Object Affordances Graph Network for Action Recognition

Haoliang Tan[1]
tan792507667@stu.xjtu.edu.cn
Le Wang*[1]
lewang@xjtu.edu.cn
Qilin Zhang[2]
samqzhang@gmail.com
Zhanning Gao[3]
zhanninggao@gmail.com
Nanning Zheng[1]
nnzheng@mail.xjtu.edu.cn
Gang Hua[4]
ganghua@gmail.com

[1] Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University, China
[2] HERE Technologies, USA
[3] Alibaba Group, China
[4] Wormpex AI Research, USA

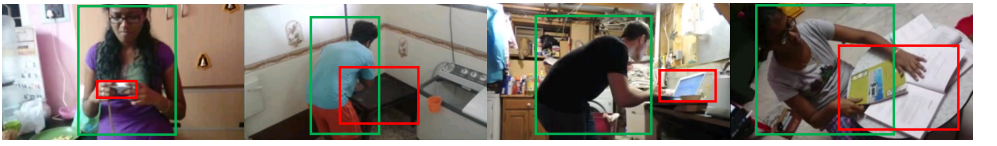
Abstract

Human actions often involve interactions with objects, and such action possibilities of objects were termed "affordances" in human-computer interaction (HCI) literature. To facilitate action recognition with object affordances, we propose the Object Affordances Graph (OAG), which cast human-object interaction cues into video representations via an iterative refinement procedure. With the spatio-temporal co-occurrences between human and objects captured, the Object Affordances Graph Network (OAGN) is subsequently proposed. To provide a fair evaluation of the role that object affordances could play on human action recognition, we have assembled a new dataset with additional annotated object bounding-boxes to account for human-object interactions. Multiple experiments on this proposed Object-Charades dataset verify the value of including object affordances in human action recognition, specifically via the proposed OAGN, which outperforms existing state-of-the-art affordance-less action recognition methods.

1 Introduction

Human action recognition [3, 8, 24, 39] is an increasingly popular topic in computer vision, especially with recent more challenging datasets [12, 30] with untrimmed videos, much more action categories and dynamic (*e.g.,* often cluttered) backgrounds. Instead of treating background objects purely as nuisances, we speculate that it could be beneficial to exploit object affordances (including background scenes) to guide action recognition. A few examples are illustrated in Figure 1, where people and their interacting objects are annotated with green and red bounding-boxes, respectively. Unsurprisingly, we can approximately

*Corresponding author



(a) Someone is playing with phone (b) Someone is washing a table (c) Someone is working on laptop (d) Someone is opening a book

Figure 1: Action possibilities of objects (*i.e.*, object affordances, or human-object interactions) could provide helpful hints in determining the action category.

deduce the action categories purely based on the occurrence of such objects. For example, the presence of multiple books (marked by a red bounding-box) in Figure 1(d) strongly indicates human actions such as reading/writing/opening/closing a book/books. Such observations motivate us to leverage the object affordances to assist the action recognition.

Typical existing affordance-less action recognition methods include two-stream convolution networks [33], 3D conventional neural networks (3D CNNs) [3, 6, 15], and long short-term memory (LSTM)-based recurrent neural networks (RNNs) [21, 25, 43]. Among them, Wang *et al*. [37] leveraged spatio-temporal object auxiliary cues with a temporal graph by densely sampling the region of interest in video frames to exploit temporal dependencies. However, such temporal graph model is derived solely from objects rather than spatio-temporal correlations between human and objects, *i.e.*, human-object interactions.

Inspired by the idea of object affordance-assisted action recognition, we propose the Object Affordances Graph (OAG) to exploit the spatio-temporal relationships between human and objects/scenes across all video frames. Specifically, OAG explicitly casts spatio-temporal human-object relationships as graph nodes, which are represented by feature maps extracted inside bounding-boxes containing human, interacting objects and the union of the them, respectively. The OAG iteratively updates each graph node to cast the human-object interaction cues into the video representation. In addition, the OAG seamlessly integrates with the CNN backbone, making end-to-end training of our proposed Object Affordances Graph Network (OAGN) possible.

To provide a fair evaluation of how much object affordances could contribute to action recognition, we propose a new dataset with annotated object bounding-boxes as additional ground truth, based on the belief that automatic object detection performance should be decoupled. This dataset is obtained by first applying Deformable R-FCN [5, 19] object detector pre-trained on MS COCO [23] dataset on the Charades dataset [30] followed by a manual review and repair step. During this manual step, low quality and false detections are removed. The obtained subset of Charades dataset together with the additional object annotations is named as the Object-Charades dataset. Experimental results on this Object-Charades dataset show that our proposed OAGN outperforms existing affordance-less baselines with less than 1% parameter overhead. These results validate the value of spatio-temporal human-object interaction cues in action recognition and the efficacy of the proposed OAGN.

Overall, the primary contributions of this paper are summarized as follows.

- Verification of the performance benefits from including spatio-temporal human-object interaction cues (*i.e.*, object affordances) in human action recognition;
- A customized Object-Charades dataset with annotations of humans and objects as additional ground truth, which resolves the dependency on automatic object detection performance;

- The incorporation of spatio-temporal human-object interaction cues as object affordances in action recognition via the proposed OAG, which aggregates both long-range spatial and long-term temporal contexts around human and objects across video frames;
- The subsequent graph-based convolution network OAGN that can be conveniently trained end-to-end.

2 Related Work

Action Recognition. With recent human action datasets [12, 20, 30, 35], deep neural network-based action recognition methods have been actively developed in recent years. Simonyan and Zisserman [33] propose the two-stream model, with multiple later variants [7, 9, 36]. 3D CNNs such as I3D [3], 3D ResNet [15], SlowFast [6], and NL I3D [38] achieve improved recognition performance. However, such existing action recognition algorithms lack explicit treatment of spatio-temporal human-object interaction cues (*i.e.*, object affordances).

Graph Neural Networks (GNNs) receive much attention recently, thanks to its ability to represent complex interdependencies [1, 2]. Herzigo *et al.* [18] propose a spatio-temporal action graph to detect driving collisions. Li *et al*. [22] propose a recurrent update scheme for the hidden state nodes for situation recognition. Yan *et al*. [39] and Si *et al*. [29] incorporate GNNs in skeleton-based action recognition. Wang *et al.* [37] combine CNNs with GNNs and exploit spatio-temporal object auxiliary cues with a temporal graph by densely sampling the region of interest in video frames to exploit temporal dependencies. However, this temporal graph accounts solely for objects rather than human-object interactions.

Human-Object Interaction (HOI) detection tasks localize human and objects with bounding-boxes and identify their interacting action [4]. Many existing methods combine both the HOI detection and human action recognition task [13, 41]. Specifically, Gkioxari *et al*. [11] propose the InteractNet to infer human-object interactions in an end-to-end manner with a modified Faster R-CNN [27]. Gupta *et al*. [14] combine human pose key points and handcrafted features for HOI detection. Qi *et al*. [26] adopt message passing over GNN-based approach. The aforementioned methods assign action labels to both the human and the object detections without explicit formulation of human-object interactions as a dedicated node in GNNs.

3 Object Affordances Graph Network

We design the OAGN to incorporate human-object interaction cues into the video representation via the OAG update iterations (as shown in Figure 2). The OAGN takes video frames and human/object bounding-boxes as inputs. We utilize various 2D/3D CNNs to extract video features (denoted as the orange-brown "Conv" layers in Figure 2) and associate them with corresponding bounding-boxes with the RoIAlign algorithm [10, 17] on each feature frame. Each human bounding-box (*e.g.*, yellow box) and each object bounding-box (*e.g.*, red and blue boxes) contribute to an individual node in the OAG, where such nodes are initialized with features corresponding to their respective bounding-boxes. Additionally, one additional node (denoted as the solid green circle in the OAG step of Figure 2) is included in each OAG, representing the abstract concept of "human-object interaction", which is conveniently initialized with the feature corresponding to the dotted green bounding-box, *i.e.*, the tight bounding-box containing both the human box (yellow) and the interacting object box

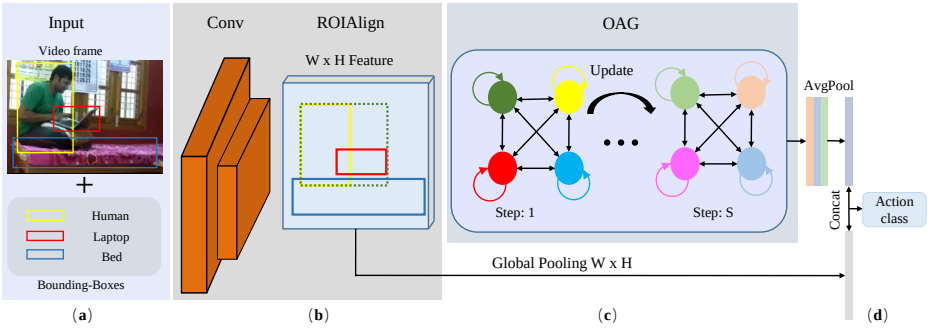


Figure 2: OAGN overview. **(a)** OAGN takes video frames and ground truth human/objects bounding-boxes as inputs. **(b)** RoIAlign [10, 17, 20] produces region-based features according to the bounding-boxes (including the union of bounding-boxes). **(c)** Graph nodes are iteratively updated in S steps, and they are initialized with extracted feature maps within respective bounding-boxes. **(d)** After practical convergence, updated nodes and video features are concatenated for action prediction.

Table 1: Parameter settings of the modified ResNet-50 [16], one option of the backbone network. The resolution of input video frame is 224×224. To preserve more spatial information in feature maps, the stride of res_5 is modified from 2 to 1.

layer	conv_1	pool_1	res_2	res_3	res_4	res_5	conv_6
config	7×7,64	3×3 max	$\begin{bmatrix} 1\times1, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix}$	$\begin{bmatrix} 1\times1, & 128 \\ 3\times3, & 128 \\ 1\times1, & 512 \end{bmatrix}$	$\begin{bmatrix} 1\times1, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix}$	$\begin{bmatrix} 1\times1, & 512 \\ 3\times3, & 512 \\ 1\times1, & 2048 \end{bmatrix}$	1×1, 512
	stride 2,2	stride 2,2	×3, stride 1	×4, stride 2	×6, stride 2	×3,stride 1	stride 1
output size	112×112	56×56	56×56	28×28	14×14	14×14	14×14

(red). Therefore, three types of nodes are present in each OAG, *i.e.*,: :q the human node(s), the object nodes and the human-object interactions nodes.

Subsequently, the OAG starts iterative graph updates (visualized as S steps in Figure 2). Although these node features are extracted from different regions and at various time stamps, we speculate that the OAG nevertheless captures long-term spatio-temporal human-object interaction cues through the message propagation mechanism. After practical convergence at the final step S, we perform an average pooling over all node features and over all video features, respectively. These two features are subsequently concatenated as the final video representation for action category prediction.

Feature extraction. Given a video clip $V = \{v_1, v_2, ..., v_T\}$ with T sampled frames*, all frames are first resized to a common resolution of 224×224 (unless otherwise specified) via bilinear interpolation, subsequently these frames are fed into one of the modified backbone CNNs, *e.g.*, a slightly modified version of ResNet-50 [16] as summarized in Table 1. These modifications include another convolution layer appended after the last residue block to restrict the number of feature channels to a predefined value† C, and a smaller stride in the convolution kernel in the last residual block to preserve higher feature map resolution.

Graph Initialization. Suppose N_t bounding-boxes‡ are present in frame v_t, $t \in \{1, 2, ..., T\}$,

*Typically one frame is sampled per six consecutive video frames.

†To reduce the number of parameters and alleviate computational complexity, typically $C = 512$.

‡Including human, objects bounding boxes and the special "human-object interaction" bounding-box.

therefore there are $N = \sum_{t=1}^{T} N_t$ bounding-boxes for video clip V. We apply RoIAlign [10, 17, 40] on each bounding-box in all feature frames, generating $7 \times 7 \times C$ features per bounding-boxes, and subsequently max-pooled to $1 \times 1 \times C$. These C-dimensional tensors are represented as the initial nodes in the graph. Let $\mathbf{X}^s \in \mathbb{R}^{N \times C}$ denote the OAG state (*i.e.*, values of all N nodes) at s-th iteration, $s \in \{0, 1, \cdots, S\}$. Each row of OAG is initialed with these C-dimensional vectors. For notational simplicity, we partition the initial state $\mathbf{X}^0 \in \mathbb{R}^{N \times C}$ into T blocks, with each block $\mathbf{X}_t^0 \in \mathbb{R}^{N_t \times C}$ representing vectors from the t-th frame, $t \in \{1, 2, ..., T\}$,

$$\mathbf{X}^0 = \begin{bmatrix} \mathbf{X}_1^0 \\ \vdots \\ \mathbf{X}_T^0 \end{bmatrix}_{N \times C}, \text{ where } \mathbf{X}_t^0 = \begin{bmatrix} \boldsymbol{x}_{t,1}^0 \\ \vdots \\ \boldsymbol{x}_{t,N_t}^0 \end{bmatrix}_{N_t \times C}, \tag{1}$$

where $\boldsymbol{x}_{t,N_t}^0 \in \mathbb{R}^{1 \times C}$ denotes the initial value of the node corresponding to the N_t-th bounding-box in frame t. Specifically with 3D CNNs, a fixed number T of frames are grouped together as a "video clip" (*e.g.*, $T = 16$ for I3D [3]) to compute features, which leads to reduced temporal resolution (T' channels, $T' < T$, *e.g.*, $T' = 2$ in I3D [3]) in extracted features. We conveniently use the bounding-boxes annotated in the frames $\left\{ \lceil \frac{T}{2T'} \rceil, \lceil \frac{T}{2T'} \rceil + \frac{T}{T'}, \cdots, T - \lfloor \frac{T}{2T'} \rfloor \right\}$ for the above initialization (*e.g.*, frame 4 and frame 12 for I3D [3]).

Graph Updating Strategy. During the graph iterations, we calculate the correlation score between each pair of nodes to generate a correlation matrix, based on which we calculate the incoming messages for each node. Subsequently, each note state is updated according to its respective incoming messages. The correlation score f between nodes $\boldsymbol{x}_{i,k}$ and $\boldsymbol{x}_{j,g}$ at the $(s-1)$-th iteration is defined as,

$$f\left(\boldsymbol{x}_{i,k}^{s-1}, \boldsymbol{x}_{j,g}^{s-1}\right) = \left(\boldsymbol{\Theta} \boldsymbol{x}_{i,k}^{s-1} + \boldsymbol{b}_{\boldsymbol{\theta}}\right)^T \left(\boldsymbol{\Phi} \boldsymbol{x}_{j,g}^{s-1} + \boldsymbol{b}_{\boldsymbol{\phi}}\right), \tag{2}$$

where $i, j \in \{1, \cdots, T\}$, $k \in \{1, \cdots, N_i\}$, $g \in \{1, \cdots, N_j\}$, $\boldsymbol{\Theta}, \boldsymbol{\Phi} \in \mathbb{R}^{C \times C}$, $\boldsymbol{b}_{\boldsymbol{\theta}}, \boldsymbol{b}_{\boldsymbol{\phi}} \in \mathbb{R}^{C \times 1}$. Parameters $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, $\boldsymbol{b}_{\boldsymbol{\theta}}$, and $\boldsymbol{b}_{\boldsymbol{\phi}}$ are shared among all graph nodes, and they are initiated by sampling a Gaussian distribution with zero mean and a standard deviation of 0.001. At the end of the graph iteration (*i.e.*, after iteration S), $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, $\boldsymbol{b}_{\boldsymbol{\theta}}$, and $\boldsymbol{b}_{\boldsymbol{\phi}}$ are jointly updated with the backbone network (*e.g.*, ResNet-50 [16]) via back-propagation. Therefore, the correlation matrix for video clip V at iteration $(s-1)$ is,

$$\mathbf{F}(\mathbf{X}^{s-1}) = \begin{bmatrix} f\left(\boldsymbol{x}_{1,1}^{s-1}, \boldsymbol{x}_{1,1}^{s-1}\right) & \cdots & f\left(\boldsymbol{x}_{1,1}^{s}, \boldsymbol{x}_{T,N_T}^{s-1}\right) \\ \vdots & \ddots & \vdots \\ f\left(\boldsymbol{x}_{T,N_T}^{s-1}, \boldsymbol{x}_{1,1}^{s-1}\right) & \cdots & f\left(\boldsymbol{x}_{T,N_T}^{s-1}, \boldsymbol{x}_{T,N_T}^{s-1}\right) \end{bmatrix}_{N \times N}. \tag{3}$$

Additionally, we exploit the softmax function to normalize the correlation matrix to ensure all its rows summing to 1. Take the node $\boldsymbol{x}_{i,k}$ as an example, $\forall s \in \{1, 2, \cdots, S\}$,

$$\sum_{n=1}^{T} \sum_{m=1}^{N_n} f_{\text{norm}}\left(\boldsymbol{x}_{i,k}^{s}, \boldsymbol{x}_{n,m}^{s}\right) = 1, \tag{4}$$

where $i \in \{1, ..., T\}$, $k \in \{1, ..., N_i\}$. Subsequently, we use this normalized correlation matrix $\mathbf{F}_{\text{norm}}(\mathbf{X}^{s-1})$ to calculate incoming messages and the graph updates,

$$\mathbf{X}^s = \mathbf{X}^{s-1} + \mathbf{F}_{\text{norm}}(\mathbf{X}^{s-1}) \mathbf{X}^{s-1}. \tag{5}$$

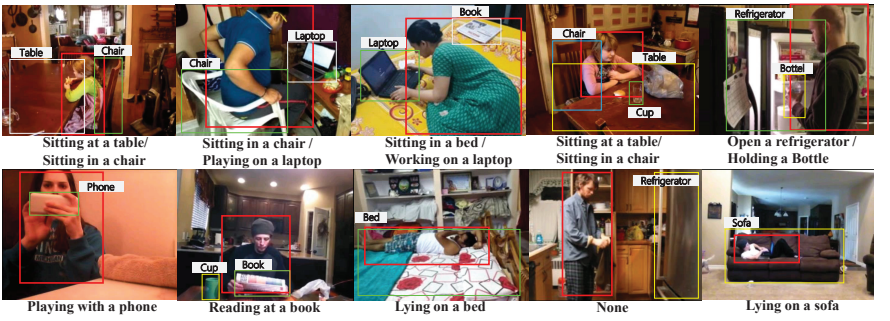


Figure 3: Samples from the Object-Charades dataset. Highlights: (1) all action instances in the dataset involve human-object interaction; (2) multiple different action instances can be present simultaneously; (3) human and object bounding-boxes are manually adjusted to be as tight as possible.

After iteration S, we obtain the updated node state $\mathbf{X}^S \in \mathbb{R}^{N\times C}$.

Video Representation Generation is achieved by merging global video-clip representations with the OAG outputs. As shown in Figure 2, we perform an average pooling on OAG outputs $\mathbf{X}^S$ and obtain $\mathbf{r}_{\text{OAG}} \in \mathcal{R}^{1\times C}$ as $\mathbf{r}_{\text{OAG}} = \mathbf{1}_{1\times N}\mathbf{X}^S/N$, where $\mathbf{1}_{1\times N}$ denote an all-ones vector of size $1\times N$. Simultaneously, the global video-clip representation are average pooled to $\mathbf{r}_{\text{Global}} \in \mathcal{R}^{1\times C}$. We concatenate them to obtain the final video representation $\mathbf{r} \in \mathcal{R}^{1\times 2C}$ for the final action prediction, $\mathbf{r} = [\mathbf{r}_{\text{OAG}}, \mathbf{r}_{\text{Global}}]$.

4 The Object-Charades Dataset

The Object-Charades dataset is a subset of the Charades dataset [30], where all action instances involve human-object interaction, and humans and their interacting objects are annotated with bounding-boxes. We briefly describe the Charades dataset and explain the Object-Charades dataset in detail.

Charades dataset is a large-scale multi-label video dataset focusing on household activities [30]. It contains 157 categories of actions and ~9K videos of about 30 seconds in length on average. Most of the action instances in the Charades dataset involve human-object interactions.

Motivation to introduce a new dataset. As our goal is to leverage the human-object interaction cues to refine the video representation and further improve action recognition accuracy. The original Charades dataset has several characteristics making it less suitable for our goal: **1)** It focuses on action recognition rather than the interaction between the human and the object for each action instance. To evaluate the influence of human-object interaction on action recognition, we purposefully select action instances with obvious interactions with objects and leave out those without. **2)** It does not provide object-level annotations. To evaluate the influence of human-object interactions on action recognition, manually annotated object-level labels should also be provided.

Bounding-box annotations. To annotate the objects involved in action instances, we first apply the Deformable R-FCN object detector [5, 19] pre-trained on MSCOCO [23] dataset on all video frames of the Charades dataset, then manually examine the results and preserve high-quality boungding-boxes. We also discard the actions with limited interacting objects

(*e.g.* floor and pictures). We believe this dataset should be more suitable for evaluating the benefit of considering human-object interaction in action recognition.

Object-Charades dataset. The Object-Charades dataset consists of 7,135 videos (5,732 training and 1,432 testing) of 52 action categories, all action instances of which involve humans interacting with objects of 18 categories. Humans and interacting objects are annotated with bounding-boxes *(x1,x2,y1,y2,class)*. Other configurations of the Object-Charades dataset are the same as those of the Charades dataset. Note that because Yuan *et al.* [42] provided bounding-box annotations for more than 5,000 video frames from only 200 videos of the test set of the Charades dataset, and thus it is not enough to evaluate our proposed OAGN method.

Evaluation metrics. Object-Charades is a multi-label video action dataset. Action classification performance is evaluated by mean average precision (mAP) metric.

5 Experiments

We conduct multiple experiments with both 2D CNNs [16] and 3D CNNs [15] as backbone networks and empirically prove that OAG can aggregate human-object interactions cues across global spatio-temporal dimensions. In addition, with larger backbone networks our OAGN achieves the state-of-the-art performance.

5.1 Implementation Details

Data pre-processing. We omit random crop, flip and other data augmentations due to the incorporation of ROIAlign into OAGN. All video frames are resized to a common resolution of 224×224 unless otherwise specified. For algorithms with 3D CNNs, we sample the video clip $V = \{v_1, v_2, \cdots, v_T\}$ so that $T = 16$, v_{i+1} and v_i are 6 video frames apart.

Backbone network. 2D ResNet [16] pre-trained on ImageNet [28] and 3D ResNet [15] pre-trained on Kinetics [3] are used as the base feature extraction networks. We restrict the number of feature channels to $C = 512$ and sets the spatial scale to 1 for the additional convolutional layer.

Graph initialization. RoIAlign is applied to the feature from the last convolution layer. Center crop is employed to generate virtual nodes for frames without any objects. A node has a dimension of 1×512 ($7\times7\times512$ via ROIAlign and $1\times1\times512$ via pooling).

Training. OAGN is implemented using PyTorch and trained on two Nvidia 1080Ti GPUs with a batch size of 50 for 2D CNNs or 20 for 3D CNNs. After nodes reach practical convergence, we perform an average pooling over all node features and over all video features, respectively, followed by a dropout with a ratio of 0.5. These two features are subsequently concatenated as the final video representation for action category prediction. We use sigmoid-based classifier for multi-label prediction. The initial learning rate is 0.08 with 2D CNN backbones and 0.12 with 3D CNNs annealed by 0.8 every 3K iterations. We apply a stage-wise strategy to train OAGN. We first train the OAG for 6K iterations with the parameters in the backbone networks fixed, then the entire OAGN end-to-end for another 10K iterations. For 3D networks, each batch contains 16 frames.

Inference. For each test video 50 clips or frames are sampled. The final mAP is acquired through max pooling over the predictions of these 50 samples.

Parameter analysis. To explore the impact of the number of iterations S used to update the OAG on the overall performance, we plug the OAG to fixed 2D and 3D backbone networks

respectively and set S to 1, 2 and 3. mAPs under different settings are shown in Figure 4. Our observations include: the number of iterations has no significant impact on the recognition accuracy; as the number of iterations increases, there is no substantial improvement on recognition accuracy; graph nodes learn sufficient spatio-temporal correlation context cue and converge in 1 iteration. Based on our observations, we set S to 1 in the following experiments for computation efficiency.

Training with larger backbones. To examine if OAGN can function properly with other CNNs and achieve better results with larger backbones, we substitute the default backbones (2D ResNet-50 and 3D ResNet-50) with 2D ResNet-101 [16] pre-trained on ImageNet [28] and I3D [3] pre-trained on Kinetics [3]. Experiment settings remain the same except for 20K more iterations for training. Results are shown in Table 2. We observe a considerable improvement of action recognition accuracy with larger backbones in place (28.6 mAP with ResNet-101 and 32.1 mAP with I3D).

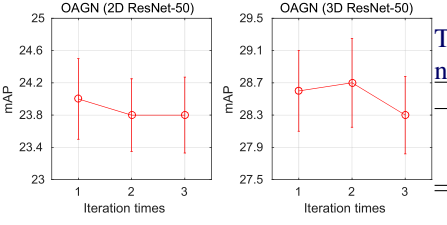


Figure 4: Impacts of the number of graph iterations on action recognition accuracy.

Table 2: OAGN with larger 2D and 3D networks.

Conv	Model/Method	input	mAP
2D	ResNet-101 [16]	1×224×224	26.8
	OAGN(ResNet-101)		27.9
	OAGN(ResNet-101)	3×224×224	**28.6**
3D	I3D [3]	16×224×224	30.5
	OAGN (I3D)		**32.1**

Qualitative and error analysis. We visualize some sample video frames and their corresponding global video-clip representations in Figure 5. These global video-clip representations are the outputs of the backbone network after layer "conv6" and are of size $512 \times 14 \times 14$. A max-pooling is carried out along the channel dimension and the 14×14 heat maps are illustrated. The highlighted areas usually contain human and their interacting objects, which provide some intuition behind the OAGN design of focusing on such regions. In Figure 6, we illustrate the action AP improvements/degradation of the OAGN (ResNet-50) over the baseline ResNet-50 in 20 action categories. The top-10 AP improvement categories and top-10 AP degradation categories are visualized in green and red colour, respectively. Figure 6 shows that the OAGN brings larger performance gains than deteriorations.

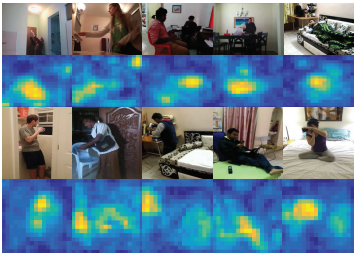


Figure 5: Global video-clip representation visualization, extracted after layer "conv6" of backbone network.

Figure 6: Action AP gains versus deteriorations of the OAGN(ResNet-50) over the baseline ResNet-50.

Table 3: Action classification performance on Object-Charades dataset in mAP(%) with (a) 2D CNNs as backbone network; (b) 3D CNNs as backbone network.

2D Conv Model	input	mAP
VGG-16 [34]	1×224×224	16.2
ResNet-50 [16]		22.0
Asyn-TF [31]		22.6
OAGN (ResNet-50)		24.5
OAGN (ResNet-101)	3×224×224	**28.6**

(a)

3D Conv Model	input	mAP
ResNet-50 [15]	16×224×224	26.8
ResNet-101 [15]		28.0
I3D [3]		30.5
OAGN(I3D)		**32.1**

(b)

Table 4: mAP(%) for action classification. (**a**) Using different part of OAGN. (**b**) Different number (T) of video frame as OAGN input.

	Method	mAP
2D 1× 224×224	Avg-Pooling	22.0
	w/ OAG	23.7
	Avg-pooling + OAG (OAGN)	**24.5**
3D 16× 224×224	Avg-Pooling	26.8
	w/ OAG	28.5
	Avg-pooling + OAG (OAGN)	**29.2**

(a)

Model	input	mAP
2D ResNet-50 [16]	1×224×224	22.0
OAGN (ResNet-50)	1×224×224	24.5
OAGN (ResNet-50)	3×224×224	**25.1**
3D ResNet-50 [15]	16×224×224	26.8
OAGN (ResNet-50)	16×224×224	**29.2**

(b)

5.2 Comparison with State-of-the-arts

VGG-16 [34]: VGG-16 is first pre-trained on ImageNet and subsequently fine-tuned. ***Asynchronous Temporal Fields (Asyn-TF)*** [31, 32]: Popular in action classification, Asyn-TF is utilized here with the Resnet-50 as its backbone network. ***3D ResNet-101*** [15]: 3D ResNet-101 is first pre-trained on Kinetics [3] with video clips of size $16 \times 112 \times 112$ and subsequently fine-tuned. During training, a dropout ratio of 0.5 is used.

All results are summarized in Table 3. Possibly due to the lack of explicit data argumentation, the reported accuracies are relatively lower. VGG-16, Asyn-TF and 3D ResNet-101 achieve mAP of 16.2, 22.6 and 28.0, respectively. Despite with fewer parameters, all OAGN varaints outperform their respective competing algorithms.

5.3 Ablation Study

To isolate the contributions of different components of the proposed OAGN, we conduct multiple ablation experiments.

Effectiveness of OAG. The results are shown in Table 4 (a). Compared with baseline "Avg-Pooling", the pure output of OAG can get higher action recognition accuracy. In addition, after combining the global video-clip representations with OAG outputs, the final action recognition accuracy can be further improved with both 2D and 3D backbone networks.

Effectiveness of joint aggregation of spatio-temporal cues. We first adopt 2D CNNs as backbones and the results are shown in Table 4 (b). By taking a single frame ($T = 1$) as input, the 2D ResNet-50 achieves 20.0 mAP and the OAGN gets 12.5% relative performance gains, indicating that the OAG can aggregate spatial human-object interaction context cues. With multiple video frames as inputs ($T = 3$), OAGN performance is further improved, indicating long-term temporal correlation cues can be captured. We also evaluate with 3D CNNs, where our OAGN gets 8.2% relative accuracy improvements.

Discussion. The performance gain shows that the OAG is effective with 2D or 3D backbones. Compared with the backbone networks, OAGN brings $\sim$2 gain in mAP with less than 1% ($\sim$0.5M) extra parameters. Possible reasons are: 1) because the nodes are initialized by extracting features from different regions across multiple video frames, the graph is able to reason the interactions between humans and objects across all spatio-temporal dimensions, which means that the output of the graph could carry global spatio-temporal correlation cues; 2) the OAG is capable of representing humans, objects, and their relationships with accurately annotated human and object bounding-boxes.

6 Conclusion.

In this paper, we propose OAGN, a graph-based convolutional network, for human action recognition. OAGN incorporates spatio-temporal human-object interaction cues as object affordances in action recognition by aggregating long-range spatial and long-term temporal context around humans and objects across video frames. We demonstrate that the proposed OAGN outperforms existing state-of-the-arts on our proposed dataset Object-Charades dataset, of which all action instances involve interaction with objects, and requires fewer parameters. We believe the experiments carried out on the OAGN reveal the importance of exploiting human-object interactions for action recognition.

Acknowledgments. This work was supported partly by National Key R&D Program of China Grant 2017YFA0700800, NSFC Grants 61629301 and 61773312, China Postdoctoral Science Foundation Grant 2019M653642, and Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001.

References

[1] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018.

[2] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

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

[4] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *WACV*, 2018.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv: 1812.03982*.

[7] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016.

[8] Zhanning Gao, Le Wang, Qilin Zhang, Zhenxing Niu, Nanning Zheng, and Gang Hua. Video imprint segmentation for temporal action detection in untrimmed videos. In *AAAI*, 2019.

[9] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.

[10] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

[11] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *CVPR*, 2018.

[12] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.

[13] Abhinav Gupta, Aniruddha Kembhavi, and Larry S Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *T-PAMI*, 31 (10):1775–1789, 2009.

[14] Tanmay Gupta, Alexander Schwing, and Derek Hoiem. No-frills human-object interaction detection: Factorization, appearance and layout encodings, and training techniques. *arXiv preprint arXiv:1811.05967*, 2018.

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, 2017.

[18] Roei Herzig, Elad Levi, Huijuan Xu, Eli Brosh, Amir Globerson, and Trevor Darrell. Classifying collisions with spatio-temporal action graph networks. *arXiv preprint arXiv:1812.01233*, 2018.

[19] Kaiming He Jian Sun Jifeng Dai, Yi Li. R-FCN: Object detection via region-based fully convolutional networks. In *NIPS*, 2016.

[20] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[21] Fu Li, Chuang Gan, Xiao Liu, Yunlong Bian, Xiang Long, Yandong Li, Zhichao Li, Jie Zhou, and Shilei Wen. Temporal modeling approaches for large-scale youtube-8m video understanding. *arXiv preprint arXiv:1707.04555*, 2017.

[22] Ruiyu Li, Makarand Tapaswi, Renjie Liao, Jiaya Jia, Raquel Urtasun, and Sanja Fidler. Situation recognition with graph neural networks. In *ICCV*, 2017.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[24] Ziyi Liu, Le Wang, Gang Hua, Qilin Zhang, Zhenxing Niu, Ying Wu, and Nanning Zheng. Joint video object discovery and segmentation by coupled dynamic markov networks. *T-IP*, 27(12):5840–5853, Dec 2018.

[25] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. In *CVPRW*, 2017.

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

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[29] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *ECCV*, 2018.

[30] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.

[31] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, 2017.

[32] Gunnar A. Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Pytorch starter code for activity classification and localization on charades. https://github.com/gsig/temporal-fields/tree/master/pytorch, 2017.

[33] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[36] 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*, 2016.

[37] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

[38] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

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

[40] Jianwei Yang, Jiasen Lu, Dhruv Batra, and Devi Parikh. A faster pytorch implementation of faster r-cnn. https://github.com/jwyang/faster-rcnn.pytorch.

[41] Bangpeng Yao and Li Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *T-PAMI*, 34(9):1691–1703, 2012.

[42] Yuan Yuan, Xiaodan Liang, Xiaolong Wang, Dit-Yan Yeung, and Abhinav Gupta. Temporal dynamic graph lstm for action-driven video object detection. In *ICCV*, 2017.

[43] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.