Learning Question-oriented Vision for Generalizable VQA

Anonymous ICCV submission

Paper ID 954

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 [2], 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 [31, 12] also pointed out that despite recent progress, today's neural network based approaches demonstrate 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 poor performance 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, a host of existing techniques show poor performance when answering real-life questions, that are often open-ended and require necessary reasoning. Though a novel method [31] with reasoning capability is proposed, the method shows difficulty in generalization.

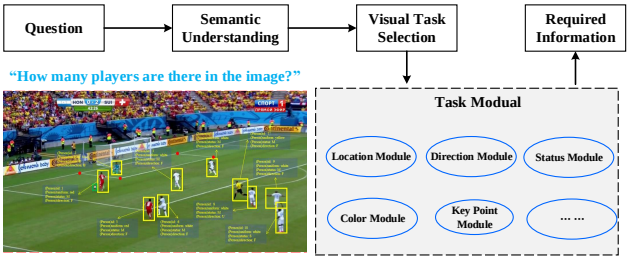

Figure 1: The image is about soccer match, where each person object is associated with attributes: id, uniform color, status (Standing, Moving, Expansion), direction (Backing, Facing, N/A), as well as location, and the soccer object is attributed with location. However, not all informations are necessary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

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 2 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) information extracted from image alone is often insufficient to answer questions, hence missing values that are crucial for question answering should be inferred by certain reasoning techniques.

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

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

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

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

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

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

4) We conduct extensive experimental studies to verify the performance of our method on both our curated VQA dataset and a public VQA dataset. We find that our method outperforms statt-of-the-art methods on both datasets.

2. Related Work

We categorize related work into following three parts.

Visual query answering. Current VQA approaches are mainly based on deep neural works. [34] introduces a spatial attention mechanism similar to the model for image captioning. Instead of computing the attention vector iteratively, [32] obtains a global spatial attention weights vector which is then used to generate a new image embedding. There has been many other improvements to the standard deep learning method, *e.g*. [10] utilized Multimodal Compact Bilinear (MCB) pooling to efficiently and expressively combine multimodal features. Another interesting idea is the implementation of Neural Module Networks [4, 13], which decomposes queries into their linguistic substructures, and uses these structures to dynamically instantiate module networks. [25] 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 [25] 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 on games [29, 28, 15] but also wildly applied in traditional vision domain. [20] 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. [11] 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. [21] proposed 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. [35] introduces reinforcement learning in the task of target-driven visual navigation. Other works aim at achieving XXX based on algorithms [16, 3].

In visual and language domain, relevant work like [24] achieves image captioning with Embedding Reward. Reinforcement learning preserves ability to effectively select preferred actions, which benefits the system in decomposing the problem into a few sub-tasks.

Graph-based VQA. [26] 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). [33] recovers a structural scene symbolization from the image which is similar to graph representation, and learns a program trace from the question, which is later executed on the scene representation to obtain the answer. Our method differs in the way of answer retrieval by graph matching instead of program execution, furthermore, our method incorporates an inference graph to infer the missing values which turns out to improve the performance significantly. [27] conducts question-query mapping and then query-KB matching which is most similar to our method. Our model is different from [27] in that we implement reinforcement learning to make both visual

processing and query generation more efficiently. Another strength of our method, again, is the introduction of inference graph which leads to powerful reasoning capability.

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 [31]. To make the paper self-contained, we cite them as follows (rephrased).

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.

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 "?".

3.2. Approach Overview

Figure 3 (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}, using the 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 contribution, we will elaborate modules QVS and VTP in Section 4, and modules VI in Section 5, 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 query-based policy. 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 Fig. 3 (b).

4.1.1 Multi-layer LSTM with Attention

The task here is to predict the most suitable action module sequence A by given a question Q and preference pre. We form the problem of seeking effective answering strategy based on question Q and preference pre as a sequence-to-sequence learning problem with attention mechanism. Inspired by [6], we input word feature of question w_i^q,

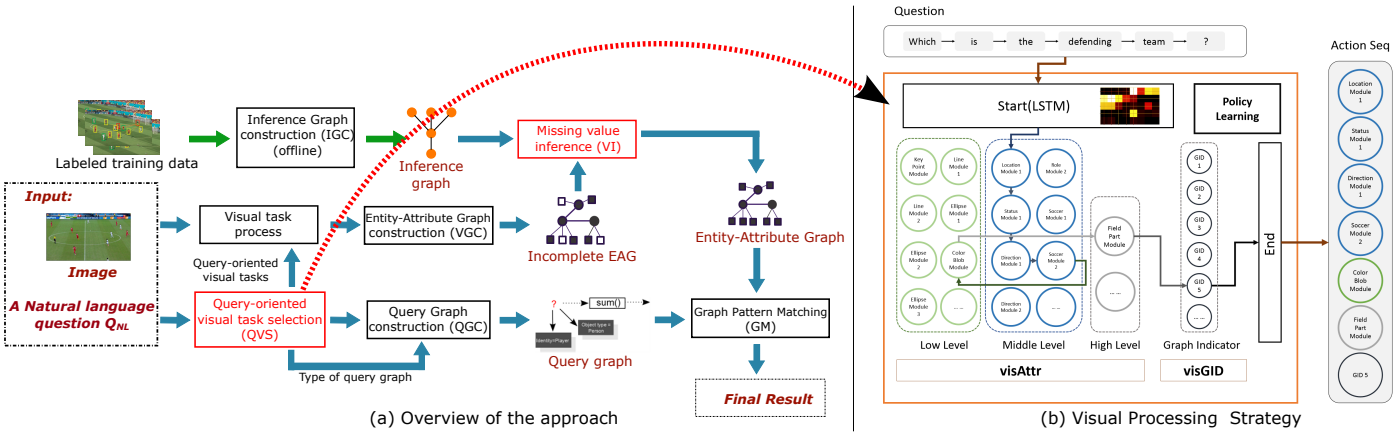

Figure 2: Overview of our approach, and graph-based representation of images and questions

$i \in \|Q\|$ into a LSTM network which is regarded as an encoder, and then output h_i as the hidden state for ith 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 weight, with higher a_{ij} in (i, j) pair, more attention will be paid in this correlation, thus the jth action module a_j in output will be more influenced by the ith input word w_i^q in question. The decode part is similar as traditional recurrent neural network (RNN). The following steps demonstrate the decoding part, aiming to get the joint distribution of the action module sequence $\mathsf{A} = [\mathsf{a}_1, \ldots, \mathsf{a}_t]$.

$$p(\mathsf{A}|Q) = \Pi_{t \in \|\mathsf{A}\|}\, p(\mathsf{a}_t|\{\mathsf{a}_1, \ldots, \mathsf{a}_{t-1}\}, c_i, Q) \tag{4}$$

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

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

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, which is explained in next section, 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 Tab. 1.

	Level	Sub-task	Descriptions
visAttr	Low	Keypoint Module	To get detailed information of the soccer field F_{keypoint}.
		Line Module	
		Ellipse Module	
		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 $P_{\text{direction}}$.
		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

The pool is consisted of two parts: the first one is *visAttr* which aims to discover the object attributes, while the other one *visGId* is an indicator showing which type of query graph can correctly represent the current question. There are three levels of *visAttr*: low, middle and high, which indicates the different difficulty level of the vision tasks.

For each vision task, the approach is not fixed, one task can be achieved by different methods with variety in time and accuracy. For instance, object detection based methods, like Faster R-CNN [23], R-FCN [8], SSD [18] or skeleton keypoints detection based method like [7] and [30] 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 [31]. Thus, the system is not only able to determine whether resemble 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 prefer-

ence. 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 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 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 $[\mathsf{a}_1, \mathsf{a}_2, \mathsf{a}_3, \ldots, \mathsf{a}_n]$. 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 will guide the vision tasks. 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 [17] 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 $L_{policy}(\theta)$.

Remark. After objects that are related to questions are identified, one can construct an Entity-Attribute graph, denoted as EAG, along the same line as [31].

5. Reasoning

According to our observation, an *incomplete* EAG isn't well satisfying of answering the question because of missing value of hidden attributes. To infer missing value of hidden attributes, an inference graph is required. 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 [9] 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:

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

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 [14]. 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:

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

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 [22]. As a consequence, the attribute class can be explicitly inferred by:

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

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, this simple model works excellently to our problem.

In the area of VQA, some simple questions may need the implicit attributes to help the answering system to find out the correct answer. Taking the soccer scene as description,

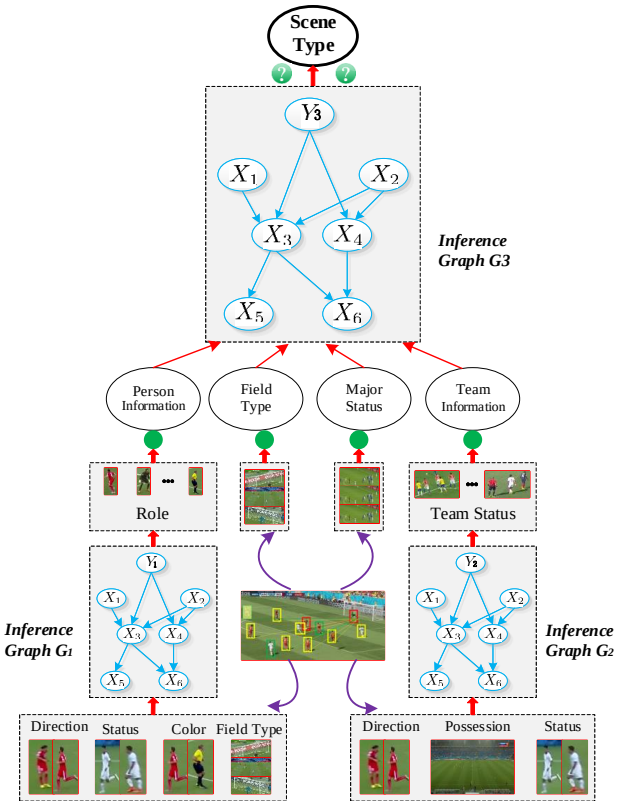

Figure 3: Schematic diagram of inference graph.

a closely related problem may be "How many players are there in the defending team at a free kick scene?". In this case, three target attributes are involved: the *role* of any detected person, the *team status*, as well as the *scene type*, all of which are implicit to us. Specifically, the inferring of the *scene type* is heavily relied on the other two hidden attributes: the *role* of a detected person and the *team status*. Therefore, our inference graph works in an indirect manner to reason the target information. Figure 4 summarizes the processes of our inference graph, where three graphs are constructed according to the difficult level of the task. First, the role of a detected person is inferred from the first inference graph 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 scene of the query image. The scene information is finally fed into the incomplete EAG so that a complete EAG can be obtained. The designed features will be discussed in the next section.

6. Experimental Studies

In this section, we conduct four sets of experiments to evaluate (1) the generalization ability of our model, (2) the performance of our question oriented visual processing module (QVS and VTP), (3) the accuracy of our inference module (VI), and (4) the overall performance of our model.

Experimental Setting.

DataSet. We used two data sets: (1) Soccer [31] that we annotated; and (2) Visual-Genome [2], one typical VQA dataset. Over Visual-Genome dataset, we extracted 2852 images with subjects of baseball, tennis and soccer. For each dataset, we split it into two parts: Part_I, which accounts for 1/3 and used for testing, and Part_{II} that accounts for 2/3 and used for training.

Questions. We used two sets of questions: (1) the set of questions, listed in Tab. 2 for Soccer; and (2) another set of questions for Visual-Genome. The questions for Visual-Genome are generated as following. We went over questions posed on Visual-Genome, extracted three kinds of typical questions, *i.e.* "what", "how many" and "where" as base questions, and constructed in total 1179 questions with base questions, where 869 for "what" type, 259 for "how many" type, and 51 for "where" type.

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?	
Q_{nl_9}	Is this image about penalty kick?	
$Q_{nl_{10}}$	Is this image about kick off?	
$Q_{nl_{11}}$	Is this image about free kick?	

Table 2: A set of questions

6.1. Generalization Ability of the Model

To show the generalization ability of our model, we compared accuracy of our model with state-of-the-art methods CNN+LSTM [5], HieCoAttenVQA [19] and Learning2Reason [13], via changes of training data. For comparison purpose, we enrich training data with those questions taking the same semantic meaning. For example, besides Q_{nl5}, we add following questions: *"Who is attacking team?"*, *"What is the uniform color of the defending team?"* and *"What is the uniform color of the attacking team?"*, that have the same semantic meaning as Q_{nl5} in training data. As shown in Tab. 3, our model always outperforms others, and moreover, is influenced least by varied training data, than other methods. The reason is that all the visual task selection is question oriented, which leads to unfixed network structure, also, the graph-based method

preserves better reasoning capability, comparing with memorizing statistic correlations between images and answers.

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 ability of our model

6.2. Performance of Visual Processing

Efficiency is one crucial factor to evaluate the performance of a VQA model. In light of this, we evaluate the running time over accuracy by using state-of-the-art methods CNN+LSTM [5] HieCoAttenVQA [19], Learning2Reason [13] and AllTasks [31]. To measure the efficiency of VQA models, we use "CPU Time" to represent the total running time of entire progress of question answering.

The results shown in Fig. 5 tell us that: (1) state-of-the-art methods perform efficiently, with running time less than 300 milliseconds, while their accuracy are all below 40%, showing that the methods are not applicable in practice; (2) AllTasks always has the highest and steady accuracy, *e.g.* 65.86%, since it processes all the visual related tasks without selection, paying the price of unexpected and long running time; and (3) our module finds a balance between efficiency and accuracy. For example, with $\gamma = 0.99$, our reinforcement learning based module spends, on average, 680 ms to answer questions, with accuracy 64.02%, which is 35.88%, 32.10% and 26.02% higher than that of CNN+LSTM, HieCoAttenVQA, and Learning2Reason, respectively. Moreover, when running time decreases, the accuracy decreases either, which is as expected.

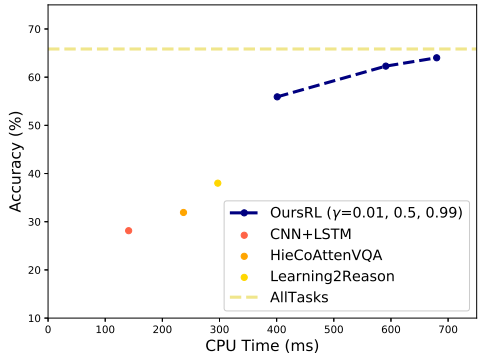

Figure 4: Balance between running time and accuracy

6.3. Performance of Inference

Accuracy is another crucial performance criteria of VQA systems. In contrast to end-to-end systems, the accuracy of our model is partly influenced by the inference quality. Hence, it is of great importance to see how accuracy is influenced by our inference module (VI). Prior work [31] already showed performance of their inference module w.r.t. a set of fixed questions on Soccer dataset. Here, we learn an inference graph for questions $Q_{nl8}, Q_{nl9}, Q_{nl10}$ and Q_{nl11} regarding field scene from Soccer dataset, and a set of inference graphs for answering questions on Visual-Genome dataset.

To evaluate accuracy, we follow F-measure [1], and define our accuracy metric as following:

$$\text{Acc} = \frac{2 \cdot (\text{recall} \cdot \text{precision})}{(\text{recall} + \text{precision})}$$

where:

$$\text{recall} = \frac{\#\text{true\_value\_inferred}}{\#\text{true\_value\_instance}}, \ \text{precision} = \frac{\#\text{true\_value\_inferred}}{\#\text{inferred\_instance}}$$

Here, #true_value_inferred is the number of all the instances, whose attribute A is inferred correctly as "v", #true_value_instance is the number of all the instances with attribute A of value "v", and #inferred_instance indicates the total number of instances whose attribute A is inferred as "v".

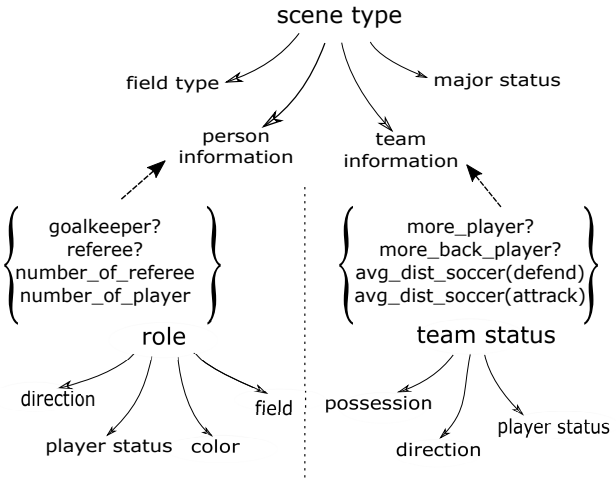

Figure 5: Attributes of scene inference graph.

Accuracy of Field Scene. We choose four typical field scenes, *i.e.* corner kick, free kick, penalty kick and kick-off, as which have distinct scene features. Practically, (1) corner kick scene always shows that a single player kicks the ball within a one-yard radius of the corner flag and most of players gather at the penalty field. (2) Free kick happens with a defensive wall mostly. (3) It is much typical for the penalty kick scene because most of players are out of penalty area except the penalty kicker and goalkeeper with

football in the penalty spot. (4) For kick-off scene, the soccer appears in the center spot and is surrounded by players from two opposing teams. Based on these observations, we design an inference graph which is shown in Fig. 6.

Using inference graph shown in Fig. 6, we have results showing accuracy of inference. As can be seen, the inference accuracy of corner kick, free kick, kick-off and penalty kick are 59.57%, 63.16%, 85.94% and 60.00%, respectively, among which the scene of kick off gains the highest inference accuracy.

Scene Type	Precision	Recall	Acc
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
Avg.	71.28	70.73	70.89

Table 4: Inference accuracy of four typical scenes (%).

In addition, we compare our approach with NuSVC, MLP, and AdaBoost in Tab. 5. 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	Acc
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 5: Accuracy comparison of field scene (%)

6.4. Overall Performance

We compare our approach with the following state-of-the-art methods: CNN+LSTM, HieCoAttenVQA, and Learn2Reason, using Soccer and Visual-Genome dataset. Table 6 lists results on Soccer. We find that (1) the average accuracy of our approach is higher than that of CNN+LSTM, HieCoAttenVQA and Learn2Reason by 20.35%, 17.64% and 15.67%, respectively; (2) our approach is typically effective in answering hard problems, *e.g.* with improvement over 60% for Q_{nl6}. Consider that questions $Q_{nl8} - Q_{nl11}$ are all about field scene, while more than 90% images of Soccer dataset are categorized as normal scene. For the fair comparison, we extract images that are categorized as one of four field scenes, and apply our approach over this subset of dataset. As shown in Tab. 6, our approach substantially outperforms others, which further verifies the advantage our approach preserves. Results on Visual-Genome are given in Tab. 7, from which we find: (1) for "what" and "where" questions, Learn2Reason performs best, while performances of four methods are quite close; (2) for "how many" questions, our method performs much better than others; and (3) the average accuracy our method surpasses CNN+LSTM, HieCoAttenVQA and Learn2Reason by 12.93%, 11.74% and 9.7%, respectively, which further verifies its advantage.

	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**
Q_{nl8}	72.43	75.14	80.15	**81.58**
Q_{nl9}	75.66	77.91	85.17	**89.47**
Q_{nl10}	77.94	76.83	81.23	**92.11**
Q_{nl11}	74.51	77.17	84.93	**89.47**
Avg.	55.63	57.51	55.24	**74.53**

Table 6: Performance comparison on Soccer dataset (%).

	CNN+LSTM	HieCoAtten	Learn2Reason	Ours
Q_{t1}	63.72	64.83	**67.12**	65.59
Q_{t2}	22.87	24.52	25.04	**82.87**
Q_{t3}	36.96	37.89	**40.12**	34.16
Avg.	50.48	51.67	53.71	**63.41**

Table 7: Performance comparison on Visual Genome dataset (%).

7. Conclusion

In this paper, we propose an innovative and efficient approach to handling the VQA problem. In the proposed method, the inputs (image and question) are first converted into entity-attribute graph and query graph, respectively; then the reinforcement learning based technique is utilized to identify correct query graph and related policies for visual processing; if information extracted from image is not sufficient to answer the question, an inference module will be invoked to infer crucial missing values; the final result will be computed by a module based on graph matching. Experimental studies show that our approach not only owns good generalization and inference ability, but also corroborates the efficiency and high accuracy when compared with other state-of-the-art baseline methods on two private and public VQA data sets. The problem of VQA has been widely studied while with slight satisfaction. We are currently exploring integrating external knowledge base to answer even more complex questions; another topic for future work is to develop techniques to automatically generate query graphs by using query logs.

References

[1] F-measure. *http://en.wikipedia.org/wiki/F-measure*. 7

[2] Visual genome dataset. http://visualgenome.org. 1, 6

[3] 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

[4] 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

[5] 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. 7

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

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

[8] 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. 5

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

[10] 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

[11] 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

[12] 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. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 1

[13] 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

[14] D. Koller, N. Friedman, and F. Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 5

[15] 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

[16] 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

[17] 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

[18] 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. 5

[19] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering, 2016. 7

[20] 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

[21] 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

[22] 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

[23] 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. 5

[24] 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

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

[26] 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. 3

[27] P. Wang, Q. Wu, C. Shen, A. Dick, and A. van den Hengel. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427, 2018. 3

[28] 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

[29] 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

[30] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016. 5

[31] P. Xiong, H. Zhan, X. Wang, B. Sinha, and Y. Wu. Visual query answering by entity-attribute graph matching and reasoning. http://120.25.121.173/research/vqa, 2019. 1, 3, 5, 6, 7

[32] 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

[33] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1039–1050, 2018. 3

[34] 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

[35] 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