

---

# Chain of Reasoning for Visual Question Answering

---

Chenfei Wu\*, Jinlai Liu\*, Xiaojie Wang, Xuan Dong

Center for Intelligence Science and Technology

Beijing University of Posts and Telecommunications

{wuchenfei, liujinlai, xjwang, dongxuan8811}@bupt.edu.cn

## Abstract

Reasoning plays an essential role in *Visual Question Answering* (VQA). Multi-step and dynamic reasoning is often necessary for answering complex questions. For example, a question “*What is placed next to the bus on the right of the picture?*” talks about a compound object “*bus on the right*,” which is generated by the relation  $\langle \text{bus}, \text{on the right of, picture} \rangle$ . Furthermore, a new relation including this compound object  $\langle \text{sign, next to, bus on the right} \rangle$  is then required to infer the answer. However, previous methods support either one-step or static reasoning, without updating relations or generating compound objects. This paper proposes a novel reasoning model for addressing these problems. A *chain of reasoning* (CoR) is constructed for supporting multi-step and dynamic reasoning on changed relations and objects. In detail, iteratively, the relational reasoning operations form new relations between objects, and the object refining operations generate new compound objects from relations. We achieve new state-of-the-art results on four publicly available datasets. The visualization of the chain of reasoning illustrates the progress that the CoR generates new compound objects that lead to the answer of the question step by step.

## 1 Introduction

“The technical issues of acquiring knowledge, representing it, and using it appropriately to construct and explain lines-of-reasoning, are important problems in the design of knowledge-based systems, which illuminates the art of Artificial Intelligence” [1]. Advances in image and language processing have developed powerful tools on knowledge representation, such as long short-term memory (LSTM) [2] and convolutional neural network (CNN) [3]. However, it is still a challenge to construct “lines-of-reasoning” with these representations for different tasks. This paper meets the challenge in visual question answering, a typical field of Artificial Intelligence.

*Visual question answering* (VQA) aims to select an answer given an image and a related question. The left part of Fig. 1 gives an example of the image and the question. Lots of work has been done on this task in recent years. Among them, reasoning, named in different ways, plays a critical role. Most of existing VQA models that enable reasoning can be divided into three categories. Firstly, relation-based method [4] views reasoning procedure as relational reasoning. It calculates the relations between image regions to infer the answer in one-step. However, one-step relational reasoning can only construct pairwise relations between initial objects, which is not always sufficient for complex questions. It is not a trivial problem to extend one-step reasoning to multi-step because of the exponential increase of computational complexity. Secondly, attention-based methods [5, 6] view reasoning procedure as to update the attention distribution on objects, such as image regions or bounding boxes, so as to gradually infer the answer. However, no matter how many times the attention distributions are updated, the objects are still from the original input, and the entire reasoning

---

\*The first two authors contributed equally.

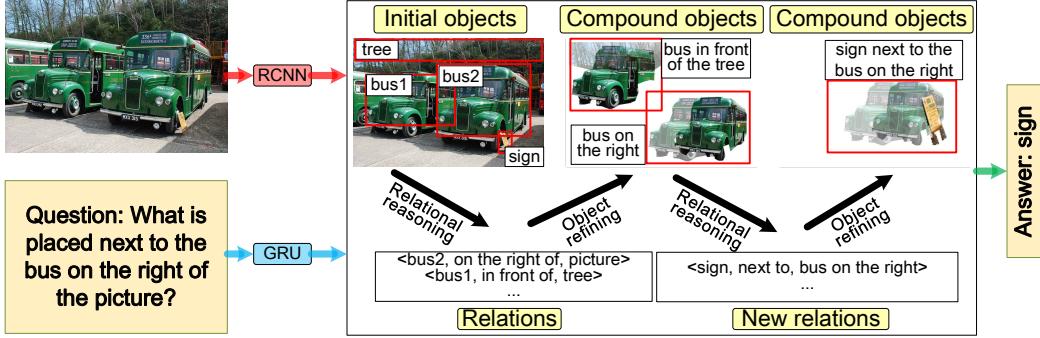


Figure 1: Chain of Reasoning for VQA. The alternate updating of objects and relations forms a chain of reasoning. The relational reasoning operation forms new relations between objects. The object refining operation generates new compound objects from relations.

procedure does not produce compound objects, such as “*sign next to the bus on the right*”, which many questions talk about. Thirdly, module-based methods [7, 8, 9] view reasoning procedure as a layout generated from manually pre-defined modules. It uses the layout to instantiate modular networks. However, the modules are pre-defined which means the reasoning procedure does not produce new modules or relations anymore. As a result, it is difficult to meet the requirements of diversity of relations in dynamic and multi-step reasoning.

This paper tries to construct a chain of reasoning (CoR) for addressing these problems. Both of the iteratively updated relations and compound objects are used as nodes in the chain. Updated relations push reasoning to involve more compound objects; compound objects maintain the intermediate conclusions of reasoning and make the next-step relational reasoning possible by lowering the computational complexity efficiently. An example of the CoR is shown in Fig. 1. Initial objects in the image are first recognized, such as two buses and a sign in the original image. All pairwise relations between these objects are then calculated, and a combination of the relations are used to generate compound objects, such as “*bus on the right*.” More complex relations are further calculated between the compound objects and initial objects to generate more complex compound objects, such as “*sign next to the bus on the right*,” which brings us the answer.

In summary, our contributions are as follows:

- We introduce a new VQA model that performs a chain of reasoning, which generates new relations and compound objects dynamically to infer the answer.
- We achieve new state-of-the-art results on four publicly available datasets. We conduct a detailed ablation study to show that our proposed chain structure is superior to stack structure and parallel structure.
- We visualize the chain of reasoning, which shows the progress that the CoR generates new compound objects dynamically that lead to the answer of the question step by step.

## 2 Related Work

Reasoning plays a crucial role in VQA. Recent studies modeled the reasoning procedure from different perspectives. In this section, we briefly review three types of existing work that enable reasoning. We also highlight differences between previous models and ours.

**Relation-based methods** The relation-based method performs one-step relational reasoning to infer the answer. [4] proposed a plug-and-play module called “Relation Networks” (RN). RN uses full arrangement to model all the interactions between objects in the image and performs multi-layer perceptrons (MLPs) to calculate all the relations. Then, the relations are summed and passed through other MLPs to infer the final answer. Modeling pairwise relationships already brings the  $O(m^2)$  computational complexity and makes it impossible to carry out multi-step reasoning. By object refining, our model lowers the computational complexity and makes the multi-step reasoning possible.

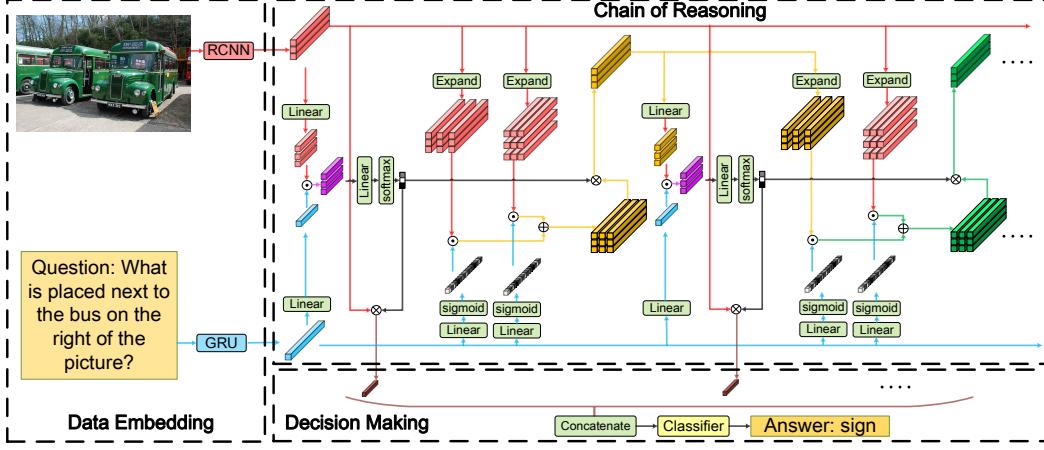


Figure 2: The overall structure of the proposed model for solving the VQA task. It consists of Data Embedding, Chain of Reasoning, and Decision Making, marked with dash lines respectively.

**Attention-based methods** Usually, attention-based methods enable reasoning by locating relevant objects in original input features, such as bounding boxes or image regions. Initially, [10] proposed one-step attention to locate relevant objects of images. Furthermore, [5, 6] proposed multi-step attention to update relevant objects of images and infer the answer progressively. Additionally, [11, 12] proposed multi-modal attention, which finds not only the relevant objects of images but also questions or answers. Recently, [13, 14, 15, 16] used bilinear fusion in attention mechanism to find more accurate objects of input features. Attention distributions in the above work are always on original input features. In contrast, our model pay attentions on not only objects in original input features but also new compound objects generated dynamically during reasoning.

**Module-based methods** Module-based methods try to define relations as modules in advance, and the reasoning procedure is determined by a layout generated from these modules. [7] proposed neural module network, which uses fixed layouts generated from dependency parses. Later, [8] proposed dynamic neural module network, which learns to optimize the layout structure by predicting a list of layout candidates. However, the layout candidates are still generated by dependency parses. To solve this problem, [9] proposed an end-to-end module network, which learns to optimize over full space of network and requires no parser at evaluation time. Our model forms new relations dynamically in the reasoning procedure, instead of choosing from a set of manually pre-defined modules.

### 3 Chain of Reasoning based model for VQA

The overall structure of our model for VQA is illustrated in Fig. 2. It consists of three parts: Data Embedding, Chain of Reasoning, and Decision Making. Data Embedding pre-processes the image and question. Chain of Reasoning is the core part of the model. Starting from outputs of Data Embedding, relational reasoning on initial objects forms new relations, and object refining generates new compound objects based on the new relations. Iteratively, these two operations on updated relations and objects build the chain of reasoning, which outputs a series of results. Decision Making makes use of all the results to select the final answer of the question. We give the details of the three parts in Section 3.1~3.3 respectively.

#### 3.1 Data Embedding

Faster-RCNN [17] is used to encode images with the static features provided by bottom-up attention [18], GRU [19] is used to encode text with the parameters initialized with skip-thoughts [20], as denoted in Eq. (1).

$$V = RCNN(image), \quad Q = GRU(question), \quad (1)$$

where  $V \in \mathbb{R}^{m \times d_v}$  denotes the visual features of the top-ranked  $m$  detection boxes and  $Q \in \mathbb{R}^{d_q}$  denotes the question embedding. Here,  $V$  is viewed as a set of  $m$  initial objects, i.e.  $V = \{v_1, v_2, \dots, v_m\}$ . From the perspective of reasoning,  $V$  can also be viewed as  $m$  initial premises.

### 3.2 Chain of Reasoning

Starting from initial objects  $O^{(1)} = V$  defined in Eq. (1), a chain of reasoning consists of a series of sub-chains and an output at each time, which is explained in Fig. 3.

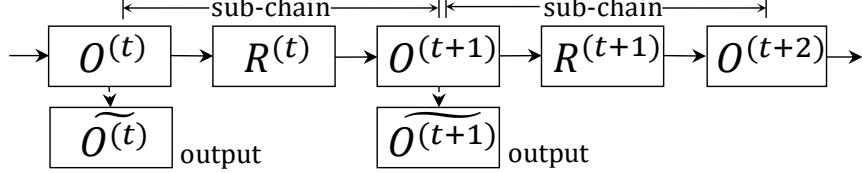


Figure 3: Sub-chains and their outputs in Chain of Reasoning.

In Fig. 3,  $O^{(t)} \in \mathbb{R}^{m \times d_v}$  is the set of initial objects at time  $t = 1$  or compound objects at time  $t > 1$ .  $O^{(t)} \in \mathbb{R}^{d_v}$  is the output of the chain at time  $t$ .  $R^{(t)} \in \mathbb{R}^{m \times m \times d_v}$  is the set of updated relations at time  $t$ .  $O^{(t+1)} \in \mathbb{R}^{m \times d_v}$  is the set of new compound objects at time  $t + 1$ . From the perspective of reasoning,  $O^{(t)}$  can also be viewed as intermediate conclusions when  $t > 1$ . We first give the details on the output at time  $t$ , and then describe how the sub-chain is formed.

The output at time  $t$  is designed to capture information provided by  $O^{(t)}$  under the guidance of question . An attention-based method is used as in Eq. (2)~(5).

$$P^{(t)} = \text{relu}(O^{(t)}W_o^{(t)}), \quad S^{(t)} = \text{relu}(QW_q^{(t)}), \quad (2)$$

$$F^{(t)} = \sum_{k=1}^K (P^{(t)}W_{p,k}^{(t)}) \odot (S^{(t)}W_{s,k}^{(t)}) \quad (3)$$

$$\alpha^{(t)} = \text{softmax}(F^{(t)}W_f^{(t)}), \quad (4)$$

$$\widetilde{O}^{(t)} = (\alpha^{(t)})^\top O^{(t)}, \quad (5)$$

where Eq. (2) maps the objects at time  $t$  to  $P^{(t)} \in \mathbb{R}^{m \times d_p}$  and maps the question feature to  $S^{(t)} \in \mathbb{R}^{d_s}$  at time  $t$ . Eq. (3) uses the Mutan fusion mechanism proposed by [16].  $K$  is the hyperparameter.  $F^{(t)} \in \mathbb{R}^{m \times d_f}$  is the fusion embedding at time  $t$ .  $\alpha^{(t)} \in \mathbb{R}^m$  in Eq. (4) is the attention distribution over the  $m$  compound objects at time  $t$ .  $\widetilde{O}^{(t)} \in \mathbb{R}^{d_v}$  in Eq. (5) is the result of attention at time  $t$ , which is also the output of chain of reasoning at time  $t$ . The output at each time  $t$  will be used for final decision making. To write simple, we omit the bias  $b$ .

The sub-chain  $O^{(t)} \rightarrow R^{(t)} \rightarrow O^{(t+1)}$  is performed in two operations. The first operation from  $O^{(t)}$  to  $R^{(t)}$  is called relational reasoning which forms new relations between objects, and the second operation from  $R^{(t)}$  to  $O^{(t+1)}$  is called object refining which generates new compound objects to start a new sub-chain. We introduce them respectively as follows.

**Relational reasoning from  $O^{(t)}$  to  $R^{(t)}$**  The  $m$  objects in  $O^{(t)}$  interact with the  $m$  initial objects in  $O^{(1)}$  under the guidance of the question  $Q$ , as denoted in Eq. (6)~(7).

$$G_l = \sigma(\text{relu}(QW_{l_1})W_{l_2}), \quad G_r = \sigma(\text{relu}(QW_{r_1})W_{r_2}), \quad (6)$$

$$R_{ij}^{(t)} = (O_i^{(t)} \odot G_l) \oplus (O_j^{(1)} \odot G_r), \quad (7)$$

where Eq (6) maps question feature to the same dimension as the object feature by a two-layer MLP with different weights respectively.  $\sigma$  is the sigmoid function.  $G_l, G_r \in \mathbb{R}^{d_v}$  are the guidances. Eq. 7 is the sum of the guided  $i$ th compound object at time  $t$  and the guided  $j$ th initial object.  $\odot$  denotes the element-wise multiplication and  $\oplus$  denotes the element-wise summation.

Notice that  $G_l$  and  $G_r$  are different guidances with different weights, but the weights in  $G_l$  and  $G_r$  are shared among all sub-chains respectively. As a result, two sets of weights are trained: the set of weights in  $G_l$  make the question focus on the compound objects and another set of weights in  $G_r$  make the question focus on the initial objects. This is in line with the reasoning procedure — the question decides what the model should do for the intermediate conclusions it already got and the initial premises. Besides, initial objects  $O_j^{(1)}$  used at each time allow the model to capture initial premises through the whole reasoning procedure.

**Object refining from  $R^{(t)}$  to  $O^{(t+1)}$**  The previous relational reasoning operation produces  $m \times m$  relations between  $m$  compound objects and  $m$  initial objects. Since modeling the pairwise relations increases the complexity of reasoning from  $O(m)$  to  $O(m^2)$ ,  $n$ -step reasoning will face the complexity of  $O(m^n)$ . In order to avoid the exponential complexity of multi-step reasoning, we refine these relations to  $m$  new compound objects, each denoted in Eq. (8):

$$O_j^{(t+1)} = \sum_{i=1}^m \alpha_i^{(t)} R_{ij}^{(t)}, \quad (8)$$

where  $O_j^{(t+1)}$  is the  $j$ th compound object at time  $t + 1$ , In Eq. (8), the attention weights of the compound objects  $\alpha^{(t)}$  are used to refine the relations  $R^{(t)}$  formed by the compound objects and the initial objects. This has two advantages: Firstly, Eq. (8) is more in line with the reasoning procedure. The  $j$ th compound object at time  $t + 1$  is determined by all the compound objects at time  $t$  and the  $j$ th initial object. This means that any conclusion generated by the next reasoning step will use all the intermediate conclusions in the previous step. At the same time, if an intermediate conclusion in the previous step is important, then its information is more likely to be used in the next step. Secondly, Eq. (8) makes it mathematically simple and computationally feasible to begin a next turn reasoning. Mathematically, we can use a single set of equations to describe the whole chain. Computationally, we can keep the complexity of  $O(nm^2)$  when we perform  $n$  sub-chains of reasoning.

### 3.3 Decision Making

The decision maker at time  $T$  gives an answer to the question by making use of all the outputs  $\widetilde{O}^{(t)}$  ( $t = 1, 2, \dots, T$ ). An concatenation is employed for integrating  $T$  outputs in Eq. (9).

$$O^* = [\text{relu}(\widetilde{O}^{(1)}W^{(1)}); \text{relu}(\widetilde{O}^{(2)}W^{(2)}); \dots; \text{relu}(\widetilde{O}^{(T)}W^{(T)})], \quad (9)$$

where  $O^* \in \mathbb{R}^{d_*}$  is the joint feature of outputs. We further fuse joint feature and question by Eq.(10).

$$H = \sum_{k=1}^K (O^* W_{o^*, k}) \odot (Q W_{q^*, k}), \quad (10)$$

where  $K \in \mathbb{R}^+$  is the hyperparameter,  $H \in \mathbb{R}^{d_h}$  is the joint embedding. Finally, a linear layer with a softmax activation function is used to predict the candidate answer distribution as shown in Eq. (11).

$$\hat{a} = \text{softmax}(HW_h), \quad (11)$$

### 3.4 Training

We first calculate the ground-truth answer distribution in Eq. (12):

$$a_i = \frac{\sum_{j=1}^N \mathbf{1}\{u_j = i\}}{N - \sum_{j=1}^N \mathbf{1}\{u_j \notin \mathcal{D}\}}, \quad (12)$$

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. (13):

$$\mathcal{L}(\hat{a}, a) = \sum_{i=1}^{|\mathcal{D}|} a_i \log \left( \frac{a_i}{\hat{a}_i} \right). \quad (13)$$

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

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	Y/N	Num.	Other	All	Y/N	Num.
Single image feature	HighOrderAtt[12]	-	-	-	-	69.4	-	-	-	-	-	-	-	69.3	-	-
MLB(7)[14]	66.77	84.54	39.21	57.81	-	66.89	84.61	39.07	57.79	-	-	-	-	-	-	-
Mutan(5)[16]	67.42	85.14	39.81	58.52	-	67.36	84.91	39.79	58.35	-	-	-	-	-	-	-
Multi image feature	DualMFA[21]	66.01	83.59	40.18	56.84	70.04	66.09	83.37	40.39	56.89	69.97	-	-	-	-	-
ReasonNet[22]	-	-	-	-	-	-	67.9	84.0	38.7	<b>60.4</b>	-	-	-	-	-	-
Single image feature	CoR-2(36boxes) (ours)	68.16	85.57	43.76	58.80	72.60	68.19	85.61	43.10	58.75	72.61	-	-	-	-	-
	CoR-3(36boxes) (ours)	<b>68.37</b>	<b>85.69</b>	<b>44.06</b>	<b>59.08</b>	<b>72.84</b>	<b>68.54</b>	<b>85.83</b>	<b>43.93</b>	59.11	<b>72.93</b>	-	-	-	-	-

Table 2: Comparision with the state-of-the-arts on the VQA 2.0 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[23]	64.73	81.29	42.99	55.55	-	-	-	-
Up-Down(36 boxes)[24]	65.32	81.82	44.21	56.05	65.67	82.20	43.90	56.26
LC_Baseline(100 boxes)[25]	67.50	82.98	46.88	58.99	67.78	83.21	46.60	59.20
LC_Counting(100 boxes)[25]	68.09	83.14	<b>51.62</b>	58.97	68.41	83.56	<b>51.39</b>	59.11
CoR-2(36 boxes) (ours)	67.96	84.7	47.1	58.42	68.15	84.82	46.8	58.52
CoR-3(36 boxes) (ours)	68.19	84.98	47.19	58.64	68.59	85.16	47.19	59.07
CoR-3(100 boxes) (ours)	<b>68.62</b>	<b>85.22</b>	47.95	<b>59.15</b>	<b>69.14</b>	<b>85.76</b>	48.4	<b>59.43</b>

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

Method	All	Obj.	Num.	Color	Loc.	WUPS0.9	WUPSO.0
QRU [26]	62.50	65.06	46.90	60.50	56.99	72.58	91.62
HieCoAtt [11]	65.4	68.0	51.0	62.9	58.8	75.1	92.0
Dual-MFA [21]	66.49	68.86	51.32	65.89	58.92	76.15	92.29
CoR-2(36 boxes) (ours)	68.67	69.76	55.14	73.36	59.52	77.47	92.68
CoR-3(36 boxes) (ours)	<b>69.38</b>	<b>70.42</b>	<b>55.83</b>	<b>74.13</b>	<b>60.57</b>	<b>78.10</b>	<b>92.86</b>

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

Question Type	MCB-A[13]	RAU[27]	CATL-QTA <sup>W</sup> [28]	CoR-2 (ours)	CoR-3 (ours)
Scene Recognition	93.06	93.96	93.80	94.48	<b>94.68</b>
Sport Recognition	92.77	93.47	95.55	<b>95.94</b>	95.90
Color Attributes	68.54	66.86	60.16	73.59	<b>74.47</b>
Other Attributes	56.72	56.49	54.36	59.59	<b>60.02</b>
Activity Recognition	52.35	51.60	60.10	60.29	<b>62.19</b>
Positional Reasoning	35.40	35.26	34.71	39.34	<b>40.92</b>
Sub. Object Recognition	85.54	86.11	86.98	88.38	<b>88.83</b>
Absurd	84.82	96.08	<b>100.00</b>	95.17	94.70
Utility and Affordances	35.09	31.58	31.48	<b>40.35</b>	37.43
Object Presence	93.64	94.38	94.55	95.40	<b>95.75</b>
Counting	51.01	48.43	53.25	57.72	<b>58.83</b>
Sentiment Understanding	66.25	60.09	64.38	66.72	<b>67.19</b>
Overall (Arithmetic MPT)	67.90	67.81	69.11	72.25	<b>72.58</b>
Overall (Harmonic MPT)	60.47	59.00	60.08	65.65	<b>65.77</b>
Overall Accuracy	81.86	84.26	85.03	86.58	<b>86.91</b>

## 4 Experiments

### 4.1 Datasets and evaluation metrics

We evaluate our model on four public datasets: the VQA 1.0 dataset [29], the VQA 2.0 dataset [30], the COCO-QA dataset[31] and the TDIUC dataset [27]. VQA 1.0 contains 614,163 samples, including 204,721 images from COCO [32]. VQA 2.0 is a more balanced version and contains 1,105,904 samples. COCO-QA is a smaller dataset that contains 78,736 samples. TDIUC is a larger dataset that contains 1,654,167 samples and 12 question types. For VQA 1.0 and VQA 2.0, we use the evaluation tool proposed in [29] to evaluate the model. For COCO-QA and TDIUC, we calculate the simple accuracy for each question type. Besides, additional WUPS [33] is calculated for COCO-QA and additional Arithmetic/Harmonic mean-per-type (MPT) [27] is calculated for TDIUC.

### 4.2 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 chain of reasoning phase, the number of hidden layer in Mutan is 510; hyperparameter  $K$  is 5. The attention hidden unit number is 620. In the decision making phase, the joint feature embedding is set to 510. All the nonlinear layers of the model all use the relu activation function and dropout [34] to prevent overfitting. All settings are commonly used in previous work. We implement the model using Pytorch. We use Adam[35] to train the model with a learning rate of  $10^{-4}$  and a batch\_size of 64. More details, including source codes, will be published in the near future.

### 4.3 Comparison with the state-of-the-art

In this section, we compare our single CoR- $T$  model with the state-of-the-art models on four datasets. CoR- $T$  means that the model consists of  $T$  sub-chains. Firstly, Tab. 1 shows the results on the VQA 1.0 dataset. Using a single image feature, CoR-3 not only outperforms all the models that use single image feature but also outperforms the state-of-the-art ReasonNet [22] model, which uses six input image features including face analysis, object classification, scene classification and so on. Secondly, Tab. 2 shows the results on the VQA 2.0 dataset. Compared with Up-Down (36 boxes) [24], which is the winning model in the VQA challenge 2017, CoR-3 (36 boxes) achieves 2.92% higher accuracy in test-std set. Compared with the most recent state-of-the-art model LC\_counting (100 boxes) [25], our single CoR-3 (100 boxes) model achieves a new state-of-the-art result of 69.14% in the test-std set. Thirdly, Tab. 3 shows the results on the COCO-QA dataset. CoR-3 improves the overall accuracy of the state-of-the-art Dual-MFA from 66.49% to 69.38%. In particular, there is an improvement of 4.51% in “Num.” and 8.24% in “Color”. Fourthly, Tab. 4 shows the results on the TDIUC dataset. CoR-3 improves the overall accuracy of the state-of-the-art CATL-QTA<sup>W</sup> [28] from 85.03% to 86.91%. There is also an improvement of 5.58% in “Counting” and 5.93% in “Color Attributes”. In summary, CoR achieves consistently best performance on all four datasets.

### 4.4 Ablation study

In this section, we conduct some ablation experiments. 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.

Tab. 5 shows the effectiveness of the chain structure. We implement MLB[14], Mutan [16] and their stack and parallel structure. The stack structure is proposed by SAN [5], which stacks 2 or 3 attention layers. The parallel structure is similar to Multi-Head Attention [36], which consists of 2 or 3 attention layers running in parallel. As shown in Tab. 5, the chain structure not only significantly improves the performance of attention models but also superior to their stack or parallel structures. For example, compared with Mutan, Mutan-Stack-3 is only 0.29% higher while CoR-3 is 1.53% higher. Furthermore, the chain structure is insensitive to the attention model. CoR-2 and CoR-3 can achieve high performance whether using Mutan or MLB.

Tab. 6 shows the effectiveness of the relational reasoning operation. Firstly, we implement CoR-2 with  $[O_i^{(t)}; O_j^{(1)}; G]W_1$ , which is proposed RN [4]. We find it lowers the performance ( $64.96\% \rightarrow 62.46\%$ ). This is because the purpose of relational reasoning here is to prepare for generating compound

Table 5: Effectiveness of the chain structure on the VQA 2.0 validation.

<b>Method</b>	MLB[14]	MLB-Stack-2	MLB-Stack-3	MLB-Parallel-2	MLB-Parallel-3	CoR-2 with MLB	CoR-3 with MLB
<b>Val</b>	62.91	63.28	63.55	63.20	63.28	64.90	<b>64.96</b>
<b>Method</b>	Mutan[16]	Mutan-Stack-2	Mutan-Stack-3	Mutan-Parallel-2	Mutan-Parallel-3	CoR-2	CoR-3
<b>Val</b>	63.61	63.78	63.90	63.66	63.80	64.96	<b>65.14</b>

Table 6: Effectiveness of relational reasoning operation on the VQA 2.0 validation.

<b>Method</b>	<b>Val</b>
CoR-2 with $[O_i^{(t)}; O_j^{(1)}; G]W_1$	62.46
CoR-2 with $(O_i^{(t)} + O_j^{(1)}) \odot G$	64.73
CoR-2 with $(O_i^{(t)} \odot G_l) \oplus (O_j^{(t)} \odot G_r)$	64.24
CoR-2	<b>64.96</b>

Table 7: Effectiveness of object refining operation on the VQA 2.0 validation.

<b>Method</b>	<b>Val</b>
CoR-2 with $\sum_{i=1}^m \alpha_i^{(t)} R_{ji}^{(t)}$	64.42
CoR-2	<b>64.96</b>

Table 8: Effectiveness of the model on different question types on the CLEVR dataset.

<b>Method</b>	Overall	Count	Exist	Compare Numbers	Query Attribute	Compare Attribute
MLB	85.0	90.0	76.7	78.8	91.1	82.7
Mutan	86.3	92.5	80.2	81.7	91.2	84.5
RN	96.4	-	-	-	-	-
CoR-2	<b>98.7</b>	<b>98.8</b>	<b>97.7</b>	<b>92.3</b>	<b>99.9</b>	<b>99.7</b>

objects, and the element-wise sum in Eq. (7) is more fine-grained. Secondly, we implement CoR-2 with  $(O_i^{(t)} \oplus O_j^{(1)}) \odot G$ , which uses a single question guidance and also lowers performance ( $64.96\% \rightarrow 64.73\%$ ). This shows that different guidances for compound objects and initial objects are beneficial to improve the performance. Thirdly, we implement CoR-2 with  $(O_i^{(t)} \odot G_l) \oplus (O_j^{(t)} \odot G_r)$ , which calculates the relations by the compound objects themselves without the initial object  $O_j^{(1)}$ .

We find it still lowers the performance ( $64.96\% \rightarrow 64.24\%$ ). This shows using initial premises  $O_j^{(1)}$  at each step is crucial and may avoid “over-reasoning” by modeling very complex relations between compound objects.

Tab. 7 shows the effectiveness of the object refining operation. We implement a similar operation  $\sum_{i=1}^m \alpha_i^{(t)} R_{ji}^{(t)}$ . Although the formula is similar to  $\sum_{i=1}^m \alpha_i^{(t)} R_{ij}^{(t)}$  in Eq. 8, the meaning is totally different.  $\sum_{i=1}^m \alpha_i^{(t)} R_{ij}^{(t)}$  generates the  $j$ th compound object by weighted sum of the relations between each compound object and the  $j$ th initial object while  $\sum_{i=1}^m \alpha_i^{(t)} R_{ji}^{(t)}$  generates that by weighted sum of the relations between each initial object and  $j$ th compound object. The former focuses on using the previous reasoning conclusions while the latter focuses on the initial premises. CoR-2 has better results and is more in line with the reasoning procedure — focusing more on previous intermediate conclusions to push the next step reasoning.

Tab. 8 shows effectiveness of the model on different question types. We conduct experiments on the state description matrix version of the CLEVR dataset [37]. CoR-2 reaches an overall accuracy of 98.7%, which outperforms MLB and Mutan on the same setup. Furthermore, CoR-2 achieves the performance of 99.9% in question type of “Query Attribute” and 99.7% in question type of “Compare Attribute”. It is worth mentioning that there is still room for improvement in “Compare Numbers” questions.

#### 4.5 Qualitative evaluation

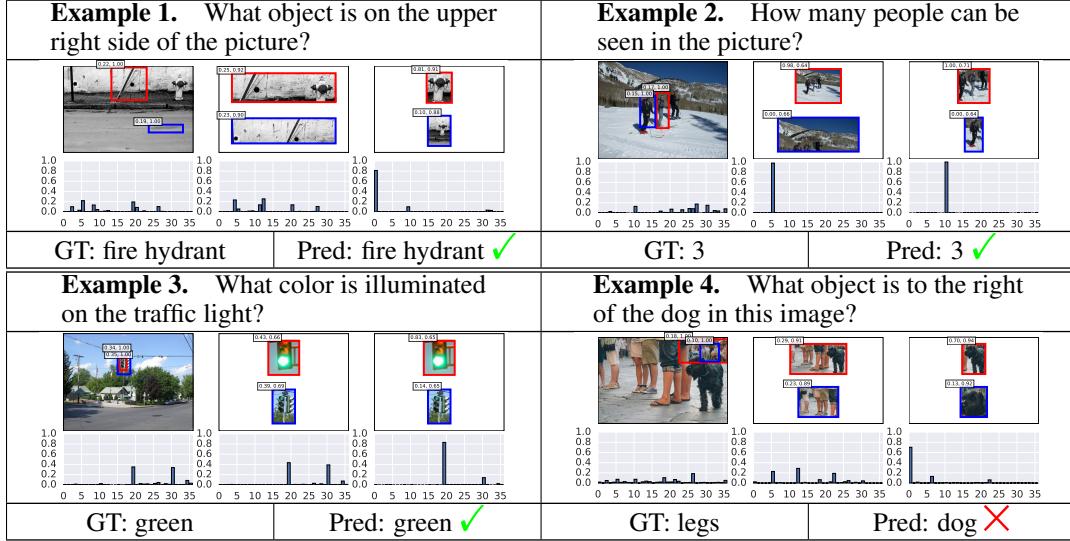


Figure 4: Visualization of the reasoning procedure of CoR-3.

In Figure 4, we visualize the compound objects generated by CoR-3 and their attention weights. Four examples are given including three success cases and one failure case. Each example contains three steps. The red box and the blue box in each step represent objects with the top two attention weights respectively. The initial objects in the first step are part of the original image and easy to visualize by the bounding box, but the compound objects in the second and third step are difficult to visualize directly. Therefore, we search from  $1105904 \times 36$  boxes ( $1105904$  is the number of samples and each sample has 36 boxes) and find the box with the most similar feature by cosine similarity to represent the compound object. The upper left corner of each box contains a tuple of the form  $(w, s)$ .  $w$  is the attention weight,  $s$  is the similarity between the searched box and the real compound object.

In **Example 1**, the left image shows a pillar (red box) and ground (blue box). Their values of  $w$  are 0.22 and 0.19 respectively. Since they are initial top two rcnn objects in  $O^{(1)}$ , the values of  $s$  are 1. The model focuses on some disperse “objects”, which can be further seen by attention distribution histogram below. The middle image shows top two compound objects in the second step. The red box focuses on “objects on the upper”. The attention weight of the red box increased slightly to 0.25. The similarity between the red box and the real compound object is 0.92. The right image shows top two more complex compound objects in the third step. The “objects on the upper right” has been focused in the red box. Interestingly, the  $w$  of the red box increases to 0.81, which means in the third step, CoR-3 is very confident that the box containing “hydrant” is exactly the final answer. Statistics show that 96.76% of the success cases satisfy the phenomenon of dispersion to concentration. In **Example 2~3**, two more success cases are shown. In **Example 4**, the model already gets the intermediate result “dog in the image” in the third step but fails to further find “leg on the right of the dog in the image”, which seems that three-step reasoning is insufficient here.

## 5 Conclusion

In this paper, we propose a novel chain of reasoning model for VQA task. The reasoning procedure is viewed as the alternate updating of objects and relations. Experimental results on four publicly available datasets show that CoR outperforms state-of-the-art approaches. Ablation study shows that proposed chain structure is superior to stack structure and parallel strucuture. The visualization of the chain of reasoning illustrates the progress that the CoR generates new compound objects that lead to the answer of the question step-by-step. In the future, we plan to apply CoR to other tasks that require reasoning like reading comprehension question answering or video question answering.

## Acknowledgments

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

## References

- [1] Edward A. Feigenbaum. The art of artificial intelligence. 1. Themes and case studies of knowledge engineering. Technical report, Stanford Univ CA Dept of Computer Science, 1977.
- [2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [3] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [4] Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.
- [5] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *CVPR*, 2016.
- [6] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, pages 451–466, 2016.
- [7] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, pages 39–48, 2016.
- [8] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *NAACL*, pages 1545–1554, 2016.
- [9] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, 2017.
- [10] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. ABC-CNN: An attention based convolutional neural network for visual question answering. *arXiv:1511.05960*, 2015.
- [11] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical Question-Image Co-Attention for Visual Question Answering. In *NIPS*, 2016.
- [12] Idan Schwartz, Alexander G. Schwing, and Tamir Hazan. High-Order Attention Models for Visual Question Answering. In *NIPS*, 2017.
- [13] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, pages 457–468, 2016.
- [14] Jin-Hwa Kim, Kyoung-Woon On, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. In *ICLR*, 2017.
- [15] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering. In *ICCV*, 2017.
- [16] Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *ICCV*, pages 2631–2639, 2017.
- [17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

- [18] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, page 6, 2018.
- [19] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv preprint arXiv:1409.1259*, 2014.
- [20] Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *NIPS*, pages 3294–3302, 2015.
- [21] Pan Lu, Hongsheng Li, Wei Zhang, Jianyong Wang, and Xiaogang Wang. Co-attending Free-form Regions and Detections with Multi-modal Multiplicative Feature Embedding for Visual Question Answering. In *AAAI*, 2018.
- [22] Ilija Ilievski and Jiashi Feng. Multimodal Learning and Reasoning for Visual Question Answering. In *NIPS*, pages 551–562, 2017.
- [23] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. Structured Attentions for Visual Question Answering. In *ICCV*, 2017.
- [24] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge. In *CVPR*, 2018.
- [25] Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Learning to Count Objects in Natural Images for Visual Question Answering. In *ICLR*, 2018.
- [26] Ruiyu Li and Jiaya Jia. Visual question answering with question representation update (qru). In *NIPS*, pages 4655–4663, 2016.
- [27] Kushal Kafle and Christopher Kanan. An Analysis of Visual Question Answering Algorithms. In *ICCV*, 2017.
- [28] Yang Shi, Tommaso Furlanello, Sheng Zha, and Animashree Anandkumar. Question Type Guided Attention in Visual Question Answering. In *ECCV*, 2018.
- [29] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, pages 2425–2433, 2015.
- [30] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, volume 1, page 9, 2017.
- [31] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *NIPS*, pages 2953–2961, 2015.
- [32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.
- [33] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, pages 1682–1690, 2014.
- [34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [35] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.
- [37] J. Johnson, B. Hariharan, L. v d Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *CVPR*, pages 1988–1997, July 2017.