# LANGUAGE AND VISUAL RELATIONS ENCODING FOR VISUAL QUESTION ANSWERING

*Fei Liu*[1,2], *Jing Liu*[1 *], *Zhiwei Fang*[1,2], *Hanqing Lu*[1]

[1] National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
*liufei2017@ia.ac.cn, {jliu, zhiwei.fang, luhq}@nlpr.ia.ac.cn*

## ABSTRACT

Visual Question Answering (VQA) involves complex relations of two modalities, including the relations between words and between image regions. Thus, encoding these relations is important to accurate VQA. In this paper, we propose two modules to encode the two types of relations respectively. The language relation encoding module is proposed to encode multi-scale relations between words via a novel masked self-attention. The visual relation encoding module is proposed to encode the relations between image regions. It computes the response at a position as a weighted sum of the features at other positions in the feature maps. Extensive experiments demonstrate the effectiveness of each modules. Our model achieves state-of-the-art performance on the VQA 1.0 dataset.

***Index Terms—*** Visual question answering, Relations, Attention

## 1. INTRODUCTION

Recently, Visual Question Answering (VQA) has gained increasing attention as an interdisciplinary subject across computer vision and natural language processing. It aims at answering a natural language question about a given image.

In a question, each word correlates with some other words, and the same word may convey different meaning in different context. To understand the textual content of the question, it is necessary to encode the dependency relationships between words. Some earlier works [1, 2] used word embedding to represent the question, which do not consider the relations between words. These methods achieved low accuracy on the VQA dataset. Currently, recurrent neural networks (RNN) [3, 4] are commonly used to encode long-range relations, but is hard to parallelize and not time-efficient due to the sequential nature of recurrent models. Very recently, some works [5, 6] have introduced self-attention mechanism instead of RNN for relation encoding, achieving state-of-the-art performance on several NLP tasks. In contrast to RNN, self-attention mechanism has more flexibility in modeling long-range relations, and its computation can be easily and significantly accelerated by existing parallel computing schemes. In [5, 6], the proposed methods take into account the relations between all words (we call *global relation*), but neglect the *local relation*. We think that local relation may be more suitable in VQA, since a few key words are enough to obtain the correct answer.

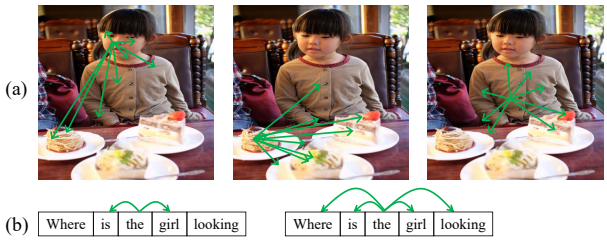


**Fig. 1**. (a) Visual relation encoding. It encodes the relations between image regions. (b) Language relation encoding. It encodes multi-scale relations between words.

In a typical image encoding process, convolutional neural networks (CNN) is employed, and then the grid features with a fixed splitting pattern on feature maps are extracted individually, while the relations between image regions are not considered. Usually, many VQA questions involve multiple objects in an image, and accurate answering such questions not only needs to recognize the objects, but also needs to capture the relations between them. For the example shown in Fig.1, the question is "where is the girl looking". The model first needs to recognize the *girl* and *food* on plate, and then correlates the two objects. Besides, in the case when some image regions contain different object parts, the local view may lead to incorrect recognition, or produce ambiguity due to similar appearance. Thus, the model needs some correlated cues from other parts of the object in adjacent image regions or other objects belonging to the same category in distant image regions, to enrich semantic information and clarify local confusion. Thus, encoding relations between image regions is important for image understanding and further beneficial for accurate question answering.

Motivated by the above observations, we propose two modules to encode language and visual relations, respec-

* Jing Liu is the corresponding author

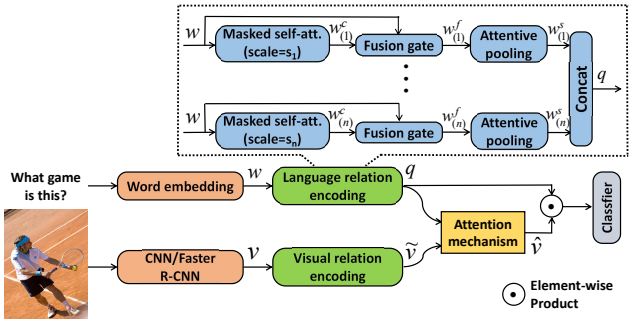


**Fig. 2**. Overview of the proposed VQA model. We highlight our main contributions in green.

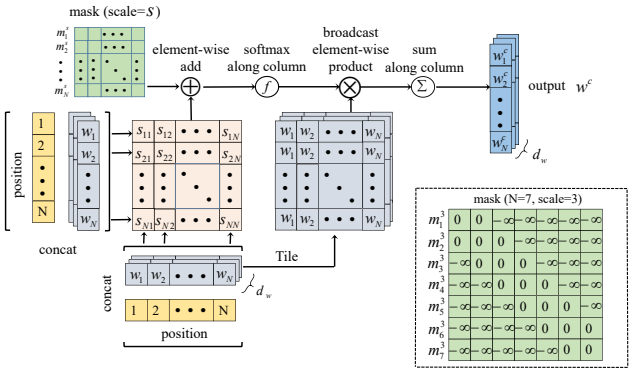


**Fig. 3**. Masked self-attention. $s_{ij}$ is computed by Eq. 3. The mask is defined in Eq. 2. On the bottom right shows an example of the mask with $N = 7$ and scale=3.

tively. To encode the relations between words, we propose a *language relation encoding* module (see the black dashed box in Fig.2). The module has multiple branches, each of which is composed of a *masked self-attention*, a *fusion gate* and an *attentive pooling*. The masked self-attention is the core component, which introduces a mask with a given scale to restrict the relation range of each word. Different branches use the masks with different scales to capture multi-scale relations including global and local relations, which gets rid of the limitation of the methods in [5, 6]. To encode the relations between image regions, we propose a *visual relation encoding* module derived from self-attention mechanism. The module updates the features at each position using a weighted sum of the most related $k$ feature vectors in the image feature maps. In other words, for each image region, the module models the relations between it and $k$ related regions. In fact, the above two methods are connected and all use attention mechanisms to perform relation encoding.

Our main contributions are as follows: **(1)** We propose two novel modules to encode relations between words and between image regions, respectively. This is the first time to explore the relations between words and between image regions in a unified framework for the VQA task. **(2)** Extensive experiments show the effectiveness of the proposed relation encoding modules. Our approach achieves new state-of-the-art results on the VQA 1.0 dataset.

## 2. PROPOSED APPROACH

Our proposed VQA model is illustrated in Fig. 2. We will elaborate each module separately below.

### 2.1. Language Relation Encoding

The proposed *language relation encoding* module has multiple parallel branches, each of which consists of one *masked self-attention*, one *fusion gate* and one *attentive pooling* (see Fig.2). The outputs of all branches are concatenated as the final question representation. For clarity, we present the forward process of one branch, and ignore the subscript $(i)$ that denotes the *i-th* branch.

The masked self-attention is illustrated in Fig. 3. A question consisting of $N$ words is first converted into a sequence of GloVe [7] vectors $\boldsymbol{w} = \{\boldsymbol{w}_i\}_{i=1}^N \in \mathbb{R}^{d_w \times N}$. To recover temporal order information that is lost in self-attention mechanism, we encode position information for each word. Specifically, we concatenate the embedding of each word with its position, which is denoted as:

$$\boldsymbol{w^p} = \{\boldsymbol{w^p}_i\}_{i=1}^N = \{[\boldsymbol{w}_i, i]\}_{i=1}^N \tag{1}$$

where $\boldsymbol{w^p} \in \mathbb{R}^{(d_w+1)\times N}$ is the word embeddings with position information encoded.

We take the *i-th* word as the query, and describe how to encode language relations. First, we calculate similarity scores between the *i-th* word and all words by Eq. 3, denoted as $\boldsymbol{s}_i = \{s_{ij}\}_{j=1}^N \in \mathbb{R}^N$. We consider the relations between the query and a few adjacent words (*i.e.* local relation). Thus, a mask $\boldsymbol{m}_i^s = \{m_{ij}^s\}_{j=1}^N \in \mathbb{R}^N$, where the superscript $s$ denotes the scale, is introduced to restrict the relation range. The mask is defined as

$$m_{ij}^s = \begin{cases} 0, & i - \frac{s-1}{2} \leqslant j \leqslant i + \frac{s-1}{2} \\ -\infty, & \text{otherwise} \end{cases} \tag{2}$$

$\boldsymbol{m}_i^s$ is added to the similarity scores $\boldsymbol{s}_i$, and then a softmax function transforms the scores to a probability distribution $\boldsymbol{\alpha}_i = \{\alpha_{ij}\}_{j=1}^N \in \mathbb{R}^N$ (Eq. 5). Due to the property of the softmax operation, the value 0 in the mask denotes reserved positions and $-\infty$ stands for unconsidered positions. Finally, the relation-encoded representation of the *i-th* word is obtained by Eq. 6.

$$s_{ij} = W_p \operatorname{ReLU}(U_p \boldsymbol{w^p}_i + V_p \boldsymbol{w^p}_j) \tag{3}$$

$$\boldsymbol{s}_i = \{s_{ij}\}_{j=1}^N \tag{4}$$

$$\boldsymbol{\alpha}_i = \operatorname{softmax}(\boldsymbol{s}_i + \boldsymbol{m}_i^s) \tag{5}$$

$$\boldsymbol{w^c}_i = \textstyle\sum_{j=1}^N \alpha_{ij} \boldsymbol{w}_j \tag{6}$$

where $W_p \in \mathbb{R}^{1\times \frac{d_w}{2}}$, $U_p \in \mathbb{R}^{\frac{d_w}{2}\times(d_w+1)}$ and $V_p \in \mathbb{R}^{\frac{d_w}{2}\times(d_w+1)}$ are learned weight matrices. $\boldsymbol{w^c} = \{\boldsymbol{w^c}_i\}_{i=1}^N \in \mathbb{R}^{d_w \times N}$ is the output of masked self-attention.

Then, we use a *fusion gate* to merge the input and the output of the masked self-attention dynamically. The fused representations of all words are computed by:

$$\boldsymbol{g} = \text{sigmoid}(W_f \boldsymbol{w^c} + U_f \boldsymbol{w}) \tag{7}$$

$$\boldsymbol{w^f} = \boldsymbol{g} \odot \boldsymbol{w^c} + (1 - \boldsymbol{g}) \odot \boldsymbol{w} \tag{8}$$

where $W_f \in \mathbb{R}^{d_w \times d_w}$, $U_f \in \mathbb{R}^{d_w \times d_w}$ are learned weights, $\odot$ represents element-wise product. In the end, we use an *attentive pooling* to compress the sequence $\boldsymbol{w^f}$ as a vector:

$$\boldsymbol{h} = W_s \, \text{ReLU}(U_s \boldsymbol{w^f}) \tag{9}$$

$$\boldsymbol{a} = \text{softmax}(\boldsymbol{h}) \tag{10}$$

$$\boldsymbol{w^s} = \textstyle\sum_{i=1}^{N} a_i \boldsymbol{w^f}_i \tag{11}$$

where $W_s \in \mathbb{R}^{1 \times \frac{d_w}{2}}$ and $U_s \in \mathbb{R}^{\frac{d_w}{2} \times d_w}$ are learned weights, $\boldsymbol{a} = \{a_1, ..., a_N\} \in \mathbb{R}^N$ is attention weights, $\boldsymbol{w^s} \in \mathbb{R}^{d_w}$ is the output of one branch. We use the concatenation of the outputs of all branches as the final question representation $\boldsymbol{q} \in \mathbb{R}^{n d_w}$ ($n$ is the number of branches).

### 2.2. Visual Relation Encoding

The visual relation encoding module computes the response at a position as a weighted sum of the most related $k$ feature vectors in the input feature maps. Given the visual features $\boldsymbol{v} = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_K\} \in \mathbb{R}^{d_v \times K}$, for the feature vector $\boldsymbol{v}_i$ that corresponds to the *i-th* position in the feature maps, we first compute the relevance scores of $\boldsymbol{v}_i$ and the feature vectors at all positions, $\boldsymbol{r}_i = \{r_{i1}, ..., r_{iK}\}$, where $r_{ij}$ is the relevance score of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, and is calculated by

$$r_{ij} = (W_r \boldsymbol{v}_i)^\top (U_r \boldsymbol{v}_j) \tag{12}$$

where $W_r, U_r \in \mathbb{R}^{\frac{d_v}{8} \times d_v}$ are learned weight matrices.

We select the most relevant $k$ feature vectors from $\boldsymbol{v}$ for relation encoding based on the relevance scores, obtaining $\boldsymbol{v}^k = \{\boldsymbol{v}_{l_1}, ..., \boldsymbol{v}_{l_k}\} \in \mathbb{R}^{d_v \times k}$. $\boldsymbol{l} = \{l_1, ..., l_k\}$ denotes the indexes of the top $k$ entries in $\boldsymbol{r}_i$. We also obtain the corresponding $k$ relevance scores $\boldsymbol{r}_i^k = \{r_{il_1}, ..., r_{il_k}\} \in \mathbb{R}^k$. The final response at the *i-th* position $\widetilde{\boldsymbol{v}}_i$ is calculated as follows.

We first normalize $\boldsymbol{r}_i^k$ using softmax (Eq.13) and linearly transform $\boldsymbol{v}^k$ using the weight matric $W_v$, then perform a weighted sum of the transformed feature vectors based on the normalized relevance scores (Eq.14). Finally, we multiply the response $\boldsymbol{o}_i$ by a scale parameter and add back the original feature vector (Eq.15). Formally,

$$\boldsymbol{\alpha}_i^k = \text{softmax}(\boldsymbol{r}_i^k) \tag{13}$$

$$\boldsymbol{o}_i = \textstyle\sum_{j=1}^{k} \alpha_{il_j}^k (W_v \boldsymbol{v}_{l_j}) \tag{14}$$

$$\widetilde{\boldsymbol{v}}_i = \lambda \boldsymbol{o}_i + \boldsymbol{v}_i \tag{15}$$

where $\widetilde{\boldsymbol{v}}_i$ is the final response at the *i-th* position, $W_v \in \mathbb{R}^{d_v \times d_v}$ is learned weight matrix, $\lambda$ is initialized as 0.

The visual relation encoding module is able to encode the dependency relationships between image regions, thus producing more expressive image representation. Its output is denoted as $\widetilde{\boldsymbol{v}} = \{\widetilde{\boldsymbol{v}}_1, ..., \widetilde{\boldsymbol{v}}_K\} \in \mathbb{R}^{d_v \times K}$.

### 2.3. Attention Mechanisms & Answer Prediction

Given $\boldsymbol{q}$ and $\widetilde{\boldsymbol{v}}$, we perform visual attention mechanism to obtain aggregated representation of the image:

$$s_i^I = W^I f_a([\widetilde{\boldsymbol{v}}_i \, ; \, \boldsymbol{q}]) \tag{16}$$

$$\boldsymbol{\alpha}^I = \text{softmax}(\boldsymbol{s}^I) \tag{17}$$

$$\hat{\boldsymbol{v}} = \textstyle\sum_{i=1}^{K} \alpha_i^I \widetilde{\boldsymbol{v}}_i \tag{18}$$

where $f_a(\cdot)$ denotes the *gated tanh* function [8] with parameters $a$. It is used to project the concatenated vector to 512-dimensional space. $W^I \in \mathbb{R}^{1 \times 512}$ is learnable weights, $\boldsymbol{s}^I$ is the scores of image regions, $\boldsymbol{\alpha}^I$ is attention weights, and $\hat{\boldsymbol{v}} \in \mathbb{R}^{d_v}$ is the attended image representation.

We then project $\hat{\boldsymbol{v}}$ and $\boldsymbol{q}$ to the same dimensional space (512 dimensions) using two *gated tanh* functions [8] with different parameters, respectively. The projected features are fused via element-wise product. Similar to [9, 10, 11], we treat VQA as a classification problem. The fused multi-modal features are fed into the *classifier* composed of 2-layer MLP with ReLU non-linearity function between the layers and a final softmax function, outputing a class probability vector. Cross-entropy loss is adopted as the objective function.

## 3. EXPERIMENTS

### 3.1. Setup

We use the VQA 1.0 [2] dataset for our experiments. VQA 1.0 is built from 204,721 MSCOCO [12] images with human annotated questions and answers. The dataset is divided into three splits: *train* (248,349 questions), *val* (121,512 questions) and *test* (244,302 questions). Following the previous works [9, 13, 14], we employ ResNet-152 [15] to produce image features of size $14 \times 14 \times 2048$ (*i.e.* $d_v = 2048, K = 196$), which are used for all experiments. When comparing with state-of-the-art methods, we also use bottom-up attention [8] features with size $36 \times 2048$ for fair comparison. The word embedding is of 300 dimension. As in [9], we choose the most frequent 3,000 answers in the *trainval* sets as the candidate answers. The model is trained using the AMSGrad [16] optimizer with an initial learning rate of $7 \times 10^{-4}$. We use the evaluation protocol of [2] in all the experiments.

### 3.2. Ablation Studies

We first investigate the influence of the number of branches and the scale (*i.e.* the relation range of each word) in the language relation encoding. The results are shown in Table 1. The baseline model has no the language relation encoding module, and averages the embeddings of all words as the question representation. We can see that the language relation encoding module largely improves the performance over the baseline. The accuracy improves as the number of branches increases. This is because more branches capture richer multi-scale relations, thus improving the understanding

| # Branch | Baseline | 1 (scale=3) | 1 (scale=7) | 1 (scale=13) | 1 (scale=*all*) | 2 (scale=7,13) | 3 (scale=7,13,*all*) |
|---|---|---|---|---|---|---|---|
| Accuracy | 61.0 | 62.2 | 62.3 | 62.4 | 62.2 | 62.7 | 62.9 |

**Table 1**. The comparison of different number of branches and scales in language relation encoding module on VQA 1.0 *val* set.

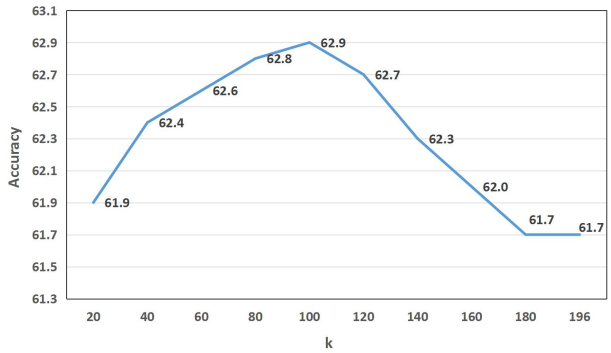


**Fig. 4**. Comparison of different values of $k$ in visual relation encoding on the VQA 1.0 *val* set.

| Model | Accuracy |
|---|---|
| Our model | 62.9 |
| Our model w/o position information | 62.6 |
| Our model w/o masked self-attention | 61.8 |
| Our model w/o fusion gate | 62.5 |
| Our model w/o attentive pooling | 62.5 |
| Our model w/o visual relation encoding | 62.0 |

**Table 2**. Ablation studies on VQA 1.0 *val* set. The table is divided into three parts. The first part shows the result of our full model. The second and last part are the ablation studies of language relation encoding and visual relation encoding, respectively.

of question. When using one branch, we can see that considering global relations (*i.e.* scale=$all$) doesn't obtain the optimal result, which indicates that local relations play a more important role than global relations in question encoding.

We then investigate the influence of the values of $k$ in visual relation encoding. As shown in Fig.4, starting from $k = 20$, the performance improves as the value of $k$ increases, then reaches the peak at $k = 100$, and finally drops as the value of $k$ increases. This phenomenon can be explained that when using a small $k$, some related image regions may be left out in relation encoding; when using a large $k$, irrelevant regions may be considered in relation encoding, the both cases will result in inaccurate relation modeling.

Finally, we conduct ablation studies to validate the effectiveness of each component. The results are shown in Table 2. Encoding position information (Eq.1) for each word improves the performance by 0.3%. Masked self-attention produces the largest performance gain (1.1%), which shows the importance of encoding the relations between words in VQA. Fusion gate results in 0.4% improvement. The *attentive pooling* in the end of each branch is used to compress the sequence as a vector.

| Model | Test-dev | | | | Test-std |
|---|---|---|---|---|---|
| | All | Other | No. | Y/N | All |
| QGHC [17] | 65.9 | 57.1 | 38.1 | 83.5 | 65.9 |
| VKMN [18] | 66.0 | 57.0 | 37.9 | 83.7 | 66.1 |
| MFH [19] | 66.8 | 57.4 | 39.7 | 85.0 | 66.9 |
| DCN [20] | 66.9 | 57.3 | 42.4 | 84.6 | 67.0 |
| DA-NTN [21] | 67.9 | 58.6 | 41.9 | 85.8 | 68.1 |
| CoR [22] | 68.4 | 59.1 | **44.1** | 85.7 | 68.5 |
| Ours | 67.2 | 57.5 | 40.6 | 85.6 | 67.4 |
| Ours + BU | **69.1** | **59.5** | **44.1** | **86.8** | **69.3** |

**Table 3**. Comparison with the state-of-the-arts on the VQA 1.0. All the reported results are obtained with a single model. BU: using bottom-up attention [8] features.

We remove the component and use *average pooling* to obtain a vector. The result shows that attentive pooling performs better than average pooling, as average pooling attaches equal importance to each word, while attentive pooling can focus on key words. Visual relation encoding achieves 0.9% accuracy improvement, which shows the importance of encoding the relations between image regions in VQA.

### 3.3. Comparison with State-of-the-arts

Table 3 shows the performance of our algorithm and state-of-the-art methods on VQA 1.0. With bottom-up attention [8] features, our approach outperforms other state-of-the-art methods in all question categories and overall accuracy. Compared with the most recent state-of-the-art model CoR [22], our model achieves new state-of-the-art results of 69.1% and 69.3% on test-dev and test-std sets, respectively.

## 4. CONCLUSIONS

In this paper, we propose two novel modules to encode language and visual relations, respectively. The language relation encoding module captures multi-scale relations between words via masked self-attention mechanisms. The visual relation encoding module encodes the relations between image regions. Extensive experiments validate the importance of relation encoding. Our model achieves new state-of-the-art performance on the VQA 1.0 dataset. The proposed relations encoding modules are applicable to a wide range of tasks involving multi-modal data.

**Acknowledgment:** This work was supported by Beijing Natural Science Foundation (4192059) and National Natural Science Foundation of China (61872366 and 61472422).

## 5. REFERENCES


[1] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, 2015.

[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] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *International Conference on Neural Information Processing Systems (NIPS)*, 2014.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *International Conference on Neural Information Processing Systems (NIPS)*, 2017.

[6] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," *arXiv preprint arXiv:1709.04696*, 2017.

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

[8] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] 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.

[10] H. Ben-younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," in *International Conference on Computer Vision (ICCV)*, 2017.

[11] M. R. Farazi and S. Khan, "Reciprocal attention fusion for visual question answering," in *The British Machine Vision Conference (BMVC)*, 2018.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014.

[13] 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.

[14] 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.

[15] 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.

[16] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations (ICLR)*, 2018.

[17] P. Gao, P. Lu, H. Li, S. Li, Y. Li, S. C.H.Hoi, and X. Wang, "Question-guided hybrid convolution for visual question answering," in *European Conference on Computer Vision (ECCV)*, 2018.

[18] Z. Su, C. Zhu, Y. Dong, D. Cai, Y. Chen, and J. Li, "Learning visual knowledge memory networks for visual question answering," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[19] Z. Yu, J. Yu, C. Xiang, J. Fan, and D. Tao, "Beyond bilinear: Generalized multi-modal factorized high-order pooling for visual question answering," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[20] D. Nguyen and T. Okatani, "Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] Y. Bai, J. Fu, T. Zhao, and T. Mei, "Deep attention neural tensor network for visual question answering," in *European Conference on Computer Vision (ECCV)*, 2018.

[22] C. Wu, J. Liu, X. Wang, and X. Dong, "Chain of reasoning for visual question answering," in *International Conference on Neural Information Processing Systems (NIPS)*, 2018.