Computer Vision and Image Understanding 214 (2022) 103301

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Simultaneous multi-person tracking and activity recognition based on cohesive cluster search

Wenbo Li [a], Yi Wei [a], Siwei Lyu [c], Ming-Ching Chang [b,*]

[a] Samsung Research AI Center, 665 Clyde Ave, Mountain View, CA, USA
[b] University at Albany, State University of New York, 1400 Washington Avenue, Albany, NY 12222, USA
[c] University at Buffalo, State University of New York, 12 Capen Hall, Buffalo, NY 14260-1660, USA

ARTICLE INFO

Communicated by Nikos Paragios

MSC:
41A05
41A10
65D05
65D17

Keywords:
Group activity
Collective activity recognition
Pairwise interaction
Multi-person tracking

ABSTRACT

We present a bootstrapping framework to simultaneously improve multi-person tracking and activity recognition at individual, interaction and social group activity levels. The inference consists of identifying trajectories of all pedestrian actors, individual activities, pairwise interactions, and collective activities, given the observed pedestrian detections. Our method uses a graphical model to represent and solve the joint tracking and recognition problems via three stages: *(i)* activity-aware tracking, *(ii)* joint interaction recognition and occlusion recovery, and *(iii)* collective activity recognition.

This full-stack problem induces great complexity in learning the representations for the sub-problems at each stage, and the complexity increases as with more stages in the system. Our solution is to make use of symbolic cues for inference at higher stages, inspired by the observations of cohesive clusters at different stages. This also avoids learning more ambiguous representations in the higher stages.

High-order correlations among the visible and occluded individuals, pairwise interactions, groups, and activities are then solved using the cohesive cluster search within a Bayesian framework. Experiments on several benchmarks show the advantages of our approach over the existing methods.

1. Introduction

Multi-person activity recognition is a major component of many applications, *e.g*, video surveillance and traffic control. The problem entails the inference of the actor activities, their motion trajectories, as well as the interactions and time dynamics of the groups for the case of multiple actors. This task is challenging, since the activities must be analyzed from both the spontaneous individual actions and the complex social dynamics involving groups and crowds (Vinciarelli et al., 2009). We aim to address the **where** and **when** problems by visual trajectory analysis, as well as the **who** and **what** problems by activity recognition.

While advanced methods for person detection are becoming more reliable (Cai et al., 2016; Yu et al., 2016), most existing activity recognition approaches rely on visual tracking following a tracking-by-detection paradigm. These methods either fail to consider social interactions while inferring activities (Ibrahim et al., 2016; Khamis et al., 2012b,a) or have difficulties recognizing the structural correlations of actions and interactions (Choi and Savarese, 2012, 2014; Deng et al., 2016). In particular, there are two major challenges: *(i)* ineffective tracking due to frequent occlusions in groups and crowds, and *(ii)* the lack of a suitable methodology to infer the complex but salient structures involving social dynamics and groups.

In this paper, we address both challenges using a bootstrapping framework to simultaneously improve the two tasks of multi-person tracking and social group activity recognition. We take person detection bounding boxes (Cai et al., 2016; Yu et al., 2016) as input to perform initial multi-person tracking. We then recognize stable group structures including the temporally cohesive *individual activities* (such as walking) and *pairwise interactions* (such as walking side-by-side, see Fig. 1 to robustly infer collective social activities (such as street crossing in a group) in multiple stages. Auxiliary inputs such as body orientation detections can be considered within the stages if available. The recognized activities and salient grouping structures are used as priors to recover occluded detections and false associations to improve performance.

We explicitly explore the correlations of *pairwise interactions* (of two individuals) and *group activities* (within the group of more individuals) during the optimization. Observe in Fig. 1 that group activities generally are identified by cohesive clusters of pairwise interactions, which we have exploited in the multi-stage inference steps. In our method, multi-person tracking and individual/group activity recognition are jointly optimized, such that consistent activity labels characterizing the dynamics of the individuals and groups can be obtained. The individual and group activities are formulated using a dynamic graphical model, and high-order correlations are represented using hypergraphs. The

* Corresponding author.
E-mail address: mchang2@albany.edu (M.-C. Chang).

https://doi.org/10.1016/j.cviu.2021.103301
Received 21 December 2020; Received in revised form 1 October 2021; Accepted 4 October 2021
Available online 29 October 2021

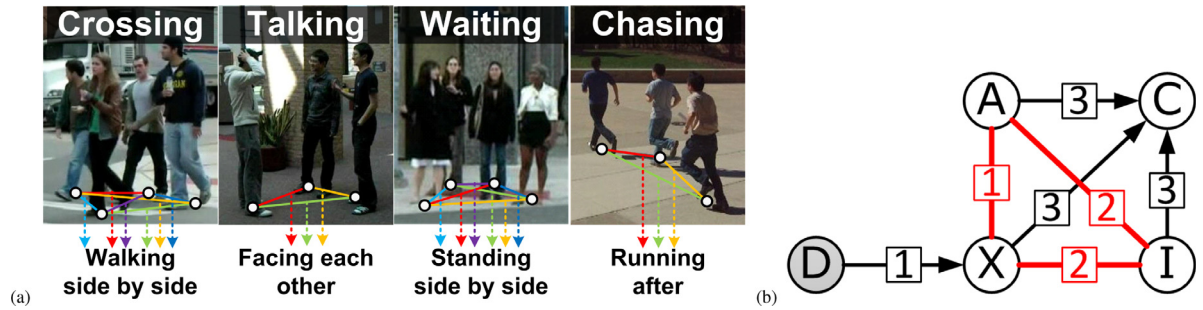

Fig. 1. (a) This work is based on two main hypotheses that: *(i)* Multi-person tracking and activity recognition can be jointly solved using an improved, unified framework. *(ii)* Group *collective activities* (crossing, talking, waiting, chasing, *etc.*) can be characterized by a *cohesive cluster of pairwise interactions* (walking side by side, facing each other, standing side by side, running after, *etc.*, a comprehensive definition is in Supplementary material Table A.4) within the group. See Section 3.1. (b) The dependency graph that can jointly infer the target tracking (X), individual activities (A), pairwise interactions (I), and collective activities (C), all from the input detections (D). Numbers on the edges indicate the inference stages in the multi-stage updating scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

simultaneous pedestrian tracking and multi-person activity recognition problems are then to be solved jointly using an efficient **cohesive cluster search** on the hypergraphs.

Main contribution of this work is two-fold. First, we propose a new framework that can jointly solve the two tasks of real-time simultaneous tracking and activity recognition. Explicit modeling of the correlations among the individual activities, pairwise interactions, and collective activities leads to a consistent solution. Second, we propose a hypergraph formulation to infer the high-order correlations among social dynamics, occlusions, groups, and activities in multi-stages. Simultaneous tracking and activity recognition are formulated as a bootstrapping framework, which can be solved efficiently using the search of cohesive clusters in the hypergraphs. This cohesive cluster search solution is general that it can be extended to include additional scenarios or constraints in new applications.

The main novelty of our work is the adaptation of cohesive cluster for trajectory tracking and activity recognition. Specifically, the optimization procedure for the cohesive cluster search preserves advantages from the previous works; new research efforts are mainly reflected in the investigation of how to construct hypergraphs for the two problems in an effective manner, such that the tracking and activity recognition can benefit each other.

Experiments on several benchmarks show the advantages of our method with improvements in both activity recognition and multi-person tracking. Our method is easily deployable to real-world applications, since: *(i)* our method does not depend on site knowledge, *i.e.*, camera calibration is not required; *(ii) online* video streams can be processed by considering a time window in a round; *(iii)* the computation can be performed in real-time (about 20 FPS, excluding the input detection steps).

2. Related works

There exists a tremendous amount of multi-person tracking, trajectory analysis and activity recognition. See Aggarwal and Ryoo (2011) and Luo et al. (2014) for survey. Our work is most related to the *collective activity* recognition, which are organized into the following three categories — recognition based on *(i)* detection, *(ii)* tracking, and *(iii)* simultaneous tracking and recognition.

2.1. Collective activity recognition based on detection

A hierarchical model is used in Lan et al. (2012) to recognize collective activities by considering the person–person and group–person contextual information. The work of Deng et al. (2015) uses hierarchical deep neural networks with a hierarchical conditional random field to recognize collective activities based on the dependencies of individual activities. This work is further extended in Deng et al. (2016), where the individual and collective activities are iteratively recognized using RNN with refinements. Multi-instance learning is used in Hajimirsadeghi et al. (2015) to recognize collective activities by inferring the cardinality relations of individual activities. A recurrent CNN is used in Bagautdinov et al. (2017) for the joint target detection and activity recognition.

Azar et al. (2019) represent the activities within each frame using an activity map (similar to multi-channel heat map), and proposed a recurrent refinement CNN for the regression purpose. The interactions and other contextual information are encoded by the convolutional computations implicitly.

2.2. Collective activity recognition based on tracking

In this category, individual target trajectories are used as the input to recognize collective activities. Collective activities are recognized in Choi et al. (2011) using random forests for the spatio-temporal volume classification. A two-stage deep temporal neural network is used in Ibrahim et al. (2016), where the first stage recognizes individual activities, and the second stage aggregates individual observations to recognize collective activities. In Antic and Ommer (2014), the key *constituents* of activities and their relationships are used to recognize collective activities. A graphical model is developed in Amer et al. (2014) to capture high-order temporal dependencies of video features. The *and-or graph* (Amer et al., 2013) is applied for video parsing and activity querying, where the detectors and trackers are launched upon receiving queries. A RNN architecture is designed in Wang et al. (2017) to model high-order social group interaction contexts.

Kong et al. (2018) propose hierarchical attention temporal networks based on the modifications of the two-stage deep temporal model proposed in Ibrahim et al. (2016). The first-stage and second-stage attention temporal networks model the correlations of individual persons and subgroups, respectively. The attention mechanisms for the two stages estimate the importance of body parts of each person and individual persons, respectively. Qi et al. (2018) propose a recurrent message-passing network to aggregate the contextual or relational information for each person, and the person-level representations are attentively pooled to form the group-level representation for classification. Similarly, Wu et al. (2019) propose to use a graph CNN to model the inter-person dependencies as the relational representations to augment person-level representations. The person-level representations are pooled for group activity classification.

2.3. Simultaneous tracking and activity recognition

Only very few works deal with the problem of simultaneous multi-person trajectory analysis and activity recognition. In Khamis et al. (2012b), per-frame and per-track cues extracted from an appearance-based tracker are combined to capture the regularity of individual actions. A network flow-based model is used in Khamis et al. (2012a) to

link detections while inferring collective activities. However, these two methods did not consider pairwise interactions for activity recognition. In Choi and Savarese (2012) and Choi and Savarese (2014), trajectory analysis and activity recognition are formulated as a joint energy maximization problem, which is solved by belief propagation with branch-and-bound. However high-order correlations among individual and pairwise activities are not considered, which limits the activity recognition performance.

2.4. Key differences between existing methods and ours

We highlight two key differences of our method against existing works. First, most existing work either use raw detections (mostly ground-truth detection annotations) or off-the-shelf associated detections (tracked trajectories) as input. These methods are developed based on an assumption that the person detection and tracking problems have been solved. However, person detection and tracking still remain challenging open problems nowadays. According to our observations, temporal information encoded by person trajectories is crucial to improve activity recognition performance. Therefore, we propose to investigate how best to boost activity recognition performance via improving the tracking performance based on real-world detection results. Therefore, we do not rely on using ground-truth detection annotations. We aim on designing a bootstrapping framework to improve both human activity recognition and trajectory analysis based on real-world data.

In target trajectory based methods, the collective activity recognition complexity increases with the number of involved people. Such complexity is reflected in the degree of intra-class variations of collective activities. Several factors and aspects are involved, including the varying involvement degrees of different number of people, viewpoint variations, background cluttering, *etc*. Therefore, we hypothesize that it is easier to learn the representation of individual activities or two-person interactions, in contrast to the learning of the full representation of collective activities. However, most existing methods focus on group-level representation that learns collective activity directly. This branch of methods become mainstream due to the prospering of deep neural networks (*e.g*, CNN, RNN, GCN, message passing, *etc*.). In contrary, our hypothesis leads to an alternative approach to avoid the ambiguities in brute-force collective activity representation/learning. Our solution is to make use of symbolic information for recognition. Our approach is also inspired from the fundamental observation of cohesive clustering of pairwise interactions that can serve strong features to represent collective activities, as illustrated in Fig. 1.

In summary, the proposed method differentiates from existing methods in that it addresses the practical challenges of the joint problems of tracking and collective activity recognition by mining symbolic cues in an effective multi-stage scheme. The result of our trajectory analysis and activity recognition is interpretable and explainable by design.

3. Method

We start with defining notations to be used in our method. Given an input video sequence, consider the most recent time window $T = [t-\tau, t]$ in an *online* fashion, and denote previous time frames $[1, t-\tau-1]$ as T'. Let D_T represent a set of target *detections* obtained using person detectors *e.g.* (Cai et al., 2016; Yu et al., 2016). Let $X_{T'}$ represent the set of existing target trajectories. Let $A_{T'}$, $I_{T'}$, and $C_{T'}$ represent the set of recognized individual activities, pairwise interactions, and collective activities, respectively. Given D_T, our approach aims to simultaneously solve the multi-person tracking and activity recognition problems, by inferring the following four terms within T: *(i) target trajectories* $X_T = \{\mathtt{x}_1, \dots, \mathtt{x}_b\}$, where b is the number of observed targets, *(ii) individual activity* labels $A_T = \{\mathtt{a}_1, \dots, \mathtt{a}_b\}$, *(iii) pairwise interaction* labels $I_T = \{\mathtt{i}_{1,2}, \mathtt{i}_{1,3}, \dots, \mathtt{i}_{2,3}, \dots, \mathtt{i}_{b-1,b}\}$, and *(iv) collective activity* labels $C_T = \{\mathtt{c}_{t-\tau}, \dots, \mathtt{c}_t\}$, where $\mathtt{c}_f$ represents the collective activity with the most involved targets in the fth frame. After a time window is processed, the method will extend target tracklets, update activity labels, and move on to the next time window: $X_{1:t} = [X_{T'}, X_T]$, $A_{1:t} = [A_{T'}, A_T]$, $I_{1:t} = [I_{T'}, I_T]$, and $C_{1:t} = [C_{T'}, C_T]$. To simplify notions, we omit the temporal indices to represent the variables within $[t-\tau, t]$ as X, A, I, C, and represent previous variables as X', A', I', C', *i.e*$X' = X_{T'}$, $A' = A_{T'}$, $I' = I_{T'}$, $C' = C_{T'}$.

3.1. Problem formulation

We aim to infer accurate trajectories of all targets (X) as well as their individual activities (A), pairwise interactions (I) and collective activities (C), all from the observed detections (D). Refer to Fig. 1. Relationship between these variables can be expressed as the joint distribution $\mathbf{Pr}(X, A, I, C|D)$ as a dependency graph. Based on the conditional independence assumption of X, A, I, C in the graphical model, $\mathbf{Pr}(X, A, I, C|D)$ can be decomposed into three terms:

$$f_1(X, D) \cdot f_2(X, A, I) \cdot f_3(X, A, I, C). \tag{1}$$

$f_1(X, D)$ is the confidence of target tracking, where the calculation will be given in Section 3.3. $f_2(X, A, I)$ models the inter-dependencies among target trajectories, individual activity and pairwise interaction labels, which is further expressed as a Markov random field (red cycle in Fig. 1):

$$f_2(X, A, I) \;\sim\; \varphi_1(X, A) \cdot \varphi_2(A, I) \cdot \varphi_3(I, X), \tag{2}$$

where φ_1, φ_2 and φ_3 are three *clique potential* functions capturing the inter-correlations between each variable pair. Derivation of these clique potentials will be given in Sections 3.3 and 3.4. $f_3(X, A, I, C)$ reflects an important assumption that collective activities can be effectively modeled by robust inference of target trajectories, individual activities and pairwise interactions; Section 3.5 will provide further details.

The inference of the joint tracking and recognition is then formulated as seeking:

$$\begin{aligned} &\arg\max_{X,A,I,C} \log \mathbf{Pr}(X, A, I, C|D) = \\ &\qquad \arg\max_{X,A,I,C} \left\{ \begin{array}{c} \log f_1(X, D) + \log \varphi_1(X, A) + \log \varphi_2(A, I) \\ + \log \varphi_3(I, X) + \log f_3(X, A, I, C) \end{array} \right\}. \end{aligned} \tag{3}$$

However, standard iterative optimization such as *block coordinate descent* is not practical due to that: *(i)* the coupling of variables X, A, I, C is still complicated; *(ii)* each of these variables represents a superset of time-dependent variables, so their joint optimization will be very inefficient; *(iii)* a real-time processing method is desired. We adopt a heuristic approximate solution using **multi-stage** *updating scheme*, which first jointly updates X, A, and then updates I, followed by the update of C. Our strategy is based on an important hypothesis that *inferring pairwise interactions I is crucial in resolving the entire optimization*, because I is the knob governing the representations in-between X, A and C. We ensure the inference or updates in each stage can finish in a few iterations to support real-time processing, while maintain sufficient accuracy.

Our updating scheme shares spirit with the standard Gibbs sampling and MH-MCMC method for the inference in probabilistic graphical models. The updating scheme takes the following three stages:

Stage 1 activity-aware tracking (Section 3.3), where individual target trajectories and activity labels are updated using:

$$(X^*, A^*) = \arg\max_{X,A} \log f_1(X, D) + \log \varphi_1(X, A). \tag{4}$$

Stage 2 joint interaction recognition and occlusion recovery (Section 3.4), where the interaction labels together with the target trajectories and activities are updated using:

$$(X^\ddagger, A^\ddagger, I^*) = \arg\max_{X^*,A^*,I} \log \varphi_2(A^*, I) + \log \varphi_3(I, X^*). \tag{5}$$

Stage 3 collective activity recognition (Section 3.5), where the collective activity labels are updated using:

$$C^* = \arg\max_{C} \log f_3(X^{\ddagger}, A^{\ddagger}, I^*, C). \tag{6}$$

We will show in Sections 3.3 and 3.4 that we model *high-order* correlations among X, A and I using two respective *hypergraphs*. The clique potentials $\varphi_1, \varphi_2, \varphi_3$ in **Stage 1** and **Stage 2** can be derived as the optimization of maximal weight search over the two hypergraphs, in order to infer X, A, I. **Stage 3** infers C using a probabilistic formulation based on the inferred X, A, I.

Notations for video activities, problem formulation and tracking are summarized in Supplementary Table A.1, where graph and hypergraph related notations are summarized in Supplementary Table A.2.

3.2. Cohesive cluster search on the hypergraph

We define an *undirected hypergraph* $\mathcal{H} = (V, E, W)$, where $V = \{\overbrace{v_1, \dots, v_p, \dots, v_n}^{n}\}$ denotes the *vertex* set of $\mathcal{H}$ and p denotes vertex index. An *undirected hyperedge* with m-incident vertices is defined as $\mathbf{e}^m = \{v_1^{\mathbf{e}}, \dots, v_m^{\mathbf{e}}\}$, where m is the *degree* of the hyperedge. The set of all m-degree hyperedges is denoted as $E = \{\mathbf{e}^m\}$. The *weights* of hyperedges are denoted as $W : E \to \mathbb{R}$, *i.e.*, each hyperedge is associated with a weight.

We use the hypergraphs to represent both (1) the detection-tracklet association for tracking (X', X), and (2) the correlations among individual activities A and pairwise interactions I. The joint problem of multi-person tracking (with possible refinements) and group activity recognition can be solved using a standard *cohesive cluster search* on the hypergraph (Liu et al., 2010). A *cluster* C within a hypergraph is a vertex set with interconnected hyperedges. We use $\kappa = |C|$ to denote the number of vertices in C, and E^C to denote the set of all incident hyperedges of C. A cluster is *cohesive* if its vertices are interconnected by a large amount of hyperedges with dense weights. Denote $\Psi(\cdot)$ the **weighting function** that measures the weight of a cluster. For a vertex $v_r \in V$, the cohesive cluster search optimization is to determine a large cluster $C(v_r)$ with dense weights:

$$C(v_r)^* = \arg\max_{C(v_r)} \Psi\left(C(v_r)\right) \text{ s.t. } C(v_r) \subset V. \tag{7}$$

We use *indicator vector* $\mathbf{y} = (y_1, \dots, y_n)$, $y_p \in \{0, 1\}$ to denote the selection of vertices from $\mathcal{H}$ to be included in C: $y_p = 1$ for $v_p \in C$, and $y_p = 0$ otherwise. The selection is constrained such that up to κ vertices including v_r are enclosed in C, such that $\sum_{p=1}^n y_p = \kappa$, and $y_r = 1$.

The design of $\Psi(\cdot)$ affects the resulting cluster C from the search. Typical $\Psi(\cdot)$ can be the total weight of all incident hyperedges. However, direct maximization of the total weight leads to a large cluster that is not necessarily cohesive. Instead, we maximize a *normalized* weight, which is the total weight divided by the cardinality of all incident hyperedges. This normalization also enables continuous optimization. For C with κ vertices and m-degree hyperedges, this normalizer is κ^m. Our weighting function $\Psi(C(v_r))$ is:

$$\frac{\sum_{\mathbf{e}^m \in E^C} W(\mathbf{e}^m)}{\kappa^m} = \sum_{v_p, \dots, v_q \in V} \left(W(\overbrace{v_1, \dots, v_m}^{m}) \cdot \overbrace{\frac{y_1}{\kappa} \cdots \frac{y_m}{\kappa}}^{m} \right) \tag{8}$$

It is intuitive to enforce that C must contain at least one hyperedge, thus κ must $\geq m$. Let $\delta_p = \frac{y_p}{\kappa}$ and $\epsilon = \frac{1}{\kappa}$. The conditions $\sum_{p=1}^n y_p = \kappa$ is then $\sum_{p=1}^n \delta_p = 1$. We relax the constraint of $y_p \in \{0, 1\}$ to be $\delta_p \in [0, \epsilon]$, so δ is a continuous variable for optimization. Eq. (7) is re-written as:

$$\begin{aligned} \max F(\delta) &= \sum_{v_p, \dots, v_q \in V} \left(W(\overbrace{v_1, \dots, v_m}^{m}) \cdot \overbrace{\delta_1 \cdots \delta_m}^{m} \right) \\ &= \sum_{v_1, \dots, v_m \in V} \left(W(v_1, \dots, v_m) \cdot \prod_{p=1}^{m} \delta_p \right) \\ &\text{s.t. } \sum_{p=1}^{n} \delta_p = 1, \delta_p \in [0, \epsilon], \delta_r = \epsilon. \end{aligned} \tag{9}$$

We follow (Liu et al., 2010) to calculate the maximizer δ^*. First, we add Lagrangian multipliers to formulation Eq. (9) to its Lagrangian function:

$$L(\delta, \lambda, \mu, \beta) = F(\delta) - \lambda(\sum_{p=1}^{n} \delta_p - 1) + \sum_{p=1}^{n} \mu_p \delta_p + \sum_{p=1}^{n} \beta_p(\epsilon - \delta_p) \tag{10}$$

where λ, μ_1, …, μ_n and β_1, …, β_n are Lagrangian multipliers. We have $\mu_p \geq 0$ and $\beta_p \geq 0$ for all $p = 1, \dots, n$.

We define the *reward* at vertex p as

$$R_p(\delta) = \sum_{v_1, \dots, v_{m-1} \in V} \left(W(v_1, \dots, v_{m-1}, p) \cdot \prod_{t=1}^{m-1} \delta_t \right) \tag{11}$$

Any local maximizer δ^* must satisfy the Karush–Kuhn Tucker (KKT) condition:

$$\begin{cases} mR_p(\delta^*) - \lambda + \mu_p - \beta_p, \ p = 1, \dots, n \\ \sum_{p=1}^n \delta_p^* \mu_p = 0, \\ \sum_{p=1}^n (\epsilon - \delta_p^*) \beta_p = 0. \end{cases} \tag{12}$$

Since δ_p^*, μ_p, β_p are all non-negative, $\sum_{p=1}^n \delta_p^* \mu_p = 0$ is equivalent to saying that if $\delta_p^* > 0$, then $\mu_p = 0$, and $\sum_{p=1}^n (\epsilon - \delta_p^*) \beta_p = 0$ is equivalent to saying that if $\delta_p^* < \epsilon$, then $\beta_p = 0$. Therefore, the KKT conditions can be rewritten as:

$$R_p(\delta^*) \begin{cases} \leq \lambda/m, & \delta_p^* = 0, \\ = \lambda/m, & 0 < \delta_p^* < \epsilon, \\ \geq \lambda/m, & \delta_p^* = \epsilon. \end{cases} \tag{13}$$

According to δ, the vertices set V can be divided into three disjoint subsets, $V_1 = \{v_p | \delta_p = 0\}$, $V_2 = \{v_p | \delta_p \in (0, \epsilon)\}$, and $V_3 = \{v_p | \delta_p = \epsilon\}$.

By the properties of maximizer δ^* characterized by Eq. (13), if δ^* is the solution to Eq. (9), there exists a constant η $(= \lambda/m)$ such that *(i)* the rewards of all vertices belonging to $V_1(\delta^*)$ are not larger than η; *(ii)* the rewards of all vertices belonging to $V_2(\delta^*)$ are equal to η; and *(iii)* the rewards of all vertices belonging to $V_3(\delta^*)$ are not smaller than η.

Let $V_d(\delta) = V_2(\delta) \cup V_3(\delta)$ denote the set of non-zero components, and let $V_u(\delta) = V_1(\delta) \cup V_2(\delta)$ denote the set of components which are smaller than ϵ. For any δ, if we want to update it to increase $F(\delta)$ in Eq. (9), then the values of some components belonging to $V_d(\delta)$ must decrease and the values of some components belonging to $V_u(\delta)$ must increase. By the above theorem regarding η, if δ is the solution to Eq. (9), then $R_p(\delta) \leq R_q(\delta), \forall v_p \in V_u(\delta), \forall v_q \in V_d(\delta)$. On the contrary, if $\exists v_p \in V_u(\delta)$, $\exists v_q \in V_d(\delta)$, $R_p(\delta) > R_q(\delta)$, then δ is not the solution to Eq. (9). In such cases, we can increase δ_p and decrease δ_q to increase $F(\delta)$ in Eq. (9), which can be written as follows:

$$\delta_l' = \begin{cases} \delta_l, & l \neq p, l \neq q, \\ \delta_l + \alpha, & l = p, \\ \delta_l - \alpha, & l = q, \end{cases} \tag{14}$$

We also define

$$R_{pq}(\delta) = \sum_{v_1, \dots, v_{m-2} \in V} \left(W(v_1, \dots, v_{m-2}, p, q) \cdot \prod_{t=1}^{m-2} \delta_t \right). \tag{15}$$

Since $R_p(\delta) > R_q(\delta)$, we can always select a proper $\alpha > 0$ to increase $F(\delta)$ in Eq. (9). According to Eq. (14) and the constraint over δ_p,

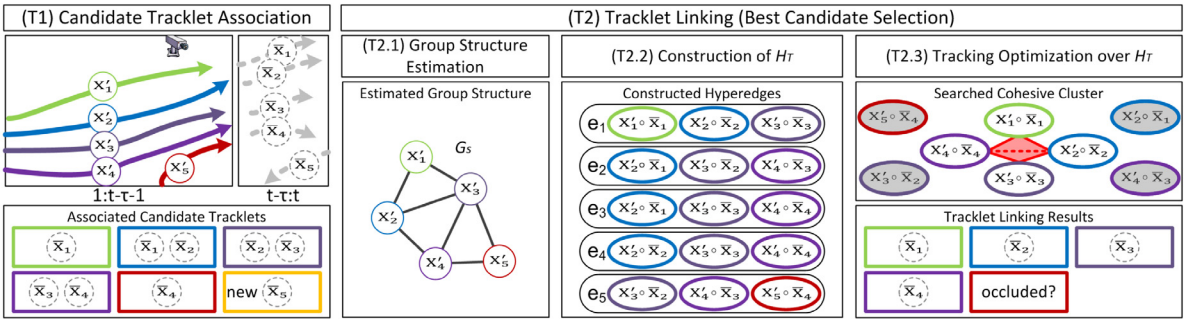

Fig. 2. Stage 1 activity-aware tracking. Given five targets $x'_1, \ldots, x'_5$, and new tracklets $\bar{x}_1, \ldots, \bar{x}_5$, step **(T1)** optimizes the association of candidate tracklets with existing target tracks. Step **(T2)** determines the best candidate assignments for tracklet linking in three steps. (T2.1) estimates the group structure using graph $\mathcal{G}_s$, where the edges represent the correlations of activities between individuals. (T2.2) constructs hypergraph $\mathcal{H}_T$ with hyperedges $e_1, \ldots, e_5$ based on the estimated group structure. (T2.3) solves the candidate tracklet linking and infers the possible occlusions in an optimization over $\mathcal{H}_T$.

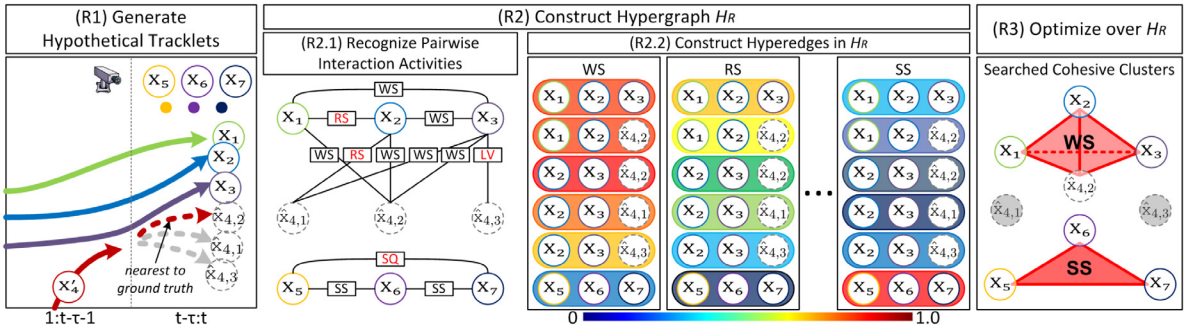

Fig. 3. Stage 2 joint interaction recognition and occlusion recovery in a road-crossing scenario, where x_1, x_2, x_3 and x'_4 are walking side-by-side across a road, while x_5, x_6, x_7 are standing side-by-side waiting. Step **(R1)** considers the linking of the occluded target x'_4 to three hypothetical tracklets $\hat{x}_{4,1}, \hat{x}_{4,2}, \hat{x}_{4,3}$. Step **(R2)** constructs hypergraph $\mathcal{H}_R$ for the inference in two steps. (R2.1) evaluates each pairwise interaction by calculating a confidence score, where wrongly assigned labels are depicted in red. (R2.2) constructs hyperedges based on the recognized pairwise interactions, where each hyperedge characterizes the likelihood of a pairwise interaction. Step **(R3)** optimizes the inference over $\mathcal{H}_R$ to jointly recognize interaction labels and resolve the tracklet linking and occlusion recovery.

$\alpha \leq \min(\delta_q, \epsilon - \delta_p)$. Since $R_p(\delta) > R_q(\delta)$, if $R_{pq} \leq 0$, then when $\alpha = \min(\delta_q, \epsilon - \delta_p)$, the increase of $F(\delta)$ reaches maximum; if $R_{pq} > 0$, then when $\alpha = \min\big(\delta_q, \epsilon - \delta_p, \frac{R_p(\delta) - R_q(\delta)}{2(m-1)R_{pq}(\delta)}\big)$, the increase of $F(\delta)$ reaches maximum.

Based on the above analysis, the computation process of the maximizer δ^* can be summarized as follows: if $\exists v_p \in V_u(\delta)$, $\exists v_q \in V_d(\delta)$, $R_p(\delta) > R_q(\delta)$, then we can update δ to increase $F(\delta)$ in Eq. (9). Such a updating procedure iterates until $R_p(\delta) \leq R_q(\delta), \forall v_p \in V_u(\delta), \forall v_q \in V_d(\delta)$. Within each iteration, we select the vertex with the largest reward from $V_u(\delta)$ and the vertex with the smallest reward from $V_d(\delta)$, and then update their corresponding components of δ.

3.3. Activity-aware tracking

Stage 1 of our method simultaneously recognizes individual activities and links tracklets in the following two steps (see Fig. 2 for a schematic overview). We use **(T)** to denote tracking steps:

- **(T1) Generate candidate tracklets** $\bar{X}$ from new detections D that maximizes $\log f_1(X, D)$ in Eq. (4).
- **(T2) Link tracklets** X' **with** $\bar{X}$ by maximizing the appearance, motion, and geometric consistencies that maximizes $\log \varphi_1(X, A)$ in Eq. (4).

(T1) Generate candidate tracklets $\bar{X}$. For each existing target $x'_i \in X'$, we generate a set of candidate tracklets $\bar{\mathbf{x}}_i = \{\bar{x}_{i,1}, \ldots, \bar{x}_{i,n}\}$ from observed detections D using the tracking method in Wen et al. (2014).[1] We employ a gating strategy to restrict the number of candidate tracklets to consider. The appearance similarity between x'_i and each tracklet $\bar{x}_j \in \bar{X}$ is calculated using the POI features (Yu et al., 2016) and Euclidean metric. If this similarity is above a threshold θ_a, $\bar{x}_j$ is added into $\bar{\mathbf{x}}_i$. Targets with no associated detection within time $[t-\tau-\tau_a, t-\tau-1]$ are discarded to reduce unnecessary computation. We use $\theta_a = 0.025$ and $\tau_a = 5$ *sec* to include a rich set of candidate tracklets for linking. If any tracklet in $\bar{X}$ ends up not linked with any target (*e.g.*, $\bar{x}_5$ in Fig. 2), a new target is created. If any target x'_i ends up with no linked tracklet for status update, it is considered occluded.[2]

(T2) Link tracklets X' **with** $\bar{X}$. After candidate tracklets $\bar{X}$ are generated, for each candidate tracklet $\bar{x}_i \in \bar{X}$, we determine its individual activity label $\bar{a}_i \in \bar{A}$ for the purpose of activity-aware tracking. We consider $n_A = 3$ individual activity labels regarding the motion pattern: *standing*, *walking*, and *running*, by calculating the velocity $\bar{v}_i$ of each $\bar{x}_i$ and modeling the posteriors using sigmoid similar to Chang et al. (2011): $p(\bar{a}_i|\bar{v}_i) \simeq p(\bar{v}_i|\bar{a}_i)p(\bar{a}_i)$. We consider social contextual cues and the correlations between individual activities in finding the best tracklet linking combinations. This also enables robust occlusion recovery for tracking. Our solution is to represent all terms using a **tracking hypergraph** $\mathcal{H}_T$. The clique potential function $\varphi_1(X, A)$ in Eq. (4) can then be inferred as

$$\varphi_1(X, A) \sim \sum_{\forall C_\alpha^T} \Psi(C_\alpha^T) \tag{16}$$

where C_α^T represents a cohesive cluster obtained from $\mathcal{H}_T$, α denotes cluster index, and $\Psi(\cdot)$ is the weighting function defined in Eq. (7).

The activity-aware tracking by linking tracklets X' with $\bar{X}$ is performed in three sub-steps: **(T2.1) estimate social group structure**

[1] All candidate tracklets and their labels are denoted with a *bar* $\bar{\cdot}$.

[2] We use trajectory prediction based on motion extrapolation in step **(R1)** of Section 3.4 to determine if the target is still within the scene.

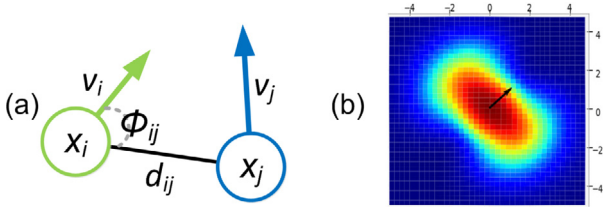

Fig. 4. Social group affinity between a pair of individuals is calculated based on: (a) distance, angle, and motion (velocity magnitude & direction). (b) visualizes such a measure at (0, 0) with direction vector (1, 1) arrow in a color map depicting the probability kernel between 0 (blue) and 1 (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

using correlations between individual activities in a graph representation. **(T2.2) construct hypergraph $\mathcal{H}_\mathcal{T}$. (T2.3) optimize tracking based on $\mathcal{H}_\mathcal{T}$.**

(T2.1) estimate social group structure. We represent the social group structure of tracked targets and the correlations between individual activities using an *undirected complete graph* $\tilde{\mathcal{G}} = \{\tilde{V}, \tilde{E}, \tilde{W}\}$ with $\tilde{V} = X'$. $\forall \mathrm{x}'_i, \mathrm{x}'_j, \exists e_{ij} = (\mathrm{x}'_i, \mathrm{x}'_j) \in \tilde{E}$. Edge weight $\tilde{W}(e_{ij})$ reflects the correlation between activities a_i and a_j of x'_i and x'_j, respectively. We define p_{corr} to reflect the correlation between activities of two targets similar to Chang et al. (2011):

$$p_{corr}(\mathrm{x}_i, \mathrm{x}_j) = g(d_{ij}, \phi_{ij}, \|v_i\|, \|v_j\|, \mathrm{a}_i, \mathrm{a}_j, \eta(\mathrm{x}_i), \eta(\mathrm{x}_j)), \tag{17}$$

where d_{ij} represents Euclidean distance between the targets. As shown in Fig. 4a, ϕ_{ij} represents the angle between the facing direction of x_i and the relative vector from x_i to x_j, and v_i represents the velocity of x_i. For a target x_i, if a_i is recognized as "standing", we use the classifier in Choi et al. (2009) to calculate the body orientation $\eta(\mathrm{x}_i)$ out of 8 quantizations. Otherwise, $\eta(\mathrm{x}_i)$ estimates motion direction from the trajectory. Edge weights of $\tilde{\mathcal{G}}$ are calculated according to Eq. (17) and refined using further grouping cues as in Chang et al. (2011). Fig. 4b visualizes the correlation defined by Eq. (17). The probability is higher on the side of a person than in the front or back, which is an implementation of Hall's *proxemics* social norms (Hall, 1966). We discard edges with weights lower than 0.3 to obtain a sparse graph denoted as $\tilde{\mathcal{G}}_s$ for computation speedup.

(T2.2) construct hypergraph $\mathcal{H}_\mathcal{T} = \{V_\mathcal{T}, E_\mathcal{T}, W_\mathcal{T}\}$ using $\tilde{\mathcal{G}}_s$ to capture the high-order correlations between activities within a group. A vertex $v_p \in V_\mathcal{T}$ represents a hypothesis of linking a tracked target with its candidate tracklet, *i.e*, $v_p = \mathrm{x}'_i \oplus \bar{\mathrm{x}}_{i,k}$ where "$\oplus$" represents the association of two tracklets. A m-degree hyperedge $\mathbf{e}^m \in E_\mathcal{T}$ represents the combination of m tracklet linking hypotheses in an assignment.

The linking of tracklets X' with $\bar{X}$ can be considered as an assignment problem with the following two **tracklet assignment constraints**: *(i)* a target cannot be linked with two or more candidate tracklets, and *(ii)* a candidate tracklet cannot be linked with two or more targets. We enforce these constraints in the construction of hyperedges in $\mathcal{H}_\mathcal{T}$. Specifically, $\forall v_p, v_q \in V_\mathcal{T}$, where $v_p = \mathrm{x}'_i \oplus \bar{\mathrm{x}}_{i,k}$ and $v_q = \mathrm{x}'_j \oplus \bar{\mathrm{x}}_{j,l}$, *if and only if* $e_{ij} = (\mathrm{x}'_i, \mathrm{x}'_j) \in \tilde{\mathcal{G}}_s$, v_p and v_q can co-exist in a hyperedge in $\mathcal{H}_\mathcal{T}$.

We further consider motion and behavior consistencies and their correlations (via $\tilde{\mathcal{G}}_s$) in determining the hyperedge weights. Specifically, we consider three affinities that determine the hyperedge weights: the *appearance* (W_a) of each tracklet, the *facing-direction* (W_d) and the *geometric similarity* (W_g) between tracked targets.

The appearance affinity between a target x'_i and a candidate tracklet $\bar{\mathrm{x}}_{i,k}$ is computed using the appearance features of tracklets as (Yu et al., 2016):

$$W_a(\mathbf{e}^m) = \sum_{\mathrm{x}'_i \oplus \bar{\mathrm{x}}_{i,k} \in \mathbf{e}^m} |\mathrm{x}'_i - \bar{\mathrm{x}}_{i,k}|. \tag{18}$$

We assume that activity states (such as walking direction) do not change abruptly in-between small linked tracklets. In other words, difference between facing directions of two targets should be small for linked tracklets:

$$W_d(\mathbf{e}^m) = \sum_{\mathrm{x}'_i \oplus \bar{\mathrm{x}}_{i,k} \in \mathbf{e}^m} \cos(\eta(\mathrm{x}'_i), \eta(\bar{\mathrm{x}}_{i,k})). \tag{19}$$

Our method aims to run on surveillance videos without calibration. To ensure smooth tracking, we use a geometric affinity term W_g to ensure that relative angles between two targets does not change abruptly:

$$W_g(\mathbf{e}^m) = \sum_{\mathrm{x}'_i \oplus \bar{\mathrm{x}}_{i,k} \in \mathbf{e}^m} \sum_{\mathrm{x}'_j \oplus \bar{\mathrm{x}}_{j,l} \in \mathbf{e}^m} \cos(\mathbf{p}'_{ij}, \bar{\mathbf{p}}_{ij}), \tag{20}$$

where $\mathbf{p}'_{ij}$ and $\bar{\mathbf{p}}_{ij}$ represent the relative image coordinate vectors between tracked targets and candidate tracklets. Final affinity value of a hyperedge $\mathbf{e}^m$ is computed by $W(\mathbf{e}^m) = \lambda_a W_a(\mathbf{e}^m) + \lambda_d W_d(\mathbf{e}^m) + \lambda_g W_g(\mathbf{e}^m)$, where λ_a, λ_d, λ_g are set as $\lambda_a = 30$, $\lambda_d = 1$, $\lambda_g = 0.5$.

(T2.3) optimize tracking based on $\mathcal{H}_\mathcal{T}$. This step aims to determine the optimal tracklet linking among candidates represented in the hypergraph $\mathcal{H}_\mathcal{T}$. The optimization is performed by the cohesive cluster search on $\mathcal{H}_\mathcal{T}$ described in Section 3.2. For each vertex v_r, such a search yields a cluster $C(v_r)$ with a score. Since a vertex may appear in multiple clusters, if any resulting cluster violates the *tracklet assignment constraints* in **(T2.2)**, such a cluster is removed to avoid further consideration. We ensure that the resulting cohesive clusters represent valid tracklet linking hypotheses that is sound and redundancy-free.[3] In case a target ends up not linked with any candidate tracklets (*e.g*, x'_5 in Fig. 3), such a target should be either outside the scene or under occlusion. We store all discovered occlusions and will try to recover them at **Stage 2** in Section 3.4. Finally, target trajectories X are updated with the newly linked tracklets in $\bar{X}$ to be X^*, and activity labels A are augmented with respective ones in $\bar{A}$ to be A^*.

3.4. Joint interaction recognition and occlusion recovery

Our approach is motivated from the observation that pairwise interactions I within a group can provide rich contextual cues to recognize the activities (as in Fig. 1) and recover possible occlusions. **Stage 2** of our method jointly resolves the two problems of (1) recognizing pairwise interactions I and (2) occlusion recovery to improve tracking. We again use a hypergraph representation to explore the high-order correlations among the interactions I, such that a similar cluster search scheme can be applied for optimization. Specifically, we construct the (activity) **recognition hypergraph** ($\mathcal{H}_\mathcal{R}$) based on the inferred target locations X^* and individual activities A^*. The optimization over $\mathcal{H}_\mathcal{R}$ maximizes the clique potential function $\log \varphi_2(A^*, I)\varphi_3(I, X^*)$ in Eq. (5) as,

$$\varphi_2(A^*, I)\varphi_3(I, X^*) \sim \sum_{\forall C^\mathcal{R}_\alpha} \Psi(C^\mathcal{R}_\alpha) \tag{21}$$

where $C^\mathcal{R}_\alpha$ represents a cohesive cluster obtained from $\mathcal{H}_\mathcal{R}$, and $\Psi(\cdot)$ is the weighting function defined in Eq. (7).

Stage 2 of our method jointly recognizes I and recovery occlusions in the following three main steps (see Fig. 3 for a schematic overview). We use **(R)** to denote recognition steps:

- **(R1) Generate hypothetical tracklets** $\hat{X}$ for occlusion recovery from given existing X' and A'.
- **(R2) Construct hypergraph** $\mathcal{H}_\mathcal{R}$ based on $X^*, A^*, \hat{X}$ to infer high-order correlations among their pairwise interactions I.
- **(R3) Optimize recognition and recovery over** $\mathcal{H}_\mathcal{R}$ to simultaneously recognize interaction I and link occluded targets with suitable hypothetical tracklets.

[3] Hypergraph clusters are processed sequentially in descending order of their scores. If any cluster violates the constraints, new cluster is discarded and any duplication is removed.

(R1) Generate hypothetical tracklets $\hat{X}$. For each possibly occluded target $\mathrm{x}'_i \in X'$, we generate a few hypothetical tracklets $\hat{\mathbf{x}}_i = \{\hat{\mathrm{x}}_{i,1}, \dots, \hat{\mathrm{x}}_{i,h}\}$ based on trajectory predictions, where h is empirically set to 9.[4] For a moving target x'_i with $\mathrm{a}'_i = \textit{walking}$, we generate $\hat{\mathbf{x}}_i$ via motion extrapolation. For a stationary target x'_i with $\mathrm{a}'_i = \textit{standing}$, we add a small perturbation to $\hat{\mathbf{x}}_i$.

(R2) Construct hypergraph $\mathcal{H}_{\mathcal{R}} = \{V_{\mathcal{R}}, E_{\mathcal{R}}, W_{\mathcal{R}}\}$, such that high-order correlations among interactions among X and $\hat{X}$ are captured for the purposes of simultaneous activity recognition and occlusion recovery. Thus, $V_{\mathcal{R}} = X \cup \hat{X}$. Each hyperedge in $E_{\mathcal{R}}$ characterizes the likelihood of a pairwise interaction $\mathrm{i} \in I$. For example in Fig. 3, $\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3$ are connected by 3 hyperedges, which correspond to interactions "WS", "RS", "SS", respectively. See Section 4 for a complete list of interaction class defined in public datasets (Choi et al., 2009; Choi and Savarese, 2012). We denote n_I the number of interaction classes.

The inference of each interaction class can be optimized independently. We can thus decompose $\mathcal{H}_{\mathcal{R}}$ into n_I sub-hypergraphs $\{\breve{\mathcal{H}}_\beta\}_{\beta=1}^{n_I}$, with $\breve{\mathcal{H}}_\beta = \{V_{\mathcal{R}}, \breve{E}_\beta, \breve{W}_\beta\}$ for the β-th interaction class. For each hyperedge $\mathbf{e}^m \in \breve{E}_\beta$, the weight $\breve{W}_\beta(\mathbf{e}^m)$ reflects how likely the interaction between the m targets are cohesive as a whole (*e.g*, all walking-side-by-wide).

We calculate the hyperedge weights in $\mathcal{H}_{\mathcal{R}}$ in two steps: **(R2.1)** evaluates each pairwise interaction with a confidence score. **(R2.2)** constructs hyperedges in $\mathcal{H}_{\mathcal{R}}$ using the average score from all involved targets.

(R2.1) recognize pairwise interaction activities. We calculate a confidence score for each possible pairwise interaction i_{ij} between the targets $\mathrm{x}_i, \mathrm{x}_j$ using a simple effective rule-based probabilistic approach as in Chang et al. (2011). Specifically, the confidence score of i_{ij} belonging to the β-th class is calculated by multiplying the following six component probabilities: *distance* (ds), *group connectivity* (gc) calculated in (17), *individual activity agreement* (aa), *distance change type* (dc), *facing direction* (dr), and *frontness/sidedness* (fs):

$$p(\mathrm{i}_{ij} = \beta | \mathrm{x}_i, \mathrm{x}_j, \mathrm{a}_i, \mathrm{a}_j) = p_{ds}(\beta|\mathrm{x}_i, \mathrm{x}_j) \cdot p_{gc}(\beta|\mathrm{a}_i, \mathrm{a}_j) \cdot p_{aa}(\beta|\mathrm{a}_i, \mathrm{a}_j) \cdot p_{dc}(\beta|\mathrm{x}_i, \mathrm{x}_j) \cdot p_{dr}(\beta|\mathrm{x}_i, \mathrm{x}_j) \cdot p_{fs}(\beta|\mathrm{x}_i, \mathrm{x}_j). \tag{22}$$

Detailed formulation of the above component probabilities and formulation are provided in Supplementary Table A.3 and Table A.4.

(R2.2) construct hyperedges in $\mathcal{H}_{\mathcal{R}}$. We consider interactions among both real and hypothetical targets during the optimization. We avoid the inclusion of multiple hypothetical tracklets of a target into a hyperedge. For each hyperedge $\mathbf{e}^m = \{\mathrm{x}_1^{\mathbf{e}}, \dots, \mathrm{x}_m^{\mathbf{e}}\} \in E_\beta$ for the β-th interaction class, we calculate the edge weight by averaging the confidence scores of the involved targets:

$$W(\mathbf{e}^m) = \frac{1}{\binom{m}{2}} \sum_{i,j} p(\mathrm{i}_{ij} = \beta | \mathrm{x}_i^{\mathbf{e}}, \mathrm{x}_j^{\mathbf{e}}, \mathrm{a}_i^{\mathbf{e}}, \mathrm{a}_j^{\mathbf{e}}). \tag{23}$$

(R3) Optimize recognition and recovery over $\mathcal{H}_{\mathcal{R}}$ cohesive cluster search on each sub-hypergraph $\breve{\mathcal{H}}_\beta$ respectively (as described in Section 3.2). This optimizes the assignment of interaction labels and the linking of probable hypothetical tracklets. Similar to (**T2.3**), for each vertex $v_r \in \mathcal{H}_{\mathcal{R}}$, we search for candidate cohesive clusters with confidence scores. We ensure that the resulting cohesive clusters are sound and redundancy-free, also not violating the *tracklet assignment constraints*. Optimization results are used to update X^*, A^*, I into $X^\ddagger, A^\ddagger, I^*$, respectively as in Eq. (5).

3.5. Collective activity recognition

Stage 3 of our method infers the collective activities C^* for each individual in a group, based on an intuition that collective activity is characterized by pairwise interactions I indexed by β within the group. Fig. 1 illustrates several examples, and Table A.4 in Supplementary shows the cohesive pairwise interaction for each collective activity class. For each target x_i within a group, we infer the most probable collective activity. The term $\log f_3(X^\ddagger, A^\ddagger, I^*, C)$ in Eq. (6) can be maximized based on a probabilistic formulation similar to Chang et al. (2011). Consider the β-th interaction for the cth collective activity, $p(c|\mathrm{x}_i)$ is calculated using:

$$p(c|\mathrm{x}_i) = 1 - \prod_{\forall j} \left(1 - p(\mathrm{i}_{ij} = \beta | \mathrm{x}_i, \mathrm{x}_j)\right), \tag{24}$$

where $p(\mathrm{i}_{ij} = \beta|\mathrm{x}_i, \mathrm{x}_j)$ is obtained in Eq. (13) after the optimization in (**R3**). The collective activity of x_i is determined by $\arg\max_c \; p(c|\mathrm{x}_i)$. We use the collective activity involving most participants as the label of the scene, to comply with the practice in major datasets (Choi and Savarese, 2012; Choi et al., 2009, 2011).[5]

4. Experimental results

Implementation. We implement our method in C++. Experiments are conducted on a machine with a i7-4800MQ CPU (2.8 GHz) and 16 GB RAM. We use the state-of-the-art person detections (Yu et al., 2016) as input, and employ deep re-identification features (Yu et al., 2016) as the appearance features for tracking. We set hyperedge degree $m = 3$ to balance the performance and speed. The whole pipeline runs in nearly real-time at approximately 20 FPS (not including the detection time). Note that input detectors can be executed in parallel for real-world applications.

Datasets. We perform evaluation on three popular collective activity recognition datasets, which are termed CAD (Choi et al., 2009), Augmented-CAD (Choi et al., 2011), and New-CAD (Choi and Savarese, 2012). Pedestrians in CAD and New-CAD are annotated with target IDs that can be used as ground truth for tracking evaluation.

CAD (Choi et al., 2009) comprises 44 video clips with annotations for $n_C = 5$ collective activities (CROSSING, WAITING, QUEUING, WALKING, TALKING), $n_I = 8$ pairwise interactions: *approaching* (AP), *leaving* (LV), *passing-by* (PB), *facing-each-other* (FE), *walking-side-by-side* (WS), *standing-in-a-row* (SR), *standing-side-by-side* (SS), *no-interaction* (NA), and $n_A = 2$ individual activities: *standing* and *walking*.

Augmented-CAD (Choi et al., 2011) is created by augmenting the CAD dataset. Collective activity WALKING is removed due to its ambiguities in definition, and 2 new collective activities DANCING, and JOGGING are included. For the newly introduced video clips, there are no annotations for interaction activities, individual activities, nor target identities.

New-CAD (Choi and Savarese, 2012) comprises 33 video clips with annotations for $n_C = 6$ collective activities: GATHERING, TALKING, DISMISSAL, WALKING-TOGETHER, CHASING, QUEUING, $n_I = 9$ pairwise interactions: *approaching* (AP), *walking-in-opposite-directions* (WO), *facing-each-other* (FE), *standing-in-a-row* (SR), *walking-side-by-side* (WS), *walking-one-after-the-other* (WR), *running-side-by-side* (RS), *running-one-after-the-other* (RR), *no-interaction* (NA), and $n_A = 3$ individual activities: *standing, walking, running*.

Experimental Setup. For evaluating activity recognition, we follow common protocols as in Choi and Savarese (2012) for CAD and New-CAD, and protocol of Choi et al. (2011) for Augmented-CAD. For evaluating tracking, we ensure fair comparison by running all tracking code using identical input detections.

Evaluation Metrics. For activity recognition, we adopt the metrics used in Choi and Savarese (2012), *i.e*, *overall classification accuracy* (OCA) and *mean-per-class-accuracy* (MCA) as in Table 1. Specifically, OCA measures the overall performance of activity recognition regardless of the occurrences of activity classes. MCA measures the average of recognition accuracy for each activity class. Note that the match-error-correction-rate used in Choi and Savarese (2012) only reflects tracking

[4] All hypothetical tracklets are denoted with a *hat* $\hat{}$ across the paper.

[5] If there are insufficient targets for interactive or collective activities (*e.g.* people leaving the scene), we keep existing labels for a short while.

Table 1
Activity recognition evaluation results in terms of accuracy and comparison with state-of-the-art methods. The CAD results are split into two tables: one for comparing methods only producing collective activity results, and the other for all three activities. Column "Time?" indicates whether this method makes use of the temporal information (✓) or not (✗).

CAD Choi et al. (2009)							
Time?	Method	Collective		Interaction		Individual	
		OCA	MCA	OCA	MCA	OCA	MCA
✗	LOG Lan et al. (2012)	79.7	78.4	-	-	-	-
	DSM Deng et al. (2015)	-	80.6	-	-	-	-
	SIE Deng et al. (2016)	-	81.2	-	-	-	-
	CK Hajimirsadeghi et al. (2015)	83.4	81.9	-	-	-	-
	CRM Azar et al. (2019)	-	85.8	-	-	-	-
✓	LCC Choi et al. (2011)	-	70.9	-	-	-	-
	FM Khamis et al. (2012b)	-	70.9	-	-	-	-
	CFT Khamis et al. (2012a)	-	75.1	-	-	-	-
	LLC Antic and Ommer (2014)	-	75.1	-	-	-	-
	HDT Ibrahim et al. (2016)	-	81.5	-	-	-	-
	HACM Kong et al. (2018)	-	84.3	-	-	-	-
	MCTS Amer et al. (2013)	-	88.9	-	-	-	-
	StagNet Qi et al. (2018)	-	89.1	-	-	-	-
	RMI Wang et al. (2017)	-	89.4	-	-	-	-
	ARG Wu et al. (2019)	-	91.0	-	-	-	-
	HiRF Amer et al. (2014)	-	92.0	-	-	-	-
	UTR-1 Choi and Savarese (2012, 2014)	79.0	79.6	56.2	50.8	-	-
	UTR-2 Choi and Savarese (2012, 2014)	79.4	80.2	45.5	36.6	-	-
	Baseline	87.8	88.0	65.4	48.4	87.3	88.3
	Ours w/o $\mathcal{H}_\mathcal{T}$	91.8	91.8	74.3	55.6	**87.7**	88.4
	Ours w/o $\mathcal{H}_\mathcal{R}$	88.0	87.8	71.4	56.5	87.3	88.4
	Ours w/o $\mathcal{H}_\mathcal{T}$, $\mathcal{H}_\mathcal{R}$	87.9	87.6	67.3	53.7	87.4	**88.6**
	Ours	**92.5**	**92.4**	**78.1**	**57.6**	87.4	88.1

Augmented CAD Choi et al. (2011)							
Method	Collective		Interaction		Individual	
	OCA	MCA	OCA	MCA	OCA	MCA
LCC Choi et al. (2011)	-	82.0	-	-	-	-
FM Khamis et al. (2012b)	-	83.7	-	-	-	-
CFT Khamis et al. (2012a)	-	85.8	-	-	-	-
LLC Antic and Ommer (2014)	-	90.1	-	-	-	-
SIE Deng et al. (2016)	-	90.2	-	-	-	-
Baseline	89.4	89.3	-	-	-	-
Ours w/o $\mathcal{H}_\mathcal{T}$	88.9	89.0	-	-	-	-
Ours w/o $\mathcal{H}_\mathcal{R}$	85.2	84.3	-	-	-	-
Ours w/o $\mathcal{H}_\mathcal{T}$, $\mathcal{H}_\mathcal{R}$	84.9	84.2	-	-	-	-
Ours	**95.1**	**94.3**	-	-	-	-

New CAD Choi and Savarese (2012)						
Method	Collective		Interaction		Individual	
	OCA	MCA	OCA	MCA	OCA	MCA
UTR-1 Choi and Savarese (2012, 2014)	80.8	77.0	54.3	46.3	-	-
UTR-2 Choi and Savarese (2012, 2014)	83.0	79.2	53.3	43.7	-	-
MCTS Amer et al. (2013)	-	84.2	-	-	-	-
RMI Wang et al. (2017)	89.4	85.2	-	-	-	-
HiRF Amer et al. (2014)	-	87.3	-	-	-	-
Baseline	95.3	89.0	70.0	72.1	85.7	92.6
Ours w/o $\mathcal{H}_\mathcal{T}$	**97.0**	**89.3**	76.5	72.2	**90.2**	**93.2**
Ours w/o $\mathcal{H}_\mathcal{R}$	96.8	88.0	74.7	74.1	90.0	93.1
Ours w/o $\mathcal{H}_\mathcal{T}$, $\mathcal{H}_\mathcal{R}$	96.8	88.0	73.9	73.2	90.0	**93.2**
Ours	**97.0**	**89.3**	**78.6**	**74.7**	90.0	**93.2**

Table 2
Tracking evaluation and comparison with state-of-the-art methods. ↑ and ↓ represent "the-higher-the-better" and "the-lower-the-better", respectively. Bold highlights best results.

Dataset	Method	Rcll ↑	Prcn ↑	FAR ↓	MT (%) ↑	ML (%) ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑
CAD (Choi et al., 2009)	H^2T (Wen et al., 2014)	**84.3**	83.8	0.90	74.5	5.4	23195	**22368**	474	722	67.6	67.5
	JPDA (Rezatofighi et al., 2015)	84.1	85.5	0.79	74.0	5.4	20339	22600	348	901	69.6	63.7
	DCEM (Milan et al., 2016)	51.4	84.3	**0.53**	32.3	16.2	**13617**	69127	801	1025	41.2	63.5
	POI (Yu et al., 2016)	82.3	76.0	1.43	72.1	5.2	36944	25146	351	1262	56.1	**67.8**
	Ours w/o $\mathcal{H}_\mathcal{T}$	84.0	86.3	0.74	74.0	5.2	18991	22717	355	623	70.4	67.6
	Ours w/o $\mathcal{H}_\mathcal{R}$	84.0	85.1	0.82	74.0	**4.8**	21002	22684	319	638	69.0	67.6
	Ours w/o $\mathcal{H}_\mathcal{T}$, $\mathcal{H}_\mathcal{R}$	84.0	84.8	0.83	**74.7**	5.2	21362	22717	360	630	68.7	67.6
	Ours	84.1	**86.6**	0.72	74.5	5.2	18461	22647	**287**	**619**	**70.9**	67.6
New CAD (Choi and Savarese, 2012)	H^2T (Wen et al., 2014)	87.4	88.3	0.37	81.5	1.6	7883	8572	117	232	75.6	**64.7**
	JPDA (Rezatofighi et al., 2015)	87.6	88.6	0.36	82.6	1.4	7660	8413	65	198	76.3	62.3
	DCEM (Milan et al., 2016)	68.5	88.5	**0.28**	41.9	7.3	**6031**	21458	220	283	59.3	62.4
	POI (Yu et al., 2016)	87.4	**89.0**	0.34	82.3	**0.8**	7331	8594	**52**	271	76.5	**64.7**
	Ours w/o $\mathcal{H}_\mathcal{T}$	87.9	**89.0**	0.35	83.9	**0.8**	7411	8226	63	198	**76.9**	**64.7**
	Ours w/o $\mathcal{H}_\mathcal{R}$	87.9	88.9	0.35	83.9	**0.8**	7439	8259	62	**195**	76.8	**64.7**
	Ours w/o $\mathcal{H}_\mathcal{T}$, $\mathcal{H}_\mathcal{R}$	88.1	88.1	0.36	83.9	**0.8**	7726	**8096**	59	202	76.7	**64.7**
	Ours	**88.2**	88.7	0.36	**84.7**	**0.8**	7630	8508	60	202	**76.9**	**64.7**

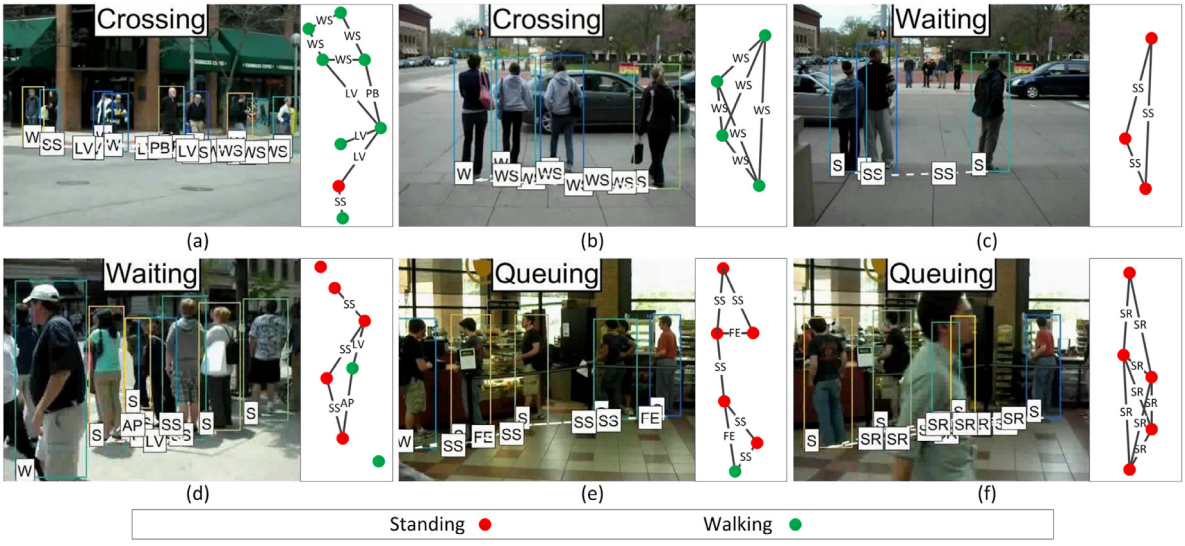

Fig. 5. Recognized collective activity examples in the CAD dataset (Choi et al., 2009): (a, b) CROSSING, (c, d) WAITING, and (e, f) QUEUING. Two individual activities *walking* (W) and *standing* (S) are detected on each target, while $n_I = 8$ possible pairwise interactions: *approaching* (AP), *leaving* (LV), *passing-by* (PB), *facing-each-other* (FE), *walking-side-by-side* (WS), *standing-in-a-row* (SR), *standing-side-by-side* (SS), *no-interaction* (NA) are detected among the pairs of targets. A top-down view of each scene is illustrated on the right.

fragmentation and identity switch. Instead, we adopt the more widely-used CLEAR MOT as tracking metrics to provide further insights for analysis.

CLEAR MOT includes 11 sub-metrics which are defined as follows. The Multi-Object Tracking Accuracy (MOTA) combines all errors (False Negatives (FN), False Positives (FP), Identity Switches (IDs))

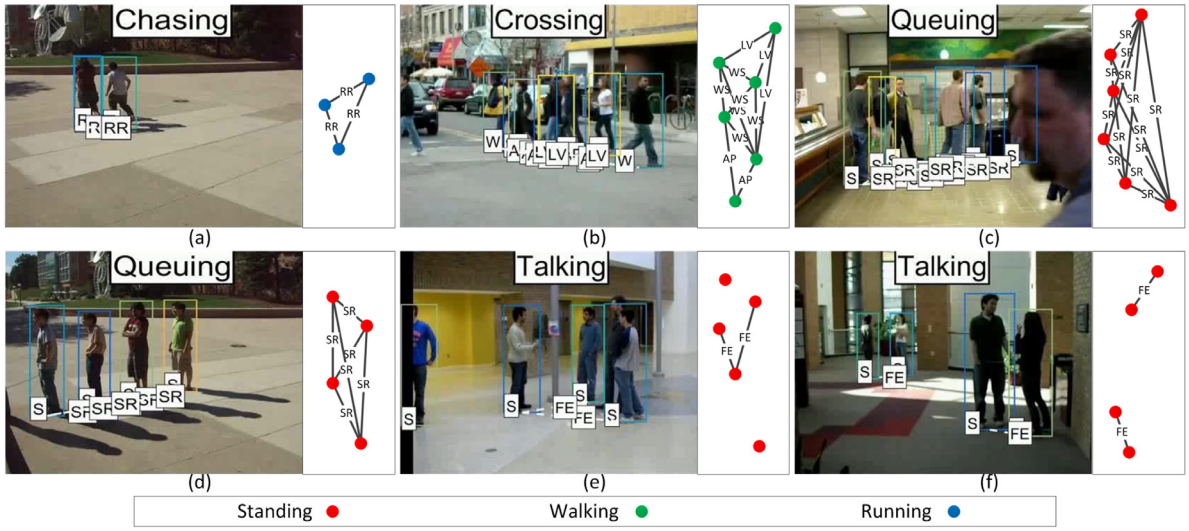

Fig. 6. Recognized collective activity examples in the NewCAD dataset (Choi and Savarese, 2012): (a) CHASING, (b) CROSSING, (c, d) QUEUING, and (e, f) TALKING. Three individual activities *walking* (W), *standing* (S), and running (R) are detected on each target, while $n_I = 9$ possible pairwise interactions: *approaching* (AP), *walking-in-opposite-directions* (WO), *facing-each-other* (FE), *standing-in-a-row* (SR), *walking-side-by-side* (WS), *walking-one-after-the-other* (WR), *running-side-by-side* (RS), *running-one-after-the-other* (RR), *no-interaction* (NA) are detected among the pairs of targets. A top-down view of each scene is illustrated on the right.

into a single number. The Multi-Object Tracking Precision (MOTP) averages the bounding box overlap over all tracked targets as a measure of localization accuracy. Mostly Lost (ML) and Mostly Tracked (MT) scores are computed on the entire trajectories and measure how many ground truth trajectories are lost (tracked for less than 20% of their life span) and tracked successfully (tracked for at least 80%). Other metrics include Recall (Rcll), Precision (Prcn), Fragmentations of the target trajectories (FM) and False Alarms per Frame (FAR). Generally speaking, MOTA is the most important sub-metric for CLEAR MOT.

Methods for Comparison. We compare our method with 17 existing activity recognition methods,[6] including 4 detection based methods (Azar et al., 2019; Deng et al., 2016, 2015; Hajimirsadeghi et al., 2015), 13 tracking based methods (Amer et al., 2013, 2014; Antic and Ommer, 2014; Choi and Savarese, 2012, 2014; Choi et al., 2011; Ibrahim et al., 2016; Khamis et al., 2012b,a; Kong et al., 2018; Qi et al., 2018; Wang et al., 2017; Wu et al., 2019), and a few baseline methods created by ourselves. The compared detection based methods take the ground-truth detection annotations as input. Our baseline methods accept the tracking results of Wen et al. (2014) as input, and recognize activities using only our probabilistic rules *e.g.*, Eqs. (17), (22), (24) with details in Supplementary Table A.4, but not the complete staged hypergraph optimizers. For tracking evaluation, we compare against 4 state-of-the-art trackers (Milan et al., 2016; Rezatofighi et al., 2015; Wen et al., 2014; Yu et al., 2016) with available code. We also develop several variants of our method by replacing the hypergraph formulations with the ordinary graph formulations. This justifies the effect of hypergraph formulations *w.r.t.* performance in both activity recognition and tracking.

4.1. Results and analysis for activity recognition

Table 1 shows the evaluation results in terms of accuracy of our method and others on three activity recognition datasets. Part of our recognition results are shown in Figs. 5 and 6. The efficacy of our method for individual activity recognition is demonstrated by the high accuracy score of approximately 90%. The performance of our method is significantly better than (Choi and Savarese, 2012, 2014), which are, to the best of our knowledge, the only works that evaluate pairwise interactions.

[6] Azar et al. (2019), Kong et al. (2018), Qi et al. (2018) and Wu et al. (2019) are methods proposed in recent one year.

Necessity of Addressing Tracking and Activity Recognition Simultaneously. We present methods which are independent with the temporal information in the upper part of Table 1 (indicating by ✗), and present those depending on the temporal information in the lower part. It appears that the temporal information based methods can achieve significantly better performance than those temporal-independent ones. Note that the temporal information of activities are provided by the person tracking results. Therefore, we argue that it is necessary for us to study how to improve the tracking performance to benefit the activity recognition.

The Basis of Symbolic Inference. In our method, the inference of higher-level activities is based on the symbolic cues instead of the group-level visual representation. The basic assumption of this method is that the lower-level activities can be inferred relatively robustly. This assumption can be demonstrated by the favorable performance of individual activity recognition and interaction recognition.

Symbolic Approaches vs. Representation Learning. Apart from our method, HiRF (Amer et al., 2014) and MCTS (Amer et al., 2013) are also based on symbolic inference. These symbolic methods perform favorably against those based on representation learning, such as HACM (Kong et al., 2018), StagNet (Qi et al., 2018), RMI (Wang et al., 2017) and ARG (Wu et al., 2019). Symbolic approaches are advantageous in avoiding intra-class variation issues in visual representation learning. Such advantage can be partially demonstrated by the performance gap between symbolic approaches (including ours) and other representation learning based approaches.

Effectiveness of Cohesive Cluster Search. HiRF (Amer et al., 2014) is the best performing symbolic inference based approaches except ours, which achieves the closest performance to ours on the CAD dataset. However, our method outperforms HiRF by a larger margin on the New CAD dataset. As discussed in Section 2, HiRF is built upon on a hierarchical conditional random field, which lacks an effective mechanism to model/identify group components and structures. Therefore, it is hard for HiRF to deal with activity classes that require accurate group estimation, *e.g.*, GATHERING, and DISMISSAL in the New CAD dataset. On the contrary, our cohesive cluster search approaches explicitly capture the group components and structures during the inference process of tracking and activity recognition, which directly contribute to our performance improvement over HiRF.

Orders of Graphs for Cohesive Cluster Search. In Tables 1 and 2, our methods "w/o $\mathcal{H}_T$" or "w/o $\mathcal{H}_R$" are ablation studies without using hypergraph, thus the cohesive cluster search is performed on graphs of

Fig. 7. Example tracking results. For each sequence, we show two frames across a long period of time, where the subjects have experienced heavy occlusions. The number of target ID changes are shown in each experiment. Colorful diamonds visualize the corresponding target ID changes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

order 2. In comparison, the inclusion of $\mathcal{H}_\mathcal{T}$ or $\mathcal{H}_\mathcal{R}$ uses hypergraphs of order 3. Comparison with the three variants of our method shows that both hypergraphs $\mathcal{H}_\mathcal{T}$ (Section 3.3) and $\mathcal{H}_\mathcal{R}$ (Section 3.4) contribute to the improvements in the collective and interaction activity recognition. The experiments show that $\mathcal{H}_\mathcal{R}$ is more influential than $\mathcal{H}_\mathcal{T}$, which is expected, as the main purpose of $\mathcal{H}_\mathcal{T}$ is to improve tracking and serve as a base for activity recognition. We also notice positive correlations

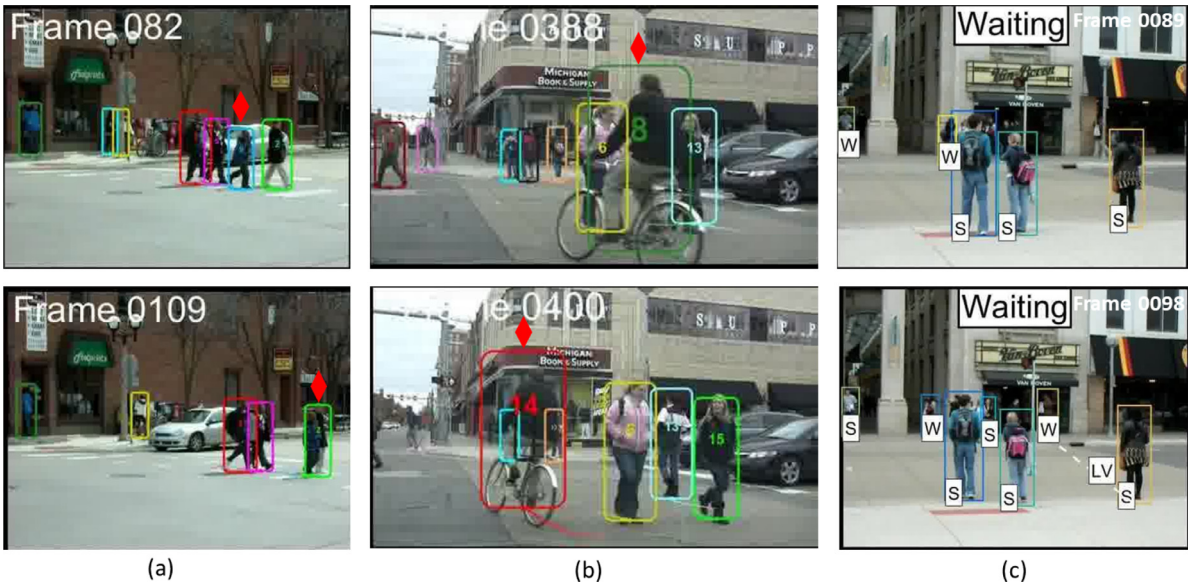

Fig. 8. Failure cases of our method on multi-person tracking and collective activity recognition. (a) shows one failure case of target missing in multi-person tracking (marked with red diamond on top of the target bounding box), (b) shows one ID switch failure (marked with red diamond on top of the target bounding box), (c) shows the activity recognition failure in the bottom frame.

between the recognition accuracy of collective activities and interactions. Finally, we do not observe performance gain when we increase the hypergraph order to 4, which comes with extra computational overhead.

4.2. Results and analysis for multi-person tracking

Table 2 shows the comparison of our method and existing tracking methods on the CAD and New-CAD. On the CAD dataset, our method achieves the best performance in most measures, *e.g*, MOTA, FM, IDs, MT, Prcn and Rcll. This is due to the incorporation of the high-order correlations in $\mathcal{H}_{\mathcal{T}}$ and $\mathcal{H}_{\mathcal{R}}$. Specifically, we use $\mathcal{H}_{\mathcal{T}}$ to model high-order correlations of individual activities, which improves the tracklet association. The use of $\mathcal{H}_{\mathcal{R}}$ to model high-order correlations of interaction activities improves occlusion recovery. This is further confirmed that after replacing $\mathcal{H}_{\mathcal{T}}$ or $\mathcal{H}_{\mathcal{R}}$ with ordinary graphs, the tracking performance decreases in most measures. New-CAD is less challenging than CAD in terms of tracking, because there are fewer occlusions and crossing occasions. Thus, many compared methods yield performances closer toward saturation. However, our method still achieves the highest score in several measures, *i.e*, MOTA, MT, and Rcll. Both (Wen et al., 2014) and our method rely on the cohesive cluster search on the hypergraph, but our method consistently outperforms (Wen et al., 2014) by a significant margin because of *(i)* the modeling of high-order correlations of individual activities, and *(ii)* successful occlusion recovery. We visualize the tracking result comparisons for several sequences in Fig. 7. For each sequence, we show two frames across a along period of time, where the subjects have experienced heavy occlusions. It is clear that our method (especially the one with hypergraph optimizers) produces the fewest ID changes, thus it is more robust than the competing methods for occlusion handling.

4.3. Failure cases

Although our method achieves better performance than the baseline methods, it fails in some extreme cases like occlusions and abrupt motion changes. We show some failure cases in Fig. 8. In column (a) and (b) we show two failure cases of multi-person tracking where we marked the failure target with a red diamond on top of its bounding box. Fig. 8(a) shows a target is missing due to the severe occlusion of two tracklets when it is hard to distinguish the two persons with appearance or movement. Since the target is absent in the view for a while, it is hard to recover the target from such severe occlusions. In Fig. 8(b), the person on bike has an ID switch in the two frames, the potential reasons are two-fold: *(i)* the movement of bike is too fast for tracker to predict its future status; *(ii)* the ambiguous appearance caused by the frequent and large overlap with other targets may confuse the tracker and thus lead to the ID switch. Fig. 8(c) shows a failure case of collective activity recognition, the activity label should have changed to be “crossing" at frame 0098 (bottom). In this case, our method is numb to the abrupt motion changes.

4.4. Time complexity analysis

We provide time complexity analysis of the cohesive cluster search, as it is the core of the whole tracking and activity recognition pipeline. As indicated in Section 3.3 and Section 3.4, the cohesive cluster search has two major phases, *i.e*, *(i)* hypergraph construction and *(ii)* optimization. Let the number of graph/hypergraph vertices for cohesive cluster search to be n, the averaging number of neighbors of each vertex to be h, the number of non-empty entries of the affinity matrix constructed during phase *(i)* to be m, and the number of iterations for optimization to be t. For phase *(i)*, since we need to compute hyperedge weights between each vertex and its neighbors, the time complexity for the hypergraph construction process is thus $O(nh)$. For phase *(ii)*, the time complexity is closely related to the detailed implementation and the used data structures. We adopt the implementation of Liu et al. (2010), so the time complexity is the same as that reported in Liu et al. (2010), which is $O(nt(h + log(n) + hlog(n)) + nm + n^2)$.

5. Conclusion

We present a novel multi-stage framework for solving the joint tasks of multi-person tracklet analysis and group activity recognition. By explicit modeling of correlations among individual activities, pairwise interactions, and collective activities using hypergraphs, we can effectively improve recognition and tracking with cohesive cluster searches. Our method can track targets with occlusion recovery, identify correlated pairwise interactions, and recognize group collective activities. Experimental evaluations demonstrate that our method outperforms state-of-the-art methods in both tasks of tracklet analysis and activity recognition. Our method runs in nearly real-time (not counting input

detections), and is applicable to a variety of real-world applications including video surveillance and situational awareness. Implementation code will be released upon the publication of this work.

CRediT authorship contribution statement

Wenbo Li: Conception and design of study, Analysis of data, Writing – original draft, Writing – review & editing. **Yi Wei:** Conception and design of study, Analysis of data, Writing – original draft, Writing – review & editing. **Siwei Lyu:** Conception and design of study, Analysis of data, Writing – original draft, Writing – review & editing. **Ming-Ching Chang:** Conception and design of study, Analysis of data, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cviu.2021.103301.

References

Aggarwal, J.K., Ryoo, M.S., 2011. Human activity analysis: A review. ACM Comput. Surv. 43 (3), 16:1–16:43.

Amer, M.R., Lei, P., Todorovic, S., 2014. HiRF: Hierarchical random field for collective activity recognition in videos. In: European Conference on Computer Vision, pp. 572–585.

Amer, M.R., Todorovic, S., Fern, A., Zhu, S., 2013. Monte Carlo tree search for scheduling activity recognition. In: IEEE International Conference on Computer Vision, pp. 1353–1360.

Antic, B., Ommer, B., 2014. Learning latent constituents for recognition of group activities in video. In: European Conference on Computer Vision, pp. 33–47.

Azar, S.M., Atigh, M.G., Nickabadi, A., Alahi, A., 2019. Convolutional relational machine for group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7892–7901.

Bagautdinov, T.M., Alahi, A., Fleuret, F., Fua, P., Savarese, S., 2017. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4315–4324.

Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N., 2016. A unified multi-scale deep convolutional neural network for fast object detection. In: European Conference on Computer Vision, pp. 354–370.

Chang, M., Krahnstoever, N., Ge, W., 2011. Probabilistic group-level motion analysis and scenario recognition. In: IEEE International Conference on Computer Vision, pp. 747–754.

Choi, W., Savarese, S., 2012. A unified framework for multi-target tracking and collective activity recognition. In: European Conference on Computer Vision, pp. 215–230.

Choi, W., Savarese, S., 2014. Understanding collective activitiesof people from videos. IEEE Trans. Pattern Anal. Mach. Intell. 36 (6), 1242–1257.

Choi, W., Shahid, K., Savarese, S., 2009. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In: IEEE International Conference on Computer Vision Workshops, pp. 1282–1289.

Choi, W., Shahid, K., Savarese, S., 2011. Learning context for collective activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3273–3280.

Deng, Z., Vahdat, A., Hu, H., Mori, G., 2016. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4772–4781.

Deng, Z., Zhai, M., Chen, L., Liu, Y., Muralidharan, S., Roshtkhari, M.J., Mori, G., 2015. Deep structured models for group activity recognition. In: British Machine Vision Conference, pp. 179.1–179.12.

Hajimirsadeghi, H., Yan, W., Vahdat, A., Mori, G., 2015. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2596–2605.

Hall, E.T., 1966. The Hidden Dimension. Anchor Books.

Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G., 2016. A hierarchical deep temporal model for group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1971–1980.

Khamis, S., Morariu, V.I., Davis, L.S., 2012a. A flow model for joint action recognition and identity maintenance. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1218–1225.

Khamis, S., Morariu, V.I., Davis, L.S., 2012b. Combining per-frame and per-track cues for multi-person action recognition. In: European Conference on Computer Vision, pp. 116–129.

Kong, L., Qin, J., Huang, D., Wang, Y., Gool, L.V., 2018. Hierarchical attention and context modeling for group activity recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1328–1332.

Lan, T., Wang, Y., Yang, W., Robinovitch, S.N., Mori, G., 2012. Discriminative latent models for recognizing contextual group activities. IEEE Trans. Pattern Anal. Mach. Intell. 34 (8), 1549–1562.

Liu, H., Latecki, L.J., Yan, S., 2010. Robust clustering as ensembles of affinity relations. In: Advances in Neural Information Processing Systems. pp. 1414–1422.

Luo, W., Zhao, X., Kim, T., 2014. Multiple object tracking: A review. CoRR, arXiv:1409.7618.

Milan, A., Schindler, K., Roth, S., 2016. Multi-target tracking by discrete-continuous energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. 38 (10), 2054–2068.

Qi, M., Qin, J., Li, A., Wang, Y., Luo, J., Gool, L.V., 2018. stagNet: An attentive semantic RNN for group activity recognition. In: European Conference on Computer Vision, pp. 104–120.

Rezatofighi, S.H., Milan, A., Zhang, Z., Shi, Q., Dick, A.R., Reid, I.D., 2015. Joint probabilistic data association revisited. In: IEEE International Conference on Computer Vision, pp. 3047–3055.

Vinciarelli, A., Pantic, M., Bourlard, H., 2009. Social signal processing: Survey of an emerging domain. Image Vis. Comput. 27 (12), 1743–1759.

Wang, M., Ni, B., Yang, X., 2017. Recurrent modeling of interaction context for collective activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3048–3056.

Wen, L., Li, W., Yan, J., Lei, Z., Yi, D., Li, S.Z., 2014. Multiple target tracking based on undirected hierarchical relation hypergraph. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1282–1289.

Wu, J., Wang, L., Wang, L., Guo, J., Wu, G., 2019. Learning actor relation graphs for group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 9964–9974.

Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J., 2016. POI: Multiple object tracking with high performance detection and appearance feature. In: European Conference on Computer Vision Workshops, pp. 36–42.

Wenbo Li is a Senior AI Research Scientist at Samsung Research America AI Center. He received his Ph.D. degree in Department of Computer Science, University at Albany, State University of New York (SUNY) in 2019. During 2014–2016, He was involved in the Ph.D. program of Department of Computer Science and Engineering, Lehigh University. He received his B.Eng. degree in School of Computer Software, Tianjin University in 2014. Dr. Li's expertise includes video analytics and image and video synthesis. He has authored more than 20 technical papers, and served as reviewers for many academic conferences and journals such as CVPR, ICCV, ECCV, NeurIPS, ICLR, AAAI, IJCAI, TPAMI, IJCV, TIP, *etc*.

Yi Wei is a Senior AI Research Scientist at Samsung Research America AI Center. He received his Ph.D. degree in Department of Computer Science, University at Albany, State University of New York (SUNY) in 2021 under the supervision of Prof. Ming-Ching Chang. He received M.S. degree in Computer Science in 2016 and B.S. degree in Computer Science in 2013 at Shandong University. His research interest is mainly focused on activity recognition and image editing.

Siwei Lyu is a SUNY Empire Innovation Professor at the Department of Computer Science and Engineering, the Director of UB Media Forensic Lab (UB MDFL). Before joining UB, Dr. Lyu was an Assistant Professor from 2008 to 2014, a tenured Associate Professor from 2014 to 2019, and a Full Professor from 2019 to 2020, at the Department of Computer Science, University at Albany, State University of New York. Dr. Lyu received his Ph.D. degree in Computer Science from Dartmouth College in 2005, and his M.S. degree in Computer Science in 2000, and B.S. degree in Information Science in 1997, both from Peking University, China. Dr. Lyu's research interests include digital media forensics, computer vision, and machine learning. Dr. Lyu has published over 170 refereed journal and conference papers.

Ming-Ching Chang is an Assistant Professor at the Department of Computer Science, College of Engineering and Applied Sciences (CEAS), University at Albany, State University of New York (SUNY). He was with the Department of Electrical and Computer Engineering from 2016 to 2018. During 2008–2016, he was a Computer Scientist at GE Global Research Center. He received his Ph.D. degree in the Laboratory for Engineering Man/Machine Systems (LEMS), School of Engineering, Brown University in 2008. He was an Assistant Researcher at the Mechanical Industry Research Labs, Industrial Technology Research Institute (ITRI) at Taiwan from 1996 to 1998. He received his M.S. degree in Computer Science and Information Engineering (CSIE) in 1998 and B.S. degree in Civil Engineering in 1996, both from National Taiwan University. Dr. Chang's expertise includes video analytics, computer vision, image processing, and artificial intelligence. His research projects are funded by GE Global Research, IARPA, DARPA, NIJ, VA, and UAlbany. He is the recipient of the IEEE Advanced Video and Signal-based Surveillance (AVSS) 2011 Best Paper Award - Runner-Up, the IEEE Workshop on the Applications of Computer Vision (WACV) 2012 Best Student Paper Award, the GE Belief - Stay Lean and Go Fast Management Award in 2015, and the IEEE Smart World NVIDIA AI City Challenge 2017 Honorary Mention Award. Dr. Chang serves as Co-Chair of the annual AI City Challenge CVPR 2018-2021 Workshop, Co-Chair of the IEEE Lower Power Computer Vision (LPCV) Annual Contest and Workshop 2019-2021, Program Chair of the IEEE Advanced Video and Signal-based Surveillance (AVSS) 2019, Co-Chair of the IWT4S 2017–2019, Area Chair of IEEE ICIP (2017, 2019–2021) and ICME (2021), TPC Chair for the IEEE MIPR 2022. He has authored more than 96 peer-reviewed journal and conference publications, 7 US patents and 15 disclosures. He is a senior member of IEEE and member of ACM.