Host–Parasite: Graph LSTM-in-LSTM for Group Activity Recognition

Xiangbo Shu, Liyan Zhang, Yunlian Sun, and Jinhui Tang, *Senior Member, IEEE*

***Abstract*— This article aims to tackle the problem of group activity recognition in the multiple-person scene. To model the group activity with multiple persons, most long short-term memory (LSTM)-based methods first learn the person-level action representations by several LSTMs and then integrate all the person-level action representations into the following LSTM to learn the group-level activity representation. This type of solution is a two-stage strategy, which neglects the "host–parasite" relationship between the group-level activity ("host") and person-level actions ("parasite") in spatiotemporal space. To this end, we propose a novel graph LSTM-in-LSTM (GLIL) for group activity recognition by modeling the person-level actions and the group-level activity simultaneously. GLIL is a "host–parasite" architecture, which can be seen as several person LSTMs (P-LSTMs) in the local view or a graph LSTM (G-LSTM) in the global view. Specifically, P-LSTMs model the person-level actions based on the interactions among persons. Meanwhile, G-LSTM models the group-level activity, where the person-level motion information in multiple P-LSTMs is selectively integrated and stored into G-LSTM based on their contributions to the inference of the group activity class. Furthermore, to use the person-level temporal features instead of the person-level static features as the input of GLIL, we introduce a residual LSTM with the residual connection to learn the person-level residual features, consisting of temporal features and static features. Experimental results on two public data sets illustrate the effectiveness of the proposed GLIL compared with state-of-the-art methods.**

***Index Terms*— Deep learning, graph LSTM (G-LSTM), group activity recognition, long short-term memory (LSTM).**

I. Introduction

SINGLE-PERSON action recognition, aiming to understand the action performed by a single person (e.g., running and jumping), has achieved great progress for the past decades [1]–[4]. Compared with the single-person action, a group/collective activity usually indicates a more complex activity scene involving at least two persons' actions, e.g., people are talking and people are queuing. Since a group activity contains several person-level actions from two or more persons, group activity recognition becomes more challenging than single-person action recognition [5]–[8].

In the early stages, researchers used various graphical models to tackle the problem of group activity recognition, e.g., hierarchical graphical models [9], AND–OR graphs [10], dynamic Bayesian networks [11], and classical neural networks [12]. Recently, witnessing the success of recurrent neural network (RNN) [13] and long short-term memory (LSTM) [14] in modeling the sequence data, researchers attempted to use RNN to address the problem of the group activity recognition [15]–[18]. A common two-stage solution is that it first learns person-level action representation by several LSTMs and then integrates all the person-level action representations to learn the group-level activity representation by another LSTM. Such a two-stage solution achieves a significant improvement of the recognition accuracy compared with traditional methods on group activity recognition.

However, the abovementioned two-stage solution ignores the important "host–parasite" relationship between the group-level activity ("host") and the person-level actions ("parasite"). Obviously, in an activity scene within multiple persons, the person-level actions and group-level activity are co-occurrence over time. Thus, the persons-level actions of the individuals and group-level activity of the scene should be simultaneously modeled by multiple RNNs. In the local view, most of the person-level actions participate in the group-level activity and decide the class of group-level activity. In the global view, a group-level activity involves several person-level actions and binds several person-level actions to a specific activity. For example, in a "walking" activity, most persons, who are walking together, decide the class "walking" of this activity. In turn, the "walking" activity involves most of the "walking" persons. We can see that the group-level activity and person-level actions in an activity scene constitute a host–parasite relationship, which cannot be simulated by the two-stage solution.

Therefore, considering the host–parasite relationship in the group activity, we propose a novel graph LSTM-in-LSTM (GLIL) to simultaneously model the person-level actions and the group-level activity in the spatiotemporal space, as shown in Fig. 1. GLIL becomes a graph LSTM (G-LSTM) in the global view and also becomes several person LSTMs (P-LSTMs)with the interactions in the local view. Specifically, P-LSTMs target to model the person-level actions with interaction among persons; meanwhile, G-LSTM targets

Manuscript received July 18, 2019; revised December 23, 2019; accepted March 2, 2020. Date of publication April 2, 2020; date of current version February 4, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001001, in part by the National Natural Science Foundation of China under Grant 61732007, Grant 61932020, Grant 61702265, and Grant 61772268, and in part by the National Natural Science Foundation of Jiangsu Province under Grant BK20170856 and Grant BK20190065. *(Corresponding author: Liyan Zhang.)*

Xiangbo Shu, Yunlian Sun, and Jinhui Tang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: shuxb@njust.edu.cn; yunlian.sun@njust.edu.cn; jinhuitang@njust.edu.cn).

Liyan Zhang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: zhangliyan@nuaa.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2978942

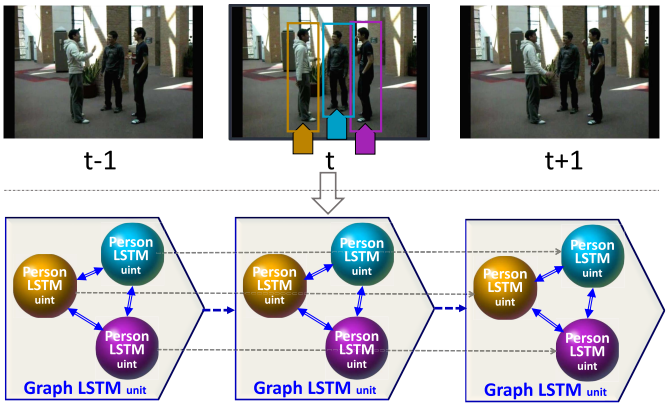


Fig. 1. Idea of the proposed GLIL for modeling a group activity. GLIL becomes P-LSTMs in the local view that models the person-level actions with the interactions among persons and becomes a G-LSTM in the global view that models the group-level activity at the same time. The G-LSTM and P-LSTMs constitute a host–parasite architecture in spatiotemporal space, which simulates the "host–parasite" relationship between the group-level activity ("host") and person-level actions ("parasite").

to model the group-level activity. G-LSTM and P-LSTMs constitute a host–parasite architecture of GLIL in the spatiotemporal space, which reveals the "host–parasite" relationship between the group-level activity and person-level actions.

The training framework of GLIL is shown in Fig. 2, which stacks a pretrained CNN, a residual LSTM (R-LSTM), the GLIL, and a softmax layer in a bottom–up way. First, we employ a pretrained CNN to extract the static features (i.e., CNN features) of each person on the person's bounding boxes. Second, we extend an R-LSTM to learn the person-level residual features of each person from their static features. Third, followed by R-LSTM, P-LSTM in GLIL learns and updates the person-level motion state of one person under the interaction with other persons, while a G-LSTM in GLIL selectively aggregates the person-level motion information from P-LSTM into a new group-level memory cell over time. Finally, we feed the group-level activity representation output from GLIL into the softmax layer at each time step and then average the outputs of all the softmax classifiers to infer the class of group activity. This means to perform an average classification score on all the frames over time.

Overall, the main contributions of this article are summarized as follows.

1) To address the problem of group activity recognition, we propose a novel GLIL framework by simultaneously modeling the person-level actions and group-level activity, where the architecture of GLIL simulates the "host–parasite" relationship between the group-level activity and the person-level actions.
2) We design several P-LSTMs to learn the person-level action representations by considering the interactions among persons under a new interaction gate and design a G-LSTM to learn the group-level activity representations.
3) We conduct experiments on two public data sets (Volleyball data set (VD) [15] and Collective Activity data set (CAD) [6]) to illustrate the effectiveness of the proposed GLIL compared with the state-of-the-art methods.

The rest of this article is organized as follows. Section II reviews some works related to RNN-based action recognition and group activity recognition. Section III introduces some preliminary works. Section IV details the proposed framework. Experiments are conducted in Section V, followed by the conclusions in Section VII.

II. Related Work

In this section, we briefly review some works related to the RNN-based action recognition and group activity recognition.

A. RNN-Based Action Recognition

Action recognition aims to recognize human action in videos [5], [19]–[21]. In the early stages, various spatiotemporal feature learning and feature extraction methods, e.g., histogram of oriented gradients (HOG) [22], histogram of optical flow (HOF) [23], dense trajectories [20], and 3-D SIFT [24], were proposed to represent the human action in videos.

For the last few years, RNNs [13] and LSTM [14] have made great progress in action recognition, due to the powerful ability for handling sequential data with variable length [15], [19], [25]–[28]. For example, Donahue *et al.* [19] proposed a long-term recurrent convolutional network for action recognition by stacking the CNN layer and RNN/LSTM layer in a bottom–up way. Subsequently, some works utilized RNN/LSTM to model the spatial relationship among data when modeling human action. For example, Wang and Wang *et al.* [29] proposed a two-stream architecture, including a temporal RNN and a spatial RNN to model temporal motions of individuals over time and spatial relation among skeleton joints.

In the meantime, kinds of RNN architectures were built to model human action based on various ideas [25], [30]–[32]. For example, to capture the change degree of motion information between two consecutive frames, Veeriah *et al.* [27] designed a derivative of the motion state between the gates in LSTM. Moreover, Shahroudy *et al.* [30] and Liu *et al.* [28] proposed to divide the memory cell in LSTM into multiple subcells corresponding to different human skeleton parts, which models the motion of skeleton parts over time.

B. Group Activity Recognition

Group activity recognition aims to automatically understand an activity performed by at least two persons, which has been developed into an attractive topic [8], [9], [33]–[35]. In the early stages, researchers designed various handcrafted features to represent person-level actions or group-level activities [6], [8], [36], [37]. Since deep learning has achieved great progress in image/video classification tasks [38]–[40], many deep learning-based activity recognition methods have been proposed in recent years [15], [17], [33], [41]. As one of the most representative works, a hierarchical LSTM model in [15] is proposed to first utilize several LSTMs to learn the person-level action representations over time and then integrate the person-level action representations of all persons

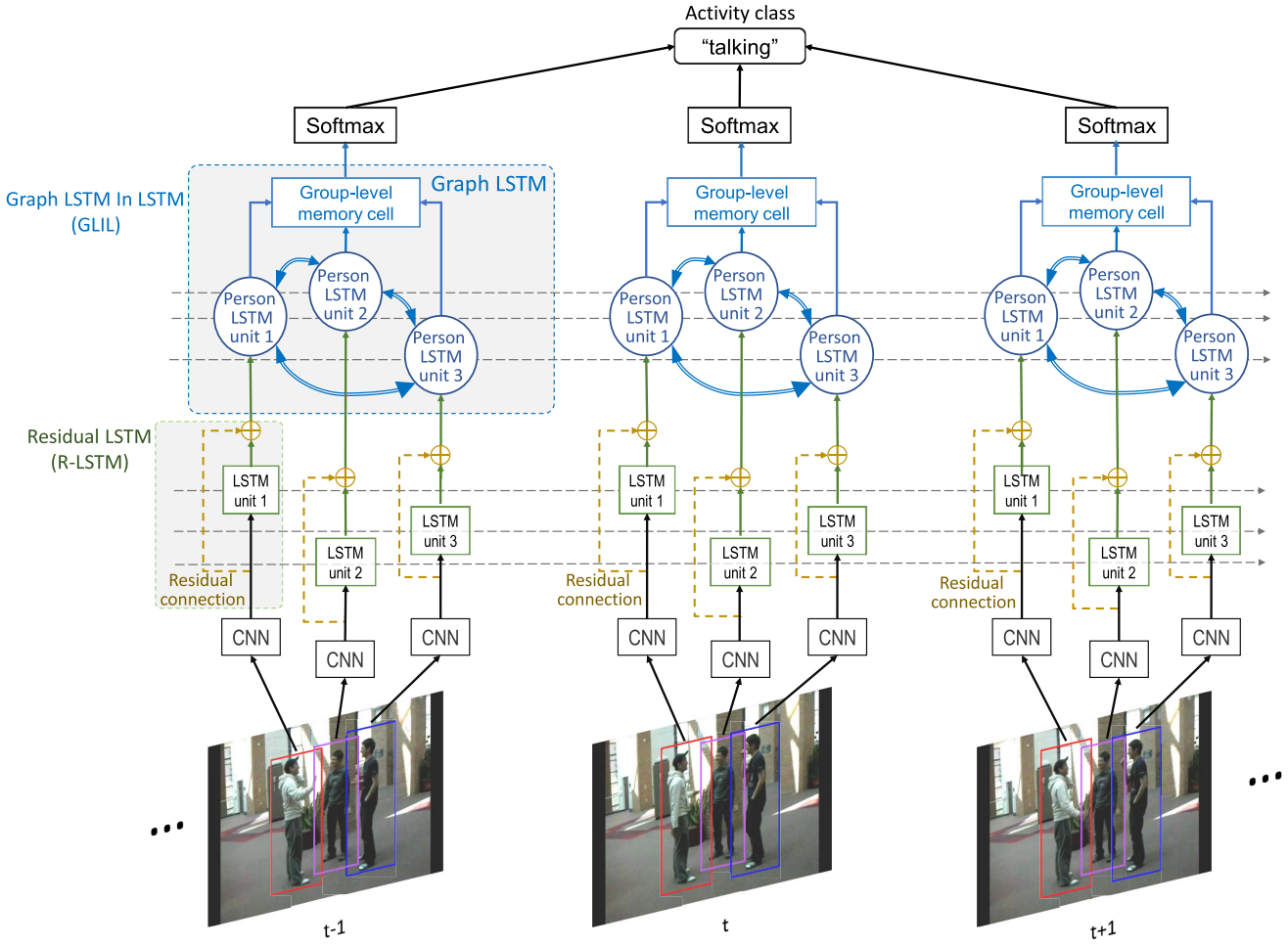


Fig. 2. Framework of the proposed GLIL for group activity recognition. For each frame, we first feed the CNN features of each person to an R-LSTM unit. Then, R-LSTM learns the person-level residual features of each person, which are treated as the inputs of the following GLIL. For a GLTL unit, one P-LSTM unit learns and updates the person-level motion state with the interactions among persons, while one G-LSTM unit selectively aggregates and stores the person-level motion information from P-LSTMs into a new group-level memory cell. Finally, the group-level activity representation output from the group-level memory cell at each time step is input to the softmax layer, and the averaged softmax score at all time steps is the prediction probability vector of group activity class.

into the following LSTM to learn the group-level activity representation over time.

Subsequently, some deep learning-based activity recognition methods assumed that persons are not independent in the group activity and considered modeling interactions among persons in a group activity [16], [18], [31], [42]. For example, Wang *et al.* [18] extended an RNN-based hierarchical framework to learn three-level motions in a step-by-step way, i.e., person-level actions, person–person interactions, and group-level activity corresponding to the individuals, multiple persons, and the activity scene. Considering the different contributions of individuals in a group activity, Tang *et al.* [31] proposed a coherence-constrained G-LSTM with spatiotemporal context coherence constraint and a global context coherence to effectively recognize group activity by modeling the relevant motions of individuals while suppressing the irrelevant motions. However, there methods model the person-level action and group-level activity in a step-by-step way, which ignores the fact that person-level action and group-level activity happen at the same time.

Therefore, some works considered modeling the person-level action and group-level activity simultaneously [33], [43], [44]. For example, Deng *et al.* [43] regarded all persons and the whole activity scene as the nodes of a graph and further exploited multiple RNNs to model the person-level action of persons and the group-level activity of the scene in a graph model. This method regards the group-level activity as a graph node, which is equal to the person-level action. In fact, in an activity scene, the person-level action and the group-level activity are not equivalent, while the truth is that the person-level actions participate in the group-level activity. A reasonable assumption is that the group-level activity and the person-level actions constitute a host–parasite relationship, as discussed in Section I. Therefore, we consider designing a new architecture of LSTM to simulate such host–parasite relationship for modeling group activity well.

Recently, Wu *et al.* [45] and Azar *et al.* [46] also explored the activity recognition and achieved the state-of-the-art performance. The difference between the proposed GLIL and these two recent methods is detailed in the following.

1) For the idea, both of Wu *et al.* [45] and Azar *et al.* [46] considered aggregating all predictions made at the level of single persons into the prediction made at the level of the whole group. GLIL directly outputs the prediction made at the level of the whole group by aggregating the single-person memories instead of the single-person predictions.
2) For the model formulation, the previous models proposed by Azar *et al.* [46] and Wu *et al.* [45] are based on the convolution network. These models only consider the spatial relationships among persons in the spatial space. Different from them, GLIL is originally based on both of the LSTM and graph structure, where the graph structure considers the spatiotemporal relation among persons in both of the temporal and spatial spaces.
3) For the architecture design, convolutional relational machine proposed by Azar *et al.* [46] can be seen as a new convolutional neural network, and Wu *et al.* [45] constructed actor relation graphs that implement on the existing graph convolutional network. Different from them, GLIL designs a bioinspired host–parasite G-LSTM, which can be seen as a new LSTM architecture.

Since generative adversarial network (GAN) has become beneficial in generating the data/feature and learning the loss function at the same time, Gammulle *et al.* [47] first utilized GAN to learn the "action code" for the group activity, which is the same as to the ground-truth label. Even so, Azar *et al.* [46] employed LSTM as the core module to learn the temporal action representation over time. Compared with LSTM, GLIL is a relatively flexible and superior architecture, which can also be embedded into the GAN framework.

III. Preliminary

This section introduces some preliminary works, such as LSTM and graph construction, which can provide basic knowledge and background.

A. Long Short-Term Memory

Given a video clip $\{\mathbf{x}^t | t = 1, \ldots, T\}$ with T frames, where $\mathbf{x}_t$ is the static feature (such as CNN feature [48]) of the tth frame, we use the standard LSTM [14] to learn a sequence of hidden states $\{\mathbf{h}^t | t = 1, \ldots, T\}$ to describe the dynamic of this video clip. The standard LSTM mainly consists of an input gate, forget gate, output gate, input modulation gate, and memory cell state, and one common LSTM unit at time step t can be repressed as follows:

$$\mathbf{i}^t = \sigma\left(\mathbf{W}_{ix} \cdot \mathbf{x}^t + \mathbf{W}_{ih} \cdot \mathbf{h}^{t-1} + \mathbf{b}_i\right); \quad (1)$$

$$\mathbf{f}^t = \sigma\left(\mathbf{W}_{fx} \cdot \mathbf{x}^t + \mathbf{W}_{fh} \cdot \mathbf{h}^{t-1} + \mathbf{b}_f\right); \quad (2)$$

$$\mathbf{o}^t = \sigma\left(\mathbf{W}_{ox} \cdot \mathbf{x}^t + \mathbf{W}_{oh} \cdot \mathbf{h}^{t-1} + \mathbf{b}_o\right); \quad (3)$$

$$\mathbf{g}^t = \varphi\left(\mathbf{W}_{gx} \cdot \mathbf{x}^t + \mathbf{W}_{gh} \cdot \mathbf{h}^{t-1} + \mathbf{b}_g\right); \quad (4)$$

$$\mathbf{c}^t = \mathbf{f}_t^s \odot \mathbf{c}^{t-1} + \mathbf{i}^t \odot \mathbf{g}^t; \quad (5)$$

$$\mathbf{h}^t = \mathbf{o}^t \odot \varphi\left(\mathbf{c}^t\right), \quad (6)$$

where $\mathbf{i}^t$, $\mathbf{f}^t$, $\mathbf{o}^t$, $\mathbf{g}^t$, and $\mathbf{c}^t$ are the input gate, forget gate, output gate, input modulation gate, and memory cell state, respectively; $\sigma(\cdot)$ is a sigmoid function; $\odot$ denotes element-wise product, $\varphi(\cdot)$ is a hyperbolic tangent tanh(·); $\mathbf{W}_{*x}$ and $\mathbf{W}_{*h}$ are weight matrices; and $\mathbf{b}_*$ is bias vector. Specifically, the input gate $\mathbf{i}^t$ controls the contributions of the newly arrived input data at time step t for updating the memory cell, while the forget gate $\mathbf{f}^t$ determines how much the contents of the previous state $\mathbf{c}^{t-1}$ contribute to deriving the current state $\mathbf{c}^t$. The output gate $\mathbf{o}^t$ learns how the output of the LSTM unit at time step t should be derived from the current state of the memory cell $\mathbf{c}_t$. More details can be found in [14].

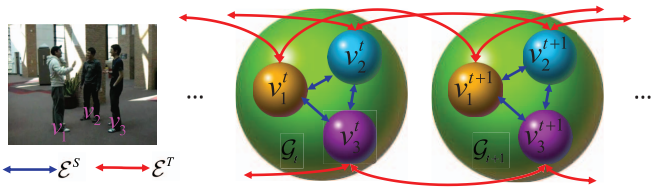


Fig. 3. Toy example of graph construction for one activity within three persons.

B. Graph Construction

This article aims to understand the complex group activity and recognizing different group activities by considering the participating degree. For a group activity, each video frame contains multiple-persons' motion information, which is interrelated in both the spatial space and temporal space. In this article, we consider constructing a graph to explore such relations among persons' motion. Specifically, the nodes of the graph can represent the state of data, and the edges can capture the spatiotemporal interactions among nodes.

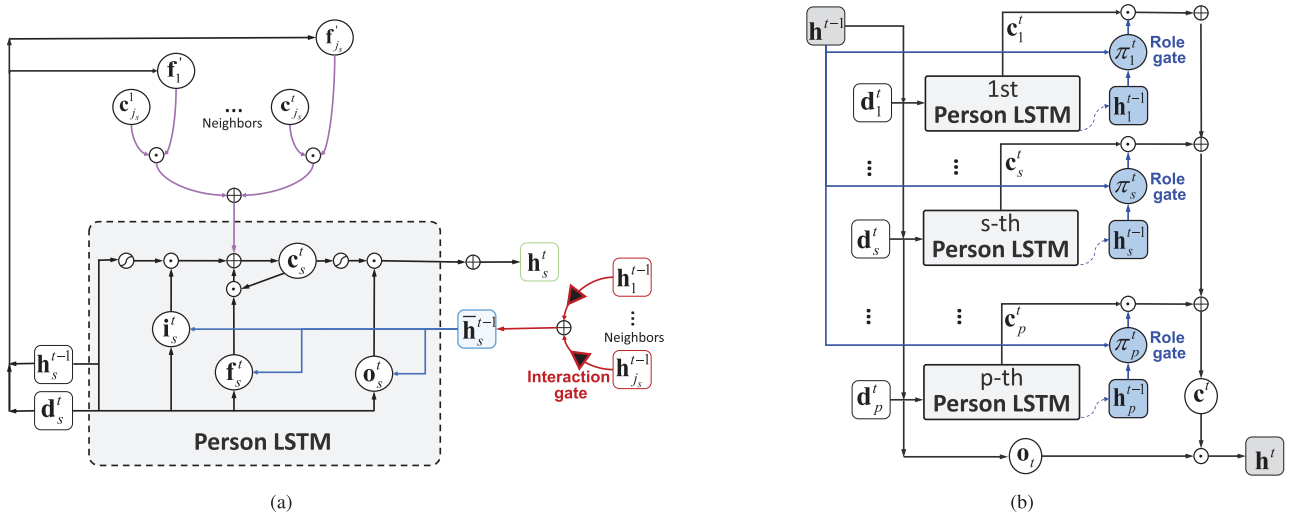
Specifically, given a video clip with T frames describing a group activity within p persons, we construct a relational graph $\mathcal{G}_t = \{\mathcal{V}, \mathcal{E}^S, \mathcal{E}^T\}$ for the tth frame by connecting a set of graph nodes $\{v_s^t | s = 1, 2, \ldots, p$, and $t = 1, 2, \ldots, T\}$ with the graph edges $\mathcal{E}^S$ and $\mathcal{E}^T$ in the spatial space and the temporal space, respectively. Here, v_s^t denotes the feature of the sth person's motion at time step t. For each node v_s^t, there are two temporal edges connecting to the previous node v_s^{t-1} and the subsequent node v_s^{t+1} in temporal space and $p - 1$ edges connecting to its neighboring nodes $\{v_{j_s}^t\}_{j_s \in \Omega(s)}$ in spatial space, where $\Omega(s) = \{1, 2, \ldots, s-1, s+1, \ldots, p\}$. Fig. 3 shows a toy example of graph construction for one activity within three persons.

IV. Proposed Framework

The framework of GLIL for modeling group activity is shown in Fig. 2, which stacks a pretrained CNN, a new R-LSTM, the GLIL (P-LSTMs or G-LSTM), and a softmax layer in a bottom–top way. These components will be introduced in this section.

A. Residual LSTM

For a video clip describing a group activity within p persons, we first employ a pretrained CNN [49] to extract the person-level static features (i.e., CNN features) of each person on person bounding box at each time step, denoted by $\{\mathbf{x}_s^t\}_{t=1}^T$, where $s = 1, 2, \ldots, p$. Subsequently, for the sth person,



(a) (b)

Fig. 4. Host–parasite architecture of the proposed GLIL. In the local view, GLIL becomes P-LSTMs that model the person-level motions by the neighboring interaction under an interaction gate. In the global view, GLIL becomes a G-LSTM that models group-level motion by integrating all person-level memory cells under a role gate. Such role gate checks the importance of one person-level motion for inferring the class of group-level activity at each time step. (a) P-LSTM (parasite architecture) in GLIL. (b) G-LSTM (host architecture) in GLIL.

we learn the person-level temporal features $\{\tilde{\mathbf{x}}_s^t\}_{t=1}^T$ over time by an LSTM with the residual connection, called R-LSTM in this article. Here, witnessing the success of the deep residual network [50], we add a residual connection across the input and the output of LSTM [51], [52]. Such residual connection can provide better flexibility to deal with the gradient vanishing or exploding in the learning process [50], [53]. R-LSTM combines the person-level static features $\{\mathbf{x}_s^t\}_{t=1}^T$ and the person-level temporal features $\{\tilde{\mathbf{x}}_s^t\}_{t=1}^T$ into the person-level residual features $\{\mathbf{d}_s^t\}_{t=1}^T$, namely $\mathbf{d}_s^t \triangleq \mathbf{x}_s^t + \tilde{\mathbf{x}}_s^t$, $s = 1, 2, \ldots, p$. Finally, the obtained person-level residual features $\{\mathbf{d}_s^t\}_{t=1}^T$ can be seen as the node v_s^t in $\mathcal{G}_t$, which is fed into the proposed GLIL.

B. Parasite Architecture of GLIL

Based on the constructed graph $\mathcal{G}_t$, we build the architecture of GLIL, as shown in Fig. 4. Here, GLIL becomes several P-LSTMs in the local view, where the architecture of one P-LSTM is shown in Fig. 4(a). In the global view, GLIL becomes a G-LSTM, as shown in Fig. 4(b), where one P-LSTM acts as a graph node. G-LSTM and P-LSTMs constitute a "host–parasite" architecture. In Fig. 4(b), the sth P-LSTM has an input gate $\mathbf{i}_s^t$, forget gate $\mathbf{f}_s^t$, output gate $\mathbf{o}_s^t$, and the neighboring forget gates $\mathbf{f}_{j_s}^t$ at time step t. These gates are decided by the input feature $\mathbf{d}_s^t$ (output from the R-LSTM) at the current time step, a person-level motion state $\mathbf{h}_s^{t-1}$ at time step $(t-1)$, and a neighboring motion state $\bar{\mathbf{h}}_s^{t-1}$ from its spatial neighbors at time step $(t-1)$, respectively. Formally, the sth P-LSTM at time step t is formulated as follows:

$$\mathbf{i}_s^t = \sigma\left(\mathbf{W}_s^i \mathbf{d}_s^t + \mathbf{U}_s^i \mathbf{h}_s^{t-1} + \mathbf{V}_s^i \bar{\mathbf{h}}_s^{t-1} + \mathbf{b}_s^i\right) \tag{7}$$

$$\mathbf{f}_s^t = \sigma\left(\mathbf{W}_s^f \mathbf{d}_s^t + \mathbf{U}_s^f \mathbf{h}_s^{t-1} + \mathbf{V}_s^f \bar{\mathbf{h}}_s^{t-1} + \mathbf{b}_s^f\right) \tag{8}$$

$$\mathbf{f}_{j_s}^t = \sigma\left(\mathbf{W}_{j_s}^f \mathbf{d}_s^t + \mathbf{U}_{j_s}^f \mathbf{h}_s^{t-1} + \mathbf{V}_{j_s}^f \bar{\mathbf{h}}_s^{t-1} + \mathbf{b}_{j_s}^f\right), \quad j_s \in \Omega(s) \tag{9}$$

$$\mathbf{o}_s^t = \sigma\left(\mathbf{W}_s^o \mathbf{d}_s^t + \mathbf{U}_s^o \mathbf{h}_s^{t-1} + V_s^o \bar{\mathbf{h}}_s^{t-1} + \mathbf{b}_s^o\right) \tag{10}$$

where $\mathbf{W}_*^*$, $\mathbf{U}_*^*$, and $\mathbf{V}_*^*$ are the weight matrices, and $\mathbf{b}_*^*$ is the bias vector.

1) Person-Level Action Representation: The output of P-LSTM at time step t is a person-level motion state $\mathbf{h}_s^t$ of the sth person (i.e., the person-level action representation $\mathbf{h}_s^t$ of the sth person at time step t), which can be computed as follows:

$$\mathbf{g}_s^t = \varphi\left(\mathbf{W}_s^g \mathbf{d}_s^t + \mathbf{U}_s^g \mathbf{h}_s^{t-1} + \mathbf{V}_s^g \bar{\mathbf{h}}_s^{t-1} + \mathbf{b}_s^g\right) \tag{11}$$

$$\mathbf{c}_s^t = \mathbf{i}_s^t \odot \mathbf{g}_s^t + \mathbf{f}_s^t \odot \mathbf{c}_s^{t-1} + \sum_{j_s \in \Omega(s)} \mathbf{f}_{j_s} \odot \mathbf{c}_{j_s}^t \tag{12}$$

$$\mathbf{h}_s^t = \mathbf{o}_s^t \odot \varphi\left(\mathbf{c}_s^t\right) \tag{13}$$

where $\mathbf{c}_s^t$ is the person-level memory cell at time step t. Equation (7)–(13) represent the basic model of P-LSTM. In the following, we will introduce some new components in P-LSTM compared with the conventional LSTM.

2) Neighboring Motion State: It is noted that the neighboring motion state $\bar{\mathbf{h}}_s^{t-1}$ in (7)–(13) is averaged by the persons-level motion states of all neighboring persons of the sth person as follows:

$$\bar{\mathbf{h}}_s^{t-1} = \sum_{j_s \in \Omega(s)} r_{j_s}^t \mathbf{h}_{j_s}^{t-1}. \tag{14}$$

In (14), $r_{j_s}^t$ denotes an interaction gate of the j_sth neighboring person corresponding to the sth person.

3) Interaction Gate: As mentioned earlier, two persons usually interact over time in a group activity. Specifically, for the sth person's motion at time step t, different neighboring persons interact with different degrees over time. In most cases, if two persons are standing closely, and performing similar motions, they are intensively interacting. Therefore, we can simultaneously use the feature similarity and location similarity of two persons to measure their interaction. Specifically, we use the central position of person's bounding box to denote the person's location. Let $\mathbf{a}_s^t$ and $\mathbf{a}_{j_s}^t$ denote the locations of the sth person and her/his j_sth neighboring person, respectively, and we design an interaction gate $r_{j_s}^t$ to quantify the interaction between two persons at time step t as

follows:

$$r_{j_s}^t = \lambda \cdot \mathbf{SimFeature}^t(s, j_s) + (1-\lambda) \cdot \mathbf{SimLocation}^t(s, j_s);$$
$$\mathbf{SimFeature}^t(s, j_s) = \frac{\mathcal{X}\left(\left|\left|\mathbf{h}_s^{t-1} - \mathbf{h}_{j_s}^{t-1}\right|\right|^2\right)}{\sum_{j \in \Omega(s)} \mathcal{X}\left(\left|\left|\mathbf{h}_s^{t-1} - \mathbf{h}_{j_s}^{t-1}\right|\right|^2\right)};$$
$$\mathbf{SimLocation}^t(s, j_s) = \frac{\mathcal{X}\left(\left|\left|\mathbf{a}_s^t - \mathbf{a}_{j_s}^t\right|\right|^2\right)}{\sum_{j_s \in \Omega(s)} \mathcal{X}\left(\left|\left|\mathbf{a}_s^t - \mathbf{a}_{j_s}^t\right|\right|^2\right)} \tag{15}$$

where $\mathcal{X}(\cdot) = 1/\exp(\cdot)$, and λ is a coefficient to balance two terms.

C. Host Architecture of GLIL

In Fig. 4(b), G-LSTM resembles a host that fosters several P-LSTMs. In G-LSTM, the person-level motion information in all person-level memory cells $\mathbf{c}_s^t$ of P-LSTM is integrated into a new graph-level memory cell $\mathbf{c}^t$ of G-LSTM. The group-level memory cell combines all person-level motions into the group-level motion. Although all person-level motion information contributes to the inference of the group activity class, their contributions are different. Thus, we hope the group-level memory cell can selectively integrate and store the useful person-level motion information when cooking group-level motion. Similar to a previous work [42], we can set a gate to control what types of person-level motion information would enter or leave the group-level memory cell over time. Here, we design a new role gate π_s^t at time step t to allow the person-level motion of the sth person to enter or level group-level memory cell.

1) Role Gate: To design the role gate π_s^t, we need to answer a question: what type of person-level motion is useful to infer the class of group activity? Obviously, if we only use one person-level action representation at the previous time step to accurately infer the class of group activity, the person-level motion at this time step is useful and to be allowed into the group-level memory cell. Therefore, we measure the consistency of the label inference of group-level activity representation and person-level action representation at the previous time step. Formally, the role gate π_s^t at time step t is defined as follows:

$$\pi_s^t = \sigma\left(\mathbf{W}_s^\pi \mathbf{q}^{t-1} + \mathbf{U}_s^\pi \mathbf{q}_s^{t-1} + \mathbf{b}_s^\pi\right), \quad s \in \{1, 2, \ldots, p\} \tag{16}$$

where $\mathbf{q}^{t-1}$ and $\mathbf{q}_s^{t-1}$ denote two predicted label vectors of the activity class, which are obtained via group-level activity representation $\mathbf{h}^{t-1}$ [defined in (22)] and person-level action representation $\mathbf{h}_s^{t-1}$, respectively. Specifically, if we let L denote the class number of group activity, $\mathbf{q}^{t-1} = [q_1^{t-1}, \ldots, q_l^{t-1}, \ldots, q_L^{t-1}]^T$ and $\mathbf{q}_s^{t-1} = [q_{s_1}^{t-1}, \ldots, q_{s_l}^{t-1}, \ldots, q_{s_L}^{t-1}]^T$ are obtained as follows:

$$q_l^{t-1} = \frac{\exp\left(z_l^{t-1}\right)}{\sum_{i=1}^L \exp\left(z_i^{t-1}\right)} \tag{17}$$

$$q_{s_l}^{t-1} = \frac{\exp\left(z_{s_l}^{t-1}\right)}{\sum_{i=1}^L \exp\left(z_{s_i}^{t-1}\right)} \tag{18}$$

where $\mathbf{z}^{t-1} = [z_1^{t-1}, \ldots, z_l^{t-1}, \ldots, z_L^{t-1}]^T$ and $\mathbf{z}_s^{t-1} = [z_{s_1}^{t-1}, \cdots, z_{s_l}^{t-1}, \ldots, z_{s_L}^{t-1}]^T$ are the confidence score vectors of group-level action representation $\mathbf{h}^{t-1}$ and person-level action representation $\mathbf{h}_s^{t-1}$ at time step $(t-1)$, respectively, that is

$$\mathbf{z}^{t-1} = \varphi\left(\mathbf{W}^z \mathbf{h}^{t-1} + \mathbf{b}^z\right) \tag{19}$$

$$\mathbf{z}_s^{t-1} = \varphi\left(\mathbf{W}_s^z \mathbf{h}_s^{t-1} + \mathbf{b}_s^z\right), \quad s \in \{1, 2, \ldots, p\}. \tag{20}$$

2) Group-Level Memory Cell: When we obtain the role gate π_s^t, we selectively integrate all person-level memory cells $\{\mathbf{c}_s^t\}_{s=1}^p$ into the group-level memory cell $\mathbf{c}^t$ via the corresponding role gates π_s^t, as follows:

$$\mathbf{c}^t = \sum_{s=1}^p \pi_s^t \odot \mathbf{c}_s^t. \tag{21}$$

3) Group-Level Activity Representation: In G-LSTM, its output at time step t is the group-level activity state $\mathbf{h}^t$ (i.e., group-level activity representation $\mathbf{h}^t$ at time step t), that is

$$\mathbf{h}^t = \mathbf{o}^t \odot \varphi\left(\mathbf{c}^t\right) \tag{22}$$

where $\mathbf{o}^t$ is the output gate of G-LSTM, which is defined as follows:

$$\mathbf{o}^t = \sigma\left(\sum_{s=1}^p \mathbf{G}_s^o \mathbf{h}_s^t + \mathbf{G}^o \mathbf{h}^{t-1} + \mathbf{b}^o\right) \tag{23}$$

where $\mathbf{G}_*^*$ is the weight matrix.

Finally, we compute the confidence score vector $\mathbf{z}^t$ of the group activity class by (19), and feed $\mathbf{z}^t$ $(t = 1, 2, \ldots, T)$ into a softmax layer, that is

$$\mathbf{y}^t = \text{softmax}\left(\mathbf{z}^t\right), \quad t = 1, 2, \ldots, T. \tag{24}$$

The outputs of all the softmax classifiers corresponding to all frames are averaged to obtain the probability class vector of group activity.

V. Experiments

In the experiments, we evaluate the performance of the proposed GLIL compared with the state-of-the-art methods on two public data sets, i.e., CAD [6], and VD [15].

A. Data Set

Two data sets used in the experiments are described as follows.

1) *CAD [6]:* It contains 44 video clips with five types of activities, i.e., "crossing," "waiting," "queuing," "walking," and "talking." Similar to [36] and [58], we select two-thirds of the video clips from each activity class to form the training set, and the rest are used for testing. The person bounding boxes (tracklets) used in the experiments are provided in [9]. Since the number of persons is varying in a range [1, 12], we randomly select five effective persons for each frame and regard them as a group activity.[1] Here, if the number of persons in one frame is less than five or some humans

[1]When the proposed method handles the activity recognition task in crowded environments, we set a fixed number of persons that can fittingly represent the activity and choose persons who emerge at most of the time steps.

dynamically exiting the scene/group at one time step, we take a full-zero matrix as a new person bounding box. Following the experimental setting in [17], [18], and [46], we merge class "walking" and "crossing" as "moving" due to the imbalanced test set.

2) *VD [15]:* It consists of 55 videos within 4830 annotated frames. This data set provides the person bounding box of each person in each frame, and a group activity class for each video clip, e.g., "left pass," "right pass," "left set," "right set," "left spike," "right spike," "left win," or "right win." Following the setting in [15], two-thirds of the annotated frames are selected for training, while the rest ones for testing. In this data set, each group activity contains two subgroups corresponding to two teams. Similar to [15] and [42], all person-level residual features in one subgroup are first input to a GLIL for learning the representations of subgroup-level activity. Then, we concatenate the activity representations of two subgroups as the representation of the whole group-level activity at each time step.

B. Experimental Setting

We use Torch toolbox and Tensorflow [59] as the deep learning platform to conduct the experiments. Following in [60], we employ the VGG16 pretrained on ImageNet to extract the person-level CNN features (on the FC-15 layer of VGG16) based on the person bounding box at each time step. In the VD and CAD data sets, we consider ten frames from each video without any resampling; namely, the length T of time steps for a video clip is set to 10. We also use the other recent networks to extract the person-level CNN features for fair comparison with prior methods. In the configuration of R-LSTM, the number of input nodes and the number of output nodes are set to 4096 for the residual connection. The number of output nodes in P-LSTM and the number of nodes in G-LSTM are set to 1024. We use the Adam algorithm [61] as the optimizer. The learning rate, momentum, and decay rate are set to 0.5×10^{-3}, 0.9, and 0.95, respectively. For the network initialization, such as traditional deep neural networks, we use the random normal distribution (mean $= 0$ and variance $= 0.01$) to initialize the parameters of GLIL. We select the parameter λ in (15) from the values of $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Through the experimental validation, we ultimately set lambda $= 0.7$ as the optimal value. We use the mean per-class accuracy (MPCA) as the performance metric.

C. Time Complexity

Let d denote the feature dimension of the person-level CNN feature. In R-LSTM, the number of the output nodes is also set d for the skip connection. For the forward propagation of R-LSTM, the time complexity mainly comes from the matrix computation in input gate, forget gate, output gate, and input modulation gate, namely $\mathcal{O}(\text{ResidualLSTM}) = O(4pd^2Tn)$, where p, T, and N are the number of persons, the number of time steps, and the number of video clips, respectively. Since we use backpropagation through time (BPTT) to minimize the loss function, the time complexity of backpropagation is equal to that of forward propagation. Thus, the time complexity of R-LSTM is $\mathcal{O}(\text{ResidualLSTM}) = 2\mathcal{O}(4pd^2Tn)$ in total. Likewise, the time complexity of P-LSTM in input gate, forget gate, output gate, input modulation gate, and neighboring forget gates is $\mathcal{O}(\text{PersonLSTM}) = 2(\mathcal{O}(4pd^2Tn) + (p-1)\mathcal{O}(pd^2Tn))$. Let m denote the number of output nodes of P-LSTM and the number of output nodes of G-LSTM, and the time complexity of G-LSTM in role gate and output gate is $\mathcal{O}(\text{GraphLSTM}) = 2(p\mathcal{O}(2LmTn) + (p+1)\mathcal{O}(LmTn) + \mathcal{O}((p+1)m^2Tn))$, where L denotes the number of the activity classes. Thus, the time complexity of the proposed GLIL is $\mathcal{O}(\text{GLIL}) = 2E(2\mathcal{O}(4pd^2Tn) + (p-1)\mathcal{O}(pd^2Tn) + p\mathcal{O}(2LmTn)) + 2E((p+1)\mathcal{O}(LmTn) + \mathcal{O}((p+1)m^2Tn))$, where E denotes the number of epochs. In the experiments, the training of GLIL begins to converge after about 60 and 110 epochs on the CAD and VD data sets, respectively. Then, the time consumption for training GLIL on CAD and VD requires about 10 and 55 h, respectively.

D. Baselines

In the experiments, several baselines are defined to illustrate the novelty of the proposed GLIL.

- **B1** *One LSTM:* This baseline treats all person-level actions as a whole to directly learn the group-level activity via an LSTM. First, multiple-person bounding boxes at one time step are merged into a bigger bounding box. Second, the CNN feature is extracted on this "bigger" bounding box at each time step. Third, we use the CNN features as inputs to train an LSTM.
- **B2** *Multiple LSTMs:* This baseline learns the person-level actions by multiple LSTMs. First, the CNN features of each person are fed into an LSTM to learn the person-level action representation. Third, all person-level action representations at one time step are concatenated into the group-level activity representation.
- **B3** *Hierarchical LSTM:* This baseline is a two-stage solution in a hierarchical way. It first learns the person-level action representations of all persons by multiple LSTMs and then learns the group-level activity representations by the following LSTM in a hierarchical way. The idea of this baseline is the same as Ibrahim *et al.* [15].
- **B4** *Hierarchical R-LSTM:* This baseline is a two-stage solution in a hierarchical way. The CNN features of each person are input to the R-LSTM, followed by a conventional LSTM. This baseline aims to test the contribution of R-LSTM compared with B3.
- **B5** *GLIL Without R-LSTM:* This baseline throws out the R-LSTM in the proposed GLIL. Thus, the CNN features of each person are directly fed into GLIL, followed by the softmax layer at each time step. This baseline aims to illustrate the power of the GLIL network.

E. Results on CAD

1) Ablation Studies: We first illustrate the novelty of the proposed GLIL by comparing it with several baselines.

TABLE I
COMPARISON AMONG DIFFERENT METHODS ON THE CAD DATA SET

Methods	Backbone	moving	waiting	queuing	talking	MPCA
Choi et al. [6]	None	90	82.9	95.4	94.9	90.8
Lan et al. [35]	None	92	69	76	99	84
Zhou et al. [53]	None	88.5	74.0	95.0	98.0	88.9
Hajimirsadeghi et al. [54]	None	87	75	92	99	88.3
Wang et al. [18]	AlexNet	94.9	63.6	100	99.5	89.4
Ibrahim et al. [15]	AlexNet	**95.9**	66.4	96.8	99.5	89.7
Li et al. [55]	Inception-v3	90.8	81.4	99.2	84.6	89.0
Yan et al. [17]	AlexNet	92.8	76.6	100	99.5	92.2
Tang et al. [56]	VGG	95.7	**89.9**	100	97.3	92.5
Gammulle et al. [46]	ResNet50	94.5	80.5	100.0	98.0	93.3
Wu et al. [44]	Inception-v3	-	-	-	-	93.7
Azar et al. [45]	Inception-V3	91.7	86.3	100.0	98.91	94.2
B1: One LSTM	VGG16	94.9	50.4	96.4	98.0	84.9
B2: Multiple LSTMs	VGG16	95.5	49.2	97.6	98.0	85.1
B3: Hierarchical LSTM	VGG16	95.5	65.3	98.0	99.5	89.6
B4: Hierarchical Residual LSTM	VGG16	95.5	68.9	99.4	99.0	90.7
B5: GLIL w/o Residual LSTM	VGG16	95.5	73.6	99.0	99.5	91.9
Graph LSTM in LSTM (GLIL)	VGG16	95.5	75.6	100.0	99.0	92.5
	ResNet50	95.5	79.7	100.0	99.5	93.7
	Inception-V3	95.5	84.7	**100.0**	**99.5**	**94.9**

As shown in Table I, GLIL achieves the best performance compared with all baselines. Since B3–B5 model group-level activity in a hierarchical way, they perform better than both B1 and B2. B4 outperforms B3, which illustrates that training the stacked LSTMs in a hierarchical way benefits from the residual connection. Compared with B5, B4 with the R-LSTM and hierarchical architecture can be learned by setting a greater number of epochs. This ensures that B4 gains more discriminative power for some activity, which has few outlier persons (noise), such as the queuing activity. However, for some activities within a certain number of outlier persons (noise), such as the waiting activity, B5 with the "host–parasite" architecture is very useful to remove these outlier persons. This is because the host–parasite architecture of GLIL (in B5) revealing the consistence of the group-level activity and person-level actions can filter the outlier persons. When we further use the person-level residual features instead of the person-level static feature, GLIL improves 0.68% again compared with B5. The confusion matrix of GLIL on CAD is shown in Fig. 5. We can see that "waiting" and "talking" activities are more confusing since they are visually similar to each other.

2) Comparison With the State of the Art: We also compare the recognition accuracy of the proposed GLIL with several competitive methods, including nondeep learning-based methods (Choi *et al.* [6], Wang *et al.* [18], Lan *et al.* [36], Zhou *et al.* [54], and Antic and Ommer [63], Kong *et al.* [64] and deep learning methods (Ibrahim *et al.* [15], Donahue *et al.* [19], Deng *et al.* [33], Shu *et al.* [42], Deng *et al.* [43], Wu *et al.* [45], Azar *et al.* [46], Gammulle *et al.* [47], Hajimirsadeghi and Mori [55], Li and Chuah [56], and Qi *et al.* [60]). The recognition accuracies obtained by these methods are shown in Table I. GLIL achieves the best average accuracy compared with alternatives. Specifically, GLIL improves approximately 7% improvements compared with one of the most representative hierarchical LSTM methods (the two-stage solution) [15] and approximately 23% improvements compared with one classical method (releases this data set) [6]. Not only that, GLIL also performs better than some semantic-based methods (e.g., Li and Chuah [56], and Qi *et al.* [60]) that use the person-level action label information (as the external prior information) to learn the network. Here, we do not use the person-level label information to learn the GLIL model. If we set the same backbone, the proposed GLIL is comparable to the state-of-the-art methods (e.g., Wu *et al.* [45], Azar *et al.* [46], and Gammulle *et al.* [47]). Some recognition results obtained by GLIL are shown in Fig. 6(a).

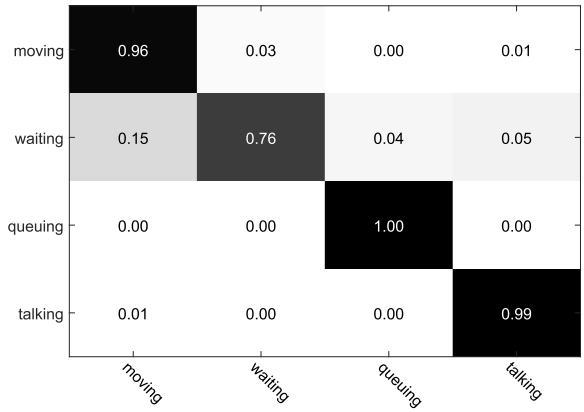


Fig. 5. Confusion matrix of the proposed GLIL (VGG16 as the backbone) on CAD.

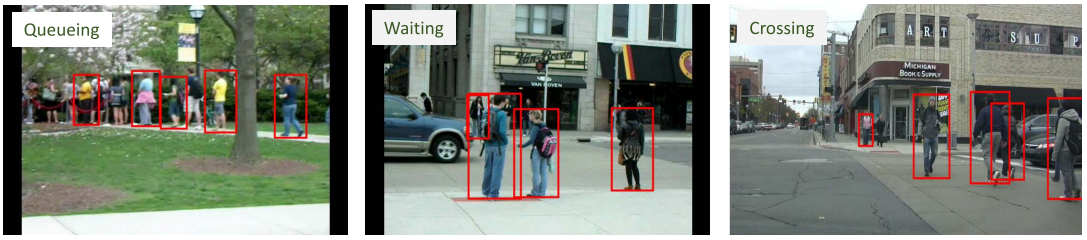
F. Results on VD

1) Ablation Studies: The recognition accuracy of the proposed GLIL compared with the baselines is shown in Table II. GLIL achieves the best average accuracy performance on all activity classes. Compared with B1 and B2, the two-stage solution (i.e., B3 and B4) has achieved a significant improvement

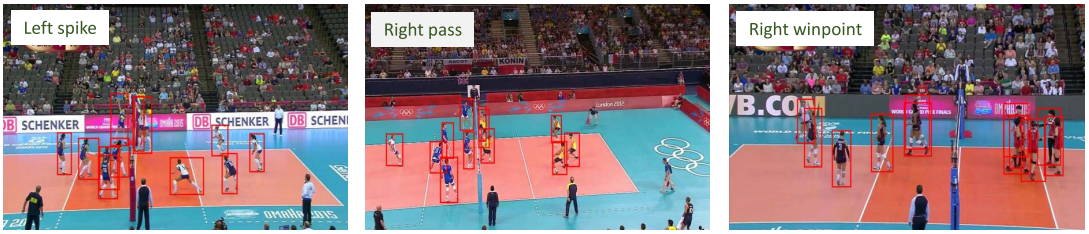
TABLE II

COMPARISON AMONG DIFFERENT METHODS ON THE VD [15]. "LPASS," "RPASS," "LSET," "RSET," "LSPIKE," "RSPIKE," "LWINPOINT," AND "RWINPOINT" DENOTE "LEFT PASS," "RIGHT PASS," "LEFT SET," "RIGHT SET," "LEFT SPIKE," "RIGHT SPIKE," "LEFT WINPOINT," AND "RIGHT WINPOINT," RESPECTIVELY

Methods	backbone	lpass	rpass	lset	rset	lspike	rspike	lwinpoint	rwinpoint	MPCA
Ibrahim et al. [15]	AlexNet	77.9	81.4	84.5	68.8	89.4	85.6	88.2	87.4	82.9
Shu et al. [16]	VGG16	-	-	-	-	-	-	-	-	83.6
Li et al. [55]	Inception-v3	55.8	69.1	67.3	52.1	82.1	79.2	-	-	67.6
Yan et al. [17]	AlexNet	85.8	88.1	90.5	80.2	92.2	87.9	89.2	90.8	88.1
Shu et al. [41]	AlexNet	83.9	88.1	90.3	80.4	93.4	89.8	88.7	92.4	88.4
Qi et al. [59]	VGG16	79	83	87	70	90	87	89	90	84.4
Tang et al. [31]	AlexNet	88.1	90.0	89.9	78.1	93.9	91.3	90.2	93.1	89.3
Ibrahim et al. [61]	VGG19	-	-	-	-	-	-	-	-	89.5
Tang et al. [56]	VGG16	-	-	-	-	-	-	-	-	89.5
Bagautdinov et al. [33]	Inception-v3	-	-	-	-	-	-	-	-	89.9
Gammulle et al. [46]	ResNet50	-	-	-	-	-	-	-	-	92.4
Wu et al. [44]	Inception-v3	-	-	-	-	-	-	-	-	91.0
Azar et al. [45]	Inception-v3	-	-	-	-	-	-	-	-	93.04
B1: One LSTM	VGG16	64.43	66.18	76.55	62.70	77.25	74.81	70.35	68.75	70.13
B2: Multiple LSTMs	VGG16	64.43	77.69	81.83	69.84	88.43	83.43	78.00	78.13	77.72
B3: Hierarchical LSTM	VGG16	80.38	83.48	87.10	71.94	91.65	87.11	89.95	89.25	85.11
B4: Hierarchical Residual LSTM	VGG16	83.16	85.98	85.64	72.74	91.51	87.11	91.97	89.51	85.95
B5: GLIL w/o Residual LSTM	VGG16	93.76	89.56	90.63	86.47	90.16	89.90	91.22	92.69	90.55
Graph LSTM in LSTM (GLIL)	VGG16	93.76	89.56	90.70	87.67	90.20	89.49	94.38	92.91	91.08
	ResNet50	94.96	89.56	90.7	87.67	94.27	89.49	94.78	**94.69**	92.02
	Inception-v3	**95.85**	**90.45**	**91.76**	**89.46**	**95.69**	**92.27**	**94.98**	93.82	**93.04**



(a)



(b)

Fig. 6. Some recognition results obtained by the proposed GLIL on data sets. On CAD, no more than five persons are randomly selected to represent the whole activity. The bounding box on (a) CAD. (b) VD.

in recognition accuracy. Specifically, the recognition results obtained by B3 and B4 successfully validate the importance of the residual connection for learning the hierarchical LSTMs. In comparison to B3 and B4, the improvement obtained by B5 demonstrates that simultaneous capturing person-level motion and group-level motion are effective for recognizing group activities. Specifically, in the lpass, rpass, lset, and rset activity scenes within some outlier persons, B5 performs much better than B4 due to the "host–parasite" architecture of B5. Furthermore, we push GLIL into a hierarchical framework within an R-LSTM and obtain better recognition accuracy. The confusion matrix of the proposed GLIL on the VD is shown in Fig. 7. We find that the confusion occurs due to visually similar motions, e.g., "right set" and "right pass."

2) Comparison With the State of the Art: We compare the performance of the proposed GLIL with several competitive methods. The recognition accuracies obtained by different methods are shown in Table II. Among these methods, Shu *et al.* [16], Bagautdinov *et al.* [34], Biswas and Gall [44], Tang *et al.* [57], and Ibrahim *et al.* [62] do not provide the recognition accuracy of each class, and Li and Chuah [56] ignores the classes of "left winpoint" and "right winpoint." As expected, GLIL achieves the best performance on average. In particular, GLIL achieves approximately 8% improvement compared with one original work [15] that releases the VD. More importantly, the MPCA obtained by GLIL is comparable to the state-of-the-art methods (e.g., Wu *et al.* [45], Azar *et al.* [46], and Gammulle *et al.* [47]). These results

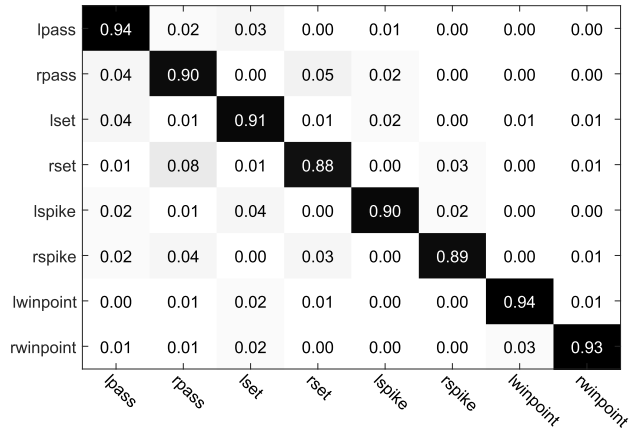


Fig. 7. Confusion matrix of the proposed GLIL (VGG16 as the backbone) on VD.

demonstrate that GLIL with a host–parasite architecture can effectively simulate the relationship between the person-level actions and group activity. Finally, we show some recognition results obtained by GLIL in Fig. 6(b).

VI. Discussion

In this article, we build a novel deep network architecture, called GLIL, by simultaneously modeling the person-level actions and group-level activity. In the architecture of GLIL, several P-LSTMs locate inside the G-LSTM, where the memory cells of P-LSTMs link to a common graph-level memory cell of graph-LSTM. It can be seen as that GLIL constructs a new "host–parasite" architecture. Such new architecture is different from the traditional hierarchical architecture, e.g., Ibrahim *et al.* [15], Yan *et al.* [17], and Wang *et al.* [18]. For traditional graph architecture, there are two nodes that are linked by edge, and several nodes construct a graph. For GLIL, two P-LSTMs are not directly linked by an edge, and all P-LSTMs are linked to a common graph-LSTM, as shown in Fig. 3(b). Thus, GLIL is also different from the Graph RNN/LSTM, Tang *et al.* [31], Deng *et al.* [33], Deng *et al.* [43], and Biswas and Gall [44]. To the best of our knowledge, GLIL is a new architecture for modeling the person-level actions and group-level activity.

VII. Conclusion and Future Work

In this article, to address the problem of group activity recognition, we propose a novel GLIL framework by modeling the person-level actions and the group-level activity simultaneously. In the host–parasite architecture of GLIL, several P-LSTMs in the local view model the person-level actions (parasites) based on the interactions among persons, while a G-LSTM in the global view models the group-level activity (host). Furthermore, to utilize the temporal features instead of the static features as the input of GLIL, we extend an R-LSTM to learn the person-level residual features (including static features and temporal features) of each person, in which a residual connection can avoid the gradient vanishing or exploding in the learning process to some extent. Experimental results on two public data sets demonstrate that the proposed GLIL has improved the recognition accuracy compared with the state-of-the-art methods. In future work, we will embed the proposed GLIL into the GAN framework to train an activity recognizer in a few-shot learning way.

References

[1] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, Jun. 2010.
[2] G. Cheng, Y. Wan, A. N. Saudagar, K. Namuduri, and B. P. Buckles, "Advances in human action recognition: A survey," 2015, *arXiv:1501.05964*. [Online]. Available: http://arxiv.org/abs/1501.05964
[3] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," 2016, *arXiv:1605.04988*. [Online]. Available: http://arxiv.org/abs/1605.04988
[4] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[5] X. Chang, W.-S. Zheng, and J. Zhang, "Learning person–person interaction in collective activity recognition," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1905–1918, Jun. 2015.
[6] W. Choi, K. Shahid, and S. Savarese, "What are they doing?: Collective activity classification using spatio-temporal relationship among people," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Sep./Oct. 2009, pp. 1282–1289.
[7] S. Shariat and V. Pavlovic, "A new adaptive segmental matching measure for human activity recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3583–3590.
[8] W. Choi, K. Shahid, and S. Savarese, "Learning context for collective activity recognition," in *Proc. CVPR*, Jun. 2011, pp. 3273–3280.
[9] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 215–230.
[10] M. R. Amer, D. Xie, M. Zhao, S. Todorovic, and S.-C. Zhu, "Cost-sensitive top-down/bottom-up inference for multiscale activity recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 187–200.
[11] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware modeling and recognition of activities in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2491–2498.
[12] A. Iosifidis, A. Tefas, and I. Pitas, "View-invariant action recognition based on artificial neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 412–424, Mar. 2012.
[13] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.
[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[15] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1971–1980.
[16] T. Shu, S. Todorovic, and S.-C. Zhu, "CERN: Confidence-energy recurrent network for group activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5523–5531.
[17] R. Yan, J. Tang, X. Shu, Z. Li, and Q. Tian, "Participation-contributed temporal dynamic model for group activity recognition," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 1292–1300.
[18] M. Wang, B. Ni, and X. Yang, "Recurrent modeling of interaction context for collective activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3048–3056.
[19] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, Apr. 2017.
[20] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. CVPR*, Jun. 2011, pp. 3169–3176.
[21] X. Shu, J. Tang, G.-J. Qi, Y. Song, Z. Li, and L. Zhang, "Concurrence-aware long short-term sub-memories for person-person action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1–8.
[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.
[23] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1932–1939.
[24] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th Int. Conf. Multimedia (MULTIMEDIA)*, 2007, pp. 357–360.

[25] W. Zhu *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," 2016, *arXiv:1603.07772*. [Online]. Available: http://arxiv.org/abs/1603.07772
[26] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1110–1118.
[27] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4041–4049.
[28] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 816–833.
[29] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 499–508.
[30] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1010–1019.
[31] J. Tang, X. Shu, R. Yan, and L. Zhang, "Coherence constrained graph LSTM for group activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 15, 2019, doi: 10.1109/TPAMI.2019.2928540.
[32] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, "Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction," 2019, *arXiv:1909.13245*. [Online]. Available: http://arxiv.org/abs/1909.13245
[33] Z. Deng *et al.*, "Deep structured models for group activity recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–12.
[34] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese, "Social scene understanding: End-to-end multi-person action localization and collective activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4315–4324.
[35] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2015, pp. 1–7.
[36] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori, "Discriminative latent models for recognizing contextual group activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1549–1562, Aug. 2012.
[37] D. Tao, L. Jin, Y. Yuan, and Y. Xue, "Ensemble manifold rank preserving for acceleration-based human activity recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1392–1404, Jun. 2016.
[38] X. Shu, G.-J. Qi, J. Tang, and J. Wang, "Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 35–44.
[39] J. Tang, X. Shu, Z. Li, G.-J. Qi, and J. Wang, "Generalized deep transfer networks for knowledge propagation in heterogeneous domains," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 4s, pp. 1–22, Nov. 2016.
[40] Y.-G. Jiang, Z. Wu, J. Tang, Z. Li, X. Xue, and S.-F. Chang, "Modeling multimodal clues in a hybrid deep learning framework for video classification," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3137–3147, Nov. 2018.
[41] Y. Zhang, X. Liu, M. Chang, W. Ge, and T. Chen, "Spatio-temporal phrases for activity recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 707–721.
[42] X. Shu, J. Tang, G. Qi, W. Liu, and J. Yang, "Hierarchical long short-term concurrent memory for human interaction recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. 17, 2019, doi: 10.1109/TPAMI.2019.2942030.
[43] Z. Deng, A. Vahdat, H. Hu, and G. Mori, "Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4772–4781.
[44] S. Biswas and J. Gall, "Structural recurrent neural network (SRNN) for group activity analysis," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1625–1632.
[45] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, "Learning actor relation graphs for group activity recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9964–9974.
[46] S. M. Azar, M. G. Atigh, A. Nickabadi, and A. Alahi, "Convolutional relational machine for group activity recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7892–7901.
[47] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Multi-level sequence GAN for group activity recognition," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2018, pp. 331–346.
[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[51] S. A. Hasan *et al.*, "Neural paraphrase generation with stacked residual LSTM networks," in *Proc. Int. Conf. Comput. Linguistics (COLING)*, 2016, pp. 2923–2934.
[52] L. Huang, J. Xu, J. Sun, and Y. Yang, "An improved residual LSTM architecture for acoustic modeling," in *Proc. 2nd Int. Conf. Comput. Commun. Syst. (ICCCS)*, Jul. 2017, pp. 101–105.
[53] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph LSTM," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 125–143.
[54] Z. Zhou, K. Li, X. He, and M. Li, "A generative model for recognizing mixed group activities in still images," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 3654–3660.
[55] H. Hajimirsadeghi and G. Mori, "Multi-instance classification by max-margin training of cardinality-based Markov networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1839–1852, Sep. 2017.
[56] X. Li and M. C. Chuah, "SBGAR: Semantics based group activity recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2876–2885.
[57] Y. Tang, Z. Wang, P. Li, J. Lu, M. Yang, and J. Zhou, "Mining semantics-preserving attention for group activity recognition," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 1283–1291.
[58] H. Hajimirsadeghi, W. Yan, A. Vahdat, and G. Mori, "Visual recognition by counting instances: A multi-instance cardinality potential kernel," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2596–2605.
[59] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, 2014, pp. 675–678.
[60] M. Qi, J. Qin, A. Li, Y. Wang, J. Luo, and L. Van Gool, "stagNet: An attentive semantic RNN for group activity recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 101–117.
[61] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[62] M. S. Ibrahim and G. Mori, "Hierarchical relational networks for group activity recognition and retrieval," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 721–736.
[63] B. Antic and B. Ommer, "Learning latent constituents for recognition of group activities in video," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 33–47.
[64] Y. Kong, Y. Jia, and Y. Fu, "Interactive phrases: Semantic descriptionsfor human interaction recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1775–1788, Sep. 2014.

Xiangbo Shu received his Ph.D. degree from the Nanjing University of Science and Technology, Nanjing, China, in July 2016.

From 2014 to 2015, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His current research interests include computer vision, multimedia computing, and deep learning. He has authored over 35 journal and conference papers in these areas, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the IEEE International Conference on Computer Vision (ICCV), and the ACM International Conference on Multimedia (ACM MM).

Dr. Shu has received the Best Student Paper Award at International Conference on MultiMedia Modeling (MMM) 2016, the Best Paper Runner-up at ACM MM 2015, the Excellent Doctoral Dissertation of Chinese Association for Artificial Intelligence (CAAI), and the Excellent Doctoral Dissertation of Jiangsu Province.

Liyan Zhang received the Ph.D. degree in computer science from the University of California at Irvine, Irvine, CA, USA, in 2014.

She is currently a Professor with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her research interests include multimedia analysis and computer vision.

Dr. Zhang received the Best Paper Award at ICMR 2013 and the Best Student Paper Award at International Conference on MultiMedia Modeling (MMM) 2016.

Yunlian Sun received the B.S. degree from Northeast Normal University, Changchun, China, in 2008, the M.E. degree from the Harbin Institute of Technology, Harbin, China, in 2010, and the Ph.D. degree from University of Bologna, Bologna, Italy, in 2014.

During the Ph.D. study, she worked as a Research Fellow with the University of Sassari, Sassari, Italy. After the Ph.D. study, she worked as a Post-Doctoral Researcher with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China. She is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. Her research interests include biometrics, pattern recognition, and computer vision.

Dr. Sun has served as a Reviewer for various journals, including the TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Pattern Recognition* (PR), *Image and Vision Computing* (IVC), and *Pattern Recognition Letters* (PRL).

Jinhui Tang (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in July 2003 and July 2008 respectively.

From 2008 to 2010, he worked as a Research Fellow with the School of Computing, National University of Singapore, Singapore. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His current research interests include multimedia computing and computer vision. He has authored over 100 journal and conference papers in these areas.

Dr. Tang was a recipient of the ACM China Rising Star Award and a co-recipient of the best paper awards at the ACM International Conference on Multimedia (ACM MM) 2007, PCM 2011, and ICIMCS 2011, the Best Paper Runner-up at ACM MM 2015, and the best student paper awards at International Conference on MultiMedia Modeling (MMM) 2016 and ICIMCS 2017. He has served as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.