

# Differential Networks for Visual Question Answering

Chenfei Wu, Jinlai Liu, Xiaojie Wang, Ruifan Li

Center for Intelligence Science and Technology  
Beijing University of Posts and Telecommunications  
{wuchenfei, liujinlai, xjwang, rfi}@bupt.edu.cn

## Abstract

The task of *Visual Question Answering* (VQA) has emerged in recent years for its potential applications. To address the VQA task, the model should fuse elements from both images and questions efficiently. Existing models fuse image feature element  $v_i$  and question feature element  $q_i$  directly, such as an element product  $v_i q_i$ . Those solutions largely ignore the following two key points: 1) Whether  $v_i$  and  $q_i$  are in the same space. 2) How to reduce the observation noises in  $v_i$  and  $q_i$ . We argue that two differences between those two feature elements themselves, like  $(v_i - v_j)$  and  $(q_i - q_j)$ , are more probably in the same space. And the difference operation would be beneficial to reduce observation noise. To achieve this, we first propose Differential Networks (DN), a novel plug-and-play module which enables differences between pair-wise feature elements. With the tool of DN, we then propose DN based Fusion (DF), a novel model for VQA task. We achieve state-of-the-art results on four publicly available datasets. Ablation studies also show the effectiveness of difference operations in DF model.

## 1 Introduction

Given an image and a related question, the Visual Question Answering (VQA) task requires the machine to determine the correct answer. From an application perspective, VQA improves human-computer interaction ability and can be applied to many scenarios such as smart home management systems and private virtual assistant (Kafle and Kanan 2017b). From a research perspective, VQA requires a joint understanding of images and questions, and can be considered as a component of Visual Turing Test (Malinowski and Fritz 2014). As a result, VQA has recently emerged as a challenging task and received more and more interest from researchers.

The basic framework of existing VQA models consists of three stages. First, the image and the question are encoded respectively. Second, the encoded image and question feature elements are fused. Third, the fused results are classified to derive the answer. Among them, the “fusion” in the second step is the key. Therefore, most studies focus on this point. Initially, linear models are used to fuse image and question feature elements (Yang et al. 2016; Lu et al. 2016;

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

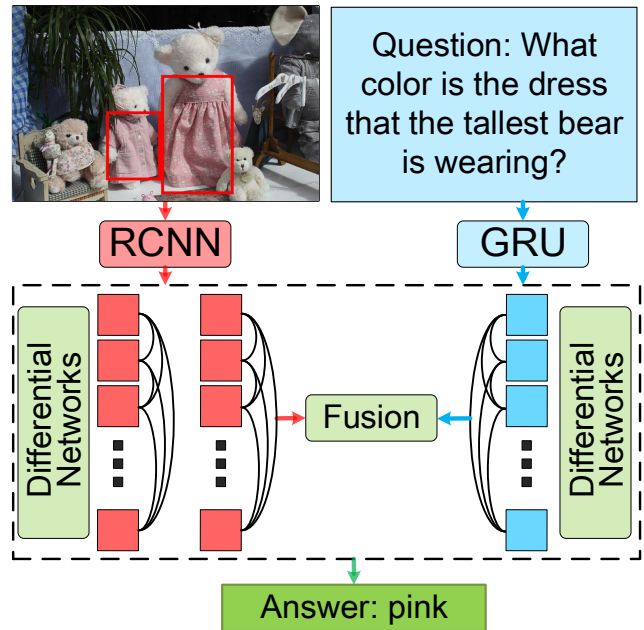


Figure 1: DN-based Fusion (DF) model for Visual Question Answering. The image and question feature elements are fed into the Differential Networks (DN) and then fused to derive the answer.

Li and Jia 2016; Nam, Ha, and Kim 2017). However, simple linear operations cannot bring fine-grained fusions between feature elements. Recently, bilinear models are used to model the fusion between image and question feature elements (Fukui et al. 2016; Kim et al. 2017; Yu et al. 2017; Ben-younes et al. 2017).

Note that the fusion in both linear and bilinear models are between the feature elements themselves, such as  $v_i q_i$  for linear models and  $v_i q_j$  for bilinear models. However, for the VQA problem, we argue that the fusion based on the difference between feature elements, like  $(v_i - v_j)(q_i - q_j)$ , would be more reasonable. This can be seen from two perspectives. For one thing, since  $v_i$  and  $q_i$  are from different modalities, it is unreasonable to fuse them directly. By contrast, after the difference operation, information from both modalities goes

to differential representations, and the fusion between them becomes more relevant. For another, the difference operation can reduce potential observation noises in  $v_i$  and  $q_i$ . For example, assuming that  $\delta$  is the noise of the feature elements  $v_i$  and  $v_j$ , then  $(v_i + \delta) - (v_j + \delta)$  can effectively filter the noise.

In this paper, we propose Differential Networks (DN), a novel plug-and-play module which enables differences between pair-wise feature elements. DN shows an interesting relation with Fully-Connected Networks (FCN). Then, with the tool of our DN, a new DN-based Fusion (DF) model for VQA task is constructed. Intuitively, we show the flowchart of our model in Fig. 1. Our DF model first makes differences on image and text feature elements respectively, and then fuse the differential representations to infer the final answer.

In summary, our contributions are as follows:

- We propose Differential Networks (DN), a novel plug-and-play module which enables differences between pair-wise feature elements;
- We propose DN based Fusion (DF), a novel model for VQA task;
- We achieve state-of-the-art results on all four datasets and conduct detailed ablation study to verify the effectiveness of differential networks.

## 2 Related Work

In this section, we introduce the related work in three consecutive sections. Firstly, we briefly review the current models for VQA task. Secondly, we focus on reviewing attention-based VQA models. Thirdly, we further review the classical fusion strategies used in these attention models.

### 2.1 Visual Question Answering (VQA)

Initially, Bayesian-based models were proposed to solve the VQA task (Malinowski and Fritz 2014; Kafle and Kanan 2016). With the success of deep neural networks, many different methods have been proposed to address the VQA task. These methods can be divided into five categories: attention-based models (Yang et al. 2016; Fukui et al. 2016), memory-based models (Xiong, Merity, and Socher 2016; Su et al. 2018), module-based models (Andreas et al. 2016; Hu et al. 2017), relation-based models (Santoro et al. 2017) and knowledge-based models (Wu et al. 2016; Wang et al. 2017). Among them, attention-based models, which select the useful part of input information with attention mechanisms, achieve significant performances on the VQA task.

In this paper, we employ the attention-based framework to validate the effectiveness of differential networks.

### 2.2 Attention-based VQA models

Due to their superior performances, attention-based VQA models have been received the most extensive studies. They focus on locating relevant objects in input features, such as bounding boxes or image regions.

Initially, (Chen et al. 2015) proposed one-step attention to locate relevant objects of images. Furthermore, (Yang

et al. 2016; Xu and Saenko 2016) proposed multi-step attention to update relevant objects of images and infer the answer progressively. Additionally, (Lu et al. 2016; Schwartz, Schwing, and Hazan 2017) proposed multi-modal attention, which finds not only the relevant objects of images but also questions or answers. Recently, (Fukui et al. 2016; Kim et al. 2017; Yu et al. 2017; Ben-younes et al. 2017) used bilinear fusion in attention mechanism to locate more accurate objects of input features.

In this paper, to validate the effectiveness of differential networks, we only use a simple one-step visual attention.

### 2.3 Fusion for Attention Mechanism

Attention mechanisms require fusion to calculate attention distributions. Therefore, the degree of fusion has a high impact on the quality of attention mechanism.

Existing attention models focusing on fusion can be divided into two categories, linear models and bilinear models. Initially, linear models are adopted to fuse image and question feature elements. (Yang et al. 2016; Lu et al. 2016) used the element-wise sum to fuse image and question feature elements, (Li and Jia 2016; Nam, Ha, and Kim 2017) used the element-wise multiplication to fuse image and question feature elements. Recently, bilinear models were used to model more fine-grained fusion between image feature and question feature elements. (Fukui et al. 2016) used the outer product to fuse image and question feature elements but resulting in the problem of dimension explosion. To solve the problem, (Kim et al. 2017) used the element-wise multiplication after the low-rank projection of image and question features. To further approximate bilinearity, (Yu et al. 2017; Ben-younes et al. 2017) used the  $k$ -calculated sum and sum-pooling of window  $k$  respectively to increase the model capacity.

In this paper, we first propose DN module, which explicitly models the differences between pair-wise feature elements. Then we propose DF model to fuse the differential representations.

## 3 Differential Networks

### 3.1 Definition and Derivation

Let  $\mathbf{x} = (x_1, x_2, \dots, x_m)^T$  be a feature vector. We perform full pairwise differences between feature elements mapping  $\mathbf{x}$  to a new vector  $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$ . Each element in  $\mathbf{y}$  is calculated in Eq. (1):

$$y_k = \sum_{i,j} (x_i - x_j) w_{ij}^{(k)}, \quad (1)$$

where  $w_{ij}^{(k)} \in \mathbb{R}$  is a learnable parameter,  $i, j \in [1, m], k \in [1, n]$ . Eq. (1) can be written in the form of a matrix, as denoted in Eq. (2):

$$y_k = \mathbf{x}^T W^{(k)} \mathbf{1} - \mathbf{x}^T (W^{(k)})^T \mathbf{1}, \quad (2)$$

where  $W^{(k)} \in \mathbb{R}^{m \times m}$  is the learnable parameter.  $\mathbf{1} \in \mathbb{R}^m$  is an all-ones vector.

Unfortunately,  $W \in \mathbb{R}^{m \times m \times n}$  is a third-order tensor, which makes the parameter scales large and difficult to train.

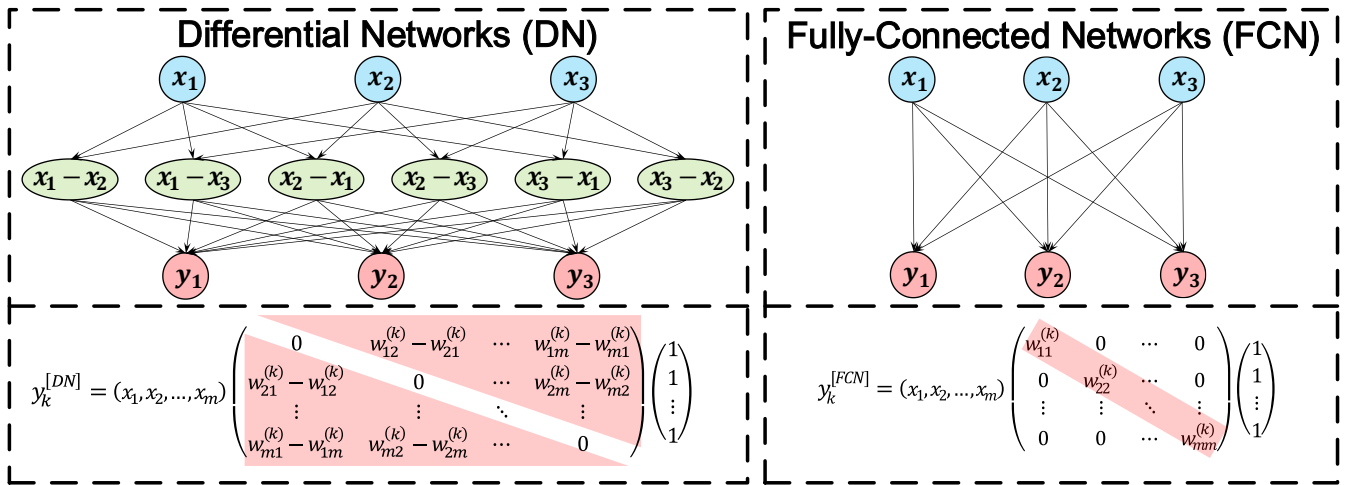


Figure 2: Comparison of Differential Networks (DN) and Fully-Connected Networks (FCN).

Inspired by Mutan (Ben-younes et al. 2017), we factorize  $W^{(k)}$  as a sum of  $S$  low-rank matrices:

$$W^{(k)} = \sum_{s=1}^S u_s^{(k)} \otimes v_s^{(k)}, \quad (3)$$

where  $u_s^{(k)} \in \mathbb{R}^m$  and  $v_s^{(k)} \in \mathbb{R}^m$  are learnable parameters,  $S$  is a hyper-parameter,  $\otimes$  represents the outer product. We substitute Eq. (3) into Eq. (2), then we have Eq. (4):

$$y_k = \sum_{s=1}^S [(\mathbf{x}^\top u_s^{(k)})(\mathbf{1}^\top v_s^{(k)}) - (\mathbf{x}^\top v_s^{(k)})(\mathbf{1}^\top u_s^{(k)})]. \quad (4)$$

Finally, Eq. (4) can be written in the form of a matrix:

$$\mathbf{y} = \sum_{s=1}^S [(\mathbf{x}^\top U_s) \odot (\mathbf{1}^\top V_s) - (\mathbf{x}^\top V_s) \odot (\mathbf{1}^\top U_s)], \quad (5)$$

where  $U_s \in \mathbb{R}^{m \times n}$ ,  $V_s \in \mathbb{R}^{m \times n}$  are learnable parameters,  $\mathbf{y} \in \mathbb{R}^n$ .  $\odot$  is the element-wise multiplication.

We call the mapping from  $\mathbf{x}$  to  $\mathbf{y}$  Differential Networks (DN) and denote Eq. (5) by Eq. (6) for simplification:

$$\mathbf{y} = DN(\mathbf{x}). \quad (6)$$

### 3.2 DN vs. FCN

In this subsection, we compare Differential Networks (DN) with Fully-Connected Networks (FCN), which is a commonly used neural network mapping  $\mathbf{x}$  to  $\mathbf{y}$  too.

For convenience of comparison, we rewrite DN as Eq. (7). It can be also rewritten as an equation in the lower left part of Fig. 2, and illustrated in the upper left part of Fig. 2. FCN can be written as Eq. (8) and rewritten as an equation in the lower right part of Fig. 2, and can be illustrated in the upper right part of Fig. 2.

$$y_k^{[DN]} = \sum_{i,j} (x_i - x_j) w_{ij}^{(k)} = \sum_{i,j} x_i (w_{ij}^{(k)} - w_{ji}^{(k)}) \quad (7)$$

$$y_k^{[FCN]} = \sum_i x_i w_{ik} = \sum_i x_i w_{ii}^{(k)}, \quad (8)$$

where in Eq. (8) we view the parameter matrix of size  $m \times n$  in fully connected networks as  $n$  diagonal matrices of size  $m \times m$ .

Comparing the illustrations of DN and FCN in the upper part of Fig. 2, we can see that the biggest difference between DN and FCN is the differential layer (light green) in the middle of DN. By the full difference between feature elements, the differential layer can effectively reduce the noise of the input information.

Comparing the formulas of DN and FCN in the lower part of Fig. 2, we have a very interesting finding. FCN has only weights for the diagonal, and the rest is zero. On the contrary, the diagonal of DN is 0, and the rest has weights. This shows that FCN focuses on the feature element itself while DN focuses on the interaction between feature elements.

## 4 Differential Fusion Model for VQA

Our DN based Fusion (DF) model is illustrated in Fig. 3. The model includes three parts. Data embedding encodes images and questions respectively. Differential Fusion is the major part of the model, which implements DN based fusion. Decision making predicts final scores of answers.

### 4.1 Data Embedding

Faster-RCNN (Ren et al. 2015) is used to encode images with the static features provided by bottom-up-attention (Anderson et al. 2018), GRU (Cho et al. 2014) is used to encode text with the parameters initialized with skip-thoughts (Kiros et al. 2015), as denoted in Eq. (9):

$$V = RCNN(image), \quad Q = GRU(question), \quad (9)$$

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

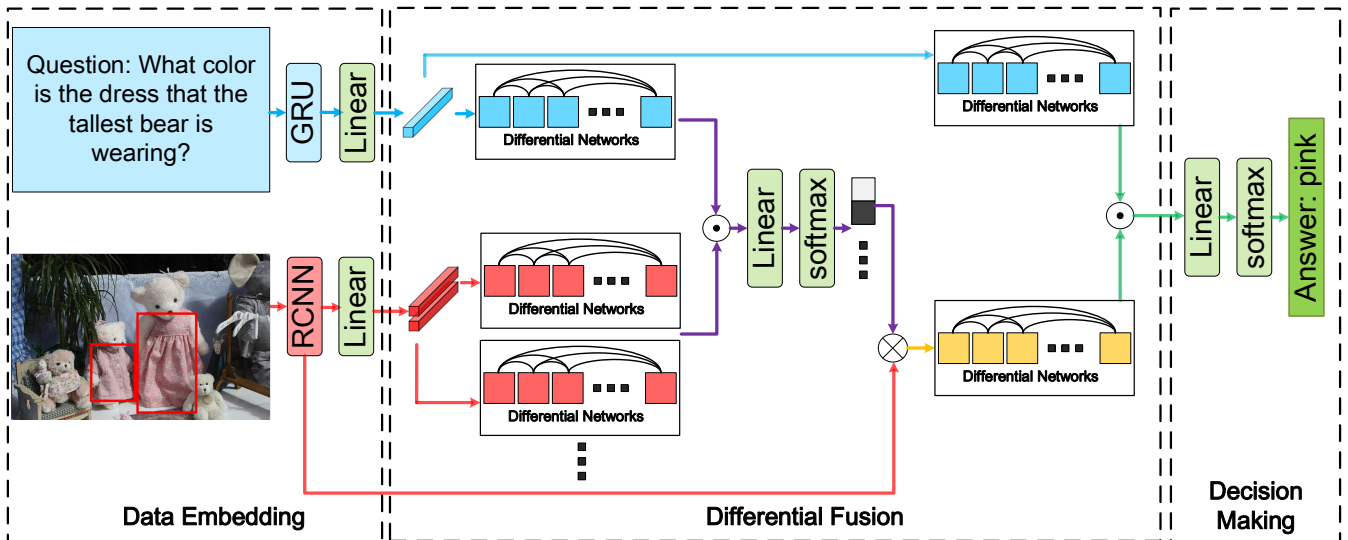


Figure 3: The overall structure of the proposed model for solving the VQA task. It consists of Data Embedding, Differential Fusion and Decision Making, marked with dash lines respectively.

Then, the visual features and question features are projected to the same dimension, as denoted in Eq. (10):

$$V^f = \text{relu}(VW_v), \quad Q^f = \text{relu}(QW_q), \quad (10)$$

where  $W_f$  and  $W_q$  are learnable parameters.  $V^f \in \mathbb{R}^{l \times d}$  and  $Q^f \in \mathbb{R}^d$  are projected features. We omit the bias  $b$  for simplicity.

## 4.2 Differential Fusion

Based on DN discussed in Sec. 3, we propose differential fusion, as denoted in Eq. (11):

$$H = \sum_{r=1}^R DN^r(V^f) \odot DN^r(Q^f), \quad (11)$$

where  $\odot$  is the element-wise operation,  $R$  is a hyperparameter,  $DN^r$  means the  $r$ th DN with different learnable parameters.  $H \in \mathbb{R}^{l \times d_h}$  is the result of the fusion. Then, multi glimpse attention distributions are calculated in Eq. (12).

$$\alpha = \text{softmax}(HW_h), \quad (12)$$

where  $W_h \in \mathbb{R}^{d_h \times g}$  is a learnable parameter,  $g$  is the number of attention glimpses. Note that here softmax is performed on the first dimension of matrix  $HW_h \in \mathbb{R}^{l \times g}$ .  $\alpha \in \mathbb{R}^{l \times g}$  is a matrix representing  $g$  attention distributions. Then, these attentions are used to attend the visual features, as denoted in Eq. (13):

$$\tilde{V} = \text{flatten}(\text{relu}(\alpha^T V W_\alpha)), \quad (13)$$

where  $W_\alpha \in \mathbb{R}^{d_v \times \frac{d'}{g}}$  is a learnable parameter,  $\text{relu}(\alpha^T V W_\alpha) \in \mathbb{R}^{g \times \frac{d'}{g}}$ ,  $\tilde{V} \in \mathbb{R}^{d'}$ . Then, the decision maker interacts  $\tilde{V}$  and  $Q^f$  again as denoted in Eq. (14):

$$F = \sum_{r=1}^R DN^r(\tilde{V}) \odot DN^r(Q^f), \quad (14)$$

where  $F \in \mathbb{R}^{d_f}$  is the result of the fusion.

## 4.3 Decision Making

In Decision Making process, a linear layer with a softmax activation function is used to predict the score of the candidate answer in Eq. (15):

$$\hat{a} = \text{softmax}(FW_f), \quad (15)$$

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

## 4.4 Training

We first calculate the ground-truth answer distribution in Eq. (16):

$$a_i = \frac{\sum_{j=1}^N \mathbb{1}\{u_j = i\}}{N - \sum_{j=1}^N \mathbb{1}\{u_j \notin \mathcal{D}\}}, \quad (16)$$

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

Finally, we use the KL-divergence as the loss function between  $a$  and  $\hat{a}$  in Eq. (17):

$$\mathcal{L}(\hat{a}, a) = \sum_{i=1}^{|\mathcal{D}|} a_i \log \left( \frac{a_i}{\hat{a}_i} \right). \quad (17)$$

## 5 Experiments

### 5.1 Datasets and evaluation metrics

We evaluate our model on four public datasets: the VQA 1.0 dataset (Antol et al. 2015), the VQA 2.0 dataset (Goyal et al. 2017), the COCO-QA dataset (Ren, Kiros, and Zemel 2015), and the TDIUC dataset (Kafle and Kanan 2017a). The VQA

Method	VQA 1.0 Test-dev					VQA 1.0 Test-std				
	Open-Ended				MC	Open-Ended				MC
	All	Y/N	Num.	Other	All	All	Y/N	Num.	Other	All
HighOrderAtt (Schwartz, Schwing, and Hazan 2017)	-	-	-	-	69.4	-	-	-	-	69.3
MLB(7) (Kim et al. 2017)	66.77	84.54	39.21	57.81	-	66.89	84.61	39.07	57.79	-
Mutan(5) (Ben-younes et al. 2017)	67.42	85.14	39.81	58.52	-	67.36	84.91	39.79	58.35	-
DualMFA (Lu et al. 2018)	66.01	83.59	40.18	56.84	70.04	66.09	83.37	40.39	56.89	69.97
ReasonNet (Ilievski and Feng 2017)	-	-	-	-	-	67.9	84.0	38.7	<b>60.4</b>	-
DF (36boxes) (ours)	<b>68.62</b>	<b>86.08</b>	<b>43.52</b>	<b>59.38</b>	<b>73.31</b>	<b>68.48</b>	<b>85.81</b>	<b>42.87</b>	59.23	<b>73.05</b>

Table 1: Comparison with the state-of-the-arts on the VQA 1.0 dataset.

Method	VQA 2.0 Test-dev				VQA 2.0 Test-std			
	All	Y/N	Num.	Other	All	Y/N	Num.	Other
MF-SIG-VG (Zhu et al. 2017)	64.73	81.29	42.99	55.55	-	-	-	-
Up-Down (36 boxes) (Teney et al. 2018)	65.32	81.82	44.21	56.05	65.67	82.20	43.90	56.26
LC_Baseline (100 boxes) (Zhang, Hare, and Prügel-Bennett 2018)	67.50	82.98	46.88	58.99	67.78	83.21	46.60	59.20
LC_Counting (100 boxes) (Zhang, Hare, and Prügel-Bennett 2018)	68.09	83.14	<b>51.62</b>	58.97	68.41	83.56	<b>51.39</b>	59.11
DF (36 boxes) (ours)	67.73	83.91	46.7	58.7	67.86	84.1	46.15	58.7
DF (100 boxes) (ours)	<b>68.31</b>	<b>84.33</b>	48.2	<b>59.22</b>	<b>68.59</b>	<b>84.56</b>	47.1	<b>59.61</b>

Table 2: Comparison with the state-of-the-arts on the VQA 2.0 dataset.

Method	All	Obj.	Num.	Color	Loc.	WUPS0.9	WUPS0.0
QRU (Li and Jia 2016)	62.50	65.06	46.90	60.50	56.99	72.58	91.62
HieCoAtt (Lu et al. 2016)	65.4	68.0	51.0	62.9	58.8	75.1	92.0
Dual-MFA (Lu et al. 2018)	66.49	68.86	51.32	65.89	58.92	76.15	92.29
DF (36 boxes) (ours)	<b>69.36</b>	<b>70.53</b>	<b>54.92</b>	<b>73.67</b>	<b>61.22</b>	<b>78.25</b>	<b>92.99</b>

Table 3: Comparison with the state-of-the-arts on the COCO-QA dataset.

Question Type	MCB-A (Fukui et al. 2016)	RAU (Kafle and Kanan 2017a)	CATL-QTA-W (Shi et al. 2018)	DF (36 boxes) (ours)
Scenen Recognition	93.06	93.96	93.80	<b>94.47</b>
Sport Recognition	92.77	93.47	95.55	<b>95.90</b>
Color Attributes	68.54	66.86	60.16	<b>74.47</b>
Other Attributes	56.72	56.49	54.36	<b>60.82</b>
Activity Recognition	52.35	51.60	60.10	<b>62.01</b>
Positional Reasoning	35.40	35.26	34.71	<b>40.76</b>
Sub. Object Recognition	85.54	86.11	86.98	<b>88.71</b>
Absurd	84.82	96.08	<b>100.00</b>	94.56
Utility and Affordances	35.09	31.58	31.48	<b>41.52</b>
Object Presence	93.64	94.38	94.55	<b>95.58</b>
Counting	51.01	48.43	53.25	<b>58.37</b>
Sentiment Understanding	66.25	60.09	64.38	<b>68.77</b>
Overall(Arithmetic MPT)	67.90	67.81	69.11	<b>72.97</b>
Overall(Harmonic MPT)	60.47	59.00	60.08	<b>65.79</b>
Overall Accuracy	81.86	84.26	85.03	<b>86.73</b>

Table 4: Comparison with the state-of-the-arts on the TDIUC dataset.

1.0 dataset contains a total of 614,163 samples and 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). The VQA 2.0 dataset contains a total of 1,105,904 samples and is divided into three splits: train(40.1%), val(19.4%), test(40.5%). It is more balanced compared to the VQA 1.0 dataset. The COCO-QA dataset contains a total of 117,684 samples and is divided into two splits: train(66.9%), test(33.1%). The TDIUC dataset contains a total of 1,654,167 samples and is divided into two splits: train(67.4%), test(32.6%). For the VQA 1.0 and VQA 2.0 dataset, we use the evaluation tool proposed in (Antol et al. 2015) to evaluate the model, as denoted in Eq. (18):

$$Acc(ans) = \min \left\{ \frac{\#\text{humans that said } ans}{3}, 1 \right\}. \quad (18)$$

For the COCO-QA dataset and TDIUC dataset, we evaluate the model in Eq. (19):

$$Acc(ans) = \mathbb{1}\{ans = \text{ground\_truth}\}. \quad (19)$$

## 5.2 Implementation details

During the data embedding phase, the image features are mapped to the size of  $36 \times 2048$  and the text features are mapped to the size of 2400. In the differential fusion phase, the number of hidden layer in DF is 510; hyperparameter  $S$  is 1,  $R$  is 5. The attention hidden unit number is 620. In the decision making phase, the number of hidden layer in DF is 510. All the nonlinear layers of the model all use the relu activation function and dropout (Srivastava et al. 2014) to prevent overfitting. All settings are commonly used in previous work. We implement the model using Pytorch. We use Adam (Kingma and Ba 2014) to train the model with a learning rate of  $10^{-4}$  and a batch size of 128. More details, including source codes, will be published in the near future.

## 5.3 Comparisons with state-of-the-arts

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

Firstly, Tab. 1 shows the comparison with the state-of-the-art models on the VQA 1.0 dataset. DF achieves new state-of-the-art results in both Multiple-Choice (MC) task and Open-Ended (OE) task. Using a single image feature, DF not only outperforms all the models that use single image feature but also outperforms ReasonNet (Ilievski and Feng 2017), which uses six input image features including face analysis, object classification, scene classification and so on. Especially, there is an improvement of 3.08% of MC task in test-std set.

Secondly, Tab. 2 shows the comparison with the state-of-the-art models on the VQA 2.0 dataset. Compared with Up-Down (36boxes) (Teney et al. 2018), which is the winning model in the VQA challenge 2017, DF (36boxes) achieves

Method	Validation
MLB	62.91
Mutan	63.61
DF with $\sum_{r=1}^R V^f W_{vf}^r \odot DN^r(Q^f)$	64.46
DF with $\sum_{r=1}^R DN^r(V^f) \odot Q^f W_{qf}^r$	64.58
DF without dropout	61.05
DF with tanh	64.78
<b>DF</b>	<b>64.89</b>

Table 5: Ablation study on the VQA 2.0 Validation.

2.19% higher accuracy. Compared with the most recent state-of-the-art model LC\_counting (100boxes) (Zhang, Hare, and Prügel-Bennett 2018), our single DF (100boxes) model achieves a new state-of-the-art result of 68.59% in the test-std set.

Thirdly, Tab. 3 shows the comparison with the state-of-the-art models on the COCO-QA dataset. DF improves the overall accuracy of the state-of-the-art Dual-MFA (Lu et al. 2018) from 66.49% to 69.36%. There is an improvement in accuracy of 3.6% in “Num.” question and 7.78% in “Color” question.

Fourthly, Tab. 4 shows the comparison with the state-of-the-art models on the TDIUC dataset. DF improves the overall accuracy of the state-of-the-art CATL-QTA-W (Shi et al. 2018) from 85.03% to 86.73%. There is also an improvement of 5.12% in “Counting” question and 5.93% in “Color Attributes” question.

In summary, the strong capability of the DF is shown in all four datasets.

## 5.4 Ablation study

In this section, we conduct some ablation studies. For a fair comparison, all the data provided in this section are trained under the VQA 2.0 training set and tested on the VQA 2.0 validation set. All the models use the exact same bottom-up-attention feature (36 boxes) extracted from faster-rcnn.

The first part of Tab. 5 compares DF with other attention models. Mutan can be viewed as the fusion between two FCNs, i.e.  $\sum_{r=1}^R V^f W_{vf}^r \odot Q^f W_{qf}^r$ . DF outperforms Mutan by 1.28%. This shows the effectiveness of the DN.

To further validate the effectiveness of DN, the models in the second part of Tab. 5 mix FCN and DN. DF-Q means using DN only for questions. In detail, DF-Q replaces Eq. (11) with  $H = \sum_{r=1}^R V^f W_{vf}^r \odot DN^r(Q^f)$  and replaces Eq. (14) with  $F = \sum_{r=1}^R \tilde{W}_v^r \odot DN^r(Q^f)$ . Similarly, DF-V means using DN only for images. As we can see, compared with DF, both DF-Q and DF-V lower the performance (64.89→64.46/64.58).

The third part of the Tab. 5 studies some tips and tricks. We find that using tanh does not affect performance much, but using relu does perform better. In addition, we find that using dropout is crucial — not using dropout will significantly lower the performance (64.89→61.05).

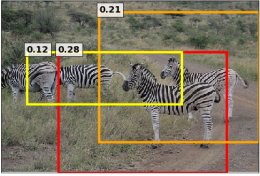
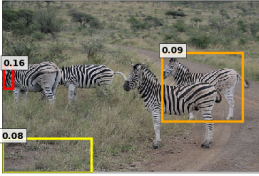
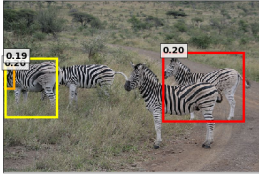
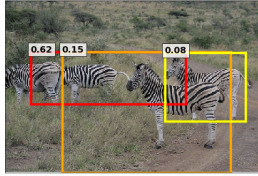
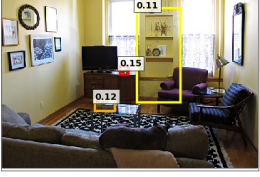
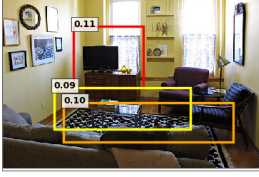
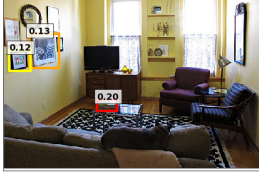


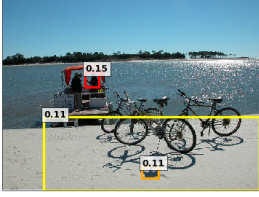




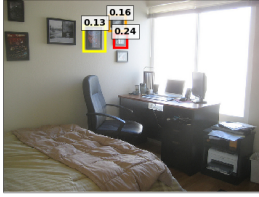
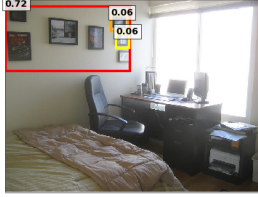
Mutan	DF-Q	DF-V	DF
<b>Example 1</b>	Question: How many zebras?		Ground-truth: 4
			
3✗	3✗	4✓	4✓
<b>Example 2</b>	Question: What is in front the window on a stand?		Ground-truth: tv
			
table ✗	tv ✓	books ✗	tv ✓
<b>Example 3</b>	Question: Are there people in the water?		Ground-truth: no
			
yes ✗	yes ✗	yes ✗	no ✓
<b>Example 4</b>	Question: How many frames are on the wall?		Ground-truth: 7
			
6 ✗	4 ✗	6 ✗	6 ✗

Figure 4: Visualization of DF and its comparative models.

### 5.5 Qualitative evaluation

In this section, we visualize some results of DF model and its comparative models in Fig. 4. Four examples are given including three success cases and one failure case of DF model. Each example compares the visualization of attentions of four models: Mutan, DF-Q, DF-V, and DF. The probability value of the attention is shown in the box upper left of each bounding box. For example, in Example 1, although both DF and DF-Q answered correctly, the bounding box for DF is more accurate and has a high attention probability of 0.62. As we can see from Example 1~3, whether DF-Q or DF-V is wrong or they are both wrong, DF still answers correctly. This shows that the difference operation plays an important role. In Example 4, all four models answered incorrectly. This shows that counting is still a challenge for attention-based models. Even so, the DF model

still boxes all the frames on the wall with a high attention probability of 0.72. This shows that by reducing the noise of the input features and map the image and the question to the same differential space, DF efficiently improves the attention accuracy and confidence.

## 6 Conclusion

In this paper, we propose a general DN module. Based on DN, we propose a new DF model for VQA task. We achieve state-of-the-art results on four publicly available datasets. In the future, we plan to use DN for other tasks and validate its generality and effectiveness.

## 7 Acknowledgments

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

## References

- Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, 6.
- Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural module networks. In *CVPR*, 39–48.
- Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. Vqa: Visual question answering. In *ICCV*, 2425–2433.
- Ben-younes, H.; Cadene, R.; Cord, M.; and Thome, N. 2017. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *ICCV*, 2631–2639.
- Chen, K.; Wang, J.; Chen, L.-C.; Gao, H.; Xu, W.; and Nevatia, R. 2015. ABC-CNN: An attention based convolutional neural network for visual question answering. *arXiv:1511.05960*.
- Cho, K.; van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv preprint arXiv:1409.1259*.
- Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 457–468.
- Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, volume 1, 9.
- Hu, R.; Andreas, J.; Rohrbach, M.; Darrell, T.; and Saenko, K. 2017. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*.
- Ilievski, I., and Feng, J. 2017. Multimodal Learning and Reasoning for Visual Question Answering. In *NIPS*, 551–562.
- Kafle, K., and Kanan, C. 2016. Answer-type prediction for visual question answering. In *CVPR*, 4976–4984.
- Kafle, K., and Kanan, C. 2017a. An Analysis of Visual Question Answering Algorithms. In *ICCV*.
- Kafle, K., and Kanan, C. 2017b. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding* 163:3–20.
- Kim, J.-H.; On, K.-W.; Kim, J.; Ha, J.-W.; and Zhang, B.-T. 2017. Hadamard product for low-rank bilinear pooling. In *ICLR*.
- Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *NIPS*, 3294–3302.
- Li, R., and Jia, J. 2016. Visual question answering with question representation update (qru). In *NIPS*, 4655–4663.
- Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical Question-Image Co-Attention for Visual Question Answering. In *NIPS*.
- Lu, P.; Li, H.; Zhang, W.; Wang, J.; and Wang, X. 2018. Co-attending Free-form Regions and Detections with Multi-modal Multiplicative Feature Embedding for Visual Question Answering. In *AAAI*.
- Malinowski, M., and Fritz, M. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 1682–1690.
- Nam, H.; Ha, J.-W.; and Kim, J. 2017. Dual attention networks for multimodal reasoning and matching. In *CVPR*.
- Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 91–99.
- Ren, M.; Kiros, R.; and Zemel, R. 2015. Exploring models and data for image question answering. In *NIPS*, 2953–2961.
- Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T. 2017. A simple neural network module for relational reasoning. In *NIPS*.
- Schwartz, I.; Schwing, A. G.; and Hazan, T. 2017. High-Order Attention Models for Visual Question Answering. In *NIPS*.
- Shi, Y.; Furlanello, T.; Zha, S.; and Anandkumar, A. 2018. Question Type Guided Attention in Visual Question Answering. In *ECCV*.
- Su, Z.; Zhu, C.; Dong, Y.; Cai, D.; Chen, Y.; and Li, J. 2018. Learning Visual Knowledge Memory Networks for Visual Question Answering. In *CVPR*, 7736–7745.
- Teney, D.; Anderson, P.; He, X.; and van den Hengel, A. 2018. Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge. In *CVPR*.
- Wang, P.; Wu, Q.; Shen, C.; van den Hengel, A.; and Dick, A. 2017. Explicit knowledge-based reasoning for visual question answering. In *IJCAI*.
- Wu, Q.; Wang, P.; Shen, C.; van den Hengel, A.; and Dick, A. 2016. Ask Me Anything: Free-form Visual Question Answering Based on Knowledge from External Sources. In *CVPR*.
- Xiong, C.; Merity, S.; and Socher, R. 2016. Dynamic memory networks for visual and textual question answering. In *ICML*.
- Xu, H., and Saenko, K. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 451–466.
- Yang, Z.; He, X.; Gao, J.; Deng, L.; and Smola, A. 2016. Stacked attention networks for image question answering. In *CVPR*.
- Yu, Z.; Yu, J.; Fan, J.; and Tao, D. 2017. Multi-modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering. In *ICCV*.
- Zhang, Y.; Hare, J.; and Prügel-Bennett, A. 2018. Learning to Count Objects in Natural Images for Visual Question Answering. In *ICLR*.
- Zhu, C.; Zhao, Y.; Huang, S.; Tu, K.; and Ma, Y. 2017. Structured Attentions for Visual Question Answering. In *ICCV*.