TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering

Yunseok Jang[1], Yale Song[2], Youngjae Yu[1], Youngjin Kim[1], Gunhee Kim[1]

[1]Seoul National University, [2]Yahoo Research

[1]{yunseok.jang, gunhee}@snu.ac.kr, {yj.yu, youngjin.kim}@vision.snu.ac.kr   [2]yalesong@yahoo-inc.com

http://vision.snu.ac.kr/projects/tgif-qa

Abstract

Vision and language understanding has emerged as a subject undergoing intense study in Artificial Intelligence. Among many tasks in this line of research, visual question answering (VQA) has been one of the most successful ones, where the goal is to learn a model that understands visual content at region-level details and finds their associations with pairs of questions and answers in the natural language form. Despite the rapid progress in the past few years, most existing work in VQA have focused primarily on images. In this paper, we focus on extending VQA to the video domain and contribute to the literature in three important ways. First, we propose three new tasks designed specifically for video VQA, which require spatio-temporal reasoning from videos to answer questions correctly. Next, we introduce a new large-scale dataset for video VQA named TGIF-QA that extends existing VQA work with our new tasks. Finally, we propose a dual-LSTM based approach with both spatial and temporal attention, and show its effectiveness over conventional VQA techniques through empirical evaluations.

1. Introduction

Vision and language understanding has emerged as a subject undergoing intense study in Artificial Intelligence. Among many tasks in this line of research, visual question answering (VQA) has been one of the most successful ones, where the goal is to learn a model that understands visual content at region-level details and finds their associations with pairs of questions and answers in the natural language form [2]. Part of the reasons for the success of VQA is that there exists a number of large-scale datasets with well-defined tasks and evaluation protocols [2, 24, 28, 42], which provided a common ground to researchers to compare their methods in a controlled setting.
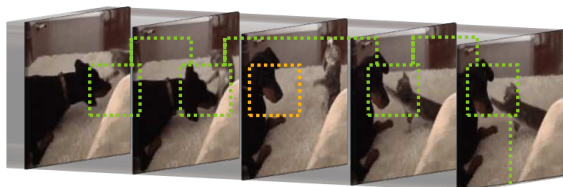
While we have seen a rapid progress in video analysis [17, 31, 34], most existing work in VQA have focused primarily on images. We believe that the limited progress in video VQA, compared to its image counter-

Image VQA

Q) What is the color of the bird?
A) White
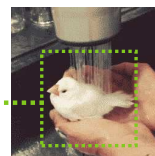
Video VQA

Q) How many times does the cat touch the dog?
A) 4 times

Figure 1. Much of conventional VQA tasks focus on reasoning from images (top). This work proposes a new dataset with tasks designed specifically for video VQA that requires spatio-temporal reasoning from videos to answer questions correctly (bottom).

part, is due in part to the lack of large-scale datasets with well-defined tasks. Some early attempts have been made to fill this gap by introducing datasets that leverage movie data [29, 33], focusing on storyline comprehension either from short video clips [29] or from movies and scripts [33]. However, existing question-answer pairs are either an extension to the conventional image VQA tasks, *e.g.*, by adding action verbs as the new answer type [29] to the existing categories of object, number, color, and location [28], or require comprehensive understanding of long textual data, *e.g.*, movie scripts [33].

In this paper, we contribute to the literature in VQA in three important ways. First, we propose three new tasks designed specifically for video VQA, which require spatio-temporal reasoning from videos to answer questions correctly. Next, we introduce a new large-scale dataset for video VQA that extends existing work in image VQA with our new tasks. Finally, we propose a dual-LSTM based approach with an attention mechanism to solve our problem, and show its effectiveness over conventional VQA tech-
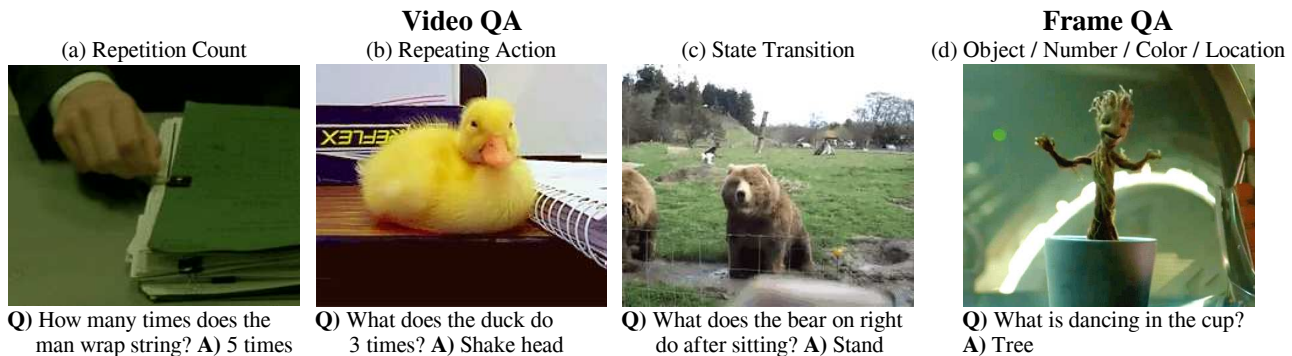
(a) Repetition Count (b) Repeating Action (c) State Transition (d) Object / Number / Color / Location

Q) How many times does the man wrap string? **A)** 5 times

Q) What does the duck do 3 times? **A)** Shake head

Q) What does the bear on right do after sitting? **A)** Stand

Q) What is dancing in the cup? **A)** Tree

Figure 2. Our TGIF-QA dataset introduces three new tasks for video QA, which require spatio-temporal reasoning from videos (*e.g.* (a) repetition count, (b) repeating action, and (c) state transition). It also includes frame QA tasks that can be answered from one of frames.

niques through empirical evaluations. Our intention is not to compete with existing literature in VQA, but rather to complement them by providing new perspectives on the importance of spatio-temporal reasoning in VQA.

Our design of video VQA tasks is inspired by existing works in video understanding, e.g., repetition counting [21] and state transitions [15], intending to serve as a bridge between video understanding and video VQA. We define three tasks: (1) count the number of repetitions of a given action; (2) detect a repeating action given its count; and (3) identify state transitions, *i.e.*, what has happened before or after a certain action state. As illustrated in Figure 2, solving our tasks requires comprehensive spatio-temporal reasoning from videos, an ideal scenario for evaluating video analysis techniques. In addition to our new tasks, we also include the standard image VQA type tasks by automatically generating question-answer pairs from video captions [28]. Following the existing work in VQA, we formulate our questions as either open-ended or multiple choice. This allows us to take advantage of well-defined evaluation protocols.

To create a benchmark for our tasks, we collected a new dataset for video VQA based on the Tumblr GIF (TGIF) dataset [22], which was originally proposed for video captioning. The TGIF dataset utilizes animated GIFs as their visual data, which have recently emerged as an attractive source of data in computer vision [13, 22] due to their concise format and cohesive storytelling nature [5]; this makes it especially ideal for vision and language understanding. We therefore extend the TGIF dataset to the VQA domain, adding 104K QA pairs from 57K animated GIFs from the TGIF dataset; we name our dataset *TGIF-QA*.

The current state-of-the-art in VQA have focused on finding visual-textual associations from images [2, 28], employing a spatial attention mechanism to learn "where to look" in an image given the question [10, 18]. While existing techniques demonstrated impressive performance on image VQA, they are inadequate for the video domain because a video contains visual information both in spatial and temporal dimensions, requiring an appropriate spatio-

temporal reasoning mechanism. In this work, we leverage spatio-temporal information from video by employing LSTMs not only for the QA pairs, as in the previous works, but also for the video input. We also evaluate spatial and temporal attention mechanisms to selectively attend to specific parts of a video. We discuss various design considerations and report empirical results in Section 5.

To summarize, our major contributions include:

1. We propose three new tasks designed specifically for video VQA, which require spatio-temporal reasoning from videos to answer questions correctly.

2. We introduce a new dataset, TGIF-QA, that consists of 104K QA pairs from 57K animated GIFs.

3. We propose a dual-LSTM based approach with an attention mechanism to solve our video QA tasks.

4. Code and the dataset are available on our project page.

2. Related Works

VQA is a relatively new problem domain first introduced by Malinowski *et al*. [24] and became popularized by Antol *et al*. [2]. Despite its short history, there has been a flourishing amount of research produced within the past few years [9, 6]. Here, we position our research and highlight key differences compared to previous work in VQA.

Datasets. Most existing VQA datasets are image-centric, *e.g.*, DAQUAR [24], abstract scenes [23], VQA [2], Visual Madlibs [40], DAQUAR-Consensus [25], FM-IQA [11], COCO-QA [28], and Visual7W [42]. Also, appearing in the same proceedings are CLEVR [16], VQA2.0 [12], and Visual Dialog [8], which all address image-based VQA. Our work extends existing works to the video domain, creating QA pairs from short video clips rather than static images.

There have been some recent efforts to create video VQA datasets based on movies. Rohrbach *et al*. [29] extended the LSMDC movie description dataset [29] to the

VQA domain. Similarly, Tapaswi *et al*. [33] introduced the MovieQA dataset by leveraging movies and movie scripts. Our work contributes to this line of research, but instead of restricting the source of video to the movie clips, here we leverage animated GIFs from the Internet, which have concise format and deliver cohesive visual stories [5, 22].

Tasks. Existing QA pairs in the VQA literature have one of the following forms: open-ended and multiple choice; we consider fill-in-the-blank as a special case of the open-ended form. Open-ended questions provide either a complete or incomplete sentence and the system must guess the correct answer word. Multiple choice questions, on the other hand, provide a number of answer candidates, either as texts [2] or bounding boxes [42], and the system must choose the correct one. Our dataset contains questions in the open-ended and multiple choice forms.

Most existing VQA tasks are image-centric and thus ask questions about visual concepts that appear only in images, *e.g.*, objects, colors, and locations [2]. In the video domain, the LSMDC-QA dataset [29] introduced the movie fill-in-the-blank task by adding action verbs to the answer set, requiring spatio-temporal reasoning from videos at the sequence level (similar to action recognition). Our tasks also require spatio-temporal reasoning from videos, but at the frame level – counting the number of repetitions and memorizing state transitions from a video requires more comprehensive spatio-temporal reasoning.

The MovieQA dataset [33] introduced an automatic story comprehension task from video and movie script. The questions are designed to require comprehensive visual-textual understanding of a movie synopsis, to the level of details of proper nouns (*e.g.*, names of characters and places in a movie). Compared to the MovieQA dataset, our task is on spatio-temporal reasoning rather than story comprehension, and we put more focus on understanding visual signals (animated GIFs) rather than textual signals (movie scripts).

Techniques. Most existing techniques in VQA are designed to solve image VQA tasks. Various techniques have demonstrated promising results, such as the compositional model [1] and the knowledge-based model [36]. The current state-of-the-art techniques employ a spatial attention mechanism with visual-textual joint embedding [10, 18]. Our work extends this line of work to the video domain, by employing spatial and temporal attention mechanisms to solve video VQA tasks.

There are very few approaches designed specifically to solve video VQA. Yu *et al*. [41] used LSTMs to represent both videos and QA pairs and adopted a semantic attention mechanism [39] on both input word representation and output word prediction. We also use LSTMs to represent both videos and QA pairs, with a different attention mechanism to capture complex spatio-temporal patterns in videos. To the best of our knowledge, our model is the first to leverage

Task		# QA pairs			# GIFs		
		Train	Test	Total	Train	Test	Total
Video QA	Rep. Count	17,670	2,409	20,079	17,670	2,409	20,079
	Rep. Action	8,958	1,151	10,109	8,958	1,151	10,109
	Transition	18,104	2,544	20,648	9,052	1,272	10,324
Frame QA	Object	16,755	5,586	22,341	15,584	3,209	18,793
	Number	8,096	3,148	11,244	8,033	1,903	9,936
	Color	11,939	3,904	15,843	10,872	3,190	14,062
	Location	2,602	1,053	3,655	2,600	917	3,517
Total		84,124	19,795	103,919	48,198	8,522	56,720

Table 1. Statistics of our dataset, organized into different tasks.

temporal attention for video VQA tasks, which turns out to improve the QA performance in our experiments.

3. TGIF-QA Dataset

Our dataset consists of 103,919 QA pairs collected from 56,720 animated GIFs. We explain our new tasks designed for video VQA and present the data collection process.

3.1. Task Definition

We introduce four task types used in our dataset. Three of them are new and unique to the video domain, including:

Repetition count. One task that is truly unique to videos would be counting the number of repetitions of an action. We define this task as an open-ended question about counting the number of repetitions of an action, *e.g.*, Figure 2 (a). There are 11 possible answers (from 0 to 10+).

Repeating action. A companion to the above, this task is defined as a multiple choice question about identifying an action that has been repeated in a video, *e.g.*, Figure 2 (b). We provide 5 options to choose from.

State transition. Another task unique to videos is asking about transitions of certain states, including facial expressions (*e.g.*, from happy to sad), actions (*e.g.*, from running to standing), places (*e.g.*, from the table to the floor), and object properties (*e.g.*, from empty to full). We define this task as a multiple choice question about identifying the state before (or after) another state, *e.g.*, Figure 2 (c). We provide 5 options to choose from.

The three tasks above require analyzing multiple frames of a video; we refer to them collectively by **video QA**.

Besides our three video QA tasks, we also include another one, which we call **frame QA** to highlight the fact that questions in this task can be answered from one of the frames in a video. Depending on the video content, it can be any frame or one particular from of a video. For this task, we leverage the video captions provided in the TGIF dataset [22] and use the NLP-based technique proposed in Ren *et al*. [28] to generate QA pairs automatically from the captions. This task is defined as an open-ended question about identifying the best answer (from a dictionary of words of type object, number, color, and location) given a question in a complete sentence, *e.g.*, Figure 2 (d).

Task	Question	Answer
Repetition count	How many times does the [SUB] [VERB] [OBJ] ?	[#Repeat]
Repeating action	What does the [SUB] do [#Repeat] times ?	[VERB] [OBJ]
State transition	What does the [SUB] do before [Next state] ?	[Previous state]
	What does the [SUB] do after [Previous state] ?	[Next state]

Table 2. Templates used for creating video QA pairs.

Category	Motion	Contact	Percp.	Comm.	Body
Examples	shake walk jump run move	stand sit hit kiss touch	look stare show hide watch	wave talk nod point face	smile blink laugh blow wink
LSMDC-QA [29]	27.98%	19.09%	**14.78%**	5.19%	4.43%
MovieQA [33]	13.90%	11.76%	4.95%	**12.17%**	2.18%
TGIF-QA(ours)	**37.05%**	**25.38%**	8.86%	7.97%	**7.07%**

Table 3. Distributions of verbs in the answers from different datasets. We show top five most common categories with example verbs. Percp.: perception, comm.: communication.

3.2. QA Collection

For the frame QA, we use the same setup of Ren *et al.* [28] and apply their method on the captions provided in the TGIF dataset [22]. As shown in Table 1, this produced a total of 53,083 QA pairs from 51,225 GIFs.

For the video QA, we generate QA pairs by using a combination of crowdsourcing and template-based approach. This produced a total of 50,836 QA pairs from 19,485 GIFs.

Crowdsourcing. We conducted two crowdsourcing studies, collecting the following information:

- Repetition: subject, verb, object, and the number of repetitions (from 2 to 10+ times) for a repeating action.

- State transition: subject, transition type (one of facial expression, action, place, or object property), previous state, next state for the changed states, if any.

We used drop-down menus to collect answers for the number of repetitions and the transition type, and used text boxes for all the others. A total of 568 workers have participated and were compensated by 5 cents per video clip.

Quality control. Our task includes many free-form input; proper quality control is crucial. Inspired by Li *et al.* [22], we filter out suspiciously negligent workers by automatic validation. Specifically, we collect a small set of video clips (159 for repetition and 172 for state transition) as the validation set, and manually annotate each example with a set of appropriate answers; we consider those the gold standard. We then include one of the validation samples to each main task and check if a worker answers it correctly by matching their answers to our gold standard set. We reject the answers from workers who fail to pass our validation, and add those workers to our blacklist so that they cannot participate in other tasks. We regularly reviewed rejected answers to correct the mistakes made by our automatic validation, removing the worker from our blacklist and adding their answers to our gold standard set.

Post processing. We lemmatize all verbs with the Word-Net lemmatizer and find the main verb in each state using the VerbNet [20]. We detect proper nouns in the collected answers using the DBpedia Spotlight [7] and replace them with the corresponding common noun, e.g., person names,

body parts, animal names, etc. We also remove any possessive determiners for the phrases used in answers.

QA generation. We generate QA pairs using the templates shown in Table 2. It is possible that the generated questions have grammatical errors; we fix those using the LanguageTool. We then generate multiple choice options for each QA pair, selecting four phrases from our dataset.

Specifically, we represent all verbs in our dictionary as a 300D vector using the GloVe word embedding [26] pretrained on the Common Crawl dataset. We then select four verbs, one by one in a greedy manner, whose cosine similarity with the verb from the answer phrase is greater than the 50th percentile, while at the same time the average cosine similarity from the current candidate verbs is maximal – this encourages the diversity in negative answers. We then randomly choose four phrases that include the candidate verbs.

For the repetition counting task, we automatically added samples that had zero count of an action, by randomly pairing a question from our question list with a GIF that was identified as having no repeating action.

3.3. Comparison with Other Video VQA Datasets

Table 4 compares our dataset with two existing video VQA datasets. LSMDC-QA refers to the subset of the data used for the VQA task in the LSMDC 2016 Challenge.

It shows that TGIF-QA is unique in terms of the objective and the sources of video and text. *i.e.*, it includes short video clips (GIFs) collected over social media, whereas the other two includes movie clips. Ours also includes both types of questions, open-ended and multiple choice, unlike other datasets. While our dataset is smaller than LSMDC-QA, we include tasks unique to video VQA. Therefore, our dataset can complement existing datasets with unique tasks.

Table 3 shows the distribution of verbs used in answers. We show top five most common verb categories obtained from the WordNet hierarchy. Most notably, TGIF-QA contains more dynamic verbs, such as the ones from the *motion* and the *contact* categories. This is an important characteristic of our dataset because it suggests the need for spatio-temporal reasoning to understand the content.

Dataset	Objective	Q. Type	Video Source	Text Source	# QA pairs	# Clips
LSMDC-QA [29]	Fill-in-the-blank for caption completion	OE	Movie	Movie caption	348,998	111,744
MovieQA [33]	Visual-textual story comprehension	MC	Movie	Movie synopsis	14,944	6,771
TGIF-QA (ours)	Spatio-temporal reasoning from video	OE & MC	Social media	Caption & crowdsourced	103,919	56,720

Table 4. Comparison of three video VQA datasets (Q.: question, OE: open-ended, and MC: multiple choice).
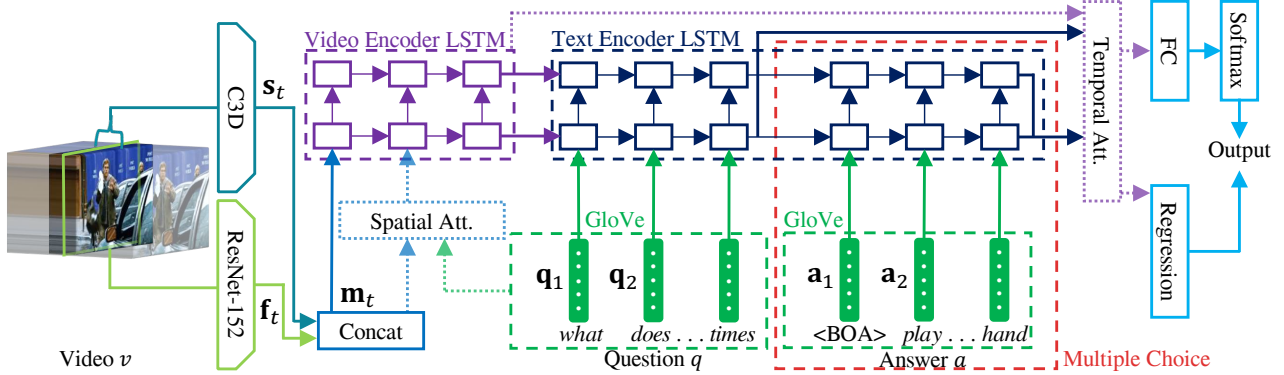

Figure 3. The proposed ST-VQA model for spatio-temporal VQA. See Figure 4 for the structure of spatial and temporal attention modules.

4. Approach

We present spatio-temporal VQA (ST-VQA) model for our task (see Figure 3). The input to our model is a tuple (v, q, a) of a video v, a question sentence q, and an answer phrase a; the answer phrase a is optional and provided only from multiple choice questions (indicated as red dashed box in Figure 3). The output is either a single word (for open-ended questions) or a vector of compatibility scores (for multiple choice questions). Our ST-VQA model captures visual-textual association between a video and QA sentences using two dual-layer LSTMs, one for each input.

4.1. Feature Representation

Video representation. We represent a video both at the frame-level and at the sequence-level. For the frame features, we use the ResNet-152 [14] pretrained on the ImageNet 2012 classification dataset [30]. For the sequence features, we use the C3D [34] pretrained on the Sport1M dataset [17]. We sample one every four frames to reduce the frame redundancy. For the C3D features, we take 16 subsequent frames centered at each time step, and pad the first or the last frame if too short. We denote the two video descriptors, ResNet-152 and C3D, by $\{\mathbf{f}_t\}_{t=1}^T$ and $\{\mathbf{s}_t\}_{t=1}^T$, respectively; T is the sequence length.

Depending on whether we use our spatio-temporal attention mechanism (explained in Section 4.4), we use different feature representations. For the ResNet-152 feature, we take the feature map of the res5c layer ($\in \mathbb{R}^{7 \times 7 \times 2,048}$) for the spatial attention mechanism and the pool5 features ($\in \mathbb{R}^{2,048}$) for the others. Similarly, for the C3D features, we take the conv5b layer ($\in \mathbb{R}^{7 \times 7 \times 1,024}$) for the spatial attention mechanism and the fc6 feature for the others.

Text representation. There are two types of text inputs: question and answer. A question is a complete sentence, while an answer is a phrase. We simply consider both as a sequence of words and represent them in the same way. For a given input, we represent each word as a 300D vector using the GloVe word embedding [26] pretrained on the Common Crawl dataset. We denote the text descriptor for questions and answers by $\{\mathbf{q}_n\}_{n=1}^N$ and $\{\mathbf{a}_m\}_{m=1}^M$, respectively; N and M are the sequence lengths.

4.2. Video and Text Encoders

Video encoder. We encode video features $\{\mathbf{s}_t\}_{t=1}^T$ and $\{\mathbf{f}_t\}_{t=1}^T$ using the video encoding LSTM, shown in the purple dashed box in Figure 3. We first concatenate the features $\mathbf{m}_t = [\mathbf{s}_t; \mathbf{f}_t]$, and feed them into the dual-layer LSTM one at a time, producing a hidden state $\mathbf{h}_t^v \in \mathbb{R}^D$ at each step:

$$\mathbf{h}_t^v = \text{LSTM}(\mathbf{m}_t, \mathbf{h}_{t-1}^v). \quad (1)$$

Since we employ a dual-layer LSTM, we obtain pairs of hidden states $\mathbf{h}_t^v = (\mathbf{h}_t^{v,1}, \mathbf{h}_t^{v,2})$. For brevity, we use the combined form $\mathbf{h}_t^v$ for the rest of the paper. We set the dimension $D = 512$.

Text encoder. We encode text features of question $\{\mathbf{q}_n\}_{n=1}^N$ and answer choices $\{\mathbf{a}_m\}_{m=1}^M$ using the text encoding LSTM, shown in the navy dashed box in Figure 3. While open-ended questions involve only a question, multiple choice questions come with a question and a set of answer candidates. We encode a question $\{\mathbf{q}_n\}_{n=1}^N$ and each of the answer choices $\{\mathbf{a}_m\}_{m=1}^M$ using a dual-layer LSTM:

$$\mathbf{h}_n^q = \text{LSTM}(\mathbf{q}_n, \mathbf{h}_{n-1}^q), \quad \mathbf{h}_0^q = \mathbf{h}_T^v. \quad (2)$$
$$\mathbf{h}_m^a = \text{LSTM}(\mathbf{a}_m, \mathbf{h}_{m-1}^a), \quad \mathbf{h}_0^a = \mathbf{h}_N^q \quad (3)$$

We set the initial hidden state $\mathbf{h}_0^q$ to the last hidden state of the video encoder $\mathbf{h}_T^v$, so that visual information is "carried over" to the text encoder – an approach similar to other sequence-to-sequence models [32, 35]. To indicate the starting point of the answer candidate, we put a special character, <BOA> (begin of answer). We also use the last hidden state of the question encoder as the initial hidden state of the answer encoder. Similar to the video encoder, we set the dimension of all the hidden states to $D = 512$.

4.3. Answer Decoders

We design three decoders that provide answers: one for the multiple choice, the other two for the open-ended.

Multiple choice. We define a linear regression function that takes as input the final hidden states from the answer encoder, $\mathbf{h}_M^a$, and computes a real-valued score for each answer candidate,

$$s = \mathbf{W}_s^\top \mathbf{h}_M^a \qquad (4)$$

where $\mathbf{W}_s \in \mathbb{R}^{1,024}$ is the model parameter. We train the decoder by minimizing the hinge loss of pairwise comparisons, $\max(0, 1 + s_n - s_p)$, where s_n and s_p are scores computed from an incorrect and correct answers, respectively. We use this decoder to solve repeating action and state transition tasks.

Open-ended, number. Similar to the above, we define a linear regression function that takes as input the final hidden states from the answer encoder, computes an integer-valued answer by adding a bias term b_s to Equation (4). We train the decoder by minimizing the ℓ_2 loss between the answer and the predicted value. We use this encoder to solve the repetition count task.

Open-ended, word. We define a linear classifier that takes as input the final hidden states from the question encoder, $\mathbf{h}_N^q \in \mathbb{R}^{1,024}$, and selects an answer from a vocabulary of words $\mathcal{V}$ by computing a confidence vector $\mathbf{o} \in \mathbb{R}^{|\mathcal{V}|}$

$$\mathbf{o} = \text{softmax}\left(\mathbf{W}_o^\top \mathbf{h}_N^q + \mathbf{b}_o\right) \qquad (5)$$

where $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{V}| \times 1,024}$ and $\mathbf{b}_o \in \mathbb{R}^{|\mathcal{V}|}$ are model parameters. We train the decoder by minimizing the softmax loss function. The solution is obtained by $y = \text{argmax}_{\mathbf{y} \in \mathcal{V}}(\mathbf{o})$. We use this encoder to solve the frame QA task.

4.4. Attention Mechanism

While our tasks require spatio-temporal reasoning from videos, the model explained so far is inadequate for such tasks because, in theory, the video encoder "squashes" necessary details of the spatio-temporal visual information into a flat representation. We now explain our spatial and temporal attention mechanisms, illustrated in Figure 4. The former allows us to learn *which regions in each frame of a video* to attend to, while the latter allows us to learn *which*
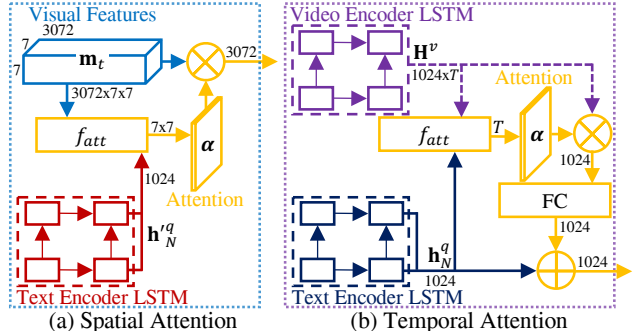

(a) Spatial Attention   (b) Temporal Attention

Figure 4. Our spatial and temporal attention mechanisms.

frames in a video to attend to solve our tasks. As such, we employ different mechanisms to model each attention type, based on Xu *et al.* [37] for spatial attention and Bahdanau *et al.* [4] for temporal attention.

Spatial attention. To learn *which regions in a frame* to attend for each word, we use visual representation that preserves spatial information and associate it with a QA pair. Also, we need textual signals when encoding each frame in the video decoder. However, as the model takes a QA pair only after encoding a video, this information is not available a priori. We solve this issue by simply defining another dual-layer LSTM that shares its model parameters with the text encoder.

Figure 4 (a) illustrates our spatial attention mechanism. For each time step t in a video sequence, we compute a 7×7 spatial attention mask $\boldsymbol{\alpha}_t = f_{att}(\mathbf{h}_N'^q, \mathbf{m}_t)$, where $\mathbf{h}_N'^q \in \mathbb{R}^{1,024}$ is the output of the text encoder and $\mathbf{m}_t \in \mathbb{R}^{7 \times 7 \times 3,072}$ is the visual feature map. We then pass the attended visual feature $\boldsymbol{\alpha}_t \mathbf{m}_t \in \mathbb{R}^{3,072}$ to the video encoder. The function $f_{att}(\cdot, \cdot)$ is a multi-layer perception (MLP) that operates over each of 7×7 spatial locations, followed by the softmax function. Our MLP is a single layer of 512 hidden nodes with the tanh activation function.

Temporal attention. To learn *which frames in a video* to attend to, we use a visual representation that preserves temporal information and associate it with a QA pair.

Figure 4 (b) shows our temporal attention mechanism. After we encode video and question sequences, we compute a $1 \times T$ temporal attention mask $\boldsymbol{\alpha} = f_{att}(\mathbf{h}_N^q, \mathbf{H}^v)$, where $\mathbf{h}_N^q \in \mathbb{R}^{1,024}$ is the last state of the text encoder and $\mathbf{H}^v \in \mathbb{R}^{T \times 1,024}$ is a state sequence from the video encoder. We then compute the attended textual signal tanh($\boldsymbol{\alpha} \mathbf{H}^v \mathbf{W}_\alpha) \oplus \mathbf{h}_N^q$, where $\mathbf{W}_\alpha \in \mathbb{R}^{1,024 \times 1,024}$ and $\oplus$ is an element-wise sum, and pass it to the answer decoder. We use the same $f_{att}(\cdot, \cdot)$ as with our spatial attention, with its MLP operating over the temporal dimension T.

4.5. Implementation Details

We use the original implementations of ResNet [14], C3D [34], and GloVe [26] to obtain features from videos

and QA text. All the other parts of our model are implemented using the TensorFlow library. Except for extracting the input features, we train our model end-to-end. For the dual-layer LSTMs, we apply layer normalization [3] to all cells, with the dropout [27] with a rate of 0.2. For training, we use the ADAM optimizer [19] with an initial learning rate of 0.001. All weights in LSTMs are initialized from a uniform distribution, and all the other weights are initialized from a normal distribution.

5. Experiments

We tackle open-ended word and multiple choice tasks as multi-class classification, and use the accuracy as our evaluation metric, reporting the percentage of correctly answered questions. For the open-ended number task, we use the mean ℓ_2 loss as our evaluation metric to account for the ordinal nature of the numerical labels. We split the data into training and test sets as shown in Table 1, following the setting in the original TGIF dataset [22].

5.1. Baselines

We compare our approach against two recent image-based VQA methods [10, 28], as well as one video-based method [41]. For a fair comparison, we re-implemented the baselines in TensorFlow and trained them from scratch using the same set of input features.

Image-based. We select two state-of-the-art methods in image-based VQA: VIS+LSTM [28] and VQA-MCB [10]. VIS+LSTM combines image representation with textual features encoded by an LSTM, after which it solves open-ended questions using a softmax layer [28]. VQA-MCB, on the other hand, uses multimodal compact bilinear pooling to handle visual-textual fusion and spatial attention [10]. This model is the winner of the VQA 2016 challenge.

Since both methods take a single image as input, we adjust them to be applicable to video VQA. We evaluate two simple approaches: *aggr* and *avg*. The *aggr* method aggregates input features of all frames in a video by averaging them, and uses it as input to the model. The *avg* method, on the other hand, solves the question using each frame of a video, one at a time, and report the average accuracy across all frames of all videos, i.e., $1/N \sum_{i=1}^{N} (1/M_i \sum_{j=1}^{M_i} \mathbb{I}[y_{i,j} = y_i^*])$, where N is the number of videos, M_i is the number of frames for the i-th video, $\mathbb{I}[\cdot]$ is an indicator function, $y_{i,j}$ is a predicted answer for the j-th frame of the i-th video, and y_i^* is an answer for the i-th video.

Video-based. We select the state-of-the-art method in video VQA, Yu *et al*. [41], which has won the retrieval track in the LSMDC 2016 benchmark. We use their retrieval model that employs the same decoder as explained in section 4.3. Although the original method used an ensemble approach, we here use a single model for a fair comparison.

Model		Repetition		State	Frame
		Count	Action	Trans.	QA
Random chance		20.3815	20.00	20.00	0.06
VIS+LSTM	aggr	6.4461	23.41	57.75	34.59
[28]	avg	6.0975	15.40	13.35	34.97
VQA-MCB	aggr	7.0613	23.33	53.64	25.70
[10]	avg	5.6164	37.59	41.52	15.49
Yu *et al*. [41]		5.9269	33.91	60.97	39.64
ST-VQA-ResNet		5.8257	48.51	61.90	45.60
ST-VQA-C3D		5.6645	48.62	60.89	45.18
ST-VQA-Concat		5.5868	<u>50.48</u>	62.09	44.96
ST-VQA-Sp.		5.5682	46.70	56.10	45.45
ST-VQA-Tp.		<u>5.4531</u>	**51.74**	<u>63.28</u>	**48.56**
ST-VQA-Sp.Tp.		**4.4621**	46.44	**63.79**	<u>47.79</u>

Table 5. Experimental results of VQA according to different problem types on our TGIF-QA dataset. (Sp.) indicates the spatial attention and (Tp.) means temporal one. We report the mean ℓ_2 loss for the repetition count task, and the accuracy for the other three tasks.

Variants of our method. To conduct an ablation study of our method, we compare five variants of our model, as shown in Table 5. The first three (ResNet, C3D, Concat) evaluate different representations for the video input; Concat uses both ResNet and C3D features. They also do not employ our spatial and temporal attention mechanisms. The next two variants (Spatial and Temporal) includes either one of the attention mechanisms. Finally, we evaluate a combination of the two attention mechanisms, by training the temporal part first and finetuning the spatial part later.

5.2. Results and Analysis

Table 5 summarizes our results. We observe that video-based methods outperform image-based methods, suggesting the need for spatio-temporal reasoning in solving our video QA tasks. We note, however, that the differences may not be seen significant; we believe this is because the C3D features already capture spatio-temporal information to some extent.

A comparison between different input features of our method (ResNet, C3D, Concat) suggests the importance of having both representations in our model. Among the three baselines, the Concat approach that uses both features achieves the best performance across all tasks.

A comparison between different attention mechanisms (Spatial and Temporal) shows the effectiveness of our temporal attention mechanism, achieving the best (or the second best) performance across all tasks. Similar results are reported in the literature; for example, in the video captioning task, Yao *et al*. [38] obtained the best result by considering both local and global temporal structures.

Finally, Figure 5 shows some qualitative examples from different approaches on the four task types of TGIF-QA. We

| | (a) Repetition Count | (b) Repeating Action | (c) State Transition | (d) Frame QA |

(a) Repetition Count

Q) How many times does the cat lick?
(Ours) 7 times
(VQA-MCB) 3 times
(VIS+LSTM) 0 times
(LSMDC-ret.) 6 times

Q) How many times does the kitten lick paw?
(Ours) 2 times
(VQA-MCB) 5 times
(VIS+LSTM) 3 times
(LSMDC-ret.) 2 times

(b) Repeating Action

Q) What does the cat do 3 times?
(Ours) Put head down
(VQA-MCB) Dance on floor
(VIS+LSTM) Move legs
(LSMDC-ret.) Move legs

Q) What does the animal do 2 times?
(Ours) Pump arms
(VQA-MCB) Stab knife
(VIS+LSTM) Play air drum
(LSMDC-ret.) Toss a pillow

(c) State Transition

Q) What does the model do after lower coat?
(Ours) Pivot around
(VQA-MCB) Hold up a decoration
(VIS+LSTM) Bend over
(LSMDC-ret.) Bend over

Q) What does the man do before walk backwards?
(Ours) Grap something
(VQA-MCB) Look at another person
(VIS+LSTM) Brush hair
(LSMDC-ret.) Brush hair

(d) Frame QA

Q) What is the color of the bulldog?
(Ours) Brown
(VQA-MCB) Red
(VIS+LSTM) White
(LSMDC-ret.) Black

Q) What runs and jumps over two hurdles ?
(Ours) Horse
(VQA-MCB) Dog
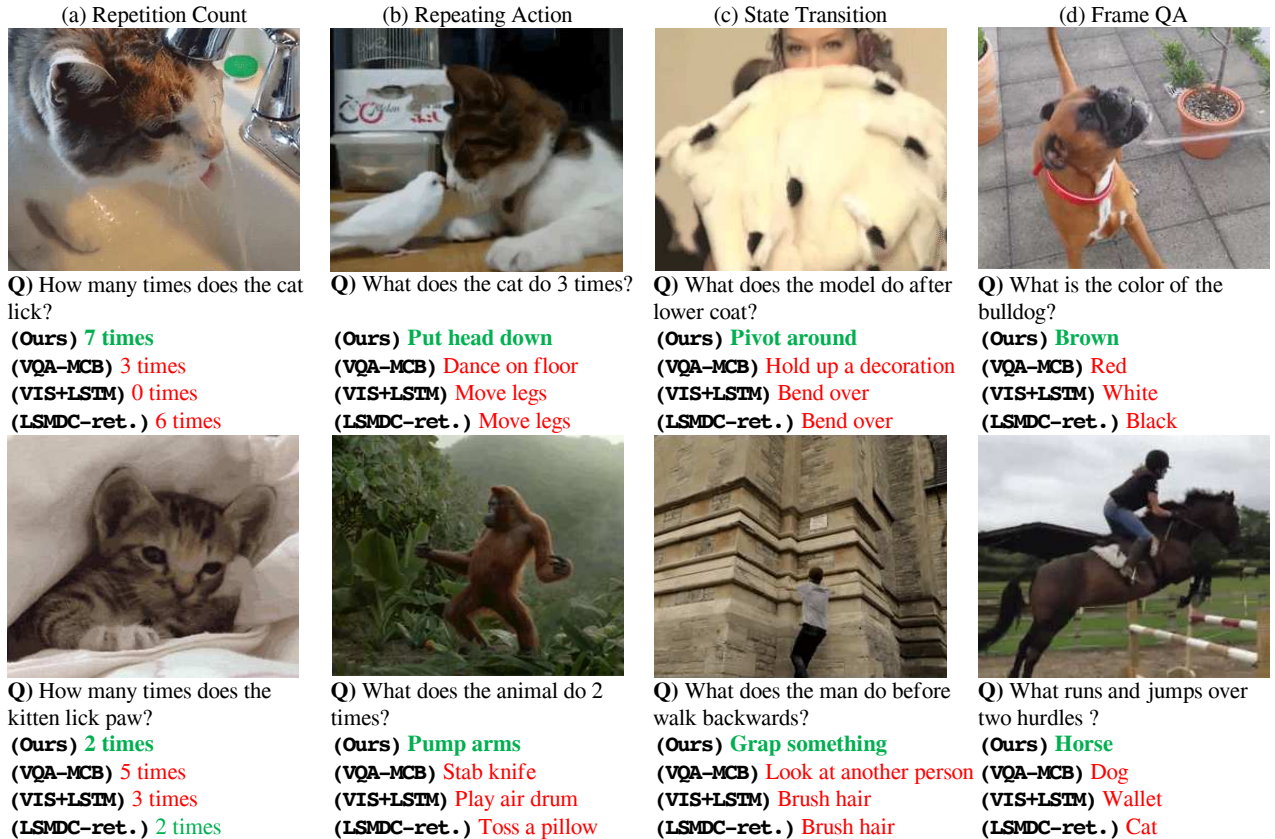**(VIS+LSTM)** Wallet
(LSMDC-ret.) Cat

Figure 5. Qualitative comparison of VQA results from different approaches, on the four task types of our TGIF-QA dataset.

observe that answering the questions indeed requires spatio-temporal reasoning. For example, the cat in Figure 5 (b) puts head down and moves legs multiple times, which cannot be answered without spatio-temporal reasoning. Our method successfully combines spatial and temporal visual representation from the input-level via ResNet and C3D features, and learns to selectively attend to them via our two attention mechanisms.

6. Conclusion

Our work complements and extends existing work on VQA with three main contributions: (i) proposing three new tasks that require spatio-temporal reasoning from videos, (ii) introducing a new large-scale dataset of video VQA with 104K QA pairs from 57K animated GIFs, and (iii) designing a dual-LSTM based approach with both spatial and temporal attention mechanisms.

Moving forward, we plan to improve our ST-VQA model in several directions. Although our model is based on a sequence-to-sequence model [35] to achieve simplicity, it can be improved in different ways, such as adopting the concept of 3D convolution [34]. Another direction is to find better ways to combine visual-textual information. Our model without the attention module (*e.g.* Concat in Table 5)

combines visual-textual information only at the text encoding step. Although our attention mechanisms explored ways to combine the two modalities to some extent, we believe there can be more principled approaches to do it efficiently, such as the recently proposed multimodal compact bilinear pooling [10].

References

[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural Module Networks. In *CVPR*, 2016.

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization. In *arXiv:1607.06450*, 2016.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*, 2015.

[5] S. Bakhshi, D. A. Shamma, L. Kennedy, Y. Song, P. de Juan, and J. J. Kaye. Fast, Cheap, and Good - Why Animated GIFs Engage Us. In *CHI*, 2016.

[6] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank. Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures. *JAIR*, 55:409–442, 2016.

[7] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *I-Semantics*, 2013.

[8] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra. Visual Dialog. In *CVPR*, 2017.

[9] F. Ferraro, N. Mostafazadeh, T.-H. Huang, L. Vanderwende, J. Devlin, M. Galley, and M. Mitchell. A Survey of Current Datasets for Vision and Language Research. In *EMNLP*, 2015.

[10] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *EMNLP*, 2016.

[11] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering. In *NIPS*, 2015.

[12] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *CVPR*, 2017.

[13] M. Gygli, Y. Song, and L. Cao. Video2GIF: Automatic Generation of Animated GIFs from Video. In *CVPR*, 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[15] P. Isola, J. J. Lim, and E. H. Adelson. Discovering States and Transformations in Image Collections. In *CVPR*, 2015.

[16] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *CVPR*, 2017.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li. Large-Scale Video Classification with Convolutional Neural Networks. In *CVPR*, 2014.

[18] J.-H. Kim, S.-W. Lee, D.-H. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal Residual Learning for Visual QA. In *NIPS*, 2016.

[19] D. P. Kingma and J. L. Ba. ADAM: A Method For Stochastic Optimization. In *ICLR*, 2015.

[20] K. Kipper-Schuler. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. PhD thesis, UPenn CIS, 2005.

[21] O. Levy and L. Wolf. Live Repetition Counting. In *ICCV*, 2015.

[22] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *CVPR*, 2016.

[23] X. Lin and D. Parikh. Don't Just Listen, Use Your Imagination: Leveraging Visual Common Sense for Non-visual Tasks. In *CVPR*, 2015.

[24] M. Malinowski and M. Fritz. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In *NIPS*, 2014.

[25] M. Malinowski, M. Rohrbach, and M. Fritz. Ask Your Neurons: A Neural-based Approach to Answering Questions about Images. In *ICCV*, 2015.

[26] J. Pennington, R. Socher, and C. D. Manning. Glove - Global Vectors for Word Representation. In *EMNLP*, 2014.

[27] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour. Dropout Improves Recurrent Neural Networks for Handwriting Recognition. In *ICFHR*, 2014.

[28] M. Ren, R. Kiros, and R. Zemel. Exploring Models and Data for Image Question Answering. In *NIPS*, 2015.

[29] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie Description. *IJCV*, 2017.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F.-F. Li. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[31] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. In *ICML*, 2015.

[32] I. Sutskever, O. Vinyals, and Q. Le. Sequence to Sequence Learning with Neural Networks. In *NIPS*, 2014.

[33] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *CVPR*, 2016.

[34] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*, 2015.

[35] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to Sequence – Video to Text. In *ICCV*, 2015.

[36] Q. Wu, C. Shen, L. Liu, A. Dick, and A. van den Hengel. What Value do Explicit High Level Concepts Have in Vision to Language Problems? In *CVPR*, 2016.

[37] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, 2015.

[38] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing Videos by Exploiting Temporal Structure. In *ICCV*, 2015.

[39] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image Captioning with Semantic Attention. In *CVPR*, 2016.

[40] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual Madlibs: Fill in the Blank Description Generation and Question Answering. In *ICCV*, 2015.

[41] Y. Yu, H. Ko, J. Choi, and G. Kim. End-to-end Concept Word Detection for Video Captioning, Retrieval, and Question Answering. In *CVPR*, 2017.

[42] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7W: Grounded Question Answering in Images. In *CVPR*, 2016.