

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

Visual Question Answering with Question Understanding and Reasoning

Anonymous ICCV submission

Paper ID ****

Abstract

Traditional techniques for visual question answering (VQA) are mostly end-to-end neural network based, which often perform poorly (e.g. inefficiency and low accuracy) due to lack of question understanding and necessary reasoning. To overcome the weaknesses, we propose a comprehensive approach with following key features. (1) It represents inputs, i.e. image Img and question Q_{nl} as entity-attribute graph and query graph, respectively, and employs graph matching to find answers; (2) it leverages reinforcement learning based model to identify correct query graph, and a set of policies that are used to guide visual tasks, based on Q_{nl} ; and (3) it trains a classifier and reasons missing values that are crucial for question answering with the classifier. With these features, our approach can not only conduct visual tasks more efficiently, but also answer questions with higher accuracy; better still, our approach also works in an end-to-end manner, owing to seamless integration of our techniques. To evaluate the performance of our approach, we conduct empirical studies on our VQA data set (Soccer-VQA) and Visual-Genome data set [1], and show that our approach outperforms the state-of-the-art method in both efficiency and accuracy.

1. Introduction

Visual Question Answering (VQA), the problem of automatically and efficiently answering questions about visual content, has attracted a wide range of attention, since it has a variety of applications in e.g. image captioning, surveillance video understanding, visual commentator robot, etc. Though important, the VQA problem brings a rich set of challenges spanning various domains such as computer vision, natural language processing, knowledge representation, and reasoning. In recent years, VQA has achieved significant progress, owing to the development of deep architectures suited for this task and the creation of large VQA datasets to train these models. However, a number of studies [29, 11] also pointed out that despite recent progress, today's neural network based approaches demon-

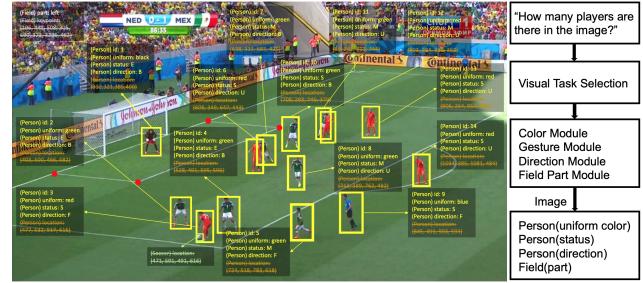


Figure 1: The image is about soccer match, where each person object is associated with attributes: id, uniform color, status (Standing, Moving, Expansion), direction (Backing, Facing, N/A), as well as location, and the soccer object is attributed with location. However, not all informations are necessary in answering one question, visual tasks are selected to achieve acquiring relevant information.

strate a few weaknesses, which greatly hinders its further development. First of all, existing techniques train deep neural networks to predict answers, where image-question pairs are jointly embedded as training data, following this way, the correlation between the question and the image is ignored, which may lead to difficulty in balancing accuracy and efficiency. Secondly, deep neural networks works as “black boxes”, it is very hard to identify the causal relations between network design and system performance, not to mention ensuring acceptable performance. Lastly, due to lack of reasoning capability, existing techniques show poor performance when answering real-life questions, that are often open-ended and require necessary reasoning.

To address the issues mentioned above, a method, that finds answers based on the understanding of the questions and necessary reasoning, is required.

Example 1: Figure 1 depicts an image about a soccer match, where each object is associated with a set of attributes. A typical question may ask “How many players are there in the image?”. Though simple, it is a challenging task to efficiently answer the query, since (1) traditionally, it often takes time to extract as much information as possible from the given image, and then answer the questions; while, only question related objects are needed; (2) infor-

108 mation extracted from image alone is often insufficient to
109 answer questions, hence missing values that are crucial for
110 question answering should be inferred by certain reasoning
111 techniques.
112

113 To tackle the issues, one may (1) model input *i.e.* image
114 and question, with graph structures that can capture information
115 from both image and question well, and ease question
116 understanding and reasoning; (2) follow the work-flow
117 given on the right hand side of Fig. 1 to identify correct
118 graph representation of the question, and a set of policies
119 that are closely related to the question and used to guide
120 forthcoming visual tasks; and (3) infer values *e.g.* “role”
121 of person objects (referee, goalkeeper or player) using well
122 trained classification model. □

123 This example suggests that we address the VQA problem
124 by modeling inputs as graphs, leveraging techniques to
125 guide question translation, visual processing, and do reasoning.
126 While to do this, two critical questions have to be
127 answered. (1) How to understand questions and carry out
128 question related visual tasks? (2) How to infer crucial information
129 to assist question answering?

131 **Contributions.** In contrast to a majority of deep neural
132 networks based VQA techniques, which not only overlooks
133 correlation between questions and images, but also lacks of
134 necessary reasoning, we provide a novel approach that
135 integrates question understanding and reasoning, for the VQA
136 problem. The main contributions of the paper are as follow.

137 (1) We model images and questions as graphs, and propose
138 to answer visual questions with graph matching. This new
139 representation and answering scheme constitute the
140 base of our techniques.

141 (2) We introduce a method to guide question translation
142 and visual processing based on reinforcement learning.
143 That is, given a question, our method can identify its correct
144 graph representation, and a set of policies to guide visual
145 tasks in a more efficient manner.

146 (3) We provide a method to infer missing values to
147 answer questions. The reasoning task relies on a classifier, that
148 is generated by offline training with supervised learning.

149 (4) We conduct extensive experimental studies to verify
150 the performance of our method on both our curated VQA
151 dataset and a public VQA dataset. We find that X, Y, and Z.

153 2. Related Work

154 We categorize related work into following three parts.

155 *Visual query answering.* Current VQA approaches are
156 mainly based on deep neural works. [32] introduces a
157 spatial attention mechanism similar to the model for
158 image captioning. Instead of computing the attention vector
159 iteratively, [30] obtains a global spatial attention weights

160 vector which is then used to generate a new image embedding.
161 [31] proposed to model the visual attention as a multivariate
162 distribution over a grid-structured conditional random field on image regions, thus multiple regions can
163 be selected at the same time. This attention mechanism
164 is called structured multivariate attention in [31]. There
165 has been many other improvements to the standard deep
166 learning method, *e.g.* [9] utilized Multimodal Compact Bi-
167 linear (MCB) pooling to efficiently and expressively com-
168 bine multimodal features. Another interesting idea is the
169 implementation of Neural Module Networks [3, 12], which
170 decomposes queries into their linguistic substructures, and
171 uses these structures to dynamically instantiate module net-
172 works. [24] proposed to build graph over scene objects and
173 question words. The visual graph is similar to ours, but the
174 query graph differs. Note that the method [24] proposed is
175 still a neural network based method as the structured repre-
176 sentations are fed into a recurrent network to form the final
177 embedding and the answer is again inferred by a classifier.

178 *Environment Exploration in Visual Field.* Reinforcement
179 driven information acquisition is not only focusing at
180 games [27, 26, 14] but also wildly applied in traditional
181 vision domain. [19] implement reinforcement learning in
182 visual object detection, by presenting a novel sequential
183 models which accumulate evidence collected at a small set
184 of image locations to detect visual objects effectively. [10]
185 forms the facial detection problem into an adaptive learn-
186 ing process, by designing an approximate optimal con-
187 trol framework, based on reinforcement learning to actively
188 search a visual field. [20] introduced a novel recurrent neu-
189 ral network model which is capable to extract information
190 from an image or video by adaptive selection for a sequence
191 of regions or locations. [33] introduced reinforcement learn-
192 ing in the task of target-driven visual navigation. Other
193 works aims to achieving based on algorithms [15, 2].

194 In visual and language domain, relevant work like [23]
195 achieves image captioning with Embedding Reward. Re-
196inforcement learning preserves ability to effectively select
197 preferred actions, which benefits the system decomposing
198 the problem into a few sub-tasks.

199 *Graph-based visual understanding.* [25] proposes a frame-
200 work to understand events and answer user queries, where
201 underlying knowledge is represented by a spatial-temporal-
202 causal And-Or graph (S/T/C-AOG).

203 3. Overview of the Approach

204 We start from representations of images and questions,
205 followed by the overview of our approach.

206 3.1. Representation of Images and Questions

207 We use the same representations as [29]. To make the
208 paper self-contained, we cite them as follows (rephrased).

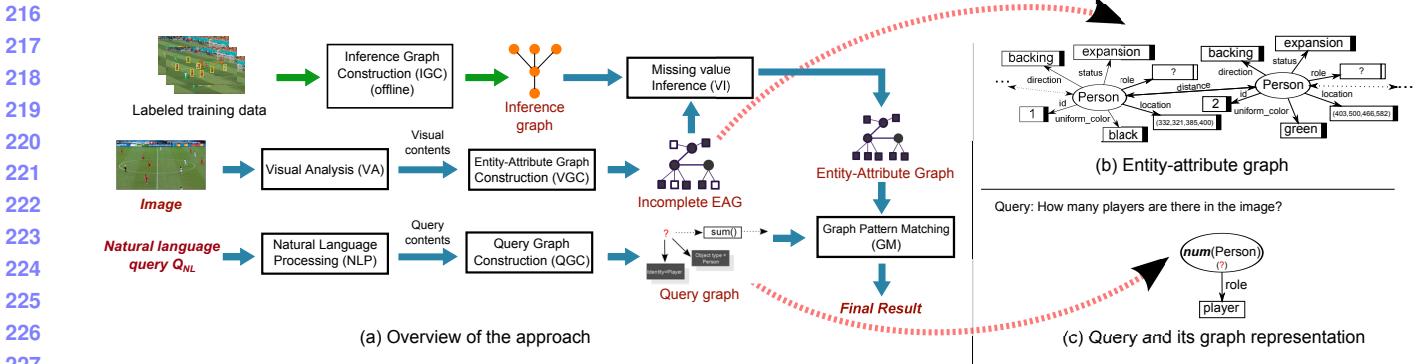


Figure 2: Overview of our approach, and graph-based representation of images and questions

Entity-Attribute Graphs. Entities are typically defined as objects or concepts that exist in the real world. An entity often carries attributes, that describe features of the entity.

Assume a set \mathcal{E} of entities, a set \mathcal{D} of values, a set \mathcal{P} of predicates indicating attributes of entities and a set Θ of types. Each entity e in \mathcal{E} has a *unique ID* and a *type* in Θ .

An *entity-attribute* graph, denoted as EAG, is a set of triples $t = (s, p, o)$, where *subject* s is an entity in \mathcal{E} , p is a *predicate* in \mathcal{P} , and *object* o is either an entity in \mathcal{E} or a value d in \mathcal{D} . It can be represented as a directed edge-labeled graph $G_{EA} = (V, E)$, such that (a) V is the set of nodes consisting of s and o for each triple $t = (s, p, o)$; and (b) there is an edge in E from s to o labeled by p for each triple $t = (s, p, o)$.

An image can be represented as an EAG with detected objects along with their detected attributes, and relationships among objects. This can be achieved via a few visual tasks. While EAG generated directly after image processing is often incomplete, *i.e.* it may miss some crucial information to answer queries. We hence refer to *entity-attribute graphs* with incomplete information as *incomplete entity-attribute graphs*, and associate nodes with white rectangles, to indicate the missing value of an entity or attribute in EAG. Figure 1(b) is an *incomplete entity-attribute graph*, in which square nodes representing person roles are associated with white rectangle.

Query Graphs. A query graph $Q(u_o)$ is a set of triples (s_Q, p_Q, o_Q) , where s_Q is either a variable z or a function $f(z)$ taking z as parameter, o_Q is one of a value d or z or $f(z)$, and p_Q is a predicate in \mathcal{P} . Here function $f(z)$ is defined by users, and variable z has one of three forms: (a) *entity variable* y , to map to an entity, (b) *value variable* $y*$, to map to a value, and (c) *wildcard* $_y$, to map to an entity. Here s_Q can be either y or $_y$, while o_Q can be y , $y*$ or $_y$. Entity variables and wildcard carry a *type*, denoting the type of entities they represent.

A query graph can also be represented as a graph such that two variables are represented as the same node if they

have the same name of y , $y*$ or $_y$; similarly for functions $f(z)$ and values d . We assume *w.l.o.g.* that $Q(x)$ is connected, *i.e.* there exists an undirected path between u_o and each node in $Q(u_o)$. In particular, u_o is a designated node in $Q(u_o)$, denoting the query focus and labeled by “?”. Take Fig. 2(c) as example. It depicts a query graph that is generated from query “*How many players are there in the image?*”. Note that the “query focus” u_o carries a function $num()$ that calculates the total number of *person* entities with *role* “player”.

3.2. Approach Overview

Figure 2(a) presents the overview of our approach. In a nutshell, our approach takes an image and a natural language question Q_{NL} as input, and answers questions with seven modules as following. (1) Upon receiving a question Q_{NL} , module QVS identifies a set of visual tasks that are query-oriented and category of the query graph that corresponds to the input question, and passes tasks and category to modules VTP and QGC, respectively. (2) Guided by the list of tasks, module VTP conducts visual tasks over the input image, and returns identified objects along with their attributes to module VGC. (3) Module VGC constructs an entity-attribute graph G_{EA} , by using identified objects and their attributes. Note that G_{EA} may be incomplete and hence unable to answer questions. (4) When G_{EA} is incomplete, module VI infers missing value with a classifier G_I , denoted as *inference graph*, and produces an updated EAG for question answering. (5) Module QGC takes category of the query graph as input, and generates a query graph $Q(u_o)$. (6) After $Q(u_o)$ and G_{EA} are generated, module GM is invoked for matching computation, and returns final result. (7) In contrast to online computation that are processed by above modules, the module IGC constructs *inference graphs* using labeled training data, offline.

As some modules employ existing techniques, to emphasize our novelty, we will elaborate modules VA and VGC in Section ??, modules IGC and VI in Section ??, and module GM in Section ?? with more details.

324

4. Question Oriented Visual Tasks

325
326
327

In this section, we introduce how we do visual tasks that are in connection with questions.

328
329

4.1. Visual Processing

330
331
332
333
334
335
336
337
338

In our approach, we build a structure which selects sub-tasks to form a policy which is based on queries. For instance, with the input *which is the defending team?*, the system first predicts the corresponding visual action sequence, which are *Human Module*, *Gesture Module*, *Direction Module*, *Soccer Module*, *Color Blob Module*, *Field Part Module* and *Graph Indicator*. Guided by such sequence, the image features are then extracted by operating relevant vision tasks. An overview is shown in Figure 3.

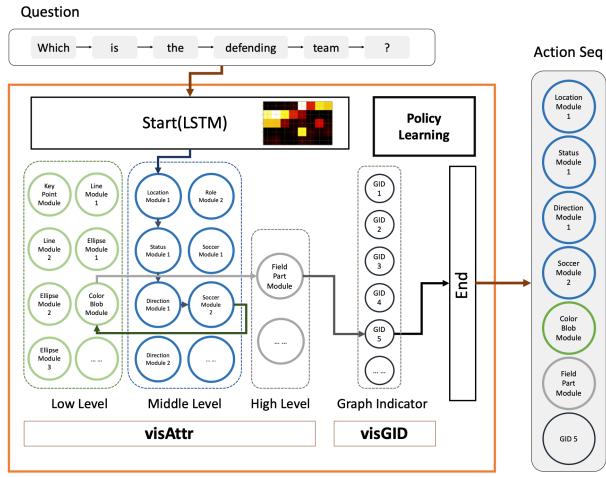


Figure 3: Visual Processing Strategy

356
357

4.1.1 Multi-layer LSTM with Attention

362
363
364
365
366
367
368
369
370
371

The task here is to predict the most suitable action modules sequence a by given questions Q and preference pre . We form the problem of seeking effective answering strategy of question Q and preference pre as a sequence-to-sequence learning problem with attention mechanism. Inspired by [5], we input word feature of questions w_i^q , $i \in \|Q\|$ into a LSTM network which is regarded as an encoder and output h_i as the hidden state for i th word in the question. By adding soft attention, the context vector c_i is calculated by the following equations.

372
373
374
375
376
377

$$c_i = \sum_{j=1}^{\|Q\|} a_{ij} h_j \quad (1)$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{\|Q\|} \exp(e_{ik})} \quad (2)$$

$$e_{ij} = a(s_{j-1}, h_i) \quad (3)$$

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

where h_i and s_j are hidden states of encoder and decoder stage, respectively. Here a_{ij} is the attention weights, with higher a_{ij} in (i, j) pair, the more attention will pay in this correlation, thus the j th output action a_i module will be more influenced by the i th input word w_i^q in question. The decode part is similar as traditional recurrent neural networks (RNN). The following steps demonstrates decoding to get the joint distribution of action module sequence $A = [a_1, \dots, a_t]$.

$$p(A|Q) = \prod_{t \in \|A\|} p(a_t | \{a_1, \dots, a_t\}, c_i, Q) \quad (4)$$

$$p(a_t | \{a_1, \dots, a_t\}, c_i, Q) = g(a_{t-1}, s_t, c_i, Q) \quad (5)$$

$$s_t = f(s_{t-1}, a_{t-1}, c_i) \quad (6)$$

where $g(\cdot)$ is a nonlinear function which outputs the probability of action module a_t . The probability distribution $p(A|Q)$ is used to predict a maximum probability action module sequence by beam search during testing time.

Guided by this action sequence $[a_1, a_2, a_3, \dots, a_n]$, actions are selected from the following visual task pool, and comes into next session, visual task selection (VTS).

4.1.2 Visual Task Selection

VTS is guided by the question feature, selecting target visual tasks from the visual task pool by Monte Carlo learning. The task pool is demonstrated in Table 1.

	Level	Sub-task	Descriptions
visAttr	Low	Keypoint Module	To get detailed information of the soccer field $F_{keypoint}$.
		Line Module	
		Ellipse Module	
	Middle	Color Blob Module	To detect blobs of different colors among a region (whole soccer field or a small bounding box).
		Location Module	To detect the location of the person $P_{location}$.
	High	Status Module	To get the person's gesture P_{status} (standing, moving, expansion).
		Direction Module	To get whether a person is facing the goal or not $P_{direction}$.
		Uniform Module	To get the uniform color of the person $P_{uniform}$.
visGID	High	Soccer Module	To detect the location of the soccer $S_{location}$.
		Field Part Module	To detect which part of the soccer field is there in the image F_{part} .
visGID	Graph ID Indicator		To indicate which type of graph will be used in the following process.

Table 1: Visual Task Pool

The pool is constructed by two parts, the first one is $visAttr$ which aims to discover the attribute of people, soccer, field and scene [29], while the other one $visGID$ is an indicator showing the current question belongs to which graph type. There are three levels of $visAttr$: low, middle and high, which represents different degree of the vision tasks.



Figure 4: Person status.

For each vision task, the approach is not fixed, one task can be achieved by different methods with variance in time and accuracy. For instance, object detection based methods, like Faster R-CNN [22], R-FCN [7], SSD [17] or skeleton keypoints detection based method like [6] and [28] can be implemented as a set of status module methods, because all of them is able to localize and distinguish people who are moving, standing or with an expansion gesture (Figure 4). Thus, the system is not only able to determine whether resembles a vision task module into action module sequence $[a_1, a_2, a_3, \dots, a_n]$, it can also select one specific approach under a vision task module, based on question and preference. For the preference, it is clarified in the next section.

4.1.3 Time and Accuracy Term

To better balance the accuracy and inference time for a given application, we proposed time and accuracy terms in loss function during training process.

$$L_{\tau\alpha}(\theta) = \ell(\theta, A|I, Q) + \gamma \sum_{i \in \|A\|} \alpha(a_i) + (1 - \gamma) \sum_{i \in \|A\|} \tau(a_i) \quad (7)$$

where $\tau(\cdot)$ and $\alpha(\cdot)$ represent the pre-tested inference time and inference accuracy, action module sequence A samples from joint distribution $p(A|Q)$, and here $\ell(\cdot)$ is the softmax loss over the predict score. For the preference term γ , it ranges from 0 to 1, which represents the preference over time and accuracy.

4.1.4 Monte Carlo Methods

The task now becomes a policy learning problem. Given a question and preference, output a policy containing a sequence of actions $[a_1, a_2, a_3, \dots, a_n]$. There is no ground truth for each steps, but only a final reward indicates that whether the prediction result is correct based on current policy. We involve the concept of Monte Carlo Methods to learn the policy which guides the vision tasks, and such policy network requires an extra reward value in loss.

$$L_{policy}(\theta) = \sum_{i \in \|A\|} \log \pi(a_i|Q, \theta) \ell(Q, A) \quad (8)$$

where a_i is the action will take, based on current status. $\pi(\cdot)$ is the policy function that maps status to actions,

here, the policy is the probability of outputing next action module a_i based on current status. And $\ell(\cdot)$ here is the softmax loss based on the whole action module sequence $[a_1, a_2, a_3, \dots, a_n]$. Since all actions are discrete, which leads to non-differentiable, and back-propagation cannot be used. Policy gradient [16] is used here during training. The object function now becomes the combination of policy gradient loss $L_{policy}(\theta)$ with the time-accuracy-balanced loss $L_{\tau\alpha}$, and optimize it by backpropagation for $L_{\tau\alpha}$, while policy gradient for $L_{policy}(\theta)$.

4.2. Construction of EAG

After objects that are related to questions are identified, we construct a graph structure, denoted as EAG, along the same line as [29].

Example 2: ADD AN EXAMPLE TO ILLUSTRATE PROGRESS IF NECESSARY! \square

5. Reasoning

According to our observation, an *incomplete* EAG isn't well satisfying of answering the question because of the insufficient attributes. To infer the hidden attributes, an inference graph is constructed accordingly. we briefly introduce the construction below.

5.1. Construction of Inference Graph

To take advantage of the prior information and increase the generalization ability of the proposed model, our inference graph is constructed using Bayesian network. Mathematically, Bayesian network [8] can be described by a pair $\mathcal{B} = < \mathcal{G}, \Theta_{\mathcal{G}} >$. Here, the notation \mathcal{G} is a directed acyclic graph, of which the i -th vertex corresponds to a random variable X_i , and the edge between two connected vertexes indicates the dependency. Additionally, the second item $\Theta_{\mathcal{G}}$ is a set of parameters used to quantify the dependencies in \mathcal{G} . Denoted by $\text{Pa}(X_i)$ the attributes of the parents of X_i , the parameter of X_i is represented by $\theta_{X_i|\text{Pa}(X_i)} = P_{\mathcal{B}}(X_i|\text{Pa}(X_i))$. With the notations above, the joint probability distribution of Bayesian network is given by:

$$P_{\mathcal{B}}(X_1, \dots, X_n) = \prod_{i=1}^n P_{\mathcal{B}}(X_i|\text{Pa}(X_i)) = \prod_{i=1}^n \theta_{X_i|\text{Pa}(X_i)} \quad (9)$$

In our inference graph, the role of Bayesian network is to predict the object class when given the attributes $\{X_i\}_{i=1}^n$ as input. In the sense of probability, the object class is also a variable [13]. Defined by $X_0 = Y$ the class variable, the network now has one extra vertex X_0 . In order to infer the class attribute, and according to the Bayesian rule, our problem becomes:

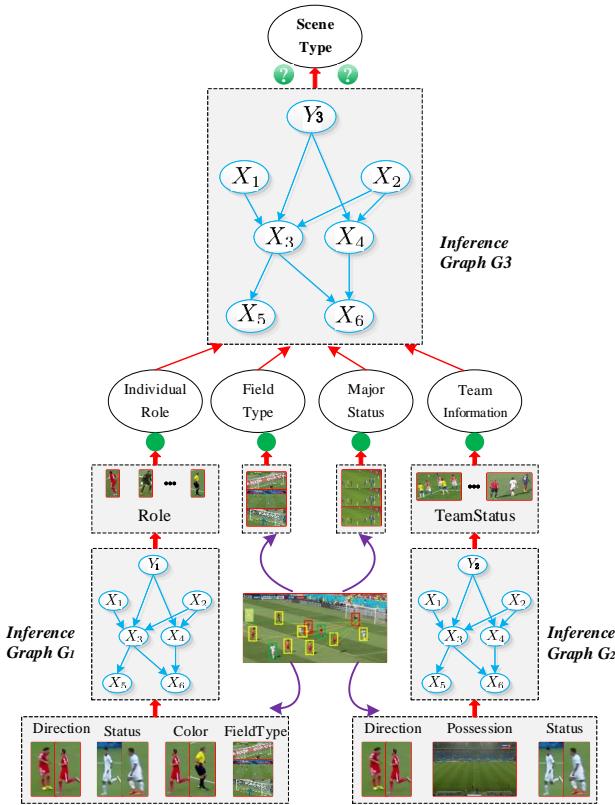


Figure 5: Schematic diagram of inference graph.

$$P_{\mathcal{B}}(Y|X) = \frac{P_{\mathcal{B}}(Y)P_{\mathcal{B}}(X|Y)}{P_{\mathcal{B}}(X)} \quad (10)$$

$$= \frac{\theta_{Y|\text{Pa}(X_0)} \prod_{i=1}^n \theta_{X_i|Y, \text{Pa}(X_i)}}{\sum_{y' \in \mathcal{Y}} \theta_{y'|\text{Pa}(X_0)} \prod_{i=1}^n \theta_{X_i|y', \text{Pa}(X_i)}}$$

where \mathcal{Y} is the set of classes.

5.2. Learning the Structure of Inference Graph

In the context of Naïve Bayes, the structure of $P_{\mathcal{B}}(Y|X)$ is simplified by taking the class variable as the root, and all attributes are conditionally independent when taking the class as a condition [21]. As a consequence, the attribute class can be explicitly inferred by:

$$P_{\mathcal{B}}(Y|X) = c \cdot \theta_Y \prod_{i=1}^n \theta_{X_i|Y} \quad (11)$$

where c is a scale factor that makes the calculation being a distribution: $c = \sum_{y' \in \mathcal{Y}} \theta_{y'} \prod_{i=1}^n \theta_{X_i|y'}$.

One can observe from Eq.(11) that Naïve Bayes simplifies the complexity of Bayesian network. As can be validated by the experimental results, the simple model works excellently to our problem.

Figure 5 summarizes the processes of our inference graph, where three graphs are constructed according to the

tasks involved. First, the role of a candidate is inferred from G_1 , in which four different kinds of features are extracted from the scene image. Then, the team status is inferred through the second inference graph G_2 , but with different features as input. Next, we use the inferred information, along with the other information can be directly detected from the scene image, to infer the information of the whole scene. The scene information is then fed into the incomplete EAG so that a complete EAG can be obtained.

6. Experimental Studies

In this section, we conduct two sets of experiments to evaluate (1) the performance of our visual processing module, (2) the accuracy of our inference module, and (3) the overall performance of our approach.

Experimental Setting.

DataSet. We used three data sets: (1) Soccer dataset that we annotated; and (2) Visual Genome dataset from <http://visualgenome.org>. We extracted images with subjects of golf and tennis. We split Soccer data into two parts: *I* (one third) and *II* (two thirds), and used *II* as training data, and *I* as testing data.

Questions. We used two sets of questions: (1) the set of questions given in Table 2 for Soccer dataset; and (2) another set of questions listed in Table ?? for X dataset.

(We enlarge the training question scale from 7 into 28, so learning correlation between question and answer does not work at this time. For question details, please refer questionset.txt.rtf.)

Id	Question	Difficulty
Q_{nl1}	Who is holding the soccer?	Easy
Q_{nl2}	What is the uniform color of the referee?	Easy
Q_{nl3}	Is there any referee in the image?	Easy
Q_{nl4}	Which team does the goalkeeper belong to?	Medium
Q_{nl5}	Who is the defending team?	Medium
Q_{nl6}	Which part of the field are the players being now?	Hard
Q_{nl7}	How many players are there in the image?	Hard
Q_{nl8}	Is this image about corner kick? (If not, just list the correct one.)	??
Q_{nl9}	Is this image about free kick? (If not, just list the correct one.)	??
Q_{nl10}	Is this image about kick off? (If not, just list the correct one.)	??
Q_{nl11}	Is this image about penalty kick? (If not, just list the correct one.)	??

Table 2: A set of questions

6.1. Generalization Ability of Model

To test the generalization of our method, we enlarge the training set by more various question with same meaning. For instance, the original question of Q_{nl5} is "Who is the

648
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701
 defending team?”, we add three more similar question asking *Who is attacking team?*, *What is the uniform color of the defending team?* and *What is the uniform color of the attacking team?*. Unlike state-of-art methods answering questions in [29], adding generalization and variation in question would not dramatically change the performance, it is because the structure is not fixed, all the visual task selection is query oriented. For [12], even though the network is not fixed, the answering part is based on neural network, and essentially it also learns the statically correlation, which leads to the weakness in logical reasoning. The result is shown in Table 3.

Various Training Questions		Original Training Questions	
Methods	Acc(%)	Methods	Acc(%)
CNN+LSTM	28.14	CNN+LSTM	46.40
HieCoAttenVQA	31.92	HieCoAttenVQA	49.11
Learning2Reason	38.00	Learning2Reason	51.08
Ours	64.02	Ours	66.75

Table 3: Generalization of Our Method

6.2. Performance of Visual Task Selecting Policy

To test the validity of reinforcement learning of selecting visual task modules, we test the inference time over accuracy with the state-of-art [4] [18] which is shown in ??.

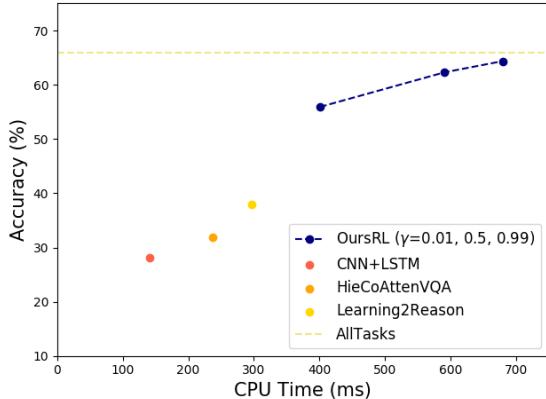


Figure 6: Inference time and accuracy.

6.3. Effectiveness of Inference

In VQA task, we aim to achieve great performance with short responding time. “CPU Time” evaluates the time cost from features extraction to question answering. All the computation work is implemented on CPU.

In terms of system accuracy, we follow the F-measure. Define that $\#true_value_inferred$ is the total number of instances whose attribute A is “ v ” and is inferred

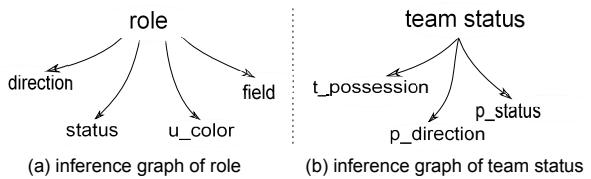


Figure 7: Inference graphs

correctly as “ v ”, $\#true_value_instance$ is the number of all the instances with attribute A of value “ v ”, and $\#inferred_instances$ indicates the total number of instances whose attribute A is inferred as “ v ”. The inference accuracy can be defined as below.

$$Acc(A = "v") = \frac{2 \cdot (recall(A = "v")) \cdot precision(A = "v"))}{recall(A = "v") + precision(A = "v"))} \quad (12)$$

where:

$$\begin{aligned} recall(A = "v") &= \frac{\#true_value_inferred}{\#true_value_instance} \\ precision(A = "v") &= \frac{\#true_value_inferred}{\#inferred_instance} \end{aligned}$$

We compare our approach to the state-of-the-art systems, i.e. CNN+LSTM[??], HieCoAttenVQA[??], and Learning2Reason[??].

	Time (ms)	Accuracy (%)
CNN+LSTM	141	28.14
HieCoAttenVQA	237	31.92
Learning2Reason	297	38.00
Ours (Without RL)	N/A	65.85
Ours ($\gamma = 0.01$)	401	55.92
Ours ($\gamma = 0.50$)	591	62.29
Ours ($\gamma = 0.99$)	680	64.02

Table 4: Inference time and accuracy comparison.

Table 4 lists the time cost and accuracy results among different approaches. For CNN+LSTM, HieCoAttenVQA, and Learning2Reason, their systems work quite efficient with responding time less than 300ms. But the accuracies of these three approaches are lower than 40%, which is unacceptable. Whereas, the system similar to [29] outperforms all other methods and reaches 65.86% accuracy, but it costs much more time. Our approach is able to keep balance between effectiveness and efficiency. For effectiveness, our approach achieves 64.02% accuracy, which is 35.88%, 32.10% and 26.02% higher than results of CNN+LSTM, HieCoAttenVQA, and Learning2Reason, respectively. For efficiency, our approach can reduce the inference time by choosing a small value of γ .

Accuracy of Role

756 Based on questions and image characteristics, four
 757 variables including *direction*, *status*, *field*, and
 758 *unique_color* (abbr. *u_color*) are adopted to calculate
 759 conditional probabilities. The inference graph is shown
 760 in Figure 7. Note that the domain of variables *direction*,
 761 *status* and *field* are given in Table ????, while variable
 762 *u_color* can have one of two values, to indicate whether a
 763 person object has the unique uniform color (=“U”) or not
 764 (=“M”).

Role	Precision	Recall	Accuracy
Goalkeeper	94.4	85.5	89.8
Referee	87.4	82.8	85.0
Player	98.8	99.3	99.0

Table 5: Inference accuracy of role (%).

771 Using the conditional probabilities, *VI* (???) infers role
 772 of each person object. The inference accuracy is shown in
 773 Table 5. It is easy to find that the inference accuracy for role
 774 player reaches 99%, which is highest among all roles. And
 775 the accuracies for different roles are all above 85%.

Accuracy of Team Status

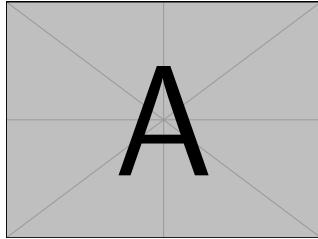


Figure 8: Attributes of Team Inference Graph.

Role	Precision	Recall	Accuracy
Defending	89.8	74.2	81.3
Attacking	79.1	92.1	85.1

Table 6: Inference accuracy of team status.

791 Table 6 gives the performance of our approach for infer-
 792 ring team status. Both of the accuracies of defending and
 793 attacking are above 80%. To be specific, the accuracy for
 794 inferring if a team is attacking achieves 85.1%, which is
 795 higher than the inference accuracy for defending status de-
 796 tection by 3.8%.

Accuracy of Field Scene

806 Field Scene includes four different scenes, *i.e.* corner
 807 kick, free kick, penalty kick and kick-off. Practically, cor-
 808 ner kick scene always happens as a single player kick the

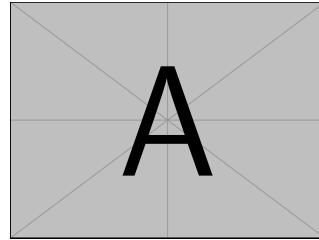


Figure 9: Attributes of Scene Inference Graph.

810 ball within a one-yard radius of the corner flag and most of
 811 players gather in the penalty field. Free kick happens out-
 812 side the penalty area and defensive wall exists mostly. It
 813 is much typical for the penalty kick scene because most of
 814 players are out of penalty area except kick player and goal-
 815 keeper with football in the penalty spot. As for kick-off
 816 scene, the ball is played in the center spot with all mem-
 817 bers of the opposing team at least 10 yards from the ball.
 818 Based on these observations, we design an inference graph
 819 in Figure 9.

820 Inference accuracy is indicated in Table 7 based on in-
 821 ference graph in Figure 9. As is shown, the inference ac-
 822 curacy of corner kick, free kick, kick-off and penalty kick
 823 are 59.57%, 63.16%, 85.94% and 60.00%, respectively. It
 824 is obvious that the scene of kick off reaches the highest ac-
 825 curacy among all.

Field Type	Precision	Recall	F1-score
Corner-kick	59.57	68.29	63.64
Free-kick	63.16	58.54	60.76
Kick-off	85.94	82.09	83.97
Penalty-kick	60.00	60.00	60.00
Average	71.28	70.73	70.89

Table 7: Inference accuracy of field scene (%) with our ap-
proach.

845 In addition, we compared our approach with NuSVC,
 846 MLP, and AdaBoost in Table 8. The results show that the
 847 average accuracy of our approach is higher than that of
 848 NuSVC and AdaBoost by 4.49% and 5.97% respectively,
 849 and slightly better than that of MLP.

	Precision	Recall	F1-score
NuSVC	68.30	65.85	66.40
MLP	70.80	70.73	70.32
AdaBoost	65.50	65.24	64.92
Ours	71.28	70.73	70.89

Table 8: Average accuracy comparison of field scene (%)

864

6.4. Overall Performance

We compared our proposed method with the following state-of-art methods: LSTM+CNN, HieCoAttenVQA, and Learn2Reason. The results are listed in [Table 9](#). The average accuracy using our approach is higher than accuracy of CNN+LSTM, HieCoAttenVQA and Learn2Reason by 20.35%, 17.64% and 15.67%, respectively.

	CNN+LSTM	HieCoAtten	Learn2Reason	Ours
Q_{nl1}	44.23	43.62	31.12	64.86
Q_{nl2}	71.31	77.66	9.4	63.74
Q_{nl3}	74.58	83.78	83.21	70.00
Q_{nl4}	40.48	39.29	51.92	62.14
Q_{nl5}	49.19	49.90	30.78	62.58
Q_{nl6}	20.56	18.70	30.0	93.33
Q_{nl7}	11.08	12.63	36.69	50.60
Avg.	46.40	49.11	51.08	66.75

Table 9: Accuracy comparison per question and average (%)

7. Conclusion

In this paper, we proposed an innovative and efficient approach to handle the VQA problem. In the proposed method, the image and question are converted to entity-attribute graph and pattern query, respectively. Then, the reinforcement learning technique is utilized to select the pattern query that is helpful to the visual tasks, in which the missing attributes are inferred by the inference graph constructed from a Bayes network. Last but not the least, the answer is found by graph matching. The generalization of the proposed scheme is significantly improved by introducing the reinforcement learning and inference graph. More importantly, the inference graph here is used in a novel way to make the graph works adaptively. To be specific, the low-level but unknown information is inferred from the known attributes by the inference network, and the high-level but unknown information is finally inferred when the unknown attributes are well inferred. The experimental results encouragingly demonstrate that the proposed scheme corroborates the efficiency and high accuracy when compared with other state-of-the-art baseline methods on two data sets.

The problem of VQA has been widely studied using the graph manner but with slight satisfaction. The reason may concern the integration of external data with complex reasoning tasks, the improvement of inference scheme, and the interactive strategy.

References

- [1] Visual genome dataset. <http://visualgenome.org>.

- | | | | | | |
|------|---|---|---|------|-----|
| [2] | F. Abtahi and I. R. Fasel. | Deep belief nets as function approximators for reinforcement learning. | In <i>Lifelong Learning, Papers from the 2011 AAAI Workshop, San Francisco, California, USA, August 7, 2011</i> , 2011. | 2 | 918 |
| | | | | | 919 |
| [3] | J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. | Neural module networks. | <i>2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)</i> , Jun 2016. | 2 | 920 |
| | | | | | 921 |
| [4] | S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. | VQA: Visual Question Answering. | <i>International Conference on Computer Vision (ICCV)</i> , 2015. | 7 | 922 |
| | | | | | 923 |
| [5] | D. Bahdanau, K. Cho, and Y. Bengio. | Neural machine translation by jointly learning to align and translate. | May 2016. | 4 | 924 |
| | | | | | 925 |
| [6] | Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. | Realtime multi-person 2d pose estimation using part affinity fields. | <i>CVPR</i> , 2017. | 5 | 926 |
| | | | | | 927 |
| [7] | J. Dai, Y. Li, K. He, and J. Sun. | R-FCN: object detection via region-based fully convolutional networks. | <i>Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain</i> , pages 379–387, 2016. | 5 | 928 |
| | | | | | 929 |
| [8] | N. Friedman, D. Geiger, and M. Goldszmidt. | Bayesian network classifiers. | <i>Machine learning</i> , 29(2-3):131–163, 1997. | 5 | 930 |
| | | | | | 931 |
| [9] | A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. | Multimodal compact bilinear pooling for visual question answering and visual grounding. | <i>arXiv preprint arXiv:1606.01847</i> , 2016. | 2 | 932 |
| | | | | | 933 |
| [10] | B. Goodrich and I. Arel. | Reinforcement learning based visual attention with application to face detection. | <i>In 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, June 16-21, 2012</i> , pages 19–24, 2012. | 2 | 934 |
| | | | | | 935 |
| [11] | Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. | Making the v in vqa matter: Elevating the role of image understanding in visual question answering. | <i>2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)</i> , Jul 2017. | 1 | 936 |
| | | | | | 937 |
| [12] | R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. | Learning to reason: End-to-end module networks for visual question answering. | <i>CoRR, abs/1704.05526</i> , 3, 2017. | 2, 7 | 938 |
| | | | | | 939 |
| [13] | D. Koller, N. Friedman, and F. Bach. | <i>Probabilistic graphical models: principles and techniques</i> . | MIT press, 2009. | 5 | 940 |
| | | | | | 941 |
| [14] | M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Perolat, D. Silver, and T. Graepel. | A unified game-theoretic approach to multiagent reinforcement learning. | <i>In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30</i> , pages 4190–4203. Curran Associates, Inc., 2017. | 2 | 942 |
| | | | | | 943 |
| [15] | S. Lange and M. Riedmiller. | Deep auto-encoder neural networks in reinforcement learning. | <i>In The 2010 International Joint Conference on Neural Networks (IJCNN)</i> , pages 1–8, July 2010. | 2 | 944 |
| | | | | | 945 |
| [16] | S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. | Improved image captioning via policy gradient optimization of | | | 946 |
| | | | | | 947 |
| | | | | | 948 |
| | | | | | 949 |
| | | | | | 950 |
| | | | | | 951 |
| | | | | | 952 |
| | | | | | 953 |
| | | | | | 954 |
| | | | | | 955 |
| | | | | | 956 |
| | | | | | 957 |
| | | | | | 958 |
| | | | | | 959 |
| | | | | | 960 |
| | | | | | 961 |
| | | | | | 962 |
| | | | | | 963 |
| | | | | | 964 |
| | | | | | 965 |
| | | | | | 966 |
| | | | | | 967 |
| | | | | | 968 |
| | | | | | 969 |
| | | | | | 970 |
| | | | | | 971 |

- 972 spider. In *The IEEE International Conference on Computer* 1026
973 *Vision (ICCV)*, Oct 2017. 5
974 [17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, 1027
975 C. Fu, and A. C. Berg. SSD: single shot multibox detector. 1028
976 In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, pages 21–37, 2016. 5
977 [18] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical 1029
978 question-image co-attention for visual question answering, 1030
979 2016. 7
980 [19] S. Mathe, A. Pirinen, and C. Sminchisescu. Reinforcement 1031
981 learning for visual object detection. In *2016 IEEE Conference 1032
982 on Computer Vision and Pattern Recognition, CVPR 2016, Las 1033
983 Vegas, NV, USA, June 27-30, 2016*, pages 2894– 1034
984 2902, 2016. 2
985 [20] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. 1035
986 Recurrent models of visual attention. In *Advances in Neural 1036
987 Information Processing Systems 27: Annual Conference on 1037
988 Neural Information Processing Systems 2014, December 8-13 1038
989 2014, Montreal, Quebec, Canada*, pages 2204–2212, 2014. 1039
990 2
991 [21] F. Petitjean, W. Buntine, G. I. Webb, and N. Zaidi. Accurate 1040
992 parameter estimation for bayesian network classifiers using 1041
993 hierarchical dirichlet processes. *Machine Learning*, 107(8-10):1303–1331, 2018. 6
994 [22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards 1042
995 real-time object detection with region proposal networks. In 1043
996 *Proceedings of the 28th International Conference on Neural 1044
997 Information Processing Systems - Volume 1, NIPS’15*, pages 1045
998 91–99, Cambridge, MA, USA, 2015. MIT Press. 5
1000 [23] Z. Ren, X. Wang, N. Zhang, X. Lv, and L. Li. Deep 1046
1001 reinforcement learning-based image captioning with embedding 1047
1002 reward. In *2017 IEEE Conference on Computer Vision and 1048
1003 Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 1049
1004 21-26, 2017*, pages 1151–1159, 2017. 2
1005 [24] D. Teney, L. Liu, and A. van den Hengel. Graph- 1050
1006 structured representations for visual question answering. 1051
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
arXiv preprint, 2017. 2
[25] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S. C. Zhu. Joint 1026
video and text parsing for understanding events and answering 1027
queries. *IEEE MultiMedia*, 21(2):42–70, 2014. 2
[26] X. Wang and T. Sandholm. Reinforcement learning to play 1028
an optimal nash equilibrium in team markov games. In 1029
S. Becker, S. Thrun, and K. Obermayer, editors, *Advances 1030
in Neural Information Processing Systems 15*, pages 1603– 1031
1610. MIT Press, 2003. 2
[27] C.-Y. Wei, Y.-T. Hong, and C.-J. Lu. Online reinforcement 1032
learning in stochastic games. In I. Guyon, U. V. Luxburg, 1033
S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and 1034
R. Garnett, editors, *Advances in Neural Information Process- 1035
ing Systems 30*, pages 4987–4997. Curran Associates, Inc., 1036
2017. 2
[28] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Con- 1037
volutional pose machines. In *CVPR*, 2016. 5
[29] P. Xiong, H. Zhan, X. Wang, B. Sinha, and Y. Wu. Visual 1038
query answering by entity-attribute graph matching and rea- 1039
soning. In *CVPR*, 2019. 1, 2, 4, 5, 7
[30] H. Xu and K. Saenko. Ask, attend and answer: Exploring 1040
question-guided spatial attention for visual question answer- 1041
ing. In *European Conference on Computer Vision*, pages 1042
451–466. Springer, 2016. 2
[31] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma. Structured 1043
attentions for visual question answering. In *Proc. IEEE Int. 1044
Conf. Comp. Vis*, volume 3, 2017. 2
[32] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: 1045
Grounded question answering in images. In *Proceedings 1046
of the IEEE Conference on Computer Vision and Pattern 1047
Recognition*, pages 4995–5004, 2016. 2
[33] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei- 1048
Fei, and A. Farhadi. Target-driven Visual Navigation in 1049
Indoor Scenes using Deep Reinforcement Learning. In 1050
IEEE International Conference on Robotics and Automation, 1051
2017. 2