Motion-Appearance Co-Memory Networks for Video Question Answering

Jiyang Gao* Runzhou Ge* Kan Chen Ram Nevatia
University of Southern California
{jiyangga, rge, kanchen, nevatia}@usc.edu

Figure 1. Answering questions in videos involves both motion and appearance analysis, and usually requires multiple cycles of reasoning, especially for transitive questions, *e.g.* " What does the woman do after eat?", we need to first localize when the woman eat, which requires motion evidence for eating and appearance evidence for the woman; and then focus on what the woman does (take a drink).

Abstract

Video Question Answering (QA) is an important task in understanding video temporal structure. We observe that there are three unique attributes of video QA compared with image QA: (1) it deals with long sequences of images containing richer information not only in quantity but also in variety; (2) motion and appearance information are usually correlated with each other and able to provide useful attention cues to the other; (3) different questions require different number of frames to infer the answer. Based these observations, we propose a motion-appearance co-memory network for video QA. Our networks are built on concepts from Dynamic Memory Network (DMN) and introduces new mechanisms for video QA. Specifically, there are three salient aspects: (1) a co-memory attention mechanism that utilizes cues from both motion and appearance to generate attention; (2) a temporal conv-deconv network to generate multi-level contextual facts; (3) a dynamic fact ensemble method to construct temporal representation dynamically for different questions. We evaluate our method on TGIF-QA dataset, and the results outperform state-of-the-art significantly on all four tasks of TGIF-QA.

∗ indicates equal contributions.

1. Introduction

Understanding video temporal structure is an important topic in computer vision. To achieve this goal, various tasks have been proposed, such as temporal action localization [28, 9], action anticipation [10] and video prediction [31]. Besides these tasks, video Question Answering (QA) [15, 29] is another challenging task, which not only requires the understanding of video temporal structure, but also joint reasoning of videos and texts. In this paper, we tackle the problem of video QA.

Image and text question answering have achieved much progress recently. The success comes in part from the application of attention mechanisms [36, 20] and memory mechanisms [19] in deep neural networks. Attention mechanisms tell the neural network "where to look", while the memory mechanism refines answers in multiple reasoning cycles. Video QA is different from image QA [22, 20] in two aspects: (1) the questions are more about temporal reasoning of the videos, *e.g.* motion transition and action counting, than spatial attributes, such as colors, spatial locations, which require effective temporal representation modeling; (2) the input source is a sequence of images, rather than a single image, which contains richer information not only in quantity but also in variety (appearance, motion, transition) to "remember", and it makes the reasoning process more complicated.

Dynamic Memory Networks (DMN) [19, 32] were originally proposed for text and image question answering. It

contained a memory module to encode the input sources multiple cycles and an attention mechanism allowing the reading process to focus on different contents in each cycle. Although DMN contains an input module and a memory module which are able to read and remember a long sequence information, which is applicable for videos, directly applying such a method to video QA task would not give satisfying results. Because it lacks motion analysis, especially joint analysis between motion and appearance in videos, and temporal modeling. To strengthen the memory mechanism, Na *et al*. [24] proposed a read-write memory network that jointly encode the movie appearance and caption content, however it lacks motion analysis and dynamic memory update. Xu *et al*. [34] exploited the appearance and motion via gradually refined attention, where the motion and appearance features are fused together.

We observe two unique attributes of answering questions in videos. The first is that the motion and appearance information are usually correlated with each other in the reasoning process. For example, in answering the question "what does the woman do after eat?" as shown in Figure 1, we need to first localize "the woman eat" action, which requires motion evidence for eating and appearance evidence for the woman; after that, we need to ignore the man's interval, and then focus on what the woman does (drinking water). Appearance and motion information are both involved in the reasoning process and provide attention cues to each other. The second attribute is that different types of questions may require representations from different amounts of frames, for example, "what is the color of the bulldog?" needs only a single frame to produce the answer, while "How many times does the cat lick" needs the understanding of the whole video.

Based on these observations, we propose a motion-appearance co-memory network for video QA. Our model is built on concepts of DMN/DMN+ [19, 32], so we share the same terms with DMN [19], such as facts, memory and attention. Specifically, a video is converted to a sequence of motion and appearance features by the two-stream models [33]. The motion and appearance features are then fed into a temporal convolutional and deconvolutional neural network to build multi-level *contextual facts*, which have the same temporal resolution but represent different contextual information. These contextual facts are used as input *facts* to the memory networks. The co-memory networks hold two separate memory states, one for motion and one for appearance. To jointly model and interact with the motion and appearance information, we design a co-memory attention mechanism that takes motion cues for appearance attention generation, and appearance cues for motion attention generation. Based on these attentions, we design *dynamic fact ensemble* method to produce temporal facts dynamically at each cycle of fact encoding. We evaluate our model on TGIF-QA dataset [15], and outperform state-of-the-art performance significantly on all four tasks in TGIF-QA.

The novelty of our method is three-fold compared with DMN/DMN+ [19, 32]:

(1) We design a co-memory attention mechanism to jointly model motion and appearance information.

(2) We use temporal conv-deconv networks to build multi-level contextual facts for video QA.

(3) We introduce a method called dynamic fact ensemble to dynamically produce temporal facts in each cycle of fact encoding.

In the following, we first introduce related work, and then outline the DMN/DMN+ framework. In Section 4, we present our motion-appearance co-memory network in detail, and in Section 5, we show the evaluation of our method on TGIF-QA.

2. Related Work

Image question answering. Image question answering aims to measure the capability of reasoning about linguistic and image inputs jointly. Many methods have been proposed [36, 4, 13, 20, 35, 6, 3, 1, 32, 22, 27, 16, 12, 39, 37, 40, 41]. Among all these models, attention mechanism [36, 4, 20, 41] provides guidance to deep models on "where to look" and memory mechanism [19, 32] allows the model to have multiple reasoning iterations and refine the answer gradually. Question-guided attention mechanism [4] uses semantic representation of a question as query to search for the regions in an image that are related to the answer. Yang *et al*. [36] presented a Stacked Attention Network (SAN) that queries an image multiple times to infer the answer progressively. Lu *et al*. [20] argued that modeling "what words to listen to" is equally important to model "where to look", and proposed a co-attention model that jointly reasons about image-guided and question-guided attention. Instead of directly inferring answers from the abstract visual features, Yu *et al*. [37] developed a semantic attention mechanism to select high-level question-related concepts. Dynamic memory network (DMN), which was first introduced by Kumar *et al*. [19] to solve text based question answering, adopted episodic memories and attention mechanisms which allow multiple cycles of reasoning. Xiong *et al*. [32] improved the memory and input module of DMN so that it can be applied to image QA.

Video question answering. Video QA is a relatively new task compared with image QA. Yu *et al*. [38] adopted a semantic attention mechanism, which combines the detected concepts in videos with text encoding/decoding to generate answers. Comparing with images, temporal domain is unique to videos. A temporal attention mechanism is leveraged to selectively attend to one or more periods of a video in [15, 23, 34]. Besides temporal attention mechanism, Jang *et al*. [15] and Xu *et al*. [34] also utilized motion

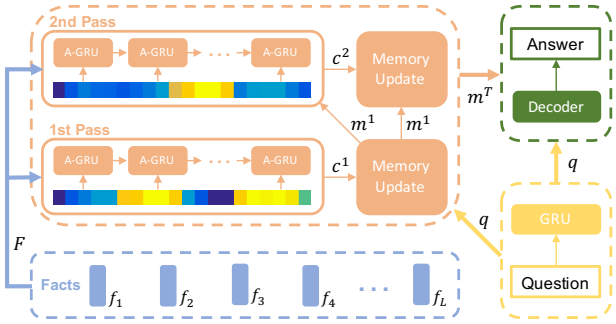


Figure 2. General Dynamic Memory Network (DMN) [19] architecture. The memory update process for the t-th cycle is : (1) the facts F are encoded by an attention-based GRU in episodic memory module, where the attention is generated by last memory m^{t-1}; (2) the final hidden state of the GRU is called contextual vector c^t, which is used to update the memory m^t together with question embedding q. The question answer is generated from the final memory state m^T.

information along with appearance information in videos. Recently Na *et al*. [24] and Kim *et al*. [17] both introduced the memory mechanism to their models for video QA. However, their models [24, 17] both lack motion analysis and dynamic memory update mechanism.

Video temporal analysis. To answer the video-based questions correctly, temporal analysis of videos is necessary. Shou *et al*. [28] presented a multi-stage Segment-CNN model to generate action proposals and localize actions in videos. Temporal Unit Regression Network (TURN) [8] and Cascaded Boundary Regression (CBR) [9] exploit the temporal boundary regression mechanism for proposal generation and and action detection. Recently Gao *et al*. [7] and Hendricks *et al*. [2] proposed to localize activities by language queries, their methods involve of joint modeling of the videos and language queries, which also related to video QA.

3. General Dynamic Memory Networks

As our work is closely related to DMN [19, 32], we begin with introducing the general framework of DMN. It contains four distinct modules: an input module, a question module, an episodic memory module and an answer module, as shown in Figure 2.

Fact module. The fact module converts the input data (*e.g*. text, image, video) into a set of vectors called *facts*, which is denoted as $F = [f_1, f_2, ..., f_L]$, where L is the total number of facts. For text-based QA, [19] used a Gated Recurrent Unit (GRU) to encode all text information; for image-based QA, [32] adopted a bi-directional GRU to encode the local region visual features to globally-aware facts.

Question module. The question module converts the question into an embedding q. Specifically, [19, 32] used a GRU to encode the question sentence and use the final hidden state of the GRU as the question embedding.

Episodic memory module. Episodic memory is designed to retrieve the relevant information from the facts. To extract information related to the questions from the facts more effectively, especially when transitive reasoning is required in questions, the episodic memory module iterates over the input facts for multiple cycles, and updates the memory after each cycle. There are two important mechanisms in the episodic memory module: an attention mechanism and a memory update mechanism.

Suppose that the updated memory after t-th cycle is m^t, the facts set $F = [f_1, f_2, ..., f_L]$, the question embedding is q, then the attention gate g_i^t is given by

$$g_i^t = F_a(f_i, m_{t-1}, q) \tag{1}$$

where F_a is an attention function which takes the fact vector f_i at step i, memory m^{t-1} at cycle $t-1$ and the question q as inputs, and outputs a scalar value g_i^t, which represents the attention value for the fact f_i in cycle t.

To effectively use the ordering and positional information in videos, an attention based GRU is designed. Instead of using the original update gate in the GRU, the attention gate g_i^t is used, the update equation for the modified GRU is

$$h_i = g_i^t \circ \tilde{h_i} + (1 - g_i^t) \circ h_{i-1} \tag{2}$$

The final hidden state of the attention based GRU is used as the contextual feature c^t for updating the episodic memory m^t. Together with the question embedding q and the memory for cycle $t-1$, the t-th cycle memory is updated by

$$m^t = F_m(m^{t-1}, c^t, q) \tag{3}$$

where F_m is a memory update function. The final memory m^T is passed to the answer module to generate the final answers, where T is the number of memory update cycle.

Answer module. The answer module takes both q and m^T to generate the models predicted answer. Different answer decoders may be applied for different tasks, *e.g*. a softmax output layer for single word answer.

4. Motion-Appearance Co-Memory Networks

In this section, we present our motion-appearance co-memory networks, including multi-level contextual facts, co-memory module and answer module. The question module remains the same as the one in traditional DMN.

4.1. Multi-level Contextual Facts

The videos are cut into small units [8] (a sequence of frames). For each video unit, we use two-stream CNN models [33] to extract unit-level motion and appearance features. More feature pre-processing details are given in Section 5. The sequence of unit-level appearance features and motion features is represented as $\{a_i\}$ and $\{b_i\}$ respectively.

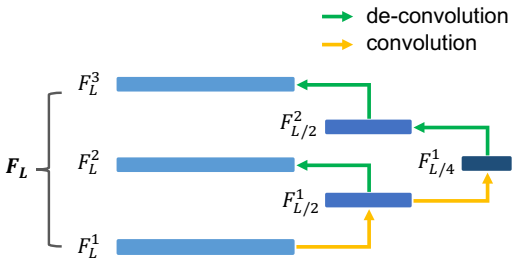


Figure 3. The input temporal representations are processed by temporal conv-deconv layers to build multi-layer contextual facts, which have the same temporal resolution but different contextual information.

To build multiple levels of temporal representations where each level represent different contextual information, we use the temporal convolutional layers to model the temporal contextual information and de-convolutional layers to recover temporal resolution, as shown in Figure 3. Specifically, the lowest level feature sequence is built directly from the unit features, $A_L^1 = \{a_i\}$, $B_L^1 = \{b_i\}$. The convolutional layers compute a feature hierarchy consisting of temporal feature sequences at several scales with a scaling step of 2, F_L^1, $F_{L/2}^2$, $F_{L/4}^3$, ..., as shown in Figure 3. Note that F could be A (for appearance features) or B (for motion features). The de-convolutional pathway hypothesizes higher resolution features F_L^2, F_L^3 by upsampling temporally coarser, but semantically stronger, feature sequences. Thus, F_L^1, F_L^2 and F_L^3 have the same resolution but different temporal contextual coverage. Note that we only show 3 levels in Figure 3, more levels could be modeled by adding more convolutional and de-convolutional layers. $\mathbf{F}_L = \{F_L^1, F_L^2, ..., F_L^N\}$ is termed as *contextual facts*.

4.2. Motion-appearance Co-Memory Module

In this part, we introduce the co-memory attention mechanism and the dynamic fact ensemble method.

Co-memory attention. The questions in video QA usually involve both appearance and motion. Appearance usually provides useful cues for motion attention, *i.e.* guides the focus on motion content, and vice versa. To allow interaction between appearance and motion, we design a co-memory attention mechanism. Specifically, two separate memory modules are used to hold motion memory m_b^t and appearance memory m_a^t, where t is the number of cycle for memory update. As indicated before, when the networks read motion facts to update motion memory, appearance memory provides useful cues to generate attentions; motion memory is also helpful for updating appearance attention. Therefore, m_b^{t-1} and m_a^{t-1} are both used to generate attentions for motion and appearance fact encoding in the t-th cycle. As we build multiple levels of facts, we generate an attention score for each fact vector at each level. The motion attention gate for fact b_j^i is $gb_{i,j}^t$ and the appearance attention for fact a_j^i is $ga_{i,j}^t$, where t means the number of cycle, i is the level of fact representation and j is the step of the facts.

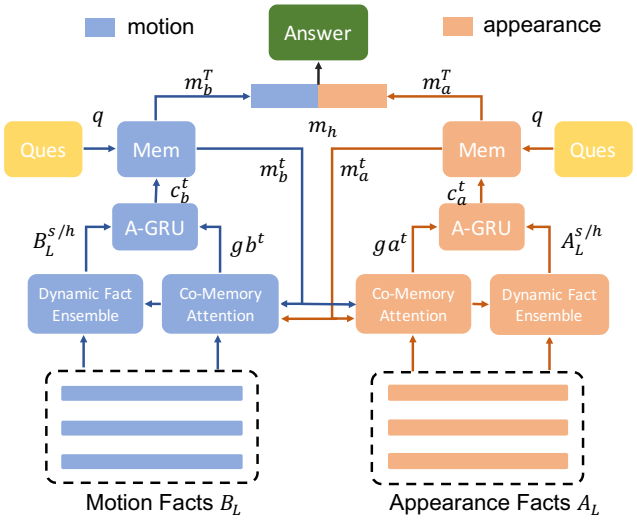


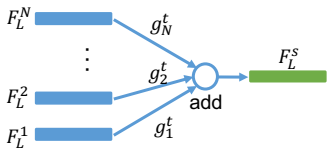
Figure 4. Co-memory attention module extracts useful cues from both appearance and motion memories to generate attention ga^t/gb^t for motion and appearance separately. Dynamic fact ensemble takes the multi-layer contextual facts $\mathbf{A}_L/\mathbf{B}_L$ and the attention scores ga^t/gb^t to construct proper facts $A_L^{s/h}/B_L^{s/h}$, which are encoded by an attention-based GRU. The final hidden state c_b^t/c_a^t of the GRU is used to update the memory m_b^t/m_a^t. The final output memory m_h is the concatenation of the motion and appearance memory, and it is use to generate answers.

$$
\begin{aligned}
za_{i,j}^t &= \tanh\left(\mathbf{W}_a^2\left(a_i^j + \mathbf{W}_a^1[m_a^{t-1}, q]\right)\right) \\
ga_{i,j}^t &= \mathbf{W}_a^4\left(za_{i,j}^t + \mathbf{W}_a^3[m_b^{t-1}, q]\right)
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
zb_{i,j}^t &= \tanh\left(\mathbf{W}_b^2\left(b_i^j + \mathbf{W}_b^1[m_b^{t-1}, q]\right)\right) \\
gb_{i,j}^t &= \mathbf{W}_b^4\left(zb_{i,j}^t + \mathbf{W}_b^3[m_a^{t-1}, q]\right)
\end{aligned} \tag{5}
$$

where $\mathbf{W}_a^1$, $\mathbf{W}_a^2$, $\mathbf{W}_a^3$, $\mathbf{W}_a^4$, $\mathbf{W}_b^1$, $\mathbf{W}_b^2$, $\mathbf{W}_b^3$ and $\mathbf{W}_b^4$ are weight parameters. $ga_{i,j}^t$ and $gb_{i,j}^t$ are attentions used in dynamic fact ensemble and memory update.

Dynamic fact ensemble. As shown in Section 4.1, we build a multi-layer contextual facts set $\mathbf{F}_L = \{F_L^1, F_L^2, ..., F_L^N\}$ for motion and appearance separately, which have the same temporal resolution, but represent different contextual information. There are two reasons that the facts should be selected dynamically: (1) Different types of questions may require different level of representations, *e.g.* the "bulldog color" and the "cat lick" questions given in Section 1; (2) During the multiple cycles of the fact reading, each cycle may focus on different level of information. We designed an attention-based fact ensemble methods shown



(a) soft fusion

Figure 5. Multi-layer contextual facts are dynamically constructed via a soft attention fusion process, which computes a weighted average facts according to the attention.

in Figure 5. For simplicity, we use $g_{i,j}^t$ to represent the attention gate, which is actually $ga_{i,j}^t$ for appearance and $gb_{i,j}^t$ for motion. We calculate Softmax over $g_{i,j}^t$ along level axis (*i.e.* i) to get attention scores $s_{i,j}^t$.

The ensemble facts can be represented as

$$F_t^s : \{f_j^t = \sum_{i=0}^{N} s_{i,j}^t f_j^i\}_{j=1}^{L} \tag{6}$$

where $f_{i,j}$ is the fact vector of level i and step j in the contextual facts $\mathbf{F}_L$. The attention scores used in the later fact encoding process are given by

$$s_j^t = \text{softmax}(\frac{1}{N}\sum_{i=0}^{N} g_{i,j}^t), j = 1, 2, ..., L \tag{7}$$

where the Softmax is computed along j axis.

Memory update. The fact encoding processes are conducted separately for motion and appearance, which adopts an attention based GRU [32] to generate contextual vectors c_a^t and c_b^t for appearance and motion in the t-th cycle. Motion memory m_b^t and appearance memory m_a^t are updated separately as follows.

$$m_a^t = \text{FC}([m_a^{t-1}, q, c_a^t]) \tag{8}$$

$$m_b^t = \text{FC}([m_b^{t-1}, q, c_b^t]) \tag{9}$$

where FC means fully-connected layer, ReLU is used as the non-linear activation. The final output memory m_h is the concatenation of m_a^T and m_b^T, where T is the number of cycles.

4.3. Answer Module

Following [15], we model the four tasks in TGIF-QA [15] into three different types: multiple-choice, open-ended numbers and open-ended words.

For multiple-choice, we use a linear regression function that takes the memory state m_h and outputs a real-valued score for each answer candidate.

$$s = \mathbf{W}_m^T m_h \tag{10}$$

where $\mathbf{W}_m$ are weight parameters. The model is optimized by hinge loss between the scores for correct answers s_p and the scores for incorrect answers s_n, $\max(0, 1 + s_n - s_p)$. This decoder is used to solve repeating action and state transition tasks.

For open-ended numbers, we also use a linear regression function which takes the memory state m_h and outputs an integer-valued answer.

$$s = [\mathbf{W}_n^T m_h + b] \tag{11}$$

where $[.]$ means rounding. We adopt ℓ_2 loss between the groundtruth value and the predicted value to train the model, which is used to solve the repetition count task.

For open-ended words, we treat this as a classification problem. A linear function that takes the final memory state m_h followed by a softmax layer is adopted to generate answers.

$$\mathbf{o} = \text{softmax}\left(\mathbf{W}_w^\top \mathbf{m}_h + \mathbf{b}\right) \tag{12}$$

where $\mathbf{W}_w$ are weight parameters and b is bias. Cross-entropy loss is used to train the model and this type of decoder is used in Frame QA task.

For each task, we train a separate model by the answer decoder and loss mentioned above. The model of each task is trained and evaluated individually.

5. Evaluation

In this section, we describe the dataset and evaluation settings, and discuss the experiment results.

5.1. Dataset

We evaluate the proposed model on TGIF-QA dataset [15], which is a large-scale dataset introduced by Jang *et al*. for Video QA. The dataset consists of 165k QA pairs collocted from 71k animated Tumblr GIFs. There are four types of tasks: repetition count, repetition action, state transition and frame QA. First three tasks are unique to videos and require temporal reasoning to answer them.

Tasks. *Repetition count* is an open-ended task to count the number of repetition of an action (*e.g.* "How many times does the cat lick?"). There are 11 possible answers (*i.e.* from 0 to 10+) in total. *Repetition action* is a 5-option multiple choice task, which is asking about the name of the action that happened specific times (*e.g.* "what does the duck do 3 times?"). *State transition* is also a 5-option multiple choice task which can be answered by understanding the transition of two states in a video (*e.g.* "What does the woman do after drink water?"). Besides, TGIF-QA also provides a traditional *frame QA* task (*i.e.* image QA). The image QA questions of previous datasets [3, 26, 21] can be answered by getting effective information from a single given image; but for frame QA in TGIF-QA dataset, the model needs to find the most relevant frame among all frames in the video

Table 1. Number of samples of different tasks in TGIF-QA dataset.

# QA pairs	Action	Trans	Count	Frame
Training	20,475	52,704	26,843	39,392
Testing	2,274	6,232	3,554	13,691
Total	22,749	58,936	30,397	53,083

to answer the question correctly. Frame QA is defined as an open-ended task. The number of QA pairs of TGIF-QA for the four tasks are shown in table 1.

Metric. For the task of repetition count, the Mean Square Error (MSE) between the predicted count value and the groundtruth count value is used for evaluation. For repetition action, state transition and frame QA, classification accuracy (ACC) is used as the evaluation metric.

5.2. Implementation Details

Appearance and motion features. Since the frames per second (FPS) of the GIFs in TGIF-QA [15] vary, we extract frames from all GIFs with the FPS that is specified by the corresponding GIF file. The long videos are cut into small units, each unit contains 6 frames.

To extract unit-level video features, we use ResNet-152 [11] to process the central frame of a unit, and the outputs of "pool5" layer ($\in \mathbb{R}^{2,048}$) of ResNet-152 is used as our appearance features. To utilize motion information, we extract optical flow inside a video unit, and use the flow CNN from two-stream model [33] to get unit-level flow features. Specifically, the two-direction dense optical flows [5] which are calculated between two adjacent frames in a six-consecutive-frame unit are fed into the pre-trained flow CNN model, which is a BN-Inception network[14]. Then we take the feature map of the "global_pool" layer ($\in \mathbb{R}^{1,024}$) as the raw optical flow features. Finally, we down-sample the feature dimension by average pooling and get a 2048-dimension vector as our two-direction optical flow feature. In this process, we padded the first or last frame if we didn't have enough frames centered at each step. We set the temporal resolution of video features to be 34, long feature sequences are cut and short one are padded.

Contextual facts. The output channel number of each layer in the conv-deconv networks is 1024, temporal conv filter size is 3 with stride 1, deconv layer with stride 2, max pool filter size is 2 with stride 2. We build $N = 3$ layers of contextual facts.

Co-memory module. The size of memory state m_a and m_b is set to be 1024. The hidden state size of the GRU for fact encoding is 512. $za^t_{i,j}$ and $zb^t_{i,j}$ in equation (4) and (5) are 512-dimensional.

Question and answer embedding. For each word in the question, we use a pre-trained word embedding model [25] to convert it to a 300-dimension vector. All words in the question are processed by a two-layer GRU, whose hidden state size is 512. The final hidden state is used as question embedding. For action transition and repeating action, the candidate answers are a sequence of words, thus we use the same method as the one for encoding questions to encode the answer.

Training details. We set the batch size to 64. Adam optimizer [18] is used to optimize the model, the learning rate is set to 0.001. For each task, we train the model for 50 epochs.

5.3. System Baselines

Besides co-memory networks, there are two direct methods to make use of motion and appearance information: fact concatenation and memory concatenation, which are used as system baselines.

Fact concatenation. This baseline method simply concatenate the input motion facts and appearance facts, $\{b_i\}$ and $\{a_i\}$ along the feature dimension. The concatenated vector $\{h_i\}$ which is $d_b + d_a$ dimensional is used as input facts for multi-level contextual fact module. Only one memory module is used.

Memory concatenation. In this baseline method, instead of concatenating the input facts, we use two separate memory modules: one for appearance, the other for motion, and concatenate the final motion memory states m_b^T and the final appearance memory states m_a^T to m_f^t together, which are used to decode answers. Co-memory attention mechanism is not used in this baseline.

5.4. Experiments on TGIF-QA

We first evaluate the co-memory attention module by comparing it with the two baseline method "fact concatenation" and "memory concatenation". Second, we evaluate the multi-level contextual facts and the dynamic fact ensemble. Finally, we compare our method with the previous state-of-the-art methods.

Co-memory attention. In this experiment, we set the layer of contextual facts to be 1, and dynamic fact ensemble is not used. The number of memory updates $T = 2$. We compare co-memory attention mechanism with "fact concatenation" (fact-concat) and "memory concatenation" (mem-concat) to see the effectiveness of co-memory attention , the results are shown in Table 2. We can see that co-

Table 2. Evaluation of co-memory attention mechanism on TGIF-QA. "Action" is repetition action (ACC %), "Trans" is state transition (ACC %), "Count" is repetition count (MSE) and "Frame" is frame QA (ACC %).

Method	Action	Trans	Count	Frame
Fact-concat	65.0	71.2	4.34	49.9
Mem-concat	64.5	70.7	4.39	50.2
Co-memory	**66.8**	**73.2**	**4.21**	**51.0**

memory attention outperforms fact-concat and mem-concat in all four tasks, which shows the effectiveness of the co-memory attention mechanism. We believe the reason is that co-memory attention exploits the knowledge that motion and appearance provide useful cues to each other in attention generation.

Contextual facts and dynamic fact ensemble. Dynamic fact ensemble collaborates with multi-level contextual facts to construct proper temporal fact representation, so we test them together. We build 3 layers of contextual facts and do experiments to test dynamic fact ensemble module. We use "fact concatenation" as the top memory network. The results are shown in Table 3: "w/o ensemble" means that we don't build the multi-level contextual facts, but just use a single temporal conv layer (filter size is 1) to convert appearance and motion features into 1024-dimension vectors, which are used as input facts.

Table 3. Evaluation of dynamic fact ensemble on TGIF-QA. "Action" is repetition action (ACC %), "Trans" is state transition (ACC %), "Count" is repetition count (MSE) and "Frame" is frame QA (ACC %).

Method	Action	Trans	Count	Frame
w/o ensemble	65.0	71.2	4.34	49.9
w/ ensemble	66.3	72.5	4.30	50.4

It can be seen that the ensemble provides better results. We believe the reason is that the attention-based fact fusion optimizes the ensemble process by using weighted average of the contextual facts, and avoids just using only one of them, which may make the facts sub-optimal.

How many cycles of memory update are sufficient? We test the co-memory attention model with different memory update times $T = 1, 2, 3$ to see how many cycles of memory update are sufficient for video QA task. The dynamic fact ensemble is not used in this experiment. The results are shown in Table 4.

Table 4. Comparison on cycles of memory update on TGIF-QA. "Action" is repetition action (ACC %), "Trans" is state transition (ACC %), "Count" is repetition count (MSE) and "Frame" is frame QA (ACC %).

Method	Action	Trans	Count	Frame
$T = 1$	65.1	69.9	4.35	50.5
$T = 2$	66.8	73.2	4.21	51.0
$T = 3$	66.5	73.1	4.24	51.1

We can see that two cycles ($T = 2$) of memory update gives the best performance on the task of "Action", "Trans" and "Count". For "Frame", $T = 2$ and $T = 3$ have similar results. Comparing the results of $T = 2$ and $T = 1$ in "Trans", we can see that $T = 2$ improves the performance by 3.3%, we believe the reason is that multiple cycles of fact reading and memory update allow the model to focus on different parts of the video in each cycle. The performance begins to saturate at $T = 3$.

Table 5. Comparison with the state-of-the-art method on TGIF-QA dataset. "Action" is repetition action (ACC %), "Trans" is state transition (ACC %), "Count" is repetition count (MSE) and "Frame" is frame QA (ACC %).

Model	Action	Trans	Frame	Count [1]
VIS+LSTM(aggr) [26]	46.8	56.9	34.6	-
VIS+LSTM(avg) [26]	48.8	34.8	35.0	-
VQA-MCB(aggr) [6]	58.9	24.3	25.7	-
VQA-MCB(avg) [6]	29.1	33.0	15.5	-
Yu *et al*. [38]	56.1	64.0	39.6	-
ST(R+C) [15]	60.1	65.7	48.2	4.38
ST-SP(R+C) [15]	57.3	63.7	45.5	4.28
ST-SP-TP(R+C) [15]	57.0	59.6	47.8	4.56
ST-TP(R+C) [15]	60.8	67.1	49.3	4.40
ST-TP(R+F)	62.9	69.4	49.5	4.32
Co-memory (w/o DFE)	66.8	73.2	51.0	4.21
Co-memory (full)	**68.2**	**74.3**	**51.5**	**4.10**

Comparison with state-of-the-art method. There are two version of TGIF-QA, we report the performance of the second version, which is released by the authors of [15] on Arxiv. The first version is originally reported in the CVPR version of [15]. State-of-the-art method [15] on TGIF-QA adopted a dual-LSTM based approach with both spatial and temporal attention. Originally, their model is trained on C3D [30] temporal feature and ResNet-152 [11] frame feature. However, our method adopts Flow CNN model (Inception) for motion and ResNet-152 for appearance. Thus, for fair comparison, we train their model (https://goo.gl/SVKTP9) with our features on all four tasks in TGIF-QA. The results are shown in Table 5. In Table 5, "SP" means spatial attention, "TP" means temporal attention, "(R+C)" means ResNet-152 features and C3D features, "(R+F)" means ResNet-152 features and Flow CNN features (our feature). We also list methods "VIS-LSTM" [26] and "VQA-MCB" [6], which are provided in [15].

There are two co-memory variants shown in Table 5: "co-memory (w/o DFE)" uses co-memory attention with $T = 2$ memory update, but not dynamic fact ensemble; "co-memory (full)" uses co-memory attention with $T = 2$ memory update and dynamic fact ensemble (soft fusion) on 3-layer contextual facts. We can see that our method outperforms the state-of-the-art method significantly on all four tasks. Some visualization examples are shown in Figure 6.

[1] We found an evaluation bug in [15] (https://goo.gl/SVKTP9) on count task. The official updated performance from the author is listed here.

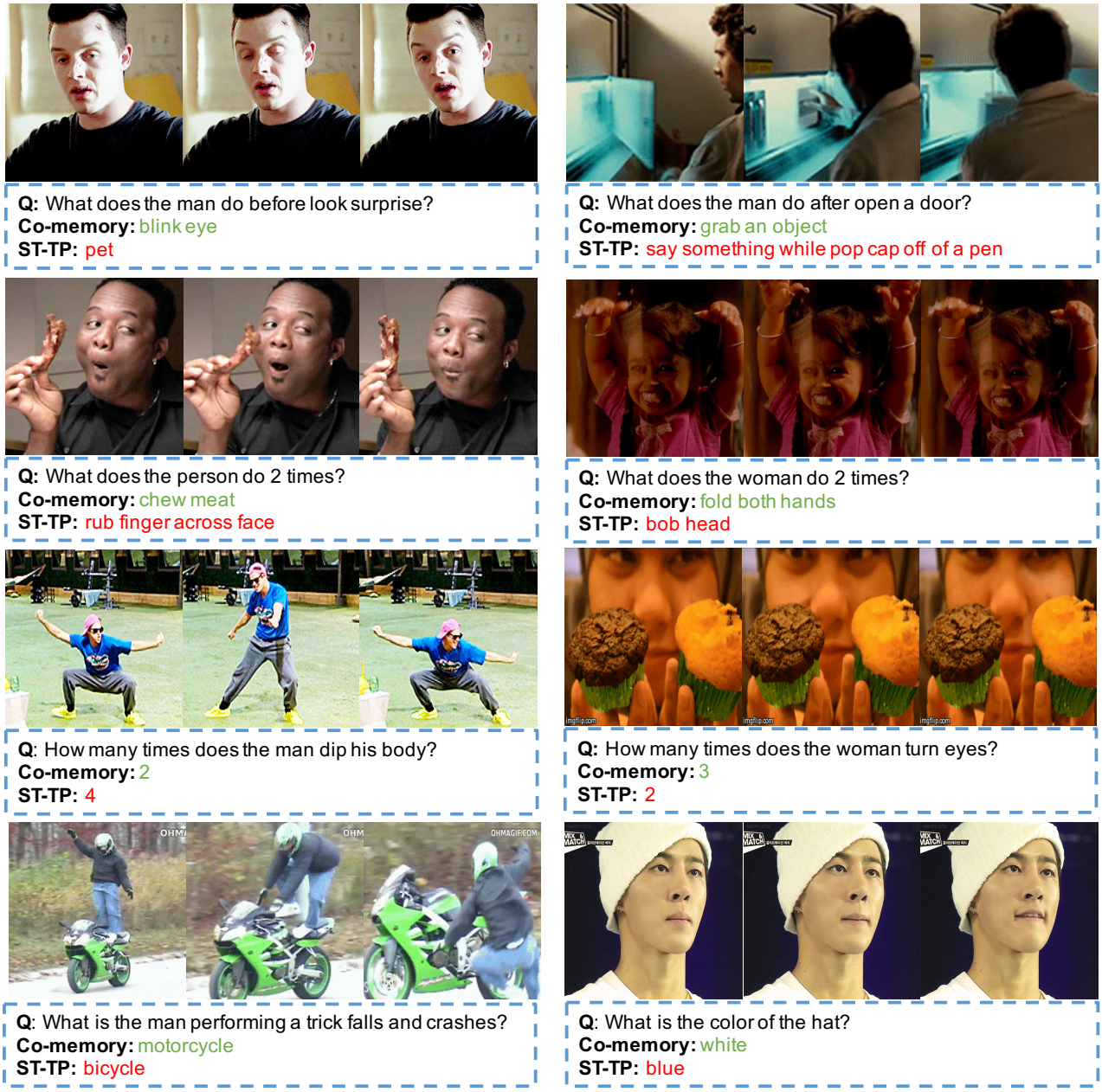


Figure 6. Examples on state transition, repetition action, repetition count and frame QA are shown in 1st, 2nd, 3rd and 4th row. ST-TP is the temporal attention model from [15]. Green is for correct prediction and red is for wrong prediction.

6. Conclusion

Comparing with image QA, video QA deals with long sequences of images, which contains richer information in both quantity and variety. In addition, motion and appearance information are both important for video analysis, and usually correlated with each other and able to provide useful attention cues to the other. Motivated by these observations, we propose a motion-appearance co-memory network for video QA. Specifically, we design a co-memory attention mechanism that utilizes cues from both motion and appearance to generate attention, a temporal conv-deconv network to generate multi-level contextual facts, and a dynamic fact ensemble method to construct temporal representation dynamically for different questions. We evaluate our method on TGIF-QA dataset, and outperforms state-of-the-art performance significantly.

Acknowledgements. This research was supported, in part, by the Office of Naval Research under grant N00014-18-1-2050.

References

[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *CVPR*, 2016. 2

[2] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 3

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 2, 5

[4] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*, 2015. 2

[5] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370, 2003. 6

[6] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 2, 7

[7] J. Gao, C. Sun, Z. Yang, and R. Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017. 3

[8] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017. 3

[9] J. Gao, Z. Yang, and R. Nevatia. Cascaded boundary regression for temporal action detection. In *BMVC*, 2017. 1, 3

[10] J. Gao, Z. Yang, and R. Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In *BMVC*, 2017. 1

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[12] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, 2017. 2

[13] I. Ilievski, S. Yan, and J. Feng. A focused dynamic attention model for visual question answering. *arXiv preprint arXiv:1604.01485*, 2016. 2

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 6

[15] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 1, 2, 5, 6, 7, 8

[16] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual qa. In *NIPS*, 2016. 2

[17] K.-M. Kim, M.-O. Heo, S.-H. Choi, and B.-T. Zhang. Deepstory: video story qa by deep embedded memory networks. In *IJCAI*, 2017. 3

[18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*, 2016. 1, 2, 3

[20] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 1, 2

[21] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014. 5

[22] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 1, 2

[23] J. Mun, P. Hongsuck Seo, I. Jung, and B. Han. Marioqa: Answering questions by watching gameplay videos. In *ICCV*, 2017. 2

[24] S. Na, S. Lee, J. Kim, and G. Kim. A read-write memory network for movie story understanding. In *ICCV*, 2017. 2, 3

[25] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 6

[26] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015. 5, 7

[27] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016. 2

[28] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 1, 3

[29] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016. 1

[30] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 7

[31] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017. 1

[32] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016. 1, 2, 3, 5

[33] Y. Xiong, L. Wang, Z. Wang, B. Zhang, H. Song, W. Li, D. Lin, Y. Qiao, L. Van Gool, and X. Tang. Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016. 2, 3, 6

[34] D. Xu, Z. Zhao, J. Xiao, F. Wu, H. Zhang, X. He, and Y. Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017. 2

[35] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 2

[36] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 2

[37] D. Yu, J. Fu, T. Mei, and Y. Rui. Multi-level attention networks for visual question answering. In *CVPR*, 2017. 2

[38] Y. Yu, H. Ko, J. Choi, and G. Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 2017. 2, 7

[39] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma. Structured attentions for visual question answering. In *ICCV*, 2017. 2

[40] Y. Zhu, D. Gordon, E. Kolve, D. Fox, L. Fei-Fei, A. Gupta, R. Mottaghi, and A. Farhadi. Visual semantic planning using deep successor representations. In *ICCV*, 2017. 2

[41] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 2