Video Language Co-Attention with Multimodal Fast-Learning Feature Fusion for VideoQA

Adnen Abdessaied[*], Ekta Sood[*], Andreas Bulling
Institute for Visualization and Interactive Systems (VIS)
University of Stuttgart, Germany
{adnen.abdessaied,ekta.sood,andreas.bulling}@vis.uni-stuttgart.de

Abstract

We propose the Video Language Co-Attention Network (VLCN) – a novel memory-enhanced model for Video Question Answering (VideoQA). Our model combines two original contributions: A multimodal fast-learning feature fusion (FLF) block and a mechanism that uses self-attended language features to *separately* guide neural attention on both static and dynamic visual features extracted from individual video frames and short video clips. When trained from scratch, VLCN achieves competitive results with the state of the art on both MSVD-QA and MSRVTT-QA with 38.06% and 36.01% test accuracies, respectively. Through an ablation study, we further show that FLF improves generalization across different VideoQA datasets and performance for question types that are notoriously challenging in current datasets, such as long questions that require deeper reasoning as well as questions with rare answers[1].

1 Introduction

Video Question Answering (VideoQA) has emerged as a challenging task at the intersection of natural language processing and computer vision. In contrast to image-based visual question answering (Lu et al., 2016; Anderson et al., 2018; Yu et al., 2019), VideoQA takes dynamic visual content (a video) as input (Xu et al., 2017; Gao et al., 2018; Li et al., 2019). This poses new challenges given that generating correct answers requires models to analyze spatial, appearance-based features of individual video frames *jointly* with the temporal, motion-based dynamics across multiple frames (Zhu et al., 2017).

However, there still is a semantic gap between the visual and language channels (Lei et al., 2018; Sun et al., 2021; Song et al., 2018) that prior work has tried to close by leveraging external memory (Kim et al., 2018, 2019; Fan et al., 2019). While external memory allows models to cache sequential information and retrieve relevant multimodal content (Patel et al., 2021), latest models still suffer from decreased performance, for example on ambiguous questions that require deeper reasoning abilities.

Moreover, current deep neural models for VideoQA are limited in that they only gradually learn during training. In contrast, human cognition leverages two different learning systems: a gradual and a fast-learning system (McClelland et al., 1995). Current networks lack a similar fast-learning mechanism, which impedes their ability to efficiently reason and generalize to unseen data.

To address these limitations, we propose the Video Language Co-Attention Network (VLCN) – a novel memory-enhanced model for VideoQA. VLCN implements a video language co-attention module that uses self- and guided-attention to align language features of the question with static and dynamic visual features extracted from videos. As such, the module offers complementary information that our network attends to, independently of each other, when visually grounding a question. Furthermore, VLCN features a novel multimodal fast-learning fusion (FLF) block that helps the model to deal with challenging questions that need deeper reasoning and understanding. Inspired by the cognitive fast-learning system (McClelland et al., 2019), we leverage the differentiable neural computer (DNC) to incorporate an external memory which the network learns how to use by freeing and reusing its memory slots.

We seamlessly integrate our novel video language co-attention module as well as the fast-learning feature fusion approach in the recent transformer-based MCAN network (Yu et al., 2019). We show that our model achieves competitive results with the state of the art on two challenging datasets – MSVD-QA and MSRVTT-QA. Our results further show that our model performs better on ambiguous questions and can better reason not only about questions with rare answers but also longer questions that require a deeper understanding of both

[*]Equal contribution.

[1]Our code is publicly available at https://git.hcics.simtech.uni-stuttgart.de/public-projects/vlcn

the question and the visual input. In addition, we show that FLF facilitates generalization across different VideoQA datasets via transfer learning.

2 Related Work

Our work is related to previous works on 1) attention mechanisms in VideoQA, and 2) memory-enhanced networks.

Attention Mechanisms in VideoQA. Neural attention mechanisms have become the de-facto standard in machine comprehension tasks (Sood et al., 2020; Yu et al., 2019; Li et al., 2019). In VideoQA, attention mechanisms are particularly important given that the information necessary to generate correct answers is scattered across frames – many of which are redundant or even irrelevant to the question at hand (Patel et al., 2021).

Ye et al. (2017) introduced the attribute-augmented attention network that learned temporally attended video representations according to semantic attributes. Xu et al. (2017) reported new state-of-the-art performance by applying question-guided attention over both the appearance and motion features of individual as well as multiple video frames. Motivated by the challenge to capture long-range dependencies, Li et al. (2019) used a transformer-based co-attention network to exploit the global dependencies of the text and the temporal dynamics of the videos. Yang et al. (2020) leveraged BERT (Devlin et al., 2018) to obtain richer contextual feature representations over the question. More recently, Seo et al. (2021) proposed a two-stream multimodal video transformer based architecture (CoMVT) that jointly attends over words in text and visual objects and scenes to learn visual-dialogue context. Although CoMVT achieves state-of-the-art results on multiple downstream VideoQA datasets, it requires a computationally-demanding pretraning stage on 1.2M instructional videos.

These previous methods have used question features to guide attention over either frame or clip-level visual features, and some applied self and co-attention to individual frames. Our work, however, is the first to use self-attention on the question which then *separately* guides the attention over both individual video frames and clips.

Memory-enhanced Networks. In parallel, other works have focused on augmenting models with external memory components to improve their reasoning capabilities particularly over long-range data that are common in many visiolinguistic tasks, e.g. images with many objects or videos with a large number of frames. One of the first methods introduced a memory component over simple facts for question answering (Weston and Bordes, 2015).

The introduction of end-to-end trainable models popularized the use of external memory components (Sukhbaatar et al., 2015). Driven by the insight that memory access is similar to neural attention (Collier and Beel, 2019), other works integrated attention mechanisms to allow networks to better interact with their external memory through read and write operations, such as the Neural Turing Machine (NTM) (Graves et al., 2014) or the Differential Neural Computer (DNC) (Graves et al., 2016). The latter includes a dynamic memory allocation scheme that enables it to learn how to effectively free and reuse memory slots.

Several works aimed to leverage the potential of memory-enhanced networks for VideoQA. Na et al. (2017) applied memory over the video frames using multi-layered CNNs read and write networks to capture richer temporal dynamics of frame-level sequence information. Xue et al. (2018) obtained syntax parse trees over questions and then stored these into memory, allowing their model to perform better on more complex questions. Fan et al. (2019) used one memory component to effectively learn global context information from appearance and motion features in combination with another question-memory to help understand the complex semantics of questions and highlight queried subjects. Gao et al. (2018) used a co-memory attention mechanism to generate attention from motion and appearance cues. More recently, Yin et al. (2020) achieved new state-of-the-art results on MSVD-QA (Xu et al., 2017) by using a DNC (Graves et al., 2016) to encode the textual information of the question and the visual information of the video.

While previous works used memory-enhanced networks to *extract* linguistic and visual features, we propose a memory-augmented block adapted from the DNC to potentially emulate the human-like fast-learning capabilites (McClelland et al., 2020) and use it to *fuse* multimodal features previously attended by an encoder-decoder transformer-based co-attention module instead.

3 Method

We propose the Video Language Co-Attention Network (VLCN) that integrates two original contributions (see Figure 1): First, we propose to use self- and guided-attention to *separately* align the language features with the static and dynamic visual features extracted from single video frames and frame sequences (clips). Second, we introduce Fast-Learning Fusion (FLF) – a novel memory-enhanced multimodal block that learns a single fused representation of all features (i.e. language, static and dynamic visual features).

3.1 Feature Representation

In contrast to images, videos consist of multiple frames that capture temporal object dynamics and motion features. Combinations of static and dy-

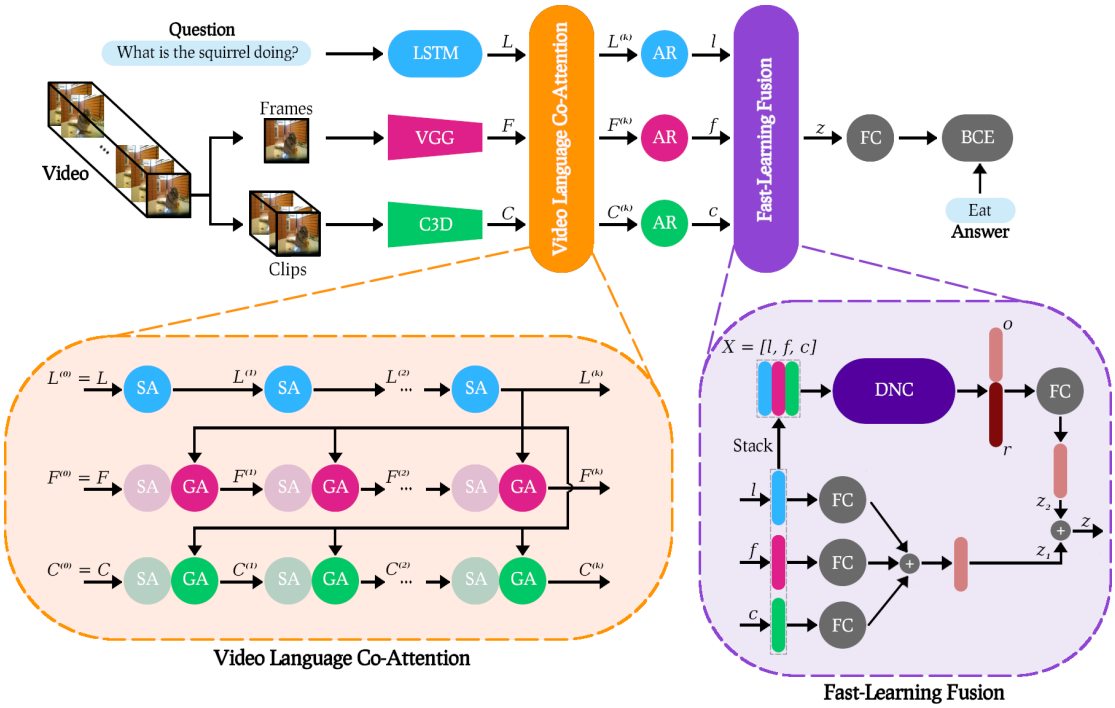

Figure 1: Architecture of the proposed Video Language Co-attention Network (VLCN). Our model aligns three different types of input (language features L, static visual features F and dynamic visual features C) using self- and guided-attention. Then, it fuses the attended reduced features (l, f and c) with the help of Fast-Learning Fusion (Fast-Learning Fusion (FLF)) – a novel memory-augmented multimodal fusion block. AR = Attention Reduction.

namic visual features have therefore become the de-facto standard for video representations (Xu et al., 2017; Le et al., 2020a) in VideoQA. We adopt the same approach in our Video Language Co-Attention Network (VLCN).

Visual Features. For each video, we first sample n_v evenly-distributed frames and clips where a clip is a sequence of 16 consecutive video frames. Then, we apply a VGG network (Simonyan and Zisserman, 2014) pre-trained on ImageNet (Russakovsky et al., 2015) and a C3D network[2] (Ji et al., 2012) pre-trained on Sports1M (Karpathy et al., 2014) on these sampled frames and clips, respectively. The activations of their last d_v-dimensional fully-connected layers are our static and dynamic visual features.

This results in a set of static frame features $F = [f_1, \ldots, f_{n_v}] \in \mathbb{R}^{n_v \times d_v}$ and a set of dynamic clip features $C = [c_1, \ldots, c_{n_v}] \in \mathbb{R}^{n_v \times d_v}$.

Language Features. Question tokens are represented using 300-D GloVe embeddings (Pennington et al., 2014) and encoded with a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997) with d_l hidden dimensions. Thus, each question is represented as a matrix $L \in \mathbb{R}^{n_l \times d_l}$, where n_l is the number of question tokens.

[2] `https://github.com/DavideA/c3d-pytorch`

3.2 Video Language Co-Attention

The intuition behind our overall approach is the way humans typically answer questions about videos: first, we read the question. Then, we consider the visual input, i.e. its static (colours, objects and shapes) and dynamic (movements and actions) visual features, to answer it. VLCN uses stacked video language co-attention layers in an encoder-decoder fashion (see Figure 1). Given a set of features, each layer simultaneously computes the self-attention of the question, frames and clips features. Then, the self-attended question features of the last layer, i.e. $L^{(K)}$, are used to *separately* guide the attention over the frames and clip features in a bottom up manner (Anderson et al., 2018). At the core of self- and guided-attention sits a multi-head attention block (Vaswani et al., 2017) that computes a scaled dot-product of a query $q \in \mathbb{R}^{1 \times d}$ and a set of n keys $K \in \mathbb{R}^{n \times d}$, where d is a common hidden dimension. A softmax function is then applied to obtain the attention weights A on the values $V \in \mathbb{R}^{n \times d}$ following:

$$A = \text{softmax}(\frac{qK^T}{\sqrt{d}})V. \quad (1)$$

Similar to (Vaswani et al., 2017), attention weights A are computed for multiple queries $Q \in \mathbb{Q}^{n \times d}$ at the same time using Equation (1). The outputs of the final video language co-attention layers

$L^{(K)} \in \mathbb{R}^{n_l \times d}$, $F^{(K)} \in \mathbb{R}^{n_v \times d}$ and $C^{(K)} \in \mathbb{R}^{n_v \times d}$ encode information about the attention weights over the question tokens and visual semantics. We reduce them to get the final attended features $l, f, v \in \mathbb{R}^d$ by linearly combining the rows of $L^{(K)}$, $F^{(K)}$ and $C^{(K)}$, respectively (see Figure 1). Taking the language features as an example, we first process $L^{(K)}$ by a multi-layer Feed-Forward Network (FFN) followed by a softmax to obtain the attention weights that we use to linearly combine the rows of $L^{(K)}$ as:

$$a = \text{softmax}(\text{FFN}(L^{(K)})) \in [0, 1]^{n_l}, \quad (2)$$

$$l = \sum_{i=1}^{n_l} a_i L^{(K)}[i, :] \in \mathbb{R}^d. \quad (3)$$

3.3 Fast-Learning Feature Fusion

We opted to use the DNC as a basis for this approach given that it is, to our knowledge, the most capable memory-augmented model to date that can be trained in an end-to-end fashion (Graves et al., 2016). In previous works (Graves et al., 2016; Yin et al., 2020), the DNC was heavily used to process long input sequences. However, its capability to treat shorter input sequences remains unexplored even though it has been argued for by cognitive science to be capable of emulating the human fast-learning system proposed in the complementary learning systems theory (McClelland et al., 1986, 2019). In our work, we leverage it —for the first time —to fuse our three multi-modal inputs (i.e. language, static, and dynamic visual features) within the VideoQA task.

Differential Neural Computer (DNC). The DNC consists of two major components: A neural network controller and an external memory. At each time-step t, the controller receives an input vector x_t and emits an output vector y_t. In addition, it receives a set of R read vectors $\{r^i_{t-1}\}^R_{i=1}$ from the $N \times W$ memory matrix M_{t-1} of the previous time-step $t-1$. Both controller inputs and the read vectors are concatenated to form the final input vector $\chi_t = [x_t; r^1_{t-1}; \ldots; r^R_{t-1}]$. Theoretically, the controller can be a network of any type. However, it is common to use an LSTM network with L hidden layers. The output vector y_t is computed via

$$y_t = W_h[h^1_t; \ldots; h^L_t] + W_r[r^1_t; \ldots; r^R_t], \quad (4)$$

where W_h and W_r are learnable weights and $\mathbf{h_t} = \{h^i_t\}^L_{i=1}$ are the hidden states of the LSTM controller. These hidden states are used to parameterize one write and R read heads to interact with the $N \times W$ external memory matrix through the so-called *fast-learning* connections. Further details on the DNC can be found in (Graves et al., 2016).

Feature Fusion. First, the reduced language and visual features l, f and c (see Figure 1) are projected and summed to get the first intermediate output z_1. Then, they are duplicated and stacked one after another to form the input sequence $X = [l, f, c] = [x_1, x_2, x_3] \in \mathbb{R}^{3 \times d}$ to the DNC. The output sequence $Y = [y_1, y_2, y_3] \in \mathbb{R}^{3 \times d}$ is summed along the first dimension to obtain the final output o and the last R read vectors are concatenated to form the global read vector r. Finally, the second intermediate output z_2 is obtained by projecting $[o; r]$ onto the same space as z_1 and the final output z is computed by summing z_1 and z_2. We concatenated the last read vectors to the DNC's output to preserve the memory information from the last step of processing the input sequences.

Answer Prediction. Given that VideoQA is formulated as a classification task, the fused features z are projected onto the answer space using a fully-connected layer. A sigmoid function is applied to train the network with binary cross-entropy (BCE) loss (see Figure 1).

4 Experiments

Datasets. We conducted experiments on two open-ended VideoQA datasets: MSVD-QA and MSRVTT-QA (Xu et al., 2017). They are, in turn, based on the Microsoft Research Video Description Corpus (MSVD) (Chen and Dolan, 2011) and the Microsoft Research Video to Text (MSRVTT) (Xu et al., 2016) datasets, respectively. Both datasets contain automatically generated questions that fall into five different categories: *what*, *who*, *how*, *when* and *where*. MSVD-QA has a total number of 1200 videos and 50 505 question-answer pairs and comes with three splits based on the videos: The training, validation, and test sets account for 61%, 13%, and 26% of the total number of videos, respectively. Similarly, MSRVTT-QA has three splits with 10 000 videos and 243 680 question-answer pairs in total. The training, validation, and test sets account for 65%, 5%, and 30% of the total number of videos, respectively. Further details on the datasets can be found in Appendix A.1.

Implementation Details. For each video, we sampled $n_v = 20$ frames and clips and used them to generate the static and dynamic visual features. We set the dimensionality of the input question features d_l and input visual features d_v (static and dynamic) to 512 and 4,096, respectively. The fused features z_1, z_2 and z had a dimension $d_z = 1,024$. Following (Vaswani et al., 2017), we set the latent dimension d of the multi-head attention block to 512 and the number of heads to eight, i.e. each had a dimensionality of 64. Since VideoQA is formulated as a classification task, similar to (Xu et al., 2017), we used the most frequent 1,000 ground-truth answers of the training and validation splits as our answer candidates. The number of video

language co-attention layers K was fixed to six. Finally, for the DNC[3] in the FLF block we used a two-layer bidirectional LSTM network (Hochreiter and Schmidhuber, 1997) with 512 hidden dimensions as a controller as well as four read and one write heads to interact with the 512×64 external memory matrix. We used Adam (Kingma and Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.98$ to optimize the weights of our model over a maximum of 30 epochs. We set the base learning rate to 10^{-4}. The batch-size was fixed to 64 and 32 during training and evaluation, respectively. We implemented our model in PyTorch (Paszke et al., 2019). It is based on a Visual Question Answering (VQA) open-source implementation[4] and will be made publicly available together with our pre-trained models. All experiments were conducted on one Nvidia Tesla V100 GPU with 32GB VRAM.

Ablated Models. In all experiments that follow we denote with *VLCN* our full model that uses a DNC inside the FLF block and whose architecture is illustrated in Figure 1. Although we experimented with training the DNC with different permutations of its inputs, we did not obtain any improvements in terms of performance when we cahnged the order of the input features. Therefore, we kept the same order that we used to encode the features (i.e. $[l, f, c]$). We additionally implemented different ablated versions of our model to study the impact of the proposed video language co-attention and fast-learning fusion:

• *MCAN*: This is the original MCAN model as proposed in (Yu et al., 2019) but adapted for VideoQA. We trained it using the concatenated static and dynamic visual features as they share the same dimensionality d_v. This model was not equipped with our novel video-language co-attention and fast-learning feature fusion.

• *VLCN−FLF*: For this model we used a simple multimodal fusion by summing the reduced features l, f and c, i.e. only the first intermediate output z_1 was passed through to the subsequent parts of the network (see Figure 1).

• *VLCN+LSTM*: For this model we only used the controller of the DNC, i.e. a two-layer bidirectional LSTM with 512 hidden dimensions, to compute the second intermediate output z_2 by summing the outputs of the LSTM and projecting them onto the same space as z_1. This model did not have the external long-term memory matrix and the fast-learning connections.

Model Training. We evaluated the robustness of our model and its ablated versions by training each five times with five different seeds. We report the performance as $\mu \pm \sigma$, where μ and σ are the average and standard deviation of the ensemble-accuracy on MSVD-QA and MSRVTT-QA *test*.

Question Length and Answer Frequency. Complementing analyses according to common question type categories (*what*, *who*, *how*, *when* and *where*), we propose two other question-binning strategies: In the first strategy, questions are put into three bins based on question length. The first bin contains questions with up to three words, the second bin between four and eight, and the last bin with more than nine words. The longer the question, the harder it should be for the model to answer as it requires deeper reasoning and understanding. In the second strategy, questions are binned according to the frequency rank of their ground-truth answers in the training and validation splits. The first bin contains questions whose ground-truths are the 100 most frequent answers. The second contains questions whose ground-truths are the next 200 most frequent answers. The last bin contains the rest of the questions, i.e. questions with the scarcest 700 answers. The rarer answers to a question are, the more difficult it should be for the model to answer correctly.

Transfer Learning of the FLF Weights. MSRVTT-QA includes more questions and longer videos compared to MSVD-QA: The average video lengths of MSRVTT-QA and MSVD-QA are 20 and 10 seconds, respectively (Aafaq et al., 2019). Performance on MSRVTT-QA should thus benefit from the knowledge acquired while training on MSVD-QA (Pan and Yang, 2010). Through transfer-learning of the *fast-learning connections* and *the DNC controller weights* learned from MSVD-QA, FLF should be able to better interact with its external memory when dealing with questions from MSRVTT-QA. To study this hypothesis, we conducted the following experiment: We trained an ensemble of five VLCNs using five different seeds on MSVD-QA. Then, we trained two further ensembles of five VLCNs on MSRVTT-QA using the same seeds: For one ensemble, we initialised the FLF weights of each model with those learned from MSVD-QA and fine-tuned them on MSRVTT-QA. We call these models *VLCN+FT*. For models in the second ensemble we trained these weights from scratch. Additionally, we experimented with fine-tuning the entire architecture of the FLF block instead. These experiments did not yield any performance improvements and we decided not to include them in this work.

5 Results

Comparison with the State of the Art. VLCN achieves competitive performance with the

[3] `https://github.com/ixaxaar/pytorch-dnc`
[4] `https://github.com/MILVLG/mcan-vqa`

Model	Question Type					All
	What	Who	How	When	Where	
ST-VQA (Jang et al., 2017)	18.10	50.00	**83.80**	72.40	28.60	31.30
Co-Mem (Gao et al., 2018)	19.60	48.70	81.60	74.10	31.70	31.70
HMEMA (Fan et al., 2019)	22.40	50.10	73.00	70.70	42.90	33.70
SSML (Amrani et al., 2020)	–	–	–	–	–	35.13
QueST (Jiang et al., 2020)	24.50	**52.90**	79.10	72.40	**50.00**	36.10
HCRN (Le et al., 2020b)	–	–	–	–	–	36.10
MA-DRNN (Yin et al., 2020)	24.30	51.60	82.00	**86.30**	26.30	36.20
CoMVT (Seo et al., 2021)						
Scratch	–	–	–	–	–	35.70
Pretrained	–	–	–	–	–	**42.60**
VLCN (Ours)	**28.42**	51.29	81.08	74.13	46.43	38.06

Table 1: Performance comparison of VLCN with the state of the art on MSVD-QA *test*. The table shows the overall accuracy as well as the accuracy with respect to individual question types in %.

Model	Question Type					All
	What	Who	How	When	Where	
ST-VQA(Jang et al., 2017)	24.50	41.20	78.00	76.50	34.90	30.90
Co-Mem (Gao et al., 2018)	23.90	42.50	74.10	69.00	**42.90**	32.00
HMEMA(Fan et al., 2019)	26.50	43.60	82.40	76.00	28.60	33.00
QueST (Jiang et al., 2020)	27.90	**45.60**	**83.00**	75.70	31.60	34.60
SSML (Amrani et al., 2020)	–	–	–	–	–	35.00
HCRN (Le et al., 2020b)	–	–	–	–	–	35.60
CoMVT (Seo et al., 2021)						
Scratch	–	–	–	–	–	37.30
Pretrained	–	–	–	–	–	**39.50**
VLCN (Ours)	**30.69**	44.09	79.82	**78.29**	36.80	36.01

Table 2: Performance comparison of VLCN with the state of the art on MSRVTT-QA *test*. The table shows the overall accuracy as well as the accuracy with respect to individual question types in %.

state of the art on both MSVD-QA and MSRVTT-QA. On MSVD-QA, our best model reaches an overall accuracy of 38.06% compared to 35.70%, 36.10%, and 36.20% achieved by CoMVT (scratch) (Seo et al., 2021), HCRN (Le et al., 2020b), and MA-DRNN (Yin et al., 2020), respectively. This corresponds to a relative improvement of 1.86% over the state of the art when the latter is trained from scratch (see Table 1). Although CoMVT can reach an overall accuracy of 42.60%, this was only possible after a computationally-demanding pretraining stage on HowToFUP (Miech et al., 2019) —a dataset consisting of 1.2M instructional videos for the task of Future Utterance Prediction (FUP). On the most diverse question types our model achieves a higher accuracy on *what* ($\sim$ 4% increase) and performs slightly worse on *who* compared to MA-DRNN. On the other types *how*, *when* and *where*, our model performs on par with the state of the art methods. As depicted in Table 2, our best VLCN model achieves an overall accuracy of 36.01% on MSRVTT-QA – the second best performance after CoMVT which achieves 37.30% accuracy when trained from scratch and 39.50% after pretaining on HowToFUP.

Ablation Study. Our analysis of the question length shows that VLCN achieves the best performance across all question length bins on MSVD-QA and on long questions, i.e. questions with length bigger than three, on MSRVTT-QA (see Tables 3 and 4). By comparing the first two rows of Table 3 and Table 4, we can see that VLCN$-$FLF outperforms MCAN across all of the question length bins of MSVD-QA and on very long questions (≥ 9) of MSRVTT-QA. This suggests that our co-attention approach helps the model make reliable predictions when the question becomes more complex compared to the simple question-guided attention over the *stacked* visual features. We hypothesize that the static and dynamic visual features offer complementary information that our network needs to attend to, independently of each other, while trying to visually ground the question. By removing the external memory of the FLF block and using a plain LSTM network, VLCN+LSTM falls behind on all question length bins resulting in an overall accuracy decrease of 0.84% and 0.8% on MSVD-QA and MSRVTT-QA, respectively, compared to VLCN (see Tables 3 and 4). We hypothesize that the proposed external memory is indispensable when answering questions that exceed the working memory capacity of the model, i.e. in this case of the LSTM network.

We then analyzed the performance of our ablated versions with respect to the answer frequency

Model	Question Length (number of words)			
	1-3	**4-8**	**$\geq$ 9**	**All**
MCAN_{avg}	35.83 ± 1.30	36.37 ± 0.33	38.13 ± 0.98	36.64 ± 0.44
$\text{VLCN}-\text{FLF}_{\text{avg}}$	37.85 ± 1.63	36.89 ± 0.28	38.32 ± 0.53	37.16 ± 0.27
$\text{VLCN}+\text{LSTM}_{\text{avg}}$	39.40 ± 1.64	36.38 ± 0.29	38.33 ± 0.61	36.82 ± 0.31
VLCN_{avg}	$\mathbf{39.48 \pm 0.73}$	$\mathbf{37.37 \pm 0.21}$	$\mathbf{38.65 \pm 0.52}$	$\mathbf{37.66 \pm 0.21}$

Table 3: Performance comparison of different ablated versions of our model on MSVD-QA *test*. The table shows the average accuracy and standard deviation $\mu \pm \sigma$ for each length bin in %.

Model	Question Length (number of words)			
	1-3	**4-8**	**$\geq$ 9**	**All**
MCAN_{avg}	$\mathbf{38.94 \pm 0.46}$	36.15 ± 0.16	33.35 ± 0.18	35.49 ± 0.16
$\text{VLCN}-\text{FLF}_{\text{avg}}$	38.49 ± 0.46	35.85 ± 0.17	33.42 ± 0.27	35.29 ± 0.16
$\text{VLCN}+\text{LSTM}_{\text{avg}}$	38.45 ± 0.35	35.82 ± 0.12	33.15 ± 0.19	35.20 ± 0.12
VLCN_{avg}	38.31 ± 0.41	36.57 ± 0.18	33.45 ± 0.15	35.77 ± 0.15
$\text{VLCN}+\text{FT}_{\text{avg}}$	38.92 ± 0.27	$\mathbf{36.78 \pm 0.02}$	$\mathbf{33.65 \pm 0.08}$	$\mathbf{36.00 \pm 0.01}$

Table 4: Performance comparison of different ablated versions of our model on MSRVTT-QA *test*. The table shows the average accuracy and standard deviation $\mu \pm \sigma$ for each length bin in %.

Model	Answer Frequency Bin			
	1-100	**101-300**	**$\geq$ 301**	**All**
MCAN_{avg}	50.40 ± 0.55	16.09 ± 0.47	2.76 ± 0.18	36.64 ± 0.44
$\text{VLCN}-\text{FLF}_{\text{avg}}$	$\mathbf{51.37 \pm 0.19}$	15.72 ± 0.90	2.49 ± 0.73	37.16 ± 0.27
$\text{VLCN}+\text{LSTM}_{\text{avg}}$	50.57 ± 0.86	16.57 ± 1.01	3.25 ± 0.72	36.82 ± 0.31
VLCN_{avg}	51.35 ± 0.36	$\mathbf{17.80 \pm 0.42}$	$\mathbf{3.35 \pm 0.17}$	$\mathbf{37.66 \pm 0.21}$

Table 5: Performance comparison of different ablated versions of our model on MSVD-QA *test*. The table shows the average accuracy and standard deviation $\mu \pm \sigma$ for each frequency bin in %.

Model	Answer Frequency Bin			
	1-100	**101-300**	**$\geq$ 301**	**All**
MCAN_{avg}	$\mathbf{48.90 \pm 0.34}$	17.08 ± 0.74	3.26 ± 0.29	35.49 ± 0.16
$\text{VLCN}-\text{FLF}_{\text{avg}}$	48.64 ± 0.11	16.90 ± 0.59	3.28 ± 0.35	35.29 ± 0.16
$\text{VLCN}+\text{LSTM}_{\text{avg}}$	48.53 ± 0.21	16.62 ± 0.44	3.39 ± 0.29	35.20 ± 0.12
VLCN_{avg}	47.70 ± 0.24	20.97 ± 0.25	5.84 ± 0.18	35.77 ± 0.15
$\text{VLCN}+\text{FT}_{\text{avg}}$	48.03 ± 0.11	$\mathbf{20.98 \pm 0.33}$	$\mathbf{5.88 \pm 0.12}$	$\mathbf{36.00 \pm 0.01}$

Table 6: Performance comparison of different ablated versions of our model on MSRVTT-QA *test*. The table shows the average accuracy and standard deviation $\mu \pm \sigma$ for each frequency bin in %.

bins (see Table 5). On MSVD-QA, VLCN achieves the best results on the most challenging questions, i.e. questions whose answers are not amongst the 100 most frequent, and performs on par with VLCN−FLF on questions with the 100 most frequent answers. Although VLCN+LSTM performs on par with VLCN and improves on the performance of MCAN and VLCN−FLF on the most challenging questions, it falls behind VLCN when it comes to the easier questions with the most frequent answers. This results in an overall accuracy decrease of 0.5% compared to VLCN.

Similarly, VLCN outperforms all of its ablated versions on the most challenging questions of MSRVTT-QA (see Table 6). In contrast to MSVD-QA, VLCN+LSTM does not reach superior results on the most challenging questions compared to MCAN and VLCN−FLF. Performance on such questions only improves when using the external memory. In fact, VLCN achieves 20.97% and 5.84% on questions with the second 100 most frequent answers and questions with the scarcest 700 answers, respectively. This translates into a relative improvement of 2.49% and 3.89% compared to the second best models on such answer frequency bins, i.e. MCAN and VLCN+LSTM, respectively (see Table 6). It is interesting to see the difficulty of answering questions with rare ground truth answers

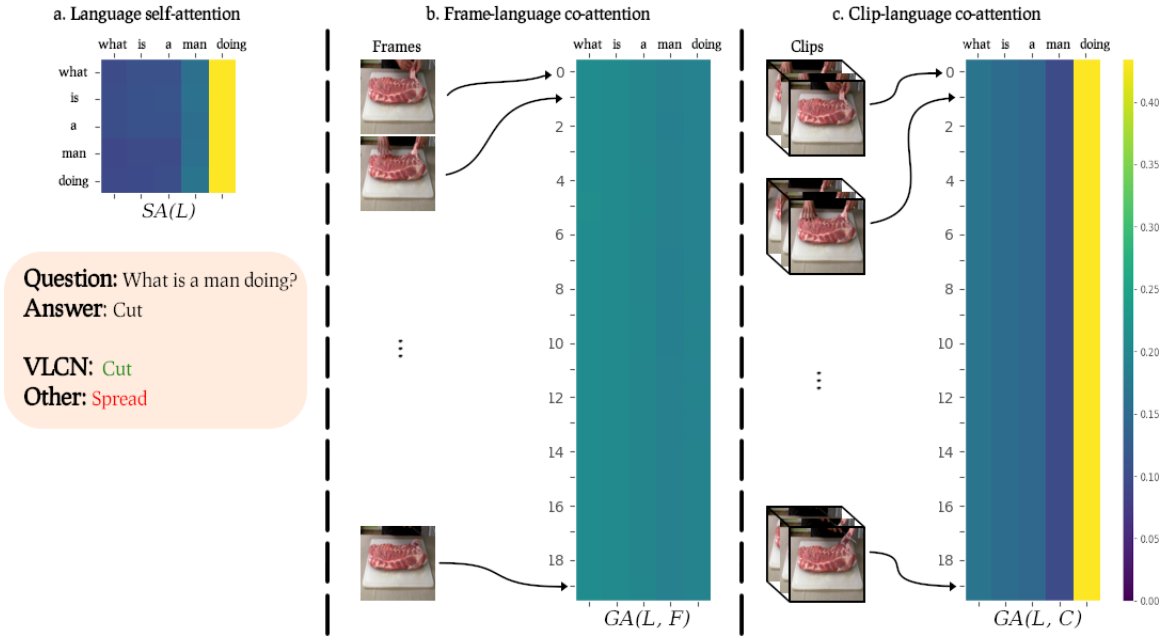

Figure 2: Visualization of the attention maps learned by the last video language co-attention layer. The indices [0, 19] indicate the individual 20 frames and clips of the video (some of which are shown).

as highlighted by the severe drop in performance for the last answer frequency bin of Table 5 and Table 6. We do not think that this is related to a language understanding problem as suggested by the error analysis we conducted on the ablated versions. Please refer to Appendix A.2 for more details.

Transfer Learning. The last two rows of Tables 4 and 6 show the importance of curriculum learning (Bengio et al., 2009). By fine-tuning the converged weights of FLF from MSVD-QA on MSRVTT-QA, VLCN+FT reaches new state of the art result on MSRVTT-QA by improving the accuracy on all question length and answer frequency bins compared to VLCN. This indicates that transfer learning of the fast-learning connections of FLF is possible and improves performance across different datasets. Further details about the effect of fine-tuning on the performance on individual question types can be found in Appendix A.3.

Qualitative Analysis. Figure 2 shows sample attention maps learned by the last video language co-attention layer together with the predictions of our model and its ablated versions. These predictions are depicted in the orange box, where *other* denotes the ablated versions of our full VLCN model. Further examples can be found in Appendix A.4. The language self-attention $SA(L)$ and the guided-attention over the clips $G(C, L)$ show that VLCN attends to the word *doing* the most. The high values of the last column of $G(L, C)$ indicates that the model is searching for possible clips that align well with the action *doing*. This highlights the importance of the independent language guided-attention over the clips. However, the guided-attention map over the frames $G(L, F)$ is flat indicating that the model is not sure which frames are important to answer the question. This uncertainty is alleviated by the efficient fast-learning feature fusion of FLF that leads our full VLCN model to predict the correct answer. While all of the ablated versions predict the wrong answer *spread*, our VLCN model answers the question correctly by predicting *cut*.

6 Conclusion

In this work, we proposed the Video Language Co-Attention Network (VLCN) for VideoQA. At its core are two distinct novel contributions: Stacked co-attention layers in an encoder-decoder framework to *separately* guide self-attended language features over both static video frame and dynamic clip features; and Fast-Learning Fusion (FLF) – a memory-enhanced multimodal block to efficiently fuse the reduced features. We demonstrated that the combination of both results in significant improvements and competitive performance with state-of-the-art models on the challenging MSVD-QA and MSRVTT-QA datasets. We also demonstrated the particular advantage of our model in dealing with long questions that require deeper reasoning or questions with rare answers. Finally, further experiments showed that our FLF block allows our model to generalize better across different datasets via transfer learning.

Acknowledgments

A. Abdessaied and A. Bulling were funded by the European Research Council (ERC; grant agreement 801708). E. Sood was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 - 390740016.

References

Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. 2019. Video Description: A Survey of Methods, Datasets, and Evaluation Metrics. *Association for Computing Machinery*, 52(6).

Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. 2020. Noise estimation using density estimation for self-supervised multimodal learning. *arXiv preprint arXiv:2003.03186*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proc. International Conference on Machine Learning (ICML)*, page 41–48.

David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 190–200.

Mark Collier and Joeran Beel. 2019. Memory-augmented neural networks for machine translation. *arXiv preprint arXiv:1909.08314*.

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

Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. 2019. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1999–2007.

Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. 2018. Motion-appearance co-memory networks for video question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6576–6585.

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

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538:471–476.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–1780.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2758–2766.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35:221–231.

Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. 2020. Divide and Conquer: Question-Guided Spatio-Temporal Contextual Attention for Video Question Answering. *Proc. Conference on Artificial Intelligence (AAAI)*, 34:11101–11108.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale Video Classification with Convolutional Neural Networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D. Yoo. 2019. Progressive attention memory network for movie story question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kyung-Min Kim, Seong-Ho Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. 2018. Multimodal dual attention memory for video story question answering. In *Proc. European Conference on Computer Vision (ECCV)*, pages 673–688.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. 2020a. Hierarchical Conditional Relation Networks for Video Question Answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. 2020b. Hierarchical conditional relation networks for video question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9972–9981.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. 2018. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*.

Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. 2019. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Proc. Conference on Artificial Intelligence (AAAI)*, volume 33, pages 8658–8665.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical Question-Image Co-Attention for Visual Question Answering. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–9.

James L McClelland, Felix Hill, Maja Rudolph, Jason Baldridge, and Hinrich Schütze. 2020. Placing language in an integrated understanding system: Next steps toward human-level performance in neural language models. *National Academy of Sciences*, 117:25966–25974.

James L McClelland, Felix Hill Maja, Rudolph, Jason Baldridge, and Hinrich Schütze. 2019. Extending machine language models toward human-level language understanding. *arXiv preprint arXiv:1912.05877*.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.

James L McClelland, David E Rumelhart, PDP Research Group, et al. 1986. *Parallel distributed processing*, volume 2. MIT press Cambridge, MA.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.

Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. 2017. A read-write memory network for movie story understanding. In *Proc. International Conference on Computer Vision (ICCV)*, pages 677–685.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035.

Devshree Patel, Ratnam Parikh, and Yesha Shastri. 2021. Recent advances in video question answering: A review of datasets and methods. *arXiv preprint arXiv:2101.05954*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252.

Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. 2021. Look Before you Speak: Visually Contextualized Utterances. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Xiaomeng Song, Yucheng Shi, Xin Chen, and Yahong Han. 2018. Explore multi-step reasoning in video question answering. In *Proc. International Conference on Multimedia (ACM-MM)*, pages 239–247.

Ekta Sood, Simon Tannert, Philipp Müller, and Andreas Bulling. 2020. Improving Natural Language Processing Tasks with Human Gaze-Guided Neural Attention. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–15.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*.

Guanglu Sun, Lili Liang, Tianlin Li, Bo Yu, Meng Wu, and Bolun Zhang. 2021. Video question answering: a survey of models and datasets. *Mobile Networks and Applications*, pages 1–34.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Sumit Chopra Jason Weston and Antoine Bordes. 2015. Memory networks. *arXiv preprint arXiv:1410.3916*.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video Question Answering via Gradually Refined Attention over Appearance and Motion. In *Proc. International Conference on Multimedia (ACM-MM)*, page 1645–1653.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296.

Hongyang Xue, Wenqing Chu, Zhou Zhao, and Deng Cai. 2018. A better way to attend: Attention with trees for video question answering. *IEEE Transactions on Image Processing*, 27(11):5563–5574.

Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. 2020. Bert representations for video question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1556–1565.

Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. 2017. Video question answering via attribute-augmented attention network learning. In *Proc. Conference on Research and Development in Information Retrieval (ACM-SIGIR)*, pages 829–832.

Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. 2020. Memory Augmented Deep Recurrent Neural Network for Video Question Answering. *IEEE Transactions on Neural Networks and Learning Systems*, 31:3159–3167.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6281–6290.

Linchao Zhu, Zhongwen Xu, Yi Yang, and Alexander G Hauptmann. 2017. Uncovering the temporal context for video question answering. *International Journal of Computer Vision*, 124(3):409–421.

	Videos	QA pairs	Question Type				
			What	Who	How	When	Where
Train	1200	30 933	19 485	10 479	736	161	72
Val	250	6 415	3995	2168	185	51	16
Test	520	13 157	8149	4552	370	58	28
All	1970	50 505	31 629	17 199	1291	270	116

Table 7: Statistics of MSVD-QA. The table shows the number of videos and question-answer pairs in the *train*, *validation*, and *test* splits as well as the number of questions per question type.

	Videos	QA pairs	Question Type				
			What	Who	How	When	Where
Train	6513	158 581	108 792	43 592	4067	1626	504
Val	497	12 278	8337	3439	344	106	52
Test	2990	72 821	49 869	20 385	1640	677	250
All	10 000	243 680	166 998	67 416	6051	2409	806

Table 8: Statistics of MSRVTT-QA. The table shows the number of videos and question-answer pairs in the *train*, *validation*, and *test* splits as well as the number of questions per question type.

A Appendix

A.1 Datasets

From Tables 7 and 8 we can see how the questions are not equally-distributed across all of the types. Question type *what* is the most diverse and accounts for 62.63% and 68.53% of the total number of questions in MSVD-QA and MSRVTT-QA, respectively. Our best VLCN model achieves new state-of-the-art performance on this question type across both datasets, i.e. 28.42% and 30.69% on MSVD-QA and MSRVTT-QA, respectively – a relative improvement of 4.12% and 2.79% over MA-DRNN (Yin et al., 2020) and QueST (Jiang et al., 2020).

A.2 Ablation Study

Tables 9 and 10 show the ensemble performance of our VLCN model and its ablated versions with respect to individual question types. On MSVD-QA, our full model achieves the best accuracy on the most diverse question type *what* and performs on par with its ablated versions on the remaining question types, i.e. *who*, *how*, *when*, and *where*. Similar results can be observed on MSRVTT-QA: Our full VLCN model achieves the best accuracy on the most diverse question type *what* as well as question type *when* and performs on par with the rest of its ablated versions on the remaining question types *who*, *how*, and *where*.

A.3 Transfer Learning

By observing the last two rows of Table 10, we can see the effect of transfer learning on the performance of our full VLCN model with respect to individual question types. In fact, by fine-tuning the fast-learning connections and the DNC weights inside the FLF block on MSRVTT-QA, we improved the performance on three different questions types, i.e. the most and second most diverse types *what* and *who* as well as question type *when*. This results in a new state-of-the-art overall accuracy of 36.01%.

A.4 Qualitative Analysis

We further show a qualitative example to highlight the supremacy of our full VLCN model over its ablated versions. In Figure 3, we can see how

Model	Question Type				
	What	Who	How	When	Where
MCAN_{avg}	26.94 ± 0.43	49.89 ± 0.43	$\mathbf{82.48 \pm 0.94}$	72.76 ± 0.69	45.71 ± 1.43
$\text{VLCN}-\text{FLF}_{\text{avg}}$	27.23 ± 0.58	50.77 ± 0.68	82.00 ± 1.00	$\mathbf{73.79 \pm 1.29}$	45.71 ± 5.72
$\text{VLCN}+\text{LSTM}_{\text{avg}}$	26.44 ± 0.69	$\mathbf{51.33 \pm 1.12}$	80.65 ± 3.41	72.06 ± 0.69	$\mathbf{47.14 \pm 5.25}$
VLCN_{avg}	$\mathbf{27.89 \pm 0.30}$	51.14 ± 0.18	81.08 ± 1.30	73.45 ± 0.85	46.43 ± 5.05

Table 9: Performance comparison of different ablated versions of our model on MSVD-QA *test*. The table shows the average accuracy and standard deviation $\mu \pm \sigma$ for each question type in %.

Model	Question Type				
	What	Who	How	When	Where
MCAN_{avg}	29.33 ± 0.03	$\mathbf{45.38 \pm 0.55}$	$\mathbf{83.33 \pm 0.61}$	75.07 ± 0.47	36.48 ± 1.35
$\text{VLCN}-\text{FLF}_{\text{avg}}$	29.15 ± 0.18	45.13 ± 0.24	83.01 ± 0.12	75.83 ± 0.76	37.28 ± 1.32
$\text{VLCN}+\text{LSTM}_{\text{avg}}$	28.92 ± 0.12	45.36 ± 0.32	83.06 ± 0.26	74.89 ± 1.57	$\mathbf{37.76 \pm 1.55}$
VLCN_{avg}	30.39 ± 0.07	43.92 ± 0.40	80.93 ± 0.90	76.87 ± 0.60	37.58 ± 1.48
$\text{VLCN}+\text{FT}_{\text{avg}}$	$\mathbf{30.59 \pm 0.10}$	44.27 ± 0.22	80.44 ± 1.14	$\mathbf{77.75 \pm 0.54}$	36.80 ± 0.44

Table 10: Performance comparison of different ablated versions of our model on MSRVTT-QA *test*. The table shows the average accuracy and standard deviation $\mu \pm \sigma$ for each question type in %.

both the language self-attention $SA(L)$ and the guided-attention over the frames $GA(L, F)$ are both flat indicating that the model is having difficulties aligning the multi-modal features. However, the guided-attention over the clips $GA(L, C)$ shows high attention values to the word *who* which is, in this case, the keyword to answer the question *who sat in his chair?* depicted in the orange box. While all of the ablated versions predict the wrong answer *lady*, our VLCN model answers the question correctly by predicting *man*.

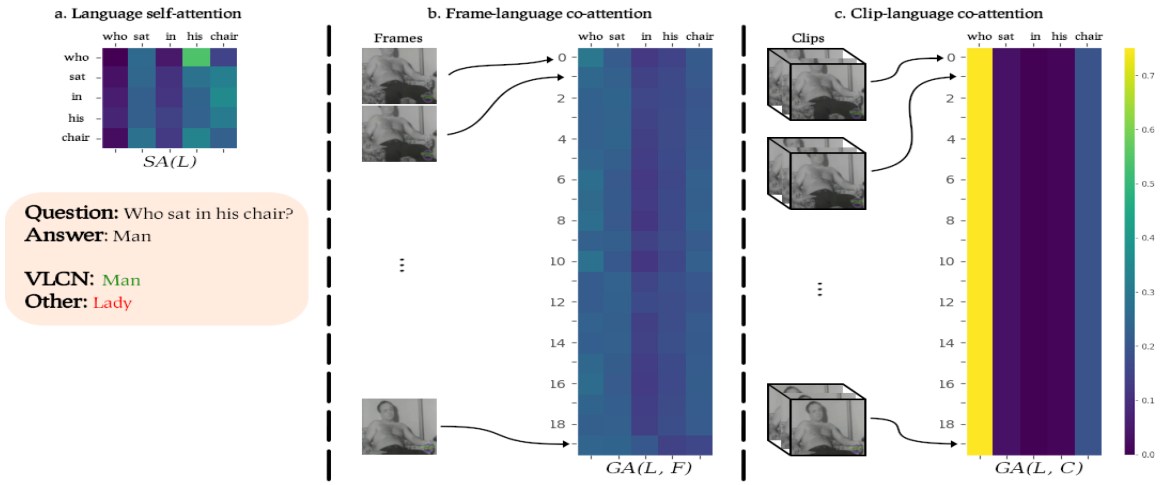

Figure 3: Visualization of the attention maps learned by the last video language co-attention layer. The indices [0, 19] indicate the individual 20 frames and clips of the video (some of which are shown).