Object-Difference Attention: A Simple Relational Attention for Visual Question Answering

Chenfei Wu, Jinlai Liu, Xiaojie Wang, Xuan Dong
Center for Intelligence Science and Technology, School of Computer Science,
Beijing University of Posts and Telecommunications
{wuchenfei,liujinlai,xjwang,dongxuan8811}@bupt.edu.cn

ABSTRACT

Attention mechanism has greatly promoted the development of *Visual Question Answering* (VQA). Attention distribution, which weights differently on objects (such as image regions or bounding boxes) in an image according to their importance for answering a question, plays a crucial role in attention mechanism. Most of the existing work focuses on fusing image features and text features to calculate the attention distribution without comparisons between different image objects. As a major property of attention, selectivity depends on comparisons between different objects. Comparisons provide more information for assigning attentions better. For achieving this, we propose an object-difference attention (ODA) which calculates the probability of attention by implementing difference operator between different image objects in an image under the guidance of questions in hand. Experimental results on three publicly available datasets show our ODA based VQA model achieves the state-of-the-art results. Furthermore, a general form of relational attention is proposed. Besides ODA, several other relational attentions are given. Experimental results show those relational attentions have strengths on different types of questions.

CCS CONCEPTS

• **Information systems** → **Question answering**; • **Computing methodologies** → **Computer vision tasks**;

KEYWORDS

Attention; Object-Difference Attention; Visual Question Answering

ACM Reference Format:
Chenfei Wu, Jinlai Liu, Xiaojie Wang, Xuan Dong. 2018. Object-Difference Attention: A Simple Relational Attention for Visual Question Answering. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240513

1 INTRODUCTION

The goal of *Visual Question Answering* (VQA) task is to output an answer for an input image and a related question. It is an essential cognitive capability of human beings, which involve both visual and linguistic computing to infer the answer. Building a machine that performs VQA as well as humans is a big challenge in the artificial intelligence community.

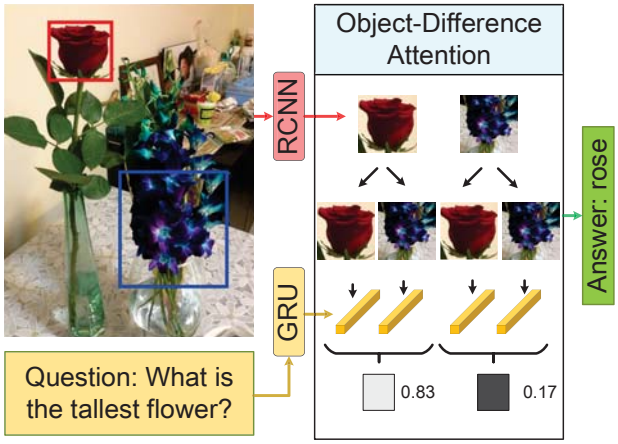

Figure 1: The proposed Object-Difference Attention Model. "Rose" is focused because of not only "rose" itself and the question, but also "the shorter orchid beside it."

Researchers from both computer vision and natural language processing have done lots of work on developing various VQA models and algorithms [13, 23], especially with the recent advances in deep neural networks [3, 24, 35]. Among them, attention mechanism, which was first introduced in [6], has been widely used in VQA [20, 21, 25, 30, 31, 34, 35, 38]. It plays a core role in various VQA models.

The nature of current attention mechanism is to assign a proper attention distribution on different objects (e.g., image regions, bounding boxes) in an image so that more attentions are paid on objects with more useful information for answering questions. Therefore, it is a major challenge to work out a proper attention distribution for a problem on hand. Many different attention models have been proposed in previous VQA work. Initially, a one-step linear fusion is proposed to calculate the attention distribution on objects [6, 20]. Later, some studies calculate more accurate attention distribution on objects through multi-step linear fusion [34, 35, 35]. Recent studies use bilinear fusion to further increase the accuracy of the attention distribution on objects [5, 8, 14, 36]. Most recently, multi-feature attention is performed to calculate multi-feature attention distribution on objects [12, 21, 30].

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240513

As we can see, an attention distribution is the normalized importance of each object in an image for answering a given question. Most of the current attention-based VQA models calculate the importance of an object by considering only the question and the object itself, although the importance is finally normalized over all objects. The object itself is, of course, a valuable factor in showing its importance for question answering, especially when the question focuses on a specific object. For example, for answering a question like *"What sport is the man playing?"* the object *"man"* might be paid the most prominent attention.

However, objects themselves are often insufficient to provide enough information on assigning proper attention distributions for lots of other types of questions. For example, for answering a question *"What is the tallest flower?"* accompanied with an image shown in Figure 1, we find it cannot receive a correct answer when a model assigns importance only depending on each flower separately. The intuition behind the situation is clear. For answering the question, all flowers should be found at first. There are a rose and an orchid in the image. Then, comparisons between different flowers are vital for giving the correct answer. Figure 1 shows both rose-based comparisons and orchid-based comparisons. The correct answer is achieved by comparisons between different objects.

The above case shows that comparisons provide more information on how to assign attentions on different objects. It is not the only case for needs of comparison. We argue that comparison is helpful and even necessary in most of the question answering. Comparison is helpful to identify which object is exactly you want; comparison is beneficial to locate the position of an object; comparison is necessary to determine if two objects belong to the same class, and so on. The problem is therefore how to include information of comparisons between different objects in attention assignment.

This paper proposes a new type of attention, which is named object-difference attention (ODA), to address the problem. For a given image and a question, information from both the question and comparisons between different objects in the image is used to assign attentions on each object. The comparison is implemented by a difference operator between two objects. ODA has lower computational complexity than some previous attentions such as that in Mutan [5]. Experimental results show our ODA based VQA model achieves the state-of-the-art performance in three different publicly available datasets, including VQA1.0, VQA2.0, and COCO-QA.

Furthermore, we extend ODA to general relational attention, and give several different types of relational attentions with different relational kernels. Experimental results show attentions with different kernels have strength in different types of question. Among them, ODA shows advantages in most types of question.

The remainders of the paper are organized as follows: Section 2 is related work, Section 3 brings our ODA model for VQA task. In section 4, we introduce a general form of relational attention and give several relational attentions with different relational kernels. Section 5 describes experiments on three public availabel datasets. Section 6 draws some conclusions.

In summary, the contributions of this paper are threefold:

- We propose a simple but effective object-difference attention (ODA) that compares objects explicitly when calculating the attention distribution.
- We introduce a general form of relational attention, which is different from previous single object based attention, and give several different attention kernels.
- We achieve the state-of-the-art results on three publicly available datasets including VQA 1.0, VQA 2.0, and COCO-QA. Different relational attentions show strengths on different types of question.

2 RELATED WORK

Most of the previous work on VQA employs attention mechanism, which dramatically improves the performance of VQA task.

Initially, a one-step linear fusion is used to calculate the attention distribution. For example, the ABC-CNN model [6] maps the question features to the visual space, and then computes the attention distribution of the image region by convolving the question features and the image features. The CoAtt model [20] connects the image feature and the question feature with a trainable parameter to obtain an affinity matrix. The image feature space and the question feature space can then be transformed with this matrix, and finally merged with a linear function to obtain the attention distribution.

Some work tries multi-step attention. The SAN model [35] maps the visual features into vectors with same dimensions as the question features. The first-step attention distribution on image regions is a linear combination of the visual features and the question features. The attention distribution is updated by fusing the visual features with updated question features. The SMem model [34] fuses image features and question features through a correlation matrix, and then sums the matrix in the textual dimension to obtain the first-hop attention distribution of image regions. Similar to the SAN, Smem takes the sum of the image features and the question features as new question features. The second-hop attention distribution is calculated by fusing these question features with image features again. The DAN model [25] also performs multiple attentions, but the image features and text features are activated separately with tanh before fused by element-wise multiplication.

Some recent work uses bilinear models to fuse image features and question features. The MCB model [14] first introduces the bilinear pooling operation to fuse image features and question features. It has big dimensions of features. To reduce the output dimension, the MLB model [36] uses Hadamard product operations to fuse image features with text features. Compared with MCB, MLB can output features that are more compact. However, it converges slowly and is sensitive to hyper-parameters. The MFB model [36] combines the advantages of MCB and MLB for both capacity and compact features. A one-dimensional non-overlapped window is used to perform sum pooling over the Hadamard product of image features and question features. The Mutan model [5] uses Tucker decomposition tensor and achieves similar expression.

Most recently, multi-feature attention is performed to calculate multi-feature attention distribution. The High-Order Attention Model [30] calculates the attention distribution of the image, the question, and the answer separately by learning high-order correlations between these features. The ReasonNet Model [12] learns to reason over a series of input features including face analysis, object classification, scene classification separately at first, then combine

the results to infer the answer. Similarly, The DualMFA model [21] uses free-form regions and detection boxes as the input features.

As we can see, previous work on the attention mechanism in the VQA task do not model the relation between objects to determine the attention distribution for answering questions.

3 ODA FOR VQA

The overall structure of our model for VQA is illustrated in Figure 2. It consists of three parts: Data Embedding, Object-Difference Attention (ODA), and Decision Making. Data Embedding extracts image features from RCNN and question features from GRU. ODA calculates an attention distribution for objects in image by explicitly comparing each image object with all other objects under the guidance of the question. Decision Making fuses the image features and question features to select the answer of the question.

We give the details of the three parts in following three subsections respectively, and then provide the loss function for training the model.

3.1 Data Embedding

Faster-RCNN [28] is used to encode images as denoted in Eq. (1), with the static features provided by bottom-up-attention [1]. GRU [7] is used to encode text as denoted in Eq. (2), with the parameters initialized with skip-thoughts [16].

$$V^f = RCNN\,(image)\,, \tag{1}$$

$$Q^f = GRU\,(question)\,, \tag{2}$$

where $V^f \in \mathbb{R}^{m\times d_v}$ denotes the visual embeddings of the top-ranked m detection boxes, and $Q^f \in \mathbb{R}^{d_q}$ denotes the question embedding.

To map V^f and Q^f to the same dimension, we employ a fully-connection layer and a one-dimensional convolution layer respectively as follows in Eq. (3)~(4).

$$V = relu\left(Conv1d\left(V^f\right)\right), \tag{3}$$

$$Q = relu\left(Linear\left(Q^f\right)\right), \tag{4}$$

where $V \in \mathbb{R}^{m\times d}$ is viewed as a set of m objects, i.e. $V = \{V_1, V_2, \ldots, V_m\}$. $Q \in \mathbb{R}^d$ is the question feature. To write simple, we omit the bias b.

3.2 Object-Difference Attention (ODA)

We define ODA as in Eq. (5):

$$\widetilde{V} = softmax\left(\left[(V_i - V_j)\odot Q\right]_{m\times md} W_f\right)^{\mathsf{T}} V, \tag{5}$$

where V_i, V_j and Q are defined in above subsection, $(V_i - V_j)\odot Q \in \mathbb{R}^d$ is the comparison results of the ith image object and the jth image object under the guidance of the question Q. As a result, $\left[(V_i - V_j)\odot Q\right]_{m\times md}$ is a matrix of size $m \times md$ with the ith row representing the comparison results between the ith object and all other objects. $W_f \in \mathbb{R}^{md\times n}$ is a learnable parameter matrix that performs n glimpses to transform the comparion results to n attention distributions. $\widetilde{V} \in \mathbb{R}^{nd}$ is the attention results.

We have three notes on ODA as follows.

Firstly, different from previous attentions, ODA includes an explicit difference operator guided by a question, which is used to explicitly compare different objects in an image. For any object V_i, the comparisons between V_i and other object guided by a question should be crucial on measuring how important V_i is for answering the question. It is the major insight of ODA. It is cognitively reasonable. According to cognitive neuroscience, one of the major property of attention mechanism is selection [9]. The nature of selection is comparison. Only by comparison, we can choose something important. An object is worthy to be assigned more attentions only when it shows much importance compared with other objects. Furthermore, comparisons between different objects help us to understand the internal structure of these objects better, which is often useful for solving problems in hand. For example, existing work in image retrieval improves retrieval performance [19] by comparing pairs of objects to capture the internal structural relations between objects.

Secondly, although ODA includes difference operator, it is not computational complexity, and even with lower computational complexity compared with some previous attentions. We use Mutan [5] as an example for comparison. The attention in Mutan in shown in Eq. (6):

$$\widetilde{V}_{Mutan} = softmax\left(\left[\sum_{s=1}^{S}\left(V_i W_1^{(s)}\odot Q W_2^{(s)}\right)\right]_{m\times d'} W_f'\right)^{\mathsf{T}} V, \tag{6}$$

where $W_1^{(s)} \in \mathbb{R}^{d\times d'}$, $W_2^{(s)} \in \mathbb{R}^{d\times d'}$, $W_f' \in \mathbb{R}^{d'\times n}$ are learnable parameters. S is a hyper-parameter. For Mutan, the amount of parameters is $\Theta_{Mutan} \approx 2Sdd'$, the time complexity is $O_{Mutan} = O(2Smdd')$. As a contrast, for ODA, the amount of model parameters is $\Theta_{ODA} \approx md$, the time complexity is $O_{ODA} = O(2m^2 d)$. As well known, normally, $m \in [36, 100]$, $d, d' > 300$, $S \approx 5$. Compared with Mutan, ODA has smaller parameter size and lower time complexity.

Thirdly, ODA can be extended easily in twofold. First, ODA is a plug-and-play module which can be easily deployed in other models that requires attention mechanism. Second, which is more exciting, ODA is a particular case of a new type of attention which is called relational attention. We will define and discuss the general form of relational attention in Section 4.

3.3 Decision Making

In order to obtain more attention information, we further calculate $\widetilde{V}$ (defined in Eq. (5)) p times with differnt learnable parameters and concatenate them in Eq (7):

$$\widetilde{Z} = \left[\widetilde{V}^{(1)}; \widetilde{V}^{(2)}; \ldots; \widetilde{V}^{(p)}\right]. \tag{7}$$

where $[;]$ represents the concatenation operation, $\widetilde{Z} \in \mathbb{R}^{ndp}$ is the concatenated image feature. If $p = 1$, we simply call the model ODA, if $p > 1$, we call the model ODA×p. For the sake of brevity, Figure 2 only shows the case of $p = 1$.

We use Mutan to fuse the image feature and the question feature, as denoted in Eq. (8):

$$H = \sum_{s=1}^{S}\left(\widetilde{Z} W_v^{(s)}\odot Q W_q^{(s)}\right), \tag{8}$$

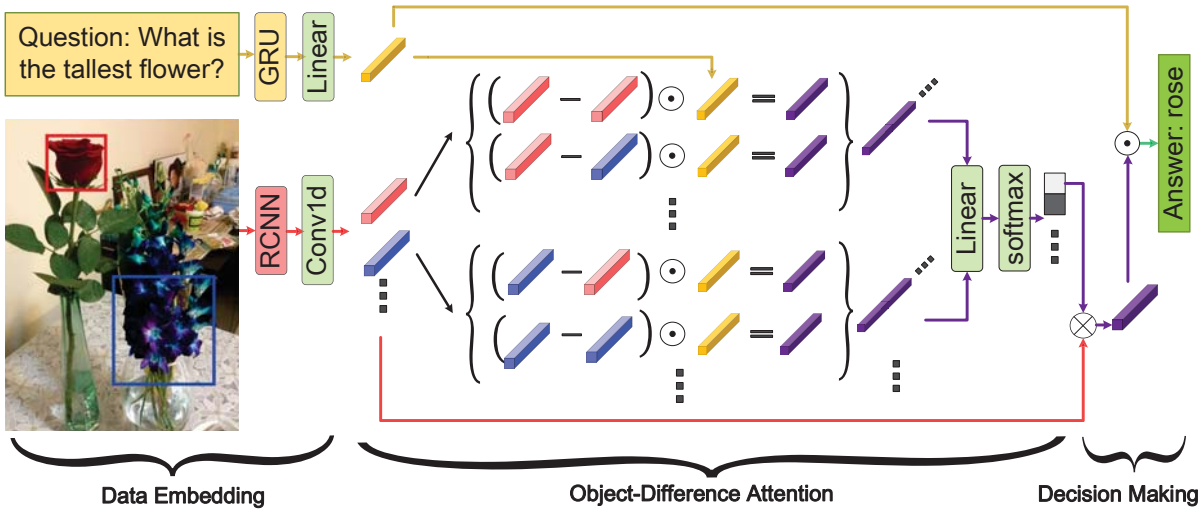

Figure 2: The overall structure of our model for VQA.

where $S \in \mathbb{R}^+$ is a hyper-parameter. $W_v^{(s)} \in \mathbb{R}^{ndp \times d_h}$, $W_q^{(s)} \in \mathbb{R}^{d \times d_h}$ are learnable parameters, $H \in \mathbb{R}^{d_h}$ is the fusion feature.

Finally, a linear layer with a sigmoid activation function is used to predict the score of the candidate answer in Eq. (9)

$$\hat{a} = \sigma(W_h H), \tag{9}$$

where $\hat{a} \in \mathbb{R}^{|\mathcal{D}|}$ is the predicted answer, $\mathcal{D}$ is the answer dictionary, $|\mathcal{D}|$ is the number of candidate answers.

3.4 Loss Function For Model Training

We first calculate the ground-truth answer distribution by making use of Eq. (10):

$$a_i = \frac{\sum_{j=1}^{N} \mathbb{1}\{u_j = i\}}{N - \sum_{j=1}^{N} \mathbb{1}\{u_j \notin \mathcal{D}\}}, \tag{10}$$

where $a \in \mathbb{R}^{|\mathcal{D}|}$ is the ground-truth answer distribution, u_i is the answer given by the ith annotator. N is the number of annotators. For example, N is 10 in the VQA 1.0 and VQA 2.0 dataset and N is 1 in the COCO-QA dataset.

The KL-divergence is then used as the loss function between a and $\hat{a}$ in Eq. (11):

$$\mathcal{L}(\hat{a}, a) = \sum_{i=1}^{|\mathcal{D}|} a_i \log\left(\frac{a_i}{\hat{a}_i}\right). \tag{11}$$

The model is trained by minimizing the loss function.

4 RELATIONAL ATTENTION

As we mentioned at the end of Section 3.2, ODA is a special case of a relational attention. We introduce a general form of relational attention (RA) and give some other cases of RA in this section.

First of all, any attention can be written as in Eq. (12):

$$\widetilde{V} = f_\theta(K, V). \tag{12}$$

It consists of three parts:

The first part is $V \in \mathbb{R}^{m \times d_v}$. It is called attention target where an attention-based model pays its attention to. m is the number of objects in the target and d_v is the dimension for each object. The purpose of attention is to get the importance distribution of these m objects. For example, V can be an image, a piece of text, or several candidate answers.

The second part is $K \in \mathbb{R}^{m \times m \times d_k}$. It is called attention kernel. The attention kernel determines how the attention distribution is calculated. If an attention kernel is obtained by a binary relational operation R between two objects V_i and V_j in the set of attention target under an external guidance Q, as shown is Eq. (13), the attention kernel is called relational attention kernel (relational kernel in brief).

$$K = R(V_i, V_j) \odot Q. \tag{13}$$

An attention with relational kernel is called a relational attention. ODA proposed in Section 3 is a relational attention, where a difference operator is used between different image objects $(V_i - V_j)$ under the guidance of question Q. We call the relational kernel in ODA object-difference kernel (ODK). Under the general definition of relational kernel, it is easy to introduce some other relational kernels. We give some of them in Table 1. For example, OUK employs union operator between different objects.

On the contrast, most of previous attention kernels are non-relational since they include no relational operation between different objects. We also list some of them in Table 1.

It is worth mentioning that this paper only gives some of the most fundamental relational kernels without external parameters. These kernels show different strengths in VQA task, as described in Section 5.4.

The third part is f_θ. It is called attention operator. It determines how an attention kernel acts on an attention target. The task itself often hints what f_θ we should use. For example, in the classification task, we need the comprehensive attention result $\widetilde{V} \in \mathbb{R}^{d_v}$. As a contrast, in the machine translation task, we need the new encoding $\widetilde{V} \in \mathbb{R}^{m \times d_v}$. For the former, we may use

Table 1: Some Attention Kernels

	Kernels	Names (brief)
Relational Attention Kernels	$(V_i - V_j) \odot Q$	Object-Difference Kernel (ODK)
	$(V_i + V_j) \odot Q$	Object-Union Kernel (OUK)
	$(V_i \odot V_j) \odot Q$	Object-Intersection Kernel (OIK)
	$(V_i \odot V_i) \odot Q$	Object-Self Kernel (OSK)
Non-Relational Attention Kernels	$V_i \odot Q$	Object Kernel (OK)
	$V_i W_1 \odot Q W_2$	MLB [14]
	$\sum_{s=1}^{S} \left(V_i W_1^{(s)} \odot Q W_2^{(s)} \right)$	Mutan [5]
	$SumPooling(V_i W_1 \odot Q W_2, S)$	MFB [36]

$f_\theta(K,V) = softmax(KW_f)^\mathsf{T} V$ with $W_f \in \mathbb{R}^{md_k}$; for the latter, we may use $f_\theta(K,V) = softmax(KW_f)^\mathsf{T} V$ with $W_f \in \mathbb{R}^{d_k}$.

In ODA, we use the former operator because we view the VQA task as a classification task.

5 EXPERIMENTS

5.1 Datasets and evaluation metrics

We evaluated our model on three public datasets: the VQA 1.0 dataset [4], the VQA 2.0 dataset [10] and the COCO-QA dataset [27].

VQA 1.0 dataset. The VQA 1.0 dataset [4] contains a total of 614,163 samples, consisting of 204,721 images, 328,120 questions, and 22,523 answers. Each sample contains an image, a question, and ten human annotated answers. Images are from Microsoft COCO image data [18]. The questions are divided into three categories: *"yes/no"*, *"number"* and *"other."* The dataset is divided into three splits: train(40.4%), val(19.8%), test(39.8%). Further, the test set includes two types: test-dev and test-std. The dataset has two subtasks: Open-Ended (OE) and Multiple-Choice (MC).

VQA 2.0 dataset. The VQA 2.0 dataset [10] contains a total of 1,105,904 samples, consisting of 204,721 images, 332,793 questions, and 29,332 answers. Specifically, for each question, there are a pair of similar images that result in two different answers to the question. As a result, VQA 2.0 dataset is more balanced compared to VQA 1.0 dataset. The VQA 2.0 dataset is also divided into three splits: train(40.1%), val(19.4%), test(40.5%). Besides, the VQA 2.0 dataset has no subtasks.

COCO-QA dataset. The COCO-QA dataset [27] contains a total of 117,684 samples, consisting of 69,172 images, 92,396 questions, and 430 answers. The dataset is only divided into two splits: train(66.9%), test(33.1%).

Evaluation metrics. For the VQA 1.0 and VQA 2.0 dataset, we use the evaluation tool proposed in [4] to evaluate the model, as denoted in Eq. (14):

$$Acc(ans) = min\left\{ \frac{\#\text{humans that said } ans}{3}, 1 \right\}. \tag{14}$$

For the COCO-QA dataset, we evaluate the model in Eq. (15):

$$Acc(ans) = \mathbb{1}\{ans = \text{ground\_truth}\}. \tag{15}$$

5.2 Implementation details

During the data-embedding phase, the image features extracted from RCNN are the size of 36×2048 and mapped to 36×310. The text features extracted from GRU are the size of 2400 and mapped to 310. In the object-difference attention phase, the attention hidden dimension is 620. Attention glimpse is 2. In the decision making phase, the Mutan dimension is 510; hyperparameter S is 5. Most of the parameters above are general settings. All the nonlinear layers of the model use the relu activation function and use dropout [32] to prevent overfitting.

We implement the model using Pytorch. We use Adam [15] to train the model. The learning rate is set to 10^{-4}, beta is (0.9,0.999) and eps is 10^{-8}. We train the model with a batch_size of 128 and 60 epochs. More details, including source codes, will be published in the near future.

5.3 Comparison with state-of-the-art

In this section, we compare ODA with the state-of-the-art models on the VQA 1.0 dataset, the VQA 2.0 dataset, and the COCO-QA dataset. In the VQA 1.0 dataset and the VQA 2.0 dataset, ODA is trained on the train+val set and test on the test-dev and test-std set respectively. In the COCO-QA dataset, ODA is trained on the train set and test on the test set.

Firstly, Table 2 shows the comparison with the state-of-the-art models on the VQA 1.0 dataset. As we can see, ODA achieves new state-of-the-art performance in all subtasks. In the Multiple-Choice task, ODA improves the overall accuracy of the state-of-the-art MFB method from 71.4% to 72.23%. In the Open-Ended task, ODA improves the overall accuracy of the state-of-the-art ReasonNet method from 67.9% to 67.97%. It is worth mentioning that ODA only uses one image feature while ReasonNet uses six input image features including face analysis, object classification, scene classification and so on.

Secondly, Table 3 shows the comparison with the state-of-the-arts on the VQA 2.0 dataset. With the same image feature (100 boxes extracted from bottom-up-attention [1]), ODA×2 improves the overall accuracy of the state-of-the-art LC_Counting model [37] from 68.09% to 68.17% in the test-dev set.

Thirdly, Table 4 shows the comparison with the state-of-the-arts on the COCO-QA dataset. ODA improves the overall accuracy of the state-of-the-art Dual-MFA model from 66.49% to 69.33%.

In summary, our ODA model, which employs a simple difference operator as relational kernel, achieves the state-of-the-art performance in all three datasets. It shows that comparisons between different objects actually play a crucial role in question answering.

5.4 Ablation study

In this section, we conduct some ablation experiments on the VQA 1.0 dataset. For a fair comparison, all the data provided in this section are trained under the training set and test on the validation set. In addition, all the models use the exactly same bottom-up-attention feature (36 boxes) extracted from faster-rcnn.

Table 5 shows the summary performance of different loss functions, different attention kernels and different state-of-the-art models on the VQA 1.0 dataset.

Firstly, we study the effectiveness of different loss functions in Figure 3. "CE" denotes the softmax cross entropy loss, "BCE" denotes the sigmoid binary cross entropy loss, and "KL" (used by ODA) denotes the Kullback-Leibler loss. We can see that KL has the

Table 2: Comparision with the state-of-the-arts on the VQA 1.0 dataset.

		VQA 1.0 Test-dev					VQA 1.0 Test-std				
		Open-Ended				MC	Open-Ended				MC
	Method	All	Y/N	Num.	Other	All	All	Y/N	Num.	Other	All
Single image feature	SAN [35]	58.70	79.30	36.60	46.10	-	58.85	79.11	36.41	46.42	-
	HieCoAtt [20]	61.80	79.70	38.70	51.70	65.80	62.06	79.95	38.22	51.95	66.07
	DAN [25]	64.3	83.0	39.1	53.9	69.1	64.2	82.8	38.1	54.0	69.0
	HighOrderAtt [30]	-	-	-	-	69.3	-	-	-	-	69.4
	MF-SIG-T3 [38]	66.00	84.33	39.34	56.37	-	65.88	84.42	38.94	55.89	70.33
	MCB [8]	64.70	82.50	37.60	55.60	69.10	-	-	-	-	-
	MLB [14]	64.89	84.13	37.85	54.57	-	65.07	84.02	37.90	54.77	68.89
	MFB [36]	66.9	84.1	39.1	58.4	71.3	66.6	84.2	38.1	57.8	71.4
	NMN [3]	58.6	81.2	38.0	44.0	-	58.7	-	-	-	-
	DNMN [2]	59.4	81.1	38.6	45.5	-	59.4	-	-	-	-
	N2NMN [11]	64.9	-	-	-	-	-	-	-	-	-
Multi image feauture	Dual-MFA [21]	66.01	83.59	40.18	56.84	70.04	66.09	83.37	40.39	56.89	69.97
	ReasonNet [12]	-	-	-	-	-	67.9	84.0	38.7	**60.4**	-
Single image feauture	ODA (36boxes) (ours)	**67.83**	**85.82**	**43.03**	**58.07**	**72.28**	**67.97**	**85.81**	**42.51**	58.24	**72.23**

Table 3: Comparision with the state-of-the-arts on the VQA 2.0 dataset.

	VQA 2.0 Test-dev			
Method	All	Y/N	Num.	Other
MF-SIG-VG (resnet) [38]	64.73	81.29	42.99	55.55
Up-Down (36boxes) [33]	65.32	81.82	44.21	56.05
LC_Baseline (100boxes) [37]	67.50	82.98	46.88	**58.99**
LC_Counting (100boxes) [37]	68.09	83.14	**51.62**	58.97
ODA (36boxes) (ours)	67.34	84.23	46.18	57.73
ODA×2 (36boxes) (ours)	67.52	84.3	46.62	57.96
ODA×2 (100boxes) (ours)	**68.17**	**84.66**	48.04	58.68

Table 4: Comparision with the state-of-the-arts on the COCO-QA dataset.

Method	All	Obj.	Num.	Color	Loc.	WUPS0.9	WUPS0.0
2VIS+BLSTM [27]	55.09	58.17	44.79	49.53	47.34	65.34	88.64
IMG-CNN [22]	58.40	-	-	-	-	68.50	89.67
DDPnet [26]	61.16	-	-	-	-	70.84	90.61
SAN [35]	61.60	65.40	48.60	57.90	54.00	71.60	90.90
QRU [17]	62.50	65.06	46.90	60.50	56.99	72.58	91.62
HieCoAtt [20]	65.40	68.00	51.00	62.90	58.80	75.10	92.00
Dual-MFA [21]	66.49	68.86	51.32	65.89	58.92	76.15	92.29
ODA (36boxes) (ours)	**69.33**	**70.48**	**54.70**	**74.17**	**60.90**	**78.29**	**93.02**

fastest convergence rate and the best convergence result (64.50). Compared with KL, BCE has a lower convergence rate but similar convergence result (64.48). In contrast, compared with KL, CE has a similar convergence rate but lower convergence result (64.22).

Secondly, we study the effectiveness of different attention kernels under different types of questions in Table 6.

It is interesting to find out that each kernel has its unique ability to answer a particular type of questions.

(1) ODK achieves the best performance on the total task, and achieves the best performance on more than half types of question, such as *"cause"*, *"color"*, *"object"*, *"other"*, *"type"* and *"yesno"* among all different kernels. It is not strange. As we have argued in section 3.2, comparisons implemented by difference operator are a major property of attention.

(2) Although ODK achieves the best performance on the total task, different kernels show different strengths on different types of question. For question types of *"time"*, OUK outperforms ODK. It seems that uniting an object with other objects may help to collect more environment information to guess the correct time. For question types of *"position"*, OIK outperforms ODA. It uses

Table 5: Ablation study on the VQA 1.0 dataset. For a fair comparison, MFB is implemented without question attention.

Method	Validation
ODA with CE loss	64.22
ODA with BCE loss	64.48
$(Vi + Vj) \odot Q$ (OUK)	64.17
$(Vi \odot Vj) \odot Q$ (OIK)	64.17
$(Vi \odot Vi) \odot Q$ (OSK)	63.87
$Vi \odot Q$ (OK)	63.82
RN [29]	58.85
MLB [14]	63.75
MFB [36]	63.87
Mutan [5]	64.06
ODA	**64.50**

Table 6: Performance of different attention kernels under different types of questions on the VQA 1.0 dataset.

	ODK	OUK	OIK	OSK	OK
action	92.42	92.44	93.07	92.71	**93.40**
cause	**25.47**	25.01	24.83	24.85	24.23
color	**78.44**	77.62	76.96	76.52	77.22
count	44.94	45.21	44.99	**45.35**	45.00
object	**48.77**	47.93	48.36	47.77	47.69
other	**58.70**	58.69	58.62	58.31	58.63
position	37.74	37.14	**38.05**	37.92	37.58
time	21.01	**21.41**	20.62	21.08	20.53
type	**54.70**	54.47	54.50	54.21	53.56
yesno	**83.41**	83.33	83.26	82.97	82.94
all	**64.50**	64.17	64.17	63.87	63.82

element-wise multiplication to obtain intersected location feature, which may help to locate the relative position. For question types of *"count"*, OSK performs best. It seems that OSK strengthens the own information of the counted object instead of interacting with others, which may help to count. For question types of *"action"*, OK performs best, maybe just watching the features of the action-related objects instead of interacting objects are helpful for this type of question.

Thirdly, we study the effectiveness of our model. In the third part of Table 5, we implement RN, MLB, MFB and Mutan in our framework (exactly same input image feature and question feature). RN models relations without attention which leads to lower results. MLB, MFH, Mutan models attention without relation. All these models are inferior to ODA.

5.5 Qualitative comparison between different attention kernels

Figure 4 shows five examples, each containing the visualization of five different kernels: ODK, OUK, OIK, OSK, OK. In each example, only one of the kernel gives the correct answer which shows different strengths of each kernel. In each picture, the object in the red box is assigned the maximum value of attention weight. The objects in the orange, yellow, and blue boxes are assigned descending attention weights.

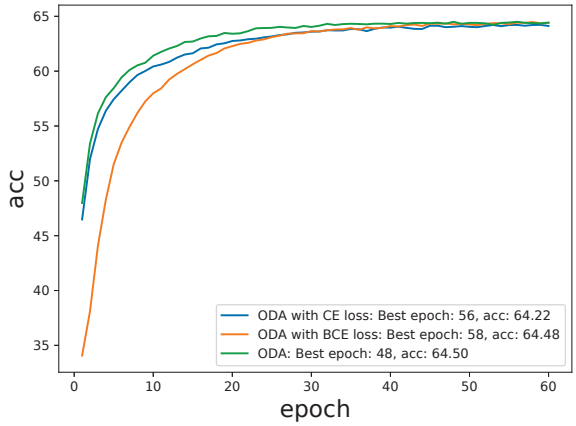

Figure 3: Performance of different loss functions on the VQA 1.0 dataset.

For example, for the question *"What type of markings does the cat have?"* in sample 1, ODK locates a red box with the attention weight of 0.42 by explicitly comparing with all other boxes in the image (See image in the first row and first column in Figure 4). The red box includes both the white and black parts of the cat. Therefore, the correct answer *"black and white"* is obtained. In contrast, other kernels either have a wrong bounding box or focus only the black parts of the cat.

6 CONCLUSIONS

In this paper, we propose a simple but effective object-difference attention (ODA) for VQA task. It compares objects explicitly by difference operator for calculating the attention distribution. It is cognitively reasonable to reflect one of major property of attention. Experimental results on three publicly available datasets show our ODA based VQA model achieves the state-of-the-art results. Furthermore, we introduce a general form of attentions with different attention kernels. Experimental results show those attentions have strengths on different types of questions. Compared to previous attention, such as multi-step attention and bilinear models, our proposed relational attentions are simple and with lower computational complexity. It gives more space for constructing new relational kernels to reveal more complex relations existing in problems, or combining several different relational kernels in a same model to make use of different strengths that have been shown in the paper.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments. This paper is supported by NSFC (No. 61273365), NSSFC (2016ZDA055), 111 Project (No. B08004), Beijing Advanced Innovation Center for Imaging Technology, Engineering Research Center of Information Networks of MOE, China.

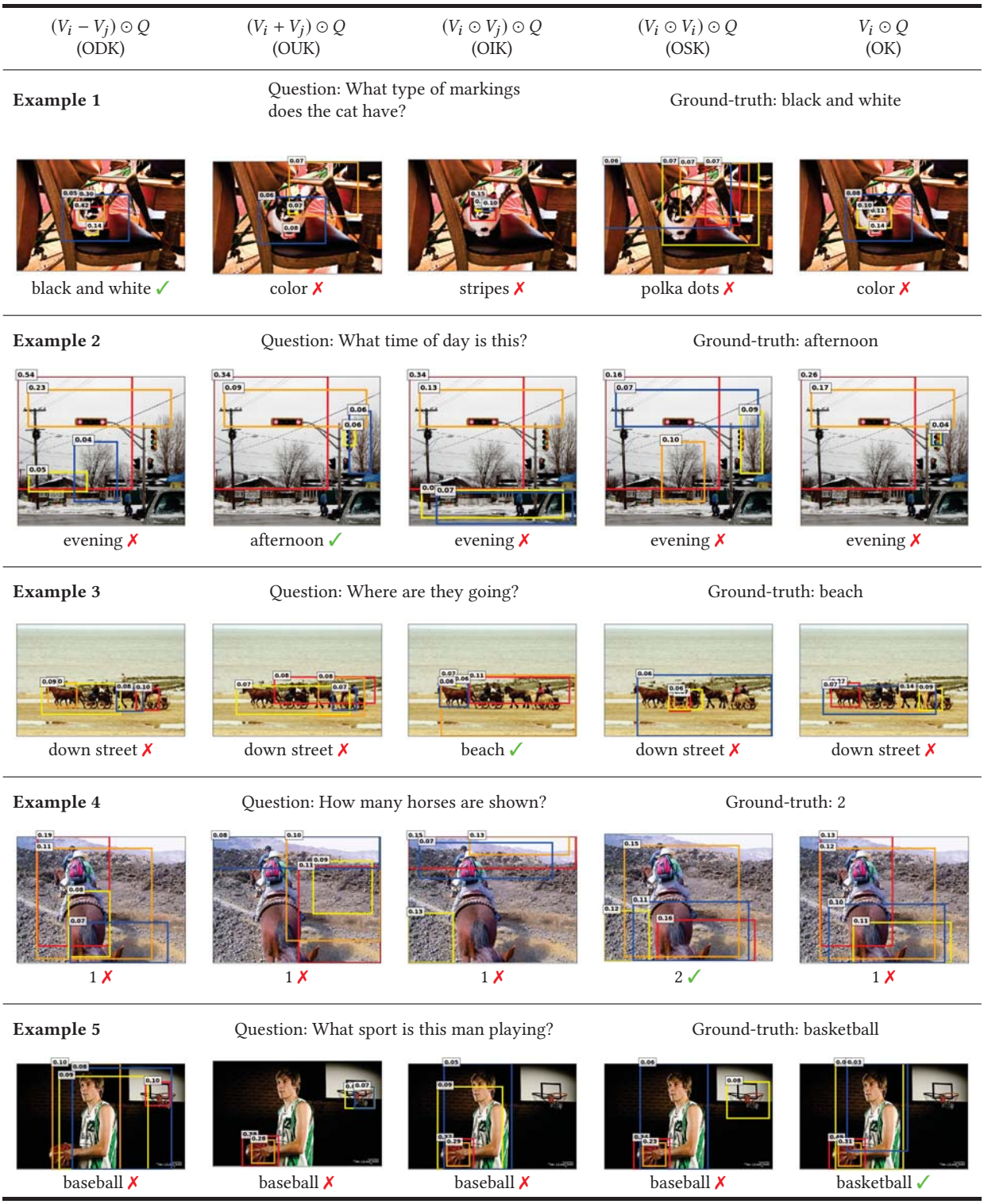

Figure 4: Visualization of different attention kernels.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *arXiv preprint arXiv:1707.07998* (2017). arXiv:1707.07998

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to Compose Neural Networks for Question Answering. In *NAACL*. 1545–1554.

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural Module Networks. In *CVPR*. 39–48.

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual Question Answering. In *ICCV*. 2425–2433.

[5] Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *ICCV*. 2631–2639.

[6] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. 2015. ABC-CNN: An Attention Based Convolutional Neural Network for Visual Question Answering. *arXiv preprint arXiv:1511.05960* (2015).

[7] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv preprint arXiv:1409.1259* (2014). arXiv:1409.1259

[8] 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 *EMNLP*. 457–468.

[9] Michael S. Gazzaniga, Richard B. Ivry, and George R. Mangun. 2013. *Cognitive Neuroscience: The Biology of the Mind, 4th Edition* (4th edition ed.). W. W. Norton & Company, New York, N.Y.

[10] 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 *CVPR*, Vol. 1. 9.

[11] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to Reason: End-to-End Module Networks for Visual Question Answering. *arXiv preprint arXiv:1704.05526* (2017).

[12] Ilija Ilievski and Jiashi Feng. 2017. Multimodal Learning and Reasoning for Visual Question Answering. In *NIPS*. 551–562.

[13] Kushal Kafle and Christopher Kanan. 2016. Answer-Type Prediction for Visual Question Answering. In *CVPR*. 4976–4984.

[14] Jin-Hwa Kim, Kyoung-Woon On, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Hadamard Product for Low-Rank Bilinear Pooling. In *ICLR*.

[15] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *NIPS*. 3294–3302.

[17] Ruiyu Li and Jiaya Jia. 2016. Visual Question Answering with Question Representation Update (Qru). In *NIPS*. 4655–4663.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft Coco: Common Objects in Context. In *ECCV*. 740–755.

[19] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. Deepfashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In *CVPR*. 1096–1104.

[20] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical Question-Image Co-Attention for Visual Question Answering. In *NIPS*.

[21] Pan Lu, Hongsheng Li, Wei Zhang, Jianyong Wang, and Xiaogang Wang. 2018. Co-Attending Free-Form Regions and Detections with Multi-Modal Multiplicative Feature Embedding for Visual Question Answering. In *AAAI*. arXiv:1711.06794

[22] Lin Ma, Zhengdong Lu, and Hang Li. 2016. Learning to Answer Questions from Image Using Convolutional Neural Network. *AAAI* (2016).

[23] Mateusz Malinowski and Mario Fritz. 2014. A Multi-World Approach to Question Answering about Real-World Scenes Based on Uncertain Input. In *NIPS*. 1682–1690.

[24] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. 2015. Ask Your Neurons: A Neural-Based Approach to Answering Questions about Images. In *ICCV*. 1–9.

[25] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual Attention Networks for Multimodal Reasoning and Matching. *CVPR* (2017).

[26] Hyeonwoo Noh, Paul Hongsuck Seo, and Bohyung Han. 2016. Image Question Answering Using Convolutional Neural Network with Dynamic Parameter Prediction. *CVPR* (2016).

[27] Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Exploring Models and Data for Image Question Answering. In *NIPS*. 2953–2961.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*. 91–99.

[29] Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A Simple Neural Network Module for Relational Reasoning. *NIPS* (2017).

[30] Idan Schwartz, Alexander G. Schwing, and Tamir Hazan. 2017. High-Order Attention Models for Visual Question Answering. In *NIPS*. arXiv:1711.04323

[31] Kevin J. Shih, Saurabh Singh, and Derek Hoiem. 2016. Where to Look: Focus Regions for Visual Question Answering. *CVPR* (2016).

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[33] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. 2017. Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge. *arXiv:1708.02711 [cs]* (Aug. 2017). arXiv:cs/1708.02711

[34] Huijuan Xu and Kate Saenko. 2016. Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering. In *ECCV*. 451–466.

[35] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked Attention Networks for Image Question Answering. In *CVPR*.

[36] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-Modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering. *ICCV 2017* (Aug. 2017). arXiv:1708.01471

[37] Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. 2018. Learning to Count Objects in Natural Images for Visual Question Answering. In *arXiv:1802.05766 [Cs]*. arXiv:cs/1802.05766

[38] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. 2017. Structured Attentions for Visual Question Answering. *ICCV 2017* (Aug. 2017). arXiv:1708.02071