International Journal of Computer Vision (2021) 129:1484–1505
https://doi.org/10.1007/s11263-020-01409-9

Spatial–Temporal Relation Reasoning for Action Prediction in Videos

Xinxiao Wu[1] · Ruiqi Wang[1] · Jingyi Hou[1] · Hanxi Lin[1] · Jiebo Luo[2]

Received: 11 December 2019 / Accepted: 26 November 2020 / Published online: 12 February 2021

Abstract
Action prediction in videos refers to inferring the action category label by an early observation of a video. Existing studies mainly focus on exploiting multiple visual cues to enhance the discriminative power of feature representation while neglecting important structure information in videos including interactions and correlations between different object entities. In this paper, we focus on reasoning about the spatial–temporal relations between persons and contextual objects to interpret the observed video part for predicting action categories. With this in mind, we propose a novel spatial–temporal relation reasoning approach that extracts the spatial relations between persons and objects in still frames and explores how these spatial relations change over time. Specifically, for spatial relation reasoning, we propose an improved gated graph neural network to perform spatial relation reasoning between the visual objects in video frames. For temporal relation reasoning, we propose a long short-term graph network to model both the short-term and long-term varying dynamics of the spatial relations with multi-scale receptive fields. By this means, our approach can accurately recognize the video content in terms of fine-grained object relations in both spatial and temporal domains to make prediction decisions. Moreover, in order to learn the latent correlations between spatial–temporal object relations and action categories in videos, a visual semantic relation loss is proposed to model the triple constraints between objects in semantic domain via VTransE. Extensive experiments on five public video datasets (i.e., 20BN-something-something, CAD120, UCF101, BIT-Interaction and HMDB51) demonstrate the effectiveness of the proposed spatial–temporal relation reasoning on action prediction.

Keywords Action prediction · Spatial–temporal relation reasoning · Long short-term graph network · Improved gated graph neural network

1 Introduction

Action prediction in videos aims to predict action category labels from partial videos that contain incomplete action executions and has achieved remarkable progress in recent years. But it still remains challenging since it is difficult to exploit sufficiently discriminative information from the partial video such as only the onset 10% of a complete video to make accurate prediction. Some existing methods (Kong et al. 2017; Kong et al. 2020; Cai et al. 2019; Wang et al. 2019; Zhao and Wildes 2019) explore enriched feature representations from partial videos by transferring discriminative information from full videos to partial videos. Several other methods (Kong et al. 2018; Lan et al. 2014; Cao et al. 2013; Ryoo 2011) learn feature templates from training videos and use the templates to perform matching for each segment of a testing video. Although learning discriminative feature representations has been widely studied by aggregating extra information such as the progress level of action execution, explicitly exploiting the interactions between objects from

Communicated by Sven J. Dickinson.

✉ Xinxiao Wu
wuxinxiao@bit.edu.cn

Ruiqi Wang
wang_ruiqi@bit.edu.cn

Jingyi Hou
houjingyi@bit.edu.cn

Hanxi Lin
hxlin@bit.edu.cn

Jiebo Luo
jluo@cs.rochester.edu

[1] Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 10081, People's Republic of China

[2] Department of Computer Science, University of Rochester, Rochester, NY 14627, USA

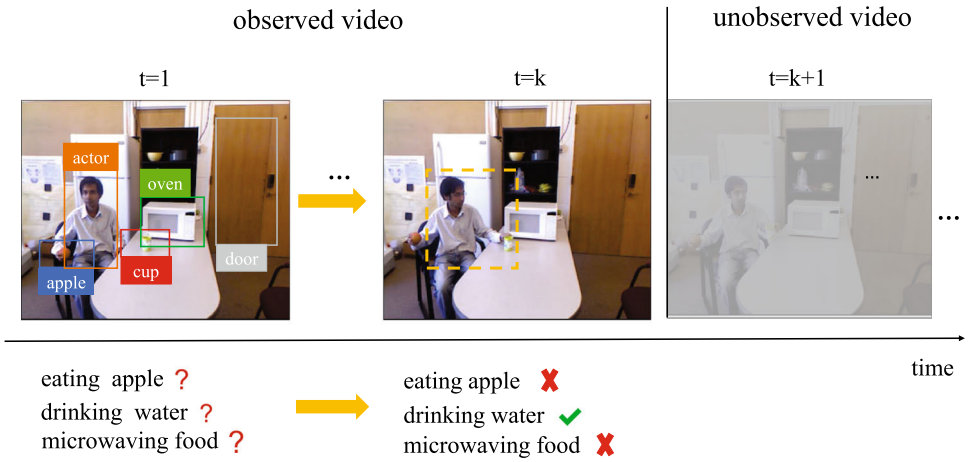


Fig. 1 Spatial and temporal relation reasoning in videos. The objects and their relative positions indicate that the actor is going to eat apple, drink water or microwave food. With the increasing input, when the actor reaches out to the cup, we can immediately make the prediction that the actor is most likely to drink water. The prediction is made by spatial and temporal relation reasoning

observed videos has received far less attention in action prediction.

This paper focuses on reasoning about both the spatial and temporal relations between persons and objects to provide an accurate visual content recognition of the observed videos for action prediction. The spatial relations between persons and objects offer a global interpretation of the scene on a per-frame base and the temporal relations characterize the underlying evolution of the spatial relations over time. For example, as shown in Fig. 1, the objects of "apple", "cup" and "microwave oven" and their relative positions with the actor suggest possible actions of "eating apple", "drinking water" and "microwaving food", respectively. This demonstrates that the spatial relations between persons and objects can help provide important cues of the actions in videos frames, thus contributing significantly to an early prediction. It can be seen from Fig. 1, the prediction of "drinking water" is accurately made only with the observation that the actor reaches out to the cup, since the ambiguity is eliminated by capturing the temporally varying relation between the actor and the cup in sequential frames. Therefore, it is more important and beneficial to infer the temporal relations in videos for action prediction by modeling the dynamic interactions between persons and objects over time.

With this in mind, we propose a novel spatial–temporal relation reasoning method for action prediction in videos. In this work, persons are considered as a type of object. For the spatial domain, we propose an improved gated graph neural network (IGGNN), to perform spatial relation reasoning of objects within video frames. Specifically, for each video frame, objects are first detected using Faster R-CNN (Girshick 2015) pre-trained on the ImageNet-1k dataset (Deng et al. 2009) to describe the local details from the regions of interest. A spatial graph is then built on the detected objects, where the node denotes each object represented by its detected bounding box and the directed edge denotes the relation between each pair of objects represented by the union bounding box of the two objects. Accordingly, the spatial relation reasoning is implemented by message propagation on the spatial graph to learn spatial relation-aware object representations.

For the temporal domain, we design a long short-term graph network (LSTGN) to perform temporal relation reasoning between sequential frames. To model the temporal dynamics of spatial relations, we build a temporal graph on the video frames. The node denotes each frame, represented by the corresponding spatial graph, which can be considered as a super node. The directed edge denotes the temporal relation between each pair of two frames, represented by the connection of spatial graphs of the two frames. The connected spatial graphs between two frames are temporally ordered and temporally isometric. Consequently, the temporal relation reasoning is implemented by message propagation on the temporal graph to learn temporal relation representation between sequential frames.

In order to learn both the short-term and long-term dynamics of the spatial relations in videos, message propagation with multi-scale temporal receptive fields is explored in LSTGN. In particular, the graph convolutional layers of LSTGN propagate information along the edges of the temporal graph, and the edges are updated by learning to accumulate the node features at different scales of the temporal repceptive fields. Through the spatial–temporal relation reasoning, for each temporal scale, a video feature representation is generated by concatenating node and edge features learned from LSTGN. The final action prediction results are produced by fusing the classification results of those video features at multiple temporal scales.

IGGNN and LSTGN are jointly learned in an end-to-end manner. During the training process, a visual semantic relation loss is proposed to learn the latent correlations between the spatial–temporal relations and action categories. By using VTransE (Hanwang et al. 2017), the semantic relation between each pair of objects is modeled as a triplet in

the semantic domain. The representations of two objects are projected into a low-dimensional relation space where their semantic relation is formulated by a vector transformation, i.e., subject + relation ≈ object. Learning such a potential relation in videos further promotes the process of relation reasoning and boosts the prediction performance.

Our main contributions are summarized as follows. First, we propose a novel spatial–temporal relation reasoning method to capture both the spatial and temporal interactions between visual objects for predicting actions from incomplete videos. It can accurately recognize the video content by reasoning on the fine-grained object relations in the spatial, temporal and semantic domains.

Second, we design a long short-term graph network to perform relation reasoning with multi-scale temporal receptive fields. It can effectively capture the dynamics of the spatial relations in varying temporal ranges and be readily integrated into other networks of various tasks such as video captioning and visual answering-questioning.

Finally, extensive experiments on five datasets demonstrate the effectiveness of the proposed spatial–temporal relation reasoning on action prediction.

2 Related Work

2.1 Action Prediction

Different from action recognition (Bhoi 2019; Wang et al. 2016; Ji et al. 2013) which can use complete spatio-temporal information of the video to make classification, action prediction aims to recognize actions before they are completely executed. Ryoo (2011) firstly defines the problem of action prediction and developed an extension of a bag-of-words model to represent the temporal distribution of action features. In Li and Fu (2014) and Cao et al. (2013), the probabilistic suffix tree and posterior probability modeling are respectively employed to learn the sequence patterns of temporal segments for action prediction. Lan et al. (2014) propose a hierarchical model to capture the structure of actions at multiple granularities and deduce the action label. Most of these methods assume that the length and the observation ratio of a test video are available, which does not hold in the real-world application. To overcome this problem, Kong et al. (2014b), Kong and Fu (2016) model the label consistency between video segments and the partial video, and employ a monotonically increasing scoring function to constrain the model.

Recently, deep learning based methods have been proposed for action prediction. Some methods (Kong et al. 2018; Wang et al. 2019; Hu et al. 2018) focus on improving the performance of the predictor to tackle the problems of sub-optimal solution and noise interference. Kong et al. (2018) measure the predictability of different actions with the help of bi-direction Long Short-Term Memory (LSTM), and exploit a memory module to remember hard ones. Wang et al. (2019) develop a teacher-student learning framework to distill progressive action knowledge from an action recognition model (teacher) to an early action prediction model (student), across different tasks. Hu et al. (2018) develop a soft label assignment mechanism to estimate the progress levels of subsequences as well as enhance the action predictor. Other methods (Kong et al. 2017; Cai et al. 2019; Kong et al. 2020; Chen et al. 2018a) make efforts to learn discriminative features for prediction. Kong et al. (2017) explore to transfer the information from full videos to partial videos, and incorporate the label consistency as additional constraints. Similarly, Cai et al. (2019) propose a two-stage learning framework to transfer the action knowledge from full videos to partial videos. Kong et al. (2020) and Chen et al. (2018a) employ the adversarial learning and the reinforcement learning, respectively, to generate more representative features for predicting actions. Pang et al. (2019) and Zhao and Wildes (2019) both aim to generate future information to relieve the shortage of contextual information in partial videos. In addition to making accurate prediction, Aliakbarian et al. (2017) add the time penalty on the loss function of LSTM in order to make prediction as early as possible.

The aforementioned methods mainly devote to promoting the expressive ability of features or improving the cognitive ability of predictor, while neglecting the contextual relationships between objects in videos for prediction. Our method simultaneously performs spatial and temporal relation reasoning to comprehensively understand visual content to help making prediction, imitating the predictive process of human beings that employs reasoning to support decision making (Evans et al. 1993).

2.2 Visual Relation Reasoning

Visual relation reasoning has achieved promising progress in various computer vision tasks, such as visual question answering (Aditya et al. 2018) and image captioning (Anderson et al. 2018). Most methods focus on relation reasoning in static images (Newell and Deng 2017; Liang et al. 2018; Zhang et al. 2019; Woo et al. 2018; Qi et al. 2019; Chen et al. 2018b). For example, Xu et al. (2019) develop a spatial-aware graph relation network to discover the relationships for each object in an image. Liao et al. (2019) propose to detect the relations in images by predicting the semantic connection between objects guided by natural language. Lu et al. (2016) exploit the language priors from semantic word embedding to finetune the likelihood of a predicted relationship.

Compared with static images, videos provide more information for reasoning visual relations such as the dynamic interactions between objects. Si et al. (2018) builds a spa-

tial reasoning network to capture the high-level feature of video frames and uses a temporal stacking strategy to recognize action. Zhou et al. (2018) conducts temporal reasoning in videos by a simple neural network. These methods only pay attention to relation reasoning on either space domain or time domain. Shang et al. (2017) is first to propose spatial–temporal relation detection in videos. Later, Tsai et al. (2019) improve upon (Shang et al. 2017) to build a spatio-temporal energy graph using conditional random field for several visual tasks in videos. Wang and Gupta (2018) modify the graph convolutional network to model the spatial and temporal relations in videos for action recognition. Recently, Herzig et al. (2019) use explicit appearance of object-object interaction to learn an inter-object graph representation for action recognition. Nicolicioiu et al. (2019) introduce a recurrent space-time graph model by processing spatial and temporal information differently. These methods make alignment for each node at each time step, and use the spatial information from neighbor nodes as well as the temporal information from the last time step to update the current node. Sun et al. (2019) propose the relational recurrent network for multi-person activity forecasting. They model the spatial–temporal relations in a fully connected graph by regarding persons in each frames as nodes, and perform relational reasoning by propagating information from nodes to its neighbors. In contrast, our method models the spatial relations in each video frame as a super node and designs a temporal graph to perform temporal relation reasoning with multi-scale temporal receptive field, which enables our model to capture both the short-term and long-term dynamics of spatial relations.

2.3 Graph Neural Network

Scarselli et al. (2008) propose Graph Neural Networks (GNNs) by unifying the recursive neural network and the Markov chain model into a common framework. It first learns node representations and then aggregates node representations to generate the final representation of the graph. Later, several other extensions of GNNs (Scarselli et al. 2008) have been proposed for various tasks. Gated Graph Neural Networks (GG-NNs) and Gate Graph Sequence Neural Networks (GGS-NNs) are proposed in Li et al. (2016). Different from GNNs, GG-NNs incorporate a node annotation as an additional input to initialize each node, use Gated Recurrent Units (GRU) (Cho et al. 2014) in the propagation model and use backpropagation algorithm to remove the constraints on parameters. GGS-NNs consist of several GG-NNs to produce the sequential output. Kipf and Welling (2017) propose Graph Convolutional Networks (GCNs) to solve the node classification problem in a semi-supervised manner. They apply a localized first-order approximation of spectral graph convolutions to layer-wise propagation rule and design a neural network model to encode the graph structure. An attention mechanism is used in Graph Attention Network (GAT) (Veličković et al. 2018) to attach different importance to different nodes. Message Passing Neural Network (MPNN) (Gilmer et al. 2017) contains a message passage phase to update each node by aggregating information from neighbors, and generates a node-based output by a read-out function.

Our work makes a modification of GG-NNs by fusing edge information into the input to perform spatial relation reasoning within video frames. We also propose LSTGN based on GCNs to perform temporal relation reasoning between video frames.

3 Our Method

3.1 Overview

Our core idea of addressing action prediction in a video is to reason about the spatial and temporal relations between different objects to comprehensively interpret the visual content of the observed video part for action classification. Our method consists of two key components: an improved gated graph neural network (IGGNN) for spatial relation reasoning within video frames and a long short-term graph network (LSTGN) for multi-scale temporal relation reasoning between sequential video frames.

For each video, we first detect objects in each frame and extract visual features from the bounding boxes of the detected objects. Then we perform spatial–temporal relation reasoning via IGGNN and LSTGN. After that, for each temporal scale, a video representation is generated by concatenating the features of nodes and edges learned from LSTGN and is fed into a classifier to generate the action category probabilities. Finally, the action prediction result is produced by fusing the category probabilities of all the temporal scales. Figure 2 illustrates the overview of our method.

3.2 Spatial Relation Reasoning

3.2.1 Spatial Graph

We build a spatial graph to model the spatial relations between objects within each video frame, denoted as $\mathcal{G}_s = (V, E, \boldsymbol{A})$, where each node $v \in V$ represents the detected visual object and each edge $e \in E$ indicates the spatial relation between two objects, represented by the union bounding box of two objects proposals. The size of the spatial graph is determined by the number of extracted objects (i.e., the number of nodes). In our experiments, different datasets have different sizes of the spatial graph since different datasets contain different numbers of objects. For the CAD120 and BIT-Interaction datasets, the average number of nodes in

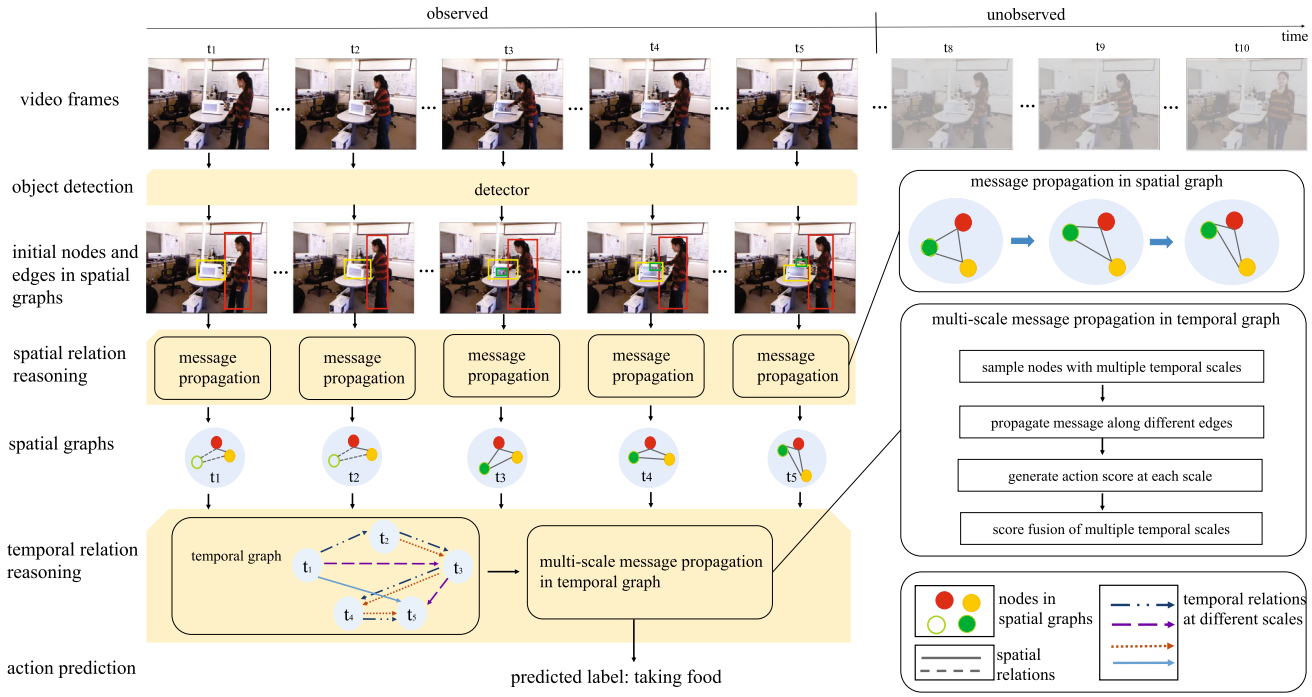


Fig. 2 Overview of the proposed method. Visual objects in each video frame are detected and input as the initial nodes and edges in spatial graphs. The spatial relation reasoning is performed by message propagation in the spatial graphs. The updated spatial graphs are treated as the super nodes in the temporal graph. The temporal relation reasoning is performed with the multi-scale message propagation. The final action category label is predicted by fusing decision scores of multiple scales. The circles represent nodes (i.e., detected objects) in the spatial graph, and the hollow circles represent virtual nodes (i.e., the objects that are not detected in this video frame but appear in the current video clip). The gray dashes represent the spatial relationship between virtual nodes and other nodes, and the gray lines represent the spatial relationship between nodes. Different arrows represent temporal relations at different temporal scales. Best viewed in color (Color figure online)

the spatial graph is 3 and 2, respectively. For the 20BN-something-something, UCF101 and HMDB51 datasets, the average number of nodes in the spatial graph is 5. The adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{|V|\times|E|}$ determines how nodes communicate with each other, where $|V|$ and $|E|$ denote the numbers of nodes and edges in the graph $\mathcal{G}_s$, respectively. The graph $\mathcal{G}_s$ is a directed fully connected graph, thus we have $|E| = |V|\times|V|$. To represent the existence of edge from node v_i to node v_j, we set $\boldsymbol{A}(i, q) = 1$, where i represents the ith row of $\boldsymbol{A}$ corresponding to the node v_i and $q = i \times |V| + j(i, j = 1, \ldots, |V|)$ represents the qth column of $\boldsymbol{A}$ corresponding to the edge e_q. Features of each node and each edge are utilized as their hidden states in message propagation for spatial relation reasoning, which are denoted as $\boldsymbol{h}_v \in \mathbb{R}^{d_v\times 1}$ and $\boldsymbol{h}_e \in \mathbb{R}^{d_e\times 1}$, respectively, where d_v and d_e represent the dimensions of node feature and edge feature, respectively. Since both the node feature and edge feature are extracted from the detected object regions, thus we have $d_v = d_e$.

3.2.2 Improved Gated Graph Neural Network

Inspired by the good performance of Graph Neural Networks on learning relations between entities, we propose an improved gated graph neural network (IGGNN) to perform spatial relation reasoning by message propagation in each spatial graph. Different from GG-NNs (Li et al. 2016) that takes only node features as input and propagates information of each node to its neighbor nodes, our IGGNN takes both the node and edge features as input, and propagates information of each node to its connected edges rather than neighbor nodes, which can learn more discriminative feature representations of nodes for action prediction. At the nth timestep of propagation, the interaction between the node v_i and its directly connected edges $\{e_1, \ldots, e_{|E|}\}$ is defined as

$$\boldsymbol{a}_{v_i}^{n\top} = \boldsymbol{A}_{v_i}^{n} \left[\boldsymbol{h}_{e_1}^{(n-1)} \ldots \boldsymbol{h}_{e_{|E|}}^{(n-1)}\right]^{\top} + \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{h}_{e_q}^{(n-1)} \in \mathbb{R}^{d_e\times 1}$ denotes the state of edge e_q at the $(n-1)$th timestep of propagation. $\boldsymbol{A}_{v_i}^{n} \in \mathbb{R}^{1\times|E|}$ denotes the ith row of adjacency matrix at the nth timestep of propagation, which represents the co-occurance relationship between the node v_i and each edge. Note that $\boldsymbol{A}_{v_i}^{n}$ is changing with the number of propagation time-steps n. $\boldsymbol{b}$ is a bias parameter. Equation 1 means that each node makes interaction with other nodes through all edges in the spatial graph, which encourages the model to reason the spatial relations in a broader region. The recurrence of message propagation is performed by a Gated Recurrent Unit (GRU) and formulated as

$$
\begin{aligned}
\boldsymbol{z}_{v_i}^n &= \sigma\left(\boldsymbol{W}^z \boldsymbol{a}_{v_i}^{n\top} + \boldsymbol{U}^z \boldsymbol{h}_{v_i}^{(n-1)}\right), \\
\boldsymbol{r}_{v_i}^n &= \sigma\left(\boldsymbol{W}^r \boldsymbol{a}_{v_i}^{n\top} + \boldsymbol{U}^r \boldsymbol{h}_{v_i}^{(n-1)}\right), \\
\widehat{\boldsymbol{h}}_{v_i}^n &= \tanh\left(\boldsymbol{W} \boldsymbol{a}_{v_i}^{n\top} + \boldsymbol{U}\left(\boldsymbol{r}_{v_i}^n \odot \boldsymbol{h}_{v_i}^{(n-1)}\right)\right), \\
\boldsymbol{h}_{v_i}^n &= \left(1 - \boldsymbol{z}_{v_i}^n\right) \odot \boldsymbol{h}_{v_i}^{(n-1)} + \boldsymbol{z}_{v_i}^n \odot \widehat{\boldsymbol{h}}_{v_i}^n,
\end{aligned}
\tag{2}
$$

where $\boldsymbol{h}_{v_i}^n \in \mathbb{R}^{d_v \times 1}$ denotes the state of node v_i at the nth timestep of propagation. $\sigma(x) = 1/\left(1 + e^{-x}\right)$ is the logistic sigmoid function and $\odot$ is the element-wise multiplication for matrix. $\boldsymbol{z}_{v_i}^n$ and $\boldsymbol{r}_{v_i}^n$ represent the update gate and reset gate, respectively, which help to preserve information characteristic from previous propagation step. $\boldsymbol{U}^z$, $\boldsymbol{U}^r$, $\boldsymbol{W}^z$, $\boldsymbol{W}^r$, $\boldsymbol{U}$ and $\boldsymbol{W}$ are parameters in GRU. The update of edge state with the message propagation is defined as

$$
\boldsymbol{h}_{e_q}^n = \boldsymbol{W}^e \boldsymbol{h}_{e_q}^{(n-1)} + \boldsymbol{d}, \tag{3}
$$

where $\boldsymbol{W}^e$ and $\boldsymbol{d}$ denote the weight parameters and bias parameter, respectively.

After the spatial relation reasoning performed by IGGNN, for the lth input video frame, the output graph-based representation $\boldsymbol{g}_l \in \mathbb{R}^{d_v \times 1} (l = 1, \ldots, L)$ is the information aggregation of all the nodes, given by

$$
\boldsymbol{g}_l = \tanh\left(\sum_{i=1}^{|V|} \alpha_i \tanh(\boldsymbol{h}_{v_i}^N)\right)\left(\sum_{i=1}^{|V|} \alpha_i = 1, \alpha_i \geq 0\right), \tag{4}
$$

where N is the last timestep of message propagation. α_i stands for the soft attention weight assigned to node v_i, formulated by

$$
\alpha_i = \mathrm{softmax}(\tanh(\boldsymbol{W}^\alpha \boldsymbol{h}_{v_i}^N + \boldsymbol{b}^\alpha)), \tag{5}
$$

where $\boldsymbol{W}^\alpha$ and $\boldsymbol{b}^\alpha$ are the weight parameter and bias parameter, respectively. In this way, each object in the frame can play a different role in predicting the action.

3.3 Temporal Relation Reasoning

3.3.1 Temporal Graph

Given an observed video with total L frames, we build a temporal graph to model the temporal relations between sequential video frames, denoted as $\mathcal{G}_t = (X, P)$, where the node set $X = \{x_1, \ldots, x_L\}$ indicates the observed video frames represented by their corresponding spatial graphs and the edge set P indicates the temporal relations between pairwise frames. Each edge is formulated by (x_l, r_{lu}, x_u), where $x_l, x_u \in X (l < u)$ and r_{lu} is a set of different scales of connections. $\mathcal{G}_t$ is a fully connected directed graph since we assume that every two frames have relation in the temporal domain and their temporal order should be maintained. Each node in graph $\mathcal{G}_t$ can be considered as a super node since each video frame is represented by a spatial graph.

3.3.2 Node Feature Representation

Let $\boldsymbol{f}_l$ denote the feature representation of the lth super node (i.e., the lth video frame) in the temporal graph $\mathcal{G}_t$. $\boldsymbol{f}_l$ consists of two parts: the spatial graph representation $\boldsymbol{g}_l \in \mathbb{R}^{d_v \times 1}$ and the object feature $\boldsymbol{O}_l \in \mathbb{R}^{d_v \times K}$.

The spatial graph representation $\boldsymbol{g}_l \in \mathbb{R}^{d_v \times 1}$ is the graph-based output of IGGNN, which captures the global interpretation of the corresponding video frame. The object feature $\boldsymbol{O}_l$ is obtained by encoding node states $\boldsymbol{h}_v^N$, which represents the detailed local visual information within each frame. Since different frames in the same video may have different numbers of visual objects, a video-level temporal alignment is designed to match the corresponding objects in different frames so that we can extract object features of the fixed size from video frames. Specifically, all the detected visual objects in the input video are firstly clustered into K clusters via K-means++ algorithm (Arthur and Vassilvitskii 2007). Each cluster has an anchor point, which is exactly the clustering center. Let $\boldsymbol{C} \in \mathbb{R}^{d_v \times K}$ represent all the anchor points and the kth column $\boldsymbol{c}_k \in \mathbb{R}^{d_v \times 1}$ of $\boldsymbol{C}$ represent the anchor point of the kth cluster. For each clustering center, we find a corresponding node in each frame by soft-assignment. For each video frame, the soft-assignment weight ω_i of the ith object ($i \in \{1, \ldots, |V|\}$) to the kth clustering center ($k \in \{1, \ldots, K\}$) is defined as

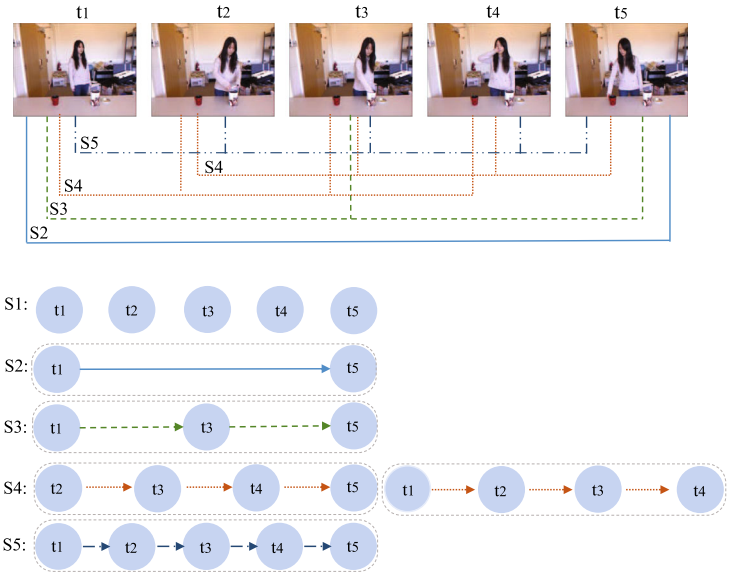
$$
\omega_i = \frac{\exp\left(-\beta \left\| \boldsymbol{h}_{v_i}^N - \boldsymbol{c}_i \right\|_2^2\right)}{\sum_{k=1}^K \exp\left(-\beta \left\| \boldsymbol{h}_{v_i}^N - \boldsymbol{c}_k \right\|_2^2\right)}, \tag{6}
$$

where β is a smoothing factor controlling the softness of the assignment. $\boldsymbol{h}_{v_i}$ is the updated feature of the node v_i from the last propagation step in IGGNN and $\boldsymbol{c}_i$ represents the anchor point of the ith cluster. This soft-assignment mechanism enables the model to allocate a virtual node when the detector is unable to detect the object corresponding to the clustering center in some frames, so that the number of objects in each frame is fixed. The object feature of the lth frame is given by

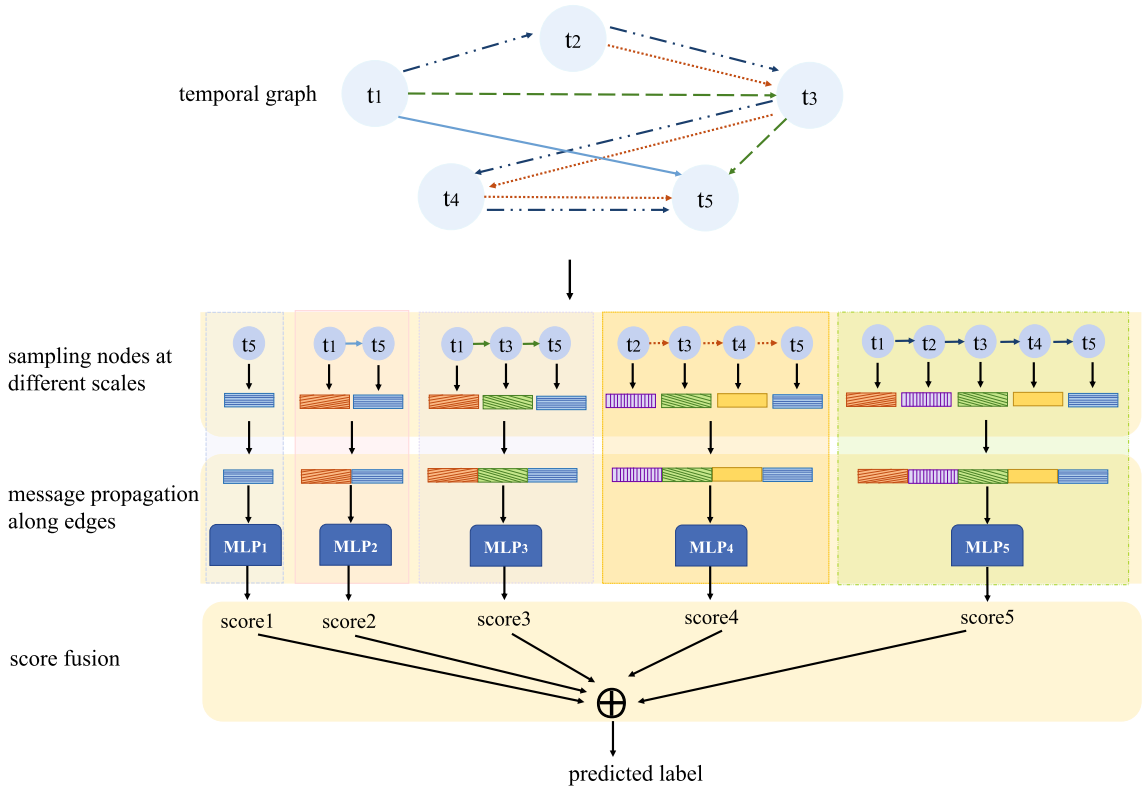
$$
\boldsymbol{O}_l = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K], \tag{7}
$$

where $\boldsymbol{s}_k$ represents the weighted sum of the residuals between all the objects and the anchor point of the kth cluster in a video, calculated by

$$
\boldsymbol{s}_k = \sum_{i=1}^{|V|} \omega_k \left(\boldsymbol{h}_{v_i}^N - \boldsymbol{c}_k\right). \tag{8}
$$



(a) Illustration of the multi-scale receptive fields.



(b) Message propagation in temporal graph for action prediction.

Fig. 3 Illustration of the multi-scale strategy in temporal relation reasoning. For zth temporal scale, we sample z video frames with equal interval, as shown in **a**. All the sampled frames and their temporal relations respectively construct the nodes and edges of the temporal graph, as shown in **b**. The message propagation is conducted on the temporal graph to update the node and edge features. During prediction, for each scale, the corresponding node features are concatenated to represent the video and generate action category probabilities. All the category probabilities from all the scales are summed up to produce the final action category label. Best viewed in color (Color figure online)

This video -level temporal alignment enables us to build a temporal graph for the holistic video by regarding each video frame as a super node.

3.3.3 Long Short-Term Graph Network

We propose a long short-term graph network (LSTGN) for performing temporal relation reasoning on the temporal graph to explore how the spatial relations change over time. Our LSTGN captures the dynamics of the spatial relations in varying temporal ranges via a multi-scale strategy. The multi-scale strategy refers to combining prediction results of temporal relation reasoning with different scales of receptive fields in the temporal domain, as shown in Fig. 3. The different scales mean sampling different numbers of video frames from a video as nodes of the temporal graph to propagate information on the temporal graph. It enables the temporal graph to update the information of each node and edge in both the short-term and long-term duration. To be specific, at the zth scale, we sample z nodes (e.g., video frames) with equal interval from the given video sequences. Among the z sampled nodes, the head node is represented by the spatial graph of the first sampled frame and the tail node is represented by the spatial graph of the last sampled frame. The edges between the two nodes are represented by the spatial graphs of the rest sampled frames. In order to take full advantage of the latest information, we sample the input video frames by including the last frame as much as possible. Taking the 4th scale for example, given a video with 10 frames, we sample four nodes that the 1st, the 4th, the 7th and the 10th frame. There are two special cases: the message propagations at the 1-st scale and the 2-nd scale. For the 1-st scale, we only sample the last frame and there is no edge. Thus, the model is unable to capture the temporal relation between frames. For the 2-nd scale, the two nodes are denoted by the spatial graphs of the sampled two frames and the edge is represented by the connection of the two spatial graphs. In this strategy, the larger the scale is, the more detailed temporal dynamics of the spatial relations can be captured. For the z-scale message propagation, given node features $\boldsymbol{f}_l, \boldsymbol{f}_u \in \mathbb{R}^{d_f}$ and edge features $\boldsymbol{v}_{lu} \in \mathbb{R}^{(z-2)\times d_f}$ for all the nodes $x_l, x_u \in X$ and edges $(x_l, r_{lu}, x_u) \in P$,

we compute updated node features $\boldsymbol{f}'_l$ and updated the edge features $\boldsymbol{v}'_{lu}$ for all the nodes and edges using two functions $g_{node}(\cdot)$ and $g_{edge}(\cdot)$, respectively. $g_{edge}(\cdot)$ takes head node feature $\boldsymbol{f}_l$, tail node feature $\boldsymbol{f}_u$ and edge feature $\boldsymbol{v}_{lu}$ as input, and outputs the updated edge feature $\boldsymbol{v}'_{lu}$, given by

$$\boldsymbol{v}'_{lu} = g_{edge}(\boldsymbol{f}_l, \boldsymbol{v}_{lu}, \boldsymbol{f}_u). \tag{9}$$

For each edge starting at node x_l and terminating at node x_u, we use $g_{node}(\cdot)$ to compute a set of updated vectors V_l^s for the head node x_l and a set of updated vectors V_u^e for the tail node x_u, formulated by

$$\begin{aligned} V_l^s &= \{g_{node}(\boldsymbol{f}_l, \boldsymbol{v}_{lu^*}, \boldsymbol{f}_{u^*}) : (x_l, r_{lu^*}, x_{u^*}) \in P\}, \\ V_u^e &= \{g_{node}(\boldsymbol{f}_{l^*}, \boldsymbol{v}_{l^*u}, \boldsymbol{f}_u) : (x_{l^*}, r_{l^*u}, x_u) \in P\}, \end{aligned} \tag{10}$$

where u^* represents the index of the tail node in edges (x_l, r_{lu^*}, x_{u^*}) that take x_l as the head node, $\boldsymbol{f}_{u^*}$ denotes the feature of the tail node and $\boldsymbol{v}_{lu^*}$ is the feature of all the edges between the head node and the tail node. Since the edges in the above two equations are different, we use notation $*$ to distinguish the different edges. The updated head node state $\boldsymbol{f}'_l$ and the updated tail node state $\boldsymbol{f}'_u$ are then calculated by

$$\begin{aligned} \boldsymbol{f}'_l &= h(V_l^s) = \sum_q \boldsymbol{f}_q(\boldsymbol{f}_q \in V_l^s), \\ \boldsymbol{f}'_u &= h(V_u^e) = \sum_q \boldsymbol{f}_q(\boldsymbol{f}_q \in V_u^e), \end{aligned} \tag{11}$$

where $h(\cdot)$ represents the sum operation to pool a set of vectors into a single vector.

Action Prediction Through the multi-scale message propagation on the temporal graph, both the short-term and long-term temporal relations are learned, which can be extracted to represent the observed video. For each scale, we concatenate the sampled node and edge features learned by LSTGN to generate the video feature, which is fed into a classifier (a fully connected layer) to produce action category probabilities.

Let $\{\succsim_j^{l_z} | j = 1, \ldots, C\}$ be the action category probabilities at the lth time step with scale of z, where C is the number of action categories. We sum up the scores of all the scales to produce the final action category probability scores $\{\succsim_j^{l} | j = 1, \ldots, C\}$ and use the softmax funtion to predict the action label:

$$\hat{y}_l = \text{softmax}(\succsim_j^l) = \frac{\succsim_j^l}{\sum_{k=1}^{C} \succsim_k^l} \tag{12}$$

3.4 Visual Semantic Relation Learning

To capture the latent correlations between the spatial–temporal object relations and action categories, we introduce a visual semantic relation loss into the joint learning procedure of IGGNN and LSTGN. The semantic relation between two objects is represented by a triplet in the semantic domain, denoted as $< subject - relation - object >$. Generally, the $subject$ and $object$ are visual objects in a video or image. The $relation$ usually refers to the relative position ("in front of", "on", etc.), action ("picking up", "eating", etc.), other verb or preposition and so on. By using the triple form, the

semantic relation between two objects is regarded as a transformation vector, which makes the *subject* state transformed into the *object* state. For example, <man−pick up−cup> represents that the state of "man" can be transformed into the state of "cup" by the relation of "pick up". On one hand, the constraints of the triple form guide the model to learn the semantic relation of <man−pick up−cup> rather than <man−pick up−oven>. On the other hand, the semantic relation of <man−pick up−cup> helps the model correlate "picking up the cup" with "taking medicine", such improving the prediction accuracy. We combine such semantic relation and the spatial–temporal relation to equip our spatial–temporal reasoning model with the association ability. Although the label of the semantic relation is not available in this paper, the hidden semantic relation can also be learned to promote the cognition and association ability of the model.

To be specific, node states $\boldsymbol{h}_v^N$ and edge states $\boldsymbol{h}_e^N$ from the last propagation step in spatial relation reasoning network represent *subject* (or *object*) and *relation*, respectively. Inspired by VTransE (Hanwang et al. 2017), we hold the assumption that a valid visual semantic relation is formulated as $subject + relation \approx object$ and apply L2 loss to the learning process. The visual semantic loss between the *object* i and the *object* j is defined as

$$L_{(i,j)} = \left\| \boldsymbol{W}_3 \cdot \boldsymbol{h}_{v_i}^N - \left(\boldsymbol{W}_1 \cdot \boldsymbol{h}_{e_q}^N + \boldsymbol{W}_2 \cdot \boldsymbol{h}_{v_j}^N \right) \right\|_2^2, \tag{13}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3$ are the mapping matrices. For a spatial graph, the visual semantic relation loss is refined by the average loss of all the relations among different objects:

$$\mathbb{L}_{\mathrm{s}}(\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3) = \frac{2}{|V|(|V|+1)} \sum_{i=1}^{|V|} \sum_{j=i+1}^{|V|} L_{(i,j)}. \tag{14}$$

3.5 Loss

Given an input video of L frames with their corresponding ground-truth labels $\{y_1, y_2, \ldots, y_L\}$, the loss for action category classification is defined as

$$\mathbb{L}_{\mathrm{r}}(\theta_1, \theta_2) = \sum_{l=1}^{L} [-y_l \log \hat{y}_l - (1 - y_l)(1 - \hat{y}_l)], \tag{15}$$

where $\theta_1 = \{\boldsymbol{U}^z, \boldsymbol{U}^r, \boldsymbol{W}^z, \boldsymbol{W}^r, \boldsymbol{U}, \boldsymbol{W}, \boldsymbol{W}^e, \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{W}^\alpha, \boldsymbol{b}^\alpha\}$ denotes the parameters of IGGNN and θ_2 denotes the parameters in g_{node} and g_{rela} of LSTGN.

Thus, the overall loss $\mathbb{L}$ consists of the classification loss $\mathbb{L}_{\mathrm{r}}$ based on the spatial–temporal relation reasoning and the visual semantic relation loss $\mathbb{L}_{\mathrm{s}}$, formulated by

$$\mathbb{L} = \mathbb{L}_{\mathrm{r}}(\theta_1, \theta_2) + \lambda \mathbb{L}_{\mathrm{s}}(\theta_3), \tag{16}$$

where $\theta_3 = \{\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3\}$ represents the parameters in visual semantic loss and λ is a trade-off parameter to control the balance between the spatial–temporal relation and the semantic relation for action prediction.

4 Experiments

4.1 Datasets

We evaluate our method on five datasets: CAD120 (Koppula et al. 2013), 20BN-something-something (Goyal et al. 2017), UCF101 (Liu et al. 2009; Soomro et al. 2012), BIT-Interaction (Kong et al. 2014a) and HMDB51 (Kuehne et al. 2011).

The **CAD120** dataset (Koppula et al. 2013) totally contains 120 vide-os of 10 different high-level action classes performed by four different persons. The 10 action classes are: "arranging objects", "cleaning objects", "having a meal", "making cereal", "microwaving food", "picking objects", "stacking objects", "taking food", "taking medicine" and "unstacking objects". Each action video can be decomposed into a sequence of sub-actions. Different action videos differ in the ordering of sub-actions and the way of action execution. The prediction is performed on the high-level action. Four-fold cross-validation is employed for evaluation and the results are reported by averaging classification accuracies across the folds. For each fold, the actions of one person are used for testing and that of the rest persons are for training.

The **20BN-something-something** dataset (Goyal et al. 2017) is a large collection of 108,499 densely-labeled video clips across 174 labels. These collected videos show basic actions of human (e.g., picking up, pulling) with everyday objects in real life, thus recognizing them requires a detailed understanding of actions and scenes. We use the standard and official subset that contains 21 action categories, including "Opening something", "Closing something", "Turning something upside down", "Pretending to turn something upside down" and so on. There are 11,101 short videos for training and 1,568 videos for validation. We report the results by averaging classification accuracies over all classes.

The **UCF101** dataset (Liu et al. 2009; Soomro et al. 2012) is a popular action prediction dataset collected from YouTube. It contains 13,320 videos covering 101 action categories. In order to provide the highest diversity, the actions in this dataset contain five types, including human-object interaction, human-human interaction, body-motion only, playing musical instruments, and sports. There are three official "train/val" splits for UCF101 and we follow the standard practice on this dataset by reporting the average results over the three splits.

The **BIT-Interaction** dataset (Kong et al. 2014a) constains 8 classes of human interactions, including "bow", "boxing", "hug", "handshake", "high-five", "kick", "pat" and "push". There are 400 videos in the dataset with 50 videos per class. According to previous work (Kong et al. 2017; Kong et al. 2020; Zhao and Wildes 2019), we use the first 34 videos in each class for training and the rest for testing.

The **HMDB51** dataset (Kuehne et al. 2011) has more than 6,000 video clips that are mostly from movies, public datasets, YouTube and Google videos. There are 51 categories such as "wave", "brush hair" and "kick ball", which contain various types of actions. Three train/val splits are provided for HMDB51. We report results by averaging over the three official splits.

4.2 Implementation Details

4.2.1 Feature Representation

For the CAD120 dataset, the publicly available Histogram of Oriented Gradient (HOG) features from Koppula et al. (2013) are used to initialize the nodes and edges of the spatial graph, respectively.

For the 20BN-something-something dataset, we train a Faster R-CNN model using ResNet-50-FPN backbone (Girshick et al. 2018). Since the ground-truth bounding box is not available in this dataset, we manually label the positions (bounding box) of objects in 10 frames sampled from each video and nearly 2,000 videos (about 20% of the training videos) are labeled. The objects are extracted by the trained detection model and the threshold of Intersection-over-Union (IoU) for proposals in non-maximum suppression is set to 0.5. We extract the features of bounding boxes from the last fully connected layer of the trained detection model. In the spatial graph, the nodes are initialized by the extracted features of the detected bounding boxes of individual objects, and the edges are initialized by the extracted features of union bounding boxes of the corresponding two objects.

For the UCF101 dataset, we employ a Faster R-CNN model (Girshick 2015) pre-trained on the ImageNet-1k dataset (Deng et al. 2009) to detect actors and objects in video frames. Then we extract three types of features to describe the actors, objects and their relations: two kinds of Two-Stream CNN features (He et al. 2016; Wang et al. 2016) and 3D CNN feature (Hara et al. 2018). For the first kind of Two-Stream CNN feature (He et al. 2016), we train two ResNets on the UCF101 dataset for the RGB and optical flow streams, respectively, following the settings in Hu et al. (2018). For the RGB stream we finetune ResNet-18 pre-trained on ImageNet and achieve 65% accuracy. For the optical flow stream we train ResNet-18 from scratch and achieve 50% accuracy. For the second kind of Two-Stream CNN feature (Wang et al. 2016), following Zhao and Wildes (2019), we use TSN (Wang et al. 2016) trained on Kinetics (Kay et al. 2017) with BN-Inception (Ioffe and Szegedy 2015) as the backbone. The feature extraction models are finetuned on the UCF101 dataset for the RGB and optical flow streams. For the 3D CNN feature, following Pang et al. (2019), we use the 3D ResNext-101 (Hara et al. 2018) trained on Kinetics (Kay et al. 2017) without finetuning. Finally, for each kind of feature, we extract features from the last convolutional layer of the feature model and make ROIPooling to generate node features and edge features as input.

For the BIT-Interaction and HMDB51 datasets, We also adopt the Faster R-CNN model (Girshick 2015) pre-trained on the ImageNet-1k dataset (Deng et al. 2009) as an object detector. For the BIT-Interaction dataset, following (Zhao and Wildes 2019), we extract features using TSN (Wang et al. 2016) trained on Kinetics (Kay et al. 2017) with BN-Inception (Ioffe and Szegedy 2015) as the backbone and finetune the model on the optical flow of BIT-Interaction dataset. For the HMDB51 dataset, we use 3D ResNext-101 (Hara et al. 2018) trained on Kinetics (Kay et al. 2017) without finetuning to extract features from the last convolutional layer. We take the feature produced by ROIPooling as the input of our model for the BIT-Interaction and HMDB51 datasets.

4.2.2 Parameter Setting

For IGGNN, the unit of GRU layer is empirically set to 512 and the number of propagation step is set to three. The feature dimensions of both initial nodes and edges are reduced to 256 by a linear layer. To ease the training of the graph network, a residual block is integrated before the last output layer which takes the initial node features and edge features as input. For LSTGN, the smoothing factor β in Eq. 6 is empirically set to 10.

The functions $g_{node}(\cdot)$ and $g_{edge}(\cdot)$ defined in Eq. 10 are both implemented by fully connected layers activated by ReLU function. The input dimension differs at different scales. Specifically, we use two fully connected layers ($512 \rightarrow 256$) for the CAD120 dataset and one fully connected layer (256) for the other four datasets. For the CAD120 and 20BN-something-something datasets, the number of scales is set to four and five, respectively. For the UCF101, BIT-Interaction and HMDB51 datasets, the number of scales is set to 8. The trade-off parameter λ is set to 0.125 for all the datasets. We use 10% of the training data as a validation set to choose the optimal values of the number of fully connected layers, the number of scale and the trade-off parameter λ for each dataset.

All the networks are trained from scratch with the learning rate of 0.00005. The Adam optimizer (Kingma and Ba 2015) and the SGD optimizer are employed with a batch size of 24 for optimization.

Table 1 Accuracies (%) of different methods on the CAD120 dataset

Method	Observation ratio				
	0.1	0.3	0.5	0.7	0.9
mem-LSTM (Kong et al. 2018)	89.59 ± 0.41	88.35 ± 0.61	87.88 ± 0.11	87.43 ± 1.11	90.35 ± 0.21
MS-LSTM (Aliakbarian et al. 2017)	81.11 ± 2.76	78.38 ± 0.96	76.06 ± 1.06	75.27 ± 2.61	70.92 ± 0.76
MSRNN (Hu et al. 2018)	89.44 ± 0.89	89.92 ± 0.40	90.73 ± 1.20	89.84 ± 1.29	90.32 ± 0.00
TRN (Zhou et al. 2018)	87.10 ± 2.42	86.29 ± 2.42	88.31 ± 2.02	89.92 ± 0.40	86.69 ± 0.41
Ours	**90.32**±1.61	**90.73**±1.21	**91.53**±0.40	**91.13**±0.81	**91.13**±0.80

Best accuracy among the compared methods are indicated in bold

Table 2 Accuracies (%) of different methods on the 20BN-something-something dataset

Method	Observation ratio				
	0.1	0.3	0.5	0.7	0.9
mem-LSTM (Kong et al. 2018)	14.92 ± 0.44	18.08 ± 0.41	20.44 ± 0.73	23.22 ± 0.63	24.46 ± 0.93
MS-LSTM(Aliakbarian et al. 2017)	17.42 ± 0.54	16.8 ± 0.02	16.71 ± 0.21	16.83 ± 0.13	17.07 ± 0.00
MSRNN (Hu et al. 2018)	20.62 ± 0.47	21.02 ± 0.23	22.45 ± 0.45	24.05 ± 0.45	27.13 ± 0.54
TRN (Zhou et al. 2018)	18.83 ± 0.50	20.09 ± 0.52	23.16 ± 0.50	27.56 ± 0.69	28.83 ± 0.64
Ours	**22.61**±1.24	**25.01**±0.47	**28.28**±0.79	**32.24**±0.47	**34.08**±0.31

Best accuracy among the compared methods are indicated in bold

4.3 Comparison with State-of-the-Art Methods

To evaluate the effectiveness of our method for predicting action, we compare our method with several state-of-the-art methods and report the action prediction accuracies at the observation ratios of {0.1, 0.3, 0.5, 0.7, 0.9}. The compared methods are listed as follows:

- **DBOW/IBOW** (Ryoo 2011) represents an action as a histogram of the spatio-temporal features to perform early action recognition.
- **MSSC** (Cao et al. 2013) formulates the prediction problem into a probabilistic framework by applying sparse coding to calculate the likelihood of each action temporal stage.
- **MTSSVM** (Kong et al. 2014b) enforces the label consistency between video segments and partial videos to maximize the discriminative power of the beginning temporal segments.
- **mem-LSTM** (Kong et al. 2018) augments bi-direction LSTM with a memory module to match characteristics of testing videos with training videos for action prediction.
- **MS-LSTM** (Aliakbarian et al. 2017) introduces a new classification loss by incorporating a time penalty to encourage the model make earlier prediction.
- **MSRNN** (Hu et al. 2018) assigns soft labels to subsequences that contain partial action executions and make regression, jointly learned with an action predictor.
- **DeepSCN** (Kong et al. 2017) exploits sequential context information extracted from full videos to enrich the feature representations of partial videos.
- **Global-Local** (Lai et al. 2018) applies metric learning to build a global-local temporal distance model with temporal saliency of video segments.
- **AAPNet** (Kong et al. 2020) builds upon DeepSCN and utilizes adversarial learning scheme to learn more discriminative features for action prediction.
- **DBDNet** (Pang et al. 2019) generates future motions and uses the synthesized future motions to reconstruct observed historical actions in order to utilize more contexual information for early action prediction.
- **RGN-KF** (Zhao and Wildes 2019) propagates feature residual across time to generate future features and takes advantage of Kalman filters to make correction for action prediction.
- **T–S** (Wang et al. 2019) proposes a teacher-student framework to distill progressive knowledge from action recognition model (teacher) to action prediction model (student).
- **Transfer** (Cai et al. 2019) learns a set of feature projection layers and classifiers from full videos and then uses them to improve the prediction of partial videos.
- **TRN** (Zhou et al. 2018) employs simple neural network to model temporal relation at multiple time scales in videos for making decision.

For the CAD120 and 20BN-something-something datasets, all the methods use the same features for fair comparison. In the compared methods (Kong et al. 2018; Aliakbarian et al. 2017; Hu et al. 2018; Zhou et al. 2018), the extracted features of bounding boxes are concatenated to represent each frame. All the models are retrained three times and we report the average accuracy and standard deviation. Tables 1 and 2 reports the results on the CAD120 dataset and the 20BN-something-something dataset, respectively. It can be observed that:

- Our method achieves better performance than other methods on both datasets, which verifies the superiority of reasoning both spatial and temporal relations between objects on action prediction.
- At the observation ratios of 0.1 and 0.3, our method achieves much higher accuracies than TRN that only performs temporal reasoning, which validates the importance of capturing the spatial relations between objects in frames.
- With the increasing input video frames, the results of 20BN-something-something dataset grow consistently while the results of CAD120 dataset have slight changes. The possible reason is that videos in 20BN-something-something dataset show more fine-grained actions and thus it requires gradually capturing discriminative temporal information as the input increases. In contrast, the early sub-actions of videos in CAD120 dataset are sufficient to distinguish different actions at the early stage, thus the results look more stable over time.

For the UCF101 dataset, we report the comparison results in Table 3 where the accuracies of all the compared methods are from Hu et al. (2018). All the methods use the ResNet-18 feature for fair comparison. From Table 3, we can observe that:

- The performance of our method at the observation ratio of 0.1 is vastly superior to the compared methods, which validates the benefit of capturing the spatial relations to an early prediction.
- Our method yields better results than most state-of-the-art methods at other observation ratios. Note that Hu et al. (2018) enhances the extracted CNN feature by employing an integral map computing technique before inputting them into the MSRNN model. Although no extra feature optimization technique is utilized in our method for CNN feature extraction, our method still achieves comparable *overall performance* at higher observation ratios compared with Hu et al. (2018), which indicates the effectiveness of the proposed spatial–temporal reasoning model.

For the UCF101 dataset, we also compare our method with several recent methods, i.e., DBDNet (Pang et al. 2019), RGN-KF (Zhao and Wildes 2019), T–S (Wang et al. 2019), Transfer (Cai et al. 2019) and AAPNet (Kong et al. 2020), shown in Table 4. All the results of these compared methods are reported directly from their original papers. For a fair comparison, we use the same Two-Stream CNN feature as RGN-KF (Zhao and Wildes 2019) and AAPNet (Kong et al. 2020), and use the same 3D CNN feature as DBDNet (Pang et al. 2019), T–S (Wang et al. 2019) and Transfer (Cai et al. 2019) to evaluate our method. From the results, it is interesting to observe that:

- Our method outperforms RGN-KF (Zhao and Wildes 2019) for most observation ratios using Two-Stream CNN feature, which shows the effectiveness of our method.
- Our method performs worse than DBDNet (Pang et al. 2019), T–S (Wang et al. 2019) and Transfer (Cai et al. 2019) using 3D CNN feature. The possible reason is that 3D CNN feature is not suitable to represent visual objects for spatial–temporal relation reasoning in our method. In our method, the objects are extracted in each video frame and then the spatial relation reasoning is performed. After that, the temporal relation reasoning is performed between video frames. Therefore, 2D CNN feature is more suitable to represent the objects while 3D CNN feature may introduce more noise from adjacent frames. So Two-Stream CNN feature is more suitable to our method, and when using Two-Stream CNN feature our method generally achieves the best results for most observation ratios.

For the BIT-Interaction dataset, we report our result by using the feature extraction model provided by Zhao and Wildes (2019) and directly copy the reported results of the compared methods from their papers. Table 5 shows the results of the BIT dataset. Our method outperforms RGN-KF (Zhao and Wildes 2019) by 5% but performs worse at the observation ratios of 0.3, 0.7 and 0.9, probably due to that the main related objects in videos of BIT-Interaction are two persons that have simple interactions, so the advantage of relation reasoning in our method is not obvious when compared with using global spatiotemporal feature for prediction in RGN-KF (Zhao and Wildes 2019).

For the HMDB51 dataset, we report our result by using the same 3D CNN feature as Cai et al. (2019) and directly copy the reported results of the compared methods from Cai et al. (2019). Table 6 illustrates the results on the HMDB51 dataset and our method achieves better or comparable results when compare with the state-of-the-art methods.

Table 3 Accuracies (%) of different methods using the ResNet-18 feature on the UCF101 dataset

Method	Observation ratio				
	0.1	0.3	0.5	0.7	0.9
DBOW (Ryoo 2011)	36.29	52.02	53.99	54.13	54.12
IBOW (Ryoo 2011)	36.29	70.23	72.56	73.84	74.72
MSSC (Cao et al. 2013)	34.05	58.32	62.52	63.55	62.67
MTSSVM (Kong et al. 2014b)	40.05	80.02	82.13	82.49	83.18
DeepSCN (Kong et al. 2017)	45.02	82.19	84.92	85.89	86.02
mem-LSTM (Kong et al. 2018)	51.02	86.75	88.37	89.22	89.97
MSRNN (Hu et al. 2018)	68.01	**88.71**	89.25	89.92	**90.23**
Ours	**80.86**	88.61	**89.31**	**90.31**	89.82

Best accuracy among the compared methods are indicated in bold

Table 4 Accuracies (%) of different recent methods using the Two-Stream CNN and 3D CNN features on the UCF101 dataset

Method	Feature	Observation ratio				
		0.1	0.3	0.5	0.7	0.9
RGN-KF (Zhao and Wildes 2019)	Two-stream CNN	**83.30**	87.78	91.50	92.03	92.85
AAPNet (Kong et al. 2020)		59.85	87.12	86.65	88.34	90.92
Ours		80.26	**89.86**	**92.87**	**94.08**	**94.43**
DBDNet (Pang et al. 2019)	3D CNN	82.67	88.35	90.58	**91.69**	**92.02**
T–S (Wang et al. 2019)		**83.32**	**88.92**	**90.85**	91.28	91.31
Transfer (Cai et al. 2019)		80.00	86.90	89.70	90.60	91.00
Ours		80.24	84.55	86.28	87.53	88.24

Best accuracy among the compared methods are indicated in bold

Table 5 Accuracies (%) of different methods on the BIT-Interaction dataset

Method	Observation ratio				
	0.1	0.3	0.5	0.7	0.9
RGN-KF (Zhao and Wildes 2019)	40.62	**67.96**	**81.25**	**92.28**	**92.19**
AAPNet (Kong et al. 2020)	39.84	64.84	80.47	88.28	91.40
DeepSCN (Kong et al. 2017)	37.5	59.38	78.13	86.72	90.63
MTSSVM (Kong et al. 2014b)	28.13	46.88	60.16	68.75	71.09
MSSC (Cao et al. 2013)	21.09	41.41	48.43	60.16	67.19
DBOW (Ryoo 2011)	23.44	41.41	47.66	54.69	55.47
IBOW (Ryoo 2011)	23.44	38.28	48.43	46.88	43.75
Ours	**46.09**	58.59	**81.25**	89.06	86.72

Best accuracy among the compared methods are indicated in bold

Table 6 Accuracies (%) of different methods on the HMDB51 dataset

Method	Observation ratio				
	0.1	0.3	0.5	0.7	0.9
Transfer (Cai et al. 2019)	43.50	51.20	56.40	**59.60**	61.10
Global-Local (Lai et al. 2018)	38.80	49.10	52.60	56.30	57.30
MTSSVM (Kong et al. 2014b)	13.60	26.70	33.80	37.50	37.50
MSSC (Cao et al. 2013)	12.40	24.90	33.80	37.80	38.80
Ours	**45.10**	**52.35**	**56.73**	59.41	**61.11**

Best accuracy among the compared methods are indicated in bold

Table 7 Accuracies (%) of ablation studies at different observation ratios on the CAD120 and 20BN-something-something datasets

Method	CAD120					20BN-something-something				
	Observation ratio					Observation ratio				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
w/o spatial and temporal	81.45	80.65	80.65	79.03	79.84	16.45	19.32	22.90	24.74	22.00
w/o temporal	82.26	86.29	86.29	86.29	85.48	19.01	21.81	24.17	25.13	23.34
w/o spatial	86.29	83.87	82.26	83.06	83.06	20.47	21.24	23.98	28.06	28.51
w/o semantic-loss	87.90	88.71	87.10	87.10	85.48	20.98	21.11	25.06	28.32	30.17
Vanilla GGNN	83.07	84.68	83.06	81.45	82.26	20.79	23.41	26.59	30.82	32.97
Ours	**90.32**	**90.73**	**91.53**	**91.13**	**91.13**	**22.61**	**25.02**	**28.28**	**32.24**	**34.08**

Best accuracy among the compared methods are indicated in bold

Table 8 Accuracies (%) of ablation studies at different observation ratios on the UCF101 datasets

Method	Observation ratio				
	0.1	0.3	0.5	0.7	0.9
w/o spatial and temporal	75.98	77.09	77.37	77.10	76.09
w/o temporal	77.96	78.15	78.41	78.50	77.23
w/o spatial	76.29	86.02	87.8	88.98	85.06
w/o semantic-loss	77.10	83.60	85.32	86.47	84.05
Vanilla GGNN	78.31	87.19	88.15	86.47	84.05
Ours	**80.86**	**88.61**	**89.31**	**90.31**	**89.82**

Best accuracy among the compared methods are indicated in bold

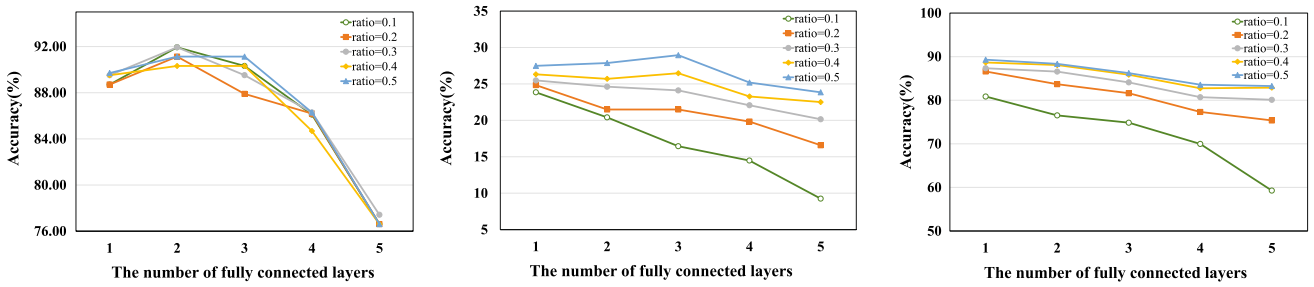
4.4 Ablation Study

To analyze the proposed approach in depth, ablation studies are conducted for empirically evaluating the importance of each individual component, including the spatial relation reasoning via IGGNN, the temporal relation reasoning via LSTGN and the visual semantic relation learning via VTransE based loss. Furthermore, we analyze the temporal relation reasoning by using different numbers of fully connected layers of $g_{node}(\cdot)$ and $g_{edge}(\cdot)$ in LSTGN. We conduct experiments on different temporal scales to go deep into the studying of multi-scale strategy in temporal relation reasoning. Finally, we analyze the effectiveness of the soft attention operator in spatial relation reasoning.

4.4.1 Evaluation on Importance of Each Component

Table 7 demonstrates the prediction accuracies of different individual components on the CAD120 dataset and the 20BN-something-something dataset. Table 8 illustrates the ablation study results of different individual components on the UCF101 dataset. "w/o spatial and temporal" represents the model without spatial and temporal reasoning, which concatenates the features of the detected bounding boxes of individual objects and the features of the union bounding boxes of the corresponding two objects as input, and directly uses cross-entropy loss as the objective function. "w/o temporal" means only performing spatial relation reasoning in video frames via IGGNN without capturing the temporal relation between objects. It takes the same input as "Ours", pools the spatial graph representations from IGGNN to represent the video-level features and uses the loss in Eq. 16 as the objective function. "w/o spatial" means only performing temporal relation reasoning via LSTGN without modeling the spatial relation between objects in each frame. It takes the same input as "Ours" and concatenates the features of the bounding boxes and the features of the union bounding boxes to represent the frame-level representations. The visual semantic loss is directly computed by the features of the bounding boxes of individual objects. The final loss in "w/o spatial" is also the same as "Ours". "w/o semantic" represents only using classification loss by removing the visual semantic relation loss during the training of the model. "vanilla GGNN" represents using GGNN to perform spatial relation reasoning and using LSTGN to perform temporal relation reasoning. It takes the same input as "Ours" and only feeds node features into GGNN. Also, it uses the loss in Eq. 16 as the objective function. From Tables 7 and 8, we can have the following observations:

- When removing the spatial relation reasoning or the temporal relation reasoning, the prediction results will substantially degrade at all the observation ratios, which validates that both these two relation reasoning are critical to the prediction performance.
- The spatial and temporal relation reasoning play different roles on different datasets. For the CAD120 dataset, the spatial relation reasoning generally contributes more than the temporal relation reasoning except at the observation ratio of 0.1, probably due to that different actions in the CAD120 dataset have similar motion patterns and are easy to be confused without the help of contextual information. Thus, it requires a more accurate recognition



(a) CAD120 **(b)** 20BN-something-something **(c)** UCF101

Fig. 4 Evaluation of different numbers of fully connected layers in LSTGN on the CAD120, 20BN-something-something and UCF101 datasets

Table 9 Accuracies (%) of different scales on the CAD120 and 20BN-something-something datasets

Scale	CAD120					20BN-something-something				
	Observation ratio					Observation ratio				
	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
1	82.26	85.48	86.29	83.87	81.45	20.77	21.15	23.47	25.42	26.15
2	89.52	89.52	90.32	88.71	87.10	20.43	21.81	24.17	25.72	26.42
3	90.32	90.32	90.32	85.49	87.10	22.28	23.26	25.70	28.85	29.11
4	**90.32**	**90.73**	**91.53**	**91.13**	**91.13**	**23.31**	24.37	27.49	30.74	31.32
5	87.91	87.91	88.71	89.52	87.90	22.61	**25.02**	**28.28**	**32.24**	**34.08**
6	86.29	88.71	88.71	88.71	89.52	22.00	22.96	26.24	30.34	31.56

Best accuracy among the compared methods are indicated in bold

of the scene via spatial relation reasoning to effectively predict the action. At the observation ratio of 0.1, the limited input leads to a light structure of LSTGN and accordingly promotes the learning of LSTGN, so "w/o spatial" achieves better performance.

- For the 20BN-something-something dataset, the spatial relation offers certain assistance at the early observation while the contribution of temporal relations becomes more significant at the later observation. The possible reason is that videos in the 20BN-something-something dataset show more fine-grained actions in simple background. Those videos lack auxiliary scene information to assist the prediction, so the temporal information is more crucial. Thus, it requires to gradually capture the temporal information with the increasing input so as to improve the prediction performance. For different scenarios, our method can effectively integrate the spatial and temporal relation reasoning with adaptive capacity to consistently improve the prediction results.
- Our method generally outperforms the "w/o semantic", which verifies that learning the visual semantic relation encourages the model to capture the association between the spatial–temporal object relations and action categories, thus promoting the performance.
- Our method outperforms "vanilla GGNN", which demonstrates that incorporating edge features in GGNN can help to improve the performance of action prediction. For the 20BN-something-something dataset, the result gap between "vanilla GGNN" and "Ours" at later observation is slighter than that at earlier observation, which also demonstrates that for this dataset, the contribution of temporal relations becomes more important at the later observation.

4.4.2 Evaluation on Different Numbers of Fully Connected Layers of $g_{node}(\cdot)$ and $g_{edge}(\cdot)$ in LSTGN

Figure 4 shows the prediction accuracies on the CAD120, 20BN-something-something and UCF101 datasets when using different numbers of fully connected (fc) layers of $g_{node}(\cdot)$ and $g_{edge}(\cdot)$ in LSTGN. It is interesting to observe that:

- For the CAD120 dataset, the results of 2 fc layers are better than that of 1 fc layer, which indicates that the deeper network captures more discriminative temporal relations among different frames.
- For the UCF101 dataset and the 20BN-something-something dataset, the optimal number of fc laryers is 1 for most observation ratios, probably due to that the visual objects in these two datasets are represented by powerful deep features and the shallower network could be satisfied.
- When the number of fc layers increases more than two, the prediction accuracies drop quickly for all the obser-

vation ratios. The probable reason is that too many parameters caused by many fc layers lead to the overfitting problem.

4.4.3 Evaluation on Multi-scale Receptive Fields in Temporal Relation Reasoning

To demonstrate the effectiveness of multi-scale receptive fields in temporal relation reasoning, we compare the prediction results of different temporal scales (from 1 to 6) on the CAD120 and 20BN-something-something datasets in Table 9. We also compare the prediction results of different temporal scales (from 1 to 10) on the UCF101 dataset in Table 10. Note that the scale 1 means only performing spatial relation reasoning and directly utilizing single node representation to classify the action. The scale n represents the fusion of results from scale 1 to scale n.

From Tables 9 and 10, we can have observe that:

- The performance obviously improves when conducting temporal relation reasoning with larger scale, which demonstrates the effectiveness of capturing both the short-term and long-term evolution of spatial relations.
- For the CAD120 dataset, when the scale is larger than 4, the performance slightly decreases, which is probably because the feature dimension of extracted temporal relations is too high to cause the overfitting problem and lead to a bad impact on the performance of the classifier. For the 20BN-something-something dataset, when the scale is larger than 5, the prediction accuracy decreases.
- For the UCF101 dataset, the optimal scale is 8 and is larger than the optimal scales on CAD120 and 20BN-something-something. The possible reason is that action videos in UCF101 are more complex and are composed of several fine-grained sub-actions, so they need a larger temporal scale to capture the motions.

4.4.4 Evaluation on the Effectiveness of Different Losses

To evaluate how the classification loss and the visual semantic relation loss affect the prediction performance, we conduct experiments with different values of the trade-off parameter λ (defined in Eq. 16) on the 20BN-something-something dataset. The results are shown in Table 11 where the larger λ represents the more effect of the visual semantic loss. It can be observed that smaller λ often achieve fairly better performance, which suggests the classification loss has much influence on the prediction results. When λ becomes zero, the accuracy significantly degrades and it can be concluded that the classification loss and the visual semantic relation loss work together to make a positive impact on the prediction performance.

Table 10 Accuracies (%) of different scales on the UCF101 datasets

Scale	Observation ratio				
	0.1	0.3	0.5	0.7	0.9
1	75.75	80.70	85.07	87.13	87.98
2	76.40	84.37	88.34	88.66	88.78
3	75.57	86.08	87.69	89.08	88.17
4	76.76	87.15	88.75	89.04	89.15
5	77.19	88.60	88.97	89.32	89.31
6	76.86	87.32	87.72	88.96	88.20
7	77.27	87.19	88.56	87.94	88.41
8	**80.86**	**88.61**	**89.31**	**90.31**	**89.82**
9	77.29	87.47	89.29	89.76	88.65
10	76.57	86.47	88.73	89.19	88.73

Best accuracy among the compared methods are indicated in bold

Table 11 Accuracies (%) of different λ on the 20BN-something-something dataset

λ	Observation ratio				
	0.1	0.3	0.5	0.7	0.9
0	20.98	21.11	25.06	28.32	30.17
1/8	**22.61**	**25.02**	**28.28**	**32.24**	**34.08**
1/6	22.39	22.19	25.70	29.27	31.06
1/4	21.94	23.15	24.94	28.64	30.68
1/2	20.60	20.92	21.94	26.02	27.23
2	21.94	22.00	25.57	28.70	30.74
4	20.60	20.98	23.15	26.21	27.30
6	21.36	21.62	23.21	25.96	26.72
8	20.79	21.05	22.64	25.89	27.49

Best accuracy among the compared methods are indicated in bold

4.4.5 Analysis on the Soft Attention Operator in Spatial Relation Reasoning

In order to make qualitative analysis on the soft attention operator in spatial relation reasoning, we visualize the attention weights of nodes (i.e., the detected objects) in Fig. 5. For each video frame, the detected objects with their corresponding attention weights are shown. It is interesting to notice that the higher the attention weight is, the object is considered to be more instructive and vice versa. Taking the action of "Turning something upside down" for example, the hand is assigned higher attention weight since the gesture is more instructive to recognize the action. For the action of "Pretending to open something without actually open it", the pillow is less instructive to the action and its attention weight is low.

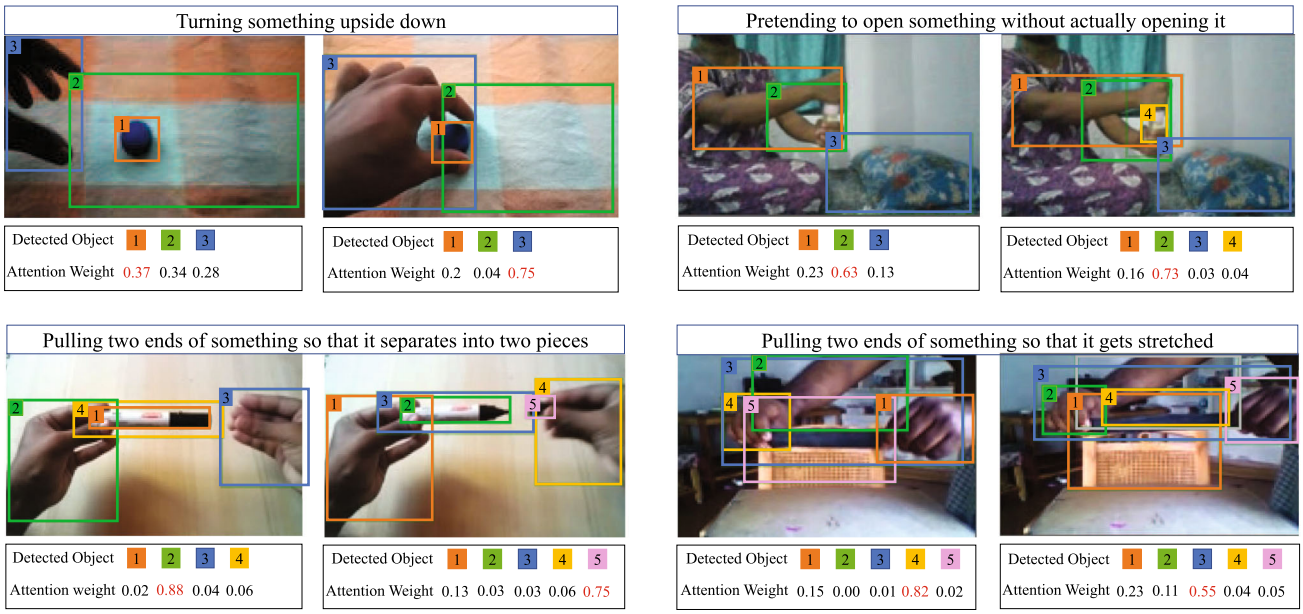


Fig. 5 Visualization of the attention weights. The colored boxes represent detected objects. The attention weight of each node (i.e., detected objects) in the spatial graph is shown below the video frames

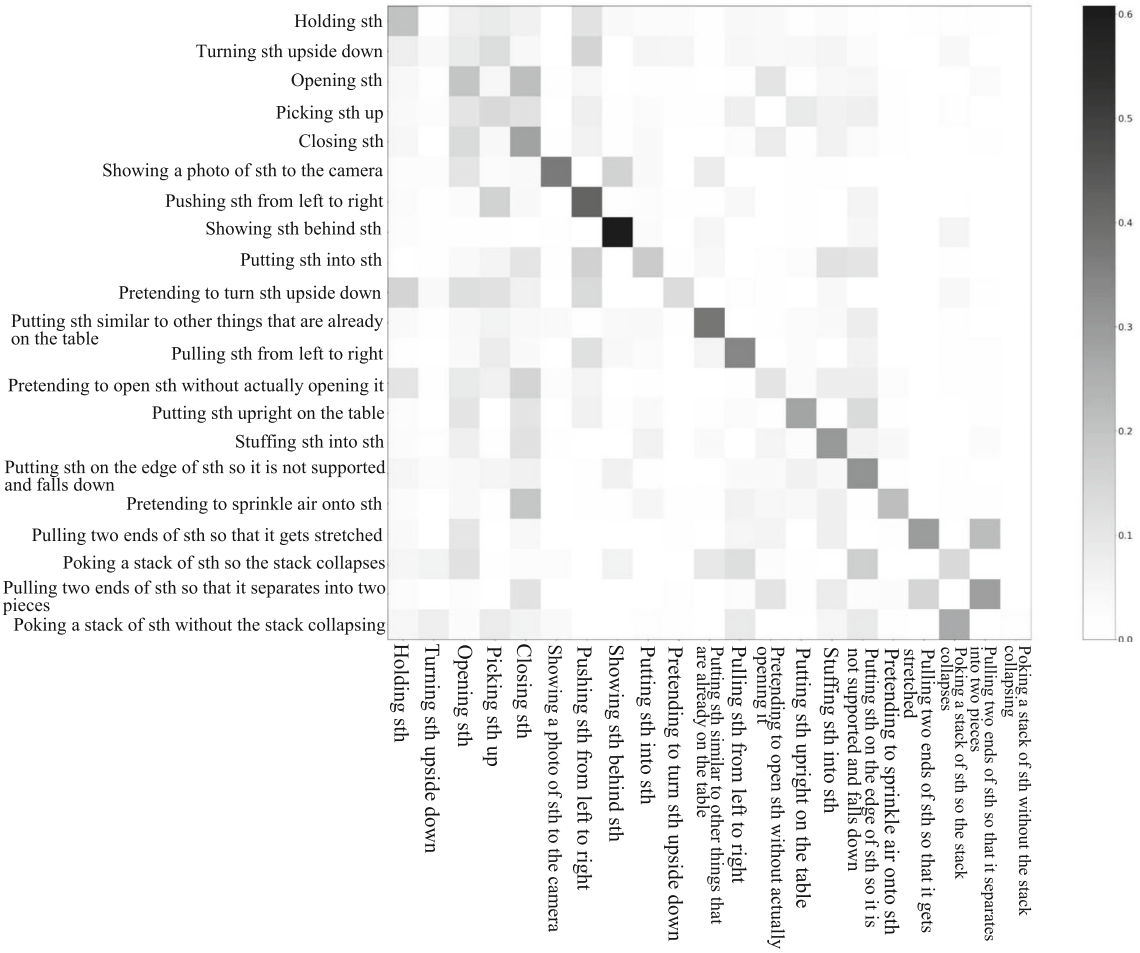


Fig. 6 Confusion matrix of the 20BN-something-something dataset. The prediction results are averaged over all the observation ratios

Fig. 7 Prediction examples on the 20BN-something-something dataset. The ground-truth label is given on the left side and predicted labels at different observation ratios are given under the video frames. The labels in blue represent correctly predicted labels and the labels in red represent wrongly predicted labels

4.5 Analysis on the Performance of Different Actions

To further discuss the prediction ability of our method, we calculate the accuracy of each action category on the 20BN-Something-Something dataset. As shown in Fig. 6, our method performs better on the actions of "Showing something behind something", "Pushing something from left to right" and "Pulling something from left to right". For the actions that are easily to be confused, e.g., "Putting something into something" and "Stuffing something into something", our model also recognizes them successfully, owing to the learned spatial–temporal relations that captures the motion dynamics in videos. For the actions of "Pulling two ends of something so that it gets stretched" and "Pulling two ends of something so that it separates into two pieces", they are usually misclassified, probably due to that the manual annotations for the object detector training have deviations, which makes the input node feature contain inaccurate spatial information, such as representing two separate objects as an integral object.

4.6 Analysis on How the Relationship Affects the Prediction Results

Figure 7 shows some exemplars of prediction results at different observation ratios on the 20BN-something-something dataset. Figure 7a shows positive examples and it suggests that our model makes effective use of relations in videos, such as relative positional relationships and their variations through time, to make accurate prediction in the early stage

Fig. 8 Examples of prediction failure statistics on the 20BN-something-something dataset. For each action category, the histograms represent the distributions of all the corresponding test videos that are classified into the six most probable categories at different observation ratios. These six categories are chosen according to the average classification probability of the test videos across all the observation ratios. The horizontal axis in each histogram indicates the category index and the corresponding labels of the six categories are provided in the right box where the green label represents the true category of the test videos and the blue labels represent false categories. The histogram bar indicates the percentage of the test videos that are classified into the corresponding category

of the video. We also provide some examples in Fig. 7b to discuss how our model gradually reasons the action. Taking the action of "Pretending to open something without actually opening it" for instance, given the first 10% frames, the relative positional relation between the hand and the box makes the action look almost still, which confuses our model to make the prediction that the action is "holding something". With the increasing input video frames, our model captures the minor changes in the appearance of the box and deduces the action of "closing something". After half of the video has been observed, the evolutions of the global spatial relations in sequential video frames are captured and the correct action label is predicted.

To further analyze how the relationship affects the prediction results, we calculate the percentage of prediction failures falling into each of the three action categories (shown in Fig. 7b). The changes of failure percentage of test videos at different observation ratios for the three action categories (i.e., "Pulling sth from left to right", "Pulling two ends of sth so that it separates into two pieces" and "Pretending to open sth without actually opening it") are shown in Fig. 8. Each histogram below the video frame represents the distribution of the test videos that are classified into the six most probable categories at the current observation ratio. These six categories are chosen according to the average classification probability of the test videos across all the observation ratios. The horizontal axis in each histogram indicates the category index and the corresponding labels of the six categories are provided in the right box where the green label represents the true category of the test videos and the blue labels represent false categories. The histogram bar indicates the percentage of the test videos that are classified into the corresponding

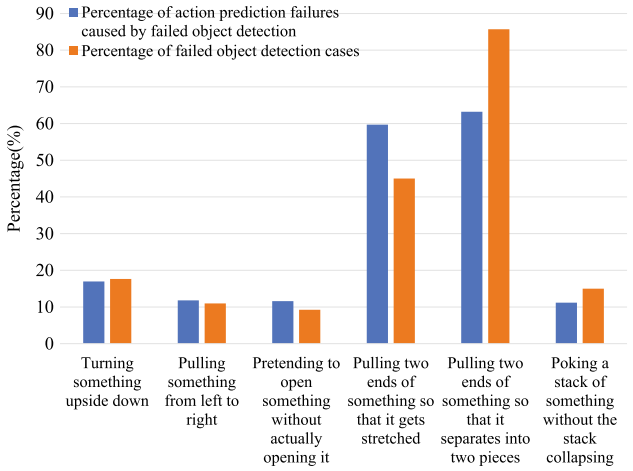


Fig. 9 Percentage of action prediction failures caused by failed object detection on the 20BN-something-something dataset. The orange histogram bar represents the percentage of failed object detection cases and the blue histogram bar stands for the percentage of action prediction failures that can be attributed to failed object detection

category. Taking the action of "Pulling something from left to right" as an example, in the early stage at the observation ratio of 10%, the most probable category is "Picking something up" (27.4.% of the test videos are classified into "Picking something up") since the relative position relationships between the hand and the object are similar in the two actions of "Pulling something from left to right" and "Picking something up". At the observation ratio of 50%, the distance between the hand and the object becomes closer and there is no change in the position of the hand and object in vertical space, thus less test videos are misclassified into "Picking something up" and more test videos are misclassified into "Putting something upright on the table" or "Putting something similar to other things that are already on the table". After observing the whole video, most test videos are correctly classified into "Pulling something from left to right".

4.7 Analysis on Failures of Our Model

We further analyze the prediction failures of our model. We choose six difficult action categories that are easily misclassified according to the confusion matrix in Fig. 6 and count the number of action prediction failures attributed to object detection failures in all the failed cases of these six action categories. Then the percentage of prediction failures that can be attributed to detection failures is calculated. The results are shown in Fig. 9 where the orange histogram bar represents the percentage of failed object detection cases and the blue histogram bar stands for the percentage of action prediction failures that can be attributed to the failed object detection. It is interesting to observe that more failed object detection cases lead to more misclassified actions. Taking the action of "Pulling two ends of something so that it separates into two pieces" for instance, there are several irrelated objects that occupy the majority of the space in the video frame, which leads to a failed detection of the related objects and misguides the model. In such videos, more than 60% failures can be attributed to the failed object detection. For the action of "Turning something upside down", the background is clear and the detector works well, thus there are relatively less failures can be attributed to the failed detection. Figure 9 also demonstrates that the performance of our model depends on the effect of the object detector to a certain extent, which is the main weakness of our method.

5 Conclusion

We have presented a spatial–temporal relation reasoning approach for action prediction from partial videos. An improved gated graph neural network has been designed to capture the spatial relations between visual objects in video frames. A long short-term graph network has been proposed to learn the varied dynamics of the spatial relations in multiple temporal scales. Thus, our method can successfully make an accurate recognition of the video content with fine-grained object relations in both spatial and temporal domains to make prediction decisions. Extensive experiments on five public action datasets have shown the superior performances of our method. In the future, we are going to incorporate prior knowledge of visual object relation into our model to further boost the spatial–temporal relation reasoning even when the object detection fails. We will also investigate on automatically learning the optimal network network structure for action prediction when handling different datasets.

Acknowledgements This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant Nos. 61673062 and 62072041.

References

Aditya, S., Yang, Y., & Baral, C. (2018). Explicit reasoning over end-to-end neural architectures for visual question answering. In *Thirty-second AAAI conference on artificial intelligence*.

Aliakbarian, M. S., Saleh, F. S., Salzmann, M., Fernando, B., Petersson, L., & Andersson, L. (2017). Encouraging lstms to anticipate actions very early. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., et al. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Eighteenth ACM-SIAM symposium on discrete algorithms*.

Bhoi, A. (2019). *Spatio-temporal action recognition: A survey*. arXiv preprint arXiv:1901.09403.

Cai, Y., Li, H., Hu, J. F., & Zheng, W. S. (2019). Action knowledge transfer for action prediction with partial videos. In *Proceedings of the AAAI conference on artificial intelligence*.

Cao, Y., Barrett, D., Barbu, A., Narayanaswamy, S., Yu, H., Michaux, A., et al. (2013). Recognize human activities from partially observed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Chen, L., Lu, J., Song, Z., & Zhou, J. (2018a). Part-activated deep reinforcement learning for action prediction. In *European conference on computer vision*.

Chen, X., Li, L. J., Fei-Fei, L., & Gupta, A. (2018b). Iterative visual reasoning beyond convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Evans, J. S. B., Over, D. E., & Manktelow, K. I. (1993). Reasoning, decision making and rationality. *Cognition*, *49*(1–2), 165–187.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th international conference on machine learning*.

Girshick, R. (2015). Fast r-CNN. In *Proceedings of the IEEE international conference on computer vision*.

Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., & He, K. (2018). *Detectron*. https://github.com/facebookresearch/detectron.

Goyal, R., Kahou, S. E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., et al. (2017). The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*.

Hanwang, Z., Kyaw, Z., Chang, S., & Chua, T. (2017). Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hara, K., Kataoka, H., & Satoh, Y. (2018). Can spatiotemporal 3d CNNs retrace the history of 2d CNNs and imagenet? In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Herzig, R., Levi, E., Xu, H., Gao, H., Brosh, E., Wang, X., et al. (2019). Spatio-temporal action graph networks. In *Proceedings of the IEEE international conference on computer vision workshops*.

Hu, J. F., Zheng, W. S., Ma, L., Wang, G., Lai, J., & Zhang, J. (2018). Early action prediction by soft regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(11), 2568–2583.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*.

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

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., et al. (2017). *The kinetics human action video dataset*. arXiv preprint arXiv:1705.06950.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International conference on learning representations*.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th International conference on learning representations*.

Kong, Y., & Fu, Y. (2016). Max-margin action prediction machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *9*, 1844–1858.

Kong, Y., Gao, S., Sun, B., & Fu, Y. (2018). Action prediction from videos via memorizing hard-to-predict samples. In *AAAI conference on artificial intelligence*.

Kong, Y., Jia, Y., & Fu, Y. (2014a). Interactive phrases: Semantic descriptionsfor human interaction recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *9*, 1775–1788.

Kong, Y., Kit, D., & Fu, Y. (2014b). A discriminative model with multiple temporal scales for action prediction. In *European conference on computer vision*.

Kong, Y., Tao, Z., & Fu, Y. (2017). Deep sequential context networks for action prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Kong, Y., Tao, Z., & Fu, Y. (2020). Adversarial action prediction networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(3), 539–553.

Koppula, H. S., Gupta, R., & Saxena, A. (2013). Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, *32*(8), 951–970.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: A large video database for human motion recognition. In *Proceedings of the international conference on computer vision*.

Lai, S., Zheng, W. S., Hu, J. F., & Zhang, J. (2018). Global-local temporal saliency action prediction. *IEEE Transactions on Image Process*, *27*(5), 2272–2285.

Lan, T., Chen, T. C., & Savarese, S. (2014). A hierarchical representation for future action prediction. In *European conference on computer vision* (pp. 689–704).

Li, K., & Fu, Y. (2014). Prediction of human activity by discovering temporal sequence patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(8), 1644–1657.

Li, Y., Tarlow, D., Brockschmidt, M., & Zemel, R. S. (2016). Gated graph sequence neural networks. In *4th International conference on learning representations*.

Liang, K., Guo, Y., Chang, H., & Chen, X. (2018). Visual relationship detection with deep structural ranking. In *AAAI conference on artificial intelligence*.

Liao, W., Rosenhahn, B., Shuai, L., & Ying Yang, M. (2019). Natural language guided visual relationship detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*.

Liu, J., Luo, J., & Shah, M. (2009). Recognizing realistic actions from videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Lu, C., Krishna, R., Bernstein, M., & Fei-Fei, L. (2016). Visual relationship detection with language priors. In *European conference on computer vision*.

Newell, A., & Deng, J. (2017). Pixels to graphs by associative embedding. In *Advances in neural information processing systems*.

Nicolicioiu, A., Duta, I., & Leordeanu, M. (2019). Recurrent spacetime graph neural networks. In *Advances in neural information processing systems*.

Pang, G., Wang, X., Hu, J. F., Zhang, Q., & Zheng, W. S. (2019). Dbdnet: Learning bi-directional dynamics for early action prediction. In *Proceedings of the 28th international joint conference on artificial intelligence*.

Qi, M., Li, W., Yang, Z., Wang, Y., & Luo, J. (2019). Attentive relational networks for mapping images to scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Ryoo, M. S. (2011). Human activity prediction: Early recognition of ongoing activities from streaming videos. In *IEEE international conference on computer vision*.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80.

Shang, X., Ren, T., Guo, J., Zhang, H., & Tat-Seng, C. (2017). Video visual relation detection. In *Proceedings of the 25th ACM international conference on multimedia*.

Si, C., Jing, Y., Wang, W., Wang, L., & Tan, T. (2018). Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European conference on computer vision*.

Soomro, K., Zamir, A. R., & Shah, M. (2012). *Ucf101: A dataset of 101 human actions classes from videos in the wild*. arXiv preprint arXiv:1212.0402.

Sun, C., Shrivastava, A., Vondrick, C., Sukthankar, R., Murphy, K., & Schmid, C. (2019). Relational action forecasting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tsai, Y. H. H., Divvala, S., Morency, L. P., Salakhutdinov, R., & Farhadi, A. (2019). Video relationship reasoning using gated spatio-temporal energy graph. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. In *ICLR*.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., et al. (2016). Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*.

Wang, X., & Gupta, A. (2018). Videos as space-time region graphs. In *Proceedings of the European conference on computer vision*.

Wang, X., Hu, J. F., Lai, J. H., Zhang, J., & Zheng, W. S. (2019). Progressive teacher-student learning for early action prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Woo, S., Kim, D., Cho, D., & Kweon, I. S. (2018). Linknet: Relational embedding for scene graph. In *Advances in neural information processing systems*.

Xu, H., Jiang, C., Liang, X., & Li, Z. (2019). Spatial-aware graph relation network for large-scale object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A., & Elhoseiny, M. (2019). Large-scale visual relationship understanding. In *Proceedings of the AAAI conference on artificial intelligence*.

Zhao, H., & Wildes, R. P. (2019). Spatiotemporal feature residual propagation for action prediction. In *Proceedings of the IEEE international conference on computer vision*.

Zhou, B., Andonian, A., Oliva, A., & Torralba, A. (2018). Temporal relational reasoning in videos. In *Proceedings of the European conference on computer vision*.