# Visual Question Answering with Adaptive 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 for the* VQA *problem. (1) It represents inputs,* i.e. *image Img and question $Q_{nl}$ as entity-attribute graph and pattern query, respectively, and employs graph matching to find answers; (2) it leverages reinforcement learning based model to identify a set of policies that are used to guide visual tasks and select the corresponding pattern query, based on $Q_{nl}$; and (3) it trains a classifier and reasons missing values that are crucial for question answering. With these features, our approach not only conducts visual tasks more efficiently, but also answers 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 soccer match data set (Soccer-VQA) and Visual-Genome data set, 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 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 also pointed out that despite recent progress, today's neural network based approach has a few weaknesses, which greatly hinders its further development. First of all, deep neural network works as "black boxes", hence it is very hard to identify the causal relations between model design and system performance. Secondly, but more importantly, there is no evidence to support the hypothesis that neural network has the capability of reasoning in solving VQA problems.

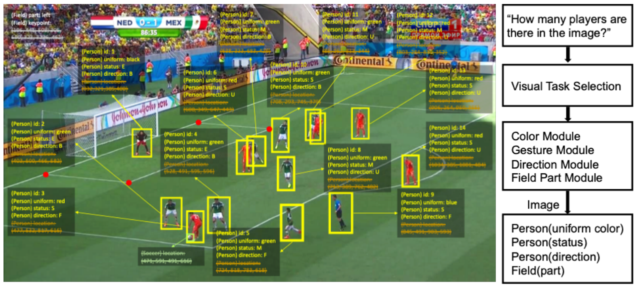


Figure 1: The image is about soccer match, where each person object is associated with attributes: id, uniform color, status (Standing, Moving, Expansion), direction (Backing, Facing, N/A), as well as location, and the soccer object is attributed with location. However, not all informations are necessary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

Traditional VQA techniques that rely on deep neural networks has received significant progress in recent years [], while due to lack of ability of question understanding and reasoning, these techniques often perform poorly, and can not be used in practical applications.

**Example 1:** Figure 1 depicts an image about a soccer match, where two teams are distinguished by red and green uniforms, and each object is associated with a set of attributes. A typical query may ask "How many players are there in the image?". Though simple, it is a challenging task to efficiently answer the query, since (1) it is often very inefficient to detect all the objects in the given image, following traditional way, while only query related objects are in demand; (2) one not only needs to identify all the *person* objects, but also have to infer their hidden attribute "role", which is often ambitious. For better efficiency, part of vision tasks are selected to accomplish acquire the necessary information for this question.

Motivated by these, one may analyze questions first and detects only those objects as well as their attributes that are in connection with questions, then infer missing values of hidden attributes, and answer questions. In this way, not only query accuracy but also query efficiency are expected to be guaranteed. □

This example suggests that we leverage adaptive query understanding and reasoning to address the VQA problem. While to do this, two critical questions have to be answered. (1) How to understand queries and carry query-related visual tasks? (2) How to infer crucial information to assist query answering?

**Contributions.** In contrast to a majority of deep learning based VQA techniques, which not only overlooks correlation between queries and images, but also lacks of necessary reasoning, we propose a novel technique that integrates adaptive query understanding and reasoning. The main contributions of the paper are as follow.

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

(2) We introduce a method to guide visual tasks based on reinforcement learning. That is, given an image and a question, our method can identify a set of visual tasks that are question related, and direct subsequent visual processing in a more efficient manner.

(3) We propose approaches to answering visual queries based on reasoning and graph matching. More specifically, we first transform a given image into an entity-attribute graph; we then develop method to infer missing value for question answering; we finally show how to answer queries with graph matching.

4) We conduct extensive experimental studies to verify the performance of our method. We find that X, Y, and Z.

## 2. Related Work

We categorize related work into following three parts.

*Visual query answering*. Current VQA approaches are mainly based on deep neural works. [30] introduces a spatial attention mechanism similar to the model for image captioning. Instead of computing the attention vector iteratively, [28] obtains a global spatial attention weights vector which is then used to generate a new image embedding. [29] proposed to model the visual attention as a multivariate distribution over a grid-structured conditional random field on image regions, thus multiple regions can be selected at the same time. This attention mechanism is called structured multivariate attention in [29]. There has been many other improvements to the standard deep learning method, *e.g.* [8] utilized Multimodal Compact Bilinear (MCB) pooling to efficiently and expressively combine multimodal features. Another interesting idea is the implementation of Neural Module Networks [2, 10], which decomposes queries into their linguistic substructures, and uses these structures to dynamically instantiate module networks. [22] proposed to build graph over scene objects and question words. The visual graph is similar to ours, but the query graph differs. Note that the method [22] proposed is still a neural network based method as the structured representations are fed into a recurrent network to form the final embedding and the answer is again inferred by a classifier.

*Environment Exploration in Visual Field*. Reinforcement driven information acquisition is not only focusing at games [25, 24, 12] but also wildly applied in traditional vision domain. [17] implement reinforcement learning in visual object detection, by presenting a novel sequential models which accumulate evidence collected at a small set of image locations to detect visual objects effectively. [9] forms the facial detection problem into an adaptive learning process, by designing an approximate optimal control framework, based on reinforcement learning to actively search a visual field. [18] introduced a novel recurrent neural network model which is capable to extract information from an image or video by adaptive selection for a sequence of regions or locations. [31] introduced reinforcement learning in the task of target-driven visual navigation. Other works aims to achieving based on algorithms [13, 1].

In visual and language domain, relevant work like [21] achieves image captioning with Embedding Reward. Reinforcement learning preserves ability to effectively select preferred actions, which benefits the system decomposing the problem into a few sub-tasks.

*Graph-based visual understanding*. [23] proposes a framework to understand events and answer user queries, where underlying knowledge is represented by a spatial-temporal-causal And-Or graph (S/T/C-AOG).

## 3. Overview of the Approach

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

### 3.1. Representation of Images and Questions

We use the same representations as [27]. To make the paper self-contained, we cite them as follows (rephrased).

**Entity-Attribute Graphs.** Entities are typically defined as

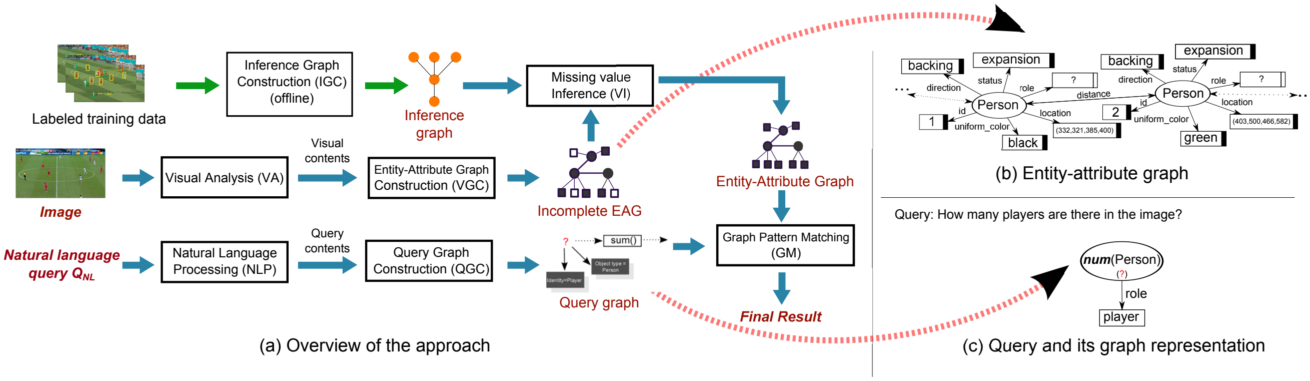

Labeled training data
Inference Graph Construction (IGC) (offline)
Inference graph
Missing value Inference (VI)
Image
Visual Analysis (VA)
Visual contents
Entity-Attribute Graph Construction (VGC)
Incomplete EAG
Entity-Attribute Graph
Natural language query $Q_{NL}$
Natural Language Processing (NLP)
Query contents
Query Graph Construction (QGC)
?
sum()
Object type = Person
Identity=Player
Query graph
Graph Pattern Matching (GM)
Final Result
(a) Overview of the approach
backing
expansion
direction
status
role
?
Person
distance
id
location
uniform_color
1
2
black
green
(332,321,385,400)
(403,500,466,582)
(b) Entity-attribute graph
Query: How many players are there in the image?
num(Person)
(?)
role
player
(c) Query and its graph representation

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

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 $\Theta$ of types. Each entity $e$ in $\mathcal{E}$ has a *unique ID* and a *type* in $\Theta$.

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 VGA in Section **??**, modules IGC and VI in Section **??**, and module GM in Section **??** with more details.

# 4. Question Oriented Visual Tasks

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

## 4.1. Visual Processing

In our approach, we build a structure which selects subtasks 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.

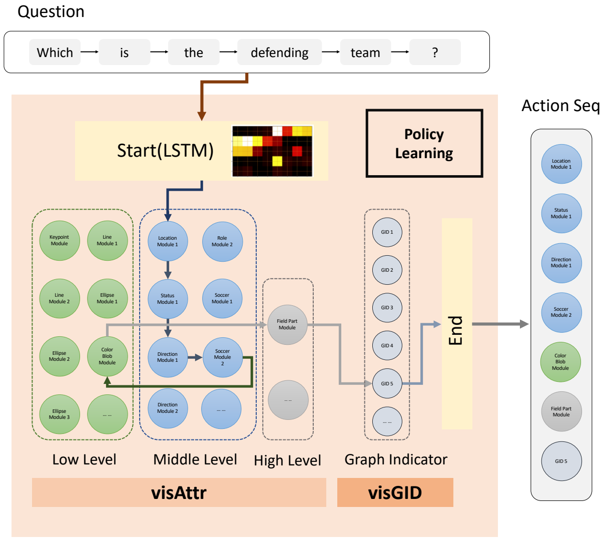


Figure 3: Visual Processing Strategy

In our approach, we build a structure which selects subtasks 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.

### 4.1.1 Multi-layer LSTM with Attention

The task here is to predict the most suitable action modules sequence $\mathsf{a}$ 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 [4], 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.

$$c_i = \sum_{j=1}^{\|Q\|} a_{ij} h_{ij} \tag{1}$$

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

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

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 $\mathsf{a}_i$ 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 $\mathsf{A} = [\mathsf{a}_1, \ldots, \mathsf{a}_t]$.

$$p(\mathsf{A}|Q) = \Pi_{t \in \|\mathsf{A}\|}\, p(\mathsf{a}_t|\{\mathsf{a}_1, \ldots, \mathsf{a}_t\}, c_i, Q) \tag{4}$$

$$p(\mathsf{a}_t|\{\mathsf{a}_1, \ldots, \mathsf{a}_t\}, c_i, Q) = g(\mathsf{a}_{t-1}, s_t, c_i, Q) \tag{5}$$

$$s_t = f(s_{t-1}, \mathsf{a}_{t-1}, c_i) \tag{6}$$

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

Guided by this action sequence $[\mathsf{a}_1, \mathsf{a}_2, \mathsf{a}_3, \ldots, \mathsf{a}_n]$, actions are selected from the following visual task pool, and comes into next session, visual task selection (VTS).

### 4.1.2 Visual Task Selection

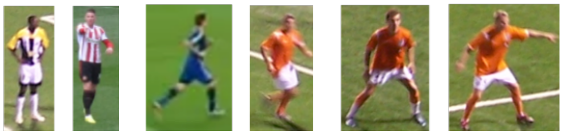
VTS is guided by the question feature, selecting target visual tasks from the visual task pool by Monte Carlo learning. The task pool is demonstrated in Table 1.

The pool is constructed by two parts, the first one is *visAttr* which aims to discover the attribute of people, soccer, field and scene [27], 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 difficulty degree of the vision tasks.

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 [20], R-FCN [6], SSD [15] or skeleton keypoints detection based method like [5] and [26] 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).

| visAttr | Level | Sub-task | Descriptions |
|---|---|---|---|
| visAttr | Low | Keypoint Module<br>Line Module<br>Ellipse Module | To get detailed information of the soccer field $F_{keypoint}$. |
| | | Color Blob Module | To detect blobs of different colors among a region(whole soccer field or a small bounding box). |
| | Middle | Location Module | To detect the location of the person $P_{location}$. |
| | | 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 $P_{uniform}$. |
| | | Soccer Module | To detect the location of the soccer $S_{location}$. |
| | High | 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



(a) Standing (b) Moving (c) Expansion

Figure 4: Person status.

Thus, the system is not only able to determine whether resembles a vision task module into action module sequence $[\mathsf{a}_1, \mathsf{a}_2, \mathsf{a}_3, \ldots, \mathsf{a}_n]$, 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.

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

where $\tau(\cdot)$ and $\alpha(\cdot)$represent the pre-tested inference time and inference accuracy, action module sequence $\mathsf{A}$ samples from joint distribution $p(\mathsf{A}|Q)$, and here $\ell(\cdot)$ is the softmax loss over the predict score. For the preference term $\gamma$, 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 $[\mathsf{a}_1, \mathsf{a}_2, \mathsf{a}_3, \ldots, \mathsf{a}_n]$. 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.

$$L_{policy}(\theta) = \sum\nolimits_{i \in \|A\|} log\pi(\mathsf{a}_i|Q, \theta)\ell(Q, \mathsf{A}) \quad (8)$$

where $\mathsf{a}_i$ 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 $\mathsf{a}_i$ based on current status. And $\ell(\cdot)$ here is the softmax loss based on the whole action module sequence $[\mathsf{a}_1, \mathsf{a}_2, \mathsf{a}_3, \ldots, \mathsf{a}_n]$. Since all actions are discrete, which leads to non-differentiable, and back-propagation cannot be used. Policy gradient [14] 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 [27].

**Example 2:** ADD AN EXAMPLE TO ILLUSTRATE PROGRESS IF NECESSARY! □

## 5. Reasoning

According to our observation, an *incomplete* EAG isn't well satisfying of answering the query 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 [7] can be described by a pair $\mathfrak{B} =< \mathcal{G}, \Theta_{\mathcal{G}} >$. 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 $\mathrm{Pa}(X_i)$ the attributes of the parents of $X_i$, the parameter of $X_i$ is represented by $\theta_{X_i|\mathrm{Pa}(X_i)} = P_{\mathfrak{B}}(X_i|\mathrm{Pa}(X_i))$. With the notations above, the joint probability distribution of Bayesian network is given by:

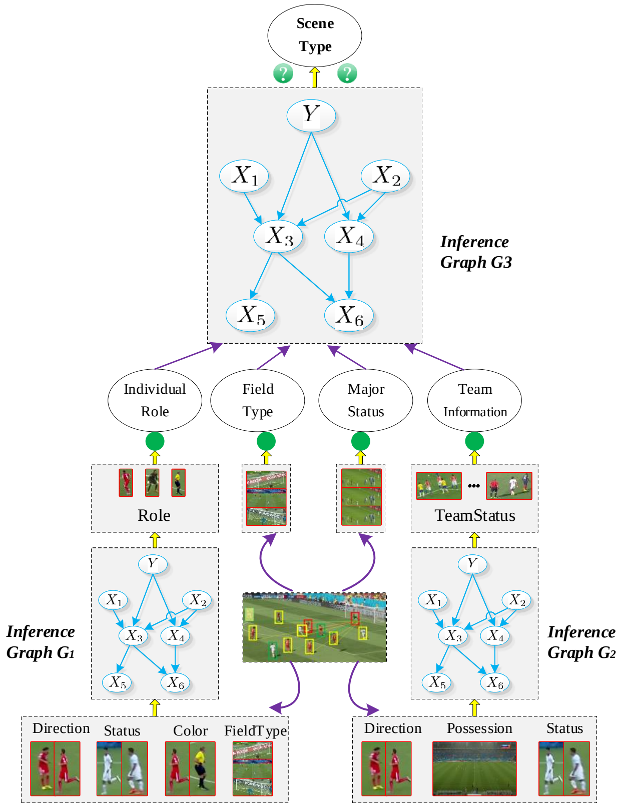


Figure 5: Schematic diagram of inference graph.

$$P_{\mathfrak{B}}(X_1,\cdots,X_n)=\prod_{i=1}^{n} P_{\mathfrak{B}}(X_i|\text{Pa}(X_i))=\prod_{i=1}^{n}\theta_{X_i|\text{Pa}(X_i)} \tag{9}$$

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 [11]. 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:

$$\begin{aligned} P_{\mathfrak{B}}(Y|X) &= \frac{P_{\mathfrak{B}}(Y)P_{\mathfrak{B}}(X|Y)}{P_{\mathfrak{B}}(X)} \\ &= \frac{\theta_{Y|\text{Pa}(X_0)}\prod_{i=1}^{n}\theta_{X_i|Y,\text{Pa}(X_i)}}{\sum_{y'\in\mathcal{Y}}\theta_{y'|\text{Pa}(X_0)}\prod_{i=1}^{n}\theta_{X_i|y',\text{Pa}(X_i)}} \end{aligned} \tag{10}$$

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 $P_{\mathfrak{B}}(Y|X)$ is simplified by taking the class variable as the root, and all attributes are conditionally independent when taking the class as a condition [19]. As a consequence, the attribute class can be explicitly inferred by:

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

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

Note 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.

Fig. 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, in which four different kinds of features are extracted from the scene image. Then, the team status is inferred through the second inference graph, 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 two datasets: (1) Soccer dataset that we annotated; and (2) X dataset from []. We extracted images with subjects of golf and tennis (report Statistics about the dataset). We split Soccer (resp. X) data into two parts:*I* (one third) and *II* (two thirds), and used *II* as training data, and *I* as testing data.

*Queries*. 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.)

### 6.1. 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 [3] [16] which is shown in Figure **??**.

Here to test the generalization, we enlarge the training set by more various question with same meaning. For instance, the original question of $Q_{nl5}$ is *"Who is the defending team?"*, we add three more similar question asking *Who is attacking team?*, *"What is the uniform color of*

| Id | Question | Difficulty |
|---|---|---|
| $Q_{nl_1}$ | Who is holding the soccer? | Easy |
| $Q_{nl_2}$ | What is the uniform color of the referee? | Easy |
| $Q_{nl_3}$ | Is there any referee in the image? | Easy |
| $Q_{nl_4}$ | Which team does the goalkeeper belong to? | Medium |
| $Q_{nl_5}$ | Who is the defending team? | Medium |
| $Q_{nl_6}$ | Which part of the field are the players being now? | Hard |
| $Q_{nl_7}$ | How many players are there in the image? | Hard |
| $Q_{nl_8}$ | Is this image about corner kick?<br>(If not, just list the correct one.) | ?? |
| $Q_{nl_9}$ | Is this image about penalty kick?<br>(If not, just list the correct one.) | ?? |
| $Q_{nl_{10}}$ | Is this image about kick off?<br>(If not, just list the correct one.) | ?? |

Table 2: A set of questions

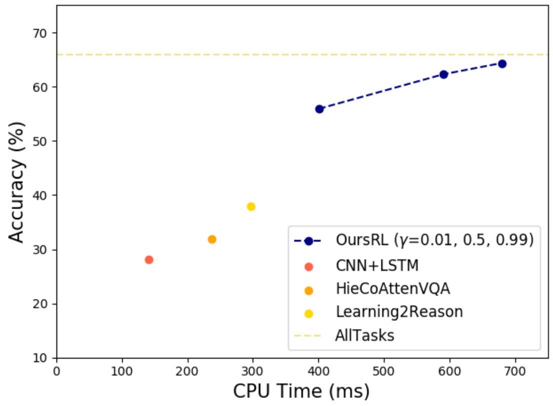
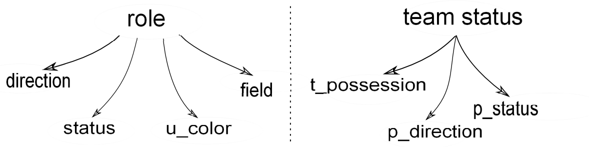


Figure 6: Inference Time and Accuracy



(a) inference graph of role (b) inference graph of team status

Figure 7: Inference Graphs

*the defending team?"* and *"What is the uniform color of the attacking team?"*. Unlike state-of-art methods answering questions in [27], 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 [10], 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.

## 6.2. 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 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.

$$Acc(A = \text{"}v\text{"}) = \frac{2 \cdot (recall(A = \text{"}v\text{"})) \cdot presision(A = \text{"}v\text{"})}{recall(A = \text{"}v\text{"}) + presision(A = \text{"}v\text{"})} \tag{12}$$

where:

$$recall(A = \text{"}v\text{"}) = \frac{\#true\_value\_inferred}{\#true\_value\_instance}$$
$$precision(A = \text{"}v\text{"}) = \frac{\#true\_value\_inferred}{\#inferred\_instance}$$

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

| | Time (ms) | Acc(%) |
|---|---|---|
| CNN+LSTM | 141 | 28.14 |
| HieCoAttenVQA | 237 | 31.92 |
| Learning2Reason | 297 | 38.00 |
| Ours ($\gamma = 0$) | 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 3: Accuracy comparison per query and average (%).

Table 3 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 [27] 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 $\gamma$.

**Accuracy of Role**

Based on queries and image characteristics, four variables including *direction*, *status*, *field*, and

$unique\_color$ (abbr. $u\_color$) are adopted to calculate conditional probabilities. The inference graph is shown in Figure 7. Note that the domain of variables $direction$, $status$ and $field$ are given in Table ????, while variable $u\_color$ can have one of two values, to indicate whether a person object has the unique uniform color (="$U$") or not (="$M$").

| Role | Precision | Recall | Accuracy |
|---|---|---|---|
| Role = "G" | 94.4 | 85.5 | 89.8 |
| Role = "R" | 87.4 | 82.8 | 85.0 |
| Role = "P" | 98.8 | 99.3 | 99.0 |

Table 4: Inference accuracy of role (%). Here "G", "R" and "P" indicate goalkeeper, referee and player, respectively.

Using the conditional probabilities, $VI$ (???) infers role of each person object. The inference accuracy is shown in Table 4. It is easy to find that the inference accuracy for role player reaches 99%, which is highest among all roles. And the accuracies for different roles are all above 85%.

### Accuracy of Team Status

### Accuracy of Field Scene

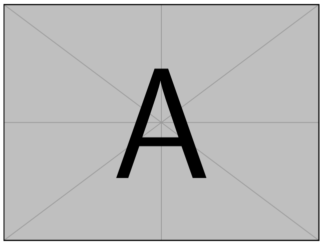


Figure 8: Attributes of Scene Inference Graph.

Field Scene includes four different scenes, *i.e.* corner kick, free kick, penalty kick and kick-off. Practically, corner kick scene always happens as a single player kick the ball within a one-yard radius of the corner flag and most of players gather in the penalty field. Free kick happens outside the penalty area and defensive wall exists mostly. It is much typical for the penalty kick scene because most of players are out of penalty area except kick player and goalkeeper with football in the penalty spot. As for kick-off scene, the ball is played in the center spot with all members of the opposing team at least 10 yards from the ball. Based on these observations, we design an inference graph in Figure 8.

Inference accuracy is indicated in Table 5 based on inference graph in Figure 8. As is shown, the inference accuracy of corner kick, free kick, kick-off and penalty kick are 59.57%, 63.16%, 85.94% and 60.00%, respectively. It is obvious that the scene of kick off reaches the highest accuracy among all.

| Field Type | Precision | Recall | F1-score |
|---|---|---|---|
| Field Scene = "Co" | 59.57 | 68.29 | 63.64 |
| Field Scene = "Fr" | 63.16 | 58.54 | 60.76 |
| Field Scene = "Ki" | 85.94 | 82.09 | 83.97 |
| Field Scene = "Pe" | 60.00 | 60.00 | 60.00 |
| Average | 71.28 | 70.73 | 70.89 |

Table 5: Inference accuracy of field scene (%) with our approach. Here "Co", "Fr", "Ki" and "Pe" indicate corner kick, free kick, penalty kick and kick-off, respectively.

In addition, we compared our approach with NuSVC, MLP, and AdaBoost in Table 6. The results show that the average accuracy of our approach is higher than that of NuSVC and AdaBoost by 4.49% and 5.97% respectively, 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 6: Average accuracy comparison of field scene (%)

## 6.3. 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 7. 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 7: Accuracy comparison per query 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] F. Abtahi and I. R. Fasel. Deep belief nets as function approximators for reinforcement learning. In *Lifelong Learning, Papers from the 2011 AAAI Workshop, San Francisco, California, USA, August 7, 2011*, 2011. 2

[2] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. 2

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 6

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. May 2016. 4

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 4

[6] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 379–387, 2016. 4

[7] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997. 5

[8] 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. *arXiv preprint arXiv:1606.01847*, 2016. 2

[9] B. Goodrich and I. Arel. Reinforcement learning based visual attention with application to face detection. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, June 16-21, 2012*, pages 19–24, 2012. 2

[10] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. *CoRR, abs/1704.05526*, 3, 2017. 2, 7

[11] D. Koller, N. Friedman, and F. Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 6

[12] 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. 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*, pages 4190–4203. Curran Associates, Inc., 2017. 2

[13] S. Lange and M. Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2010. 2

[14] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Improved image captioning via policy gradient optimization of spider. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, pages 21–37, 2016. 4

[16] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering, 2016. 6

[17] S. Mathe, A. Pirinen, and C. Sminchisescu. Reinforcement learning for visual object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2894–2902, 2016. 2

[18] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2204–2212, 2014. 2

[19] F. Petitjean, W. Buntine, G. I. Webb, and N. Zaidi. Accurate parameter estimation for bayesian network classifiers using hierarchical dirichlet processes. *Machine Learning*, 107(8-10):1303–1331, 2018. 6

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 91–99, Cambridge, MA, USA, 2015. MIT Press. 4

[21] Z. Ren, X. Wang, N. Zhang, X. Lv, and L. Li. Deep reinforcement learning-based image captioning with embedding reward. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1151–1159, 2017. 2

[22] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. *arXiv preprint*, 2017. 2

[23] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S. C. Zhu. Joint video and text parsing for understanding events and answering queries. *IEEE MultiMedia*, 21(2):42–70, 2014. 2

[24] X. Wang and T. Sandholm. Reinforcement learning to play an optimal nash equilibrium in team markov games. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1603–1610. MIT Press, 2003. 2

[25] C.-Y. Wei, Y.-T. Hong, and C.-J. Lu. Online reinforcement learning in stochastic games. 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*, pages 4987–4997. Curran Associates, Inc., 2017. 2

[26] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016. 4

[27] P. Xiong, H. Zhan, X. Wang, B. Sinha, and Y. Wu. Visual query answering by entity-attribute graph matching and reasoning. In *CVPR*, 2019. 2, 4, 5, 7

[28] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016. 2

[29] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma. Structured attentions for visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, 2017. 2

[30] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004, 2016. 2

[31] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. In *IEEE International Conference on Robotics and Automation*, 2017. 2