Enhancing Visual Question Answering Using Dropout

Zhiwei Fang[1,2], Jing Liu[1,2], Yanyuan Qiao[2], Qu Tang[1], Yong Li[3], Hanqing Lu[1,2]
[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Business Growth BU, JD.com, China
{zhiwei.fang,jliu,luhq}@nlpr.ia.ac.cn;qiaoyanyuan16@mails.ucas.ac.cn;tangquu@gmail.com;liyong5@jd.com

ABSTRACT

Using dropout in Visual Question Answering (VQA) is a common practice to prevent overfitting. However, in multi-path networks, the current way to use dropout may cause two problems: the co-adaptations of neurons and the explosion of output variance. In this paper, we propose the *coherent dropout* and the *siamese dropout* to solve the two problems, respectively. Specifically, in coherent dropout, all relevant dropout layers in multiple paths are forced to work coherently to maximize the ability of preventing neuron co-adaptations. We show that the coherent dropout is simple in implementation but very effective to overcome overfitting. As for the explosion of output variance, we develop a siamese dropout mechanism to explicitly minimize the difference between the two output vectors produced from the same input data during training phase. Such mechanism can reduce the gap between training and inference phases and make the VQA model more robust. Extensive experiments are conducted to verify the effectiveness of coherent dropout and siamese dropout. And the results also show that our methods can bring additional improvements on the state-of-the-art VQA models.

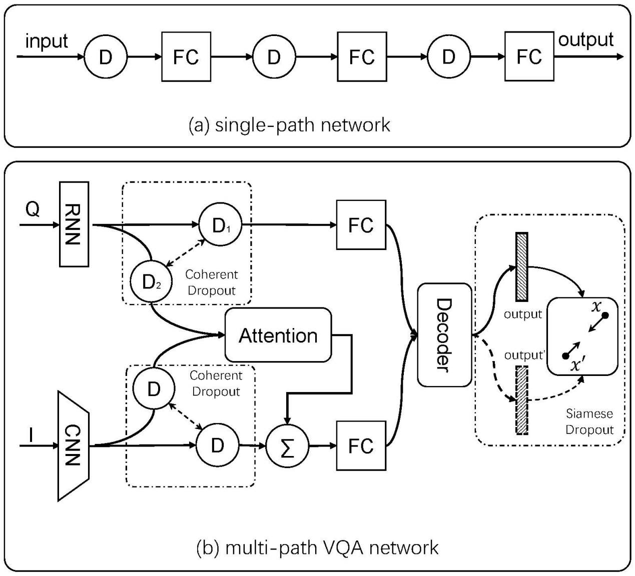
KEYWORDS

Visual Question Answering, Coherent Dropout, Siamese Dropout

ACM Reference Format:
Zhiwei Fang, Jing Liu, Yanyuan Qiao, Qu Tang, Yong Li, Hanqing Lu. 2018. Enhancing Visual Question Answering Using Dropout. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240662

1 INTRODUCTION

In VQA, a system is required to automatically generate natural language answers to totally free-form, open-ended textual questions about unconditional images. This is a challenging task since it needs not only the comprehensive understanding of both structural language information and non-structural image information but also learning semantic knowledge from them and reasoning among them. In recent few years, VQA has attracted a lot of attention: a number of benchmarks have been released for VQA [4, 7, 9, 10, 18, 23, 25, 32, 33, 38, 39] and a variety of methods have been proposed [2, 3, 5, 8, 16, 22, 24–26, 31, 33, 35, 36]. Many of the state-of-the-art methods usually adopt a similar architecture: a convolutional neural network (CNN) [11, 19, 27, 30] for image encoding; a recurrent neural network (RNN) [6, 13, 14] for question encoding; a module for multi-modal feature fusion and a decoder for answer prediction [34]. Since attention mechanism is essential in VQA, the attention module is also often included in the architecture. Thus in VQA model, there are many cases of feature map reuse and there usually exist multiple paths between input and output. Such multi-path architecture is the current main stream in VQA.



(a) single-path network

(b) multi-path VQA network

Figure 1: Single-path network and multi-path network. Ⓓ denotes dropout layer and *FC* denotes fully-connected layer. In multi-path VQA model, we propose the *corrent dropout* and *siamese dropout* to improve the final performance. In coherent dropout, the dropout masks of different layers are forced the be the same. In siamese dropout, two outputs are produced by a siamese model with dropout layers and the difference between them is minimized by the siamese dropout loss.

VQA model is heavily data-dependent [1, 10, 15], and the deep-learning-based models usually suffer serious overfitting. In order to alleviate this problem, many researchers introduce dropout [28] technique into their model to improve the generalization performance [2, 5, 8, 16, 22, 35, 36]. In these methods, the dropout is used

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

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240662

in a simple way: where a feature is expected to be used, a dropout layer is placed there first. This results in that there usually exists a dropout layer in the front of each path, as is shown in Fig.1 (b). However, in multi-path networks, using dropout in such way may cause some problems.

The first problem is the co-adaptations. Co-adaptation of neurons is the inner reason to cause overfitting of neural networks [12, 28]. It indicates such a case where several neurons always depend on each other to fit a noise pattern. And dropout is proposed to prevent such complex co-adaptations on the training data [12]. On each presentation of each training case, the hidden units are randomly omitted from the network, so a hidden unit cannot rely on other hidden neurons. Generally speaking, in single-path feed-forward neural networks (Fig.1 (a)), dropout can work well because a neural layer (e.g. fully-connected layer) is connected to only one dropout layer, thus there can not be any conflicts between any two dropout layers. But in multi-path networks, the ability of dropout to prevent co-adaptations may be weakened or damaged, if the dropout layers are used in an unsuitable way. Take the reuse of question feature in VQA as an example (Fig.1 (b)). The question feature is used twice: one for visual spacial attention module and the other for answer decoder module, in two paths, respectively. In each path, the output of RNNs is firstly fed into a dropout layer D_1 or D_2. In general, D_1 and D_2 work independently and their outputs are different. Now we consider two neurons n_1 and n_2 in the output layer of RNNs. Suppose that in D_1, n_1 is omitted while in D_2, n_2 is omitted, then in each of the two paths, n_1 and n_2 can not depend on each other because only one of them has a nonzero gradient and it's impossible to optimize both of them at the same time. Since current dropout layer doesn't directly omit the neurons but their output features, D_1 and D_2 will back propagate the gradient of n_2 and n_1 to RNNs, respectively. If we take D_1 and D_2 together as a "black box" and see from the point of RNNs, we will find that none of n_1 and n_2 are omitted and they can still be optimized simultaneously. This will decrease the independence of neurons and increase the possibility of overfitting.

Another problem is the explosion of the output variance. Namely, for the same input, a model with dropout layers will produce different outputs in different time. This because the dropout masks are changing every time in each forward pass. For VQA model, the variance can enlarge the gap between training and inference phases and make the model sensitive to input noises [29]. For example, assume that the output variance is very high, then for a given input, the outputs of its two forward passes can be very different; if one output gives the correct answer and the other one should provide the wrong answer in a high probability. This is not what we want. Current methods usually assume that this problem can be implicitly handled if we force the outputs to fit a fixed target attached to the specific input question during training. But in VQA, such assumption may not be so effective because the ground-truth answer for given input image and question is not unique [4]. This results in that the target for a specific question during training is not fixed but sampled from a set of ground-truth labels [8, 16, 35]. Hence in VQA, using dropout can make the problem even serious.

In order to enhance the VQA model, in this paper, we propose the *coherent dropout* and the *siamese dropout* to solve the above two problems.

For co-adaptations of neurons, the reason is that in multi-path networks, the multiple dropout layers share the same input feature vector but work independently. This may cause a lot conflicts among the dropout masks of different dropout layers. In coherent dropout, we force the dropout layers work coherently to reduce the conflicts. Specifically, during training, the dropout layers which share the same input must use the same randomly sampled dropout masks. In such case, if a neuron is omitted in one layer, then it must be omitted in all the relevant dropout layers. This can ensure that in backward pass, there is no gradient for this neuron. The equivalent form of coherent dropout is placing the dropout layer in the root path instead of the branch path. In our experiments, we demonstrate that with just a little modification by converting common dropout to coherent dropout, the state-of-the-art methods can achieve better performances.

As for the explosion of output variance, the core idea behind *siamese dropout* is to explicitly minimize the distance of outputs. During training, we first keep another identical copy of the VQA model. The two models share the same weights and inputs in each iteration, but use different dropout masks. Then between their outputs, we append a new constrain which forces the distance between them to be as small as possible (See Fig.1 (b)). We show that the siamese dropout constrain can effectively decrease the variance of outputs caused by dropout and reduce the gap between training and inference phases.

Finally, with the proposed techniques on dropout, the VQA models can become more generalized and can go deeper in several modules, especially the question encoding module. Compared with CNN encoder for image, the question encoder (RNNs) is usually very shallow. Simply stacking multiple RNNs in question encoder often suffers serious overfitting. In this paper, with the help of residual connection and coherent dropout, we demonstrate that the power of question encoder can be improved by simply stacking more RNNs.

To summarize, the main contributions of this study is three-fold: Firstly, we proposed a simple but effective *coherent dropout* to improve the ability of dropout to prevent overfitting in VQA model. Secondly, we develop a *siamese dropout* mechanism to handle the high output variance of VQA model during training. Finally, based on the coherent dropout and residual contributions, we develop a deeper and more powerful question encoder by stacking multiple RNNs. We conduct extensive experiments to demonstrate the effectiveness of our methods. The results show that with the proposed techniques, the performances of current state-of-the-art methods such as BottomUp [2], MCB [8] and MUTAN [5] can be further improved.

2 RELATED WORK

2.1 Dropout

Dropout is thought to be an effective way to prevent overfitting, especially for deep neural networks. It is proposed by Hinton et al. in [12, 28]. The key idea is to randomly drop units from the neural network during training. They think this can prevent the co-adaptations of neurons. Co-adaptation is a concept from the theory of the role of sex in evolution [21] and the motivation of dropout is also from it [28]. Hinton et al. believe that such process of dropout

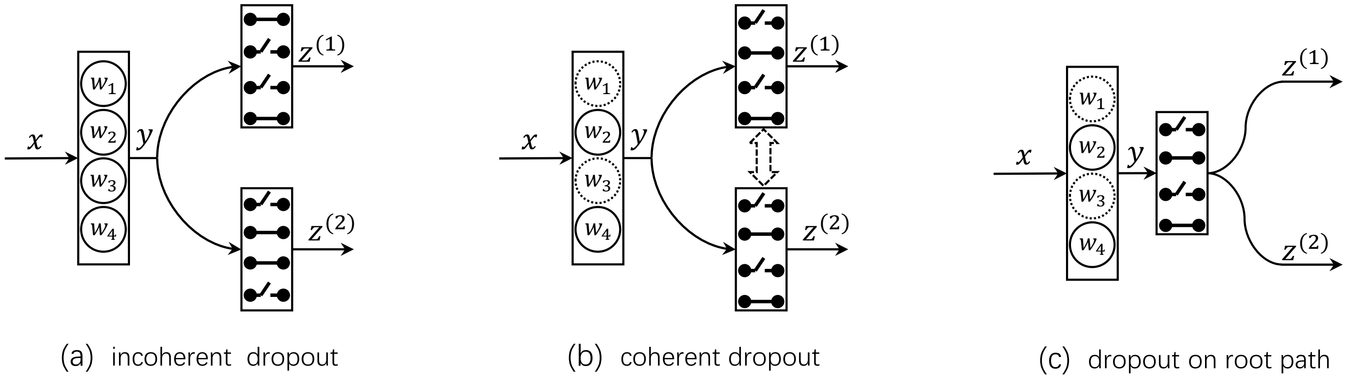


Figure 2: Structures of different dropouts. In incoherent dropout (a), outputs are dropped out while the neurons can still get gradient. In coherent dropout (b), all dropout layers share the same dropout mask and the dropping of neurons is totally controlled by the mask. (c) is the equivalent form of coherent dropout which is used in implementation.

can force the neurons learn something without the dependence on other neurons, which can be also explained as a kind of model ensemble. They demonstrate that using dropout in input layers and in hidden layers can both improve the neural model significantly. In this paper, we also follow such interpretation of the effect of dropout in [12, 28] to explain the motivation of our coherent dropout.

Now dropout has been widely used in various kinds of neural networks such convolutional neural networks [19, 27] and recurrent neural networks [37]. Current powerful CNNs are usually fully-convolutional networks [11, 30] without any fully-connected layer as hidden layer, thus they seldom use dropout. Some classical CNNs such as [19, 27] have several hidden fully-connected layers and need to use dropout to prevent overfitting. In these CNNs, since there are only one single path from input to output, the dropout layers can work independently and do not affect each other. However, in VQA model, it is very common that a feature is used by several modules and thus there are usually multiple paths with dropout, which may cause dropout can not work independently. This is the reason why we study dropout in VQA model in this paper.

2.2 Dropout in VQA

Dropout is widely used in VQA models to prevent overfitting [5, 8, 16, 22, 35, 36]. For example, MCB[8], Fukui et al. place a dropout layer on top of their Compact Bilinear Pooling (CBP) module to weak the side effects of the high-dimensional output of CBP. Besides, they also use dropout layers in and after the LSTM encoder. Similarly, Ben-younes et al. use dropout technique in the proposed Multimodal Tucker Fusion Layer of MUATN model [5]. In these models, the dropout is placed in a simple way: when one module needs to use a feature, then the model designer places a dropout layer before the module, which will cause the similar case in Fig.1 (b). There are also some methods that do not include any dropout layers, such as [2]. However, in [2], Anderson et al. have to adopt model ensemble to prevent overfitting while the number of models for ensemble is quite big – 30 models for best performance! As a contrast, MCB[8] uses 7 models for ensemble, MUTAN [5] uses 5. Different from the methods above, in this study, we demonstrate that if a VQA model introduces dropout in a coherent way, then it can achieve a higher performance. Coherent dropout is simple but effective.

3 APPROACH

In this section, we first describe the details of formulation of coherent dropout and siamese dropout. Then we discuss how to use the coherent dropout to design a deeper and more powerful question encoder.

3.1 Coherent Dropout

Assume that there is a hidden layer (e.g. a fully-connected layer) with a set of neurons $\boldsymbol{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_i, ..., \boldsymbol{w}_N\}$ where $\boldsymbol{w_i}$ is the weight vector of the i^{th} neuron. Without loss of generality, we assume that the bias term is contained in $\boldsymbol{w}_i$. Then the output $\boldsymbol{y}$ of the hidden layer is fed into multiple dropout layers in multiple paths. Let $l \in \{1, 2, ..., L\}$ index the dropout layers and $\boldsymbol{z}^{(l)}$ is the output vector of l^{th} dropout layer. Since the current dropout layer is implemented to drop out the output data instead of the neurons [28], then for a given input vector $\boldsymbol{x}$, $\boldsymbol{z}^{(l)}$ is computed by:

$$\begin{aligned} \boldsymbol{y} &= f(\boldsymbol{W}\boldsymbol{x}) \\ \boldsymbol{z}^{(l)} &= \boldsymbol{m}^{(l)} * \boldsymbol{y} \end{aligned} \tag{1}$$

where $*$ denotes the Hadamard product, f is any activation function such as *sigmoid* or *ReLU* and $\boldsymbol{m}^{(l)}$ is the dropout mask which is sampled from a bernoulli distribution $Ber(p)$. Generally speaking, when the number of neurons N is big, the probability of $\boldsymbol{m}^{(l)} = \boldsymbol{m}^{(n)}$ $(l, n \in L, l \neq n)$ is pretty close to 0. Then the i^{th} element of $\boldsymbol{z}^{(l)}$ can be described:

$$\begin{aligned} m_i^{(l)} &\sim Ber(p) \\ y_i &= f(\boldsymbol{w}_i \boldsymbol{x}) \\ z_i^{(l)} &= m_i^{(l)} y_i \end{aligned} \tag{2}$$

Now consider the co-adaptation of two neurons $\boldsymbol{w}_i, \boldsymbol{w}_j \in \boldsymbol{W}$. We know that if two neurons are no optimized simultaneously, then there is not co-adaptation between them. During the backward pass of neural networks, the gradient of $\boldsymbol{w}_i$ can be computed by:

$$\begin{aligned}\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{w_i}} &= \sum_{l=1}^{L}\left\{\frac{\partial \boldsymbol{J}}{\partial z_i^{(l)}}\frac{\partial z_i^{(l)}}{\partial y_i}\frac{\partial y_i}{\partial \boldsymbol{w_i}}\right\}\\ &= \frac{\partial y_i}{\partial \boldsymbol{w_i}}\sum_{l=1}^{L}\left\{\frac{\partial \boldsymbol{J}}{\partial z_i^{(l)}} m_i^{(l)}\right\}\end{aligned} \tag{3}$$

where $\boldsymbol{J}$ is the objective function. Firstly, from Equation 2, we can see that for single-path network (i.e., $L = 1$), dropping out output data is equivalent to dropping out neurons, because when $L = 1$, the gradient of i^{th} neuron is controlled by the dropout mask m_i. However, such equivalence relation does not exist in the case of multi-path networks. Equation 3 implies that even if there are no co-adaptations between two neurons in each of the dropout layer (i.e., there are at last one 0 in $\{m_i^{(l)}, m_j^{(l)}\}, \forall l$), it is also very possible that both of the two neurons can get non-zero gradient and thus be optimized simultaneously, When $L > 1$, if the i^{th} neuron is expected to be omitted, there must be $\sum_{l=1}^{L}\left\{\frac{\partial \boldsymbol{J}}{\partial z_i^{(l)}} m_i^{(l)}\right\} = 0$. Usually, such condition is hard to be met if the dropout layers work independently.

But in our coherent dropout (Fig.2(b)), the dropout layers do not work independently but share the same dropout mask during one forward pass, which can be described as:

$$\begin{aligned}\boldsymbol{m}^{(l)} &= \boldsymbol{m}^{(n)}, \forall l, n \in [1, L]\\ m_i^{(l)} &= m_i^{(n)}, \forall l, n \in [1, L], i \in [1, N]\end{aligned} \tag{4}$$

Let $\boldsymbol{m}^{(1)} = \ldots = \boldsymbol{m}^{(L)} = \boldsymbol{m}$, then the gradient of neuron $\boldsymbol{w}_i$ is as follow:

$$\begin{aligned}\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{w_i}} &= \frac{\partial y_i}{\partial \boldsymbol{w_i}}\sum_{l=1}^{L}\left\{\frac{\partial \boldsymbol{J}}{\partial z_i^{(l)}} m_i^{(l)}\right\}\\ &= m_i\frac{\partial y_i}{\partial \boldsymbol{w_i}}\sum_{l=1}^{L}\frac{\partial \boldsymbol{J}}{\partial z_i^{(l)}}\end{aligned} \tag{5}$$

Equation 5 implies that, in coherent dropout, the neuron's gradients are controlled by the dropout mask. In another word, coherent dropout is more efficient to prevent the complex co-adaptations in multi-path neural networks.

In VQA model, there are many places where coherent dropout can be adopted. For example, both question feature and image feature are at least used twice in attention module and feature-fusion module. In multi-glimpse attention mechanism, there also exists such case that a feature is used to product multiple attention maps. In [31], the mixed feature of textual and visual information is also used by multiple classifiers to predict the correct answer. And some non-linear layers such as the gated tanh layer [2] in VQA is also in multi-path structure. In Section 3.3, we further explore the way to adopt the coherent dropout into question encoder to make it deeper and more powerful.

In implementation, the coherent dropout is yet very simple. There is no need to actually control the masks of multiple dropout layers. The only thing you need to do is placing a dropout layer in root path and removing the ones in branch paths, which is shown in Fig.2 (c). In the following experiment section, we will demonstrate that with just such simple modification of network structure, the performance can be improved significantly.

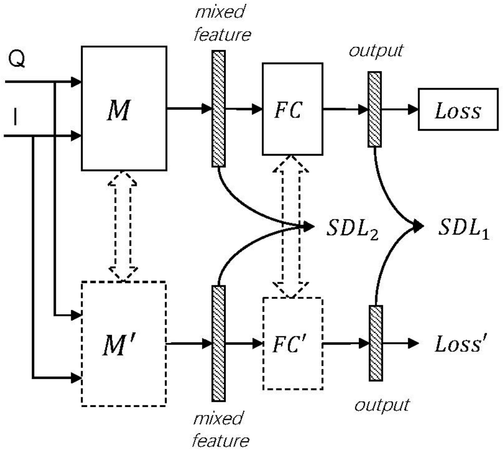


Figure 3: Structure of siamese dropout for VQA. M **and** M' **denotes the main bodies of two weight-shared VQA model.** FC **and** FC' **denotes two weight-shared fully-connected layers.** SDL **means the siamese dropout loss and it can be applied on different feature vector such as the output (**SDL_1**) or the mixed feature (**SDL_2**). Due to the existence of dropout, the outputs of the two models are not the same and the variance is controlled by our siamese dropout loss.**

3.2 Siamese Dropout

The output variance in VQA model can not be ignored. We propose a siamese dropout mechanism to explicitly control such variance during training. The structure of siamese dropout is shown in Fig.3. Consider we have a VQA model $\mathcal{M}_\theta$ that takes as input $(\boldsymbol{Q}, \boldsymbol{I})$ whose target label is $\boldsymbol{A}$, where θ denotes the parameters. Let $\ell(\mathcal{M}_\theta(\boldsymbol{Q}, \boldsymbol{I}), \boldsymbol{A})$ be the loss value for the sample triplet $(\boldsymbol{Q}, \boldsymbol{I}, \boldsymbol{A})$. During training, we keep an identical copy of original model $\mathcal{M}_\theta$ as $\mathcal{M}_\theta^{'}$, i.e., the two networks share the same parameters θ. For each of the two model, there is a loss for the output :

$$\begin{aligned}\boldsymbol{o} &= \mathcal{M}_\theta(\boldsymbol{Q}, \boldsymbol{I})\\ l &= \ell(\boldsymbol{o}, \boldsymbol{A})\\ \boldsymbol{o}' &= \mathcal{M}_\theta^{'}(\boldsymbol{Q}, \boldsymbol{I})\\ l' &= \ell(\boldsymbol{o}', \boldsymbol{A})\end{aligned} \tag{6}$$

where l, l' are the loss values of $\mathcal{M}_\theta, \mathcal{M}_\theta^{'}$, respectively. Then the overall loss function can be given:

$$L = \frac{1}{2}(l + l') + \gamma SDL(\boldsymbol{o}, \boldsymbol{o}') \tag{7}$$

where γ is the loss weight factor and $SDL(\boldsymbol{o}, \boldsymbol{o}')$ is the *siamese dropout* loss, which is given by:

$$SDL(\boldsymbol{o}, \boldsymbol{o}') = \frac{1}{n} \parallel \tanh(\boldsymbol{o}) - \tanh(\boldsymbol{o}') \parallel_2 \tag{8}$$

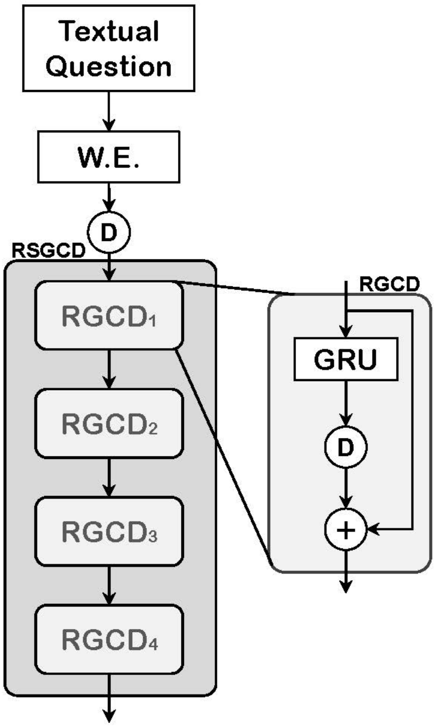


Figure 4: Residual Stacked GRUs with Coherent Dropout (RSGCD). Ⓓ denotes dropout layer. ⊕ denotes element-wise sum. W.E. stands for word embedding. Our question encoder of RSGCD consists of several modules of Residual GRU with Coherent Dropout (RGCD) which are stacked serially. The coherent dropout makes it robust to overfitting and the residual connections prevent gradient vanishing.

where n is the length of $\boldsymbol{o}$ and tanh is used to limit the scale of the output vector. Since the siamese dropout loss will always be 0 if there are no any dropout layers in the model, the aim of Equation 8 is to minimize the variance caused by dropout.

We also notice that the siamese dropout loss can be adopted not only on the final output vector, but also on some other feature maps produced by hidden layers. The motivation is that even though the input units of a hidden layer are randomly dropped out, the learned representation should also be stable. Let $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_K\}$ is a set of such suitable feature maps, then the overall loss function can be extended as:

$$L_{ext} = \frac{1}{2}(l + l') + \sum_{i=1}^{K}[\gamma_i SDL(\boldsymbol{v}_i, \boldsymbol{v}'_i)] \tag{9}$$

where γ_i is the loss weight factor for $SDL(\boldsymbol{v}_i, \boldsymbol{v}'_i)$.

Although the siamese dropout leads to the increase of computation (additional one forward pass), it can explicitly decrease the output variance caused by dropout. And its effect is significant. We demonstrate that the siamese dropout can effectively decrease the gap between training and inference phases and provide additional improvements to VQA model.

3.3 Stacked-Residual GRUs

In VQA, the amount of textual data is very limited. Although there may be hundreds of thousands of question-answer pairs in dataset, the length of questions are usually very small, typically less than 15, let alone some of the words are wh-phrases (e.g. what kind of, where is the, etc.) without clear semantics. Thus designing a powerful question encoder is very essential to visual question answering. In some other area such as Image Caption, stacking multiple RNNs is usually effective. However, in VQA, such stacking method is not significant, or even causes bad effect. The reason may be the overfitting of question encoder on the limited textual training data.

In this section, we will utilize our coherent dropout and residual connections to design a deeper stacked RNNs question encoder. We choose GRU as the implementation of RNN and design a deep GRU structure termed as *Residual Stacked GRUs with Coherent Dropout* (RSGCD) which is illustrated in Fig.4. RSGCD consists of several modules of *Residual GRU with Coherent Dropout* (RGCD) which are stacked one by one. The input of RSGCD is the output features of word embedding and then fed into the following RGCD modules serially. In each RGCD module, the input is first processed by a GRU and then a dropout layer. We use the sum of input and the output of dropout layer as the final result of a RGCD module. Note that the dropout layer in RGCD needs to be before the element-wise sum operation to make sure that it is coherent dropout. Placing dropout layer after the sum operation results in twice dropouts for input features, which is equivalent to a incoherent dropout. On the other hand, the residual connections are also essential for RSGCD because when we stack more GRUs, the problems of gradient vanishing becomes serious. The residual can act as a high-way for gradient propagation which connects GRU's input and output feature maps. In RGCD, the GRUs are expected to learn the *residual transformation* rather than common transformation. We demonstrate that with the coherent dropout, the RSGCD can reach 3 or more GRUs and bring more improvement to VQA models. We believe that the RSGCD is instructive for the using of coherent dropout in models with complex skip-connections.

4 EXPERIMENT

4.1 Dataset

VQA-v1. The VQA-v1 dataset [4] consists of ~200K images from the MS-COCO dataset [20] with approximately 3 questions per image and 10 answers per question. All questions are divided into three categories by the type of their answers: "Yes/No", "Number" and "Other" but their answers are not strictly limited by the question types (e.g., the answers for "Yes/No" questions may not have to be {yes, no}, the ground-truth may also be "I don't know" or some other words). There are three splits in VQA-v1: *train* (~248K questions), *val* (~122K questions) and *test* (~244K questions). The ground truth answers are available only for training and validation sets while the evaluation for testing set can be only done on server of the dataset.

VQA-v2. The VQA-v2 dataset [10] is the extended version of VQA-v1 dataset [4]. It doubles the number of questions and makes them more balanced. Namely, in order to improve the importance of

Table 1: The overall accuracies of models under different configuration about coherent dropout and siamese dropout. G-tanh denotes the gated tanh [2]. SDL denotes the siamese dropout loss.

	Method	Accuracy			
		Yes/No	Number	Other	Overall
A	Baseline	83.25	36.89	51.17	61.39
B	Baseline-Dropout	82.7	36.27	50.27	60.4
	Baseline+Coherent G-tanh	83.12	35.42	51.61	61.73
	Baseline+Coherent Dropout	**83.31**	**37.27**	**52.83**	**62.28**
C	Baseline+SDL@(output)	83.18	**37.22**	52.62	62.12
	Baseline+SDL@(mixed_feature)	83.03	36.88	52.78	62.11
	Baseline+SDL@(output+mixed_feature)	83.27	36.82	52.88	62.24
	Baseline+SDL@(output+mixed_feature+q_feature)	83.28	37.05	**52.88**	**62.27**
	Baseline+SDL@(output+mixed_feature+v_feature)	**83.32**	36.85	52.82	62.23
D	Baseline+Coherent Dropout+SDL@(output+mixed_feature)	**83.88**	**37.92**	**53.32**	**62.82**

visual images in VQA, each question in this dataset is asked about two images with two different answers. There are ~443K questions in *train* split, ~214K questions in *val* split and ~453K questions in *test* split.

Evaluation Metric. In evaluation, the algorithm is only allowed to product one natural language answer for one question. We use the accuracy metric proposed in [4] to evaluate the performances of models which can be described as:

$$min(1, \frac{\#humans\ that\ provided\ that\ answer}{3}) \tag{10}$$

In this metric, once the presented answer matches a ground truth answer, then it gets $\frac{1}{3}$. But the score can not be larger than 1 even if the matching times are more than 3.

4.2 Experimental Setup

Our baseline model structure is similar to that in [31] with two modifications: (1) Before each linear layer (including the ones in the gated tanh module [2]), we insert a dropout layer. (2) The hidden size of GRU is set to 300 rather than 512. When given a question and an image, they are first embedded into a question feature and a image feature map by RNNs and CNNs, respectively. Then the question feature and image feature are fed into an attention module to produce the visual attention. The attended image feature and question feature are merged by Hadamard product and produce the mixed_feature. Finally, the mixed_feature is used by the classifiers to predict the answers.

We use the Adamax [17] optimizer to optimize our VQA model. The learning rate is fixed as 7e-3 and no weight decay is used. We use a batch size of 128 to train the model for 100 epochs. All experiments (except for the ones in Table 4) are conducted on VQA-v1 dataset: the models are trained on *train* split and tested on *val* split. Before training, we collect the top 3000 frequent answers of *train* split as the answer vocabulary. In this paper, all dropout layers use the same dropout ratio of 0.5.

4.3 Experiments on Coherent Dropout

In order to demonstrate the effectiveness of the coherent dropout, we compare the model performances with/without our coherent dropout. The results are listed in A,B blocks of Table 1. In addition to the baseline configuration (denoted by **Baseline**), there are another three models:

- **Baseline - Dropout**: All dropout layers in **Baseline** are removed.
- **Baseline + Coherent G-tanh**: We move the dropout layers in the two branch paths of the gated tanh (G-tanh) [2] module to the root path, which is termed **Coherent G-tanh**.
- **Baseline + Coherent Dropout**: All dropout layers in baseline model are replaced by coherent dropout.

Firstly, we notice that without dropout, the **Baseline - Dropout** model suffers overfitting and is much poorer than **Baseline** model. This implies that dropout is effective on preventing overfitting. Secondly, when the dropout is partially (**Baseline + Coherent G-tanh**) and then fully **Baseline + Coherent Dropout** replaced by coherent dropout, the performance keeps increasing step by step. And the final accuracy of **Baseline + Coherent Dropout** is much higher than **Baseline** model, which proves that our coherent dropout can benefit VQA model a lot.

The similar conclusion can also be seen from the accuracy curves during training, which are illustrated in Fig.5. In the curves, **Baseline + Coherent Dropout** model outperforms all the other models while the **Baseline - Dropout** gets into overfitting at very beginning epochs. It is noteworthy that during epochs from 40 to 100, the performances of **Baseline** and **Baseline + Coherent G-tanh** start to decrease, while the accuracy of **Baseline + Coherent Dropout** is still at a high level and very stable. This implies that coherent dropout is more powerful to prevent overfitting in multi-path networks.

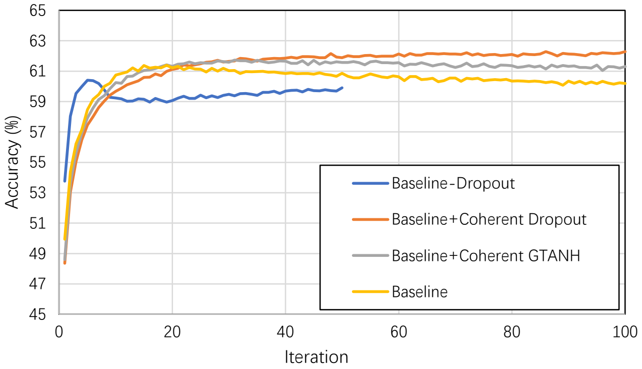


Figure 5: Comparison of the accuracies on validation set during training.

4.4 Experiments on Siamese Dropout

Table 1, block C lists the experimental results on siamese dropout. "SDL@($\boldsymbol{x}$)" stands for a siamese dropout loss (SDL) which is computed from the feature map $\boldsymbol{x}$. Here we choose 5 typical feature maps in VQA model:

- output: the final output probability vector.
- mixed_feature: the output of the multi-modal feature fusion module.
- q_feature: the question feature produced by question encoder, usually LSTM or GRU.
- v_feature: the image feature produced by CNNs.

Firstly, we evaluate the performance of SDL on single feature map. When compared with **Baseline** model, we can see that both SDL@(output) and SDL@(mixed_feature) perform better, which demonstrates the effectiveness of our siamese dropout. Then we test the model with both of the two siamese dropout losses and the results (SDL@(output+mixed_feature)) show that the accuracy can be further improved. However, the improvements of SDL@(output) and SDL@(mixed_feature) are partially superposed. This may be because both of the two siamese dropout losses are doing the same thing: reducing the output variance. We also notice that, when adding q_feature and v_feature to SDL, the increase is not significant. The reason may be due to that there are few dropout layers before those two feature maps.

Table 2: The testing accuracies on the conditions whether the dropout is closed or open in inference phase. "Mod.": Model; "Acc.": Accuracy; "Dro.St.": Dropout Status in inference phase.

Acc. Dro.St. Mod.	closed	open
Baseline	61.39	57.34
Baseline+SDL	62.12	60.01

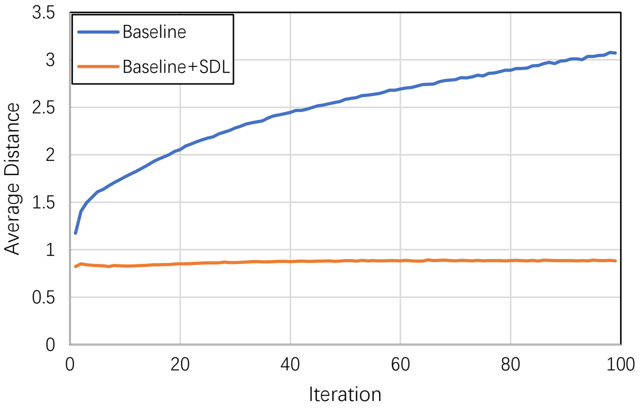


Figure 6: The comparison of the output variance between models with/without siamese dropout.

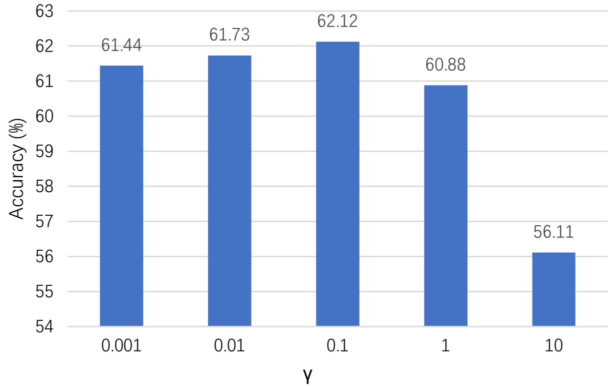


Figure 7: The influence of the loss weight factor γ on the performance of siamese dropout.

In order to explicitly demonstrate that our siamese dropout can decrease the output variance, we record the values of the output distance $\frac{1}{n} \parallel \boldsymbol{o} - \boldsymbol{o'} \parallel_2$ in each epoch, and the result curves are in Fig.6. The curves imply that the effect of siamese dropout is significant: In **Baseline** model, the distance between the two outputs keeps increasing while in **Baseline+SDL**, it is still controlled at a very low level. We further conduct experiments to explore whether the decrease of output variance can reduce the gap between training and inference phase and the results are listed in Table 2. In this table, each model is evaluated twice with different dropout status: closed or open. If dropout is closed, then it acts as an identity layer which is our usual practice. If dropout is open, then it will act as in training phase and conduct randomly dropping out even if it is in inference phase. Consider an idealistic case that the siamese dropout loss is optimized to 0, and then the results of the two dropout status will be the same. Thus difference between performances in the two dropout status can represent the gap of training and inference. As we can see that in **Baseline** model, the gap is 4.01% while in **Baseline+SDL**, it is only 2.11%. This demonstrates that our siamese dropout can effectively reduce the gap and make VQA model more robust.

In addition, we also discuss the influence of the loss weight factor γ in Equation 7 (see Fig.7). The results show that γ should not be too big or siamese dropout loss may cause bad effects. In the overall loss function, the siamese dropout loss is supposed to act as an auxiliary role, or it may lead the optimization to the wrong direction.

4.5 Ablation Analysis

Table 3: The overall accuracies when using different combinations of the three techniques. CD:coherent dropout;SD: siamese dropout; RSGCD: residual stacked GRUs with coherent dropout.

No.	CD	SD	RSGCD	Accuracy(%)
1				61.39
2	✓			62.28
3		✓		62.12
4			✓	62.11
5	✓	✓		62.79
6	✓		✓	62.55
7		✓	✓	62.67
8	✓	✓	✓	**63.01**

In this paper, we propose three techniques about dropout: coherent dropout, siamese dropout, and RSGCD. In Table 3, we compare the performances of models with one or several of the three techniques. Here, the siamese dropout uses the same configuration of SDL@(outperforms) and the RSGCD includes three stacked GRUs. Note that the dropout in RSGCD encoder is always the coherent dropout, i.e., in No.4 and No.7, only the dropout in RSGCD is coherent while the other is incoherent. From experiments of No.1-4, we can see that any of the three techniques can bring much improvement to baseline model, which demonstrates their effectiveness. And the results of No.5-7 show that although the effects of the three techniques can not be linearly superposed, the performance of the combination of two techniques is also higher than that of single one. This implies that there are complementarity among the siamese dropout, coherent dropout and RSGCD. Thus as shown in No.8, the best model can be obtained by using all of the three techniques. And the final improvement is also significant: 1.62% higher than baseline model.

4.6 Effect on State-of-the-art Methods

In order to verify the effectiveness of the proposed methods, we adopt our coherent dropout, siamese dropout and RSGCD question encoder into the state-of-the-art methods and the results are shown in Table 4. For MCB† [8], we use coherent dropout, SDL@(output) and RSGCD encoder with 3 LSTMs. For MUTAN [5], since it doesn’t provide single model performance on *test* split, we train the model by ourselves [1] without Visual Genome [18] dataset. For MUTAN†, we use coherent dropout, SDL@(output) and RSGCD encoder with 4 GRUs. For BottomUp†, we use coherent dropout, SDL@(output + mixed_feature) and RSGCD with 4 GRUs. We notice that for each of the state-of-the-art approaches, our methods can still enhance the model and improve the performance.

Table 4: The improvements brought by our method on the state-of-the-art methods. † denotes the model uses coherent dropout, siamese dropout and RSGCD. “Ver.” stands for which dataset is used to train the model.

		Accuracy			
Method	Ver.	Yes/No	Number	Other	Overall
MCB[8]	v1	82.30	37.20	57.40	65.40
MCB†	v1	**83.20**	**37.60**	**58.00**	**66.10**
MUTAN[5]	v1	84.08	40.00	54.82	65.23
MUTAN†	v1	**85.16**	**40.17**	**56.60**	**66.54**
BottomUp[2, 31]	v2	82.20	43.90	56.26	65.32
BottomUp†	v2	**83.37**	**45.40**	**57.79**	**66.92**

5 CONCLUSION

In visual question answering, many models use dropout to prevent overfitting. We notice that in multi-path networks, the current way to use dropout can cause two problems: the co-adaptations of neurons and the increase of output variance. In order to prevent co-adaptations, we propose the *coherent dropout* which is simple in implementation but effective on preventing overfitting. As for controlling the output variance, we develop a *siamese dropout* mechanism in training, which can explicitly minimize the output variance and reduce the gap between training and inference phase. We also explore how to utilize the coherent dropout to design a deeper question encoder and find that the residual stacked GRUs with coherent dropout structure is very flexible to stack more GRUs. We conduct extensive experiments to prove the effectiveness of the proposed methods and verify that our approach can also bring additional improvements on the state-of-the-art methods.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (61472422 and 61332016).

REFERENCES

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2017. Don’t Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. *arXiv preprint arXiv:1712.00377* (2017).

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-up and top-down attention for image captioning and VQA. *arXiv preprint arXiv:1707.07998* (2017).

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 39–48.

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. 2425–2433.

[5] Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. Mutan: Multimodal tucker fusion for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, Vol. 1. 3.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase

[1] The code is in open access on https://github.com/Cadene/vqa.pytorch provided by the authors of MUTAN.

representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[7] L ea Yu. 2015. Visual madlibs: Fill in the blank image generation and question answering. arXiv preprint. *arXiv* 1506 (2015), 3.

[8] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847* (2016).

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

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, Vol. 1. 9.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[12] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Michael I Jordan. 1997. Serial order: A parallel distributed processing approach. In *Advances in psychology*. Vol. 121. Elsevier, 471–495.

[15] Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 1983–1991.

[16] Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Hadamard Product for Low-rank Bilinear Pooling. In *The 5th International Conference on Learning Representations*.

[17] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). arXiv:1412.6980 http://arxiv.org/abs/1412.6980

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123, 1 (2017), 32–73.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[21] Adi Livnat, Christos Papadimitriou, Nicholas Pippenger, and Marcus W Feldman. 2010. Sex, mixability, and modularity. *Proceedings of the National Academy of Sciences* 107, 4 (2010), 1452–1457.

[22] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*. 289–297.

[23] Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*. 1682–1690.

[24] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2016. Dual attention networks for multimodal reasoning and matching. *arXiv preprint arXiv:1611.00471* (2016).

[25] Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Image question answering: A visual semantic embedding model and a new dataset. *Proc. Advances in Neural Inf. Process. Syst* 1, 2 (2015), 5.

[26] Idan Schwartz, Alexander Schwing, and Tamir Hazan. 2017. High-Order Attention Models for Visual Question Answering. In *Advances in Neural Information Processing Systems*. 3667–3677.

[27] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[29] Dendi Suhubdy. 2018. Fraternal Dropout. (2018).

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. 2015. Going deeper with convolutions. Cvpr.

[31] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. 2017. Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge. (2017). arXiv:1708.02711 http://arxiv.org/abs/1708.02711

[32] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2017. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence* (2017).

[33] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick. 2015. Explicit knowledge-based reasoning for visual question answering. *arXiv preprint arXiv:1511.02570* (2015).

[34] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2017. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding* 163 (2017), 21–40.

[35] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 21–29.

[36] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis*, Vol. 3.

[37] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* (2014).

[38] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and yang: Balancing and answering binary visual questions. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 5014–5022.

[39] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4995–5004.