Object-Relation Reasoning Graph for Action Recognition

Yangjun Ou, Li Mi, Zhenzhong Chen*
School of Remote Sensing and Information Engineering, Wuhan University, China
{ouyangjun,milirs,zzchen}@whu.edu.cn

Abstract

Action recognition is a challenging task since the attributes of objects as well as their relationships change constantly in the video. Existing methods mainly use object-level graphs or scene graphs to represent the dynamics of objects and relationships, but ignore modeling the fine-grained relationship transitions directly. In this paper, we propose an Object-Relation Reasoning Graph (OR^2G) for reasoning about action in videos. By combining an object-level graph (OG) and a relation-level graph (RG), the proposed OR^2G catches the attribute transitions of objects and reasons about the relationship transitions between objects simultaneously. In addition, a graph aggregating module (GAM) is investigated by applying the multi-head edge-to-node message passing operation. GAM feeds back the information from the relation node to the object node and enhances the coupling between the object-level graph and the relation-level graph. Experiments in video action recognition demonstrate the effectiveness of our approach when compared with the state-of-the-art methods.

1. Introduction

Action recognition is one of the fundamental tasks in the field of video understanding [8], and it remains an active topic in the vision research community. The goal of action recognition is to identify activities in the video according to object states. As it is difficult to capture object transitions by a global representation of the video [2, 25], object-based action recognition has attracted increasing attention.

Object-based methods [30] mainly represent the objects as the nodes of a graph and obtain the action category through graph reasoning at the object-level. However, these methods often fail to explicitly model the interaction between objects. To consider object interactions in action recognition, current efforts [14] decompose the objects and relationships in a video according to the event segmentation theory [18], where events can be divided into consistent groups or represented as hierarchical structures. However, these methods represent the dynamics of objects and relationships by aggregating them from the scene graph, and ignore modeling and reasoning about fine-grained relationship transitions.

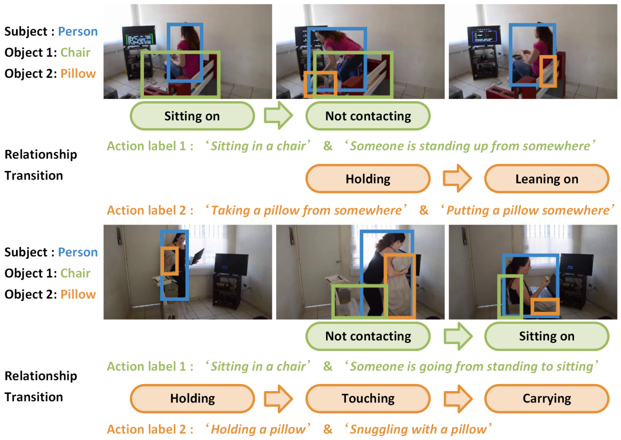

Subject : Person
Object 1: Chair
Object 2: Pillow
Sitting on
Not contacting
Relationship Transition
Action label 1 : 'Sitting in a chair' & 'Someone is standing up from somewhere'
Holding
Leaning on
Action label 2 : 'Taking a pillow from somewhere' & 'Putting a pillow somewhere'
Subject : Person
Object 1: Chair
Object 2: Pillow
Not contacting
Sitting on
Relationship Transition
Action label 1 : 'Sitting in a chair' & 'Someone is going from standing to sitting'
Holding
Touching
Carrying
Action label 2 : 'Holding a pillow' & 'Snuggling with a pillow'

Figure 1. Illustration of fine-grained relationship transition in the videos. Two examples of different action labels with same subject-object pairs but different relationship transitions are presented.

Fine-grained relationship transition plays an important role in distinguishing action categories, especially actions with similar characteristics. Figure 1 shows two examples of similar video content. In both cases, the subject is a person and the objects are a chair and a pillow. However, there are some slight differences in the relationship transitions between the subject and objects, resulting in different actions. For the visual relationship between the <person-chair> pair, when the transition is '***sitting on***' → '***not contacting***', it represents the action of '*Someone is standing up from somewhere*'. However, when the transition is '***not contacting***' → '***sitting on***', it represents another action of '*Someone is going from standing to sitting*'. For the visual relationship between the <person-pillow> pair, when the transition is '***holding***' → '***leaning on***', it represents the action of '*Taking a pillow from somewhere*' and '*Putting a pillow somewhere*'. However, in the transition of '***holding***'

[*] Corresponding author: Zhenzhong Chen.

→ '***touching***' → '***carring***', the actions change to '*Holding a pillow*' and '*Snuggling with a pillow*'. Therefore, modeling the relationship transition of independent subject-object pairs across key frames is crucial for the action recognition task. It helps the network to reason the video actions in a human-like way, thus to enhance the explainability of the network and get more precise action categories.

In this paper, we propose an Object-Relation Reasoning Graph (OR^{2}G) to model the fine-grained transition of the objects and relationships in a video as shown in Figure 2. Based on the above analysis, we split multiple objects and relationships into independent items, and further decompose the actions in detail. Firstly, for fine-grained modeling of object attribute transition, we propose an actor-centric object-level graph (OG). By modeling the dependencies between the subject and the objects in an actor-centric way, the object-level graph captures more critical information about object interactions. Secondly, considering the impact of relationship transition between the subject and the objects, a relation-level graph (RG) is proposed to model the dependencies among fine-grained relationships along the temporal dimension. Finally, to enhance the coupling between the two graphs, we propose a graph aggregating module (GAM). With a multi-head attention edge-to-node message passing operation, the information of relation-level graph feeds back to the object-level graph in spatial dimension.

Our contributions can be summarized as follows:

- OR^{2}G is proposed to model fine-grained attribute and relationship transition for the challenging problem of distinguishing actions with similar characteristics. By reasoning on both object-level graph and relation-level graph, OR^{2}G explains more clearly how actions occur through the slight transition of attributes and relationships, and create interpretable representation for a variety of complex actions.
- A graph aggregating module which adopts multi-head attention edge-to-node message passing operation is proposed to make the two graphs more coupled. The information of relation-level graph feeds back to the object-level graph in the spatial dimension, permitting a more reasonable utilization of relationship transition information.

The remainder of this article is structured as follows. Section 2 gives an overview of related work. Section 3 presents our proposed method. Section 4 provides the implementation details and experimental results on the public action recognition dataset, followed by the conclusions in Section 5.

2. Related Work

During the past decades, action recognition has attracted a lot of attention in the computer vision community. In this section, we mainly focus on the study of action recognition, graph structure and visual reasoning.

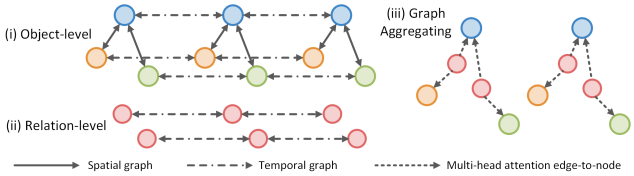


Figure 2. Object-Relation Reasoning Graph (OR^{2}G) is proposed to distinguish actions with similar characteristics by fine-graied attribute and relationship transition modeling.

2.1. Action Recognition

Most of the early works for action recognition have focused on designing hand-crafted features, such as the Improved Dense Trajectory (IDT) [27]. The strategies are still widely used and show very competitive results in different video related tasks. Recently, due to the great advances of deep learning, a large number of CNN-based approaches have been proposed and surpass such traditional approaches.

The existing deep-learning methods for action recognition can be classified into two types. The first is based on two-stream networks [6, 7, 23, 28, 32], which take RGB frames and optical flows as input for each stream. Simonyan et al. [23] first proposed the two-stream ConvNet architecture for action recognition. Wang et al. [28] proposed a sparse temporal sampling strategy for the two-stream structure to model long range relationship in the time domain. The second type is based on 3D convolutional neural networks (3D CNN) [2, 25, 35], which are designed to capture the spatial-temporal features jointly. The first 3D CNN for action recognition is C3D [25], which models the spatial and temporal features together. By inflating the filters and pooling kernels of very deep image classification ConvNets into 3D, Carreira et al. [2] proposed the I3D network to learn seamless spatio-temporal features. Recently, Kondratyuk et al. [16] use Neural Architecture Search to get network structure with the best performance for action recognition.

2.2. Graph Structure in Videos

When used to recognize actions, two types of graph structures are utilized: skeleton-based graphs and object-based graphs. Skeleton-based graphs are constructed to model the skeleton information based on a fixed graph structure. Yan et al. [36] introduced the graph convolution operation to skeleton-based action recognition and proposed a novel spatial-temporal graph convolutional network to learn the spatial and temporal pattern from skeleton data automatically. Cheng et al. [4] proposed a shift graph con-

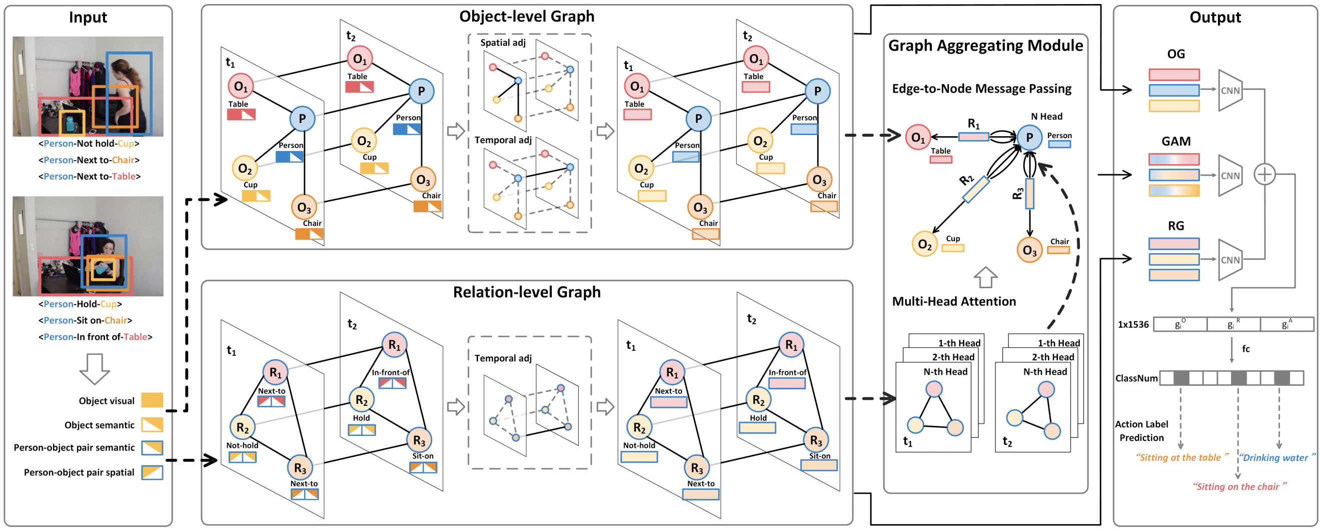


Figure 3. The overall architecture of OR^{2}G. Firstly, visual, spatial and semantic features are extracted from the object locations, object categories and the visual relationship categories in key frames to construct the nodes in the graphs. Secondly, the node features of the two graphs are refined at the object-level and the relation-level, respectively. Then, the information of relation-level graph is fed back to the object-level graph by a graph aggregating module. Finally, the node information of three modules are concatenated to learn the representation of video for the action recognition task.

volutional network to combine novel shift graph operation with lightweight point-wise convolutions for skeleton-based action recognition. Object-based graphs are utilized in non-skeleton-based action recognition, where the nodes represent objects or regions in a visual scene. Wang et al. [30] captured the important cues by representing videos as space-time region graphs, and then used Graph Convolutional Network (GCN) [15] to perform long-range temporal modeling of human-object and object-object relationships. Herzig et al. [10] modeled the video context by a disentangled graph embedding derived from several inter-object graphs with spatial and temporal hierarchy.

2.3. Visual Reasoning

For visual reasoning in videos, methods can be divided into two families, frame-level reasoning and object-level reasoning. Frame-level reasoning conducts relational reasoning among video frames. For example, Zhou et al. [38] introduced an interpretable module to learn and reason about temporal dependencies between video frames. Huang et al. [12] proposed a Graph-based Temporal Reasoning Module to learn the relations of action segments. Zhang et al. [37] proposed a learnable temporal relation reasoning graph to capture the appearance features among regions and the temporal relation between video sequences simultaneously. Different from frame-level reasoning, object-level reasoning relies on an object-level graph to model the interactions among objects or regions in videos. Baradel et al. [1] introduced an Object Relation Network to the action recognition task and proposed a novel model to achieve object-level reasoning in videos. Sun et al. [24] proposed a weakly supervised actor-centric relational network to accumulate pair-wise relation information for action classification. Chen et al. [3] proposed an approach to reason between regions by a fully-connected graph and projected the node features to the coordinate space. Zhuo et al. [39] integrated scene graph generation methods into a video action recognition framework. Wu et al. [34] built an object-level graph to capture the appearance and position relation between actors through Graph Convolutional Network. Materzynska et al. [20] proposed a spatial-temporal interaction network that operated on object-centric features and performed spatial interaction reasoning to obtain a classification decision in compositional action recognition.

3. Methodology

To reason the attribute transitions as well as the relationship transitions in the video, we decompose action into a series of objects and relationships according to the event segmentation theory, and propose an Object-Relation Reasoning Graph (OR^{2}G) for the action recognition task. The overall architecture of the proposed OR^{2}G is shown in Figure 3. It is mainly constructed of three parts, the object-level graph, the relation-level graph and the graph aggregating module.

3.1. Object-level Graph Reasoning

The nodes in the object-level graph represent persons or objects in video frames, and we refer to them as person nodes or object nodes, respectively. The object-level graph is constructed to obtain the relationships between the person node and the object nodes, as well as the attribute transitions of object nodes.

To obtain sufficient information about the objects, two kinds of high-level features are used. The first is the visual feature v_i^O extracted by ResNet [9], and the second is the semantic feature s_i^O obtained by embedding the object category to the semantic feature space. The location and category information of the objects are either provided by the dataset or extracted by a fine-tuned Faster R-CNN [21]. Attributes of the object nodes are obtained by concatenating the two features as $x_i^O = [v_i^O, s_i^O]$.

After extracting the node information, edges are added between nodes to build the graph. Considering the information flows in spatial and temporal dimensions, we add two kinds of edges to the graph. Firstly, we add spatial edges between the person node and object nodes in the same frame to evaluate the correlation of person-object pairs. The spatial adjacency matrix can be formulated as:

$$\mathbf{A}_{S_{ij}}^O = \begin{cases} 1, & o_i/o_j = \text{`person'} \; and \; t_i = t_j \\ 0, & otherwise \end{cases} \tag{1}$$

where o_i and o_j represent the object categories of node i and j, respectively. t_i and t_j represent the video key frames that the objects belong to. Since the actions are based on the action subject, we construct an actor-centric object-level graph whose central node is the person node. In each identical key frame, the object nodes only connect to the person node, while between different key frames, the person node and the object nodes only connect to themselves by temporal edges. The temporal adjacency matrix can be formulated as:

$$\mathbf{A}_{T_{ij}}^O = \begin{cases} 1, & o_i = o_j \; and \; t_i - t_j < T_1 \\ 0, & otherwise \end{cases} \tag{2}$$

where T_1 is the threshold of the distance between adjacent frames. We combine spatial and temporal edges by straight combination to generate the overall adjacency matrix of the object-level graph. The computational formulas for straight combination are:

$$\begin{gathered} \mathbf{A}_{ij}^O = \mathbf{A}_{S_{ij}}^O \oplus \mathbf{A}_{T_{ij}}^O \\ \mathbf{G}^O = \sigma(\hat{\mathbf{D}^O}^{-\frac{1}{2}} \hat{\mathbf{A}^O} \hat{\mathbf{D}^O}^{-\frac{1}{2}} \mathbf{X}^O \mathbf{W}^O) \end{gathered} \tag{3}$$

where $\oplus$ denotes the OR operation. $\mathbf{A}^O$ is the overall adjacency matrix for object-level graph, and $\hat{\mathbf{A}^O}$ is adjacency matrix $\mathbf{A}^O$ with added self-connections I_N. $\hat{\mathbf{D}^O}$ is the degree matrix of $\hat{\mathbf{A}^O}$. $\mathbf{X}^O$ is the input features of the object nodes in the graph. $\mathbf{W}^O$ is the weight matrix of the layer. Once the graph is constructed, the node information is updated by the graph convolution operation.

3.2. Relation-level Graph Reasoning

The nodes in the relation-level graph represent the relationships between the subject and the objects, and we refer to them as relation nodes. The function of the relation-level graph is to obtain the fine-grained relationship transitions of relation nodes.

Similar to the object-level graph, we also used two kinds of high-level features for the relation-level graph. One is the spatial feature sp_i^R of the subject and the objects, which is extracted using relative spatial position descriptors. Different from the visual feature and the semantic feature, the spatial feature is a relative feature which represents the relative position of two bounding boxes. To obtain the relative spatial feature of the bounding box, we adopt the idea of box regression [11], in which the relative spatial feature sp_i^R is defined as:

$$\begin{aligned} sp_i^R = [&\Delta(b_i, b_p); \Delta(b_i, b_{ip}); \Delta(b_p, b_{ip}); \\ &\text{iou}(b_i, b_p); \text{dis}(b_i, b_p)] \end{aligned} \tag{4}$$

where b_p is the bounding box of the subject. b_i is the bounding box of the object. b_{ip} is the union of b_i and b_p. $\Delta(b_i, b_p)$ is the box delta that regresses the bounding box b_i to b_p. $\text{dis}(b_i, b_p)$ and $\text{iou}(b_i, b_p)$ are the normalized distance and IoU between b_i and b_p, respectively.

The other one is the semantic feature sm_i^R obtained by embedding the visual relationship category between the subject and the objects to the semantic feature space. The visual relationship category labels are either provided by the dataset or the fine-tuned visual relationship detection network [11, 19] as introduced in Section 4.1. Features of the relation nodes are obtained by concatenating the two features as $x_i^R = [sp_i^R, sm_i^R]$.

We add temporal edges between the relation nodes of same subject-object pair at adjacent frames to evaluate relationship transition, which can be formulated as:

$$\mathbf{A}_{T_{ij}}^R = \begin{cases} 1, & r_i = r_j \; and \; t_i - t_j < T_2 \\ 0, & otherwise \end{cases} \tag{5}$$

where r_i and r_j represent the subject-object pairs that compose the visual relationship. The computational formula for the relation-level graph is:

$$\mathbf{G}^R = \sigma(\hat{\mathbf{D}_T^R}^{-\frac{1}{2}} \hat{\mathbf{A}_T^R} \hat{\mathbf{D}_T^R}^{-\frac{1}{2}} \mathbf{X}^R \mathbf{W}_T^R) \tag{6}$$

where $\hat{\mathbf{A}_T^R}$ is temporal adjacency matrix $\mathbf{A}_T^R$ with added self-connections I_N. $\hat{\mathbf{D}_T^R}$ is the degree matrix of $\hat{\mathbf{A}_T^R}$. $\mathbf{X}^R$ is the input features of the relation nodes in the graph. $\mathbf{W}_T^R$ is the weight matrix of the layer.

3.3. Graph Aggregating Module

The object-level graph transfers and reasons the information of the object nodes, while the relation-level graph transfers and reasons the information of the relation nodes. However, the two graphs are still relatively independent and do not interact with each other. To make the two graphs more coupled, we proposed a graph aggregating module with a multi-head attention edge-to-node message passing operation. In this way, the information of the relation-level graph feeds back to the object-level graph in spatial dimension.

In the relation-level graph, the information of each node corresponds to the information of each edge in the object-level graph, so we take the node embeddings of the relation-level graph as the edge embeddings of the object-level graph. The graph aggregating module is actually updating the object-level graph according to the relation-level graph.

In the general graph convolutional network, the edge-to-node message passing operation [11] can be formulated as:

$$x_i' = f_e(\frac{1}{d_i} \sum_{e_{ij} \in \mathcal{E}} e_{ij}) \tag{7}$$

where x_i' is the updated node embedding. $\mathcal{E}$ denotes the edge set. e_{ij} is the edge embedding between the object node i and j. d_i is the amount of edges connected to node i. f_e is the mapping between the edge and the node.

In the proposed method, when updating the central node (person node) with the edge embedding in each frame, a multi-head attention edge-to-node message passing operation is proposed for the special actor-centric structure in the object-level graph. In the actor-centric graph, each object node has only one edge connected to it, while the person node has multiple edges connected to it. According to eq. 7, each object node is only updated by one connected edge embedding, while there are multiple edge embeddings feed back to the person node embedding with the same weight. However, the importance of each subject-object pair varies for different actions in the multi-label action recognition task. So in this section, we optimize the edge-to-node message passing operation based on the multi-head attention mechanism [26], which is inspired from the self-attention mechanism and also described as the mapping from $Query(\mathbf{Q})$ to $Key(\mathbf{K})$-$Value(\mathbf{V})$. Each head in the multi-head attention mechanism linearly transforms $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ through the parameter matrix $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$. This process can be formulated as:

$$head_i(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{W}_i^Q(\mathbf{K}\mathbf{W}_i^K)^T}{\sqrt{d_k}})\mathbf{V}\mathbf{W}_i^V \tag{8}$$

where i is the index of heads, and the parameter matrix of each head is not shared. d_k is the dimension of $\mathbf{K}$.

According to eq. 7 and 8, the updating of the person node in each frame can be formulated as:

$$\begin{aligned} \mathbf{G}_p^A = \frac{1}{d_p} \sum [&head_1(\mathbf{G}_p^R, \mathbf{G}_p^R, \mathbf{G}_p^R), ..., \\ &head_h(\mathbf{G}_p^R, \mathbf{G}_p^R, \mathbf{G}_p^R)]\mathbf{W}^A \end{aligned} \tag{9}$$

where h is the number of heads, $\mathbf{G}_p^A$ is the updated embedding of the central node, $\mathbf{G}_p^R$ is the feature matrix of all the edges connected to node p, d_p is the number of edges connected to the person node, $\mathbf{W}^A$ is the weight matrix of output embedding.

The updated central node, the updated object nodes, and the output of the object-level graph are concatenated and mapped to generate the entire output of the graph aggregating module. This process can be formulated as:

$$\mathbf{G}^A = f_e([\mathbf{G}_p^A, \mathbf{G}^R, \mathbf{G}^O]) \tag{10}$$

3.4. Multi-class Action Recognition

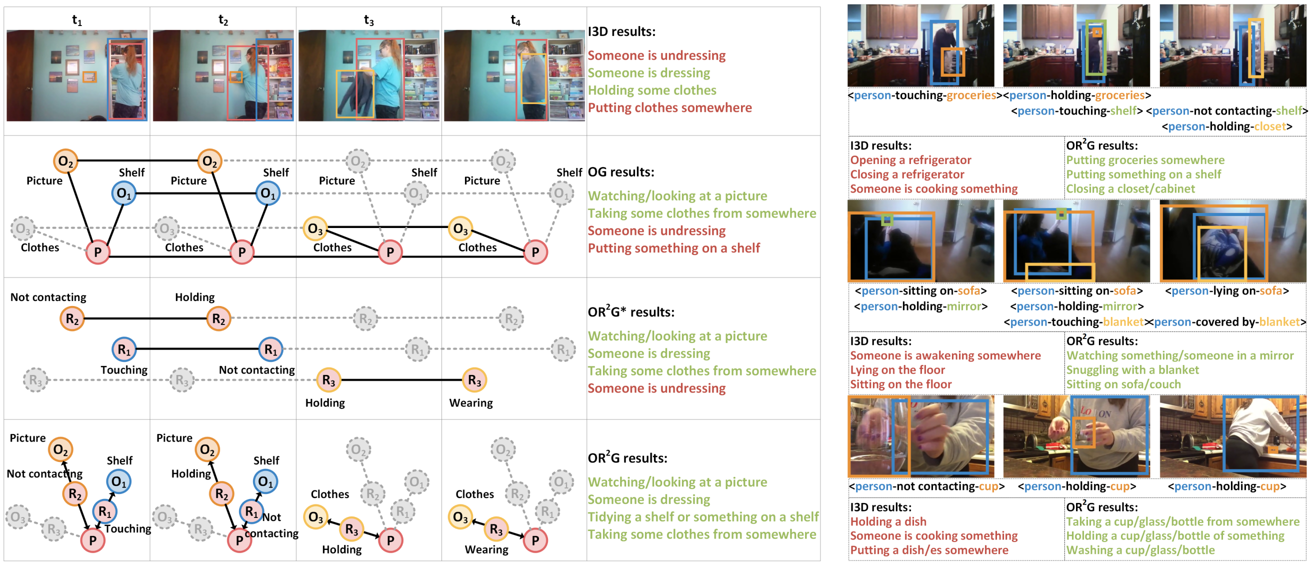
After performing the spatial-temporal graph convolution, all the update features are rearranged to compose three 3D features, i.e., the object-level feature, the relation-level feature and the aggregating feature. The height and channel number of the two features are equal to the number of frames and to the dimension of the updated features, respectively. The width of the object-level feature is equal to the number of object nodes in each frame, while the width of the relation-level feature is equal to the number of relation nodes in each frame. The rearranged features are processed by several convolutional layers and pooling layers to obtain $1 \times d$ dimension representations g_i^O, g_i^R and g_i^A, which contain the information of the OG, RG and GAM, respectively. In addition, the video is processed by a pre-trained I3D with non-local blocks [29] (I3D-NL) network to obtain another $1 \times d$ dimension global feature g_i^G.

These features are then concatenated together as $g_i = [g_i^O, g_i^R, g_i^A, g_i^G]$ for action recognition. The confidence of each class is $y_i = sigmoid(\mathbf{W}_f g_i)$, where $\mathbf{W}_f$ is the embedding matrix that maps interaction embeddings to match the action categories. Binary cross entropy loss is used in the training.

4. Experiments

4.1. Implemention Details

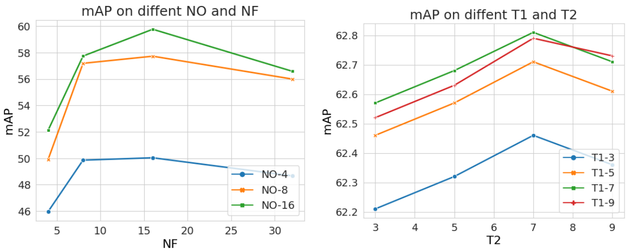
Dataset. The Charades dataset [22] contains 9,848 videos with an average length of 30 seconds. There are 157 action classes and multiple actions can happen at the same time. The Action Genome dataset [14] is annotated based on the Charades dataset. The dataset decomposes actions and focuses on video clips where the actions occur. It contains a total number of 234K key frames, on which 476K object bounding boxes and 1.72M relationships are annotated. There are a total number of 35 object categories and 25 relationship categories, among which the relationship categories are divided into 3 types, namely 3 classes of attention relationships, 6 classes of spatial relationships and 16 classes of contacting relationships. The multi-class action recognition task provides a video sequence as input



(a) Examples of I3D, OG, OR^{2}G* and OR^{2}G with detail structures

(b) Several examples of I3D and OR^{2}G

Figure 4. The visualization results of different structures. Top-K results are listed. The correct predictions are marked in green and the incorrect predictions are marked in red.



(a) Object and frame numbers

(b) T1 and T2 threshold values

Figure 5. Hyper parameter experiments.

and expects multiple action labels as output. Performance is measured by mean Average Precision (mAP).

Backbones. We used the ResNet-152 network as the visual feature extractor for bounding box regions in the video frame. The ResNet-152 network is pre-trained on the ImageNet dataset [17] and takes object image cropped from each bounding box as input. As in the state-of-the-art methods used for comparison, we also adopt the I3D-NL network as the video feature extractor. The backbone of the I3D-NL network is ResNet-101. The I3D-NL network is pre-trained on the Kinetics-400 dataset [2] and takes RGB video frames as input.

Evaluation Modes. Following the standard evaluation in [14], we used two standard evaluation modes for all the ablation study experiments: (1) Oracle (OR^{2}G Oracle): Object locations, object categories and human-object relationships (attention, spatial and contact relationships) are provided by the ground truth of the Action Genome dataset [14]. (2) Prediction (OR^{2}G): Faster R-CNN [21] with ResNet-101 is used as the backbone for region proposals and object detection. The network is then fine-tuned on object locations and object categories in the Action Genome dataset. A graph convolutional network with FC layer [11, 19] is used for visual relationship detection. The network is then fine-tuned on human-object relationship categories in the Action Genome dataset. For a fair comparison, we used the same train/val splits as in the Charades dataset [22].

Experimental Setup. All the experiments are conducted under the same design. The input of the network contains N_f key frames for each video. For the training set, the frames are extracted randomly over the whole video, while for the validation set, the frames are extracted evenly over the whole video. The frames are sent into the network in the order of time, and when the total number of frames is less than N_f, the missing frames are filled with zeroes. N_o objects are selected for each video, with the actor node ranking the first. To select the other objects, the number of occurrences of objects in all frames is counted and sorted. For the training set, the top $N_o/2$-1 most frequent objects are selected, and the other $N_o/2$ objects are randomly selected from the remaining objects. For the validation set, the top $N_o - 1$ most frequent objects are selected. When the total number of objects is less than N_o or when the objects do not appear in the current frame, the missing objects are filled with zeroes.

Table 1. Structure analysis.

Evaluation Mode	Method	OG edge	Combine type	OG	RG	GAM	mAP
Oracle	OG-S	Spa	-	✓	✗	✗	57.95
	OG-T	Tem	-	✓	✗	✗	58.94
	OG-w	S+T	weighted	✓	✗	✗	59.53
	OG	S+T	straight	✓	✗	✗	59.77
	OR^{2}G*	S+T	straight	✓	✓	✗	62.81
	OR^{2}G	S+T	straight	✓	✓	✓	**63.28**
Prediction	OG-S	Spa	-	✓	✗	✗	31.30
	OG-T	Tem	-	✓	✗	✗	32.06
	OG-w	S+T	weighted	✓	✗	✗	32.63
	OG	S+T	straight	✓	✗	✗	32.65
	OR^{2}G*	S+T	straight	✓	✓	✗	33.60
	OR^{2}G	S+T	straight	✓	✓	✓	**34.24**

Table 2. Feature analysis.

Evaluation Mode	Method	Object feature	Relation feature	mAP
Oracle	OG	V	-	34.56
	OG	S	-	58.78
	OG	V + S	-	59.77
	OR^{2}G*	V + S	Sp	60.28
	OR^{2}G*	V + S	Sm	62.65
	OR^{2}G*	V + S	Sm+Sp	**62.81**
Prediction	OG	V	-	25.51
	OG	S	-	32.36
	OG	V + S	-	32.65
	OR^{2}G*	V + S	Sp	32.89
	OR^{2}G*	V + S	Sm	33.17
	OR^{2}G*	V + S	Sm+Sp	**33.60**

4.2. Hyper Parameters

Object and frame numbers. The videos in the Action Genome dataset contain different numbers of key frames and objects. We vary the numbers of sampling frames N_f and objects N_o into the object-level graph and show the results in Figure 5a. N_f is taken as 4, 8, 16 and 32 frames, respectively. N_o is taken as 4, 8 and 16 objects from each video segment, respectively. It can be seen from the experimental results that the best performance can be obtained when N_o is set to 16. The mAP of the validation set increases and then decreases as the number of input video frames N_f grows, and the optimal experimental results are obtained when N_f is set to 16.

T1 and T2 threshold values. We compare the performance of different distance thresholds T_1 in the object-level graph and T_2 in the relation-level graph, respectively. As shown in Figure 5b, it can be seen from the experimental results that the best performance can be obtained when T_1 is set to 7. For the threshold of T_2, the mAP of the validation set increases and then decreases as the threshold of T_2 grows, and the optimal experimental results are obtained when T_2 is set to 7.

4.3. Ablation Study

Structure Analysis

• Components of OR^{2}G. Table 1 explores the effectiveness of each module of the proposed OR^{2}G. The baseline corresponds to the performance obtained by the object-level reasoning graph (OG). No relation-level information is present in this baseline. OR^{2}G* adds a relation-level graph to the baseline and gains 2.9 points compared to our baseline in Oracle evaluation mode. This indicates that modeling of the fine-grained relationship transition is able to extract representative features for distinguishing different actions. OR^{2}G is the proposed method and gains 0.4 points compared to OR^{2}G* in Oracle evaluation mode, which proves that the graph aggregating module improves the coupling of the two graphs and makes better use of relationship without introducing additional information.

To show the effect of our proposed modules more intuitively, we give some examples and visualize them in Figure 4. As shown in Figure 4a, I3D tends to extract the global representation of the video, and its result only concentrates on actions related to the 'clothes'. In the result of OG, the action related to the 'picture' (i.e., '*Watching/looking at a picture*') is also correctly recognized with the addition of fine-grained object information. But due to the absence of relationship cues, OG fails to recognize the action of '*someone is dressing*'. This problem is solved by OR^{2}G* with the additional relationship transition information. Only the proposed OR^{2}G recognizes the action of '*Tidying a shelf or something on a shelf*', because it makes more reasonable use of the relationship transition with the graph aggregation module, which gives different attention weights to different visual relationships (i.e., the <person-picture> pair and the <person-shelf> pair).

• Construction of OG. In addition, we also compare the performance of the two kinds of adjacent edges in the object-level graph. OG-S is the method with the spatial edges given by Eq. 1, while OG-T is the method with temporal edges given by Eq. 2. The two kinds of adjacent edges represent different ways of message passing, so we compare the performance of graphs with different combination types for spatial and temporal edges. OG-w shows the result of the weighted combination of the two graphs, while OG shows the result of the straight combination in Eq. 3. Compared with the results of spatial edges or temporal edges, both the combined graphs achieve better performance, which proves the complementarity between the two edges. When comparing the two combined graphs, it can be found that the straight combination is slightly better than the weighted combination.

Table 3. Backbone analysis in two evaluation modes.

Evaluation Mode	Backbone	mAP
Oracle	Resnet	63.28
	Resnet+I3D	**67.51**
Prediction	Resnet	34.24
	Resnet+I3D	**44.91**

Feature Analysis

• Object features. We compare the performance of different object features to evaluate their effectiveness in Table 2. V represents the visual feature and S represents the semantic feature. Comparing the three OG with different object features as input, it can be seen that the S+V achieves the best performance, which verifies the complementarity between the two feature types.

• Relation features. The experiments of relation features are also shown in Table 2. Sm and Sp are semantic feature and spatial feature for the relation-level graph, respectively. Comparing the three OR^{2}G* with different relation features as input, it can be seen that the Sm+Sp achieves the best performance, which verifies that both the semantic feature and the spatial feature are beneficial for the relation-level reasoning.

Backbone Analysis

The backbones for the proposed OR^{2}G are ResNet and I3D. When the backbone is ResNet, only g_i^O, g_i^R and g_i^A are concatenated for classification. When the backbone is I3D, that's the OR^{2}G we proposed, in which g_i^O, g_i^R, g_i^A together with the I3D feature g_i^G are concatenated for classification. It can be seen from Table 3 that the accuracy of OR^{2}G is greatly improved with the addition of the I3D feature.

Prediction Evaluation Mode

The overall trend of the Prediction evaluation mode is consistent with that of the Oracle evaluation mode in ablation study experiments, and similar conclusions can be obtained. Comparing the results of the two evaluation modes, we find that with the ground truth object locations, object categories and human-object relationships provided by Action Genome, the improvement in mAP can be as high as 23%. It means that the performance of the proposed OR^{2}G can be further improved with the improvement on object information or visual relationships information.

4.4. Comparison with the State-of-the-art Methods

To demonstrate the effectiveness of the proposed OR^{2}G, we compare the proposed method with related works for the multi-label action recognition task. Table 4 shows the results of comparing our proposed method to the existing methods in the Charades dataset. Timeception [13] and LFB [33] model the action using long-range temporal information, while SlowFast [5] models the action based on the difference of speed between action subject and background. It is difficult for these methods to capture the object transitions. STRG [30] and SVAG [31] model the actions based on objects and voxels, respectively, ignoring explicitly modeling the interaction between objects. Though SGFB [14] takes visual relationships into account, it ignores modeling and reasoning about fine-grained relationship transitions. In the case of the Prediction evaluation mode, our proposed OR^{2}G is superior to these methods and achieves the state-of-the-art performance with 44.9% mAP. For a fair comparison, we also evaluated our method in the Oracle evaluation mode and compared it with the SGFB method. The mAP of OR^{2}G is higher than that of SGFB by 7%, indicating that our method is able to reason about actions more accurately after being provided with effective annotations of objects and visual relationships.

Table 4. Action recognition on Charades validation set in mAP (%).

Method	Backbone	Pre-train	mAP
I3D + NL [2, 29]	R101-I3D-NL	Kinetics-400	37.5
STRG [30]	R101-I3D-NL	Kinetics-400	39.7
Timeception [13]	R101	Kinetics-400	41.1
SlowFast [5]	R101	Kinetics-400	42.1
SlowFast+NL [5, 29]	R101-NL	Kinetics-400	42.5
LFB [33]	R101-I3D-NL	Kinetics-400	42.5
SVAG [31]	R101-NL	Kinetics-400	44.1
SGFB [14]	R101-I3D-NL	Kinetics-400	44.3
OR^{2}G (ours)	R101-I3D-NL	Kinetics-400	**44.9**
SGFB Oracle [14]	R101-I3D-NL	Kinetics-400	60.3
OR^{2}G Oracle (ours)	R101-I3D-NL	Kinetics-400	**67.5**

5. Conclusions

In this paper, an Object-Relation Reasoning Graph (OR^{2}G) is proposed for action recognition. The proposed OR^{2}G uses graph convolutional network at object-level and relation-level to reason the fine-grained object and relationship transitions through the visual, spatial and semantic features of the objects and visual relationships in the video. Specifically, the graph aggregating module is proposed to make more reasonable use of relationship transition information. Ablation experiments verified the effectiveness of the object-level graph, the relation-level graph and the graph aggregating module. Experiments on the Charades dataset show that the proposed method improves on state-of-the-art performance in both Oracle and Prediction evaluation modes.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grant No.62036005).

References

[1] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *Proceedings of the European Conference on Computer Vision*, 2018. 3

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 6, 8

[3] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3

[4] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 8

[6] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[7] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[8] Tal Hassner. A critical review of action recognition benchmarks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 4

[10] Roei Herzig, Elad Levi, Huijuan Xu, Hang Gao, Eli Brosh, Xiaolong Wang, Amir Globerson, and Trevor Darrell. Spatio-temporal action graph networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 3

[11] Yue Hu, Siheng Chen, Xu Chen, Ya Zhang, and Xiao Gu. Neural message passing for visual relationship detection. In *Proceedings of the International Conference on Machine Learning Workshops*, 2019. 4, 5, 6

[12] Yifei Huang, Yusuke Sugano, and Yoichi Sato. Improving action segmentation via graph-based temporal reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[13] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 8

[14] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 5, 6, 8

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*, 2017. 3

[16] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 6

[18] Christopher A Kurby and Jeffrey M Zacks. Segmentation in the perception and memory of events. *Trends in Cognitive Sciences*, 12(2):72–79, 2008. 1

[19] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of the European Conference on Computer Vision*, 2016. 4, 6

[20] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 4, 6

[22] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of the European Conference on Computer Vision*, 2016. 5, 6

[23] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014. 2

[24] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision*, 2018. 3

[25] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 1, 2

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 5

[27] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013. 2

[28] 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 *Proceedings of the European Conference on Computer Vision*, 2016. 2

[29] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5, 8

[30] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision*, 2018. 1, 3, 8

[31] Yang Wang, Gedas Bertasius, Tae-Hyun Oh, Abhinav Gupta, Minh Hoai, and Lorenzo Torresani. Supervoxel attention graphs for long-range video modeling. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2021. 8

[32] Yunbo Wang, Mingsheng Long, Jianmin Wang, and Philip S Yu. Spatiotemporal pyramid network for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[33] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 8

[34] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3

[35] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[36] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 2

[37] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *IEEE Transactions on Image Processing*, 29:5491–5506, 2020. 3

[38] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision*, 2018. 3

[39] Tao Zhuo, Zhiyong Cheng, Peng Zhang, Yongkang Wong, and Mohan Kankanhalli. Explainable video action reasoning via prior knowledge and state transitions. In *Proceedings of the ACM International Conference on Multimedia*, 2019. 3