Enhancing the association in multi-object tracking via neighbor graph

Tianyi Liang\textsuperscript{1,2} | Long Lan\textsuperscript{3} | Xiang Zhang\textsuperscript{3} | Xindong Peng\textsuperscript{1} | Zhigang Luo\textsuperscript{4}

\textsuperscript{1}College of Computer, National University of Defense Technology, Changsha, China
\textsuperscript{2}Shandong Sport University, Jinan, China
\textsuperscript{3}Institute for Quantum Information & State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, China
\textsuperscript{4}Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha, China

Abstract
Most modern multi-object tracking (MOT) systems for videos follow the tracking-by-detection paradigm, where objects of interest are first located in each frame then associated correspondingly to form their intact trajectories. In this setting, the appearance features of objects usually provide the most important cues for data association, but it is very susceptible to occlusions, illumination variations, and inaccurate detections, thus easily resulting in incorrect trajectories. To address this issue, in this study we propose to make full use of the neighboring information. Our motivations derive from the observations that people tend to move in a group. As such, when an individual target's appearance is remarkably changed, the observer can still identify it with its neighbor context. To model the contextual information from neighbors, we first utilize the spatiotemporal relations among trajectories to efficiently select suitable neighbors for targets. Subsequently, we construct neighbor graph for each target and corresponding neighbors then employ the graph convolutional networks (GCNs) to model their relations and learn the graph features. To the best of our knowledge, it is the first time to explicitly leverage neighbor cues via GCN in MOT. Finally, standardized evaluations on the MOT16 and MOT17 data sets demonstrate that our
approach can remarkably reduce the identity switches whilst achieve state-of-the-art overall performance.

KEYWORDS
data association, graph convolutional networks, multi-object tracking

1 | INTRODUCTION

Multi-object tracking (MOT) aims to predict the trajectories of all concerned objects in video sequences. It has been a long-standing research topic in computer vision since many applications, such as video surveillance, autonomous driving, and sport event analysis, rely on it. In recent years, due to the advances of high-performance object detection, trackers following the tracking-by-detection paradigm made remarkable progress and dominate this community. Nonetheless, tracking multiple objects accurately in complex real-world scenes is still very challenging.

The basic pipeline of tracking-by-detection is first localizing objects of interest in each video frame and then associating them with certain metrics to form the trajectories. Under the online tracking protocol, this pipeline can be concisely defined as associating detection responses in the current frame to existing trajectories. To this end, most state-of-the-art trackers adopt the re-identification (ReID) model to extract the individual appearance features (embeddings) of each detection and take them as cues for data association. Due to the rapid progress of the deep ReID technique\cite{1,2}, the ReID-based feature seems to be robust in most cases. However, it should not be ignored that this scheme runs upon an important prerequisite, that is, the bounding box of detection should be accurate enough. Once inaccurate, distractions will be brought into the feature extraction, resulting in the error associations. More importantly, in real-world scenes the frequent occlusions, illumination variations, and cluttered backgrounds often dramatically change the appearance of targets, which makes the association based on individual appearance features difficult to correctly infer. In summary, the individual appearance features are very susceptible to negative factors, such as occlusions, illumination variations, and inaccurate bounding boxes.

So how to overcome the intrinsic defects of individual features and make the association more robust? A feasible idea is to treat targets at a group level and utilize the group information to improve tracking. The early attempts\cite{3,4} borrow some ideas from group tracking\cite{5,6}, where they model the tracked objects in mutual context with each other to benefit robust tracking. Nonetheless, as works\cite{3,4} were proposed in the predeep learning era, the hand-engineered methods they used to model the mutual relations are not powerful and flexible enough. Recent works\cite{7,8} revisit the exploration of group-level information and introduce the data-driven approaches to achieve better performance, but obvious limitations still exist in their solution. Concretely, the work\cite{7} is computationally expensive because it focuses on the pixel-level neighboring relations. Another work\cite{7} simply aggregates the neighbor context via MultiLayer Perceptron (MLP), which is not fine-grained enough to capture complex interactions within the neighbor context. In fact, the similar idea, termed collaborative filtering, is widely accepted in the research of recommender system\cite{9}, where proper neighbors are first grouped then utilized to estimate the target user’s preference. Here we argue that the philosophy of collaborative filtering can be extended to enhance the association in the tracking-by-detection paradigm.
In this study, inspired by the philosophy of collaborative filtering, we propose to enhance the association through making full use of the neighboring information, rather than solely focusing on the individual features. Our main idea is concisely shown in Figure 1: for the tracking of multiple pedestrians, although the target pedestrian’s appearance at current frame is seriously changed by occlusions, we still can associate it with the correct trajectory since the neighboring pedestrians provide important complementary information. Here we term the detection and trajectory waiting to be matched as the targets for convenience. This motivation also conforms with the observation that people tend to walk in a group, and the entire group is relatively stable and consistent in a long term. As such, when a part of the group suffers from occlusions, illumination variations, or inaccurate bounding boxes, we can identify them with help of their neighbors in the group. In other words, the features learned from groups are more powerful than the individuals. The neighboring tracked objects become a kind of attribute of the target to support the association.

Nonetheless, how to fuse the features of targets and neighbors is nontrivial. In addition, in the MOT task, a large number of intraclass objects (e.g., pedestrians) often simultaneously appear while most of them are noises for the targets. Therefore, it is necessary to efficiently filter the noises and select the most suitable neighbors. To achieve the two goals, we first design a spatiotemporal relation-based strategy to select neighbors for the targets. This strategy only uses the spatiotemporal information generated by the tracking itself thus is lightweight and efficient. Then, with the selected neighbors, we construct a pair of neighbor graphs for the target detection and trajectory. Specifically, the nodes of a neighbor graph are the target (detection or trajectory) and its neighbors, where the neighbor nodes link with the target node. With that, we employ the graph convolutional networks (GCNs) to extract the graph embeddings then use them for data association. Compared with the prior works, our approach can get better embeddings since the proposed GCN-based extractor is more powerful than the hand-engineered methods and MLP to capture the complex relations in a graph. Moreover, as our framework explicitly models the neighbor relations on target level rather than pixel level, it is thus more computationally efficient than work.

We evaluate our approach on the most widely used MOT Challenge benchmark. It achieves state-of-the-art overall performance on both MOT16 and MOT17 data sets,
wherein the score of identity switches (IDs) gets noticeable improvements. In the rest of this paper, we first review the relevant works in Section 2, then we formulate our problem in Section 3.1 and present the contributions, that is, the spatiotemporal relation-based neighbor selection and neighbor graph learning in Sections 3.2 and 3.3, respectively. The proposed neighbor selection method only relies on the spatiotemporal cues generated by the tracking itself, thus is pragmatic and strikes a good balance between accuracy and speed. The neighbor graph learning enhances the association via jointly considering the target and its neighboring information. To the best of our knowledge, we are the first to exploit neighboring target cues via GCN in MOT. We finally discuss the experiment setup, results, and ablative studies in Section 4 and give conclusions in Section 5.

2 | RELATED WORK

As our contributions involve data association, ReID-based feature learning, GCN, and group tracking, we thus briefly review related works in these areas.

2.1 | Data association

The tracking-by-detection framework consists of two components: an object detector to localize all the objects of interest and a data association model to form the trajectories. Despite the two parts are equivalently crucial, most MOT works mainly concentrate on the data association because object detection is a separated research direction. Specifically, the association methods can be broadly classified into two categories: batch mode and online mode. The batch mode simultaneously considers a long-time range of frames to output the final trajectories at once, therefore it runs offline and can be treated as a global optimization problem. A variety of global optimization algorithms, such as graph segmentation13 and Markov random field,14 has been applied in this setting. On the contrary, the online mode only focuses the association in two adjacent frames, which can be solved by the bipartite matching algorithms, like, the Hungarian algorithm.15 Compared with the batch mode association, the online mode is more challenging since it cannot utilize the future frames to handle occlusions or detection missing, but most recent trackers dedicate to tracking online because this mode is much closer to the human-like ability. On the other hand, existing data associations in both batch and online modes only consider the individual features of targets while ignoring its neighboring information. Our method fills this gap and improves the performance. Besides, in this study our method is implemented and tested in an online setting, but it can also be easily combined with the batch mode.

2.2 | ReID-based appearance model

Due to the complex situations in real-world scenes, multiple cues, including appearances,16 motions,17,18 and interactions,19 are jointly exploited to distinguish and re-identify targets. Among all of these, the appearance cues are most widely studied because the motions and interactions are hard to predict under long-term intra-object occlusions. To extract discriminative appearance features, most modern MOT trackers adopt the deep ReID model as the
feature extractor. For example, the DeepSORT tracker\textsuperscript{16} employs a Resnet\textsuperscript{20}-based ReID model to learn a discriminative appearance embedding and therefore benefits the affinity measurement. Works\textsuperscript{13,19} adopt the attention mechanism to refine the appearance embedding by reducing the noise from the background. Very recently some works\textsuperscript{21,22} unified the detection and ReID embedding models, which is named Joint-Detection-Embedding (JDE), with respect to the Separate-Detection-Embedding (SDE). The JDE models append an embedding head on the detector heatmap then perform detecting and feature embedding simultaneously, leading to the real-time inference speed without too much performance sacrifice. Nonetheless, no matter SDE or JDE methods, existing works only focus the individual feature learning while ignoring the neighboring information, thus they are very susceptible to harmful factors, such as occlusions and illumination variations. Instead, our solution incorporates the neighboring information to obtain more robust appearance features. Note that in this study our model follows the JDE framework but can also be combined with the SDE.

2.3 | Graph convolutional networks in MOT

GCNs\textsuperscript{10} extend deep convolutional networks to process graph data, which can solve the graph-based tasks\textsuperscript{23,24} with an end-to-end manner. It has been applied in computer vision tasks, including semantic segmentation,\textsuperscript{25} action recognition,\textsuperscript{26} and single object tracking.\textsuperscript{27} For MOT, prior works\textsuperscript{28,29} adopt GCN to achieve the end-to-end learning. They first extract targets’ appearance and motion features via CNN and LSTM, respectively, then fusing them and make association inference through GCN. As GCN is differentiable, their entire frameworks can thus be trained in the end-to-end style. The main drawback of these works\textsuperscript{28,29} is they still rely on the individual features while overlooking the neighboring information. By contrast, we leverage the neighbor relations via GCN so as to enhance the data association. Wang et al.\textsuperscript{7} employ GCN to model the interactions between targets thereby making the learned features more discriminative, but we think they indeed tend to the pixel-level interaction modeling on the heatmap while our scheme focuses on the image patch-level (target) interactions and thus is more efficient. Graph similarity model (GSM)\textsuperscript{8} also introduce the graph representation concept to the data association like us, but they do not use GCN to capture the representation, resulting in fewer performance gains than us.

2.4 | Group tracking

Similar to our work, the group tracking task also uses group-level information to improve tracking, but it treats groups as the tracked units rather than individual objects. For example, works\textsuperscript{3,30} use the social force model which is based on motion directions to detect and track groups. Solera et al.\textsuperscript{6} locate groups using both motion and visual cues then associate the groups via correlation clustering. All these works follow the batch mode and are, therefore, inappropriate for the time-critical applications. Perhaps the works\textsuperscript{3,4} are mostly related to our scheme where the group cues are utilized to improve the tracking of multiple individuals, but for the work,\textsuperscript{4} it performs tracking through simultaneously using numbers of single object trackers, therefore is less efficient than the tracking-by-detection approach. For the work,\textsuperscript{3} it merely adopts the motions cues to model the interactions and run at a batch mode. Besides, the group relations in References [3,4] are modeled by the hand-engineered features, which are
rigid and need online adaption. In contrast, our scheme exploits relations between objects through GCN, obtains more informative relation representations, and requires no finetuning during tracking. Castellano et al.\textsuperscript{31} also design a graph-based approach for detecting and tracking people. The main difference between their and our works is that they aim to predict the clustering tendency of people through the spatial graph, while we focus on improving data association with graph convolution theories.

3  |  APPROACH

In this section we present our approach in detail. We first give the mathematical formulation of our scheme in Section 3.1, then we describe how to select suitable neighbors for the targets (Section 3.2) and introduce the methods of building the neighbor graph and learning graph features via GCN (Section 3.3). Finally, we show the procedures of make data association using the neighbor graph features (Section 3.4). The entire framework is depicted in Figure 2.

3.1  |  Problem formulation

Given a detection response $d_i^t$ and a tracklet $T_k^{t-n}$, where $d_i^t$ denotes the four-dimensional (4D) coordinates of the bounding box of the $i$th detection at frame $t$ and $T_k^{t-n}$ denotes the tracklet of a target with identity $k$ which is last tracked at frame $t - n$ ($0 < n < t$). Let $T_k^{t-n} = \{d_p^k\}$, where $d_p^k$ is the $p$th bounding box of tracklet $T_k^{t-n}$ and $t_k \leq t - n$. On the basis of these, we can model the matching probability between $d_i^t$ and $T_k^{t-n}$ as $P(k|d_i^t, T_k^{t-n})$. Since most trackers adopt the appearance and motions cues to measure the similarity between detections and tracklets thereby reason about the data association, we can thus formulate the tracking problem as

$$P(k|d_i^t, T_k^{t-n}) \propto \alpha \text{Cosine}(\varphi(d_i^t), \varphi(T_k^{t-n})) + \beta \text{IoU}(d_i^t, T_k^{t-n}),$$

(1)

**FIGURE 2** The pipeline of our framework. We implement our framework in the Joint-Detection-Embedding style. The backbone first outputs the heatmap of the input image with down-sample 4. Then the detection and embedding heads localize the objects and extract their features. With these, we perform neighbor selection and build neighbor graph for the targets. Finally, we learn the graph features via GCN, and use them to make associations. GCN, graph convolutional network; ReID, re-identification [Color figure can be viewed at wileyonlinelibrary.com]
where $\phi(d_i^t)$ and $\phi(T_k^{t-n})$ are the appearance embedding features, $\text{IoU}(d_i^t, T_k^{t-n})$ computes the Intersection over Union (IoU) between bounding boxes so that model the motion features. $\alpha$ and $\beta$ are hyper-parameters to balance the terms. Here we treat the $\text{IoU}(d_i^t, T_k^{t-n})$ as a constant term because our work focuses on improving the appearance embedding.

As our previous discussion, the individual appearance embedding is easily affected by occlusions and illumination variations, therefore we propose to jointly consider the neighbors’ information when learning the appearance embedding to make the embedding more robust:

\[
\phi\left(d_i^t, \{d_{ij}^t\}_{j=1}^K\right) = w_i \phi''\left(d_i^t\right) + \sum_{j=1}^K w_{ij} \phi''\left(d_{ij}^t\right),
\]

\[
\phi\left(T_k^{t-n}, \{T_{kj}^{t-n}\}_{j=1}^K\right) = w_k \phi''\left(T_k^{t-n}\right) + \sum_{j=1}^K w_{kj} \phi''\left(T_{kj}^{t-n}\right),
\]

where $\{d_{ij}^t\}_{j=1}^K$ and $\{T_{kj}^{t-n}\}_{j=1}^K$ are the neighbor sets of $d_i^t$ and $T_k^{t-n}$, respectively. $w_i$ and $w_{ij}$ are the weights for the target $d_i^t$ and its $j$th neighbor $d_{ij}^t$. $w_k$ and $w_{kj}$ have the similar settings for $T_k^{t-n}$. These weights, $w_i, w_{ij, k_i, w_k, w_{kj}}$ account for fusing the embeddings from the targets and corresponding neighbors. To be concise, we abbreviate $\phi\left(d_i^t, \{d_{ij}^t\}_{j=1}^K\right)$ and $\phi\left(T_k^{t-n}, \{T_{kj}^{t-n}\}_{j=1}^K\right)$ as $\phi'(d_i^t)$ and $\phi'(T_k^{t-n})$. With Equations (2) and (3), Equation (1) can be reformulated as

\[
P\left(k \mid d_i^t, \{d_{ij}^t\}_{j=1}^K; T_k^{t-n}, \{T_{kj}^{t-n}\}_{j=1}^K\right) \propto \alpha \text{Cosine}(\phi'(d_i^t), \phi'(T_k^{t-n})) + \beta \text{IoU}(d_i^t, T_k^{t-n}).
\]

Since we only focus on improving the appearance embedding and treat the motion term as a constant, therefore the variables in Equation (4) are neighbor sets $\{d_{ij}^t\}_{j=1}^K$, $\{T_{kj}^{t-n}\}_{j=1}^K$, embeddings $\phi'(d_i^t)$, $\phi'(T_k^{t-n})$ and weights $w_i, w_{ij, k_i, w_k, w_{kj}}$. As such, the solution of Equation (4) is composed of two steps: first, find the proper neighbor sets $\{d_{ij}^t\}_{j=1}^K$, $\{T_{kj}^{t-n}\}_{j=1}^K$; second, find the optimal embeddings $\phi'(d_i^t)$, $\phi'(T_k^{t-n})$ and weights $w_i, w_{ij, k_i, w_k, w_{kj}}$ to combine the information from the targets and neighbors. To solve them, we propose a spatiotemporal relation-based neighbor selection to find the neighbors, then cast the target and corresponding neighbors into a graph and employ GCN to learn the embeddings and weights through node embedding and message passing. In this study, we name the graph consisting of a target and corresponding neighbors as a neighbor graph.

### 3.2 Spatiotemporal relation-based neighbor selection

For a pair of to-be-matched targets $d_i^t$ and $T_k^{t-n}$, it is nontrivial to select their suitable neighbors because many intraclass objects (e.g., pedestrians) usually appear at the video frame $t$ and $t - n$ whilst most of them are not the proper ones. To achieve that goal, we propose to obtain the neighbors by making full use of the spatiotemporal information produced by the tracking itself.

Specifically, give a detection response set $D_n^t = \{d_{ij}^t\}_{i=0,...,n}$ at current frame $t$ and a tracklet set $T^{t-1} = \{T_k^t\}$ where $t_k \leq t - 1$, we first compute the affinity matrix according to their individual appearances and motion cues then feed it to the Hungarian algorithm to solve the association problem. This is the traditional data association and we term it as an initial association. The initial association outputs a matched tracklet set $M^t = \{T_k^t\}$, an unmatched trajectory set $U^t = \{T_k^{t-1}\}$ where $t_k < t$, and an unmatched detection set $U^t = \{d_i^t\}$. To make the
initial association more robust, we remove the matched tracklet-detection pairs whose affinity scores are lower than $\tau_1$ from $M^t$ and put them into $U^t_d$ and $U^t_n$, respectively. On this basis, the temporal and spatial relations are utilized to select the suitable neighbors.

**Temporal relation:** Given a tracklet $T^t_k \in U^t_d$ (i.e., a tracklet that is last tracked at frame $t-n$), the temporal relation requires its neighbors should appear at both frames $t - n$ and $t$. In other words, the neighbors of $T^t_k$ are the tracklets which are jointly matched at frames $t-n$ and $t$. Formally, let $N_{T^t_k}^{temp}$ be the neighbor set of $T^t_k$ which stratifies the temporal restriction:

$$N_{T^t_k}^{temp} = M^{t-n} \cap M^t. \quad (5)$$

**Spatial relation:** Empirical observations from the pedestrian tracking tell that the spatial distances among the cowalkers are closer than the others. On the basis of this, we define the spatial relation for a target and its neighbors are that their Euclid-distances should be smaller than other observations. As such, for $T^t_k \in U^t_d$, we can further refine its temporal neighbor set $N_{T^t_k}^{temp}$ using the spatial restriction, thereby getting its final neighbor set $N_{T^t_k}$:

$$N_{T^t_k} = \left\{ T^t_k \left| \text{Dist} \left( T^t_k, T^t_k \right) < \text{Dist} \left( T^t_k, T^t_k \right) \cap \left( T^t_k, T^t_k \right) \in N_{T^t_k}^{temp}, T^t_k \notin N_{T^t_k} \right\}. \quad (6)$$

where $\text{Dist}(\cdot, \cdot)$ computes the Euclid-distance between two bounding boxes according to their center points. $j = 1, ..., K$ denotes the $j$th neighbor. For a detection response $d^t_i \in U^t_d$, its neighbor set which jointly satisfies the temporal and spatial relations is

$$N_{d^t_i} = \left\{ d^t_i \left| \text{Dist} \left( d^t_i, d^t_i \right) < \text{Dist} \left( d^t_i, d^t_i \right), \left( d^t_i, d^t_i \right) \notin U^t_d, d^t_i \notin N_{d^t_i} \right\}. \quad (7)$$

Note that here we do not set a distance threshold to filter out the unqualified ones from the selected neighbors because, in the scenario of vision MOT, such as MOT Challenge, the video can be filmed by the first or bird view, static or moving camera, resulting in frequent and dramatic variations of objects’ scale. Therefore, it is very difficult to set a proper distance threshold for various scenarios. Our spatiotemporal relation-based approach is more suitable than a threshold in such a task setting.

Our neighbor selection approach is time-efficient with no need for extra training data, which is tailored for the task as MOT because this kind of task is time-critical. In contrast, some other fields such as recommender systems and social network analysis which can run complex and slow neighbor-search procedures offline.

### 3.3 Learning appearance features from neighbor graph

With the neighbors selected by the spatiotemporal relations, the following question is how to represent the relations between the target and corresponding neighbors, thereby effectively fuse their information. The optimal representation should be able to guide the target to fully incorporate the information from its neighbor context, so that makes the target’s feature embedding being more discriminative and less affected by harmful factors, such as occlusions and illumination variations. To achieve that, we model each target and corresponding neighbors as
a graph, then employ GCN\textsuperscript{10} to learn their graph embeddings. The motivations of that are, first, the relations in a group of target and neighbors can be naturally interpreted as a graph structure. Second, Equations (3) and (4) which abstract the feature fusing process in our solution share the same philosophy with the mathematical definitions of node aggregation in GCN. In this study we propose the neighbor graph to model the target–neighbor relations, where the target is placed in the center position and all the neighbors connect to it. Figure 3 depicts the neighbor graph structure and the graph convolution on it.

In particular, considering a neighbor graph $G$ consists of $N$ nodes and a set of edges. If the target in $G$ has $K$ neighbors, then $N = K + 1$. For the $N$ nodes of $G$, they are assigned with the appearance feature vectors of the target and corresponding neighbors, that is, $\varphi(X) \in \mathbb{R}^{N \times d}$ where $d$ is the feature dimension of each node. As the target node may represent a detection or trajectory, thus the feature vector $\varphi(X)$ input into $G$ is processed differently, depending on the type of the target. In particular, suppose we want to build a pair of neighbor graph $G_{\text{det}}$ and $G_{\text{traj}}$ for $d_i$ and $T_k^{t-n}$, respectively. For the $G_{\text{det}}$, the feature vectors associated with it are extracted from $d_i$ and its neighbor $d_{ij}$ using $\varphi(d_i)$ and $\varphi(d_{ij})$. But for the $G_{\text{traj}}$, the input feature vectors of $T_k^{t-n}$ and its neighbors are computed as

\begin{equation}
\varphi(T_k^{t-n}) = \mu \varphi(T_k^{t-n-1}) + (1 - \mu) \varphi(T_k^{t-n}),
\end{equation}

where $\varphi(T_k^{t-n})$ is the smoothed feature of a trajectory at frame $t - n$, and $\varphi(T_k^{t-n})$ denotes the appearance feature of the associated detection at frame $t - n$. The momentum term $\mu$ is set to 0.9.

We use $A \in \mathbb{R}^{N \times N}$ to denote the adjacent matrix of neighbor graph $G$. Let the target node as the first node in the $G$, then the adjacent matrix is

\begin{equation}
A_{ij} = \begin{cases} 
1 & \text{if } i = 1 \text{ or } j = 1 \text{ or } i = j, \\
0 & \text{otherwise},
\end{cases}
\end{equation}

where $i, j \in \{1, \ldots, N\}$. Let $\hat{A}$ denote the normalized adjacent matrix, the layerwise propagations of GCN is computed as

\begin{equation}
Z^{(l+1)} = \sigma(\hat{A}Z^{(l)}W^{(l)}),
\end{equation}

[FIGURE 3] Visualization of the neighbor graph structure. The star and triangles in the graph denote the target and neighbors, respectively, and the arrow represents the message propagation in graph convolution. Best view in color [Color figure can be viewed at wileyonlinelibrary.com]
where $Z^{(l)}$ is the activations of the $l$th layer and $W^{(l)}$ is the learnable matrix, $Z^{(0)} = \varphi(X)$. In our implementation, we use a three-layer GCN and adopt ReLU as the activation function $\sigma$. The network in Equation (10) merges features of nodes into a 2048-dimension feature vector, then feeds it into a fully connected layer for outputting the 2048-dimension feature embedding. At the training phase, the Siamese GCN is adopted, and for a pair of $Z^{(l)}_1$ and $Z^{(l)}_2$, the loss $L$ for backup propagations is computed by the cosine distances between predictions and labels:

$$L\left(Z^{(l)}_1, Z^{(l)}_2, \text{label}\right) = \begin{cases} 1 - \text{Cosine}\left(Z^{(l)}_1, Z^{(l)}_2\right) & \text{if label} = 1, \\ \max\left(0, \text{Cosine}\left(Z^{(l)}_1, Z^{(l)}_2\right)\right) & \text{if label} = 0. \end{cases} \tag{11}$$

### 3.4 Association

This round of data association is performed on the unmatched set in the initial association phase. For each trajectory and detection in the unmatched set, we build the neighbor graph for them and learn their appearance embeddings through GCN. With these graph embeddings, we compute the corresponding affinity matrix then send it to the Hungarian algorithm to achieve data association. The same as the postprocess in the initial association, we filter the matching pairs which affinity scores are lower than the threshold $\tau^2$.

## 4 EXPERIMENT

### 4.1 Data sets and evaluation metrics

Data sets: As the prior JDE frameworks,\textsuperscript{21,22} we train our entire model on a collection of object detection, person ReID, and tracking data sets. Specifically, we purely train the detection branch in our model on the ETH\textsuperscript{33} and the CityPerson\textsuperscript{34} data sets, while jointly train the ReID and detection branches on the data sets of CalTech,\textsuperscript{35} MOT17,\textsuperscript{12} CUHK-SYSU,\textsuperscript{36} and PRW.\textsuperscript{37} For the testing, we evaluate our tracker on the MOT16\textsuperscript{12} and MOT17 data sets which share the same 14 sequences of video (seven for training and seven for testing). The differences between MOT16 and MOT17 data sets are that more kinds of public detections and finer annotations provided in the latter one.

Evaluation metrics: We adopt the CLEAR MOT Metrics\textsuperscript{38} to evaluate our work. In particular, metrics used in our evaluations are multiple object tracking accuracy (MOTA), false positive (FP), false negative (FN), ID-switches, identification F1 score (IDF1), the number of mostly tracked (MT) targets (>80% recovered), and the number of mostly lost (ML) targets (<20% recovered). Among these metrics, MOTA summarizes FP, FN, and IDS factors while seriously determined by the first two. IDs measures the tracking consistency and stability (i.e., lower IDs are better) thus directly reflects the tracker’s data association ability.

### 4.2 Implementation details

We implement our tracker in the JDE framework with reference to the work.\textsuperscript{22} In particular, we use the modified DLA-34 network\textsuperscript{39} as our backbone. For an input image with the size of $H_{\text{image}} \times W_{\text{image}}$, the backbone outputs a heatmap in shape of $C \times H_{\text{image}}/4 \times W_{\text{image}}/4$. 

6722 | WILEY

LIANG ET AL.
The settings of the detection and embedding heads upon the heatmap layer are the same as Reference [22]. For the head of neighbor graph learning, it consists of three layers of GCN and is placed after the above two heads. This subnetwork receives a neighbor graph containing 1 target and $K$ neighbors. When training and testing, if the number of neighbors is less than $K$, we copy the target to serve as neighbors. In the extreme case $K = 0$, the neighbor graph is dropped.

The DLA-34 backbone is initialized with the parameters pretrained on the COCO detection data set.40 The GCN model is pretrained on the person search data set CUHK-SYSU.36 We train and finetune the entire model using the Adam optimizer for 30 epochs, where the learning rate starts with $1 \times 10^{-4}$, then decays to $1 \times 10^{-5}$ and $1 \times 10^{-6}$ at 20 and 27 epochs. The input image is resized to $1088 \times 608$ and goes through a series of augmentation as scaling, rotation, and jittering. The thresholds $\tau_1$ and $\tau_2$ for associations are set to 0.8 and 0.95, respectively, and the number of neighbors $K$ is set to 3. The model is implemented with the PyTorch library and trained on 4 Nvidia RTX 2080Ti GPUs using batch size 12.

### 4.3 Comparison with the state-of-the-arts

We compare our method with the state-of-the-art trackers, including two graph-related methods (GSM8 and GSDT7), on the MOT Challenge platform.11 The platform guarantees fairness and objectiveness with two measures: first, it provides unified data for training and testing; second, testing results must be uploaded to the platform for standardized evaluations, with the rule that any tracker has at most three chances for evaluating, which avoids overfitting and tricking. Results reported in Table 1 show that our tracker achieves the highest MOTA score on both of the MOT16 and MOT17 testing, which demonstrates the superiority of our method. In particular, our approach gains a remarkable advantage in terms of FP score and that is because the proposed neighbor graph mechanism can enable the tracker to filter the FP detections more precisely. The appearance features produced by the neighbor graph became more discriminative since it incorporates the context information from neighbors. As such, with these enhanced features, the tracker identifies the false detections with more confidence. More importantly, using the enhanced appearance features our tracker can make more precise data association, which is reflected in the favorable scores of ML, IDF1, and IDs. Among the online trackers in Table 1, only Tracktor41 has lower IDs than us, and that is mainly because our method achieves a much lower FN value than.41 The lower FN value means our tracker is capable of finding more trajectories, which in turn bring more potential ID-switches. Works19,42 gain some advantages in terms of IDF1 and IDs, but that is because they run in the batch mode thus can utilize the global frames of a video. Compare to the two offline trackers, our method achieves a better MOTA score and is capable of tracking online at 17.3 Hz, with a single Nvidia 2080Ti GPU and Intel i5-9400F CPU. Note that we evaluate our method under the public detection protocol, therefore we only keep the bounding boxes output by our model that is close to the public detections.

### 4.4 Ablation study

To quantify the impacts of each contribution in our framework, we conduct a series of ablative experiments on the MOT17 training set. The MOT17 training set contains seven video
sequences recorded in different scenes. Therefore, we iteratively take one sequence as the validation set and the other six sequences for training, finally summarize all the results. In other words, we follow the $k$-fold cross-validation. Evaluations are performed with the MOTChallenge Devkit.*

**Neighbor graph versus no neighbor graph:** To investigate the effectiveness of the proposed neighbor graph mechanism, we first implement our method with the JDE tracker,22 then compare them on the MOT benchmark. For the sake of fairness, the network structure, hyperparameters, and training strategies of the two trackers are identical, except that our tracker is equipped with the proposed neighbor graph framework. The experiments are conducted on the MOT17 train set using the public and private detection.

As is shown in Table 2, our method remarkably optimizes IDs scores (16.5% and 19.2% gains), which demonstrates the proposed neighbor graph framework can significantly improve the data association procedures. Specifically, our contributions reinforce the individual appearance by incorporating its neighboring information, thus make it more distinctive and reduce the ambiguousness caused by harmful factors, such as occlusions. Figure 4 shows an example supporting our work, where the tracker without the neighbor graph mechanism suffers ID-switches when occlusions occur, while our neighbor graph guarantees the tracking consistency. On the other hand, the MOTA scores are also improved but on a smaller scale than the gains in IDs, and that is because the MOTA is an imbalance summation of FP, FN, and IDs where the FP and FN values are particularly larger than IDs. Accordingly, the improvements of IDs are easily overwhelmed by the FP and FN. We argue that IDs is equivalently important to MOTA since it truly measures the consistency and stability in tracking, thus our work focus on IDs optimization.

To evaluate the computational time of our tracker, we test it and the baseline tracker22 on the same devices (i5-9400F CPU × 1 and Nvidia 2080ti GPU × 1). As the right-most column in

| Data set | Tracker | Year | MOTA↑ | IDF1↑ | MT↑ (%) | ML↓ (%) | FP↓ | FN↓ | IDs↓ |
|----------|---------|------|-------|-------|---------|---------|-----|-----|------|
| MOT16    | DASOT   | 2020 | 46.1  | 49.4  | 14.6    | 41.6    | 8222| 89,204| 802  |
|          | MOTDT   | 2018 | 47.6  | 50.9  | 15.2    | 38.3    | 9253| 85,431| 792  |
|          | LSST    | 2019 | 49.2  | 56.5  | 15.2    | 38.3    | 7187| 84,875| 792  |
|          | HDRTR   | 2018 | 53.6  | 46.6  | 15.2    | 38.3    | 4714| 79,353| 792  |
|          | Tracktor| 2019 | 54.4  | 52.5  | 19      | 36.9    | 3280| 79,149| 682  |
|          | GSM     | 2020 | 57.0  | 58.2  | 22.0    | 34.5    | 4322| 73,573| 859  |
| Ours     |         | 2020 | **57.7** | **62.6** | **18.8** | **32.8** | 2432| 74,006| 732  |
| MOT17    | DASOT   | 2020 | 49.5  | 51.8  | 20.4    | 34.6    | 33,640| 247,370| 4142 |
|          | MOTDT   | 2018 | 50.9  | 52.7  | 17.5    | 35.7    | 24,769| 250,768| 2474 |
|          | LSST    | 2019 | 54.7  | 62.9  | 20.4    | 40.1    | 26,091| 228,434| 3726 |
|          | Tracktor| 2019 | 53.5  | 52.3  | 19.5    | 36.6    | 12,201| 248,047| 2072 |
|          | TT      | 2020 | 54.9  | 63.1  | 24.4    | 38.1    | 20,236| 233,295| 1088 |
|          | GSM     | 2020 | 56.4  | 57.8  | 22.2    | 34.5    | 14,379| 230,174| 2763 |
|          | GSDT    | 2020 | 56.4  | 42.0  | 16.7    | 40.8    | 17,421| 223,974| 4572 |
| Ours     |         | 2020 | **58.4** | **62.9** | **20.8** | **31.3** | **6526** | 225,507| 2425 |

**Note:** The symbol “*” means the tracker runs in the batch mode and the bold values mean the best scores.

Abbreviations: DASOT, data association and single object tracking; FN, false negative; FP, false positive; GSM, graph similarity model; HDRTR, hierarchical deep tracklet re-identification; ID, identity; IDF1, identification F1; ML, mostly lost; MOT, multi-object tracking; MOTA, multiple object tracking accuracy; MT, mostly tracked.
Table 2 shows, the two trackers' speed is very close (17.3 Hz vs. 19.0 Hz). Our tracker is slightly slower than the baseline because the neighbor graph mechanism brings some extra computational cost. While considering the notable improvements we achieved, we can conclude that our contributions do not sacrifice much speed to improve performance, thus achieving a good balance between accuracy and efficiency.

**Impact of the neighbor size**: As a neighbor graph consists of one target node and $K$ neighbor nodes, it is intuitive to regard the $K$ as an important hyper-parameter and keep tuning it to find the optimal setting. In this section, we conduct a series of experiments to explore how does the neighbor size $K$ influence the framework performance. We set the value of $K$ ranging from 1 to 10 and report the corresponding changes of IDs and MOTA in Figure 5. According to Figure 5, the proposed framework reaches the optimal performance when $K = 3$, and both smaller and larger $K$ settings will result in the suboptimal performance. The reasons behind this phenomenon can be summarized as the small $K$ setting does not provide sufficient neighboring information to supplement the individual target features. On the other hand, the large $K$ setting introduces more noises to the neighbor graph learning, resulting in the suboptimal graph features.

**Effectiveness of the spatiotemporal relations**: In this section, we conduct experiments on the MOT17 train set to quantify how much the spatiotemporal relations work for the neighbor selection. As is introduced in Section 3.2, the temporal and spatial relations jointly guarantee the neighbor consistency for a pair of targets, therefore we drop them alternatively in the experiments to explore the performance changes. Specifically, when dropping the spatial relation, neighbors are randomly selected from the set $N_{\text{temp}}$ without consideration of distance. When the temporal relation is dropped, the neighbors are selected only depending on distances, requiring no temporal consistency. From the results in Figure 6 we can see once the temporal or spatiorelation is dropped, the tracking performance, especially the ID-switches, suffers a decline. This is because without using the temporal and spatiorelations, the model cannot select the most appropriate neighbors to build the optimal neighbor graph, which further results in the degenerated graph representation for the target. Note that the MOTA gains are not the main concern in this experiment because the spatiotemporal relations we used to support the neighbor graph primarily affect the optimization of IDs, while MOTA is an imbalance summation of FP, FN, and IDs as our previous discussion.
FIGURE 4  Tracking examples from the MOT17-02 and MOT17-11 sequences. The vanilla tracker (no neighbor graph) suffers ID-switches at frame 476 (ID 67 → 71) and 754 (ID 107 → 118) due to the intraclass occlusions. By contrast, our neighbor graph-equipped tracker performs more robust in this case and avoids the identity confusions. Note that the tracks are identified by the color of bounding boxes, where the color is randomly initialized in each tracking round. ID, identity; MOT, multi-object tracking [Color figure can be viewed at wileyonlinelibrary.com]

FIGURE 5  Impact of the neighbor size $K$ on tracking performance. Experiments are conducted on the MOT17 train set and evaluated using the MOTChallenge-Devkit. MOT, multi-object tracking; MOTA, multiple object tracking accuracy [Color figure can be viewed at wileyonlinelibrary.com]
CONCLUSIONS AND FUTURE WORKS

The individual features of tracking targets are easily affected by the negative issues, such as occlusions, pose variations, and inaccurate detections, thus resulting in the mismatch of data association. In this study, we are inspired by collaborative filtering and propose to address the aforementioned issue via exploiting the neighboring information. To this end, we first leverage the temporal and spatial cues from the tracking itself to efficiently select suitable neighbors for the targets. With that, we design the neighbor graph to represent the target and corresponding neighbors and use GCN to model their mutual relations thereby enrich the target’s appearance embeddings. Over extensive experiments on the MOT benchmarks, our contributions are demonstrated effective to significantly enhance the data association via reducing the ID-switches, as well as achieve state-of-the-art overall performance. On the other hand, as our discussion in Section 4.2, when the neighbors are sparse, we straightly copy the target as pseudoneighbors to enable the neighbor graph. We think there is room to improve that although this case is rare. In more detail, we plan to explore the generative adversarial networks to generate high-quality pseudoneighbors. Besides, we also consider further exploiting the neighboring cues to improve the object detection component in the Joint-Detection-Embedding framework, which means the neighbor information will jointly enhance the embedding learning and object detection tasks.

ACKNOWLEDGMENT

This study was partially supported by the National Natural Science Foundation of China (No. 61906210)

ENDNOTES

* https://github.com/xstgavin/amilan-motchallenge-devkit.

ORCID

Tianyi Liang https://orcid.org/0000-0003-2508-6288
Long Lan https://orcid.org/0000-0002-4238-8985
Xiang Zhang https://orcid.org/0000-0002-5201-3802
Xindong Peng https://orcid.org/0000-0001-9080-6267
Zhigang Luo https://orcid.org/0000-0002-1063-9970

FIGURE 6 Influences of spatiotemporal relations on tracking performance. “None” means no temporal or spatiorelation is used. Experiments are conducted on the MOT17 train set and evaluated using the MOTChallenge-Devkit. MOT, multi-object tracking; MOTA, multiple object tracking accuracy [Color figure can be viewed at wileyonlinelibrary.com]
REFERENCES

1. Wang K, Ma Z, Chen S, Yang J, Zhou K, Li T. A benchmark for clothes variation in person re-identification. *Int J Intell Syst*. 2020;35:1881-1898.

2. Zeng K, Ning M, Wang Y, Guo Y. Hierarchical clustering with hard-batch triplet loss for person re-identification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE; 2020:13657-13665.

3. Leal-Taixe L, Pons-Moll G, Rosenhahn B. Everybody needs somebody: modeling social and grouping behavior on a linear programming multiple people tracker. In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. Piscataway, NJ: IEEE; 2011:120-127.

4. Duan G, Ai H, Cao S, Lao S. Group tracking: exploring mutual relations for multiple object tracking. In: *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*. Vol III. Berlin, Heidelberg: Springer; 2012:129-143.

5. Luber M, Stork JA, Tipaldi GD, Arras KO. People tracking with human motion predictions from social forces. In: *2010 IEEE International Conference on Robotics and Automation*. Piscataway, NJ: IEEE; 2010: 464-469.

6. Solera F, Calderara S, Ristani E, Tomasi C, Cucchiara R. Tracking social groups within and across cameras. *IEEE Trans Circuits Syst Video Technol*. 2017;27:441-453.

7. Wang Y, Weng X, Kitani K. Joint detection and multi-object tracking with graph neural networks. *arXiv preprint*. 2020;arXiv:2006.13164.

8. Liu Q, Chu Q, Liu B, Yu N. GSM: graph similarity model for multi-object tracking. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann; 2020:530-536.

9. Chen H, Wang S, Jiang N, Li Z, Yan N, Shi L. Trust-aware generative adversarial network with recurrent neural network for recommender systems. *Int J Intell Syst*. 2021;36:778-795.

10. Kipf TN. Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations (ICLR 2017)*. 2017.

11. Dendorfer P, Ošep A, Milan A, et al. MOTChallenge: a benchmark for single-camera multiple target tracking. *arXiv preprint*. 2020;arXiv:2010.07548.

12. Milan A, Leal-Taixé L, Reid ID, Roth S, Schindler K. MOT16: a benchmark for multi-object tracking. *arXiv preprint*. 2016;arXiv:1603.00831.

13. Tang S, Andriluka M, Andres B, Schiele B. Multiple people tracking by lifted multicut and person re-identification. In: *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Piscataway, NJ: IEEE; 2017:3701-3710.

14. Lan L, Wang X, Hua G, Huang ST, Tao D. Semi-online multi-people tracking by re-identification. *Int J Comput Vision*. 2020;128:1-19.

15. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In: *2016 IEEE International Conference on Image Processing (ICIP)*. Piscataway, NJ: IEEE; 2016:3464-3468.

16. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: *2017 IEEE International Conference on Image Processing (ICIP)*. Piscataway, NJ: IEEE; 2017:3645-3649.

17. Kramer KA, Kramer KA, Stubberud SC. Tracking of multiple target types with a single neural extended Kalman filter. *Int J Intell Syst*. 2010;25:440-459.

18. Sadeghian A, Alahi A, Savarese S. Tracking the untrackable: learning to track multiple cues with long-term dependencies. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Piscataway, NJ: IEEE; 2017:300-311.

19. Babaei M, Athar A, Rigoll G. Multiple people tracking using hierarchical deep tracklet re-identification. *arXiv preprint*. 2018;arXiv:1811.04091.

20. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE; 2016:770-778.

21. Wang Z, Zheng L, Liu Y, Wang S. Towards real-time multi-object tracking. 2019. arXiv preprint arXiv:1909.12605.

22. Zhang T, Wang C, Wang X, Zeng W, Liu W. FairMOT: on the fairness of detection and re-identification in multiple object tracking. *arXiv preprint*. 2020;arXiv:2004.01888.
23. Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs advances. In: Neural Information Processing Systems. New York, NY: Curran Associates; 2017:1024-1034.

24. Wang Z, Chen T, Ren J, Yu W, Cheng H, Lin L. Deep reasoning with knowledge graph for social relationship understanding. In: 27th International Joint Conference on Artificial Intelligence (IJCAI 2018). San Mateo, CA: Morgan Kaufmann; 2018:1021-1028.

25. Qi X, Liao R, Jia J, Fidler S, Urtasun R. 3D graph neural networks for RGBD semantic segmentation. In: 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway, NJ: IEEE; 2017:5209-5218.

26. Yan S, Xiong Y, Lin D, Tang. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: 18 AAAI Conference on Artificial Intelligence (AAAI). San Mateo, CA: Morgan Kaufmann; 2018:7444-7452.

27. Gao J, Zhang T, Xu C. Graph convolutional tracking. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE; 2019:4649-4659.

28. Jiang X, Li P, Li Y, Zhen X. Graph neural based end-to-end data association framework for online multiple-object tracking. arXiv preprint. 2019;arXiv:1907.05315.

29. Ma C, Li Y, Yang F, et al. Deep Association: end-to-end graph-based learning for multiple object tracking with conv-graph neural network. In: Proceedings of the 2019 International Conference on Multimedia Retrieval. New York, NY: ACM; 2019:253-261.

30. Mazzon R, Poiesi F, Cavallaro A. Detection and tracking of groups in crowd. In: 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance. Piscataway, NJ: IEEE; 2013:202-207.

31. Castellano G, Castiello C, Mencar C, Vessio G. Crowd detection in aerial images using spatial graphs and fully-convolutional neural networks. IEEE Access. 2020;8:64534-64544.

32. Ktena SI, Parisot S, Ferrante E, et al. Distance metric learning using graph convolutional networks: application to functional brain networks International Conference on Medical Image Computing and Computer-Assisted Intervention; 2017:469-477.

33. Ess A, Leibe B, Schindler K, Gool LV. A mobile vision system for robust multi-person tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE; 2008:1-8.

34. Zhang S, Benenson R, Schiele B. CityPersons: a diverse dataset for pedestrian detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE; 2017:4457-4465.

35. Dollar P, Wojek C, Schiele B, Perona P. Pedestrian detection: a benchmark. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE; 2009:304-311.

36. Xiao T, Li S, Wang B, Lin L, Wang X. Joint detection and identification feature learning for person search. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE; 2017:3376-3385.

37. Gheissari N, Sebastian TB, Hartley R. Person reidentification using spatiotemporal appearance. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06). Piscataway, NJ: IEEE; 2006:1528-1535.

38. Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: the CLEAR MOT metrics. Eurasip J Image Video Process. 2008;2008:1-10.

39. Zhou X, Wang D, Krähenbühl P. Objects as points. arXiv preprint. 2019;arXiv:1904.07850.

40. Lin T-Y, Maire M, Belongie SJ, et al. Microsoft COCO: common objects. In: Context European Conference on Computer Vision. Berlin, Heidelberg: Springer; 2014:740-755.

41. Feng W, Hu Z, Wu W, Yan J, Ouyang W. Multi-object tracking with multiple cues and switcher-aware classification. arXiv preprint. 2019;arXiv:1901.06129.

42. Zhang Y, Sheng H, Wu Y, et al. Long-term tracking with deep tracklet association. IEEE Trans Image Process. 2020;29:6694-6706.

43. Chu Q, Ouyang W, Liu B, Zhu F, Yu N. DASOT: a unified framework integrating data association and single object tracking for online multi-object tracking. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020). San Mateo, CA: Morgan Kaufmann; 2020:10672-10679.
44. Chen L, Ai H, Zhuang Z, Shang C. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE International Conference on Multimedia and Expo (ICME). Piscataway, NJ: IEEE; 2018:1-6.

45. Bergmann P, Meinhardt T, Leal-Taixe L. Tracking without bells and whistles. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway, NJ: IEEE; 2019:941-951.

How to cite this article: Liang T, Lan L, Zhang X, Peng X, Luo Z. Enhancing the association in multi-object tracking via neighbor graph. Int J Intell Syst. 2021;36:6713-6730. https://doi.org/10.1002/int.22565