Applied Intelligence (2025) 55:33
https://doi.org/10.1007/s10489-024-05866-4

ETTrack: enhanced temporal motion predictor for multi-object tracking

Xudong Han[1] · Nobuyuki Oishi[1] · Yueying Tian[1] · Elif Ucurum[1] · Rupert Young[1] · Chris Chatwin[1] · Philip Birch[1]

Accepted: 6 November 2024 / Published online: 27 November 2024

Abstract

Many Multi-Object Tracking (MOT) approaches exploit motion information to associate all the detected objects across frames. However, traditional tracking-by-detection (TBD) methods, relying on the Kalman Filter, often work well in linear motion scenarios but struggle to accurately predict the locations of objects undergoing complex and non-linear movements. To overcome these limitations, we propose ETTrack, a novel motion prediction method with an enhanced temporal motion predictor. Specifically, the motion predictor integrates a transformer model and a Temporal Convolutional Network (TCN) to capture both long-term and short-term motion patterns, and it predicts the future motion of individual objects based on the historical motion information. Additionally, we propose a novel Momentum Correction Loss function that provides additional information regarding the motion direction of objects during training. This allows the motion predictor to rapidly adapt to sudden motion variations and more accurately predict future motion. Our experimental results demonstrate that ETTrack achieves a competitive performance compared with state-of-the-art trackers on DanceTrack and SportsMOT, scoring 56.4% and 74.4% in HOTA metrics, respectively. Our work provides a robust solution for MOT in complex dynamic environments, which enhances the non-linear motion prediction capabilities of tracking algorithms.

Keywords Multi-object tracking · Motion model · Kalman filter · Transformer · Temporal convolutional network

1 Introduction

Multi-Object Tracking (MOT) is an important technology in the field of computer vision and aims to track multiple objects in video sequences. Accurate MOT benefits downstream computer vision tasks such as action recognition, scene understanding and pose tracking. It also plays a significant role in applications, including mobile robotics [34, 44], autonomous driving [31, 41], smart devices [29]and sports analytics [33]. With recent progress in object detection, tracking-by-detection (TBD) methods [4, 5, 10, 37, 49] have become the most popular paradigm. These approaches typically comprise two subtasks: detecting objects in each frame; and associating them across various frames. The core of the TBD paradigm is data association, which requires a motion predictor to predict the motion of an object. Most of these trackers adopt the Kalman filter [16] as a motion predictor, with the assumption of constant velocity in both the prediction and filtering processes. The Kalman Filter works well when tracked objects have linear and regular movements on the pedestrian dominant dataset [25]. However, these trackers struggle to perform well in situations [9, 32] where dancers dance in all directions of the stage and players have rapid motion in sports scenes. In these cases, the Kalman Filter fails in handling non-linear motion and diverse motion. This observation motivates us to rethink current motion predictors and recognize their limitations. Then, we focus on developing

✉ Philip Birch
P.M.Birch@sussex.ac.uk

Xudong Han
xh218@sussex.ac.uk

Nobuyuki Oishi
n.oishi@sussex.ac.uk

Yueying Tian
yt322@sussex.ac.uk

Elif Ucurum
e.ucurum@sussex.ac.uk

Rupert Young
r.c.d.young@sussex.ac.uk

Chris Chatwin
c.r.chatwin@sussex.ac.uk

[1] School of Engineering and Informatic, University of Sussex, Falmer BN1 9QT, East Sussex, United Kingdom

a motion predictor that increases the accuracy of object association and tracking performance.

To overcome the limitations of the Kalman Filter, some deep-learning-based motion predictors have been applied to MOT. For example, [17, 26] adopt Recurrent Neural Networks (RNNs) to predict the object position based on the historical trajectories of objects by exploiting their sequence processing capabilities. Chaabane et al. [6] employs a Long Short-Term Memory-based (LSTM) to capture motion constraints by considering the motion information of objects as input. Xiao et al. [40] proposes a Transformer-based motion predictor to capture long-range dependences for modeling motions. However, these deep-learning-based approaches have two limitations. First, owing to their simple network structures (RNN and LSTM), they are unable to effectively handle input sequences with high variability and have difficulty modeling complex and long-range temporal dependencies [23]. Despite the strength of the transformer in capturing long-range temporal dependencies using a self-attention mechanism, the transformer-based model inherently possesses less inductive bias when compared to other methods such as Convolutional Neural Networks (CNNs) [14, 38]. In many scenarios [9, 32], especially those involving immediate, short-term motions, the available data for these motions is relatively limited. Transformers, due to their lack of inductive bias, often require large amounts of data to learn effectively. This means that with limited short-term motion data, Transformers may struggle to learn accurate motion patterns. Second, current deep-learning-based approaches only consider the historical trajectories of objects as inputs and lack the capacity to integrate reliable additional information, resulting in unreliable position prediction in complex and non-linear scenes. Nonetheless, integrating additional information is possible, such as the appearance features of objects [17], the interaction between objects [15] and camera motion [43]. However, when significant occlusions and complex actions are present in these scenes [32], the performance is compromised.

In this paper, we propose an enhanced temporal motion predictor named ETTrack that integrates a Temporal Transformer model [35] and a Temporal Convolutional Network (TCN) [2] for MOT. In MOT, where capturing immediate, short-term motion patterns is crucial and where available training data may be limited, the lack of inductive bias can negatively impact performance. To address the limitations of the transformer-based method, our approach integrates a TCN that introduces a degree of recency bias. This enables the motion predictor to capture short-term motion patterns more effectively and enhances prediction performance in scenarios with non-linear movements and high variability. Specifically, the TCN, which employs dilated causal convolution, is inherently more suitable for modeling fine-grained and short-term dependencies, particularly short-term minor changes. Leveraging the inductive biases of the TCN, our model can learn effective motion representations from limited data, overcoming the huge data requirements of Transformers in short-term motion scenarios. Furthermore, we utilize a specialized Temporal Transformer architecture that employs only the encoder component of the vanilla Transformer model. The integration of the Temporal Transformer with the TCN enables our motion predictor to capture both local and global motion patterns comprehensively. In this integrated framework, the TCN excels at capturing fine-grained motion details, while the Temporal Transformer builds upon this to comprehend broader and long-term motion trends.

In addition to the motion predictor, we propose a novel loss function called Momentum Correction Loss (MCL), which serves as a regularizer for the primary motion prediction task. During training, the MCL guides the motion predictor by encouraging the predicted motion directions to align with the actual motion directions. This alignment is particularly important in scenarios involving rapid posture changes and swift movements, where the motion direction of an object can shift significantly. Under such conditions, relying solely on past trajectories diminishes the efficacy of modeling motion information, as the motion predictor may struggle to swiftly acquire and incorporate new motion cues and make prompt adjustments. Furthermore, in MOT tasks, the position of an object is typically represented by a bounding box. Therefore, in scenarios where the posture of some objects suddenly changes [32], we consider not only the motion directions of the object's center point but also those of its four corners. By incorporating the motion directions of the bounding box corners, we capture changes in the object's scale and aspect ratio that the center point alone cannot provide. This is particularly important when an object undergoes pose variations that cause its bounding box to change size or shape. With this added motion direction information, the motion predictor can rapidly adapt to motion variations and predict future motion more accurately. The contributions of this study can be summarized as follows:

1. We propose ETTrack, an enhanced temporal motion predictor that integrates a Temporal Convolutional Network (TCN) and a Temporal Transformer model to better handle complex and non-linear object movements. It can effectively capture both short-term and long-term motion patterns, improving object tracking performance.
2. We introduce a novel Momentum Correction Loss (MCL) function that incorporates motion direction into our motion predictor during training to improve adaptation to sudden motion changes. This enhances the motion predictor's accuracy in rapid movements and posture changes.

3. We demonstrate that our proposed method achieves competitive performance on challenging datasets, such as DanceTrack [32] and SportsMOT [9], where non-linear movements, diverse poses, and severe occlusion are present. Furthermore, our method achieves comparable results on MOT17 [25].

2 Related works

2.1 Tracking-by-detection methods

As detection and re-identification techniques have advanced rapidly, tracking-by-detection methods [4, 37] have become the predominant paradigm in MOT. These methods first detect objects and then associate them using appearance and motion information. Alternatively, joint-detection-tracking methods, such as JDE [36] and FairMOT [48], incorporate the detection and ReID model for joint training, offering comparable performance with low computational costs. However, joint-detection-tracking methods may face reduced efficiency owing to conflicts between the detection and tracking optimization goals in their unified network. Additionally, ByteTrack [49] utilizes a simple yet effective data association method BYTE to significantly enhance the tracking accuracy and robustness. Tracking-by-detection methods demonstrate that a robust detector combined with a simple association approach can attain good tracking results. Therefore, we chose to follow the ByteTrack algorithm replacing the Kalman Filter [16] with a deep-learning-based motion model.

2.2 Motion models in MOT

Several mainstream MOT algorithms [4, 5, 7, 37] use motion models. Typically, SORT-series trackers [4, 5, 49] utilize the Bayesian estimation [20] as a motion model to predict the subsequent state by maximizing the posterior estimation. For instance, SORT [4] utilizes the classic Kalman Filter [16], assuming linear motion for object estimation, and the Hungarian matching algorithm [19] to match predictions and detections. OC_SORT [5] enhances robustness in handling occlusions by prioritizing object observations rather than linear state estimations, but it still suffers from long-term occlusion and struggles in recovering lost objects undergoing non-linear motion. However, as has been emphasized, Kalman Filter based methods presuppose a constant motion, which does not accurately describe the change in object positions when undergoing complex interactions within a scene. Therefore, some MOT methods introduce deep-learning-based motion models [17, 26] for non-linear motion modeling. For example, [26] presents a novel tracker based on recurrent neural networks (RNNs) for online MOT. Kesa et al. [17]proposes a joint learning architecture for improved MOT and trajectory forecasting by leveraging the capabilities of RNNs and adding additional appearance information, thereby surpassing the limitations imposed by using a traditional Kalman Filter. The DEFT algorithm [6] uses an LSTM to capture the motion constraints of objects. [40] proposes a Transformer-based motion model to capture long-range dependencies for modeling motions.

However, these current approaches lack the capacity to model more complex temporal dependencies and integrate reliable additional information, which leads to in inadequate motion prediction capabilities in complex and non-linear scenarios. Transformer-based models, while capturing long-range dependencies, lack inductive biases for local context and require more data to train the motion predictor effectively. In this paper, the proposed ETTrack integrates a TCN with a Temporal Transformer to model short-term and long-term dependencies. This approach aims to effectively captures motion patterns across different time scales and enhance the motion predictor's ability to capture fine-grained motion details.

2.3 Transformer-based methods

Since the Transformer [35] has become popular in computer vision, many methods [8, 24, 30, 46, 50] for the MOT task have been proposed to leverage its powerful attention mechanism to extract deep representations from both visual information and object trajectories. For example, TrackFormer [24] and MOTR [46] extend from Deformable DETR [53]. They utilize both track queries and standard detection queries to predict object bounding boxes and associate the same objects in subsequent frames. TransTrack [30] employs only Transformers as its feature extractor and propagates track queries once to obtain the position of objects in the subsequent frame. TransMOT [8] uses convolutional neural networks (CNNs) as a detector to extract features and employs spatial-temporal transformers to learn an affinity matrix. Recently, MOTRv2 [50] combines a separate detector with MOTR [46] to address the conflict between the detection and association. Additionally, MeMOTR [12] enhances multi-object tracking by using a long-term memory-augmented Transformer to capture long-term temporal information, improving target association.

However, despite their success, Transformer-based methods have several disadvantages. Transformer-based methods require extensive training time and computational resources due to their high, which limits their real-time capability. In contrast, the proposed method utilizes the powerful temporal dependency modeling capabilities of a Transformer to model the movement of objects. In addition, the ETTrack method relies solely on trajectory data as input, which significantly decreases the computation time required for motion model inference during the runtime.

3 Method

In this work, we propose an enhanced temporal motion predictor that effectively utilizes motion cues to track objects with complex motion patterns. Our primary objective is to achieve precise estimates of non-linear uncertainties by integrating a Temporal Transformer with a Temporal Convolutional Network (TCN) that surpasses the performance of some deep-learning-based motion models. In addition, we propose a momentum correction loss function to enhance the motion predictor by using motion direction information.

3.1 Problem formulation

The trajectory of an individual object i contains a sequence of bounding boxes $\mathbf{B} = \{b_{t_1}, b_{t_2}, \ldots, b_{t_N}\}$, where t stands for the timestamp, and N is the total number of frames. The bounding boxes are represented as b=(x, y, w, h). The aim of MOT is to assign a unique identifier to all the frame-wise bounding boxes. This assignment aims to establish a comprehensive association between all the bounding boxes.

Our goal is to create a motion predictor that predicts the locations of objects. When the historical trajectory length of objects is set to p, the historical trajectory of objects can be denoted as a sequence $\mathbf{X} \equiv \{\{X_i^k\}_{k=1}^K\}_{i=s-p}^{s-1} \equiv \{\{X_{s-p}^k\}, \{X_{s-p+1}^k\}, ..., \{X_{s-1}^k\}\} \in \mathbb{R}^{p\times 8}$, where K is the total number of objects across all frames and s is the frame index. The object k at the moment $s-1$ is denoted by $\mathbf{X}_{s-1}^k = \left(x_{s-1}^k, y_{s-1}^k, w_{s-1}^k, h_{s-1}^k, \Delta x_{s-1}^k, \Delta y_{s-1}^k, \Delta w_{s-1}^k, \Delta h_{s-1}^k\right)$, where (x_{s-1}^k, y_{s-1}^k) are the center coordinate of the corresponding bounding box, (w_{s-1}^k, h_{s-1}^k) stand for the width and height of the bounding box, respectively, and $\mathbf{V}_{s-1}^k = \left(\Delta x_{s-1}^k, \Delta y_{s-1}^k, \Delta w_{s-1}^k, \Delta h_{s-1}^k\right)$ are the offsets in the center point, width and height. These historical trajectories are fed into the motion predictor to predict the offsets $\mathbf{V}_s^k = \left(\Delta x_s^k, \Delta y_s^k, \Delta w_s^k, \Delta h_s^k\right)$ at the current moment s.

Predicting offsets (i.e., position changes) between successive frames offers two advantages for motion prediction. First, predicting position changes simplifies the overall prediction task because it involves analyzing relative movements, which typically exhibit less variation and more predictable patterns than the absolute positions. By focusing on relative movements, the complexity of the prediction process is reduced. Second, our motion predictor becomes less sensitive to specific starting points or trajectory shapes. This advantage enables the motion predictor to effectively capture the underlying movement dynamics, thereby improving its ability to generalize across different scenarios and unseen data.

To obtain the predicted position at the current moment s, the motion predictor adds bounding boxes $\{X_{s-1}^k\}_{k=1}^K$ from the last frame to the predicted offsets to generate the predicted current bounding boxes $\mathbf{X}_s^k = \{x_s^k, y_s^k, w_s^k, h_s^k\} = \{x_{s-1}^k + \Delta x_s^k, y_{s-1}^k + \Delta y_s^k, w_{s-1}^k + \Delta w_s^k, h_{s-1}^k + \Delta h_s^k\}$.

3.2 Overall framework

The overall framework of the motion predictor is illustrated in Fig. 1. Our proposed motion predictor processes historical trajectories $\mathbf{X} \in \mathbb{R}^{p\times 8}$, where p is the number of past frames. The trajectories are initially processed through a fully connected layer, which maps them into a higher-dimensional feature space $\mathbf{H} \in \mathbb{R}^{p\times c}$ to enhance representation capacity for temporal modeling, where c is the feature dimension. Next, the TCN processes $\mathbf{H}$ to extract short-term temporal dependencies and fine-grained motion details. The output of the TCN, denoted as $\mathbf{T} \in \mathbb{R}^{p\times c}$, serves as the input to the Temporal Transformer. The Transformer models long-term dependencies by employing multi-head self-attention mechanisms, capturing global motion trends across the entire sequence. At the current time step s, the motion predictor generates track predictions P_s based on the processed historical trajectories. An object detector is deployed to obtain detections D_s at time s. With the track predictions and detections, data association is accomplished using the IoU Distance and the Hungarian matching algorithm. Different numbers in the tracking results represent different object identities.

3.3 Motion predictor

The MOT task requires identification of the spatial and temporal locations of objects, specifically their trajectories. It has been demonstrated that the Temporal Transformer can capture global temporal dependencies and comprehend the overall context of motion sequences, offering a comprehensive perspective of long-range motion interactions and patterns. Moreover, the Temporal Convolutional Network (TCN) [2] has proven effective in identifying complex local temporal dependencies in motion sequences, resulting in precise analysis of short-term motion patterns. Thus, the integration of the Temporal Transformer and TCN allows the model to provide a comprehensive understanding of the motion patterns.

3.3.1 Temporal transformer

The Temporal Transformer is designed to effectively capture the long-term historical motion patterns of individual objects.

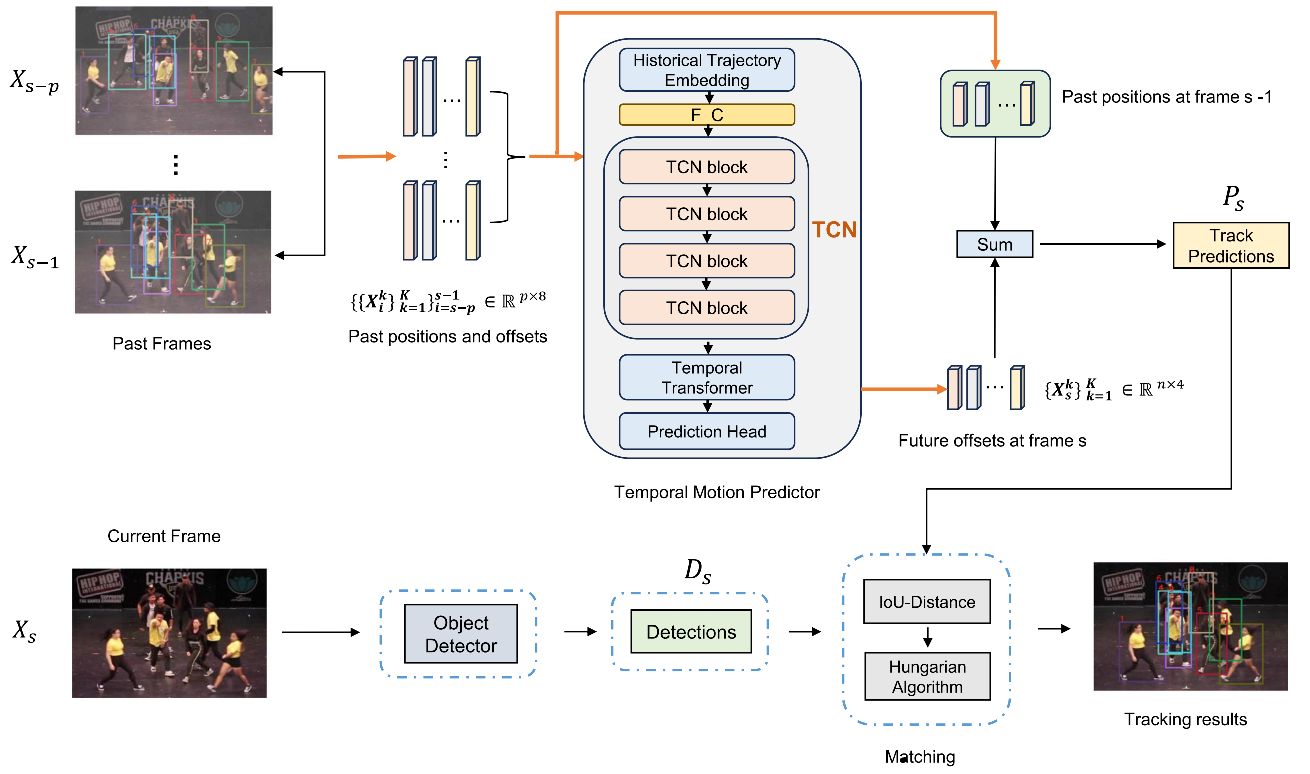

X_{s-p}
X_{s-1}
Past Frames
$\{\{X_i^k\}_{k=1}^{K}\}_{i=s-p}^{s-1} \in \mathbb{R}^{p\times 8}$
Past positions and offsets
Historical Trajectory Embedding
F C
TCN block
TCN block
TCN block
TCN block
TCN
Temporal Transformer
Prediction Head
Temporal Motion Predictor
Past positions at frame s -1
Sum
P_s
Track Predictions
$\{X_s^k\}_{k=1}^{K} \in \mathbb{R}^{n\times 4}$
Future offsets at frame s
Current Frame
X_s
Object Detector
D_s
Detections
IoU-Distance
Hungarian Algorithm
Matching
Tracking results

Fig. 1 The overall framework of ETTrack

This is achieved by utilizing a standard transformer encoder, which comprises a multi-head self-attention (MHSA) mechanism. The MHSA enables the encoder to consider various aspects of the trajectory sequence and identify the most critical features for predicting the future positions of objects. The structure of the Temporal Transformer is shown in Fig. 2.

In the Temporal Transformer, the input sequence of token T is the output of the Temporal Convolutional Network. The self-attention of the temporal transformer learns the query matrix $Q = f_Q(T)$, key matrix $K = f_K(T)$, and value matrix $V = f_V(T)$. Self-attention of a single head is calculated as:

$$Attention(Q, K, V) = \frac{softmax(QK^T)}{\sqrt{d_k}} V, \qquad (1)$$

where : $\frac{1}{\sqrt{d_k}}$ accounts for the numerical stability of self-attention. In (1), $softmax$ is the distribution function, which depends on the properties of the model. The application principles of the Temporal Transformer involve utilizing multiple attention matrices for recurrent processing, enabling more effective handling of complex temporal dependencies. This is achieved through the implementation of multi-head attention, which involves embedding the outputs of multiple self-attention mechanisms. This method enables the model to simultaneously consider information from various subspaces of representation at different positions, thereby enhancing its ability to process and understand complex information. With n heads, the multi-head attention can be represented as:

$$Multi\ \ Attention = f_h(\text{Attention}(Q_i, K_i, V_i)_{i=1}^{n}), \qquad (2)$$

where f_h is a fully connected feed-forward network. A positional encoding method is implemented to provide the Transformer encoder with positional information and to enable the attention layer to perform multi-head self-attention on the output of the TCN. Finally, the output of n heads is cascaded and fed into f_h to obtain the Transformer output.

The Temporal Transformer is an important implementation of the Transformer model for modeling object motion by processing sequential data in MOT. It enhances object association by providing precise motion predictions, facilitating the association of detections across frames, particularly when objects undergo complex or non-linear movements. In our experiments, we demonstrate that the temporal Transformer can efficiently capture the temporal interdependencies within the input trajectory sequences.

3.3.2 Temporal convolutional network

The Temporal Convolutional Network (TCN) is an innovative structure optimized for sequential data processing that effectively addresses temporal dependencies using a deep learning approach. A standard TCN is composed of multiple

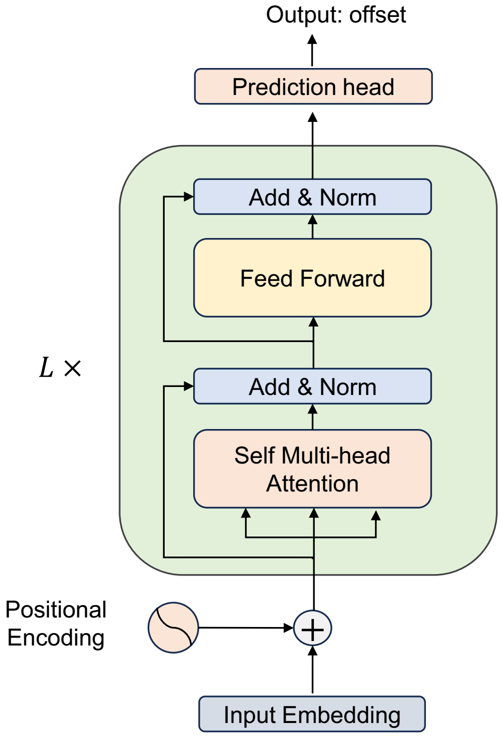


Fig. 2 The network structure of Temporal Transformer in the motion predictor

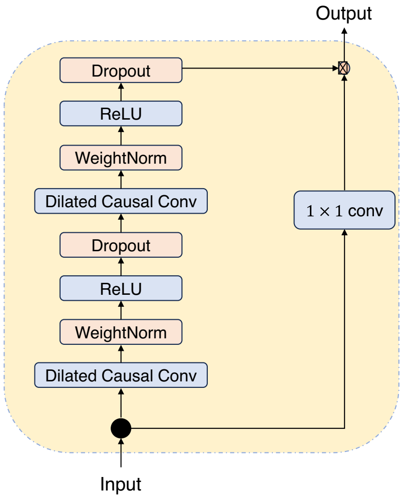


Fig. 3 Structure of the TCN block

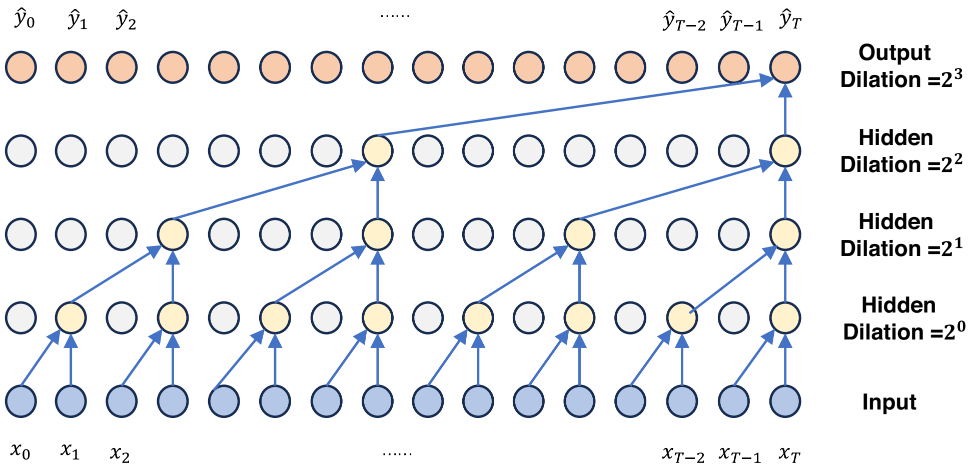
TCN blocks, each designed to capture temporal patterns and ensure robust feature representation. Details of the TCN block are shown in Fig. 3. Each TCN block consists of successive layers, beginning with a pair of causal convolutions that ensure the model's predictions depend solely on past and present information rather than future data. While causal convolutions are effective in modeling short-term dependencies, they have limitations in capturing long-term dependencies due to their limited receptive fields. To enhance the TCN's capacity to model longer temporal interdependencies, we can increase the network depth or filter size; however, this results in higher computational complexity.

To address this challenge, dilated causal convolution employs dilation factors, represented by $dil \in F$, where F is a set of dilation factors used to control the spacing between elements in the convolution operation. This allows the network to enlarge its receptive field without significantly increasing computational demands. The dilation factor, denoted by dil, can be exponentially increased, as shown in Fig. 4. Using dilated causal convolution, we can calculate the feature map f as follows:

$$f(t) = (I *_{dil} W)(t) \sum_{i=0}^{K-1} w(i) I(t - i dil), \tag{3}$$

where I is the input, W represents the filter, and K is the filter size. By stacking these convolutions, a TCN can expand its receptive field and effectively capture longer dependencies.

Fig. 4 visualization of a stack of dilated causal convolutions (K = 7, Dilation (F) = [$\mathbf{2^0}$,$\mathbf{2^1}$,$\mathbf{2^2}$,$\mathbf{2^3}$])



Following dilated causal convolution, two layers of non-linearity are introduced using the ReLU activation function. This non-linearity is crucial for the model to capture complex patterns and relationships in data. Weight normalization is added to the one-dimensional convolutions to improve the training stability and speed. In addition, a dropout block is added after each activation function. Finally, a residual connection is integrated into the layer to enhance the predictive performance of the model. Residual blocks [45] equipped with identity mapping can be denoted as:

$$T_{j+1} = T_j + \mathcal{R}(T_j, W_j), \tag{4}$$

Where T_j and T_{j+1} represent the input and output of the $(j{+}1)$th TCN block, respectively, W_j is the trainable parameter matrix of the residual blocks, and $\mathcal{R}(\cdot)$ denotes a residual function.

Although the TCN is designed to enhance the capacity to model long-term dependencies using dilated causal convolution, it is equally optimized for modeling short-term dependencies, allowing it to accurately capture local patterns and dynamics in sequence data. For example, suppose an object is moving steadily along a straight path but suddenly changes direction or speed due to an abrupt maneuver. The TCN captures this immediate change by focusing on recent time steps, effectively modeling the sudden local variation in motion. The Temporal Transformer then integrates these short-term features with the entire motion sequence to understand how the abrupt change fits into the overall trajectory, enabling accurate prediction of the object's future position by accounting for both recent deviations and long-term trends.

The TCN effectively mitigates the limitations of the Temporal Transformer in modeling short-term dependencies, particularly when the tracking task is particularly dependent on localized features in the short term. Moreover, compared to the Temporal Transformer, which requires complex self-attention computation, the TCN typically has a simpler model structure. This not only alleviates computational demands but also allows TCN models to perform training and inference more rapidly in real-world scenarios.

3.3.3 Momentum correction loss (MCL)

To enhance the motion predictor model and obtain more reliable future predictions, we incorporate contextual information, such as motion direction. Designing effective loss functions plays a crucial role in enhancing tracking performance. For instance, the Shrinkage Loss proposed in [21] improves the discriminative ability of tracking models by penalizing easy negative samples more than hard ones. We propose a novel loss called Momentum Correction Loss (MCL), which serves as a regularizer for the motion prediction loss. The MCL function is designed to align the predicted motion directions with the actual motion directions, enhancing the model's responsiveness to sudden movements. As illustrated in Fig. 5, we obtain the predicted and real motion directions, represented by the blue and green arrows, respectively. The proposed MCL operates by calculating the angular difference between the predicted and real motion directions at key points of the bounding box.

Given two points (x_1, y_1) and (x_2, y_2), representing the positions of an object at two consecutive time steps, the motion direction θ is calculated using the arctangent function:

$$\theta = \arctan\left(\frac{y_1 - y_2}{x_1 - x_2}\right). \tag{5}$$

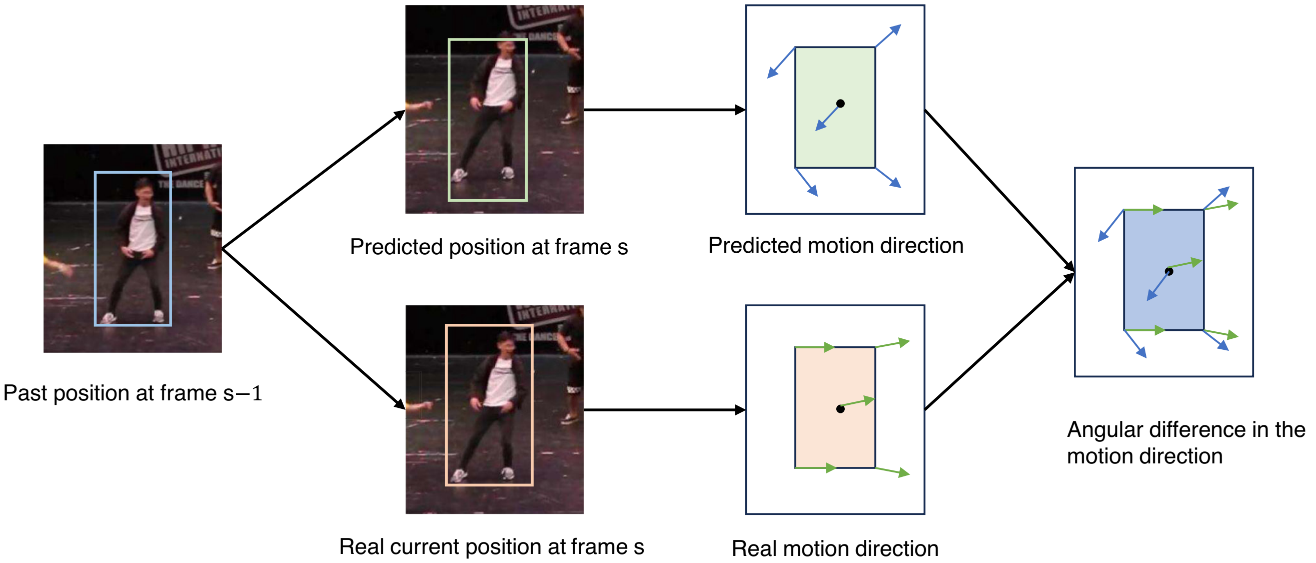

Past position at frame s−1
Predicted position at frame s
Real current position at frame s
Predicted motion direction
Real motion direction
Angular difference in the motion direction

Fig. 5 Illustration of momentum correction loss

In MOT, the position of an object is typically represented by a bounding box. To adjust to sudden pose changes and swift movements, we consider the motion directions at both the center point and the four corners of the bounding box. Therefore, the predicted object motion direction can be represented as $\theta_p = \{\theta_c, \theta_{lt}, \theta_{rt}, \theta_{lb}, \theta_{rb}\}$, where $\{\theta_c\}$ is the motion direction at the center point and $\{\theta_{lt}, \theta_{rt}, \theta_{lb}, \theta_{rb}\}$ represent the motion direction at the four corners of the object. The real object motion direction can also be denoted as $\theta_t = \{\theta'_c, \theta'_{lt}, \theta'_{rt}, \theta'_{lb}, \theta'_{rb}\}$. In general, the direction of motion of the center point of an object represents the overall direction of motion. In scenarios where the posture of some objects suddenly changes [32], it is necessary to consider not only the direction of motion of the center point of an object but also the direction of motion of the four corners of the bounding box. As illustrated in Fig. 5, we can capture more pose changes using the four corners of the bounding box. Thus, the Momentum Correction Loss can be calculated as:

$$\mathcal{L}_{MCL} = \frac{1}{5}\sum_{i} |\theta_i - \theta'_i|, i \in (c, lt, rt, lb, rb). \tag{6}$$

This loss encourages the model to minimize the angular difference between predicted and actual motion directions across all five key points, ensuring that the predicted movements are directionally consistent with the true object motions. By incorporating MCL into the total loss function alongside the motion prediction loss, we provide the model with additional supervision that improves its ability to adapt to rapid motion changes and enhances overall prediction accuracy.

3.4 Training and inference

Training. In the training phase, we train the motion predictor by utilizing the historical positions of n frames and predict the object positions of the subsequent frames. We use the L1 loss function as the prediction loss to supervise the training process and improve the capacity to address outliers. Specifically, given the predicted offset $\widehat{\mathbf{V}} = \left\{\widehat{v}_x, \widehat{v}_y, \widehat{v}_w, \widehat{v}_h\right\}$, and the corresponding attributes of the ground truth V, the prediction loss $\mathcal{L}_{pred}$ is obtained by:

$$\mathcal{L}_{pred}(\widehat{\mathbf{V}}, \mathbf{V}) = \sum_{i} |\hat{v}_i - v_i|, i \in (x, y, w, h). \tag{7}$$

The final loss function sums both the motion prediction loss $\mathcal{L}_{pred}$ and momentum correction loss $\mathcal{L}_{MCL}$:

$$\mathcal{L}_{final} = \mathcal{L}_{pred} + \beta\mathcal{L}_{MCL}, \tag{8}$$

where β is a crucial hyper-parameter that determines the extent to which the momentum correction loss function is influential. It is worth noting that additional direction information is required only during the training phase. By contrast, during the prediction phase, the model relies solely on the observed trajectory to predict the future trajectory. **Inference.** Our motion predictor is applied to the ByteTrack [49] platform, which introduces a two-step association algorithm that utilizes object detection thresholds to track every detection box using tracklets and recovers occluded objects based on similarities. We use the motion predictor to output the prediction boxes P_s in the current frame. By utilizing a YOLOX [13] detector, the detection boxes D_s are obtained in the current frame. ByteTrack's association algorithm employs the Hungarian algorithm to assign detections to tracklets based on the Intersection-over-Union (IoU) between P_s and D_s. These successfully associated detection boxes are incorporated into the historical trajectories for updates, and the unassigned detections are then initialized as new trajectories. For inactive trajectories, our motion predictor continues to generate prediction boxes, that are subsequently appended to trajectories to facilitate model inference. When detections are reconnected to these inactive trajectories, the inactive trajectories may also be retracked. If the inactive time exceeds a given threshold, these preserved prediction boxes are deleted. The pseudo-code of ETTrack is shown in Algorithm 1.

4 Experiments

4.1 Datasets and evaluation metrics

Datasets To conduct a comprehensive evaluation of our method, we performed experiments on various MOT benchmarks including DanceTrack [32], SportsMOT [9] and MOT17 [25]. MOT17 is a widely used foundational benchmark in MOT, in which the motion of pedestrians is mostly linear. In contrast, the SportsMOT dataset captures the intricate motions and similar appearances of athletes in sports scenes, incorporating videos from high-profile events, such as the Olympic Games and NBA, thereby demanding high tracking precision. The DanceTrack dataset presents a particularly complex challenge in terms of object tracking. This is because it consists of objects that look very similar to each other, frequently become occluded from view, and exhibit unpredictable movement patterns. Consequently, it is challenging for any tracking algorithm to conclusively demonstrate its ability to handle complex scenarios effectively. Our aim is to propose a motion predictor that can effectively improve the tracking performance in challenging situations, particularly when the Kalman Filter fails under

Algorithm 1 Pseudo-code of ETTrack.

```
Input: A video sequence $V$; object detector $\mathbf{D}$; motion predictor $\mathbf{T}$;
    detection score threshold $\tau$
Output: Tracks $\mathcal{T}$ of the video
initialization: $\mathcal{T} \Leftarrow \emptyset$
for $frame\ f\ in\ V$ do
    /*Detection */
    $D_f \Leftarrow \mathbf{D}(f)$
    $D_{high} \Leftarrow \emptyset$
    $D_{low} \Leftarrow \emptyset$
    for $d\ in\ D_f$ do
        if $d.score \geq \tau$ then
            $D_{high} \Leftarrow D_{high} \cup \{d\}$
        else
            $D_{low} \Leftarrow D_{low} \cup \{d\}$
        end if
    end for
    /*motion predictor */
    for $trks$ in $\mathcal{T}$ do
        $trks \Leftarrow \mathbf{T}(trks)$
    end for
    /*first association */
    Associate $\mathcal{T}$ and $D_{high}$ using IOU
    $D_{remian} \Leftarrow$ remaining detected boxes from $D_{high}$
    $\mathcal{T}_{remian} \Leftarrow$ remaining predicted boxes from $\mathcal{T}$
    /*second association */
    Associate $\mathcal{T}_{remian}$ and $D_{low}$ using IOU
    $\mathcal{T}_{re-remian} \Leftarrow$ remaining predicted boxes from $\mathcal{T}_{remian}$
    /*delete unmatched tracks */
    $\mathcal{T} \Leftarrow \mathcal{T} \backslash \mathcal{T}_{re-remian}$
    /*initialize new tracks */
    for $d$ in $D_{remian}$ do
        $\mathcal{T} \Leftarrow \mathcal{T} \cup \{d\}$
    end for
end for
Return $\mathcal{T}$
```

diverse scenarios. SportsMOT and DanceTrack are ideal datasets for evaluating the tracking performance.

Metrics The performance of our algorithm is evaluated using the Higher-Order Tracking Accuracy (HOTA) metric [22], Multi-Object Tracking Accuracy (MOTA) [3], Identity F1 (IDF1) Score [28], Association Accuracy (AssA) [22] and Detection Accuracy (DetA) [22]. We adapt HOTA [22] as the primary metric. HOTA combines several sub-metrics and provides a balanced view by considering both Detection Accuracy (DetA) and Association Accuracy (AssA). The HOTA is defined as:

$$\mathrm{HOTA} = \sqrt{\mathrm{DetA} \cdot \mathrm{AssA}} \tag{9}$$

MOTA is a metric used to measure the accuracy of detection. The MOTA can be expressed as:

$$\mathrm{MOTA} = 1 - \frac{|\mathrm{FN}| + |\mathrm{FP}| + |\mathrm{IDSW}|}{|\mathrm{gtDet}|} \tag{10}$$

, where FN, FP, IDSW, and gtDet represent the numbers of false negatives, false positives, IDs and ground truth detections respectively. IDF1 evaluates identity preservation ability and is used to measure association performance. The IDF1 can be calculated as:

$$\mathrm{IDF1} = \frac{|\mathrm{IDTP}|}{|\mathrm{IDTP}| + 0.5|\mathrm{IDFN}| + 0.5|\mathrm{IDFP}|} \tag{11}$$

, where IDTP, IDFN, IDFP denote the numbers of true positive IDs, false negative IDs, and false positive IDs respectively. These metrics are widely used to effectively evaluate the algorithm performance.

4.2 Implementation details

We train the motion predictor solely on the corresponding tracking datasets without integrating any external samples. In the experiments, we use the publicly available YOLOX [13] detector weights developed by ByteTrack [49] for a fair comparison. For the motion predictor, the historical trajectories as input sequences are initially encoded into a vector of size 32 by a fully connected layer, followed by ReLU activation. Subsequently, a TCN is employed to further encode input with feature size of 32. The TCN comprises 4 TCN blocks. Additionally, a dropout ratio of 0.1 is implemented during data processing to prevent overfitting. It is worth noting that all the Temporal Transformer layers also accept inputs with a feature size of 32 and consists of 6 layers employing multi-head self-attention with 8 heads. We optimize the network using the Adam algorithm [18] algorithm with a learning rate of 0.0015 and a batch size of 16 for 50 epochs. The maximum historical trajectory length p is set to 10. On the training datasets, we conduct hyper-parameter optimization of β. We achieve best tracking results using $\beta = 0.3$ on the DanceTrack validation sets. All the experiments are conducted using a GeForce RTX 3090 GPU.

4.3 Benchmark evaluation

Here, we present benchmark results for multiple datasets, such as DanceTrack, SportsMOT, and MOT17. For fair comparison, all tracking-by-detection methods are evaluated using the same detection results and standardized evaluation protocols. This ensures that performance differences are solely due to the tracking methods themselves.

It is important to note that our experiments are conducted under deterministic conditions with fixed random seeds and consistent data splits, resulting in identical outputs across runs. Given this determinism, performance variability

Table 1 Comparison of our method with start-of-the-art MOT algorithms on the DanceTrack test sets

Tracker	HOTA↑	MOTA ↑	DetA↑	AssA↑	IDF1↑
CenterTrack [51]	48.1	86.8	78.1	22.6	35.7
FairMOT [48]	39.7	82.2	66.7	23.8	40.8
QDTrack [11]	45.7	83	72.1	29.2	44.8
TransTrack [30]	45.5	88.4	75.9	27.5	45.2
TraDes [39]	43.3	86.2	74.5	25.4	41.2
GTR [52]	48	84.7	72.5	31.9	50.3
MORT [46]	54.2	79.7	73.5	**40.2**	51.5
MotionTrack [40]	52.9	91.3	80.9	34.7	53.8
SORT [4]	47.9	91.8	72	31.2	50.8
DeepSORT [37]	45.6	87.8	71	29.7	47.9
ByteTrack [49]	47.3	89.5	71.6	31.4	52.5
OC_SORT [5]	55.1	89.4	80.3	38	54.2
ETTrack(ours)	**56.4**	**92.2**	**81.7**	39.1	**57.5**

The best results of methods are marked in bold
Methods in bottom block use the same YOLOX detector

parameters such as standard deviation are negligible, and statistical significance testing is not applicable in this context.

DanceTrack To demonstrate the performance of ETTrack with non-linear object motion and diverse scenarios, the results for the DanceTrack dataset are presented in Table 1. The performance of our method is tested on the DanceTrack test sets. The results demonstrate that ETTrack performs competitively compared with the other methods. Specifically, ETTrack achieved 56.4% HOTA, 92.2% DetA and 57.5% IDF1, which is better than the OC_SORT method with an enhanced Kalman Filter and recovery strategy. These experimental results provide convincing evidence that ETTrack handle effectively complex and non-linear motion patterns in dance sequences. It is noteworthy that a significant gain of 3.2% in IDF1 indicates better identity preservation, which shows ETTrack's robustness in maintaining object identities despite frequent occlusions and abrupt motion changes.

SportsMOT To further evaluate the performance of ETTrack in non-linear scenarios, we conduct experiments on the SportsMOT benchmark. All methods used the same YOLOX detector trained on the SportsMOT training sets with or without validation sets for a fair comparison. As shown in Table 2, these methods with $*$ show that their YOLOX detectors are trained on SportsMOT train and validation sets. The

Table 2 Comparison of our method with start-of-the-art MOT algorithms on the SportsMOT test sets

Tracker	HOTA↑	IDF1↑	AssA↑	MOTA↑	DetA↑	LocA↑	IDs↓	Frag↓
FairMOT [48]	49.3	53.5	34.7	86.4	70.2	83.9	9928	21673
QDTrack [11]	60.4	62.3	47.2	90.1	77.5	88	6377	11850
CenterTrack [51]	62.7	60	48	90.8	82.1	90.8	10481	5750
TransTrack [30]	68.9	71.5	57.5	92.6	82.7	91	4492	9994
BoT-SORT [1]	68.7	70	55.9	94.5	84.4	90.5	5729	5349
ByteTrack [49]	62.8	69.8	51.2	94.1	77.1	85.6	3267	4499
OC_SORT [5]	71.9	72.2	59.8	94.5	86.4	92.4	3093	3474
ETTrack	72.2	72.5	60.1	94.9	86.9	92.5	4075	5279
$*$ByteTrack [49]	64.1	71.4	52.3	95.9	78.5	85.7	3089	4216
$*$MixSort_Byte [9]	65.7	74.1	54.8	96.2	78.8	85.7	**2472**	4009
$*$OC_SORT [5]	73.7	74	61.5	96.5	88.5	92.7	2728	**3144**
$*$MixSort_OC [9]	74.1	74.4	62	96.5	88.5	92.7	2781	3199
$*$ETTrack	**74.3**	**74.5**	**62.1**	**96.8**	**88.8**	**92.8**	3862	4298

The best results of methods are marked in bold
Methods in middle and bottom block use the same YOLOX detector
Methods with $*$ show that their YOLOX detectors are trained on the SportsMOT train and validation sets

Table 3 Tracking performance of investigated algorithms on MOT17 dataset

Tracker	HOTA↑	MOTA ↑	IDF1↑	FP↓	FN ↓	IDs↓ ↓	AssA↑	AssR↑
FairMOT [48]	59.3	73.7	72.3	27500	117000	3,303	58.0	63.6
QDTrack [11]	53.9	68.7	66.3	26600	147000	3,378	52.7	57.2
TransTrack [30]	54.1	75.2	63.5	50200	864000	3,603	47.9	57.1
TransCenter [42]	54.5	73.2	62.2	23100	124000	4,614	49.7	54.2
MORT [46]	57.2	71.9	68.4	21100	136000	2,115	55.8	59.2
TransMOT [8]	61.7	76.7	75.1	36200	93200	2,346	59.9	66.5
GTR [52]	59.1	75.3	71.5	26800	110000	2,859	61.6	–
ByteTrack [49]	63.1	**80.3**	77.3	25500	**83700**	2,196	62.0	**68.2**
StrongSORT [10]	**63.5**	78.5	**78.3**	–	–	**1,446**	63.7	–
OC_SORT [5]	63.2	78.0	77.5	**15100**	108000	1,950	**63.2**	67.5
ETTrack	61.9	79.0	75.9	23100	93300	2,118	60.5	67.0

The best results of methods are marked in bold
Methods in bottom block use the same YOLOX detector

evaluation results presented in Table 2 show that ETTrack achieves 74.3% in HOTA, 74.5% in IDF1, 62.1% in AssA, 96.8% in MOTA, and 88.8% in DetA. Compared with MixSort_OC [9] that designs appearance based association to enhance OC_SORT, our method still outperforms it. Specifically, ETTrack outperforms ByteTrack by up to 10.2% in HOTA and 3.1% in IDF1. These results indicate the effectiveness of ETTrack in dealing with non-linear motions in sports scenarios. These scenarios often involve rapid, unpredictable movements and interactions between multiple objects. Our method effectively captures these complex motion patterns, leading to notable improvements in both HOTA and IDF1 metrics.

MOT17 Table 3 presents the tracking performance on the test set of MOT17 to validate the generalizability of the proposed motion model, which covers linear object motion. MOT17 primarily features linear and predictable pedestrian motion where simpler motion models like the Kalman Filter are highly effective, so ETTrack's advanced temporal modeling does not provide significant advantages in this context. The results show that although our method achieves results comparable to existing benchmarks, it slightly underperforms relative to the current state-of-the-art methods. Despite being optimized for complex and non-linear motion patterns, our approach still attains competitive performance on MOT17. Thus, ETTrack consistently demonstrates robust generalizability across different types of motion patterns.

4.4 Ablation study

A series of ablation studies are conducted on the validation set of the DanceTrack to assess the impact of model components, momentum correction loss, and some hyper-parameters on our proposed method.

Impact of motion modeling To assess the effectiveness of our motion predictor, we conduct a comparative study using various existing motion models. We incorporate different motion models into the tracking process, as shown in Table 4. Obviously, the Kalman Filter surpasses the IoU association method, demonstrating the considerable potential of motion models in tracking objects, especially when appearance information is unreliable. As shown in Table 4, our method has a significant advantage over the Kalman Filter and Vanilla Transformer [35], as measured by the HOTA and IDF1. Our motion predictor outperforms the Kalman Filter, which relies on linear motion, by up to 6.5% in HOTA and 1.9% in IDF1. Regarding computational complexity and real-time performance, we compare the number of parameters, FLOPs (Floating Point Operations), and inference speeds of the models. The Vanilla Transformer, with a

Table 4 Comparison of different motion models on the DanceTrack validation sets

Tracker	FLOPs	Param	HOTA↑	MOTA ↑	DetA ↑	AssA↑	IDF1↑	FPS ↑
No motion	–	–	44.7	87.3	79.6	25.3	36.8	24.2
Kalman Filiter	–	–	46.8	87.5	70.2	31.3	52.1	22.7
Vanilla Transformer	16.57M	2.9M	51.9	89.3	78.3	35.5	52.7	19.2
Ours	8.64M	2.1M	**53.3**	**90.0**	**78.5**	**36.3**	**54.0**	20.1

The best results of methods are marked in bold

Table 5 Evaluation of different model components

	HOTA↑	MOTA↑	DetA ↑	AssA↑	IDF1↑
W/o TCN	52.2	89.8	**78.7**	35.7	53.0
Ours	**53.3**	**90.0**	78.5	**36.3**	**54.0**

"W/o" means that the TCN is removed from the motion predictor
The best results are marked in bold

Table 6 Comparison with/without MCL

	HOTA↑	MOTA↑	DetA ↑	AssA↑	IDF1↑
W/o MCL	52.5	89.9	**78.6**	35.1	53.1
Ours	**53.3**	**90.0**	78.5	**36.3**	**54.0**

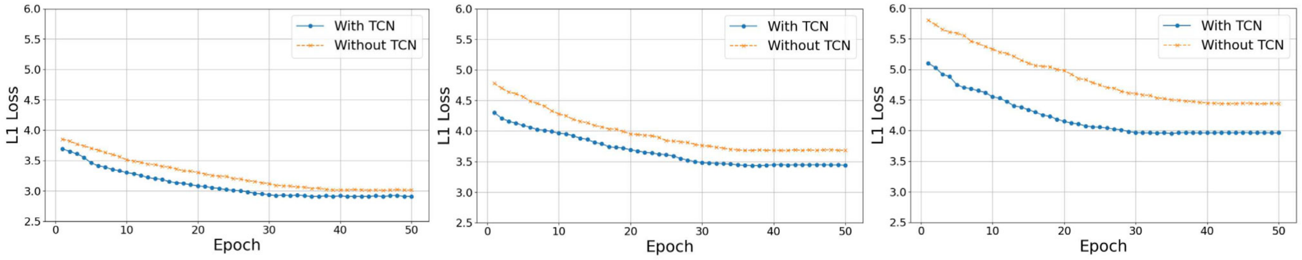
"W/o" means that the no motion direction information is input to motion predictor
The best results are marked in bold

larger parameter count of 2.9 million and the highest FLOPs at 16.57 million, is slightly slower at 19.2 FPS. Our motion predictor maintains a balanced parameter count of 2.1 million and the lowest computational cost with 8.64 million FLOPs. It attains a competitive inference speed of 20.1 FPS and is efficient for real-time applications. These results show that our motion predictor, combining a TCN and a Temporal Transformer models, learns motion patterns more effectively and predicts object positions more accurately while maintaining acceptable computational complexity and real-time performance.

Impact of model components We conduct an ablative experiment to assess the impact of the core components on our proposed model. Specifically, the TCN is deactivated in our motion predictor to examine the resulting changes in the tracking performance. As shown in Table 5, when the TCN is removed from the motion predictor, the HOTA and IDF1 decreased by 1.1% and 1.0%, respectively. To further investigate the TCN's capacity to introduce beneficial inductive biases, we train models with and without the TCN component using 10%, 50%, and 100% of the training data, measuring the L1 loss over 50 epochs. Figure 6 illustrates the training curves for both models under different data sizes. With only 10% of the training data, the model incorporating the TCN converges significantly faster and achieves lower training loss than the model without the TCN, demonstrating the TCN's effectiveness in facilitating efficient learning from limited data. As the amount of training data increases to 50% and 100%, the performance gap narrows but remains noticeable, which indicates that the TCN continues to contribute positively even with more data. By integrating the TCN into our motion predictor, we introduce beneficial inductive biases that efficiently model short-term dependencies. This synergy mitigates the Transformer's limitations in modeling localized temporal features, particularly under limited training data scenarios. Consequently, our design effectively predicts object motions and thus proves its efficacy in both performance and data efficiency. Nevertheless, we may face limitations in scenes with highly irregular motion or when training data is severely limited, as the model relies on learned temporal patterns for accurate predictions. In the future, we will focus on enhancing the temporal modeling capabilities of our motion predictor or employing data augmentation techniques to address these limitations.

Impact of the Momentum Correction Loss An ablation study examines how the momentum correction loss (8) behaves, as summarized by the results shown in Table 6. We measure the tracking performance when momentum correction loss is used to train the motion predictor. HOTA and IDF1 levels increase by 0.8% and 0.9%, respectively. The results demonstrate the impact of utilizing the motion direction information in future motion prediction models. There are many sudden changes in pose and swift movements in datasets such as DanceTrack, which makes predictions that rely on past trajectories insufficient. Our future research will explore additional possibilities for incorporating more information into motion prediction.

Impact of historical trajectory length To demonstrate how the tracking performance is affected by the length of the historical trajectory, we evaluate our proposed method at different p values. The results, as listed in Table 7, indicate that



100 percentage of training dataset

50 percentage of training dataset

10 percentage of training dataset

Fig. 6 The influence of training set amount on prediction performance

Table 7 Evaluation of p on the DanceTrack validation sets

p	HOTA↑	MOTA↑	DetA ↑	AssA↑	IDF1↑
5	51.5	89.9	77.8	35.1	52.1
8	53.1	89.9	78.2	35.5	53.4
10	**53.3**	**90.0**	**78.5**	36.3	**54.0**
13	53.2	90.0	78.4	**36.4**	53.8
15	52.9	90.1	78.1	36.2	53.1

The best results are marked in bold

a very small historical trajectory length fails to provide sufficient information, resulting in unreliable predictions. Our results indicate that extending the historical trajectory length provides a more comprehensive analysis of object motion. However, a very large historical trajectory length tends to provide a considerable degree of noise, which in turn negatively affects the tracking performance. Therefore, we selected T to be 10, which accounts for 0.5 seconds of the object's historical trajectory, based on a video frame rate of 20 FPS. The results presented in Table 7 demonstrate the significance of choosing an appropriate historical trajectory length to achieve superior performance in object-tracking tasks.

Impact of the weight of the Momentum Correction Loss Finally, we also explore the effect of the hyper-parameter β, which determines the degree to which the momentum correction loss affects the final objective function. As shown in Table 8, the best results are obtained on the DanceTrack validation sets when β was set to 0.3.

4.5 Qualitative results

Figure 7 shows qualitative comparisons of ETTrack and OC_SORT. The first row shows that OC_SORT causes ID switching between frame #240 and #265 owing to the object occlusion or non-linear motion. The Kalman Filter's assumption of linear motion prevents OC_SORT from accurately predicting sudden pose changes, resulting in false matches. In the second and third row, OC_SORT still leads to ID switch due to severe occlusion or non-linear motion, respectively between frames #315 to #332 and frames #132 to #146. By contrast, ETTrack maintains consistent identities despite severe occlusions and non-linear movements. Unlike OC_SORT, which suffers from identity switches due to its reliance on linear motion assumptions, ETTrack effectively handles these complex scenarios. Figure 7 illustrates examples where ETTrack successfully tracks objects through complex motions. These qualitative results complement our quantitative findings and provide visual evidence of ETTrack's superior performance in challenging scenarios.

Table 8 Evaluation of β in (8)

β	HOTA↑	MOTA↑	DetA ↑	AssA↑	IDF1↑
0	52.2	89.9	78.7	35.5	53.1
0.1	52.6	89.9	78.6	35.6	53.3
0.2	52.9	**90.1**	**78.9**	36.0	53.7
0.3	**53.3**	90.0	78.5	**36.3**	**54.0**
0.4	52.7	89.8	78.5	36.2	53.5
0.5	51.7	89.8	78.6	35.6	52.9

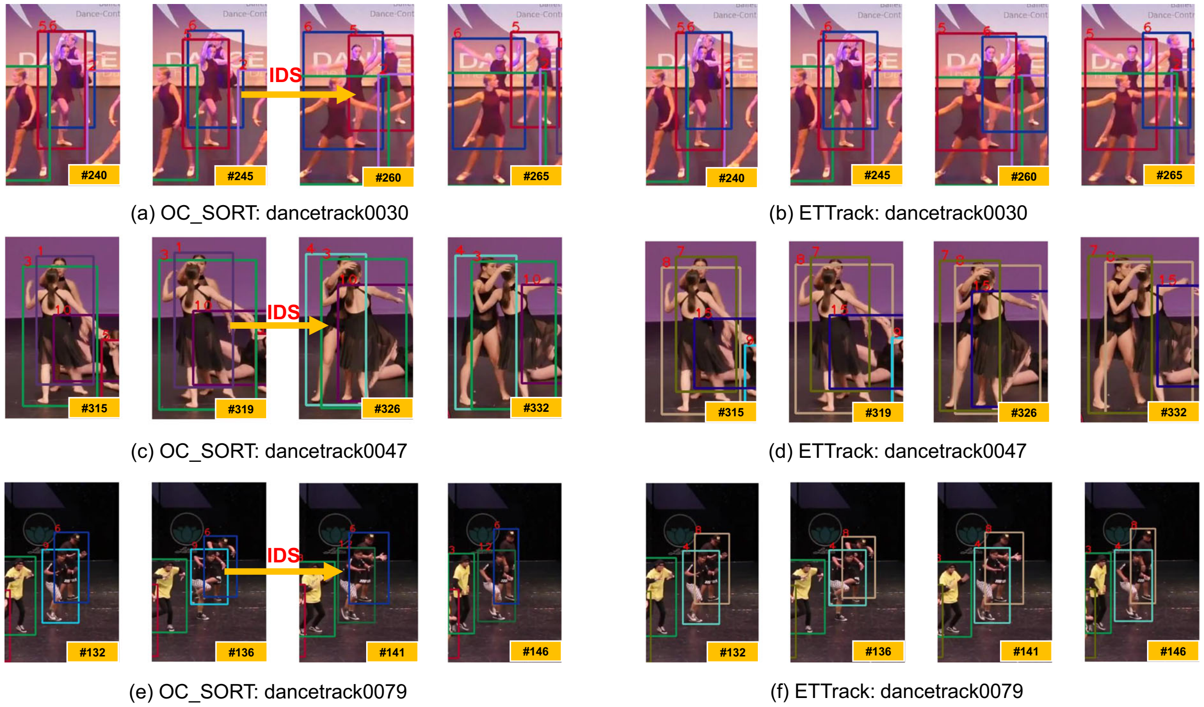
The best results are marked in bold

The visualized results of ETTrack on the test set of SportsMOT and MOT17 are shown in Fig. 8. ETTrack provides accurate predictions on the test set of SportsMOT. It is demonstrated that our method can predict the positions of objects accurately in sports scenarios where the objects exhibit rapid and non-linear motions. Figure 8 also shows several ETTrack's tracking results on the test set of MOT17. It can be observed that although the MOT17 dataset is designed to track pedestrians in scenarios where motion patterns are generally linear, our method still delivers impressive tracking results.

5 Conclusion

In this paper, we propose a motion-based MOT called ETTrack, which uses an enhanced temporal motion predictor to improve object association and tracking performance in non-linear motion. The motion predictor integrates a Temporal Transformer model and a Temporal Convolutional Network (TCN) to effectively capture both local and global historical motion patterns. In addition, the proposed method introduces a novel Momentum Correction Loss to guide the motion predictor during training and improve its ability to handle complex movements. As a result, compared to other motion models based on the Kalman Filter and deep learning, ETTrack achieves a competitive performance on DanceTrack and SportsMOT, achieving scores of 56.4% and 74.4% in HOTA metrics, respectively. It also performs comparably on pedestrian-centric datasets like MOT17, which demonstrates its adaptability to different motion patterns. Furthermore, the proposed ETTrack method holds significant potential for applications beyond traditional MOT scenarios. Specifically, it can be extended to microscopic image analysis fields, such as microbiological image analysis [47] and cell image analysis [27], where accurate tracking of microscopic entities is essential. By effectively handling complex and non-linear motion patterns, ETTrack could significantly contribute to advancements in these fields and broaden the impact of our work.

Despite these advancements, ETTrack still has certain limitations as it fails to consider the effects of camera movement on coordinate system transformations in dynamic scenes



(a) OC_SORT: dancetrack0030

(b) ETTrack: dancetrack0030

(c) OC_SORT: dancetrack0047

(d) ETTrack: dancetrack0047

(e) OC_SORT: dancetrack0079

(f) ETTrack: dancetrack0079

Fig. 7 Qualitative comparison of OC_SORT and ETTrack (Ours). OC_SORT leads to ID switch due to non-linear motion or severe occlusion, but ETTrack still maintains the identity

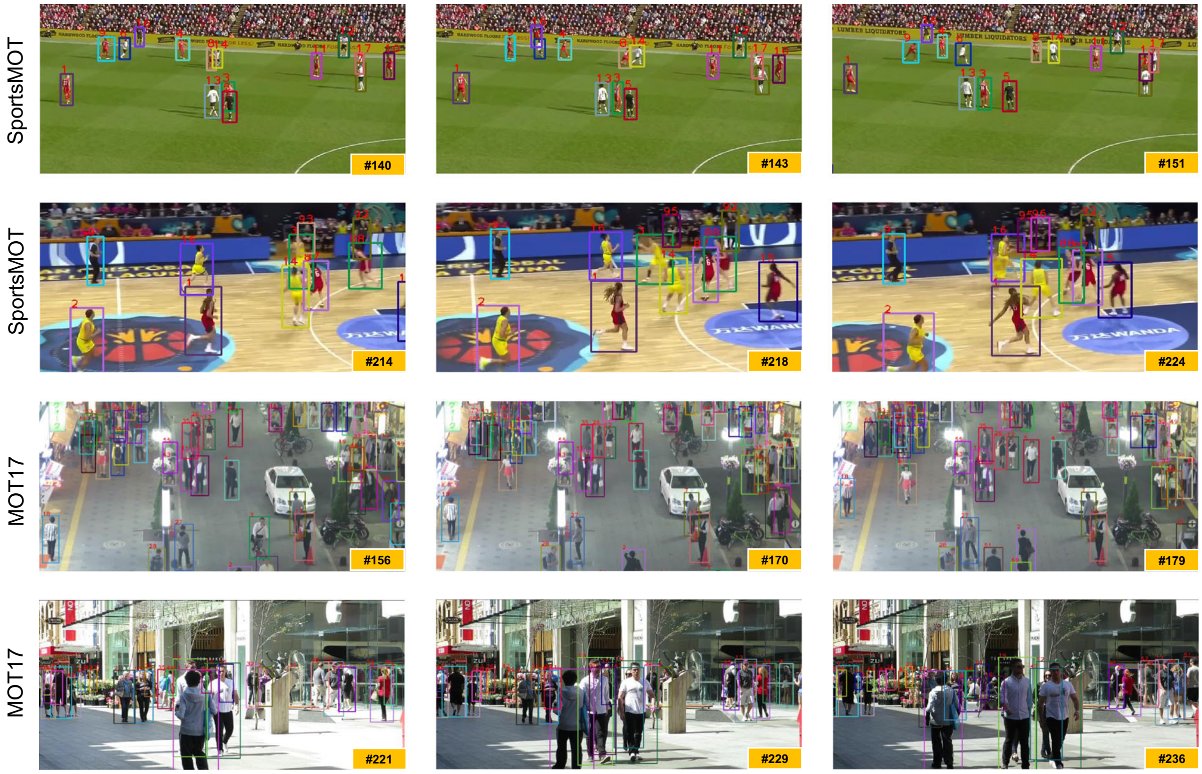


Fig. 8 The visualizations of ETTrack's tracking results on the test set of SportsMOT and MOT17. Boxes of the same color denote the same object

when modeling object motion. Additionally, ETTrack also faces challenges in scenes with highly irregular motion or severely limited training data due to its reliance on learned temporal patterns. In future work, we will enhance the model's generalization and robustness by exploring data augmentation techniques and incorporating camera motion information.

Author Contributions Conceptualization: Xudong Han and Nobuyuki Oishi; Methodology: Xudong Han and Yueying Tian; Formal analysis and investigation: Xudong Han and Elif Ucurum; Writing - original draft preparation: Xudong Han and Philip Birch; Writing - review and editing: Xudong Han and Rupert Young; Supervision: Rupert Young, Chris Chatwin and Philip Birch.

Funding The authors did not receive support from any organization for the submitted work.

Data Availability and Access The datasets analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing Interests The authors declare that they have no relevant financial or non-financial interests to disclose in this paper.

Ethical and Informed Consent for Data Used The data used in this paper are all publicly available and have no ethical violations.

References

1. Aharon N, Orfaig R, Bobrovsky BZ (2022) Bot-sort: Robust associations multi-pedestrian tracking. arXiv:2206.14651
2. Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271
3. Bernardin K, Stiefelhagen R (2008) Evaluating multiple object tracking performance: the clear mot metrics. EURASIP J Image and Video Process 2008:1–10. https://doi.org/10.1155/2008/246309
4. Bewley A, Ge Z, Ott L, et al (2016) Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP), IEEE, pp 3464–3468, https://doi.org/10.1109/icip.2016.7533003
5. Cao J, Pang J, Weng X, et al (2023) Observation-centric sort: Rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9686–9696, https://doi.org/10.1109/cvpr52729.2023.00934
6. Chaabane M, Zhang P, Beveridge JR, et al (2021) Deft: Detection embeddings for tracking. arXiv:2102.02267
7. Choi W (2015) Near-online multi-target tracking with aggregated local flow descriptor. In: Proceedings of the IEEE international conference on computer vision, pp 3029–3037, https://doi.org/10.1109/iccv.2015.347
8. Chu P, Wang J, You Q, et al (2023) Transmot: Spatial-temporal graph transformer for multiple object tracking. In: Proceedings of the IEEE/CVF Winter Conference on applications of computer vision, pp 4870–4880, https://doi.org/10.1109/wacv56688.2023.00485
9. Cui Y, Zeng C, Zhao X, et al (2023) Sportsmot: A large multi-object tracking dataset in multiple sports scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 9921–9931, https://doi.org/10.1109/iccv51070.2023.00910
10. Du Y, Zhao Z, Song Y, et al (2023) Strongsort: Make deepsort great again. IEEE Trans Multimed. https://doi.org/10.1109/tmm.2023.3240881
11. Fischer T, Huang TE, Pang J, et al (2023) Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/tpami.2023.3301975
12. Gao R, Wang L (2023) Memotr: Long-term memory-augmented transformer for multi-object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 9901–9910, https://doi.org/10.1109/iccv51070.2023.00908
13. Ge Z, Liu S, Wang F, et al (2021) Yolox: Exceeding yolo series in 2021. arXiv:2107.08430
14. Gulati A, Qin J, Chiu CC, et al (2020) Conformer: Convolution-augmented transformer for speech recognition. arXiv:2005.08100
15. Han S, Wang H, Yu E, et al (2023) Ort: Occlusion-robust for multi-object tracking. Fundamental Res. https://doi.org/10.1016/j.fmre.2023.02.003
16. Kalman RE, et al (1960) Contributions to the theory of optimal control. Bol soc mat mexicana 5(2):102–119. https://doi.org/10.1109/9780470544334.ch8
17. Kesa O, Styles O, Sanchez V (2022) Multiple object tracking and forecasting: Jointly predicting current and future object locations. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 560–569, https://doi.org/10.1109/wacvw54805.2022.00062
18. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. arXiv:1412.6980
19. Kuhn HW (1955) The hungarian method for the assignment problem. Naval Res logistics quarterly 2(1-2):83–97. https://doi.org/10.1002/nav.20053
20. Lehmann EL, Casella G (2006) Theory of point estimation. Springer Sci Business Med, https://doi.org/10.1007/b98854
21. Lu X, Ma C, Shen J, et al (2020) Deep object tracking with shrinkage loss. IEEE Trans on Pattern Anal Mach Intell 44(5):2386–2401. https://doi.org/10.1109/tpami.2020.3041332
22. Luiten J, Osep A, Dendorfer P, et al (2021) Hota: A higher order metric for evaluating multi-object tracking. Int J Comput Vis 129:548–578. https://doi.org/10.1007/s11263-020-01375-2
23. Luo W, Yang B, Urtasun R (2018) Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 3569–3577, https://doi.org/10.1109/cvpr.2018.00376
24. Meinhardt T, Kirillov A, Leal-Taixe L, et al (2022) Trackformer: Multi-object tracking with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8844–8854, https://doi.org/10.1109/cvpr52688.2022.00864

25. Milan A, Leal-Taixé L, Reid I, et al (2016) Mot16: A benchmark for multi-object tracking. arXiv:1603.00831
26. Milan A, Rezatofighi SH, Dick A, et al (2017) Online multi-target tracking using recurrent neural networks. In: Proceedings of the AAAI conference on Artificial Intelligence, https://doi.org/10.1609/aaai.v31i1.11194
27. Rahaman MM, Li C, Yao Y, et al (2021) Deepcervix: A deep learning-based framework for the classification of cervical cells using hybrid deep feature fusion techniques. Comput Bio Med 136:104649. https://doi.org/10.1016/j.compbiomed.2021.104649
28. Ristani E, Solera F, Zou R, et al (2016) Performance measures and a data set for multi-target, multi-camera tracking. In: European conference on computer vision, Springer, pp 17–35, https://doi.org/10.1007/978-3-319-48881-3_2
29. Sun J, Liu Y (2023) Design of 360° dead-angle-free smart desk lamp based on visual tracking. HighTech and Innov J 4(4):761–767. https://doi.org/10.28991/hij-2023-04-04-05
30. Sun P, Cao J, Jiang Y, et al (2020a) Transtrack: Multiple object tracking with transformer. arXiv:2012.15460
31. Sun P, Kretzschmar H, Dotiwalla X, et al (2020b) Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2446–2454, https://doi.org/10.1109/cvpr42600.2020.00252
32. Sun P, Cao J, Jiang Y, et al (2022) Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 20993–21002, https://doi.org/10.1109/cvpr52688.2022.02032
33. Torres-Ronda L, Beanland E, Whitehead S, et al (2022) Tracking systems in team sports: a narrative review of applications of the data and sport specific analysis. Sports Med-Open 8(1):15. https://doi.org/10.1186/s40798-022-00408-z
34. Trujillo D, Morales LA, Chávez D, et al (2023) Trajectory tracking control of a mobile robot using neural networks. Emer Sci J 7(6):1843–1862. https://doi.org/10.28991/esj-2023-07-06-01
35. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Neural Information Processing Systems,Neural Inf Process Syst https://doi.org/10.48550/arXiv.1706.03762
36. Wang Z, Zheng L, Liu Y, et al (2020) Towards real-time multi-object tracking. In: European conference on computer vision, Springer, pp 107–122, https://doi.org/10.48550/arXiv.1909.12605
37. Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP), IEEE, pp 3645–3649, https://doi.org/10.1109/icip.2017.8296962
38. Wu H, Xiao B, Codella N, et al (2021a) Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 22–31, https://doi.org/10.1109/iccv48922.2021.00009
39. Wu J, Cao J, Song L, et al (2021b) Track to detect and segment: An online multi-object tracker. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12352–12361, https://doi.org/10.1109/cvpr46437.2021.01217
40. Xiao C, Cao Q, Zhong Y, et al (2023) Motiontrack: Learning motion predictor for multiple object tracking. arXiv:2306.02585
41. Xu N, Lin W, Lu X et al (2024) Video object tracking: Tasks, datasets, and methods. Synth Lectures on Comput Vis. https://doi.org/10.1007/978-3-031-44660-3
42. Xu Y, Ban Y, Delorme G, et al (2022) Transcenter: Transformers with dense representations for multiple-object tracking. IEEE Trans Pattern Anal Mach Intell 45(6):7820–7835. https://doi.org/10.1109/tpami.2022.3225078
43. Yang J, Ge H, Su S, et al (2022) Transformer-based two-source motion model for multi-object tracking. Applied Intell pp 1–13. https://doi.org/10.1007/s10489-021-03012-y
44. Yuan Y, Iqbal U, Molchanov P, et al (2022) Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11038–11049, https://doi.org/10.1109/cvpr52688.2022.01076
45. Zagoruyko S, Komodakis N (2016) Wide residual networks. arXiv:1605.07146
46. Zeng F, Dong B, Zhang Y, et al (2022) Motr: End-to-end multiple-object tracking with transformer. In: European Conference on Computer Vision, Springer, pp 659–675, https://doi.org/10.1007/978-3-031-19812-0_38
47. Zhang J, Li C, Kosov S, et al (2021a) Lcu-net: A novel low-cost u-net for environmental microorganism image segmentation. Pattern Recogn 115:107885. https://doi.org/10.1016/j.patcog.2021.107885
48. Zhang Y, Wang C, Wang X, et al (2021b) Fairmot: On the fairness of detection and re-identification in multiple object tracking. Int J Comput Vis 129:3069–3087. https://doi.org/10.1007/s11263-021-01513-4
49. Zhang Y, Sun P, Jiang Y, et al (2022) Bytetrack: Multi-object tracking by associating every detection box. In: European conference on computer vision, Springer, pp 1–21, https://doi.org/10.1007/978-3-031-20047-2_1
50. Zhang Y, Wang T, Zhang X (2023) Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 22056–22065, https://doi.org/10.1109/cvpr52729.2023.02112
51. Zhou X, Koltun V, Krähenbühl P (2020) Tracking objects as points. In: European conference on computer vision, Springer, pp 474–490, https://doi.org/10.48550/arXiv.2004.01177
52. Zhou X, Yin T, Koltun V, et al (2022) Global tracking transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8771–8780, https://doi.org/10.1109/cvpr52688.2022.00857
53. Zhu X, Su W, Lu L, et al (2020) Deformable detr: Deformable transformers for end-to-end object detection. arXiv:2010.04159

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xudong Han is a PhD student at the Industrial Informatics and Signal Processing Research Group, University of Sussex. He received the M.Sc. degree in Robotics and Autonomous Systems from the University of Sussex, Brighton, U.K., in 2022. His research interests are multiple object detection and tracking, depth estimation and trajectory prediction.

Nobuyuki Oishi is a PhD student at the Wearable Technologies Lab, University of Sussex. His research focuses on advancing wearable Inertial Measurement Unit (IMU)-based Human Activity Recognition (HAR) systems by leveraging human motion data and physics simulations. His work aims to address critical challenges in the field, including the lack of sufficient training data and the cost of annotating sensor data.

Yueying Tian is a Ph.D. student in the Industrial Informatics and Signal Processing Research Group within the School of Engineering and Informatics at the University of Sussex. She acquired her Master of Science degree (2022, Cardiff), from the school of Computer Science & Informatics at the University of Cardiff. Her research interests are in the field of medical imaging using machine learning techniques, generative models, image processing and object detection.

Elif Ucurum is a Ph.D. student within the School of Engineering and Informatics at the University of Sussex. She holds a BSc in Electrical and Electronics Engineering and earned an MSc in Image and Video Communications and Signal Processing from the University of Bristol in 2020. Her research focuses on video-based small object detection, with an emphasis on drone detection and tracking.

Dr Rupert Young is a Reader in Engineering at the University of Sussex; he graduated from the University of Glasgow, from where he also gained his PhD in optical signal processing. Since 1995 he has been with the Department of Engineering and Design, University of Sussex, in which he served as Head of Department from 2006 to 2011. He has published over 300 refereed journal and conference papers on various aspects of digital signal and image processing, pattern recognition, electro-optics and communications.

Professor Chris Chatwin holds the Chair in Engineering at the University of Sussex, UK; where, *inter alia*, he is a Research Director of the "Industrial Informatics & Signal Processing Research Group.". He has published two research level monographs and more than four hundred international papers which focus on: industrial informatics, middleware for business systems, security systems & e-commerce optics, signal processing, laser systems, digital image processing, biometrics,. He is a member of: the Institution of Electrical and Electronic Engineers (Senior); Association for Computing Machinery, British Computer Society, the Association of Industrial Laser Users. He is a Chartered Engineer, Euro-Engineer, International Professional Engineer, Chartered Physicist and a Fellow of: The Institution of Engineering and Technology, The Institution of Mechanical Engineers, The Institute of Physics. Profile: http://www.sussex.ac.uk/profiles/9815

Philip Birch is a Reader in Engineering at the University of Sussex, UK. He received a BSc in Physics from the University of Durham, UK in 1994 and a PhD from the same institution in 1998. His current research interests are in optical imagining, signal processing, and computer vision for object detection and tracking. He has over 170 publications in these areas.