Video Question Answering Using Language-Guided Deep Compressed-Domain Video Feature

Nayoung Kim^{+}, Seong Jong Ha*, Je-Won Kang$^{+,\dagger}$
$^{+}$Department of Electronic and Electrical Engineering, Ewha W. University, Korea
*Vision AI Lab, AI Center, NCSOFT
†Graduate Program in Smart Factory, Ewha W. University, Korea
1210513skdud@ewhain.net, seongjongha@ncsoft.com, jewonk@ewha.ac.kr

Abstract

Video Question Answering (Video QA) aims to give an answer to the question through semantic reasoning between visual and linguistic information. Recently, handling large amounts of multi-modal video and language information of a video is considered important in the industry. However, the current video QA models use deep features, suffered from significant computational complexity and insufficient representation capability both in training and testing. Existing features are extracted using pre-trained networks after all the frames are decoded, which is not always suitable for video QA tasks. In this paper, we develop a novel deep neural network to provide video QA features obtained from coded video bit-stream to reduce the complexity. The proposed network includes several dedicated deep modules to both the video QA and the video compression system, which is the first attempt at the video QA task. The proposed network is predominantly model-agnostic. It is integrated into the state-of-the-art networks for improved performance without any computationally expensive motion-related deep models. The experimental results demonstrate that the proposed network outperforms the previous studies at lower complexity. https://github.com/Nayoung-Kim-ICP/VQAC

1. Introduction

Recent advances in artificial intelligence (AI) have brought significant attention to the multidisciplinary research area of computer vision (CV) and natural langue processing (NLP). Video question answering (QA) aims to give a reasonable answer by jointly conducting visual understanding and language-specific reasoning. It has a number of real-time emerging intelligent applications such as human-AI interactions and communication systems.

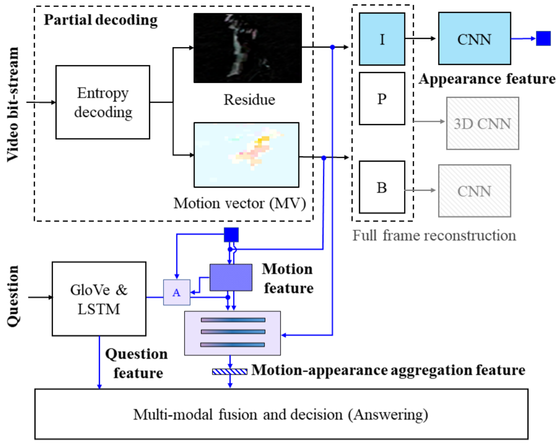

Figure 1. Motivation of the proposed VQAC-baseline network architecture. It retrieves residue and motion vectors (MVs) from a coded bit-stream from only the partial decoding to save computational resources. The compressed-domain features are used for generating a motion-appearance aggregation (MA+) feature.

Previous video QA studies have focused on developing sophisticated deep learning models to resolve diverse reasoning problems in multimodal data. In recent studies [14, 43, 9, 45, 12], the QA models incorporated external memories [14, 12] and attention mechanisms [42, 45] to improve performance. Nevertheless, the previous studies straightforwardly used the same baseline neural network architecture for extracting video features and question features. A convolutional neural network (CNN) and 3D-CNN [34, 6] are used for an appearance feature and a motion feature, respectively. A recurrent neural network (RNN) is used for a question feature [11, 25]. Given with the separately generated features, the previous models were used to understand semantic relations to answer the question. However, several studies indicated that the current approaches suffered from significantly degraded performance when the features lacked sufficient representation capabilities [5, 10]. It is problematic that the current baseline model naively uses a pre-trained neural network for extracting features.

Few studies have redesigned a baseline structure to provide more efficient features in the video QA task, likely because many computational resources are required to exploit QA features both for training and testing. 3D-CNN is extremely complex, despite being developed for representing a homogenous motion. For lightweight features, we introduce compressed-domain features that are included in a bitstream of coded video data. Video compression enables a sequence of frames to be reconstructed using only a few anchor frames named an intra-coded frame (I-frame) with complete RGB data and several ingredients for prediction such as residue and a motion vector (MV). Because most video content is compressed in advance and the residue and MVs are readily obtained as intermediate outputs during decompression, various CV tasks could be facilitated in the previous studies [33, 31, 3, 38].

This paper proposes a time-efficient video QA network using compressed-domain video features (VQAC) to improve performance at lower complexity. Previous QA works are difficult to apply directly to compressed video data. Conventional video features such as C3D [34] and I3D [6] can only be created if complete video frames are available after decompression. However, full decompression requires extra latency and extensive storage, which further deteriorates computational complexity for feature extraction.

In our framework, for an appearance feature, a pre-trained CNN [30, 16] is applied only to I-frames to avoid any delay or latency, as depicted in Fig. 1, because non-anchor P- and B-frames are only available after the I-frames are fully reconstructed. For motion, residue and MVs are first retrieved with only the partial decoding of P- and B-frames to avoid their full reconstruction. The compressed-domain features are then used for generating motion features to replace the existing 3D-CNN.

Our approach is the first attempt to apply compressed-domain features to video QA tasks. Previously, Shou et al. [29] and Wu et al. [38] proposed to exploit compressed-domain features in action recognition tasks, identifying only a few representative motions in a video. Compared with other CV tasks, the video QA model needs to achieve a more comprehensive and semantically aligned interpretation of a video and query. However, it is computationally intractable to learn such features in end-to-end, considering the nature of multimodal data. These problems motivate us to apply the compressed-domain features to the QA task. While the previous studies have only few choices of the pre-trained features, the compressed features are readily obtained by decompressing existing data.

The VQAC network produces video QA features that consider different modalities and more efficient alignments. Fig. 1 shows an overall scheme of the VQAC-baseline network. The network creates a motion-appearance aggregation (MA+) feature as output. It is promptly generated by warping the current appearance feature using a MV and adapting to temporal dynamics using a residue. The MA+ feature is fused with question features to a decision network to exploit inter-modal correlation, which is crucial for video-related multimodal tasks. The VQAC-baseline can be used as a standalone model to operate very fast.

Furthermore, the network can be integrated into the existing video QA models because the baseline network is predominantly model agnostic. Previous studies [14, 12] have attempted to improve performance by understanding global contexts over a video. The current state-of-the-art networks commonly use memory modules to retain global appearance and motion features using read and write operations. Therefore, we present a VQAC-integration model that uses both the proposed QA features and some global features by combining the baseline with the existing model, which maintains global video and question features.

Our primary contributions are summarized as follows:

- We present a VQAC-baseline network to resolve the major drawbacks of the previous video QA features: significant computational complexity and insufficient representation capability. We introduce compressed-domain features and develop several dedicated modules to both the video QA and the video compression system, which is the first attempt at the video QA task.

- We develop a VQAC-integration network to integrate the baseline model for improved performance without any computationally expensive motion models [34]. The VQAC-integration model outperforms the previous studies for various video QA datasets.

2. Related Works

2.1. Previous video QA studies

Previous image and video QA studies attempted to build deep learning models to train a joint representation of visual and language information [2, 23, 7]. For image QA tasks, CNNs such as VGG [30] and ResNet [16] are used to extract appearance features, and RNNs such as long short-term memory (LSTM) are used to encode a sequence of word embedding initialized with GloVe[25] or BERT[11]. For video QA tasks, motion features are included using 3D-CNNs such as C3D [34] and I3D [6]. However, straightforward QA models using CNNs and RNNs could not preserve key information in long video or wordy sentences [47, 41].

The problems are caused by forgotten QA features in the previous video frames. Several studies have attempted to manage critical features over an entire video and sentences using attention and fusion mechanisms [47, 18, 14, 41, 12, 4]. Temporal visual attention was proposed to exploit temporal correlation among successive frames [27, 44]. The mechanism was extended to use spatiotemporal attention [34, 47, 18, 45, 45]. In [21], a CNN-LSTM network was used for learning cross-modal features. In [15], a self-attention mechanism was applied to each frame without extracting motion.

Memory modules have been efficiently used for preserving the global contexts to improve the performance [40, 14, 22, 12, 43]. Xiong et al. developed a dynamic memory network [40] to manage long-term and short-term contexts. In [14], the memory module was used for combining motion and appearance features in co-memory attention for reasoning. In [12], a heterogeneous memory module was used to train joint attention for global video features.

2.2. Compressed-domain features in video

Video coding standard In existing video coding standards [36, 32], a video sequence is divided into a group of picture (GOP), and each frame within a GOP is coded as an I-, P-, and B- frame. An I-frame is the first frame of the GOP to maintain full RGB pixels as an anchor. The subsequent P- and B-frames are then coded with temporal prediction using a block-based MV. The prediction is conducted by finding the closest matching block of a previously coded frame as a reference frame, and a (x, y) vector of the current block to the reference block is determined as the MV. Because the current block and the matching block are usually not the same, the transformed residue is sent to a decoder.

An I-frame is independently decoded because it uses no temporal prediction. In contrast, P- and B-frames are fully reconstructed after all the reference frames are available. In the worst case, an inter-coded frame can start decoding after all the other frames in the same GOP are fully reconstructed. However, the MV and residue can be obtained immediately as they are partially decoded.

Compressed-domain features MVs and residue as compressed-domain features have been widely used for many vision tasks, such as action recognition [29, 38, 46], saliency detection [20], and video summarization [1]. Because an MV is produced to minimize the difference between the current and the reference block, it can reflect a locally temporal change of a foreground object. Furthermore, the residue can represent the abrupt changes in RGB values. The amount of residue tends to be larger in boundaries of fast-moving objects and scene changes.

Compressed-domain features provided several advantages. First, the computational costs are much lower than other deep features, such as deep flows [17]. Although deep features require abundant video data for training, the compressed-domain features can be obtained from the decoding process. The compressed-domain features do not even require full-frame reconstruction. The decoding process consists of entropy decoding, inverse transform and quantization, and motion-compensation. The features can be extracted while skipping the motion-compensation process, which is the most complicated decoding process. Second, the compressed-domain features do not suffer from delay or dependency problems caused by temporal prediction because they are instantly obtained. The advantages have increased the number of use cases of compressed features in CV. However, such efforts have rarely been attempted in QA tasks.

3. Proposed Method

3.1. VQAC-baseline network

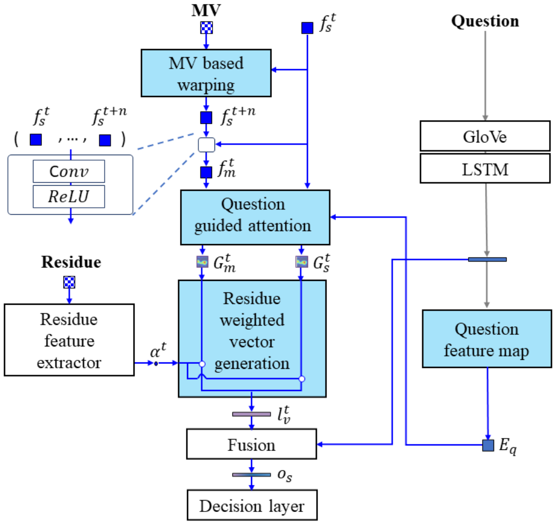

Figure 2. VQAC-baseline network architecture, in which the core modules such as MV-based warping, question guided attention, question feature map, and residue weighted vector generation are indicated by different colors.

Fig. 2 illustrates a block diagram of the VQAC-baseline network architecture exploiting compressed-domain features. Core modules are represented by colored boxes. An appearance feature $f_s^t \in \mathbb{R}^{d_h \times d_w \times d_c}$ is extracted from every I-frame at time t using a pretrained CNN where d_h, d_w and d_c are feature dimensions of height, width, and channel, respectively. The size of a GOP is set to 16 in experiments. A question feature $f_w \in \mathbb{R}^{N_w \times d_r}$ is initialized with Glove [25] and encoded with LSTM as in [12] where N_w is the number of input words and d_r is dimension of word features. We remove the conventional 3D-CNN for extracting a motion feature. Instead, a motion feature f_m^t is obtained by temporal convolution using an MV.

The features are combined to create an MA+ feature l_v^t to adapt for time-varying characteristics in a video to answer a question. Intuitively, when a scene changes abruptly and an object moves fast, the network can obtain more reliable features by analyzing local information. Otherwise, the complicated motion dynamics are mixed with the feature. In contrast, when a video displays a homogenous motion, it is more advantageous to watch a small difference in a long sequence to filter out redundancies.

This mechanism is accomplished promptly by the compressed-domain features in the network. Furthermore, the feature is generated with examining at the region of interest associated with a question to exploit inter-modal correlation. We illustrate the details in the next subsection.

3.1.1 Motion vector (MV)-based feature warping

In this subsection, we explain how to create a motion feature $f_m^t \in \mathbb{R}^{d_h \times d_w \times d_c}$. Because there is neither an appearance feature nor motion feature in the adjacent frames, an MV $mv^t \in \mathbb{R}^{h \times w \times 2}$ is used for estimating the appearance feature f_s^{t+1} in the adjacent frame by warping the current appearance feature f_s^t. This scheme significantly reduces computational complexity because it avoids reconstructing all the frames and applies feature extraction by CNNs individually. The chunk of the motion-estimated features around the current time t can be generated with arbitrary reference frames as in [38]. Therefore, f_s^{t+n} is generated by displacing the pixels of the current feature with a block-based motion estimation as follows:

$$f_s^{t+n}(u) = f_s^{t+n-1}(u + \frac{1}{r} mv^{t+n-1}(\frac{u}{r})), \tag{1}$$

where n is the number of adjacent features, r is a scaling factor calculated by h/d_h, and u is a spatial coordinate of features. This approach can keep the dimension of the appearance feature to help attention to a question and avoid repeated extraction whereas the previous studies use the limited (1-D) feature, directly extracted from a FC layer of a pretrained network.

The method in [13] recognizes a motion using low- and high-temporal resolution pathways. A high-temporal resolution pathway can capture local temporal changes in the video. Motivated by [13], we express the motion to temporally high-resolution features. The purpose of the following equation is to combine information on the temporal movement of appearances in adjacent frames as

$$f_m^t = ReLU(Conv([f_s^t, ..., f_s^{t+n}])), \tag{2}$$

where [.] is the concatenate operation in the channel axis and $Conv$ is a 1×1 convolution layer with stride 1, producing a channel dimension of feature $d_c \times n$ to d_c. As in [13], f_m^t goes through no temporal downsampling before the mixture in the last convolution layer in Eq.(2) but is slightly modified to increase the number of spatial channels using a pretrained CNN on each n temporal features.

3.1.2 Question guided attention

Spatial attention with a question was proposed initially to capture more relevant objects in a frame on a visual question answering task [42, 8]. We extend this scheme to observe which regions are attended in both motion and appearance features based upon the question feature.

Question feature map For the question-guided attention, we create a question feature map $E_q \in \mathbb{R}^{(d_h \times d_w) \times 1}$, starting with a word feature vector from an LSTM encoder as

$$E_q = W_1 \sum_{j=1}^{N_w} (f_w^j)^T, \tag{3}$$

where $W_1 \in \mathbb{R}^{(d_h \times d_w) \times d_r}$ are learnable parameters used for projection on the same space with f_s and f_m. We select an LSTM encoder because it requires lower time complexity compared with other encoders ($e.g.$, BERT [11]), showing suitable performance in experiments.

Attention map A_s^t and A_m^t refer to the attention maps formed by a question, focusing on the relevant regions and movements in the current frame and the adjacent frames. They are generated from the corresponding feature maps and the question, given as

$$Z_s^t = W_4 \tanh\left(W_2 E_q + W_3 f_s^t + b_1\right) + b_2, \tag{4}$$

$$A_s^t = \frac{exp(Z_s^t)}{\sum_u exp(Z_s^t(u))}, \tag{5}$$

where the transform matrices $W_2 \in \mathbb{R}^{1 \times d_s}$, $W_3 \in \mathbb{R}^{d_c \times d_s}$ and $W_4 \in \mathbb{R}^{d_s \times d_c}$ and offsets $b_1 \in \mathbb{R}^{d_s}$ and $b_2 \in \mathbb{R}^{d_c}$ are learnable parameters. d_s is the hidden size. In implementation, we reshape $f_s^t \in \mathbb{R}^{d_h \times d_w \times d_c}$ to $f_s^t \in \mathbb{R}^{(d_h \times d_w) \times d_c}$ in Eq.(4) to apply attention on each channel in f_s^t.

Then, A_m^t is computed as same in Eq.(4) and Eq.(5). A_a^t and A_m^t share the same learnable parameters for activations aligned more closely to the similar locations of objects.

Finally, the spatially activated areas G_s^t and spatially activated movements G_m^t are designated using the guided local attention and video features. G_s^t and G_m^t are given as $f_s^t \odot A_s^t$ and $f_m^t \odot A_m^t$, respectively. $\odot$ is an operation of element-wise product. The aim of G_s^t and G_m^t is to focus on specific parts and behaviors on the relevant objects to answer given a question in the video frames.

3.1.3 Motion-appearance aggregation (MA+) feature

When the scene changes in adjacent frames or objects appear or disappear based on time t frames, the motion's expression accuracy is naturally reduced if we express motion using these adjacent frames. If motion feature by 3D-CNN, such as I3D[6] or C3D[34], is used an existing model [41, 14, 12], it is inherently vulnerable because of limited representation capability to describe diverse contexts in dynamic scenes. Therefore, in this subsection, we explain how to generate an MA+ feature vector l_v^t to overcome those problems and closely synchronize motion and appearance features at time t. When the motion and appearance features are combined, we use a motion-control parameter α^t to control the degree to which the motion is reliable at the time. The learnable parameter is used for adjusting the temporal changes based on residual information, given as

$$\alpha^t = enc_r(R_s^t), \tag{6}$$

where enc_r is a residue feature extractor consists of a pooling operation and two FC layers and a following sigmoid function, as depicted in Fig. 3. R_s^t is the residue.

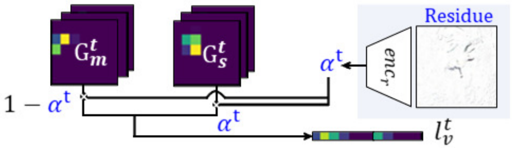

Figure 3. Residue feature extractor and a motion-control parameter to create MA+ feature l_v^t.

In video compression, the residue is given after the prediction. The amount of the residue becomes larger as more temporal changes occur in RGB pixels. Consequently, in Eq.(6), f_m^t would be generated inappropriately due to inaccurate MVs, which may degrade performance.

Therefore, the proposed algorithm produces a small weighting factor to the feature vector to represent temporal changes. Mathematically, the video feature $l_v^t \in \mathbb{R}^{d_h \times d_w \times d_c}$ is computed as follows:

$$l_\alpha^t = \alpha^t W_5 G_s^t(u) + (1-\alpha^t) W_6 G_m^t(u), \tag{7}$$

where all the matrices W are the learnable parameters. According to Eq.(7), if there are abrupt temporal changes between scenes and α^t goes nearly to 1, l_v^t is derived mostly from $G_s^t(u)$. This implies the network examines an input video almost frame-by-frame at the time. In contrast, when there is a slight motion and α^t approaches 0, the network considers only the locally temporal changes in $G_m^t(u)$. This mechanisms can synchronize a motion feature and adaptively determine the amounts for the temporal changes.

We then, use a 1×1 convolutional layer with stride 1, which converts dimension d_c to $d_c/8$ to reduce the channel dimension, and, apply an FC layer using reshape 1 dimension vector as follows:

$$l_v^t = FC(ReLU(Conv(l_\alpha^t))), \tag{8}$$

where $l_v^t \in \mathbb{R}^{d_s}$. It is an 1-D vector but produced by coupling all the appearance, motion, and language features.

3.1.4 Multi-modal fusion and decision

For the decision, we use a logit output vector O_s as

$$O_s = W_7 \sum_{t=1}^{N_v} l_v^t + W_8 \sum_{j=1}^{N_w} f_w(j), \tag{9}$$

where W_7 and W_8 are the learnable parameters, and N_w and N_v are the numbers of input words and video frames, respectively. The decision layer consists of two FC layers. The two layers have 1,024 and 1,000 dimensional outputs when the number of answer sets is 1,000. The answer is obtained by maximizing the softmax function O_s.

3.2. VQAC-integration network

The VQAC-integration network is proposed to reflect global and local information jointly. The previous video QA works have exploited global visual features over an entire video and global question features in a sentence to improve performance using external memory modules. Despite its time-efficiency, the VQAC-baseline performance can be further improved with the global features provided by the existing video QA modules in addition to the local features from the VQAC-baseline network.

In the straightforward integration, the network might combine the feature O_s from the VQAC-baseline and the global features O_g only in the decision network. However, the performance can be limited because the two features have not been jointly created. The VQAC-integration network also attempts to solve this problem. Furthermore, although the VQAC network borrows global memory structures in state-of-the-art networks, it does not use 3D-CNN to extract motion features.

3.2.1 Architecture of VQAC-integration network

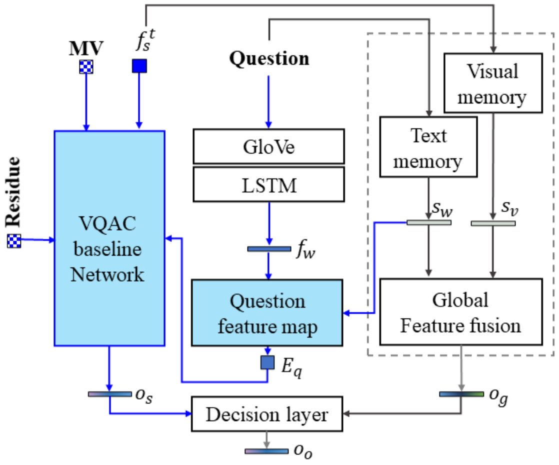

Figure 4. VQAC-integration network architecture incorporated into the existing memory-based architecture. The global visual and question features are further used to improve the performance and fused in the decoder to answer the question.

We illustrate the block diagram of the VQAC-integration network in Fig.4. The network architecture is designed to examine a video and question along with global dependency. The VQAC-integration network obtains the memories S_v and S_w to extract global video and question features, respectively. In experiments, heterogeneous memory [12] and motion-appearance co-memory [14] are used, but the baseline network can also be applied to any memory models if global features are offered.

The network first obtains global question and video features using existing architectures. It then generates an enhanced question feature map and reviews locally spatial and temporal information for improved question-guided attention. The question feature map is further enhanced by reading the global question feature S_w from [12] and integrating it into Eq.(3), given as

$$E_q = W_1 \sum_{j=1}^{N_w} f_w^j + W_9 \sum_{j=1}^{N_w} S_w^j, \tag{10}$$

where W_9 is a learnable parameter.

The quality of the question feature map is improved by reading the global features with the encoded feature word-by-word. Later, E_q is used for guiding which parts of the video frame should be activated. E_q used in Eq.(4) is replaced with the improved one.

Multimodal fusion and decision: We compute the global output vector O_g:

$$O_g = W_{10} \sum_{j=1}^{N_w} S_w(j) + W_{11} \sum_{j=1}^{N_v} S_v(j). \tag{11}$$

For the decision, O_g and O_s are concatenated and decoded to produce the logit O_o for multi-modal fusion of the global features and local features as depicted in Fig. 4. The decision layer has the same architecture in Fig. 2.

4. Experimental Results

Implementation details We implemented the proposed network using Pytorch [24] with an NVIDIA Quadro RTX 6000. In the network, f_s^t has a $28 \times 28 \times 512$ dimension, extracted at the end of the layer 2 of the Resnet152 [16]. Glove 300D [26] is used for question word embedding. For a codec, we use an H.264/AVC decoder [37] to decompress bitstreams and obtain MVs and residue. Whereas the size of a macroblock is 16×16, the partition of sub-blocks is set to four 8×8. We set d_s and n to 512 and 2, respectively.

Training details We used the cross-entropy loss function [28] in training and performed backpropagation using the Adam optimizer [19].

Datasets We conducted performance comparisons using MSR-VTT QA and MSVD QA [41] datasets. MSVD QA has 1,970 video clips and 50,505 QA sets. MSR-VTT QA has 10,000 video clips and 243,680 QA sets. Each dataset is divided into training, validation, and testing sets as in [41, 43, 12]. They are widely used for VideoQA to quantitatively evaluate the performance because they also contain long, high-fidelity sentences.

Measurement metric Top-1 accuracy compares the correct answer with the predicted answer corresponding to the highest probability. Top-k accuracy is also considered. For a question such as "Who is walking down a path?," the semantically reasonable answer might be one of "Man," "Someone," and "Human," although the correct answer was "Person." Therefore, we use the Mean Rank (MR) and Mean Rank Reciprocal (MRR) [35], and Wu-Palmer Similarity (WUPS) scores [39] to compare the accuracies. MR is calculated as the rank in the query. MRR is the reciprocal value of MR. WUPS measures the semantic similarity.

4.1. Performance evaluation and analysis

Table 1. Performance comparisons using the top-1 accuracy

		Performance (%) in MSVD QA			
Method	MEM	What (8149)	Who (4,552)	Others (456)	All (13,157)
E-VQA [41]	-	9.7	42.2	**80.5**	23.4
DLAN [47]	-	21.1	46.0	79.8	31.7
AMU [41]	-	20.6	47.5	80.3	32.0
ST-VQA [18]	-	18.1	50.0	79.0	31.2
VQAC(Base)	-	13.4	**55.6**	77.9	31.5
CO-MEM [14]	✓	19.6	48.7	77.6	31.7
HME [12]	✓	22.4	50.1	70.9	33.7
VQAC(CO)	✓	22.9	50.8	74.3	34.3
VQAC(HME)	✓	**26.9**	**53.6**	68.5	**37.8**

		Performance (%) in MSR-VTT QA			
Method	MEM	What (49,869)	Who (20,385)	Others (2,567)	All (72,821)
E-VQA [41]	-	18.9	38.7	74.8	26.4
DLAN [47]	-	25.4	42.8	73.8	32.0
AMU [41]	-	26.2	43.0	73.3	32.5
ST-VQA [18]	-	24.5	41.2	73.4	30.9
VQAC(Base)	-	24.5	43.3	73.8	31.5
CO-MEM [14]	✓	25.4	43.5	70.3	32.0
HME [12]	✓	26.5	43.6	75.5	33.0
VQAC(CO)	✓	27.2	44.1	75.9	33.6
VQAC(HME)	✓	**29.1**	**46.5**	**77.2**	**35.7**

Compared methods We conduct the performance evaluation of the proposed algorithm compared with the recent VideoQA algorithms DLAN [47], ST-VQA [18], E-VQA [41], AMU [41], CO-MEM [14], and HME [12]. The VQAC-baseline network is referred to as VQAC(Base). We also use external global memories VQAC(HME) and VQAC(CO), where heterogeneous Memory [12] and motion-appearance co-memory [14] were used for global features, respectively.

Quantitative performance analysis Table 1 illustrates the accuracies of the different question types of MSVD and MSR-VTT QA. Five QA algorithms including VQAC(Base) from the top use no external memory modules to preserve global features, whereas the subsequent four algorithms use memory modules. In the comparisons, VQAC(HME) presents the highest average performance among the compared algorithms. Both in MSVD QA and MSR-VTT QA, VQAC(HME) and VQAC(CO) improve the performance by approximately 1.7%∼4.1% over the original HME and CO-MEM, respectively. This result implies that VQAC-baseline improves performance.

Table 2 presents the performance comparisons using different metrics such as the top-10, MR, MRR, and WUPS scores. We also choose HME and AMU for the comparisons because the codes are available. E-MN is also reported in [41]. VQAC(HME) provides the highest performance among the previous algorithms for all metrics. For instance, the proposed algorithm yields approximately 5.6% higher top-10 accuracy in MSVD QA and approximately 3.2% in MSR-VTT QA than HME. The MR value represents the ranking of a predicted answer, so a lower value indicates higher performance.For WUPS scores, VQAC(HME) exhibits significantly improved performance in WUPS 0.0 and comparable performance in WUPS 0.9.

Table 2. Performance comparisons in MSVD QA and MSR-VTT QA using the top-10, MR, MRR, and WUPS accuracies

Method	Performance in MSVD QA dataset				
	Top-10	MR	MRR	WUPS 0.9	WUPS 0.0
E-MN[41]	57.7 %	5.19	0.41	35.7 %	70.0 %
AMU[41]	65.5 %	4.50	0.46	38.9 %	70.0 %
HME[12]	64.9 %	4.48	0.46	41.2 %	72.8 %
VQAC(HME)	**70.5** %	**3.90**	**0.51**	**45.0** %	**72.9** %
Method	Performance (%) in MSR-VTT QA dataset				
	Top-10	MR	MRR	WUPS 0.9	WUPS 0.0
E-MN[41]	60.0 %	5.00	0.42	35.8 %	65.5 %
AMU[41]	62.1 %	4.83	0.43	35.9 %	66.2 %
HME[12]	64.9 %	4.50	0.46	40.7 %	68.3 %
VQAC(HME)	**68.1** %	**4.18**	**0.49**	**42.4** %	**69.2** %

Time complexity measurement We measure the inference time in MSVD QA dataset, in which the resolution of the video clips is 512 × 512 and illustrate the results in Table 3. For the comparisons, we choose AMU [41] and HME [12] because their source codes are available and the time complexity can be measured in the same platform.

Table 3. Measurement time complexity (*min*) in MSVD QA dataset. The total time comprises the measurement time in decompressing video frames (Dec.) and extracting the corresponding features (Ext.) and that in executing the network models (Exe.).

Module	Dec. & Ext. Time			Model Exe. Time		Total
	f_s (T_I)	f_m (T_M)	3D-CNN (T_p)	s_v, s_w	Others	(*min*)
Time	4.0	0.7	7.4			
AMU	✓	-	✓	-	0.3	11.7
HME	✓	-	✓	0.3	0.2	11.9
VQAC(HME)	✓	✓	-	0.3	0.5	5.5
VQAC(Base)	✓	✓	-	-	0.2	4.9

In Table 3, we consider the measurement time in decompressing video frames (Dec) and extracting the corresponding features (Ext) to prepare the features. Once the features are available, the execution time (Exe) to operate the model is added. Thus, the total time is the summation of Ext, Dec, and Exe. The time is measured in on average for every 1,000 videos. As depicted in Table 3, VQAC(Base) is the fastest at approximately 4.9 *min*, and VQAC(HME) is the next at approximately 5.5 *min*. The measurement time is only around 46.2 % that of HME (i.e. 11.9 *min*).

For more details, we profiled the measurement time. f_s is used as an appearance feature. It takes 3.2 *min* to decompress 20 I-frames from the bit-streams and 0.8 *min* to extract the features from a pretrained CNN in MSVD QA dataset. It is used for all the compared methods. However, the main difference is computing time in the motion feature. VQAC(Base) and VQAC(HME) take 0.4 *min* for retrieving MVs from the bit-streams and 0.3 *min* for subsequent processes to generate motion features, thus providing 0.7 *min*. In contrast, HME and AMU require substantially more time due to 3D-CNN. 3D-CNN uses 15 additional P-frames, so it requires more decompression time approximately at 6.4 *min*. It consumes 7.4 *min* in total, by considering feature extractions in 3D-CNN, which takes 1.0 *min*.

Although VQAC(HME) uses global features, it takes less time than the other compared algorithms. In Exe, though the execution time for s_v and s_w required for read/write operations to memories is dominant, it does not occupy a large portion in the total time. "Others" include the measurement time required during attention, fusion, etc.

Qualitative performance evaluation In Fig. 5, we visualize Video QA examples in MSVD dataset. The examples in the left column show the answers related to motion. For example, when a question "What are two man doing?" is given in the first row, VQAC(HME) produces the correct answer "Fight," but HME answers "Stand." It is shown that the VQAC(HME) successfully follows the subject.

4.2. Ablation study

Ablation tests are conducted in the following condition. We turn off each of the core modules and compare the performance versus the VQAC or VQAC(HME). Results represented with [w/H] is compared with VQAC(HME). Otherwise, they are compared with VQAC(Base).

- [w/o MV] is tested to see the changes w/o features using an MV. Only the appearance is used for l_v^t in Eq.(7).

Table 4. Results of the ablation tests by turning off each module to see the performance changes. w/H means a test in VQCA(HME).We also conduct bootstrap sampling five times and report μ and σ on second lien.

Test	[w/o MV]	[w/o R]	[w/ DMC]	[w/ SF]	[w/H + w/o MV]	[w/H + w/o R]	[w/ H + w/o s_w]	[w/ H + w/ 3D]
(%)	30.3(▽ 1.1)	31.0 (▽ 0.4)	30.7 (▽ 0.7)	31.2 (▽ 0.2)	34.2 (▽ 3.5)	36.5 (▽ 1.2)	35.4 (▽ 2.3)	36.0 (▽ 1.7)
$\mu(\sigma)$	29.3 (0.03)	30.3 (0.58)	30.0 (0.06)	30.9 (0.24)	30.9 (0.16)	35.5 (0.08)	33.3 (0.06)	34.6 (0.04)

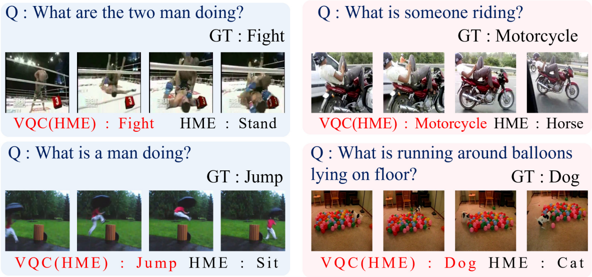
Q : What are the two man doing?
GT : Fight
VQC(HME) : Fight HME : Stand
Q : What is someone riding?
GT : Motorcycle
VQC(HME) : Motorcycle HME : Horse
Q : What is a man doing?
GT : Jump
VQC(HME) : Jump HME : Sit
Q : What is running around balloons lying on floor?
GT : Dog
VQC(HME) : Dog HME : Cat

Figure 5. Qualitative performance evaluation in several data samples of MSVD dataset.

- [w/o R] is tested to see the effect of residue. α^t is set to 0.5, so enc_r becomes inactive. The appearance and motion play equal roles in Eq.(7).

- [w/ 3D] and [w/ SF] are tested when f_m^t in Eq.(2) is replaced with C3D [34] and the slow-fast [13], respectively.

- [w/ DMC] is tested when the compressed-domain features develop for the other CV task such as action recognition [29] are used instead of f_m^t.

- [w/ H + w/o s_w] is tested when the global question feature s_w in VQCA(HME) is removed in Eq.(10). E_q does not consider s_w.

Table 4 illustrates the results of the ablation tests. In [w/o MV] and [w/H + w/o MV], we observe 1.1% and 3.5% performance drops at approximately, respectively, in the top-1 performance. This result indicates the efficiency of the proposed compressed-domain features. Furthermore, we observe some drops in [w/ SF] though the motion feature is generated from [13]. [w/ DMC] presents a 0.7% performance drop. This result shows that the compressed-domain feature [29] for action recognition is not suitable for video QA because it has no consideration of the multimodality (language). In fact, action recognition needed to discern only few representative motions in a video. In [w/o R] and [w/H+w/o R], the performance drops by approximately 0.4% and 1.2%, respectively. [w/ H + w/o s_w] presents a 2.3% performance drop. These results confirm that our enhanced E_q using global feature functions efficiently. Instead of creating motion features using motion vectors, [w/ H + w/ 3D] produces a 1.7% drop in performance and increases complexity. It is more efficient to create motion features using the proposed algorithm for both the performance and speed.

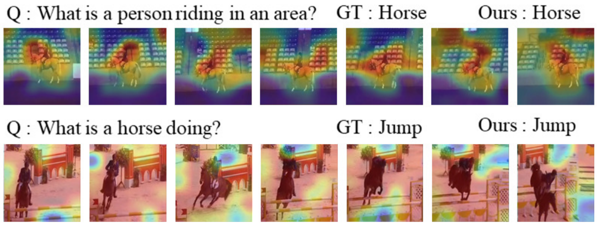
Q : What is a person riding in an area?
GT : Horse
Ours : Horse
Q : What is a horse doing?
GT : Jump
Ours : Jump

Figure 6. Visualization of the activation map G_s^{t+} in a several data samples in MSVD dataset.

Visualization of the question guided attention In Fig. 6, we visualize some examples with G_s^{t+}. The red region is the importance of the attention, whereas the blue region is the opposite. The examples in the first row illustrate answers related to appearance. For example, when a question, "What is a person riding in an area?," is given, the proposed model makes a correct answer "Horse." The attention successfully follows the movements of the horse. The second example illustrates answers related to motion. When a question in the fourth row, "What is a horse doing?," is given, the proposed network produces the correct answer "Jump" by accurately recognizing the subject.

5. Conclusion

In this paper, a deep neural network that exploits compressed-domain features was proposed to yield video QA features. The proposed network considered inter-modal correlation and computational complexity. The proposed network provides the baseline framework but also the integrated to the state-of-the-art networks for improved performance.

Acknowledgement

This work was supported by NCSOFT, in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(No.NRF-2019R1C1C1010249), and in part by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-2020-0-01460) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

References

[1] Jurandy Almeida, Neucimar J. Leite, and Ricardo da S Torres. Online video summarization on compressed domain. *Journal of Visual Communication and Image Representation*, 24(6):729–738, 2013.

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

[3] R Venkatesh Babu, KR Ramakrishnan, and SH Srinivasan. Video object segmentation: a compressed domain approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):462–474, 2004.

[4] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, 2017.

[5] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in Neural Information Processing Systems*, pages 839–850, 2019.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition: a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[7] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. Abc-cnn: An attention based convolutional neural network for visual question answering, 2015. arXiv preprint arXiv:1511.05960.

[8] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*, 2015.

[9] Shaoxiang Chen and Yu-Gang Jiang. Motion guided spatial attention for video captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8191–8198, 2019.

[10] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*, 2019.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *CVPR*, 2019.

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019.

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

[15] Peng Gao, Hongsheng Li, Shuang Li, Pan Lu, Yikang Li, Steven C.H. Hoi, and Xiaogang Wang. Question-guided hybrid convolution for visual question answering. In *ECCV*, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitsky, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

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

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. arXiv preprint arXiv:1412.6980.

[20] Se-Ho Lee, Je-WonKang, and Chang-Su Kim. Compressed domain video saliency detection using global and local spatiotemporal features. *Journal of Visual Communication and Image Representation*, 35(1):169–183, 2016.

[21] Shuang Li, Tong Xiao, Hongsheng Li, Wei Yang, and Xiaogang Wang. Identity-aware textual-visual matching with latent co-attention. In *ICCV*, 2017.

[22] 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.

[23] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*, 2014.

[24] 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.

[25] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

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

[27] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension, 2016. arXiv preprint arXiv:1611.01603.

[28] John Shore and Rodney Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on information theory*, 26(1):26–37, 1980.

[29] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *CVPR*, 2019.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Orachat Sukmarg and Kamisetty R Rao. Fast object detection and segmentation in mpeg compressed domain. In *2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No. 00CH37119)*, volume 3, pages 364–368. IEEE, 2000.

[32] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[33] Kunio Takaya. Detection and segmentation of moving objects in video. In *2006 Canadian Conference on Electrical and Computer Engineering*, pages 2069–2073. IEEE, 2006.

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

[35] Ellen M. Voorhees. The trec-8 question answering track report. In *Trec*, 2019.

[36] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

[37] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

[38] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Compressed video action recognition. In *CVPR*, 2018.

[39] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Association for Computational Linguistics*, 1994.

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

[41] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM international conference on Multimedia*, 2017.

[42] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016.

[43] Tianhao Yang, Zheng-Jun Zha, Hongtao Xie, Meng Wang, and Hanwang Zhang. Question-aware tube-switch network for video question answering. In *ACM international conference on Multimedia*, 2019.

[44] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. Video question answering via attribute-augmented attention network learning. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017.

[45] Zheng-Jun Zha, Jiawei Liu, Tianhao Yang, and Yongdong Zhang. Spatiotemporal-textual co-attention network for video question answering. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 15(2):1–18, 2019.

[46] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, 2016.

[47] Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, Yueting Zhuang, Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, and Yueting Zhuang. Video question answering via hierarchical spatio-temporal attention networks. In *IJCAI*, 2017.