HAIR: Hierarchical Visual-Semantic Relational Reasoning for Video Question Answering

Fei Liu[1,2] Jing Liu[1,2*] Weining Wang[1] Hanqing Lu[1,2]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
liufei2017@ia.ac.cn {jliu, weining.wang, luhq}@nlpr.ia.ac.cn

Abstract

Relational reasoning is at the heart of video question answering. However, existing approaches suffer from several common limitations: (1) they only focus on either object-level or frame-level relational reasoning, and fail to integrate the both; and (2) they neglect to leverage semantic knowledge for relational reasoning. In this work, we propose a ***H****ierarchical Visu****A****l-Semantic Relat****I****onal* ***R****easoning (HAIR) framework to address these limitations. Specifically, we present a novel graph memory mechanism to perform relational reasoning, and further develop two types of graph memory: a) visual graph memory that leverages visual information of video for relational reasoning; b) semantic graph memory that is specifically designed to explicitly leverage semantic knowledge contained in the classes and attributes of video objects, and perform relational reasoning in the semantic space. Taking advantage of both graph memory mechanisms, we build a hierarchical framework to enable visual-semantic relational reasoning from object level to frame level. Experiments on four challenging benchmark datasets show that the proposed framework leads to state-of-the-art performance, with fewer parameters and faster inference speed. Besides, our approach also shows superior performance on other video+language task.*

1. Introduction

Video Question Answering (VideoQA), an emerging task that requires machines to answer questions about videos in a natural language form, has recently drawn increasing interests from researchers. The task is particularly challenging, as it requires fine-grained understanding of video content involving various complex relations such as object-object relation, frame-frame relation *etc.* Thus, relational reasoning plays an important role in solving VideoQA problem. Recent works [9, 12, 14, 28, 20, 43] have introduced memory networks [44, 35], attention mechanisms [46] or Graph Convolutional Networks (GCNs) [22] for relational reasoning in VideoQA. Although achieving promising results, these existing approaches suffer from two common limitations.

*Corresponding author.

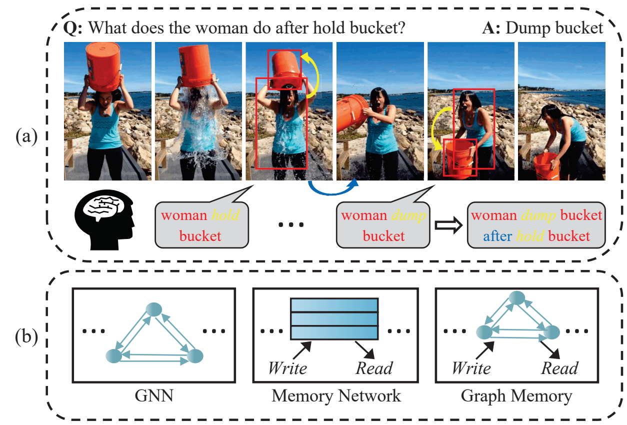

Figure 1. (a) Hierarchical relational reasoning. Humans perform object-level first and then frame-level relational reasoning for understanding the whole video content. (b) A concise comparison of vanilla GNN, memory network and our graph memory.

First, current approaches for VideoQA only focus on either object-level [14] **or frame-level relational reasoning** [9, 12, 26, 51, 20]**, and do not integrate the both in a hierarchical manner.** Given a video clip and an associated question, as shown in Figure 1(a), a typical reasoning process for human is that we first recognize relevant objects and their interaction in each video frame (*e.g. woman hold bucket, woman dump bucket*), and then correlate these frames to understand a sequence of actions and their temporal relationship (*e.g. woman dump bucket after hold bucket*). Finally, the correct answer can be naturally derived based on the understanding of video content. Such a process of relational reasoning is conducted in a hierarchical way, *i.e.*, from object level to frame level. It is desired to endow the machines with the same characteristic as human. However, none of current approaches have attempted to explicitly perform *hierarchical* relational reasoning. These approaches may miss the modeling of some crucial relations that are necessary for answering questions correctly.

Second, current approaches for VideoQA only consider visual information for relational reasoning, and neglect the reasoning in the semantic space. In [26, 20, 28], the proposed approaches perform relational reasoning over video frame features extracted by CNN. Huang *et al*. [14] and Jin *et al*. [19] exploited object-level visual information using RCNN. These methods neglect to leverage semantic knowledge for relational reasoning, possibly leading to the misunderstanding of visual content due to the inherent semantic gap. Compared to visual information, semantic knowledge (*e.g*. the attributes and classes of multiple objects) provides more explicit and richer cues to benefit the reasoning, which has been demonstrated in the image recognition domain [29, 7].

In this work, in an effort to address the aforementioned limitations, we put forward a **H**ierarchical Visu**A**l-Semantic Relat**I**onal **R**easoning (HAIR) framework, which jointly performs *visual* and *semantic* relational reasoning in a hierarchical structure (Figure 2). The core component of the framework is the graph memory mechanism, inspired by graph neural network (GNN) [40] and memory network [44]. The GNN can pass message among nodes, which is a natural choice to perform relational reasoning. While the memory network is able to gradually distill query-related information through read and write operations. Here, we marry GNN with memory network to inherit the advantages of the both, enabling more efficient relational reasoning. A concise comparison of vanilla GNN, memory network and our graph memory is shown in Figure 1(b). Moreover, we develop two types of graph memory mechanisms: a) *visual graph memory*, which exploits *visual* information of video for relational reasoning, and gradually learns query-related relation-aware *visual* representation; b) *semantic graph memory*, where we represent object classes and attributes as nodes and build edges to encode commonsense *semantic* relationships. It explicitly leverages *semantic* knowledge to facilitate relational reasoning. The two graph memory mechanisms work cooperatively and interact with each other via learnable visual-to-semantic and semantic-to-visual node mapping. Finally, taking advantage of the proposed graph memory mechanisms, we build a hierarchical structure, from object to frame level, thus enabling hierarchical visual-semantic relational reasoning.

In summary, the contributions of this work are threefold: (1) We present graph memory, a novel relational reasoning mechanism. Furthermore, we develop visual graph memory and semantic graph memory to reason over different types of information. (2) We propose a hierarchical visual-semantic relational reasoning (HAIR) framework to integrate object-level and frame-level relational reasoning in a hierarchical manner. (3) Experimental results show that our framework achieves state-of-the-art performance on four datasets for VideoQA, with fewer parameters and faster inference speed. Our approach also shows superior performance on other video+language tasks, *e.g*., language-based temporal grounding.

2. Related Work

Video Question Answering. The Video Question Answering (VideoQA) task is an extension of Image Question Answering (ImageQA). Compared with the well-studied ImageQA which focuses on understanding static images [2, 52, 1, 30, 31], VideoQA is much more challenging because of the existence of extra temporal domain. When solving the VideoQA problem, one repuires to figure out various complex relations, such as spatial, temporal, visual and semantic relations to reason about answer. A lot of efforts have been made to explore relational reasoning in VideoQA. In [28, 26, 20, 27, 18], the proposed methods represent each video frame as global feature vector, hence only frame-level relational reasoning is considered. In particular, Li *et al*. [28] and Kim *et al*. [20] used a self-attention [46] based technique to model global dependencies among frames of a video. Jiang *et al*. [18] proposed heterogeneous graph alignment (HGA) network. These approaches lack the exploitation of fine-grained information on spatial dimension, and are thus struggling to answer questions involving multiple objects and their relations. To alleviate this issue, Huang *et al*. [14] proposed to reason over detected objects with location-aware graph convolutional network, but failed to explore frame-level relational reasoning. Unlike these works that focus on either frame-level or object-level relational reasoning, our HAIR framework mimics the cognition process of human [10, 23, 39] and performs *hierarchical* relational reasoning.

GNN & Memory Network. Graph Neural Network (GNN) is able to easily pass message among nodes and update node representation iteratively, which is very suitable to learn relational reasoning. As a result, GNN has been broadly applied in many fields, such as image domain (including image recognition [8, 45], pose estimation [3], *etc.*) and video domain (including action recognition [42, 41], video object segmentation [48], *etc.*). However, for multimodal tasks, the relational reasoning needs to absorb necessary query information and should be under the dynamic guidance of query, in order to retrieve relevant information at each iteration step. For these, GNN cannot handle them well, although some works [36, 11] attempted to represent node as the fusion of visual and query feature. Memory network is first introduced in [49, 44], which allows the model to explicitly retrieve and store information by read and write operations. It has been proven to be effective in multimodal

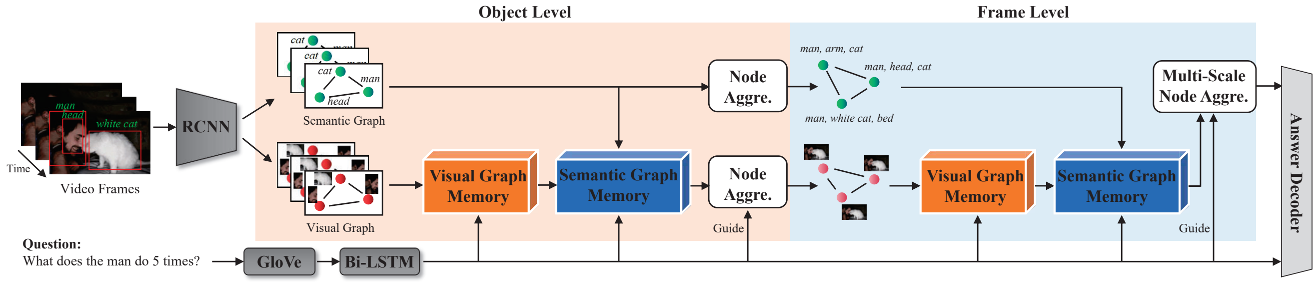

Figure 2. Hierarchical Visual-Semantic Relational Reasoning (HAIR) framework for VideoQA. It first extracts the input representations, and constructs visual graph and semantic graph at object level. After that, both graph memory mechanisms perform object-level relational reasoning over visual and semantic representations, respectively. Node aggregation is used to aggregate nodes for each frame and build new graphs at frame level. Next, both gragh memory mechanisms perform frame-level relational reasoning over visual and semantic representations, respectively. Multi-scale node aggregation captures multi-scale temporal information and produces global representation of video. Finally, the answer decoder fuses the multimodal representations to infer the answer.

QA task [50, 12, 9], where memory network is able to gradually and dynamically learn query-related information. Inspired by these, we marry GNN with memory network to enable dynamic relational reasoning under the guidance of query. We call it *graph memory*. We show the proposed graph memory performs much better than GNN and other variants in Sec. 4.3.

Relational Reasoning. Relational reasoning has been explored in other video understanding tasks besides VideoQA. Huang *et al*. [15] proposed a dynamic graph module to model object-object interactions in video activities. Ma *et al*. [33] utilized an LSTM to model interactions between arbitrary subgroups of objects. However, these methods only perform relational reasoning over *visual object*, possibly resulting in incomplete understanding of video due to the lack of frame-level reasoning and semantic knowledge. Mavroudi *et al*. [34] proposed to build an additional symbolic graph using action categories. However, their method only operates at object level. In comparison, our HAIR is a hierarchical relational reasoning framework. We believe this is the first attempt to: (1) consider semantic knowledge to facilitate relational reasoning; and (2) explore both object-level and frame-level relational reasoning in a hierarchical way for VideoQA.

3. Our Approach

In this section, we present an end-to-end trainable framework – Hierarchical Visual-Semantic Relational Reasoning (HAIR) for VideoQA. The overall architecture is illustrated in Figure 2. We begin with the introduction of the both graph memory mechanisms (*i.e*. visual graph memory and semantic graph memory) in Sec. 3.1, then present the overall architecture in Sec. 3.2.

3.1. Graph Memory

The graph memory consists of a fully-connected graph and read-write controllers. The fully-connected graph allows to fully explore the relations among nodes. The controllers carry query information and interact with the node representations by a series of read and write operations. We develop two types of graph memory: visual graph memory and semantic graph memory, to reason over different representations.

3.1.1 Visual Graph Memory

The visual graph memory performs iterative relational reasoning over visual representations, as shown in Figure 3. Since our approach contains read and write operations of memory network, we follow a similar style to describe our graph memory.

Read Operation. Let $\boldsymbol{q}^{(0)} \in \mathbb{R}^d$ denote the initial state of read controller and $\boldsymbol{v}_i^{(0)} \in \mathbb{R}^d$ denote the initial representation of the i-th graph node. At each reasoning step $k \in \{1, ..., K_v\}$, the read controller attentively reads the content $\boldsymbol{r}^{(k)}$ from all nodes:

$$a_i'^{(k)} = \boldsymbol{V}_r^a \tanh(\boldsymbol{W}_r^a \boldsymbol{q}^{(k-1)} + \boldsymbol{U}_r^a \boldsymbol{v}_i^{(k-1)}) \tag{1}$$

$$a_i^{(k)} = \exp(a_i'^{(k)}) / \textstyle\sum_j \exp(a_j'^{(k)}) \tag{2}$$

$$\boldsymbol{r}^{(k)} = \textstyle\sum_i a_i^{(k)} \boldsymbol{v}_i^{(k-1)} \tag{3}$$

where $\boldsymbol{W}_r^a, \boldsymbol{U}_r^a$ and $\boldsymbol{V}_r^a$ are learnable weights (bias term is omitted for simplicity). Once acquiring the node content $\boldsymbol{r}^{(k)}$, the read controller updates its state as follows:

$$\widetilde{\boldsymbol{q}}^{(k)} = \boldsymbol{W}_r^h \boldsymbol{q}^{(k-1)} + \boldsymbol{U}_r^h \boldsymbol{r}^{(k)} \tag{4}$$

$$\boldsymbol{g}^{(k)} = \sigma(\boldsymbol{W}_r^g \boldsymbol{q}^{(k-1)} + \boldsymbol{U}_r^g \boldsymbol{r}^{(k)}) \tag{5}$$

$$\boldsymbol{q}^{(k)} = \boldsymbol{g}^{(k)} \circ \widetilde{\boldsymbol{q}}^{(k)} + (\boldsymbol{1} - \boldsymbol{g}^{(k)}) \circ \boldsymbol{q}^{(k-1)} \tag{6}$$

where $\boldsymbol{W}$s and $\boldsymbol{U}$s are learnable weights. σ and $\circ$ represent the sigmoid function and Hadamard product, respectively. The update gate $\boldsymbol{g}^{(k)}$ controls how much previous state to be preserved.

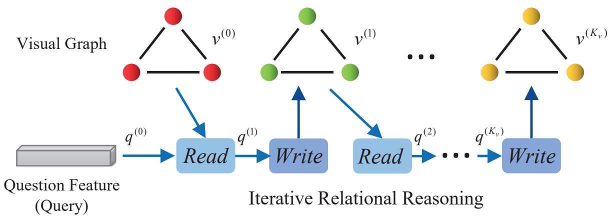

Figure 3. Illustration of Visual Graph Memory (VGM).

Write Operation. After the read operation, we need to update the node representations with new query information and relations among nodes. At each step k, the write controller updates the i-th node by considering its previous representation $\boldsymbol{v}_i^{(k-1)}$, current content from the read controller $\boldsymbol{q}^{(k)}$ and the representations from other nodes $\{\boldsymbol{v}_j^{(k-1)}\}_{j \neq i}$. Concretely, we first aggregate the information from neighbor nodes to capture the context:

$$e'^{(k)}_{i,j} = \mathrm{MLP}([\boldsymbol{v}_i^{(k-1)}; \boldsymbol{v}_j^{(k-1)}]) \quad (7)$$

$$e^{(k)}_{i,j} = \exp(e'^{(k)}_{i,j}) / \textstyle\sum_{j \neq i} \exp(e'^{(k)}_{i,j}) \quad (8)$$

$$\boldsymbol{c}_i^{(k)} = \textstyle\sum_{j \neq i} e^{(k)}_{i,j} \boldsymbol{v}_j^{(k-1)} \quad (9)$$

where MLP is Multi-Layer Perceptron consisting of two linear layers with the ReLU activation in between, $e^{(k)}_{i,j}$ is the relation weight from the j-th to i-th node, and $[\cdot;\cdot]$ denotes the feature concatenation. After obtaining the context representation $\boldsymbol{c}_i^{(k)}$, the write controller updates the node representation as:

$$\widetilde{\boldsymbol{v}}_i^{(k)} = \boldsymbol{W}_u^v \boldsymbol{q}^{(k)} + \boldsymbol{U}_u^v \boldsymbol{v}_i^{(k-1)} + \boldsymbol{V}_u^v \boldsymbol{c}_i^{(k)} \quad (10)$$

$$\boldsymbol{g}_i^{(k)} = \sigma(\boldsymbol{W}_u^g \boldsymbol{q}^{(k)} + \boldsymbol{U}_u^g \boldsymbol{v}_i^{(k-1)} + \boldsymbol{V}_u^g \boldsymbol{c}_i^{(k)}) \quad (11)$$

$$\boldsymbol{v}_i^{(k)} = \boldsymbol{g}_i^{(k)} \circ \widetilde{\boldsymbol{v}}_i^{(k)} + (\boldsymbol{1} - \boldsymbol{g}_i^{(k)}) \circ \boldsymbol{v}_i^{(k-1)} \quad (12)$$

As shown in Eq.1-12, our graph memory retains the advantage of GNN and is capable of modeling the relations among visual representations. Meanwhile, it possesses the read and write controllers of memory network, thus enabling dynamic interaction between query and visual representations and dynamic selection of relevant information (due to the internal gating mechanism).

The full process of iterative reasoning can be written as:

$$\boldsymbol{v}^{(K_v)} = \mathrm{VGM}(\boldsymbol{q}^{(0)}, \boldsymbol{v}^{(0)}) \quad (13)$$

where VGM represents visual graph memory, $\boldsymbol{q}^{(0)}$ is the initial state of the read controller, $\boldsymbol{v}^{(0)} = \{\boldsymbol{v}_i^{(0)}\}_{i=1}^{|V|}$ is the initial *visual* representations of graph nodes (where $|V|$ is the number of nodes), and $\boldsymbol{v}^{(K_v)}$ is the updated representations after K_v reasoning steps.

3.1.2 Semantic Graph Memory

The semantic graph memory leverages semantic knowledge and performs iterative relational reasoning over semantic representations, as shown in Figure 4. It has three inputs: the initial state of the read controller $\boldsymbol{q}^{(0)} \in \mathbb{R}^d$, the initial representations of the semantic graph $\boldsymbol{s}^{(0)} \in \mathbb{R}^{|S| \times d}$, and the updated representations of the visual graph $\boldsymbol{v}^{(K_v)} \in \mathbb{R}^{|V| \times d}$, where $|S|$ and $|V|$ represent the number of nodes. As a first step, we enhance the semantic representations using visual evidence. To achieve this, we introduce a learnable visual-to-semantic node mapping mechanism:

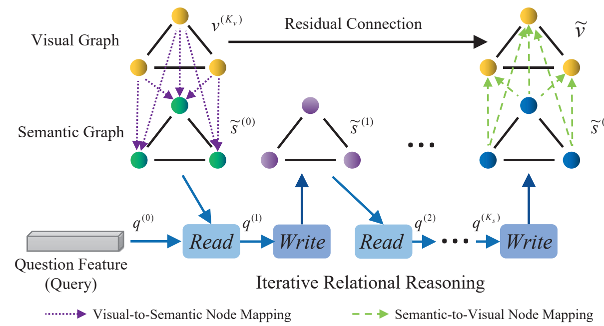

Figure 4. Illustration of Semantic Graph Memory (SGM).

$$\phi_{v_j \to s_i} = \exp(\boldsymbol{W}_i^{vs} \boldsymbol{v}_j^{(K_v)}) / \textstyle\sum_{i'=1}^{|S|} \exp(\boldsymbol{W}_{i'}^{vs} \boldsymbol{v}_j^{(K_v)}) \quad (14)$$

$$\boldsymbol{f}_i^{vs} = \textstyle\sum_{j=1}^{|V|} \phi_{v_j \to s_i} \boldsymbol{W}_p^v \boldsymbol{v}_j^{(K_v)} \quad (15)$$

where $\phi_{v_j \to s_i}$ represents the confidence of mapping the feature from the j-th *visual* node to the i-th *semantic* node, $\boldsymbol{W}^{vs} = \{\boldsymbol{W}_i^{vs}\}_{i=1}^{|S|} \in \mathbb{R}^{|S| \times d}$ is a trainable weight matrix for calculating voting weights, and $\boldsymbol{W}_p^v \in \mathbb{R}^{d \times d}$ is a projection weight matrix. The representation of each semantic node is updated as: $\widetilde{\boldsymbol{s}}_i^{(0)} = [\boldsymbol{s}_i^{(0)}; \boldsymbol{f}_i^{vs}]$.

Then, we perform iterative relational reasoning over the enhanced semantic representations $\widetilde{\boldsymbol{s}}^{(0)}$. The read and write operations are identical with those in the visual graph memory, defined in Eq.1-12. After K_s reasoning steps, we obtain the updated semantic representations $\widetilde{\boldsymbol{s}}^{(K_s)} = \{\widetilde{\boldsymbol{s}}_i^{(K_s)}\}_{i=1}^{|S|}$, which is then mapped back into visual space to enrich the visual representation with global semantic knowledge via a semantic-to-visual node mapping:

$$\phi'_{s_j \to v_i} = \boldsymbol{W}_i^{sv} [\widetilde{\boldsymbol{s}}_j^{(K_s)}; \boldsymbol{v}_i^{(K_v)}] \quad (16)$$

$$\phi_{s_j \to v_i} = \exp(\phi'_{s_j \to v_i}) / \textstyle\sum_{j=1}^{|S|} \exp(\phi'_{s_j \to v_i}) \quad (17)$$

$$\boldsymbol{f}_i^{sv} = \textstyle\sum_{j=1}^{|S|} \phi_{s_j \to v_i} \boldsymbol{W}_p^s \widetilde{\boldsymbol{s}}_j^{(K_s)} \quad (18)$$

where $\boldsymbol{W}_i^{sv} \in \mathbb{R}^{1 \times 2d}$ and $\boldsymbol{W}_p^s \in \mathbb{R}^{d \times d}$ are learnable projection weights. Through the two node mapping mechanisms, the visual graph memory and the semantic graph memory work cooperatively and interact with each other, to achieve a better relational reasoning and a more comprehensive understanding of video content. The final representation of the i-th visual node is obtained using a residual connection: $\widetilde{\boldsymbol{v}}_i = \boldsymbol{v}_i^{(K_v)} + \boldsymbol{f}_i^{sv}$.

The entire process can be concisely written as:

$$\widetilde{\boldsymbol{v}} = \mathrm{SGM}(\boldsymbol{q}^{(0)}, \boldsymbol{s}^{(0)}, \boldsymbol{v}^{(K_v)}) \quad (19)$$

3.2. Overall Architecture

In this subsection, we present the overall architecture of our hierarchical visual-semantic relational reasoning (HAIR) framework (see Figure 2), based on the definition of the graph memory in Sec. 3.1.

Input Embedding. Given a video containing T frames, we use a modified Faster R-CNN [38] pre-trained on the VGenome [25] to extract the visual features of N objects from each frame. To capture the object's spatial location, we introduce a 4-dimensional location feature from the object's relative bounding box coordinates $[x_{\min}/W_{\mathrm{fr}}, y_{\min}/H_{\mathrm{fr}}, x_{\max}/W_{\mathrm{fr}}, y_{\max}/H_{\mathrm{fr}}]$, where W_{fr} and H_{fr} are frame width and height respectively. Then, the visual object feature and the location feature are projected into the d-dimensional space with two learned linear layers, and are summed up as the initial visual representation $\boldsymbol{v}_t^{(0)} = \{\boldsymbol{v}_{t,n}^{(0)}\}_{n=1}^N$, where $t \in \{1, ..., T\}$ is the frame index and $\boldsymbol{v}_{t,n}^{(0)} \in \mathbb{R}^d$ is the representation of the n-th object in the t-th frame. In the meanwhile, we extract classes and attributes of the detected objects, *e.g.*, "white cat", using the same Faster R-CNN. These semantic knowledge is embedded by a pre-trained word embedding model (fastText [4] in our case), and are then linearly projected into a d-dimensional space to produce the initial semantic representations $\boldsymbol{s}_t^{(0)} = \{\boldsymbol{s}_{t,n}^{(0)}\}_{n=1}^N$.

For the question, we first embed each word into a 300-dimensional vector, which is initialized with pre-trained GloVe vectors [37]. To obtain contextual representation, we further pass these embedding vectors through a Bi-LSTM [13]. The final question embedding is denoted as $\boldsymbol{q}^{(0)} \in \mathbb{R}^d$.

Reasoning at Object Level. After obtaining the input embeddings $\boldsymbol{v}_t^{(0)}$, $\boldsymbol{s}_t^{(0)}$, and $\boldsymbol{q}^{(0)}$, we use them to initialize the visual graph, the semantic graph, and the read controller, respectively. Then, both graph memory mechanisms perform iterative relational reasoning over visual object representations and semantic object representations, respectively.

$$\boldsymbol{v}_t^{(K_v)} = \mathrm{VGM}(\boldsymbol{q}^{(0)}, \boldsymbol{v}_t^{(0)}) \quad (20)$$

$$\widetilde{\boldsymbol{v}}_t = \mathrm{SGM}\,(\boldsymbol{q}^{(0)}, \boldsymbol{s}_t^{(0)}, \boldsymbol{v}_t^{(K_v)}) \quad (21)$$

where $\widetilde{\boldsymbol{v}}_t = \{\widetilde{\boldsymbol{v}}_{t,n}\}_{n=1}^N$ is updated representation for the t-th frame, encoding query-relevant object-level visual and semantic relations.

Node Aggregation. We aggregate graph nodes for each frame, and build new graph by using the aggregated representation of each frame as nodes, thus enabling subsequent frame-level relational reasoning. To be specific, for visual graph, nodes are aggregated via question-guided attention [1]: $\bar{\boldsymbol{v}}_t = \mathrm{Attn}(\widetilde{\boldsymbol{v}}_t, \boldsymbol{q}^{(0)})$, where $\bar{\boldsymbol{v}}_t \in \mathbb{R}^d$ is the aggregated visual representation of the t-th frame. We inject the temporal location information into $\bar{\boldsymbol{v}}_t$ following [46]. For semantic graph, we aggregate nodes using average pooling: $\bar{\boldsymbol{s}}_t = \frac{1}{N}\sum_{n=1}^N \boldsymbol{s}_{t,n}^{(0)}$, where $\bar{\boldsymbol{s}}_t \in \mathbb{R}^d$ is the aggregated semantic representation of the t-th frame.

Reasoning at Frame Level. We construct two new graphs and initialize their node states with the frame-level representations: $\boldsymbol{v}^{(0)} = \{\bar{\boldsymbol{v}}_t\}_{t=1}^T$ and $\boldsymbol{s}^{(0)} = \{\bar{\boldsymbol{s}}_t\}_{t=1}^T$. The read controller is initialized with the question embedding $\boldsymbol{q}^{(0)}$. Afterwards, both graph memory mechanisms perform iterative relational reasoning over visual frame representations and semantic frame representation, respectively.

$$\boldsymbol{v}^{(K_v)} = \mathrm{VGM}(\boldsymbol{q}^{(0)}, \boldsymbol{v}^{(0)}) \quad (22)$$

$$\widetilde{\boldsymbol{v}} = \mathrm{SGM}\,(\boldsymbol{q}^{(0)}, \boldsymbol{s}^{(0)}, \boldsymbol{v}^{(K_v)}) \quad (23)$$

where $\widetilde{\boldsymbol{v}} \in \mathbb{R}^{T \times d}$. Through such iterative relational reasoning at frame level, the model learns to gradually attend to the key frames and capture the appropriate relations between frames (as shown in Figure 6). Moreover, by incorporating high-level semantic knowledge, the yielded video representation is more discriminative.

Multi-Scale Node Aggregation. Answering different questions usually needs temporal information of different durations. To this end, we design a multi-scale node aggregation method to aggregate $\widetilde{\boldsymbol{v}}$ into a holistic representation. The component consists of H parallel heads. Each head includes a linear layer that reduces the input dimension by $1/H$, a temporal average pooling with different kernel size that captures multi-scale temporal information, and a question-guided attention [1] that aggregates nodes with attention weights. We concatenate the output of each head as final output, denoted as $\hat{\boldsymbol{v}} \in \mathbb{R}^d$. Note that all nodes are arranged in time order before applying the temporal pooling.

Answer Decoder. Following previous work [26, 9], we adopt different answer decoders depending on the question type. (1) For ***open-ended*** questions, we treat them as classification tasks. The video representation $\hat{\boldsymbol{v}}$ is fused with the question embedding $\boldsymbol{q}^{(0)}$ to compute scores on all candidate answers: $\boldsymbol{p} = \mathrm{MLP}([\hat{\boldsymbol{v}}\,; \boldsymbol{q}^{(0)}])$. The cross-entropy is used as the loss function. (2) For ***counting*** questions, the model is required to predict a number ranging from 0 to 10. We leverage a linear layer followed by a rounding function upon the fused representation to predict the number: $num = \mathrm{round}(\boldsymbol{W}_p \boldsymbol{f}_{vq})$, where $\boldsymbol{f}_{vq} = \mathrm{ReLU}(\boldsymbol{W}_f[\hat{\boldsymbol{v}}\,; \boldsymbol{q}^{(0)}])$. The loss for this question type is Mean Squared Error (MSE). (3) For ***multi-choice*** questions, each answer choice is concatenated with the question to form a query. We feed each pair of query and video into the network. As a result, we obtain a set of query representations $\{\boldsymbol{q}_a^{(0)}\}_{a=1}^{|\mathcal{A}|}$ and video representations $\{\hat{\boldsymbol{v}}_a\}_{a=1}^{|\mathcal{A}|}$, where $|\mathcal{A}|$ is the number of answer choices. The score of each answer choice is computed as $p_a = \mathrm{MLP}([\hat{\boldsymbol{v}}_a\,; \boldsymbol{q}_a^{(0)}])$. A softmax function is applied to process the scores. We use the cross-entropy loss function.

4. Experiments

4.1. Experimental Setup

Datasets. Four datasets are used in our experiments. TGIF-QA [16] is currently the most prominent benchmark dataset for the VideoQA task, which contains 165K QA pairs collected from 72K animated GIFs. There are four task types: (1) *Count*: an open-ended counting task that retrieves the number of repetition of an action; (2) *Action*: a multiple-choice task that aims to recognize the action repeated for a given number of times; (3) *Transition*: a multiple-choice task asking about the transition of two states; and (4) *Frame QA*: an open-ended task similar to ImageQA, which can be answered from a single video frame. MSVD-QA [51] is a small dataset of 51K QA pairs which are automatically generated from the descriptions of MSVD videos [5]. All questions are open-ended and divided into five types – *what, who, how, when* and *where*. MSRVTT-QA [51] is a larger dataset containing 243K QA pairs. Youtube2Text-QA [53] includes open-ended and multiple-choice questions, which are divided into three types (*i.e*. *what, who* and *other*). More statistics of the four datasets are in the Supp. material.

We adopt accuracy as the evaluation metric for all tasks except the *count* task on TGIF-QA dataset. For *count*, we use Mean Square Error (MSE) to measure the performance.

Implementation Details. We evenly sample 10 frames to represent the video and select 6 detected objects with the highest scores per frame. The dimensionality of the joint embedding space d is 512. The number of visual and semantic reasoning steps, K_v and K_s, are set to 2 and 2, respectively. We use 4 heads in the multi-scale node aggregation. The kernel sizes of temporal pooling in each head are respectively set to 1, 2, 3 and 4, and the stride size is 1. Models are trained using the Adam optimizer [21] with an initial learning rate of 1e-4 and a batch size of 64. The entire training takes approximately 12 hours on one Nvidia Tesla V100 GPU. The results are reported at the epoch giving the best validation performance.

4.2. State of the Art Comparison

We compare our HAIR with state-of-the-art methods on four challenging datasets. Table 1 shows the performance comparison on TGIF-QA dataset. Only with ResNet visual feature, HAIR outperforms previous methods (even those that use more visual features) on *Action* (+2.8%), *Trans.* (+0.9%) and *FrameQA* (+3.9%) tasks. The improvement is particularly noticeable on *FrameQA* task, where object-level relational reasoning is required. It is noted that L-GCN [14] uses GCN [22] to reason about object-object relations while PSAC [28] applies self-attention to

Table 1. Comparison with state-of-the-art methods on TGIF-QA dataset. For *Count* task, the lower the better. Visual features are: R(ResNet), C(C3D), F(FlowCNN), RX(ResNext).

Method	Action	Trans.	FrameQA	Count
ST-VQA (R+C) [16]	60.8	67.1	49.3	4.40
Co-Mem (R+F) [12]	68.2	74.3	51.5	4.10
PSAC (R) [28]	70.4	76.9	55.7	4.27
HME (R+C) [9]	73.9	77.8	53.8	4.02
L-GCN (R) [14]	74.3	81.1	56.3	3.95
HCRN (R+RX) [26]	75.0	81.4	55.9	**3.82**
HAIR (R)	**77.8**	**82.3**	**60.2**	3.88

Table 2. Comparison with state-of-the-art methods: Co-Mem [12], AMU [51], HME [9], QueST [17] and HCRN [26] on MSVD-QA and MSRVTT-QA datasets.

Dataset	Co-Mem	AMU	HME	QueST	HCRN	HAIR
MSVD-QA	31.7	32.0	33.7	36.1	36.1	**37.5**
MSRVTT-QA	32.0	32.5	33.0	34.6	35.6	**36.9**

Table 3. Comparison with state-of-the-art methods on Youtube2Text-QA dataset.

Task	Method	What	Who	Other	All
Multiple-Choice	HME [9]	83.1	77.8	**86.6**	80.8
	L-GCN [14]	86.0	81.5	80.6	83.9
	HAIR	**87.8**	**82.4**	81.4	**85.3**
Open-Ended	HME [9]	29.2	28.7	**77.3**	30.1
	L-GCN [14]	24.5	53.2	70.4	38.0
	HAIR	**32.4**	**54.7**	72.2	**43.0**

model frame-frame relations, but they fail to integrate object-level and frame-level relational reasoning. Table 2 shows the performance comparison on MSVD-QA and MSRVTT-QA datasets. It can be seen from the table that our model HAIR significantly outperforms existing methods on both datasets, establishing new state-of-the-art results of 37.5% and 36.9% on MSVD-QA and MSRVTT-QA, respectively. Table 3 shows the performance comparison on Youtube2Text-QA dataset. Our HAIR achieves remarkable improvements (+1.4% for multiple-choice task and +5% for open-ended task) over L-GCN [14] in overall accuracy. These facts prove the effectiveness and generality of our approach on different task types and datasets.

4.3. Ablation Studies

Hierarchical Relational Reasoning. We first conduct experiments to investigate the effect of *hierarchical* relational reasoning. As shown in the first block of Table 4, ablating any hierarchical level (*i.e*. object level or frame level) leads to severe performance degradation on all task types. We observe "object level only" performs better than "frame level only". This indicates object-level relational reasoning plays a more important role in VideoQA. However, few of previous work explore such relational reasoning. We also exper-

Table 4. Ablation studies of our model on TGIF-QA dataset. For *Count* task, the lower the better.

Setting	Action	Trans.	FrameQA	Count
Object level only	73.5	79.2	57.1	4.08
Frame level only	71.2	78.0	55.9	4.15
Two-stream	75.3	80.7	57.8	4.01
w/o visual	70.6	77.2	57.4	4.13
w/o semantic	74.6	80.6	56.0	4.06
w/o visual+semantic	68.4	76.1	54.7	4.28
GCN	73.4	79.0	56.2	4.07
GCN (fusion)	75.1	81.4	57.7	3.95
Self-attention	73.9	80.5	56.7	4.06
Memory network	72.4	78.1	54.2	4.16
Full	77.8	82.3	60.2	3.88

Table 5. Comparison of inference time, model size and memory footprint.

Method	Inference Time	Model Size	Memory Footprint
HME [9]	3.2s	43.3M	3055MB
HCRN [26]	0.6s	42.8M	2111MB
Ours	0.5s	24.2M	2541MB

iment with two-stream framework. One stream is the object level and another is the frame level. The worse results from the two-stream framework suggest the superiority of our hierarchical framework.

Visual-Semantic Relational Reasoning. We then analyze the impact of visual-semantic relational reasoning in the second block of Table 4. Generally, "w/o visual" produces larger performance drop compared with "w/o semantic". However, on FrameQA task, "w/o visual" (*i.e.* using only semantic knowledge for reasoning) achieves surprisingly better performance than "w/o semantic". The reason is that the semantic knowledge can provide explicit answer cues for some FrameQA questions. For example, the class "cat" can be directly utilized to answer the question "What jumps up at itself in the mirror?". When disabling both graph memory mechanisms, we observe the performance further degenerates, showing the complementarity between visual and semantic relational reasoning.

Graph Memory. We propose a novel relational reasoning mechanism – graph memory, which elegantly combines the ideas of GNN and memory network. We also investigate other relational reasoning modules in the third block of Table 4. "GCN" denotes graph convolutional network [22], and "GCN (fusion)" denotes using the fusion of multimodal features as node representation. We can see that GCN variants underperform our graph memory, due to the disability of dynamic query guidance and dynamic feature selection. Self-attention [46] is applied to model the dependencies of frames in [20, 28]. We stack a few self-attention layers to keep the same reasoning steps as ours, and replace our graph memory in HAIR framework. As shown in the table, self-

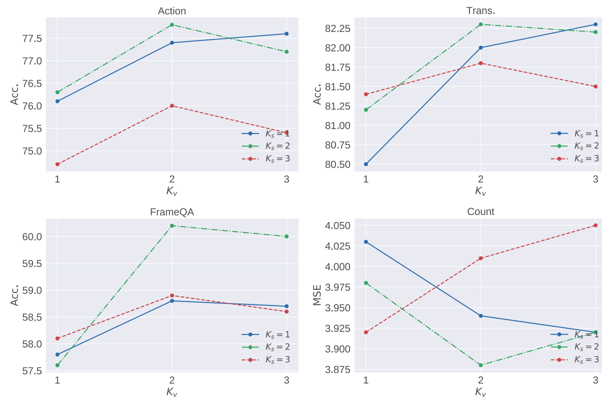

Figure 5. Comparison of different visual relational reasoning steps (K_v) and semantic relational reasoning steps (K_s) on TGIF-QA.

attention deliver worse results than ours. Memory network [44] has been introduced to solve QA problem [50, 12]. It is capable of performing iterative reasoning in a dynamic way, but can not explicitly model relations, thus leading to performance drop. These results demonstrate the superiority of our graph memory mechanism.

# of Reasoning Steps. It is also of interest to explore how many steps of *visual* and *semantic* relational reasoning are sufficient for VideoQA task. We test our model with different reasoning steps. The results are exhibited in Figure 5. We have the following observations: (1) When $K_v = 2$ and $K_s = 2$, the best performance is obtained on all four tasks. (2) When $K_s = 1$ (*i.e.* blue line), increasing K_v from 1 to 3 can constantly boost the performance. It seems that more *visual* reasoning steps can make up for the lack of *semantic* reasoning to some extent. This may be because more iterations can distill some *semantic* knowledge from *visual* information, which is similar to that deeper CNN layers usually carry high-level *semantic* information compared to shallow layers. (3) Increasing K_s from 2 to 3 produces larger performance drop compared to increasing K_v from 2 to 3. This phenomenon can be explained that *semantic* knowledge is already explicit and high-level representation, and thus using more semantic relational reasoning steps would smooth (or blur) the semantics.

Model Efficiency Comparison. Table 5 shows the inference time, model size (#param), and memory footprint of different methods. We run our method and the released codes of HME[1] [9] and HCRN[2] [26] on one Nvidia Tesla V100 GPU with batch size 32. It can be observed that our HAIR is more efficient than HME and HCRN (recent SOTAs), with nearly half params and faster inference time.

Performance on Other Video+Language Task. To further validate the effectiveness and generality of our hierarchical visual-semantic relational reasoning, we conduct experiment on other video+language task, *e.g.*, language-

[1] https://github.com/fanchenyou/HME-VideoQA
[2] https://github.com/thaolmk54/hcrn-videoqa

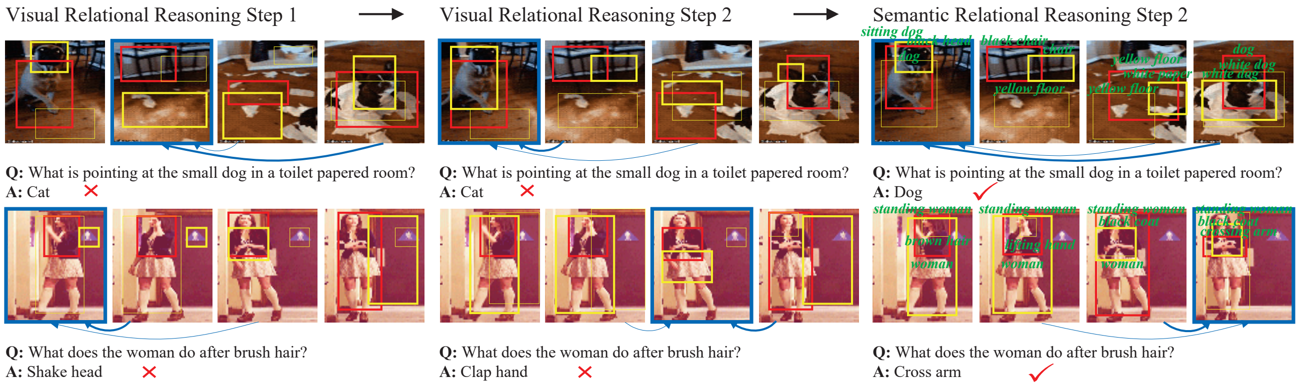

Figure 6. Visualization of relational reasoning process of our HAIR. In each frame, we show the most attended object (red box), and two most related objects (yellow box) with different line width indicating the relations between them and the red box. The most attended frame is highlighted with blue box. The blue arrows with different line width denotes the relation weights from other frames to the most attended frame. When reaching the semantic relational reasoning step, we show the semantic knowledge (*i.e.* classes and attributes) on the top (or bottom due to space limitation) of the boxes. See more examples in the Supp.

Table 6. Performance comparison on the language-based temporal grounding task.

Method	IoU@0.3	IoU@0.5	IoU@0.7
CBP [47]	54.3	35.8	17.8
ABLR [54]	55.7	36.8	-
DEBUG [32]	55.9	39.7	-
HVTG [6]	57.6	40.2	18.3
HAIR	57.3	40.5	18.2

based temporal grounding. We adopt ActivityNet Captions dataset [24] for performance comparison and "R@1, IoU@x" as the evaluation metrics. We use a similar prediction module and loss function to [6]. As shown in Table 6, our HAIR achieves promising results.

4.4. Qualitative Analysis

To provide more insights about our HAIR, we show the visualization of relational reasoning process in Figure 6. Initially, the model fails to focus on the relevant object and frame (*e.g.*, the object "sign", "door" and the 1st frame are focused on in the second example), and fails to model accurate object-object relations and frame-frame relations (*e.g.*, the relation between the "woman" and the "sign" and the relation between the 1st and the 2nd frame are modeled). As the iteration (step) goes on, the model gradually learns to attend to the most relevant object and frame (*e.g.*, the object "crossing arm" and the 4th frame), and model accurate object-object relations and frame-frame relations (*e.g.*, the relations between the "woman" and the "crossing arm", between the 3rd and the 4th frame). In particular, without explicit semantic knowledge, the model mistakenly recognizes the object and the action, although more visual relational reasoning steps have been conducted. After leveraging the semantic knowledge for relational reasoning, the model finally gives the correct answer. These visualizations

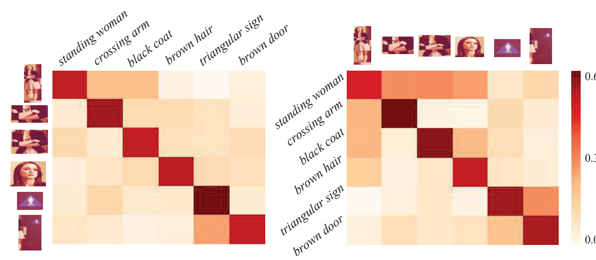

Figure 7. Visualization of attention weights of visual-to-semantic (*left*) and semantic-to-visual (*right*) node mapping.

help explain our approach. Some failure examples are provided in the Supp. material. We take the 4th frame in the second example and visualize the attention of visual-to-semantic and semantic-to-visual node mapping mechanisms at object level. As shown in Figure 7, the proposed node mapping mechanisms are able to collect the related information from another representation to enhance the current representation and benefit the relational reasoning.

5. Conclusion

In this paper, we propose a hierarchical visual-semantic relational reasoning (HAIR) framework for VideoQA, which integrates object-level and frame-level relational reasoning in a hierarchical way and explores high-level semantic knowledge to facilitate relational reasoning. The basic unit is graph memory, which can achieve relational reasoning under dynamic guidance of query and also enable dynamic information selection. Extensive experiments demonstrate the effectiveness and generality of our method.

Acknowledgment: This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106400), National Natural Science Foundation of China (61922086, 61872366), and Beijing Natural Science Foundation (4192059, JQ20022).

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018. 2, 5

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, pages 2425–2433, 2015. 2

[3] Yanrui Bin, Zhao-Min Chen, Xiu-Shen Wei, Xinya Chen, Changxin Gao, and Nong Sang. Structure-aware human pose estimation with graph convolutional networks. *Pattern Recognition*, 106:107410, 2020. 2

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 5

[5] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, pages 190–200, 2011. 6

[6] Shaoxiang Chen and Yu-Gang Jiang. Hierarchical visual-textual graph for temporal activity localization via language. In *ECCV*, 2020. 8

[7] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, pages 7239–7248, 2018. 2

[8] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *CVPR*, pages 5177–5186, 2019. 2

[9] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *CVPR*, pages 1999–2007, 2019. 1, 3, 5, 6, 7

[10] Karl Friston. Hierarchical models in the brain. *PLoS Comput Biol*, 4(11):e1000211, 2008. 2

[11] Difei Gao, Ke Li, Ruiping Wang, Shiguang Shan, and Xilin Chen. Multi-modal graph neural network for joint reasoning on vision and scene text. In *CVPR*, pages 12746–12756, 2020. 2

[12] Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. Motion-appearance co-memory networks for video question answering. In *CVPR*, pages 6576–6585, 2018. 1, 3, 6, 7

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5

[14] Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. Location-aware graph convolutional networks for video question answering. In *AAAI*, pages 11021–11028, 2020. 1, 2, 6

[15] Hao Huang, Luowei Zhou, Wei Zhang, Jason J Corso, and Chenliang Xu. Dynamic graph modules for modeling object-object interactions in activity recognition. In *BMVC*, 2019. 3

[16] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, pages 2758–2766, 2017. 6

[17] Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *AAAI*, pages 11101–11108, 2020. 6

[18] Pin Jiang and Yahong Han. Reasoning with heterogeneous graph alignment for video question answering. In *AAAI*, pages 11109–11116, 2020. 2

[19] Weike Jin, Zhou Zhao, Mao Gu, Jun Yu, Jun Xiao, and Yueting Zhuang. Multi-interaction network with object relation for video question answering. In *ACM MM*, pages 1193–1201, 2019. 2

[20] Kyung-Min Kim, Seong-Ho Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. Multimodal dual attention memory for video story question answering. In *ECCV*, pages 673–688, 2018. 1, 2, 7

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1, 6, 7

[23] Daniel C Krawczyk, M Michelle McClelland, and Colin M Donovan. A hierarchy for relational reasoning in the prefrontal cortex. *Cortex*, 47(5):588–597, 2011. 2

[24] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, pages 706–715, 2017. 8

[25] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 5

[26] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *CVPR*, pages 9972–9981, 2020. 1, 2, 5, 6, 7

[27] Xiangpeng Li, Lianli Gao, Xuanhan Wang, Wu Liu, Xing Xu, Heng Tao Shen, and Jingkuan Song. Learnable aggregating net with diversity learning for video question answering. In *ACM MM*, pages 1166–1174, 2019. 2

[28] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *AAAI*, pages 8658–8665, 2019. 1, 2, 6, 7

[29] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *NeurIPS*, pages 1853–1863, 2018. 2

[30] Fei Liu, Jing Liu, Zhiwei Fang, Richang Hong, and Hanqing Lu. Visual question answering with dense inter-and intra-modality interactions. *IEEE Transactions on Multimedia*, 2020. 2

[31] Fei Liu, Jing Liu, Richang Hong, and Hanqing Lu. Erasing-based attention learning for visual question answering. In *ACM MM*, pages 1175–1183, 2019. 2

[32] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. Debug: A dense bottom-up grounding approach for natural

language video localization. In *EMNLP*, pages 5147–5156, 2019. 8

[33] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding. In *CVPR*, pages 6790–6800, 2018. 3

[34] Effrosyni Mavroudi, Benjamín Béjar Haro, and René Vidal. Representation learning on visual-symbolic graphs for video understanding. In *ECCV*, pages 71–90, 2020. 3

[35] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *EMNLP*, 2016. 1

[36] Will Norcliffe-Brown, Efstathios Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. In *NeurIPS*, pages 8344–8353, 2018. 2

[37] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 5

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 5

[39] Morteza Sarafyazd and Mehrdad Jazayeri. Hierarchical reasoning by neural circuits in the frontal cortex. *Science*, 364(6441), 2019. 2

[40] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008. 2

[41] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, pages 7912–7921, 2019. 2

[42] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 12026–12035, 2019. 2

[43] Gursimran Singh, Leonid Sigal, and James J Little. Spatio-temporal relational reasoning for video question answering. In *BMVC*, 2019. 1

[44] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *NeurIPS*, pages 2440–2448, 2015. 1, 2, 7

[45] Zichang Tan, Yang Yang, Jun Wan, Guodong Guo, and Stan Z Li. Relation-aware pedestrian attribute recognition with graph convolutional networks. In *AAAI*, pages 12055–12062, 2020. 2

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 1, 2, 5, 7

[47] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *AAAI*, pages 12168–12175, 2020. 8

[48] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, pages 9236–9245, 2019. 2

[49] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. 2

[50] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, pages 2397–2406, 2016. 3, 7

[51] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*, pages 1645–1653, 2017. 1, 6

[52] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *CVPR*, pages 21–29, 2016. 2

[53] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. Video question answering via attribute-augmented attention network learning. In *ACM SIGIR*, pages 829–832, 2017. 6

[54] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*, pages 9159–9166, 2019. 8