Perceptual Visual Reasoning with Knowledge Propagation

Guohao Li*
ligh16@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Xin Wang†
xin_wang@tsinghua.edu.cn
Tsinghua University
Beijing, China

Wenwu Zhu†
wwzhu@tsinghua.edu.cn
Tsinghua University
Beijing, China

ABSTRACT

Visual Question Answering (VQA) aims to answer natural language questions given images, where great challenges lie in comprehensive understanding and reasoning based on the rich contents provided by both questions and images. Most existing literature on VQA fuses the image and question features together with attention mechanism to answer the questions. In order to obtain a more human-like inferential ability, there have been some preliminary module-based approaches which decompose the whole problem into modular sub-problems. However, these methods still suffer from unsolved challenges such as lacking sufficient explainability and logical inference — no doubt the gap between these preliminary studies and the real human reasoning behaviors is still extremely large. To tackle the challenges, we propose a Perceptual Visual Reasoning (PVR) model which advances one important step towards the more explainable VQA in this paper. Our proposed PVR model is a module-based approach which incorporates the concept of *logical and/or* for logic inference, introduces a richer group of perceptual modules for better logic generalization and utilizes the supervised information on each sub-module for more explainability. Knowledge propagation is therefore enabled by resorting to the modular design and supervision on sub-modules. We carry out extensive experiments with various evaluation metrics to demonstrate the superiority of the proposed PVR model against other state-of-the-art methods.

KEYWORDS

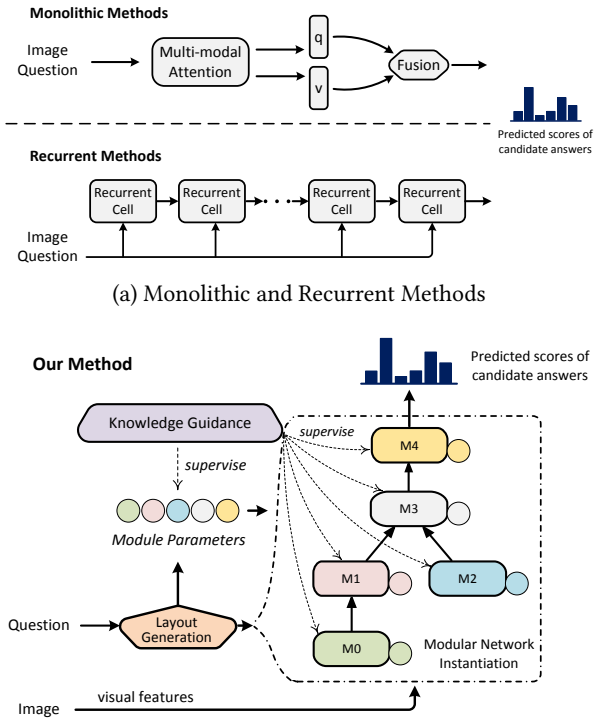
neural module networks, compositional reasoning, visual question answering

ACM Reference Format:
Guohao Li, Xin Wang, and Wenwu Zhu. 2019. Perceptual Visual Reasoning with Knowledge Propagation. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350922

*Beijing National Research Center for Information Science and Technology (BNRist).
†Corresponding Authors.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350922



(a) Monolithic and Recurrent Methods

(b) Our Proposed Method

Figure 1: Monolithic, Recurrent Approaches v.s. Our Proposed Approach

1 INTRODUCTION

Visual Question Answering (VQA) is regarded as one of the most compelling problems in both multimedia and computer vision due to its tough requirement for comprehensively understanding and reasoning based on the given cross-modal information. As a key component in visual Turing test, VQA has drawn an increasing number of research attention from both academia and industry.

One natural question is how well can a VQA algorithm answer the given visual questions compared with humans. Many existing works on VQA have been developed aiming to select candidate options as correctly as a human for multi-choice questions [33] and produce answers as similar as a human for open-domain questions [5]. One widely adopted solution so far is to locate the most relevant visual region of images based on attentions obtained from the given questions [7, 20, 28, 31], which can be referred to as monolithic methods. Another type of solution is to utilize a recurrent

cell whose output at the current step will be used as input to the next running round [11, 25] for the purpose of simulating simple sequential reasoning process. The general concepts of these two types of solutions are depicted in Figure 1 (a).

However, both types of solutions fail to model comprehensive reasoning when it comes to the scenario in which we need to locate different objects, tell their colors, and compare their properties. Although to ameliorate the ability of sophisticated reasoning, some pioneering works [4, 10, 14] propose the module network which utilizes several sub-modular networks to simulate the compositional reasoning process of human in VQA tasks, their model still suffers from two challenging problems: i) the "black box" issue (unexplainability) in deep neural networks — it is unclear whether these sub-modules are doing exactly the tasks that they are designed for; ii) the common issue faced by all existing methods — it is far from simulating human logic inferences. In general, there still exists an extremely large gap between real human reasoning and current VQA models.

In this paper, we propose a Perceptual Visual Reasoning (PVR) model to tackle these challenges and move one important step towards the more explainable VQA, as is shown in Figure 1 (b). The proposed PVR model decomposes the given questions into several sub-tasks with a hierarchical structure, chooses the most appropriate perceptual module for each sub-task, and feeds them with personalized inputs. Each perceptual module is designed to realize a certain function with a joint perception of questions words, image pixels and feedbacks from other modules in the hierarchical structure. We incorporate the *logical and/or* to simulate the process of logic inference. We design a richer group of perceptual modules than previous modular networks such that the PVR model can possess a more powerful and flexible generation ability. To further enhance the capability of explanation, guidance information tailored for each perceptual module is employed to supervise the learning process of each module. Thus the knowledge from the supervised guidance information is able to propagate from top perceptual modules to bottom ones through the connected hierarchical structure. Therefore, our proposed PVR model is capable of simulating compositional inferring procedure and producing more explainable intermediate results. Besides the VQA *Accuracy* metric that is widely used in previous works, we also compare the proposed PVR model with several state-of-the-art approaches based on five additional metrics, e.g., *Consistency* and *Grounding*, for more comprehensive evaluations.

To summarize, this paper makes the following contributions.

- We propose the perceptual visual reasoning (PVR) model, which is a general modular taxonomy design for compositional and explainable visual reasoning on real images.
- We unify visual and logic modules in a perceptual modular framework to simulate the process of human inference.
- We employ guidance information tailored for different perceptual modules as auxiliary supervisions on sub-tasks, which results in both improved model performance and better interpretability.
- We conduct extensive experiments to demonstrate the advantages of proposed PVR model against other comparative methods in terms of various evaluation metrics.

2 RELATED WORK

In this section, we review related works on VQA by dividing them into three groups.

Fusion of image and question features. The necessity of simultaneously analyzing image and text information makes VQA a multimodal task in essence [1, 5, 8, 13, 15, 18, 26]. Therefore, there have been many works on VQA trying to improve the accuracy through fusing the image and question features in many different ways such as combining CNN features of images and LSTM features of questions together [7, 16, 17, 19, 20, 22–24, 27, 29, 30, 32]. Most works make full use of the rich image information by resorting to the attention mechanism which can highlight the regions that are important for getting the correct answers [20, 23, 24, 29–31]. In particular, Yu et al. [29, 30] predict multiple attention maps for getting complementary visual features, Lu et al. [20] and Nguyen et al. [23] adopt the idea of synchronous or asynchronous co-attention schemes to bridge the correlations between images and questions. Although these fusion-based approaches have achieved remarkable performance gains, they fail to answer questions requiring complicated logical inference for real images and are still far from real human reasoning process.

Recurrent Approaches. In contrast with the above monolithic approaches, recurrent approaches [11, 25, 28] perform multiple-step reasoning to answer complicated visual questions. Each reasoning step is implemented using a general-purpose reasoning block which takes results from the previous iteration as input and outputs an updated result to its next iteration. In particular, Yang et al. [28] stack the attention layers to form multi-step visual attentions, Perez et al. [25] adopt multiple conditional normalization layers to fuse visual and textual information, Hundson et al. [11] perform multi-step reasoning using a MAC cell that can extract visual information and update the internal memory. Although these approaches can simulate sequential reasoning processes, especially in simple synthetic visual reasoning scenarios (e.g., CLEVR [13]), they do not explicitly interpret the reasoning procedure into a series of semantic sub-tasks.

Modular Approaches. Given that the procedure of visual reasoning is essentially compositional, modular approaches [3, 4, 9, 10, 14, 21] model human's visual capability through primitive modules and investigate visual reasoning problem in a modular way. The major difference between modular and recurrent approaches is that modular approaches explicitly decompose questions into semantic subtasks and assemble specialized modules to handle these sub-tasks. This compositional design enables more transparent reasoning procedures. However, existing modular approaches mainly focus on simple and easy tasks for synthetic datasets [13] and fail to perform more complex visual reasoning in real-world cases. For example, the state-of-the-art modular approaches [14, 21] on CLEVR design fine-grained dataset-specific modules, e.g., *filter_rubber_material*, which cannot generalize to the real-world scenario and is very far away from the real human reasoning behaviors.

In this work, we address these difficulties with a library of general perceptual modules suitable for the real-world scenario, each of which is capable of utilizing the supervised guidance information for semantics-to-visual perception, helping itself to decouple and cooperate with other modules.

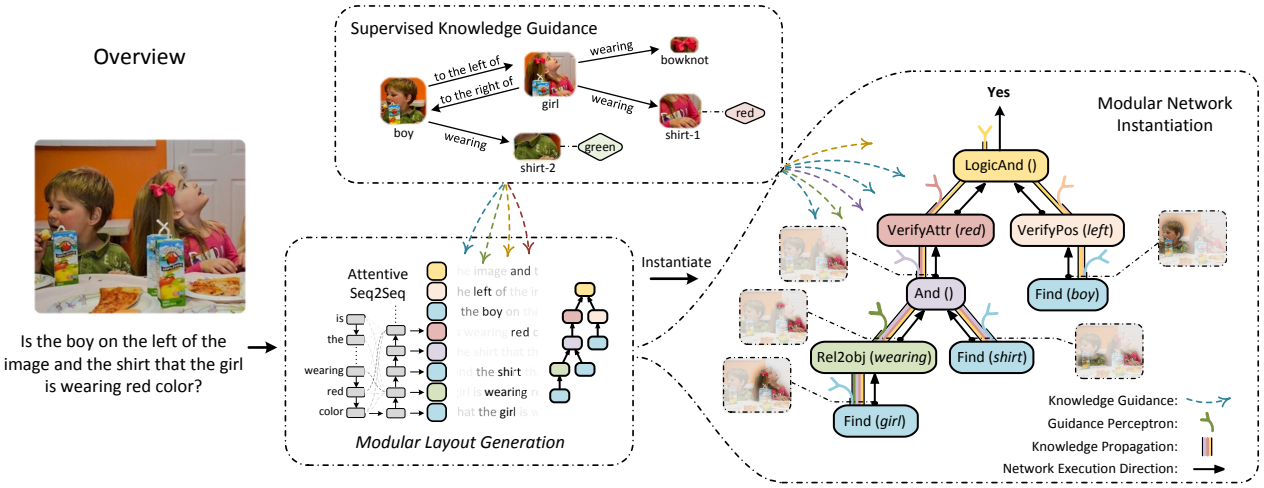


Figure 2: Overview of our proposed Perceptual Visual Reasoning (PVR) model. Given a pair of image and question, our model constructs a series of perceptual modules (depicted as small colorful rectangles) using an *Attentive Seq2Seq* model, where each module comes with a personalized textual input and visual input for semantics-to-visual perception. The perceptual modules are assembled into a tree-structured reasoning layout and then instantiated into a neural network which takes image features as input and executes the modules in a bottom-up manner until getting the final answer. We visualize the intermediate attention maps for some modules, e.g., the bottom `Find(girl)` module locates the girl in the image and the `Rel2obj(wearing)` module locates the shirt and bowknot she wears. During training time, we leverage the guidance knowledge (visual objects coordinates, attributes, and relationships) to find the appropriate inputs for different perceptual modules and supervise their learning procedures. Supervised guidance information (depicted as dashed colorful curve arrows) is designed to help semantics-to-visual perception in each module and propagated in a top-down manner. The module perception is visualized by colorful fork marks, and the knowledge propagation is visualized by rainbow strips between modules. We note that the guidance knowledge is only used in training to help module learning, and will be discarded during test time.

3 PERCEPTUAL VISUAL REASONING WITH KNOWLEDGE PROPAGATION

In this section, we detailedly discuss the Perceptual Visual Reasoning (PVR) model which targets at explainable and compositional reasoning in real-world visual scenes. Figure 2 gives an overview of our PVR model based on an example. Concretely, the PVR model consists of three components:

(1) **Modular Layout Generation.** This component generates several sub-tasks with a hierarchical structure based on the given question and assembles various functional perceptual modules with personalized inputs together to perform these sub-tasks.
(2) **Modular Network Instantiation.** This component dynamically instantiates a modular deep neural network according to the generated modular layout and execute these instantiated neural modules in a bottom-up way.
(3) **Supervised Knowledge Guidance.** This component supervises the learning procedure of the above two components in the training stage such that each perceptual module is functioning exactly as it is designed and the guidance knowledge can propagate from top modules down to bottom modules.

We first describe our module designs in Sec. 3.1 and then present the modular layout generation in Sec. 3.2, followed by the modular network instantiation and knowledge propagation among perceptual modules in Sec. 3.3. We remark that the executions of both modular layout generation and modular network instantiation are under the supervision of knowledge guidance.

3.1 Module Designs

A module can be regarded as a progressive summing up from existing reasoning states to a new reasoning state. Each module accepts zero, one or more inputs, and then generates one output. This many-to-one design establishes a hierarchical tree-based reasoning procedure and enables us to simulate the reasoning process by exploiting its compositional nature.

PVR adopts a hierarchical tree-based modular network to connect low-level visual perception with high-level logic inference, decoupling perceptual modules with specific functionalities to encourage compositional reasoning and allowing perceptual modules to accept both textual and visual inputs to improve versatility in complicated real-world visual scenarios. These designs aim at simulating human-like inferring procedure in complicated visual scenarios and producing more explainable intermediate results.

As shown in Table 1, we propose a library of modules designed for various sub-tasks ranging from lower-level visual perception to higher-level logic inference. We divide our modules into four categories based on their designed functionalities.

Table 1: The list of modules in our model. Modules are classified into four categories and operate with three kinds of data types (*att, bool, ans*). Modules that accept textual parameters (x_{txt}) are marked with ✓. Modules also use the image features (x_{vis}) and question features (x_q) in implementation. The $\odot$ operation is element-wise multiplication, vec **is an operation flattening attention maps and adding extra dimensions(e.g., max, min, average over attention maps). $P(b)$ is the probability being *true* that b represents. The** padding **operation converts logical outputs to answers.**

Module Type	Module Name	x_{txt}	Inputs	Output	Implementation Details
Attention	`Find`	✓	-	att	$a_{out} = \text{conv}_2(\text{conv}_1(x_{vis}) \odot W x_{txt})$
	`Filter[Attr|Pos]`	✓	att	att	$a_{out} = \text{minimum}(a_1, \text{conv}_2(\text{conv}_1(x_{vis}) \odot W x_{txt}))$
	`FilterDiff`		[att, att]	att	$a_{out} = \text{minimum}(a_1, a_1 - a_2)$
	`Rel2[subj|obj]`	✓	att	att	$a_{out} = \text{conv}_2(\text{conv}_1(x_{vis}) \odot W_1 \sum(a \odot x_{vis}) \odot W_2 x_{txt})$
	`And`		[att, att]	att	$a_{out} = \text{minimum}(a_1, a_2)$
Logic	`Exist`		att	bool	$b_{out} = W_b^T \text{vec}(a)$
	`Verify[Attr|Pos]`	✓	att	bool	$b_{out} = W_b^T (W_1 \sum(a \odot x_{vis}) \odot W_2 x_{txt})$
	`VerifyRel, Same`	✓	[att, att]	bool	$b_{out} = W_b^T (W_1 \sum(a_1 \odot x_{vis}) \odot W_2 \sum(a_2 \odot x_{vis}) \odot W_3 x_{txt})$
	`SameAll`	✓	att	bool	$b_{out} = W_b^T \left[W_1 \sum(a \odot x_{vis}) \odot W_2 x_{txt}, W_3 x_q\right]$
Inference	`LogicNot`		bool	bool	$P(b_{out}) = 1 - P(b)$
	`LogicAnd`		[bool, bool]	bool	$P(b_{out}) = P(b_1)P(b_2)$
	`LogicOr`		[bool, bool]	bool	$P(b_{out}) = 1 - (1 - P(b_1))(1 - P(b_2))$
Answer	`AnswerLogic`		bool	ans	$y_{out} = padding(b)$
	`QueryName`		att	ans	$y_{out} = W_y^T \left(W_1 \sum(a \odot x_{vis}) \odot W_2 x_q\right)$
	`QueryPos`		att	ans	$y_{out} = W_y^T (W_1 \sum(a \odot x_{vis}))$
	`QueryAttr, Choose`	✓	att	ans	$y_{out} = W_y^T (W_1 \sum(a \odot x_{vis}) \odot W_2 x_{txt})$
	`ChooseRel`	✓	[att, att]	ans	$y_{out} = W_y^T [W_1 \sum(a_1 \odot x_{vis}), W_2 \sum(a_2 \odot x_{vis})]$
	`Common`		[att, att]	ans	$y_{out} = W_y^T \left[W_1 \sum(a_1 \odot x_{vis}) \odot W_2 \sum(a_2 \odot x_{vis}), W_3 x_q\right]$

1) **Attention Modules** localize the most relevant visual regions. Each attention module generates an attention map over the image based on the input attention map(s) (if any) and its specified functionality. Essentially, attention modules perform sub-tasks such as *detecting visual objects* or *modeling visual relations*.

2) **Logic Modules** analyze the localized visual information and generate a logical output. They may be used for checking the existences or verifying the properties of some attended subjects.

3) **Inference Modules** perform basic logical inference, e.g., *not, and, or*. They accept logical input(s) and generate a logical output.

4) **Answer Modules** serve as the top-level module. In the VQA task, an answer module generates the final answer based on the intermediate results produced by bottom modules.

Modules are designed for specific functionalities with semantic meanings. For example, a `Find` module can localize visual regions given a name; a `Rel2subj` module would localize the *subject* visual regions given the *relation* and *object* by modeling *(subject, relation, object)* relationships; a `VerifyAttr` module would check whether the localized visual regions satisfies some specified attributes. Furthermore, our knowledge propagation strategy encourages these modules to learn its expected behavior in the training stage, which will be elaborated in Sec. 3.3.

Modules can accept textual parameters to improve perceptual versatility in the complicated real-world visual scenario. A module is usually instantiated with a textual parameter to specify a particular functionality. For example, `Find`[*cat*] and `Find`[*dog*] instantiates the same `find` module with different textual parameters, where one would localize *cat* while the other would localize *dog* in the image. Besides the textual parameters, a module can optionally take image features and question features as inputs.

Formally, a module F_m is a parameterized function that receives zero, one or more reasoning states $\{S_{m_i}\}$ and outputs one reasoning state $S_m = F_m(\{S_{m_i}\}|x_{vis}, x_q, x_{txt}; \theta_m)$, where θ_m is internal neural network parameters, x_{vis}, x_q, and x_{txt} are image features, question features and textual parameters respectively. We will explain the generation for x_q and x_{txt} in Sec. 3.2. In our implementation, reasoning state S_m can be one of the following three tensors depending on the module types: 1) An attention map over objects-based image features, denoted as a_m; 2) A representation of the probability being logical *true* or *false*, denoted as b_m; 3) A probability distribution over possible answers, denoted as y_m.

Modules are usually implemented as small differentiable neural networks. We list the implementation details in Table 1. For logical inference modules, e.g., `LogicAnd` and `LogicOr` modules, we derive formulations based on the assumption that events are independent with each other. Note that although there are various types of modules, they are all unified in our modular framework as building blocks for visual reasoning.

3.2 Modular Layout Generation

Based on the semantics and logic in the given question, we construct a hierarchical tree-based modular layout — a tree composed of modules. The bottom modules (tree leaves) are usually used for low-level visual perceptions and the top-most module (tree root) is used to generate an answer. We execute these modules in a bottom-up manner to reach the final answer.

Unlike previous works whose module's textual parameters are either hard-coded [14, 21] or set without any guidance [9, 10], the PVR model not only predicts the overall modular layout but also learns to fit each module with personalized textual parameters at

the same time. As such, PVR ensures that each module receives appropriate textual parameters and therefore generates explainable results, which is of great importance in constructing an explainable visual reasoning system.

We note that a valid tree-based modular layout can be represented as a unique sequence after a postorder traversal and vice versa. Thus a module sequence (if valid) can be reverted to a unique tree structure following the topological restrictions of modules. Therefore, the module layout prediction problem can be formulated as a sequence-to-sequence problem. We adopt the Attentive Recurrent Neural Network [6] to address the sequence-to-sequence layout prediction problem, adding an extra loss in the training stage to personalize the textual input for each module.

For a question q with K words, question words are first embedded into vectors $\{w_k\}_{k=1,\cdots,K}$. We use a multi-layer LSTM as encoder, feeding the word sequences into it to get a sequence of hidden vectors $\{h_k^{enc}\}_{k=1,\cdots,K}$ at each encoder time-step k. The final encoder states are used as question features x_q in module execution afterward. In the decoding stage, a decoder is implemented as an LSTM that has the same structure with encoder but with different network parameters. Similar to words, each module is regarded as a token and also embedded into vectors. At each decoder time-step t, the decoder fuses its hidden output and soft attended input sequence, to predict a module token $m^{(t)}$, whose embedding is fed back into the decoder at next time-step to predict the next module token.

Given the decoder output at t time-step h_t^{dec} and the question sequence encoding $\{h_k^{enc}\}_{k=1,\cdots,K}$, the decoding procedure at t time-step is described as follows:

$$u_{tk} = v^T \tanh(W_1 h_t^{dec} + W_2 h_k^{enc}), \tag{1}$$

$$\alpha_{tk}^{(txt)} = \frac{\exp(u_{tk})}{\sum_{i=1}^{K} \exp(u_{ti})}, \qquad c_t = \sum_{k=1}^{K} \alpha_{tk}^{(txt)} h_k^{enc}, \tag{2}$$

$$p(m^{(t)}|m^{(1)},\cdots,m^{(t-1)},q) = \text{softmax}(W_3 h_t^{dec} + W_4 c_t), \tag{3}$$

where the $W_{\{1,2,3,4\}}$ and v are network parameters to be learned from data; $\alpha_{tk}^{(txt)}$ is the attention weights over input quesiton sequence, and used for modeling $p(m^{(t)}|m^{(1)},\cdots,m^{(t-1)},q)$ — the probability of predicting module token $m^{(t)}$ at the t time-step. For a module layout sequence $l = [m^{(1)},\cdots,m^{(T)}]$, the probability of predicting l is $p(l|q) = \prod_{m^{(t)}\in l} p(m^{(t)}|m^{(1)},\cdots,m^{(t-1)},q)$.

We greedily select module token with the highest probability as the t_{th} module at decoder time-step t, and aggregate textual parameter from word embeddings $x_{txt}^{(t)} = \sum_{k=1}^{K} \alpha_{tk}^{(txt)} w_k$ to specify a particular functionality for the t_{th} module.

At training time, we exploit dataset annotations to fabricate the ground-truth layout l^* and ground-truth question attention $\boldsymbol{\alpha}^{(txt)*}$ as auxiliary supervise information. Given the training entry $(q, l^*, \boldsymbol{\alpha}^{(txt)*})$, we maximize the likelihood of predicting ground-truth layout l^*, as well as jointly minimize the KL-divergence between the predicted question attention $\boldsymbol{\alpha}^{(txt)}$ and ground-truth question attention $\boldsymbol{\alpha}^{(txt)*}$:

$$\mathcal{L}^{(gen)} = -p(l^*|q;\theta) + \beta \frac{1}{|\mathcal{T}|} \sum_{m_t\in\mathcal{T}} KL(\boldsymbol{\alpha}_t^{(txt)*}, \boldsymbol{\alpha}_t^{(txt)}). \tag{4}$$

where $\mathcal{T}$ is the set of modules that accept textual parameters.

This joint training loss ensures that PVR learns to infer appropriate modular layout for further reasoning from the given question, as well as feed modules with personalized textual parameters.

3.3 Modular Network Instantiation and Knowledge Propagation

We have so far designed a collection of neural modules and described how to construct an appropriate hierarchical tree-based modular layout for reasoning. In this section, we turn to *Modular Network Instantiation* where modules in the generated modular layout are dynamically assembled into a modular neural network. Each module in this modular neural network takes outputs from its child modules and feeds its output to its parent module until obtaining a final answer from the top-most module. Formally, given question q and image v, the modules eventually calculate a probability distribution over candidate answers and select the one with the highest probability as the final answer:

$$\hat{y} = \text{argmax} \;\; P(y|q,v;\theta). \tag{5}$$

Existing modular approaches optimize their modules using a single performance-oriented classification loss. In simple visual scenario (e.g., CLEVR [13]), some modular approaches succeed in both the performance and explainability [21]. However, in complicated real-world scenario [8, 12], existing methods fail to produce clear reasoning evidence and tend to sacrifice explainability for better performance. The major challenge here is that modules have difficulties in decoupling functionalities from each other when being optimized jointly.

Instead of optimizing modules with supervision merely from the ground-truth answer, PVR takes each module as an agent capable of perceiving external supervision from guidance knowledge and leverages this rich auxiliary information to help optimizing module parameters. Since these perceptual modules are organized in a tree structure, we allow the guidance knowledge tailored for each perceptual module to propagate from top modules to bottom ones through the connected hierarchical structure in a top-down manner. The guidance knowledge is expected to help modules to learn specialized decoupled functionalities.

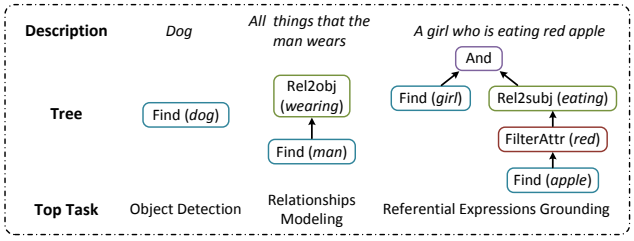


Figure 3: Examples showing sub-trees can be regarded as sub-tasks.

In an alternative view, such an approach enables multi-task learning for visual reasoning, where each sub-tree in the modular layout can be regarded as a sub-task to be executed. As the examples in Fig. 3, `Find[dog]` is a single module in the tree structure describing an *Object Detection* sub-task for "dog(s)"; The sub-tree `Rel2obj[wearing](Find[man])` designates a sub-task of locating

Table 2: Results on GQA dataset. The *Grounding* results are reported on the validation set, while other results on the test set.

Methods	Binary	Open	Consistency	Plausibility	Validity	Distribution	Grounding	Accuracy
Global Prior	42.94	16.62	51.69	74.81	88.86	93.08	-	28.90
Local Prior	47.90	16.66	54.04	84.31	84.33	13.98	-	31.24
CNN	36.05	1.74	62.40	34.84	35.78	19.99	-	17.82
LSTM	61.90	22.69	68.68	**87.30**	96.39	17.93	-	41.07
CNN+LSTM	63.26	31.80	74.57	84.25	96.02	7.46	-	46.55
BottomUp [2]	66.64	34.83	78.71	84.57	96.18	5.98	-	49.74
N2NMN [10]	72.59	40.30	83.64	84.21	96.29	5.81	58.94	55.44
MAC [11]	71.23	38.91	81.59	84.48	96.16	**5.34**	88.29	54.06
PVR (Ours)	**74.58**	**42.10**	**85.85**	84.96	**96.47**	5.64	**97.44**	**57.33**

"all things that the man wears", which involves the task of *Relationships Modeling*. Modules can be further organized for more complicated *Referential Expressions Grounding* sub-tasks, such as `And(Find[girl],Rel2subj[eating](FilterAttr[red](Find[apple]))` which refers to "a girl who is eating red apple" in the image. Beyond the *Attention* modules, the high-level modules (i.e., *Logic*, *Inference* and *Answer* modules) also specify sub-tasks, and we regard them as *VQA* sub-tasks.

However, we remark that our work is different from conventional multi-task learning problem in several aspects:

1) *Dynamic* — tasks are changing from one training instance to another training instance;

2) *Consistent* — tasks with similar routines are addressed by the same set of neural modules;

3) *Compositional* — tasks are composed by several sub-tasks;

4) *Plug and play* — multi-task supervised information is optional and can be used as long as it is available.

Here, we explain what the guidance knowledge is and how we collect them in our PVR model. For *Attention* modules, guidance information is the bounding boxes of visual regions that modules should focus on. For high-level modules (i.e., *Logic*, *Inference* and *Answer* modules), guidance information is the expected scores over candidate answers. In our implementation, we use the guidance information for all *Attention* modules and the top-most *Answer* module. We collect the guidance knowledge by referring to ground-truth scene graphs of the input images and pre-executing the modular networks in a symbolic manner. When ground-truth scene graphs are not available, an alternative is to utilize state-of-the-art scene graph parsers to collect guidance knowledge. Note that the guidance knowledge is only used at training time to help module learning, and is discarded at test time.

Next, we formulate the training procedure for *Attention* modules in detail. Suppose an image has K objects-based visual features, the k_{th} of which corresponds to a bounding-box proposal $bbox_k$. The output of an attention module is therefore an attention map $a_{1\ldots K}$ over the K bounding-box proposals $bbox_{1\ldots K}$, where $\sum_k a_k = 1$. On the other side, the guidance knowledge is L ground-truth bounding-boxes $bbox^*_{1\ldots L}$, indicating the expected attention regions. To close the gap between the module outputs and guidance knowledge, we propose to align object-based attention maps in $C * C$ grid cells in the whole image and minimize the KL-divergence between the output attentions and the guidance attentions.

The alignment operation from object bounding-box $bbox$ to $C*C$ grid cells is defined as:

$$G^{(i,j)}(bbox) = \frac{Area(Intersect(bbox, grid^{(i,j)}))}{Area(bbox)}, \tag{6}$$

where $grid^{(i,j)}$ is the i_{th} row, j_{th} column grid cell, $Intersect$ generates intersection region of two boxes, $Area$ calculates the area of one box, and $\boldsymbol{G}(bbox)$ is a $C * C$ matrix summing up to 1. We align module output attentions and guidance attentions into grids as follows:

$$\boldsymbol{\alpha}^{(vis)} = \sum_{k=1}^{K} a_k \boldsymbol{G}(bbox_k), \quad \boldsymbol{\alpha}^{(vis)*} = \frac{1}{L}\sum_{l=1}^{L} \boldsymbol{G}(bbox^*_l), \tag{7}$$

where the $\boldsymbol{\alpha}^{(vis)}$ and $\boldsymbol{\alpha}^{(vis)*}$ are $C * C$ matrices summing up to 1.

Given a training entry $(q, v, y^*, \boldsymbol{\alpha}^{(vis)*})$, we jointly minimize the softmax cross-entropy loss on the final answer scores and the KL-divergence of the output attentions and the guidance attentions in every attention module:

$$\mathcal{L}^{(exe)} = -\sum_i y^*_i \log P(y_i|q, v; \theta) + \gamma \frac{1}{|\mathcal{A}|} \sum_{m_t \in \mathcal{A}} KL(\boldsymbol{\alpha}^{(vis)*}_t, \boldsymbol{\alpha}^{(vis)}_t), \tag{8}$$

where y^* is the one-hot ground-truth answer, θ is the network parameters of modules, $\mathcal{A}$ is the set of attention modules, γ is a factor balancing the losses.

3.4 Putting All Together

Finally, the *Modular Layout Generation* and *Modular Network Instantiation* under *Supervised Knowledge Guidance* can be jointly optimized by the total loss:

$$\mathcal{L} = \mathcal{L}^{(gen)} + \eta \mathcal{L}^{(exe)}. \tag{9}$$

We close this section by pointing out that our proposed PVR model encourages explainable and compositional visual reasoning by unifying various types of modules, as well as supervising their inputs (i.e., textual parameters) and outputs (i.e., attention maps) simultaneously to regularize individual module behaviors.

4 EMPIRICAL EXPERIMENTS

We evaluate our model on the recent GQA [12] dataset for real-world visual reasoning. The dataset contains 113K real-world images and 22M multi-step compositional questions that require complex and multiple reasoning skills to answer, such as recognition, relation reasoning, logical inference, and comparisons. In this dataset, each image is annotated with a scene-graph providing a structural representation of semantics, while each question is associated with a functional program that specifies the reasoning steps to answer the question. In contrast to previous synthetic CLEVR [13] dataset, GQA dataset focuses on real-world visual reasoning with a much larger semantic space and more diverse visual concepts.

In general, the experiments demonstrate the following advantages of our proposed Perceptual Visual Reasoning (PVR) model. First, our model outperforms the state-of-the-art methods on standard VQA accuracy and exhibits superior performances on metrics reflecting model consistency and explainability (Section 4.1). Second, our model can provide clear reasoning evidence with semantic meaning at each reasoning step (Section 4.2).

To gain better insights into the PVR model, we conduct extensive ablation studies in section 4.3.

4.1 Model Performance

We use the official metrics for GQA dataset [12] to evaluate the model performance, including standard VQA *Accuracy* metrics (for open and binary questions), a *Consistency* metric measuring responses consistency across different questions, a *Grounding* metric measuring the degree to which the model grounds its reasoning in the image, *Validity* and *Plausibility* metrics measuring whether the answer is valid or reasonable for the question regardless of the image, a *Distribution* metric measuring the overall match between the true answer distribution and the predicted distribution.

We compare the proposed PVR model with several baseline methods (LSTM-CNN, etc.) and state-of-the-art methods. The bottom-up attention [2] is the winner of the 2017 VQA challenge, representing the state-of-the-art of monolithic methods. The N2NMN [10] is a modular approach that can handle real-world visual scenes. Here we use the same visual features as other methods for N2NMN for fair comparisons. Note that most state-of-the-art modular approaches [14, 21] design specialized modules for CLEVR [13] dataset, e.g., *filter_rubber_material*, thus are difficult to adapt to the real-world scenario. The MAC model [11] that performs multi-step reasoning based on a powerful recurrent cell is the state-of-the-art recurrent method for both the CLEVR [13] and GQA [12] visual reasoning datasets.

The quantitative model performances of baseline methods and ours are listed in Table 2.[1] Compared with MAC, PVR gains 3.3% improvement on VQA accuracy (3.3% and 3.2% for binary and open questions, respectively). As for the *Consistency* metric, PVR obtains even greater improvement (4.3%), demonstrating that our model evinces more consistent behavior when answering different questions, which is essentially the merits of our decoupled modular design. For the *Plausibility*, *Validity* and *Distribution* metrics, PVR also achieves competitive results, but we do not emphasize these metrics, because they are either irrelevant to visual information or heavily dependent on dataset distribution, providing limited insights for visual reasoning methods.

We further compare PVR with MAC in terms of the accuracy per question type in Table 3, where all results are reported on the validation set. Each question in GQA is associated with two types: structural type (e.g., *query*, *verify*, *logical*) and semantic type (e.g., *relation*, *attribute*, *object*). Overall, PVR outperforms MAC on every question type. For example, the significant improvement on questions involving *attribute* semantics (4.00%) indicates that our attribute-related modules (`FilterAttr`, `VerifyAttr`, etc.) effectively capture the diversity of object attributes. Although the improvement for *logical* questions is relatively small (0.95%) due to the simple implementations of logical inference modules, our model is capable of revealing explicit reasoning procedures for logical questions. After inspecting both the successful and failed cases, we believe that the appropriate modular layout together with the well-trained attention modules can benefit most types of questions.

Table 3: Accuracy per question type on GQA validation set.

Q Types	MAC [11]	PVR (ours)	Acc. Boost
All	62.07	64.47	+2.40
Open	46.65	49.29	+2.64
Binary	78.52	80.67	+2.15
Query	46.65	49.29	+2.64
Compare	66.80	69.06	+2.26
Choose	78.60	83.47	+4.87
Logical	81.23	82.18	+0.95
Verify	78.65	79.89	+1.24
Global	65.28	65.88	+0.60
Object	83.37	83.67	+0.30
Attribute	67.65	71.65	+4.00
Relation	53.59	55.36	+1.77
Category	55.44	59.16	+3.72

4.2 Explainability

We adopt the *Grounding* metric proposed in GQA [12] to quantitatively evaluate the degree to which the model grounds its reasoning in the image. As is shown in Table 2, PVR achieves 97.44% grounding score while MAC obtains 88.29% on the validation set of GQA dataset. The metric shows that PVR effectively grounds its reasoning in the image instead of making educated guess based on language priors or dataset biases.

Furthermore, we provide visualized examples in Fig. 4 to show that our proposed PVR model can perform explainable visual reasoning in real-world scenarios through enabling every module to carry out specialized functionality at each reasoning step.

4.3 Module Performances and Ablations

We conduct a more in-depth study on the GQA validation set by analyzing the performance of individual components and modules in our model, also by evaluating model ablations for obtaining more insightful results.

[1] In Table 2, we report GQA results on the test set, except that the *Grounding* results are reported on the validation set. Because the official evaluation server, which generates GQA test set results, does not support evaluating the *Grounding* score. We instead calculate the *Grounding* score on the validation set using the official evaluation scripts offline. Due to the limit on the number of submissions to the official evaluation server, we report the GQA results on the validation set in the rest of the paper.

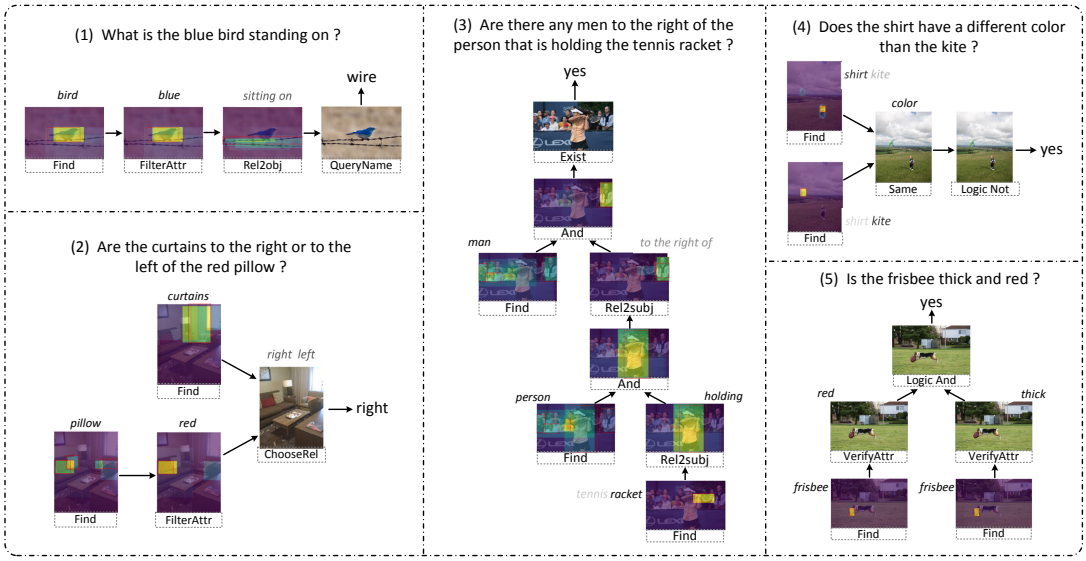


Figure 4: Real Examples visualizing the reasoning process of our PVR model. The intermediate outputs of attention modules are depicted with overlaid attention maps. We also show the textual parameters that are fed to modules.

The modular layout generation component performs nearly perfect with a 99.4% accuracy. In order to measure the degree to which modules receive proper textual parameters, we define a *Textual Attention Score* metric. For each module, the metric is defined as the sum of properly assigned question attention weights to this module. By averaging over all modules, PVR achieves a nearly perfect score of 97.8%.

Table 4: Module Performances.

Metrics	Find	Filter Attr	Rel2 subj	Verify Attr	Exist	Query Name
Att.	0.490	0.466	0.387	-	-	-
Acc.	-	-	-	78.8	82.9	44.0

We further analyze the performances of individual modules of our PVR model. For attention modules, we design a *Grid Attention Score* (Att.) metric $\sum \textbf{minimum}(\boldsymbol{\alpha}_t^{(vis)*}, \boldsymbol{\alpha}_t^{(vis)})$ to measure the degree to which the attention module t attends to expected regions, where $\boldsymbol{\alpha}_t^{(vis)*}, \boldsymbol{\alpha}_t^{(vis)}$ are $C * C$ grid cells as defined in Sec. 3.3, the **minimum** is an element-wise minimum operation. For high-level modules, we use the accuracy (Acc.) metric. We list the performances of several modules in Table 4.

As expected, increasing task complexity leads to the continual performance drops for `Find`, `FilterAttr` and `Rel2subj` modules. Furthermore, PVR performs well on tasks such as checking existences or verifying attributes while struggling on more difficult tasks, e.g., inferring name from visual attention.

To gain further insights into the contributions of PVR, we conduct several ablation studies. We perform experiments using a simplified library of modules similar to [10], including `Find`, `Filter`, `Relate`, `Exist` and `Query` modules. This variant leads to a 4.37% drop in the accuracy, 8.47% drop in the consistency score and 38.7% drop in the grounding score. The ablations demonstrate that compared with the simplified version, the hierarchical modular design in PVR is effective in learning decoupled module functionalities from data. In another ablation experiment, removing the guidance knowledge leads to 0.82% drop in the accuracy, 1.76% drop in the consistency score and 24.6% drop in the grounding score, which reveals that the supervised guidance knowledge can benefit both the performances and explainability of our PVR model.

5 CONCLUSIONS

In this work, we propose the Perceptual Visual Reasoning (PVR) model, a module-based approach for real-world visual reasoning. Our PVR model solves the real-world visual reasoning problem by decomposing a given question into several correlated sub-tasks and progressively solving these sub-tasks. We design a library of hierarchical neural modules to bridge low-level visual perception with high-level logic inference in a unified framework, where each module is capable of perceiving external guidance information to learn its specialized functionality. These design choices encourage an explainable and compositional reasoning process. We validate the superiority of our model in both performances and explainability, showing that PVR is able to outperform state-of-the-art models on the GQA dataset and produce transparent, explainable intermediate results in the reasoning process.

ACKNOWLEDGMENTS

This research is supported by National Program on Key Basic Research Project No.2015CB352300, National Natural Science Foundation of China Major Project No.U1611461, China Postdoctoral Science Foundation No.BX201700136 and Shenzhen Nanshan District Ling-Hang Team Grant under No.LHTD20170005.

REFERENCES

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4971–4980.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6077–6086.

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705* (2016).

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

[5] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*. 2425–2433.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

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

[8] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6904–6913.

[9] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2018. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 53–69.

[10] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. 804–813.

[11] Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067* (2018).

[12] Drew A Hudson and Christopher D Manning. 2019. GQA: A New Dataset for Compositional Question Answering over Real-World Images. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[13] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2901–2910.

[14] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*. 2989–2998.

[15] Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In *Proceedings of the IEEE International Conference on Computer Vision*. 1965–1973.

[16] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: Understanding data visualizations via question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5648–5656.

[17] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2016. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325* (2016).

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

[19] Junwei Liang, Lu Jiang, Liangliang Cao, Li-Jia Li, and Alexander G Hauptmann. 2018. Focal visual-text attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6135–6143.

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

[21] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. 2018. Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4942–4950.

[22] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 299–307.

[23] Duy-Kien Nguyen and Takayuki Okatani. 2018. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6087–6096.

[24] Badri Patro and Vinay P Namboodiri. 2018. Differential attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7680–7688.

[25] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[26] Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Exploring models and data for image question answering. In *Advances in neural information processing systems*. 2953–2961.

[27] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. 2018. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4223–4232.

[28] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 21–29.

[29] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Proceedings of the IEEE international conference on computer vision*. 1821–1830.

[30] Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. 2018. Beyond bilinear: generalized multimodal factorized high-order pooling for visual question answering. *IEEE transactions on neural networks and learning systems* 99 (2018), 1–13.

[31] Yiyi Zhou, Rongrong Ji, Jinsong Su, Yongjian Wu, and Yunsheng Wu. 2017. More Than An Answer: Neural Pivot Network for Visual Qestion Answering. In *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 681–689.

[32] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. 2017. Structured attentions for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. 1291–1300.

[33] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4995–5004.