Beyond RNNs: Positional Self-Attention with Co-Attention for Video Question Answering

Xiangpeng Li[1], Jingkuan Song[1]*, Lianli Gao[1], Xianglong Liu[2]
Wenbing Huang[3], Xiangnan He[4], Chuang Gan[5]
[1]Center for Future Media and School of Computer Science and Engineering,
University of Electronic Science and Technology of China,
[2]Beihang University, [3]Tecent AI lab, [4]National University of Singapore, [5]MIT-IBM Watson AI Lab
{xiangpengli.cs, jingkuan.song}@gmail.com, lianli.gao@uestc.edu.cn, xlliu@nlsde.buaa.edu.cn,
hwenbing@126.com, ganchuang1990@gmail.com, xiangnanhe@gmail.com

Abstract

Most of the recent progresses on visual question answering are based on recurrent neural networks (RNNs) with attention. Despite the success, these models are often time-consuming and having difficulties in modeling long range dependencies due to the sequential nature of RNNs. We propose a new architecture, Positional Self-Attention with Co-attention (PSAC), which does not require RNNs for video question answering. Specifically, inspired by the success of self-attention in machine translation task, we propose a Positional Self-Attention to calculate the response at each position by attending to all positions within the same sequence, and then add representations of absolute positions. Therefore, PSAC can exploit the global dependencies of question and temporal information in the video, and make the process of question and video encoding executed in parallel. Furthermore, in addition to attending to the video features relevant to the given questions (i.e., video attention), we utilize the co-attention mechanism by simultaneously modeling "what words to listen to" (question attention). To the best of our knowledge, this is the first work of replacing RNNs with self-attention for the task of visual question answering. Experimental results of four tasks on the benchmark dataset show that our model significantly outperforms the state-of-the-art on three tasks and attains comparable result on the Count task. Our model requires less computation time and achieves better performance compared with the RNNs-based methods. Additional ablation study demonstrates the effect of each component of our proposed model.

1 Introduction

In recent years, breaking the semantic gap of vision and language is a hot topic in artificial intelligence. A lot of research achievements are made centering on computer vision (CV) (Yang et al. 2018; Gao et al. 2017) and natural language processing (NLP), especially for the task of visual question-answering (VQA) (Gao et al. 2018a; Yang et al. 2016; Yu et al. 2017; Anderson et al. 2017; Palangi et al. 2018; Song et al. 2018). It is still a critical challenge towards machine intelligence, but its achievement can be beneficial for various real-life applications.

In general, we can divide the VQA task into two categories: image question-answering (Kim, Jun, and Zhang 2018a; Yang et al. 2016; Xiong, Merity, and Socher 2016; Kim, Jun, and Zhang 2018b; Gao et al. 2018b) and video question-answering (Jang et al. 2017; Gao et al. 2018a; Kim et al. 2017; Zeng et al. 2017). Compared with image-based question-answering, video question-answering is more challenging. Given a question, a video QA model is required to locate and explore a sequence of frames to firstly recognize a set of specific objects and activities and secondly identify the relationship between objects and the relationship between objects and actions. In artificial intelligence, the above two steps are usually conducted independently.

Previous visual question-answering models (Kim, Jun, and Zhang 2018a; Yang et al. 2016; Xiong, Merity, and Socher 2016; Jang et al. 2017; Gao et al. 2018a) are primarily based on RNNs, especially Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which have good performance on modeling sequential data and potential to solve the gradient vanishing problem. For instance, Jang *et al.* (Jang et al. 2017) proposed a two-staged LSTM to encode both video frames and question information for answer prediction. Gao *et al.* (Gao et al. 2018a) extended a dynamic memory network to form a new motion-appearance co-memory network. In these models, LSTM is an essential part for catching data dependencies. However, experimental results (Vaswani et al. 2017) demonstrated that LSTM has weakness in modeling the long range dependencies and it cannot ensure data encoding to be conducted in parallel. Therefore, the training may be time-consuming, especially for encoding long sequential data (e.g., a text paragraph). To solve this problem, Vaswani *et al.* (Vaswani et al. 2017) proposed an attention mechanism, named Self-Attention, to replace traditional RNNs for machine translation. The proposed attention mechanism incorporated external information to assist a model to assign different weights to data items based on their importance. Experimental results demonstrate its effective role in catching long range dependencies, and it reaches a new state-of-the-art performance for machine transition.

In this work, we introduce a simple yet interpretable network named Positional Self-Attention with Co-Attention (PSAC) for video question-answering. It consists of two positional self-attention blocks to replace LSTM for modeling

*Jingkuan Song is the corresponding author.

data dependencies, and a video-question co-attention block to simultaneously attend both visual and textual information for improving answer prediction. We summarize the contributions of our model as below:

- To better exploit the global dependencies of the sequential input (i.e., a video and a question), and make video and question encoding processes conducted in parallel, we present a novel positional self-attention mechanism. To our knowledge, this is the fist try in the visual QA task, where a traditional RNN is replaced by a self-attention to boost the performance and training efficiency.
- We propose a new co-attention mechanism (i.e., video-to-question and question-to-video attention) to enable our model attending to both relevant and important visual and textual features, which removes the irrelevant video and textual information to guarantee the generation of accurate answers.
- We conduct experiments on the large-scale TGIF-QA dataset and the experimental results on four tasks demonstrate the efficiency and effectiveness of our proposed Positional Self-Attention with Co-Attention architecture.

2 Related Work

In this section, we discuss related work of our method. Specifically, we introduce relevant works in two aspects: visual question-answering (image QA and video QA) and self-attention mechanism.

2.1 Visual Questioning-Answering

In image-based question-answering, existing question-answering methods are mainly focusing on using a LSTM network to encode question sequence and fusing question representation and image feature together to predict an answer. Yang *et al.* (Yang et al. 2016) modified the basic model and proposed Stacked Attention Network (SAN) which uses a multiple-layer SAN. In SAN, it queries an image multiple times to infer the answer progressively. Nam *et al.* (Nam, Ha, and Kim 2017) proposed a dual attention network which attends to specific regions in images and word in text through multiple steps and gathers essential information from both modalities to help predict an answer. Top-down model (Anderson et al. 2017) was proposed by combining bottom-up and top-down attention mechanism that allows attentions to be computed at the level of objects and other salient image regions. Xiong *et al.* (Xiong, Merity, and Socher 2016) introduced dynamic memory network to image question-answering which has a memory component and an attention to assist the prediction of correct answers. For video question-answering, Jang *et al.* (Jang et al. 2017) proposed a spatial-temporal model which gathers the visual information from spatial aspect and motion information from temporal aspect. Motion-appearance dynamic memory network (Gao et al. 2018a) adopted two dynamic memory network to construct a co-memory structure which deals with visual static feature and motion flow feature at the same time. Fusion of multiple features can boost the performance.

2.2 Self-Attention Mechanism

Attention module assigns different weights to different data to allow the model focusing on important data. In recent years, attention mechanism(Xu et al. 2015) has been widely applied in lots of research topics and experimental results have proved the effectiveness of this module. Vaswani *et al.* (Vaswani et al. 2017) modified traditional attention and proposed Self-Attention which calculates the response at a position in a sequence by attending to all positions. Yu *et al.* (Yu et al. 2018) adopted self-attention and convolution to construct a QAnet for reading comprehension, where convolution extracts local interactions and self-attention extracts the global interactions between sequence. Zhang *et al.* (Zhang et al. 2018) proposed the Self-Attention Generative Adversarial Network (SAGAN) in which self-attention attains a better balance between the ability to model dependencies and computational efficiency. Zhou *et al.* (Zhou et al. 2018) employed self-attention mechanism to propose a new model which enables the use of efficient non-recurrent structure during encoding and leads to performance improvements in dense video captioning.

3 Proposed Method

In this section, we first present our positional self-attention with co-attention (PSAC) architecture which is proposed to address the problem of video question answering. The architecture is shown in Fig. 1 and it consists of three key components: Video-based Positional Self-Attention Block (VPSA), Question-based Positional Self-Attention Block (QPSA) and Video-Question Co-Attention Block (VQ-Co). Specifically, both VPSA and QPSA utilize the same positional self-attention mechanism, excepting that VPSA is focused on attending to video frame features, while QPSA is focused on attending to a textual feature, derived by concatenating question word features with character features. Through VPSA and QPSA, frame features and question features are both updated. Next, our VQ-Co block takes the two updated video and question features as inputs and then simultaneously compute attentions for them. Multiple features are fused together for final answer prediction.

2个主部分+一个融合

多个特征融合在一起来回答最后的问题

3.1 Positional Self-Attention Block

To catch better long range dependencies and position information, we propose a positional self-attention block to replace RNNs. Given a query and a set of key-values, attention mechanism calculates a weighted sum of values based on the similarity of query and keys. In positional self-attention model, we regard $\mathbf{F}^{\mathrm{Q}}$, $\mathbf{F}^{K}$ and $\mathbf{F}^{V}$ as query, key and value. And they are projected using fully connected layer over sequential feature $\mathbf{F}$ respectively. Supposed that each feature is represented as $\mathbf{F}^{*} \in \mathbb{R}^{n \times d_k}$ and a scaled dot product attention (SDPA) is defined as below mathematically:

$$SDPA(\mathbf{F}^{\mathrm{Q}}, \mathbf{F}^{K}, \mathbf{F}^{V}) = softmax(\frac{\mathbf{F}^{K}(\mathbf{F}^{Q})^{T}}{\sqrt{d_k}})\mathbf{F}^{V} \quad (1)$$

where n is the length of the sequence, and d_k denotes feature dimension of each projected feature. In order to enable

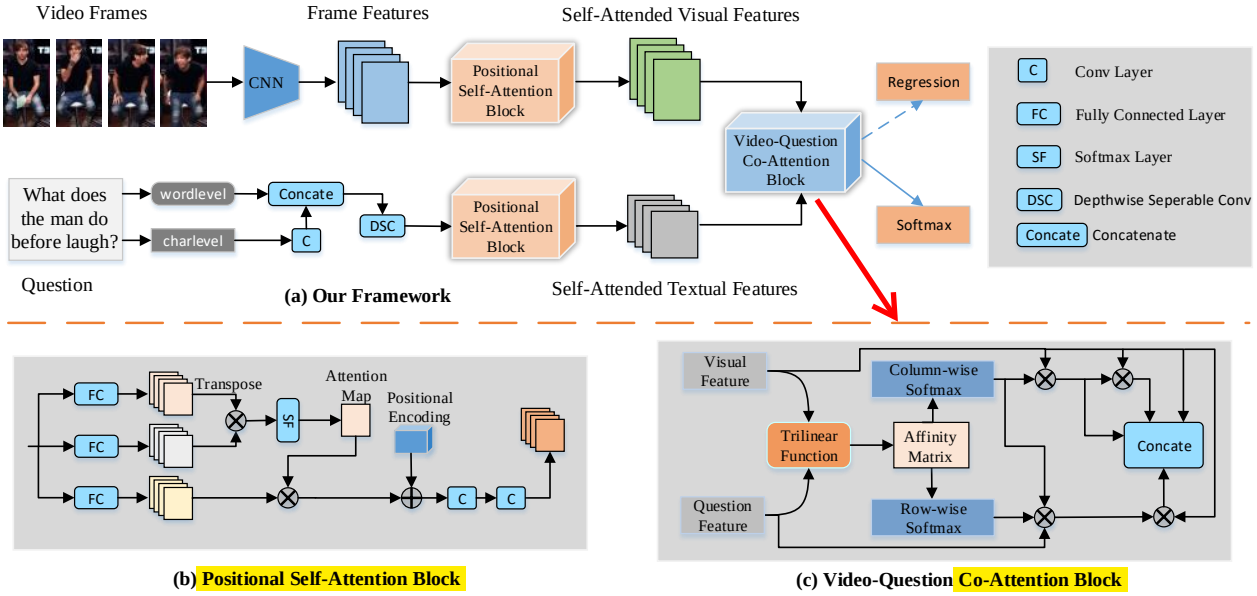

Figure 1: The overview of our proposed framework, Positional Self-Attention wiht Co-attention for Video Question-Answering. There are three key components: Video-based Positional Self-Attention Block, Question-based Positional Self-Attention Block and Video-Question Co-Attention Block.

the model to jointly attend to information from different representation subspaces at different positions, positional self-attention adopts l-scale dot product attention concurrently. Positional self-attention concatenates the outputs of all scale l dot product attention models and then utilizes a linear layer to project the concatenated feature to a fixed dimensional feature. We can formulate the self-attention calculation process as:

$$\mathbf{J} = concat(h_1, ..., h_l)W_o \tag{2}$$

$$h_i = SDPA(FW_i^Q, FW_i^K, FW_i^V) \tag{3}$$

where W_o and W_i are parameters to be learned. However, transmission loss may occur in self-attention operations. Thus we add a residual connection to $\mathbf{J}$ and then apply a layer normalization. Therefore, the original mapping $\mathbf{J}$ is recast into $\mathbf{O}$.

$$\mathbf{O} = LayerNorm(\mathbf{J} + \mathbf{F}) \tag{4}$$

Compared with traditional RNN networks, such as LSTM, self-attention has ability to ensure computational efficiency and to derive long-rang dependencies, but it ignores the positional information of the sequential input (Gehring et al. 2017). To remedy this weakness, we define a positional matrix $\mathbf{P} \in \mathbb{R}^{n \times d_k}$ to encode the sequence geometric position information about $\mathbf{F}$. $\mathbf{P}$ is computed by using sine and cosine functions at different positions:

$$\mathbf{p}_{pos,2j} = sin(pos/10000^{2j/d_k}) \tag{5}$$

$$\mathbf{p}_{pos,2j+1} = cos(pos/10000^{2j/d_k}) \tag{6}$$

where pos is the position and j is the dimension. With $\mathbf{P} \in \mathbb{R}^{n \times d_k}$, we add it to the attended feature followed by two convolutional layers with a ReLU activation function. Therefore, the final self-attended feature is defined as:

$$\mathbf{O}_f = ReLU((\mathbf{O} + \mathbf{P})W_1 + b_1)W_2 + b_2 \tag{7}$$

where W_1 and W_2 all denote the convolutional operations and b_* is the bias. For simplicity, we define our positional self-attention mechanism as:

$$\mathbf{F}_o = PositionalSelfAttention(\mathbf{F}) \tag{8}$$

where $\mathbf{F}$ is the sequential input features, and $\mathbf{F}_o$ is the positional self-attended feature.

3.2 Video-based Positional Self-Attention Block

Given a video, we firstly conduct the video pre-processing step by extracting N equal-spaced frames and then applying a pre-trained CNN network to obtain N frame features. Each frame feature's dimension is d_v. After the video pre-processing, we define the extracted video features as $\mathbf{V} \in \mathbb{R}^{N \times d_v}$. Next, we apply the previous defined positional self-attention mechanism to encode the input $\mathbf{V}$.

$$\mathbf{V}_o = PositionalSelfAttention(\mathbf{V}) \tag{9}$$

where $\mathbf{V}_o$ indicates the positional self-attended visual feature, which contains both video long-term structures as well as frame spatial position information.

3.3 Question-based Positional Self-Attention Block

The goal of Question-based Positional Self-Attention Block (QPSA) is to extract the semantic long range dependencies

of the given question Q. The information of the question can be described in two levels: word and character. Given a question Q, we suppose the embedded word and character representations of question are $\mathbf{W} \in \mathbb{R}^{M \times d_w}$ and $\mathbf{C} \in \mathbb{R}^{M \times r \times d_c}$, respectively. Where M denotes the sentence length, r denotes the word length, and d_w and d_c represent the word embedding and character embedding dimensions.

To form a representative question feature, we firstly use a convolutional layer to encode the character $\mathbf{C} \in \mathbb{R}^{M \times r \times d_c}$ and then concatenate the output of the convolutional feature with word-level feature $\mathbf{W} \in \mathbb{R}^{M \times d_w}$. The question embedding process is defined below.

$$\mathbf{Q}_e = Concat(\mathbf{W}, Conv2D(\mathbf{C})) \tag{10}$$

where $\mathbf{Q}_e$ is the combined question feature. In addition, two-layer highway network (Srivastava, Greff, and Schmidhuber 2015) is used after the concatenation of character feature and word feature in language translation task. This is due to that the highway network can solve training difficulties with the model parameters growing. However, the representative ability of concatenation is constrained, thus we need a convolutional layer to further fuse the word level and character level features. Compared with traditional convolutional layer, the depthwise separable convolution (Chollet 2017) has been proved better in parameter efficiency and has better generalization ability. As a result, we adopt a depthwise separable convolutional layer to further encode our question feature $\mathbf{Q}_e$:

$$\mathbf{Q}_d = ReLU((\mathbf{Q}_e W_d + b_d)W_b + b_p) \tag{11}$$

where W_d denotes depthwise convolution parameters, W_b denotes pointwise convolution parameters in the depthwise separable convolution module, and b_* is the bias.

To our knowledge, encoding the long range dependencies of a question is important for extracting useful information cues. In our framework, we use positional self-attention mechanism to exploit the long range dependencies for the given question. After the positional self-attention mechanism, our question $\mathbf{Q}_d$ is mapped to $\mathbf{Q}_o$, which indicates the positional self-attended question features.

$$\mathbf{Q}_o = PositionalSelfAttention(\mathbf{Q}_d) \tag{12}$$

3.4 Video-Question Co-Attention Layer

After the two positional self-attention blocks, we obtain two attended features: $\mathbf{V}_o$ and $\mathbf{Q}_o$, but the last dimension of them are different. In order to conduct the further operations, we firstly project them into a ρ-dimension common space. With the projected $\mathbf{V}_o$ and $\mathbf{Q}_o$, we apply our proposed video-question co-attention layer on them to boost the question answering performance.

Here we generalize a co-attention model for two multi-channel inputs, where $\mathbf{V}_o \in \mathbb{R}^{N \times \rho}$ and $\mathbf{Q}_o \in \mathbb{R}^{M \times \rho}$, and it generates two attention maps. One is used to attend to $\mathbf{V}_o$, and the other is applied to attend to $\mathbf{Q}_o$. To derive the attention maps, we follow previous work (Seo et al. 2016) to construct a similarity matrix, denoted as $\mathbf{S}$, by employing a multi-element function. It integrates $\mathbf{Q}_o$, $\mathbf{V}_o$ and $\mathbf{Q}_o \odot \mathbf{V}_o$ to compute $\mathbf{S}$.

$$\mathbf{S} = W_s[\mathbf{Q}_o, \mathbf{V}_o, \mathbf{Q}_o \odot \mathbf{V}_o] \tag{13}$$

where W_s denotes the parameter that to be trained and $\odot$ means the element-wise multiplication and $\mathbf{S} \in \mathbb{R}^{N \times M}$. With the similarity matrix $\mathbf{S}$, we now use it to obtain two attention maps and the attended vectors in both directions.

Video-to-Question Attention. Video-to-question (V2Q) attention aims to locate which question vectors are most relevant to each self-attended video features. The attention map $\mathbf{S}_q$ is computed by normalizing each row of the $\mathbf{S}$ with a softmax function. Thus we apply the computed attention map $\mathbf{S}_q$ to the question feature $\mathbf{Q}_o$ to obtain the attended question feature $\mathbf{A} = \mathbf{S}_q \cdot \mathbf{Q}_o$.

Question-to-Video Attention. Question-to-video (Q2V) attention aims to find which visual vectors have the highest similarity to one of the question vectors, and are hence critical for predicting answers for questions. To compute the video attention map, we normalize each column of the $\mathbf{S}$ with a softmax function to get $\mathbf{S}_v$. Next, our video attention weight $\mathbf{B}$ is obtained by $\mathbf{B} = \mathbf{S}_q \cdot \mathbf{S}_v^{\mathbf{T}} \cdot \mathbf{V}_o^{\mathbf{T}}$.

Co-attention. To yield the final feature $\mathbf{O}_f$ for answer prediction, we combine the three attended features together, including $\mathbf{V}_o$, $\mathbf{A}$, and $\mathbf{B}$ through the following operation.

$$\mathbf{O}_f = Concat(\mathbf{V}_o, \mathbf{A}, \mathbf{V}_o \odot \mathbf{A}, \mathbf{V}_o \odot \mathbf{B})W_f \tag{14}$$

where W_f is the parameter to be learned.

3.5 Answer Module and Loss Function

For multiple choice task (i.e. Transition and Action), a linear regression function is adopted. It takes final combined feature $\mathbf{O}_f$ as the input to compute a real-valued score for each option, as below.

$$p = W_r^{\mathbf{T}} \mathbf{O}_f \tag{15}$$

where W_r is the parameter to be trained. For this task, we use multi-hinge loss of each question $max(0, 1 - s_p + s_n)$ where s_p and s_n are scores calculated from correct candidates and incorrect candidates.

As for open-ended question (i.e. FrameQA), we use a linear classifier and a softmax function to project $\mathbf{O}_f$ to the answer space.

$$p = softmax(W_x^{\mathbf{T}} \mathbf{O}_f + b_p) \tag{16}$$

where W_x is the parameters to be learned. Like other open-ended question-answering tasks, we use cross entropy loss between predicted answer and groundtruth answer as our loss function.

We also regard Count task as an open-ended task, but it requires a model to predict a number ranging from 0 to 10. Therefore, we define a linear regression function to predict the real-valued number directly.

$$p = W_c^{\mathbf{T}} \mathbf{O}_f \tag{17}$$

where W_c is parameters to be trained. To reduce the gap between the predicted answer and true answer, we use Mean Square Error (MSE) loss to train our count model.

4 Experiments

4.1 Dataset

Following (Gao et al. 2018a), we evaluate our method on TGIF-QA dataset (Jang et al. 2017). It consists of 103,919

Table 1: QA-pairs and GIFs construction of TGIF-QA dataset.

QA pairs	Action		Trans		Count		FrameQA	
	QA-pairs	GIFs	QA-pairs	GIFs	QA-pairs	GIFs	QA-pairs	GIFs
Train	20475	3543	52704	34526	26843	12382	39392	27327
Test	2274	614	6232	4881	3554	2580	13691	10028
Total	22749	3851	58936	38021	30397	13316	53083	36633

question-answer pairs collected from 56,720 animated GIFs. In addition, all QA-pairs are split into four tasks: Action, Transition (Trans.), Count and FrameQA.

- **Action** It is a multi-choice task and each question is attached with 5 options. All questions are asking about a certain action, which has occurred fixed number of times. For example "What does the woman do 3 times?".
- **Transition (Trans.)** It is a multi-choice task and each question has 5 options. Questions are all about transitions of certain states, including facial expression, actions, places and object properties. For instance, "What does the man in sweater do after lay on bed?".
- **Count** This is an open-ended task which counts the number of repetition of a certain action. Even though it is an open-ended question, all the answers are numbers and there are 11 possible answers, ranging from 0 to 10. For example, "How many times does the woman sway hips?".
- **FrameQA** It is an open-ended task and it is similar to image-QA. Each question in this task can be answered from one of the frames in a video. For instance, "What is the color of the woman's hair?".

For each task, the total number of QA-pairs and GIFs as well as the numbers of training/testing are displayed in Tab. 1.

4.2 Evaluation Metrics

For multi-choice tasks, including Action and Trans., we use accuracy as the evaluation metric to evaluate video QA models. For open-ended tasks, the FrameQA also adopts accuracy, while the Count task utilizes the Mean Square Error (MSE) between the predict answer and the ground truth answer to measure the model's performance. Note that for a model, the accuracy should be as higher as better, while the MSE score is exactly opposite.

4.3 Implementation Details

Given a video, we equally select 35 frames. For each frame, we take the output of $pool5$ layer of ResNet-152 (He et al. 2016) as visual features. The dimension of each frame feature is 2048. Given a question, we firstly covert all words to lower cases and then get rid of all punctuations. Next, each question is split by blank space to acquire words, while each word is further spitted to obtain characters. Specifically, each word is transfered to a 300-D feature vector by a pre-trained GloVe (Pennington, Socher, and Manning 2014) and each character is finally embedded into a 64-D vector.

In order to train the model, we employ the Adamax optimizer. For both Count and FrameQA, we set the size of minibatch as 128. For Action and Trans., the size of minibatch is set as 16. For all CNN layers, the dropout rate is 0.2. Following (Vaswani et al. 2017), the number of scaled parallel attention l in both VPSA and QPSA is set as 8.

Table 2: Comparison with the state-of-the-art methods on TGIF-QA dataset. Action, Trans., FrameQA and Count are four tasks in TGIF-QA dataset. R denotes ResNet feature, C denotes C3D feature and F denotes flow CNN.

Model	Action	Trans	FrameQA	Count
Random Chance	20.0	20.0	0.06	20.4
VIS+LSTM(aggr)	46.8	56.9	34.6	5.09
VIS+LSTM(avg)	48.8	34.8	35.0	4.80
VQA-MCB(aggr)	58.9	24.3	25.7	5.17
VQA-MCB(avg)	29.1	33.0	15.5	5.54
Yu *et al.*	56.1	64.0	39.6	5.13
ST(R+C)	60.1	65.7	48.2	4.38
ST-SP(R+C)	57.3	63.7	45.5	4.28
ST-SP-TP(R+C)	57.0	59.6	47.8	4.56
ST-TP(R+C)	60.8	67.1	49.3	4.40
Co-memory(R+F)	68.2	74.3	51.5	**4.10**
PSAC(R)	**70.4**	**76.9**	**55.7**	4.27

4.4 Comparison with State-of-the-art Methods

In this subsection, we introduce several state-of-the-art baseline methods and show the comparisons of our proposed model with baselines. The comparison results are shown in Tab. 2. We have the following observations:

- Random Chance is to select an answer in the answer vocabulary randomly. As expected, it performs the worst compared with the other methods.
- In order to compare with the best image-based VQA methods, including VIS+LSM and VQA-MCB (i.e., the winner of the VQA 2016 challenge), Jang *et al.* proposed two ways to extend them for video question answering task. They are early fusion based approaches including VIS+LSTM (aggr) and VQA-MCB (aggr), and late fusion based methods including VIS+LSTM (avg) and VQA-MCB (avg). For four models, they select one in four frames and then use the pre-trained ResNet-152 model to extract frame features. In addition, the aggr. methods obtain the input visual feature by averaging all frame features, while agv. methods take one frame at a time to predict the answer and then average the accuracy across all frames of all videos. Compared with Random Chance, they perform better but the performance is still quite low. This is because both aggr. and avg. based methods ignores

Table 3: The effect of VPSA. The w/ L VPSA indicates our model with L stacked VPSA layers. The experiments are conducted on the TGIF-QA dataset.

Encoder layer	Action	Trans	FrameQA	Count
w/o VPSA	69.0	76.3	55.5	4.41
w/ 1 VPSA	**70.4**	**76.9**	55.7	**4.27**
w/ 2 VPSA	68.2	76.7	**56.1**	4.32
w/ 3 VPSA	69.2	75.7	55.9	4.40
w/ 4 VPSA	66.4	76.2	55.4	4.38
w/ 5 VPSA	64.6	75.9	55.5	4.36
w/ 6 VPSA	67.3	76.2	55.6	4.44

long-term dependencies in videos.

- Later on, Yu *et al.* proposed a high-level concept word detector to generate a list of concepts words. Next, the detected words are combined through a semantic attention for captioning, retrieval and question answering. The results show that it further improves the performance of video question answering on the TGIF-QA dataset, reaching 56.1% Action, 64.0% Trans, 39.6% FrameQA and 5.13 Count.
- Both ST and its variants (Jang et al. 2017) and CO-memory (Gao et al. 2018a) are video-based methods. Compared with ST and its variants with C3D features and ResNet features as the visual input, Co-memory (R+F) with optical flow features and ResNet features performs better. Specially, Co-memory (R+F) achieves the lowest score on the Count task with 4.10.
- By observing the whole Tab. 2, we reach a conclusion that our model performs best in terms of Action, Trans. and FrameQA tasks. Compared with Co-memory (R+F), our model achieves higher accuracy, with an increase of 2.2% for Action task, 2.6% for Trans., and 4.2% for FrameQA. For Count task, our model performs slightly worse than Co-memory (R+F), 4.27 vs 4.10 respectively. However, in essential both ST-based methods and Co-memory (R+F) are not comparable, because they utilize extra visual features either C3D feature or optical flow feature. Our model only makes use of the ResNet visual feature. Therefore, we believe that the results demonstrate the effectiveness of our proposed framework.

4.5 Ablation Study

Why Video Positional Self-Attention? Here, we conduct several experiments to study the effect of video positional self-attention block (VPSA). Firstly, we remove the VPSA block and directly use original frame features to replace positional self-attended visual features to conduct question answering. The experimental result is shown in Tab. 3. Without VPSA, we find out that the performance of our PSAC model decreases by 1.4% on Action task, 0.6% on Trans., 0.2% on FrameQA and increases MSE 0.14 on Count. Furthermore, we continue to explore the role of L stacked VPSA layers. From Tab. 3, we can see that for Action, Trans., and Count, the best results are obtained when $L = 1$, and for FrameQA, the best results (56.1%) is achieved when $L = 2$. Therefore, we set $L = 1$ in the rest of this paper.

Table 4: Comparison results and time cost (seconds) on visual positional self-attention (VPSA) and Bi-LSTM layer.

model	program	Action	Trans	FrameQA	Count
PSAC	result	70.4	76.9	55.7	4.27
	s/epoch	269	698	122	74
Bi-LSTM	result	67.6	75.1	53.9	4.39
	s/epoch	337	870	143	83

Table 5: The effect of character embedding: without or with word character information.

	Action	Trans	FrameQA	Count
PSAC w/o Char.	66.89	75.74	54.80	4.40
PSAC w/ Char.	**70.36**	**76.89**	**55.67**	**4.27**

In addition, we compare our VPSA to the recurrent layers which are commonly used for mapping one variable-length sequence of features to another sequence with equal length. Here, we adopt the bidirectional LSTM (Bi-LSTM) to replace the VPSA, thus the self-attended visual features are replaced by a set of Bi-LSTM hidden states. The dimension of all hidden states are 1024. The comparison results are shown in Tab. 4. Compared with Bi-LSTM layer, our VPSA brings improvements for all four tasks. In theory, given a n length sequence, where a recurrent layer requires $O(n)$ steps, while VPSA only requires a constant number of sequentially executed operations. Experiments empirically show that for each epoch, PSAC with VPSA respectively takes 269s, 698s, 122s and 74s for training Action, Trans., FrameQA and Count, while PSAC with Bi-LSTM spends 337s, 870s, 143s and 83s. The time difference is small, due to the reason that we are dealing with short videos ($n = 35$). If n is big enough, the time difference can be more obvious.

Why Word Character? In this section, we conduct two experiments: PSAC w/o Char. and PSAC w/ Char. The experimental results are shown in Tab. 5. The results demonstrate the effects of character. Without word character, PSAC decreases the performance by 3.47%, 1.15% and 0.87% on Action, Trans. and FrameQA respectively, and increases the MSE error to 4.40 for Count. The positive impact of word character further proves that a video question answering model could gain benefit by fully exploring the question multi-level information (e.g., word and character level).

Why Video-Question Co-attention? We compare the performance of non-video question co-attention and video question co-attention method in Tab. 6. We see that co-attention outperforms non co-attention method on three tasks, including Action, Trans., and Count, respectively reaching 70.36%, 76.89% and 4.27 of MSE error. For FrameQA, with or without co-attention, their performance are almost the same. Compared with other three tasks Action, Trans., and Count, which require to analyze multiple frames, FrameQA only needs one of the frame to answer the question. Therefore, the nature of FrameQA task makes assigning weights for multiple frames ineffective. In addition,

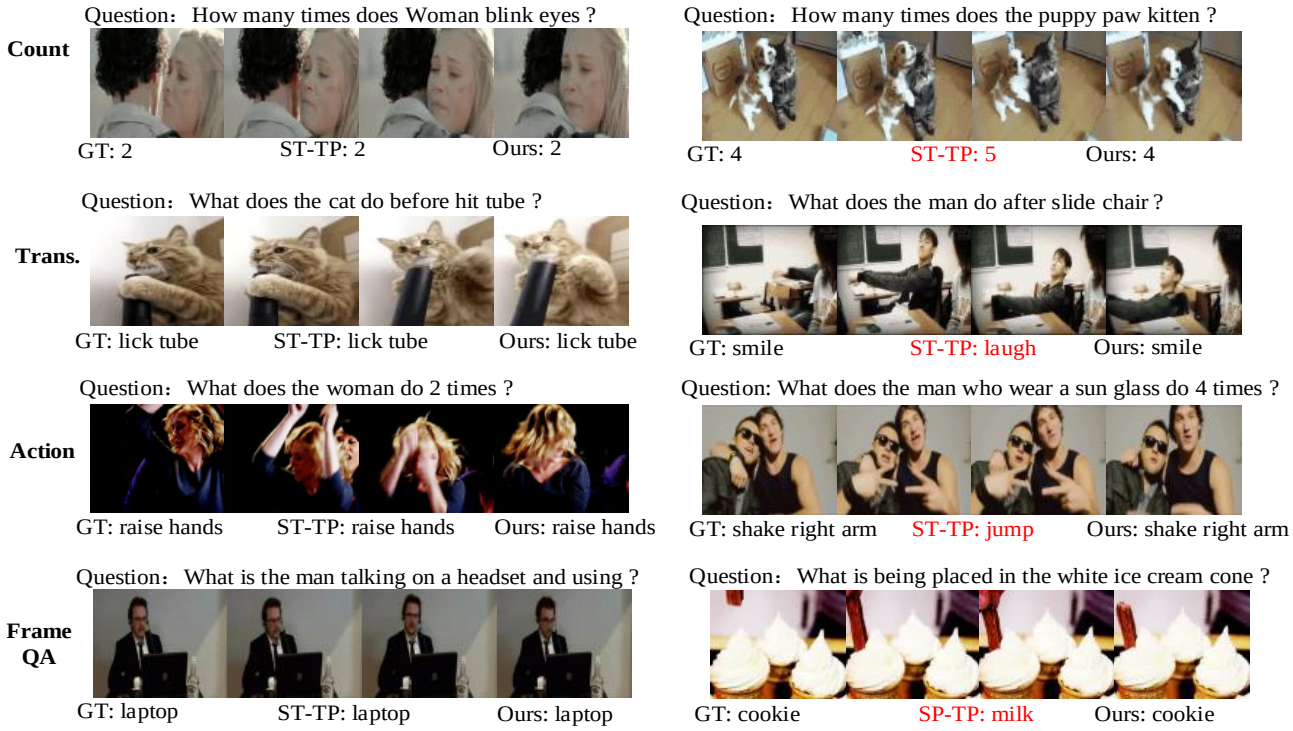

Figure 2: For each task, two examples are provided. Answers are predicted by ST-TP(R+C) model and our PSAC(R) model. The incorrect answers are marked with red.

Table 6: The effect of video-question co-attention block

	Action	Trans	FrameQA	Count
without QA-Co	67.15	75.73	**55.84**	4.32
with Q2V	68.38	76.33	55.49	4.50
with QA-Co	**70.36**	**76.89**	55.67	**4.27**

we further study the effect of question-to-video (Q2V) attention and the results is shown in the second row of Tab. 6. The performance of with Q2V is higher than without QA-Co but lower than with QA-Co. The comparison results demonstrate the positive role of Q2V and V2Q.

4.6 Qualitative Analysis

In this section, we provide some qualitative results produced by ST-TP(R+C) (Jang et al. 2017) and our PSAC(R). We cannot provide the answers from Co-memory(R+F), because the code has not been released yet. All the qualitative examples are displayed in Fig. 2. The first column shows some positive examples where both ST-TP(R+C) and our PSAC(R) can provide the accurate answers. The second column demonstrates some examples where ST-TP(R+C) fails to provide the correct answer. For Count task, our PSAC(R) model can recognize and locate the action precisely. For example, our PSAC(R) can correctly count the number of times of "puppy paw kitten". For Trans. task, our PSAC(R) can precisely identify the action after "slide chair" that is "smile", while the answer for ST-TP(R+C) is "laugh". The answer for ST-TP(R+C) is acceptable but not precise. For Action task, when ask "what does the man who wear glass do 4 times", our PSAC(R) provides an accurate answer "shake right arm". Even though ST-TP(R+C) analyzes both appearance and motion information, it provides a wrong answer "jump". For FrameQA task, compared with ST-TP(R+C) with an answer "milk", our model PSAC(R) provides an accurate answer "cookie". To conclude, the second column demonstrates the effectiveness of our PSAC(R) for video question answering task, even without analyzing the video motion features.

5 Conclusion

In this work, we introduce a simple and interpretable network named Positional Self-Attention with Co-Attention for video question-answering, which adopts positional self-attention block to replace LSTM to model the data dependencies. Besides, video-question co-attention is used to help attend to both visual and textual information which makes the answers more accurate. Experimental results on TGIF-QA demonstrate that our approach outperforms state-of-the-art methods significantly on three tasks and achieve comparable result on Count task. Our model requires less computation time and achieves better performance compared with the RNNs-based methods.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2014J063, No. ZYGX2016J085) and the National Natural Science Foundation of China (Grant No. 61772116, No. 61502080, No. 61632007, No. 61602049). This work was partly supported by the 111 Project No. B17008.

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2017. Bottom-up and top-down attention for image captioning and VQA. *CoRR* abs/1707.07998.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. 1800–1807.

Gao, L.; Guo, Z.; Zhang, H.; Xu, X.; and Shen, H. T. 2017. Video captioning with attention-based LSTM and semantic consistency. *IEEE Trans. Multimedia* 19(9):2045–2055.

Gao, J.; Ge, R.; Chen, K.; and Nevatia, R. 2018a. Motion-appearance co-memory networks for video question answering.

Gao, L.; Zeng, P.; Song, J.; Liu, X.; and Shen, H. T. 2018b. Examine before you answer: Multi-task learning with adaptive-attentions for multiple-choice VQA. In *ACM Multimedia*, 1742–1750.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. 1243–1252.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2680–8.

Kim, K.; Heo, M.; Choi, S.; and Zhang, B. 2017. Deepstory: Video story QA by deep embedded memory networks. In *IJCAI*, 2016–2022.

Kim, J.; Jun, J.; and Zhang, B. 2018a. Bilinear attention networks. *CoRR* abs/1805.07932.

Kim, J.; Jun, J.; and Zhang, B. 2018b. Bilinear attention networks. *CoRR* abs/1805.07932.

Nam, H.; Ha, J.; and Kim, J. 2017. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2156–2164.

Palangi, H.; Smolensky, P.; He, X.; and Deng, L. 2018. Question-answering with grammatically-interpretable representations. In *AAAI*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *ACL*, 1532–1543.

Seo, M. J.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *CoRR* abs/1611.01603.

Song, J.; Zeng, P.; Gao, L.; and Shen, H. T. 2018. From pixels to objects: Cubic visual attention for visual question answering. In *IJCAI*, 906–912.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Highway networks. *CoRR* abs/1505.00387.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 6000–6010.

Xiong, C.; Merity, S.; and Socher, R. 2016. Dynamic memory networks for visual and textual question answering. In *ICML*, 2397–2406.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A. C.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2048–2057.

Yang, Z.; He, X.; Gao, J.; Deng, L.; and Smola, A. J. 2016. Stacked attention networks for image question answering. In *CVPR*, 21–29.

Yang, Y.; Jie, Z.; Jiangbo, A.; Yi, B.; Alan, H.; and Tao, S. H. 2018. Video captioning by adversarial lstm. *IEEE Transactions on Image Processing* 27(11):5600–5611.

Yu, Y.; Ko, H.; Choi, J.; and Kim, G. 2017. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 3261–3269.

Yu, A. W.; Dohan, D.; Luong, M.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR* abs/1804.09541.

Zeng, K.; Chen, T.; Chuang, C.; Liao, Y.; Niebles, J. C.; and Sun, M. 2017. Leveraging video descriptions to learn video question answering. In *AAAI*, 4334–4340.

Zhang, H.; Goodfellow, I. J.; Metaxas, D. N.; and Odena, A. 2018. Self-attention generative adversarial networks. *CoRR* abs/1805.08318.

Zhou, L.; Zhou, Y.; Corso, J. J.; Socher, R.; and Xiong, C. 2018. End-to-end dense video captioning with masked transformer. volume abs/1804.00819.