Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01426-y

ORIGINAL ARTICLE

SCGTracker: object feature embedding enhancement based on graph attention networks for multi-object tracking

Xin Feng[1] · Xiaoning Jiao[1] · Siping Wang[1] · Zhixian Zhang[1] · Yan Liu[2]

Received: 17 July 2023 / Accepted: 9 March 2024

Abstract
Multi-object tracking (MOT) is a task to identify objects in videos, however, objects with similar appearance or occlusion may cause frequent ID switching, which is the main challenge of current MOT. In this paper, we propose a novel self-cross graph neural network-based multi-object tracking method, which we termed as SCGTracker. This method seamlessly integrates object detection and tracking through graph neural networks, building upon the foundation of the JDE paradigm. Specifically, we construct graph structures to capture the correlation between objects in both spatial and temporal dimensions. To further tackle the frequent ID switching problem, we employ an attention mechanism to aggregate object context information within the same frame and across different frames, updating the object information via graph neural networks to derive highly distinctive appearance features. Ultimately, the obtained strongly distinguishable object appearance features serve to mitigate the issue of frequent object ID switches. In experiments conducted on the MOT17 test set, our proposed method yields promising results, achieving a 73% Multiple Object Tracking Accuracy (MOTA) and a 73.2% ID $F1$ score. Furthermore, it demonstrates a substantial reduction in ID switches compared with state-of-the-art methods.

Keywords Embedding enhancement · Graph neural networks · Joint detection and embedding · Multi-object tracking

Introduction

Multi-object tracking (MOT) entails the analysis of video footage to identify and track one or more targets. To achieve this, the targets of interest need to be detected in each frame of the video. Correctly associating identical targets across successive frames, as well as accurately handling newly appearing or disappearing targets, is crucial.

In scenarios like video surveillance [27] and autonomous driving [16], multi-object tracking algorithms are frequently employed to detect and track pedestrian targets, aiding in the comprehension and analysis of their movement trajectories. This tracking capability facilitates early warnings of abnormal pedestrian behavior or contributes to effective vehicle control. However, pedestrian targets are subject to various factors, including changes in the external environment, pedestrian posture variations, and object occlusion [21]. Consequently, maintaining a consistent ID for a specific pedestrian target throughout the tracking process becomes challenging, leading to a degradation in tracking effectiveness, as illustrated in Fig. 1.

To address the challenge of frequent switching of pedestrian target IDs, multi-object tracking (MOT) algorithms predominantly focus on enhancing the appearance feature representation of pedestrian targets. The prevalent approach [2, 28, 31, 33] involves training convolutional neural network (CNN) models using both historical frames and current frames as inputs. This allows the CNN to learn associations between historical frames and the current frame, utilizing these associations to improve the feature representation of the current frame. However, a notable drawback is that features of pedestrian objects are typically extracted independently of each other, with minimal consideration given to the interactions between objects. Consequently, several studies have tackled the multi-target tracking task as both a spatio-temporal graph modeling problem. For example, the graph neural networks (GNNs) [41] is employed to capture the interrelationships and contextual information between

✉ Xin Feng
xfeng@cqut.edu.cn

[1] College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 404100, China

[2] School of Artificial Intelligence, Chongqing University of Technology, Chongqing 404100, China

Published online: 11 May 2024

Fig. 1 Illustration of tracking results of FairMOT [39] and our SCGTracker. In the image frames, two pedestrian objects with ID_1 and ID_2 (shown in red and yellow bounding boxes, respectively) are moving toward opposite direction in frame 1, while they are partially obscured at frame 2, and are separated again at frame 3. In (**a**), FairMOT [39] takes the original ID_2 object (in yellow bounding box) as a newly emerging object at frame 3,and assigns a new ID to it. Whereas in (**b**) our proposed SCGTracker maintains the original ID for ID_2 at frame 3. This observation is also applicable to s–6

objects. Studies [13, 36] involve a graph representation for potential connections between trajectories and detection results. TransMOT [4] establishes connections between trajectories in both time and spatial domain using the transformer encoders, treating the connection of tracked objects as sparse graphs. However, the majority of existing graph-based MOT algorithms fall short in addressing the interrelationships among targets within the same frame and do not consider the object occlusion scenarios, where the extracted features may be compromised, potentially leading to correlation errors and error propagation over consecutive frames.

In response to these challenges, we present a joint object detection and tracking method by incorporating a Self-Cross attention Graph to improve the feature representative ability for better Multi-Object Tracking, which we term as SCGTracker. SCGTracker seamlessly integrates object detection and tracking, leveraging the intrinsic characteristics of objects as moving objects with low constant velocity and predictable spatial relationships between neighboring objects. SCGTracker is built on the highly efficient joint detection and embedding (JDE [31]) framework. According to the JDE detected objects and their corresponding feature embeddings, we propose to model the relationships between individual objects in both spatial and temporal domains through building the spatial–temporal object graph for two consecutive frames in a video stream. To reduce the number of ID switch caused mainly by occlusion objects' tracking, we propose to apply the self and cross-attention mechanism in the spatial–temporal object graph. In special, the self-attention aggregates the information of all neighboring objects in a frame for each object. While the cross-attention is to map objects with similar contexts in consecutive frames into a shared space by aggregating relevant information across frames. Through message passing, the self-cross attention enhances the objects' feature by considering both the spatial relationship between objects in a frame and temporal correspondence between two objects across frames. SCGTracker is an efficient online MOT method that optimizes the association of targets in consecutive frames. Through extensive experiments, it is shown to obtain the best performance in terms of both tracking accuracy and the number ID switches.

The contribution of this study can be divided into three aspects.

1. The SCGTracker, our proposed solution, is an end-to-end framework designed for seamless integration of pedestrian target detection and tracking, utilizing graph neural networks. Through this innovative approach, we aim to improve the features associated with pedestrian targets and achieve a globally optimized solution for both detection and tracking tasks.
2. We examine the interrelationships among targets within the same frame by constructing the object map in the spatial dimension for that frame. Additionally, recognizing the smaller target displacement between consecutive frames, we model the targets in successive frames to create an object graph spanning different frames in the temporal dimension.
3. We incorporate a graph neural network, specifically a Self-Cross Attention Graph, to improve the miss tracking of the occluded targets. This is accomplished by spatially aggregating target context information within the same

frame through the self-attention mechanism, temporally aggregating target information across consecutive frames using the cross-attention mechanism, and updating target features through message passing to derive highly discriminative pedestrian object appearance features.

Related work

The tracking-by-detection (TBD)-based MOT algorithms

Numerous Multi-Object Tracking (MOT) algorithms adopt the tracking-by-detection framework, which entails dividing the multi-target tracking task into object detection and trajectory association. A robust detector, such as Faster R-CNN [25], CenterNet [6], or YOLOv5 [43], is crucial for predicting the object's bounding box. These bounding boxes are then linked through data association to establish trajectories. To accomplish this, Bewley et al. [2] initially proposed using Kalman filtering [32] to predict the position of bounding boxes from the previous frame in the current frame. They then utilized the Hungarian algorithm [20] and IOU distance to match these predicted positions with the bounding boxes in the current frame for trajectory association. Subsequently, Wojke et al. [33] introduced the Re-ID network to extract appearance features of the bounding boxes, resulting in improved performance. However, this method demands substantial computational resources due to the necessity for additional Re-ID networks.

The joint detection and tracking (JDE)-based MOT algorithms

The Joint Detection and Embedding (JDE)-based MOT algorithm integrates target detection and re-identification (Re-ID) tasks within a single network. Zhou et al. [41] suggested predicting the offset of object centroids between consecutive frames and utilized it for data association. Wang [31] enhanced the original detection task by incorporating the Re-ID task, achieved by modifying the predictor head of the detector. Computational efficiency was further optimized through feature sharing and multi-task learning. Zhang et al. [39] proposed an architecture based on unanchored target detection, employing different feature maps for the detection and Re-ID tasks to alleviate competition between them. Despite these advancements within the JDE paradigm, there remains potential for further improvement in the accuracy of trackers.

The graph neural network-based MOT algorithms

LGM [8] proposes transforming the target association problem into a graph matching problem by modeling a graph based on relationships between trajectories and detections. It relaxes the undirected graph matching into a continuous quadratic programming problem. TrackMPNN [24] introduces a framework based on dynamic undirected graphs, leveraging Message Passing Graph Neural Networks (GNNs) [41] to generate likelihood for associating each target. Reference [13] constructs an undirected graph between trackers and detections, incorporating target appearance features as node features and pose features as edge features. Node features are updated through node similarity, and aggregated updated edge features. TransMOT [4] establishes trajectory links by constructing encoders in both temporal and spatial domains, treating tracked targets as a sparse band-weighted graph. The decoder component predicts the correspondence between the output of the encoders and the graph representation of the current frame. However, this structure requires substantial computational resources.

Many existing algorithms in this domain often neglect the interdependencies among targets within the same frame, resulting in a diminished correlation between consecutive frames. Moreover, they frequently overlook the impact of occlusion, where the features of a detected target are influenced by unfavorable factors. As a result, the interaction between the detected target and the trajectory target through the graph neural network [41] may inadvertently compromise the initially favorable features of the trajectory target.

Methodology

In the naturally captured videos, there is an assumption that the relationships between multiple moving objects are invariant in a short time period, and even some object is temporally occluded by the obstacles, the relative relationship between this object and others will be maintained. Hence, besides the appearance feature of the individual object, the relative relationship between objects in frame$t$ and the correspondence relationship across consecutive frames are also important known information for objects correlation in MOT. Motivated by this assumption, in this paper, we propose a joint object detection and tracking method based on JDE [31] framework, which we termed as SCGTracker. As shown in Fig. 2, SCGTracker takes two consecutive frames as inputs, and the CNN-based joint object detection and feature embedding are applied to both frames. Then a spatial–temporal object graph is built by taking the feature embedding as node description, the relative position description in a frame as spatial edges, and the object correspondence between two consecutive frames as temporal edges. To improve the

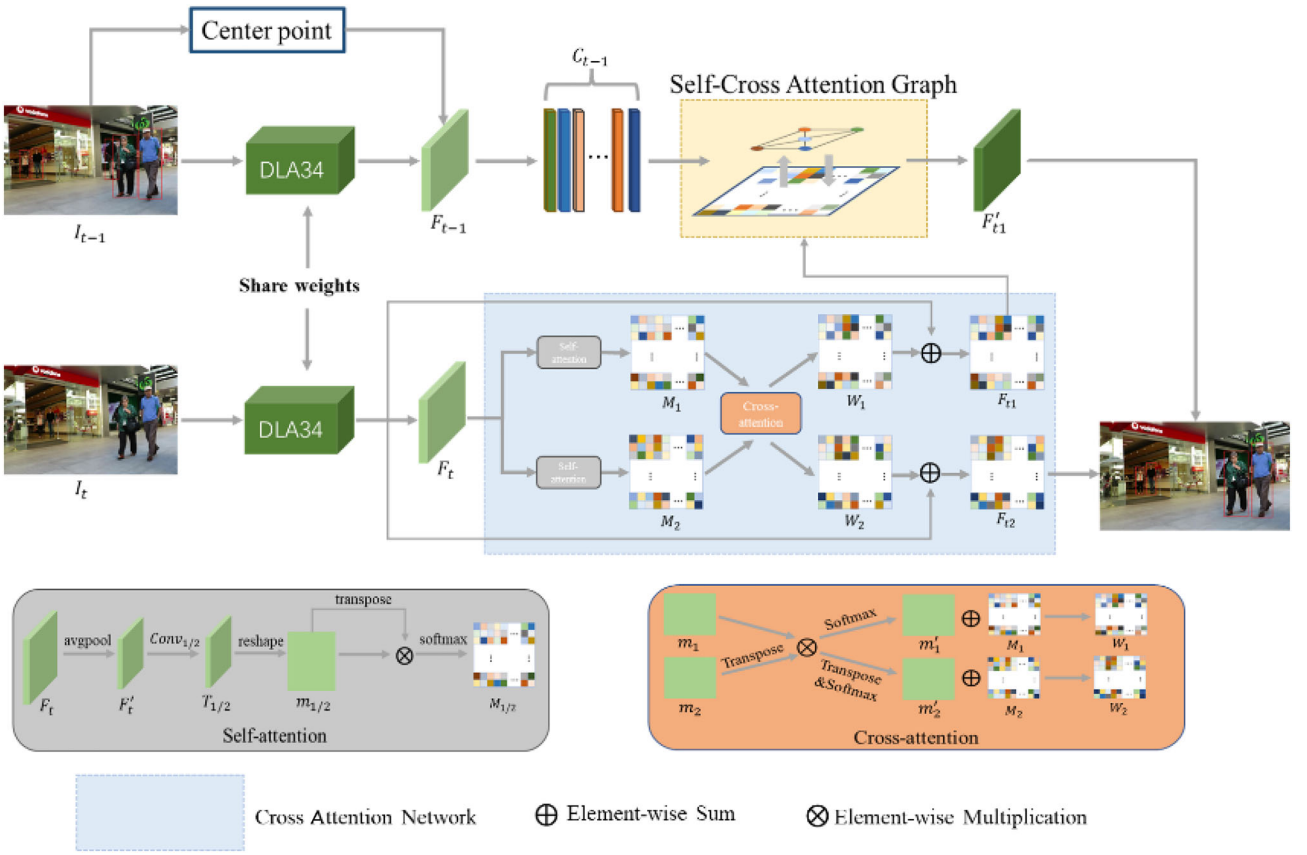

Fig. 2 Overview of SCGTracker. The current frame image I_t is sent to the backbone network to obtain the feature graph F_t. The cross-attention network [12] is used to decompose F_t into two separate feature graphs: F_{t1} for the Re-ID task and F_{t2} for the target detection task. The target feature vector C_{t-1} extracted from the previous frame I_{t-1} is combined with the feature graph F_{t1} and passed through the self-cross graph module to obtain the target feature vector at the moment of I_t

feature representative ability for re-identification of temporary occluded objects, a self-cross attention is applied to the spatial–temporal object graph. During the training of the self-cross attention-based spatial–temporal object graph, the message passing process transfers the correlation information of neighboring objects for a given object, and the aggregation step collects all the context information to update the given object's feature description. As a result, in addition to the appearance feature, the relative position and the temporal correlation are all taken into consideration for target matching, leading to better tracking accuracy and less ID switches in contrast to the MOT methods that only match on object appearance feature. This paper provides some definitions as follows (see Table 1):

Notations:

Table 1 Abbreviated statement expression

Abbreviation	Full name
MOT	Multi-object tracking
CAN	Cross Attention Network
SCG	Self-Cross Attention Graph

I_t:	Represents the current frame image
F_t	The features extracted from the backbone network for image I_t
F_{t1}	Features from F_t used for Re-Identification (Re-ID) tasks

I_t:	Represents the current frame image
I_{t-1}	Represents the previous frame image
F_{t-1}	Features extracted from the backbone network for image I_{t-1}
C_{t-1}	Represents the feature vector extracted from I_{t-1} for a specific target
D_p^{t-1}	Represents the feature information of the p-th target in I_{t-1}
$D^{t-1} = \{D_1^{t-1}, D_2^{t-1}, \cdots, D_{n_{t-1}}^{t-1}\}$	Represents the feature information for all targets in I_{t-1}
n_{t-1}	The total number of targets in I_{t-1}
G_{t-1}	Represents the target graph for I_{t-1}
G_t	Represents the target graph for I_t
E_{self}:	Represents the edges within G_t and G_{t-1}
E_{cross}	Represents the edges between G_t and G_{t-1}
$C_{t-1}{}'$	Represents the target feature vector of I_{t-1} after Self-Cross Attention Graph updates
$F_{t1}{}'$	Represents the target feature map of I_t after Self-Cross Attention Graph updates
$\varphi_{x,y}$	Represents the target feature extracted from $F_{t1}{}'$ with (x, y) as the center

Architecture of proposed method

SCGTracker is designed to achieve efficient and accurate tracking of multi-objects, analyzing their movement trajectories for real-time tracking applications such as autonomous driving. The approach employs a graph neural network [41] to map two pedestrian targets with high similarity in consecutive frames to a common space. This mapping aggregates object information to generate highly expressive object appearance features, thereby preventing confusion in object association. Leveraging these tools, SCGTracker offers a reliable method for pedestrian target tracking.

Our strategy for pedestrian target detection involves employing an enhanced version of the Deep Aggregation Network (DLA-34) [35] as our backbone network. We adhere to the concept of detecting pedestrian targets based on their centroids. We input two consecutive frames into the network, utilizing the image I_{t-1} at time $t-1$ as input to obtain the feature map Ft-1. Given that we can directly acquire position information for the pedestrian target center in the I_{t-1} image, we extract the appearance feature vector C_{t-1} of the pedestrian target based on that position information within the feature map F_{t-1}. This approach enables effective detection of pedestrian targets and extraction of their features for tracking.

In addition to the feature map F_{t-1}, we input the image I_t at time t into our backbone network. The feature map F_t obtained from this process encompasses information regarding object class confidence, object size, and object appearance. Recognizing the distinctions in tasks, we employ the Cross Attention Network (CAN) module to understand both the commonalities and specificities of detection and Re-ID task features. The CAN module learns self-relationships between different feature channels to enhance the feature representation of each task. Simultaneously, it employs a cross-relationship mechanism to capture shared information between the two tasks for commonality learning. Finally, we decompose the feature map F_t into two separate feature maps: F_{t1} for the Re-ID task and F_{t2} for the object detection task.

To facilitate the data association process, we propose constructing keypoints based on the appearance features of pedestrian objects. Specifically, we use the appearance feature vector C_{t-1} of pedestrian objects detected in image I_{t-1} as the keypoint information D_p^{t-1}, which is then organized into $D^{t-1} = \{D_1^{t-1}, D_2^{t-1}, \ldots, D_{n_{t-1}}^{t-1}\}$. Here, n_{t-1} represents the maximum number of objects detected in I_{t-1}. For image I_t, as the position of pedestrian objects is not directly available, we employ the feature vector at each position of the Re-ID task's feature graph F_{t1} as the keypoint information. To aggregate this information, we utilize the graph neural network [41] Self-Cross Attention Graph to combine D^{t-1} and F_{t1}. This process merges the appearance features of pedestrian objects from the previous frame with those in the current frame, resulting in a more expressive representation of pedestrian object appearance in the current frame (the specific algorithm flow is illustrated in Fig. 2) (see Table 2).

Self-cross attention graph

In real-world environments, pedestrian targets can be obscured or affected by motion blur, introducing complexity to the tracking process. Existing tracking algorithms frequently employ the Re-ID features of targets directly in the data association link, without accounting for potential interdependencies between targets. This methodology may undermine the correlation between different frames, leading to a consistent switch in the ID of the same pedestrian target.

Table 2 SCGTracker training process

Step	Process
Input	Current frame image I_t, the previous frame image I_{t-1}
Step 1	Utilize a model to obtain the feature map F_t for the current frame I_t Extract the target feature vectors C_{t-1} from the previous frame I_{t-1}
Step 2	Send the feature map F_t to the CAN module, decomposing it into two parts: F_{t1}: Features used for Re-ID tasks; F_{t2}: Features used for object detection tasks
Step 3	Build graphs G_{t-1} and G_t based on the extracted target feature vectors C_{t-1} and F_{t1}
Step 4	Feed graphs G_{t-1} and G_t into the Self-Cross Attention Graph module
Step 5	Utilize self-attention mechanisms to aggregate and update feature information on each node Node Aggregation: $h_i^l = [x_i^l \| \| m^l_{E_{\text{self}} \to i}]$, where G_{t-1} and G_t connect node i to all other nodes through the edge E_{self} Node Update:$x_i^{l+1} = x_i^l + MLP(h_i^l)$
Step6	Utilize cross-attention mechanisms for further information exchange Node Aggregation: $h_i^l = [x_i^l \| \| m^l_{E_{\text{cross}} \to i}]$, where E_{cross} connects node i in G_{t-1} to all nodes in G_t Node Update:$x_i^{l+1} = x_i^l + MLP(h_i^l)$
Output	Complete interaction between F_{t1} and C_{t-1} to obtain updated feature information, resulting in $F_{t1}{}'$ and $C_{t-1}{}'$. Send the feature map F_{t2} to the detection branch, which outputs the object positions on the image I_t

As a result, the tracking outcomes become unstable, causing a notable deterioration in Multi-Object Tracking performance.

To tackle this challenge, we introduce the Self-Cross Attention Graph. The primary innovation of this approach lies in its spatial modeling of targets within the same frame. It achieves this by leveraging the self-attention mechanism [29] to comprehensively capture information within the target area. Moreover, it aggregates target context information within the same frame and updates target features through message passing. The method further extends spatial correlation to the temporal dimension by modeling targets between successive frames, exploiting the consistent contextual relationships of pedestrian targets over a short period. The cross-attention mechanism [12] enhances focus on target information, and message passing is employed to bring targets with similar contexts in different frames closer in terms of spatial distance. Consequently, this process significantly improves the representation of target features.

The Self-Cross Attention Graph entails the construction of object graphs within the same frame (the previous frame graph G_{t-1} and the current frame graph G_t) and between consecutive frames. The nodes in these graphs correspond to the keypoints in the two images. Intra-frame object graphs connect node i to all other nodes via E_{self}, while object graphs between frames connect node i to all the keypoints in the other object graph through E_{cross}. Both E_{self} and E_{cross} represent undirected edges.

The nodes in the graph G_{t-1} and graph G_t are updated with representation information through messages propagated by their edges E. Following the message passing phase, the SCGTracker derives the updated pedestrian target feature vector $C_{t-1}{}'$ in the image I_{t-1} and F_{t1} of image I_t. From the feature map $F_{t1}{}'$, the Re-ID feature $\varphi_{x,y}$ of the pedestrian target in the current frame is directly extracted, with (x, y) serving as the center.

Attentional aggregation Multi-target tracking tasks often face challenges such as occlusion, changes in attitude, scale variations, and the presence of external or invisible regions, all of which can degrade target feature information. As a remedy, we have introduced an attention mechanism designed to prioritize the undisturbed portion of the target, thereby minimizing sensitivity to disruptive factors.

Self-attention mechanism [29]: Given the similarity in structure between two object graphs in consecutive frames, aggregating self-information within the same object graph can be beneficial for identifying similar nodes.

Cross-attention mechanism [12]: To augment the expressiveness of pedestrian object features in the current frame, a comprehensive comparison is needed between all the keypoints D_p^{t-1} in the previous frame and the feature graph F_{t1} of the current frame. This entails searching for contextual clues that aid in distinguishing a true match from other similarities and filtering out keypoints in the current frame that correspond to D_p^{t-1}. This iterative process focuses attention on specific locations, facilitating information transfer and the completion of different object graphs (as illustrated in Fig. 3).

To implement self-attention-based information [29] aggregation within the object graph, we connect node i to all other nodes in the same graph using both edge E_{self} in graph G_{t-1} and edge E_{self} within G_t. Additionally, node i in graph G_{t-1} undergoes cross-attention-based [12] information aggregation with nodes in between G_t by connecting to them. Information between nodes is aggregated through edges E_{self} and E_{cross}, and the representation of nodes is updated on each layer of the graph neural network [41].

In the aggregation process, the attention mechanism is utilized to account for the relationship between a node and its neighbors during information aggregation. The node aggregation formula is as follows:

$$H_i^l = \left[x_i^l \| \| m^l_{E \to i} \right]. \tag{1}$$

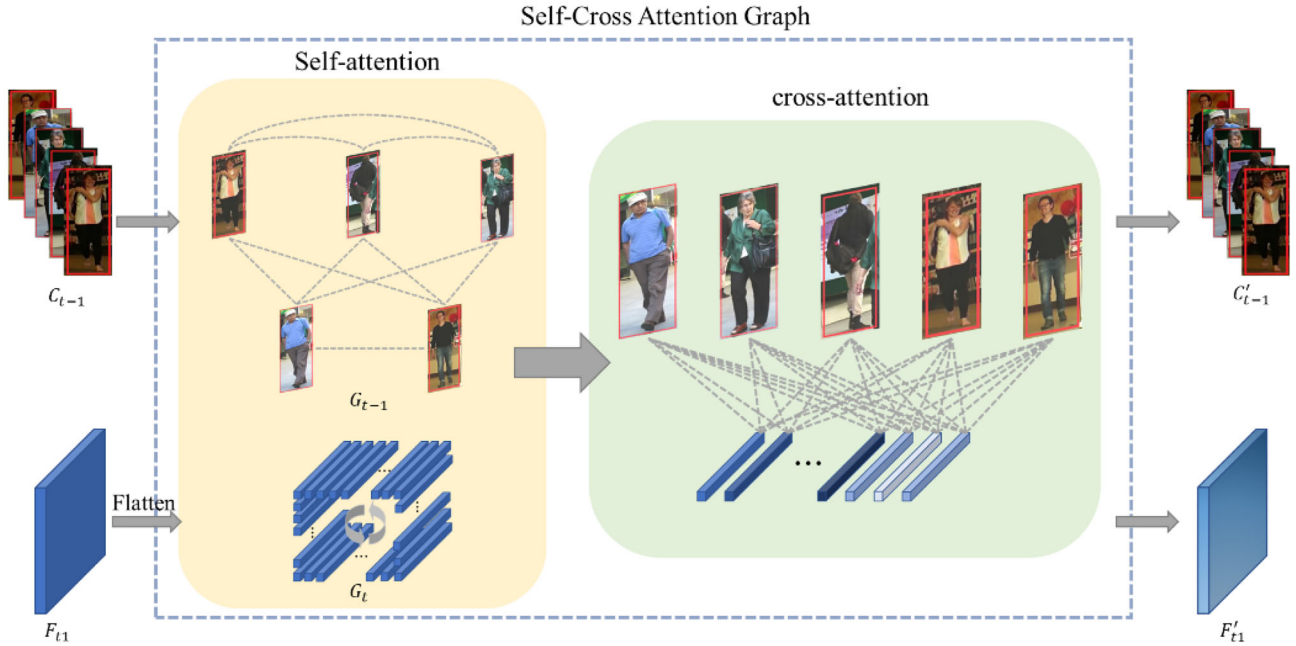

Fig. 3 Self-Cross Attention Graph. object graphs G_{t-1} and G_t constructed from object feature vectors at the moments I_{t-1} and I_t. First, intra-frame object information aggregation is performed, followed by inter-object graph information aggregation based on cross-attention [12]. Through this process, we obtain more expressive pedestrian target appearance features

Here, x_i^l represents node i in graph G_t, and l represents the number of layers in the graph neural network [41]. The node's information transfer, denoted as $m_{E\to i}^l$, is the result of aggregation from all nodes $\{j : (i, j) \in E\}$, where $E \in \{E_{\text{self}}, E_{\text{cross}}\}$. The notation $[\cdot||\cdot]$ indicates node information splicing.

We employ the attention mechanism to perform aggregation and compute node information transfer $m_{E\to i}$. To obtain the attention of other nodes towards node i, we compute the representation of node i as a query object q_i and retrieve the value v_j of certain nodes based on their properties (e.g., keyword k_j). Subsequently, we calculate a weighted average of this information to obtain the attention.

$$m_{E\to i}^l = \sum_{j:(i,j)\in E} \alpha_{i,j} v_j \tag{2}$$

The attention weight $\alpha_{i,j}$ between nodes i and j is obtained by querying the softmax of the similarity between object q_i and keyword k_j:

$$\alpha_{i,j} = Softmax\left(q_i^T k_j\right). \tag{3}$$

To calculate the query object q_i, keyword k_j, and value v_j, we use linear projection of the depth feature of the graph neural network [41]:

$$\begin{aligned} q_i &= W_1^l x_i^l + b_1^l, \\ k_j &= W_2^l x_j^l + b_2^l, \\ v_j &= W_3^l x_j^l + b_3^l \end{aligned} \tag{4}$$

To enhance the representational power of the model, we use a multi-headed attention mechanism with h attention heads in practice.

$$m_{E\to i}^l = W^l\left(m_{E\to i}^{l,1}||m_{E\to i}^{l,2}||\cdots||m_{E\to i}^{l,h}\right) \tag{5}$$

During the update process of the graph neural network [41], the neighborhood information H_i^l obtained from aggregation is utilized to update the features of the current node i:

$$x_i^{l+1} = x_i^l + MLP\left(H_i^l\right). \tag{6}$$

The Self-Cross Attention Graph incorporates two types of aggregation mechanisms: self-attention [29] and cross-attention [12]. In the self-attention mechanism, the input is G_t/G_{t-1}, which aggregates contextual information of nodes within the same feature map, enhancing the expressiveness of the node features in G_t/G_{t-1}. Simultaneously, this information is fed into the cross-attention mechanism [12] to enable interaction between nodes connected through continuous edges. The node information messages from G_{t-1} are utilized to strengthen the node features in G_t, resulting in improved outcomes during the data association phase.

Network loss

Our proposed network output comprises a detection task and a pedestrian target feature re-identification task. The learning task for target detection adheres to the loss function design of the target centroid-based detection network. We utilize cross-entropy loss [17] to compute the target centroid category L_{cls}. For calculating the target centroid offset loss L_{off} and the target area size loss L_{size}, we employ L_1 Loss [40].

The losses for the detection task are calculated as follows during testing:

$$L_{\text{det}} = L_{\text{cls}} + \lambda_{\text{off}} L_{\text{off}} + \lambda_{\text{size}} L_{\text{size}}, \tag{7}$$

where $\lambda_{\text{off}} = 1$ and $\lambda_{\text{size}} = 0.1$.

To learn and identify features with different identities in the Re-ID task, we treat it as a classification task. During training, we consider all objects in the dataset with the same identity ID as the same class and use their IDs as classification labels for the Re-ID task. To obtain the target center location (C_x^i, C_y^i) on the heat map, we use $b^i = (x_1^i, y_1^i, x_2^i, y_2^i)$. The Re-ID feature vector $E_{(C_x^i, C_y^i)}$ is extracted from the target centroid location (C_x^i, C_y^i) and mapped to the class distribution vector $P = \{p(k), k \in [1, k]\}$ using a fully connected layer and softmax operation. The classification labels for the Re-ID task are encoded using one-hot encoding $L^i(k)$. Re-ID losses are then computed as follows:

$$L_{\text{identity}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} L^i(k)\log(p(k)), \tag{8}$$

where K represents the number of IDs of all targets in the training set. To summarize, the overall loss L_{total} is calculated as follows:

$$L_{\text{total}} = \frac{1}{2}\left(\frac{1}{e^{w_1}} L_{\text{det}} + \frac{1}{e^{w_2}} L_{\text{identity}} + w_1 + w_2\right), \tag{9}$$

where w_1 and w_2 are learnable parameters that balance these two tasks.

Experiment

Training details & parameter settings

Datasets We conducted our experiments on the MOTchallenge, specifically utilizing the MOT 16 [19] and MOT 17 [19] pedestrian datasets. Both datasets consist of the same set of videos, with seven videos assigned for training and seven for testing. However, it is important to note that while MOT 16 provides only one detector, MOT 17 offers three detectors, namely Faster R-CNN [25] and SDP [9]. Additionally, we employed MOTSynth, a large-scale synthetic dataset designed to replace real data, for pedestrian detection, tracking, and segmentation. MOTSynth [7] encompasses a wide range of variations, including changes in environment, camera perspective, object texture, lighting conditions, weather, seasonal changes, and object identity. By leveraging this diversity, MOTSynth [7] aims to bridge the gap between synthetic and real data, enhancing the robustness and generalizability of our methods.

Table 3 Evaluation metrics for multi-object tracking algorithms

Evaluation metrics	Description of indicators
False negative	Number of real trajectories not predicted
False positive	Number of predicted trajectories that are not true trajectories
ID Switch	Number of times trajectory identifiers were exchanged
Mostly lost tracklets	Maximum 20 percent of trajectories predicted during tracking, i.e. number of almost lost trajectories
Mostly tracked tracklets	Number of tracked trajectories for which 80 percent of the trajectories were predicted for the tracking process
MOTA	Accuracy of multi-object tracking with bias detection effect
IDF1	The ratio of the number of correct ids to the sum of true ids and detected ids is a composite indicator of the accuracy of the biased ids

Evaluation metrics The MOT dataset not only offers data support for video sequences but also provides a range of related metrics for evaluating multi-object tracking algorithms comprehensively. These metrics [1] assess various aspects of performance, including detection and identity tracking. Table 3 presents the algorithm evaluation criteria and their descriptions provided by the MOT dataset.

In the context of algorithm research, it is essential to concentrate on specific metrics that align with targeted business requirements. When evaluating multi-target tracking algorithms, certain metrics offer particularly informative insights. These include MOTA, IDF1, IDSwitch, ML, and MT. MOTA (Multiple Object Tracking Accuracy) and IDF1 (ID F1 Score) are comprehensive metrics that provide a holistic assessment of algorithm performance. MOTA emphasizes detector performance, while IDF1 prioritizes accuracy in trajectory matching. The formulas for MOTA and IDF1 are as follows:

$$\text{MOTA} = 1 - \frac{\sum_t FN + FP + ID}{\sum_t gt}, \tag{10}$$

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

We conducted the experiments on Ubuntu 20.04 LTS and trained the model using GeForce RTX3090. To train the network on the MOT17 [19] dataset and accelerate the process, we first pre-trained on the CrowdHuman [26] dataset. This pre-training helped improve the human detection performance while providing strong domain generalization. The network takes inputs with an image resolution of 1088×608, is trained for 40 epochs, has an initial learning rate of 0.00001, and employs a batch size of 12. We applied a learning rate reduction by a factor of 10 after every 20 training cycles.

Experimental results and analysis

Experimental results and analysis of different modules To verify the impact of the cross-attention network CAN and the graph neural network [41] SCG on the multi-object tracking algorithm. We used the MOT20 [5] training set as the validation set and performed ablation experiments on it.

Table 4 displays the experimental results module on the validation set for the different modules. It is evident that when using only SCG, our method exhibits a slight improvement in the MOTA metric and a significant decrease in the FP metric. These outcomes indicate that SCG can effectively enhance the representation of appearance features of pedestrian targets

After incorporating only the Cross Attention Network (CAN), a notable enhancement in tracking performance and tracking persistence was observed. This observation suggests a genuine competition between the Re-ID task and the object detection task. To enhance the model's performance, we decoupled these two tasks. The best results on the validation set were achieved when all modules were added to the model.

Experimental results and analysis of the number of layers in the graph neural network We conducted an experiment to investigate the influence of the number of layers in the graph neural network [41] on the overall performance of the algorithm. To identify the most suitable number of Self-Constructing Graph (SCG) layers, we systematically increased the number of layers in the graph neural network [41]. In our study, we introduced two hyperparameters to control the depth of the graph neural network [41] specifically for the self-attention and cross-attention mechanisms [12]. The parameter l_s determines the depth of the GNN for self-attention [29], indicating the number of layers through which information is propagated and aggregated within each node's local neighborhood. Similarly, the parameter l_c controls the depth of the GNN for cross-attention.

Table 5 depicts the tracking results obtained by the SCG with varying numbers of layers on the MOT17 [19] validation set. Notably, when the number of SCG layers reaches 3, the tracking performance shows a decline compared to cases where fewer layers are utilized. This phenomenon can be attributed to the increased neighborhood aggregation of GNN nodes, causing the loss of node diversity within the graph. Consequently, the vector representations become more similar, resulting in smoother node features. In comparison to using only one SCG layer, employing two SCG layers yields the best results across various indicators. This observation can be explained by the fact that when the number of graph neural network layers is insufficient, the information propagation path becomes limited, impeding the network's ability to capture long-range relationships and contextual information between nodes. Consequently, the network may struggle to capture global patterns and structures within the graph data. For the final algorithm configuration, we opt for a two-layer graph neural network with $l_s = 1$ and $l_c = 1$

Experimental results and analysis of different features To assess the effectiveness of augmenting object features, our study compares the use of solely object appearance features to the inclusion of object location information.

Table 6 illustrates a decrease of 0.5% in both MOTA and IDF1 when incorporating Gemo., suggesting that adding location information may introduce similar target location characteristics, potentially causing tracking algorithm errors by incorrectly associating distinct targets as a single target. In light of this observation, we chose to exclusively use the appearance features of the objects

Experimental results and analysis of the number of layers of the graph neural network We investigate the influence of the number of multi-head attention mechanisms on the overall performance of the algorithm.

We conducted ablation experiments, varying the number of attention heads, denoted as "h ". The results, presented in Table 7, demonstrate the performance of the model under different configurations.

Interestingly, we observed that the best overall metrics were achieved when the number of attention heads was set to 3, with the exception of the FP metric. This can be attributed to the fact that each attention head focuses on different subspaces of features, allowing for a more comprehensive understanding of the data. However, when the number of attention heads is 4, the model may overly emphasize noise or less significant features in the training data, leading to reduced generalization ability. On the other hand, a smaller number of attention heads may limit the model's capacity to explore the diversity present in the data.

Table 4 Experimental results of different modules on the verification set

SCG	CAN	MOTA (%)↑	FP ↓	FN ↓	IDs ↓	MT ↑	ML ↓	IDF1 (%)↑
		39.6	23,798	77,120	6228	321	778	**41.8**
√		39.8	**21,576**	775,899	7115	281	804	39.7
	√	**40.5**	21,841	767,858	6255	327	**766**	40.9
√	√	**40.5**	22,844	**767,048**	**6030**	**335**	847	**41.8**

Bold values represent the best result or a secondary result

Table 5 Number of self-attention layers and cross-attention layers

Layers	MOTA (%)↑	FP ↓	FN ↓	IDs ↓	MT ↑	ML ↓	IDF1 (%)↑
$l_s = 0, l_c = 0$	77.1	19,149	56,418	1656	972	114	79.1
$l_s = 1, l_c = 0$	78.2	16,797	55,362	1278	984	115	79.2
$l_s = 0, l_c = 1$	78.6	17,130	53,709	1389	1002	123	81.0
$l_s = 1, l_c = 1$	78.7	18,000	52,536	1206	1008	117	81.2
$l_s = 1, l_c = 2$	77.9	18,615	54,633	1371	996	117	80.5
$l_s = 2, l_c = 1$	78.1	17,550	54,783	1395	999	114	80.7

Table 6 Ablation study on the effect of using geometric features during affinity computation

Appear	Geom	MOTA (%)↑	FP ↓	FN ↓	IDs ↓	MT (%)↑	ML (%)↓	IDF1 (%)↑
√		**73.0**	**23,772**	125,442	2997	**43.0**	**17.6**	**73.2**
√	√	72.5	28,110	**124,347**	**2799**	41.4	19.9	72.7

Bold values represent the best result or a secondary result

Table 7 Number of different multi-head attentions in results on MOT17 validation set

h	MOTA (%)↑	FP↓	FN↓	IDs↓	MT↑	ML↓	IDF1 (%)↑
1	76.6%	19,890	56,940	1611	948	129	78.8%
2	86.5%	7188	36,630	1815	1125	87	85.2%
3	87.1%	7479	34,251	1758	1173	75	86.0%
4	85.1%	8403	39,720	1932	1059	87	82.9%

Table 8 Comparison of efficiency between SCGTracker and FairMOT

	FLOPs (G)↓	Parameters (M)↓	FPS ↑
Fairmot	87.5	16.5	15.99
SCGTracker	88.1	16.7	14.57

We choose the optimal number of attention heads, in this case, 3, allows for a balance between capturing relevant features and avoiding overfitting or underutilization of important information, resulting in improved model performance and generalization ability.

In terms of computational and parameter requirements (as illustrated in Table 8), our model exhibits a slight increase compared to FairMOT [39]. This is primarily due to our algorithm's focus on addressing the issue of frequent ID switching among high-density pedestrians. The attention module we have devised involves calculating and modeling correlations between multiple elements, resulting in higher computational complexity. However, the attention mechanism also enhances the model's modeling and representation capabilities, despite typically having a relatively small number of parameters. When evaluating the frames per second (FPS) on the MOT17 [19] dataset, SCGTracker demonstrates competitive performance while simultaneously improving tracking accuracy.

The loss diagram of MOT17 [19] is illustrated in Fig. 4. The metric train_hm_loss reflects the detection loss, while train_id_loss pertains to the loss of features. On the other hand, train_loss represents the overall or total loss. Notably, when the epoch reaches 15, the curve of train_id_loss starts to exhibit a gradual smoothness. In contrast, train_hm_loss continues to display a downward trend at 15 epochs, but eventually stabilizes around 29 epochs. It is worth mentioning that train_loss reaches a plateau by epoch 29, indicating a

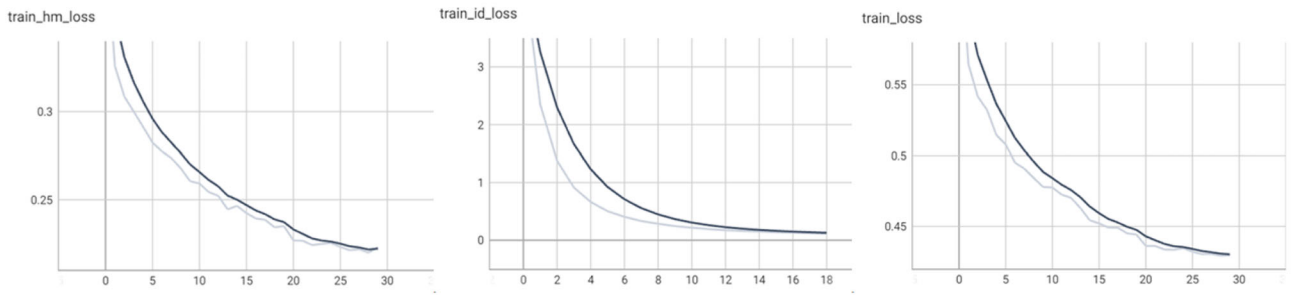

Fig. 4 The loss chart of MOT17 [19]

Table 9 Comparison of ours and other algorithms on MOT16 test set

Algorithms	MOTA (%) ↑	FP ↓	FN ↓	IDs ↓	MT ↑	ML ↓	IDF1 ↑
QDTrack [22]	69.8	9861	44,050	1097	41.6	19.8	67.1
TraDes [34]	70.1	8091	45,210	1144	37.3	20.0	64.7
KDMOT [38]	74.3	–	–	797	40.4	17.6	74.7
CSTrack [15]	70.7	10,286	41,974	1071	38.2	17.8	71.8
GSDT [30]	74.5	8913	36,428	1229	41.2	17.3	68.1
FairMOT [39]	74.9	–	–	1074	44.7	15.9	72.8
SCGTracker (ours)	74.6	10,589	30,538	1021	44.3	16.1	73.1

Table 10 Comparison of ours and other algorithms on MOT17 test set

Algorithms	MOTA (%) ↑	FP ↓	FN ↓	IDs ↓	MT ↑	ML ↓	IDF1 ↑
CTracker [11]	66.6	22,284	160,491	5529	32.2	24.2	71.6
SGT [10]	76.4	25,974	102,885	4101	48.0	11.7	72.8
GSDT [30]	73.2	26,397	120,666	3891	41.7	17.5	66.5
CenterTrack [42]	67.8	18,498	160,332	3039	34.6	24.6	64.7
TraDes [34]	69.1	20,892	150,060	3555	36.4	21.5	63.9
TrackFormer [18]	74.1	34,602	108,777	2829	–	–	68
MOTR [37]	73.4	–	–	2439	–	–	68.6
FairMOT [39]	73.7	27,507	117,477	3303	43.2	17.3	72.3
SCGTracker (ours)	73.0	23,772	125,442	2997	43.0	17.6	73.2

diminishing improvement in the overall loss. Consequently, the training process is halted at epoch 26 to prevent further training iterations that would yield minimal gains.

Comparison with other algorithms: To demonstrate the state-of-the-art performance of our algorithm on the MOT challenge dataset, we conducted a comparative analysis with other top-performing tracking algorithms. By examining Tables 9 and 10, it becomes apparent that our algorithm does not exhibit a substantial improvement in the MOTA metric for the MOT16 [19] and MOT17 [19] datasets. This is primarily attributed to the fact that our algorithm primarily focuses on addressing the issue of ID switching in high-density pedestrian datasets.

One crucial measure to evaluate the effectiveness of our approach is the IDF1 metric, which quantifies the number of instances where the track ID number differs from the initial track ID number. Additionally, the IDs metric indicates the frequency of trajectory identity exchanges. It is worth noting that our method achieved the best results in terms of IDF1 and IDs, indicating that it effectively alleviates the problem of pedestrian ID switching in dense scenes.

While the improvement in the MOTA metric may not be substantial, the exceptional performance in IDF1 and IDs demonstrates the efficacy of our method in mitigating the challenges associated with ID switching in crowded pedestrian scenarios. This showcases the unique contribution and value of our approach in addressing this specific problem, even if it does not lead to a significant improvement in overall MOTA performance (The best results are highlighted in red, and the second-best results are highlighted in blue).

To assess the robustness of our algorithm, we conducted an experiment on the MOTSynth [7] dataset, comparing it

Table 11 Comparison of ours and other algorithms on MOTSynth test set

Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDs ↓
GNMOT [14]	44.7	44.1	299	586	**11,445**	169,662	4997
GCNNMatch [23]	46.8	58.2	459	348	50,979	125,778	**2499**
MPNTracker [3]	49.6	59.4	474	303	45,414	121,893	2571
GMTracker [8]	51.7	61.3	519	303	44,232	115,755	2583
GSDT [30]	**60.1**	**62.9**	**558**	**291**	**23,697**	**108,117**	2676
SCGTracker (ours)	**57.5**	**64.3**	**579**	**288**	33,177	**107,838**	**2229**

Bold values represent the best result or a secondary result

with existing graph neural network-based multi-target tracking algorithms. The results of this experiment are presented in Table 11, alongside the findings from other relevant papers.

Upon analyzing Table 11, we observe that while our algorithm does not achieve the highest scores in terms of MOTA indicators, it performs on par with other methods in terms of MOTA, ML, MT, and IDs. Here, MT represents the proportion of ground-truth trajectories covered by track hypotheses that overlap with at least 80% of their respective ground-truth trajectory, while ML represents the proportion of ground-truth trajectories covered by track hypotheses that overlap with up to 20% of their respective ground-truth trajectory.

This suggests that our algorithm effectively obtains discriminative target features, contributing to a reduction in the number of target ID switches. This ability to capture strongly discriminative target features is a notable strength of our algorithm, contributing to its robustness in multi-target tracking scenarios.

Visualization results

As depicted in Fig. 5, three distinct trace scenarios were chosen for effective display plots from the test set. The first row of the figure illustrates the detection effect graph without utilizing the decoupling module (CAN), while the second row demonstrates the effect of decoupling with the Cross Attention Network (CAN).

As depicted in the detection effect plots, the response area of the pedestrian targets enclosed within the red circles in each image of the first row is notably smaller than that of the second row. This observation suggests that the competition between the detection task and the Re-ID task has a substantial impact, not only on the detection task but also on the tracking task. Hence, it is imperative to decouple them using a cross-attention network.

The tracking performance of our method on the MOT17 test set is illustrated in Fig. 6. Each row of the figure corresponds to a video sequence from the MOT17 test set, and each column from left to right represents our tracking results every 30 frames.

Discussion

The SCGTracker is an online algorithm designed for end-to-end multi-target tracking. It leverages an attention mechanism to aggregate information surrounding the targets. Additionally, it employs message passing to interact with target feature information, thereby identifying highly discriminative characteristics. However, there are some notable drawbacks that need to be addressed. First, the performance in target detection falls short of expectations, as evidenced by unsatisfactory results obtained from the MOTA indicator. As previously discussed, our method primarily focuses on enhancing target features while disregarding the crucial data association module. Second, the SCGTracker fails to fully exploit the positional information of the targets. Our experiments reveal that incorporating the positional information at the pixel level in the current frame may introduce similar target position features, resulting in errors within the tracking algorithm. Resolving these aforementioned issues constitutes a significant research area within the context of the MOT framework based on graph neural networks.

Conclusion

We conducted a literature review on the application of Graph Neural Networks (GNNs) [41] for enhancing target re-identification (Re-Id). Our findings reveal that existing algorithms often overlook the interdependencies among targets within the same frame. Additionally, when facing occlusion, the features of the detected target can unintentionally compromise the high-quality features of the trajectory

Fig. 5 Impact of CAN on detection tasks

Fig. 6 Tracking effects on the MOT17 test set

target. To address this issue, our paper introduces the construction of object graphs for each frame and between consecutive frames. We leverage the self-attention mechanism to aggregate target features within the same frame and employ cross-attention to gather information from pedestrian targets in two consecutive frames, effectively capturing their correlations. The target features are then updated using a Graph Neural Network [41]. Experimental evaluations on the MOT17 dataset demonstrate that our proposed method is highly competitive compared to state-of-the-art tracking methods. In fact, it achieves comparable or superior results across almost all evaluation metrics.

Acknowledgements This work is partially supported by the Key Project of Chongqing Technology Innovation and Application Development under Grant No.cstc2021jscx-dxwtBX0018, Natural Science Foundation of Chongqing, China under Grant No.CSTB2022NSCQ-MSX0493, Chongqing Postgraduate Scientific Research Innovation Project and the Action Plan for the High-quality Development of Postgraduate Education of Chongqing University of Technology (Grant No. gzlcx20222062, No. gzlcx20233218).

Data availability All data that support the findings of this study are included in this manuscript, and data available on request from the authors.

Declarations

Conflict of interest There is no conflict of interest.

References

1. Bernardin K, Stiefelhagen R (2008) Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP J Image Video Process 2008:1–10. https://doi.org/10.1007/s40747-020-00206-8
2. Bewley A, Ge Z, Ott L, et al (2016) Simple online and real-time tracking. In: 2016 IEEE International Conference on image processing (ICIP), pp. 3464–3468. https://doi.org/10.1109/ICIP.2016.7533003
3. Bras 'o G, Leal-Taix'e L (2020) Learning a neural solver for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, 2020, pp 6247–6257. https://doi.org/10.48550/arXiv.1912.07515.
4. 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, 2023, pp 4870–4880. https://doi.org/10.48550/arXiv.2104.00194
5. Dendorfer P, Rezatofighi H, Milan A, et al (2020) Mot20: A benchmark for multi-object tracking in crowded scenes. arXiv preprint arXiv:2003.09003
6. Duan K, Bai S, Xie L, et al (2019) Centernet: keypoint triplets for object detection. In: Proceedings of the IEEE/CVF International Conference on computer vision, 2019, pp 6569–6578. https://doi.org/10.1109/ICCV.2019.00667.
7. Fabbri M, Brasó G, Maugeri G, et al (2021) Motsynth: How can synthetic data help pedestrian detection and tracking? In: Proceedings of the IEEE/CVF International Conference on computer vision. 2021, pp 10849–10859. https://doi.org/10.1109/ICCV48922.2021.01067.
8. He J, Huang Z, Wang N, et al (2021) Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, 2021, pp pp 5299–5309. https://doi.org/10.48550/arXiv.2103.16178.
9. Hezaveh YD, Dalal N, Marrone DP et al (2016) Detection of lensing substructure using ALMA observations of the dusty galaxy SDP.81. Astrophys J 823(1):37. https://doi.org/10.1088/0004-637X/823/1/37
10. Hyun J, Kang M, Wee D, et al (2023) Detection recovery in online multi-object tracking with sparse graph tracker. In: Proceedings of the IEEE/CVF Winter Conference on applications of computer vision, 2023, pp 4850–4859. https://doi.org/10.48550/arXiv.2205.00968.
11. Jiang H, Wang M, Liu D et al (2021) Ctrack: acoustic device-free and collaborative hands motion tracking on smartphones. IEEE Internet Things J 8(19):14658–14671. https://doi.org/10.1007/s40747-020-00206-8
12. Ke L, Li X, Danelljan M, et al (2021) Prototypical cross-attention networks for multiple object tracking and segmentation. In: Advances in neural information processing systems 34, pp 1192–1203. https://doi.org/10.48550/arXiv.2106.11958.
13. Lee J, Jeong M, Ko BC (2021) Graph convolution neural network-based data association for online multi-object tracking. IEEE Access 9:114535–114546. https://doi.org/10.1109/ACCESS.2021.3105118
14. Li J, Gao X, Jiang T (2020) Graph networks for multiple object tracking. In: Proceedings of the IEEE/CVF Winter Conference on applications of computer vision, pp 719–728. IEEE. https://doi.org/10.1109/WACV45572.2020.9093347
15. Liang C, Zhang Z, Zhou X et al (2022) Rethinking the competition between detection and reid in multiobject tracking. IEEE Trans Image Process 31:3182–3196. https://doi.org/10.48550/arXiv.2010.12138
16. Liang H, Song H, Yun X et al (2022) Traffic incident detection based on a global trajectory spatiotemporal map. Complex Intell Syst. https://doi.org/10.1007/s40747-020-00206-8
17. Martinez M, Stiefelhagen R (2019) Taming the cross entropy loss. In: Pattern Recognition: 40th German Conference, GCPR 2018, Proceedings 40, pp 628–637. Springer. https://doi.org/10.1007/978-3-030-12939-2 43
18. Meinhardt, T., Kirillov, A., Leal-Taixe, L., et al. (2022). Trackformer: Multiobject tracking with transformers. In Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, pp 8844–8854. IEEE. https://doi.org/10.48550/arXiv.2101.02702
19. Milan A, Leal-Taixe L, Reid I, et al (2016) Mot16: a benchmark for multiobject tracking. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (pp. 807–823). Springer. https://doi.org/10.1007/978-3-319-46478-7 40
20. Mills-Tettey GA, Stentz A, Dias MB (2007) The dynamic Hungarian algorithm for the assignment problem with changing costs. Robotics Institute, Pittsburgh, PA, Tech Rep CMU-RI-TR-07-27. https://doi.org/10.1177/0278364911404579
21. Ning C, Menglu L, Hao Y et al (2021) Survey of pedestrian detection with occlusion. Complex Intell Syst 7:577–587. https://doi.org/10.1007/s40747-020-00206-8
22. Pang J, Qiu L, Li X, et al (2021) Quasi-dense similarity learning for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, pp 164–173. IEEE. https://doi.org/10.48550/arXiv.2006.06664
23. Papakis I, Sarkar A, Karpatne A (2020) Gcnnmatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization. arXiv preprint arXiv:201000067. https://doi.org/10.1007/s40747-020-00206-8
24. Rangesh A, Maheshwari P, Gebre M, et al (2021) TrackMPNN: a message passing graph neural architecture for multi-object tracking. arXiv preprint arXiv:2101.04206.
25. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE International Conference on computer vision, pp 1440–1448. https://doi.org/10.1109/ICCV.2015.169.
26. Shao S, Zhao Z, Li B, et al (2018) CrowdHuman: a benchmark for detecting humans in a crowd. arXiv preprint arXiv:1805.00123. https://doi.org/10.48550/arXiv.1805.00123
27. Sun S, Akhtar N, Song H et al (2019) Deep affinity network for multiple object tracking. IEEE Trans Pattern Anal Mach Intell 43(1):104–119. https://doi.org/10.1109/TPAMI.2019.2896507

28. Tang S, Andriluka M, Andres B, et al (2017) Multiple people tracking by lifted multicut and person re-identification. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 3539–3548. https://doi.org/10.1109/CVPR.2017.394
29. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In: Advances in Neural Information Processing Systems 30, pp 5998–6008. https://doi.org/10.5555/3295222.3295349
30. Wang Y, Kitani K, Weng X (2021) Joint object detection and multi-object tracking with graph neural networks. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 13708–13715. https://doi.org/10.1109/ICRA.2021.9553818
31. Wang Z, Zheng L, Liu Y, et al (2020) Towards real-time multi-object tracking. In: Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, Springer, pp 107–122. https://doi.org/10.48550/arXiv.1909.12605
32. Willner D, Chang C, Dunn K (1976) Kalman filter algorithms for a multi-sensor system. In: 1976 IEEE Conference on Decision and Control including the 15th Symposium on adaptive processes, IEEE, pp 570–574. https://doi.org/10.1109/CDC.1976.267794
33. 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
34. Wu J, Cao J, Song L, et al (2021) Track to detect and segment: An online multi-object tracker. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 12352–12361. https://doi.org/10.1109/CVPR42934.2021.01253
35. Yu F, Wang D, Shelhamer E, et al (2018) Deep layer aggregation. In: Proceedings of the IEEE Conference on computer vision and pattern recognition pp 2403–2412. https://doi.org/10.48550/arXiv.1707.06484
36. Zaech JN, Liniger A, Dai D et al (2022) Learnable online graph representations for 3D multi-object tracking. IEEE Robot Autom Lett 7(2):5103–5110. https://doi.org/10.48550/arXiv.2104.11747
37. Zeng F, Dong B, Zhang Y, et al (2022) MOTR: End-to-end multiple-object tracking with transformer. In: European Conference on computer vision, pp 659–675. Springer. https://doi.org/10.48550/arXiv.2105.03247
38. Zhang W, He L, Chen P, et al (2021) Boosting end-to-end multi-object tracking and person search via knowledge distillation. In: Proceedings of the 29th ACM International Conference on multimedia, pp 1192–1201. ACM. https://doi.org/10.1145/3474085.3481546
39. Zhang Y, Wang C, Wang X et al (2021) Fairmot: On the fairness of detection and re-identification in multiple object tracking. Int J Comput Vision 129:3069–3087. https://doi.org/10.1007/s40747-020-00206-8
40. Zhao H, Gallo O, Frosio I et al (2016) Loss functions for image restoration with neural networks. EEE Trans Comput Imaging 3(1):47–57. https://doi.org/10.1109/TCI.2016.2644865
41. Zhou J, Cui G, Hu S et al (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81. https://doi.org/10.48550/arXiv.1812.08434
42. Zhou X, Koltun V, Krahenbiihl P (2020) Tracking objects as points. In: Proceedings of the 16th European Conference on computer vision (ECCV2020), Glasgow, UK, August 23–28, 2020, Part IV, pp 474–490. Springer. https://doi.org/10.48550/arXiv.2004.01177
43. Zhu X, Lyu S, Wang X, et al (2021) Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. In: Proceedings of the IEEE/CVF International Conference on computer vision, pp 2778–2788. https://doi.org/10.1109/ICCVW54120.2021.00312