Heterogeneous Memory Enhanced Multimodal Attention Model for Video Question Answering

Chenyou Fan[1,*], Xiaofan Zhang[1], Shu Zhang[1], Wensheng Wang[1], Chi Zhang[1], Heng Huang[1,2,*]

[1]JD.COM, [2] JD Digits

*chenyou.fan@jd.com, *heng.huang@jd.com

Abstract

In this paper, we propose a novel end-to-end trainable Video Question Answering (VideoQA) framework with three major components: 1) a new heterogeneous memory which can effectively learn global context information from appearance and motion features; 2) a redesigned question memory which helps understand the complex semantics of question and highlights queried subjects; and 3) a new multimodal fusion layer which performs multi-step reasoning by attending to relevant visual and textual hints with self-updated attention. Our VideoQA model firstly generates the global context-aware visual and textual features respectively by interacting current inputs with memory contents. After that, it makes the attentional fusion of the multimodal visual and textual representations to infer the correct answer. Multiple cycles of reasoning can be made to iteratively refine attention weights of the multimodal data and improve the final representation of the QA pair. Experimental results demonstrate our approach achieves state-of-the-art performance on four VideoQA benchmark datasets.

1. Introduction

Video Question Answering (VideoQA) is to learn a model that can infer the correct answer for a given question in human language related to the visual content of a video clip. VideoQA is a challenging computer vision task, as it requires to understand a complex textual question first, and then to figure out the answer that can best associate the semantics to the visual contents in an image sequence.

Recent work [2,3,10,11,15,29] proposed to learn models of encoder-decoder structure to tackle the VideoQA problem. A common practice is to use LSTM-based encoders to encode CNN features of video frames and embeddings of question words into encoded visual sequence and word sequence. Proper reasoning is then performed to produce the correct answer, by associating the relevant visual contents with the question. For example, learning soft weights
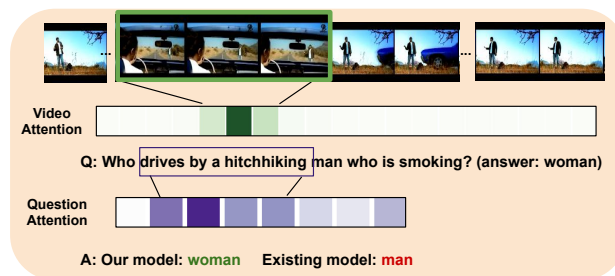

Figure 1. VideoQA is a challenging task as it requires the model to associate relevant visual contents in frame sequence with the real subject queried in question sentence. For a complex question such as "Who drives by a hitchhiking man who is smoking?", the model needs to understand that the driver is the queried person and then localize the frames in which the driver is driving in the car.

of frames will help attend to events that are queried by the questions, while learning weights of regions in every single frame will help detect details and localize the subjects in the query. The former one aims to find relevant frame-level details by applying temporal attention to encoded image sequence [10, 15, 27]. The latter one aims to find region-level details by spatial attention [2, 12, 26, 29].

Jang *et al.* [10] applied spatiotemporal attention mechanism on both spatial and temporal dimension of video features. They also proposed to use both appearance (*e.g.*, VGG [22]) and motion features (*e.g.*, C3D [24]) to better represent video frames. Their practice is to make early fusion of the two features and feed the concatenated feature to a video encoder. But such straightforward feature integration leads to suboptimal results. Gao *et al.* [5] proposed to replace the early fusion with a more sophisticated co-memory attention mechanism. They used one type of feature to attend to the other and fused the final representations of these two feature types at the final stage. However, this method doesn't synchronize the attentions detected by appearance and motion features, thus could generate incorrect attentions. Meanwhile, this method will also miss the attention which can be inferred by the combined appearance and

motion features, but not individual ones. The principal reason for the existing approaches to fail to identify the correct attention is that they separate feature integration and attention learning steps. **To address this challenging problem, we propose a new heterogeneous memory to integrate appearance and motion features and learn spatiotemporal attention simultaneously.** In our new memory model, the heterogeneous visual features as multi-input will co-learn the attention to improve the video understanding.

On the other hand, VideoQA becomes very challenging if the question has complex semantics and requires multiple steps of reasoning. Several recent work [5, 15, 32] tried to augment VideoQA with differently embodied memory networks [23, 25, 26]. Xu *et al*. [27] proposed to refine the temporal attention over video features word by word with a conventional LSTM question encoder plus an additional LSTM based memory unit to store and update the attention. However, this model is easily trapped into irrelevant local semantics, and cannot understand the question based on the global context. Both Zeng *et al*. [32] and Gao *et al*. [5] used external memory (memory network [23] and episodic memory [26] respectively) to make multiple iterations of inference by interacting the encoded question representation with video features conditioning on current memory contents. However, similar to many other work [2, 10, 26], the question representation used in these approaches is only a single feature vector encoded by an LSTM (or GRU) which lacks capability to capture complex semantics in questions such as shown in Fig. 1. Thus, it is desired to design a new powerful model for understanding the complex semantics of questions in VideoQA. **To tackle this problem, we design novel network architecture to integrate both question encoder and question memory which can augment each other. The question encoder learns meaningful representation of question and the re-designed question memory understands the complex semantics and highlights queried subjects by storing and updating global contexts.**

Moreover, we design a multimodal fusion layer which can attend to visual and question hints simultaneously by aligning relevant visual contents with key question words. After gradually refining the joint attention over video and question representations and fusing them with learned soft modality weights, the multi-step reasoning is achieved to infer the correct answer from the complex semantics.

Our major contributions can be summarized as follows: 1) we introduce a heterogeneous external memory module with attentional read and write operations such that the motion and appearance features are integrated to co-learn attention; 2) we utilize the interaction of visual and question features with memory contents to learn global context-aware representations; 3) we design a multimodal fusion layer which can effectively combine visual and question

features with softly assigned attentional weights and also support multi-step reasoning; and 4) our proposed model outperforms the state-of-the-art methods on four VideoQA benchmark datasets.

2. Related Work

Visual Question Answering (VQA) is an emerging research area [1–3, 11, 15, 17, 29] to reason the correct answer of a given question which is related to the visual content of an image. Yang *et al*. [29] proposed to encode question words into one feature vector which is used as query vector to attend to relevant image regions with stack attention mechanism. Their method supports multi-step reasoning by repeating the query process while refining the query vector. Anderson *et al*. [2] proposed to align questions with relevant object proposals in images generated by Faster R-CNN [20] and compute the visual feature as a weighted average over all proposals. Xiong *et al*. [26] proposed to encode image and question features as facts and attend to relevant facts through attention mechanism to generate a contextual vector. Ma *et al*. [15] proposed a co-attention model which can attend to not only relevant image regions but also important question words simultaneously. They also suggested to use external memory [21] to memorize uncommon QA pairs.

Video Question Answering (VideoQA) extends VQA to video domain which aims to infer the correct answer given a relevant question of the visual content of a video clip. VideoQA is considered to be a challenging problem as reasoning on video clip usually requires memorizing contextual information in temporal scale. Many models have been proposed to tackle this problem [5, 10, 27, 30–32]. Many work [5, 10, 30] utilized both motion (i.e. C3D [24]) and appearance (i.e. VGG [22], ResNet [8]) features to better represent video frames. Similar to the spatial mechanism widely used in VQA methods to find relevant image regions, many VideoQA work [5, 10, 27, 30] applied temporal attention mechanism to attend to most relevant frames of a video clip. Jang [10] utilized both appearance and motion features as video representations and applied spatial and temporal attention mechanism to attend to both relevant regions of a frame and frames of a video. Xu *et al*. [27] proposed to refine the temporal attention over frame features at each question encoding step word by word. Both Zeng *et al*. [32] and Gao *et al*. [5] proposed to use external memory (Memory Network [23] and Episodic Memory [26] respectively) to make multiple iterations of inference by interacting the encoded question feature with video features conditioning on current memory contents. Their memory designs maintain a single hidden state feature of current step and update it through time steps. However, this could hardly establish long-term global context as the hidden state feature is updated at every step. Neither are their models able to synchronize appearance and motion features.
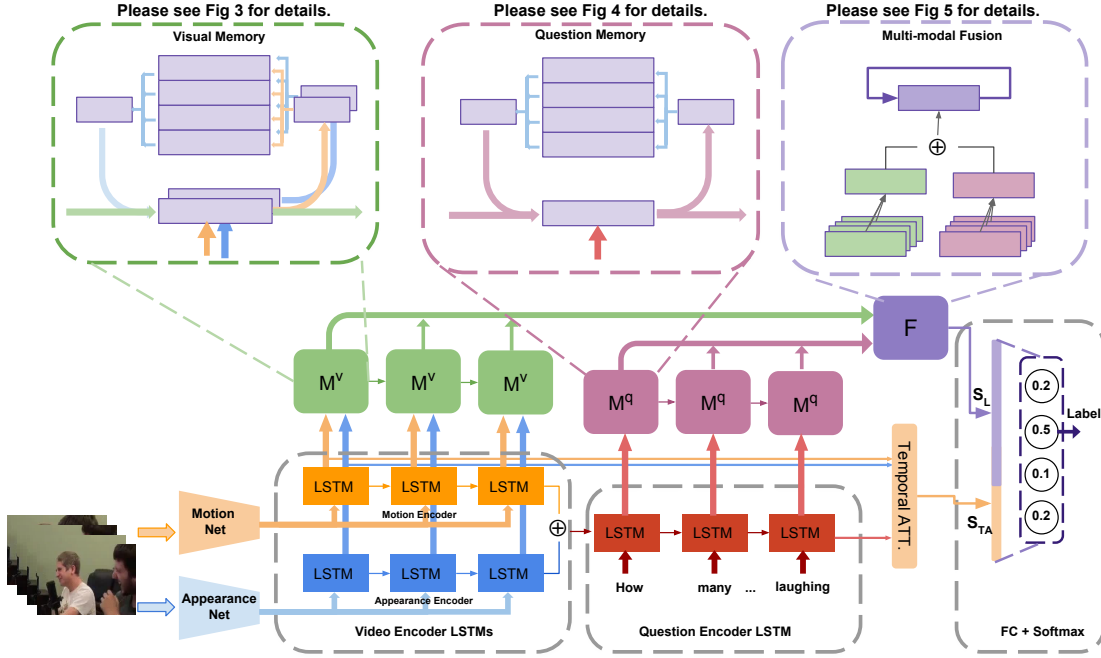
Figure 2. Our proposed VideoQA pipeline with highlighted visual memory, question memory, and multimodal fusion layer.

Our model differs from existing work such that 1) we design a heterogeneous external memory module with attentional read and write operations that can efficiently combine motion and appearance features together; 2) we allow interaction of visual and question features with memory contents to construct global context-aware features; and 3) we design a multimodal fusion layer which can effectively combine visual and question features with softly assigned attentional weights and also support multi-step reasoning.

3. Our Approach

In this section, we illustrate our network architecture for VideoQA. We first introduce the LSTM encoders for video features and question embeddings. Then we elaborate on the design of question memory and heterogeneous video memory. Finally, we demonstrate how our designed multimodal fusion layer can attend to relevant visual and textual hints and combine to form the final answer representation.

3.1. Video and text representation

Video representation. Following previous work [5, 10, 27], we sample a fixed number of frames (*e.g.*, 35 for TGIF-QA) for all videos in that dataset. We then apply pre-trained ResNet [8] or VGG [22] network on video frames to extract video appearance features, and use C3D [24] network to extract motion features. We denote appearance features as $\mathbf{f}^a = [\mathbf{f}^a_1, \cdots, \mathbf{f}^a_{N_v}]$, and motion features as $\mathbf{f}^m = [\mathbf{f}^m_1, \cdots, \mathbf{f}^m_{N_v}]$, in which N_v is number of frames. The dimensions of ResNet, VGG and C3D features are 2048, 4096 and 4096. We use two separate LSTM encoders to process motion and appearance features individually first, and late fuse them in the designed memory module which will be discussed in §3.2. In Fig. 2, we highlight the appearance encoder in blue and the motion encoder in orange. The inputs fed into the two encoders are raw CNN motion features $\mathbf{f}^m$ and appearance features $\mathbf{f}^a$, and the outputs are encoded motion and appearance features denoted as $\mathbf{o}^m = [\mathbf{o}^m_1, \cdots, \mathbf{o}^m_{N_v}]$ and $\mathbf{o}^a = [\mathbf{o}^a_1, \cdots, \mathbf{o}^a_{N_v}]$.

Question representation. Each VideoQA dataset has a pre-defined vocabulary which is composed of the top K most frequent words in the training set. The vocabulary size K of each dataset is shown in Table 1. We represent each word as a fixed-length learnable word embedding and initialize with the pre-trained GloVe 300-D [19] feature. We denote the question embedding as a sequence of word embeddings $\mathbf{f}^q = [\mathbf{f}^q_1, \cdots, \mathbf{f}^q_{N_q}]$, in which N_q is number of words in the question. We use another LSTM encoder to process question embedding $\mathbf{f}^q$, as highlighted in red in Fig. 2. The outputs are the encoded text features $\mathbf{o}^q = [\mathbf{o}^q_1, \cdots, \mathbf{o}^q_{N_q}]$.

3.2. Heterogeneous video memory

Both motion and appearance visual features are crucial for recognizing the objects and events associated with the questions. Because these two types of features are heterogeneous, the straightforward combination cannot effectively learn the video content. Thus, we propose a new heterogeneous memory to integrate motion and appearance visual features, learn the joint attention, and enhance the spatial-temporal inference.
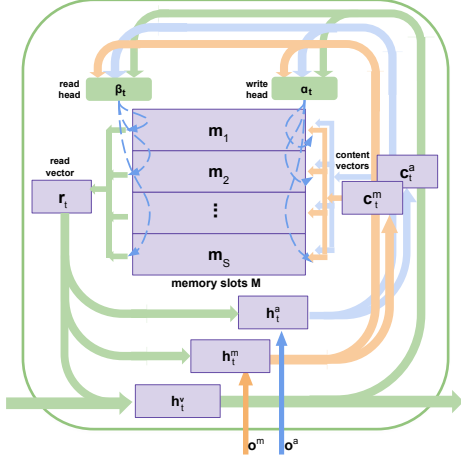
Figure 3. Our designed heterogeneous visual memory which contains memory slots M, read and write heads α, β, and three hidden states $\mathbf{h}^m$, $\mathbf{h}^a$ and $\mathbf{h}^v$.

Different to the standard external memory, our new heterogeneous memory accepts multiple inputs including encoded motion features $\mathbf{o}^m$ and appearance features $\mathbf{o}^a$, and uses multiple write heads to determine the content to write. Fig. 3 illustrates the memory structure, which is composed of memory slots $\mathbf{M} = [\mathbf{m}_1, \cdots, \mathbf{m}_S]$ and three hidden states $\mathbf{h}^m$, $\mathbf{h}^a$ and $\mathbf{h}^v$. We use two hidden states $\mathbf{h}^m$ and $\mathbf{h}^a$ to determine motion and appearance contents which will be written into memory, and use a separate global hidden state $\mathbf{h}^v$ to store and output global context-aware feature which integrates motion and appearance information. We denote the number of memory slots as S, and sigmoid function as σ. For simplicity, we combine superscript m and a for identical operations on both motion and appearance features.

Write operation. Firstly we define the motion and appearance content $\mathbf{c}_t^{m/a}$ to write to memory at t-th time as non-linear mappings from input and previous hidden state

$$\mathbf{c}_t^{m/a} = \sigma(\mathbf{W}_{oc}^{m/a}\mathbf{o}_t^{m/a} + \mathbf{W}_{hc}^{m/a}\mathbf{h}_{t\text{-}1}^{m/a} + \mathbf{b}_c^{m/a}) \quad (1)$$

Then we define $\boldsymbol{\alpha}_t^{m/a}=\{\alpha_{t,1}^{m/a}, \ldots, \alpha_{t,S}^{m/a}\}$ as the write weights of $\mathbf{c}_t^{m/a}$ to each of S memory slot given by

$$\mathbf{a}_t^{m/a} = \mathbf{v}_a^\top \tanh(\mathbf{W}_{ca}^{m/a}\mathbf{c}_t^{m/a} + \mathbf{W}_{ha}^{m/a}\mathbf{h}_{t\text{-}1}^{m/a} + \mathbf{b}_a^{m/a})$$

$$\alpha_{t,i}^{m/a} = \frac{\exp(a_{t,i}^{m/a})}{\sum_{j=1}^{S} \exp(a_{t,j}^{m/a})} \quad \text{for } i = 1 \ldots S \quad (2)$$

satisfying $\boldsymbol{\alpha}_t^{m/a}$ sum to 1. Uniquely, we also need to integrate motion and appearance information and make a unified write operation into current memory. Thus we estimate the weights $\boldsymbol{\epsilon}_t \in \mathbb{R}^3$ of motion content $\boldsymbol{\alpha}_t^m$, appearance content $\boldsymbol{\alpha}_t^a$ and current memory content $\mathbf{M}_{t\text{-}1}$ given by

$$\mathbf{e}_t = \mathbf{v}_e^\top \tanh(\mathbf{W}_{he}\mathbf{h}_{t\text{-}1}^v + (\mathbf{W}_{me}\mathbf{c}_t^m + \mathbf{W}_{ae}\mathbf{c}_t^a) + \mathbf{b}_e)$$

$$\epsilon_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{3} \exp(e_{t,j})} \quad \text{for } i = 1 \ldots 3 \quad (3)$$

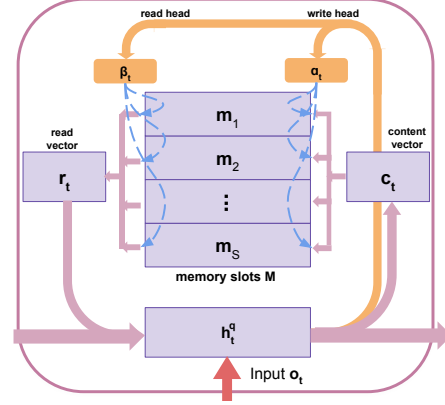Figure 4. Our re-designed question memory with memory slots M, read and write heads α, β, and hidden states $\mathbf{h}^q$.

The memory $\mathbf{M}$ can be updated at each time step by

$$\mathbf{M}_t = \epsilon_{t,1}\boldsymbol{\alpha}_t^m\mathbf{c}_t^m + \epsilon_{t,2}\boldsymbol{\alpha}_t^a\mathbf{c}_t^a + \epsilon_{t,3}\mathbf{M}_{t\text{-}1} \quad (4)$$

in which the write weights $\boldsymbol{\alpha}_t^{m/a}$ for memory slots determine how much attention should different slots pay to current inputs, while the modality weights $\boldsymbol{\epsilon}_t$ determine which of motion or appearance feature (or none of them if non-informational) from current inputs should the memory pay more attention to. Through this designed memory-write mechanism, we are able to integrate motion and appearance features to learn joint attention, and memorize different spatio-temporal patterns of this video in a synchronized and global context.

Read operation. The next step is to perform an attentional read operation from the memory $\mathbf{M}$ to update memory hidden states. We define the weights of reading from memory slots as $\boldsymbol{\beta}_t=\{\beta_{t,1}, \ldots, \beta_{t,S}\}$ given by

$$\mathbf{b}_t = \mathbf{v}_b^\top \tanh(\mathbf{W}_{hb}\mathbf{h}_{t\text{-}1}^v + (\mathbf{W}_{mb}\mathbf{c}_t^m + \mathbf{W}_{ab}\mathbf{c}_t^a) + \mathbf{b}_b)$$

$$\beta_{t,i} = \frac{\exp(b_{t,i})}{\sum_{j=1}^{S} \exp(b_{t,j})} \quad \text{for } i = 1 \ldots S \quad (5)$$

The content $\mathbf{r}_t$ read from memory is the weighted sum of each memory slot $\mathbf{r}_t = \sum_{i=1}^{S} \beta_{t,i} \cdot \mathbf{m}_i$ in which both motion and appearance information has been integrated.

Hidden states update. The final step is to update all three hidden states $\mathbf{h}^a$, $\mathbf{h}^m$ and $\mathbf{h}^v$

$$\mathbf{h}_t^{m/a} = \sigma(\mathbf{W}_{hh}^{m/a}\mathbf{h}_{t\text{-}1}^{m/a} + \mathbf{W}_{oh}^{m/a}\mathbf{o}_t^{m/a} \quad (6)$$
$$+ \mathbf{W}_{rh}^{m/a}\mathbf{r}_t + \mathbf{b}_h^{m/a})$$
$$\mathbf{h}_t^v = \sigma(\mathbf{W}_{hh}^v\mathbf{h}_{t\text{-}1}^v + \mathbf{W}_{rh}^v\mathbf{r}_t + \mathbf{b}_h^v) \quad (7)$$

The global memory hidden state at all time steps $\mathbf{h}_{1:N_v}^v$ will be taken as our final video features. In next section, we will discuss how to generate global question features. In Section 3.4, we will introduce how to interact video and question features for answer inference.

3.3. External question memory

The existing deep learning based VideoQA methods often misunderstand the complex questions because they understand the questions based on local word information. For example, for question "Who drives by a hitchhiking man who is smoking?", traditional methods are easily trapped by the local words and fail to generate the right attention to the queried person (the driver or the smoker). To address this challenging problem, we introduce the question memory to learn context-aware text knowledge. The question memory can store the sequential text information, learn relevance between words, and understand the question from the global point of view.

We redesign the memory networks [6, 16, 23, 25] to persistently store previous inputs and enable interaction between current inputs and memory contents. As shown in Fig. 4, the memory module is composed of memory slots $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_S]$ and memory hidden state $\mathbf{h}^q$. Unlike the heterogeneous memory discussed previously, one hidden state $\mathbf{h}^q$ is necessary for the question memory. The inputs to the question memory are the encoded texts $\mathbf{o}^q$.

Write operation. We first define the content to write to the memory at t-th time step as $\mathbf{c}_t^q$ which is given by

$$\mathbf{c}_t^q = \sigma(\mathbf{W}_{oc}\mathbf{o}_t^q + \mathbf{W}_{hc}\mathbf{h}_{t-1}^q + \mathbf{b}_c) \qquad (8)$$

as a non-linear mapping from current input $\mathbf{o}_t^q$ and previous hidden state $\mathbf{h}_{t-1}^q$ to content vector $\mathbf{c}_t^q$. Then we define the weights of writing to all memory slots $\boldsymbol{\alpha}_t = \{\alpha_{t,1}...\alpha_{t,i}...\alpha_{t,S}\}$ such that

$$\mathbf{a}_t = \mathbf{v}_a^\top \tanh(\mathbf{W}_{ca}\mathbf{c}_t^q + \mathbf{W}_{ha}\mathbf{h}_{t-1}^q + \mathbf{b}_a)$$

$$\alpha_{t,i} = \frac{\exp(a_{t,i})}{\sum_{j=1}^{S} \exp(a_{t,j})} \qquad \text{for } i = 1 \dots S \qquad (9)$$

satisfying $\boldsymbol{\alpha}_t$ sum to 1. Then each memory slot $\mathbf{m}_i$ is updated by $\mathbf{m}_i = \alpha_{t,i}\mathbf{c}_t + (1 - \alpha_{t,i})\mathbf{m}_i$ for $i = 1 \dots S$.

Read operation. The next step is to perform attentional read operation from the memory slots $\mathbf{M}$. We define the normalized attention weights $\boldsymbol{\beta}_t = \{\beta_{t,1}...\beta_{t,i}...\beta_{t,S}\}$ of reading from memory slots such that

$$\mathbf{b}_t = \mathbf{v}_b^\top \tanh(\mathbf{W}_{cb}\mathbf{c}_t^q + \mathbf{W}_{hb}\mathbf{h}_{t-1}^q + \mathbf{b}_b)$$

$$\beta_{t,i} = \frac{\exp(b_{t,i})}{\sum_{j=1}^{S} \exp(b_{t,j})} \qquad \text{for } i = 1 \dots S \qquad (10)$$

The content $\mathbf{r}_t$ read from memory is the weighted sum of each memory slot content $\mathbf{r}_t = \sum_{i=1}^{S} \beta_{t,i} \cdot \mathbf{m}_i$.

Hidden state update. The final step of t-th iteration is to update the hidden state $\mathbf{h}_t^q$ as

$$\mathbf{h}_t^q = \sigma(\mathbf{W}_{oh}\mathbf{o}_t^q + \mathbf{W}_{rh}\mathbf{r}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1}^q + \mathbf{b}_h) \qquad (11)$$

We take the memory hidden state of all time steps $\mathbf{h}_{1:N_q}^q$ as the global context-aware question features which will be used for inference in Section 3.4.
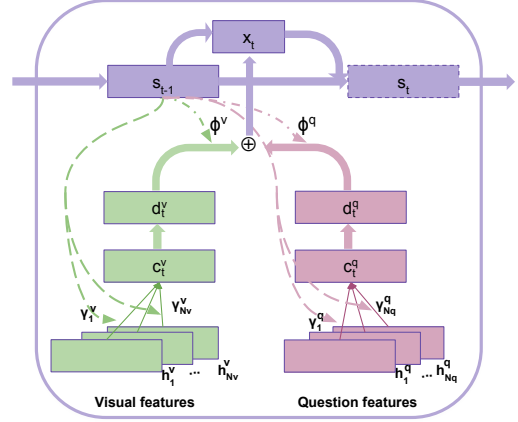

Figure 5. Multimodal fusion layer. An LSTM controller with hidden state $\mathbf{s}_t$ attends to relevant visual and question features, and combines them to update current state.

3.4. Multimodal fusion and reasoning

In this section, we design a dedicated multimodal fusion and reasoning module for VideoQA, which can attend to multiple modalities such as visual and textual features, then make multi-step reasoning with refined attention for each modality. Our design is inspired by Hori *et al.* [9] which proposed to generate video captions by combining different types of features such as video and audio.

Fig. 5 demonstrates our designed module. The hidden states of video memory $\mathbf{h}_{1:N_v}^v$ and question memory $\mathbf{h}_{1:N_q}^q$ are taken as the input features. The core part is an LSTM controller with its hidden state denoted as $\mathbf{s}$. During each iteration of reasoning, the controller attends to different parts of the video features and question features with temporal attention mechanism, and combines the attended features with learned modality weights ϕ_t, and finally updates its own hidden state $\mathbf{s}_t$.

Temporal attention. At t-th iteration of reasoning, we first generate two content vectors $\mathbf{c}_t^v$ and $\mathbf{c}_t^q$ by attending to different parts of visual features $\mathbf{h}_t^v$ and question features $\mathbf{h}_t^q$. The temporal attention weights $\gamma_{1:N_v}^v$ and $\gamma_{1:N_q}^q$ are computed by

$$\mathbf{g}^{v/q} = \mathbf{v}_g^{v/q\top} \tanh(\mathbf{W}_g^{v/q}\mathbf{s}_{t-1} + \mathbf{V}_g^{v/q}\mathbf{h}^{v/q} + \mathbf{b}_g^{v/q})$$

$$\gamma_i^{v/q} = \frac{\exp(g_i^{v/q})}{\sum_{j=1}^{N_{v/q}} \exp(g_j^{v/q})} \qquad \text{for } i = 1 \dots N_{v/q} \qquad (12)$$

and shown by the dashed lines in Fig. 5. Then the attended content vectors $\mathbf{c}_t^{v/q}$ and the transformed $\mathbf{d}_t^{v/q}$ are

$$\mathbf{c}_t^{v/q} = \sum_{i=1}^{N_{v/q}} \gamma_i^{v/q}\mathbf{h}_i^{v/q}, \; \mathbf{d}_t^{v/q} = \text{ReLU}(\mathbf{W}_d^{v/q}\mathbf{c}_t^{v/q} + \mathbf{b}_d^{v/q}) \qquad (13)$$

Multimodal fusion. The multimodal attention weights $\phi_t = \{\phi_t^v, \phi_t^q\}$ are obtained by interacting the previous hid-

den state $\mathbf{s}_{t\text{-}1}$ with the transformed content vectors $\mathbf{d}_t^{v/q}$

$$\mathbf{p}_t^{v/q} = \mathbf{v}_p^\top \tanh(\mathbf{W}_p^{v/q}\mathbf{s}_{t\text{-}1} + \mathbf{V}_p^{v/q}\mathbf{d}_t^{v/q} + \mathbf{b}_p^{v/q})$$
$$\phi_t^{v/q} = \frac{\exp(p_t^{v/q})}{\exp(p_t^v) + \exp(p_t^q)} \tag{14}$$

The fused knowledge $\mathbf{x}_t$ is computed by the sum of $\mathbf{d}_t^{v/q}$ with multimodal attention weights $\phi^{v/q}$ such that

$$\mathbf{x}_t = \phi_t^v \mathbf{d}_t^v + \phi_t^q \mathbf{d}_t^q \tag{15}$$

Multi-step reasoning. To complete t-th iteration of reasoning, the hidden state $\mathbf{s}_t$ of LSTM controller is updated by $\mathbf{s}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{s}_{t\text{-}1})$. This reasoning process is iterated for L times and we set $L = 3$. The optimal choice for L is discussed in §4.4. The hidden state $\mathbf{s}_L$ at last iteration is the final representation of the distilled knowledge. We also apply the standard temporal attention on encoded video features $\mathbf{o}^m$ and $\mathbf{o}^a$ as in ST-VQA [10], and concatenate with $\mathbf{s}_L$ to form the final answer representation $\mathbf{s}_A$.

3.5. Answer generation

We now discuss how to generate the correct answers from answer features $\mathbf{s}_A$.

Multiple-choice task is to choose one correct answer out of K candidates. We concatenate the question with each candidate answer individually, and forward each QA pair to obtain the final answer feature $\{\mathbf{s}_A\}_{i=1}^K$, on top of which we use a linear layer to provide scores for all candidate answers $\mathbf{s} = \{s^p, s_1^n, \cdots, s_{K-1}^n\}$ in which s^p is the correct answer's score and the rest are $K-1$ incorrect ones. During training, we minimize the summed pairwise hinge loss [10] between the positive answer and each negative answer defined as

$$L_{mc} = \sum_{i=1}^{K-1} \max(0, m - (s^p - s_i^n)) \tag{16}$$

and train the entire network end-to-end. The intuition of L_{mc} is that the score of the true QA pair should be larger than any negative pair by a margin m. During testing, we choose the answer of highest score as the prediction. In Table 1, we list the number of choices K for each dataset.

Open-ended task is to choose one correct word as the answer from a pre-defined answer set of size C. We apply a linear layer and softmax function upon $\mathbf{s}_A$ to provide probabilities for all candidate answers such that $\mathbf{p} = \text{softmax}(\mathbf{W}_p^\top \mathbf{s}_L + \mathbf{b}_\mathbf{p})$ in which $\mathbf{p} \in \mathbb{R}^C$. The training error is measured by cross-entropy loss such that

$$L_{open} = -\sum_{c=1}^C \mathbb{1}\{y = c\} \log(p_c) \tag{17}$$

in which y is the ground truth label. By minimizing L_{open} we can train the entire network end-to-end. In testing phase, the predicted answer is provided by $c^* = \arg\max_c(\mathbf{p})$.

3.6. Implementation details

We implemented our neural networks in PyTorch [18] and updated network parameters by Adam solver [13] with batch size 32 and fixed learning rate 10^{-3}. The video and question encoders are two-layer LSTMs with hidden size 512. The dimension D of the memory slot and hidden state is 256. We set the video and question memory sizes to 30 and 20 respectively, which are roughly equal to the maximum length of the videos and questions. We have released our code for boosting further research[1].

4. Experiments and Discussions

We evaluate our model on four benchmark VideoQA datasets and compare with the state-of-the-art techniques.

4.1. Dataset descriptions

In Table 1, we show the statistics of the four VideoQA benchmark datasets and the experimental settings from their original paper including feature types, vocabulary size, sampled video length, number of videos, size of QA splits, answer set size for open-ended questions, and number of options for multiple-choice questions.

TGIF-QA [10] contains 165K QA pairs associated with 72K GIF images based on the TGIF dataset [14]. TGIF-QA includes four types of questions: 1) counting the number of occurrences of a given action; 2) recognizing a repeated action given its count; 3) identifying the action happened before or after a given action, and 4) answering image-based questions. **MSVD-QA** and **MSRVTT-QA** were proposed by Xu *et al.* [27] based on MSVD [4] and MSVTT [28] video sets respectively. Five different question types exist in both datasets, including what, who, how, when and where. The questions are open-ended with pre-defined answer sets of size 1000. **YouTube2Text-QA** [30] collected three types of questions (what, who and other) from YouTube2Text [7] video description corpus. The video source is also MSVD [4]. Both open-ended and multiple-choice tasks exist.

4.2. Result analysis

TGIF-QA result. Table 2 summarizes the experiment results of all four tasks (Count,Action,Trans.,FrameQA) on TGIF-QA dataset. We compare with state-of-the-art methods ST-VQA [10] and Co-Mem [5] and list the reported accuracy in the original paper. For *repetition counting* task (column 1), our method achieves the lowest average L_2 loss compared with ST-VQA and Co-Mem (4.02 v.s. 4.28 and 4.10). For *Action* and *Trans.* tasks (column 2,3), our method significantly outperforms the other two by increasing accuracy from prior best 0.682 and 0.743 to 0.739 and 0.778 respectively. For *FrameQA* task (column 4), our method also

---

[1]https://github.com/fanchenyou/HME-VideoQA

Dataset	Feature	Vocab size	Video len	Video num	Question num			Ans size	MC num
					Train	Val	Test		
TGIF-QA [10]	ResNet+C3D	8,000	35	71,741	125,473	13,941	25,751	1746	5
MSVD-QA [27]	VGG+C3D	4,000	20	1,970	30,933	6,415	13,157	1000	NA
MSRVTT-QA [27]	VGG+C3D	8,000	20	10,000	158,581	12,278	72,821	1000	NA
Youtube2Text-QA [30]	ResNet+C3D	6,500	40	1,970	88,350	6,489	4,590	1000	4

Table 1. Dataset statistics of four VideoQA benchmark datasets. The columns from left to right indicate dataset name, feature types, vocabulary size, sampled video length, number of videos, size of QA splits, answer set size (Ans size) for open-ended questions, and number of options for multiple-choice questions (MC num).

Method	Question type			
	Count (loss)	Action	Trans.	FrameQA
ST-VQA [10]	4.28	0.608	0.671	0.493
Co-Mem [5]	4.10	0.682	0.743	0.515
Ours	**4.02**	**0.739**	**0.778**	**0.538**

Table 2. Experiment results on TGIF-QA dataset.

achieves the best accuracy of 0.538 among all three methods, outperforming the Co-Mem by 4.7%.

Method	Question type and # instances					
	What 8419	Who 4552	How 370	When 58	Where 28	All 13427
ST-VQA [10]	0.181	**0.500**	**0.838**	0.724	0.286	0.313
Co-Mem [5]	0.196	0.487	0.816	**0.741**	0.317	0.317
AMU [27]	0.206	0.475	0.835	0.724	**0.536**	0.320
Ours	**0.224**	**0.501**	0.730	0.707	0.429	**0.337**

Table 3. Experiment results on MSVD-QA dataset.

MSVD-QA result. Table 3 summarizes the experiment results on MSVD-QA. It's worth mentioning that there is high class imbalance in both training and test sets, as more than 95% questions are *what* and *who* while less than 5% are *how*, *when* and *where*. We list the numbers of their test instances in the table for reference. We compare our model with the ST-VQA [10], Co-Mem [5] and current state-of-the-art AMU [27] on MSVD-QA. We show the reported accuracy of AMU in [27], while we accommodate the source code of ST-VQA and implement Co-Mem from scratch to obtain their numbers. Our method outperforms all the others on both *what* and *who* tasks, and achieves best overall accuracy of 0.337 which is 5.3% better than prior best (0.320). Even though our method slightly underperforms on the How, When and Where questions, the difference are minimal (40,2 and 3) regarding the absolute number of instances due to class imbalance.

Method	Question type					
	What	Who	How	When	Where	All
ST-VQA [10]	0.245	0.412	0.780	**0.765**	0.349	0.309
Co-Mem [5]	0.239	0.425	0.741	0.690	**0.429**	0.320
AMU [27]	0.262	0.430	0.802	0.725	0.300	0.325
Ours	**0.265**	**0.436**	**0.824**	0.760	0.286	**0.330**

Table 4. Experiment results on MSRVTT-QA dataset.

MSRVTT-QA result. In Table 4, we compare our model with the ST-VQA [10], Co-Mem [5] and AMU [27] on MSRVTT-QA. Similar to the trend on MSVD-QA, our method outperforms the other models on three major question types (*what*, *who*, *how*), and achieves the best overall accuracy of 0.330.

Task	Method	Question type and # instances				
		What 2489	Who 2004	Other 97	All 4590	Avg. Per-class
Multi-choice	r-ANL [30]	0.633	0.364	0.845	0.520	0.614
	Ours	**0.831**	**0.778**	**0.866**	**0.808**	**0.825**
Open-ended	r-ANL [30]	0.216	**0.294**	**0.804**	0.262	0.438
	Ours	**0.292**	0.287	0.773	**0.301**	**0.451**

Table 5. Experiment results on YouTube2Text-QA dataset.

YouTube2Text-QA result. In Table 5, we compare our methods with the state-of-the-art r-ANL [30] on YouTube2Text-QA dataset. It's worth mentioning that r-ANL utilized frame-level attributes as additional supervision to augment learning while our method does not. For multiple-choice questions, our method significantly outperforms r-ANL on all three types of questions (*What, Who, Other*) and achieves a better overall accuracy (0.808 v.s. 0.520). For open-ended questions, our method outperforms r-ANL on *what* queries and slightly underperforms on the other two types. Still, our method achieves a better overall accuracy (0.301 v.s. 0.262). We also report the per-class accuracy to make direct comparison with [30], and our method is better than r-ANL in this evaluation method.

4.3. Attention visualization and analysis

In Figs. 1 and 6, we demonstrate three QA examples with highlighted key frames and words which are recognized by our designed attention mechanism. For visualization purpose, we extract the visual and textual attention weights from our model (Eq. 12) and plot them with bar charts. Darker color stands for larger weights, showing that the corresponding frame or word is relatively important.

Fig. 1 shows the effectiveness of understanding complex question with our proposed question memory. This question intends to query the female driver though it uses another relative clause to describe the man. Our model focuses on the correct frames in which the female driver is driving in the car and also focuses on the words which describe the woman but not the man. In contrast, ST-VQA [10] fails to
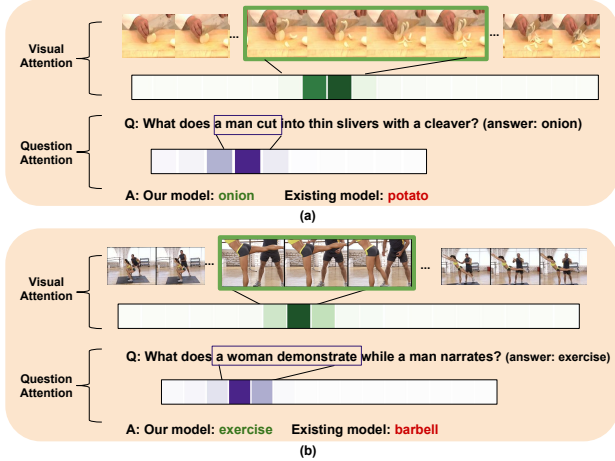
Figure 6. Visualization of multimodal attentions learned by our model on two QA exemplars. Highly attended frames and words are highlighted.

identify the queried person as its simple temporal attention is not able to gather semantic information in the context of a long sentence.

In Fig. 6(a), we provide an example showing that our video memory is learning the most salient frames for the given question while ignoring others. In the first half of the video, it's difficult to know whether the vegetable is onion or potato, due to the lighting condition and camera view. However, our model smartly pays attention to frames in which the onion is cut into pieces by combining both question words "a man cut" and the motion features, and thus determines the correct object type by onion pieces (but not potato slices) from appearance hint.

Fig. 6(b) shows a typical example illustrating that jointly learning motion and appearance features as our heterogeneous memory design is superior to attending to them separately such as Co-Mem [5]. In this video, a woman is doing yoga in a gym, and there is a barbell rack at the background. Our method successfully associated the woman with the action of exercising, while Co-Mem [5] incorrectly pays attention to the barbell and fails to utilize motion information as they separately learn motion and appearance attentions.

4.4. Ablation study

We perform two ablation studies to investigate the effectiveness of each component of our model. We first study how many iterations of reasoning is sufficient in the designed multimodal fusion layer. After that, we make a comparison of variants of our model to evaluate the contribution of each component.

Reasoning iterations. To understand how many iterations of reasoning are sufficient for our VideoQA tasks, we test different numbers and report their accuracy. The validation accuracy on MSVD-QA dataset increases from 0.298

to 0.306 when the number of reasoning iteration L increases from 1 to 3, and seems to saturate at $L = 5$ (0.307), while drops to 0.304 at $L = 7$. To balance performance and speed, we choose $L = 3$ for our experiments throughout the paper.

Dataset	EF	LF	E-M	V-M	Q-M	V+Q
MSVD	0.313	0.315	0.318	0.320	0.315	**0.337**
MSRVTT	0.309	0.312	0.319	0.325	0.321	**0.330**

Table 6. Ablation study of different architectures.

Different architectures. To understand the effectiveness of our designed memory module, we compare several variants of our models and evaluate on MSVD-QA and MSRVTT-QA, as shown in Table 6. *Early Fusion (EF)* is indeed ST-VQA [10] which concatenates raw video appearance and motion features at an early stage, before feeding into the LSTM encoder. *Late Fusion (LF)* model uses two separate LSTM encoders to encode video appearance and motion features and then fuses them by concatenation. *Episodic Memory (E-M)* [26] is a simplified memory network embodiment and we use it as the visual memory to compare against our design. *Visual Memory (V-M)* model uses our designed heterogeneous visual memory (M^v in Fig. 2) to fuse appearance and motion features and generate global context-aware video features. *Question Memory (Q-M)* model uses our redesigned question memory only (M^q in Fig. 2) to better capture complex question semantics. Finally, *Visual and Question Memory (V+Q M)* is our full model which has both visual and question memory.

In Table 6, we observe consistent trend that using memory networks (e.g., E-M,V-M,V+Q) to align and integrate multimodal visual features is generally better than simply concatenating them (e.g., EF,LF). In addition, our designed visual memory (V-M) has shown its strengths over episodic memory (E-M) and other memory types (Table 3-5). Furthermore, using both visual memory and question memory (V+Q) increases the performance by 2-7%.

5. Conclusion

In this paper, we proposed a novel end-to-end deep learning framework for VideoQA, with designing new external memory modules to better capture global contexts in video frames, complex semantics in questions, and their interactions. A new multimodal fusion layer was designed to fuse visual and textual modalities and perform multi-step reasoning with gradually refined attention. In empirical studies, we visualized the attentions generated by our model to verify its capability of understanding complex questions and attending to salient visual hints. Experimental results on four benchmark VideoQA datasets show that our new approach consistently outperforms state-of-the-art methods.

References

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C Lawrence Zitnick, Dhruv Batra, and Devi Parikh. VQA: Visual question answering. In *ICCV*, 2015.

[2] 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*, 2018.

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016.

[4] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

[5] Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. Motion-appearance co-memory networks for video question answering. In *CVPR*, 2018.

[6] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. In *arXiv preprint arXiv:1410.5401*, 2014.

[7] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *ICCV*, 2017.

[10] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017.

[11] Aniruddha Kembhavi, MinJoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Textbook question answering for multimodal machine comprehension. In *CVPR*, 2017.

[12] Jin-Hwa Kim, Sang-Woo Lee, Donghyun Kwak, Min-Oh Heo, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Multimodal residual learning for visual QA. In *NIPS*, 2016.

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

[14] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *CVPR*, 2016.

[15] Chao Ma, Chunhua Shen, Anthony Dick, Qi Wu, Peng Wang, Anton van den Hengel, and Ian Reid. Visual question answering with memory-augmented networks. In *CVPR*, 2018.

[16] Ying Ma and Jose Principe. A taxonomy for neural memory networks. In *arXiv preprint arXiv:1805.00327*, 2018.

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

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *EMNLP*, 2014.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[21] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[23] Sainbayar Sukhbaatar, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NIPS*, 2015.

[24] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatio-temporal features with 3d convolutional networks. In *ICCV*, 2015.

[25] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *ICLR*, 2015.

[26] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016.

[27] 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 *ACMMM*, 2017.

[28] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016.

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

[30] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. Video question answering via attribute-augmented attention network learning. In *SIGIR*, 2017.

[31] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 2016.

[32] Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. Leveraging video descriptions to learn video question answering. In *AAAI*, 2017.