Improved Fusion of Visual and Language Representations
by Dense Symmetric Co-Attention for Visual Question Answering

Duy-Kien Nguyen[1] and Takayuki Okatani[1,2]
[1]Tohoku University   [2]RIKEN Center for AIP
{kien, okatani}@vision.is.tohoku.ac.jp

Abstract

A key solution to visual question answering (VQA) exists in how to fuse visual and language features extracted from an input image and question. We show that an attention mechanism that enables dense, bi-directional interactions between the two modalities contributes to boost accuracy of prediction of answers. Specifically, we present a simple architecture that is fully symmetric between visual and language representations, in which each question word attends on image regions and each image region attends on question words. It can be stacked to form a hierarchy for multi-step interactions between an image-question pair. We show through experiments that the proposed architecture achieves a new state-of-the-art on VQA and VQA 2.0 despite its small size. We also present qualitative evaluation, demonstrating how the proposed attention mechanism can generate reasonable attention maps on images and questions, which leads to the correct answer prediction.

1. Introduction

There has been a significant progress in the study of visual question answering (VQA) over a short period of time since its introduction, showing rapid boost of performance for common benchmark datasets. This progress has been mainly brought about by two lines of research, the development of better attention mechanisms and the improvement in fusion of features extracted from an input image and question.

Since introduced by Bahdanau et al. [3], attention has been playing an important role in solutions of various problems of artificial intelligence ranging from tasks using single modality (e.g., language, speech, and vision) to multimodal tasks. For VQA, attention on image regions generated from the input question was first introduced [28] and then several extensions have been proposed [17, 31, 4]. Meanwhile, researchers have proposed several methods for

feature fusion [5, 14, 32], where the aim is to obtain better fused representation of image and question pairs. These studies updated the state-of-the-art for common benchmark datasets at the time of each publication.

We observe that these two lines of research have been independently conducted so far. This is particularly the case with the studies of feature fusion methods, where attention is considered to be optional, even though the best performance is achieved with it. However, we think that they are rather *two different approaches towards the same goal*. In particular, we argue that a better attention mechanism leads to a better fused representation of image-question pairs.

Motivated by this, we propose a novel co-attention mechanism for improved fusion of visual and language representations. Given representations of an image and a question, it first generates an attention map on image regions for each question word and an attention map on question words for each image region. It then performs computation of attended features, concatenation of multimodal representations, and their transformation by a single layer network with ReLU and a residual connection. These computations are encapsulated into a composite network that we call *dense co-attention* layer, since it considers every interaction between any image region and any question word. The layer has fully symmetric architecture between the two modalities, and can be stacked to form a hierarchy that enables multi-step interactions between the image-question pair.

Starting from initial representations of an input image and question, each dense co-attention layer in the layer stack updates the representations, which are inputted to the next layer. Its final output are then fed to a layer for answer prediction. We use additional attention mechanisms in the initial feature extraction as well as the answer prediction layer. We call the entire network including all these components the *dense coattention network* (DCN). We show the effectiveness of DCNs by several experimental results; they achieve the new state-of-the-art for VQA 1.0 and 2.0 datasets.

2. Related Work

In this section, we briefly review previous studies of VQA with a special focus on the developments of attention mechanisms and fusion methods.

2.1. Attention Mechanisms

Attention has proved its effectiveness on many tasks and VQA is no exception. A number of methods have been developed so far, in which question-guided attention on image regions is commonly used. They are categorized into two classes according to the type of employed image features. One is the class of methods that use visual features from some region proposals, which are generated by Edge Boxes [22, 11] or Region Proposal Network [24]. The other is the class of methods that use convolutional features (i.e., activations of convolutional layers) [4, 5, 12, 13, 14, 17, 19, 20, 27, 28, 32].

There are several approaches to creation and use of attention maps. Yang *et al.* [28] developed stacked attention network that produces multiple attention maps on the image in a sequential manner, aiming at performing multiple steps of reasoning. Kim *et al.* [13] extended this idea by incorporating it into a residual architecture to produce better attention information. Chen *et al.* [4] proposed a structured attention model that can encode cross-region relation, aiming at properly answering questions that involve complex inter-region relations.

Earlier studies mainly considered question-guided attention on image regions. In later studies, the opposite orientation of attention, i.e., image-guided attention on question words, is considered additionally. Lu *et al.* [17] introduced the co-attention mechanism that generates and uses attention on image regions and on question words. To reduce the gap of image and question features, Yu *et al.* [31] utilized attention to extract not only spatial information but also language concept of the image. Yu *et al.* [32] combined the mechanism with a novel multi-modal feature fusion of image and question.

We point out that the existing attention mechanisms only consider a limited amount of possible interactions between image regions and question words. Some consider only attention on image regions from a *whole* question. Co-attention additionally considers attention on question words but it is created from a *whole* image. We argue that this can be a significant limitation of the existing approaches. The proposed mechanism can deal with every interaction between any image region and any question word, which possibly enables to model unknown complex image-question relations that are necessary for correctly answering questions.
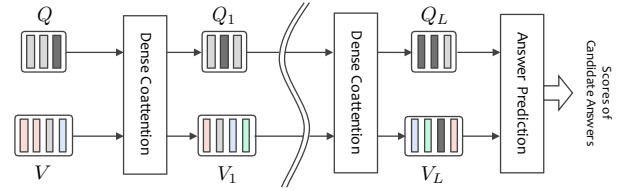

Figure 1: The global structure of the dense co-attention network (DCN).

2.2. Multimodal Feature Fusion

The common framework of existing methods is that visual and language features are independently extracted from the image and question at the initial step, and they are fused at a later step to compute the final prediction. In early studies, researchers employed simple fusion methods such as the concatenation, summation, and element-wise product of the visual and language features, which are fed to fully connected layers to predict answers.

It was then shown by Fukui *et al.* [5] that a more complicated fusion method does improve prediction accuracy; they introduced the bilinear (pooling) method that uses an outer product of two vectors of visual and language features for their fusion. As the outer product gives a very high-dimensional feature, they adopt the idea of Gao *et al.* [6] to compress the fused feature and name it the Multimodal Compact Bilinear (MCB) pooling method. However, the compacted feature of the MCB method still tends to be high-dimensional to guarantee robust performance, Kim *et al.* [14] proposed low-rank bilinear pooling using Hadamard product of two feature vectors, which is called the Multimodal Low-rank Bilinear (MLB) pooling. Pointing out that MLB suffers from slow convergence rate, Yu *et al.* [32] proposed the Multi-modal Factorized Bilinear (MFB) pooling, which computes a fused feature with a matrix factorization trick to reduce the number of parameters and improve convergence rate.

The attention mechanisms can also be considered feature fusion methods, regardless of whether it is explicitly mentioned, since they are designed to obtain a better representation of image-question pairs based on their interactions. This is particularly the case with co-attention mechanisms in which the two features are treated symmetrically. Our dense co-attention network is based on this observation. It fuses the two features by multiple applications of the attention mechanism that can use more fine-grained interactions between them.

3. Dense Co-Attention Network (DCN)

In this section, we describe the architecture of DCNs; see Fig.1 for its overview. It consists of a stack of *dense co-attention layers* that fuses language and visual features

repeatedly, on top of which an *answer prediction layer* that predict answers in a multi-label classification setting [24]. We first explain the initial feature extraction from the input question and image (Sec.3.1) and then describe the dense co-attention layer (Sec.3.2) and the answer prediction layer (Sec.3.3).

3.1. Feature Extraction

We employ pretrained networks that are commonly used in previous studies [13, 29, 14, 4] for encoding or extracting features from images, questions, and answers, such as pretrained ResNet [9] with some differences from earlier studies.

3.1.1 Question and Answer Representation

We use bi-directional LSTM for encoding questions and answers. Specifically, a question consisting of N words is first converted into a sequence $\{e_1^Q, ..., e_N^Q\}$ of GloVe vectors [21], which are then inputted into a one-layer bi-directional LSTM (Bi-LSTM) with a residual connection as

$$\overrightarrow{q_n} = \text{Bi-LSTM}(\overrightarrow{q_{n-1}}, e_n^Q), \tag{1}$$

$$\overleftarrow{q_n} = \text{Bi-LSTM}(\overleftarrow{q_{n+1}}, e_n^Q). \tag{2}$$

We then create a matrix $Q = [q_1, ..., q_N] \in \mathbb{R}^{d \times N}$ where $q_n = [\overrightarrow{q_n}^\top, \overleftarrow{q_n}^\top]^\top$ $(n = 1, ..., N)$. We will also use $s_Q = [\overrightarrow{q_N}^\top, \overleftarrow{q_1}^\top]^\top$, concatenation of the last hidden states in the two paths, for obtaining representation of an input image (Sec.3.1.2). We randomly initialized the Bi-LSTM. It is worth noting that we initially used a pretrained two-layer Bi-LSTM that yields Context Vectors (CoVe) in [18], which we found does not contribute to performance.

We follow a similar procedure to encode answers. An answer of M words is converted into $\{e_1^A, ..., e_M^A\}$ and then inputted to the same Bi-LSTM, yielding the hidden states $\overrightarrow{a_m}$ and $\overleftarrow{a_m}$ $(m = 1, ..., M)$. We will use $s_A = [\overrightarrow{a_M}^\top, \overleftarrow{a_1}^\top]^\top$ for answer representation.

3.1.2 Image Representation

As in many previous studies, we use a pretrained CNN (i.e., a ResNet [9] with 152 layers pretrained on ImageNet) to extract visual features of multiple image regions, but our extraction method is slightly different. We extract features from four conv. layers and then use a question-guided attention on these layers to fuse their features. We do this to exploit the maximum potential of the subsequent dense co-attention layers. We conjecture that features at different levels in the hierarchy of visual representation [33, 30] will be necessary to correctly answer a wide range of questions.

To be specific, we extract outputs from the four conv. layers (after ReLU) before the last four pooling layers. These are tensors of different sizes (i.e., $256 \times 112 \times 112$,
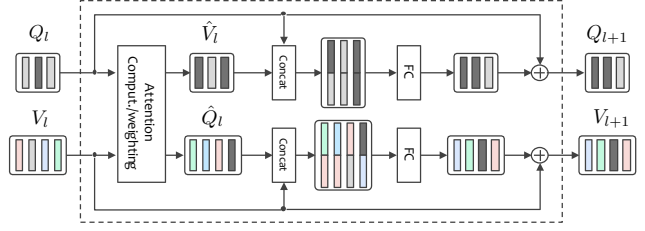

Figure 2: The internal structure of a single dense co-attention layer of layer index $l + 1$.

$512 \times 56 \times 56, 1024 \times 28 \times 28$, and $2048 \times 14 \times 14$) and are converted into tensors of the same size ($d \times 14 \times 14$) by applying max pooling with a different pooling size and one-by-one convolution to each. We also apply l_2 normalization on the depth dimension of each tensor as in [2]. We reshape the normalized tensors into four $d \times T$ matrices, where $T = 14 \times 14$.

Next, attention on the four layers is created from s_Q, the representation of the whole question defined above. We use a two-layer neural network having 724 hidden units with ReLU non-linearity to project s_Q to the scores of the four layers as

$$[s_1, s_2, s_3, s_4] = \text{MLP}(s_Q), \tag{3}$$

which are then normalized by softmax to obtain four attention weights $\alpha_1, ..., \alpha_4$. The weighted sum of the above four matrices is computed, yielding a $d \times T$ matrix $V = [v_1, ..., v_T]$, which is our representation of the input image. It stores the image feature at the t-th image region in its t-th column vector of size d.

3.2. Dense Co-Attention Layer

3.2.1 Overview of the Architecture

We now describe the proposed dense co-attention layer; see Fig.2. It takes the question and image representations Q and V as inputs and then outputs their updated versions. We denote the inputs to the $(l + 1)$-st layer by $Q_l = [q_{l1}, ..., q_{lN}] \in \mathbb{R}^{d \times N}$ and $V_l = [v_{l1}, ..., v_{lT}] \in \mathbb{R}^{d \times T}$. For the first layer inputs, we set $Q_0 = Q = [q_1, ..., q_N]$ and $V_0 = V = [v_1, ..., v_T]$.

The proposed architecture has the following properties. First, it is a co-attention mechanism [17]. Second, the co-attention is *dense* in the sense that it considers every interaction between any word and any region. To be specific, our mechanism creates one attention map on regions per each word and creates one attention map on words per each region (see Fig.3). Third, it can be stacked as shown in Fig.1.

3.2.2 Dense Co-attention Mechanism

Basic method for attention creation For the sake of explanation, we first explain the basic method for creation of
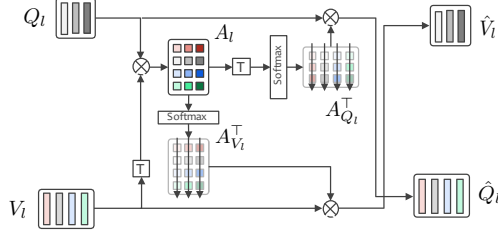
Figure 3: Computation of dense co-attention maps and attended representations of the image and question.

attention maps, which we will extend later. Given Q_l and V_l, two attention maps are created as shown in Fig.3. Their computation starts with the affinity matrix

$$A_l = V_l^\top W_l Q_l, \tag{4}$$

where W_l is a learnable weight matrix. We normalize A_l in row-wise to derive attention maps on question words conditioned by each image region as

$$A_{Q_l} = \text{softmax}(A_l), \tag{5}$$

and also normalize A_l in column-wise to derive attention maps on image regions conditioned by each question word as

$$A_{V_l} = \text{softmax}(A_l^\top). \tag{6}$$

Note that each row of A_{Q_l} and A_{V_l} contains a single attention map.

Nowhere-to-attend and memory It often occurs at the creation and application of each attention map that there is no particular region or word that the model should attend. To deal with such cases, we add K elements to N question words as well as to T image regions, as in [26]. In [26], the authors only use $K = 1$, but we found it effective to use $K > 1$, which is expected to additionally serve as a memory for storing useful information [8]. More specifically, incorporating two matrices $M_{Q_l} \equiv [q_{l\oslash_1}, ..., q_{l\oslash_K}] \in \mathbb{R}^{d \times K}$ and $M_{V_l} \equiv [v_{l\oslash_1}, ..., v_{l\oslash_K}] \in \mathbb{R}^{d \times K}$, which are learnable parameters, we augment the matrix Q_l and V_l in the row direction as $\tilde{Q}_l = [q_{l1}, ..., q_{lN}, q_{l\oslash_1}, ..., q_{l\oslash_K}] \in \mathbb{R}^{d \times (N+K)}$ and $\tilde{V}_l = [v_{l1}, ..., v_{lT}, v_{l\oslash_1}, ..., v_{l\oslash_K}] \in \mathbb{R}^{d \times (T+K)}$. This augmentation of Q_l and V_l provides A_l of size $(T + K) \times (N + K)$; A_{Q_l} and A_{V_l} are of size $(T + K) \times (N + K)$ and $(N + K) \times (T + K)$, respectively.

Parallel attention In several studies [12, 25], multiple attention maps are created and applied to target features in a parallel manner, which provides multiple attended features, and then they are *fused by concatenation*. In [25], features are first linearly projected to multiple lower-dimensional spaces, for each of which the above attention function is

performed. We adopt a similar approach that uses multiple attention maps here, but we use *average* instead of concatenation for fusion of the multiple attended features, because we found it works better in our case.

To be specific, we linearly project the d-dimensional features (stored in the columns) of $\tilde{V}_l$ and $\tilde{Q}_l$ to multiple lower dimensional spaces. Let h be the number of lower dimensional spaces and $d_h (\equiv d/h)$ be their dimension. We denote the linear projections by $W_{\tilde{V}_l}^{(i)} \in \mathbb{R}^{d_h \times d}$ and $W_{\tilde{Q}_l}^{(i)} \in \mathbb{R}^{d_h \times d}$ $(i = 1, \ldots, h)$. Then the affinity matrix between the projected features in the i-th space is given as

$$A_l^{(i)} = (W_{\tilde{V}_l}^{(i)} \tilde{V}_l)^\top (W_{\tilde{Q}_l}^{(i)} \tilde{Q}_l). \tag{7}$$

Attention maps are created from each affinity matrix by column-wise and row-wise normalization as

$$A_{Q_l}^{(i)} = \text{softmax}\left(\frac{A_l^{(i)}}{\sqrt{d_h}}\right), \tag{8}$$

$$A_{V_l}^{(i)} = \text{softmax}\left(\frac{A_l^{(i)\top}}{\sqrt{d_h}}\right). \tag{9}$$

As we employ multiplicative (or dot-product) attention as explained below, average fusion of multiple attended features is equivalent to averaging our attention maps as

$$A_{Q_l} = \frac{1}{h} \sum_{i=1}^{h} A_{Q_l}^{(i)}, \tag{10}$$

$$A_{V_l} = \frac{1}{h} \sum_{i=1}^{h} A_{V_l}^{(i)}. \tag{11}$$

Attended feature representations We employ multiplicative attention to derive attended feature representations $\hat{Q}_l$ and $\hat{V}_l$ of the question and image, as shown in Fig.3. As A_{Q_l} and A_{V_l} store attention maps in their rows and their last K rows correspond to "nowhere-to-attend" or memory, we discard them when applying them to $\tilde{Q}_l$ and $\tilde{V}_l$ as

$$\hat{Q}_l = \tilde{Q}_l A_{Q_l}[\texttt{1:T}, :]^\top, \tag{12}$$

and

$$\hat{V}_l = \tilde{V}_l A_{V_l}[\texttt{1:N}, :]^\top, \tag{13}$$

respectively[1]. Note that $\hat{Q}_l$ is the same size as V_l (i.e. $d \times T$) and $\hat{V}_l$ is the same size as Q_l (i.e. $d \times N$).

3.2.3 Fusing Image and Question Representations

After computing the attended feature representations $\hat{Q}_l$ and $\hat{V}_l$, we fuse the image and question representations, as

---

[1]The notation $(\texttt{1:T}, :)$ indicates the submatrix in the first T rows, as in Python.

shown in the right half of Fig.2. The matrix $\hat{V}_l$ stores in its n-th column the attended representation of the entire image conditioned on the n-th question word. Then, the n-th column vector $\hat{v}_{ln}$ is fused with the representation q_{ln} of n-th question word by concatenation to form $2d$-vector $[q_{ln}^\top, \hat{v}_{ln}^\top]^\top$. This concatenated vector is projected back to a d-dimensional space by a single layer network followed by the ReLU activation and residual connection as

$$q_{(l+1)n} = \text{ReLU}\left(W_{Q_l}\begin{bmatrix} q_{ln} \\ \hat{v}_{ln} \end{bmatrix} + b_{Q_l}\right) + q_{ln}, \quad (14)$$

where $W_{Q_l} \in \mathbb{R}^{d \times 2d}$ and $b_{Q_l} \in \mathbb{R}^d$ are learnable weights and biases. An identical network (with the same weights and biases) is applied to each question word $(n = 1, \ldots, N)$ independently, yielding $Q_{l+1} = [q_{(l+1)1}, \ldots, q_{(l+1)N}] \in \mathbb{R}^{d \times N}$.

Similarly, the representation v_{lt} of t-th image region is concatenated with the representation $\hat{q}_{lt}$ of the whole question words conditioned on the t-th image region, and then projected back to a d-dimensional space as

$$v_{(l+1)t} = \text{ReLU}\left(W_{V_l}\begin{bmatrix} v_{lt} \\ \hat{q}_{lt} \end{bmatrix} + b_{V_l}\right) + v_{lt}, \quad (15)$$

where $W_{V_l} \in \mathbb{R}^{d \times 2d}$ and $b_{V_l} \in \mathbb{R}^d$ are weights and biases. The application of an identical network to each image region $(t = 1, \ldots, T)$ yields $V_{l+1} = [v_{(l+1)1}, \ldots, v_{(l+1)T}] \in \mathbb{R}^{d \times T}$.

It should be noted that the above two fully-connected networks have different parameters (i.e., W_{Q_l}, W_{V_l} etc.) for each layer l.

3.3. Answer Prediction

Given the final outputs Q_L and V_L of the last dense co-attention layer, we predict answers. As they contain the representation of N question words and T image regions, we first perform self-attention function on each of them to obtain aggregated representations of the whole question and image. This is done for Q_L as follows: i) compute 'scores' $s_{q_{L1}}, \ldots, s_{q_{LN}}$ of $q_{L1}, \ldots, q_{LN}$ by applying an identical two-layer MLP with ReLU nonlinearity in its hidden layer; ii) then apply softmax to them to derive attention weights $\alpha_1^Q, \ldots, \alpha_N^Q$; and iii) compute an aggregated representation by $s_{Q_L} = \sum_{n=1}^N \alpha_n^Q q_{Ln}$. Following the same procedure with an MLP with different weights, we derive attention weights $\alpha_1^V, \ldots, \alpha_T^V$ and then compute an aggregated representation s_{V_L} from $v_{L1}, \ldots, v_{LT}$.

Using s_{Q_L} and s_{V_L} thus computed, we predict answers. We consider three methods to do this here. The first one is to compute inner product between the sum of s_{Q_L} and s_{V_L} and s_A, the answer representation defined in Sec.3.1.1, as

(score of the answer encoded as s_A)

$$= \sigma\left(s_A^\top W\left(s_{Q_L} + s_{V_L}\right)\right), \quad (16)$$

Table 1: Ablation study on each module of DCNs using the validation set of the Open-Ended task (VQA 2.0). * indicates modules employed in the final model.

Category	Detail	Accuracy
Attention direction	I ← Q	60.95
	I → Q	62.63
	I ↔ Q*	62.94
Memory size (K)	1	62.53
	3*	62.94
	5	62.83
Number (h) of parallel attention maps	2	62.82
	4*	62.94
	8	62.81
Number (L) of stacked layers	1	62.43
	2	62.82
	3*	62.94
	4	62.67
Attention in answer prediction layer	Attention used*	62.94
	Avg of features	61.63
Attention in image extraction layer	Attention used*	62.94
	Only last conv layer	62.39

where σ is the logistic function and W is a learnable weight matrix. The second and third ones are to use a MLP for computing scores for a set of predefined answers, which is a widely used approach in recent studies. The two differ in how to fuse s_{Q_L} and s_{V_L}, i.e., summation

$$(\text{score of answers}) = \sigma\left(\text{MLP}\left(s_{Q_L} + s_{V_L}\right)\right), \quad (17)$$

or concatenation

$$(\text{score of answers}) = \sigma\left(\text{MLP}\left(\begin{bmatrix} s_{Q_L} \\ s_{V_L} \end{bmatrix}\right)\right), \quad (18)$$

where MLP is a two layer MLP having 1024 hidden units with ReLU non-linearity. The first one is the most flexible, as it allows us to deal with any answers that are not considered at the time of training the entire network.

4. Experiments

In this section, we present results of the experiments conducted to evaluate the proposed method.

4.1. Datasets

We used two most popular datasets, VQA [2] and VQA 2.0 [7], for our experiments. VQA (also known as VQA 1.0) contains human-annotated question-answer pairs on 204,721 images from Microsoft COCO dataset [16]. There are three predefined splits of questions, *train*, *val* and *test* or *test-standard*, which consist of 248,349, 121,512, and

Table 2: Results of the proposed method along with published results of others on VQA 1.0 in similar conditions (i.e., a single model; trained without an external dataset).

Model	Test-dev				Test-standard			
	Overall	Other	Number	Yes/No	Overall	Other	Number	Yes/No
VQA team [2]	57.75	43.08	36.77	80.50	58.16	43.73	36.53	80.569
SMem [27]	57.99	43.12	37.32	80.87	58.24	43.48	37.53	80.80
SAN [28]	58.70	46.10	36.60	79.30	58.90	-	-	-
FDA [11]	59.24	45.77	36.16	81.14	59.54	-	-	-
DNMN [1]	59.40	45.50	38.60	81.10	59.40	-	-	-
HieCoAtt [17]	61.00	51.70	38.70	79.70	62.10	-	-	-
RAU [20]	63.30	53.00	39.00	81.90	63.20	52.80	38.20	81.70
DAN [19]	64.30	53.90	39.10	83.00	64.20	54.00	38.10	82.80
Strong Baseline [12]	64.50	55.20	39.10	82.20	64.60	55.20	39.10	82.00
MCB [5]	64.70	55.60	37.60	82.50	-	-	-	-
N2NMNs [10]	64.90	-	-	-	-	-	-	-
MLAN [31]	64.60	53.70	40.20	83.80	64.80	53.70	40.90	83.70
MLB [14]	65.08	54.87	38.21	84.14	65.07	54.77	37.90	84.02
MFB [32]	65.90	56.20	39.80	84.00	65.80	56.30	38.90	83.80
MF-SIG-T3 [4]	66.00	56.37	39.34	84.33	65.88	55.89	38.94	84.42
DCN (16)	66.43	56.23	42.37	84.75	66.39	56.23	41.81	84.53
DCN (17)	**66.89**	57.31	**42.35**	**84.61**	**67.02**	**56.98**	**42.34**	**85.04**
DCN (18)	66.83	**57.44**	41.66	84.48	66.66	56.83	41.27	84.61

Table 3: Results of the proposed method along with published results of others on VQA 2.0 in similar conditions (i.e., a single model; trained without an external dataset). DCN(number) indicates the DCN equipped with the prediction layer that uses equation (number) for score computation. *: trained with external datasets. ‡: the winner of VQA challenge 2017, unpublished.

Model	Test-dev				Test-standard			
	Overall	Other	Number	Yes/No	Overall	Other	Number	Yes/No
VQA team-Prior [7]	-	-	-	-	25.98	01.17	00.36	61.20
VQA team-Language only [7]	-	-	-	-	44.26	27.37	31.55	67.01
VQA team-LSTM+CNN [7]	-	-	-	-	54.22	41.83	35.18	73.46
MCB [5] reported in [7]	-	-	-	-	62.27	53.36	38.28	78.82
MF-SIG-T3 * [4]	64.73	55.55	42.99	81.29	-	-	-	-
Adelaide Model * ‡ [24]	62.07	52.62	39.46	79.20	62.27	52.59	39.77	79.32
Adelaide + Detector * ‡ [24]	65.32	56.05	44.21	81.82	65.67	56.26	43.90	82.20
DCN (16)	**66.87**	**57.26**	46.61	83.51	66.97	**57.09**	46.98	83.59
DCN (17)	66.72	56.77	**46.65**	**83.70**	**67.04**	56.95	**47.19**	83.85
DCN (18)	66.60	56.72	46.60	83.50	67.00	56.90	46.93	**83.89**

244,302 questions, respectively. There is also a 25% subset of the test-standard set referred to as *test-dev*. All the questions are categorized into three types: yes/no, number, and other. Each question has 10 free-response answers. VQA 2.0 is an updated version of VQA 1.0 and is the largest as of now. Compared with VQA 1.0, it contains more samples (443,757 *train*, 214,354 *val*, and 447,793 *test* questions), and is more balanced in term of language bias. We evaluate our models on the challenging Open-Ended task of both

datasets.

As in [24], we choose correct answers appearing more than 5 times for VQA and 8 times for VQA 2.0 to form the set of candidate answers. Following previous studies, we train our network on train + val splits and report the test-dev and test-standard results from the VQA evaluation server (except for the ablation test shown below). We use the evaluation protocol of [2] in all the experiments.

What are these animals	What are these animals	
Pred: Giraffes, Ans: Giraffes		

What are these animals	What are these animals	
Pred: Cows, Ans: Cows		

Is it cloudy · Is it cloudy
Pred: No, Ans: No

Is it cloudy · Is it cloudy
Pred: Yes, Ans: Yes

Figure 4: Typical examples of attended image regions and question words for complementary image-question pairs from VQA 2.0 dataset. Each row contains visualization for two pairs of the same question but different images and answers. The original image and question are shown along with their attention maps generated in the answer prediction layer. The brightness of image pixels and redness of words indicate the attention weights.

4.2. Experimental Setup

For both of the datasets, we use the Adam optimizer with the parameters $\alpha = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. During the training procedure, we make the learning rate (α) decay at every 4 epochs for VQA and 7 epochs for VQA 2.0 with an exponential rate of 0.5. All models are trained up to 16 and 21 epochs on VQA and VQA 2.0, respectively. To prevent overfitting, dropouts [23] are used after each fully connected layers with a dropout ratio $p = 0.3$ and after the LSTM with a dropout ratio $p = 0.1$. The batch size is set to 160 and 320 for VQA and VQA 2.0. We set the dimension d of the feature space in the dense co-attention layers (equivalently, the size of its hidden layers) to be 1024.

4.3. Ablation Study

The architecture of the proposed DCN is composed of multiple modules. To evaluate the contribution of each module to final prediction accuracy, we conducted ablation tests. Using VQA 2.0, we evaluated several versions of DCNs with different parameters and settings by training them on the train split and calculating its performance on the val split. The results are shown in Table 1.

The first block of the table shows the effects of image-question co-attention. The numbers are performances obtained by a DCN with only question-guided attention on images $(I \leftarrow Q)$, with only image-guided attention on question words $(I \rightarrow Q)$, and the standard DCN with co-attention $(I \leftrightarrow Q)$. The single-direction variants generates only attention in either side of the two paths in the dense co-attention layer; the rest of the computations remain the same. The network with co-attention performs the best, verifying the effectiveness of our co-attention implementation.

The second block of the table shows the impacts of K, which is the row size of M_{Q_l} and M_{V_l} that are used for augmenting Q_l and V_l. This augmentation is originally introduced to be able to deal with "nowhere to attend", which can be implemented by $K = 1$ [26]. However, we found that the use of $K > 1$ improves performance to a certain extent, which we think is because M_{Q_l} and M_{V_l} work as external memory that can be used through attention mechanism [8]. As shown in the table, $K = 3$ yields the best performance.

The third and fourth blocks of the table show choices of the number h of parallel attention maps and L of stacked layers. The best result was obtained for $h = 4$ and $L = 3$.

The last two blocks of the table show effects of the use of attention in the answer prediction layer and the image extraction layer; the use of attention improves accuracy by about 1.3% and 0.5%, respectively.

4.4. Comparison with Existing Methods

Table 2 shows the performance of our method on VQA 1.0 along with published results of others. The entries 'DCN (n)' indicate which method for score computation is employed from (16)-(18). It is seen from the table that our method outperforms the best published result (MF-SIG-T3) by a large margin of $0.9\% \sim 1.1\%$ on both test-dev and test-standard sets. Furthermore, the improvements can be seen in all of the entries (*Other* with 1.1%, *Number* with 3.4%, *Yes/No* with 0.6% on test-standard set) implying the capacity of DCNs to model multiple types of complex relations between question-image pairs. Notably, we achieve significant improvements of 3.0% and 3.4% for the question type *Number* on test-dev and test-standard sets, respectively.

Table 2 also shows the performances of DCNs with a different answer prediction layer that uses (16), (17), and (18) for score computation. It is seen that (17) shows at least comparable performance to the others and even attains the best performance of 67.02% in test-standard set.

Table 3 shows comparisons of our method to previous published results on VQA 2.0 and also that of the winner of VQA 2.0 Challenge 2017 in both test-dev and test-standard sets. It is observed in Table 3 that our approach outperforms the state-of-the-art published method (MF-SIG-T3) by a large margin of 2.1% on test-dev set, even though the MF-SIG-T3 model was trained with VQA 2.0 and an augmented dataset (Visual Genome [15]). It is noted that the improvements are seen in all the question types (*Other* with 1.71%, *Number* with 3.66%, and *Yes/No* with 2.41%). Comparing our DCN with the winner of VQA 2.0 Challenge 2017, Adelaide model. Our best DCN (17) delivers 1.5% and 1.37% improvements in every question types over the Adelaide+Detector on test-dev and test-standard, respectively. It is worth to point out that the winner method uses a detector (Region Proposal Network) trained on annotated regions of Visual Genome dataset [15] to extract visual features; and that the model is trained using also an external dataset, i.e., the Visual Genome question answering dataset.

It should also be noted that while achieving the best performance in VQA dataset, the size of the DCNs (i.e., the number of parameters) is comparable or even smaller than the former state-of-the-art methods, as shown in Table 4.

4.5. Qualitative Evaluation

Complementary image-question pairs are available in VQA 2.0 [7], which are pairs of the same question and different images with different answers. To understand the behaviour of the trained DCN, we visualize attention maps that the DCN generates for some of the complementary image-question pairs. Specifically, we show multiplication of an input image and question with their attention maps $\alpha_1^V, \ldots, \alpha_T^V$ and $\alpha_1^Q, \ldots, \alpha_N^Q$ (defined in Sec.3.3) generated in the answer prediction layer. A typical example is shown

Table 4: Model sizes of DCNs and several bilinear fusion methods. The numbers include the parameters of LSTM networks and exclude those of ResNets.

Model	No. params
MCB [5]	63M
MLB [14]	25M
MFB [32]	46M
DCN (18)	32M
DCN (17)	31M
DCN (16)	28M

in Fig.4. Each row shows the results for two pairs of the same question and different images, from which we can observe that the DCN is able to look at right regions to find the correct answers. Then, the first column shows the results for two pairs of the same image and different questions. It is observed that the DCN focuses on relevant image regions and question words to produce answers correctly. More visualization results including failure cases are provided in the supplementary material.

5. Conclusion

In this paper, we present a novel network architecture for VQA named the dense co-attention network. The core of the network is the dense co-attention layer, which is designed to enable improved fusion of visual and language representations by considering dense symmetric interactions between the input image and question. The layer can be stacked to perform multi-step image-question interactions. The layer stack combined with the initial feature extraction step and the final answer prediction layer, both of which have their own attention mechanisms, form the dense co-attention network. The experimental results on two datasets, VQA and VQA 2.0, confirm the effectiveness of the proposed architecture.

Acknowledgement

References

[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *The*

Association for Computational Linguistics (HLT-NAACL), 2016. 6

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 3, 5, 6

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[4] Z. Chen, Z. Yanpeng, H. Shuaiyi, T. Kewei, and M. Yi. Structured attentions for visual question answering. In *International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 3, 6

[5] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 1, 2, 6, 8

[6] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[7] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 6, 8

[8] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 4, 7

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[10] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 6

[11] I. Ilievski, S. Yan, and J. Feng. A focused dynamic attention model for visual question answering. *arXiv preprint arXiv:1604.01485*, 2016. 2, 6

[12] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017. 2, 4, 6

[13] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal Residual Learning for Visual QA. In *International Conference on Neural Information Processing Systems (NIPS)*, 2016. 2, 3

[14] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang. Hadamard product for low-rank bilinear pooling. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 2, 3, 6, 8

[15] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 2017. 8

[16] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L.

Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 5

[17] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *International Conference on Neural Information Processing Systems (NIPS)*, 2016. 1, 2, 3, 6

[18] B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in translation: Contextualized word vectors. *arXiv preprint arXiv:1708.00107*, 2017. 3

[19] H. Nam, J. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. *arXiv preprint arXiv:1611.00471*, 2016. 2, 6

[20] H. Noh and B. Han. Training recurrent answering units with joint loss minimization for VQA. *arXiv preprint arXiv:1606.03647*, 2016. 2, 6

[21] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 3

[22] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 2014. 7

[24] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *arXiv preprint arXiv:1708.02711*, 2017. 2, 3, 6

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 4

[26] C. Xiong, V. Zhong, and R. Socher. Dynamic coattention networks for question answering. In *International Conference on Learning Representations (ICLR)*, 2017. 4, 7

[27] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 6

[28] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 6

[29] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *International Conference on Neural Information Processing Systems (NIPS)*, 2014. 3

[30] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *ICML Workshop on Deep Learning*, 2015. 3

[31] D. Yu, J. Fu, Y. Rui, and T. Mei. Multi-level attention networks for visual question answering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 6

[32] Z. Yu, J. Yu, J. Fan, and D. Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question

answering. In *International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 6, 8

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014. 3