Abstract

A compositional question refers to a question that contains multiple visual concepts (e.g., objects, attributes, and relationships) and requires compositional reasoning to answer. Existing VQA models can answer a compositional question well, but cannot work well in terms of reasoning consistency in answering the compositional question and its sub-questions. For example, a compositional question for an image is: “Are there any elephants to the right of the white bird?” and one of its sub-questions is “Is any bird visible in the scene?”. The models may answer “yes” to the compositional question, but “no” to the sub-question. This paper presents a dialog-like reasoning method for maintaining reasoning consistency in answering a compositional question and its sub-questions. Our method integrates the reasoning processes for the sub-questions into the reasoning process for the compositional question like a dialog task, and uses a consistency constraint to penalize inconsistent answer predictions. In order to enable quantitative evaluation of reasoning consistency, we construct a GQA-Sub dataset based on the well-organized GQA dataset. Experimental results on the GQA dataset and the GQA-Sub dataset demonstrate the effectiveness of our method.

1. Introduction

Compositional visual question answering (VQA) [3, 9, 27] is the task of providing an answer to a compositional question about an image based on the content of the image. A compositional question refers to a question that contains multiple visual concepts (e.g., objects, attributes, and relationships). The task requires a comprehensive understanding of the multi-modal inputs and compositional relational reasoning based on the understanding.

Existing reasoning models for compositional VQA can answer a compositional question well, but cannot work well in terms of reasoning consistency in answering the compositional question and its sub-questions. A compositional question usually contains various known visual concepts in the image. To answer the question, reasoning models are supposed to infer unknown concepts based on the known concepts. For a compositional question “Are there any elephants to the right of the white bird?” (shown in Fig. 1 (a)), the known concepts are “white” and “bird”, while the unknown concepts are “elephants” and “to the right of”. Obviously, the compositional question reveals answers to sub-questions about these known concepts, such as “Is any bird visible in the scene?”, and “What the white animal is called?”. The question requires a stronger reasoning abil-
ity to answer than its sub-questions. However, although a model accurately answers the compositional question, it may fail to provide correct answers for its sub-questions as shown in Fig. 1. The inconsistency indicates errors in the reasoning process for the compositional question.

This paper proposes a dialog-like reasoning method that integrates the reasoning processes for sub-questions into the reasoning process for a compositional question to maintain the reasoning consistency in compositional VQA. The method represents an input image as a structured visual graph and performs iterative language-guided graph convolution to learn contextual visual representations for compositional reasoning. The number of required iterations for each question is determined in advance. In the reasoning process of a compositional question, if the current number of iterations equals the required number of iterations of a sub-question, we use an answer classifier to answer the corresponding sub-question using current visual representations. A consistency constraint is further introduced to penalize inconsistent answer predictions. By answering sub-questions with the guidance of the compositional question, the method is supposed to capture the correlations of the question and its sub-questions for better consistency. The answering process for the sub-questions can also be regarded as the intermediate supervision of the reasoning process for the compositional question.

To enable the quantitative evaluation of reasoning consistency in compositional VQA, we build a GQA-Sub dataset based on the GQA dataset [11], a well-organized large-scale dataset for compositional VQA. We automatically decompose compositional questions of the GQA into sub-questions, and carefully balance the obtained sub-questions to avoid biases. Given a compositional question and its sub-questions, reasoning consistency can be quantitatively measured by evaluating whether predictions for these questions are contradictory. Experimental results on the GQA and the GQA-Sub demonstrate the effectiveness of our method. The data and code are publicly available at https://github.com/jingchenchen/ReasoningConsistency-VQA.

The contributions of this paper are two-fold:

1. We propose a dialog-like reasoning method that integrates reasoning processes for the sub-questions into the reasoning process for a compositional question to ensure reasoning consistency in compositional VQA.

2. We present a GQA-Sub dataset to evaluate the reasoning consistency of compositional VQA models.

2. Related Work

Consistency in VQA Consistency in VQA is defined as being able to answer questions posed from different semantic perspectives about a certain visual fact without any contradiction [19]. Specifically, a visual fact is defined as a triplet in a scene graph of an image. Recent work has proposed various datasets to measure the consistency of VQA models. Ribeiro et al. [20] first studied the inconsistency of VQA models and argue that we need to consider the relationship between predictions to measure true understanding. They automatically generate implications for questions in the VQA v1 dataset [3] to evaluate the consistency. The implications are questions about the same fact as the original question. Given a question-answer pair as “What room is this? bathroom”, they generate implications such as “Is this a bathroom?” and “Is there a bathroom in the picture?”. Ray et al. [19] generate entailed questions for questions in the VQA v2 dataset [5]. The entailed questions are also about the same fact as the original question. Selvaraju et al. [22] distinguish reasoning questions and perception questions and build datasets based on the VQA v2 to test the consistency of models for reasoning questions and perception questions. Yuan et al. [28] transform counting question of the VQA v2 by performing object-oriented partition, reordering, or reversion, to test the perception ability of VQA models. Shah et al. [23] and Whitehead et al. [26] rephrase questions in the VQA v2 to evaluate the VQA models’ robustness for linguistic variations. Previous work generates questions about one visual fact in the original question to measure the consistency of VQA models. By contrast, we focus on generating sub-questions about known visual facts in the compositional questions to test whether the VQA models are really capable of compositional reasoning.

Hudson and Manning [11] devise a consistency metric by generating entailed questions for compositional questions in the GQA [11]. In Sec. 4, we analyze the differences between entailed questions of the GQA and our sub-questions in detail.

Compositional VQA. Existing methods for compositional VQA can be mainly divided into two categories: modular and holistic. Modular methods [2, 4, 8, 24] assemble various modules according to an input question and execute the modules for reasoning. Holistic methods [9, 10, 12, 13, 16, 29] use a single model for different inputs to achieve reasoning. Our work aims to ensure the reasoning consistency for compositional VQA and is orthogonal to previous work.

3. Method

3.1. Overview

The compositional VQA task is to provide an answer \( A \) for a natural language compositional question \( Q \) about an image \( I \). The image \( I \) can be represented by a set of objects \( O_v = \{o_i\}_{i=1}^M \), where \( M \) is the number of objects in the image. A compositional question \( Q \) contains multiple concepts such as objects, relations, and attributes.
We decompose the question $Q$ into a set of sub-questions $Q = \{Q_i\}_{i=1}^N$, where $N$ is the number of sub-questions. We use $Q^f = \{Q\} \cup Q$ to denote a set of questions consisting of the question $Q$ and its sub-questions $Q$.

As shown in Fig. 2, we represent an input image as a visual graph to characterize its relational layout. For each question $q \in Q^f$, the proposed method performs $T_q$ iterations of graph convolution guided by the question $q$ to learn contextual visual representations to answer the question. Specifically, the number of iterations $T_q$ is determined by performing Part-of-speech (POS) tagging for the question via the Spacy tool [7] and computing the number of nouns in it. An answer classifier $\mathcal{F}$ is used to fuse the question and the graph to predict the answer. In particular, in the reasoning process of the compositional question $Q$, after the $t$-th iteration, where $t \in \{1, 2, ..., T_Q\}$, we use the classifier $\mathcal{F}$ to answer a sub-question $Q^i_t \in Q$, if and only if $t = T_{Q_i}$, according to the current graph.

### 3.2. Feature Encoding

For an input image, we build a visual graph $G = \{V, E\}$, where $V = \{v_i\}_{i=1}^M$ is a set of nodes and $E = \{e_{ij}\}_{i,j=1}^M$ denotes the relationships among objects. An undirected edge $e_{ij}$ connects each pair of nodes $v_i$ and $v_j$. Thus $G$ is a fully-connected graph. For each node $v_i$, we extract an appearance feature $l_i \in \mathbb{R}^{d_l \times 1}$ via the bottom-up attention model [1]. We encode the location and size information via a spatial feature $b_i \in \mathbb{R}^{d_b \times 1}$. Then we obtain the node embedding $v_i = W_v[l_i; b_i]$ by concatenating the two features and projecting the concatenated feature to a common space $\mathbb{R}^{d_v \times 1}$, where $W_v \in \mathbb{R}^{d_v \times (d_l + d_b)}$ is a learnable matrix.

$[\cdot \cdot \cdot]$ denotes the concatenation operation of two vectors. We do not obtain representations for edges in $G$ because the representations are not involved in graph convolution.

For a question $q \in Q^f$ containing $K$ words $\{w_k\}_{k=1}^K$, we use a Bi-LSTM [21] to encode the question and project it into the common space to obtain the sentence-level representation $v_{q}^{att} \in \mathbb{R}^{d_q \times 1}$. The word representation $w_k \in \mathbb{R}^{d_w \times 1}$ of the word $w_k$ is obtained by concatenating corresponding forward and backward hidden vectors and projecting it into the common space.

### 3.3. Architecture

The proposed method uses the language-guided graph convolution to learn context-aware visual representations for each question $q \in Q^f$, to achieve reasoning. For each node $v_i$, we first build a local representation $v_i^{loc} = v_i$ and initialize a contextual representation $v_i^{ctx} \in \mathbb{R}^{d_{ctx}}$ from a learned parameter as [9].

At the $t$-th iteration of graph convolution, the two representations are fused to obtain a joint representation $v_{i,t} \in \mathbb{R}^{3d_{ctx} \times 1}$ as

$$v_{i,t} = [v_i^{loc}, v_{i,t-1}^{ctx}; (W_1 v_i^{loc}) \circ (W_2 v_{i,t-1}^{ctx})],$$

where $\circ$ denotes element-wise multiplication. $W_1 \in \mathbb{R}^{d \times d}$ and $W_2 \in \mathbb{R}^{d \times d}$ are learnable matrices. We use a language attention to generate a command vector $c_t \in \mathbb{R}^{d \times 1}$ as

$$c_t = \sum_{k=1}^{K} \alpha_t^{L,k} w_k,$$

$$\alpha_t^{L,k} = \text{Softmax}_k (W_3 (w_k \circ (W_4 (W_5 q)))),$$
where $W_3 \in \mathbb{R}^{1 \times d}$, $W'_4 \in \mathbb{R}^{d \times d}$, and $W_5 \in \mathbb{R}^{d \times d}$ are learnable matrices.

Then we update the contextual representation $v_{i,t}^{ctx}$ for each node $v_i$ by using the joint representations $\{\tilde{v}_{i,t}\}_{t=1}^M$ of both the connected nodes and the node itself, based on the command vector $c_t$. The $v_{i,t}^{ctx}$ is computed by

$$v_{i,t}^{ctx} = W_6 \left[ v_{i,t-1}^{ctx}; \tilde{v}_{i,t} \right],$$

$$\tilde{v}_{i,t}^{ctx} = \sum_{j=1}^M \beta_{i,j}^t \odot (W_7 \tilde{v}_{j,t}),$$

where $W_6 \in \mathbb{R}^{d \times 2d}$, $W_7 \in \mathbb{R}^{d \times 3d}$, $W_8 \in \mathbb{R}^{d \times 3d}$, $W_9 \in \mathbb{R}^{d \times d}$, and $W_{10} \in \mathbb{R}^{d \times 3d}$ are learnable matrices.

After the graph convolution, an answer classifier is used for answer prediction. The answer classifier fuses local and contextual representations of nodes to obtain output representations, and uses the aggregated representations to obtain the predicted answer distribution $y_{pred}^q$ as

$$y_{pred}^q = W_{ans} \left( W_{11} \left[ \sum_{i=1}^{M} \alpha_i^V v_{i,T_q}; q \right] \right),$$

$$\alpha_i^V = \text{Softmax}_i \left( W_{12} \left( v_{i,T_q}^{out} \circ (W_{13} q) \right) \right),$$

$$v_{i,T_q}^{out} = W_{14} \left[ v_{i,T_q}^{loc}; v_{i,T_q}^{ctx} \right],$$

where $W_{ans} \in \mathbb{R}^{N_{ans} \times d}$, $W_{11} \in \mathbb{R}^{d \times 2d}$, $W_{12} \in \mathbb{R}^{1 \times d}$, $W_{13} \in \mathbb{R}^{d \times d}$, and $W_{14} \in \mathbb{R}^{d \times 2d}$ are learnable matrices. $N_{ans}$ is the number of answer categories. For simplicity, we use a function $F$ to represent the classifier. Then we have $y_{pred}^q = F(q, V_{T_q}^p)$, where $V_{T_q}^p$ denotes node representations after $T_q$ iterations of graph convolution guided by a question $q$.

For a compositional question $Q$, after the $t$-th iteration, if $t$ equals $T_Q$, the number of required iterations for a sub-question $Q_i$, we use the answer classifier to predict the answer for the sub-question $Q_i$ as $y_{pred}^{Q_i} = F(Q_i, V_{T_{Q_i}}^Q)$, where $y_{pred}^{Q_i}$ is the predicted answer distribution for a sub-question $Q_i$ in the reasoning process for the compositional question $Q$.

### 3.4. Optimization

We use the cross-entropy loss to supervise the answering process of the proposed method. For each questions $q \in Q^f$, the VQA loss is computed by $L_{vqa} = -y_{gt}^q \log(y_{pred}^q)$, where $y_{gt}^q$ is an one-hot label for question $q$. Considering we answer sub-questions $Q$ in the reasoning process for the corresponding compositional question $Q$, we have an extra VQA loss for sub-questions. Specifically, for a sub-question $Q_i$, the loss is given by $L_{vqa} = -y_{gt}^{Q_i} \log(y_{pred}^{Q_i})$.

We introduce a consistency constraint to encourage the model to answer the compositional question and the sub-questions consistently. We believe that sub-questions $Q$ should be answered more confidently and more accurately than the compositional question $Q$ because the answering of the sub-question is the basis of answering the compositional question. We use $a_{pred}^{Q_i}$ to denote the predicted probability for the ground-truth answer of a sub-question $Q_i$ in the reasoning process of the compositional question $Q$, and $a_{pred}^{Q}$ is the predicted probability for the ground-truth answer of a $Q$. The consistency constraint enforces the $a_{pred}^{Q}$ to be higher than $a_{pred}^{Q_i}$ as

$$L_{cons} = \max(\log(a_{pred}^{Q_i}) - \log(a_{pred}^{Q}), 0).$$

The consistency constraint is also used to guarantee the compatibility of $a_{pred}^{Q}$ and $a_{pred}^{Q_i}$.

The overall objective of the proposed method is given by

$$L = L_{vqa} + \lambda_{sub} L_{vqa}^{sub} + \lambda_{cons} L_{cons},$$

where $\lambda_{sub}$ and $\lambda_{cons}$ are two hyper-parameters that balance the loss terms. Considering that at the beginning of the training stage, the predictions of the model may not be accurate enough for consistency constraint to be effective, we first set $\lambda_{cons}$ in Eq. (6) as 0 and train the model for several epochs, and then train it with the full objective.

### 3.5. Implementation Details

For the visual input, the dimensions of the appearance features $d_a$, the spatial features $d_b$, and the common space $d$ are set as 2048, 96, and 512, respectively. We keep the top 48 bounding boxes ranked by confidence score as [4] with the positional information of each bounding box in the form of [top-left-x, top-left-y, bottom-right-x, bottom-right-y], normalized by the image width and height. For the language input, we first use the pre-trained GloVe [18] to initialize the 300-d embedding of words and then input them to the Bi-LSTM. The number of training epochs is set as 25 and we first train the model for 5 epochs without the consistency constraint. The hyper-parameter $\lambda_{sub}$ and $\lambda_{cons}$ are set as 0.05 and 0.01, respectively.

### 4. GQA-Sub dataset

In this section, we introduce the GQA-Sub dataset, which enables the quantitative evaluation of reasoning consistency for compositional VQA models. The GQA-Sub dataset is constructed based on the GQA dataset [11], a large-scale dataset for real-world visual reasoning and compositional question answering. The GQA contains four splits: a train split for training, a validation split and a test-dev split for validation, and a test split for online testing. We only generate sub-questions for questions of the train split and the validation split of GQA because the ground-truth scene graphs of the two splits are available.
4.1. Sub-question Generation Pipeline

In the following, we illustrate how we obtain sub-questions by decomposing compositional questions of the GQA dataset. The ground-truth scene graphs for images and functional programs for questions are used for sub-question generation. The graphs and programs are provided by the GQA. As preparations, we manually annotate patterns for each type of question to generate sub-questions.

**Language Graph Generation.** To decompose a compositional question into multiple sub-questions, we first construct a language graph to represent the known visual concepts of the question. Generally, a “known concept” can be determined by only checking a question, while an “unknown concept” cannot be determined until a corresponding image is observed. As shown in Fig. 3 (a), for a compositional question “What is the name of the animal that is watching the black television above the DVD player” with the answer “cat”, the known concepts in the question are “DVD player”, “above”, “television”, and “black” while the unknown concepts are “cat” and “watching”. We obtain the known visual concepts of the question based on its functional program and confirm these concepts indeed exist in the scene graph as shown in Fig. 3 (b). A language graph as shown in Fig. 3 (c) is further constructed to represent these concepts. In particular, we add a kind of directed edges to denote the referential relationship in the language graph. The “television” is determined by the “DVD player” and “above”, which means there may be more than one black television but only one is above the DVD player. Thus we add an edge from the “DVD player” to the “television”.

**Language Graph Traversing.** Then we traverse the language graph to generate sub-questions, as shown in Fig. 3 (c) and (d). We sequentially select visual concepts from the root node, whose in-degree is zero, to other nodes alongside the edges representing referential relationships. Specifically, we generate three kinds of sub-questions about the three kinds of visual concepts: existential questions about objects, relational questions about relations, and questions about an attribute. For a given visual concept, various types of questions can be generated. For each question type, we randomly select a pattern and generate a sub-question.

**Decoys.** We generate decoys for two kinds of questions: (1) questions that are about verifying and with an answer “no” such as “Is the dog white” for a black dog, (2) questions about choosing such as “Is the dog brown or black”. We exploit the questions of the GQA dataset and obtain a set of high-quality decoys for each concept such as “white” and “brown” for “black”.

**Balancing.** Finally, we balance the generated sub-questions to avoid language biases. We perform three times of sampling for these generated sub-questions. Please refer to the supplementary material for details of the sampling.

4.2. Dataset Analysis

After all the stages, we obtain 351, 272 and 45, 043 sub-questions for the train split and the validation split of the GQA dataset, respectively. These sub-questions form two splits: a train-sub split and a validation-sub split for the proposed GQA-Sub dataset. As shown in Fig. 4 (a), the generated sub-questions are informative and expressive. Besides, the types of our sub-questions are diverse.

We compare our sub-questions and the entailed questions, which are used to evaluate the consistency metric in the GQA dataset, in Fig. 4 (b). It is shown that the entailed questions are different from our sub-questions. Firstly, some entailed questions are paraphrases of the original question. Thus the questions do not require different strengths of reasoning ability to answer. Secondly, some entailed questions are redundant and similar. The number of entailed questions for compositional questions varies significantly. While the first question only has one entailed question, the second question has 27 entailed questions. We do not show all entailed questions for simplicity. Among the 27 entailed questions, most of them are similar or even
Figure 4. (a) Examples of the GQA-Sub dataset. For each image on the left, we list a compositional question and the sub-questions on the right side. (b) Comparisons of the entailed questions in GQA and our sub-questions. For each example, we list a compositional question and its sub-questions on the left side, show an image in the middle, and list entailed questions of the compositional question on the right.

the same. Thus these entailed questions can not be used for evaluation of the reasoning consistency.

4.3. Reasoning Consistency Score

For a reasoning model to be consistent for questions and the sub-questions, the answers to all the questions should be correct. To measure the reasoning consistency of models, a simple solution is to compute the accuracy of models for answering sub-questions. But we argue that this score only measures the ability to answer sub-questions correctly rather than the ability to answer compositional questions and sub-questions consistently. An ideal consistency score should take into account both two kinds of questions.

We devise a reasoning consistency score \( RC(k) \), which measures the ability of reasoning models to answer compositional questions that have at least \( k \) sub-questions consistently. The \( RC(k) \) is computed by

\[
RC(k) = \frac{\sum_{Q \subseteq Q, N > = k} \text{Correct}^f(Q, \{Q_i\}_{i=1}^N)}{\sum_{Q \subseteq Q, N > = k} \text{Correct}(Q) }, \tag{7}
\]

where \( Q \) denotes the set of compositional questions. \( \text{Correct}^f(\cdot, \cdot) \) and \( \text{Correct}(\cdot) \) are two indicator functions. For a question \( q \), we set \( \text{Correct}(q) = 1 \) if it is correctly answered by a model and 0 otherwise. \( \text{Correct}^f(Q, \{Q_i\}_{i=1}^N) \) indicates whether a compositional question \( Q \) and all its sub-questions \( \{Q_i\}_{i=1}^N \) are correctly answered. We set it as 1 if \( Q \) and \( \{Q_i\}_{i=1}^N \) are correctly answered and 0 otherwise. Obviously, the value of \( RC(k) \) is in the range of \([0, 1]\). A model with higher reasoning consistency score is more consistent than a model with a lower score. This definition disentangles the accuracy and the reasoning consistency of compositional VQA. A model with lower accuracy may still have a higher reasoning consistency score if most correctly answered compositional questions are consistently correctly answered.

5. Experiments

We evaluate the proposed method on the GQA dataset [11] and our GQA-Sub dataset. In this section, we first analyze the reasoning consistency of the proposed method. Then we present ablation studies to investigate the effectiveness of several critical components in it. Further, we compare the accuracy of the proposed method with state-of-the-art compositional VQA methods. Finally, qualitative results of the proposed method are shown.

5.1. Reasoning Consistency Performance

We first evaluate three state-of-the-art compositional VQA models: MAC [10], LCGN [9], and MMN [4]. The MAC and LCGN are typical holistic methods for compositional VQA while the MMN is a modular method. The results of these methods and the proposed method are listed in Tab. 1, where “Acc” is the accuracy on the validation split of the GQA dataset, “Acc (sub)” is the accuracy on the validation-sub split of the GQA-Sub dataset, and “RC(k)” is the reasoning consistency scores with different \( k \). For more fair comparisons, we trained the three methods in a

<table>
<thead>
<tr>
<th>Methods</th>
<th>Acc</th>
<th>Acc (sub)</th>
<th>RC(1)</th>
<th>RC(2)</th>
<th>RC(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language-only</td>
<td>43.86</td>
<td>41.16</td>
<td>31.60</td>
<td>18.81</td>
<td>7.31</td>
</tr>
<tr>
<td>Visual-only</td>
<td>56.63</td>
<td>53.97</td>
<td>46.63</td>
<td>28.16</td>
<td>16.26</td>
</tr>
<tr>
<td>MAC [10]</td>
<td>62.08</td>
<td>62.63</td>
<td>56.10</td>
<td>41.67</td>
<td>33.96</td>
</tr>
<tr>
<td>MAC + DA</td>
<td>61.04</td>
<td>71.42</td>
<td>65.78</td>
<td>54.09</td>
<td>43.14</td>
</tr>
<tr>
<td>LCGN [9]</td>
<td>64.16</td>
<td>63.74</td>
<td>57.37</td>
<td>44.32</td>
<td>35.09</td>
</tr>
<tr>
<td>LCGN + DA</td>
<td>64.14</td>
<td>73.46</td>
<td>68.93</td>
<td>58.94</td>
<td>50.05</td>
</tr>
<tr>
<td>MMN [4]</td>
<td>65.05</td>
<td>64.46</td>
<td>58.79</td>
<td>43.98</td>
<td>33.96</td>
</tr>
<tr>
<td>MMN + DA</td>
<td>65.65</td>
<td>74.60</td>
<td>69.59</td>
<td>57.98</td>
<td>48.17</td>
</tr>
<tr>
<td>Ours</td>
<td>66.26</td>
<td>76.02</td>
<td>71.47</td>
<td>61.94</td>
<td>52.80</td>
</tr>
</tbody>
</table>

Table 1. Results of our method and the state-of-the-art. The “DA” means the data augmentation.
data augmentation manner by using both the train split of the GQA and the train-sub of the GQA-Sub for training. We also evaluate the performance of a visual-only model and a question-only model. The former model only takes the local representations of nodes as input for answer prediction while the latter only uses the question representations. Since the original MMN uses the validation split of the GQA dataset, we retrain an MMN model by removing this split and remaining other settings to evaluate its consistency. We didn’t evaluate the transformer-based method because these methods [14, 16, 25] use the validation set of the GQA dataset in the pre-training stage.

It is shown from Tab. 1 that the state-of-the-art reasoning models suffer from inconsistency. These methods are capable of answering compositional questions with at least one sub-questions. But among these corrected answered compositional questions, only 60% are consistently answered. As the $k$ grows, the reasoning consistency score of all methods gradually declines. The language-only model and the visual-only model perform worst in all the cases, which is in line with our intuition. The proposed method outperforms the other methods in terms of reasoning consistency in all the cases. Note that our method also achieves superior performance compared with the state-of-the-art trained with data augmentation. There are two reasons for this. Firstly, the proposed method integrates the reasoning processes for the sub-questions into the reasoning process for compositional questions to encourage the model to exploit the correlations of the two kinds of questions. Secondly, the consistency constraint directly penalizes inconsistent answer predictions of models.

### 5.2. Ablation Studies

To investigate the effectiveness of several important components of our method, we train different variants of our model by ablating certain components. The results of those models are shown in Tab. 2.

We first evaluate the effectiveness of dialog-like reasoning. We train a graph reasoning model that only performs language-guided graph convolution. The model is named “Graph”. The comparisons between the “Graph” and “Ours” demonstrate that dialog-like reasoning can significantly improve reasoning consistency. We also observe that dialog-like reasoning is beneficial to the accuracy for the compositional questions. The possible reason is that the answering processes for the sub-questions can provide intermediate supervision for the answering of the compositional question. To evaluate the effectiveness of the consistency constraint, we remove the constraint of our method and obtain a model called “Ours (w/o CC)”. The model is also inferior to our full model. To further investigate the effect of the order of sub-questions, we’ve trained a model by randomly setting the required iterations of graph convolution. The comparisons between the obtained model “Ours (rand)” and our full model show the positive effect of the order of sub-questions.

Then we compare our method with a graph reasoning model trained via data-augmentation and named “Graph + DA”, since the dialog-like reasoning uses the sub-questions for training. We found that the simple data augmentation strategy can improve the accuracy for the sub-questions but doesn’t benefit the accuracy for the compositional questions considerably. The comparison between the “Graph + DA” and “Ours” demonstrates that the proposed method indeed improves the accuracy and consistency. We observe our full model significantly outperforms the model for compositional questions with at least two questions.

### 5.3. Compositional Reasoning Performance

In this part, we compare the accuracy of the proposed method on the test split of the GQA dataset with the state-of-the-art. In the implementation, we follow [4, 25] and first train the model with the “all” split of questions of the GQA dataset and fine-tune the model with the balanced split of the GQA. The results are shown in Tab. 3. We observe that the proposed method achieves competitive results compared with the state-of-the-art methods. The reason why we do not surpass all methods is that we mainly focus on maintaining reasoning consistency in compositional VQA. We

<table>
<thead>
<tr>
<th>Methods</th>
<th>Required Inputs</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottom-Up [1]</td>
<td>V+L</td>
<td>49.74</td>
</tr>
<tr>
<td>MAC [10]</td>
<td>V+L</td>
<td>54.06</td>
</tr>
<tr>
<td>NMN [2]</td>
<td>V+L+Program</td>
<td>55.70</td>
</tr>
<tr>
<td>BAN [15]</td>
<td>V+L</td>
<td>57.10</td>
</tr>
<tr>
<td>LCGN [9]</td>
<td>V+L</td>
<td>57.07</td>
</tr>
<tr>
<td>RPR [12]</td>
<td>V+L</td>
<td>59.43</td>
</tr>
<tr>
<td>LXMERT [25]</td>
<td>V+L</td>
<td>60.33</td>
</tr>
<tr>
<td>12-in-1 [16]</td>
<td>V+L</td>
<td>60.65</td>
</tr>
<tr>
<td>MMN [4]</td>
<td>V+L+Program</td>
<td>60.83</td>
</tr>
<tr>
<td>Ours</td>
<td>V+L</td>
<td>59.58</td>
</tr>
</tbody>
</table>

Table 3. Results of our method and the state-of-the-art on the test split of the GQA dataset.
only use simple graph convolutions to learn contextual visual representations for compositional reasoning and didn’t explore more sophisticated architecture or training strategies to further improve the question-answering ability. For example, transformer-based methods such as [14, 16, 25] use large corpus vision-and-language dataset in pre-training to learn multi-modality representations. The MMN [17] uses annotated structural scene graphs as the supervision for modules at the training stage.

5.4. Quantitative Results

We provide qualitative examples in Fig. 5 to show the effectiveness of our method. For a compositional question and its sub-questions about an input image, we provide visual attention maps, numbers of required iterations, and predicted answers of our method. The compositional question in the first example is about whether the woman determined by a backpack holds a racket. We observe that our method focuses on the backpack to answer the first sub-question about it. Then our method focuses on the woman and the backpack simultaneously to answer the following two sub-questions. Finally, our method figures out there is no racket held by the woman and answers “no”. Throughout the reasoning process, the method attends to critical object(s). As a result, our method answers the compositional questions and their sub-questions consistently. We believe that the attention maps and predicted answers for the sub-questions can serve as explanations for humans to be more convinced that the model relies on compositional reasoning to predict the answer rather than dataset biases. For more quantitative results, please refer to the supplementary material.

6. Conclusion and Future Work

In this paper, we have presented a dialog-like reasoning method to maintain reasoning consistency in compositional VQA. We integrate the reasoning processes for the sub-questions into the reasoning process for a compositional question like a dialog task. Based on the GQA dataset, we constructed a GQA-Sub dataset, which enables the quantitative evaluation of reasoning consistency for compositional VQA models. Experimental results show that our method can answer a compositional question and its sub-questions consistently and accurately.

In our implementation, the number of required iterations for each question is determined by using an off-the-shelf POS tagging method. The tagging method is not specifically designed for our task and may introduce noises. In the future, we will devise a policy network for the reasoning model to determine whether the current contextual representations are sufficient to answer a specific question. Thus we can use reinforcement learning to enable the model to learn to integrate reasoning processes for the sub-questions into the reasoning process for a compositional question in an end-to-end manner for better performance.

Acknowledgments

This work was supported by the Natural Science Foundation of China (NSFC) under Grants No. 62172041 and No. 62176021.
References


