Learning Actor Relation Graphs for Group Activity Recognition

Jianchao Wu Limin Wang Li Wang Jie Guo Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University, China

Abstract

Modeling relation between actors is important for recognizing group activity in a multi-person scene. This paper aims at learning discriminative relation between actors efficiently using deep models. To this end, we propose to build a flexible and efficient Actor Relation Graph *(ARG) to simultaneously capture the appearance and position relation between actors. Thanks to the Graph Convolutional Network, the connections in ARG could be automatically learned from group activity videos in an end-to-end manner, and the inference on ARG could be efficiently performed with standard matrix operations. Furthermore, in practice, we come up with two variants to sparsify ARG for more effective modeling in videos: spatially localized ARG and temporal randomized ARG. We perform extensive experiments on two standard group activity recognition datasets: the Volleyball dataset and the Collective Activity dataset, where state-of-the-art performance is achieved on both datasets. We also visualize the learned actor graphs and relation features, which demonstrate that the proposed ARG is able to capture the discriminative relation information for group activity recognition.* [1]

1. Introduction

Group activity recognition is an important problem in video understanding [56, 47, 52, 14] and has many practical applications, such as surveillance, sports video analysis, and social behavior understanding. To understand the scene of multiple persons, the model needs to not only describe the individual action of each actor in the context, but also infer their collective activity. The ability to accurately capture relevant relation between actors and perform relational reasoning is crucial for understanding group activity of multiple people [30, 1, 7, 23, 39, 12, 24, 59]. However, modeling the relation between actors is challenging, as we only have access to individual action labels and collective activity labels, without knowledge of the underlying interaction information. It is expected to infer relation between actors from other aspects such as *appearance similarity* and *relative location*. Therefore, it is required to model these two important cues when we design effective deep models for group activity understanding.

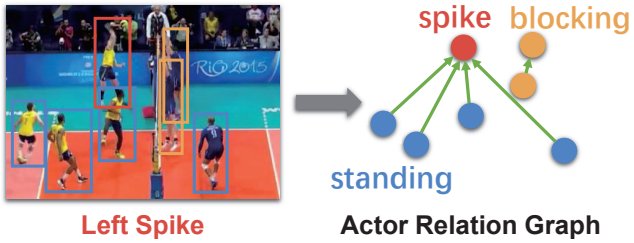

Figure 1: Understanding group activity in multi-person scene requires accurately determining relevant relation between actors. Our model learns to represent the scene by actor relation graph, and performs reasoning about group activity ("left spike" in the illustrated example) according to the graph structure and nodes features. Each node denotes an actor, and each edge represents the relation between two actors

Recent deep learning methods have shown promising results for group activity recognition in videos [3, 24, 45, 12, 32, 59, 23, 39]. Typically, these methods follow a two-stage recognition pipeline. First, the person-level features are extracted by a convolutional neural network (CNN). Then, a global module is designed to aggregate these person-level representations to yield a scene-level feature. Existing methods model the relation between these actors with an inflexible graphical model [23], whose structure is manually specified in advance, or using complex yet unintuitive message passing mechanism [12, 39]. To capture temporal dynamics, a recurrent neural network (RNN) is usually used to model temporal evolution of densely sampled frames [3, 24]. These models are generally expensive at computational cost and sometimes lack the flexibility dealing with group activity variation.

In this work, we address the problem of capturing appearance and position relation between actors for group activity recognition. Our basic aim is to model actor relation in a more flexible and efficient way, where the graphical connection between actors could be automatically learned from video data, and inference for group activity recogni-

[1] The code is available at https://github.com/wjchaoGit/Group-Activity-Recognition

tion could be efficiently performed. Specifically, we propose to model the actor-actor relation by building a *Actor Relation Graph* (ARG), illustrated in Figure 1, where the node in the graph denotes the actor's features, and the edge represents the relation between two actors. The ARG could be easily placed on top of any existing 2D CNN to form a unified group activity recognition framework. Thanks to the operation of graph convolution [29], the connections in ARG can be automatically optimized in an end-to-end manner. Thus, our model can discover and learn the potential relations among actors in a more flexible way. Once trained, our network can not only recognize individual actions and collective activity of a multi-person scene, but also on-the-fly generate the video-specific actor relation graph, facilitating further insights for group activity understanding.

To further improve the efficiency of ARG for long-range temporal modeling in videos, we come up with two techniques to sparsify the connections in ARG. Specifically, in spatial domain, we design a *localized* ARG by forcing the connection between actors to be only in a local neighborhood. For temporal information, we observe that slowness is naturally video prior, where frames are densely captured but semantics varies very slow. Instead of connecting any pair frame, we propose a *randomized* ARG by randomly dropping several frames and only keeping a few. This random dropping operation is able to not only greatly improve the modeling efficiency but also largely increase the diversity of training samples, reducing the overfitting risk of ARG.

In experiment, to fully utilize visual content, we empirically study different methods to compute pair-wise relation from the actor appearance features. Then we introduce constructing multiple relation graphs on an actors set to enable the model to focus on more diverse relation information among actors. We report performance on two group activity recognition benchmarks: the Volleyball dataset [25] and the Collective Activity dataset [7]. Our experimental results demonstrate that our ARG is able to obtain superior performance to the existing state-of-the-art approaches.

The major contribution of this paper is summarized as follows:

- We construct flexible and efficient actor relation graphs to simultaneously capture the appearance and position relation between actors for group activity recognition. It provides an interpretable mechanism to explicitly model the relevant relations among people in the scene, and thus the capability of discriminating different group activities.

- We introduce an efficient inference scheme over the actor relation graphs by applying the GCN with sparse temporal sampling strategy. The proposed network is able to conduct relational reasoning over actor interactions for the purpose of group activity recognition.

- The proposed approach achieves the state-of-the-art results on two challenging benchmarks: the Volleyball dataset [25] and the Collective Activity dataset [7]. Visualizations of the learned actor graphs and relation features show that our approach has the ability to attend to the relation information for group activity recognition.

2. Related Work

Group activity recognition. Group activity recognition has been extensively studied from the research community. The earlier approaches are mostly based on a combination of hand-crafted visual features with probability graphical models [1, 31, 30, 43, 6, 8, 17] or AND-OR grammar models [2, 46]. Recently, the wide adoption of deep convolutional neural networks (CNNs) has demonstrated significant performance improvements on group activity recognition [3, 24, 41, 45, 12, 32, 59, 23, 39]. Ibrahim *et al*. [24] designed a two-stage deep temporal model, which builds a LSTM model to represent action dynamics of individual people and another LSTM model to aggregate person-level information. Bagautdinov *et al*. [3] presented a unified framework for joint detection and activity recognition of multiple people. Ibrahim *et al*. [23] proposed a hierarchical relational network that builds a relational representation for each person. There are also efforts that explore modeling the scene context via structured recurrent neural networks [12, 59, 39] or generating captions [32]. Our work differs from these approaches in that it explicitly models the interactions information via building flexible and interpretable ARG. Moreover, instead of using RNN for information fusion, we employ GCN with sparse temporal sampling strategy which enables relational reasoning in an efficient manner.

Visual relation. Modeling or learning relation between objects or entities is an important problem in computer vision [35, 9, 22, 68, 57]. Several recent works focus on detecting and recognizing human-object interactions (HOI) [66, 16, 67, 5, 40], which usually requires additional annotations of interactions. In scene understanding, a lot of efforts have been made on modeling pair-wise relationships for scene graph generation [26, 34, 63, 65, 33, 62]. Santoro *et al*. [44] proposed a relation network module for relational reasoning between objects, which achieves superhuman performance in visual question answering. Hu *et al*. [21] applied an object relation module to object detection, and verified the efficacy of modeling object relations in CNN based detection. Besides, many works showed that modeling interactions information can help action recognition [60, 36, 15, 37, 50]. We show that explicitly exploit-

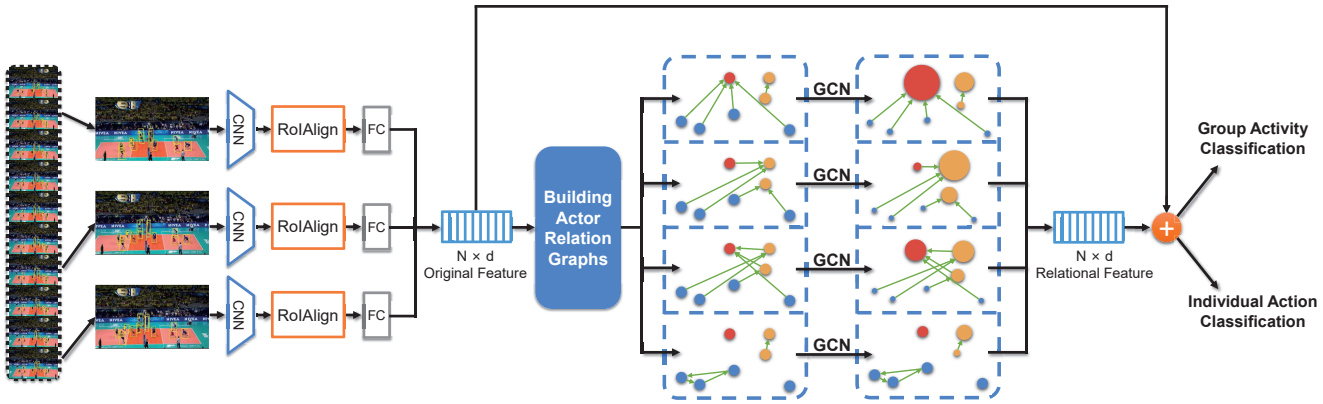

Figure 2: An overview of our network framework for group activity recognition. We first extract feature vectors of actors from sampled video frames. We use a d-dimension vector to represent an actor bounding box. And the total number of bounding boxes in sampled frames equals N. Multiple actor relation graphs are built to capture relation information among actors. Afterwards, Graph Convolutional Networks are used to perform relational reasoning on graphs. The outputs of all graphs are then fused to produce the relational feature vectors of actors. Finally, original feature and relational feature are aggregated and fed into classifiers of group activity and individual action.

ing the relation information can achieve significant gain on group activity recognition accuracy.

Neural networks on graphs. Recently, integrating graphical models with deep neural networks is an emerging topic in deep learning research. A considerable amount of models has arisen for reasoning on graph-structured data at various tasks, such as classification of graphs [13, 10, 38, 11, 27], classification of nodes in graphs [29, 18, 55], and modeling multi-agent interacting physical systems [28, 49, 4, 20]. In our work, we apply the Graph Convolutional Network (GCN) [29] which was originally proposed for semi-supervised learning on the problem of classifying nodes in a graph. There are also applications of GCNs to single-human action recognition problems [64, 61]. However, it would be inefficient to compute all pair-wise relation across all video-frame to build video as a fully-connected graph. Therefore, we build multi-person scene as a sparse graph according to relative location. Meanwhile, we propose to combine GCN with sparse temporal sampling strategy [58] for more efficient learning.

3. Approach

Our goal is to recognize group activity in multi-person scene by explicitly exploiting relation information. To this end, we build *Actor Relation Graph* (ARG) to represent multi-person scene, and perform relational reasoning on it for group activity recognition. In this section, we will give detailed descriptions of our approach. First, we present an overview of our framework. Then, we introduce how to build ARG. Finally, we describe the efficient training and inference algorithms for ARG.

3.1. Group Activity Recognition Framework

The overall network framework is illustrated in Figure 2. Given a video sequence and the bounding boxes of the actors in the scene, our framework takes three key steps. First, we uniformly sample a set of K frames from the video and extract feature vectors of actors from sampled frames. We follow the feature extraction strategy used in [3], which adopts Inception-v3 [51] to extract a multi-scale feature map for each frame. Besides that, we also have conducted experiments on other backbone models to verify the generality and effectiveness of our approach. We apply RoIAlign [19] to extract the features for each actor bounding box from the frame feature map. After that, a fc layer is performed on the aligned features to get a d dimensional appearance feature vector for each actor. The total number of bounding boxes in K frames is denoted as N. We use a $N \times d$ matrix $\mathbf{X}$ to represent feature vectors of actors.

Afterwards, upon these original features of actors, we build actor relation graphs, where each node denotes an actor. Each edge in the graphs is a scalar weight, which is computed according to two actors' appearance features and their relative location. To represent diverse relation information, we construct multiple relation graphs from a same set of actors features.

Finally, we perform learning and inference to recognize individual actions and group activity. We apply the GCN to conduct relational reasoning based on ARG. After graph convolution, the ARGs are fused together to generate relational representation for actors, which is also in $N \times d$ dimension. Then two classifiers respectively for recognizing individual actions and group activity will be applied on the pooled actors' relational representation and the original rep-

resentation. We apply a fully connected layer on individual representation for individual action classification. The actor representations are maxpooled together to generate scene-level representation, which is used for group activity classification through another fully connected layer.

3.2. Building Actor Relation Graphs

As mentioned above, ARG is the key component in our framework. We utilize the graph structure to explicitly model pair-wise relation information for group activity understanding. Our design is inspired by the recent success of relational reasoning and graph neural networks [44, 29].

Graph definition. Formally, the nodes in our graph correspond to a set of actors $A = \{(\mathbf{x}_i^a, \mathbf{x}_i^s)|i = 1, \cdots, N\}$, where N is the number of actors, $\mathbf{x}_i^a \in \mathbb{R}^d$ is actor i's appearance feature, and and $\mathbf{x}_i^s = (t_i^x, t_i^y)$ is the center coordinates of actor i's bounding box. We construct graph $\mathbf{G} \in \mathbb{R}^{N \times N}$ to represent pair-wise relation among actors, where relation value $\mathbf{G}_{ij}$ indicates the importance of actor j's feature to actor i.

In order to obtain sufficient representational power to capture underlying relation between two actors, both appearance features and position information need to be considered. Moreover, we note that appearance relation and position relation have different semantic attributes. To this end, we model the appearance relation and position relation in a separate and explicit way. The relation value is defined as a composite function below:

$$\mathbf{G}_{ij} = h\left(f_a(\mathbf{x}_i^a, \mathbf{x}_j^a), f_s(\mathbf{x}_i^s, \mathbf{x}_j^s)\right), \tag{1}$$

where $f_a(\mathbf{x}_i^a, \mathbf{x}_j^a)$ denotes the appearance relation between two actors, and the position relation is computed by $f_s(\mathbf{x}_i^s, \mathbf{x}_j^s)$. The function h fuses appearance and position relation to a scalar weight.

In our experiments, we adopt the following function to compute relation value:

$$\mathbf{G}_{ij} = \frac{f_s(\mathbf{x}_i^s, \mathbf{x}_j^s) \exp\left(f_a(\mathbf{x}_i^a, \mathbf{x}_j^a)\right)}{\sum_{j=1}^{N} f_s(\mathbf{x}_i^s, \mathbf{x}_j^s) \exp\left(f_a(\mathbf{x}_i^a, \mathbf{x}_j^a)\right)}, \tag{2}$$

where we perform normalization on each actor node using softmax function so that the sum of all the relation values of one actor node i will be 1.

Appearance relation. Here we discuss different choices for computing appearance relation value between actors:

(1) *Dot-Product*: The dot-product similarity of appearance features can be considered as a simple form of relation value. It is computed as:

$$f_a(\mathbf{x}_i^a, \mathbf{x}_j^a) = \frac{(\mathbf{x}_i^a)^{\mathrm{T}} \mathbf{x}_j^a}{\sqrt{d}}, \tag{3}$$

where $\sqrt{d}$ acts as a normalization factor.

(2) *Embedded Dot-Product*: Inspired by the Scaled Dot-Product Attention mechanism [54], we can extend the dot-product operation to compute similarity in an embedding space, and the corresponding function can be expressed as:

$$f_a(\mathbf{x}_i^a, \mathbf{x}_j^a) = \frac{\theta(\mathbf{x}_i^a)^{\mathrm{T}} \phi(\mathbf{x}_j^a)}{\sqrt{d_k}}, \tag{4}$$

where $\theta(\mathbf{x}_i^a) = \mathbf{W}_\theta \mathbf{x}_i^a + \mathbf{b}_\theta$ and $\phi(\mathbf{x}_j^a) = \mathbf{W}_\phi \mathbf{x}_j^a + \mathbf{b}_\phi$ are two learnable linear transformations. $\mathbf{W}_\theta \in \mathbb{R}^{d_k \times d}$ and $\mathbf{W}_\phi \in \mathbb{R}^{d_k \times d}$ are weight matrices, $\mathbf{b}_\theta \in \mathbb{R}^{d_k}$ and $\mathbf{b}_\phi \in \mathbb{R}^{d_k}$ are weight vectors. By learnable transformations of original features, we can learn the relation value between two actors in a subspace.

(3) *Relation Network*: We also evaluate the Relation Network module proposed in [44]. It can be written as:

$$f_a(\mathbf{x}_i^a, \mathbf{x}_j^a) = \mathrm{ReLU}\left(\mathbf{W}[\theta(\mathbf{x}_i^a), \phi(\mathbf{x}_j^a)] + \mathbf{b}\right), \tag{5}$$

where $[\cdot, \cdot]$ is the concatenation operation and $\mathbf{W}$ and $\mathbf{b}$ are learnable weights that project the concatenated vector to a scalar, followed by a ReLU non-linearity.

Position relation. In order to add spatial structural information to actor graph, the position relation between actors needs to be considered. To this end, we investigate two approaches to use spatial features in our work:

(1) *Distance Mask*: Generally, signals from local entities are more important than the signals from distant entities. And the relation information in the local scope has more significance than global relation for modeling the group activity. Based on these observations, we can set $\mathbf{G}_{ij}$ as zero for two actors whose distance is above a certain threshold. We call the resulted ARG as *localized ARG*. The f_s is formed as:

$$f_s(\mathbf{x}_i^s, \mathbf{x}_j^s) = \mathbb{I}\left(d(\mathbf{x}_i^s, \mathbf{x}_j^s) \leq \mu\right), \tag{6}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $d(\mathbf{x}_i^s, \mathbf{x}_j^s)$ denotes the Euclidean distance between center points of two actors' bounding boxes, and μ acts as a distance threshold which is a hyper-parameter.

(2) *Distance Encoding*: Alternatively, we can use the recent approaches [54] for learning position relation. Specifically, the position relation value is computed as

$$f_s(\mathbf{x}_i^s, \mathbf{x}_j^s) = \mathrm{ReLU}\left(\mathbf{W}_s \mathcal{E}(\mathbf{x}_i^s, \mathbf{x}_j^s) + \mathbf{b}_s\right), \tag{7}$$

the relative distance between two actors is embedded to a high-dimensional representation by $\mathcal{E}$, using cosine and sine functions of different wavelengths. The feature dimension after embedding is d_s. We then transform the embedded feature into a scalar by weight vectors $\mathbf{W}_s$ and $\mathbf{b}_s$, followed by a ReLU activation.

Multiple graphs. A single ARG $\mathbf{G}$ typically focuses on a specific relation signal between actors, therefore discarding a considerable amount of context information. In order

to capture diverse types of relation signals, we can extend the single actor relation graph into multiple graphs. That is, we build a group of graphs $\mathcal{G} = (\mathbf{G}^1, \mathbf{G}^2, \cdots, \mathbf{G}^{N_g})$ on a same actors set,where N_g is the number of graphs. Every graph $\mathbf{G}^i$ is computed in the same way according to Eq. (2), but with unshared weights. Building multiple relation graphs allows the model to jointly attend to different types of relation between actors. Hence, the model can make more robust relational reasoning upon the graphs.

Temporal modeling. Temporal context information is a crucial cue for activity recognition. Different from prior works, which employ Recurrent Neural Network to aggregate temporal information on dense frames, our model merges the information in the temporal domain via a sparse temporal sampling strategy [58]. During training, we randomly sample a set of $K = 3$ frames from the entire video, and build temporal graphs upon the actors in these frames. We call the resulted ARG as *randomized ARG*. At testing time, we can use a sliding window approach, and the activity scores from all windows are mean-pooled to form global activity prediction.

Empirically we find that sparsely sampling frames when training yields significant improvements on recognition accuracy. A key reason is that, existing group activity recognition datasets (e.g., Collective Activity dataset and Volleyball dataset) remain limited, in both size and diversity. Therefore, randomly sampling the video frames results in more diversity during training and reduces the risk of overfitting. Moreover, this sparse sampling strategy preserves temporal information with dramatically lower cost, thus enabling end-to-end learning under a reasonable budget in both time and computing resources.

3.3. Reasoning and Training on Graphs

Once the ARGs are built, we can perform relational reasoning on them for recognizing individual actions and group activity. We first review a graph reasoning module, called Graph Convolutional Network (GCN) [29]. GCN takes a graph as input, performs computations over the structure, and returns a graph as output, which can be considered as a "graph-to-graph" block. For a target node i in the graph, it aggregates features from all neighbor nodes according to the edge weight between them. Formally, one layer of GCN can be written as:

$$\mathbf{Z}^{(l+1)} = \sigma\left(\mathbf{G}\mathbf{Z}^{(l)}\mathbf{W}^{(l)}\right), \tag{8}$$

where $\mathbf{G} \in \mathbb{R}^{N\times N}$ is the matrix representation of the graph. $\mathbf{Z}^{(l)} \in \mathbb{R}^{N\times d}$ is the feature representations of nodes in the l^{th} layer, and $\mathbf{Z}^{(0)} = \mathbf{X}$. $\mathbf{W}^{(l)} \in \mathbb{R}^{d\times d}$ is the layer-specific learnable weight matrix. $\sigma(\cdot)$ denotes an activation function, and we adopt ReLU in this work. This layer-wise propagation can be stacked into multi-layers. For simplicity, we only use a layer of GCN in this work.

The original GCN operates on a single graph structure. After GCN, the way to fuse a group of graphs together remains an open question. In this work, we employ the late fusion scheme, namely fuse the features of same actor in different graphs after GCN:

$$\mathbf{Z}^{(l+1)} = \sum_{i=1}^{N_g} \sigma\left(\mathbf{G}^i\mathbf{Z}^{(l)}\mathbf{W}^{(l,i)}\right), \tag{9}$$

where we employ element-wise sum as a fusion function. We also evaluate concatenation as fusion function. Alternatively, a group of graphs can also be fused by early fusion, that is, fused via summation to one graph before GCN. We compare different methods of fusing a group of graphs in our experiments.

Finally the output relational features from GCN are fused with original features via summation to form the scene representation. As illustrated in Figure 2, the scene representation is fed to two classifiers to generate individual actions and group activity predictions.

The whole model can be trained in an end-to-end manner with backpropagation. Combining with standard cross-entropy loss, the final loss function is formed as

$$\mathcal{L} = \mathcal{L}_1(y^G, \hat{y}^G) + \lambda\mathcal{L}_2(y^I, \hat{y}^I), \tag{10}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the cross-entropy loss, y^G and y^I denote the ground-truth labels of group activity and individual action, $\hat{y}^G$ and $\hat{y}^I$ are the predictions to group activity and individual action. The first term corresponds to group activity classification loss, and the second is the loss of the individual action classification. The weight λ is used to balance these two tasks.

4. Experiments

In this section, we first introduce two widely-adopted datasets and the implementation details of our approach. Then, we perform a number of ablation studies to understand the effects of proposed components in our model. We also compare the performance of our model with the state of the art methods. Finally, we visualize our learned actor relation graphs and features.

4.1. Datasets and Implementation Details

Datasets. We conduct experiments on two publicly available group activity recognition datasets, namely the Volleyball dataset and the Collective Activity dataset.

The Volleyball dataset [25] is composed of 4830 clips gathered from 55 volleyball games, with 3493 training clips and 1337 for testing. Each clip is labeled with one of 8 group activity labels (right set, right spike, right pass, right winpoint, left set, left spike, left pass and left winpoint). Only the middle frame of each clip is annotated with the

Method	Accuracy
base model	89.8%
dot-product	**91.3%**
embedded dot-product	**91.3%**
relation network	90.7%

(a) Exploration of different appearance relation functions.

Method	Accuracy
no position relation	91.3%
distance mask	**91.6%**
distance encoding	91.5%

(b) Exploration of different position relation functions.

Number	1	4	8	16	32
Accuracy	91.6%	92.0%	92.0%	**92.1%**	92.0%

(c) Exploration of number of graphs.

Method	Accuracy
early fusion	90.8%
late fusion (summation)	**92.1%**
late fusion (concatenation)	91.9%

(d) Exploration of different methods for fusing multiple graphs.

Method	Accuracy
single frame	92.1%
TSN (3 frames)	92.3%
temporal-graphs (3 frames)	**92.5%**

(e) Exploration of temporal modeling methods.

Table 1: Ablation studies for group activity recognition accuracy on the Volleyball dataset.

players' bounding boxes and their individual actions from 9 personal action labels (waiting, setting, digging, failing, spiking, blocking, jumping, moving and standing). Following [24], we use 10 frames to train and test our model, which corresponds to 5 frames before the annotated frame and 4 frames after. To get the ground truth bounding boxes of unannotated frames, we use the tracklet data provided by [3].

The Collective Activity dataset [7] contains 44 short video sequences (about 2500 frames) from 5 group activities (crossing, waiting, queueing, walking and talking) and 6 individual actions (NA, crossing, waiting, queueing, walking and talking). The group activity label for a frame is defined by the activity in which most people participate. We follow the same evaluation scheme of [39] and select $1/3$ of the video sequences for testing and the rest for training.

Implementation details. We extract 1024-dimensional feature vector for each actor with ground-truth bounding boxes, using the methods mentioned in Section 3.1. During ablation studies, we adopt Inception-v3 as backbone network. We also experiment with VGG [48] network for fair comparison with prior methods. Due to memory limits, we train our model in two stages: first, we fine-tune the ImageNet pre-trained model on single frame randomly selected from each video without using GCN. We refer to the fine-tuned model described above as our base model throughout experiments. The base model performs group activity and individual action classification on original features of actors without relational reasoning. Then we fix weights of the feature extraction part of network, and further train the network with GCN.

We adopt stochastic gradient descent with ADAM to learn the network parameters with fixed hyper-parameters to $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$. For the Volleyball dataset, we train the network in 150 epochs using mini-batch size of 32 and a learning rate ranging from 0.0002 to 0.00001. For the Collective Activity dataset, we use mini-batch size of 16 with a learning rate of 0.0001, and train the network in 80 epochs. The individual action loss weight $\lambda = 1$ is used. Besides, the parameters of the GCN are set as $d_k = 256, d_s = 32$, and we adopt the $1/5$ of the image width to be the distance mask threshold μ.

Our implementation is based on PyTorch deep learning framework. The running time for inferring a video is approximately 0.2s on a single TITAN-XP GPU.

4.2. Ablation Studies

In this subsection, we perform detailed ablation studies on the Volleyball dataset to understand the contributions of the proposed model components to relation modeling using group activity recognition accuracy as evaluation metric. The results are shown in Table 1.

Appearance relation. We begin our experiments by studying the effect of modeling the appearance relation between actors and different functions to compute appearance relation value. Based on single frame, we build single ARG without using position relation. The results are listed in Table 1a. We first observe that explicitly modeling the relation between actors brings significant performance improvement. All models with GCN outperform the base model. Then it is shown that the dot-product and embedded dot-product yield same recognition accuracy of 91.3%, and perform better than the relation network. We conjecture that dot-product operation is more stable for representing relation information. In the following experiments, embedded dot-product is used to compute appearance relation value.

Position relation. We further add spatial structural information to ARG. In Section 3.2, we present two methods to use spatial features: distance mask and distance encoding. Results on comparing the performance of these two methods are reported in Table 1b. We can see that these two

methods both obtain better performance than those without using spatial features, demonstrating the effectiveness of modeling position relation. And the distance mask yields slightly better accuracy than distance encoding. In the rest of the paper, we choose distance mask to represent position relation.

Multiple graphs. We also investigate the effectiveness of building a group of graphs to capture different kinds of relation information. First, we compare the performance of using different number of graphs. As shown in Table 1c, we observe that building multiple graphs leads to consistent and significant gain compared with only building single graph, and is able to further boost accuracy from 91.6% to 92.1%. Then we evaluate three methods to fuse a group of graphs: (1) early fusion, (2) late fusion via summation, (3) late fusion via concatenation. The results of experiments using 16 graphs are summarized in Table 1d. We see that the late fusion via summation achieves the best performance. We note that the early fusion scheme, which aggregates a group of graphs by summation before GCN, results in the performance drops dramatically. This observation indicates that the relation values learned by different graphs encode different semantic information and will cause confusion for relational reasoning if they are fused before graph convolution. We adopt $N_g = 16$ and late fusion via summation in the following experiments.

Temporal modeling. With all the design choices set, we now extend our model to temporal domain. As mentioned in Section 3.2, we employ sparse temporal sampling strategy [58], and uniformly sample a set of $K = 3$ frames from the entire video during training. In the simplest setting, we can handle the input frames separately, then fuse the prediction scores of different frames as Temporal Segment Network (TSN) [58]. Alternatively, we can build temporal graphs upon the actors in input frames and fuse temporal information by GCN. We report the accuracies of these two temporal modeling methods in Table 1e. We see that TSN modeling is helpful to improve the performance of our model. Moreover, building temporal graphs further boosts accuracy to 92.5%, which demonstrates that temporal reasoning helps to differentiate between group activity categories.

4.3. Comparison with the State of the Art

Now, we compare our best models with the state-of-the-art methods in Table 2. For fair comparison with prior methods, we report our results with both Inception-v3 and VGG backbone network. Meanwhile, we perform proposal-based experiment. We train a Faster-RCNN [42] with training data. Using the bounding boxes from Faster-RCNN at testing time, our model can still achieve promising accuracy.

Table 2a shows the comparison with previous results on the Volleyball dataset for group activity and individual action recognition. Our method surpasses all the existing methods by a good margin, establishing the new state-of-the-art. Our model with Inception-v3 utilizes the same feature extraction strategy as [3], and outperforms it by about 2% on group activity recognition accuracy, since our model can capture and exploit the relation information among actors. And, we also achieve better performance on individual action recognition task. Meanwhile, our method outperforms the recent methods using hierarchical relational networks [23] or semantic RNN [39], mostly because we explicitly model the appearance and position relation graph, and adopt more efficient temporal modeling method.

Method	Backbone	Group activity	Individual action
HDTM [24]	AlexNet	81.9%	-
CERN [45]	VGG16	83.3%	-
stagNet (GT) [39]	VGG16	89.3%	-
stagNet (PRO) [39]	VGG16	87.6%	-
HRN [23]	VGG19	89.5%	-
SSU (GT) [3]	Inception-v3	90.6%	81.8%
SSU (PRO) [3]	Inception-v3	86.2%	77.4%
OURS (GT)	Inception-v3	92.5%	83.0%
OURS (PRO)	Inception-v3	91.5%	-
OURS (GT)	VGG16	91.9%	**83.1%**
OURS (GT)	VGG19	**92.6%**	82.6%

(a) Comparison with state of the art on the Volleyball dataset.

Method	Backbone	Group activity
SIM [12]	AlexNet	81.2%
HDTM [24]	AlexNet	81.5%
Cardinality Kernel [17]	None	83.4%
SBGAR [32]	Inception-v3	86.1%
CERN [45]	VGG16	87.2%
stagNet (GT) [39]	VGG16	89.1%
stagNet (PRO) [39]	VGG16	87.9%
OURS (GT)	Inception-v3	**91.0%**
OURS (PRO)	Inception-v3	90.2%
OURS (GT)	VGG16	90.1%

(b) Comparison with state of the art on the Collective dataset.

Table 2: Comparison with state of the art methods. GT and PRO indicate using ground-truth and proposal-based bounding boxes, respectively.

We further evaluate the proposed model on the Collective Activity dataset. The results and comparison with previous methods are listed in Table 2b. Our temporal multiple graphs model again achieves the state-of-the-art performance with 91.0% group activity recognition accuracy. This outstanding performance shows the effectiveness and generality of proposed ARG for capturing the relation information in multiple people scene.

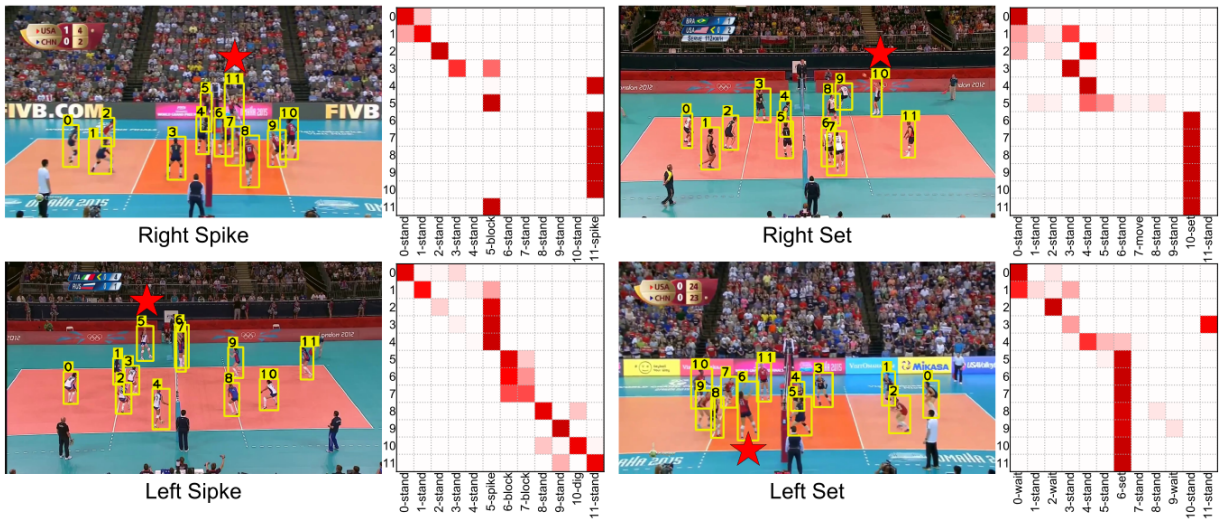

Figure 3: Visualization of learned actor relation graphs. Each row shows two examples. For each example, we plot: (1) input frame with group-truth bounding boxes and group activity label; (2) matrix $\mathbf{G}$ of learned relation graph with ground-truth individual action labels. The actor who has max column sum of $\mathbf{G}$ in each frame is denoted with red star.

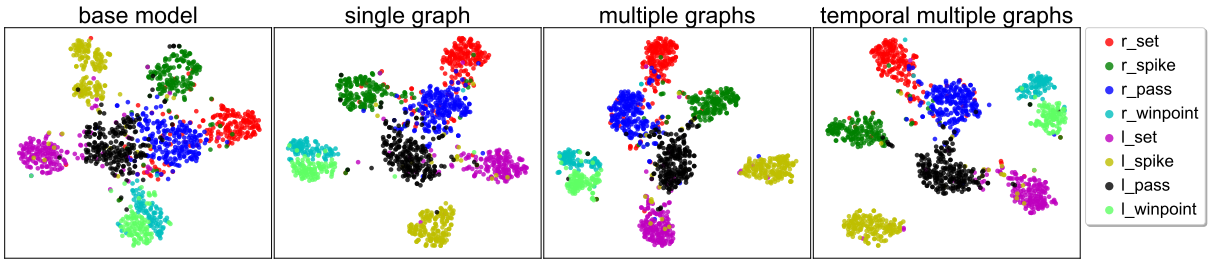

Figure 4: t-SNE [53] visualization of embedding of video representation on the Volleyball dataset learned by different model variants: base model, single graph, multiple graphs, temporal multiple graphs. Each video is visualized as one point and colors denote different group activities (better view in color version).

4.4. Model Visualization

Actor relation graph visualization We visualize several examples of the relation graph generated by our model in Figure 3. We use the single graph model on single frame, because it is easier to visualize. Visualization results facilitate us understanding how ARG works. We can see that our model is able to capture relation information for group activity recognition, and the generated ARG can automatically discover the key actor to determine the group activity in the scene.

t-SNE visualization of the learned representation. Figure 4 shows the t-SNE [53] visualization for embedding the video representation learned by different model variants. Specifically, we project the representations of videos on the validation set of Volleyball dataset into 2-dimensional space using t-SNE. We can observe that the scene-level representations learned by using ARG are better separated. Moreover, building multiple graphs and aggregating temporal information lead to better differentiate group activities. These visualization results indicate our ARG models are more effective for group activity recognition.

5. Conclusion

This paper has presented a flexible and efficient approach to determine relevant relation between actors in a multi-person scene. We learn *Actor Relation Graph* (ARG) to perform relational reasoning on graphs for group activity recognition. We also evaluate the proposed model on two datasets and establish new state-of-the-art results. The comprehensive ablation experiments and visualization results show that our model is able to learn relation information for understanding group activity. In the future, we plan to further understand how ARG works, and incorporate more global scene information for group activity recognition.

Acknowledgement

This work is supported by the National Science Foundation of China under Grant No.61321491, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Mohamed Rabie Amer, Peng Lei, and Sinisa Todorovic. Hirf: Hierarchical random field for collective activity recognition in videos. In *ECCV*, pages 572–585, 2014. 1, 2

[2] Mohamed R. Amer, Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song Chun Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *ECCV*, pages 187–200, 2012. 2

[3] Timur M. Bagautdinov, Alexandre Alahi, François Fleuret, Pascal Fua, and Silvio Savarese. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In *CVPR*, pages 3425–3434, 2017. 1, 2, 3, 6, 7

[4] Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *NIPS*, pages 4502–4510, 2016. 3

[5] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *WACV*, pages 381–389, 2018. 2

[6] Wongun Choi and Silvio Savarese. Understanding collective activitiesof people from videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(6):1242–1257, 2014. 2

[7] Wongun Choi, Khuram Shahid, and Silvio Savarese. What are they doing? : Collective activity classification using spatio-temporal relationship among people. In *ICCV Workshops*, pages 1282–1289, 2009. 1, 2, 6

[8] Wongun Choi, Khuram Shahid, and Silvio Savarese. Learning context for collective activity recognition. In *CVPR*, pages 3273–3280, 2011. 2

[9] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *CVPR*, pages 3298–3308, 2017. 2

[10] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *ICML*, pages 2702–2711, 2016. 3

[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3837–3845, 2016. 3

[12] Zhiwei Deng, Arash Vahdat, Hexiang Hu, and Greg Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *CVPR*, pages 4772–4781, 2016. 1, 2, 7

[13] David K. Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*, pages 2224–2232, 2015. 3

[14] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alex G Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, pages 2568–2577, 2015. 1

[15] Georgia Gkioxari, Ross B. Girshick, and Jitendra Malik. Actions and attributes from wholes and parts. In *ICCV*, pages 2470–2478, 2015. 2

[16] Abhinav Gupta, Aniruddha Kembhavi, and Larry S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1775–1789, 2009. 2

[17] Hossein Hajimirsadeghi, Wang Yan, Arash Vahdat, and Greg Mori. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In *CVPR*, pages 2596–2605, 2015. 2, 7

[18] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1025–1035, 2017. 3

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 3

[20] Yedid Hoshen. VAIN: attentional multi-agent predictive modeling. In *NIPS*, pages 2698–2708, 2017. 3

[21] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, pages 3588–3597, 2018. 2

[22] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, pages 4418–4427, 2017. 2

[23] Mostafa S. Ibrahim and Greg Mori. Hierarchical relational networks for group activity recognition and retrieval. In *ECCV*, pages 742–758, 2018. 1, 2, 7

[24] Mostafa S. Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *CVPR*, pages 1971–1980, 2016. 1, 2, 6, 7

[25] Mostafa S. Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *CVPR*, pages 1971–1980, 2016. 2, 5

[26] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015. 2

[27] Steven M. Kearnes, Kevin McCloskey, Marc Berndl, Vijay S. Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):595–608, 2016. 3

[28] Thomas N. Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard S. Zemel. Neural relational inference for interacting systems. In *ICML*, pages 2693–2702, 2018. 3

[29] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. 2, 3, 4, 5

[30] Tian Lan, Leonid Sigal, and Greg Mori. Social roles in hierarchical models for human activity recognition. In *CVPR*, pages 1354–1361, 2012. 1, 2

[31] Tian Lan, Yang Wang, Weilong Yang, Stephen N. Robinovitch, and Greg Mori. Discriminative latent models for recognizing contextual group activities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(8):1549–1562, 2012. 2

[32] Xin Li and Mooi Choo Chuah. SBGAR: semantics based group activity recognition. In *ICCV*, pages 2895–2904, 2017. 1, 2, 7

[33] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. In *ECCV*, pages 346–363, 2018. 2

[34] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and caption regions. *CoRR*, abs/1707.09700, 2017. 2

[35] Xiaodan Liang, Lisa Lee, and Eric P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*, pages 4408–4417, 2017. 2

[36] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding. In *CVPR*, pages 6790–6800, 2018. 2

[37] Bingbing Ni, Xiaokang Yang, and Shenghua Gao. Progressively parsing interactional objects for fine grained action detection. In *CVPR*, pages 1020–1028, 2016. 2

[38] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023, 2016. 3

[39] Mengshi Qi, Jie Qin, Annan Li, Yunhong Wang, Jiebo Luo, and Luc Van Gool. stagnet: An attentive semantic RNN for group activity recognition. In *ECCV*, pages 104–120, 2018. 1, 2, 6, 7

[40] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, pages 407–423, 2018. 2

[41] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander N. Gorban, Kevin Murphy, and Li Fei-Fei. Detecting events and key actors in multi-person videos. In *CVPR*, pages 3043–3053, 2016. 2

[42] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 7

[43] Michael S. Ryoo and Jake K. Aggarwal. Stochastic representation and recognition of high-level group activities: Describing structural uncertainties in human activities. In *CVPR Workshops*, page 11, 2009. 2

[44] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, pages 4974–4983, 2017. 2, 4

[45] Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. CERN: confidence-energy recurrent network for group activity recognition. In *CVPR*, pages 4255–4263, 2017. 1, 2, 7

[46] Tianmin Shu, Dan Xie, Brandon Rothrock, Sinisa Todorovic, and Song-Chun Zhu. Joint inference of groups, events and human roles in aerial videos. In *CVPR*, pages 4576–4584, 2015. 2

[47] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 1

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[49] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *NIPS*, pages 2244–2252, 2016. 3

[50] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, pages 335–351, 2018. 2

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 3

[52] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1

[53] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2):2579–2605, 2008. 8

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 6000–6010, 2017. 4

[55] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2017. 3

[56] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, pages 1430–1439, 2018. 1

[57] Limin Wang, Yu Qiao, and Xiaoou Tang. MoFAP: A multi-level representation for action recognition. *International Journal of Computer Vision*, 119(3):254–271, 2016. 2

[58] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016. 3, 5, 7

[59] Minsi Wang, Bingbing Ni, and Xiaokang Yang. Recurrent modeling of interaction context for collective activity recognition. In *CVPR*, pages 7408–7416, 2017. 1, 2

[60] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 2

[61] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, pages 413–431, 2018. 3

[62] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. In *NIPS*, pages 558–568, 2018. 2

[63] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 3097–3106, 2017. 2

[64] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, pages 7444–7452, 2018. 3

[65] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. In *ECCV*, pages 690–706, 2018. 2

[66] Bangpeng Yao and Fei-Fei Li. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, pages 17–24, 2010. 2

[67] Bangpeng Yao and Fei-Fei Li. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1691–1703, 2012. 2

[68] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, pages 3107–3115, 2017. 2