Visual Question Reasoning on General Dependency Tree

Qingxing Cao Xiaodan Liang Bailing Li Guanbin Li Liang Lin*

School of Data and Computer Science, Sun Yat-sen University, China
caoqx@mail2.sysu.edu.cn, xdliang328@gmail.com, liblin3@mail2.sysu.edu.cn, liguanbin@mail.sysu.edu.cn, linliang@ieee.org

Abstract

The collaborative reasoning for understanding each image-question pair is very critical but under-explored for an interpretable Visual Question Answering (VQA) system. Although very recent works also tried the explicit compositional processes to assemble multiple sub-tasks embedded in the questions, their models heavily rely on the annotations or hand-crafted rules to obtain valid reasoning layout, leading to either heavy labor or poor performance on composition reasoning. In this paper, to enable global context reasoning for better aligning image and language domains in diverse and unrestricted cases, we propose a novel reasoning network called Adversarial Composition Modular Network (ACMN). This network comprises of two collaborative modules: i) an adversarial attention module to exploit the local visual evidence for each word parsed from the question; ii) a residual composition module to compose the previously mined evidence. Given a dependency parse tree for each question, the adversarial attention module progressively discovers salient regions of one word by densely combining regions of child word nodes in an adversarial manner. Then residual composition module merges the hidden representations of an arbitrary number of children through sum pooling and residual connection. Our ACMN is thus capable of building an interpretable VQA system that gradually dives the image cues following a question-driven reasoning route and makes global reasoning by incorporating the learned knowledge of all attention modules in a principled manner. Experiments on relational datasets demonstrate the superiority of our ACMN and visualization results show the explainable capability of our reasoning system.

Corresponding author is Liang Lin. This work was supported by the State Key Development Program under Grant 2016YFB1001004, the National Natural Science Foundation of China under Grant 61622214 and Grant 61702565, Guangdong Natural Science Foundation Project for Research Teams under Grant 2017A030312006, and was also sponsored by CCF-Tencent Open Research Fund.

1. Introduction

The task of Visual Question Answering (VQA) is to predict the correct answer given an image and a textual question. The key to this task is the capability of co-reasoning over both image and language domains. However, most of the previous methods [21, 20, 18] work more like a black-box manner, i.e., simply mapping the visual content to the textual words by crafting neural networks. The main drawback of these methods is the lack of interpreting ability to the results, i.e., why these answers are produced? Moreover, it has been shown that their accuracy may be achieved by
over-fitting the data bias in the VQA benchmark [8], and the absence of explicitly exploiting structures of text and image leads to unsatisfying performance on relational reasoning [13]. Very recently, a few pioneering works [11, 15, 26] take advantage of the structure inherently contained in text and image, which parses the question-image input into a tree or graph layout and assembles local features of nodes to predict the answer. For example, layout “more(find(ball), find(yellow))” means the module should locate the ball and the yellow object on the image first, then compose the two results to answer whether there are more balls than yellow objects. However, these methods would either rely on hand-designed rules for understanding questions or train a layout parser from scratch which suffers large decay in performance. We argue those limitations severely prohibit their application potentials in understanding general image-question pairs that may contain diverse and open-ended question styles.

To achieve a general and powerful reasoning system with the ability to enable reasoning over any dependency trees of questions rather than fixed layouts in prior works, we propose a novel Adversarial Composition Modular Network (ACMN) that designs two collaborative modules to perform tailored reasoning operations for addressing two most common word relations in the questions. As shown in Figure 11, given a specific dependency tree of each question by an off-the-shelf dependency parser, we construct a reasoning route following the dependency layout that is a tree-structure composed of clausal predicate relation and modifier relation. Our module network then alternatively performs two collaborative modules on each word node for global reasoning: 1) exploit local visual evidence of each word guided by exploiting regions of its child nodes in an adversarial way in terms of nodes with modifier relations; 2) integrate the hidden representations of child nodes via residual composition with respect to nodes with clausal predicate relation. Notably, in contrast to previous methods, our ACMN aims at a general and interpretable reasoning VQA framework that does not require any complicated handcrafted rules or ground-truth annotation to obtain a specific layout.

Specifically, we observe that the frequently used types of dependency relations can be categorize in two sets: whether the head is a predicate that describes the relation of its children (e.g. color ← is, is→nose), or a word decorated by its child (e.g. furthest→object). We refer the first set as clausal predicate relation and the second is modifier relation. Our ACMN designs adversarial attention modules for encoding modifier relation and residual composition modules for clausal predicate relations.

Firstly, for child nodes with modifier relations, we apply the adversarial attention mechanism similar to [28]. To enable effectively mining all visual evidence, we enforce each parent node explore new regions by masking out attentive regions of its child nodes at each step. More specifically, we sum up the attention maps from child nodes and mask out features weighted by the mined attention map in a soft manner. We then perform attention operation on manipulated hidden representations to extract new local visual evidence for the parent node. Secondly, for those child nodes with clausal predicate relation, our residual composition module integrates the hidden representations weighted by attention maps of its child nodes using bilinear fusion. In order to retain the information from the child nodes and deal with an arbitrary number of child nodes, the module learns a residual that will be added to the input on the sum of child nodes to modify their hidden representations. Finally, the final hidden representation of the root node goes through a multi-layer perceptron to predict the final answer.

Extensive experiments show that our model can achieve state-of-art VQA performance on both natural image VQA benchmark VQAv2 dataset and CLEVR relational dataset. And qualitative results further demonstrate the interpretable capability of our ACMN on collaborative reasoning over image and language domains.

Our contributions are summarized as follows: 1) We present a general and interpretable reasoning VQA system following a general dependency layout composed by modifier relations and clausal predicate relations. 2) a novel adversarial attention module is proposed to enforce efficient visual evidence mining for modifier relations while a residual composition module for integrating knowledge of child nodes for clausal predicate relations.

2. Related Works

Visual question answering The visual question answering task requires co-reasoning over both image and text to infer the correct answer.

The baseline method proposed in VQA dataset [2] to solve this task uses a CNN-LSTM based architecture, which consist of a convolution neural network to extract image features, and a LSTM for literal question feature encoding. It combines these two features to predict the final answer. In recent years, a large number of works followed this pipeline and have achieved substantial improvements over the baseline model. Among these works, the attention mechanism [12, 20, 24, 52, 51, 20] and the joint embedding of image and question representation [8, 15] have been widely studied. Attention mechanism learns to focus on the most discriminate sub-region instead of the whole image, provides a certain extent of reasoning to the answer. Different attention methods such as stacked attention [3] and co-attention between question and image on different levels [20] constantly improve the performance of the VQA task. As for the multi-modal joint embedding, Fukui et al. [8], Kim et al. [10] and Hedi et al. [5] exploited the compact bilinear method to fuse the embedding of im-
Figure 2: The modules in our ACMN: a) each ACMN module is composed of an adversarial attention module and a residual composition module; b) adversarial attention module; c) residual composition module. The blue arrows indicate the modifier relation and the yellow arrows represent the clausal predicate relation. Each node receives the output attention maps and the hidden features from its children, as well as the image feature and word encoding. The adversarial attention module is employed to generate a new attention map conditioned on image feature, word encoding and previously attended regions given by modifier-dependent children. The residual composition module is learned to evolve higher-level representation by integrating features of its children and local visual evidence.

3. Adversarial Composition Modular Network

3.1. Overview

Given the free-form questions $Q$ and images $I$, our proposed ACMN model learns to predict the answers $y$ and their corresponding explainable attention maps. Specifically, we first generate the structure layout given the input question $Q$ by parsing it into a tree structure using an off-the-shelf universal Stanford Parser [6]. To reduce the computational complexity, we prune the leaf-nodes that are not noun, then categorize the labels of dependency relations such as “nominal modifier” (e.g. (left, object)), “nominal
subject” (e.g. (is, color)) into two classes: the modifier relation \( M \) and the clausal predicate relation \( P \).

Our ACMN model is constituted by a set of network modules \( f \) on each word node in the layout from bottom to top. Suppose a node is \( x \), and its \( n \) children are denoted as \( \{x_1^c, x_2^c, ..., x_n^c\} \). The module \( f \) has three inputs: the image feature \( v \), the word encoding \( w \), and its children’s outputs \( [att_i^c, h_i^c] = f(x_i^c) \). It outputs a new attention region \( att_{out} \), and a hidden feature \( h_{out} \), which are generated by the adversarial attention module \( f_a \) and residual composition module \( f_h \) respectively, as shown in Figure 4.

The spatial feature \( v \) is extracted for each image via any pre-trained convolutional neural network on ImageNet (e.g. conv5 features from ResNet-152 [10] or conv4 features from ResNet-101 [10]). The word embedding vector \( w \) is obtained with a Bi-LSTM [23]. Specifically, each word in the question is first embedded as a 300 dimension vector, then the question is fed into a bidirectional LSTM. The final word embedding \( w \) is the hidden vector of Bi-LSTM at its corresponding position.

3.2. Adversarial Attention Module

Specifically, as shown in Figure 5A, we first filter the child nodes whose relation is modifier \( M \) and perform adversarial attention module on the parent node \( x \). The input attention map \( att_{in} \) of each node \( x \) is first obtained by summing attention maps \( \{att_i^c\} \) of its modifier-dependent child nodes \( x_i \). The adversarial mask is generated by subtracting \( att_{in} \) by \( 1 \) followed by a ReLU layer to keep the results non-negative. Then the mask is used to softly weight the spatial feature \( v \) via a multiplication operation. Finally, the adversarial module \( f_a \) outputs a new attention map \( att_{out} \) conditioned on the input word embedding \( w \) and weighted spatial features. We further apply Softmax to regularize the resulting attention map into the range of \([0, 1]\). The visual representation \( h' \) of the node \( x \) is then generated by the weighted sum of each grid features in \( v \) given the attended weight \( att_{out} \).

3.3. Residual Composition Module

As shown in Figure 5A, the residual composition module \( f_h \) first sums the hidden features \( \{h_i^c\} \) of its children with clausal predicate relation \( P \) into \( h_{in} \), and then concatenate \( h_{in} \) with extracted local evidence \( h' \), and finally combine with word encoding \( w \) to generate a new hidden feature \( h_{out} \). A fully connected layer is applied to project both the concatenated hidden \([h_{in}, h']\) and word encoding \( w \) feature to 2048 dimension feature vector. Then we perform element-wise multiplication on the two features, project it to 128 dimension vector, and add it with all of its children’s hidden feature \( \{h_i^c\} \) as the output hidden representation \( h_{out} \).

<table>
<thead>
<tr>
<th>Clausal Predicate Relation</th>
<th>Relation Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSUBJ</td>
<td>Nominal subject</td>
</tr>
<tr>
<td>NSUBJPASS</td>
<td>Passive nominal subject</td>
</tr>
<tr>
<td>CSUBJ</td>
<td>Clausal subject</td>
</tr>
<tr>
<td>CSUBJPASS</td>
<td>Clausal passive subject</td>
</tr>
<tr>
<td>DOBJ</td>
<td>Direct object</td>
</tr>
<tr>
<td>IOBJ</td>
<td>Indirect object</td>
</tr>
<tr>
<td>CCOMP</td>
<td>Clausal complement</td>
</tr>
<tr>
<td>XCOMP</td>
<td>Open clausal complement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modifier Relation</th>
<th>Relation Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMOD</td>
<td>Nominal modifier</td>
</tr>
<tr>
<td>AMOD</td>
<td>Adjectival modifier</td>
</tr>
<tr>
<td>NUMMOD</td>
<td>Numeric modifier</td>
</tr>
<tr>
<td>ADVMOD</td>
<td>Adverbal modifier</td>
</tr>
<tr>
<td>APPOS</td>
<td>Appositional modifier</td>
</tr>
<tr>
<td>ACL</td>
<td>Clausal modifier of noun</td>
</tr>
<tr>
<td>DET</td>
<td>Determiner</td>
</tr>
<tr>
<td>CASE</td>
<td>Prepositions, postpositions...</td>
</tr>
<tr>
<td>COMPOUND</td>
<td>Compound</td>
</tr>
</tbody>
</table>

Table 1: The two major categories of relations classified by the universal dependency parser.

3.4. The proposed ACMN model

Given the tree-structured layout of the dependency tree, our ACMN module is sequentially used on each word node to mine visual evidence and integrate features of its child nodes from bottom to top, and then predict the final answer at the root of the tree. Formally, each ACMN module can be written as:

\[
att_{in} = \sum_{(x, x_i) \in M} att_i^c, \\

h_{in} = \sum_{(x, x_i) \in P} h_i^c, \\

att_{out} = f_a(\sum_{i} att_i^c, v, w), \\

h' = att_{out} \ast v, \\

h_{out} = f_h([h_{in}, h'], w) + \sum_{i} h_i^c,
\]

Where \((x, x_i^c)\) represents the relation of node \( x \) and its child \( x_i^c \). Because the nodes with modifier relations \( M \) can modify their parent node by referring to a more specific object, we thus generate a more precise attention map as \( att_{out} \). On the other hand, the clausal predicate relation \( P \) suggests the parent node is a predicate of child nodes, we thus integrate features of child nodes to enhance the representation given the predicate word.

After propagating through all word nodes with a sequence of adversarial attention module and residual composition module, the output features of the root node \( h_{root} \) are passed through a three Multi-Layer Perceptron to predict the final answer \( y \). Our model that is stacked by a list of adversarial attention modules and residual composition mod-
ules following a tree-structured layout. Weights are share across modules with same height in order to learn different levels of semantic representation. The whole model can be trained end-to-end with only the supervision signal $y$.

3.5. Modifier Relation and Clausal Predicate Relation

A dependency-based parser is to draw directed edges from head words to dependent words in a sentence. It also labels the head-dependent relations to provide an approximation to the relationship between predicates and their arguments. One of the most widely used head-dependent relation sets is the Universal Dependencies (UD) \cite{ud}. It has a total of 42 relations that can be clustered into 9 categories. But the frequently used relations concentrate on only two of them: the core dependents of clausal predicate and the noun dependents, as shown in Figure \ref{fig:dependency_relations}. In this work, we make some small modification to noun dependents sets and refer these two kinds of relationships as clausal predicate relation $P$ and modifier relation $M$. The details of both sets are shown on Table \ref{tab:relation_sets}. For those relations that belong to neither of the two sets, we will pass both the attention map and the hidden representation to the parent nodes.

Dependants of clausal predicate relation $P$ describe syntactic roles with respect to a predicate that often describes how to compose its children. For example, in question *What color is the nose of the plane?*, word *is* is the head of *color* and *nose*, and their relations are *(is, color) = direct object* and *(is, nose) = nominal subject*. So the word *is* tells us how to compose word *color* and *nose*, such as using a function “describes(color, nose)” in a modular network \cite{modular_network}. Thus, our residual composition module learns to compose features $\{h_i\}$ of its children nodes with clausal predicate relation $P$ conditioned on current word embedding $w$ of a parent node.

The modifier relations $M$ categorize the ways words that can modify their parents. For example, the modifier relation $M$ of the question *What size is the cylinder that is left of the brown metal thing* from CLEVR dataset \cite{clevr} can be the relation *(left, brown metal thing) = nominal modifier*. The reason is that the word *left* indicates the region related to *brown metal thing* instead of *cylinder*, which is similar to “transform(left, thing)” relation in the modular network \cite{modular_network}. Thus we can obtain a modified attention map for the part node according to attention maps $\{att_i^c\}$ of its children given the current word encoding $w$ via our adversarial attention module.

4. Experiment

We validate the effectiveness and interpretation capability of our models on both two synthetic datasets (i.e., CLEVR and Sort-of-CLEVR) that mainly focus on relation reasoning and one natural dataset (i.e., VQA v2) with diverse image-question pairs in the wild.

![Figure 3: The statistic of clausal predicate relation and modifier relation in the questions of VQA v2 \cite{vqa} dataset training split and CLEVR dataset \cite{clevr} training split.](image)

4.1. Datasets

The CLEVR \cite{clevr} is a synthesized dataset with 100,000 images and 853,554 questions. The images are photo-realistic rendered images with objects of random shapes, colors, materials and sizes. The questions are generated using sets of functional programs, which consists of functions that can filter certain color, shape, or compare two objects. Thus, the reasoning routes required to answer each question can be precisely determined by its underlying function program. Unlike natural image dataset, it requires model capable of reasoning on relations to answer the questions.

The Sort-of-CLEVR \cite{sort_of_clevr} consists of synthesized images of 2D colored shapes. Each image has exactly 6 objects that can be unambiguously identified by 6 colors, and the objects have random shapes (square or circle) and positions. Each image is associated with 20 questions asking about the shape or position of a certain object, 10 of which is non-relational questions that query the object by its unambiguous colors and another 10 are relational questions that query the object with furthest or closest relation to another unambiguous colored object. It is visually simpler than the CLEVR, but also requires the model capable of relational reasoning. Since the original dataset is not released, we generate a set following their detailed description, including 9800 images for training and 200 for testing.

The VQA v2 \cite{vqa} contains 204,721 natural images from COCO \cite{coco} and 1,105,904 free-form questions. Compared with its first version \cite{vqa1}, this dataset focuses on reducing dataset biases through balanced pairs: for each question, there are pair of images which the answers to that question are different.

4.2. Implementation details

For the CLEVR dataset, we employ the same setting used in \cite{our_model1, our_model2} to extract the image feature and words encoding. We first resize all images to $224 \times 224$, then ex-
Figure 4: Two examples of the dependency trees of questions and corresponding regions attended by our model at each step on CLEVR dataset. The question is shown on the bottom. The image and dependency parse tree are shown on the left. The arrows in the dependency tree are drawn from the head words to the dependent words. The blue arrows indicate the modifier relation $M$, and the yellow arrows indicate the clausal Predicate relation $P$. The curved arrows point to the pruned leaf words that are not a noun. Thus word “there” and “have” is the root node for each example respectively. The regions with high attention weight are shown as bright areas in the images on the right. Those nodes without obvious bright region indicate our model equally attend all regions of the image, thus no specific salient regions correspond to this node.

Table 2: Comparisons in terms of question answering accuracy on the CLEVR dataset. The performance of question types \textit{Exist}, \textit{emphCount}, \textit{Compare Integer}, \textit{Query}, \textit{Compare} are reported on each column. LBP-SIG [33] and RN [22] only report total accuracy of question types \textit{Compare Integer}, \textit{Query}, \textit{Compare}, their performance on these types are merged.

<table>
<thead>
<tr>
<th>Method</th>
<th>\textit{Exist}</th>
<th>\textit{Count}</th>
<th>\textit{Equal}</th>
<th>\textit{Less}</th>
<th>\textit{More}</th>
<th>\textit{Size}</th>
<th>\textit{Color}</th>
<th>\textit{Material}</th>
<th>\textit{Shape}</th>
<th>\textit{Size}</th>
<th>\textit{Color}</th>
<th>\textit{Material}</th>
<th>\textit{Shape}</th>
<th>\textit{Overall}</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP-SIG [33]</td>
<td>79.63</td>
<td>61.27</td>
<td>80.69</td>
<td>88.59</td>
<td>97.9</td>
<td>50.1</td>
<td>53.9</td>
<td>48.6</td>
<td>51.1</td>
<td>95.5</td>
<td>95.5</td>
<td>99.0</td>
<td>80.0</td>
<td>97.0</td>
</tr>
<tr>
<td>RN [22]</td>
<td>97.8</td>
<td>90.1</td>
<td>93.6</td>
<td>88.1</td>
<td>74.0</td>
<td>84.1</td>
<td>53.9</td>
<td>48.6</td>
<td>51.1</td>
<td>95.5</td>
<td>95.5</td>
<td>99.0</td>
<td>80.0</td>
<td>97.0</td>
</tr>
<tr>
<td>N2NMN scratch [11]</td>
<td>72.7</td>
<td>55.1</td>
<td>85.1</td>
<td>79.0</td>
<td>84.1</td>
<td>89.4</td>
<td>52.5</td>
<td>85.4</td>
<td>86.7</td>
<td>78.9</td>
<td>83.7</td>
<td>90.0</td>
<td>90.0</td>
<td>95.1</td>
</tr>
<tr>
<td>N2NMN cloning expert [12]</td>
<td>83.3</td>
<td>63.3</td>
<td>68.2</td>
<td>87.2</td>
<td>85.4</td>
<td>93.1</td>
<td>84.8</td>
<td>91.5</td>
<td>90.6</td>
<td>92.6</td>
<td>82.8</td>
<td>89.6</td>
<td>90.0</td>
<td>95.1</td>
</tr>
<tr>
<td>N2NMN policy search [11]</td>
<td>85.7</td>
<td>68.5</td>
<td>73.8</td>
<td>89.7</td>
<td>87.7</td>
<td>94.8</td>
<td>93.3</td>
<td>93.1</td>
<td>89.3</td>
<td>97.8</td>
<td>94.5</td>
<td>96.6</td>
<td>95.1</td>
<td>88.6</td>
</tr>
<tr>
<td>PE-semi-9K [13]</td>
<td>89.7</td>
<td>79.7</td>
<td>85.2</td>
<td>76.1</td>
<td>77.9</td>
<td>98.8</td>
<td>98.4</td>
<td>98.1</td>
<td>97.3</td>
<td>99.8</td>
<td>98.5</td>
<td>98.9</td>
<td>98.4</td>
<td>96.9</td>
</tr>
<tr>
<td>PE-Strong [13]</td>
<td>97.7</td>
<td>92.7</td>
<td>98.0</td>
<td>99.0</td>
<td>89.8</td>
<td>92.6</td>
<td>86.45</td>
<td>92.35</td>
<td>90.65</td>
<td>98.5</td>
<td>97.44</td>
<td>94.93</td>
<td>97.37</td>
<td>89.31</td>
</tr>
<tr>
<td>\textbf{Ours}</td>
<td>94.21</td>
<td>81.37</td>
<td>75.06</td>
<td>88.23</td>
<td>81.51</td>
<td>92.61</td>
<td>86.45</td>
<td>92.35</td>
<td>90.65</td>
<td>98.5</td>
<td>97.44</td>
<td>94.93</td>
<td>97.37</td>
<td>89.31</td>
</tr>
</tbody>
</table>

Table 2: Comparisons in terms of question answering accuracy on the CLEVR dataset. The performance of question types \textit{Exist}, \textit{emphCount}, \textit{Compare Integer}, \textit{Query}, \textit{Compare} are reported on each column. LBP-SIG [33] and RN [22] only report total accuracy of question types \textit{Compare Integer}, \textit{Query}, \textit{Compare}, their performance on these types are merged.
Table 3: Comparisons in terms of question answering accuracy on the Sort-of-CLEVR dataset.

<table>
<thead>
<tr>
<th></th>
<th>Non-relational</th>
<th>Relational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours-w/o residual</td>
<td>99.05</td>
<td>93.50</td>
</tr>
<tr>
<td>Ours-DualPath</td>
<td>98.05</td>
<td>91.10</td>
</tr>
<tr>
<td>Ours-relocate</td>
<td>98.20</td>
<td>90.10</td>
</tr>
<tr>
<td>Ours-concat</td>
<td>99.10</td>
<td>91.15</td>
</tr>
<tr>
<td>CNN+MLP [22]</td>
<td>-</td>
<td>63</td>
</tr>
<tr>
<td>CNN+RN [22]</td>
<td>-</td>
<td>94.0</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>99.85</strong></td>
<td><strong>96.20</strong></td>
</tr>
</tbody>
</table>

Table 3 shows the comparisons among our model, its variants and prior works on Sort-of-CLEVR dataset. As described in [22], since the visual elements in this dataset are quite simple, a simple CNN+MLP baseline model can achieve over 94% accuracy for non-relational questions but fail for relational questions. We thus mainly focus on comparing results for relational questions. The results of two baselines (i.e. “CNN+MLP [22]” and “CNN+RN [22]”) are originally reported in [22]. The actual accuracy number for non-relational questions are not reported since both models achieve nearly 100%. We can see that our ACMN achieves superior results over two previous methods for answering relational questions that require the model has strong capability in relation reasoning rather than overfitting the dataset bias as previous works.

Figure 4 shows the resulting attention regions following the general dependency tree for the questions achieved by our ACMN. The images and the dependency parse trees are shown on the left. We highlight the regions with high attention weights, and slightly brighter the image if our model equally attends all of the regions. The first example shows that our model can first locate the “purple object”, while the phase “same material” alone does not correspond to any object, our model doesn’t focus on any specific region. Later, our model attends all object except the purple object given phase “any other things”, and “are there” locate the objects that have the same material and predict the answer “yes”. The second example illustrates the process of locating the “big thing” and the “metal sphere”. Then our model composes their visual features to answer whether these two objects have the same color.

4.3.2 Sort-of-CLEVR dataset

Table 3 shows the comparisons among our model, its variants and prior works on Sort-of-CLEVR dataset. As described in [22], since the visual elements in this dataset are quite simple, a simple CNN+MLP baseline model can achieve over 94% accuracy for non-relational questions but fail for relational questions. We thus mainly focus on comparing results for relational questions. The results of two baselines (i.e. “CNN+MLP [22]” and “CNN+RN [22]”) are originally reported in [22]. The actual accuracy number for non-relational questions are not reported since both models achieve nearly 100%. We can see that our ACMN achieves superior results over two previous methods for answering relational questions that require the model has strong capability in relation reasoning rather than overfitting the dataset bias as previous works.

Figure 5 shows the resulting attention regions following the general dependency tree for the questions achieved by our ACMN, which clearly demonstrates its promising interpretable ability. The first example locates the “gray object”, then it transforms its attention regions to its “furthest” ob-
Figure 5: Examples of parse trees and corresponding regions attended by our ACMN on Sort-of-CLEVR dataset. Same with Figure 4, the edges in dependency tree is drawn from head words to dependent words. The attended regions are highlighted for different nodes.

Figure: What is the shape of the object that is closest to the blue object?

Table 4: The question answering accuracy on VQAv2 test-dev and test-std. The “1st ensemble” and “1st single” denotes the first place method of the 2017 VQA Challenge with and without ensemble respectively.

“Ours-w/o residual” with “ours”. Furthermore, we combine the residual and dense connection to form a dual-path tree-structured network, resulting in a variant “Ours-DualPath”. This network has more parameters and exploits previous nodes’ knowledge in a more direct way. Specifically, we concatenate all previous hidden representation $h$ and use an extra fully connected layer to project them into a $256$-d feature vector. “Ours-DualPath” achieves $91.1\%$ accuracy, indicates that the extra fully connected layer hurts the performance since nodes in a general dependency parse tree may contain duplicate information. Our residual connection can handle these trivial nodes, demonstrate the effectiveness of our residual composition.

Adversarial attention module We also evaluate the results of other attention modules to demonstrate the effectiveness of our adversarial attention module. One commonly used attention module is the Relocate module in [11] which used the soft-attention encoding applied in [11], resulting in our variant “Ours-relocate”. Another option for attention module is to directly concatenate image features with the input attention maps $\text{att}^t_{in}$ instead of using an adversarial mask, that is “Ours-concat”. The proposed adversarial attention module is demonstrated to obtain better question answering performance over these two attention alternatives, benefiting from the adversarial-mask driven exploration of unseen regions.

5. Conclusion

In this paper, we propose a novel ACMN module network equipped with an adversarial attention module and a residual composition module for visual question reasoning. In contrast to previous works that rely on the annotations or hand-crafted rules to obtain valid layouts, our ACMN model can automatically perform interpretable reasoning process over a general dependency parse tree from the question, which can largely broaden its application fields. The adversarial attention module encourages the model to attend the local visual evidence for each modifier relation while the residual composition module can learn to compose representations of children for the clausal predicate relation while retaining the information flow from its indirect child nodes. Experiments show that our model outperforms previous modular networks without using any specified groundtruth layouts or complicated hand-crafted rules.
References


