Adaptive Interaction Modeling via Graph Operations Search

Haoxin Li[1], Wei-Shi Zheng[2,3,5,*], Yu Tao[2,4], Haifeng Hu[1,*], Jian-Huang Lai[2]
[1]School of Electronics and Information Technology, Sun Yat-sen University, China
[2]School of Data and Computer Science, Sun Yat-sen University, China
[3]Peng Cheng Laboratory, Shenzhen 518005, China
[4]Accuvision Technology Co. Ltd.
[5]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
lihaoxin05@gmail.com, wszheng@ieee.org, gytaoyu@hotmail.com
huhaif@mail.sysu.edu.cn, stsljh@mail.sysu.edu.cn

Abstract

Interaction modeling is important for video action analysis. Recently, several works design specific structures to model interactions in videos. However, their structures are manually designed and non-adaptive, which require structures design efforts and more importantly could not model interactions adaptively. In this paper, we automate the process of structures design to learn adaptive structures for interaction modeling. We propose to search the network structures with differentiable architecture search mechanism, which learns to construct adaptive structures for different videos to facilitate adaptive interaction modeling. To this end, we first design the search space with several basic graph operations that explicitly capture different relations in videos. We experimentally demonstrate that our architecture search framework learns to construct adaptive interaction modeling structures, which provides more understanding about the relations between the structures and some interaction characteristics, and also releases the requirement of structures design efforts. Additionally, we show that the designed basic graph operations in the search space are able to model different interactions in videos. The experiments on two interaction datasets show that our method achieves competitive performance with state-of-the-arts.

1. Introduction

Video classification is one of the basic research topics in computer vision. Existing video classification solutions can be mainly divided into two groups. The first one is the two-stream network based methods [28, 33, 8], which model appearance and motion features with RGB and optical flow streams respectively; the second type is the 3D convolution neural networks (CNN) based methods [29, 4, 32, 26, 31, 23], which model spatiotemporal features with stacked 3D convolutions or the decomposed variants. While these methods work well on scene-based action classification, most of them obtain unsatisfactory performance on recognizing interactions, since they haven't effectively or explicitly modeled the relations.

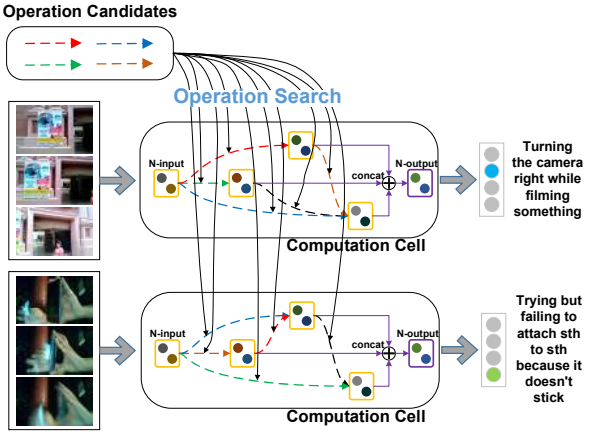


Figure 1. Illustration of our method. We search adaptive network structures to model the interactions in different videos, in which the candidate basic operations (dashed arrows) are selected (solid arrows) to construct adaptive structures for different videos.

To model the interactions in videos, some methods employ specific structures [40, 14, 16] to capture temporal relations. Others model the relations between entities. Non-local network [34] and GloRe [7] design networks with self-attention and graph convolution to reason about the relations between semantic entities. CPNet [22] aggregates features from potential correspondences for representation learning. Space-time region graphs [35] are developed to model the interactions between detected objects with graph

*Corresponding author

convolution network (GCN).

However, existing methods have to manually design network structures for interaction modeling, which requires considerable architecture engineering efforts. More importantly, the designed structures are fixed so that they could not adaptively model different interactions. For example, the two videos in Figure 1 contain the interactions with greatly different complexities and properties, *i.e*. the upper one mainly concerns the motions of the background while the lower one involves complicated relations among objects, where which kind of structures should be used to adequately model the interactions is not completely known in advance, so that it requires to construct adaptive structures for more effective interactions modeling.

Instead of designing fixed network structures manually, we propose to automatically search adaptive network structures directly from training data, which not only reduces structures design efforts but also enables adaptive interaction modeling for different videos. As briefly illustrated in Figure 1, different operations are adaptively selected to construct the network structures for adaptive interaction modeling for different videos, which is implemented by differentiable architecture search. To construct the architecture search space, we first design several basic graph operations which explicitly capture different relations in videos, such as the temporal changes of objects and relations with the background. Our experiments show that the architecture search framework automatically constructs adaptive network structures for different videos according to some interaction characteristics, and the designed graph operations in the search space explicitly model different relations in videos. Our method obtains competitive performance with state-of-the-arts in two interaction recognition datasets.

In summary, the contribution of this paper is two-fold. (1) We propose to automatically search adaptive network structures for different videos for interaction modeling, which enables adaptive interaction modeling for different videos and reduces structures design efforts. (2) We design the search space with several basic graph operations, which explicitly model different relations in videos.

2. Related Work

2.1. Action and Interaction Recognition

In the deep learning era, action recognition obtains impressive improvements with 2D [28, 33, 8] or 3D [15, 29, 26, 4, 32, 31, 23] CNNs. 2D CNNs use RGB frames and optical flows as separate streams to learn appearance and motion representations respectively, while 3D CNNs learn spatiotemporal features with 3D convolutions or the decomposed counterparts. Some other works [19, 16] learn spatiotemporal representations by shifting feature channels or encoding motion features together with spatiotemporal features, which achieve high performance and efficiency. As for temporal-based actions, TRN [40] and Timeception [14] design specific structures to model the temporal relations.

To model interactions, Gupta *et al*. [11] apply spatial and functional constraints with several integrated tasks to recognize interactions. InteractNet [9] and Dual Attention Network [37] are proposed to model the interactions between human and objects. Some other works model the relations between entities for interaction recognition. Non-local network [34] models the relations between features with self-attention. CPNet [22] aggregates correspondences for representation learning. GCNs are employed to model the interactions between nodes [35, 7]. These specific structures in the above methods are non-adaptive. In practice, however, we do not know what kinds of interactions are contained in videos, and the non-adaptive structures could not sufficiently model various interactions, which requires adaptive structures for effective modeling.

In this work, we propose to automatically search adaptive network structures with differentiable architecture search mechanism for interaction recognition.

2.2. Graph-based Reasoning

Graph-based methods are widely used for relation reasoning in many computer vision tasks. For example, in image segmentation, CRFs and random walk networks are used to model the relations between pixels [5, 3, 18]. GCNs [12, 17] are proposed to collectively aggregate information from graph structures and applied in many tasks including neural machine translation, relation extraction and image classification [1, 2, 25, 36]. Recently, GCNs are used to model the relations between objects or regions for interaction recognition. For example, Chen *et al*. [7] adopt GCN to build a reasoning module to model the relations between semantic nodes, and Wang *et al*. [35] employ GCN to capture the relations between detected objects.

In this paper, we design the search space with basic operations based on graph. We propose several new graph operations that explicitly model different relations in videos.

2.3. Network Architecture Search

Network architecture search aims to discover optimal architectures automatically. The automatically searched architectures obtain competitive performance in many tasks [42, 20, 43]. Due to the computational demanding of the discrete domain optimization [43, 27], Liu *et al*. [21] propose DARTS which relaxes the search space to be continuous and optimizes the architecture by gradient descent.

Inspired by DARTS, we employ differentiable architecture search mechanism to automatically search adaptive structures directly from training data, which facilitates adaptive interaction modeling for different videos and releases the requirement of structures design efforts.

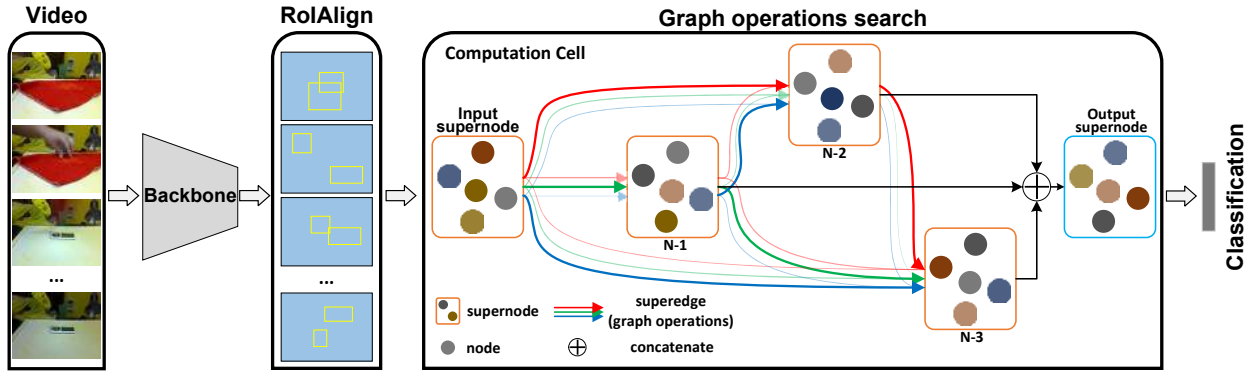


Figure 2. **Overall framework**. Some frames are sampled from a video as the input to our model. We extract basic features of the sampled frames with a backbone CNN, and extract class-agnostic bounding box proposals with RPN model. Then we apply RoIAlign to obtain the features of proposals and regard them as node features. In the graph operations search stage, we search for a computation cell, where the supernodes are transformed by the selected graph operations on the superedges (see Section 3.2 and 3.3 for details), to construct adaptive structures. The searched structures are used to model the interactions in the corresponding videos. Finally, the node features are pooled into a video representation for interaction recognition.

3. Proposed Method

In order to learn adaptive interaction modeling structure for each video, we elaborate the graph operations search method in this section. We design the architecture search space with several basic graph operations, where the candidate operations are enriched in addition to graph convolution by several proposed new graph operations modeling different relations, *e.g*. the temporal changes and relations with background. We further develop the search framework based on differentiable architecture search to search adaptive structure for each video, which enables adaptive interaction modeling for different videos.

3.1. Overall Framework

We first present our overall framework for interaction recognition in Figure 2. Given a video, we sample some frames as the input to our model. We extract basic features of the sampled frames with a backbone CNN. At the same time, we extract class-agnostic RoIs for each frame with Region Proposal Network (RPN) [13]. Then we apply RoIAlign [13] to obtain features for each RoI. All the RoIs construct the graph for relation modeling. The nodes are exactly the RoIs, and edges are defined depending on the specific graph operations introduced in Section 3.2, in which different graph operations would indicate different connections and result in different edge weights. To obtain adaptive network structures, we employ differentiable architecture search mechanism to search adaptive structures in which graph operations are combined hierarchically. The interactions are modeled with the searched structures by transforming the node features with the selected graph operations. Finally, the output node features are pooled into a video representation for interaction classification.

In the following subsections, we describe the search space with basic graph operations and the architecture search framework in details.

3.2. Search Space with Graph Operations

To search the network structures, we firstly need to construct a search space. We search for a computation cell to construct the network structures, as illustrated in Figure 2. A computation cell is a directed acyclic computation graph with N ordered supernodes ("supernode" is renamed from "node" to avoid confusion with the nodes in the graphs constructed from RoIs). Each supernode contains all the nodes and each superedge indicates the candidate graph operations transforming the node features. In the computation cell, the input supernode is the output of the previous one, and the output is the channel-wise concatenated node features of all the intermediate supernodes.

Each intermediate supernode can be obtained by summing all the transformed predecessors (the ordering is denoted as "N-1", "N-2", "N-3" in Figure 2) as follows,

$$\boldsymbol{X}^{(j)} = \sum_{i<j} o^{ij}(\boldsymbol{X}^{(i)}), \tag{1}$$

where $\boldsymbol{X}^{(i)}$, $\boldsymbol{X}^{(j)}$ are the node features of the i-th and j-th supernode, and o^{ij} is the operation on superedge (i, j). Thus the learning of cell structure reduces to learning the operations on each superedge, so that we design the candidate operations in the following.

We design the basic operations based on graph for explicit relation modeling. In addition to graph convolution, we propose several new operations, *i.e*. *difference propagation*, *temporal convolution*, *background incorporation* and *node attention*, which explicitly model different relations in videos and serve as basic operations in the search space.

3.2.1 Feature Aggregation

Graph convolution network (GCN) [17] is commonly used to model relations. It employs feature aggregation for relation reasoning, in which each node aggregates features from its neighboring nodes as follows,

$$\boldsymbol{z}_i = \delta\left(\sum_j a_{ij}^f \cdot \boldsymbol{W}_f \boldsymbol{x}_j\right), \tag{2}$$

where $\boldsymbol{x}_j \in \mathbb{R}^{C_{in}}$ is the feature of node-j with C_{in} dimensions, $\boldsymbol{W}_f \in \mathbb{R}^{C_{out} \times C_{in}}$ is the feature transform matrix applied to each node, $a_{ij}^f = \boldsymbol{x}_i^\mathsf{T} \boldsymbol{U}_f \boldsymbol{x}_j$ is the affinity between node-i and node-j with learnable weights $\boldsymbol{U}_f$, δ is a nonlinear activation function and the $\boldsymbol{z}_i \in \mathbb{R}^{C_{out}}$ is the updated feature of node-i with C_{out} dimensions. Through information aggregation on the graph, each node enhances its features by modeling the dependencies between nodes.

3.2.2 Difference Propagation

In videos, the differences between objects are important for recognizing interactions. But GCN may only aggregate features with weighted sum, which is hard to explicitly capture the differences. Therefore, we design an operation *difference propagation* to explicitly model the differences.

By slightly modifying Equation (2), the differences can be explicitly modeled as follows,

$$\boldsymbol{z}_i = \delta\left(\sum_{j, j \neq i} a_{ij}^d \cdot \boldsymbol{W}_d (\boldsymbol{x}_i - \boldsymbol{x}_j)\right), \tag{3}$$

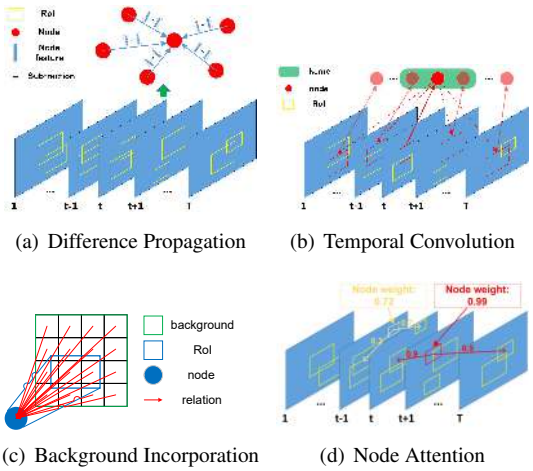
where the symbols share similar meanings of those in Equation (2). The item $(\boldsymbol{x}_i - \boldsymbol{x}_j)$ in Equation (3) explicitly models the differences between node-i and node-j, and then the differences are propagated on the graph, as shown in Figure 3(a). *Difference propagation* focuses on the differences between nodes to model the changes or differences of objects, which benefits recognizing interactions relevant to the changes or differences.

3.2.3 Temporal Convolution

Nodes in videos are inherently in temporal orders. However, both *feature aggregation* and *difference propagation* model the features in unordered manners and ignore the temporal relations. Here we employ *temporal convolution* to explicitly learn temporal representations.

In temporal convolutions, we firstly obtain node sequences in temporal order. Given node-i in the t-th frame, we find its nearest node (not required to represent the same object) in each frame measured by the inner product of node features and arrange them in temporal order for a sequence,

$$\boldsymbol{X}_i = [\boldsymbol{x}_i^0, \cdots, \boldsymbol{x}_i^t, \cdots, \boldsymbol{x}_i^{T-1}], \tag{4}$$



(a) Difference Propagation (b) Temporal Convolution

(c) Background Incorporation (d) Node Attention

Figure 3. Illustration of proposed graph operations. (a) Difference Propagation, each node propagates the differences to its neighboring nodes. (b) Temporal Convolution, each node learns temporal features with convolution over node sequences along the video. (c) Background Incorporation, each node aggregates the relations with the background. (d) Node Attention, each node learns attention weights to indicate its importance.

where $\boldsymbol{x}_i^0, \cdots, \boldsymbol{x}_i^{T-1}$ denote the nearest nodes in frame $0, \cdots, T-1$ with reference to the given node $\boldsymbol{x}_i^t$.

Then we conduct temporal convolutions over the node sequence as shown in Figure 3(b),

$$\boldsymbol{z}_i = \delta(\boldsymbol{W}_t * \boldsymbol{X}_i), \tag{5}$$

where $*$ denotes temporal convolution and $\boldsymbol{W}_t$ is the convolution kernel. The *temporal convolution* explicitly learns the temporal representations to model the significant appearance changes of the node sequence, which is essential for identifying interactions with temporal relations.

3.2.4 Background Incorporation

The node features derived from RoIAlign exclude the background information. However, background is useful since the objects probably interact with the background. This inspires us to design the *background incorporation* operation.

In each frame, the detected objects have different affinities with different regions in the background, as illustrated in Figure 3(c). Denote the feature of node-i in the t-th frame as $\boldsymbol{x}_i^t \in \mathbb{R}^{C_{in}}$ and the background feature map corresponding to the t-th frame as $\boldsymbol{y}^t \in \mathbb{R}^{h \times w \times C_{in}}$. The affinity between $\boldsymbol{x}_i^t$ and $\boldsymbol{y}_j^t$ ($j = 1, \cdots, h \times w$) can be calculated as $a_{ij}^b = \boldsymbol{x}_i^{t\mathsf{T}} \boldsymbol{U}_b \boldsymbol{y}_j^t$ with learnable $\boldsymbol{U}_b$. The a_{ij}^b indicates the relations between the node and the background with spatial structure, which could be transformed into node features,

$$\boldsymbol{z}_i^r = \boldsymbol{V}_b \boldsymbol{a}_i^b, \tag{6}$$

where $\boldsymbol{a}_i^b = [a_{i1}^b; a_{i2}^b; \cdots; a_{i(h\cdot w)}^b] \in \mathbb{R}^{h\cdot w}$ is the affinity vector, and $\boldsymbol{V}_b \in \mathbb{R}^{C_{out}\times(h\cdot w)}$ is the transform matrix transforming the affinity vector into node features.

In addition, the background features can be aggregated according to the affinity a_{ij}^b to model the dependencies between detected objects and the background,

$$\boldsymbol{z}_i^a = \sum_{j=1,\cdots,h\times w} a_{ij}^b \cdot \boldsymbol{W}_b \boldsymbol{y}_j. \tag{7}$$

Finally, the updated node features are the combination of the two features above followed by a nonlinear activation,

$$\boldsymbol{z}_i = \delta(\boldsymbol{z}_i^r + \boldsymbol{z}_i^a). \tag{8}$$

3.2.5 Node Attention

The graph contains hundreds of nodes but they contribute differently to recognizing interactions. Some nodes irrelevant to the interaction serve as outliers that interfere the interaction modeling, so it is reasonable to weaken the outliers with attention scheme.

The outliers are often the nodes wrongly detected by RPN, which usually have few similar nodes and their similar nodes do not locate regularly at specific regions or along the videos, as briefly illustrated in Figure 3(d). So that we calculate the attention weights according to the similarities and relative positions to the top-M similar nodes.

$$\begin{aligned}
&\boldsymbol{z}_i = w_i \cdot \boldsymbol{x}_i, \\
&w_i = \sigma(\boldsymbol{W}_n [\boldsymbol{a}_i^n; \Delta \boldsymbol{s}_i]), \\
&\boldsymbol{a}_i^n = \left[a_{ij_1}^n; a_{ij_2}^n; \cdots; a_{ij_M}^n\right], \\
&\Delta \boldsymbol{s}_i = \begin{bmatrix} \boldsymbol{s}_i - \boldsymbol{s}_{j_1} \\ \boldsymbol{s}_i - \boldsymbol{s}_{j_2} \\ \cdots \\ \boldsymbol{s}_i - \boldsymbol{s}_{j_M} \end{bmatrix},
\end{aligned} \tag{9}$$

where w_i is the attention weight of $\boldsymbol{x}_i$, which is calculated from similarity vector $\boldsymbol{a}_i^n$ and relative positions $\Delta \boldsymbol{s}_i$, σ is the sigmoid nonlinear function, j_m is the node index of node-i's m-th similar nodes measured by inner product, and $a_{ij_m}^n$ is the inner product of node features between node-i and node-j_m, and $\boldsymbol{s}_i = [x_i; y_i; t_i]$ is the normalized spatial and temporal positions of node-i. With the attention weights, we are able to focus on informative nodes and neglect the outliers.

The graph operations above explicitly capture different relations in videos and serve as the basic operations in the architecture search space, which facilitates structure search in Section 3.3.

3.3. Searching Adaptive Structures

With the constructed search space, we are able to search adaptive structures for interaction modeling. We employ differentiable architecture search mechanism in DARTS [21] to develop our search framework, and revise the learning of operation weights to facilitate search of adaptive interaction modeling structures.

DARTS. DARTS utilizes continuous relaxation to learn specific operations (o^{ij} in Equation (1)) on the superedges. The softmax combination of all the candidate operations are calculated as the representation of each supernode,

$$\bar{o}^{ij}(\boldsymbol{X}^{(i)}) = \sum_{o\in\mathbb{O}} \frac{exp(\alpha_o^{ij})}{\sum_{o'\in\mathbb{O}} exp(\alpha_{o'}^{ij})}\, o(\boldsymbol{X}^{(i)}), \tag{10}$$

where $\mathbb{O}$ is the set of candidate operations, o represents a specific operation, α_o^{ij} is the operation weight of operation o on superedge (i, j), and the $\bar{o}^{ij}(\boldsymbol{X}^{(i)})$ is the mixed output. In this way, the cell structure learning reduces to the learning of operation weights α_o^{ij}.

To derive the discrete structure after the search procedure converges, the operation with strongest weight is selected as the final operation on superedge (i, j),

$$o^{ij} = \arg\max_{o\in\mathbb{O}}\ \alpha_o^{ij}. \tag{11}$$

Adaptive Structures. Since the interactions differ from video to video, we attempt to learn adaptive structures for automatical interaction modeling. However, the operation weights α_o^{ij} in Equation (10) is non-adaptive. So that we modify the α_o^{ij} to be adaptive by connecting them with the input video through a fully-connected (FC) layer,

$$\alpha_o^{ij} = \boldsymbol{A}_o^{ij} \boldsymbol{X}, \tag{12}$$

in which $\boldsymbol{X}$ is the global feature of input video (global average pooling of the backbone feature) and $\boldsymbol{A}_o^{ij}$ is the learnable structure weights corresponding to operation o on superedge (i, j). In this way, adaptive structures are constructed for different videos to model the interactions.

Unlike alternatively optimizing the model in training and validation set to approximate the architecture gradients in DARTS, we jointly optimize the structure weights and the weights in all graph operations in training set to learn adaptive structures.

Fixing Substructures. It is time consuming to search stable structures with too many candidate operations. We attempt to reduce the number of basic operations by combining several operations into fixed substructures and regarding the fixed substructures as basic operations in the search space. For example, we connect *feature aggregation* and *node attention* sequentially into a fixed combination, and put it after the other 3 graph operations to construct 3 fixed substructures for search (as shown on the superedges in Figure 4).

By this means, we accelerate search by simplifying the search space and also deepen the structures because each superedge contains multiple graph operations.

Diversity Regularization. We find that the search framework easily selects only one or two operations to construct structures, because these operations are easier to optimize. However, other operations are also effective on interaction modeling, so we hope to keep more operations activated in the searched structures. We introduce the variance of operation weights as an auxiliary loss to constraint that all the operations would be selected equally,

$$L_{var} = \frac{1}{|\mathbb{O}| - 1} \sum_{o \in \mathbb{O}} (\alpha_o - \bar{\alpha})^2, \tag{13}$$

where $\alpha_o = \sum_{(i,j)} \alpha_o^{ij}$, $\bar{\alpha}$ is the mean of α_o. The variance loss is added to the classification loss for optimization.

4. Experiments

4.1. Datasets

We conduct experiments on two large interaction datasets, Something-Something-V1(Sth-V1) and Something-Something-V2(Sth-V2) [10] (see Figure 7 and 8 for some example frames). Sth-V1 contains 108,499 short videos across 174 categories. The recognition of them requires interaction reasoning and common sense understanding. Sth-V2 is an extended version of Sth-V1 which reduces the label noises.

4.2. Implementation Details

In the training, we employ stagewise training of the backbone and the graph operations search for easier convergence. And we optimize the weights in all graph operations and the structure weights ($\boldsymbol{A}_o^{ij}$ in Equation (12)) alternately to search adaptive structures.

In the structures search stage, we include the *zero* and *identity* as additional candidate operations. Following [6], we add dropout after *identity* to avoid its domination in the searched structures. We use 3 intermediate supernodes in each computation cell. The weight for auxiliary variance loss L_{var} (Equation (13)) is set to 0.1.

More details about the model, training procedure and data augmentation are included in supplementary materials.

4.3. Analysis of Architecture Search Framework

In this section, we analyze our architecture search framework. First we compare the interaction recognition accuracy of our searched structures with our baselines, and the results are shown in Table 1. It is observed that our searched structures obtain about 3% improvements over the baselines, *i.e. global pooling* (global average pooling of the backbone feature) and *pooling over RoIs* (average pooling over all the RoI features), indicating that the searched structures are effective to model interactions and improve recognition performance. In the following, we show the searched structures and analyze the effects of adaptive structures.

Search schemes	V1 Val[1]Acc	V2 Val[1]Acc
global pooling	48.1	60.3
pooling over RoIs	48.3	60.3
non-adaptive (only testing)[2]	50.2	62.4
non-adaptive (training and testing)[3]	50.8	63.1
adaptive	51.4	63.5

[1] Something-Something-V1 validation set and Something-Something-V2 validation set
[2] Only one searched structure (corresponding to most training videos) is used for testing.
[3] The structure are non-adaptive both in training and testing.

Table 1. Interaction recognition accuracy (%) comparison of different search schemes.

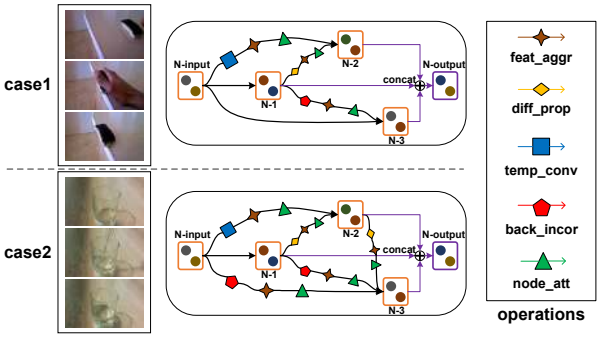


Figure 4. Two example videos and their corresponding structures. In the figure, "feat_aggr", "diff_prop", "temp_conv", "back_incor", "node_att" represent *feature aggregation*, *difference propagation*, *temporal convolution*, *background incorporation* and *node attention*, respectively.

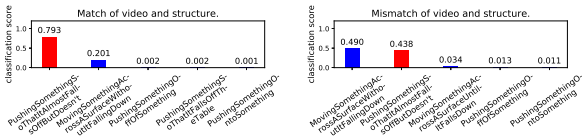
4.3.1 Searched Structures

Figure 4 shows two examples of the input videos and the corresponding searched structures. From the searched structures we observe that our architecture search framework learns adaptive structures for different input videos. The main differences between the two structures are the superedges entering "N-3", where *case1* learns simple structure but *case2* selects complicated structure with more graph operations. Perhaps *case2* is confusing with other interactions and requires complicated structures to capture some detailed relations for effective interaction modeling.

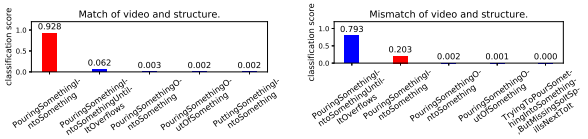
Mismatch of videos and structures. To validate the specificity of adaptive structures, we swap the two searched structures in Figure 4 to mismatch the input videos, and use them to recognize the interactions. The results are compared in Figure 5. We observe that the mismatch of videos and structures leads to misclassification, which reveals that different videos require different structures for effective interaction modeling, since different interactions of different complexities are involved.

4.3.2 Analysis of Adaptive Structures

To understand the relations between the adaptive structures and the interaction categories, we statistically analyze the proportion of videos per class corresponding to different searched structures in validation set. Figure 6 compares the

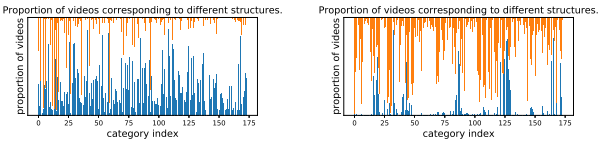


(a) Match and mismatch classification comparison of case 1.



(b) Match and mismatch classification comparison of case 2.

Figure 5. Top 5 classification score comparison of match and mismatch of videos and structures. (a) and (b) show the results of the two cases in Figure 4. The red bars indicate the groundtruth categories.



(a) Something-Something-V1 (b) Something-Something-V2

Figure 6. The proportion of videos per class corresponding to different structures. (a) and (b) show the results on the two datasets. The bars with different colors indicate different structures.

results of two searched structures indicated with different colors. We observe that the searched structures are strongly correlated to the interaction categories, where each structure corresponds to some specific interaction categories. For examples, in Something-Something-V1 dataset, the structure indicated with orange bars mainly corresponds to the interactions of indexes $\{2, 4, 6, 12, 15, \textit{et al.}\}$, which are about the motions of the camera. While the structure indicated with blue bars includes the interactions about moving/pushing objects (of indexes $\{8, 26, 29, 30, 41, \textit{et al.}\}$). This reveals that our architecture search framework learns to roughly divide the videos into several groups according to some characteristics in the interactions, and search specialized structures for different groups for adaptive interaction modeling. In other words, the adaptive structures automatically model interactions in a coarse (groups) to fine (specialized structure for each group) manner.

We further quantitatively compare the interaction recognition accuracy of non-adaptive and adaptive search schemes in Table 1. We make the following observations: On the one hand, adaptive scheme gains better performance than non-adaptive schemes. On the other hand, using only one searched structure for testing leads to obvious performance degradation, since different structures are searched to match different groups during training but only one structure is used for testing, which is insufficient to model the interactions in all groups. These observations further indicate the effectiveness of the adaptive structures.

Operations	V1 Val Acc	V2 Val Acc
global pooling	48.1	60.3
pooling over RoIs	48.3	60.3
feature aggregation	49.9	62.0
difference propagation	49.5	61.8
temporal convolution	48.7	61.0
background incorporation	49.7	62.4
node attention	49.8	61.8

Table 2. Interaction recognition accuracy (%) comparison of different graph operations.

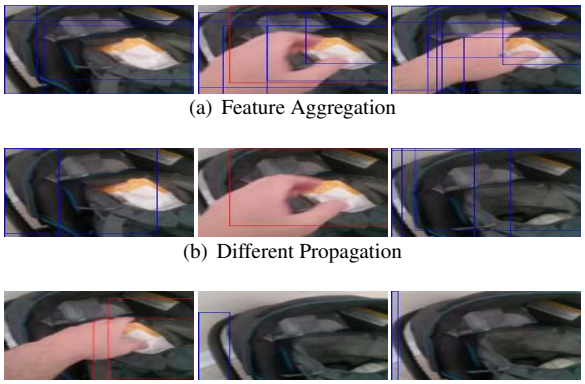
We also validate that learning with fixed substructures gains slight improvements, diversity regularization helps to learn structures with multiple operations, and the adaptive structures can transfer across datasets. For more details, please refer to our supplementary materials.

4.4. Analysis of Graph Operations

In this section, we analyze the role of each graph operation in interaction modeling. Firstly, we compare the recognition accuracy of different operations by placing them on top of the backbone, and the results are shown in Table 2. It is seen that all the operations improve the performance over baselines, indicating that explicitly modeling the relations with graph operations benefits interaction recognition. Different graph operations gain different improvements, which depends on the significance of different relations in the datasets. In the following, we visualize some nodes and cases to demonstrate the different effects of different graph operations in interaction modeling.

Top activated nodes. We visualize the nodes with top affinity values of some operations for the same video in Figure 7. The *feature aggregation* focuses on the apparently similar nodes to model the dependencies among them as shown in Figure 7(a). On the contrary, the *difference propagation* models the significant changes of some obviously different nodes in Figure 7(b). In Figure 7(c), the nodes with high attention weights are the hand or the bag, and the nodes with low attention weights are some outliers, which indicates that the *node attention* helps to concentrate on important nodes and eliminate the interference of outliers.

Successful and failed cases. We show some successful and failed cases to indicate the effects of different operations in Figure 8. In Figure 8(a), the *feature aggregation* successfully recognizes the interaction due to the obvious dependencies between the paper and the mug. However, it fails when detailed relations in Figure 8(b) and 8(c) are present. In Figure 8(b), the *difference propagation* and the *temporal convolution* could capture that the lid is rotating so that they correctly recognize the interaction. In Figure 8(c), the *background incorporation* is able to capture the relations between the towel and the water in the background so that

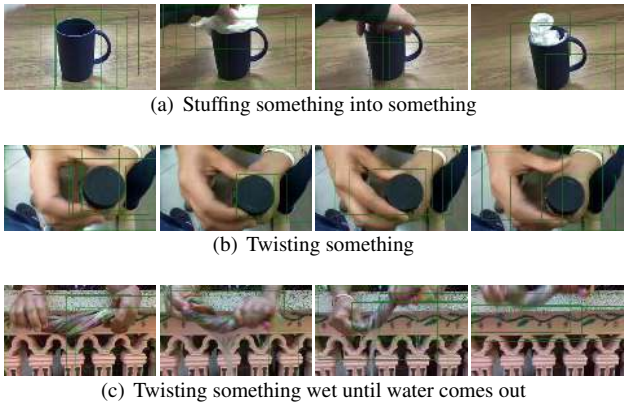


(a) Feature Aggregation

(b) Different Propagation

(c) Node Attention

Figure 7. Top activated nodes of different operations on the same interaction "Pulling something out of something". In (a) and (b), the red node is the reference node and the blue nodes are the top activated nodes. In (c), The red nodes have the highest attention weights while the blue ones have the lowest attention weights.



(a) Stuffing something into something

(b) Twisting something

(c) Twisting something wet until water comes out

Figure 8. Successful and failed cases of different graph operations. The green bounding boxes are RoIs extracted from RPN.

it makes correct prediction, but other operations ignoring the background information are hard to recognize such an interaction with the background.

More case study and analysis about graph operations are included in supplementary materials.

4.5. Comparison with State-of-the-arts

We compare the interaction recognition accuracy with recent state-of-the-art methods, and the results are show in Table 3. Except for STM [16], our method outperforms other methods, which indicates the effectiveness of our method. We model the interactions with adaptive structures, which enhances the ability of interaction modeling and boosts the performance.

Among the recent state-of-the-arts, I3D+GCN [35] also uses graph operation over object proposals to recognize interactions. Our method surpasses it with a margin about 7%, perhaps because we have trained a better backbone with our data augmentation techniques (see Section 4.2 for details),

Methods	V1 Val Acc	V2 Val Acc
I3D+GCN [35] (ECCV'18)	43.3	-
NonLocalI3D+GCN [35] (ECCV'18)	46.1	-
CPNet [22] (CVPR'19)	-	57.6
TSM [19] (ICCV'19)	44.8[1]	58.7[1]
ECO [41] (ECCV'18)	46.4	-
TrajectoryNet [39] (NeurIPS'18)	47.8	-
S3D [38] (ECCV'18)	48.2	-
ir-CSN-152 [30] (ICCV'19)	48.4	-
GST [23] (ICCV'19)	48.6	62.6
discriminative filters [24] (ICCV'19)	50.1[2]	-
STM [16] (ICCV'19)	50.7	**64.2**
adaptive structures search (Ours)	**51.4**	63.5

[1] Only RGB results are reported for fair comparison.
[2] Only the results with the same backbone (ResNet50) as ours are reported.

Table 3. Interaction recognition accuracy (%) comparison with state-of-the-arts.

and our adaptive structures with multiple graph operations learn better interaction representations.

STM [16] proposes a block to encode spatiotemporal and motion features, and stacks it into a deep network, which obtains better performance on Something-something-V2 dataset than ours. However, we adaptively model interactions with different structures, which provides more understanding about the relations between the interactions and the corresponding structures, instead of only feature encoding in STM. In addition, our structures are automatically searched, which releases the structures design efforts.

5. Conclusion

In this paper, we propose to automatically search adaptive network structures for interaction recognition, which enables adaptive interaction modeling and reduces structures design efforts. We design the search space with several proposed graph operations, and employ differentiable architecture search mechanism to search adaptive interaction modeling structures. Our experiments show that the architecture search framework learns adaptive structures for different videos, helping us understand the relations between structures and interactions. In addition, the designed basic graph operations model different relations in videos. The searched adaptive structures obtain competitive interaction recognition performance with state-of-the-arts.

Acknowledgement

This work was supported partially by the National Key Research and Development Program of China (2018YFB1004903), NSFC(U1911401,U1811461), Guangdong Province Science and Technology Innovation Leading Talents (2016TX03X157), Guangdong NSF Project (No. 2018B030312002), Guangzhou Research Project (201902010037), and Research Projects of Zhejiang Lab (No. 2019KD0AB03). The principal investigator for this work is Wei-Shi Zheng.

References

[1] Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, 2017.

[2] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 273–283, 2018.

[3] Gedas Bertasius, Lorenzo Torresani, Stella X Yu, and Jianbo Shi. Convolutional random walk networks for semantic image segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 858–866, 2017.

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[5] Siddhartha Chandra, Nicolas Usunier, and Iasonas Kokkinos. Dense and low-rank gaussian crfs using deep embeddings. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 5103–5112, 2017.

[6] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. *arXiv preprint arXiv:1904.12760*, 2019.

[7] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 433–442, 2019.

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, 2016.

[9] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8359–8367, Salt Lake City, UT, USA, 2018.

[10] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database for learning and evaluating visual common sense. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017.

[11] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[12] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[14] Noureldien Hussein, Efstratios Gavves, and Arnold W.M. Smeulders. Timeception for complex action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 254–263, 2019.

[15] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[16] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2000–2009, 2019.

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

[18] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *Advances in Neural Information Processing Systems*, pages 1858–1868, 2018.

[19] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019.

[20] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search. In *International Conference on Learning Representations*, 2018.

[21] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[22] Xingyu Liu, Joon-Young Lee, and Hailin Jin. Learning video representations from correspondence proposals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4273–4281, 2019.

[23] Chenxu Luo and Alan L. Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 5512–5521, 2019.

[24] Brais Martinez, Davide Modolo, Yuanjun Xiong, and Joseph Tighe. Action recognition with spatial-temporal discriminative filter banks. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 5482–5491, 2019.

[25] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1105–1116, 2016.

[26] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542, 2017.

[27] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4780–4789, 2019.

[28] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[30] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 5552–5561, 2019.

[31] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018.

[32] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, 2018.

[33] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[34] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.

[35] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *The European Conference on Computer Vision (ECCV)*, pages 413–431, 2018.

[36] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6857–6866, 2018.

[37] Tete Xiao, Quanfu Fan, Dan Gutfreund, Mathew Monfort, Aude Oliva, and Bolei Zhou. Reasoning about human-object interactions through dual attention networks. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3919–3928, 2019.

[38] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *The European Conference on Computer Vision (ECCV)*, pages 318–335, 2018.

[39] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *Advances in Neural Information Processing Systems*, pages 2204–2215, 2018.

[40] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *The European Conference on Computer Vision (ECCV)*, pages 831–846, 2018.

[41] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *The European Conference on Computer Vision (ECCV)*, pages 713–730, 2018.

[42] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

[43] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710, 2018.