24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Group Behavior Recognition Using Attention- and Graph-Based Neural Networks

Fangkai Yang[1†], Wenjie Yin[1†], Tetsunari Inamura[2], Mårten Björkman[1], Christopher Peters[1]

Abstract. When a conversational group is approached by a newcomer who wishes to join it, the group may dynamically react by adjusting their positions and orientations in order to accommodate it. These reactions represent important cues to the newcomer about if and how they should plan their approach. The recognition and analysis of such socially complaint dynamic group behaviors have rarely been studied in depth and remain a challenging problem in social multi-agent systems. In this paper, we present novel group behavior recognition models, attention-based and graph-based, that consider behaviors on both the individual and group levels. The attention-based category consists of Approach Group Net (*AGNet*) and Approach Group Transformer (*AGTransformer*). They share a similar architecture and use attention mechanisms to encode both temporal and spatial information on both the individual and group levels. The graph-based models consist of Approach Group Graph Convolutional Networks (*AG-GCN*), which combine Multi-Spatial-Temporal Graph Convolutional Networks (*MST-GCN*) on the individual level and Graph Convolutional Networks (*GCN*) on the group level, with multi-temporal stages. The individual level learns the spatial and temporal movement patterns of each agent, while the group level captures the relations and interactions of multiple agents. In order to train and evaluate these models, we collected a full-body motion-captured dataset of multiple individuals in conversational groups. Experiments performed using our models to recognize group behaviors from the collected dataset show that AG-GCN, with additional distance and orientation information, achieves the best performance. We also present a multi-agent interaction use case in a virtual environment to show how the models can be practically applied.

1 Introduction

A common pattern of multi-agent interactions arises in small groups where people gather and stand in an environment to converse. This pattern is referred as *free-standing conversational groups* [11] which are ubiquitous in daily life. When humans or robots approach to join these groups, one vital ability is to present social compliance. The newcomer should adopt behaviors in a socially-acceptable manner in order to make individuals in the group feel comfortable [20, 34]. However, group dynamics are not appropriately considered in previous works so that the conversational groups are assumed to be static when approached by a newcomer [2, 21]. As observed in real scenarios such as during coffee breaks and in poster sessions [1, 13] or in human-robot interaction experiments [29], the group members react to the newcomer as they either ignore the newcomer or adjust their positions and orientations to better accommodate them (Figure 1).

Due to the lack of such a capability to recognize dynamic group behaviors, recent works [18, 29] use humans to teleoperate robots to approach groups leveraging the human perception on the dynamic group behaviors. Such teleoperation suffers from limitations that the control needs experienced operators and it is hard to keep consistency among different situations. This motivates us to collect data that can be used to train machine learning models in order to recognize and understand group dynamics. It aims to support research, especially in human-agent interaction, by providing human-group interaction data to better understand and learn human behaviors in groups.

Behavior recognition methods have been widely used in real-world scenarios [12, 37], but with fewer applications for human-human/robot interaction on group level. It is challenging to recognize group behaviors, and the difficulty lies in modeling the relations among group members and the lack of datasets for training. In this paper, we present novel machine learning models that trained on our collected dataset to recognize group behaviors. They are categorized into attention-based and graph-based models. The attention-based models share a similar architecture but differ in the attention mechanism where AGNet uses LSTM-based soft attention and AG-Transformer uses the Transformer model with self-attention. Among these two categories, the Approach Group Graph Convolutional Network (AG-GCN), which combines Multi-Spatial-Temporal Graph Convolutional Neural Networks (MST-GCN) and Graph Convolutional Networks (GCN), achieves the best performance. It builds a spatial-temporal graph from a sequence of body markers. The movement of each agent is modeled through a multi-temporal stage graph on an individual-level, and a group-level graph is combined to encode the group spatial relationships. In order to apply our trained model, we present a virtual online group interaction use case based on a cloud-based VR platform [25]. Each participant controls a virtual character through a VR device. Motion data are fed to the trained model to recognize group behaviors in real-time.

The major contributions of the paper are summarized as follows:

- We propose novel machine learning models for group behavior recognition when a group is approached by a newcomer, supported by a new full-body motion-captured dataset that we collected.
- We present a multi-agent interaction use case in virtual environments to recognize group dynamics in real-time using our models.

2 Related Work

2.1 Multi-agent Interactions in Groups

There have been numerous studies on multi-agent interactions in the field of Artificial Intelligence [23], Social Science and Cogni-

† Authors contributed equally, {fangkai, yinw}@kth.se.
[1] KTH Royal Institute of Technology, Stockholm, Sweden.
[2] National Institute of Informatics, Tokyo, Japan.

tive Science [24], with fewer focused on group interaction, specifically situations in which a newcomer approaches to join a group. In a free-standing conversational group, Kendon [16] proposed the *F-formation* system to define the positions and orientations of individuals within a group. F-formations and other group formation models have been studied computationally [6, 21, 28], and have been used as a basis for group joining behaviors of a mobile robot or an agent [3, 20, 26]. These works focus on navigating a robot or an agent to approach a group in a safe and socially-acceptable manner. However, they rely on hand-crafted features. Other recent works [10, 34, 35] have made use of data-driven methods to generate such joining group behaviors, but they were trained using synthetic data or prior computational models due to the lack of real-life datasets. All aforementioned works assume the newcomer would eventually approach to join the group which is not aware of the newcomer, i.e., the group members have no reaction to the newcomer but stand still in a group. However, as observed in publicly available datasets concerning cocktail parties or poster sessions [1, 13], the free-standing conversational groups would have interactions when approached by a newcomer as they make adjustments in positions and orientations to better accommodate the newcomer or ignore them. Our dataset contains these group behaviors with detailed 3D full-body information that could be used to learn group behaviors utilizing our models.

2.2 Behavior Recognition Methods

Analysis of multi-agent interactions benefits from human motion recognition that the group behavior composed by the action of each group member is more interpretable. Human behavior recognition has been explored by many researchers, and it could be grouped into vision-based approaches and skeleton-based approaches. While the vision-based approaches has be addressed in numerous works [37], the complex factors such as scenario, occlusion, pose estimation error limit the performance of vision-based approaches. On the other side, the skeleton data recorded by Motion Capture systems (MoCap) are stable with respect to external factors, we thus collect our data using motion capture and focus related works on skeleton-based approaches (see [12] for a review).

From the model perspective, behavior recognition approaches also could be categorized into machine learning algorithms with hand-crafted features and end-to-end deep learning methods [37]. Historically, hand-crafted machine learning algorithms are highly active in the topic of human behavior recognition. These works have been using Hidden Markov Models [31], K-means and SVM [15] to learn behavior recognition models. However, these methods rely on hand-crafted features. With the dawn of deep learning, deep learning algorithms have been used to achieve great success. Recurrent Neural Networks (RNNs), specially Long Short-Term Memory (LSTM) models, have shown extraordinary performance on human behavior recognition by capturing the sequential information [8, 38]. In addition, attention-based models [30, 36] utilize attention mechanisms together with RNNs to focus on important information in behavior recognition. Recent graph-based methods use Graph Convolutional Networks (GCN) [17, 22] on constructed human skeleton graphs for behavior recognition. Yan et al. [33] propose the Spatial-Temporal Graph Convolutional Networks (ST-GCN) to extract spatial and temporal features. Li et al. [19] use the Actional-Structural Graph Convolutional Networks (AS-GCN), which combines actional and structual links into a generalized skeleton graph. Inspired by the success of attention- and graph-based methods, in this work, we adopt both approaches for multi-agent interactions.

All of the above methods focus on single person behavior recognition. As for group behavior recognition, Ibrahim et al. [14] propose a two-stage LSTM model to recognize group activity. [5] uses a set of interconnected RNNs to jointly capture the actions of individuals. However, these RNN-based methods ignore the physical relations among group members and suffer from the lack of flexibility. We adopt graph-based models to represent the relations among agents. In [32], an actor relation graph is proposed to capture the appearance and position relation between actors, but it simplifies the individual behaviors as the changes in body positions without considering skeletons or body joints. In our work, the skeleton of each individual is used for training which contains more detailed behavioral information in behavior recognition. In contrast to previous works, we develop AG-GCN that combines Multi-Spatial-Temporal GCN and Group-GCN to address the graphical relations between body markers and group members. Meanwhile, we include body distance and head orientation for better interaction recognition performance.

3 Methods

In this section, we present a novel dataset of group behaviors collected with motion capture (Section 3.1). Then we develop novel behavior recognition models, i.e., attention-based models (Section 3.2) and graph-based models (Section 3.3), trained on our dataset.

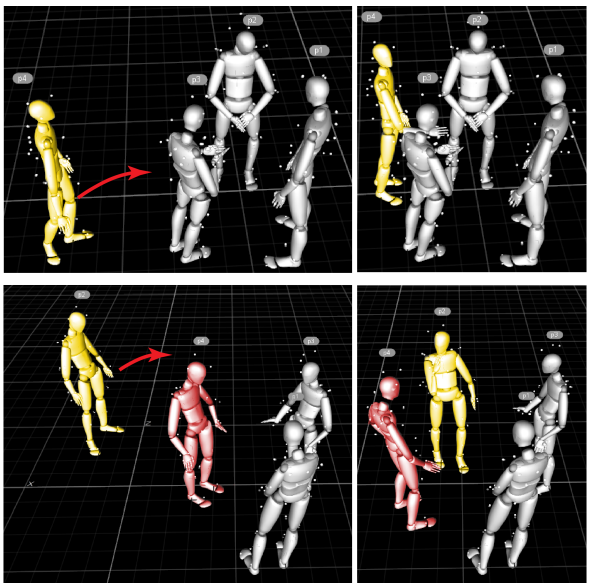


Figure 1: Two group behaviors when the newcomer (yellow character) approaches to join the group. The red arrow indicates the movement of the newcomer. The group members stand still and ignores the newcomer purposefully (top). The group members accommodate the newcomer, with one group member (red character) moving backwards in order to make space for them (bottom). All skeletons above are reconstructed from our collected data.

3.1 Dataset Collection

To provide a scenario for group interaction behaviors, we adopted a game scene called *Who's the Spy*. Forty participants (27F:13M) aged between 22-35 years old (M=25.8, SD=3.2) were recruited from the local city and the university through public bulletins and online advertisement to participate in the motion capture sessions. Three small booklets were distributed to three group players, and each booklet

contains 40 word cards with an order that ensures only one different but synonymous word exists in one game round. For example, the first word cards from the three booklets are *Bee*, *Bee*, and *Butterfly*, where *Butterfly* is the spy word. Each player takes turns to play as the newcomer, and shifts after 10 game rounds. In each round, group members take turns to describe the word at hand, and the newcomer observes the group from 1.5 meters outside the group. Once the spy is determined, the newcomer will approach and join the group to inform the judgment.

The motion data of each participant was recorded with a motion capture suit with 37 markers and a NaturalPoint Optitrack system[1]. The group behaviors are labeled into two general types, *Accommodate* and *Ignore*, which corresponds to the behaviors of the group as the newcomer is approaching to join. These group behaviors are observed in real-life datasets [1, 13] and experiments [29]. Figure 1 shows two randomly sampled behaviors from our dataset using reconstructed skeletons from full-body markers.

3.2 Attention-Based Models

In this section, we present attention-based models that we train and evaluate on our collected dataset. Attention mechanisms-based recurrent neural networks have been successfully applied to human behavior recognition [30, 36]. Such models benefit from the attention mechanism that it enables the model to automatically focus on important spatial and temporal information during individual behavior recognition. Inspired by that, we develop attention-based models for group behavior recognition during human-group interactions. Two models are presented in this section including Approach Group Net (*AGNet*) and Approach Group Transformer (*AGTransformer*). These two models share a similar architecture with different types of agent encoders.

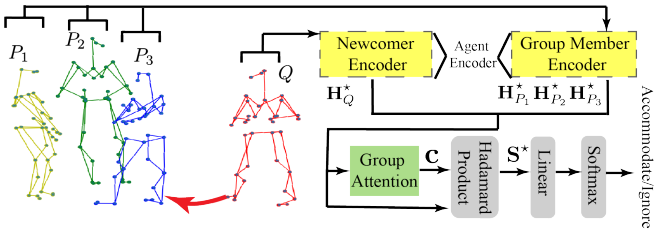


Figure 2: The overview of the attention-based architecture. The full-body markers of three group members are encoded through a shared *Group Member Encoder* and the newcomer is encoded through another *Newcomer Encoder*. Both encoders are instantiated from the *Agent Encoder* which encodes the importance of the spatial and temporal information. The output from the agent encoders is fed to the *Group Attention Module* in order to find which group member exerts more impact on the overall group behaviors. The output is then used to classify the group behavior to be *Accommodate* or *Ignore*

The overview of the attention-based architecture is shown in Figure 2. There are three humans P_1, P_2, P_3 in a conversational group and a newcomer Q approaches to join the group. The input is a sequence of tensors which contains 3D positions of full-body markers from each agent during a period of time. The *Agent Encoder* (dashed yellow boxes in Figure 2) is developed to extract the temporal and spatial information of all the markers from both group members (*Group Member Encoder*) and the newcomer (*Newcomer Encoder*). Note that the details of the Agent Encoder will be discussed later in separate models (see Section 3.2.1&3.2.2). The output of the Agent Encoder encodes the full-body markers of an agent with the focus on important markers at important time frames. It has the form $\mathbf{H}^\star_{P_i} = [h^1_{P_i}, h^2_{P_i}, \cdots, h^K_{P_i}]^T$ and $\mathbf{H}^\star_Q$, where $i = 1, 2, 3$ for three group members, and K is the hidden layer dimension.

Utilizing the Agent Encoder, the network is able to encode the important spatial and temporal information from each group member and the newcomer. We thus need a higher-level Group Attention (GA) module to find out which group member exerts a larger impact on the overall group behaviors. The group attention score is computed with two fully-connected layers with *tanh* and *softmax* activations:

$$\mathbf{C} = softmax(tanh(\mathbf{W}_{group}\mathbf{H}^\star)) \tag{1}$$

where $\mathbf{W}_{group}$ is a weight matrix and $\mathbf{H}^\star = [\mathbf{H}^\star_{P_1}, \mathbf{H}^\star_{P_2}, \mathbf{H}^\star_{P_3}, \mathbf{H}^\star_Q]$. The group attention score is then used to modulate the output of the Agent Encoders:

$$\mathbf{S}^\star = \mathbf{C} \odot \mathbf{H}^\star \tag{2}$$

where $\mathbf{S}^\star$ is a $K \times 4$ matrix which encodes the importance of both group members and the newcomer, and $\odot$ represents Hadamard product. This output from the group attention module is fed to a fully-connected layer with a softmax activation to classify the type of the group behavior.

Two attention-based models will be presented as follows, and they share the similar architecture as shown in Figure 2 with differences in the Agent Encoder.

3.2.1 *The AGNet architecture*

The Agent Encoder of the AGNet (see Figure 3) is implemented with two modules: a temporal attention module, separate for each marker, followed by a body attention module which learns the subset of the body markers that play an important role in the full-body behaviors.

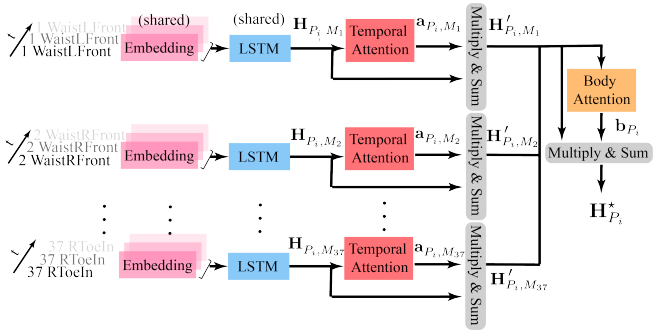


Figure 3: The Agent Encoder of the AGNet.

Temporal Attention Module A shared Long Short-Term Memory (LSTM) is used to extract the temporal information for each of the 37 markers independently, i.e., given a marker M_m of a group member P_i, the output of the LSTM encoder is a $K \times T$ matrix of hidden states $\mathbf{H}_{P_i,M_m} = [\mathbf{H}^1_{P_i,M_m}, \mathbf{H}^2_{P_i,M_m}, \cdots, \mathbf{H}^T_{P_i,M_m}]$, where T is the temporal length of the input data matrix, and $\mathbf{H}^t_{P_i,M_m} = [h^{t,1}_{P_i,M_m}, h^{t,2}_{P_i,M_m}, \cdots, h^{t,K}_{P_i,M_m}]^T$, where K is the hidden layer dimension. These outputs are then fed into the *Temporal Attention Module*. Wang et al. [30] justified that a 1×1 convolutional layer can help to reduce the number of trainable parameters compared with a fully-connected layer, i.e., limits irrelevant temporal connections. We thus use a 1×1 convolutional layer with a softmax activation to learn the temporal attention score $\mathbf{a}_{P_i,M_m} = softmax(\mathbf{W}_{temp}\mathbf{H}_{P_i,M_m})$,

[1] https://optitrack.com/

where $\mathbf{W}_{temp} \in \mathbb{R}^{1\times K}$ is a weight matrix. The temporal attention score is further used to combine the original output of the LSTM encoder for different moments in time:

$$\mathbf{H}'_{P_i,M_m} = \sum_{t=1}^{T} a^t_{P_i,M_m} \mathbf{H}^t_{P_i,M_m} \tag{3}$$

Body Attention Module The temporal attention module has encoded the information from each marker separately. We thus need a body attention module to learn a body attention score for each marker, in order to better understand the subset of the full-body markers that play an important role in the full-body behaviors. Similar to [30], two fully-connected layers with *tanh* activation and *softmax* activation are used to compute the body attention score:

$$\mathbf{b}_{P_i} = softmax(tanh(\mathbf{W}_{body}\mathbf{H}'_{P_i})) \tag{4}$$

where $\mathbf{H}'_{P_i} = [\mathbf{H}'_{P_i,M_1}, \mathbf{H}'_{P_i,M_2}, \cdots, \mathbf{H}'_{P_i,M_m}]$, and $\mathbf{W}_{body}$ is a weight matrix. The body attention score is thus used to merge the output of the temporal attention module for different markers:

$$\mathbf{H}^{\star}_{P_i} = \sum_{m=1}^{37} b_{P_i,M_m} \mathbf{H}'_{P_i,M_m} \tag{5}$$

As mentioned above each $\mathbf{H}^{\star}_{P_i}$ then goes through group attention. The vector $\mathbf{H}^{\star}_{Q}$ representing the newcomer is computed in a similar manner, but in another instance of the Agent Encoder, i.e. Newcomer Encoder.

3.2.2 *The AGTransformer architecture*

The Transformer model [27] has proven to be successful in learning a better representation of each element in a sequence for machine translation tasks. This has inspired us to apply Transformer layers in our AGTransformer model as a self-attention mechanism to deal with sequential information. The Agent Encoder of the AGTransformer uses two transformer layers, Marker Transformer and Body Transformer. The Marker Transformer learns a deeper representation for each body marker by capturing its self-attention on different time frames, and the Body Transformer captures the self-attention of each marker related to others. Besides a marker embedding layer which embeds all input markers into a fixed dimension vector, we have a positional embedding layer which encodes the temporal position of each marker similar to the word position in [27].

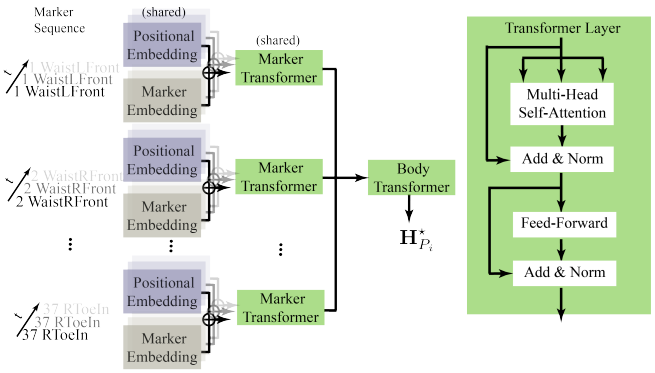


Figure 4: The Agent Encoder of the AGTransfomer.

Transformer Layer The Transformer layer (see the right green box in Figure 4) contains a multi-head self-attention layer and a feed-forward layer, and each of these two layers has a residual connection followed by a standard normalization step. The multi-head self-attention is defined as:

$$MH(\mathbf{H}) = Concat(head_1, head_2, \cdots, head_h)\mathbf{W}^h \tag{6}$$

where $\mathbf{H}$ is the embedded matrix of each marker on sequential frames, $\mathbf{W}^h \in \mathbb{R}^{h\times K}$ represents a weight matrix, and $head_i = Attention(\mathbf{HW}^{\mathbf{Q}}_i, \mathbf{HW}^{\mathbf{K}}_i, \mathbf{HW}^{\mathbf{V}}_i)$ is a scaled dot-product attention:

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = softmax(\frac{\mathbf{QK^T}}{\sqrt{d}}\mathbf{V}) \tag{7}$$

Here $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ represent query, key, and value vectors of length d (see [27] for details), $\mathbf{W}^{\mathbf{Q}}_i, \mathbf{W}^{\mathbf{K}}_i, \mathbf{W}^{\mathbf{V}}_i \in \mathbb{R}^{K\times d}$ are projection matrices, and K is the embedding dimensionality. The output is then fed into a feed-forward layer. Note that both dropout and LeakyReLU are used in the multi-head self-attention layer and the feed-forward layer to avoid overfitting. The output is passed through Multilayer perceptrons (MLP) to reduce dimensionality, e.g., MLP after the Marker Transformer reduces the temporal dimension.

We use two Transformer layers in our AGTransformer architecture in order to learn a representation for an agent by learning the self-attention of the body markers during a period of time. The final output is embedded to the same dimension as the output from the AGNet Agent Encoder through MLP.

In summary, the attention-based models share the same architecture with differences in the Agent Encoder. On the individual level, the full-body markers of each agent are encoded through the Agent Encoder, and on the group level, the output from each agent encoder is fed to the Group Attention module to combine them with attention weights before it is sent to a classifier.

3.3 Graph-Based Models

We introduce Approach Group Graph Convolutional Networks (AG-GCN) for the group behavior recognition. Figure 5 shows an overview of AG-GCN. The input data is a skeleton graph. The skeleton graph construction is described in Section 3.3.1. Hierarchically, the model consists of two levels, the individual level, and the group level. We discuss these two levels in Section 3.3.2. Finally, we dive into the multi-temporal model in Section 3.3.3.

3.3.1 *Skeleton Graph Construction*

We create a spatial-temporal graph from the sequence of marker coordinates. The format of the input data of the graph neural network is significantly different from the one in the attention models. As described in Section 3.2, the features of all markers are concatenated to one vector. In the graph neural network, we convert the data to a graph structure based on the spatial structure of the human skeleton with related markers. For example, as shown in Figure 6(a), markers (each one has an ID) are connected to construct the skeleton graph.

Spatially, the skeleton graph can be represented as an undirected graph $G_S = (V_S, E_S)$, where V_S is the set of nodes, and E_S is the set of edges. Each marker is a node, and there exist edges between connected markers. For the skeleton with N nodes, $V_S = \{v_i \mid i = 1, ..., N\}$. There also exist temporal connections that connect the same marker nodes in consecutive frames. For a sequence

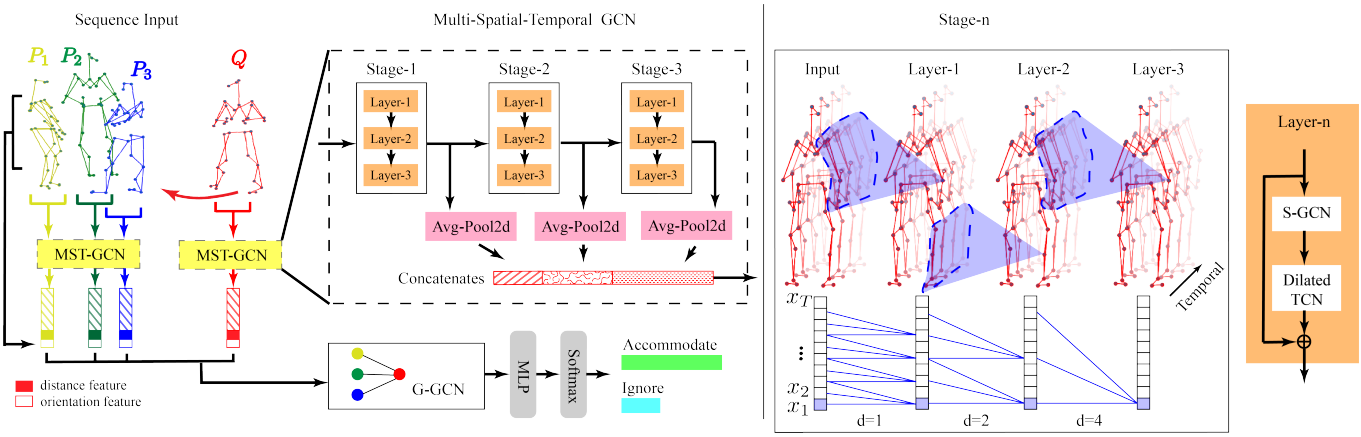
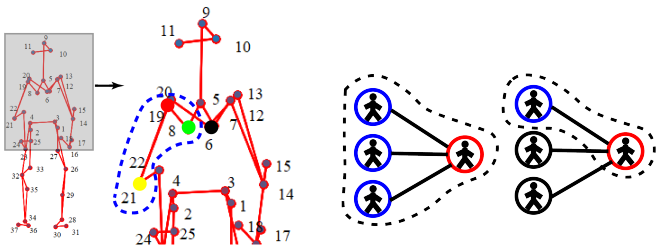


Figure 5: Overview of the *Approach Group Graph Convolutional Neural Network (AG-GCN)* for group behavior analysis. The full-body markers are connected as skeleton graphs and fed into the *Multi-spatial-temporal Graph Convolutional Network (MST-GCN)* which encodes the marker's spatial relationships and movement over time. The group members (P1, P2, P3) share the same model, while the newcomer (Q) is modeled through another *MST-GCN* model. The output from the *MST-GCN* module is then fed to the *Group Graph Convolutional Neural Network (G-GCN)* Module which encodes the group's spatial relationships. The output is used to classify the group behavior to either *Accommodate* or *Ignore*.



(a) Spatial configuration partition (For intra-body)

(b) Distance partition (For intra-group)

Figure 6: The partition strategy: (a) Spatial configuration partition strategy. On the individual level, the nodes in a neighbor set are divided into three sets: the node itself, the nodes that are closer to the center of the graph, and the nodes that are farther away. (b) Distance partition strategy. On the group level, the nodes in a neighbor set are divided into two sets: the node itself and the neighbor nodes.

with T frames, $v_{t,i}$ connects to $v_{t-1,i}$ and $v_{t+1,1}$ along the temporal dimension. For the node v_i, the temporal connections are represented as $G_i = (V_i, E_i)$, where $V_i = \{v_{t,i} \mid t = 1, ..., T\}$, and E_i represents the temporal edges. The whole graph sequence is composed of the spatial graph and temporal graph.

3.3.2 *Spatial Graph Convolutional Neural Network*

Spatial Graph Convolutional Neural Network (S-GCN) extends convolution operations on images [7] to graphs. On graphs, we can define a sampling function on a node and its neighbor set. Unlike image convolutions, in a skeleton graph, the nodes within a neighborhood do not have a fixed spatial order. To address this problem, Kipf et al. [17] proposed a strategy that is equivalent to calculate the inner product of the average feature vector in the set and a weight vector. Yan et al. proposed spatial configuration partition and distance partition [33]. In our implementations, we follow the same idea. Hierarchically, the whole model is divided into two levels, the individual level and group level. The individual level adopts the spatial configuration partition and the group level adopts the distance partition. The spatial configuration partition divides neighborhoods into three subsets, i.e., the node itself, the nodes that are closer to the center of the graph, and the nodes that are farther away.

For example, as shown in Figure 6(a), we assume the chest (marker-6, the black node) is the center of the body. Within the neighbor set of marker-19 (within the blue dotted line), there are three subsets: the marker-8 (the green node, closer to the center), the marker-19 (the red node, the node itself), and the marker-22 (the yellow node, farther away from the center). On the group level, the distance partition divided nodes of agents set into two subsets, i.e., the node itself and the neighbor nodes. When the newcomer (agent in the red circle) is the root node (to the left in Figure 6(b)), there are two subsets, the newcomer itself and all other group members (agents in the blue circles). When one of the group members is the root node (to the right in Figure 6(b)), the two subsets are this group member and the newcomer. After dividing the points in a neighbor sets into several subsets, we can determine the spatial order based on the order of the subsets. For example, in distance partition, the index of the root node itself is 0, the index of the neighbor nodes is 1.

Using neighborhood subsets defined for both individual and group levels, graph convolutions are performed by the corresponding networks, S-GCN (to the right in Figure 5) and G-GCN (at the bottom left). With x_i and y_i being the feature maps of node v_i before and after a graph convolution operation, a graph convolution can be defined as:

$$y_i = \sum_{v_j \in S_i} \frac{x_j}{D_{v_i}(v_j)} w(l_{v_i}(v_j)), \tag{8}$$

where S_i is the set of neighbor nodes of v_i, w is a weight function, $l_{v_i}(v_j)$ is a mapping from v_j to the index of its corresponding subset and the normalizing term $D_{v_i}(v_j)$ is the number of nodes in this subset. Essentially, an average is computed for each subset, with the output being a weighted sum of these averages.

To improve the results, we calculate distance and angle features, and concatenate these features to the end of the output of the previous level as the input of the group level. As stated in [34, 35], body distances and head orientations between two people in group interactions have shown to be essential factors in group behavior analysis. For group members, the distance feature is the distance to the newcomer, the angle feature is the angle between the head orientations of the member and the newcomer. For the newcomer, the values of

these two features are the average value of group member features.

3.3.3 *Multi-Temporal Convolutional Neural Network*

Before delving into the multi-temporal convolutional neural network, we first study the temporal convolutional neural network (TCN). Instead of ordinary convolutions along the temporal dimension, TCN uses dilated convolutions (shown in the 'Stage-n' box of Figure 5) to enable larger receptive fields for higher layers of the network [4]. Given the frames of a sequence $x_{1:T} = (x_1, ..., x_T)$, and a filter kernel $f_k, k = 0, ..., K-1$, the dilated convolution operation on x_t, in the temporal domain, is defined as:

$$y_t = \sum_{i=0}^{K-1} f_k \; x_{t-dk}, \tag{9}$$

where d is the dilation factor, and K is the filter size. Residual connections are further adopted to promote gradients flow to speed up training and improve accuracy. In the residual block, the inputs are added to the outputs (orange rectangle 'Layer-n' in Figure 5). From one layer to the next, the dilation factor d is doubled.

In the skeleton graph construction (Section 3.3.1), nodes are connected to the same nodes of consecutive frames in the temporal domain. Similar to TCN applied to image sequences [4], TCN on graph sequences can be extended to multiple stages. In the multi-stage TCN model [9], the input to the first stage is the original sequence. Each stage generates a refined prediction based on the previous stage:

$$\begin{aligned} Y^0 =& X_{1:T}, \\ Y^s =& \mathcal{F}(Y^{s-1}), \end{aligned} \tag{10}$$

where $X_{1:T}$ is the original sequence, $\mathcal{F}$ is each stage, and Y^s is the output of stage s. We stack several stages sequentially, and concatenate the prediction of each stage, as illustrated in Figure 5.

In summary, in contrast with ST-GCN [33] that only has one stage with multiple layers without dilation, MST-GCN computes a feature vector that is a concatenation of features from a series of stages. Each stage consists of a number of residual layers, where each such layer includes a spatial GCN followed by a dilated TCN over the temporal domain, with a residual connection. AG-GCN further adds a group GCN on the output of the all MST-GCNs of the group, before the combined result is sent to a classifier.

4 Experiment

Data Preparation We run a sliding window over data sequences to pre-process the data. The window length is 180 with an overlap ratio of 0.75. All samples are down-sampled to 60 frames. 5-fold cross-validation is further applied to make full use of the data.

Implementation Details The data source[2] and code[3] can be found here. AGNet is trained with 16 embedding dimensions and the LSTM encoder contains 3-layer LSTM networks. The dimensions of the hidden state is 16 for each layer. Dropout with probability of 0.5 is used after each LSTM layer. AGTransformer also has 16 embedding dimensions, and the transform layer has 8 heads with the query, the key, and the value vector size set to be 64.

As for the model details of AG-GCN, hierarchically, the AG-GCN model is composed of two levels. For the individual level, there are three temporal stages, and each stage has three layers. The number of channels in these three temporal stages is 64, 128, and 256. A pooling layer with a stride of 2 exists between every two stages. The size of the temporal kernel is 9 and the dilation factor is doubled at each layer. An average pooling is performed after each stage, and we concatenate the features as the input of the group level. For the group level, the number of channel is 64. The GCN is connected with a fully connected layer and softmax classifier.

[2] https://www.csc.kth.se/~chpeters/ESAL/infrastructure.html
[3] https://github.com/YIN95/Group-Behavior-Recognition

4.1 Experimental Results

The classification performances of the attention-based neural networks and the graph-based neural networks are presented in Table 1. We can see that the graph-based models generally perform better than the attention-based ones with higher F1 scores, and AG-GCN with additional features of body distance and head orientation achieved the best performance (highest F1 score).

Table 1: Confusion matrix and F1 score for group behavior classification. "GT" means *Ground truth*, "A" means *Accommodate* and "I" stands for *Ignore*. For the meaning of each abbreviation of networks, please refer to Section 4.1.

	GT	A	I	F1 score
AGNet	A	3940 (76.27%)	1226 (23.73%)	0.754
	I	1345 (38.42%)	2156 (61.58%)	
AGTransformer	A	4825 (93.40%)	341 (6.60%)	0.842
	I	1473 (42.07%)	2028 (57.93%)	
ST-GCN + Group Attention	A	4713 (91.23%)	453 (8.77%)	0.892
	I	688 (19.65%)	2813 (80.35%)	
ST-GCN + Group GCN	A	4763 (92.20%)	403 (7.80%)	0.919
	I	438 (12.51%)	3063 (87.49%)	
MST-GCN + Group Attention	A	4806 (93.03%)	360 (6.97%)	0.926
	I	411 (11.73%)	3090 (88.27%)	
AG-GCN (MST-GCN + Group GCN)	A	4822 (93.32%)	344 (6.68%)	0.930
	I	389 (11.11%)	3112 (88.89%)	
AG-GCN (dis & ori)	**A**	**4839 (93.67%)**	**327 (6.33%)**	**0.941**
	I	**276 (7.88%)**	**3225 (92.12%)**	

AGTransformer achieves comparable True Positive (TP) with the graph-based models. A possible reason is that the transformer layer provides a better capability to capture the sequential information of all markers than a naive attention module in AGNet. AGTransformer has high False Positives (FP) as it recognized some Ignore behaviors to be Accommodate. The reason might be that the markers are passed to the Body Transformer layers without ordering, and it makes AGTransformer fail in attending to the right markers which are representative in Accommodate behaviors. Note that Ignore behaviors are not static and contain body motions, and these motions are acted mostly within the group rather than to the newcomer.

Then we evaluate the graph-based networks by analyzing the effectiveness of the proposed modules in AG-GCN. We first compare

the spatial-temporal graph convolutional neural network (ST-GCN) [33] with Group Attention (GA) to AGNet, letting ST-GCN replace the body attention module and temporal attention module in AGNet. Seen from Table 1, ST-GCN+GA significantly outperforms the attention-based methods. In AGNet and AGTransformer, all body markers are simply concatenated as the input features without spatial ordering. However, in ST-GCN, the spatial information among markers is naturally preserved by using the graph structure, and the motion trajectory is expressed in the form of temporal-edges. The use of graph convolutional networks enhances the association among body markers and reduces the complexity of networks.

We also evaluate the efficiency of using graph neural networks on group level by combining ST-GCN with a group-level GCN (ST-GCN+Group GCN), in effect replacing the Group Attention module in ST-GCN+GA with a GCN. On the individual level, each agent is thus modeled by ST-GCN, while on the group level, a GCN is utilized to model the spatial relationship among group members. With these two levels, the performance is improved further.

In ST-GCN+GA and ST-GCN+Group GCN, a single-stage temporal convolutional network (TCN) without dilation is adopted. To verify the multi-temporal-stage architecture is better than a single-temporal-stage one, we train multi-temporal-stage networks that have the same number of parameters as the single-stage one. We can observe in Table 1 that AG-GCN (MST-GCN+GA) outperforms ST-GCN+GA. Applying multiple temporal stages to the ST-GCN, the AG-GCN enhances the F1 score to 0.930. Even if AG-GCN already performs quite well, the recognition performance can still be improved by concatenating the features of body distance and head orientation to the output of each MST-GCN.

5 Virtual Reality Use Case

In this section, we present a use case in a virtual environment to show how the model is being applied. There is a common pattern of online multi-agent interactions that small groups of people gather and stand in an environment to converse, e.g., VRChat[4]. The group behaviors should account for the interactions between the group and a newcomer that the group members either ignore the newcomer or react to them by adjusting their positions and orientations to accommodate the newcomer in the group formation better. The motivation for establishing such a virtual reality interaction environment is to show the capability of perceiving the group behaviors is desirable for artificial-intelligent agents, in a case that the agents should be socially-acceptable when approaching free-standing conversational groups. Such capability is also essential in the pedagogical domain where students could learn when and how to join a group politely in this virtual environment.

To generate such a virtual scenario for supporting multi-agent interactions, a cloud-based VR platform, SIGVerse[5] [25], is used. MIXAMO[6] 3D humanoid character with no facial features is used in order to ensure that the perception of the characters will exclusively result from the body behaviors. Similar to the aforementioned data collection scenario, four participants are engaged in the use case (see Figure 7 top-left), i.e., three group members are in a free-standing conversational group and one newcomer approaches the group to join the conversation.

Each of these four participants controls one virtual character by a VR device (see Figure 7 bottom), and four VR devices are thus used including three Oculus Rift S and one Oculus Rift CV1. The VR devices track the head and hand movements, and these data are simultaneously transferred to control the virtual characters. Note that the lower body motions are resolved by the built-in Inverse Kinematics (IK) system in SIGVerse. The full-body motion data are passed to our trained AG-GCN model to determine whether the group members accommodate or ignore the newcomer in real-time (see Figure 7 top-right). Note that the participants can operate and control the virtual characters from separate places since the scenario is cloud-based.

Figure 7: The virtual reality use case. The perspective view of the multi-agent interaction scenario, where a newcomer approaches to join a conversational group (top-left). The first-person view of the newcomer with the normalized probability of group behaviors represented by color bars (top-right). Each participant controls one virtual character with one VR device (bottom).

6 Conclusion

In this paper, we propose novel attention- and graph-based models to recognize dynamic group behaviors when a group is approached by a newcomer. A novel full-body motion capture dataset of conversational groups is collected to understand and learn group behaviors. The provided experimental results show the graph-based models outperform attention-based models by leveraging graph convolutional networks to learn the representations within the agent body and the human group. We further present a use case in a virtual environment to recognize group dynamics in real-time using a trained AG-GCN.

In social scenarios, dynamic group behavior recognition has rarely been studied due to its complexity and the lack of data. This has motivated us to apply our models to dynamic group behaviors in a ubiquitous scenario where a conversational group is approached by a newcomer. We believe our models and methods, which involve virtual environments, are suitable for extension to other group behaviour scenarios with minor modifications to the architecture. In the future, we plan to generate autonomous and socially-acceptable behaviors for an agent/robot to approach or coordinate with groups.

Acknowledgements

This research has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n. 824160 (EnTimeMent). This research is also supported by the 2019 NII International Internship Program and Inamura Lab.

[4]https://www.vrchat.com/
[5]http://www.sigverse.org/wiki/en/
[6]https://www.mixamo.com/

REFERENCES

[1] Xavier Alameda-Pineda, Jacopo Staiano, Ramanathan Subramanian, Ligia Batrinca, Elisa Ricci, Bruno Lepri, Oswald Lanz, and Nicu Sebe, 'Salsa: A novel dataset for multimodal group behavior analysis', *IEEE transactions on pattern analysis and machine intelligence*, **38**(8), 1707–1720, (2015).

[2] Xavier Alameda-Pineda, Yan Yan, Elisa Ricci, Oswald Lanz, and Nicu Sebe, 'Analyzing free-standing conversational groups: A multimodal approach', in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 5–14. ACM, (2015).

[3] Philipp Althaus, Hiroshi Ishiguro, Takayuki Kanda, Takahiro Miyashita, and Henrik I Christensen, 'Navigation for human-robot interaction tasks', in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2, pp. 1894–1900. IEEE, (2004).

[4] Shaojie Bai, J Zico Kolter, and Vladlen Koltun, 'An empirical evaluation of generic convolutional and recurrent networks for sequence modeling', *arXiv preprint arXiv:1803.01271*, (2018).

[5] Sovan Biswas and Juergen Gall, 'Structural recurrent neural network (srnn) for group activity analysis', in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1625–1632. IEEE, (2018).

[6] Marco Cristani, Loris Bazzani, Giulia Paggetti, Andrea Fossati, Diego Tosato, Alessio Del Bue, Gloria Menegaz, and Vittorio Murino, 'Social interaction discovery by statistical analysis of f-formations.', in *BMVC*, volume 2, p. 4, (2011).

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, 'Deformable convolutional networks', in *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, (2017).

[8] Yong Du, Wei Wang, and Liang Wang, 'Hierarchical recurrent neural network for skeleton based action recognition', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110–1118, (2015).

[9] Yazan Abu Farha and Jurgen Gall, 'Ms-tcn: Multi-stage temporal convolutional network for action segmentation', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3575–3584, (2019).

[10] Yuan Gao, Fangkai Yang, Martin Frisk, Daniel Hernandez, Christopher Peters, and Ginevra Castellano, 'Social behavior learning with realistic reward shaping', in *2019 28th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, (2019).

[11] Erving Goffman, *Encounters: Two studies in the sociology of interaction*, Ravenio Books, 1961.

[12] Fei Han, Brian Reily, William Hoff, and Hao Zhang, 'Space-time representation of people based on 3d skeletal data: A review', *Computer Vision and Image Understanding*, **158**, 85–105, (2017).

[13] Hayley Hung and Ben Kröse, 'Detecting f-formations as dominant sets', in *Proceedings of the 13th international conference on multimodal interfaces*, pp. 231–238. ACM, (2011).

[14] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori, 'A hierarchical deep temporal model for group activity recognition', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1971–1980, (2016).

[15] Ioannis Kapsouras and Nikos Nikolaidis, 'Action recognition on motion capture data using a dynemes and forward differences representation', *Journal of Visual Communication and Image Representation*, **25**(6), 1432–1445, (2014).

[16] Adam Kendon, *Conducting interaction: Patterns of behavior in focused encounters*, volume 7, CUP Archive, 1990.

[17] Thomas N Kipf and Max Welling, 'Semi-supervised classification with graph convolutional networks', *arXiv preprint arXiv:1609.02907*, (2016).

[18] Annica Kristoffersson, Kerstin Severinson Eklundh, and Amy Loutfi, 'Measuring the quality of interaction in mobile robotic telepresence: A pilot's perspective', *International Journal of Social Robotics*, **5**(1), 89–101, (2013).

[19] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian, 'Actional-structural graph convolutional networks for skeleton-based action recognition', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2019).

[20] Sai Krishna Pathi, 'Join the group formations using social cues in social robots', in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, (2018).

[21] Francesco Setti, Oswald Lanz, Roberta Ferrario, Vittorio Murino, and Marco Cristani, 'Multi-scale f-formation discovery for group detection', in *2013 IEEE International Conference on Image Processing*, pp. 3547–3551. IEEE, (2013).

[22] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu, 'Skeleton-based action recognition with directed graph neural networks', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7912–7921, (2019).

[23] Peter Stone and Manuela Veloso, 'Multiagent systems: A survey from a machine learning perspective', *Autonomous Robots*, **8**(3), 345–383, (2000).

[24] Ron Sun et al., *Cognition and multi-agent interaction: From cognitive modeling to social simulation*, Cambridge University Press, 2006.

[25] Jeffrey Too Chuan Tan and Tetsunari Inamura, 'Sigverse-a cloud computing architecture simulation platform for social human-robot interaction', in *2012 IEEE International Conference on Robotics and Automation*, pp. 1310–1315. IEEE, (2012).

[26] Xuan-Tung Truong and Trung-Dung Ngo, 'To approach humans?: A unified framework for approaching pose prediction and socially aware robot navigation', *IEEE Transactions on Cognitive and Developmental Systems*, **10**(3), 557–572, (2018).

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', in *Advances in neural information processing systems*, pp. 5998–6008, (2017).

[28] Marynel Vázquez, Aaron Steinfeld, and Scott E Hudson, 'Parallel detection of conversational groups of free-standing people and tracking of their lower-body orientation', in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3010–3017. IEEE, (2015).

[29] Jered Vroon, Michiel Joosse, Manja Lohse, Jan Kolkmeier, Jaebok Kim, Khiet Truong, Gwenn Englebienne, Dirk Heylen, and Vanessa Evers, 'Dynamics of social positioning patterns in group-robot interactions', in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 394–399. IEEE, (2015).

[30] Chongyang Wang, Min Peng, Temitayo A Olugbade, Nicholas D Lane, Amanda C De C Williams, and Nadia Bianchi-Berthouze, 'Learning bodily and temporal attention in protective movement behavior detection', *arXiv preprint arXiv:1904.10824*, (2019).

[31] Di Wu and Ling Shao, 'Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–731, (2014).

[32] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu, 'Learning actor relation graphs for group activity recognition', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9964–9974, (2019).

[33] Sijie Yan, Yuanjun Xiong, and Dahua Lin, 'Spatial temporal graph convolutional networks for skeleton-based action recognition', in *Thirty-Second AAAI Conference on Artificial Intelligence*, (2018).

[34] Fangkai Yang and Christopher Peters, 'App-LSTM: Data-driven generation of socially acceptable trajectories for approaching small groups of agents', in *Proceedings of the 7th International Conference on Human-Agent Interaction*, pp. 144–152. ACM, (2019).

[35] Fangkai Yang and Christopher Peters, 'AppGAN: Generative adversarial networks for generating robot approach behaviors into small groups of people', in *2019 28th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, (2019).

[36] Ming Zeng, Haoxiang Gao, Tong Yu, Ole J Mengshoel, Helge Langseth, Ian Lane, and Xiaobing Liu, 'Understanding and improving recurrent networks for human activity recognition by continuous attention', in *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pp. 56–63. ACM, (2018).

[37] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du, and Duan-Sheng Chen, 'A comprehensive survey of vision-based human action recognition methods', *Sensors*, **19**(5), 1005, (2019).

[38] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie, 'Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks', in *Thirtieth AAAI Conference on Artificial Intelligence*, (2016).