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

Fei Liu, Jing Liu, Weining Wang, Hanqing Lu
1National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
2School 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 Hierarchical Visual-Semantic Relational Reasoning (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.

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 \textit{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 \textit{et al.} [14] and Jin \textit{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 \textbf{H}ierarchical \textbf{V}isual-\textbf{S}emantic \textbf{R}elational \textbf{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 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) \textit{visual graph memory}, which exploits visual information of video for relational reasoning, and gradually learns query-related relation-aware \textit{visual} representation; b) \textit{semantic graph memory}, where we represent object classes and attributes as nodes and build edges to encode commonsense \textit{semantic} relationships. It explicitly leverages \textit{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, \textit{e.g.}, language-based temporal grounding.

2. Related Work

\textbf{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 requires 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 \textit{et al.} [28] and Kim \textit{et al.} [20] used a self-attention [46] based technique to model global dependencies among frames of a video. Jiang \textit{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 \textit{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 \textit{hierarchical} relational reasoning.

\textbf{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], \textit{etc.}) and video domain (including action recognition [42, 41], video object segmentation [48], \textit{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
 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 \( q^{(0)} \in \mathbb{R}^d \) denote the initial state of read controller and \( v^{(0)} \in \mathbb{R}^d \) 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 \( r^{(k)} \) from all nodes:

\[
\begin{align*}
    q^{(k)} &= \mathbf{V}_v^a \tanh(\mathbf{W}_v^a q^{(k-1)} + \mathbf{U}_v^a v^{(k-1)}) \quad (1) \\
    q^{(k)} &= \exp(a^{(k)}_i) / \sum_j \exp(a^{(k)}_j) \quad (2) \\
    r^{(k)} &= \sum_i a^{(k)}_i v_i^{(k-1)} \quad (3)
\end{align*}
\]

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

\[
\begin{align*}
    \tilde{q}^{(k)} &= \mathbf{W}_v^r q^{(k-1)} + \mathbf{U}_v^r r^{(k)} \quad (4) \\
    g^{(k)} &= \sigma(\mathbf{W}_v^g q^{(k-1)} + \mathbf{U}_v^g r^{(k)}) \quad (5) \\
    q^{(k)} &= g^{(k)} \odot \tilde{q}^{(k)} + (1 - g^{(k)}) \odot q^{(k-1)} \quad (6)
\end{align*}
\]

where \( \mathbf{W}_v^a, \mathbf{U}_v^a \) and \( \mathbf{V}_v^a \) are learnable weights. \( \sigma \) and \( \odot \) represent the sigmoid function and Hadamard product, respectively. The update gate \( g^{(k)} \) controls how much previous state to be preserved.
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 $v_i^{(k-1)}$, current content from the read controller $q_i^{(k)}$, and the representations from other nodes $\{v_j^{(k-1)}\}_{j \neq i}$. Concretely, we first aggregate the information from neighboring nodes to capture the context:

$$c_i^{(k)} = \text{MLP}([v_i^{(k-1)}; v_j^{(k-1)}])$$

(7)

$$e_{i,j}^{(k)} = \exp(e_{i,j}^{(k)}) / \sum_{j \neq i} \exp(e_{i,j}^{(k)})$$

(8)

$$c_i^{(k)} = \sum_{j \neq i} e_{i,j}^{(k)} v_j^{(k-1)}$$

(9)

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

$$v_i^{(k)} = W_u v_i^{(k)} + U_u c_i^{(k)} + V_u v_{j}^{(k-1)}$$

(10)

$$g_i^{(k)} = \sigma(W^g u v_i^{(k)} + U^g u v_{j}^{(k-1)} + V^g u v_{j}^{(k-1)})$$

(11)

$$v_i^{(k)} = g_i^{(k)} \circ v_i^{(k-1)} + (1 - g_i^{(k)}) \circ v_i^{(k-1)}$$

(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:

$$v_i^{(K_v)} = VGM(q_i^{(0)}, v_i^{(0)})$$

(13)

where VGM represents visual graph memory, $q_i^{(0)}$ is the initial state of the read controller, $v_i^{(0)} = \{v_i^{(0)}\}_{i=1}^{|V|}$ is the initial visual representations of graph nodes (where $|V|$ is the number of nodes), and $v_i^{(K_v)}$ is the updated representation 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 $q_i^{(0)} \in \mathbb{R}^d$, the initial representations of the semantic graph $s_i^{(0)} \in \mathbb{R}^{|S| \times d}$, and the updated representations of the visual graph $v_i^{(K_v)} \in \mathbb{R}^{|V| \times d}$, 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:

$$\phi_{vs} = \exp(W_{vs} s_i^{(K_v)}) / \sum_{j'=1}^{|S|} \exp(W_{vs} s_j^{(K_v)})$$

(14)

$$f_i^{vs} = \sum_{j=1}^{|V|} \phi_{vs} s_j v_i^{(K_v)}$$

(15)

where $\phi_{vs}$ represents the confidence of mapping the feature from the $j$-th visual node to the $i$-th semantic node, $W_{vs} \in \mathbb{R}^{|S| \times |V|}$ is a trainable weight matrix for calculating voting weights, and $W_p \in \mathbb{R}^{|d| \times d}$ is a projection weight matrix. The representation of each semantic node is updated as:

$$s_i^{(0)} = [s_i^{(0)}, f_i^{vs}]$$

Then, we perform iterative relational reasoning over the enhanced semantic representations $s_i^{(0)}$. 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 $s_i^{(K_s)} = \{s_i^{(K_s)}\}_{i=1}^{|S|}$, which is then mapped back into visual space to enrich the visual representation with global semantic knowledge via a semantic-to-visual node mapping:

$$\phi_{sv}^{(s)} v_i^{(K_s)} = W_{sv} s_i^{(K_s)} v_i^{(K_s)}$$

(16)

$$\phi_{sv}^{(s)} v_i^{(K_s)} = \exp(\phi_{sv}^{(s)} v_i^{(K_s)}) / \sum_{j=1}^{|S|} \exp(\phi_{sv}^{(s)} v_i^{(K_s)})$$

(17)

$$f_i^{sv} = \sum_{j=1}^{|S|} \phi_{sv}^{(s)} s_j v_i^{(K_s)}$$

(18)

where $W_{sv} \in \mathbb{R}^{|d| \times |S|}$ and $W_p \in \mathbb{R}^{|d| \times d}$ 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:

$$\tilde{v}_i = v_i^{(K_v)} + f_i^{sv}.$$
The entire process can be concisely written as:

$$\bar{v} = \text{SGM}(q^{(0)}, s^{(0)}, v^{(K_v)})$$  \hspace{1cm} (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 $[x_{\text{min}}/W_{\text{fr}}, y_{\text{min}}/H_{\text{fr}}, x_{\text{max}}/W_{\text{fr}}, y_{\text{max}}/H_{\text{fr}}]$, where $W_{\text{fr}}$ and $H_{\text{fr}}$ 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 $v_1^{(0)} = \{v_{t,n}^{(0)}\}_{n=1}^{N}$, where $t \in \{1, ..., T\}$ is the frame index and $v_{t,n}^{(0)} \in \mathbb{R}^d$ 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 $s_t^{(0)} = \{s_{t,n}^{(0)}\}_{n=1}^{N}$.

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 $q^{(0)} \in \mathbb{R}^d$.

Reasoning at Object Level. After obtaining the input embeddings $v_1^{(0)}$, $s_1^{(0)}$, and $q^{(0)}$, 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.

$$v_t^{(K_v)} = \text{VGM}(q^{(0)}, v_t^{(0)})$$  \hspace{1cm} (20)

$$\bar{v}_t = \text{SGM}(q^{(0)}, s_t^{(0)}, v_t^{(K_v)})$$  \hspace{1cm} (21)

where $\bar{v}_t = \{\bar{v}_{t,n}\}_{n=1}^{N}$ 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]: $\bar{v}_t = \text{Attn}(\bar{v}_t, q^{(0)})$, where $\bar{v}_t \in \mathbb{R}^d$ is the aggregated visual representation of the $t$-th frame. We inject the temporal location information into $\bar{v}_t$ following [46]. For semantic graph, we aggregate nodes using average pooling: $\tilde{s}_t = \frac{1}{N} \sum_{n=1}^{N} s_{t,n}^{(0)}$, where $\tilde{s}_t \in \mathbb{R}^d$ 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: $\bar{v}^{(0)} = \{\bar{v}_t\}_{t=1}^{T}$ and $s^{(0)} = \{\tilde{s}_t\}_{t=1}^{T}$. The read controller is initialized with the question embedding $q^{(0)}$.

Afterwards, both graph memory mechanisms perform iterative relational reasoning over visual frame representations and semantic frame representation, respectively.

$$v^{(K_v)} = \text{VGM}(q^{(0)}, \bar{v}^{(0)})$$  \hspace{1cm} (22)

$$\bar{v} = \text{SGM}(q^{(0)}, s^{(0)}, v^{(K_v)})$$  \hspace{1cm} (23)

where $\bar{v} \in \mathbb{R}^{T \times d}$. 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 $\bar{v}$ 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 $\hat{v} \in \mathbb{R}^d$. 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 $\hat{v}$ is fused with the question embedding $q^{(0)}$ to compute scores on all candidate answers: $p = \text{MLP}([\hat{v} : q^{(0)}])$. 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: $\text{num} = \text{round}(W_p f_{eq}, where f_{eq} = \text{ReLU}(W_j [\hat{v} : q^{(0)}]))$. 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 $\{q^{(0)}_a\}_{a=1}^{|A|}$ and video representations $\{\hat{d}_a\}_{a=1}^{|A|}$, where $|A|$ is the number of answer choices. The score of each answer choice is computed
as $p_a = \text{MLP}([\bar{v}_a; q_a^{(0)}])$. 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 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>
<thead>
<tr>
<th>Method</th>
<th>Action</th>
<th>Trans.</th>
<th>FrameQA</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-VQA (R+C) [16]</td>
<td>60.8</td>
<td>67.1</td>
<td>49.3</td>
<td>4.40</td>
</tr>
<tr>
<td>Co-Mem (R+F) [12]</td>
<td>68.2</td>
<td>74.3</td>
<td>51.5</td>
<td>4.10</td>
</tr>
<tr>
<td>PSAC (R) [28]</td>
<td>70.4</td>
<td>76.9</td>
<td>55.7</td>
<td>4.27</td>
</tr>
<tr>
<td>HME (R+C) [9]</td>
<td>73.9</td>
<td>77.8</td>
<td>53.8</td>
<td>4.02</td>
</tr>
<tr>
<td>L-GCN (R) [14]</td>
<td>74.3</td>
<td>81.1</td>
<td>56.3</td>
<td>3.95</td>
</tr>
<tr>
<td>HCRN (R+RX) [26]</td>
<td>75.0</td>
<td>81.4</td>
<td>55.9</td>
<td>3.82</td>
</tr>
<tr>
<td>HAIR (R)</td>
<td>77.8</td>
<td>82.3</td>
<td>60.2</td>
<td>3.88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Co-Mem</th>
<th>AMU</th>
<th>QueST</th>
<th>HCRN</th>
<th>HAIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSVD-QA</td>
<td>31.7</td>
<td>32.0</td>
<td>33.7</td>
<td>36.1</td>
<td>37.5</td>
</tr>
<tr>
<td>MSRVTT-QA</td>
<td>32.0</td>
<td>32.5</td>
<td>33.0</td>
<td>34.6</td>
<td>35.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>Method</th>
<th>What</th>
<th>Who</th>
<th>Other</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple-Choice</td>
<td>HME [9]</td>
<td>83.1</td>
<td>77.8</td>
<td>86.6</td>
<td>80.8</td>
</tr>
<tr>
<td></td>
<td>L-GCN [14]</td>
<td>86.0</td>
<td>81.5</td>
<td>80.6</td>
<td>83.9</td>
</tr>
<tr>
<td></td>
<td>HAIR</td>
<td>87.8</td>
<td>82.4</td>
<td>81.4</td>
<td>85.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>Method</th>
<th>What</th>
<th>Who</th>
<th>Other</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open-Ended</td>
<td>HME [9]</td>
<td>29.2</td>
<td>28.7</td>
<td>77.3</td>
<td>30.1</td>
</tr>
<tr>
<td></td>
<td>L-GCN [14]</td>
<td>24.5</td>
<td>53.2</td>
<td>70.4</td>
<td>38.0</td>
</tr>
<tr>
<td></td>
<td>HAIR</td>
<td>32.4</td>
<td>54.7</td>
<td>72.2</td>
<td>43.0</td>
</tr>
</tbody>
</table>
Table 4. Ablation studies of our model on TGIF-QA dataset. For Count task, the lower the better.

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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Inference Time</th>
<th>Model Size</th>
<th>Memory Footprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>HME [9]</td>
<td>3.2s</td>
<td>43.3M</td>
<td>3055MB</td>
</tr>
<tr>
<td>HCRN [26]</td>
<td>0.6s</td>
<td>42.8M</td>
<td>2111MB</td>
</tr>
<tr>
<td>Ours</td>
<td>0.5s</td>
<td>24.2M</td>
<td>2541MB</td>
</tr>
</tbody>
</table>

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

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

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-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 $K_v = 1$ (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 HME1 [9] and HCRN2 [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-
Table 6. Performance comparison on the language-based temporal grounding task.

<table>
<thead>
<tr>
<th>Method</th>
<th>IoU@0.3</th>
<th>IoU@0.5</th>
<th>IoU@0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBP [47]</td>
<td>54.3</td>
<td>35.8</td>
<td>17.8</td>
</tr>
<tr>
<td>ABLR [54]</td>
<td>55.7</td>
<td>36.8</td>
<td>-</td>
</tr>
<tr>
<td>DEBUG [32]</td>
<td>55.9</td>
<td>39.7</td>
<td>-</td>
</tr>
<tr>
<td>HVTG [6]</td>
<td>57.6</td>
<td>40.2</td>
<td>18.3</td>
</tr>
<tr>
<td>HAIR</td>
<td>57.3</td>
<td>40.5</td>
<td>18.2</td>
</tr>
</tbody>
</table>

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 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


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


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


