Progressive Relation Learning for Group Activity Recognition

Guyue Hu[1,3*] Bo Cui[1,3] Yuan He[1,3] Shan Yu[1,2,3]
[1]Brainnetome Center, National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences (CASIA)
[2]CAS Center for Excellence in Brain Science and Intelligence Technology
[3]University of Chinese Academy of Sciences
{guyue.hu, bo.cui, yuan.he, shan.yu}@nlpr.ia.ac.cn

Abstract

Group activities usually involve spatiotemporal dynamics among many interactive individuals, while only a few participants at several key frames essentially define the activity. Therefore, effectively modeling the group-relevant and suppressing the irrelevant actions (and interactions) are vital for group activity recognition. In this paper, we propose a novel method based on deep reinforcement learning to progressively refine the low-level features and high-level relations of group activities. Firstly, we construct a semantic relation graph (SRG) to explicitly model the relations among persons. Then, two agents adopting policy according to two Markov decision processes are applied to progressively refine the SRG. Specifically, one feature-distilling (FD) agent in the discrete action space refines the low-level spatiotemporal features by distilling the most informative frames. Another relation-gating (RG) agent in continuous action space adjusts the high-level semantic graph to pay more attention to group-relevant relations. The SRG, FD agent, and RG agent are optimized alternately to mutually boost the performance of each other. Extensive experiments on two widely used benchmarks demonstrate the effectiveness and superiority of the proposed approach.

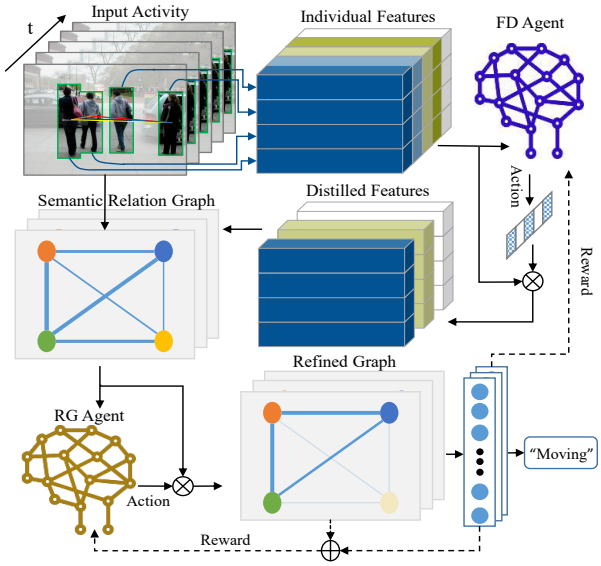


Figure 1: The overview of proposed method. A feature-distilling (FD) agent progressively selects the most informative frames of the low-level spatiotemporal individual features. A relation-gating (RG) agent further progressively refines the high-level semantic relation graph (SRG) to discover group-relevant relations.

1. Introduction

Group activity recognition, which refers to discern the activities involving a large number of interactive individuals, has attracted growing interests in the communities of computer vision [9, 32, 31, 36, 22]. Unlike conventional video action recognition that only concentrates on the spatiotemporal dynamics of one or two persons, group activity recognition further requires understanding the group-relevant interactions among many individuals.

In the past a few years, a series of approaches combine the hand-crafted feature with probability graph [7, 17, 26]. Recently, the LSTM, strucural RNNs and message passing neural network (MPNN) are also applied to model the interactions among persons, subgroups and groups [22, 32, 4]. The interaction relations in these methods are *implicitly* contained in the ordered RNNs or the passing messages of MPNN. Moreover, not all the existing relations are relevant to the group activity and the pairwise relations may contain many edges that are coupled from spurious noise, such as cluttered background, inaccurate human detection, and interaction between outlier persons (*e.g.*, the "Waiting" per-

*Corresponding Author

son in Fig. 1). Due to the relations in previous methods are modeled *implicitly*, it is unable to determine whether one specific relation is group-relevant or not.

In addition, although a large number of persons may involve in a group activity, usually only a few actions or interactions in several key frames essentially define the group activity. Yan *et al*. [36] heuristically defined the key participants as the ones with "long motion" and "flash motion". Qi *et al*. [22] applied a "self-attention" mechanism to attend to important persons and key frames. Nevertheless, these methods are limited to the coarse individual (person) level, and have not dug into the fine-grained relation level to consider which relations are vital (*e.g*., regulating 15 pairwise relations is more fine-grained than attending 6 persons).

To move beyond such limitations, we propose a progressive relation learning framework to effectively model and distill the group-relevant actions and interactions in group activities. Firstly, we build a graph to explicitly model the semantic relations in group activities. Then, as illustrated in Fig. 1, two agents progressively refine the low-level spatiotemporal features and high-level semantic relations of group activities. Specifically, at the feature level, a feature-distilling agent explores a policy to distill the most informative frames of low-level spatiotemporal features. At the relation level, a relation-gating agent further refines the high-level relation graph to focus on the group-relevant relations.

In summary, the contributions of this paper can be summarized as: (1) A novel progressive relation learning framework is proposed for group activity analysis. (2) Beyond distilling group-relevant information at the course individual (person) level, we proposed a RG agent to progressively discover group-relevant semantic relations at the fine-grained relation level. (3) A FD agent is proposed to further progressively filter the frames of low-level spatiotemporal features that used for constructing the high-level semantic relation graph.

2. Related Works

Reinforcement Learning. Reinforcement learning (RL) has benefited many fields of computer vision, such as image cropping [18] and visual semantic navigation [38]. Regarding the optimization policy, RL can be categorized into the value-based methods, policy-based methods, and their hybrids. The value-based methods (*e.g*., deep Q-learning [21]) are good at solving the problems in low dimensional discrete action space, but they fail in high dimensional continuous space. Although the policy-based methods (*e.g*., policy gradient [29]) are capable to deal with the problems in continuous space, they suffer from high variance of gradient estimation. The hybrid methods, such as Actor-Critic algorithms [16], combine their advantages and are capable for both of discrete and continuous action spaces. Moreover, by exploiting asynchronous updating, the Asynchronous Advantage Actor-Critic (A3C) algorithm [20] has largely improved the training efficiency. Therefore, we adopt the A3C algorithm to optimize both of our RG agent in continuous action space and our FD agent in discrete action space.

Graph Neural Network. Due to the advantages of representing and reasoning over structured data, the graph neural network (GNN) has attracted increasing attention [35, 34, 13, 12, 3]. Graph convolutional network (GCN) generalizes CNN on graph, which therefore can deal with non-Euclidean data [5]. It has been widely applied in computer vision, *e.g*., point cloud classification [27], action recognition [37], and traffic forecasting [39]. Another class of GNN combines graph with RNN, in which each node captures the semantic relation and structured information from its neighbors through multiple iterations of passing and updating, *e.g*., message-passing neural network [10], graph network block [24]. Each relation in the former class (*i.e*., GCN) is represented by a scalar in its adjacency matrix that is not adequate for modeling the complex context information in group activity. Therefore, our semantic relation graph is built under the umbrella of the latter class that each relation is explicitly represented by a learnable vector.

3. Method

3.1. Individual Feature Extraction

Following [36], the person bounding boxes are firstly obtained through the object tracker in the Dlib library [15]. As shown in Fig. 2, the visual feature (*e.g*., appearance and pose) $x_{p_i}^{vis}$ of each person i is extracted through a convolutional neural network (called Person-CNN). Then, the spatial visual feature is fed into a long short-term memory network (called Person-LSTM) to model the individual temporal dynamic $x_{p_i}^{tem}$. Finally, we concatenate the stacked visual features $\boldsymbol{x}_p^{vis}$ and temporal dynamics $\boldsymbol{x}_p^{tem}$ of all persons as the basic spatiotemporal features, *i.e*., $\boldsymbol{x}_p = [\boldsymbol{x}_p^{vis}, \boldsymbol{x}_p^{tem}]$. These basic representations contain no context information, such as the person to person, person to group, and group to group interactions. Besides, the spatial distance vectors $\{|dx|, |dy|, |dx + dy|, \sqrt{(dx)^2 + (dy)^2}\}$ and direction vectors $\{arctan(dy, dx), arctan2(dy, dx)\}$ between each pair of persons are concatenated as the original interaction features $\boldsymbol{x_e}$, where dx and dy are the displacements along horizontal and vertical axes, respectively.

3.2. Semantic Relation Graph

Inferring semantic relations over inherent structure in a scene is helpful to suppress noises, such as inaccurate human detection, mistaken action recognition, and outlier people not involved in a particular group activity. To achieve it, we explicitly model the structured relations through a graph network [24]. Let us put aside the two agents in Fig. 2 and explain how to build the baseline semantic re-

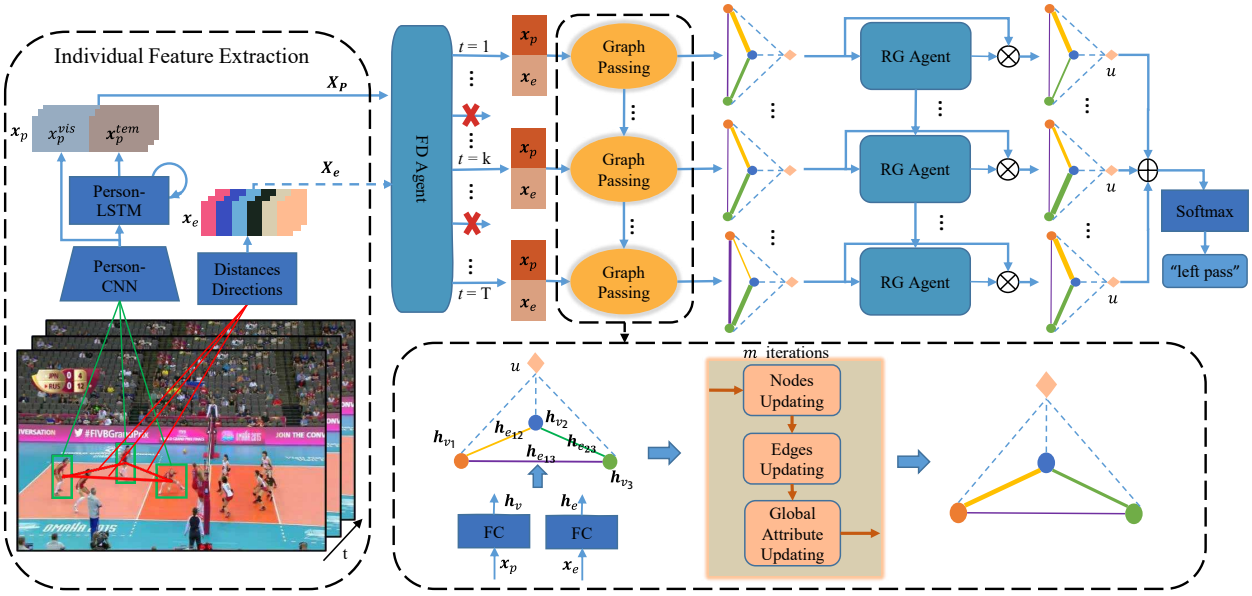


Figure 2: The detailed framework of our method. The low-level spatiotemporal features of persons are extracted by a CNN and a LSTM. The feature-distilling (FD) agent selects the informative frames of features. Then the distilled features are used to build a high-level semantic relation graph (SRG), and a relation-gating (RG) agent further refines the SRG. "FC" denotes fully connected layer. Finally, the activity category is predicted according to the sum of global attributes at all the times.

lation graph first. Let a graph $\boldsymbol{G} = (\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{u}$ is the global attribute (*i.e.*, activity score), $\boldsymbol{V} = \{v_i\}_{i=1}^{N_v}$ and $\boldsymbol{E} = \{e_{ij}\}_{i,j=1}^{N_v}$ are respectively the person nodes and the relation edges among them. The attributes of person nodes $\boldsymbol{H_v}$ and the attributes of relation edges $\boldsymbol{H_e}$ are respectively initialized with the embeddings of low-level spatiotemporal features $\boldsymbol{X_p}$ and original interaction features $\boldsymbol{X_e}$.

During graph passing, each node v_i collects the contextual information $\boldsymbol{h_{ve}^{ij}}$ from each of its neighbors v_j ($j \in \mathcal{N}(v_i)$) via a collecting function ϕ_{ve}, and aggregates all collected information via an aggregating function ψ_v, *i.e.*,

$$\boldsymbol{h_{ve}^{ij}} = \phi_{ve}(\boldsymbol{h_{e_{ij}}}, \boldsymbol{h_{v_j}}) = \text{NN}_{ve}\left([\boldsymbol{h_{e_{ij}}}, \boldsymbol{h_{v_j}}]\right) \quad (1)$$

$$\overline{\boldsymbol{h}}_{\boldsymbol{e_i}} = \psi_v(\boldsymbol{h_{ve}^{i}}) = \sum_{j \in \mathcal{N}(v_i)} \boldsymbol{h_{ve}^{ij}} \quad (2)$$

where the collecting function ϕ_{ve} is implemented by a neural network NN_{ve}, and $[\cdot]$ denotes concatenation. Then, the aggregated contextual information $\overline{\boldsymbol{h}}_{\boldsymbol{e_i}}$ updates the node attributes via a node updating function ϕ_v (network NN_v),

$$\boldsymbol{h'_{v_i}} = \phi_v(\overline{\boldsymbol{h}}_{\boldsymbol{e_i}}, \boldsymbol{h_{v_i}}) = \text{NN}_v\left([\overline{\boldsymbol{h}}_{\boldsymbol{e_i}}, \boldsymbol{h_{v_i}}]\right). \quad (3)$$

After that, each edge $\boldsymbol{h_{e_{ij}}}$ enrolls message from the sender $\boldsymbol{h'_{v_i}}$ and receiver $\boldsymbol{h'_{v_j}}$ to update its edge attributes via an edge updating function ϕ_e (network NN_e),

$$\hat{\boldsymbol{h}}_{\boldsymbol{e}_{ij}} = \phi_e(\boldsymbol{h_{v'_i}}, \boldsymbol{h_{v'_j}}, \boldsymbol{h_{e_{ij}}}) = \text{NN}_e\left([\boldsymbol{h_{v'_i}}, \boldsymbol{h_{v'_j}}, \boldsymbol{h_{e_{ij}}}]\right) \quad (4)$$

To simplify the problem, we consider the graph is undirected (*i.e.*, $\boldsymbol{h'}_{\boldsymbol{e_{ij}}} = \boldsymbol{h'}_{\boldsymbol{e_{ji}}} = (\hat{\boldsymbol{h}}_{\boldsymbol{e_{ij}}} + \hat{\boldsymbol{h}}_{\boldsymbol{e_{ji}}})/2$) and has no self-connection. Finally, the global attribute $\boldsymbol{u}$ is updated based on semantic relations in the whole relation graph, *i.e.*,

$$\boldsymbol{u'} = \boldsymbol{W_u}\left(\sum_{i=1}^{N_v}\sum_{j>i}^{N_v} \boldsymbol{h'_{e_{ij}}}\right) + \boldsymbol{b_u} \quad (5)$$

where $\boldsymbol{W_u}$ is parameter matrix and $\boldsymbol{b_u}$ is bias. The NN_e, NN_{ve} and NN_v are implemented with LSTM networks.

Since propagating information over the graph once captures at most pairwise relations, we update the graph for $\boldsymbol{m}$ iterations to encode high-order interactions. After the propagations, the graph automatically learns the high-level semantic relations from the low-level individual features in the scene. Finally, the activity score can be obtained by appending a softmax layer to the $\boldsymbol{u}$ after the last iteration.

3.3. Progressively Relation Gating

Although the above fully-connected semantic graph is capable of explicitly modeling any type of relation, it contains many group-irrelevant relations. Therefore, we introduce a relation-gating agent to explore an adaptive policy to select group-relevant relations. The decision process is formulated as a Markov Process $\mathcal{M} = \{S, A, \mathcal{T}, r, \gamma\}$.

States. The state S consists of three parts $S = \{S_g, S_l, S_u\}$. S_g is the whole semantic graph, represented by the stack of all relation triplets (*"sender", "relation",*

"receiver"), which provides the global information about the current scene. S_l is concatenation of the relation triplet $(\boldsymbol{h}_{\boldsymbol{v}_i}, \boldsymbol{h}_{\boldsymbol{e}_{ij}}, \boldsymbol{h}_{\boldsymbol{v}_j})$ corresponding to one specific relation $\boldsymbol{h}_{\boldsymbol{e}_{ij}}$ that will be refined, which provides the local information for the agent. $S_l \in \mathbb{R}^{D_v+D_e+D_v}$, where D_v and D_e denote the attribute dimensions of N_v nodes and N_e relations, respectively. $S_u = \boldsymbol{u}$ is global attributes of the relation graph at the current state, where $\boldsymbol{u}$ is the activity scores.

Action. Inspired by the information gates in the LSTMs, we introduce a gate g_{ij} for each relation edge. The action A of the agent is to generate the gate $g_{ij} \in [0, 1]$. Then, it is applied to adjust the corresponding relation at each reinforcement step, *i.e.*, $\boldsymbol{h}_{\boldsymbol{e}_{ij}} = g_{ij} \cdot \boldsymbol{h}_{\boldsymbol{e}_{ij}}$. Since the semantic relation graph is undirected, we normalize the values of gates before gating operation, *i.e.*, $g_{ij} = g_{ji} = (g_{ij} + g_{ji})/2$.

Reward. The reward $r(S, A)$, reflecting the efficacy of action A w.r.t the state S, consists of three parts. 1) To encourage the relation gates $\boldsymbol{G} = \{g_{ij}\}_{i,j=1}^{N_v}$ to selects group-relevant relations, we propose a structured sparsity reward. We define structured sparsity as the $L_{2,1}$ norm of $\boldsymbol{G}$, *i.e.*,

$$L_{2,1}(\boldsymbol{G}) = \sum_{i=1}^{N_v} \|\boldsymbol{g}_{i,:}\|_2 = \sum_{i=1}^{N_v} \left(\sqrt{\sum_{j=1}^{N_v} |g_{ij}|^2} \right) \tag{6}$$

where $\boldsymbol{g}_{i,:}$ is row vectors of $\boldsymbol{G}$. As illustrated in Fig. 3a, unlike L_1 norm that tends to uniformly make all gating elements sparse, the $L_{2,1}$ norm can encourage the rows of $\boldsymbol{G}$ to be sparse. Thus, the structured sparsity is very helpful to attend to a few key participants which have wide influence to others. The structured sparsity reward at the τth reinforcement step is defined to encourage the agent to gradually attend to a few key participants and relations, *i.e.*,

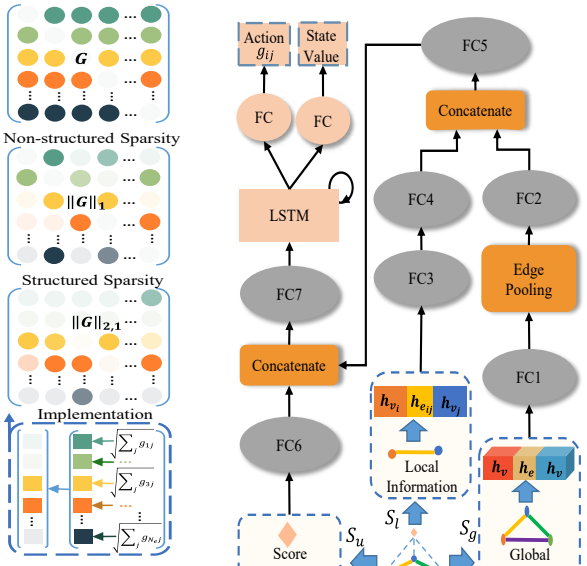
$$r_{sparse} = -sgn\left(L_{2,1}\left(\boldsymbol{G}_\tau\right) - L_{2,1}\left(\boldsymbol{G}_{\tau-1}\right)\right) \tag{7}$$

where $r_{sparse} \in \{-1, 1\}$ and the sgn is sign function. 2) To encourage the posterior probability to evolve along an ascending trajectory, we introduce an ascending reward with respect to the probability of groundtruth activity label, *i.e.*,

$$r_{ascend} = sgn\left(\boldsymbol{p}_\tau^c - \boldsymbol{p}_{\tau-1}^c\right) \tag{8}$$

where $\boldsymbol{p}_\tau^c$ is predicted probability of the groundtruth label at the τth step. $r_{ascend} \in \{-1, 1\}$ reflects the probability improvement of the groundtruth. 3) To ensure that the model tends to predict correct classes, inspired by [30], a strong stimulation Ω is enforced when the predicted class shifts from wrong to correct after a step, and a strong punishment $-\Omega$ is applied if the turning goes otherwise, *i.e.*,

$$r_s = \begin{cases} \Omega, & \text{if stimulation} \\ -\Omega, & \text{if punishment} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$



(a) Comparison of sparsity

(b) Structure of the RG agent

Figure 3: (a) Comparing the L_1 and $L_{2,1}$ norms of gating matrix $\boldsymbol{G}$, where the transparency denotes the value of each gate. The $\|\boldsymbol{G}\|_1$ encourages uniform sparsity while $\|\boldsymbol{G}\|_{2,1}$ encourages structured row sparsity. The implementation of $\|\boldsymbol{G}\|_{2,1}$ is illustrated in the bottom. (b) The RG agent takes in the global information S_g, the local information S_l for specific relation, and the global scene attribute S_u. "FC1", ..., "FC7" are fully connected layers, and "Edge Pooling" denotes average pooling along the edge dimension. Finally, the left branch (*Actor*) and the right branch (*Critic*) outputs an action and a value for the current state, respectively.

Finally, the total reward for the RG agent is

$$r = r_{sparse} + r_{ascend} + r_{shift}. \tag{10}$$

Relation-gating Agent. Since searching high dimensional continuous action space is challenging for reinforcement learning, we compromise to let the agent output one gating value at a time and cycle through all edges within each reinforcement step. The architecture of the RG agent is shown in Fig. 3b, which is under an Actor-Critic framework [16]. Inspired by human's decision making that historical experience can assist the current decision, a LSTM block is used to memorize the information of the past states. The agent maintains both a policy $\pi(A_\tau|S_\tau;\theta)$ (also named *Actor*) to generate actions (gates) and an estimation of value function $V(S_\tau;\theta_v)$ (also named *Critic*) to assess values for corresponding states. Specifically, the *Actor* outputs a mean μ_{ij} and a standard deviation σ_{ij} of action distribution $\mathcal{N}(\mu_{ij}, \sigma_{ij})$. The action g_{ij} is sampled from the Gaussian distribution $\mathcal{N}(\mu_{ij}, \sigma_{ij})$ during training, and is set as μ_{ij} directly during testing.

Optimization. The agent is optimized with the classical A3C algorithm [20] for reinforcement learning. The policy and the value function of the agent are updated after every τ_{max} (updating interval) steps or when a terminal state is reached. The accumulated reward at the step τ is $R_\tau = \sum_{i=0}^{k-1} \gamma^i r_{\tau+i} + \gamma^k V(S_{\tau+k}; \theta_v)$, where γ is the discount factor, r_τ is the reward at the τth step, and k varies from 0 to τ_{max}. The advantage function can be calculated by $R_\tau - V(S_\tau; \theta_v)$, and the entropy of policy π is $H(\pi(S_\tau; \theta))$. Eventually, the gradients are accumulated via Eq. 11 and Eq. 12 to respectively update the value function and the policy of agent [20].

$$d\theta_v \leftarrow d\theta_v + \nabla_{\theta_v} \left(R_\tau - V(S_\tau; \theta_v)\right)^2 / 2 \tag{11}$$

$$\begin{aligned} d\theta \leftarrow d\theta &+ \nabla_\theta log\pi(A_\tau | S_\tau; \theta) \left(R_\tau - V(S_\tau; \theta_v)\right) \\ &+ \beta \nabla_\theta H(\pi(S_\tau; \theta)) \end{aligned} \tag{12}$$

where β controls the strength of entropy regularization.

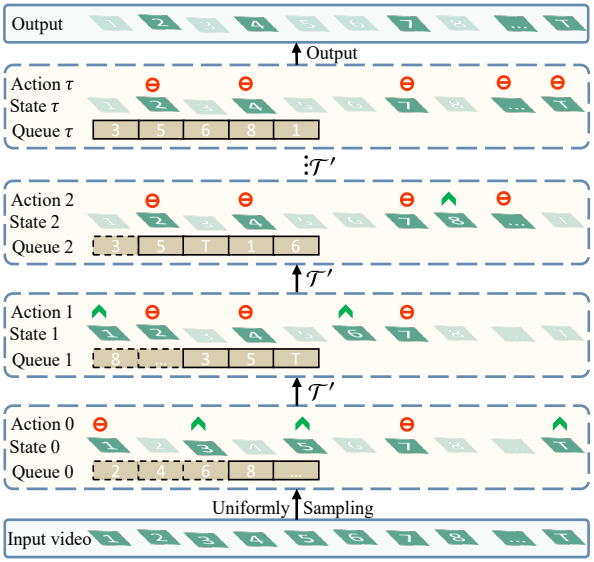
3.4. Progressively Feature Distilling

To further refine the low-level spatiotemporal features used for constructing graph, we introduced another feature-distilling agent. It is aimed at distilling the most informative frames of features, which is also formulated as a Markov Decision Process $\mathcal{M}' = \{S', A', \mathcal{T}', r', \gamma'\}$.

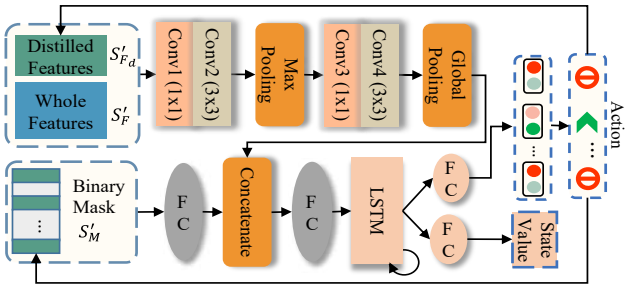
State. The state of the FD agent consists of three components $S' = \{S'_F, S'_{F_d}, S'_M\}$. The whole feature tensor of an activity $S'_F \in \mathbb{R}^{N \times T \times D_F}$ provides the global information about the activity clip, where N, T and D_F are respectively the numbers of person, frame and feature dimension of the feature tensor. The local feature $S'_{F_d} \in \mathbb{R}^{N \times T_d \times D_F}$ carries the *implicit* information of the distilled frames, where T_d is the number of frames to be kept. In order to be *explicitly* aware of the distilled frames, the state of FD agent also contains the binary mask S'_M of the distilled frames.

Action. As shown in Fig. 4a, the FD agent outputs two types of discrete actions for each selected frame, *i.e.* "*stay distilled*" indicating the frame is informative that the agent determines to keep it, and "*shift to alternate*" indicating the agent determines to discard the frame and take in an alternate. The shifting may be frequent at the beginning but will gradually become stable after some explorations (Fig. 4a). In order to give equal chance for all alternates to be enrolled, the latest discarded frames are appended to the end of a queue and have the lowest priority to be enrolled again.

Feature-distilling Agent. The FD agent in Fig. 4b is also constructed under the Actor-Critic [16] framework. The agent takes in the global knowledge from the whole feature S'_F, the *implicit* local knowledge from the distilled features S'_{F_d}, and the *explicit* local knowledge from the binary frame mask S'_M. Finally, the agent outputs an action vector for the T_d distilled feature frames and a value for the current state. The action vector is sampled from the policy



(a) Illustration of the feature-distilling process



(b) Structure of the feature-distilling agent

Figure 4: (a) The FD agent has two discrete actions, *i.e.*, "*stay distilled*" (red icon) and "*shift to alternate*" (green icon). The "Queue" is a queue which contains the alternate feature frames, and $\mathcal{T}'$ is the deterministic state transition function. (b) The convolutional layers Conv1 and Conv3 (with kernel of 1x1) are used for channel squeezing, and Conv2 and Conv4 (with kernel of 3x3) are used for feature extracting. The"FC" denotes fully connected layer.

distribution during training, and is directly set as the action type with max probability during testing.

Optimization and Rewards. The optimization algorithm (A3C) and object function are same as the RG agent. The reward only contains the components about trajectory ascending and class shifting introduced above, *i.e.*,

$$r' = r_{ascend} + r_{shift}. \tag{13}$$

3.5. Training Procedure

In the proposed approach, the agents and the graph need to be updated respectively on CPU (to exploit numerous CPU cores/threads for asynchronous updating workers according to A3C algorithm [20]) and GPU. In addition, the

graph is updated after each video batch, but the agents are updated many times during each video when the number of reinforcement step reaches the updating interval τ_{max} or a terminal state is reached. Thus, the graph and agents are updated on different devices with different updating periods, and it is unable to optimize them with conventional end-to-end training. Therefore, we adopt alternate training. More details of the standard flowchart of A3C algorithm can be found in the *Supplementary Material*.

Individual Feature Preparation. Following [31], we finetune the Person-CNN (VGG16 [28]) pretrained on ImageNet [23] with individual action labels to extract visual features, and then train the Person-LSTM with individual action labels to extract temporal features. To lower the computation burden, the extracted individual features are saved to disk and only need reloading after this procedure.

Alternate Training. There are totally 9 separated training stages. At each stage, only one of the three components (SRG, trained with 15 epochs; FD- or RG-agent, trained with 2 hours) is trained and the remaining two are frozen (or removed). In the first stage, the SRG (without agents) is trained with the extract features to capture the context information within activities. In the second stage, the SRG is frozen, and the FD agent is introduced and trained with the rewards provided by the frozen SRG. In the third stage, the SRG and FD agent are frozen, the RG agent is introduced and trained with the rewards provided by the frozen SRG and FD agent. After that, one of the SRG, FD agent and RG agent is trained in turn with the remaining two be frozen in the following 6 stages.

4. Experiments

4.1. Datasets

Volleyball Datasets [14]. The Volleyball dataset is currently the largest dataset for group activity recognition. It contains 4830 clips of 55 volleyball videos. Each clip is annotated with 8 group activity categories (*i.e.*, right set, right spike, right pass, right winpoint, left winpoint, left pass, left spike and left set), and its middle frame is annotated with 9 individual action labels (*i.e.*, waiting, setting, digging, falling, spiking, blocking, jumping, moving and standing). We employ the metrics of Multi-class Classification Accuracy (MCA) and Mean Per Class Accuracy (MPCA) to evaluate the performance following [36].

Collective Activity Dataset (CAD) [8]. The CAD contains 2481 activity clips of 44 videos. The middle frame of each clip is annotated with 6 individual action classes (*i.e.*, NA, crossing, walking, waiting, talking and queueing), and the group activity label is assigned as the majority action label of individuals in the scene. Following [32], we merge the classes "walking" and "crossing" as "moving" and report the MPCA to evaluate the performance.

Since the existing datasets lack sufficient diversity of background [32], it is too difficult to distinguish useful objects (*e.g.*, volleyball) from noisy background without any annotation. Following [14, 19, 36, 25, 22, 31], we ignore the background and only focus on interactions among persons.

4.2. Implementation Details

For fair comparison with previous methods [22, 31], we use the same backbone network (Person-CNN) VGG16 [28]. It outputs 4096-d features and the Person-LSTM equipped with 3000 hidden neurons takes in all the features in T (T=10) time steps. In the SRG, the embedding sizes of node and edge are 1000 and 100 respectively, and the graph passes 3 iterations at each time. Thus, the number of hidden neurons in updating functions NN_{ve}, NN_{v}, and NN_{e} are 1000, 1000 and 100, respectively. In the RG agent, the fully connected layers FC1, FC2, ..., FC7 are respectively contains 512, 256, 512, 256, 256, 64 and 256 neurons, and its LSTM network contains 128 hidden nodes. In the FD agent, the number of feature frames to be kept T_d is practically set as 5. In Fig. 4b, the neuron number of the two FC layers from the left to right is 64 and 256, the channels of Conv1, Conv2, Conv3, Conv4 are respectively 1024, 1024, 256, 256, and the LSTM network contains 128 neurons.

During training, we use RMSprop/Adam (SRG/Agents) optimizer with an initial learning rate of 0.00001/0.0001 (SRG/Agents) and a weight decay of 0.0001. The batch size is 8/16 (CAD/Volleyball) for SRG training. The discount factor γ, entropy factor β and the number of asynchronous workers in A3C for both agents are respectively set as 0.99, 0.01 and 16. In practice, the updating interval τ_{max} and Ω (in Eq. 9) are set as 5/5 and 15/20 (RG/FD agent), respectively. In Volleyball dataset, following [14], the 12 players are split into two subgroups (*i.e.*, the left team and the right team) according to positions, and the RG agent are shared by the two subgroups in our framework, and finally the outputs of the two subgroups are averaged. In CAD dataset, since the number of individuals is varying from 1 to 12, we select 5 effective persons for each frame and fill zeros for the frames contain less than 5 persons following [36].

4.3. Baseline and Variants for Ablation Studies

To examine the effectiveness of each component in the proposed method, we conduct ablation studies with the following baseline and variants. *stagNet w/o Atten.* [22]: this baseline constructs a message passing graph network with the similar low-level features as our SRG. It implicitly represents the interactions by the passing messages, while our SRG explicitly models relations in a full graph network. *Ours-SRG*: this variant only contains the SRG of the proposed method. *Ours-SRG+T. A.*: this variant contains our SRG and a temporal attention over feature frames. *Ours-SRG+R. A.*: this variant contains our SRG and a relation

Table 1: Comparisons of recognition accuracy (%) on Volleyball dataset. "OF" denotes additional optical flow input.

Methods	Backbone	OF	MCA	MPCA
HDTM [14]	AlexNet	N	81.9	82.9
SBGAR [19]	Inception-v3	Y	66.9	67.6
CERN-2 [25]	VGG16	N	83.3	83.6
SSU [2]	Inception-v3	N	89.9	-
SRNN [4]	AlexNet	N	83.5	-
PC-TDM [36]	AlexNet	Y	87.7	88.1
stagNet [22]	VGG16	N	89.3	-
SPA+KD [31]	VGG16	N	89.3	89.0
SPA+KD+OF [31]	VGG16	Y	90.7	90.0
ARG [33]	VGG16	N	**91.9**	-
CRM [1]	I3D	Y	**93.0**	-
Baseline [22]	VGG16	N	87.9	-
Ours-SRG	VGG16	N	88.3	88.5
Ours-SRG+T. A.	VGG16	N	88.6	88.7
Ours-SRG+R. A.	VGG16	N	88.7	89.0
Ours-SRG+FD	VGG16	N	89.5	89.2
Ours-SRG+RG	VGG16	N	89.8	91.1
Ours-PRL	VGG16	N	**91.4**	**91.8**

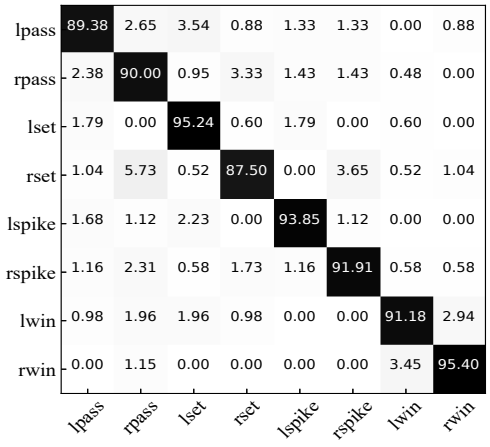


Figure 5: Confusion matrix on the Volleyball dataset.

attention that directly learns relation gates. *Ours-SRG+FD*: this variant contains both the SRG and FD agent, and they are trained alternately to boost each other. *Ours-SRG+RG*: this variant contains both the SRG and RG agent, and they are alternately trained. *Ours-SRG+FD+RG (PRL)*: our progressive reinforcement learning framework that contains all the proposed three components, including the SRG, the FD agent, and the RG agent.

4.4. Results on the Volleyball Dataset

To examine the effectiveness of each component, we compare the proposed PRL against the above baseline and variants. As Table 1 shows, although building graphs on similar low-level features, our semantic relation graph is superior to the baseline (stagNet w/o Atten. [22]) because our semantic relations are explicitly modeled while the baseline only implicitly contains them in the passing messages. Our SRG+FD boosts the SRG over 1.2% (MCA) and 0.7% (MPCA) by applying the FD agent to filter out ambiguous frames of features, and our SRG+RG also improves the performance of the SRG over 1.5% (MCA) and 2.6% (MPCA) by exploiting the RG agent to refine the relations. Our PRL achieves better performance by combining the advantages from the two agents. Note that the PRL eventually improves 3.1% (MCA) over the original SRG, which is even larger than the sum of increments from the two agents, 2.7% (MCA), indicating that the two agents can boost each other through the alternate training procedure. Besides, the agent-equipped variants SRG+FD and SRG+RG respectively perform better than corresponding attention-equipped variants SRG+T. A. and SRG+R. A. by 0.9% and 1.1% (MCA). The superiority of the agents probably owe to two reasons: 1) The attention variants can only learn from the annotated activity labels, while our RL-based agents can also learn from the historical experience during the policy exploring processes. 2) The attention variants only updates for each video batch, while our agents are updated many times during each single video (cf. training flowchart) that can achieve more fine-grained and video-specific adjustments.

Then, we compare the proposed PRL with other state-of-the-art methods. As shown in Table 1, our PRL is on par with the state-of-the-art method that has no extra optical flow input (ARG [33]). Our PRL even outperforms most of the methods that exploit optical flow input (including SBGAR [19], PC-TDM [36], and SPA+KD+OF [31]). Although CRM [1] performs somewhat better than our PRL, it is unfair to compare with. Because the CRM not only exploits extra optical flow input but only utilizes a much larger backbone (I3D [6]) than ours (VGG16 [28]).

In addition, the confusion matrix of the proposed PRL is shown in Fig. 5. As we can see, our PRL achieves promising recognition accuracies ($\geq$ 90%) on most of the activities. The main failure cases are from "set" and "pass" within the left and right subgroups, which is probably due to the very similar actions and positions of the key participants. We also visualized several refined semantic relation graphs in Fig. 6, where the relations with top5 gate values are shown and the importance degree of persons are indirectly computed by summing the connected relation gates (normalized over all persons). In Fig. 6a, benefited from the rewards of structured sparsity, our RG agent successfully discovers the subset of relations related to the "digging" person is the key to determine the activity "left pass". In Fig. 6b, the model predicts "right winpoint" mainly based on two relation clusters, including the cluster characterized

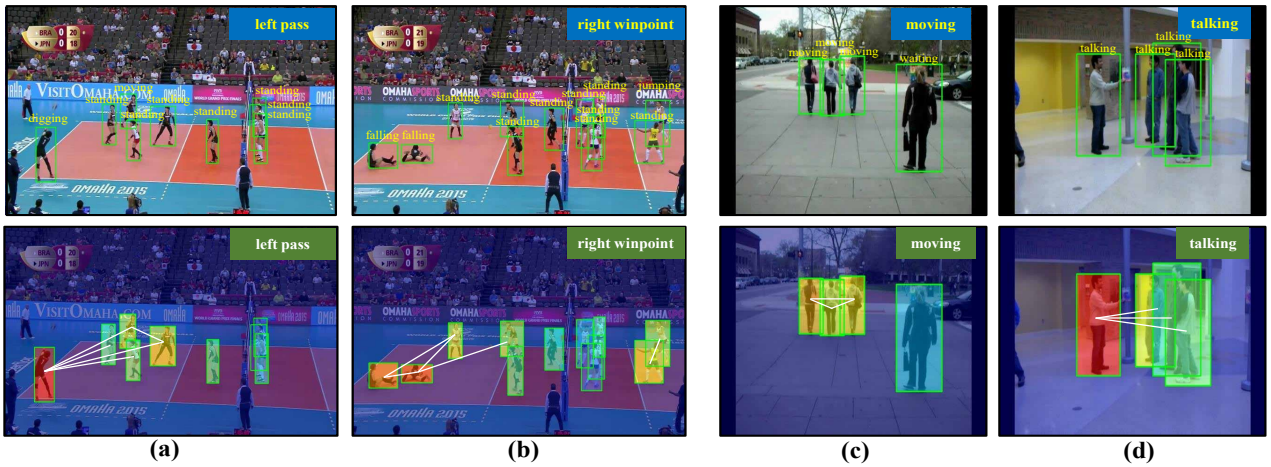


Figure 6: Visualization of the refined SRGs. The first row contains the obtained tracklets and the groundtruth labels of activity and person actions. The second row contains the refined SRGs and the predicted activity labels. The color of person represents its importance degree. To facilitate visualization, only the relations with top5/top3 (Volleyball/CAD) gate values are shown (the white lines). The samples of (a,b) and (c,d) are from the Volleyball and CAD datasets, respectively.

Table 2: Comparisons of recognition accuracy (%) on CAD dataset. "OF" denotes additional optical flow input.

Methods	Backbone	OF	MPCA(%)
HDTM [14]	AlexNet	N	89.6
CERN-2 [25]	VGG16	N	88.3
SBGAR [19]	Inception-v3	Y	89.9
PC-TDM [36]	AlexNet	Y	92.2
SPA+KD [31]	VGG16	N	92.5
SPA+KD+OF [31]	VGG16	Y	**95.7**
CRM [1]	I3D	Y	94.2
Baseline [22]	VGG16	N	87.7*
Ours-SRG	VGG16	N	89.4
Ours-SRG+R. A.	VGG16	N	90.0
Ours-SRG+T. A.	VGG16	N	90.1
Ours-SRG+FD	VGG16	N	91.1
Ours-SRG+RG	VGG16	N	91.4
Ours-PRL	VGG16	N	**93.8**

* MPCA is unavailable, MCA is listed instead.

by the two "falling" persons in the left team and the cheering cluster in the right team.

4.5. Results on the Collective Activity Dataset

Table 2 shows the comparison with different methods on the CAD dataset. Following [36, 31], the results regarding MPCA of several methods are calculated from the reported confusion matrices in [11, 14, 25, 19]. Our PRL outperforms the state-of-the-art method (SPA+KD [31]) without extra optical flow input by a margin of 1.3%. Although the SPA+KD+OF [31] performs better than our PRL, its main improvement (3.2%) is owed to the extra optical flow information (cf. Table 2). The backbone of CRM [1] (I3D) is much larger than ours (VGG19), making it less comparable. The detailed confusion matrix of our PRL on the CAD dataset can also be found in the *Supplementary Material*.

Furthermore, we analyze the results by visualizing the final SRGs. For the "Moving" activity in Fig. 6c, our method concentrates on the relations among the three moving persons to suppress the noisy relations caused by the "Waiting" person. Similarly, in Fig. 6d, our method successfully attends to the relations connected to the "Talking" person and weakens the relations among the three audiences.

5. Conclusion

In this work, we propose a novel progressive relation learning method to model and distill the group-relevant actions and interactions in group activities. A graph built on the spatiotemporal features and the interactions of individuals is used to explicitly model the semantic relations in group activities. A feature-distilling agent is proposed to progressively distill the most informative frames of the low-level features, and the relation-gating agent is proposed to refine the high-level relations in the semantic relation graph. Eventually, our PRL achieves promising results on two widely used datasets for group activity recognition.

Acknowledgements: This work was jointly supported by the National Key Research and Development Program of China (No. 2017YFA0105203), the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDB32040200), and the Hundred Talent Program of the Chinese Academy of Sciences (for S.Y.).

References

[1] Sina Mokhtarzadeh Azar, Mina Ghadimi Atigh, Ahmad Nickabadi, and Alexandre Alahi. Convolutional relational machine for group activity recognition. In *CVPR*, pages 7892–7901, 2019.

[2] 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.

[3] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[4] Sovan Biswas and Juergen Gall. Structural recurrent neural network (SRNN) for group activity analysis. In *WACV*, pages 1625–1632, 2018.

[5] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.*, 34(4):18–42, 2017.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[7] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, pages 215–230. Springer, 2012.

[8] 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. IEEE, 2009.

[9] 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.

[10] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017.

[11] 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.

[12] Guyue Hu, Bo Cui, and Shan Yu. Joint learning in the spatio-temporal and frequency domains for skeleton-based action recognition. *IEEE Transactions on Multimedia*, 2019.

[13] Guyue Hu, Bo Cui, and Shan Yu. Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1216–1221, 2019.

[14] 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.

[15] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[16] Vijay R. Konda and John N. Tsitsiklis. Actor-critic algorithms. In *NIPS*, pages 1008–1014, 1999.

[17] 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.

[18] Debang Li, Huikai Wu, Junge Zhang, and Kaiqi Huang. A2-RL: aesthetics aware reinforcement learning for image cropping. In *CVPR*, pages 8193–8201, 2018.

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

[20] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[22] 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.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[24] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin A. Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *ICML*, pages 4467–4476, 2018.

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

[26] 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.

[27] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *CVPR*, pages 29–38, 2017.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[29] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, pages 1057–1063, 1999.

[30] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *CVPR*, pages 5323–5332, 2018.

[31] Yansong Tang, Zian Wang, Peiyang Li, Jiwen Lu, Ming Yang, and Jie Zhou. Mining semantics-preserving attention for group activity recognition. In *ACM MM*, pages 1283–1291. ACM, 2018.

[32] Minsi Wang, Bingbing Ni, and Xiaokang Yang. Recurrent modeling of interaction context for collective activity recognition. In *CVPR*, 2017.

[33] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *CVPR*, pages 9964–9974, 2019.

[34] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[35] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[36] Rui Yan, Jinhui Tang, Xiangbo Shu, Zechao Li, and Qi Tian. Participation-contributed temporal dynamic model for group activity recognition. In *ACM MM*, pages 1292–1300. ACM, 2018.

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

[38] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.

[39] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*, pages 3634–3640, 2018.