STDFormer: Spatial-Temporal Motion Transformer for Multiple Object Tracking

Mengjie Hu, Xiaotong Zhu, Haotian Wang, Shixiang Cao, Chun Liu, and Qing Song

***Abstract*— Mainstream multi-object tracking methods exploit appearance information and/or motion information to achieve interframe association. However, dealing with similar appearance and occlusion is a challenge for appearance information, while motion information is limited by linear assumptions and is prone to failure in nonlinear motion patterns. In this work, we disregard appearance clues and propose a pure motion tracker to address the above issues. It dexterously utilizes Transformer to estimate complex motion and achieves high-performance tracking with low computing resources. Furthermore, contrastive learning is introduced to optimize feature representation for robust association. Specifically, we first exploit the long-range modeling capability of Transformer to mine intention information in temporal motion and decision information in spatial interaction and introduce prior detection to constrain the range of motion estimation. Then, we introduce contrastive learning as an auxiliary task to extract reliable motion features to compute affinity and introduce bidirectional matching to improve the affinity computation distribution. In addition, given that both tasks are dedicated to narrowing the embedding distance between the motion features of the tracked object and the detection features, we design a joint-motion-and-association framework to unify the above two tasks in one framework for optimization. The experimental results achieved with three benchmark datasets, MOT17, MOT20 and DanceTrack, verify the effectiveness of our proposed method. Compared with state-of-the-art methods, the proposed STDFormer sets a new state-of-the-art on DanceTrack and achieves competitive performance on MOT17 and MOT20. This demonstrates the advantage of our method in handling associations under similar appearance, occlusion or nonlinear motion. At the same time, the significant advantages of the proposed method over Transformer-based and contrastive learning-based methods suggest a new direction for the application of Transformer and contrastive learning in MOT. In addition, to verify the generalization of STDFormer in unmanned aerial vehicle (UAV) videos, we also evaluate STDFormer on VisDrone2019. The results show that STDFormer achieves state-of-the-art performance on VisDrone2019, which proves that it can handle small-scale object associations in UAV videos well. The code is available at https://github.com/Xiaotong-Zhu/STDFormer.**

Manuscript received 12 November 2022; revised 4 February 2023 and 10 March 2023; accepted 28 March 2023. Date of publication 3 April 2023; date of current version 30 October 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC3302200. This article was recommended by Associate Editor D. Gragnaniello. *(Corresponding author: Qing Song.)*

Mengjie Hu, Xiaotong Zhu, Haotian Wang, Chun Liu, and Qing Song are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: mengjie.hu@bupt.edu.cn; 17zxt826@bupt.edu.cn; lingxin@bupt.edu.cn; chun.liu@bupt.edu.cn; priv@bupt.edu.cn).

Shixiang Cao is with the Beijing Institute of Space Mechanics and Electricity, Beijing 100081, China (e-mail: cshixiang0110@126.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3263884.

Digital Object Identifier 10.1109/TCSVT.2023.3263884

***Index Terms*— Multi-object tracking, joint-motion-and-association, spatial-temporal transformer, contrastive learning, bidirectional matching.**

I. Introduction

MULTI-OBJECT tracking (MOT) plays an essential role in computer vision. It is widely used in video surveillance, autonomous driving, motion recognition and crowd behavior analysis. The goal of multi-object tracking is to find objects of interest in a video sequence and match the same objects frame-by-frame. This task can be divided into two subtasks: object detection and data association. According to the combination of the two subtasks, the mainstream methods can be separated into two paradigms: a) tracking-by-detection [1], [2], [3], [4] and b) joint-detection-and-tracking [5], [6], [7], [8], [9]. Researchers [6] have discovered that tracking loss and detection loss are incompatible and even somewhat impair detection performance when training a single backbone network jointly. Therefore, we believe that tracking-by-detection, which decouples the two subtasks, is a better solution. Thanks to the rapid development of object detection, an increasing number of tracking-by-detection methods have started to use powerful detectors that are already in place to implement high-performance tracking. Compared with object detection, data association methods have developed relatively slowly. Moreover, existing data association methods still have some limitations in multi-object tracking. To further improve the performance of multi-object tracking, more work is needed to develop data association methods.

Most of the data association techniques currently in use rely on appearance and motion data, with the former predominating and the latter typically employed as a secondary association. Unfortunately, appearance information performs poorly for occluded, blurred or similar objects due to noisy detections or similar appearances. Inspired by trajectory prediction [10], [11], [12], we recognize that if a high-performance motion model can perfectly predict the object trajectory, the model can mitigate the false associations caused by these aforementioned problems, as shown in Figure 1. Moreover, some recent studies have demonstrated that accurate tracking can be achieved by relying on motion information alone. Hence, we aim to construct a powerful motion model for robust data association.

In past studies on MOT tasks, the motion information was encoded either by conventional filtering or data-driven methods. Compared with conventional filtering [1], [4], [13], [14], [15], [16], [17], [18], data-driven

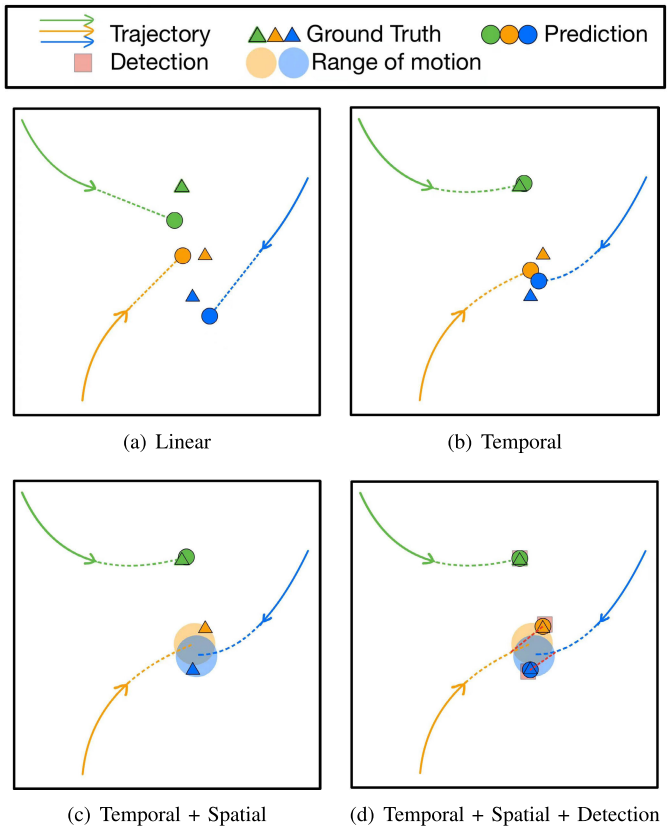


Fig. 1. Illustration of the idea of STDFormer. (a) is linear motion estimation, and (b) (c) (d) is nonlinear motion estimation. (a) The motion prediction based on the linear model has a large deviation from the true trajectory of the nonlinear motion. (b) Temporal-based motion estimation can more accurately determine the purpose of nonlinear motion accurately by mining historical motion information, and eventually improve motion prediction (**green**). However, targets may collide in space when motion estimation only considers temporal information (**yellow** and **blue**). (c) Conflicts can be avoided by adding spatial information based on temporal prediction, which is more consistent with the social attributes for people to flee from danger. (d) Detection constraints can further modify our predictions to be more biased toward ground truth, and can also indicate avoidance directions for conflicting trajectories (dashed **red** lines).

methods [19], [20], [21], [22], [23] have attracted less attention since most studies typically use motion models as an auxiliary clue. However, data-driven motion models have inherent advantages in handling nonlinear and sophisticated motion patterns which can not be expressed by filtering-based methods. Among data-driven methods, RNNs make up the majority of existing model frameworks since they are capable of handling long-range motion context dependencies. As the RNN blocks propagate, the relationship between two long-range motion information becomes very weak. RNN-based methods have limitations for modeling long-range motion. In summary, our motivation is to explore a new framework for constructing a powerful motion model for robust association under similar appearance, occlusion, or nonlinear motion.

Recent literature in sequence modeling confirmed that Transformer outperforms RNNs in long-range modeling and parallel computing due to the attention mechanism. Consequently, we leverage the Transformer architecture to model long-range motion information. On the other hand, to achieve accurate trajectory prediction, we deeply mine the influence of spatial-temporal motion information and observations, as shown in Figure 1. Contemporary trajectory prediction methods emphasize that the motion of an object is determined by multiple factors. As shown in Figure 1(b), the temporal information contains the motion intention of the object. Intention dependence is the dominant factor for the movement of an object in any scene. That is, each object has its own intended destination. Affected by social neighbors and the physical environment, objects make temporary changes during their progress. As shown in Figure 1(c), the spatial interaction constrains the movement distribution of the object in the next step. To obtain more accurate motion prediction, we also introduce the detections of the latest frame as prior knowledge to further restrict the object's potential positions, as shown in Figure 1(d).

To accomplish the aforementioned goals, we propose a Transformer-based multi-object tracking model with joint Spatial-Temporal motion and prior Detection, called STDFormer. STDFormer performs object motion prediction and affinity matrix calculation between detections and tracks by utilizing spatial-temporal constraints as well as potential detection. As shown in Figure 2, our model employs a parallel framework for jointly learning motion prediction and association, referred to as **joint-motion-and-association**. Unlike joint-detection-and-tracking, our approach achieves win-win cooperation because the jointly optimized feature space is exceptionally harmonious for both tasks. We believe that the motion prediction module in our framework can implicitly utilize the association information encoded in the interactive features of the current frame, making the feature representations for the motion prediction module close to the real detection feature representations. Specifically, the network consists of four components. Among them, our core design is to propose the STD (Spatial-Temporal-Detection) module in feature interaction, which utilizes the attention mechanism of Transformer to effectively realize the interaction of spatial, temporal and detection information. As for the token mechanism, we propose a learnable *trajectory token*, which obtains information about the entire trajectory by aggregating the features of all tracking boxes for a single trajectory in temporal attention. Each *trajectory token* represents an object and is used in spatial attention, detection attention and affinity calculation. For the association task branch, discriminative features and effective affinity calculation are crucial. However, existing methods optimize the affinity matrix between detections and tracks are very dependent on high-quality detection results, whereas missed and coarse detections seriously damage feature representation learning. To enhance the association task, we introduce contrastive learning for the first time in MOT tasks to learn motion representations, and utilize bidirectional matching to optimize the final affinity calculation.

To summarize, our contributions are as follows:

- We present a parallel framework for motion prediction and affinity calculation, named STDFormer. STDFormer improves the performance of both tasks via joint optimization.
- To the best of our knowledge, we are the first to explicitly use a Transformer architecture to model long-range

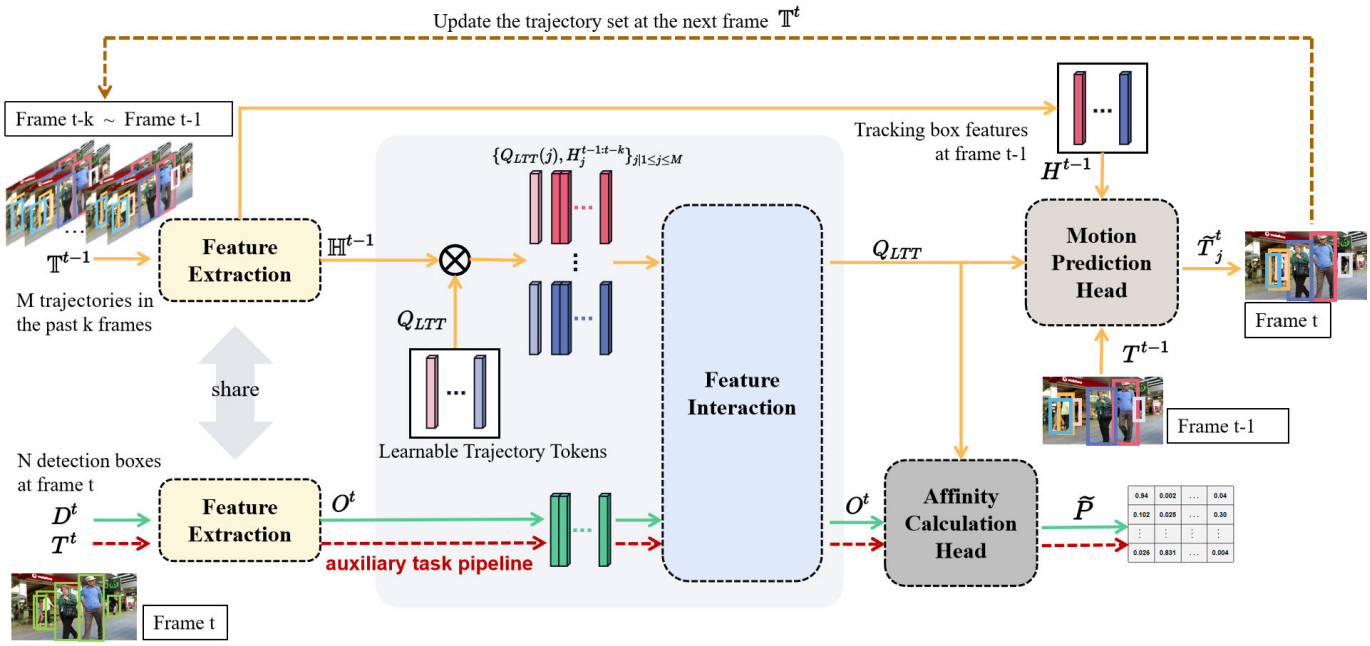


Fig. 2. The pipeline of the proposed joint-motion-and-association framework, STDFormer, where ⊗ represents concatenate features. The solid line is the data flow for training and testing. It takes the tracking boxes of the existing trajectories in the past k frames and the detection boxes of the current frame as input and outputs the localization of the trajectories in the current frame and the association probability matrix between the detections and the trajectories. In training, we add an auxiliary task, which inputs the ground truth of the tracking boxes in the current frame to the network instead of the detection boxes, and its data flow (**red dotted line**) is consistent with the detection data flow (**green solid line**). In testing, the predicted tracking boxes are used to update the trajectory set as input for the next frame (**brown dashed line**). The pipeline consists of four components: feature extraction, feature interaction, motion prediction head, and affinity calculation head (the detector component is omitted). Feature interaction is the core component of the whole method, which is used to mine the spatial-temporal information of the trajectories and interact with detection.

motion information in MOT without considering appearance information.

- We generate a learnable *trajectory token* by aggregating previous tracking box features in temporal attention to efficiently switch to spatial attention, detection attention and affinity calculation.
- We initially apply contrastive learning in MOT tasks to learn motion representation, and we further leverage bidirectional matching to improve affinity calculation.

II. Related Work

In this paper, with the aid of the existing detectors, we formulate multi-object tracking problem as a motion tracking model and utilize the Transformer architecture to mine long-range context dependency. Additionally, we employ contrastive learning to learn the highly discriminative feature representation and compute the matching similarity during the training process. Thus, we review the most relevant tasks of multi-object tracking, which are motion modeling, Transformer tracking, and contrastive learning.

A. Motion Modeling

Motion models aim to exploit motion information to predict spatial-temporal variations in trajectories, and infer spatial affinity between predictions and detections. The existing research methods of motion prediction can be roughly divided into two categories: traditional filtering-based methods and data-driven methods.

The common idea behind filtering-based motion models is to apply a Bayesian estimation framework. Following Bayesian methodology, different Bayesian filtering techniques have been developed for different scenarios. As one of the most classic Bayesian filters, the vanilla Kalman filter [24] with the assumption of uniform motion has gradually become the most popular motion model in multi-object tracking tasks [1], [4], [5], [16], [17], [25]. However, it is not suitable for nonlinear motion. To solve this problem, [26], [27] utilized the local linearization method based on the Kalman filter to address nonlinear motion, and particle filters [28] use a large number of random sampling points to approximate the posterior probability density function. The former methods have a large tracking error when applied to highly nonlinear motion; in contrast, the latter is close to optimal Bayesian estimation when the number of sampling points tends to infinity and can correctly handle nonlinear motion estimation. Considering the target interaction and variable number of targets in MOT, in some studies [29], [30] more advanced particle filter variants were applied, for example, Reversible-Jump Markov Chain Monte Carlo (RJMCMC) particle filter [31]. Nonetheless, particle filter-based methods have seldom been applied in MOT, given that a large number of sampled particles leads to a sharp increase in computation.

As traditional filtering-based methods cannot describe the motion pattern of objects accurately, more complex data-driven motion methods have been proposed to achieve more complicated state prediction. In the existing multi-object tracking literature, data-driven approaches have mainly been based

on RNN architecture to mine the temporal information of motion. [32] introduced recurrent networks to simulate a Bayesian filter for motion estimation. After [32], different types and combinations of RNNs were used [19], [20], [21], [22], [23] to realize deterministic or probability distribution prediction of object motion. However, the relationship between long-range motion information will become very weak as RNN blocks propagate. In addition, graph models are also considered for modeling object motion. Reference [33] proposed two modules, box embedding and tracklet embedding, which used the attention and reconstruction mechanisms of deep graph convolutional networks to model local and global motion information, respectively. It should be pointed out that the global motion modeling in [33] was based on tracklets. It did not establish connections to box nodes of non-adjacent frames, which made it unable to mine long-range motion information from low-level features.

Recently, Transformer has been shown to perform better than RNNs in long-range modeling and parallel computing. Compared with the graph models, Transformer does not need to pre-design graph. It can directly build a fully connected graph through self-attention. Inspired by this, our approach adopts the Transformer architecture for long-range modeling of motion information to extract more accurate motion features and mine more useful motion cues.

B. Transformer In Tracking

Transformers have achieved great success in natural language processing and have gradually been applied to fields such as computer vision in recent years. The existing Transformer-based MOT methods can be categorized as short-term models or long-term models, depending on the range of information.

Short-term models only consider the local information of adjacent frames to learn and infer object trajectories. Reference [34] leverages features from previously detected objects as queries to discover associated objects in subsequent frames. Reference [35] introduces pixel-level dense queries with Transformers and proposes a dual decoder to output center heatmap and object size, as well as tracking displacements in adjacent frames. Reference [36] uses previously detected results as references to aggregate the corresponding features from the combined features of the adjacent frames. For each reference, [36] then concurrently predicts the one-to-one track state.

In contrast, long-term models have access to longer-range information beyond two frames, which theoretically can obtain more accurate results by using more contextual cues. References [37] and [38] achieve detection and data association synchronously by integrating object and autoregressive track queries as input to the Transformer decoder in the next time step. These methods implicitly apply long-term temporal information by propagating track queries frame-by-frame. References [39] and [40] integrate long-term temporal information by focusing on all past embeddings for each individual object and use this information to predict the appropriate embedding for the current time step. Reference [41] constructs a spatial map for objects appearing in each frame of the past T frames, and leverages Transformer architecture to jointly learn the spatial and temporal relationships of small trajectories as well as candidate trajectories for efficient association. However, these methods which explicitly utilize multi-frame information, usually require constructing a large spatial-temporal memory to store past observations of tracked objects, which consumes expensive storage and computing resources.

All of the aforementioned approaches, which explicitly use multi-frame information, take full advantage of Transformer's long-term modeling capabilities. The high resource cost of this technique is due to the storage of high-dimensional visual features and motion features. To address this problem, our method only stores and encodes low-dimensional position information of objects in the past multiple frames and discards appearance information.

C. Contrastive Learning

Contrastive learning is an effective representation learning method. It has strong discriminative power to distinguish the same object from other objects by pushing away negative embedding distances and narrowing the positive ones. This technology has recently obtained amazing results in various fields, such as computer vision [42], [43], [44], natural language understanding [45], [46], [47], and text-image matching. Contrastive learning applied specifically to multi-object tracking has received less attention. The few relevant studies [48], [49] also focused on appearance features, which can be learned effectively by contrastive learning. However, for motion features, determining how to design positive and negative samples effectively is very important. Recently, in the field of trajectory prediction, [50] was the first to learn motion representation by contrastive learning. Reference [50] proposes a social sampling strategy. It constructs the positive event from the ground-truth location of the primary agent and the negative events from the regions of other neighbors, given that one location cannot be occupied by multiple agents at the same time. Our work is inspired by the safety of dealing with the sampling strategy proposed by [50].

Our work attempts to aggregate the historical features of the tracks to obtain the trajectory information of the objects in the current frame and compute the similarity with the detections. Both subtasks require the trajectory token features to be as close as possible to the motion embedding of objects in the current frame. Therefore, our method takes all the objects appearing in the current frame as samples. The samples belonging to the same objects as the historical tracks are positive samples, and the samples belonging to different objects are negative samples. This method enables the trajectory token features of tracks to effectively learn the behavioral intentions of objects. To the best of our knowledge, we are the first to utilize the contrastive idea for motion representation learning in MOT.

III. Methodology

In this section, we first introduce the overall framework of STDFormer (Figure 2). Then, we provide a detailed description of the model, training and inference process.

A. Overview

Given a sequence of video frames, the goal of MOT is to detect and associate the targets frame-by-frame. In this paper, we address the problem of online multi-object tracking in a scene by following a tracking-by-detection paradigm.

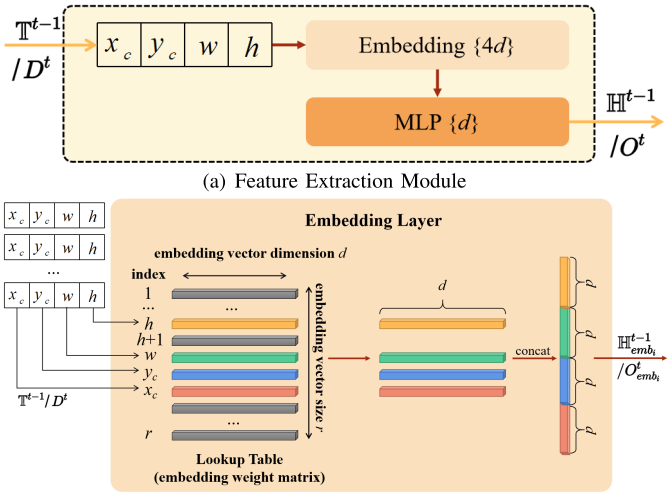
Before introducing the pipeline of our tracking algorithm, we define some symbolic expressions. Specifically, let $D^t = \{d_1, \ldots, d_N\}$ denote the detections of N objects at frame T. Each detection $d_i = [x, y, w, h]$ is represented as the center point and size of the bounding box. Let $\mathbb{T}^{t-1} = \{T_1^{t-1:t-k}, \ldots, T_M^{t-1:t-k}\}$ denote the trajectories of M tracked objects at frame $t-1$. Every trajectory $T_j^{t-1:t-k} = [T_j^{t-1}, \ldots, T_j^{t-k}]$ consists of the tracking boxes from the j-th tracked item over the previous k frames in reverse order of time. Note that if the length of a trajectory T_j at frame t is s ($s < k$), we need to pad $T_j^{t-1:t-k}$. Specifically, we assume that the state of the short trajectory T_j is standing still until the trajectory initialization, so we repeat the initial tracking box T_j^{t-s} of the trajectory $k-s$ times and regard them as hypothetical tracking boxes for T_j from frame t-k to frame t-s+1:

$$T_j^{t-1:t-k} = [\overbrace{T_j^{t-1}, \ldots, T_j^{t-s}}^{s}, \overbrace{T_j^{t-s}, \ldots, T_j^{t-s}}^{k-s}] \tag{1}$$

The j-th tracked object's tracking box at frame t $T_j^t = [x, y, w, h]$ takes the same definition as detection, where $t \in \{t-1, \ldots, t-k\}$. Based on the definition, the tracking boxes of all tracked objects at frame t can be represented as $T^t = \{T_1^t, \ldots, T_M^t\}$. Let $\Delta^{t-1:t} = \{\delta_1, \ldots, \delta_M\}$ denote the tracked objects' displacements between frame $t-1$ and frame t. Each tracked object's displacement $\delta_j^{t-1:t} = [dx, dy, dw, dh]$ is expressed as the change in the center point and size of the tracking box between two frames. Let $A_{N:M}^t$ denote the affinity matrix at frame t, which indicates the similarity of detections and tracked objects.

The workflow of STDFormer contains two stages: 1) At frame t, we apply a high-performance detector [51] to identify and locate all targets D^t; 2) the proposed STDFormer takes object trajectories $\mathbb{T}^{t-1}$ up to time $t-1$ and detections D^t at time t as input and outputs each tracked object's displacements $\Delta^{t-1:t}$ as well as the affinity matrix $A_{N:M}^t$. During training, we introduce an auxiliary task to learn more accurate motion feature representation. This stage is similar to the second stage, except that the input now uses tracking boxes T^t from the t-th frame instead of detections D^t. During inference, we propose a step-by-step association strategy, which utilizes an affinity matrix to match first and then adopts the IoU similarity between detections and predicted tracking boxes to match the remaining detections and tracks.

Specifically, as shown in Figure 2, STDFormer consists of four main components: 1) a feature extraction module that encodes the current frame's detection information along with the tracked objects' historical motion data from the previous frames; 2) a feature interaction module that takes advantage of Transformer's attention mechanism to aggregate trajectories' spatial-temporal features and detection features; 3) a motion prediction head that generates displacements of tracked objects in adjacent frames, based on the difference between tracking box features in the previous frame and aggregated features of the trajectories; and 4) an affinity calculation head that computes the affinity matrix between detections and tracked objects for data association.



(a) Feature Extraction Module

(b) Embedding Layer

Fig. 3. Feature extraction module. The center points and sizes of the input tracking boxes and detection boxes are denoted as (x_c, y_c, w, h). The embedding layer maps each dimension of the low-dimensional input to the high-dimensional space and connects them into embedding vectors ($R^4 \Rightarrow R^{4d}$). The MLP layer fuses and reduces the dimension of the embedding vectors of each dimension ($R^{4d} \Rightarrow R^d$), and finally outputs the extracted feature vector $\mathbb{H}^{t-1}$ and O^t.

B. Feature Extraction

Both historical motion information and detection information represent spatial information. Therefore, they can be embedded into the same feature space by a shared feature extractor. As shown in Figure 3, taking the input object trajectories up to time $t-1$ $\mathbb{T}^{t-1} \in R^{M \times k \times 4}$ and detections at time t $D^t \in R^{N \times 4}$, we utilize an embedding layer to extract their features. The embedding layer is a lookup table that can be trained. It creates a weight matrix $W \in R^{r \times d}$, where r is the embedding vector size and d is the embedding vector dimension. This layer takes only positive integers (indices) as input and converts them into fixed-size embedding vectors. Specifically, the embedding layer converts each integer i into the i-th row of the embedding weight matrix. As shown in Figure 3(b), in this work, we take the center point coordinates and bounding box size values of trajectory and detections as a set of positive integer indices, retrieve the corresponding embedding vectors from the embedding weight matrix and concatenate them as the output of the embedding layer. We define the extracted features for trajectories and detections as $\mathbb{H}_{emb}^{t-1} \in R^{M \times k \times 4d}$ and $O_{emb}^t \in R^{N \times 4d}$, where d indicates the embedded dimension. The calculation process of the embedding layer is as follows:

$$x_{emb} = W[x_c] \tag{2}$$

$$y_{emb} = W[y_c] \tag{3}$$

$$w_{emb} = W[w] \tag{4}$$

$$h_{emb} = W[h] \tag{5}$$

$$\mathbb{H}_{emb}^{t-1} / O_{emb}^t = concat(x_{emb}, y_{emb}, w_{emb}, h_{emb}) \tag{6}$$

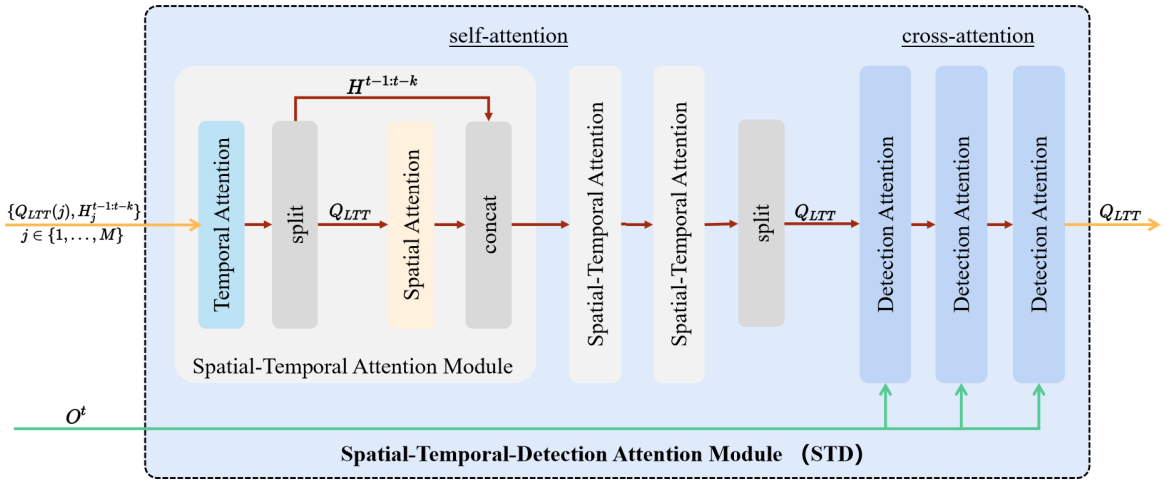


Fig. 4. Feature interaction module. STD concatenates the learnable trajectory tokens Q_{LTT} and the extracted features of the historical tracking boxes $H^{t-1:t-k}$ as input, and outputs the iteratively updated trajectory tokens Q_{LTT} after three spatial-temporal attention submodules. Each spatial-temporal attention submodule is composed of a temporal attention layer and a spatial attention layer, and realizes feature interaction based on a self-attention mechanism. Then, the updated trajectory tokens Q_{LTT} and the extracted feature of the detection boxes O^t are fed to the three detection attention layers, and Q_{LTT} is used as the query vector to realize the cross-attention with O^t. Finally, the STD outputs the updated trajectory tokens Q_{LTT}.

Furthermore, we use a 1-layer Multi-Layer Perceptron (MLP) to fuse the features of the last dimension and map them to the final embedded representation, $\mathbb{H}^{t-1} \in R^{M\times k\times d}$ and $O^t \in R^{N\times d}$.

In summary, our embedding layer maps low-dimensional spatial data to high-dimensional feature space, which obtains more discriminative feature representations and is propagated to the next feature interaction module.

C. Transformer for Feature Interaction

In this subsection, we demonstrate how to leverage Transformer's attention mechanism for feature interaction: feature interaction using Spatial attention, Temporal attention, and Detection attention, STD for short.

After feature extraction, $\mathbb{H}^{t-1} = \{H_1^{t-1:t-k}, \ldots, H_M^{t-1:t-k}\}$ and $O^t = \{O_1, \ldots, O_N\}$ are passed through the STD module to implement feature interaction. As shown in Figure 4, the STD module consists of three different types of attention submodules: a) temporal attention, b) spatial attention, c) detection attention. In fact, the three attention modules all follow the same encoder layer paradigm [52], except that the query and key are used as input. We first briefly introduce the tokens involved in the three submodules, and then elaborate on the detailed differences of each module.

1) Token Mechanism: The token embeddings are delivered as input to the attention module with a 1D sequence format. In the STD module, tokens can be classified into the following three categories according to the source of embedding:

- detection token: embedding of a detection at frame t $O_i \in O^t$, where $O_i \in R^d$, $i \in \{1, \ldots, N\}$.
- track token: embedding of a history tracking box for a single tracked object at frame $t-1$ over the past k frames $H_j^p \in H_j^{t-1:t-k} = \{H_j^{t-1}, \ldots, H_j^{t-k}\}$, where $H_j^p \in R^d$, $j \in \{1, \ldots, M\}$, $p \in \{t-1, \ldots, t-k\}$.
- trajectory token: learnable embedding of a tracked object at frame $t-1$. It aggregates the historical tracking boxes' features of a single object, whose state at the output of the attention module implicitly serves as the tracked object's tracking box embedding at frame t. We define the Learnable Trajectory Token of the j-th tracked object as $Q_{LTT}(j) \in R^d$.

2) Temporal Attention: The long-range motion information of a tracked object implicitly indicates its motion trend. For each tracked object, we independently execute the interaction of temporal motion features over the previous k frames. Therefore, considering that each tracking target has a temporal attention layer, the temporal attention module dynamically adjusts the number of parallel temporal attention layers along with the tracking target. As illustrated in Figure 4, we construct the encoded query and key vectors for the intra-track temporal attention of the j-th tracked object at frame $t-1$ in a self-attention manner as follows:

- query: embeddings of a tracklet token and k track tokens of the target in the past k frames $\{Q_{LTT}(j), H_j^{t-1}, \ldots, H_j^{t-k}\}$, where $j \in \{1, \ldots, M\}$.
- key: similar to the query.

In particular, position embeddings with sinusoidal format [53] are added to the above embeddings to retain temporal information. DIn contrast to the standard Transformer, we only add them once to the relevant temporal embeddings of the feature extraction module output. The inputs of temporal attention become time-dependent by adding the position embeddings to the temporal embeddings, which is essential for STD mining movement trends.

3) Spatial Attention: The interaction of spatial features reflects social interaction. In social interactions, targets' decisions frequently follow logical social norms. Regarding spatial information, we exploit the interaction mechanism of self-attention to measure the relative spatial position between the targets and their neighbors, which assists them in making movement decisions. Thus, the query and key vectors for the inter-track spatial attention module are as follows:

- query: embeddings of the tracklet tokens of all the targets at frames $t-1$ $\{Q_{LTT}(1), \ldots, Q_{LTT}(M)\}$.
- key: similar to the query.

4) Detection Attention: The detections provide the potential distribution position of targets in the current frame, which can be used as prior knowledge to constrain the possible position of the tracking targets in the current frame. In the detection attention module, we focus on the relative spatial position between the tracked objects' prediction and the detections. Unlike the spatial attention module, we aim to narrow the feature representation of matched pairs in the embedding space by measuring relative positions and performing the interaction between detection and tracking in a cross-attention manner. Its query and key vectors are as follows:

- query: embeddings of the tracklet tokens of all the targets at frames $t-1$ $\{Q_{LTT}(1), \ldots, Q_{LTT}(M)\}$.
- key: embeddings of all the detection tokens at frame t $\{O_1, \ldots, O_N\}$.

As demonstrated in Figure 4, we interleave the temporal attention and spatial attention modules by L times in the STD module to aggregate the spatial-temporal motion features of the tracklets. Then, we iterate the detection attention module L times to make the aggregated motion features interact with the detection features of the current frame. After feature interaction, we have M trajectory tokens for tracked objects $\{Q_{LTT}(1), \ldots, Q_{LTT}(M)\}$.

D. Motion Prediction Head

The motion prediction task aims to forecast the position of each tracked object in the current frame. However, inspired by [54] and [55], directly training a model to adapt to shapes of various objects is a challenging task, which results in poor performance in precise localization. In contrast, predicting the candidate box offset is simpler. Thus, STDFormer takes the position of the target in the previous frame as prior information and predicts the change in its position in the current frame instead of directly regressing the position of the target in the current frame.

To improve the learning ability of the model, we assess the object motion distribution using several benchmark datasets in Appendix A and finally propose two motion prediction head variants based on the data distribution characteristics:

1) Linear Motion Prediction Head (LMPH): This variant directly predicts the displacement δ of each tracked object between the adjacent frames, which is suitable for datasets with simple motion patterns and small offset variance;
2) Exponential Motion Prediction Head (EMPH): This variant scales the bounding box of the tracked object in the previous frame by predicting an exponential adjustment factor ζ to obtain the tracking box in the current frame, which is suitable for datasets with complex motion patterns and large offset variance.

The quantitative results in Section IV-C further verify the superiority of the two variants in different scenarios. More details about object motion analysis on benchmark datasets can be found in Appendix A. The detailed motion prediction head architecture is shown in Figure 5.

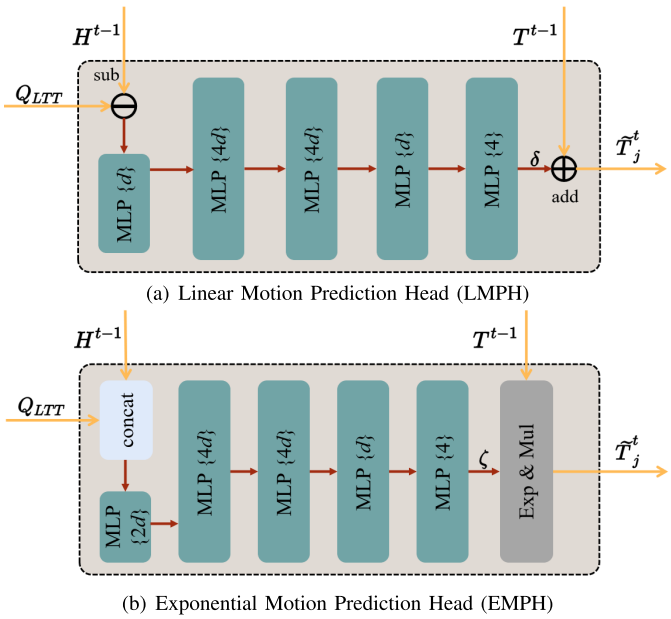


Fig. 5. Motion prediction head, where $\ominus$ represents feature addition and $\oplus$ represents vector addition. There are two variants of this module: (a) Linear Motion Prediction Head (LMPH): This variant subtracts the trajectory token features after the feature interaction module and the tracking box features of the previous frame after the feature extraction module. Then, the subtracted features are passed to a five-layer MLP, which outputs the displacement δ of each tracked target between adjacent frames. Finally, the displacement is added to the tracking box of the previous frame to obtain the tracking box prediction of each tracked target in the current frame. (b) Exponential Motion Prediction Head (EMPH): This variant first concatenates the trajectory token features after the feature interaction module and the tracking box features of the previous frame after feature extraction module. Then, the concatenated features are passed to a five-layer MLP, which outputs the exponential adjustment factor ζ of each tracked target between adjacent frames. Finally, the exponential adjustment factor is exponentially processed, and multiplied by the tracking box of the previous frame to obtain the tracking box prediction of each tracked target in the current frame.

1) Linear Motion Prediction Head: As depicted in Figure 5, to learn the displacement, we first utilize this simple subtraction between trajectory token features from the feature interaction module and the motion features from feature extraction module for each tracked object as the displacement feature B_j:

$$B_j = Q_{LTT}(j) - H_j^{t-1}, \tag{7}$$

where $Q_{LTT}(j)$ is the implicit motion feature of tracked object j at frame t and H_j^{t-1} is the motion feature of tracked object j at frame $t-1$.

After obtaining the displacement features, we propose a five-layer MLP with a nonlinear operator that takes the displacement feature B_j as input and produces a scalar value δ_j that represents the displacement of the tracking target j between frame $t-1$ and frame t:

$$\delta_j = MLP(B_j). \tag{8}$$

Finally, to obtain the position prediction $\tilde{T}_j^t$ of tracked object j at frame t, we perform simple additive post-processing:

$$\tilde{T}_j^t = T_j^{t-1} + \delta_j. \tag{9}$$

2) Exponential Motion Prediction Head: As shown in Figure 5(b), the main difference from LMPH is the processing of input and output. Instead of subtracting, we first concatenate trajectory token features from the feature interaction module and the motion features from the feature extraction module for each tracked object. Then, the concatenated features are also fed to a five-layer MLP and MLP outputs a scalar as an exponential adjustment factor ζ. Finally, an exponential operation is performed on this factor to obtain a scaling factor that is multiplied by the bounding box of the previous frame to obtain the position of the tracking target in the current frame. The specific calculation process is as follows:

$$\zeta = MLP(concat(Q_{LTT}(j), H_j^{t-1})), \tag{10}$$

$$\widetilde{T}_j^t = exp(\zeta) \cdot T_j^{t-1} \tag{11}$$

3) Box Loss: We describe motion prediction as a tracking box regression task. L1 loss, the most popular regression loss, is sensitive to bounding box size. To alleviate this problem, we employ a linear combination of smooth L1 loss [56] and scale-invariant generalized IoU loss [57] to optimize the prediction of tracking boxes:

$$\mathcal{L}_{box}(T_j^t, \widetilde{T}_j^t) = \lambda_{l1}\mathcal{L}_{l1}(T_j^t, \widetilde{T}_j^t) + \lambda_{iou}\mathcal{L}_{iou}(T_j^t, \widetilde{T}_j^t), \tag{12}$$

where $\lambda_{l1}, \lambda_{iou} \in R$ are hyperparameters, $\widetilde{T}_j^t$ is the estimated tracking box of tracked object j at frame t, T_j^t is the ground truth of the tracking box of tracked object j at frame t, $\mathcal{L}_{l1}$ is the smooth L1 loss and $\mathcal{L}_{iou}$ is the GIoU loss.

E. Affinity Calculation Head

The goal of the affinity calculation head is to compute the pairwise similarity scores of detected and tracked objects for data association. The process of this module is shown in Figure 6. Given the features of N detections from the feature extraction module and the trajectory token features of M tracked objects from the feature interaction module, we calculate the affinity matrix at frame t $A_{N:M}^t$ as the inner product between each pair of detection features O_i and tracked object trajectory token features $Q_{LTT}(j)$:

$$A_{N:M}^t(i, j) = O_i \cdot Q_{LTT}(j), \tag{13}$$

where $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M\}$.

1) Bidirectional Matching: A well-performing affinity matrix should satisfy bidirectional optimal matching. In other words, when detection-to-track or track-to-detection pairs identify the best match, symmetrical track-to-detection or detection-to-track scores should be the highest. As a result, we use the *dual-softmax* proposed by [58], which modifies the initial affinity matrix by introducing a prior probability matrix $\widetilde{P}_{prior} \in R^{N \times M}$ generated in the cross direction. We can filter out the challenging cases with a high detection-to-track affinity score but a low track-to-detection affinity score by acalculating the dot product between the prior probability matrix and the initial affinity matrix.

Specifically, we successively apply softmax on the two dimensions of $A_{N:M}^t$ to obtain the probability $\widetilde{P} \in R^{N \times M}$ of soft mutual nearest neighbor matching:

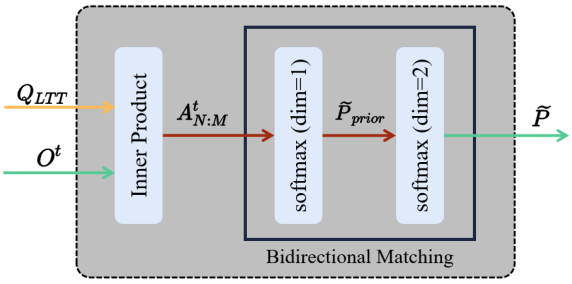


Fig. 6. Affinity calculation head. Given the trajectory token features Q_{LTT} after the feature interaction module and the detection box features O^t after the feature extraction module, this module first calculates the affinity matrix $A_{N:M}^t$ between the detection and tracks based on the inner product. The affinity matrix $A_{N:M}^t$ is then fed to the bidirectional matching submodule to obtain a detection and tracking association probability matrix $\widetilde{P}$.

$$\widetilde{P}_{prior} = softmax(A_{N:M}^t/\tau, dim = 1), \tag{14}$$

$$\widetilde{P}(i, j) = softmax(\widetilde{P}_{prior} \cdot A_{N:M}^t, dim = 0)_{ij} \tag{15}$$

where $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M\}$ and τ is a temperature parameter.

2) Label Assignment: Unlike box loss, the ground truth of the affinity matrix cannot be given directly or indirectly by the dataset when computing affinity loss, because the performance of the chosen detector greatly influences the detection directly tied to the affinity matrix. For this reason, we refer to [59] and use the Hungarian algorithm to find an optimal bipartite matching between predicted detection and ground truth tracked objects for label assignment. In contrast to [59], our approach takes only the box loss into account when calculating the matching cost and ignores the categorization loss. For more implementation details, please refer to [59].

3) Affinity Loss: After the label assignment, we obtain the ground truth affinity matrix. Each row and column of the matrix is either a one-hot vector, or an all-zero vector, where 1 to indicates a matching detection-track pair at that location and 0 indicates a mismatch. We can also treat the ground truth affinity matrix as a probability matrix, denoted as P.

During training, we set a *max objects* hyperparameter, Z, to achieve parallel training. When the number of tracked or detected targets as input is less than the *max objects*, we pad the input with 0. In other words, the dimension of our output must be $Z \times Z$. When $M < Z$ and/or $N < Z$, the output will be extended to dimension $Z \times Z$ by padding with 0. Obviously, the number of zeros in P is much greater than the number of ones. To address the imbalance between positive and negative samples, we use focal loss [60] as affinity loss:

$$\mathcal{L}_{aff}(\widetilde{P}(i, j)) = -\alpha(1 - \widetilde{P}(i, j))^{\gamma} \log(\widetilde{P}(i, j)), \tag{16}$$

where α and γ are hyperparameters.

F. Training

To summarize the above, whether focusing on motion prediction or affinity calculation, the primary task is to learn the motion feature representation of tracked objects in the

current frame. However, as mentioned in Section III-E, the performance of the affinity calculation head is highly dependent on the prediction of the detector. Missed, wrong, and rough detections cause serious damage to the motion feature representation due to the feature interaction. In this regard, we believe that the ideal situation of the current frame is that no new targets appear, no old targets are lost, and each tracked target can find its accurate localization in the current scene. Motivated by this assumption, we introduce an auxiliary task to discover more trustworthy motion feature representations.

The most intuitive approach is to follow the STD pipeline above, using the ground truth tracked boxes T^t at frame t instead of detections D^t as input, and retrain the STD. However, as shown in Figure 2, instead of training twice, we achieve parallel training by adding an auxiliary branch.

Specifically, the auxiliary task shares the feature processing flow of detection in STD and outputs a track-to-track affinity matrix whose ground truth is the identity matrix. We define this track-to-track affinity matrix at frame t as $A^t_{M:M}$:

$$A^t_{M:M}(i,j) = Q_{LTT}(i) \cdot V_j, \tag{17}$$

where V_j is the embedding of the j-th ground truth tracked boxes T^t_j at frame t from the feature extraction module.

In fact, our auxiliary task subtly introduces the idea of contrastive learning. Contrastive learning requires that each sample has one positive sample and several negative samples. Like [50], we take the aggregated historical motion information of each tracked object as a sample, and we hope to make the feature representation of the sample close to the motion feature representation of the corresponding target in the current frame through contrastive learning. Therefore, we regard the ground truth position of the tracked object in the current frame as a positive sample, and the ground truth position of the rest of the tracked objects in the current frame as a negative sample. Obviously, it is easy to meet the above requirements in the setting of our auxiliary task. Furthermore, [50] pointed out that such a sampling strategy can effectively not only narrow the feature representation of similar samples, but also push the samples away from the negative sample points, avoiding location conflicts and eventually effectively introducing security considerations.

1) Contrastive Loss: After obtaining the track-to-track affinity matrix at frame t as $A^t_{M:M}$, we also perform the *dual-softmax* operation (Equation 14, 15) on it. Unlike Equation 16, the affinity loss of our auxiliary task replaces the focal loss with cross entropy loss which is called Dual Softmax Loss (DSL) in [58]. It can be regarded as a variant of InfoNCE Loss, a contrastive loss that is frequently employed. This contrastive loss optimization seeks to learn well-performing historical aggregated representations by maximizing the mutual information between historical aggregated representations and ground truth motion representations. Specifically, the contrastive loss of the auxiliary task is calculated as follows:

$$Pr(i,j) = softmax(A^t_{M:M}/\tau, dim = 1), \tag{18}$$

$$\mathcal{L}_{aux} = -\frac{1}{M}\sum_{i}^{M} \log \frac{exp(Q^{(}_{LTT}i) \cdot V_i^{+} \cdot Pr(i,i))}{\sum_{j=1}^{M} exp(Q^{(}_{LTT}i) \cdot V_j \cdot Pr(i,j))}, \tag{19}$$

where Pr is the prior matrix of $A^t_{M:M}$ (same as Equation 14), τ is a temperature hyperparameter.

2) Total Loss: We jointly train the motion prediction, affinity calculation and auxiliary task branches by adding the losses (i.e., Equations 12,16,19) together. In particular, we leverage the uncertainty loss proposed in [61] to automatically balance multitask learning:

$$\mathcal{L}_{task} = w_1\mathcal{L}_{box} + w_2\mathcal{L}_{aff}, \tag{20}$$

$$\mathcal{L}_{total} = \frac{1}{2}\left(\frac{1}{e^{w_3}}\mathcal{L}_{task} + \frac{1}{e^{w_4}}\mathcal{L}_{aux} + w_3 + w_4\right), \tag{21}$$

where w_1 and w_2 are fixed hyperparameters that balance the motion prediction task and the affinity calculation task, respectively, and w_3 and w_4 are learnable parameters that balance the above two main tasks and the auxiliary task. Finally, it should be emphasized that the auxiliary task is only used during training.

G. Inference

During inference, we process the video stream online in a frame-by-frame manner. We construct a temporal history buffer of T frames to store each tracked object's tracking boxes. Following [4], we divide all detection boxes into high score detection boxes and low score detection boxes according to the detection score threshold θ_{det}. For each individual frame t, given all the high score detections for the current frame and a sequence of the tracked objects' historical tracking boxes for the previous frame, STD takes them as input for inference and outputs an affinity matrix with probability format (after **dual softmax**) as well as the tracked objects' tracking boxes prediction in the current frame.

Data Association. After network inference, we perform association by using the network outputs. We follow the standard online tracking paradigm to associate boxes. In the first frame, we first initialize some tracklets based on the high score detection boxes. In the following frames, we link the detection boxes to the existing tracklets according to a step-by-step association strategy as follows:

1) high score detection boxes are associated with all existing tracklets, including tracked, lost, and unconfirmed tracklets;
2) low score detection boxes are associated with the unmatched tracklets in the previous step;
3) the high score detection boxes are associated with all the remaining unmatched tracklets.

In each step of the above association process, we obtain a cost matrix C by performing a weighted sum operation on the center point distance cost C_{cdist}, spatial overlap rate (GIoU) cost C_{giou} and affinity cost C_{aff} between the prediction boxes and the detection boxes. The calculation process is as follows:

$$C = \lambda_{cdist}C_{cdist} + \lambda_{giou}C_{giou} + \lambda_{aff}C_{aff} \tag{22}$$

We found that a large amount of noise and false detections in low score detection boxes can seriously damage the tracking performance through ablation experiments. Therefore, we only feed high score detection boxes to STDFormer, which causes the model to not output an affinity matrix between the prediction boxes and the low score detection boxes and step 2 of the above association does not consider the affinity cost. After obtaining the cost matrix, we feed it to the Hungarian algorithm and filter matching pairs that are far away according to a fine-grained spatial overlap rate threshold design.

In addition, determining the birth and death of a trajectory is also an important part of MOT. When a detection box fails to match an existing tracked object, we can assume that a new target has entered the scene and constructing a unique identification is necessary. To avoid false positive detection boxes, we initialize a new trajectory only if the object exists for more than three consecutive frames (except the first frame). On the other hand, when the tracking object does not find a matching detection box in the current scene, we need to distinguish whether the object is lost or has left the current scene. We propose a boundary judgment method. Specifically, when the center point of the predicted tracking box is less than x pixels from the image frame boundary, we consider the object to have already left the current scene and destroy the corresponding identity of the tracking object. In contrast, when the object is far from the image frame boundary, we consider the object to be only temporarily lost. Furthermore, when the object is lost in fewer than s frames, we fill the tracklet of the lost tracked object with the predicted tracking box, and reactivate this tracked object if the object reappears. When the object is missing more than s frames, we also consider the tracking object to have left the current scene and destroy its identity.

IV. Experiments

In this section, to assess the performance of the proposed approach, we describe experiments conducted on MOTChallenge. First, we introduce evaluation datasets, evaluation metrics and implementation details. Then, we compare the proposed method with state-of-the-art methods and show quantitative results on MOTChallenge. In addition, we provide a qualitative analysis of the results. Finally, we demonstrate the effectiveness of each module through ablative studies.

A. Datasets and Metrics

1) Datasets: We evaluate STDFormer on the MOT17 [62], MOT20 [63] and DanceTrack [64] datasets in accordance with the "private detection" protocol. All three datasets are related to pedestrian tracking. The difference is that the pedestrian motion patterns of MOT17 and MOT20 are relatively simple and close to linear motion, while the motion of DanceTrack is complex and highly nonlinear. The details of these datasets are as follows:

1) MOT17: This dataset includes seven scenes of indoor and outdoor public places with pedestrians. The videos of each scene are divided into two segments for training and testing. Specifically, this dataset contains 14 video sequences, 7 of which are used for training and 7 for testing. The training datasets consist of 15948 frames with a total of 1638 identities and 336891 labeled boxes. The test datasets consist of 17757 frames with a total of 2355 identities and 564228 labeled boxes.
2) MOT20: This dataset consists of 8 video sequences from 3 different scenes, half of which are used as the training dataset and half as the testing dataset. The training dataset consists of 15948 frames with a total of 1638 identities and 1336920 labeled boxes. The test dataset consists of 4479 frames with a total of 1501 identities and 765465 labeled boxes. Obviously, compared with MOT17, the pedestrian density in the scene of MOT20 datasets is higher, and the average crowd density reaches 246 pedestrians per frame. All sequences were shot from above.
3) DanceTrack: This dataset includes 100 videos (40 training videos, 25 validation videos and 35 test videos) covering group dance, kung fu, gymnastics, and other activities. It contains 990 unique instances with an average length of 52.9 s, 105k frames and 877k high-quality bounding boxes by 20 FPS annotation. In addition, the targets in DanceTrack are very clear and in close range, so object detection generally does not limit the algorithm, and emphasis is placed on evaluating the data association performance of the algorithm. Furthermore, since the targets in DanceTrack have similar or even identical appearances and there are a large number of occlusions, position interleaving, and complex nonlinear motion patterns among the targets, the dataset encourages the algorithm to mine matching cues other than appearance, such as motion trajectories.

Note that MOT17 and MOT20 are both benchmark datasets for MOTChallenge and the most commonly used benchmarks for multi-object tracking. Both only have training and testing sets, while validation sets are not available. Therefore, in the ablation experiments, we follow the experimental setup in most of the literature [1], [4], [5], [13]. We divide each video in the MOT17 training set into two equal parts. The first half is used for training, and the second half is used for validation. In addition, we also evaluate STDFormer on the VisDrone2019 [65] dataset to verify the effectiveness of the proposed method in UAV videos. More details about STDFormer in unmanned aerial vehicle videos can be found in Appendix B.

2) Metrics: We use the CLEAR MOT Metrics [66], [67] and HOTA [68] to quantitatively evaluate the overall tracking accuracy. All the metrics are listed as follows:

- MOTA(↑): Multi-Object tracking accuracy.
- IDF1(↑): ID F1 score.
- HOTA(↑): Higher order tracking accuracy.
- MT(↑): Mostly tracked targets.
- ML(↑): Mostly lost targets.
- FP(↑): The total number of false positives.
- FN(↑): The total number of false negatives (missed targets).
- AssA(↑): Association accuracy.
- ID Sw.(↑): Number of identity switches.

TABLE I

COMPARISON OF OUR METHOD WITH STATE-OF-THE-ART MOT ALGORITHMS UNDER THE "PRIVATE DETECTOR" PROTOCOL ON THE MOT17 TEST SET. THE BEST RESULTS OF THE FOUR GROUPS OF METHODS ARE MARKED IN YELLOW, GREEN, BLUE AND RED, RESPECTIVELY

	Methods	MOTA↑	IDF1↑	HOTA↑	MT↑	ML↓	FP↓	FN↓	AssA↑	ID Sw.↓	Frag↓
Benchmark	TubeTK [71]	63.0	58.6	48.0	31.2%	19.9%	27060	177483	45.1	4137	5727
	CenterTrack [72]	67.8	64.7	52.2	34.6%	24.6%	**18498**	160332	51.0	3039	6102
	TraDes [6]	69.1	63.9	52.7	36.4%	21.5%	20892	150060	50.8	3555	4833
	MAT [73]	69.5	63.1	53.8	43.8%	18.9%	30660	138741	51.4	**2844**	**3726**
	SOTMOT [74]	71.0	71.9	-	42.7%	15.3%	39537	118983	-	5184	-
	GSDT [75]	73.2	66.5	-	41.7%	17.5%	26397	120666	-	3891	-
	FairMOT [5]	73.7	72.3	59.3	43.2%	17.3%	27507	117477	58.0	3303	8073
	CSTrack [76]	74.9	72.6	59.3	41.5%	17.5%	23847	114303	57.9	3567	7668
	SGT [77]	76.3	72.4	60.6	47.9%	**11.7%**	25983	102984	**58.6**	4578	6960
	STDFormer-LMPH	78.4	**73.1**	**60.9**	49.6%	12.7%	29514	**87132**	58.4	5091	5853
	STDFormer-EMPH	**78.8**	71.5	59.9	**49.7%**	13.1%	24627	89862	56.6	4998	5379
Motion-based	ByteTrack [4]	80.3	77.7	63.1	**53.2%**	14.5%	25491	**83721**	62.0	2196	2277
	OCSORT [13]	78.0	77.5	63.2	41.0%	20.9%	**15129**	107055	**63.2**	1950	**2040**
	BoT-SORT [15]	**80.6**	**79.5**	**64.6**	-	-	22524	85398	-	**1257**	-
	STDFormer-LMPH	78.4	73.1	60.9	49.6%	**12.7%**	29514	87132	58.4	5091	5853
	STDFormer-EMPH	78.8	71.5	59.9	49.7%	13.1%	24627	89862	56.6	4998	5379
Transformer-based	TransTrack [34]	75.2	63.5	54.1	**55.3%**	10.2%	50517	**86442**	47.6	3603	4872
	TrackFormer [37]	74.1	68.0	57.3	47.3%	10.4%	34602	108777	54.1	2829	4221
	TransCenter [35]	73.2	62.2	54.5	40.8%	18.5%	23112	123738	49.7	4614	9519
	MOTR [38]	71.9	68.4	57.2	38.9%	24.1%	21123	135561	55.8	**2115**	**3897**
	MO3TR-PIQ [39]	77.6	72.9	60.3	-	-	**21045**	102531	-	2847	-
	MeMOT [40]	72.5	69.0	56.9	43.8%	18.0%	37221	115248	55.2	2724	-
	GTR [78]	75.3	71.5	59.1	-	-	26793	109854	57.0	2859	-
	TR-MOT [36]	76.5	72.6	59.7	-	-	-	-	-	-	-
	STC [79]	75.8	70.9	59.8	53.6%	**7.6%**	44952	87039	-	4533	-
	STDFormer-LMPH	78.4	**73.1**	**60.9**	49.6%	12.7%	29514	87132	**58.4**	5091	5853
	STDFormer-EMPH	**78.8**	71.5	59.9	49.7%	13.1%	24627	89862	56.6	4998	5379
Contrastive learning-based	QDTrack [49]	68.7	66.3	53.9	40.6%	21.9%	26589	146643	52.7	3378	8091
	Semi-TCL [48]	73.3	**73.2**	59.8	41.3%	18.7%	**22944**	124980	**59.4**	**2790**	8010
	STDFormer-LMPH	78.4	73.1	**60.9**	49.6%	**12.7%**	29514	**87132**	58.4	5091	5853
	STDFormer-EMPH	**78.8**	71.5	59.9	**49.7%**	13.1%	24627	89862	56.6	4998	**5379**

- Frag(↑): The total number of times a trajectory is fragmented.

(↑) means that the higher the score is, the better the performance. (↓) means that the lower the score is, the better the performance.

B. Implementation Details

We implemented our proposed method in PyTorch and trained it on a server equipped with 4 NVIDIA TITAN Xp GPUs, Intel(R) Core(TM) i7-6800K CPU, and 32 GB memory. For fair comparison and high-speed inference, we inferred on a single Tesla V100 GPU and a single Tesla A100 GPU, which are widely used in baseline methods [4], [34], [38], [40]. On the MOT17 test set, we achieved approximate real-time tracking on a single Tesla V100 GPU with 24 FPS running speed and 2860 MB GPU memory. We achieved up to 28 FPS running speed on a single Tesla A100 GPU. For detection, we adopted YOLOX-X [51] as the detector and the input image size was 1440×800. YOLOX-X is widely used by several recent state-of-the-art MOT algorithms based on the tracking-by-detection paradigm [4], [13], [14], [15] because of its balance between accuracy and speed. The training and inference of the detector stand on the giants' shoulders, which exactly adhere to the strategy of [14]. Next, we focus on the implementation details of STDFormer.

1) Training Schemes: We trained STDFormer using AdamW optimizer [69] with an initial learning rate of 10^{-3} and weight decay of 10^{-1}. For better optimization results, we employed a cosine annealing scheduler [70] with warm-up and restart for AdamW [69]. Due to the sparseness and scale difference of these three datasets, we set different training epochs and batch sizes. For MOT17, the batch size was set to 128 and the training epoch was set to 1500 with a total training time of $9h$. For MOT20, the batch size was set to 64 and the training epoch was set to 750 with a total training time of $26h$. For DanceTrack, the batch size was set to 512 and the training epoch was set to 1270 with a total training time of $35h$.

2) Hyperparameter Setting: In this work, we set $k = 15$, $r = 2500$, $d = 128$ and $L = 3$ for model building. We let $\theta_{det} = 0.6$, $\lambda_{cdist} = 0.1$, $\lambda_{giou} = 2.0$, $\lambda_{aff} = 2.0$ and $x = 50$ for data association. In addition, the hyperparameters for the construction of the box loss were: $\lambda_{l1} = 5$ and $\lambda_{iou} = 2$. The hyperparameters for the construction of the affinity loss were: $\alpha = 0.25$ and $\gamma = 2.0$. The hyperparameters for the construction of the total loss were: $w_1 = 1.0$ and $w_2 = 1.0$. The temperature hyperparameters for *dual-softmax* was: $\tau = 1000$. The effect of the hyperparameter k, L and x were conducted in the ablative studies.

C. Quantitative Results

We compared the proposed method with several state-of-the-art methods on the MOT17, MOT20 and DanceTrack test sets. Table I, Table II and Table III present an overview of the comparative results on these three datasets. To better prove the effectiveness of our method's core design, we grouped and compared the state-of-the-art methods according to the method correlation. Considering that our proposed method is

TABLE II

COMPARISON OF OUR METHOD WITH STATE-OF-THE-ART MOT ALGORITHMS UNDER THE "PRIVATE DETECTOR" PROTOCOL ON THE MOT20 TEST SET. THE BEST RESULTS OF THE FOUR GROUPS OF METHODS ARE MARKED IN YELLOW, GREEN, BLUE AND RED, RESPECTIVELY

	Methods	MOTA↑	IDF1↑	HOTA↑	MT↑	ML↓	FP↓	FN↓	AssA↑	ID Sw.↓	Frag↓
Benchmark	MLT [80]	48.9	54.6	43.2	30.9%	22.1%	45660	216803	44.1	2187	3067
	FairMOT [5]	61.8	67.3	54.6	68.8%	7.6%	103440	88901	54.7	5243	7874
	CSTrack [76]	66.6	68.6	54.0	50.4%	15.5%	25404	144358	54.0	3196	7632
	GSDT [75]	67.1	67.5	53.6	53.1%	13.2%	31913	135409	52.7	3131	9875
	SOTMOT [74]	68.6	71.4	-	64.9%	9.7%	57064	101154	-	4209	-
	SGT [77]	72.8	70.5	56.9	64.3%	12.6%	25165	112897	55.3	2649	5486
	STDFormer-LMPH	76.2	72.1	60.2	71.5%	8.6%	32336	87542	57.7	3166	5245
	STDFormer-EMPH	75.8	72.3	60.0	67.4%	12.0%	22056	100991	58.0	2329	4228
Motion-based	ByteTrack [4]	77.8	75.2	61.3	69.2%	9.5%	26249	87594	59.6	1223	1460
	OCSORT [13]	75.7	76.3	62.4	45.5%	12.9%	19067	105894	62.5	942	1086
	BoT-SORT [15]	77.7	76.3	62.6	-	-	22521	86037	-	1212	-
	STDFormer-LMPH	76.2	72.1	60.2	71.5%	8.6%	32336	87542	57.7	3166	5245
	STDFormer-EMPH	75.8	72.3	60.0	67.4%	12.0%	22056	100991	58.0	2329	4228
Transformer-based	TransTrack [34]	65.0	59.4	48.9	50.1%	13.4%	27191	150197	45.2	3608	11352
	TrackFormer [37]	68.6	65.7	54.7	53.6%	14.6%	20348	140373	53.0	1532	2474
	TransCenter [35]	58.5	49.6	43.5	48.6%	14.9%	64217	146019	37.0	4695	9581
	MO3TR-PIQ [39]	72.3	69.0	57.3	-	-	12738	128439	-	2200	-
	MeMOT [40]	63.7	66.1	54.1	57.5%	14.3%	47882	137983	55.0	1938	-
	TR-MOT [36]	67.1	59.1	50.4	-	-	-	-	-	-	-
	STC [79]	73.0	67.5	56.3	67.0%	11.8%	30215	107701	-	2011	-
	STDFormer-LMPH	76.2	72.1	60.2	71.5%	8.6%	32336	87542	57.7	3166	5245
	STDFormer-EMPH	75.8	72.3	60.0	67.4%	12.0%	22056	100991	58.0	2329	4228
Contrastive learning-based	Semi-TCL [48]	65.2	70.1	55.3	61.3%	10.5%	61209	114709	56.3	4139	8508
	STDFormer-LMPH	76.2	72.1	60.2	71.5%	8.6%	32336	87542	57.7	3166	5245
	STDFormer-EMPH	75.8	72.3	60.0	67.4%	12.0%	22056	100991	58.0	2329	4228

TABLE III

COMPARISON OF OUR METHOD WITH STATE-OF-THE-ART MOT ALGORITHMS ON THE DANCETRACK TEST SET. THE BEST RESULTS OF THE FOUR GROUPS OF METHODS ARE MARKED IN YELLOW, GREEN, BLUE AND RED, RESPECTIVELY

	Methods	MOTA↑	IDF1↑	HOTA↑	DetA↑	AssA↑
Benchmark	CenterTrack [72]	86.8	35.7	41.8	78.1	22.6
	FairMOT [5]	82.2	40.8	39.7	66.7	23.8
	TraDes [6]	86.2	41.2	43.3	74.5	25.4
	STDFormer-LMPH	91.6	55.0	52.8	80.4	34.8
	STDFormer-EMPH	91.7	60.5	57.8	80.5	41.7
Motion-based	ByteTrack [4]	89.6	53.9	47.7	71.0	32.1
	OCSORT [13]	89.4	54.2	55.1	80.3	38.0
	STDFormer-LMPH	91.6	55.0	52.8	80.4	34.8
	STDFormer-EMPH	91.7	60.5	57.8	80.5	41.7
Transformer-based	TransTrack [34]	88.4	45.2	45.5	75.9	27.5
	MOTR [38]	79.7	51.5	54.2	73.5	40.2
	STDFormer-LMPH	91.6	55.0	52.8	80.4	34.8
	STDFormer-EMPH	91.7	60.5	57.8	80.5	41.7
Contrastive learning-based	QDTrack [49]	83.0	44.8	45.7	72.1	29.2
	STDFormer-LMPH	91.6	55.0	52.8	80.4	34.8
	STDFormer-EMPH	91.7	60.5	57.8	80.5	41.7

a motion-based method that takes full advantage of the Transformer's superiority and extracts more discriminative motion features through comparative learning, we divide the compared methods into four groups: 1) benchmark methods, which are the SOTA method released on the benchmark dataset in recent years except for the methods mentioned in the following three groups; 2) motion-based tracking methods, which only rely on the motion features of objects and discard appearance cues; 3) Transformer-based tracking methods, which utilize a regular Transformer framework; and 4) contrastive learning-based tracking methods, which use contrastive learning to learn more discriminative features of the tracked objects. Note that although the methods in the first group are not closely related to our method, they are all representative and often used for comparison in MOT. To prove the competitiveness of our method, this group of comparison is indispensable. In addition, since DanceTrack is an MOT benchmark dataset released in 2022, there were fewer methods available for comparison.

For a more comprehensive comparison, we tested and compare two different configurations of STDFormer on each benchmark dataset: STDFormer-LMPH and STDFormer-EMPH. Their main difference was the motion prediction head. The former uses the Linear Motion Prediction Head (LMPH), while the latter uses the Exponential Motion Prediction Head (EMPH). The performance of STDFormer-LMPH and STDFormer-EMPH was comparable on MOT17 and MOT20. In terms of the three main evaluation metrics of MOTA, IDF1 and HOTA, the overall performance of STDFormer-LMPH was slightly better than that of STDFormer-EMPH. On DanceTrack, STDFormer-EMPH was significantly better than STDFormer-LMPH based on all evaluation metrics. This was mainly due to the difference in the offset distribution and the exponential adjustment factor

distribution on the three benchmark datasets. More detailed analysis is provided in Appendix A. To better compare with other methods, we chose STDFormer-LMPH as the comparison method on MOT17 and MOT20, and we chose STDFormer-EMPH as the comparison method on DanceTrack. The following will be referred to as STDFormer.

The quantitative results show that STDFormer achieved state-of-the-art performance on the DanceTrack test set, while STDFormer achieved comparable performance to that other state-of-the-art methods on the MOT17 and MOT20 test sets. The significant gains of STDFormer on DanceTrack demonstrated its superiority in handling complex nonlinear motion estimation. Furthermore, the analysis of the group comparison was as follows:

1) **Benchmark tracking:** Compared with benchmark methods on three benchmark datasets, the proposed method achieved state-of-the-art performance based on most evaluation metrics. In terms of the three main evaluation metrics of MOTA, IDF1 and HOTA, the proposed method achieved significant improvement compared with the baseline method. For example, 2.1%, 0.5% and 0.3% improvements for MOT17, 3.4%, 0.7% and 3.3% improvements for MOT20, and 4.9%, 13.8% and 9.5% improvements for DanceTrack. The better performance demonstrated the effectiveness of the proposed method in MOT.
2) **Motion-based tracking:** Compared with motion-based methods, the proposed method achieved state-of-the-art performance on DanceTrack, while achieving performance comparable to that of other state-of-the-art algorithms on MOT17 and MOT20. On DanceTrack, the proposed method achieved the best performance based on all evaluation metrics, especially in the three important association metrics of IDF1, HOTA and AssA, which were improved by 6.3%, 2.7% and 3.7%, respectively. This demonstrated that the proposed method significantly outperformed baselines on complex nonlinear object motion modeling and achieved robust association under similar appearance, occlusion, or nonlinear motion. On MOT17 and MOT20, the proposed method did not achieve state-of-the-art performance on the main evaluation metrics of MOTA, IDF1 and HOTA. However, we achieve the best or second-best performance in MT and ML. For example, on the MOT17 test set (Table I), the proposed method ranked first with an ML of 12.7% and second with an MT of 49.6%. On the MOT20 test set (Table II), the proposed method achieved the best performance in terms of both MT (71.5%) and ML (8.6%). The higher MT and lower ML value indicated that our method was better at recovering objects from occlusion or drift than filter-based methods. Note that methods in this group follow the tracking-by-detection paradigm and use the same detector [51]. In the motion part, ByteTrack uses standard Kalman filtering and trajectory interpolation while both OC-SORT and BoT-SORT modify Kalman filtering to adapt to more complex situations. Furthermore, BoT-SORT also models camera motion.
3) **Transformer-based tracking:** Compared with existing Transformer-based methods, the proposed method achieved the best tracking performance on MOT17, MOT20 and DanceTrack. On DanceTrack, STDFormer achieves 91.7 MOTA, 60.5 IDF1, 57.8 HOTA, 80.5 DetA and 41.7 AssA, surpassing the baseline methods by 3.3%, 9.0%, 3.6%, 4.6% and 1.5%. Unlike the baseline methods, the proposed method only encodes the object position information without considering the object appearance. This showed that the new paradigm of Transformer-based methods had a great advantage in tracking under nonlinear motion. On MOT17 and MOT20, the three main evaluation metrics of MOTA, IDF1 and HOTA and the associated accuracy metric of AssA ranked first with 78.4, 73.1, 60.9, 58.4 and 76.2, 72.1, 60.2, 57.7 respectively. In more detail, on MOT17 test set (Table I), our method increased by 0.8%, 0.2%, 0.6% and 1.4% on MOTA, IDF1, HOTA and AssA, respectively. In the crowded MOT20 test set (Table II), the benefits of the proposed method were more significant. MOTA, IDF1, HOTA and AssA increased by 3.9%, 3.1%, 2.9% and 2.7%, respectively. Comparing Table I and Table II, we found that the performance of most Transformer-based methods on the MOT20 test set decreased significantly. In contrast, the performance loss of our method on MOT20 was much lower, and some metrics (MT and ML) even increase. In addition, 71.5% MT and 8.6% ML were much better than the second-best record on the list, increasing by 14.0% and 4.8%, respectively. These results demonstrated that our method could effectively handle the occlusion problem in dense scenes. In summary, the good performance showed that the idea of using Transformer to encode low-dimensional position information for multi-object tracking was effective. Note that since Transformer has been applied in MOT in the past two years, there have been relatively few related studies. To make a full comparison, we only paid attention to whether the Transformer architecture is applied in the pipeline when selecting the methods, regardless of whether the tracking paradigm, detector or association strategy was closely related to the proposed method and whether the Transformer architecture was applied in the detection part or in the association part or even in both parts. In this work, one of our motivations was to explore a new Transformer application paradigm in MOT, trying to achieve better tracking performance with fewer computing and storage resources, so the comparison of this group of methods is meaningful. In addition, the selected methods were all published in the past two years and are comparable to a certain extent.
4) **Contrastive learning-based tracking:** Compared with some existing contrastive learning-based studies, the proposed method outperformed the baseline by a large margin on MOT17, MOT20 and DanceTrack. On DanceTrack, our method improved MOTA, IDF1, HOTA, DetA and AssA by 8.7%, 15.7%, 12.1%, 8.4% and 12.5%, respectively, compared with the

baseline method. The large performance advantage demonstrated that our contrastive learning strategy could learn more discriminative feature representations for complex nonlinear object motion tracking. On MOT17, our method achieved state-of-the-art performance with 78.4 MOTA, 60.9 HOTA, 49.6% MT, 12.7% ML, 87132 FN and 25917 Frag. We achieved the second-best values in terms of 73.1 IDF1 and 58.4 AssA, slightly lower than the performance of Semi-TCL [48]. On MOT20, our method outperformed other methods based on all proposed metrics. Furthermore, like the comparison results of Transformer-based methods, our method did not fluctuate much in tracking performance on MOT17 and MOT20 compared to other contrastive learning-based methods. This indicated that our contrastive learning strategy could learn more reliable feature representations to keep tracking or recover lost objects from occlusion or drift. Note that the contrastive learning-based baseline methods adopt selection criteria similar to the Transformer-based baseline methods.

D. Qualitative Analysis

1) Visualization: To more intuitively show that STDFormer can achieve more robust association and more stable tracking than the baseline mentioned in IV-C under similar appearance, occlusion and nonlinear motion, we provide some visualization results of difficult cases on DanceTrack that STDFormer was able to handle but ByteTrack was not (Figure 7). Specifically, we selected samples from diverse scenes, including pop dance, gymnastics and street dance. Objects in pop dance videos (dancetrack0013, dancetrack0017) have frequent crossover. Objects in gymnastics videos (dancetrack0054, dancetrack0059) exhibit diverse body gestures, frequent pose variation and complicated motion patterns. Street dance videos (dancetrack0084, dancetrack0093) present difficult scenes in low lighting apart from frequently occluded objects. Additionally, the objects in these videos have similar appearances due to similar or even identical clothes. As shown in Figure 7, ByteTrack caused ID switching due to object occlusion or complex nonlinear motion, especially the object with id 918 on dancetrack0093 (Figure 7(k)) experienced multiple ID switches. In contrast, STDFormer did not exhibit any identity conversion and effectively preserved the identity. This demonstrated that the proposed method could effectively improve multi-object tracking under occlusion and nonlinear motion situations.

In Figure 8, we also show the tracking results of Bytetrack and the proposed method on MOT17. Although STDFormer did not achieve superior tracking performance over motion-based methods on MOT17, the visualization results showed that it still had an advantage in solving the occlusion problem.

2) Limitations: STDFormer has several limitations, which are the problems we will focus on in future work:

1) **Linear motion noise/Detector noise:** We focus on improving multi-object tracking under occlusion and nonlinear motion and pay insufficient attention to the noise interference existing in linear motion prediction. Affected by detection noise, STDFormer easily interprets bad detection results as nonlinear motion signals and accumulates errors. We believe that it is necessary to explore a reasonable correction mechanism for STDFormer to address the incorrect values in the prediction process and enhance the smoothness of the trajectory.
2) **Tracking in dynamic scenes:** Our method only models the object motion without considering the influence of coordinate system transformation caused by camera motion in the dynamic scene (e.g., MOT17-06, MOT17-12 and MOT17-14). Motion prediction produces large deviations in dynamic scenes. In particular, because the camera motion is ignored, it is easy to misjudge the object leaving the scene as still remaining in the current scene and make the wrong association. Therefore, in the face of multi-object tracking in dynamic scenes, STDFormer needs to model camera motion additionally.
3) **Tracking at the edge:** Compared with the objects in the center of the scene, the object motion modeling at the edge is more complex. Objects at edges away from the camera direction are small and dense. Only relying on the center point distance and GIoU matching of the object bounding boxes for association is prone to ID switching. In addition, due to the small motion displacement of the objects at the edge, the object after the ID switching may continue to maintain the wrong ID in the future. False tracking will interfere with STDFormer's judgment of the object's motion intention, especially ID switch with a object wandering at the edge or a new object entering the scene. It is difficult to judge whether the object will continue to leave the scene, stop at the edge or even re-enter the scene according to the trajectory trend, so the wrong tracking is difficult to be corrected. On the other hand, for the object close to the camera direction and gradually away from the edge of the scene, its motion offset is large. As mentioned in Appendix A, the object motion offset at some edges on MOT17 and MOT20 can reach tens or even hundreds (e.g., MOT17-08, MOT20-06 and MOT20-08), which is highly volatile and difficult to predict compared with the offset at the center of the scene. These are issues with motion-based methods that should be studied in future work.

In summary, the quantitative experiments and visualization results on MOT17, MOT20 and DanceTrack demonstrate that STDFormer is an effective and powerful tracker. It focuses on improving the nonlinear motion modeling of objects to mine object motion intention and decision information to achieve robust association under similar appearance, occlusion, and nonlinear motion. Meanwhile, STDFormer achieves state-of-the-art performance compared to Transformer-based and contrastive learning-based methods. This points to a new feasible direction for the application of Transformer and contrastive learning in MOT.

(a) ByteTrack: dancetrack0013 (b) STDFormer: dancetrack0013

(c) ByteTrack: dancetrack0017 (d) STDFormer: dancetrack0017

(e) ByteTrack: dancetrack0054 (f) STDFormer: dancetrack0054

(g) ByteTrack: dancetrack0059 (h) STDFormer: dancetrack0059

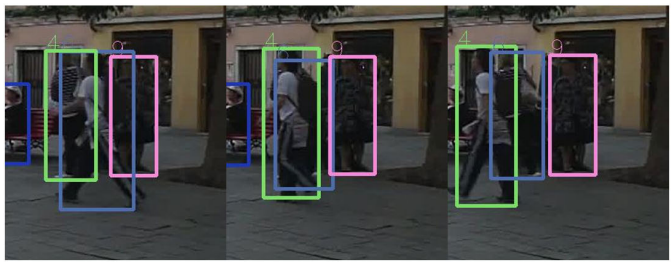
(i) ByteTrack: dancetrack0084 (j) STDFormer: dancetrack0084

(k) ByteTrack: dancetrack0093 (l) STDFormer: dancetrack0093

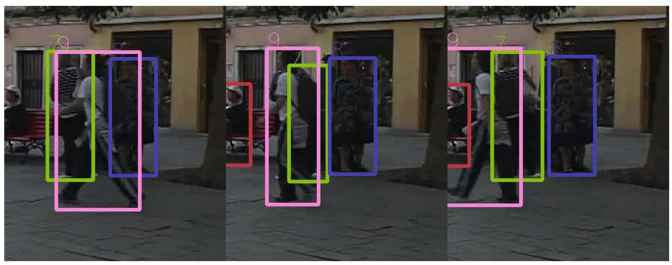
Fig. 7. Visualization results of STDFormer and ByteTrack on DanceTrack. ByteTrack leads to ID switch due to object occlusion or complex nonlinear motion, but STDFormer effectively preserves the identity. To be precise, the problem occurs with ByteTrack's objects: (a) ID switch between #82 and #86; (c) ID switch between #97 and #101; (e) ID switch between #535 and #537; (g) ID switch between #576 and #579; (i) ID switch between #692 and #693; (k) ID switch multiple times between #917 and #918. We pick samples from different scenes, including pop dance, gymnastics, and pop dance.

E. Ablative Studies

In this subsection, we evaluate different components of STDFormer on the MOT17 validation set using private detection and show the individual contributions of key components and strategies to facilitate tracking performance. Note that to optimize the association performance of STDFormer, we use the three main association metrics of HOTA, IDF1 and AssA as the judgment basis in the following



(a) ByteTrack: MOT17-01



(b) STDFormer: MOT17-01

Fig. 8. Visualization results of STDFormer and ByteTrack on MOT17-01. ByteTrack switches IDs between #4 and #6 due to object occlusion, while STDFormer effectively preserves identities based on motion trends.

TABLE IV

COMPARISONS ON DIFFERENT LENGTH OF HISTORICAL MOTION MEMORY

k	MOTA↑	IDF1↑	HOTA↑	AssA↑	ID Sw.↓	Time(ms)↓
5	89.07	82.793	75.544	72.748	433	**11.34**
10	88.647	82.637	75.602	73.02	462	11.52
15	89.091	**83.787**	**76.257**	**73.975**	430	12.14
20	88.746	82.379	75.35	72.485	469	12.38
25	88.842	82.744	75.228	72.176	468	12.69
30	**89.297**	82.947	75.474	72.603	**401**	13.12

experiments to determine the best parameter values and component design.

1) Effect of Historical Motion Memory Length: STDFormer uses long-range motion information of tracked objects to mine their behavioral intent. Too long trajectory information may have redundancy or noise effects, while too short trajectory information may not be enough to reflect the potential motion awareness of the objects. To choose an appropriate length of temporal information, we use trajectory histories of different lengths to train the model and infer on a single V100 GPU. Table IV shows the effect of different track history lengths k from 5 to 30 on tracking.

The results support our analysis that raising k can encourage association performance until the bottleneck is encountered. According to our experimental results, HOTA, IDF1, and AssA performed the best when k was increased to 15. This can be explained by the fact that adding historical motion information can alleviate the influence of short-term motion noise on motion direction estimation. However, continuing to increase k above 15 restrains association performance. Long-range movement behavior causes misjudgment of the current movement intention. We also focused on the effect of increasing history motion length on inference time in addition to association performance. As k varies from 5 to 30, the association time increased from 11.34 ms to 13.12 ms and the running speed of STDFormer decreased from 24.8 FPS to 23.7 FPS with detection time around 29 ms. As can be seen, increasing the history motion length had a negligible impact on the overall inference time. Therefore, we mainly considered the association performance. The results showed that our method had the best association performance when k=15, so we finally chose the historical motion memory length of 15 frames. This setting improved by 0.84%, 0.66% and 0.955% over the second place in HOTA, IDF1 and AssA, respectively. Moreover, the association performance changed sharply in the neighborhood of the optimal value of k, and the remaining values of k changed gently for the association performance of the proposed method, which further showed that 15 frames was a very suitable historical motion memory length.

TABLE V

COMPARISONS WITH AND WITHOUT PRIOR DETECTION

prior detection	MOTA↑	IDF1↑	HOTA↑	AssA↑	ID Sw.↓
w/o	**89.306**	82.011	74.854	71.285	421
w/ high score	89.091	**83.787**	**76.257**	**73.975**	430
w/ all	82.017	72.854	70.492	65.506	**336**

2) Effect of Prior Detection: We tried to constrain the range of motion prediction by incorporating the high score detection boxes of the current frame in the process of modeling object motion. As shown in Table V, we verified the effectiveness of prior detection by adding or not adding a detection branch. In addition, we explored the choice of prior detection. **"w/o"** means that no detection information was input to STDFormer. **"w/ high score"** indicates that only high score detections were input to STDFormer. **"w/ all"** indicates that both high score detection boxes and low score detection boxes were input to STDFormer.

We found that STDFormer achieved the best association performance by adding high score detection boxes in the process of predicting object motion, especially in the three main association evaluation metrics of HOTA, IDF1 and AssA. It shows that adding high confidence prior information imposes effective constraints on motion estimation, thereby improving the association accuracy of STDFormer. On the other hand, although STDFormer had significantly better association performance after adding prior information, its detection metric MOTA was lower than the setting without prior detection. This is because noise and false detection boxes in high score detection boxes led to a decrease in localization accuracy and an increase in FN and FP. In addition, there were more false detection boxes and noise in low score detection boxes. Using low score detection boxes as prior information not only failed to improve the association accuracy, but also greatly diminished its localization accuracy, which leads to a significant decline in STDFormer's tracking performance. Therefore, although there may be some occluded objects in low score detection boxes and considering that all detection boxes can help reduce missing objects, fragmented trajectories and ID switching, we do not recommend inputting all detection boxes into STDFormer. In summary, prior detection helped to improve the performance of association, but it was limited by the impact of detection accuracy to some extent. Considering everything together, we chose to use high score detection boxes as the prior information available to STDFormer.

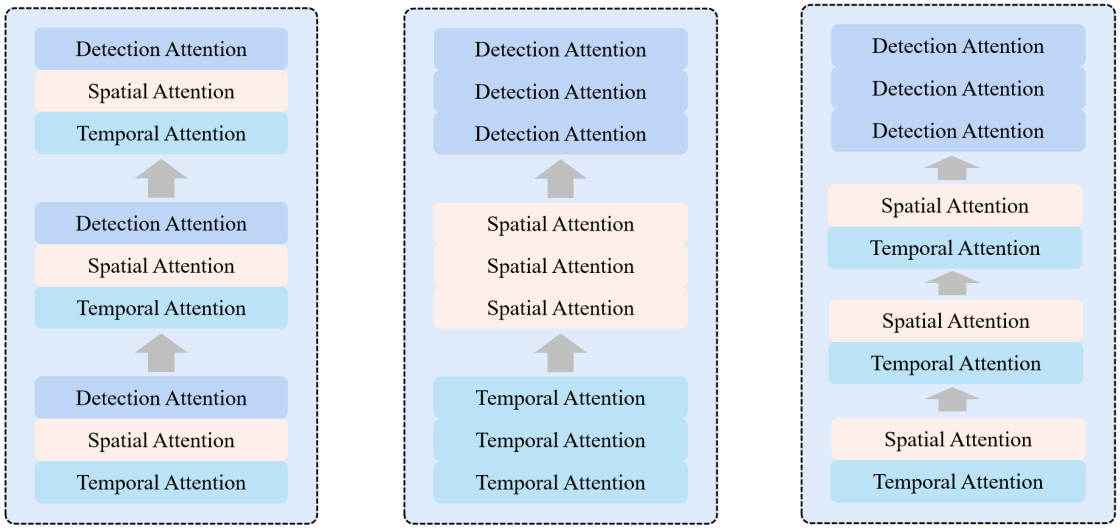


Fig. 9. Several different feature interaction designs. (a) Intra-feature interaction: Temporal features, spatial features, and detection features sequentially implement self-interaction, and there is no interaction between different types of features. (b) Joint spatial-temporal-detection interaction: Three types of features interact in each submodule. (c) Joint spatial-temporal and separate detection interaction: First realize the interaction of spatial-temporal features in each submodule, and then implement the self-interaction of detection separately.

TABLE VI
COMPARISONS ON DIFFERENT FEATURE INTERACTION MODE

interaction mode	MOTA↑	IDF1↑	HOTA↑	AssA↑	ID Sw.↓
a	87.222	75.585	71.617	66.263	715
b	88.968	82.029	74.783	71.423	489
c	**89.091**	**83.787**	**76.257**	**73.975**	**430**

TABLE VII
COMPARISONS ON DIFFERENT MATCHING STRATEGIES

matching strategies	MOTA↑	IDF1↑	HOTA↑	AssA↑	ID Sw.↓
$P_{softmax}$	88.768	80.564	74.089	70.287	507
P_{dsl}	88.851	81.457	74.272	70.527	494
Ours	**89.091**	**83.787**	**76.257**	**73.975**	**430**

3) Effect of Feature Interaction Mode: Feature interaction is the core module of the proposed method. The interaction patterns of temporal features, spatial features and detection features are crucial for the model to extract effective motion cues. Figure 9 shows several different feature interaction designs. *Mode a* iteratively encodes temporal features, spatial features and detection features in sequence, and there is no obvious interaction among different types of features. *Mode b* interleaves the three types of features by L times, interacting with the three features in each iteration. *Mode c* is the feature interaction method proposed by our method, which first interleaves the spatial-temporal motion features of the objects to obtain trajectory token features, and then interactively encodes the trajectory token features and detection features.

Table VI displays the comparison results of different feature interaction modes. They show that the feature interaction method adopted by STDFormer was superior in both localization and association. Compared with the split temporal attention and spatial attention in *Mode a*, the interleaved coding of spatial-temporal attention in *Mode b* and *Mode c* was more effective in mining the motion information of the objects. In addition, compared with the addition of detection information in the process of extracting trajectory motion information in *Mode b*, the use of detection information in *Mode c* to fine-tune the extracted spatial-temporal motion information was clearly better for making correct decisions.

4) Effect of Bidirectional Matching Optimization: STDFormer achieves bidirectional optimal matching of affinity matrices by introducing dual-softmax. As shown in Table VII, we demonstrate the effectiveness of our proposed bidirectional optimization method by comparing three different matching strategies. $P_{softmax}$ indicates performing single-dimensional softmax matching processing on the affinity matrix. The calculation process is as follows:

$$P_{softmax} = softmax(A^t_{N:M}/\tau, dim = 2) \tag{23}$$

P_{dsl} indicates that the bidirectional softmax matching process in LoFTR is used for the affinity matrix. The calculation process is as follows:

$$\begin{aligned} P_{dsl} = &\, softmax(A^t_{N:M}/\tau, dim = 1) \\ &\cdot softmax(A^t_{N:M}/\tau, dim = 2) \end{aligned} \tag{24}$$

Ours is the bidirectional softmax matching strategy we applied, and the calculation process is shown in Equation Equations 14 and 15.

The experimental results demonstrated that STDFormer's bidirectional matching method provided the best optimization effect. The comparative analysis of the three matching strategies not only showed that bidirectional matching was better than single-directional matching but also demonstrated that the sequential softmax was better than the parallel softmax optimization in two dimensions of the affinity matrix.

5) Effect of the Contrastive Learning Task: To learn the accurate motion feature representation of the trajectory prediction branch, we introduce an auxiliary task based on contrast learning. Table VIII shows the effects of adding and not adding auxiliary tasks for model learning.

TABLE VIII
COMPARISONS WITH AND WITHOUT THE CONTRASTIVE LEARNING TASK

contrastive learning	MOTA↑	IDF1↑	HOTA↑	AssA↑	ID Sw.↓
-	88.733	80.184	74.052	69.966	496
✓	**89.091**	**83.787**	**76.257**	**73.975**	**430**

TABLE IX
COMPARISONS ON DIFFERENT ASSOCIATION STRATEGIES

association strategies		MOTA↑	IDF1↑	HOTA↑	AssA↑	ID Sw.↓
IAC	BJF					
-	-	88.76	81.364	74.485	70.825	483
✓	-	89.083	83.494	76.082	73.624	446
-	✓	88.786	80.951	74.338	70.634	476
✓	✓	**89.091**	**83.787**	**76.257**	**73.975**	**430**

TABLE X
COMPARISONS ON DIFFERENT BOUNDARY JUDGMENT THRESHOLDS

x	MOTA↑	IDF1↑	HOTA↑	AssA↑	ID Sw.↓
0	89.083	83.494	76.082	73.624	446
5	89.1	83.671	76.141	73.756	431
10	89.096	83.674	76.13	73.739	430
20	89.094	83.749	76.205	73.869	428
30	**89.102**	83.756	76.211	73.877	**427**
40	89.085	83.761	76.208	73.878	431
50	89.091	**83.787**	**76.257**	**73.975**	430
60	89.094	83.747	76.229	73.919	431

The results were consistent with our previous analysis. The addition of auxiliary tasks improved the MOTA, IDF1 and HOTA metrics, greatly improving the association accuracy and localization accuracy of the model. We also verified that the model learned a more accurate motion feature expression with the constraints of contrastive learning.

6) Effect of Association Strategies: The proposed stepwise association strategy has two novel designs compared to conventional association methods:

1) Incorporating Affinity Cost (IAC): Compared with existing motion association methods, in addition to the position distance and/or detection box overlap ratio, we also use the affinity cost of detection feature representation and motion feature representation as one of the assignment criteria.
2) Boundary Judgment Filtering (BJF): It is used to distinguish whether the disappearing target is temporarily lost or the target has left the current screen effectively.

As shown in Table IX, we separately verified the individual contributions of each component by different combinations of association strategies. The results showed that both of our proposed association strategies were effective in promoting the association accuracy of tracking, especially the IAC strategy.

Furthermore, we also explored the optimal boundary judgment threshold x through experiments. Table X showed the effect of different boundary judgment thresholds from 0 to 60. The results suggested that boundary judgment filtering enabled our method to obtain the highest HOTA, IDF1 and AssA when $x = 50$. Therefore, we finally set the boundary judgment threshold to 50. As can be seen, this setting had little difference in association metric compared with the boundary judgment threshold belonging to the range of 20 to 60 and was superior to no boundary judgment or when the border judgment range was close to 0 pixels. For example, it improved HOTA, IDF1 and AssA by 0.293%, 0.175% and 0.351% respectively compared with the boundary judgment threshold of 0.

V. CONCLUSION

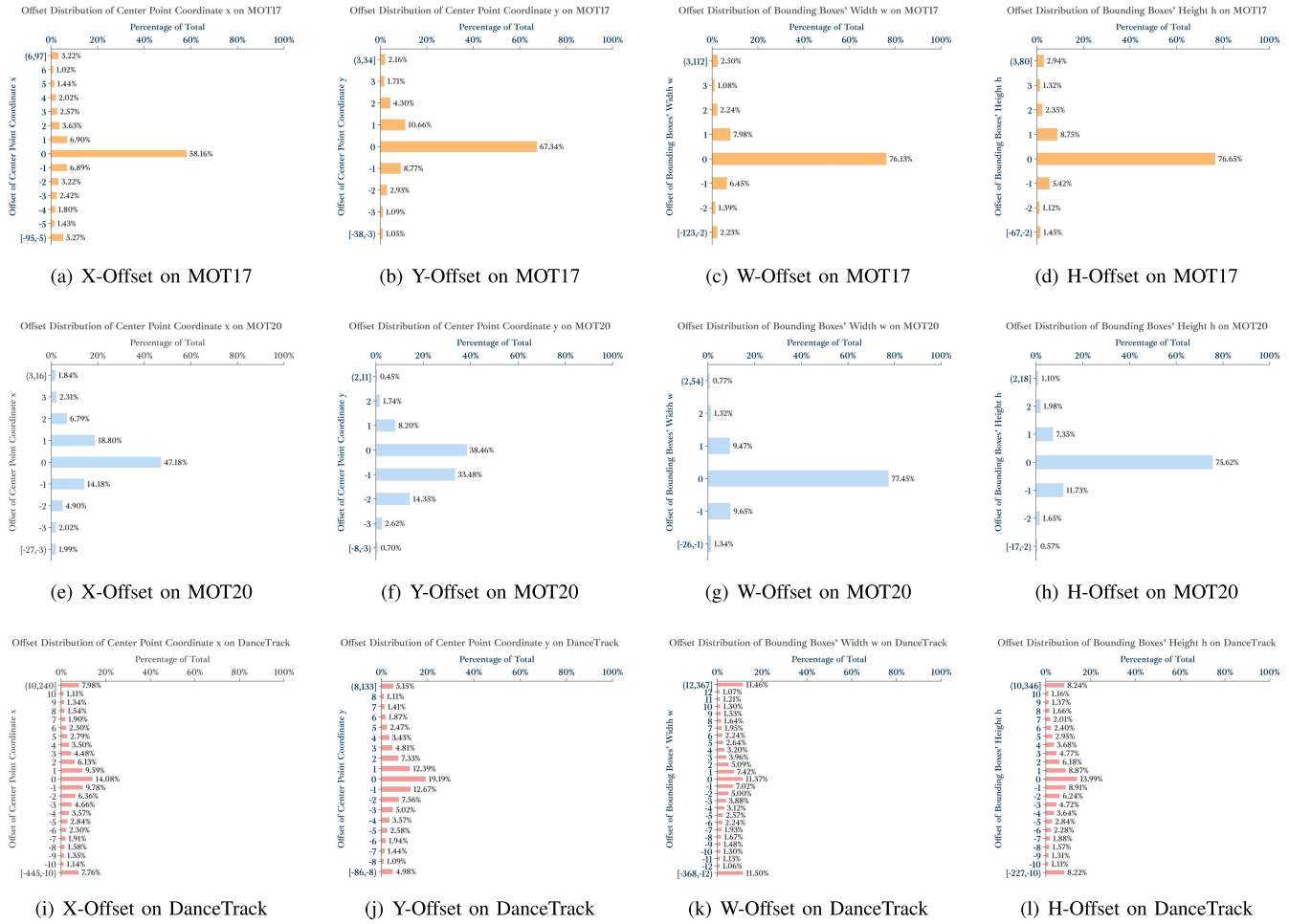
In this paper, we proposed STDFormer for online multi-object tracking by jointly performing motion estimation and data association. STDFormer mines motion cues contained in temporal motion and spatial interaction of targets by the attention mechanism of Transformer. We also introduced detection constraints considering prior knowledge. STDFormer achieves switching between temporal attention and spatial attention/detection attention by representing the tracked targets as the embeddings of dynamically updated and aggregated temporal information. To learn accurate motion feature representations, we introduced an auxiliary task based on contrastive learning to mitigate the influence of detection noise. In addition, we employed bidirectional matching to improve the one-to-many/many-to-one matching problem of single-directional matching. Evaluation on the MOTChallenge benchmark datasets demonstrated the superiority of our proposed method. The proposed method achieved significant improvement over existing Transformer-based and contrastive learning-based methods, although it still has some limitations in simple motion modes and dynamic scenes. In future work, we will conduct in-depth research to empower the STDFormer model by introducing camera motion information.

APPENDIX A
OBJECT MOTION ANALYSIS ON BENCHMARK DATASETS

The convergence speed and performance of the model are closely related to the data distribution. To make STDFormer converge quickly and locate the tracked objects' positions accurately, we analyzed the object motion on different benchmark datasets.

We reported the offset distribution of tracked objects' bounding boxes between adjacent frames on the MOT17, MOT20 and DanceTrack training sets in Figure 10, specifically including the offset distribution of object center point coordinates, bounding box height and width. The variance of the offset distribution on DanceTrack was large, while the variance of the offset distribution on MOT17 and MOT20 was small. We conducted an in-depth analysis to assess this.

MOT17 and MOT20. Except for the center point coordinate x, the offset value in the interval $[-2,2]$ accounted for more than 90% of MOT17. Due to the dense scene and overhead perspective, the offset on MOT20 was smaller, most of which was in the $[-1,1]$ interval. Furthermore, the percentages of 0 offsets far exceeded the others on both MOT17 and MOT20. This is because the video frame rate of MOT17 and MOT20 is high and the pedestrian motion pattern is simple in static scenes, so the percentage of no obvious change in position between adjacent frames is large. At the same time, there were also some offsets as high as dozens or even hundreds. These large offsets were mostly distributed in dynamic scenes with camera motion on MOT17.



(a) X-Offset on MOT17 (b) Y-Offset on MOT17 (c) W-Offset on MOT17 (d) H-Offset on MOT17

(e) X-Offset on MOT20 (f) Y-Offset on MOT20 (g) W-Offset on MOT20 (h) H-Offset on MOT20

(i) X-Offset on DanceTrack (j) Y-Offset on DanceTrack (k) W-Offset on DanceTrack (l) H-Offset on DanceTrack

Fig. 10. Percentage bar graph of Offset on benchmark datasets. For visualization purposes, we do not show the offset percentage bars below 1% separately. (a)-(d): The percentages of the center point coordinate x, center point coordinate y, bounding box width w, and bounding box height h of the target in adjacent frames on the MOT17 training set are displayed in turn; (e)-(h): The percentages of the center point coordinate x, center point coordinate y, bounding box width w, and bounding box height h of the target in adjacent frames on the MOT20 training set are displayed in turn; (i)-(l): The percentages of the center point coordinate x, center point coordinate y, bounding box width w, and bounding box height h of the target in adjacent frames on the DanceTrack training set are displayed in turn.

A small number of large offsets occurred in the static scenes on MOT17 and MOT20, which usually appeared when pedestrians approached the camera and left the scene. Overall, object motion in MOT17 and MOT20 scenes was relatively simple, and the main challenges were camera motion in dynamic scenes and object motion at the edge of the scene.

DanceTrack. Compared with MOT17 and MOT20, the offset on DanceTrack was larger, mostly in the [-10,10] interval. The percentage of large offsets with an absolute offset value above 10 cannot be ignored, and some even reached 300 or even 400, while the 0 offsets were much smaller than those on MOT17 and MOT20. In addition, the continuity of the offset was not strong. For example, the offset was consistently 0 and suddenly changed to 30 at a certain point in time. This is because DanceTrack is full of dance videos and the target movement range is large and the movement pattern is nonlinear. The object motion of DanceTrack is more complex, and the main challenge was more abrupt motion and nonlinear motion.

According to the analysis of Figure 10, it is feasible to directly predict the offset of object motion on MOT17 and MOT20. However, it is difficult to directly predict the offset on DanceTrack because the model did not easily converge. Therefore, we considered another commonly used method of regressing the bounding box based on the candidate box. Specifically, instead of directly outputting the offsets of the two bounding boxes, we obtained the bounding box of the current frame by predicting an exponential adjustment factor to scale the bounding box of the previous frame. This exponential adjustment factor is calculated as follows:

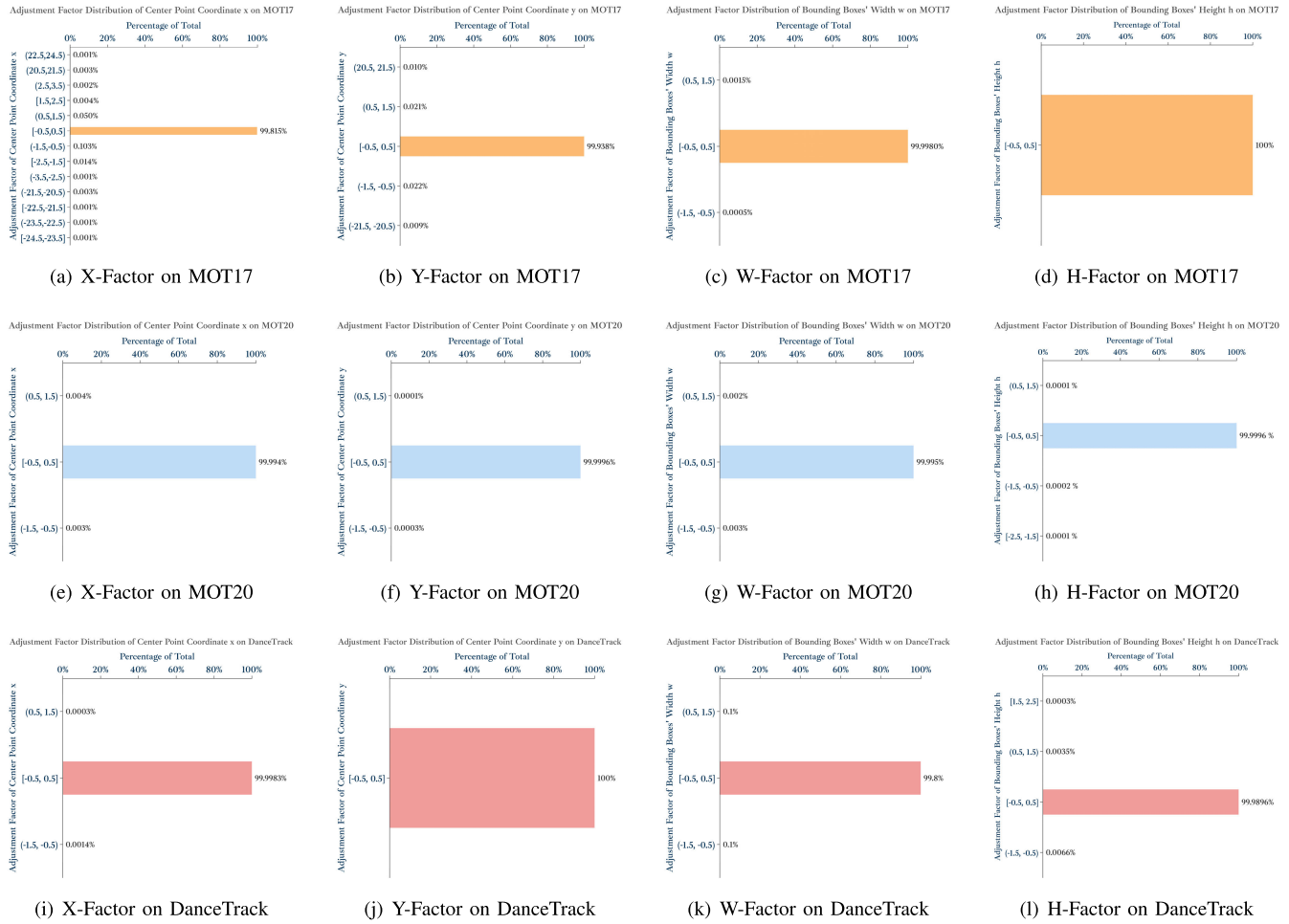
$$\zeta_x = \log(\frac{x_t}{x_{t-1}}) \tag{25}$$

$$\zeta_y = \log(\frac{y_t}{y_{t-1}}) \tag{26}$$

$$\zeta_w = \log(\frac{w_t}{w_{t-1}}) \tag{27}$$

$$\zeta_h = \log(\frac{h_t}{h_{t-1}}) \tag{28}$$

Similarly, we report the exponential adjustment factor distribution of tracked objects' bounding boxes between adjacent frames on the MOT17, MOT20 and DanceTrack training sets in Figure 11, specifically including the exponential adjustment factor distribution of object center point coordinates, bounding



(a) X-Factor on MOT17 (b) Y-Factor on MOT17 (c) W-Factor on MOT17 (d) H-Factor on MOT17

(e) X-Factor on MOT20 (f) Y-Factor on MOT20 (g) W-Factor on MOT20 (h) H-Factor on MOT20

(i) X-Factor on DanceTrack (j) Y-Factor on DanceTrack (k) W-Factor on DanceTrack (l) H-Factor on DanceTrack

Fig. 11. Percentage bar graph of Exponential Adjustment Factor on benchmark datasets. (a)-(d): The percentages of the center point coordinate x, center point coordinate y, bounding box width w, and bounding box height h of the target in adjacent frames on the MOT17 training set are displayed in turn; (e)-(h): The percentages of the center point coordinate x, center point coordinate y, bounding box width w, and bounding box height h of the target in adjacent frames on the MOT20 training set are displayed in turn; (i)-(l): The percentages of the center point coordinate x, center point coordinate y, bounding box width w, and bounding box height h of the target in adjacent frames on the DanceTrack training set are displayed in turn.

box height and width. The distributions of the exponential adjustment factors on the three benchmark datasets were similar, and the variances were all extremely small. The values of the exponential adjustment factors were nearly or completely in the $[-0.5,0.5]$ interval, and the model was easy to regress.

In summary, given that the variance of the offset distribution and the variance of the exponential adjustment factor on MOT17 and MOT20 were both small, the model converged well, and the performance of the model was not much different whether it is directly predicting the offset or the exponential adjustment factor. For further comparison, the performance of the model based on offset regression may be better due to the larger variance of the offset distribution and more discrimination. The variance of the offset distribution on DanceTrack was much larger than the variance of the exponential adjustment factor and the model based on the exponential adjustment factor converged better. To a certain extent, the performance of the model was also significantly better than that of the model based on offset regression. The above conclusions are supported by the quantitative results in Section IV-C. Therefore, STDFormer-LMPH is recommended when dealing with simple motion scenes, and STDFormer-EMPH is recommended when dealing with complex nonlinear motion scenes.

APPENDIX B
STDFORMER IN UNMANNED AERIAL VEHICLE VIDEOS

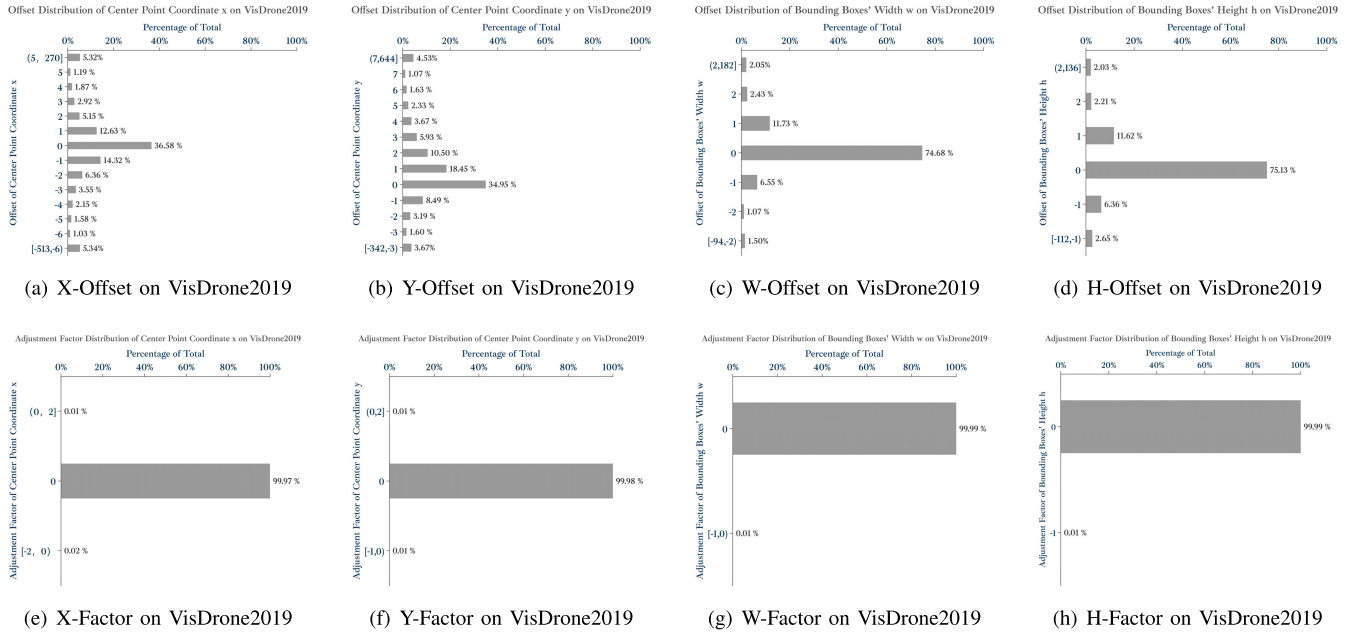
Recently, multi-object tracking techniques based on unmanned aerial vehicle (UAV) videos have received a lot of attention. Compared with the targets captured in natural scenes, the targets captured by the mobile UAV platform are smaller in scale and have higher density in crowded scenes. Due to the differences of targets in different scenes, some multi-object tracking algorithms applied to natural scenes can not be directly used in UAV videos. To demonstrate the generalizability of our method in different scenarios, we evaluate the performance on a UAV-captured MOT dataset. Specifically, we compare the proposed method with the state-of-the-art methods on the VisDrone2019 dataset.

A. VisDrone2019 Dataset

VisDrone2019 is a large-scale benchmark dataset for facilitating object detection and tracking research on UAV videos.

TABLE XI
COMPARISON OF OUR METHOD WITH STATE-OF-THE-ART MOT ALGORITHMS ON THE VISDRONE2019 TEST-DEV SET. THE BEST RESULTS OF THE METHODS ARE MARKED IN **RED**

Methods	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓	FM↓
MOTDT [81]	-0.8	68.5	21.6	87	1196	44548	185453	1437	3609
SORT [1]	14.0	73.2	38.0	506	545	80845	112954	3629	4838
IOUT [82]	28.1	74.7	38.9	467	670	36158	126549	2393	3829
GOG [83]	28.7	76.1	36.4	346	836	**17706**	144657	1387	**2237**
MOTR [38]	22.8	72.8	41.4	272	825	28407	147937	**959**	3980
TrackFormer [37]	25	73.9	30.5	385	770	25856	141526	4840	4855
UAVMOT [84]	36.1	74.2	51.0	520	574	27983	115925	2775	7396
STDFormer-EMPH	**45.9**	**77.9**	**57.1**	**684**	**538**	21288	**101506**	1440	3471



(a) X-Offset on VisDrone2019 (b) Y-Offset on VisDrone2019 (c) W-Offset on VisDrone2019 (d) H-Offset on VisDrone2019

(e) X-Factor on VisDrone2019 (f) Y-Factor on VisDrone2019 (g) W-Factor on VisDrone2019 (h) H-Factor on VisDrone2019

Fig. 12. Percentage bar graph of Offset and Percentage bar graph of Exponential Adjustment Factor on VisDrone2019 datasets. (a)-(d): The offset percentages of the center point coordinate x, center point coordinate y, bounding box width w, and bounding box height h of the target in adjacent frames on the VisDrone2019 training set are displayed in turn; (e)-(h): The exponential adjustment factor percentages of the center point coordinate x, center point coordinate y, bounding box width w, and bounding box height h of the target in adjacent frames on the VisDrone2019 training set are displayed in turn.

It was collected from 14 different cities in China by the AISKYEYE team at Lab of Machine Learning and Data Mining, Tianjin University, China. This dataset contains 96 sequences. 56 sequences are used as a training set for training the algorithms. 7 sequences are used as a validation set to verify the performance of algorithms. 17 sequences are used as a test-development set for public evaluation. 16 sequences are used as a test-challenge set for the workshop competition. The AISKYEYE team has annotated the targets with bounding boxes, categories, and tracking ids in each frame. Note that the VisDrone2019 dataset has ten categories, but we generally only consider five categories (pedestrian, car, van, truck and bus) in the evaluation of multi-object tracking methods.

B. Implementation Details

STDFormer's experimental environment on the VisDrone2019 dataset was the same as IV-B. We used the training set together with the validation set for training and evaluated our method on the VisDrone2019 test-development set using the official VisDrone MOT toolkit.

Unlike single-class tracking (pedestrian) on MOT17, MOT20 and DanceTrack datasets, tracking on VisDrone2019 dataset is a multi-class tracking task. Therefore, we did some additional designs to avoid id switching between different categories of targets. The main modifications were as follows:

- We retrained a multi-class detector. The original detector was a single-class detector, which only needed to distinguish whether the object was a pedestrian or not. The new detector was extended to 5 classes (pedestrian, car, van, truck and bus).
- We added object category information to STDFormer's input. Specifically, we converted the input (x_c, y_c, w, h) to (x_c, y_c, w, h, c), where c referred to the category id of the target.
- We added category matching cost in matching. Specifically, we filtered out some false matches by setting the cost of matching pairs that did not belong to the same class to infinity.

C. Comparison With State-of-the-Arts

To further demonstrate its effectiveness, the proposed method is compared with previous SOTA methods and benchmark methods on the VisDrone2019 dataset. As shown

in Table XI, STDFormer sets a new state-of-the-art, outperforming the baseline by a large margin in UAV videos. This shows that our method generalizes well to UAV videos. Specifically, we achieved 45.9 MOTA, 57.1 IDF1 and 77.9 MOTP on the VisDrone2019 test-development set, which surpassed the second place by 9.8%, 6.1% and 1.8% respectively. STDFormer has little difference with other methods in detection metrics like MOTP, but it has improved significantly in association metrics like IDF1 and finally achieved a large increase in tracking accuracy (MOTA). It can be seen that the proposed method can achieve a large leap in association performance on the basis of slightly improving the detection performance, which reflects that compared with other methods, STDFormer can better handle the association problem of small-scale targets in UAV videos. Note that, as shown in Figure 12, the variance of the offset distribution on VisDrone2019 is much larger than the variance of the exponential adjustment factor and the model based on the exponential adjustment factor converged better. Thus, we finally choose STDFormer-EMPH as the comparison method.

However, while our method outperforms existing methods, the tracking performance based on UAV videos is still far inferior to that in natural scenes. An important influencing factor is that the images on VisDrone2019 were all captured by a moving unmanned aerial vehicle platform, and its camera movement is more obvious than in natural scenes. In a dynamic scene, we need to consider the impact of camera motion on it in addition to its own motion when modeling the object's motion. Therefore, we believe that our method will achieve greater performance gains in multi-object tracking tasks for UAV videos after adding camera motion compensation.

REFERENCES

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

[2] H. Sheng, Y. Zhang, J. Chen, Z. Xiong, and J. Zhang, "Heterogeneous association graph fusion for target association in multiple object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3269–3280, Nov. 2019.

[3] H. Sheng et al., "Hypothesis testing based tracking with spatio-temporal joint interaction modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 2971–2983, Sep. 2020.

[4] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," 2021, *arXiv:2110.06864*.

[5] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, 2021.

[6] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, p. 12352.

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

[8] S. You, H. Yao, and C. Xu, "Multi-object tracking with spatial–temporal topology-based detector," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 3023–3035, May 2022.

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

[10] K. Chen, X. Song, and X. Ren, "Modeling social interaction and intention for pedestrian trajectory prediction," *Phys. A, Stat. Mech. Appl.*, vol. 570, May 2021, Art. no. 125790.

[11] X. Li, Y. Liu, K. Wang, Y. Yan, and F.-Y. Wang, "Multi-target tracking with trajectory prediction and re-identification," in *Proc. Chin. Automat. Congr. (CAC)*. IEEE, 2019, pp. 5028–5033.

[12] X. Weng, Y. Yuan, and K. Kitani, "PTP: Parallelized tracking and prediction with graph neural networks and diversity sampling," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4640–4647, Jul. 2021.

[13] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric SORT: Rethinking SORT for robust multi-object tracking," 2022, *arXiv:2203.14360*.

[14] Y. Du et al., "StrongSORT: Make DeepSORT great again," 2022, *arXiv:2202.13514*.

[15] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," 2022, *arXiv:2206.14651*.

[16] H. Nodehi and A. Shahbahrami, "Multi-metric re-identification for online multi-person tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 1, pp. 147–159, Jan. 2021.

[17] Y. G. Lee, T. Zheng, and J. N. Hwang, "Online-learning-based human tracking across non-overlapping cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2870–2883, Oct. 2017.

[18] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, "GIAOTracker: A comprehensive framework for MCMOT with global information and optimizing strategies in VisDrone 2021," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2809–2819.

[19] 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.* Cham, Switzerland: Springer, 2019, pp. 411–423.

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

[21] X. Wan, J. Wang, and S. Zhou, "An online and flexible multi-object tracking framework using long short-term memory," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1230–1238.

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

[23] 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. (CVPR)*, Jun. 2021, p. 14329.

[24] R. E. Kalman, "Contributions to the theory of optimal control," *Boletín Sociedad Matemática*, vol. 5, no. 2, pp. 102–119, 1960.

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

[26] G. L. Smith, S. F. Schmidt, and L. A. McGee, *Application of Statistical Filter Theory to the Optimal Estimation Of Position and Velocity on Board a Circumlunar Vehicle*. Washington, DC, USA: National Aeronautics and Space Administration, 1962.

[27] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Proc. SPIE*, vol. 3068, pp. 182–193, Apr. 1997.

[28] F. Gustafsson et al., "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

[29] J. Xiang, N. Sang, J. Hou, R. Huang, and C. Gao, "Multitarget tracking using Hough forest random field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2028–2042, Nov. 2016.

[30] Z. Weng, J. Yang, Q. Zhang, and Z. Guo, "Multi-target tracking based on motion estimation and RJ-MCMC particle filter," in *Proc. Int. Conf. Inf. Sci. Syst.*, Apr. 2018, pp. 161–165.

[31] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1819, Nov. 2005.

[32] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–8.

[33] G. Wang, R. Gu, Z. Liu, W. Hu, M. Song, and J.-N. Hwang, "Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9876–9886.

[34] P. Sun et al., "TransTrack: Multiple object tracking with transformer," 2020, *arXiv:2012.15460*.

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

[36] M. Chen, Y. Liao, S. Liu, F. Wang, and J.-N. Hwang, "TR-MOT: Multi-object tracking by reference," 2022, *arXiv:2203.16621*.

[37] 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. (CVPR)*, Jun. 2022, pp. 8844–8854.

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

[39] T. Zhu et al., "Looking beyond two frames: End-to-end multi-object tracking using spatial and temporal transformers," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Oct. 10, 2022, doi: 10.1109/TPAMI.2022.3213073.

[40] J. Cai et al., "MeMOT: Multi-object tracking with memory," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8090–8100.

[41] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "TransMOT: Spatial–temporal graph transformer for multiple object tracking," 2021, *arXiv:2104.00194*.

[42] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[43] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[44] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 319–345.

[45] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, "A theoretical analysis of contrastive unsupervised representation learning," 2019, *arXiv:1902.09229*.

[46] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," 2018, *arXiv:1803.02893*.

[47] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-Gram features," 2017, *arXiv:1703.02507*.

[48] W. Li, Y. Xiong, S. Yang, M. Xu, Y. Wang, and W. Xia, "Semi-TCL: Semi-supervised track contrastive representation learning," 2021, *arXiv:2107.02396*.

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

[50] Y. Liu, Q. Yan, and A. Alahi, "Social NCE: Contrastive learning of socially-aware motion representations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, p. 15118.

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

[52] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8922–8931.

[53] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[54] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[55] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[56] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[57] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[58] X. Cheng, H. Lin, X. Wu, F. Yang, and D. Shen, "Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss," 2021, *arXiv:2109.04290*.

[59] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[60] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[61] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

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

[63] P. Dendorfer et al., "MOT20: A benchmark for multi object tracking in crowded scenes," 2020, *arXiv:2003.09003*.

[64] P. Sun et al., "DanceTrack: Multi-object tracking in uniform appearance and diverse motion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, p. 20993.

[65] P. Zhu et al., "Detection and tracking meet drones challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7380–7399, Nov. 2022.

[66] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *J. Image Video Process.*, vol. 2008, pp. 1–10, Feb. 2007.

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

[68] J. Luiten et al., "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 548–578, Oct. 2020.

[69] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[70] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

[71] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "TubeTK: Adopting tubes to track multi-object in a one-step training model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6308–6318.

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

[73] 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, Mar. 2022.

[74] L. Zheng, M. Tang, Y. Chen, G. Zhu, J. Wang, and H. Lu, "Improving multiple object tracking with single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2453–2462.

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

[76] 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.

[77] J. Hyun, M. Kang, D. Wee, and D.-Y. Yeung, "Detection recovery in online multi-object tracking with sparse graph tracker," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4850–4859.

[78] X. Zhou, T. Yin, V. Koltun, and P. Krähenbühl, "Global tracking transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8771–8780.

[79] A. Galor, R. Orfaig, and B.-Z. Bobrovsky, "Strong-TransCenter: Improved multi-object tracking based on transformers with dense representations," 2022, *arXiv:2210.13570*.

[80] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke, and Z. Xiong, "Multiplex labeling graph for near-online tracking in crowded scenes," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7892–7902, Sep. 2020.

[81] 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 (ICME)*, Jul. 2018, pp. 1–6.

[82] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[83] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. CVPR*, Jun. 2011, pp. 1201–1208.

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

Mengjie Hu received a Ph.D. degree from Beihang University, Beijing, China, in 2017. She is currently a Lecturer with the Beijing University of Posts and Telecommunications. Her current research interests include computer vision and machine learning, especially visual object tracking and 3D scene understanding.

Xiaotong Zhu received the B.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2021, where she is currently pursuing the master's degree. Her research interests include multi-object tracking and trajectory prediction.

Haotian Wang received the B.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2022, where he is currently pursuing the M.S. degree. His current research interests include object detection and object tracking, especially multi-object tracking in satellite videos.

Shixiang Cao received the Ph.D. degree in remote sensing from Beihang University, Beijing, China, in 2014. He is currently a Senior Engineer with the Beijing Institute of Space Mechanics and Electricity. His current research interests include optical detective sensor design, motion analysis, remote sensing image processing, and algorithm implementation from the computer vision community.

Chun Liu received the B.S. degree in mechanical design, manufacturing, and automation, and the M.S. degree in mechatronic engineering from China University of Geosciences, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree in measurement and control technology from the University of Kassel, Kassel, Germany, in 2014. She is currently a Lecturer with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing. Her research interests include deep learning, intelligent computation, computer vision, and data mining.

Qing Song received the Ph.D. degree from Tianjin University, Tianjin, China, in 2006. She is currently a Scientific Researcher with the Beijing University of Posts and Telecommunications (BUPT), where she is engaged in computer vision technology study. She is the Founder of the Pattern Recognition and Intelligent Vision Laboratory (PRIV). She oversees many national, provincial, and ministerial projects and enterprise cooperation projects. She has published more than 100 academic papers in international journals and conferences.