Dual Capsule Attention Mask Network with Mutual Learning for Visual Question Answering

Weidong Tian[1,3], **Haodong Li**[1,3], **Zhong-Qiu Zhao**[1,2,3,4,*]
[1]College of Computer and Information, Hefei University of Technology
[2]Guangxi Academy of Science
[3]Intelligent Manufacturing Institute of HFUT
[4]Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology
wdtian@hfut.edu.cn, lhd_hfut@163.com, z.zhao@hfut.edu.cn

Abstract

A Visual Question Answering (VQA) model processes images and questions simultaneously with rich semantic information. The attention mechanism can highlight fine-grained features with critical information, thus ensuring that feature extraction emphasizes the objects related to the questions. However, unattended coarse-grained information is also essential for questions involving global elements. We believe that global coarse-grained information and local fine-grained information can complement each other to provide richer comprehensive information. In this paper, we propose a dual capsule attention mask network with mutual learning for VQA. Specifically, it contains two branches processing coarse-grained features and fine-grained features, respectively. We also design a novel stackable dual capsule attention module to fuse features and locate evidence. The two branches are combined to make final predictions for VQA. Experimental results show that our method outperforms the baselines in terms of VQA performance and interpretability and achieves new SOTA performance on the VQA-v2 dataset.

1 Introduction

In recent years, visual question answering (VQA) has received extensive research attention in the fields of computer vision and multimedia computing. The goal of VQA is to answer questions related to the content of images correctly (Antol et al., 2015). It has a wide range of practice applications, such as helping people with visual impairment and human-computer Q&A.

In the early stage, most VQA models extract features from images and questions independently (Malinowski et al., 2015; Gao et al., 2015; Ren et al., 2015). These methods fail to capture the fine-grained key features and include much unnecessary information. Afterward, the attention mechanism becomes popular and is introduced in many fields (Lu et al., 2016; Cai and Hu, 2020). The VQA models with the attention mechanism extract critical information from one modality guided by another modality (Lu et al., 2016; Anderson et al., 2018; Yu et al., 2019). Consider the question related to Figure 1 (Left) "What is on the little girl's head?" The attention method needs to encourage the model to focus on the "girls head" in the question and related regions in the image to produce the correct answer as "helmet." In this case, local fine-grained input(features which have critical information with attention processing) can help the model eliminate distractions and generate correct answers. However, the attention mechanism is not a panacea. Some questions may mislead visual attention and lead to wrong answers. For example, in Figure 6 (line 3, left), the question word "boxes" makes the model focus on the printer(which looks like a box) in the picture and leads to the wrong answer. Also, in some scenarios, the model needs to focus on multiple objects for reasoning, but the question cannot explicitly remind which object require its attention. For example, consider the ques-

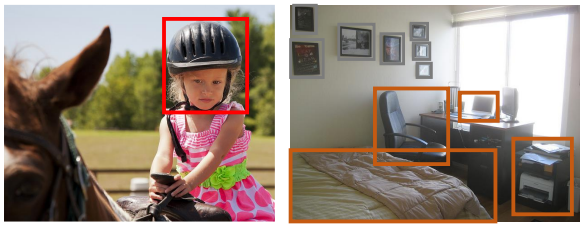


Figure 1: Samples in the VQA-v2 dataset. (Left): The fine-grained features with attention have the critical information required for the answer inference, which can help the model generate the correct answer by eliminating the interference of irrelevant factors. (Right): Unattended coarse-grained features have richer semantic information, which can help the answer inference when the attention mechanism is of limited use.

*Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 5678–5688
October 12–17, 2022.

tion related to Figure 1 (Right) "Could this be a multi-purpose room?" The model needs to take the bed, chair, computer, and printer into account for reasoning, but there are no words in the question that can help identify the relevant objects. In these cases, global coarse-grained input(features that have all information without attention processing) can provide more comprehensive information for generating answers. The challenge is to make the model focus on key features while maintaining a reference to the global information.

To overcome the above limitations, we propose a novel Dual Capsule Attention Mask Network (DCAMN) with mutual learning to process multimodal information at different granularities. We believe that coarse-grained information and fine-grained information can complement each other to provide richer comprehensive information for answer reasoning. Inspired by mutual learning and its variants (Zhang et al., 2018; Song and Chai, 2018), we design a two-branch network. The first branch processes the entire features of visual and language, analyzes global information at the coarse-grained level and fuses the features to produce predictions. We also design a Stackable Dual Capsule Attention Module (SDCAM) to model cross-modal deep interactions between the image and the question. The second branch masks visual features and language features with the attention weights generated by the SDCAM of the first branch to get fine-grained features, which enables the network to focus on key regions of the image and keywords of the question. Finally, we combine results from two branches to get final predictions. In contrast to other multi-granularity work (Nguyen et al., 2021), our DCAMN does not introduce additional information such as predicates, while utilizes the attention weights from the first branch to filter features for the second branch.

As a novel method of multi-modal fusion, SDCAM can output precise attention weights, which not only mask fine-grained features but also help locate evidence(grounds for the answer). By analyzing evidence, we can learn what information the network concerns more and how it makes decisions. Also, the stacking strategy for SDCAM improves attention accuracy and the fusion of visual and language representations. In DCAMN, there is only one language module to encode the questions, while images are presented to the two branches for processing separately. Compared with using two independent peer networks, sharing the language module between two branches can reduce the burden of parameters and calculation. Moreover, early blocks acquire gradients from both branches during backpropagation, which reduces the risk of gradient vanishing (Song and Chai, 2018). The knowledge of two branches at different perspectives and granularities is learned by another branch for information supplement and regularization, which can improve the generalization capability and VQA performance of DCAMN.

The main contributions of this paper are as follows: (1) A novel dual capsule attention mask network with mutual learning is proposed, which can process coarse-grained features and fine-grained features separately. Two branches can learn from each other, and their combination can improve the VQA performance. (2) A stackable dual capsule attention module is proposed, which provides precise co-attention weights for masking out features and locating evidence. (3) We propose to share the language module between two branches of different granularities, which can reduce parameter requirements and the risk of gradient vanishing. (4) Extensive experiments are conducted to evaluate the proposed method. Our method has significant advantages over the baselines in terms of interpretability and accuracy and achieves state-of-the-art performance on the VQA-v2 dataset.

2 Related Work

2.1 Visual Question Answering

The rapid development of VQA has benefited from many aspects. The latest studies in visual and language feature representation are applied to VQA to improve the ability to extract and process features (Jiang et al., 2020; Devlin et al., 2019). Better multimodal fusion methods, such as MCB (Fukui et al., 2016), MLB (Kim et al., 2017), MUTAN (Ben-younes et al., 2017), etc., are proposed to capture the high-level interactions between visual and language features. The transformer significantly contributes to the improvement of VQA (Yu et al., 2019; Zhou et al., 2021) and makes large-scale pre-training possible (Chen et al., 2020; Li et al., 2020). To extract useful information from the cumbersome features, many approaches introduce the attention mechanism to refine key information (Lu et al., 2016; Anderson et al., 2018; Gao et al., 2019; Yu et al., 2019). However, The attention mechanism may perform poorly on ques-

tions involving background. (Sharma and Jalal, 2022). Moreover, some words in the questions may mislead the question-based attention. These situations indicate that the attention mechanism is not a panacea, and a method that can integrate attention features and global features is necessary.

2.2 Mutual Learning

Most distillation-based model compression methods distill large and powerful networks into smaller and efficient networks (Romero et al., 2015; Zhang and Ma, 2021). However, its two-step strategy(train the teacher first and then train students) is time-consuming. So mutual learning is proposed (Zhang et al., 2018), which enables networks to be trained in parallel. In Zhang et al. (2018), each student model learns from the predictions of the other members, and the whole network requires complex asynchronous updates among different students. Another approach (Song and Chai, 2018) advocates that all students share the same early module, aggregating the gradient flow from all branches. This strategy reduces the training computational complexity and facilitates the supervision of the shared layers. Our network is based on the latter.

2.3 Capsule Network

The idea of grouping neurons is proposed early in Hinton et al. (2011). Following this, the dynamic routing method with capsules is formally introduced in Sabour et al. (2017). After dynamic routing, Hinton et al. (2018) implement EM routing of matrix capsules. The applications in other domains prove the universality of capsule networks (Duarte et al., 2018; Jaiswal et al., 2018; Zhao et al., 2019). Zhou et al. (2019), inspired by dynamic-routing implementation of capsule networks (Sabour et al., 2017), proposed CapsAtt model replacing the previously multi-level attentions.

3 Method

Given an image **I** and a question **Q**, the purpose of VQA is to output the correct answer $a \in \mathbf{A}$, where **A** indicates the candidate word list for answers. We follow the transformer design of MCAN (Yu et al., 2019) and TRAR (Zhou et al., 2021) and use them as our backbones with their encoder-decoder units. The overall framework of DCAMN is shown in Figure 2.

3.1 Question Representation and Image Representation

DCAMN models language features and visual features by encoding layers and decoding layers, respectively. The question **Q** is first processed by Glove word embedding (Pennington et al., 2014) accompanied by LSTM (Hochreiter and Schmidhuber, 1997) and then is presented to the encoding layers to get the question feature matrix $\mathbf{Y} \in \mathbb{R}^{k \times d}$, where d is the latent dimensionality in multi-head attention of encoder-decoder units, and k denotes the number of words in the question. Following BUTD (Anderson et al., 2018), we extract the salient region features $\mathbf{X}_0 \in \mathbb{R}^{m \times d}$ from the image **I** as the visual input by a pre-trained Faster R-CNN. $\mathbf{X}_0$ are then fed into two independent decoding structures to obtain the visual feature matrices $\mathbf{X}_t^1 \in \mathbb{R}^{m \times d}$ and $\mathbf{X}_t^2 \in \mathbb{R}^{m \times d}$ $(t = [1, 2, \cdots, T])$, where $\mathbf{X}_t^b$ represents the output of the t-th decoder block of branch b ($b \in [1, 2]$); T denotes the number of decoding layers; m is the number of bounding boxes.

3.2 Stackable Dual Capsule Attention Module and Decoder Block

We design an efficient multi-modal fusion method SDCAM. The attention of SDCAM is transformed from capsule dynamic routing, and the stacking strategy is also employed to improve visual-language co-attention performance. The details of SDCAM are shown in Figure 3. Specifically, it alternately performs attention and fusion on visual features and language features. The visual input **X** of SDCAM is $\mathbf{X}_t^1$ or $\mathbf{X}_t^2$ and the language input is **Y**, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]$, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k]$. In SDCAM, $\mathbf{x}_i (i = 1, \cdots, m)$ denotes the image feature of the i-th bounding box and $\mathbf{y}_i (i = 1, \cdots, k)$ denotes the language feature of the i-th word, and they are considered as the underlying capsules which need to be routed.

First, the high-level capsule $\mathbf{S}_t$ of the visual attention module is initialized by the fusion feature $\mathbf{F}_{t-1}$. When $t = 1$, $\mathbf{F}_0$ is a language feature processed by the reduction model, which is a simple self-attention module proposed in Yu et al. (2019). The high-level capsule for the visual attention module is calculated by:

$$\mathbf{S}_t^{p+1} = \mathbf{S}_t^p + \sum_{i=1}^{m} c_i \mathbf{x}_i^w, \tag{1}$$

$$\mathbf{x}_i^w = \sigma(\mathbf{W}_x \mathbf{x}_i), i \in [1, m], \tag{2}$$

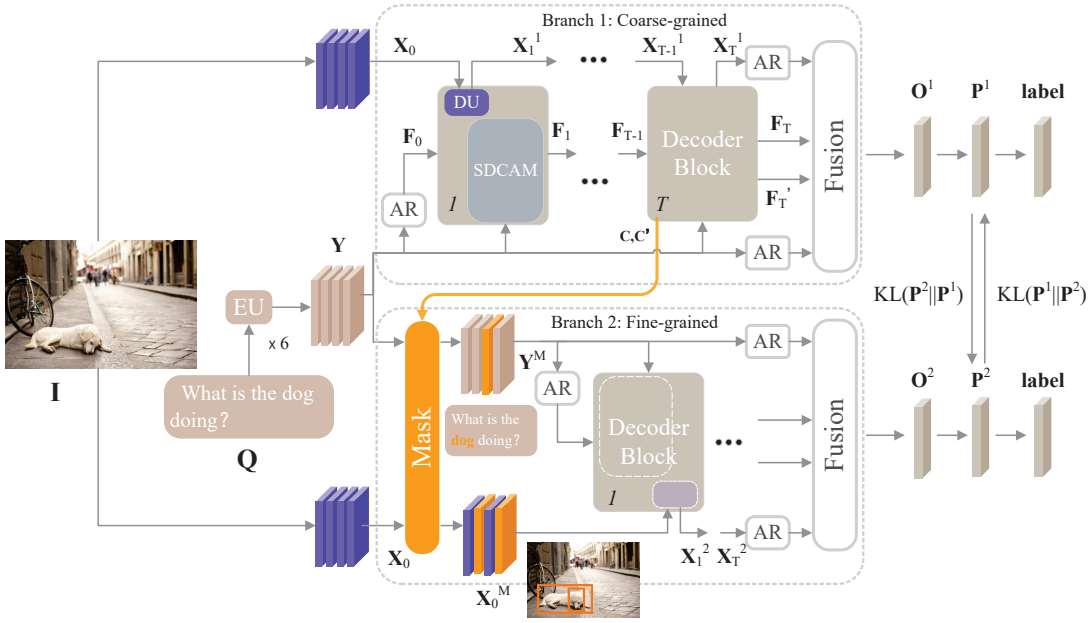


Figure 2: The overall framework of DCAMN. Coarse-grained branch 1 and fine-grained branch 2 handle coarse-grained features and fine-grained features, respectively. The fine-grained features are obtained by masking with attention weights provided by the last SDCAM of branch 1. AR, EU, and DU indicate reduction model, encoder unit, and decoder unit, respectively. We combine the two predictions $\mathbf{P}^1$ and $\mathbf{P}^2$ to get the final prediction.

where $\mathbf{S}_t^p$ denotes the high-level capsule of the visual attention module in the t-th SDCAM unit after $p \in [0, R-1]$ routing iterations, R represents the total number of routing iterations, c_i is the coupling coefficient between the underlying capsule $\mathbf{x}_i$ and high-level capsule $\mathbf{S}_t$, which can be interpreted as the contribution to the high-level capsule, $\mathbf{x}_i$ is multiplied by the projection matrix $\mathbf{W}_x \in \mathbb{R}^{d \times d}$ and passed through the activation function σ to get $\mathbf{x}_i^w$.

The value of the coupling coefficient c_i depends on b_i. In the routing algorithm, the value of b_i is used to measure the similarity between the underlying capsule $\mathbf{x}_i$ and the high-level capsule $\mathbf{S}_t$, and is obtained by:

$$b_i = b_i + (\mathbf{x}_i^w)^{\mathsf{T}} \cdot \mathbf{S}_t^p, i \in [1, m]. \quad (3)$$

We initialize b_i to be 0. In each routing iteration, b_i is added with the dot product of the high-level capsule and the underlying capsule for updating, and then softmax is applied to obtain the coupling coefficient c_i as follows:

$$c_i = softmax(b_i), i \in [1, m]. \quad (4)$$

When the processing of the visual attention module finishes, the fusion feature $\mathbf{S}_t$ will be used to initialize the high-level capsule $\mathbf{S}_t^{'}$ of the language attention module. The language attention module is similar to the visual one, except that the input of the language attention module is $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k]$.

After the language attention module, $\mathbf{S}_t^{'}$, which is also denoted as the fusion feature output $\mathbf{F}_t$ of the SDCAM, is fed to the next SDCAM for initializing its high-level feature $\mathbf{S}_{t+1}$. The whole procedure is defined as:

$$\mathbf{F}_t, \mathbf{F}_t^{'} = \mathrm{SDCAM}_t(\mathbf{X}, \mathbf{Y}, \mathbf{F}_{t-1}), t \in [1, T]. \quad (5)$$

In decoder block t, visual features and language features are fed to a DU and a SDCAM for processing:

$$\mathbf{X}_t = \mathrm{DU}_t(\mathbf{X}_{t-1}, \mathbf{Y}), \quad (6)$$

$$\mathbf{F}_t, \mathbf{F}_t^{'} = \mathrm{SDCAM}_t(\mathbf{X}_t, \mathbf{Y}, \mathbf{F}_{t-1}), t \in [1, T], \quad (7)$$

where DU_t is the t-th decoder unit and denotes a Guided-Attention Unit (Yu et al., 2019). The DU_t takes $\mathbf{X}_{t-1}$ and $\mathbf{Y}$ as inputs. Its output $\mathbf{X}_t$ is fed to SDCAM_t and the next decoder block. The whole procedure of decoder block t is defined as:

$$\mathbf{F}_t, \mathbf{F}_t^{'}, \mathbf{c}, \mathbf{c}^{'}, \mathbf{X}_t = \mathrm{DecoderBlock}_t(\mathbf{X}_{t-1}, \mathbf{Y}, \mathbf{F}_{t-1}), t \in [1, T], \quad (8)$$

where $\mathbf{c}$ and $\mathbf{c}^{'}$ are coupling coefficients generated from the SDCAM, which can be used to label the weights of bounding boxes and question words, respectively. In this way, we can learn what the model concerns in the inference process.

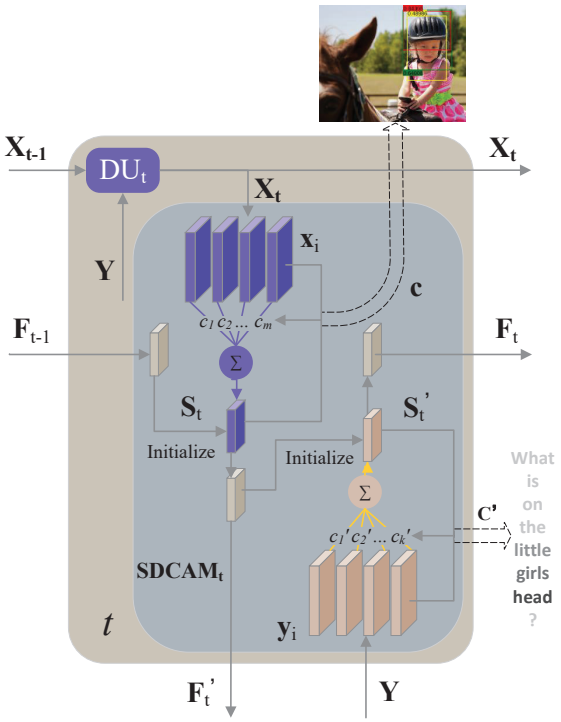


Figure 3: Illustration of decoder block t. It consists of a SDCAM and a DU. The attention map of the image is obtained by labeling bounding boxes with the weight $\mathbf{c}$ from SDCAM, and the importance of each word in the question is reflected by the weight $\mathbf{c}'$.

3.3 Double Branches with Coarse and Fine Granularities

In coarse-grained branch 1, coarse-grained visual feature $\mathbf{X}_0$ and language feature $\mathbf{Y}$ are sent to T decoder blocks to obtain $\mathbf{F}_T$, $\mathbf{F}_T'$ and $\mathbf{X}_T^1$. Note that we let all SDCAMs in decoder blocks of branch 1 share parameters to reduce the parameter burden. After this, $\mathbf{X}_T^1$ and $\mathbf{Y}$ pass through the attentional reduction model and are summed. The result is then concatenated with the sum of $\mathbf{F}_T$ and $\mathbf{F}_T'$ to get the fused feature $\mathbf{U}^1 \in \mathbb{R}^{d_f}$, where d_f is the dimension of the fused feature. Afterward, $\mathbf{U}^1$ is passed through a linear layer to get the vector $\mathbf{O}^1 \in \mathbb{R}^N$, where N indicates the number of candidate answers.

The fine-grained visual features and language features are processed in fine-grained branch 2 which has independent decoder blocks. To obtain critical features with visual-language relevance, we mask $\mathbf{X}_0$ and $\mathbf{Y}$ with the attention weights from the last SDCAM of coarse-grained branch 1. Specifically, in the visual attention module of SDCAM, the coupling coefficient $c_i (i = [1, 2, \cdots, m])$ corresponds to the visual feature $\mathbf{x}_i \in \mathbf{X}_0 (i = [1, 2, \cdots, m])$. We sort $\mathbf{x}_i$ from large to small according to the coupling coefficient c_i. The bottom $p\%$ of $\mathbf{x}_i$ are selected to be multiplied by the mask coefficient $\gamma (< 1)$ while the rest remain unchanged. The language feature $\mathbf{Y}$ is processed in the same way. Mask enables the network to focus on critical information. For example, in Figure 2, the word "dog" in the question and the bounding boxes about the dog are more significant after masking.

In fine-grained branch 2, the masked features $\mathbf{X}_0^M$ and $\mathbf{Y}^M$ are processed in the same way as branch 1. They are sent to T decoder blocks and fused to obtain $\mathbf{U}^2 \in \mathbb{R}^{d_f}$ and the vector $\mathbf{O}^2 \in \mathbb{R}^N$.

We apply a softmax function on $\mathbf{O}^1$ and $\mathbf{O}^2$ to obtain predictions of branch 1 and branch 2:

$$\mathbf{P}^i = softmax(\mathbf{O}^i), i \in [1, 2], \tag{9}$$

where $\mathbf{P}^1$ and $\mathbf{P}^2$ are predictions of branch 1 and branch 2, respectively. We sum $\mathbf{P}^1$ and $\mathbf{P}^2$ and take the candidate answer that has the maximum probability in their results as the final answer a.

3.4 Mutual Learning and Loss Function

For each branch, we utilize the prediction from the other branch to supervise. The Kullback Leibler (KL) Divergence is used to calculate the loss of mutual learning:

$$L_M = D_{kl}(\mathbf{P}^2 \| \mathbf{P}^1) + D_{kl}(\mathbf{P}^1 \| \mathbf{P}^2). \tag{10}$$

L_M forces the two branches to match the probability distribution of each other and learn from each other. In addition to L_M, binary cross-entropy (BCE) is employed to compute the loss between predictions and labels:

$$\begin{aligned} L_{BCE} = & BCE(sigmoid(\mathbf{O}^1), \mathbf{label}) \\ & + BCE(sigmoid(\mathbf{O}^2), \mathbf{label}). \end{aligned} \tag{11}$$

The two branches independently compute their BCE losses. The total loss is the sum of L_M and L_{BCE}.

In contrast to the tedious training method of traditional mutual learning (Zhang et al., 2018) that performs asynchronous updates among different sub-networks, DCAMN only has a total loss L, which means that parameter optimization in different branches can be performed simultaneously and training efficiency can be improved.

4 Experiments

4.1 Datasets

We use the VQA-v2 dataset (Goyal et al., 2017) to evaluate our model. The dataset is split into three

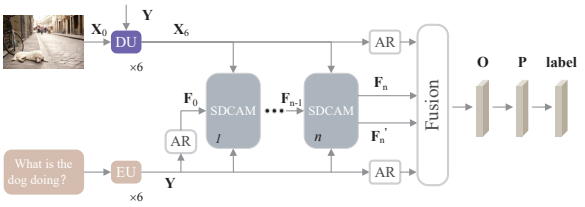


Figure 4: SDCAM-last-layer-n.

subsets: *train*, *val*, and *test*, which contain 80k, 40k, and 80k images and 444k, 214k, and 448k QA pairs, respectively. It has three types of questions: *yes/no*, *number*, and *other*. *test-std* and *test-dev*, as the subsets of the *test* set, are provided to evaluate model performance online. In addition, we also use the VQA-CPv2 dataset (Agrawal et al., 2018) to test models' robustness for the question biases. The VQA-CPv2 dataset is a VQA dataset with a particular answer distribution used to evaluate the ability of the model against language priors.

4.2 Implementation Details

The hyperparameter setting of our method is described in this subsection. We set the hidden layer dimension of LSTM to be 512. The dimension d_f of the fused feature is 1024. The latent dimensionality d in multi-head attention is set to be 512. Features in the multi-head attention are split into 8 heads with 64 dimensions for each head. We set the number of candidate answers N and the number of decoding layers T to 3129 and 6. Following Sabour et al. (2017), the number of routing iterations R is set to 3. We set mask coefficient γ and mask probability p to 0.5 and 30, which are experimentally selected.

We set the number of epochs to 15. The batch size is 64. The learning rate is initialized to $2.5e^{-5}$, gradually grows to $1e^{-4}$, and is decayed by 0.2 in the last three epochs. We use the adam optimizer and set the parameters β_1 and β_2 to be 0.9 and 0.98, respectively. We use the *train* and *val* sets for training and a subset of Visual Genome (Krishna et al., 2017) for data augmentation, and test for online evaluation on *test-dev* and *test-std*.

4.3 Ablation Studies

We use MCAN (Yu et al., 2019) and TRAR (Zhou et al., 2021) as the baselines. SDCAM-last-layer-n represents that the final output of encoder-decoder is processed by n SDCAMs in a single branch, as illustrated in Figure 4. Table 1 shows that SDCAM-last outperforms the baseline MCAN, which validates the effectiveness of SDCAM as a novel fusion method. SDCAM-mid indicates that the intermediate features from decoder units are processed by SDCAM, which is similar to coarse-grained branch 1 of Figure 2. SDCAM-mid outperforms SDCAM-last, which demonstrates the effectiveness of reusing intermediate information. Besides improving VQA performance, SDCAM has a further contribution to enhancing the interpretability of DCAMN. From the result of Table 1 and Table 2, we can see that each component (mutual learning, SDCAM, and the masking strategy) contributes to performance improvement. Moreover, The mutual learning in DCAMN will degenerate into the modules ensemble if the additional loss L_M is removed. By comparing the performance of DCAMN with or without KLloss, we can determine that DCAMN is not just a model simply using the modules ensemble because it enables branches to learn from each other to improve effectiveness.

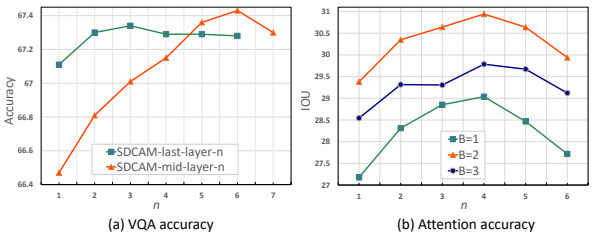
Figure 5(a) shows that the VQA accuracy of SDCAM-last and SDCAM-mid grows and saturates as the number of stacks increases. This proves that stacking can improve the performance of SDCAM and enable it to perform a better fusion.

Model	All	Other	Yes/No	Num.
MCAN(baseline)	67.2	58.7	84.84	48.69
SDCAM-last-layer-3	67.34	58.81	85.15	48.42
SDCAM-mid-layer-6	67.43	58.89	85.15	48.78
SDCAM-mid+Mutual	68.05	59.63	85.60	49.43
SDCAM-mid+Mutua+Mask	**68.14**	**59.66**	**85.74**	**49.65**

Table 1: Ablation studies using MCAN as the backbone on VQA-v2 *val* set. Mutual means using the two-branch mutual learning strategy. Mask means using our masking mechanism.

SA	TD	SDCAM	Mutual	Mask	KLloss	Accuracy
						67.6
✓						67.48
	✓					67.55
		✓				67.76
			✓		✓	68.27
		✓	✓		✓	68.30
✓			✓	✓	✓	68.09
	✓		✓	✓	✓	68.19
		✓	✓	✓		68.00
		✓	✓	✓	✓	68.40

Table 2: Ablation studies using TRAR as the backbone on VQA-v2 *val* set. SA and TD mean using the Self-Attention in MCAN and the Top-Down mechanism in BUTD for co-attention to replace SDCAM, respectively. KLloss means the loss L_M.



(a) VQA accuracy (b) Attention accuracy

Figure 5: Effect of SDCAM stacking numbers n on VQA accuracy and attention accuracy. layer-n indicates that the SDCAM is stacked n times. The attention weights used for the attention accuracy calculation are provided by the n-th SDCAM of SDCAM-mid-layer-n.

Method	B=1	B=2	B=3	B=AU
DFAF (2019)	5.07	11.27	14.18	17.55
MCAN (2019)	15.43	18.77	20.01	23.91
BUTD (2018)	26.46	27.80	26.92	33.97
SDCAM-mid-layer-6	27.72	29.94	29.12	37.15
DCAMN-branch-1	28.05	30.26	29.21	37.38
DCAMN-branch-2	28.29	30.25	28.98	37.18
DCAMN-branch-sum	28.78	30.68	29.45	37.80
SDCAM-mid-layer-4	**29.04**	**30.94**	**29.79**	**37.93**

Table 3: Attention accuracy comparison of various methods. B is the number of candidate bounding boxes. DCAMN-branch-b denotes the attention accuracy of the last SDCAM in branch b of DCAMN. branch-sum means attention weights of two branches are added together. We evaluate methods on the VQS validation set.

4.4 Qualitative Analysis

To explore the inference process of the model and determine whether the correct answer is obtained by reasoning based on features rather than exploiting the statistics of the dataset, we visualize the attention weights of SDCAM in Figure 6.

From the first row, we can see that SDCAM achieves more accurate attention both on visual regions and on question words. In the first question, SDCAM locates the "drainage" in the image accurately and gets the correct answer. By analyzing examples of MCAN and BUTD, we can see that the models usually get the wrong answers when they focus on the incorrect visual regions and words. This demonstrates the ability of SDCAM to find evidence and help analyze the reasoning process.

Moreover, from the second row, we can observe that as the number of stacked layers increases, the interest regions of SDCAM tend to be more accurate, and the distribution of weights tends to be more focused. In the question "What is being celebrated?" SDCAM focuses on the people at first, but after several iterations, the center of attention shifts to the cake, and the model finally gets the correct answer. The next question is similar to this one. These cases prove that SDCAM can focus on different objects exactly as reasoning requires.

In the third row, we show some examples of wrong predictions. With the attention map of SDCAM, we can know the reason for wrong predictions. For example, in the question "Are there any boxes in the room?" SDCAM focuses on the printer that looks like a box and incorrectly answers yes.

4.5 Analysis of the Attention Accuracy

We quantify the accuracy of visual attention of SDCAM and other methods, as shown in Table 3. VQS is a dataset in which for each triad (image, question, and answer), there is a corresponding image segmentation mask depicting the contours which need attention in the image (Gan et al., 2017). We calculate the Intersection of Union (IOU) between the VQS segmentation mask and the attention mask of DCAMN as the metric to evaluate the attention accuracy, where the attention mask of SDCAM is the union of top B bounding boxes in terms of the attention weights rank. B=AU (automatically) indicates that the optimal result for different bounding box numbers is incorporated into the overall IOU statistics. From Table 3, we can see that SDCAM has higher attention accuracy than other methods. The results of two-branches DCAMN are not better than those of single branch SDCAM, which indicates that SDCAM attention accuracy is hardly influenced by mutual learning.

In Figure 5(b), we show the performance of SDCAM-mid with different stacking numbers n. With increasing n, the attention accuracy becomes higher and reaches a maximum at $n = 4$. This confirms that the stacking strategy can help SDCAM focus on the correct regions.

4.6 Comparison with SOTAs

We compare our DCAMN [1] with the state-of-the-art methods on the VQA-v2 dataset, and the results are shown in Table 4. We can see that DCAMN outperforms other methods and $\text{DCAMN}_{\text{mcan}}$ and $\text{DCAMN}_{\text{trar}}$ have higher performance than the baseline MCAN and TRAR, respectively, which proves the validity of the proposed method. It is worth noting that for the baseline TRAR, the improvement on the number type question is particu-

[1]Code is available at https://github.com/HFUTLHD/DCAMN-VQA-master.

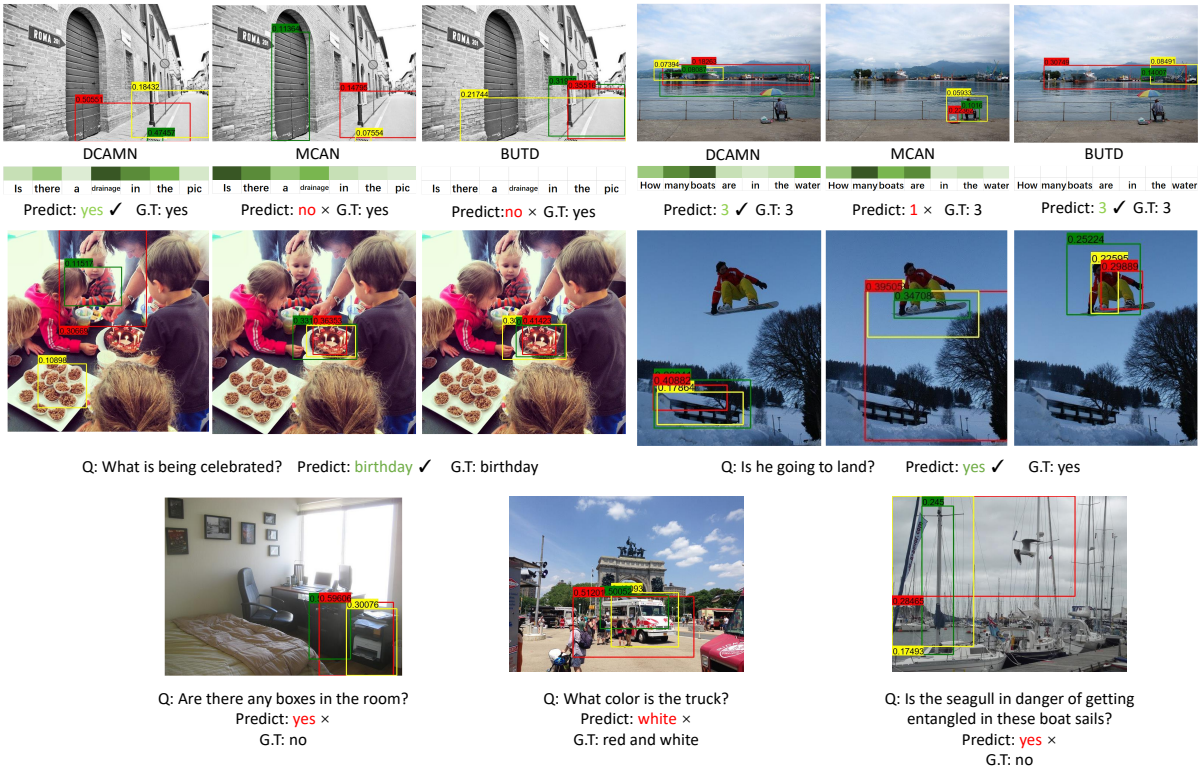


Figure 6: Visualization of attention weights for DCAMN, MCAN and BUTD on visual regions and question words. We mark the top three bounding boxes based on attention weights rank and label their weights above. The second row shows the visual attention distribution of the image in the 1st, 3rd, and 6th SDCAM of DCAMN. The third row shows some incorrect samples.

	Test-dev				Test-std
Method	All	Yes/no	Num.	Others	All
BUTD (2018)	65.32	81.82	44.21	56.05	65.67
DFAF (2019)	70.22	86.09	53.32	60.49	70.34
CFR (2021)	72.5	-	-	-	-
MCAN (2019)	70.63	86.82	53.26	60.72	70.90
TRAR (2021)	72.62	88.11	55.33	63.31	72.93
$\text{DCAMN}_{\text{mcan}}$	71.77	87.80	53.96	62.12	72.19
$\text{DCAMN}_{\text{trar}}$*	**73.67**	**88.86**	**58.02**	**64.22**	**74.05**

Table 4: VQA performance comparison with state-of-the-art approaches on the VQA-v2 dataset. $\text{DCAMN}_{\text{mcan}}$ and $\text{DCAMN}_{\text{trar}}$ denote DCAMNs using MCAN and TRAR as the backbone, respectively. * means using the 16 × 16 grid features.

larly significant, with a 2.7 point gain. We attribute such a marked improvement to the fact that the objects which require counting are more prominent after masking, which facilitates more accurate modeling in the second branch. From the results in Table 5, we can see that DCAMN outperforms the baselines and other methods on the VQA-CPv2 dataset, which proves the effectiveness of DCAMN against the language priors.

Method	Accuracy
BUTD (2018)	39.06
QCG (2018)	39.32
BAN (2018)	40.06
MCAN (2019)	43.29
TRAR (2021)	42.30
$\text{DCAMN}_{\text{trar}}$	43.03
$\text{DCAMN}_{\text{mcan}}$	**44.09**

Table 5: VQA performance comparison with other methods on VQA-CPv2 test.

5 Conclusion

In this paper, we propose a dual capsule attention mask network for VQA. DCAMN can process features at different granularities to take global information into account and focus on critical information. Combining the views of the two branches at different perspectives and granularities can improve the generalization capability of the model and make more accurate predictions. In addition, the proposed SDCAM can effectively fuse multimodal features and locate evidence, which also enhances the interpretation capability of the network. Experiments show that DCAMN outperforms other methods in terms of interpretability and accuracy and achieves new SOTA performance for VQA.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 61976079, the Open Project of the Key Laboratory of Digital Publishing and Knowledge Service of Cultural Resources of the State Press and Publication Administration of China under Grant 104-4331201902, the Guangxi Key Research and Development Program under Grant AB22035022, and Anhui Key Research and Development Program under Grant 202004a05020039.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4971–4980. Computer Vision Foundation / IEEE Computer Society.

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 *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. Computer Vision Foundation / IEEE Computer Society.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433. IEEE Computer Society.

Hedi Ben-younes, Rémi Cadène, Matthieu Cord, and Nicolas Thome. 2017. MUTAN: multimodal tucker fusion for visual question answering. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2631–2639. IEEE Computer Society.

Jiahui Cai and Jianguo Hu. 2020. 3d rans: 3d residual attention networks for action recognition. *Vis. Comput.*, 36(6):1261–1270.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: universal image-text representation learning. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings*, volume 12375 of *Lecture Notes in Computer Science*, pages 104–120. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Duarte, Yogesh S. Rawat, and Mubarak Shah. 2018. Videocapsulenet: A simplified network for action detection. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7621–7630.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, Austin, Texas. Association for Computational Linguistics.

Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. 2017. VQS: linking segmentations to questions and answers for supervised attention in VQA and question-focused semantic segmentation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1829–1838. IEEE Computer Society.

Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. Are you talking to a machine? dataset and methods for multilingual image question. In *Advances in neural information processing systems*, pages 2296–2304.

Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven C. H. Hoi, Xiaogang Wang, and Hongsheng Li. 2019. Dynamic fusion with intra- and inter-modality attention flow for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6639–6648. Computer Vision Foundation / IEEE.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6325–6334. IEEE Computer Society.

Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. 2011. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning - ICANN 2011 - 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I*, volume 6791 of *Lecture Notes in Computer Science*, pages 44–51. Springer.

Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix capsules with EM routing. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, and Premkumar Natarajan. 2018. Capsulegan: Generative adversarial capsule network. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11131 of *Lecture Notes in Computer Science*, pages 526–535. Springer.

Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik G. Learned-Miller, and Xinlei Chen. 2020. In defense of grid features for visual question answering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10264–10273. Computer Vision Foundation / IEEE.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1571–1581.

Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Hadamard product for low-rank bilinear pooling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*, volume 12375 of *Lecture Notes in Computer Science*, pages 121–137. Springer.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*, pages 289–297.

Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. 2015. Ask your neurons: A neural-based approach to answering questions about images. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1–9. IEEE Computer Society.

Binh X. Nguyen, Tuong Do, Huy Tran, Erman Tjiputra, Quang D. Tran, and Anh Nguyen. 2021. Coarse-to-fine reasoning for visual question answering. *CoRR*, abs/2110.02526.

Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. 2018. Learning conditioned graph structures for interpretable visual question answering. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8344–8353.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Mengye Ren, Ryan Kiros, and Richard S. Zemel. 2015. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2953–2961.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. Fitnets: Hints for thin deep nets. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3856–3866.

Himanshu Sharma and Anand Singh Jalal. 2022. An improved attention and hybrid optimization technique for visual question answering. *Neural Process. Lett.*, 54(1):709–730.

Guocong Song and Wei Chai. 2018. Collaborative learning for deep neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1837–1846.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6281–6290. Computer Vision Foundation / IEEE.

Linfeng Zhang and Kaisheng Ma. 2021. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4320–4328. Computer Vision Foundation / IEEE Computer Society.

Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 2019. 3d point capsule networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1009–1018. Computer Vision Foundation / IEEE.

Yiyi Zhou, Rongrong Ji, Jinsong Su, Xiaoshuai Sun, and Weiqiu Chen. 2019. Dynamic capsule attention for visual question answering. In *Proceedings of the AAAI conference on artificial intelligence*, pages 9324–9331. AAAI Press.

Yiyi Zhou, Tianhe Ren, Chaoyang Zhu, Xiaoshuai Sun, Jianzhuang Liu, Xinghao Ding, Mingliang Xu, and Rongrong Ji. 2021. TRAR: routing the attention spans in transformer for visual question answering. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2054–2064. IEEE.