Video Question Answering with Iterative Video-Text Co-Tokenization

AJ Piergiovanni, Kairo Morton, Weicheng Kuo, Michael S. Ryoo, and Anelia Angelova

Google Research
ajpiergi@google.com

Abstract. Video question answering is a challenging task that requires understanding jointly the language input, the visual information in individual video frames, as well as the temporal information about the events occurring in the video. In this paper, we propose a novel multi-stream video encoder for video question answering that uses multiple video inputs and a new video-text iterative co-tokenization approach to answer a variety of questions related to videos. We experimentally evaluate the model on several datasets, such as MSRVTT-QA, MSVD-QA, IVQA, outperforming the previous state-of-the-art by large margins. Simultaneously, our model reduces the required GFLOPs from 150-360 to only 67, producing a highly efficient video question answering model. [1]

Keywords: video question answering, video-text joint learning, video understanding, efficient vision models

1 Introduction

Video Question and Answering (VideoQA) [30, 99, 49, 38, 82] targets the challenging problem of answering a variety of questions about a video, which requires both natural language understanding and understanding of the video content, including reasoning about activities, objects, sequence of events, and interactions within the video. VideoQA is the video counterpart of Visual Question and Answering (VQA) [14, 33, 2, 25], a long-standing task in computer vision of answering questions towards an image. VideoQA is a very important multi-modal visual-language task for natural interaction with videos, aiming to understand what is happening in a video with the help of text-based specification (Figure 1). It can satisfy information needs from videos and allow for rich user engagement such as searching for highlights, events, objects, or specific scenes in a video, e.g., "Where is the first goal scored in the game?", "How many goals were scored?" "Why was the umpire ruling considered controversial?".

VideoQA has the inherent challenges of VQA tasks: it needs to understand the visual and language inputs and how they relate to each other. Additionally, VideoQA, needs to address multiple challenging video understanding tasks, such as action recognition, action detection and segmentation [62, 9, 44], but unlike

---
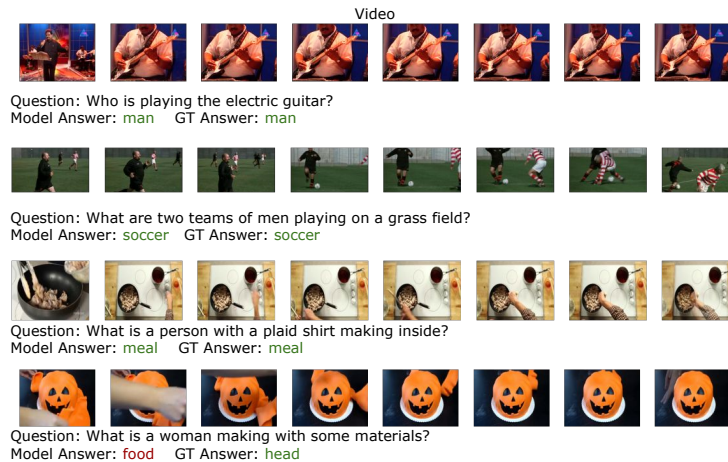[1] Code: https://sites.google.com/view/ videoqa-cotokenization

Fig. 1: We consider the challenging task of Video Question Answering which is a multi-modal information-seeking task, where natural language questions or tasks specifications are issued towards a video. The answers are in natural language open-vocabulary text. Example VideoQA outputs of our model are shown.

them, needs to work in the open-set domain[2], where questions can be issued about unseen object categories or unknown activities. VideoQA needs a deeper understanding of the video input to begin with, which requires adequate spatio-temporal understanding.

VideoQA faces other challenges too: it requires the ability to process larger visual inputs e.g., 30-100x the number of frames than VQA, and to answer much harder questions, for example, why a certain event is happening, or what has happened before a certain action, where the timeframes might span small or large portions of the video. Furthermore, VideoQA inherits the efficiency challenges of video processing, which very few of the prior approaches have addressed.

The visual question and answering problem has been explored by many in the image and video research domains [2, 88, 3, 46]. However, many previous works with regards to this problem utilize multiple backbone feature extractors for individual image frames, for the video input and for the text input separately and apply cross-attention between modalities as an after thought following feature extraction [47, 88]. However, we believe, and will show in this paper, that the interaction between these modalities during the visual feature extraction process is key in attempting to solve this problem. We propose to jointly learn the video and language representations, where their interaction allows reasoning across these two modalities. Furthermore, by viewing question text not just as additional information or as a part of the task, but as the lens through which visual

---

[2] Unlike standard video understanding tasks, e.g., action recognition, action segmentation, where the set of action classes is pre-defined and known, the open-set VideoQA involves answering natural questions about novel or unseen actions and/or objects.

data is understood, it is clear that these cross-modality interactions must occur at various stages throughout the process of extracting visual features. Specifically, we propose a novel video-text iterative co-tokenization approach which learns efficient joint representations iteratively[3].

Our work effectively demonstrates the benefits of using text during the video understanding tasks for a variety of open ended VideoQA tasks and shows improvements over the state-of-the-art (SOTA) on MSVD-QA [83], MSRVTT-QA [35], and IVQA [88] datasets. Extensive ablation experiments confirm the benefit of each component. Our paper makes the following contributions:

- Novel video-language interaction learning for videos and text, especially focusing on learning both spatial and temporal features, which outperforms several SOTA on several VideoQA datasets.
- Novel multi-stream video encoder with iterative video-text co-tokenization which uses multiple inputs and iteratively selects efficient features
- An efficient approach, greatly reducing the FLOPs over baselines, which is important to save compute for video methods and allows scalability.

2   Related work

Video understanding. Video understanding is a fundamental visual recognition task, conducted in video inputs [31, 70, 65, 9, 81, 96, 71, 20, 79, 29, 8, 64, 63, 40, 102, 51]. What makes it challenging is the joint processing of spatial and temporal information and the sheer volume and diversity of visual inputs. Video understanding tasks include action classification, action detection, object segmentation in videos, etc. Text is not used in the training and recognition process.

Video and language. Using language alongside videos [45, 12] has opened several interesting problems in multi-modal video-language learning and language-guided video understanding, for example, VideoQA [30, 99], video captioning [28, 41, 15, 98], text-to-video retrieval [84, 61, 45, 97, 6, 32, 76], referring expression comprehension for videos [7, 26, 37] and others [26, 23, 74]. In addition to the challenges of the video understanding tasks, video+language tasks bring in their own challenges of understanding text in the context of video, analyzing the video content according to the text input, or in some tasks, natural language text generation.

Previous video-language methods use pre-training from separate video models and text models which are typically pre-trained on disjoint datasets [66, 21, 100, 47, 58, 52, 89]. Pre-training video and text jointly has been shown to be very beneficial to a number of downstream tasks [57, 100]. End-to-end joint training with multi-modal inputs from the target datasets is also gaining popularity recently [45]. In the above-mentioned approaches, Transformer-based models [73], adapted to videos are often used to join the two modalities [66, 100], e.g., with masked-learning or other objectives [98, 54]; in other works, standard representations e.g., I3D, S3D [9, 81] for video and word2vec embeddings for text, have also been explored for joint training [57].

---

[3] We consider video inputs of 32 frames which spans up to 10 seconds of video.
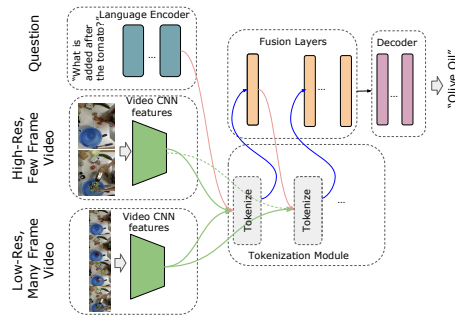
Fig. 2: Main architecture. Video features interact with text features. This is done efficiently using multi-stream video encoder to learn spatial and temporal features fusion with text features. Iterative co-tokenization fusion module learns a reduced number of useful tokens by progressively using the fused video-language representation to affect the next tokenization (red arrows). This results in a very efficient approach as compact tokenized representations are learned throughout. The decoder, which produces the answer, is a standard text-generation decoder.

VideoQA. Video Question and Answering covers a broad range of questions towards a video which require both understanding of the video input and the input text [30, 99, 49, 38, 82, 3, 55, 3, 85, 24, 94, 91, 101, 80, 75, 46, 68, 48, 88]. Several research methods have been developed for VideoQA [4, 42, 99, 27, 43, 90, 18, 39, 72, 59, 34, 49, 69, 87, 10]. Large-data pre-training is commonly used for VideoQA [100, 57]. ClipBert [45] propose end-to-end learning of video and language tasks, which are typically tested on text-to-video retrieval and VideoQA [30]. Kim et al. [38] propose multi-modal question and answering where additionally a text input such as captioning can be included in the input.

Video-language cross-modal learning. Image-language cross-modal learning has been developed for cross-modality retrieval, e.g., initially for images and text [93, 16] and for videos and text [86, 17]. These approaches are followed by using the popular transformer architecture [73] to encode separately each modality into a joint space, applied to both image-text [56] and to video-text retrieval [66, 58, 56]. A similar transformer-based architecture is commonly used for learning jointly image and language features via joint attention or cross-modal co-attention modules [54, 11, 67, 77, 50, 92] This is also common in video-language joint learning, e.g., [100, 45]. In the context of VideoQA, Zhu et al. [100] use the transformer architecture with pre-computed bounding boxes from an object detector, the features of which are fed to transformer layers. Hero [47] propose cross-modal transformer for multi-modal fusion based on Uniter [11]. Fan et al. [18] also propose a multimodal fusion layer. In complement to these cross-modal approaches, ours specifically learns efficient representations to reflect the interactions of these modalities at each level of abstraction of feature learning. Our multistream formulation is also conceptually related to SlowFast [20] for videos and multi-stream object detection networks for object understanding, where different network streams were utilized to look at different resolutions.
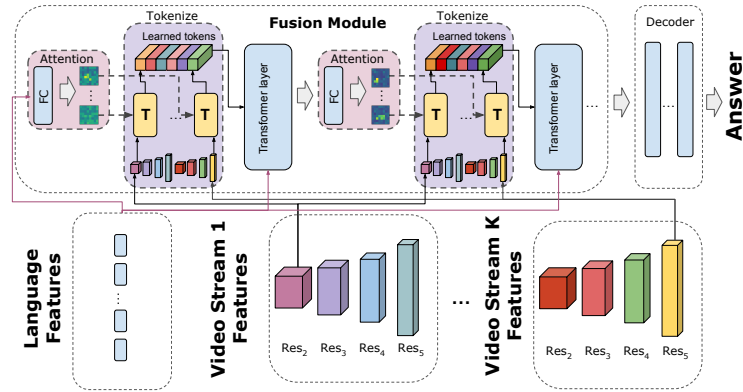
Fig. 3: Illustration of the iterative fusion module in detail. The features are used to generate N attention maps, which are applied to each stream and (optionally) scale of the video features. This results in a fixed number of tokens for the K streams and S scales (K and S are typically very small). The tokens are fed through a transformer layer, generating a new feature, which is used to generate new attention maps. This processes is repeated multiple times. The yellow boxes (denoted as 'T') represent the application of an attention map to a feature map.

3 Approach

3.1 Multi-Stream Video Encoder Overview

Our approach to the challenging VideoQA task, is video-language learning where early visual and language features are jointly learned. Specifically, we propose a multi-stream video encoder that is capable of efficiently learning both spatial and temporal features, and a new fusion method that can adaptively combine the video and text features. Our multi-stream encoder features the following components: First, the video features are extracted forming multiple learnable video representation inputs (Section 3.2); the text input is also preliminarily encoded. Secondly, a condensed representation for each input stream is learned via learned tokenization (Section 3.3). Importantly, the features are combined by an iterative video-text co-tokenization fusion mechanism which learns the most appropriate compact feature representations iteratively based on the previous features (Section 3.4). Figure 2 gives an overview of the main architecture.

At a high-level, our approach can be thought of as an encoder-decoder structure, however here the multi-stream video encoder is tasked with interdependent, video-language feature learning, which produces compact and efficient features. A text generation decoder directly outputs natural language free-form text.

3.2 Video understanding at different timeframes

Actions and events in videos span a wide range of timeframes. Questions about videos also require different time scales. For example, 'What color is the apple?' only requires understanding a single frame, while 'What happens after cutting

the apple?' requires localizing two specific actions segments in the video in order to answer, and 'What is being made in the video?' requires understanding the entire video. Being able to answer all these questions, with a model that is computationally efficient is a challenge, as, unlike images, the model must consider features of large video inputs with multiple spatial and temporal resolutions.

To address this, we propose a multi-stream approach with fusion, that extracts video features at various time and space scales. We consider them jointly together, as well as with the text/question input.

Video features sub-inputs. The video encoder, V, takes a video as input, e.g., a $x = T \times H \times W$ tensor where T is the number of frames and H and W are height/width of the image frame. It processes the video, producing the output features f_v. The baseline model uses a single video input, while the multi-stream model takes the video at different space- and time-scales as input. For example, one stream can take many frames at low spatial resolution, while another stream can take few frames at high spatial resolution. We can here use any number of video streams, each learning different spatio-temporal features. Importantly, subsequent components of the multi-stream visual encoder will enable learning various inter-relations of these inputs. This is critical for VideoQA, since the question can refer to the full video or specific spatial or temporal segments in it, or require the comprehension of events or actions in the video across different duration timeframes, or be very specifically pinpointed in time and space.

The features from the multi-stream visual encoder can further be mutli-scale, taking features from different points in the network, e.g., after each residual block in X3D. We denote the output(s) of the multi-stream encoder as $f_v = V(x)$, where f_v in general a set of multi-scale features over multiple feature levels: $f_v = \{f_{vi}\}_{i=1}^{S}$ where S is the total number of video features from different streams.

3.3   Learning to tokenization

One important aspect for fusing multi-modal models is to achieve effective representations of each modality but also do so efficiently. A possible naive approach is to apply either global average pooling, reducing the entire vision stream to a single representation, or concatenate many frames together. The first averages all the spatio-temporal information into a single representation, while the latter maintains a lot of redundant information and is computationally expensive.

Instead, we here first 'compress' each modality by learning to tokenize, based on TokenLearner [63]. The advantage is that it learns to select a small set of tokens, conditioned on the inputs. This greatly reduces the number of spatio-temporal tokens, while learning to maintain the needed information. Due to its adaptive nature, it can change the tokens based on the video and text inputs, allowing it to focus on the most important information.

We modify the TokenLearner approach to better suit the problem here, since in [63] it is only applied to single image frames. In our case, we also have both videos and text as our inputs, and videos are of multiple resolutions. We here enable both video and text to condition how video tokens are learned. Intuitively,

this will allow TokenLearner to better select important spatio-temporal regions in the video not only based on visual information itself but also based on the text, and adaptively tokenize such regions.

Specifically, we learn to tokenize by first taking the input representation r with shape $L \times F$, where L is the length of the sequence and F is the feature dimension. This initially is the text feature, $r = f_t$. We also take a specific video feature f_{vi}, where i is one of the multi-scale, multi-stream features which has shape $T \times H \times W \times C$. Given r, we learn $\phi(r)$ (implemented with a linear layer in our version), to produce a $L \times (T \cdot H \cdot W)$ tensor where $T \cdot H \cdot W$ is the temporal and spatial size of the video feature. Another linear layer is used to make the feature have shape $C \times (T \cdot H \cdot W)$. This is reshaped and added with the video feature: $f = \phi(r) + f_{vi}$. Next, the function $\psi(\cdot)$ (implemented with convolutional layers) is applied to convert f into a feature with shape $T \times H \times W \times N$, where N is the number of desired tokens. A softmax function (σ) is applied over this, along the N-axis, selecting the spatio-temporal features for each token. Enabling the spatial attention mechanism, this is multiplied with the video representation, f_{vi}. More specifically, the attention mask $\sigma(\psi(\phi(r) + f_{vi}))$ is transposed to shape $N \times T \times H \times W$, and is tensor-dot-producted with f_{vi} while treating $T \times H \times W$ as the dimensions to contract. Overall, this can be expressed as:

$$f_0^i = \sigma(\psi(\phi(r) + f_{vi})) \cdot f_{vi} \tag{1}$$

The resulting feature representation f_0^i will have a shape of $N \times C$, abstracting the entire video as a set of N tokens per each video feature f_{vi}.

This allows the tokenizations to reduce many video streams into a few tokens. Generally, for our multi-stream, multi-scale inputs, the final feature representation f_0 is obtained by concatenating all f_0^i in the first axis, forming a representation with shape of $(NS) \times C$ where S is the number of scales and streams that are used. Importantly, these adaptive tokens are learned according to the optimization loss of the final task, which aims to improve accuracy of the produced outputs or answers. These tokens are then fused with a learning mechanism described in the below subsections.

3.4   Video-text iterative co-tokenization

We here describe how we apply the above-mentioned joint tokenization approach iteratively at various levels of feature abstraction. We utilize self-attention transformer layers to combine the text and video features. Different from previous works, we will use the features of the transformer layers to produce a few informative tokens from the data (blue arrows on Fig. 2), which are fed to the next transformer fusion layer (red arrows on Fig. 2). Importantly, we use the token learning mechanism presented above at every layer; This allows the model to change its (video feature) selections differently at different layers.

To do this, we start by using tokenization to select NS visual tokens from the video input, as described above, initially using the text feature. These are then concatenated with the encoded text representation: $[f_t, f_0]$ along the token axis. These are then passed through 1 transformer layer (H), $r_1 = H([f_t, f_0])$.

The outputs, r_1 are then used as input to Eq. 1 to generate NS new tokens, f_1, which are again concatenated with the text representation, added to the previous encoded, and passed through the next transformer layer: $r_l = H([f_t, f_{l-1}] + r_{l-1})$ for $l > 1$. The tokenziation is done for each of the multi-stream, multi-scale features f_{vi}, and is repeated L times, where L is the number of transformer layers (Fig. 3).

This approach allows the model to adaptively and iteratively select different visual features, from multiple scales and streams, refining the input to best align with the text. It results in a highly efficient method, due to the iterative tokenization.

3.5 Implementation details

For better comparison to SOTA, we use standard model components, the T5 language model [60], and for video, we use popular video representations: 3D ResNets, 2D ResNets + temporal pooling, and X3D [19]. The encoded text and video features (learnable end-to-end) are entered into the multi-stream video encoder for learning the interaction between these features. A language-based decoder, again T5 [60], then takes the fused features to generate the output text. We set $N = 8, S = 4, C = 768; K = 3$

Video Preprocessing: To preprocess the video data, each individual frame is resized a fixed size, e.g., 224 by 224 pixels, and normalized such that the value for each pixel ranges from -1 to 1. We sample T frames from each video, evenly spaced across the video, i.e., the frames-per-second per video varies to maximize the temporal extent. In the multi-stream setting, T and $H \times W$ are different for each stream. We describe these settings for the models below.

Text Preprocessing: The text is tokenized using T5's standard 32k word vocabulary, with a max length of 32 tokens per example for both input and output. The model is trained to minimize the output per-token cross-entropy.

4   Experiments

We conduct experiments across different VideoQA datasets in order to determine the benefits of the approach, pretraining, efficiency and scaling.

4.1   Datasets

IVQA The IVQA dataset [88] is a new, human annotated dataset for VideoQA consisting of 'how-to' videos. It has 10k clips with 1 question and 5 answers per question. It follows an evaluation metric similar to the VQA2.0 dataset [2], where its accuracy is computed for 5 choose 4 ground-truth (GT) answers.

MSRVTT-QA [82] is based on the MSRVTT descriptions dataset [35], with automatically generated QA pairs from the descriptions. It has 243k VideoQA pairs and is evaluated by answer accuracy.

MSVD-QA [82] is based on the MSVD datasets with automatically generated QA pairs. It has 50k VideoQA pairs and is evaluated by answer accuracy.

TGIF-QA [30] consists of short GIF video clips with accompanying questions and answers related to the video. There are four types of questions ranging from questions about objects in still frames, to activities, repetition counting and

Table 1: MSRVTT-QA [35] and MSVD-QA [83] datasets. Accuracy (%).

Model	MSRVTT-QA	MSVD-QA	GFLOPs
ST-VQA [30]	31.3	30.9	-
Co-Mem [22]	31.7	32.0	-
AMU [82]	32.0	32.5	-
HME [18]	33.7	33.0	-
HRA [13]	34.4	35.1	-
HCRN [43]	36.1	35.6	-
ClipBERT [45] 8x2	37.4	-	-
OCRL+LOGNet [53]	38.2	36.0	-
VQA-T [88] with HowTo100M	40.4	43.5	75+
VQA-T [88] with HowToVQA69M	41.5	46.3	75+
SiaSamRea [89]	41.6	45.5	-
MERLOT[95] with YT-Temporal-180M* [95]	43.1	-	-
Ours with HowTo100M	**45.7**	**48.6**	67

Table 2: Results comparing to SOTA approaches on the IVQA [88] dataset.

Model	Pre-training dataset	Accuracy (%)	GFLOPs
VQA-T [88]	HowTo100M	28.1	-
VQA-T [88]	HowToVQA69M	35.4	-
Ours	HowTo100M	**38.2**	67

sequences of events. TGIF-QA is evaluated in the multiple-choice answer setting, which we show is trivial for a strong (and language-only) model (see supp.). Instead, we use TGIF-QA in an open-vocabulary generative setting to study the effects of our approach on both single-frame and temporal-based questions.

4.2 Evaluation setup

We follow the standard metrics for each dataset, described above. Since our model is generative, we use three different evaluation settings. (1) Open-ended generation, where the text decoder is used as-is with beam search [78], and generates any text. We then check string equality of the generated text to the GT answer. (2) Vocabulary-specific fully-connected (FC) layer, where we use the target vocabulary for each dataset from [88], and train a new FC layer on top of the final language features to classify the answer. (3) Masked-vocabulary generation, where we keep the text decoder from the model, but mask out tokens not in the target vocabulary. This lets us preserve the learned token embedding, but restrict the vocabulary to that of previous works, so as to be directly comparable.

5 Experimental Results

We compare our model with the SOTA approaches (Section 5.1) and then examine the model in a number of ablations. Section 5.4, reports on model efficiency.

Video

Question: What does the person on left do before taking the other to the ground?
Model Answer: kick   GT Answer: hang on someone's body

Question: What does man one toss to the other?
Model Answer: towel   GT Answer: towel

Question: What is a person doing?
Model Answer: mixing   GT Answer: stirring

Question: What is the man pouring antifreeze into??
Model Answer: car   GT Answer: car

Fig. 4: Example results of our method.

5.1 VideoQA results. Comparison to SOTA

Table 1 shows our results comparing to the MSRVTT-QA and MSVD-QA datasets, which are commonly used for VideoQA evaluation. As seen, our results outperform the SOTA, even though our pretraining is done on the weaker captioning dataset (HowTo100M), instead of the VideoQA counterpart, HowToVQA69M, which has been shown to be superior for VideoQA tasks [88].

Table 2 compares our approach on the challenging new IVQA dataset. As seen, our results outperform SOTA with both HowToVQA69M and HowTo100M. Importantly, with the same pre-training HowTo100M, our approach outperforms the SOTA by more than 10% in absolute values.

In Table 1 of the supp. material, we show a surprising but instructive result on TGIF-QA, where our method, using a medium size pretrained text model T5 (T5-Base) [60], is able to accomplish close to 100% results on the multiple choice questions. This is due to the fact that the limited selection of answers is easy to guess even without video (it performs randomly for the action counting category as it needs the video input for these). We acknowledge that this contemporary text model is stronger than text models used previously, but is important to note that the multiple-choice setting is too easy. We experiment with the harder open-vocabulary setting in our ablations. The datasets evaluated above, MSRVTT-QA, MSVD-QA, IVQA are open-ended and thus do not suffer from this problem.

Visualizations. In Fig. 4 we show example results of the approach. In Fig. 5, we visualize the learned attention maps for different types of questions. In Fig. 6, we see a set of attention maps for each transformer layer. These figures show that the model is adapting the tokens based on both the video and text.

5.2 Open-vocabulary answer generation

In the previous SOTA results (as presented in Section 5.1), prior approaches still use a limited vocabulary of answers, e.g., 4000 answers [88]. However in

real-life scenarios, it is desirable that the generated answers are free-form text. In this section we show that generating open-vocabulary answers is a much more challenging setting for VideoQA. With our experiments, we would like to encourage future results to report this setting, as well.

In Table 3 we compare the open-ended, vs. FC vocabulary vs. masked vocabulary. The open vocabulary is the most challenging, as it requires the model to generate (in free form) the exact sequence of tokens to match the GT word. The FC vocabulary is restricted to 4k tokens, matching [88] vocabulary, but this model throws away the learned token embeddings, losing information. The masked vocabulary preserves all the token information, but removes the unneeded tokens, closely matching the previous settings, while maintaining the learned features.

Table 3: Comparing the performance of the same model with a fixed vocabulary (as reported in SOTA) and with open vocabulary, which is more challenging.

Vocabulary / Dataset	IVQA	MSRVTT-QA	MSVD-QA
Open 32k vocabulary	21.4	33.7	32.5
Fixed 4k vocabulary (FC)	37.4	42.9	45.9
Fixed 4k vocabulary (Masked)	**38.2**	**45.7**	**48.6**

5.3   Fusion techniques

We conduct ablations on the multi-stream video encoder itself and its fusion mechanisms. Table 4 has the results – open vocabulary and no pre-training are used in these experiments. We see that each of the components of multi-stream video encoder contribute, where we note large collective contributions of the novel tokenization-based fusion techniques and also improvements compared to a single-stream model. Our approach is also very efficient, despite being multi-stream, due to the iterative tokenization approach and efficient backbones. Note that multiple frame models hurt the single-frame QA setting, but greatly benefit the time-based questions. Further, we see the most gains from the approach in the multi-frame questions, showing the benefit of the approach.

5.4   Video multi-stream encoder model efficiency

In Table 5 we experiment with the effect of scaling the model to more streams and larger models and see that our approach is much more efficient, in addition to being accurate, even with more than one streams. Our multi-stream approach, with only 67 GFLOPs, allows using various model sizes to improve performance and save FLOPs/params. We also note that to our estimates, the popular video vision transformer-based model ViViT [5] requires 2010 GFLOPs if adapted to VideoQA with T5, which would be infeasible both for training or inference (ViViT does not report VideoQA results). We used 64 TPUs for 72 hours to

pretrain (4608 TPU hours) and 4 TPUs for 8 hours to finetune (32 TPU hours). Overall, this is fairly modest, e.g., compared to MERLOT [95] which used 1024 TPUs for 30 hours (30k TPU hours) for pretraining.

Table 4: Ablations on multi-stream video encoder and fusion techniques. These models are trained from scratch and in the open-vocab generative setting. The ablations are cumulative, e.g., the last row uses 2-stream + Tok + MS + Co-Tok. Adding each component benefits. Tokenization reduces FLOPs notably and brings small performance improvements, whereas multi-scale and iterative co-tokenization bring larger improvements for very modest additional FLOPs. These models were trained with 32 frames, 224x224 images. The two streams are 32x224x224 and 32x128x128.

Model	GFLOPs	TGIF Frame-QA Single frame	TGIF Action (What happens X times?)	TGIF Trans. (What happens after X?)	IVQA	MSRVTT QA
Single-frame	–	24.4	0.7	1.5	8.4	7.2
Single-stream	150	21.5	8.2	9.2	14.2	24.8
2-stream	47	24.2	8.8	9.2	14.5	24.7
+ Transformer	49	24.5	9.1	10.9	14.4	25.3
+ Tokenization	40	24.7	9.7	11.6	14.9	25.5
+ Multi-Scale	41	26.2	11.5	12.2	15.2	26.2
+ Iterative Co-Tok.	42	**27.3**	**11.8**	**12.5**	**15.5**	**27.6**

Table 5: Efficiency comparisons. While we use multiple streams, they take much fewer FLOPs. They also outperform the strongest X3D-XL model. Open vocabulary setting, no pre-training. 2-str (X3D-S 8x224x224, X3D-M 16x112x112), 3-str (X3D-S 8x224x224, X3D-M 16x112x112, X3D-M 32x64x64), 3-str (3x X3D-L 8x224x224, 16x112x112, 32x64x64). Note that these models use fewer frames than the ones in Table 4.

Model	IVQA	MSRVTT-QA	MSVD-QA	GFLOPS	Params
X3D-S	9.4	24.8	22.4	82	311M
X3D-XL	10.3	27.8	23.2	150	380M
2D-RN-50	2.2	6.4	6.5	306	332M
3D-RN-50	8.9	24.4	23.2	362	341M
2-stream (X3D-S, X3D-M)	9.2	25.3	23.5	40	321M
3-stream (X3D-S, X3D-M, X3D-M)	10.3	28.2	23.8	42	335M
3-stream (3x X3D-L)	**12.4**	**30.5**	**25.7**	67	345M

5.5   Pretraining

In Table 6 we explore different versions of pretraining (PT): contrastive, generative, Kinetics [36] classification, etc. for the VideoQA task. The models are pretrained using either HowTo100M with the automatic captions, YouTube8M [1] with automatic captions (when YouTube8M is split into 5 second clips, similar to HowTO100M clip duration, it has 250M clips, about 2x the size of HowTo100M). We pretrain the models using contrastive training (e.g., [58]) or language-generative training, in a completion setting where half the caption is

Table 6: Comparing different pretrianing methods for the VideoQA task. All use the single-stream baseline X3D-M model.

Method	IVQA	MSRVTT-QA	MSVD-QA
Random init	7.2	8.3	4.8
Kinetics-600 + T5	10.8	26.8	23.2
Contrastive (YouTube8M)	10.8	25.8	26.2
Contrastive (HowTo100M)	11.0	26.2	25.5
Generative (YouTube8M)	15.2	27.8	28.2
Generative (HowTo100M)	**15.6**	**29.4**	**28.6**

used as input and half used as the target text. We also compare to Kinetics 600 classification pretraining of the video model and using the pretrained T5 model [60] for text. We find HowTo100M to be strongest of these pretrainings, and for our model, the generative training was better than contrastive. Both were generally better than independent Kinetics + T5 pretraining.

5.6   Exploring the effects of temporal features

In Table 7 we compare the effects of temporal features on TGIF-QA and MSRVTT-QA, IVQA. We compare the ResNet 1-frame, 16-frame and 32-frame model, as well as the X3D video models. We can see that, for single frame questions, the 2D ResNet does better. However, for temporal questions, X3D is far superior. This further confirms the benefit of multi-stream models for VideoQA tasks.

Table 7: Comparing temporal features. Here the ResNet is pretrained on image-text data, e.g., CC12M. We find that single-frame PT ResNet does very well on the single frame QAs, but quite poorly on the temporal questions. Adding frames help, but in general, we find that learning video specific features helps the most. 2D ResNets are shown in the top half and X3D-M in the bottom.

Model	TGIF Frame-QA (Single frame)	TGIF Action (What happens X times?)	TGIF Transition (What happens after X?)	IVQA	MSRVTT-QA
1 frame	37.4	0.8	2.9	12.7	22.3
16 frame	35.4	2.5	3.5	18.6	23.5
32 frame	36.2	3.3	3.9	19.3	23.9
16 frame	26.2	12.4	13.4	13.5	24.5
32 frame	25.5	13.1	15.4	14.2	24.8

6   Conclusions

In this paper we propose a novel multi-stream video text encoder with iterative video-text co-tokenization, which efficiently extracts and fuses information

from video and text inputs. We demonstrate its benefits to challenging VideoQA tasks, outperforming SOTA on the standard benchmarks and metrics and also report results in the more challenging open-vocabulary setting. Our approach is efficient taking only 67 GFLOPs. One limitation of the approach is for longer answer. Though our model is able to generate longer answers as it is generative, the challenge is the evaluation metric. The metric we use is string equality for accuracy. However, for long answers this will be quite hard.
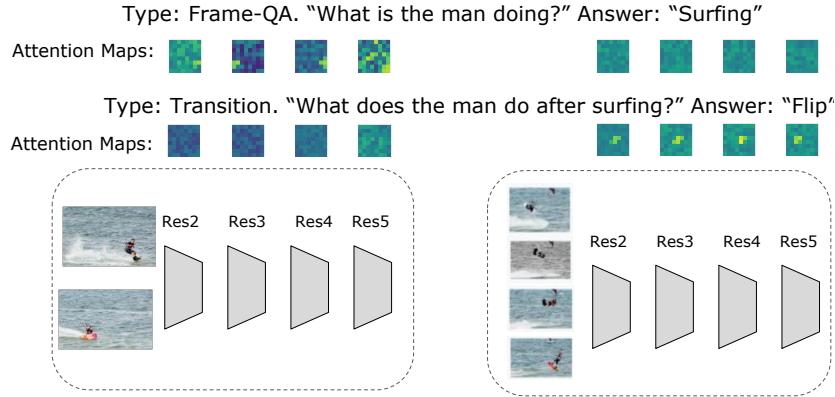
Fig. 5: Visualization of the learned attentions for two different types of questions. We can see it learns to select the spatial stream for the "Frame-QA" type of question, where temporal information is not needed, but for the "Transition" question, it focuses more heavily on the stream with many frames.
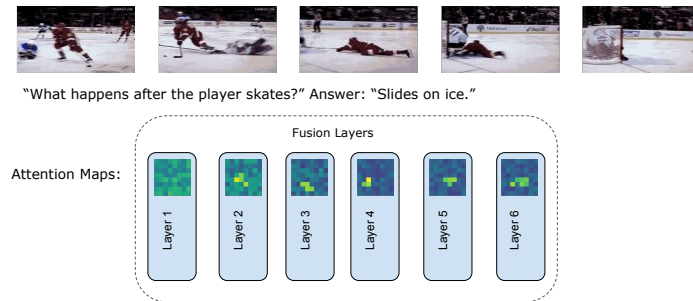

Fig. 6: Attention maps changing after each layer they are applied to. The first one focuses over the whole video, each after focuses on more specific regions. In this example, it captures the hockey player falling and sliding on the ice.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)
2. Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C.L., Batra, D., Parikh, D.: Vqa: Visual question answering. In: ICCV (2015)
3. Alamri, H., Cartillier, V., Das, A., Wang, J., Cherian, A., Essa, I., Batra, D., Marks, T.K., Hori, C., Anderson, P., Lee, S., Parikh, D.: Audio-visual scene-aware dialog. In: CVPR (2019)
4. Anna Rohrbach, Atousa Torabi, M.R.N.T.C.P.H.L.A.C., Schiele, B.: Movie description. In: IJCV (2017)
5. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lucic, M., Schmid, C.: Vivit: A video vision transformer. In: ICCV (2021)
6. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: ICCV (2021)
7. Bellver, M., Ventura, C., Silberer, C., Kazakos, I., Torres, J., i Nieto, X.G.: Refvos: A closer look at referring expressions for video object segmentation. In: https://arxiv.org/abs/2010.00263 (2020)
8. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? (2021)
9. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
10. Chadha, A., Arora, G., Kaloty, N.: iperceive: Applying common-sense reasoning to multi-modal dense video captioning and video question answering. In: WACV (2021)
11. Chen, Y.C., Li, L., Yu, L., Kholy, A.E., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Universal image-text representation learning. In: ECCV (2020)
12. Chen, Z., Ma, L., Luo, W., Wong, K.Y.K.: Weakly-supervised spatio-temporally grounding natural sentence in video. In: In Proc. 57th Annual Meeting of the Association for Computational Linguistics (2019)
13. Chowdhury, M.I.H., Nguyen, K., Sridharan, S., Fookes, C.: Hierarchical relational attention for video question answering. In: 25th IEEE International Conference on Image Processing (ICIP)
14. Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M.F., Parikh, D., Batra, D.: Visual dialog. In: CVPR (2017)
15. Deng, C., Chen, S., Chen, D., He, Y., Wu, Q.: Sketch, ground, and refine: Top-down dense video captioning. In: CVPR (2021)
16. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
17. Dong, J., Li, X., Xu, C., Ji, S., He, Y., Yang, G., Wang, X.: Dual encoding for zero-example video retrieval. In: CVPR (2019)
18. Fan, C., Zhang, X., Zhang, S., Wang, W., Zhang, C., Huang, H.: Heterogeneous memory enhanced multimodal attention model for video question answering. In: CVPR (2019)
19. Feichtenhofer, C.: X3D: expanding architectures for efficient video recognition. In: CVPR (June 2020)
20. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV (2019)

21. Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: ECCV (2020)
22. Gao, J., Ge, R., Chen, K., Nevatia, R.: Motion appearance co-memory networks for video question answering. In: CVPR (2018)
23. Gao, J., Sun, C., Yang, Z., Nevatia, R.: Tall: Temporal activity localization via language query. In: ICCV (2017)
24. Garcia, N., Otani, M., Chu, C., Nakashima, Y.: Knowit VQA: Answering knowledge-based questions about videos. In: AAAI (2020)
25. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017)
26. Hendricks, L.A., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: ICCV (2017)
27. Hori, C., Alamri, H., Wang, J., Wichern, G., Hori, T., Cherian, A., Marks, T.K., Cartillier, V., Lopes, R.G., Das, A., Essa, I., Batra, D., Parikh, D.: End-to-end audio visual scene-aware dialog using multimodal attention-based video features. In: ICASSP (2019)
28. Huang, G., , B.P., Zhu, Z., Rivera, C., Soricut, R.: Multimodal pretraining for dense video captioning. In: AACL-IJCNLP (2020)
29. Hussein, N., Gavves, E., Smeulders, A.W.: Timeception for complex action recognition. In: CVPR (2019)
30. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In: CVPR (2017)
31. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition **35**(1), 221–231 (2013)
32. Jianfeng Dong, X.L., Xu, C., Yang, X., Yang, G., Wang, X., Wang, M.: Dual encoding for video retrieval by text. In: T-PAMI (2021)
33. Jiang, H., Misra, I., Rohrbach, M., Learned-Miller, E., Chen, X.: In defense of grid features for visual question answering. In: CVPR (2020)
34. Jiang, P., Han, Y.: Reasoning with heterogeneous graph alignment for video question answering. In: AAAI (2020)
35. Jun Xu, Tao Mei, T.Y., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: CVPR (2016)
36. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The Kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
37. Khoreva, A., Rohrbach, A., Schiele, B.: Video object segmentation with language referring expressions. In: 14th Asian Conference on Computer Vision (ACCV) (2018)
38. Kim, J., Ma, M., Pham, T., Kim, K., Yoo, C.D.: Modality shifting attention network for multi-modal video question answering. In: CVPR (2020)
39. Kim, K.M., Choi, S.H., Kim, J.H., Zhang, B.T.: Multimodal dual attention memory for video story question answering. In: ECCV (2018)
40. Korbar, B., Tran, D., Torresani, L.: Scsampler: Sampling salient clips from video for efficient action recognition. In: ICCV (2019)
41. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: ICCV (2017)
42. Kyung-Min Kim, Min-Oh Heo, S.H.C., Zhang, B.T.: Deepstory: Video story QA by deep embedded memory networks. In: IJCAI (2017)
43. Le, T.M., Le, V., Venkatesh, S., Tran, T.: Hierarchical conditional relation networks for video question answering. In: CVPR (2020)

44. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: CVPR (2017)
45. Lei, J., Li, L., Zhou, L., Gan, Z., Berg, T.L., Bansal, M., Liu, J.: Less is more: Clipbert for video-and-language learning via sparse sampling. In: CVPR (2021)
46. Lei, J., Yu, L., Bansal, M., Berg, T.L.: TVQA: Localized, compositional video question answering. In: EMNLP (2018)
47. Li, L., Chen, Y.C., Cheng, Y., Gan, Z., Yu, L., Liu, J.: Hero: Hierarchical encoder for video+ language omni-representation pre-training. In: EMNLP (2020)
48. Li, L., Lei, J., Gan, Z., Yu, L., Chen, Y.C., Pillai, R., Cheng, Y., Zhou, L., Wang, X.E., Wang, W.Y., et al.: Value: A multi-task benchmark for video-and-language understanding evaluation. In: 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks (2021)
49. Li, X., Song, J., Gao, L., Liu, X., Huang, W., He, X., Gan, C.: Beyondrnns: Positional self-attention with co-attention for video question answering. In: AAAI (2020)
50. Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., Gao, J.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: ECCV (2020)
51. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: ICCV (2019)
52. Lin, X., Bertasius, G., Wang, J., Chang, S.F., Parikh, D.: Vx2text: End-to-end learning of video-based text generation from multimodal inputs. In: CVPR (2021)
53. Long Hoang Dang, Thao Minh Le, V.L.T.T.: Object-centric representation learning for video question answering. In: IJCNN (2021)
54. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: CVPR (2019)
55. Maharaj, T., Ballas, N., Rohrbach, A., Courville, A., Pal, C.: A dataset and exploration of models for understanding video data through fill-in-theblank question-answering. In: CVPR (2017)
56. Miech, A., Alayrac, J.B., Laptev, I., Sivic, J., Zisserman, A.: Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In: CVPR (2021)
57. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: CVPR (2020)
58. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: ICCV (2019)
59. Park, J., Lee, J., Sohn, K.: Bridge to answer: Structure-aware graph interaction network for video question answering. In: CVPR (2021)
60. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. In: Journal of Machine Learning Research (2020)
61. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: CVPR (2015)
62. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. In: CVPR (2012)
63. Ryoo, M.S., Piergiovanni, A., Arnab, A., Dehghani, M., Angelova, A.: Tokenlearner: Adaptive space-time tokenization for videos (2021)
64. Ryoo, M.S., Piergiovanni, A., Tan, M., Angelova, A.: AssembleNet: Searching for multi-stream neural connectivity in video architectures. In: ICLR (2020)

65. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NeurIPS. pp. 568–576 (2014)
66. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: ICCV (2019)
67. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. In: EMNLP (2019)
68. Tapaswi, M., Zhu, Y., Rainer Stiefelhagen, A.T., Urtasun, R., Fidler, S.: MovieQA: Understanding stories in movies through questionanswering. In: CVPR (2016)
69. Thao Minh Le, Vuong Le, S.V., Tran, T.: Neural reasoning, fast and slow, for video question answering. In: IJCNN (2020)
70. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: C3d: generic features for video analysis. CoRR, abs/1412.0767 **2**(7), 8 (2014)
71. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR. pp. 6450–6459 (2018)
72. Tsai, Y.H.H., Divvala, S., Morency, L.P., Salakhutdinov, R., Farhadi, A.: Video relationship reasoning using gated spatio-temporal energy graph. In: CVPR (2019)
73. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
74. Wang, J., Ma, L., Jiang, W.: Temporally grounding language queries in videos by contextual boundary-aware prediction. In: AAAI (2020)
75. Wang, X., Wu, J., Chen, J., Li, L., Wang, Y.F., Wang, W.Y.: Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In: ICCV (2019)
76. Wang, Z., Wu, Y., Narasimhan, K., Russakovsky, O.: Multi-query video retrieval. In: ArXiv:2201.03639 (2022)
77. Weijie Su, Xizhou Zhu, Y.C.B.L.L.L.F.W.J.D.: Vl-bert: Pre-training of generic visual-linguistic representations. In: ICLR (2020)
78. Wiseman, S., Rush, A.M.: Sequence-to-sequence learning as beam-search optimization. In: EMNLP (2016)
79. Wu, C.Y., Krahenbuhl, P.: Towards long-form video understanding. In: CVPR (2021)
80. Xiao, J., Shang, X., Yao, A., Chua, T.S.: Next-qa:next phase of question-answering to explaining temporal actions. In: CVPR (2021)
81. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: ECCV. pp. 305–321 (2018)
82. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: ACM Multimedia (2017)
83. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: ACM Multimedia (2017)
84. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: CVPR (2016)
85. Xu, L., Huang, H., Liu, J.: Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events. In: CVPR (2021)
86. Xu, R., Xiong, C., Chen, W., Corso, J.J.: Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In: AAAI (2015)

87. Xue, H., Chu, W., Zhao, Z., Cai, D.: A better way to attend: Attention with trees for video question answering. In: IEEE Transactions on Image Processing (2018)
88. Yang, A., Miech, A., Sivic, J., Laptev, I., Schmid, C.: Just ask: Learning to answer questions from millions of narrated videos. In: ICCV (2021)
89. Yu, W., Zheng, H., Li, M., Ji, L., Wu, L., Xiao, N., Duan, N.: Learning from inside: Self-driven siamese sampling and reasoning for video question answering. Advances in Neural Information Processing Systems **34**, 26462–26474 (2021)
90. Yu, Y., Kim, J., Kim, G.: A joint sequence fusion model for video question answering and retrieval. In: ECCV (2018)
91. Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., Tao, D.: Activitynet-qa: A dataset for understanding complex web videos via question answering. In: AAAI (2019)
92. Yu, Z., Yu, J., Cui, Y., Tao, D., Tian, Q.: Deep modular co-attention networks for visual question answering. In: CVPR (2019)
93. Yunchao Gong, Liwei Wang, M.H.J.H., Lazebnik, S.: Improving image-sentence embeddings using large weakly annotated photo collections. In: ECCV (2014)
94. Zadeh, A., Chan, M., Liang, P.P., Tong, E., Morency, L.P.: Social-iq: A question answering benchmark for artificial social intelligence. In: CVPR (2019)
95. Zellers, R., Lu, X., Hessel, J., Yu, Y., Park, J.S., Cao, J., Farhadi, A., Choi, Y.: Merlot: Multimodal neural script knowledge models (2021)
96. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: ECCV. pp. 803–818 (2018)
97. Zhou, L., Xu, C., Corso, J.J.: Towards automatic learning of procedures from web instructional videos. In: AAAI
98. Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: CVPR (2018)
99. Zhu, L., Xu, Z., Yan, Y., Hauptmann, A.G.: Uncovering the temporal context for video question answering. In: IJCV (2018)
100. Zhu, L., Yang, Y.: Actbert: Learning global-local video-text representations. In: CVPR (2020)
101. Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J.: Cross-task weakly supervised learning from instructional videos. In: CVPR (2019)
102. Zolfaghari, M., Singh, K., Brox, T.: Eco: efficient convolutional network for online video understanding. In: ECCV (2018)