, the potential associations will accumulate, leading to a rise in ambiguous matches. Consequently, relying solely on appearance features for re-identifying long-lost targets is insufficient. Additionally, existing simple motion models [13], [24] cannot aid in re-identification due to the error accumulation issue in long-term motion prediction. Therefore, [20] devises a sophisticated motion model for long-term motion prediction, enabling successful re-identification. Nevertheless, [20] requires complex motion modeling for different scenes, which makes the model less robust. Unfortunately, using the motion model to accomplish the re-identification of long-lost targets remains a challenging research direction and is often overlooked. Furthermore, all the aforementioned methods focus on addressing the long-term tracking issue by extracting more robust features, but they overlook the impact of track management on long-term tracking scenarios.

In this paper, we propose a novel algorithm for long-term multi-object tracking based on the TBD paradigm, namely **L**ong-**T**erm **Track** (LTTrack). In particular, our LTTrack focuses on three perspectives to tackle the unique challenges posed by long-term tracking scenarios: fragmentation prevention, long-term motion prediction, and lost track management.

In terms of track fragmentation, we propose the Position-Based Association (PBA) module to ensure robust association and reduce fragmentation in crowded scenes with heavy occlusion. Interaction among targets is an important contextual cue. However, previous methods only use it for motion prediction [24]–[27] or missed detection recovery [28], without explicitly leveraging it as an association cue. Therefore, we innovatively utilize the interaction feature for data association. Specifically, our PBA module uses the relative positions of targets to encode interaction features and the absolute positions to encode motion features. Afterward, the two features are combined to compute the affinity between detections and tracks for association.

In terms of long-term motion prediction, as mentioned above, how to approach error accumulation and design a simple yet effective long-term motion model is a key problem in long-term tracking. We argue that the reason for error accumulation lies in the tendency of existing motion models to rely solely on short-term motion features for motion prediction. For example, existing models tend to predict the velocity of a target in the next frame based on its velocity in adjacent frames. For long-term motion prediction across multiple frames, subsequent motion predictions are built upon prior predictions that may contain errors, resulting in error accumulation. Hence, relying solely on short-term motion features is inadequate. It is crucial to consider the overall motion trends of tracks. Consequently, we propose the Long-Term Motion (LTM) model, which extracts short-term motion features, interaction features, and long-term motion features from tracks for long-term prediction. For long-term motion feature extraction, we observe that variations in the dimensions of the bounding box can reflect the target motion magnitude. Furthermore, the historical velocities of the target can also reveal its motion speed trend. Therefore, the historical variations in the tracking box and the velocity of tracks are encoded as long-term motion features to represent the motion trends.

In terms of lost track management, it is essential to ensure that long-lost targets have the opportunity for re-identification, hence they should not be promptly deleted. However, retaining long-lost tracks can lead to an excessive number of tracks in the track pool, expanding the search space for data association and disrupting the association of normal tracks. Therefore, it is necessary to separate long-lost tracks from others for more effective management. In light of this analysis, we designate tracks lost for more than m_{lost} frames as "zombie tracks" and propose the Zombie Track Re-Match (ZTRM) association strategy to deal with zombie tracks individually based on long-term motion prediction and appearance features.

As shown in Fig. 1, LTTrack outperforms existing methods as the loss time increases. Furthermore, we conduct extensive experiments on MOT17 [16], MOT20 [17], and DanceTrack [29] datasets. The experimental results illustrate our LTTrack can achieve competitive performance with SOTA methods.

The main contributions are summarized as follows:

- The LTM module is proposed to address the error accumulation problem and attain accurate long-term motion prediction by introducing long-term motion features.
- The PBA module is proposed to solve association failure due to the ineffectiveness of appearance and motion features in crowded scenes using the interaction feature.
- The ZTRM module is designed to manage long-lost tracks using a separate association strategy.
- A tracker LTTrack is proposed by combining the above innovative modules to realize stable long-term tracking.

II. Related Work

In this section, we delve into the aspects of existing MOT research relevant to our work, including motion models, association algorithms, and track management strategies.

A. Motion Model in MOT

The motion model in MOT is used to predict the positions of tracks in the next frame. Existing motion models can be classified into filter-based models and data-driven models [24].

Filter-based methods model motion prediction as a state estimation problem, employing Bayesian estimation to predict the motion state of the track. As one of the Bayesian filters, the Kalman Filter (KF) [30] finds wide applications in MOT. [11], [12], [31]–[33] assume that the target undergoes linear motion between consecutive frames and utilize KF as a linear motion model for motion prediction. However, in complex scenes, targets often engage in non-linear motion, resulting in a significant decrease in the effectiveness of KF. To address this issue, [13], [18], [34]–[36] improve the implementation of KF in MOT. For instance, OC-SORT [13] introduces an observation-centric tracking method to reduce error accumulation and enhance the robustness of KF. BoT-SORT [35] enhances KF by directly regressing the center coordinates, width, and height of the tracking box, leading to more precise motion prediction. However, the aforementioned methods primarily concentrate on reducing estimation errors of KF but may still struggle to model non-linear motion accurately. Consequently, these methods continue to underperform in complex scenarios [9].

To overcome the limitations of filter-based methods, data-driven methods incorporate deep neural networks to enable non-linear motion prediction. [37]–[41] predict the motion of the target in the subsequent frames through iterative regression. [42], [43] leverage RNNs to extract temporal features of tracks, which are subsequently used to predict the motion state of the target in the next frame. [27] utilizes Transformer to model motion information across multiple frames for intricate motion prediction. Considering the mutual influence of adjacent targets, [24]–[26], [44] model the interactions between targets for motion prediction. However, these methods are unsuitable for long-term motion prediction. When dealing with scenarios involving long-term occlusion of targets and requiring motion prediction for multiple frames, these methods may suffer from significant bias due to error accumulation. To track long-term occluded targets, QuoVadis [20] applies trajectory forecasting techniques to MOT. It initially constructs tracks in the Bird's Eye View (BEV) space using homography transformation. Then, it employs a trajectory forecasting model to predict the motion of the target across multiple frames. This approach effectively alleviates error accumulation in long-term motion prediction. However, it introduces a homography transformation module, which requires precise homography transformations for every tracking scenario. This limits the robustness and practicality of the model.

Comparing our LTM with existing motion models, in terms of model structure, LTM shares similarities with [25]: both models extract temporal and interaction features to support motion prediction. However, the key difference is that LTM focuses on long-term tracking. Specifically, LTM encodes the long-term motion trend of the track to reduce error accumulation in long-term motion prediction. In terms of application scenarios, both LTM and [20] are designed for long-term tracking. However, unlike [20] which incorporates an extra homography estimation model and a trajectory forecasting model, LTM consists of a simple network structure while achieving superior performance.

B. Data Association in MOT

The data association in MOT aims at linking detections and tracks with the same identity in a frame. A kind of simple association method [13], [31], [45], [46] calculates the Intersection over Union (IoU) between the detection boxes and the tracking boxes to assess the motion affinity between the two sets. Then the motion affinity is input to the Hungarian algorithm [47] to complete association. Although such methods are efficient and fast due to their sole reliance on motion information, they may not be suitable for complex scenarios with dense objects, irregular motion, etc. In these cases, densely distributed targets often exhibit similar positions and the motion cues are ambiguous. For complex scenarios, [18], [22], [48]–[51] combine appearance and motion affinities for association. These approaches extract appearance features using a ReID network. Then, the appearance affinity between tracks and detections is calculated based on appearance features. Afterward, the association is performed based on appearance and motion affinities. For example, [18], [22], [50], [51] compute a weighted sum of appearance affinity and motion affinity for more robust association. [49] designs a multi-level association strategy, which relies on the appearance affinity for association when the motion cue is unreliable. While the fusion of appearance and motion features is effective in many scenarios, it faces challenges in cases of heavy occlusion and complex motion modes. In such situations, appearance and motion features can become unreliable, inducing decreased accuracy in existing association methods.

Existing methods primarily focus on extracting robust appearance or motion features for association without considering other information in the scene, leading to performance degradation in challenging scenes. We note that in crowded scenes where appearance and motion features are unreliable, interaction features can effectively distinguish between different targets. Therefore, the PBA module is proposed to leverage interaction features to assist the association in crowded scenes.

C. Track Management in MOT

The track management strategy in MOT is employed to handle data association results, including track initialization, update, and deletion [52]. For the newly appeared targets during the tracking process, the track management strategy initializes them as new tracks and adds these tracks to the track pool. For successfully associated tracks and detections, the states of tracks are updated according to the matched detection. For tracks that fail to be associated, the track management strategy needs to determine whether the track is temporarily lost or has terminated, thereby deciding whether to remove the track from the track pool. Existing methods [10], [13], [49], [50], [53] introduce a threshold n for loss duration to decide whether a track is lost or terminated. If a track has been lost for more than n frames, it is considered terminated. The value of n is determined through experimental studies. However, relying solely on n to filter out terminated tracks poses limitations in long-term tracking scenarios. For re-identifying long-lost targets, a relatively large value of n is necessary to alleviate the erroneous deletion of these targets. Nevertheless, setting a large value for n results in a surplus of terminated tracks within the track pool, which hinders the association of regular tracks. To address this issue, ByteTrack [11] dynamically adjusts the value of n based on the characteristics of the tracking scene. However, this manual adjustment strategy leads to a lack of robustness and practicality of [11].

Different from existing methods, we preserve long-lost tracks in the track pool, preventing their premature deletion. Moreover, to mitigate ambiguous association caused

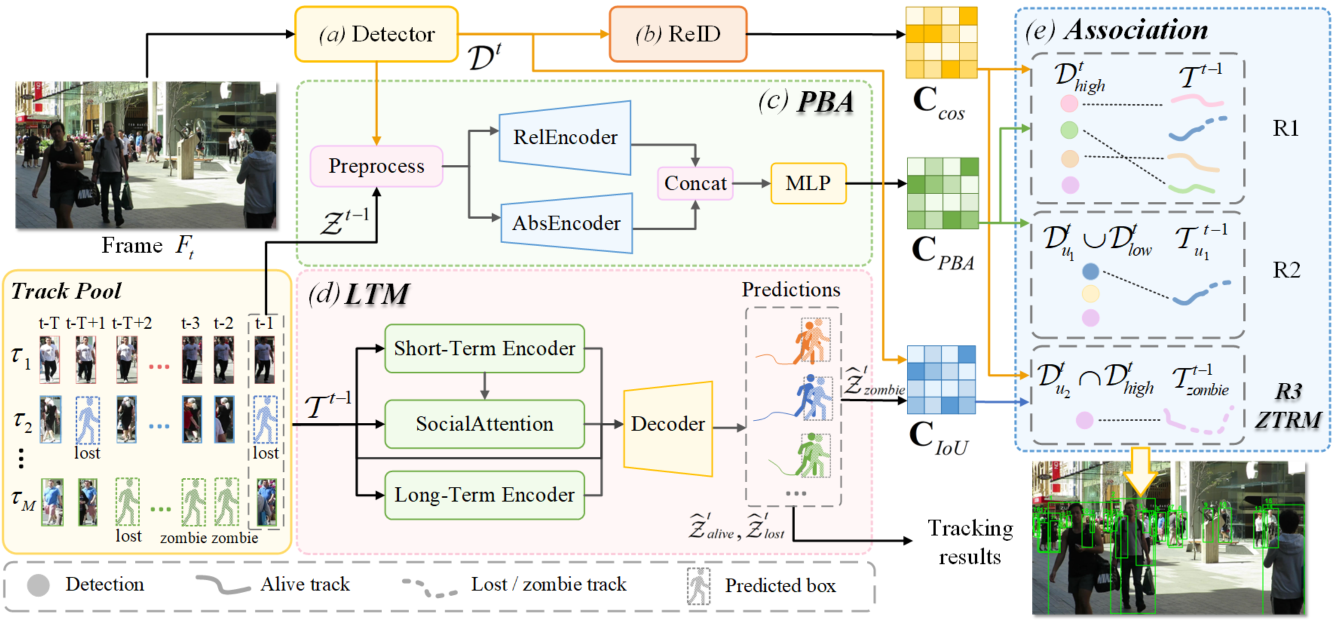


Fig. 2. **Overview of our LTTrack.** Given the frame F_t and tracks $\mathcal{T}^{t-1}$ up to frame $t-1$, LTTrack outputs the tracking results, i.e., tracks $\mathcal{T}^t$ up to frame t. There are five components in LTTrack: ***(a)*** a detector to obtain detections $\mathcal{D}^t$ in F_t, ***(b)*** a ReID module to extract appearance features for the calculation of appearance affinity between $\mathcal{D}^t$ and $\mathcal{T}^{t-1}$, ***(c)*** the proposed PBA to compute interaction and motion affinities based on position information, ***(d)*** the proposed LTM to predict positions of tracks in F_t, ***(e)*** and an association algorithm with the proposed ZTRM to associate $\mathcal{T}^{t-1}$ and $\mathcal{D}^t$ based on appearance, motion and interaction affinity, outputting tracks $\mathcal{T}^t$ up to frame t.

by retaining long-lost tracks, we design the ZTRM strategy, which performs the association between long-lost tracks and detections separately.

III. Methodology

In this section, we first introduce the overview of LTTrack and then present the details of each proposed module.

A. Overview of LTTrack

Initially, we establish several symbolic expressions. We denote the t-th frame in the video as F_t and represent the set of N detections in F_t as $\mathcal{D}^t = \{\boldsymbol{d}_i^t\}_{i=1}^N$, where each detection is defined by the center point coordinates, width, height, and detection confidence, i.e. $\boldsymbol{d}_i^t = (x, y, w, h, c)$. We denote the set of M tracks up to frame $t-1$ as $\mathcal{T}^{t-1} = \{\boldsymbol{\tau}_j\}_{j=1}^M$, where $\boldsymbol{\tau}_j = \{\boldsymbol{b}_j^k\}_{k=t-L}^{t-1}$ represents a track with ID j and length L, consisting of tracking boxes $\boldsymbol{b}_j^k = (x, y, w, h)$ for target j in L different frames. Besides, the set of positions of tracks at frame $t-1$ is denoted as $\mathcal{Z}^{t-1} = \{\boldsymbol{b}_j^{t-1}\}_{j=1}^M$, and the predicted positions of tracks at frame t is represented as $\hat{\mathcal{Z}}^t = \{\hat{\boldsymbol{b}}_j^t\}_{j=1}^M$.

At the beginning of tracking, we directly initialize the detections $\mathcal{D}^1$ of the first frame as tracks of length 1. For the subsequent frames, as shown in Fig. 2, the workflow of LTTrack contains five stages: (a) Given F_t as input, a detector is introduced to detect objects in F_t and output detections $\mathcal{D}^t$. (b) A ReID model is employed to extract the appearance features for the computation of the appearance cost matrix $\mathbf{C}_{cos}$, which represents the appearance affinity between detections and tracks. (c) The proposed PBA module utilizes $\mathcal{D}^t$ and positions $\mathcal{Z}^{t-1}$ of tracks at frame $t-1$ to compute the position cost matrix $\mathbf{C}_{PBA}$, which reflects motion and interaction affinities. (d) The proposed LTM module is applied to predict the positions $\hat{\mathcal{Z}}^t$ of tracks at frame t. And the predictions $\hat{\mathcal{Z}}^t_{zombie}$ for zombie tracks is used to calculate the IoU cost matrix $\mathbf{C}_{IoU}$ with $\mathcal{D}^t$. (e) Three rounds of association are conducted with different cost matrices as input to associate tracks with detections and update $\mathcal{T}^{t-1}$ to $\mathcal{T}^t$.

Throughout the tracking process, we categorize tracks into four states: alive, lost, zombie, and terminated. The trajectory set is partitioned into three subsets: $\mathcal{T}_{alive}$, $\mathcal{T}_{lost}$, $\mathcal{T}_{zombie}$ to store tracks with the corresponding state, while terminated tracks will be removed from $\mathcal{T}$. The newly appearing targets in each frame will be initialized as alive tracks and added to $\mathcal{T}_{alive}$. If a track is successfully associated with a detection at frame t, its state remains alive and the matched detection is taken as the tracking box of the track at frame t. When a track fails to associate with any detections at frame t, its state becomes lost. When a track remains in the lost state for more than m_{lost} frames, the lost track turns into a zombie track. For tracks in the lost or zombie state from frame t_{lost} to t_L, we utilize the prediction boxes generated by LTM to represent tracking boxes of the track during the lost period, denoted as $\boldsymbol{\tau}_j = \{\boldsymbol{b}_j^{t_1}, \boldsymbol{b}_j^{t_2}, \ldots, \boldsymbol{b}_j^{t_{lost}-1}, \hat{\boldsymbol{b}}_j^{t_{lost}}, \ldots, \hat{\boldsymbol{b}}_j^{t_L}\}$. Once a lost or zombie track is successfully associated with a detection (i.e. re-identified), its state is restored to alive. When a track fails to associate with any detection for more than m_{zombie} frames, we consider the track terminated and remove it from $\mathcal{T}_{zombie}$.

B. Long-Term Motion

Existing motion models suffer from severe error accumulation in long-term motion prediction. We analyze the reason for this as follows. Existing models may introduce errors

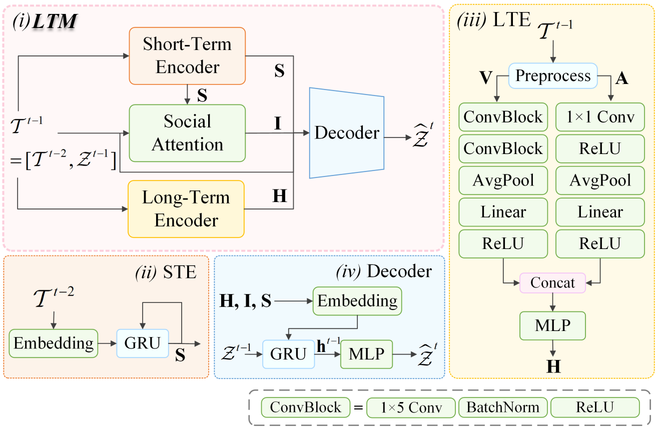


Fig. 3. **Illustration of LTM.** LTM takes tracks $\mathcal{T}^{t-1}$ up to frame $t-1$ as input, and outputs position predictions $\hat{\mathcal{Z}}^t$ of tracks at frame t. There are four submodules in LTM: the Short-Term Encoder (STE) to extract short-term motion features, the Social Attention to extract interaction features among tracks, the Long-Term Encoder (LTE) to extract long-term motion features, and the Decoder to fuse the three features and output predictions $\hat{\mathcal{Z}}^t$.

in their predictions. This is often manageable in short-term motion prediction because the matched detections can correct these errors. However, for long-term motion prediction, without detections to provide corrective information, errors tend to accumulate because the subsequent frame predictions are based on previous predictions with errors. After several frames of prediction, the predicted positions may deviate significantly from the actual locations. Consequently, the tracks may fail to associate with detections accurately.

To mitigate the challenge of error accumulation in long-term motion prediction, we propose that the motion model should not rely solely on the motion states from adjacent frames for motion prediction. Instead, it should consider the overall motion trend of the track. Based on the above analysis, we propose the Long-Term Motion (LTM) model. Building on the common motion model structure that includes trajectory encoding and interaction feature extraction, our LTM integrates an extra branch for long-term motion feature extraction. This helps to minimize error accumulation in motion prediction for consecutive frames. We argue that changes in the height and width of the tracking box can indicate the target motion magnitude. For example, the tracking box of a dancing person experiences more significant changes compared to the tracking box of a static person. Furthermore, the historical velocity of a track can reveal the tendency of the target to move at a particular speed, which can facilitate characterizing the long-term motion trend of the target. For instance, older individuals typically move more slowly than younger ones. Thus, the long-term motion trend of a target is depicted by encoding the historical widths, heights, and velocities of the track.

The input of LTM is the set of tracks $\mathcal{T}^{t-1}$ up to frame $t-1$, and the output is a set of prediction boxes $\hat{\mathcal{Z}}^t$ representing the possible positions of tracks at frame t. As shown in Fig. 3 (i), there are four submodules in LTM: the short-term motion encoder, the long-term motion encoder, the Social Attention [54] module, and the prediction decoder. The workflow of LTM is as follows: (a) Preprocessing $\mathcal{T}^{t-1}$. We preserve the tracking boxes in the last s frames for tracks in $\mathcal{T}^{t-1}$. For tracks shorter than s frames, pad the track sequences with zero vectors at the beginning to make the length of these tracks equal to s. (b) Input $\mathcal{T}^{t-2}$ ($\mathcal{T}^{t-1}$ without tracking boxes at frame $t-1$) to the short-term encoder to extract the short-term motion features $\mathbf{S} \in \mathbb{R}^{M \times d_{LTM}}$, where M is the number of tracks in $\mathcal{T}^{t-1}$ and d_{LTM} is the embedding dimension. (Fig. 3 (ii)). (c) Extract long-term motion features $\mathbf{H} \in \mathbb{R}^{M \times d_{LTM}}$ using the long-term encoder (Fig. 3 (iii)). (d) Input $\mathbf{S}$ and $\mathcal{T}^{t-1}$ to the Social Attention module to extract interaction features $\mathbf{I} \in \mathbb{R}^{M \times d_{LTM}}$. (e) Integrate $\mathbf{H}$, $\mathbf{I}$ and $\mathbf{S}$ to form a new hidden state $\mathbf{h}^{(t-2)'}$, take tracking boxes $\mathcal{Z}^{t-1}$ of tracks at frame $t-1$ as observations and apply a single layer of GRU following a Multi-Layer Perceptron (MLP) to decode the prediction boxes $\hat{\mathcal{Z}}^t \in \mathbb{R}^{M \times 4}$ of tracks at frame t (Fig. 3 (iv)). The above procedure is formulated as follows:

$$\begin{aligned} \mathbf{S} &= \mathrm{STE}\left(\mathcal{T}^{t-2}\right) \\ \mathbf{H} &= \mathrm{LTE}\left(\mathcal{T}^{t-1}\right) \\ \mathbf{I} &= \mathrm{SA}\left(\mathcal{T}^{t-1}, \mathbf{S}\right) \\ \hat{\mathcal{Z}}^t &= \mathrm{Decoder}\left(\mathcal{Z}^{t-1}, \mathbf{H}, \mathbf{I}, \mathbf{S}\right) \end{aligned} \tag{1}$$

where STE, LTE, and SA represent the short-term encoder, long-term encoder, and Social Attention, respectively. Notably, since $\mathcal{Z}^{t-1}$ is taken as observations to input to GRU in the Decoder, we input $\mathcal{T}^{t-2}$ instead of $\mathcal{T}^{t-1}$ to STE to prevent redundant encoding of $\mathcal{Z}^{t-1}$.

Similar to existing data-driven motion models, we encode track sequences and introduce interaction features for motion prediction. Specifically, we apply a single layer of GRU to encode track sequences, and the output hidden states $\mathbf{h}^{t-1}$ at frame $t-1$ are taken as the short-term motion features. What sets LTM apart from existing methods is the additional design of a long-term motion feature encoder, aimed at dealing with error accumulation in long-term motion prediction. The structure of the long-term motion feature encoder is shown in Fig. 3 (iii). For track preprocessing, we denote the target historical velocities as the positional offsets of the top-left and bottom-right corner points of the tracking boxes between adjacent frames. Additionally, we assume the speed of a target is 0 at the starting frame. To encode the historical widths and heights of a track, we divide the width and height of the tracking box by the y-coordinate of its center point, which is taken as the relative width and height of the tracking box. The reason for calculating the relative width and height is that the variation of the width and height of the tracking box not only reflects the target motion magnitude but is also affected by the distance between the target and the camera. Specifically, when a target moves toward the camera, its tracking box usually becomes larger, and the y-coordinate of the center point also increases. We mitigate this effect by calculating the relative width and height. The computation of velocity and relative width and height of a target at frame t is as follows:

$$\mathbf{v}^t = \left(x_1^t - x_1^{t-1}, y_1^t - y_1^{t-1}, x_2^t - x_2^{t-1}, y_2^t - y_2^{t-1}\right) \tag{2}$$

$$\mathbf{a}^t = \left(\frac{w^t}{y_c^t}, \frac{h^t}{y_c^t}\right) \tag{3}$$

where (x_1, y_1) and (x_2, y_2) represent the coordinates of the top-left and bottom-right corner points of the tracking box. y_c^t, w^t, h^t denote the y-coordinate of the center point, width, and height of the tracking box. The historical relative widths and heights of tracks in $\mathcal{T}^{t-1}$ is denoted as $\mathbf{A} \in \mathbb{R}^{M\times 30\times 2}$, while the historical velocities is denoted as $\mathbf{V} \in \mathbb{R}^{M\times 30\times 4}$.

We take a 1×1 convolutional layer to merge the two channels representing relative heights and relative widths in $\mathbf{A}$. Subsequently, the result is passed through a ReLU activation layer, an average pooling layer, a linear layer, and another ReLU activation layer to encode the historical relative widths and heights of tracks. To encode historical velocities $\mathbf{V}$ of tracks, we employ two ConvBlocks, an average pooling layer, a linear layer, and a ReLU activation layer to construct the encoder. Each ConvBlock comprises a 1×5 convolutional layer, a BatchNorm layer, and a ReLU activation layer. Finally, we concatenate the encoded historical widths and heights with the encoded historical velocities and use an MLP for feature fusion to obtain the long-term motion features $\mathbf{H}$. These features are then input to the decoder for motion prediction.

C. Position-Based Association

In dense scenes, closely positioned targets tend to occlude each other, and the appearance features of occluded targets are susceptible to contamination. Consequently, distinguishing these targets based on their position and appearance characteristics becomes a challenging task. However, we observe that in dense scenes, within a group of adjacent targets, each target maintains a unique interaction relationship with its neighbors, which is reflected by the relative positions between the target and its neighbors. As illustrated in Fig. 4, four densely distributed and visually similar targets exhibit distinctive relative position properties that differentiate each target from its counterparts. Moreover, in consecutive frames, the positions of a target will undergo gradual transitions instead of abrupt changes, leading to a high similarity in the relative position properties of the same target between adjacent frames. Hence, we contend that the relative positions of targets can serve as interaction cues for distinguishing between different targets, thus assisting in the association in dense scenes.

In light of the above analysis, we propose the PBA module, which extracts absolute and relative position features to assist data association in crowded scenes. As depicted in Fig. 4, the structure of PBA comprises two branches: the relative position encoder (RelEncoder) and the absolute position encoder (AbsEncoder). PBA takes as input the detections $\mathcal{D}^t$ at frame t and the tracking boxes $\mathcal{Z}^{t-1}$ of tracks at frame $t-1$. The output of PBA is the cost matrix $\mathbf{C}_{PBA}$, representing the motion and interaction affinities between $\mathcal{D}^t$ and $\mathcal{Z}^{t-1}$.

For relative position encoding, we first compute the coordinate differences between bounding boxes in the same set for the input $\mathcal{D}^t$ and $\mathcal{Z}^{t-1}$, as follows:

$$(r_x, r_y, r_w, r_h) = (x_{c1}-x_{c2}, y_{c1}-y_{c2}, w_1-w_2, h_1-h_2) \quad (4)$$

where $(x_{c1}, y_{c1}, w_1, h_1)$ and $(x_{c2}, y_{c2}, w_2, h_2)$ represent the center point coordinates, widths, and heights of two bounding boxes in the same set. (r_x, r_y, r_w, r_h) is taken as the relative coordinates. We retain the k nearest relative coordinates for each bounding box and require that the two bounding boxes used to calculate the relative coordinates have overlapping areas. These selected relative coordinates are then stacked to form relative coordinate matrices for detections and tracks at frame t, denoted as $\mathbf{P}_D \in \mathbb{R}^{N\times k\times 4}$ and $\mathbf{P}_T \in \mathbb{R}^{M\times k\times 4}$, where N and M represents the number of detections and tracks. In cases where the number of bounding boxes that have an overlap with a particular bounding box is less than k, we pad the relative coordinate matrix with zeros to reach the desired size of k. Then, we concatenate $\mathbf{P}_D$ and $\mathcal{D}^t$, $\mathbf{P}_T$ and $\mathcal{Z}^{t-1}$, respectively, and map them to high-dimension features $\mathbf{F}_D \in \mathbb{R}^{N\times 8d_{PBA}}$ and $\mathbf{F}_T \in \mathbb{R}^{M\times 8d_{PBA}}$, where $8d_{PBA}$ is the embedding dimension. The formulas are as follows:

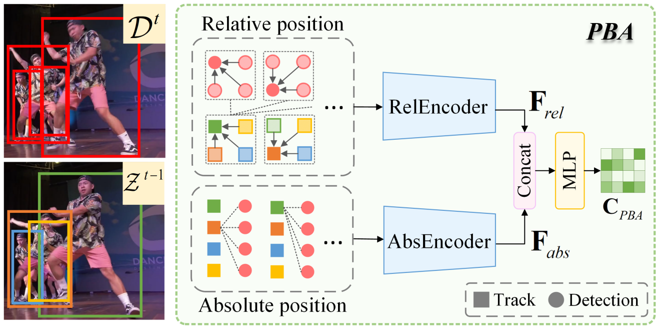


Fig. 4. **Illustration of PBA.** PBA accepts detections $\mathcal{D}^t$ at frame t and positions $\mathcal{Z}^{t-1}$ of tracks at frame $t-1$ as input, encodes the absolute and relative position similarity between them, and produces a position cost matrix $\mathbf{C}_{PBA}$. The absolute position is denoted by bounding box coordinates, while the relative position is indicated by the difference of bounding box coordinates.

$$\begin{aligned}\mathbf{F}_D &= \phi\left(\text{concat}\left(\mathbf{P}_D, \mathcal{D}^t\right), \mathbf{W}_{rel}\right)\\ \mathbf{F}_T &= \phi\left(\text{concat}\left(\mathbf{P}_T, \mathcal{Z}^{t-1}\right), \mathbf{W}_{rel}\right)\end{aligned} \quad (5)$$

where $\mathbf{W}_{rel}$ is the weight of the linear transformation, and ϕ represents ReLU activation function. Subsequently, we construct $\mathbf{F}_D$ and $\mathbf{F}_T$ as relative coordinate matrix $\mathbf{F} \in \mathbb{R}^{M\times N\times 16d_{PBA}}$. Afterward, an MLP is utilized to compute the interaction affinity features $\mathbf{F}_{rel} \in \mathbb{R}^{M\times N\times 2d_{PBA}}$.

For absolute position encoding, as shown in Eq. (6), we calculate the pairwise differences between the center-point coordinates and the aspect ratios of the bounding boxes in $\mathcal{D}^t$ and $\mathcal{Z}^{t-1}$, which serves to represent the motion affinity between detections and tracks. Subsequently, an MLP is employed to obtain the motion affinity features $\mathbf{F}_{abs} \in \mathbb{R}^{M\times N\times d_{PBA}}$.

$$(a_x, a_y, a_w, a_h) = \left(x_{c1} - x_{c2}, y_{c1} - y_{c2}, \log\left(\frac{w_1}{w_2}\right), \log\left(\frac{h_1}{h_2}\right)\right) \quad (6)$$

Finally, we concatenate $\mathbf{F}_{rel}$ and $\mathbf{F}_{abs}$, and feed them into an MLP to compute the final position cost matrix $\mathbf{C}_{PBA} \in \mathbb{R}^{M\times N}$, which is further used to perform association.

$$\mathbf{C}_{PBA} = \text{MLP}\left(\text{concat}\left(\mathbf{F}_{rel}, \mathbf{F}_{abs}\right)\right) \quad (7)$$

D. Zombie Track Re-Match

To re-identify long-lost targets and achieve stable long-term tracking, we propose an improved association algorithm based on the two-stage association algorithm BYTE [11]. Contrary to

Algorithm 1 Pseudo-code of Association Algorithm.

```
Input: $\mathcal{T}^{t-1} = \{\mathcal{T}^{t-1}_{alive}, \mathcal{T}^{t-1}_{lost}, \mathcal{T}^{t-1}_{zombie}\}$, $\mathcal{D}^t = \{\mathcal{D}^t_{high}, \mathcal{D}^t_{low}\}$
Output: $\mathcal{T}^t = \{\mathcal{T}^t_{alive}, \mathcal{T}^t_{lost}, \mathcal{T}^t_{zombie}\}$
  $\hat{\mathcal{Z}}^t \leftarrow \mathrm{LTM}(\mathcal{T}^{t-1})$    // motion prediction by LTM

  // First association (R1)
  $\mathcal{Z}^{t-1} \leftarrow$ positions of tracks in $\mathcal{T}^{t-1}$ at frame $t-1$
  $\mathbf{C}_1 \leftarrow \mathrm{ReID}(\mathcal{T}^{t-1}, \mathcal{D}^t_{high}) + \mathrm{PBA}(\mathcal{Z}^{t-1}, \mathcal{D}^t_{high})$
  Associate $\mathcal{T}^{t-1}$, $\mathcal{D}^t_{high}$ by Hungarians with $\mathbf{C}_1$
  Generate $\mathcal{M}_1$, $\mathcal{T}^{t-1}_{u_1}$, and $\mathcal{D}^t_{u_1}$

  // Second association (R2)
  $\mathcal{D}^t_{u_1} \leftarrow \mathcal{D}^t_{u_1} \cup \mathcal{D}^t_{low}$
  $\mathcal{Z}^{t-1} \leftarrow$ positions of tracks in $\mathcal{T}^{t-1}_{u_1}$ at frame $t-1$
  $\mathbf{C}_2 \leftarrow \mathrm{PBA}(\mathcal{Z}^{t-1}, \mathcal{D}^t_{u_1})$
  Associate $\mathcal{T}^{t-1}_{u_1}$, $\mathcal{D}^t_{u_1}$ by Hungarians with $\mathbf{C}_2$
  Generate $\mathcal{M}_2$, $\mathcal{T}^{t-1}_{u_2}$, and $\mathcal{D}^t_{u_2}$

  // Third association (R3-ZTRM)
  $\mathcal{D}^t_{u_2} \leftarrow \mathcal{D}^t_{u_2} \cap \mathcal{D}^t_{high}$
  $\mathbf{C}_3 \leftarrow \mathrm{ReID}(\mathcal{T}^{t-1}_{zombie}, \mathcal{D}^t_{u_2}) + \mathrm{IoU}(\hat{\mathcal{Z}}^t_{zombie}, \mathcal{D}^t_{u_2})$
  Associate $\mathcal{T}^{t-1}_{zombie}$, $\mathcal{D}^t_{u_2}$ by Hungarians with $\mathbf{C}_3$
  Generate $\mathcal{M}_3$, $\mathcal{T}^{t-1}_{u_3}$, and $\mathcal{D}^t_{u_3}$

  // Track management
  update $\mathcal{T}^t_{alive}$ by $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$
  update $\mathcal{T}^t_{alive}$ by generating new tracks from $\mathcal{D}^t_{u_3}$
  update $\mathcal{T}^t_{lost}, \mathcal{T}^t_{zombie}$ by unmatched tracks $\mathcal{T}^{t-1}_{u_2}$ and $\hat{\mathcal{Z}}^t$
  update $\mathcal{T}^t_{zombie}$ by unmatched zombie tracks $\mathcal{T}^{t-1}_{u_3}$ and $\hat{\mathcal{Z}}^t$
  return $\mathcal{T}^t$
```

BYTE, we incorporate both LTM and PBA into our association algorithm and design a specific association strategy ZTRM for long-term lost targets. In particular, we define tracks with the lost time exceeding m_{lost} frames as "zombie tracks" and utilize ZTRM as an additional round of association to re-identify zombie tracks. The pseudo-code of the association algorithm is shown in Algorithm 1. Taking the association process at frame t as an example, following BYTE, we divide the detections into two sets: high-score detections $\mathcal{D}^t_{high}$ and low-score detections $\mathcal{D}^t_{low}$, based on the detection confidence scores. Subsequently, we perform association as follows:

(R1) The first round of association matches the alive and lost tracks $\{\mathcal{T}^{t-1}_{alive}, \mathcal{T}^{t-1}_{lost}\}$ with high-score detections $\mathcal{D}^t_{high}$. We evaluate the overall similarity by calculating the appearance, motion, and interaction affinities between tracks and detections. To calculate the appearance affinity, we employ a ReID network to obtain the detection appearance feature and calculate a track appearance feature by combining the appearance features of historical detections contained in the track. Afterward, we compute the cosine distance matrix between the two sets of appearance features to represent the appearance affinities, following [55]. To compute motion and interaction affinities, $\mathcal{D}^t_{high}$ and $\mathcal{Z}^{t-1}$ are fed into the PBA, outputting the cost matrix $\mathbf{C}_{PBA}$. Then, the final cost matrix $\mathbf{C}_1$ between $\mathcal{T}^{t-1}$ and $\mathcal{D}^t_{high}$ is computed by adding the cosine distance matrix and $\mathbf{C}_{PBA}$. Finally, $\mathbf{C}_1$ is input to the Hungarian algorithm [47] to obtain the optimal matching result. The outputs of the first association are successfully matched detection-track pairs $\mathcal{M}_1 = \{(\boldsymbol{\tau}_j, \boldsymbol{d}^t_i) \,|\, \boldsymbol{\tau}_j \in \mathcal{T}^{t-1}, \boldsymbol{d}^t_i \in \mathcal{D}^t_{high}\}$, unmatched high-score detections $\mathcal{D}^t_{u_1}$, and unmatched tracks $\mathcal{T}^{t-1}_{u_1}$.

(R2) The low-score detections $\mathcal{D}^t_{low}$ are combined with the unmatched high-score detections $\mathcal{D}^t_{u_1}$ as the detections for the second association. And the unmatched tracks $\mathcal{T}^{t-1}_{u_1}$ from the first association are taken as the tracks for the second association. Because the low-score detections usually have a heavy occlusion or image blur, resulting in unreliable appearance features, we disregard the appearance affinity and solely focus on the motion and interaction affinities for the second association. The PBA is employed to compute the cost matrix $\mathbf{C}_2$ reflecting motion and interaction affinities between tracks and detections. The outputs of the second association are matched detection-track pairs $\mathcal{M}_2 = \{(\boldsymbol{\tau}_j, \boldsymbol{d}^t_i) \,|\, \boldsymbol{\tau}_j \in \mathcal{T}^{t-1}_{u_1}, \boldsymbol{d}^t_i \in \mathcal{D}^t_{low} \cup \mathcal{D}^t_{u_1}\}$, unmatched detections $\mathcal{D}^t_{u_2}$ and unmatched tracks $\mathcal{T}^{t-1}_{u_2}$.

(R3-ZTRM) In the third association, we associate zombie tracks $\mathcal{T}^{t-1}_{zombie}$ with high-score detections in $\mathcal{D}^t_{u_2}$. We merge the appearance and motion affinities to associate zombie tracks with detections. The appearance affinity is calculated in the same way as used in the first association. For the computation of motion affinity, LTM is utilized to predict the motion of the zombie tracks during the loss period. Then, the IoU between the predicted boxes $\hat{\mathcal{Z}}^t_{zombie}$ and detection boxes is computed to represent the motion affinity. The outputs of the third association are matched detection-track pairs $\mathcal{M}_3 = \{(\boldsymbol{\tau}_j, \boldsymbol{d}^t_i) \,|\, \boldsymbol{\tau}_j \in \mathcal{T}^{t-1}_{zombie}, \boldsymbol{d}^t_i \in \mathcal{D}^t_{u_2} \cap \mathcal{D}^t_{high}\}$, unmatched high-score detections $\mathcal{D}^t_{u_3}$ and unmatched zombie tracks $\mathcal{T}^{t-1}_{u_3}$.

After three rounds of association, for all successfully matched detection-track pairs, the matched detection box is used to indicate the position of the track at frame t and set the track state to alive. For unmatched tracks $\mathcal{T}^{t-1}_{u_2}$ in the second association, the track is considered lost. If the track has been lost for more than m_{lost} frames, it turns into a zombie track. For the unmatched zombie tracks $\mathcal{T}^{t-1}_{u_3}$ in the third association, if the track has been lost for more than m_{zombie} frames, the corresponding target is considered terminated, and the track will be deleted. For the lost and zombie tracks, the prediction boxes generated by LTM are used to indicate the positions of these tracks at frame t. Finally, we initialize the unmatched high-score detections $\mathcal{D}^t_{u_3}$ after the third association as new tracks and add them into $\mathcal{T}^t_{alive}$.

In the outlined association algorithm, we include a third association, ZTRM, for zombie tracks to tackle the challenge of re-identifying tracks lost for long periods. Notably, the association of zombie tracks is separate from the association of other tracks with alive or lost states, which will not interfere with the association between other tracks and detections.

E. Training

LTM Training: We create a training sample for LTM by combining ground truth bounding boxes $\mathcal{D}_{GT}$ in a frame from the training set with tracks up to the preceding frame. Taking frame t as an example, the object bounding boxes $\mathcal{D}^t_{GT}$ at frame t are used to supervise training. Meanwhile tracks

$\mathcal{T}^{t-1}$ up to frame $t-1$ are input to the LTM, outputting the predicted positions $\hat{\mathcal{Z}}^t$ of tracks at frame t. The IoU loss [56] is employed to supervise the LTM as follows:

$$\mathcal{L}_{LTM} = 1 - \text{IoU}\left(\hat{\mathcal{Z}}^t, \mathcal{D}_{GT}^t\right)^2 \tag{8}$$

PBA Training: We sample two adjacent frames from videos in the training set as a training sample for PBA. The inputs of PBA are ground truth object bounding boxes from the two adjacent frames, while the output is the cost matrix $\mathbf{C}_{PBA} \in \mathbb{R}^{N_1 \times N_2}$, where N_1 and N_2 are numbers of bounding boxes in the two frames. $\mathbf{C}_{PBA}$ indicates the distances between targets from the two frames. The greater the distance between two targets, the lower the probability that they have the same identity. We construct the ground truth matrix $\mathbf{A} \in \{0,1\}^{N_1 \times N_2}$ based on the target ID. Specifically, if the target i at frame t has the same ID as the target j at frame $t+1$, the elements in column j and row i of $\mathbf{A}$ will be assigned a value of 0, and 1 otherwise. We train PBA using the binary cross-entropy loss as follows:

$$\mathcal{L}_{PBA} = -\mathbf{A}\log\left(\mathbf{C}_{PBA}\right) - \left(1-\mathbf{A}\right)\log\left(1-\mathbf{C}_{PBA}\right) \tag{9}$$

IV. Experiments

In this section, we first discuss our evaluation settings (Section IV-A, IV-B). Then, we conduct ablation studies on LTTrack and analyze the effectiveness of the three proposed modules for MOT (Section IV-C). Afterward, we compare our proposed model with the SOTA methods on three benchmark datasets (Section IV-D). Finally, we present the visualization results (Section IV-E) and analyze the limitations of our tracking method (Section IV-F).

A. Datasets and Metrics

Datasets: We evaluate our LTTrack on three public datasets: MOT17 [16], MOT20 [17], and DanceTrack [29]. MOT17 and MOT20 are pedestrian tracking datasets where most object motion is linear, while MOT20 presents a much greater challenge with crowded scenes. DanceTrack is a challenging dataset, containing dancing scenarios where targets have similar appearances and complex motions. For ablation studies, we split the MOT17 training set following the popular convention [11] that the first half of each video is used for training and the second half for validation. For comparison with SOTA methods, we train LTM and PBA on the MOT17, MOT20, and DanceTrack training sets, respectively, and evaluate our tracker on the corresponding test sets.

Metrics: The identification F1 score (IDF1), higher order tracking accuracy (HOTA), association accuracy (AssA), association recall (AssR), detection accuracy (DetA), multiple object tracking accuracy (MOTA), the number of ID switches (ID Sw), the ratio of mostly tracked targets (MT), and the ratio of mostly lost targets (ML) are employed to evaluate tracking performance [19], [57]–[59]. Specifically, IDF1, AssA, MT, ML, and IDSW are used to evaluate association performance. MOTA is highly related to detection performance. HOTA unifies the performance of detection and association in a balanced manner. Since we focus on the association performance, HOTA and IDF1 are taken as the primary evaluation metrics.

TABLE I
Evaluation of each component in LTTrack on MOT17 validation set. "w/o" means abandoning the corresponding module.

Methods	NUM	HOTA↑	IDF1↑	AssA↑	MOTA↑
w/o PBA	①	66.27	77.41	67.70	75.31
w/o LTM	②	67.02	78.82	69.42	75.25
w/o ZTRM	③	67.16	78.63	69.36	75.41
LTTrack	④	**67.69**	**79.64**	**70.39**	**75.78**

B. Implementation Details

Our LTTrack is evaluated under the "private detection" protocol. For a fair comparison, we adopt the publicly available YOLOX-X detector [60] trained by ByteTrack [11] for MOT17 and MOT20, trained by OC-SORT [13] for DanceTrack. All experiments are conducted with a single RTX 2080 Ti GPU.

Training: For both LTM and PBA training, the AdamW optimizer [61] is utilized with parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. Besides, we use the cosine annealing scheduler [62] with initial learning rates of 10^{-4} for LTM and 10^{-5} for PBA. The training epochs are set to 10 for MOT17, 20 for MOT20, and 30 for DanceTrack.

Hyperparameter settings: Following BYTE [11], we retain high-score and low-score detections by setting the thresholds at 0.6 and 0.1, respectively. SBS50 from FastReID [63] is used to extract ReID features. We set $s = 30$, $d_{LTM} = 32$ for LTM and $k = 4$, $d_{PBA} = 8$ for PBA. The hyperparameters for track state alternation are: $m_{lost} = 20$, $m_{zombie} = 130$ for MOT17, and $m_{lost} = 30$, $m_{zombie} = 100$ for MOT20 and DanceTrack. The effect of the hyperparameters s, k, m_{lost}, and m_{zombie} was investigated in the ablative studies.

C. Ablation Studies

Analysis of each component: We verify the contribution of the proposed modules in our LTTrack by removing them from LTTrack. Specifically, for the model without PBA (①), we use IoU instead of PBA to compute the cost matrix for the first two rounds of association. For the model without LTM (②), the Kalman Filter (KF) improved by OC-SORT [13] is utilized to replace LTM as the baseline motion model. For the model without ZTRM (③), we directly eliminate ZTRM from LTTrack. Besides, for reliable verification, all other settings of these models are the same.

As shown in Table I, our proposed modules can improve all metrics, indicating the effectiveness of the three modules. Specifically, without PBA, the performance in IDF1, HOTA, and AssA is dropped severely (① vs. ④), indicating the effectiveness of PBA in distinguishing identities. Without LTM, the decrease in all metrics also demonstrates the capability of LTM for more accurate motion prediction (② vs. ④). Without ZTRM, we also get inferior performance (③ vs. ④), illustrating that ZTRM can effectively manage long-lost tracks.

Analysis of LTM: To reflect the superiority of our proposed LTM in long-term motion prediction, we compare the performance of LTM with the baseline motion model KF from [13]. In particular, we also evaluate the performance of LTM without the long-term feature extraction branch, namely STM, to further prove the effectiveness of long-term features in motion

TABLE II
EVALUATION OF LTM IN LONG-TERM LOSE SCENES ON THE MOT17 VALIDATION SET.

Settings	NUM	Methods	IDF1↑	HOTA↑	MOTA↑
$30 \leq \ell < 60$	①	KF	61.86	47.77	**58.08**
	②	STM	61.67	47.77	57.96
	③	LTM	**62.15**	**48.17**	57.72
$60 \leq \ell < 90$	①	KF	65.69	47.35	52.65
	②	STM	67.18	47.23	52.71
	③	LTM	**67.92**	**47.54**	**53.05**
$90 \leq \ell$	①	KF	56.48	46.31	54.22
	②	STM	54.92	44.49	**55.32**
	③	LTM	**58.64**	**47.34**	55.29

TABLE III
EVALUATION OF PBA IN OCCLUSION SCENES ON THE MOT17 VALIDATION SET.

Settings	Methods	IDF1↑	HOTA↑	MOTA↑
$a < 0.25$	IoU	84.81	71.20	82.35
	PBA	**85.59**	**71.68**	**82.51**
	Δ	+0.78	+0.48	+0.16
$0.25 \leq a < 0.5$	IoU	79.02	67.36	**80.10**
	PBA	**80.32**	**67.86**	79.71
	Δ	+1.30	+0.50	-0.39
$0.5 \leq a < 0.75$	IoU	71.24	64.79	76.52
	PBA	**73.59**	**67.83**	**76.66**
	Δ	+2.35	+3.04	+0.14
$0.75 \leq a$	IoU	62.48	67.64	78.44
	PBA	**66.03**	**71.74**	**79.31**
	Δ	+3.55	+4.10	+0.87

TABLE IV
COMPARISONS ON ZTRM AND BASELINES WITH DIFFERENT N.

Settings	HOTA↑	IDF1↑	MOTA↑	AssA↑
Baseline(n=10)	66.72	77.95	75.67	68.40
Baseline(n=30)	67.16	78.63	75.41	69.36
Baseline(n=60)	66.74	78.30	75.08	68.77
Baseline(n=90)	66.49	78.27	75.22	68.27
ZTRM	**67.69**	**79.64**	**75.78**	**70.39**

prediction. Given the scarcity of scenes involving long-term target loss in existing MOT datasets, evaluating the long-term motion prediction performance of models directly on the entire dataset would be inadequate. Therefore, we conduct a separate evaluation focusing on tracking long-lost targets. Specifically, we select the targets with a duration of loss exceeding 30 frames from every frame of the MOT17 validation set. These targets are then categorized into three groups based on the loss frame count ℓ for evaluation, as shown in the first column of Table II. Furthermore, we modified the computation of HOTA, IDF1, and MOTA metrics to evaluate the performance of models in tracking targets with different loss durations.

The experimental results are shown in Table II and the best results are marked in bold. Comparing ① and ③, the LTM exhibits noticeable performance improvements over the KF in addressing long-term loss. This evidence demonstrates that our model is capable of precise motion prediction, especially in scenarios involving long-term loss, ensuring accurate long-term tracking. Comparing ② and ③, it is evident that the LTM outperforms the STM, with the performance gap of IDF1 and HOTA increasing as the loss duration lengthens. This demonstrates that our proposed long-term motion feature plays an important role in accurate motion prediction. In addition, it can be observed that LTM does not significantly outperform STM on MOTA. This is because MOTA primarily measures the short-term tracking capability and is biased towards measuring detection [58], while LTM improves on STM by incorporating a long-term feature extraction branch to improve motion prediction and long-term association. As a result, both methods exhibit similar MOTA scores.

Analysis of PBA: In practice, existing methods take motion information (used to compute IoU) along with appearance information to handle the occlusion issue. In contrast, our PBA introduces interaction information in addition to motion information to tackle the challenge of extreme occlusion. To verify the influence of interaction information on data association in occlusion scenes, we split targets in the MOT17 validation set into four classes according to occlusion ratio a, which is defined as the ratio of the maximum occluded area to the bounding box area. Then, we compare the performance of IoU and our proposed PBA under various occlusion ratio conditions. In particular, we gather various classes of targets in the validation set and compute the HOTA, IDF1, and MOTA scores to evaluate the performance of models in tracking targets with varying degrees of occlusion.

As illustrated in Table III, our PBA is superior to IoU in handling various levels of occlusion. Furthermore, in scenarios with severe occlusions ($0.75 \leq a$), PBA-based methods notably outperform IoU-based methods. This observation provides strong evidence that PBA is effective in addressing occlusion problems, and interaction information can assist in challenging data association. Additionally, it can be observed that the introduction of PBA does not improve the MOTA score compared to other metrics. The computation of MOTA is biased to detection evaluation, while our PBA aims to improve association accuracy in crowded scenes, thus changes in the MOTA score are insignificant.

Analysis of ZTRM: To verify the effectiveness of ZTRM in re-identifying long-lost tracks, we remove ZTRM from LTTrack and adjust the max-alive age n of the lost track to create different baselines for comparison. The results in Table IV support our previous analysis that setting n too small or too large can lead to performance degradation. The reason is that setting n too small can increase the likelihood of target re-identification failure while setting n too large can result in an increased number of terminated tracks in the track pool, hindering the association of other tracks. Compared to baselines, our ZTRM exhibits superior performance, highlighting the effectiveness of ZTRM in managing lost tracks.

Analysis of track length s: In the preprocessing stage of LTM, the tracking boxes of input tracks in the last s frames are retained for subsequent feature extraction. To ensure sufficient track information for feature extraction, it is recommended to set s to a relatively high value. However, setting s too high can introduce noisy data and interfere with the extraction of interaction features. To determine the appropriate value for s,

TABLE V
EVALUATION OF DIFFERENT TRACK LENGTH s IN LTM.

s	HOTA↑	IDF1↑	MOTA↑	AssA↑
20	67.19	78.95	75.10	69.52
30	**67.69**	**79.64**	**75.78**	**70.39**
40	67.21	78.84	75.25	69.55
50	67.58	79.41	75.39	70.20
60	67.51	79.32	75.35	70.09

TABLE VI
EVALUATION OF DIFFERENT NUMBER k OF RELATIVE POSITIONS IN PBA.

k	HOTA↑	IDF1↑	MOTA↑	AssA↑
2	66.43	77.72	75.07	68.18
3	66.93	78.06	74.88	69.28
4	**67.69**	**79.64**	**75.78**	**70.39**
5	66.61	77.90	74.63	68.95
6	66.03	77.05	74.95	67.40

we sequentially set s values from 20 to 60 to train multiple LTM models, assessing their performance on the MOT17 validation set. The experimental results in Table V demonstrate that the model works best when $s = 30$.

Analysis of relative position number k: To characterize the interaction information of target b with the surrounding targets, PBA selects the top k target boxes that have the highest degree of overlap with b to calculate the relative positions. When the number of bounding boxes overlapping with the bounding box of b is less than k, zero padding is used to round up the relative coordinate count to k. Setting a small value for k makes the extracted interaction information incomplete, while setting a large value for k introduces redundant and meaningless zero vectors into the PBA encoding, affecting the accuracy of the extracted features. To select the optimal value for k, we sequentially test k from 2 to 6 on the MOT17 validation set. Table VI shows the experimental outcomes and the best model performance is achieved with $k = 4$.

Analysis of thresholds m_{lost}, m_{zombie} for track state alteration: As stated in Section III-D, a lost track will turn into a zombie track after failing to be re-identified for m_{lost} frames, while a zombie track will be deleted after failing to be re-identified for m_{zombie} frames. We apply grid research [64] to find the best combination of m_{lost} and m_{zombie}. As illustrated in Fig. 5, the HOTA metric reaches the best performance when $m_{lost} = 20$ and $m_{zombie} = 130$.

D. Comparison with State-of-the-Art Methods

For comparison with SOTA methods, we evaluate LTTrack on the test sets of DanceTrack, MOT17, and MOT20. The results are used to compare LTTrack with the SOTA methods under the "private detection" protocol. The hyperparameter settings correspond to the results of ablation studies in Section IV-C. For a fair comparison, we also apply linear interpolation after LTTrack, as [11] [13]. **"(w/ LI)"** in the following tables indicates that interpolation is employed in the model.

DanceTrack: As depicted in Table VII, our LTTrack outperforms the SOTA TBD trackers on the DanceTrack benchmark for most metrics. In comparison to ByteTrack [11] and OC-SORT [13], which partly constitute our baseline, LTTrack outperforms them across all metrics (i.e., +3.7% HOTA and +5.6% IDF1 compared to OC-SORT). The results affirm the effectiveness of our method in dealing with challenging scenarios, including complex motion and severe occlusion.

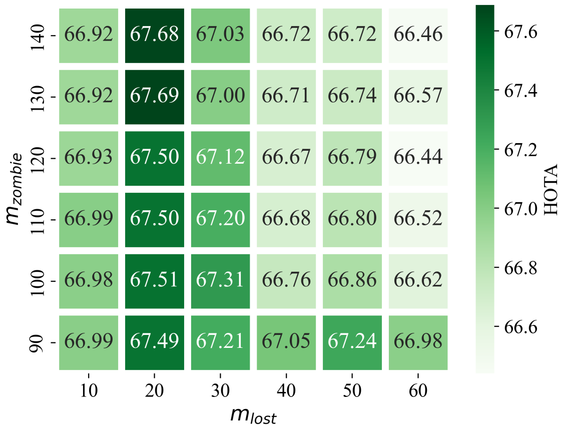


Fig. 5. The grid research results of various combination of m_{lost} and m_{zombie}. The values in the grid indicate the HOTA scores for different combinations of m_{lost} (horizontal axis) and m_{zombie} (vertical axis). The HOTA achieves optimal performance when $m_{lost} = 20$, $m_{zombie} = 130$.

TABLE VII
COMPARISON RESULTS UNDER THE "PRIVATE DETECTOR" PROTOCOL ON THE DANCETRACK TEST SET. THE TOP THREE BEST RESULTS OF TBD METHODS ARE MARKED IN THE ORDER OF RED BLUE AND GREEN.

Methods	Ref.	HOTA↑	IDF1↑	AssA↑	DetA↑
TBQ:					
TransTrack [40]	arxiv 2020	45.5	45.2	27.5	75.9
MeMOTR [9]	ICCV 2023	68.5	71.2	58.4	80.5
MOTRv2 [8]	CVPR 2023	69.9	71.7	59.0	83.0
TBD:					
FairMOT [22]	IJCV 2021	39.7	40.8	23.8	66.7
ByteTrack [11]	ECCV 2022	47.7	53.9	32.1	71.0
STDFormer [27]	TCSVT 2023	**57.8**	**60.5**	**41.7**	80.5
QuasiDense [65]	TPAMI 2023	54.2	50.4	36.8	80.1
StrongSORT++ [18]	TMM 2023	55.6	55.2	38.6	**80.7**
GHOST [50]	CVPR 2023	**56.7**	**57.7**	39.8	**81.1**
OC-SORT [13]	CVPR 2023	55.1	54.9	**40.4**	80.4
LTTrack	ours	**58.8**	**60.5**	**43.0**	**81.1**

MOT17: As displayed in Table VIII, our LTTrack ranks within the top three compared to SOTA methods on the MOT17 test set, achieving 64.3 HOTA, 79.2 IDF1, and 64.8 AssA. Specifically, LTTrack outperforms OC-SORT [13] on HOTA by 1.1%, IDF1 by 1.7%. This superiority can be attributed to our utilization of LTM and ZTRM to improve long-term tracking capabilities and PBA to handle crowded scenes. However, since our method specifically focuses on long-term tracking, which represents a relatively limited scene within the MOT17 dataset, our model did not exhibit significant superiority across the entire MOT17 dataset. Observably, our model displays suboptimal performance in MT, ML, ID Sw, and MOTA. This deficiency stems from the inclusion of low-score detections during association, which aims to reduce false negative detections (FNs) and associate as many occluded

TABLE VIII
Comparison results under the “private detector” protocol on the MOT17 test set. “(w/ LI)” means linear interpolation is used. The top three best results of TBD methods are marked in the order of red blue and green.

Methods	Ref.	HOTA↑	IDF1↑	MOTA↑	AssA↑	AssR↑	MT↑	ML↓	ID Sw↓
TBQ:									
TransCenter [4]	TPAMI 2022	-	65.4	76.4	-	-	51.7%	11.6%	6402
TrackFormer [5]	CVPR 2022	57.3	68.0	74.1	54.1	58.0	47.3%	10.4%	2829
MeMOT [7]	CVPR 2022	56.9	69.0	72.5	55.2	-	43.8%	18.0%	2724
MeMOTR [9]	ICCV 2023	58.8	71.5	72.8	58.4	63.0	41.4%	19.2%	1902
MOTRv2 [8]	CVPR 2023	62.0	75.0	78.6	60.6	-	-	-	-
TBD:									
CTracker [46]	ECCV 2020	49.0	57.4	66.6	45.2	48.1	32.2%	24.2%	5529
FairMOT [22]	IJCV 2021	59.3	72.3	73.7	58.0	63.6	43.2%	17.3%	3303
MOTFR [12]	TCSVT 2022	61.8	76.3	74.4	62.6	67.8	46.1%	17.6%	2652
QuoVadis [20]	NIPS 2022	63.1	77.7	**80.3**	62.1	68.8	**55.5%**	**10.8%**	2103
MAATrack [36]	WACV 2022	62.0	75.9	79.4	60.2	67.3	**57.6%**	**12.0%**	1452
ByteTrack [11]	ECCV 2022	63.1	77.3	**80.3**	62.0	68.2	53.2%	14.5%	2196
CSTrack [41]	TIP 2022	59.3	72.6	74.9	57.9	63.2	41.5%	17.5%	3567
STDFormer [27]	TCSVT 2023	59.9	71.5	78.8	56.6	-	49.7%	13.1%	4998
DcMOT [48]	TCSVT 2023	61.3	75.2	74.5	-	-	42.0%	16.9%	2682
FDTrack [49]	TCSVT 2023	61.3	75.6	76.8	-	-	43.1%	16.9%	3705
QuasiDense [65]	TPAMI 2023	63.5	77.5	78.7	62.6	69.3	54.0%	**12.6%**	1935
StrongSORT [18]	TMM 2023	63.5	78.5	78.3	63.7	63.6	-	-	**1446**
StrongSORT++ [18]	TMM 2023	**64.4**	**79.5**	79.6	64.4	**71.0**	53.6%	13.9%	**1194**
MotionTrack [24]	CVPR 2023	**65.1**	**80.1**	**81.1**	**65.1**	**70.8**	**55.5%**	16.7%	**1140**
GHOST [50]	CVPR 2023	62.8	77.1	78.7	-	-	-	-	2325
OC-SORT [13]	CVPR 2023	63.2	77.5	78.0	63.2	67.5	41.0%	20.9%	1950
LTTrack	ours	63.8	79.1	78.0	**64.6**	69.6	45.50%	13.10%	2220
LTTrack (w/ LI)	ours	**64.3**	**79.2**	79.0	**64.8**	**70.2**	51.5%	12.9%	1902

TABLE IX
Comparison results under the “private detector” protocol on the MOT20 test set. “(w/ LI)” means linear interpolation is used. The top three best results of TBD methods are marked in the order of red blue and green.

Methods	Ref.	HOTA↑	IDF1↑	MOTA↑	AssA↑	AssR↑	MT↑	ML↓	ID Sw↓
TBQ:									
TransTrack [40]	arxiv 2020	48.9	59.4	65	45.2	-	50.1%	13.4%	3608
TransCenter [4]	TPAMI 2022	-	58.1	72.5	-	-	64.7%	12.0%	2332
TrackFormer [5]	CVPR 2022	54.7	65.7	68.6	53.0	57.4	53.6%	15.0%	2474
TBD:									
FairMOT [22]	IJCV 2021	54.6	67.3	61.8	54.7	60.7	68.8%	**7.6%**	5243
MOTFR [12]	TCSVT 2022	57.2	71.7	69.0	57.1	62.6	65.7%	10.3%	3648
QuoVadis [20]	NIPS 2022	61.5	75.7	**77.8**	59.9	67.0	**69.2%**	**9.5%**	1187
MAATrack [36]	WACV 2022	57.3	71.2	73.9	55.1	61.1	59.7%	12.3%	1331
ByteTrack [11]	ECCV 2022	61.3	75.2	**77.8**	59.6	66.2	**69.2%**	**9.5%**	1223
CSTrack [41]	TIP 2022	54.0	68.6	66.6	54.0	57.6	50.4%	15.5%	3196
STDFormer [27]	TCSVT 2023	60.0	72.3	75.8	58.0	-	67.4%	12.0%	2329
DcMOT [48]	TCSVT 2023	53.8	67.4	59.7	-	-	66.7%	**7.6%**	5636
FDTrack [49]	TCSVT 2023	59.9	75.7	75.0	-	-	62.8%	9.7%	2226
QuasiDense [65]	TPAMI 2023	60.0	73.8	74.7	58.9	65.7	64.2%	13.0%	**1042**
StrongSORT [18]	TMM 2023	61.5	75.9	72.2	63.2	59.9	-	-	1066
StrongSORT++ [18]	TMM 2023	62.6	**77.0**	73.8	**64.0**	**69.6**	62.1%	14.9%	**770**
MotionTrack [24]	CVPR 2023	**62.8**	76.5	**78.0**	61.8	68.0	**71.3%**	**9.5%**	
GHOST [50]	CVPR 2023	61.2	75.2	73.7	-	-	-	-	1264
OC-SORT [13]	CVPR 2023	62.1	75.9	75.5	62.0	-	-	-	**913**
LTTrack	ours	**62.7**	**77.3**	75.2	**63.3**	**69.6**	66.7%	11.2%	1609
LTTrack (w/ LI)	ours	**63.2**	**77.3**	76.0	**63.5**	**70.3**	**70.9%**	10.5%	1183

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2024.3404275

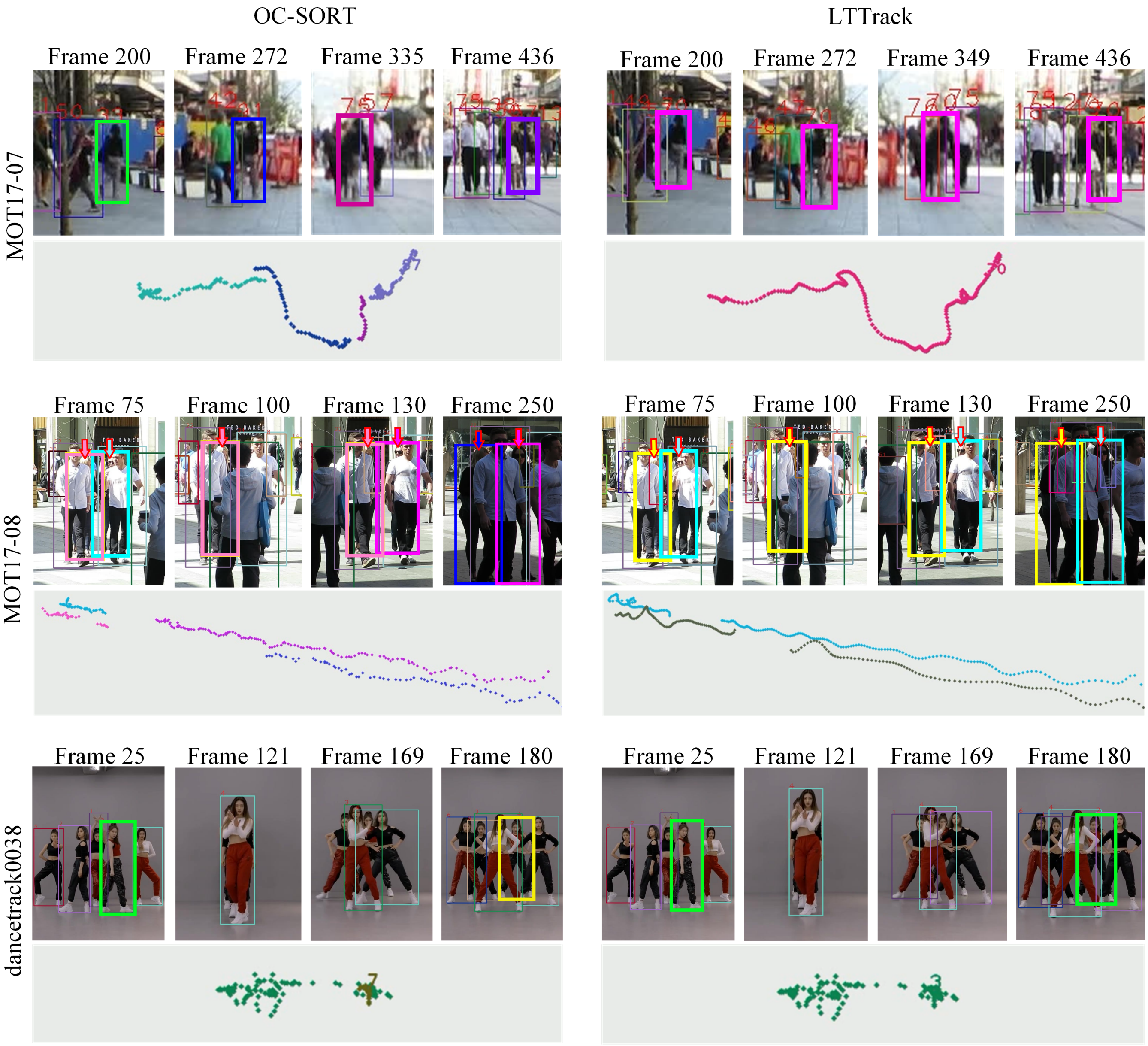


Fig. 6. **Visualization results of LTTrack and OC-SORT [13].** Different colors indicate different identities. Targets of interest are highlighted with bolder bounding boxes. The dotted lines below represent the tracks of targets of interest.

targets as possible. Nevertheless, this strategy inevitably introduces false positive detections (FPs), consequently generating spurious tracks and leading to degradation in performance across MT, ML, and ID Sw. Furthermore, while retaining low-score detections diminishes FNs, it disturbs the delicate balance between FPs and FNs, resulting in more FPs. Consequently, the MOTA score of LTTrack is suboptimal. Nevertheless, the competitive scores of LTTrack in metrics measuring association performance (IDF1, AssA) demonstrate the effectiveness of our method.

MOT20: On the MOT20 test set (Table IX), LTTrack surpasses the previous SOTA methods in most metrics. Compared with OC-SORT [13], part of which is included in our baseline, LTTrack improved by 1.1%, 1.4%, 0.5%, and 1.5% on HOTA, IDF1, MOTA, and AssA, respectively. The results demonstrate the robustness and effectiveness of our LTTrack in handling crowded scenarios with occlusion challenges. Similar to the test results on MOT17, LTTrack does not achieve optimal performance on the four metrics: MOTA, MT, ML, and ID Sw. The reason is that no superior detectors were used, and low-score detections were kept during association.

Further Analysis: Comparing the test results of TBQ and TBD methods on the three benchmark datasets, it is evident that TBQ algorithms outperform TBD algorithms on the DanceTrack dataset while on the MOT17 and MOT20 datasets, the results are reversed. We attribute this phenomenon to the fact that the DanceTrack dataset is large and the scenes are mostly camera stationary, so the TBQ algorithm converges well during training. On the contrary, the MOT17 and MOT20 datasets are prone to overfitting due to their small size, resulting in query duplication in crowded scenes [8]. In addition, the MOT17 and MOT20 datasets contain camera motion scenes. However, current TBQ algorithms lack effective solutions for camera motion scenes. Furthermore, it can be observed that simple TBD algorithms [11], [13], [18] exhibit noteworthy performance degradation on the DanceTrack dataset. The reason is that the complexity of the target motion and similar appearances on DanceTrack pose a significant challenge to simple TBD algorithms. In contrast, the TBQ method utilizes deep learning models to effectively model complex target motion, resulting in superior performance on DanceTrack. Our proposed method, which belongs to the TBD algorithm, achieves comparable results on all three datasets to SOTA methods. To analyze the reasons for this, on the one hand, LTTrack introduces LTM to accurately model complex motions and PBA to distinguish targets with similar appearances, leading to better performance than other TBD trackers in the DanceTrack dataset. On the other hand, LTTrack is designed for long-term tracking with LTM, PBA, and ZTRM, resulting in more robust tracking results in MOT17 and MOT20 datasets.

E. Qualitative Results

To more intuitively reflect the ability of LTTrack to address occlusion and achieve stable long-term tracking, we compare the visualization results of LTTrack and OC-SORT [13] in the test sets of MOT17 and DanceTrack in Fig. 6. In *MOT17-07*, the highlighted target is frequently occluded in dense crowds, leading to track fragmentation in the tracking result of OC-SORT. However, with the integration of PBA, LTTrack mitigates track fragmentation effectively. In *MOT17-08*, prolonged occlusions result in significant track interruptions of the highlighted targets. OC-SORT incorrectly deletes the long-lost target and initializes the reappeared targets as new tracks. Conversely, LTTrack leverages LTM and ZTRM to model long-term motion and successfully restores zombie tracks after long-term loss. In *dancetrack0038*, complex target motions and prolonged occlusions occur from frame 121 to frame 169. OC-SORT initializes the reappearing target after a long-term loss as a new track erroneously, whereas LTM effectively recovers zombie tracks with the help of ZTRM.

F. Limitations

Although our method excels in long-term tracking scenes, we also recognize certain limitations within this paradigm. As shown in Fig. 7 (a), false positive detections lead to the generation of spurious tracks. This kind of failure can be attributed to the retention of low-score detections in LTTrack, which was originally meant to reduce false negatives and ensure the re-identification of lost targets. Therefore, balancing false negative and false positive detections remains a problem requiring a solution. As shown in Fig. 7 (b), ZTRM incorrectly associates a newly appeared target with a zombie track that has left the scene. In cases where a target re-enters the scene after leaving, LTTrack preserves tracks that have exited the scene for m_{zombie} frames to recover zombie tracks. This practice results in interference with the initialization of tracks for newly entered targets. Hence, future works can focus on designing a more adaptable track management strategy that accommodates scenarios involving tracks entering and exiting the scene.

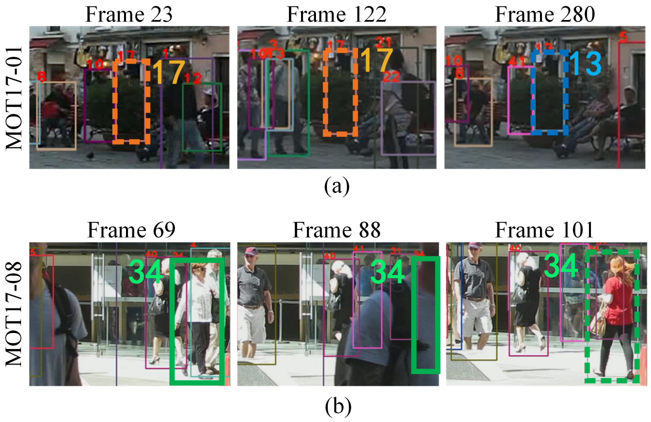


Fig. 7. **Illustration of failure cases.** (a) A failure case caused by false positive detections. (b) A failure case caused by ZTRM. The dashed boxes indicate the failure tracking results.

V. Conclusion

Our work presents an effective and simple solution called LTTrack to address the commonly overlooked but practical problem of long-term tracking. To handle occlusion and ensure consistent long-term tracking, we devise the PBA module, which innovatively utilizes interaction cues for association. In addition, we propose a motion model LTM to achieve precise multi-frame motion prediction in cases of trajectory loss. This is accomplished by extracting the long-term motion features of the track, which in turn aid in re-identifying the long-lost target. For the long-lost track management, which is overlooked by existing methods, we define long-lost tracks as zombie tracks, and develop a unique track management policy, ZTRM, for zombie tracks, so that long-lost tracks will not be deleted by mistake. Combining the above three modules, a MOT framework LTTrack for long-term tracking is proposed. Extensive experiments are conducted on the MOT17, MOT20, and DanceTrack benchmarks to verify the effectiveness of LTTrack. The results show that our LTTrack achieves comparable performance to SOTA methods and presents superior performance in long-term tracking scenarios.

Appendix A
Results on VisDroneMOT

In this section, we provide more experimental results on the VisDrone2019 MOT benchmark [67] to further evaluate the effectiveness of our LTTrack.

Dataset and Metrics: The VisDrone2019 [67] is a large-scale drone video dataset that includes five tracking categories: pedestrian, car, bus, truck, and van. Captured from the perspective of drones, the targets in VisDrone2019 typically exhibit small scales, blurred appearances, long motion tracks, and intense camera motion, posing great challenges to existing MOT methods. For comparison with SOTA methods, we

TABLE X
COMPARISON RESULTS ON THE VISDRONE2019 TEST-DEV SET. THE TOP THREE BEST RESULTS ARE MARKED IN THE ORDER OF RED BLUE AND GREEN.

Methods	Ref.	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	ID Sw↓
MOTDT [21]	ICME 2018	21.6	-0.8	68.5	87	1196	44548	185453	**1437**
FairMOT [22]	IJCV 2021	48.3	**36.4**	**75.7**	574	**525**	31346	**110498**	4052
MOTR [6]	ECCV 2022	41.4	22.8	72.8	272	825	28407	147937	**959**
TrackFormer [11]	CVPR 2022	30.5	25.0	73.9	385	770	**25856**	141526	4840
UAVMOT [66]	CVPR 2022	**51.0**	36.1	74.2	520	574	**27983**	115925	2775
PID-MOT [51]	TCSVT 2023	50.2	33	74.1	**686**	**424**	53691	**96541**	3529
STDFormer [27]	TCSVT 2023	**57.1**	**45.9**	**77.9**	**684**	**538**	**21288**	**101506**	1440
LTTrack	ours	**57.5**	**43.0**	**75.1**	**746**	**729**	35294	138824	**1429**

train LTM and PBA on the training set together with the validation set of VisDrone2019 and evaluate our LTTrack on the VisDrone2019 test-dev set.

Following [67], the IDF1, MOTA, multiple object tracking precision (MOTP), MT, ML, FP, FN, and ID Sw are employed to evaluate tracking performance. In particular, IDF1 and MOTA are taken as the primary evaluation metrics.

Implementation Details: We retrain YOLOX-X as a multi-class detector for VisDrone2019. For the training of LTM and PBA, the number of training epochs is set to 30. Other settings are the same as Section IV-B. For the hyperparameter settings, we set $m_{lost} = 30$ and $m_{zombie} = 130$.

Comparison with State-of-the-Art Methods: As illustrated in Table X, our LTTrack achieves 57.5 on IDF1 and 43.0 on MOTA, which is competitive with the SOTA methods on the VisDrone2019 test-dev set. The results confirm the effectiveness of our method in tackling challenging scenes, including complex motion and blurred appearance. Nevertheless, compared to the performance of LTTrack on the pedestrian tracking benchmarks [16], [17], [29], its performance on VisDrone2019 is notably inferior. We contend that there are two principal factors contributing to this outcome. First, our retrained detector YOLOX is not suitable for detecting small targets from the perspective of a drone, as indicated by the lower MOTA and MOTP scores. Secondly, the scenarios included in VisDrone2019 are all multi-class multi-object tracking from a drone's perspective with intense camera motion. However, our method is initially designed for pedestrian tracking in natural scenes. Specifically, our LTM solely extracts motion features based on trajectory information, without fully considering camera motion, which can be unreliable in scenes captured by drones with intense camera motion.

APPENDIX B
ANALYSIS OF INFERENCE SPEED

In this section, we analyze the impact of the three proposed modules, namely LTM, PBA, and ZTRM, on the inference speed of LTTrack. Moreover, a comparison between LTTrack and two efficient trackers [11], [13] is also performed to reveal the speed issues of different MOT trackers. Notably, it is difficult to fairly compare the speed of different trackers because the runtime of each tracker depends on the hardware on which they are executed. To ensure fairness, we record the runtime and frames per second (FPS) of trackers on a single 2080Ti GPU with a batch size of 1 and YOLOX as the detector. Additionally, both the association time and the total time for each tracker are reported. The total time encompasses both the detection time and the association time.

TABLE XI
EVALUATION OF INFERENCE SPEED ON THE MOT17 VALIDATION SET.

Methods	Association		Total	
	FPS↑	Time(s)↓	FPS↑	Time(s)↓
Baseline	40.92	64.83	16.70	158.80
Baseline+LTM	29.64	89.49	15.01	176.71
Baseline+LTM+PBA	27.77	95.53	14.46	183.45
Baseline+LTM+PBA+ZTRM	27.34	97.00	14.36	184.69
ByteTrack [11]	66.09	40.13	17.34	152.97
OC-SORT [13]	51.63	51.40	17.24	153.79

As shown in Table XI, in terms of association speed, adding LTM significantly slows down the association process (-11.28 FPS), while the PBA and ZTRM cause negligible computational costs (-1.87 FPS and -0.43 FPS). Compared to the two efficient trackers [11], [13], our LTTrack is considerably slower. However, in terms of the total speed, all three modules have caused a slight reduction in the tracking speed. Moreover, LTTrack exhibits a minor speed difference compared to ByteTrack and OC-SORT. The reason is that the detection step consumes significantly more time than the association step in the tracking process, and variations in the association time have minimal impact on the tracking speed. Hence, the detection module is the key factor affecting the speed of the TBD tracker, rather than association.

REFERENCES

[1] H.-k. Chiu, J. Li, R. Ambruş, and J. Bohg, "Probabilistic 3d multi-modal, multi-object tracking for autonomous driving," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2021, pp. 14 227–14 233.

[2] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3153–3160.

[3] K. Zhang, Y. Li, J. Wang, E. Cambria, and X. Li, "Real-time video emotion recognition based on reinforcement learning and domain knowledge," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1034–1047, 2021.

[4] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "Transcenter: Transformers with dense representations for multiple-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7820–7835, 2023.

[5] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8844–8854.

[6] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "Motr: End-to-end multiple-object tracking with transformer," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 659–675.

[7] J. Cai, M. Xu, W. Li, Y. Xiong, W. Xia, Z. Tu, and S. Soatto, "Memot: Multi-object tracking with memory," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8090–8100.

[8] Y. Zhang, T. Wang, and X. Zhang, "Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22 056–22 065.

[9] R. Gao and L. Wang, "Memotr: Long-term memory-augmented transformer for multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 9901–9910.

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

[11] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 1–21.

[12] J. Kong, E. Mo, M. Jiang, and T. Liu, "Motfr: Multiple object tracking based on feature recoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7746–7757, 2022.

[13] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 9686–9696.

[14] S. You, H. Yao, B.-K. Bao, and C. Xu, "Utm: A unified multiple object tracking model with identity-aware feature enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 21 876–21 886.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[16] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[17] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. D. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," *arXiv preprint arXiv:2003.09003*, 2020.

[18] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Trans. Multimedia*, vol. 25, pp. 8725–8737, 2023.

[19] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 17–35.

[20] P. Dendorfer, V. Yugay, A. Osep, and L. Leal-Taixé, "Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 15 657–15 671.

[21] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2018, pp. 1–6.

[22] 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, no. 11, pp. 3069–3087, 2021.

[23] 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.*, 2022, pp. 8834–8843.

[24] Z. Qin, S. Zhou, L. Wang, J. Duan, G. Hua, and W. Tang, "Motiontrack: Learning robust short-term and long-term motions for multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17 939–17 948.

[25] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann, and S. Gould, "Probabilistic tracklet scoring and inpainting for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14 329–14 339.

[26] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "Transmot: Spatial-temporal graph transformer for multiple object tracking," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 4859–4869.

[27] M. Hu, X. Zhu, H. Wang, S. Cao, C. Liu, and Q. Song, "Stdformer: Spatial-temporal motion transformer for multiple object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 11, pp. 6571–6594, 2023.

[28] Y. Wu, H. Sheng, S. Wang, Y. Liu, Z. Xiong, and W. Ke, "Group guided data association for multiple object tracking," in *Proc. Asia. Conf. Comput. Vis.*, 2022, pp. 485–500.

[29] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, and P. Luo, "Dancetrack: Multi-object tracking in uniform appearance and diverse motion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 20 961–20 970.

[30] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Engineer.*, vol. 82, pp. 35–45, 1960.

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

[32] 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. Workshops*, 2016, pp. 36–42.

[33] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 107–122.

[34] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, and X. Pan, "MAT: motion-aware multi-object tracking," *Neurocomputing*, vol. 476, pp. 75–86, 2022.

[35] N. Aharon, R. Orfaig, and B. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *arXiv preprint arXiv:2206.14651*, 2022.

[36] D. Stadler and J. Beyerer, "Modelling ambiguous assignments for multi-person tracking in crowds," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops*, 2022, pp. 133–142.

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

[38] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 474–490.

[39] Q. Liu, Q. Chu, B. Liu, and N. Yu, "GSM: graph similarity model for multi-object tracking," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 530–536.

[40] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple-object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[41] 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.

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

[43] N. Ran, L. Kong, Y. Wang, and Q. Liu, "A robust multi-athlete tracking algorithm by exploiting discriminant features and long-term dependencies," in *Proc. Int. Conf. Multimedia Model.*, 2019, pp. 411–423.

[44] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 300–311.

[45] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proc. IEEE Int. Conf. Adv. Video Signal-Based Surveill.*, 2017, pp. 1–6.

[46] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 145–161.

[47] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[48] K. Deng, C. Zhang, Z. Chen, W. Hu, B. Li, and F. Lu, "Jointing recurrent across-channel and spatial attention for multi-object tracking with block-erasing data augmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 4054–4069, 2023.

[49] Y. Jin, F. Gao, J. Yu, J. Wang, and F. Shuang, "Multi-object tracking: Decoupling features to solve the contradictory dilemma of feature requirements," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 5117–5132, 2023.

[50] J. Seidenschwarz, G. Brasó, V. C. Serrano, I. Elezi, and L. Leal-Taixé, "Simple cues lead to a strong multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13 813–13 823.

[51] W. Lv, N. Zhang, J. Zhang, and D. Zeng, "One-shot multiple object tracking with robust id preservation," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2023.

[52] Z. Sun, J. Chen, C. Liang, W. Ruan, and M. Mukherjee, "A survey of multiple pedestrian tracking based on tracking-by-detection framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 5, pp. 1819–1833, 2021.

[53] Z. Zou, J. Huang, and P. Luo, "Compensation tracker: Reprocessing lost object for multi-object tracking," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 307–317.

[54] J. Amirian, J. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with gans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 2964–2972.

[55] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification," in *Proc. IEEE Int. Conf. Image Process.*, 2023, pp. 3025–3029.

[56] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proc. ACM Int. Conf. Multimedia.*, 2016, p. 516–520.

[57] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.

[58] J. Luiten, A. Osep, P. Dendorfer, P. H. S. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 548–578, 2021.

[59] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2953–2960.

[60] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: exceeding YOLO series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[61] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[62] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent.*, 2017.

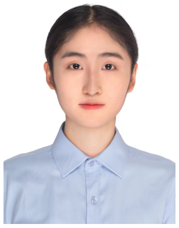
[63] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, "Fastreid: A pytorch toolbox for general instance re-identification," in *Proc. ACM Int. Conf. Multimedia.*, 2023, pp. 9664–9667.

[64] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[65] T. Fischer, T. E. Huang, J. Pang, L. Qiu, H. Chen, T. Darrell, and F. Yu, "Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 15 380–15 393, 2023.

[66] S. Liu, X. Li, H. Lu, and Y. He, "Multi-object tracking meets moving uav," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8876–8885.

[67] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7380–7399, 2022.



Jiaping Lin received a B.S. degree from Sichuan University, Chengdu, China, in 2022. She is currently pursuing a Master's degree from Sichuan University, Chengdu, China. Her research interests include multi-object tracking and object detection.

Gang Liang received a Ph.D. degree in computer science from Sichuan University, Chengdu, China, in 2007. He is currently an Associate Professor with the School of Cyber Science and Engineering, Sichuan University, Chengdu, China. His research interests include computer vision, multimedia forensics, social networks, and machine learning.

Rongchuan Zhang received a B.S. degree from Sichuan University, Chengdu, China, in 2022. She is currently pursuing a Master's degree from Sichuan University, Chengdu, China. Her current research interests include multimedia forensics, computer vision, and deep learning.