Learning Visual Knowledge Memory Networks for Visual Question Answering

Zhou Su[1]*, Chen Zhu[2], Yinpeng Dong[3], Dongqi Cai[4], Yurong Chen[4], Jianguo Li[4]
[1]Tencent Wechat, [2]ShanghaiTech University, [3]Tsinghua University, [4]Intel Labs China
zhousu@tencent.com, zhuchen@shanghaitech.edu.cn, dyp17@mails.tsinghua.edu.cn
{dongqi.cai,yurong.chen,jianguo.li}@intel.com

Abstract

Visual question answering (VQA) requires joint comprehension of images and natural language questions, where many questions can't be directly or clearly answered from visual content but require reasoning from structured human knowledge with confirmation from visual content. This paper proposes visual knowledge memory network (VKMN) to address this issue, which seamlessly incorporates structured human knowledge and deep visual features into memory networks in an end-to-end learning framework. Comparing to existing methods for leveraging external knowledge for supporting VQA, this paper stresses more on two missing mechanisms. First is the mechanism for integrating visual contents with knowledge facts. VKMN handles this issue by embedding knowledge triples (subject, relation, target) and deep visual features jointly into the visual knowledge features. Second is the mechanism for handling multiple knowledge facts expanding from question and answer pairs. VKMN stores joint embedding using key-value pair structure in the memory networks so that it is easy to handle multiple facts. Experiments show that the proposed method achieves promising results on both VQA v1.0 and v2.0 benchmarks, while outperforms state-of-the-art methods on the knowledge-reasoning related questions.

1. Introduction

Visual Question Answering (VQA) is an emerging interdisciplinary research field in the last few years, which attracts broad attention from both computer vision and natural language processing (NLP) communities. VQA aims to help computer automatically answer natural language question about an image. The question answers can be divided into the following types: yes/no answers, multi-choice answers, numeric answers, and open-ended word/phrase answers (for questions about *what,where,who,...*). VQA requires comprehensive image and language understanding, which constitutes a truly AI-complete task with similar spirit as Turing test [15, 27, 28].

(a) (b) (c)

Figure 1: Examples of different question objective categories in VQA. (a) is an example of *apparent objective* that the answer is clearly visible and can be easily answered using recognition results. (b) is an example of *indiscernible objective* that the answer is unclear for visual recognition, and may need constraints from common sense. (c) is an example of of *invisible objective* that the answer requires deduction/reasoning from external knowledge.

Basically, VQA is formulated as a classification problem in most researches, in which images and questions are input, with answers as output categories (due to a limited number of possible answers). As the VQA task was proposed after deep learning approaches had already gained wide popularity, almost all current VQA solutions use CNN to model image input and recurrent neural network (RNN) to model the question [49]. Attention mechanism has been heavily investigated in VQA. This includes visual attention [46, 44, 35] which focuses on handling the problem *where to look*, and question attention [33, 47, 11, 26] which focuses on solving the problem *where to read*. As images and questions are two different modalities, it is straightforward to jointly embed two modalities together for a unified description of the image/question pair. Some works [14, 19, 20, 50, 48] even consider putting the attention mechanism and multi-modal joint embedding in one unified framework.

However, VQA is significantly more complex than other vision and language tasks such as image captioning, since explicit information (middle-level recognition results such as objects, attributes, or even image captions, etc) are not enough for accurate VQA. For instance, we investigated the

*This work was done when Zhou Su worked at Intel Labs China, and Yinpeng Dong was intern at Intel Labs China. Jianguo Li is the corresponding author.

VQA v1.0 dataset [2], which is based on the COCO dataset [24]. As each image has 5 manual caption annotations, we match question answers to words in image captions, which only produces 2,907 exact matches, while the remaining nearly 50K answers do not appear in captions. Basically, we could divide the question objectives of VQA into three categories: (a) *Apparent objective* which answers in the query image could be directly obtained from recognition results (objects, attributes, captions, etc); (b) *Indiscernible objective* which answer targets are usually too small or unclear in the query image, and thus requires supporting facts for correct answers; (c) *Invisible objective* which requires deduction of common sense, topic-specific or even encyclopedic knowledge about the content of the image. Please refer to Figure 1 for some examples of these three cases. The external knowledge information at least could help VQA on the latter two categories. A supporting data is that we find more than 49,866 answers in VQA v1.0 dataset appear in knowledge base from the visual genome [22]. A few pioneering works [39, 40, 42] study the problem on *how to reason* with prior knowledge information for the VQA task. They only involve one supporting fact to help the decision, which may introduce knowledge ambiguity/inaccuracy due to the inaccurate knowledge extraction procedure, and further yield wrongly question answering.

Before the VQA task appears, NLP community has extensively studied the text only question-answering (QA) problem [38, 21]. Classical methods read documents directly and use information retrieval methods to find answers [21]. Thereafter, knowledge base (KB) such as Freebase [7] organizes information into structured triples: $<s, r, t>$, where s is the subject, t is the target, and r is the relation between s and t. Then, question answering is converted into a database query problem [6, 12]. Most recently, memory networks have been proposed to combine document reading and knowledge base retrieval [41, 29, 36] for accurate QA.

Inspired by the development of memory networks based text QA methods, this paper proposes visual knowledge memory network (VKMN) for accurate *reasoning* with a pre-built visual knowledge base. Figure 2 illustrates how the proposed VKMN model works on visual question answering. VKMN extracts multiple related knowledge facts from the question, jointly embeds knowledge facts with visual attentive features into visual knowledge attentive features, and stores them in a key-value pair for easy and efficient reading from memory. The memory reading will make the visual-question addend to the highly relevant/correlated knowledge facts, and thus gives much more accurate question answering. The major contributions of this paper are:

(1) We propose VKMN, a simple yet efficient end-to-end trainable framework, which inherits the merits from attention based methods and joint-embedding based methods, while avoids the knowledge inaccuracy limitation of current knowledge-based solutions.

(2) We build a visual-question specific knowledge base, which does not contain irrelevant knowledge entries as generic knowledge base like Freebase [7].

(3) We conduct extensive experiments on the VQA v1.0 and v2.0 benchmark datasets, and show that the proposed method achieves competitive accuracy, while outperforms state-of-the-art methods on knowledge-reasoning related questions.

2. Related Work

Attention mechanisms in VQA. Attention Mechanisms in Neural Networks are loosely based on the visual attention mechanism found in human brains. It was first proposed and successfully applied in tasks like machine translation [5], image captioning [45], while then became popular in VQA. The main idea is that specific parts of the input (image and/or question) are more informative/effective than others for answering a given question. Numerous VQA methods [44, 35, 26, 46, 3] have incorporated spatial attention to learn specific CNN features according to the input question, rather than using holistic global features from the entire image. Some have also incorporated attention into the text representation. For instance, recent works [26] focus on the co-attention models that jointly exploit visual attention and question attention advantages through a unified hierarchical architecture. Furthermore, there are some researches combining attention mechanism with multi-modal representation through joint embedding visual attention features and text question features. This includes some state-of-the-art VQA methods: Multimodal Compact Bilinear pooling (MCB) [14], Multimodal Low-rank Bilinear pooling (MLB) [20], Multi-modal Factorized Bilinear pooling (MFB) [48], and Structured Visual Attention (SVA) [50].

This paper does not study the spatial attention mechanisms, while directly leverage existing multi-modal attention method as one component in our method.

Knowledge base and VQA. Knowledge base (KB) is a technology used to store structured fact entries in the triple form $<s, r, t>$, where s is the subject, t is the target, and r is the relation between s and t. There is an increasing interest in the NLP community using KBs for question answering [6, 12]. However, there are still limited works on leveraging KBs for VQA. As is known, VQA is an interdiscipline task in vision and language just like image/video captioning [45, 13, 34], but moves beyond for even deeper image understanding, since it often requires information not contained in the image itself, which can range from "common sense" knowledge to some topic-specific knowledge. For instance, understanding the image content is not enough to answer the question in Figure 1(c). The VQA system must first recognize that the "fruit" entity is "banana", and

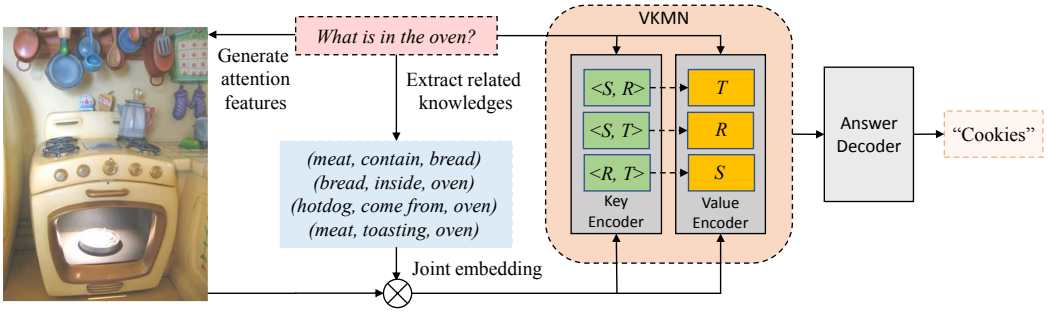

Figure 2: Illustration of VKMN for the VQA task. Note that three replicated memory sub-blocks (different combination of s, r, t as key-part or value-part) are used to handle the ambiguity on which part of the knowledge triple is missing in the query question.

inference based on knowledge about "animal loving banana" to get the answer "monkey". We also argue that for some cases such as Figure 1(b), even if the target content is visible, it may be too small and/or unclear to yield a wrong recognition result. KB will then act as an probabilistic prior to adjust the decision score on a list of candidates and output a correct result. We will verify these two cases with some qualitative examples in our experiments.

Substantial studies have focused on building large-scale structured Knowledge Bases (KBs) for QA, such as DBpedia [4], Freebase [7], ConcepNet [25], etc. Zhu *et al.* [51] even built large-scale multimodal knowledge base for VQA purpose. A few works [39, 40, 42] also tried to introduced KBs in the VQA task. Wang *et al.* [39] proposes "Ahab" method to reason about the content of an image, which first detects concepts in the query image and then links them to the relevant parts of DBpedia KB. It learns the mapping of images/questions to queries over the constructed knowledge graph for final question answering. This method is restricted to questions parsed with manually designed templates. [40, 42] improve "Ahab" by introducing long-short term memory (LSTM) and a data-driven approach to learn the mapping of images/questions to queries with the knowledge fact meeting the search conditions in a KB.

All these methods only involve one supporting knowledge fact to help the decision, so that they suffer greatly from the knowledge inaccuracy problem due to the inaccurate extraction procedure. This paper tries to alleviate this issue with memory network mechanism to handle multiple knowledge facts expanding from questions.

Memory networks. Memory networks were first proposed for modern large-scale question answering (QA) systems in [41], which introduces a long-term memory block for reading/writing simple facts when reasoning with inference components in the QA task. Sukhbaatar *et al.* [36] improve memory networks with an end-to-end learning fashion, which requires fewer supervision signals during training stage and has more practicality. To bridge the gap between KB query/inference and documents reading, Miller *et al.* [29] propose Key-Value Memory Networks (KV-MemNN) which can read documents and answer questions more viable and effective. The KV-MemNN performs QA by first storing facts in a key-value structured memory before reasoning over them for the answer.

Memory networks were later introduced into the VQA task in the form of dynamic memory networks [43], which consists of four modules: input module for encoding input with a set of "facts" vectors, question module for deriving a vector representation of the question, an episodic memory module for retrieval attention facts to the question, and answer module for combining final memory state and the question vector to predict the output. Similarly, spatial memory networks [44] also store CNN features from image grid regions/patches into memory, and selects certain parts of the information with an explicit attention mechanism for question answering.

We also try to utilize memory network for VQA. We have at least two differences to [43, 44]. First, we design a key-value memory network for visual knowledge attentive features rather than a simple dynamic memory network as in [43, 44]. Second, the memory network we used focuses more on multiple visual knowledge encoding, while [44] focuses more on spatial patches encoding.

3. Method

3.1. Overview of the Method

We propose the visual knowledge memory network (VKMN) aiming to answer the visual question more accurately with an auxiliary visual knowledge base. VKMN is derived from key-value memory network [29], which has been proven effective in the QA task. VKMN constructs a visual-question related knowledge base in advance, which allows great flexibility for designing key and value items based on the visual features and (part-of) knowledge triples. Basically, our VKMN model consists of four modules:

(1) *Input module*, which encodes input images with CNN model and questions with a RNN model, and further obtains the query representation by jointly embedding the output of these two models;

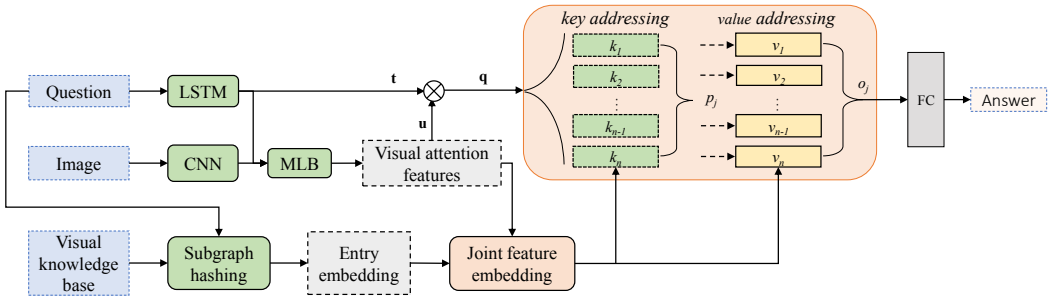

Figure 3: Diagram of Visual Knowledge Memory Network based VQA system

(2) *Knowledge spotting module*, which retrieves related knowledge entries based on query questions or automatic image captions from the pre-built visual knowledge base by sub-graph hashing;

(3) *Joint visual and knowledge embedding module*, which embeds visual features and knowledge triples (or part of the triples) jointly for the ease of storing in key-value memory networks;

(4) *Memory module*, which receives query question, reads the stored key-value to form the visual-knowledge feature, and predicts the answers.

Figure 2 illustrates how the proposed VKMN model works on visual question answering, and Figure 3 gives detailed diagram on how different modules interact in VKMN. We will elaborate each module separately below.

3.2. Encoding of Image/Question Inputs

The input image I and question q need to be processed into feature vectors before feeding into memory networks. We process input images with an ImageNet pre-trained CNN model, and questions with a LSTM model. The feature vectors from two modalities could be jointly embedded into a single visual attentive description as defined in Eq.1 for answer prediction. Several methods are proposed to learn the multimodal joint embedding in an end-to-end manner for VQA, including the VQA 2016 challenge winner solution MCB [14], and the state-of-the-art solution MLB [20].

In this paper, we directly leverage MLB for visual-question pair encoding. We denote the MLB with spatial attention output (aka the first MLB stage output) as $\mathbf{u}$, and the LSTM encoded question vector as $\mathbf{t}$, where $\mathbf{t}$ is already projected to the same dimensional space as $\mathbf{u}$ with one internal fully-connection (FC) layer, i.e., $\mathbf{t}, \mathbf{u} \in \mathcal{R}^d$. The query representation is the joint embedding of $\mathbf{t}$ and $\mathbf{u}$ with low-rank bilinear pooling [20] as

$$\mathbf{q} = \mathbf{t} \odot \mathbf{u}, \tag{1}$$

where $\odot$ indicates Hadamard product between two vectors (element-wise product). $\mathbf{q}$ is the visual attentive description of the visual-question pair for query purpose.

3.3. Knowledge Spotting with Sub-graph Hashing

Before elaborating the details of visual knowledge representation, we present how to spot knowledge entries related to the visual question. Firstly, given all the knowledge triples $<s_i, r_i, t_i>$ in the pre-built Visual Knowledge Base, we generate the entity set $E = \{s_i, t_i\}$, and relation set $R = \{r_i\}$. We call $S = E \cup R$ the entry set, which contains all the different entries in the knowledge base. Then we extract entries whenever one phrase in questions (or automatic generated captions) matches one item in the entry set S using sub-graph hashing methods [8]. To alleviate the inaccuracy of extracted visual knowledge, we restrict that each knowledge triple should contain at least two entries extracted from a question (or automatic generated captions), which is more stringent than that in MemNN [41, 36] and KV-MemNN [29]. A small subset of N knowledge triples $\{<s_1, r_1, t_1>, \cdots, <s_N, r_N, t_N>\}$ are created afterwards. To handle the long-tail effects in the visual knowledge base, we perform knowledge triple expansion on the knowledge graph to include the direct neighborhood of those extracted N knowledge triples. Figure 4 illustrates one example of subgraph hashing and knowledge triple expansion. Finally, we setup a memory network which could store M knowledge entries ($M > N$). If the size of expanded knowledge subset is less than M, we append null entries.

3.4. Joint Embedding of Visual and Knowledge

In the QA task [29], people just take the knowledge triple or part-of-knowledge triple to compose the key-value pair in key-value memory networks. In VQA, visual contents are essential to answering the visual questions. Our start point is the spatial attentive visual feature $\mathbf{u}$ from the input module and knowledge entry $\mathbf{e}$ from knowledge spotting module. We are required to learn a joint embedding to combine $\mathbf{u} \in \mathcal{R}^d$ and $\mathbf{e}$ together. As $\mathbf{e}$ is text representation, we impose a mapping function $\Phi(\cdot)$ to get real-valued feature vector $\Phi(\mathbf{e}) \in \mathcal{R}^{d_e}$. Here $\Phi(\cdot)$ could be either bag-of-words (BoW) representation, word2vec transformation, or even knowledge embedding like TransE [9]. The feature dimension of $\mathbf{u}$ and $\Phi(\mathbf{e})$ are usually different, i.e., $d \neq d_e$. We project them into the same space, and apply MLB [20]

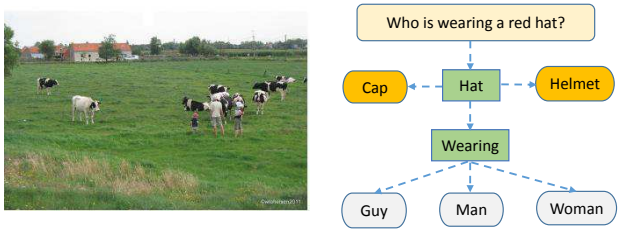

Figure 4: Illustration of knowledge subgraph expansion. We do question sentence parsing to get the target 'hat' and relation 'wearing'. We expand the subject with query from knowledge-base (results in yellow), and also expand the target (results in gray) based on subject plus relation. Dashed-line indicates expansion results.

to capture their joint representation as

$$\mathbf{x} = \Psi(\mathbf{e}, \mathbf{u}) = \sigma(W_e \Phi(\mathbf{e})) \odot \sigma(W_u \mathbf{u}), \tag{2}$$

where $\sigma(\cdot)$ is hyperbolic tangent function (favors over sigmoid function in our experiments), W_u and W_e are the matrices to project $\mathbf{u}$ and $\Phi(\mathbf{e})$ into the same dimensional space. $\mathbf{x}$ is called visual knowledge attentive description, since it integrates visual feature $\mathbf{u}$ with knowledge entry $\mathbf{e}$.

We argue that $\mathbf{x}$ captures more fine-grained information about the knowledge than the spatial attentive visual feature $\mathbf{u}$ from MLB, in which the attention is based on the whole question. We verified this point in the ablation study (see row 4 & 6 of Table 1).

3.5. Visual Knowledge Memory Network

We presented general joint embedding of visual features and knowledge entries in previous subsection. As VKMN defines memory slots as key-value vector pairs like $(\mathbf{k}_1, \mathbf{v}_1), \cdots, (\mathbf{k}_M, \mathbf{v}_M)$, in this subsection, we discuss the key-value pair design first, and the inference procedure with memory networks later.

Key-value design. The design of which part should be key, and which part should be value is quite flexible. In traditional QA problem, given the knowledge triple $<s,\ r,\ t>$, people usually consider the first two (s and r) as key, while take the last one t as value. However, this is not true for VQA, as we do not know which part of the knowledge triple is missing in the visual question. There are three combinations of keys while taking the remained item as value: (1) (s, r) as the key; (2) (s, t) as the key; (3) (r, t) as the key. In practice, we build three memory blocks for these three cases separately as shown in Figure 2, and name it *triple replication*. This is useful to distinguish questions like "what is the toothbrush used for?" and "what is used for brushing teeth?". For simplicity, we only elaborate the case (s, r) as the key item and t as the value item in the following study. We can employ Eq.2 to obtain the embedding of keys and values. Suppose $\mathbf{e} = (e_1, e_2, e_3)$, in which e_1, e_2 and e_3 correspond to s, r, t according to the designed key-value mapping. To ensure that key representation $\mathbf{k}_i$ and value representation $\mathbf{v}_i$ are of the same dimensionality, we make the additive assumption similar to the continuous bag-of-words (CBOW), and derive the $\mathbf{k}_i$ and $\mathbf{v}_i$ as below:

$$\mathbf{k}_i = \Psi(e_1, \mathbf{u}_i) + \Psi(e_2, \mathbf{u}_i); \tag{3}$$

$$\mathbf{v}_i = \Psi(e_3, \mathbf{u}_i). \tag{4}$$

This additive assumption also suits for TransE encoding [9], since it adopts shared weights to encode the knowledge triple $<s,\ r,\ t>$ as a whole, and outputs the same dimensional feature vector for s, r, and t respectively.

With the designed key-value pairs stored in VKMN, the inference consists of three steps: addressing related knowledge, reading corresponding value and answering the question. We discuss them step by step below.

Key addressing. Given a query $\mathbf{q}$, we address each candidate memory slot and assign a relevance probability by comparing the question to each key:

$$p_i = Softmax(\mathbf{q} \cdot A\mathbf{k}_i), \tag{5}$$

where $\cdot$ denotes inner product, A is the parameter matrix for memory networks which projects $\mathbf{k}_i$ into the same feature space as $\mathbf{q}$, and $Softmax(z_i) = e^{z_i}/\sum_j e^{z_j}$.

Value reading. In the value-reading step, the values of memory slots are read by weight averaging with the addressing possibilities, and the returned vector $\mathbf{o}$ is defined as:

$$\mathbf{o} = \sum\nolimits_i p_i A\mathbf{v}_i \tag{6}$$

The original key-value memory networks [29] supports iterative refining the query and reading the memories. In this paper, we only make one-step update of query with $\mathbf{q}' = \mathbf{q} + \mathbf{o}$ after receiving $\mathbf{o}$.

Question answering. The number of different answers is fixed for the VQA task so that the question answering is re-casted to be a classification problem. We predict the answer based on $\mathbf{q}'$ using a FC layer with weight matrix W_o as below

$$\hat{a} = \arg\max Softmax(W_o \mathbf{q}'). \tag{7}$$

All the parameter matrix W_u, W_e, A, and W_o in VKMN are end-to-end trained with stochastic gradient descent (SGD) based backpropagation algorithm.

4. Experiments

We evaluate the proposed model on the VQA v1.0 [2] and v2.0[16] datasets. We will elaborate the details of the dataset, the procedure of building visual knowledge base, ablation studies on different configurations and the benchmark results below separately.

4.1. Datasets

The *VQA v1.0* dataset [2] consists of 204,721 MS-COCO images [24], with 614,163 associated question-answer pairs. There are three designed data splits: train (248,349 questions), validation (121,512 questions) and test (244,302 questions). In addition, 25% subset of test partition is splitted as test-dev set.

The *VQA v2.0* dataset [16] further balanced the v1.0 dataset by collecting complementary images such that each question in the balanced dataset has a pair of similar images but with different answers to the question. It finally consists of 1,105,904 questions from 204,721 images, which is about twice larger than v1.0 dataset.

For simplicity, we restricted the target answers to the top 2,000 most frequent answers in the train+val set as most existing works [20, 14]. Therefore, our training data consists of data with the top 2,000 answers in the train+val set, which covers about 90% question-answer pairs in the train+val set.

The *Visual Genome* dataset [22] contains 108,249 images labeled with question-answer pairs, objects and attributes. An average of 17 question-answer pairs are collected for each image. Moreover, there are *Visual Genome Relationship* (VGR) annotations available. *VGR* provides relationship descriptions of different objects on images in the triple form $<s,\ r,\ t>$, which perfectly fits the needs of our visual knowledge base. Detailed usage settings are given in section 4.2.

4.2. Building Visual Knowledge Base

As the general purpose knowledge bases like Freebase contains billions of knowledge facts, while most knowledge entries are irrelevant to the visual questions, we build our own knowledge base for the purpose of VQA, in which each entry has a structure of $<s,\ r,\ t>$. The entries in the knowledge base (named visual knowledge base) come from two sources: (i) knowledge entries extracted from the question-answer pairs in the VQA v1.0 train+val set [2]; (ii) knowledge triples from the *VGR* dataset [22].

First, we perform information extraction from the question-answer pairs into the VQA v1.0 train+val set, to obtain a bunch of structured knowledge entries. Compared to existing knowledge bases like DBpedia [4] and ConceptNet [25], knowledge entries we extracted are more closely bound up with questions associated with their images, and the expression is more precise and compact than original question-answer pairs in the VQA dataset [2]. The knowledge entry extraction procedure works as below. We first parse the part-of-speech (POS) tags and dependency tree of each question via the Stanford Parser [10]. Then we extract triples from QA pairs using question tags, dependencies, and answers. All words in the triples are lemmatized so that inflectional variants of the same word explicitly share the same representation. Finally, we build a *relation set* $R = \{r\}$, where r is all possible relation phrases in *VGR* dataset[22]. For each extracted triple $<s,\ r,\ t>$ from the VQA dataset, we replace r with the most similar/closed one in set R. Lemmatisation and relation replacement can help reducing data sparseness, especially for capturing "long-tail" entries.

Second, the *VGR* dataset [22] consists of 1,531,448 knowledge triples describing object relationships, about 14 triples per image. We filter the original triple sets in VGR dataset to obtain 40,480 unique triples by removing triples with appearance frequency (either s, r, or t) less than 3.

After combining these two parts together, we obtain a visual knowledge base with about 159,970 unique triples.

Model	All	Y/N	Num	Other
(1) BoW (GLoVe)	65.5	83.8	38.4	56.1
(2) Blind model (text only)	49.3	77.9	34.3	28.3
(3) $\mathbf{q}$ only no KB (MLB) [20]	65.1	84.1	38.2	54.9
(4) No triple replication	64.5	84.0	37.3	53.9
Baseline (full VKMN)	66.0	83.7	37.9	57.0

Table 1: Ablation studies of the design choice on the *Open-Ended* task of the VQA v1.0 test-dev dataset.

4.3. Implementation Details

We implement our model using the Torch framework with RNN packages from [23]. In the input module, we followed MLB [20] and used ResNet-152[17] as the backbone network for visual feature extraction. The MLB attention feature vector $\mathbf{u}$ is of 2,400 dimension, and the question embedding vector $\mathbf{t}$ is also projected to the same dimension. The joint visual and knowledge embedding module output 300-dimensional features by $\Phi(e)$. We tried different embedding methods, such as BOW (GloVe) [32], and TransE [9], and finally pick TransE [9]. In the memory module, we set the memory-slot number to 8. If the extracted subgraph size is less than 8, we fill empty slots with zeros.

4.4. Ablation Studies

There are several configurations of VKMN which may impact the final accuracy, including knowledge encoding method $\Phi(\cdot)$, blind model with text feature only, and the triple replicated memory blocks, etc. We examine the impact of these configurations with evaluation on the Real Image Open-Ended task using the *test-dev* split of the VQA v1.0 dataset. To avoid interaction between these factors, we adopt the variable-controlling approach to ease our study. We take our designed model as baseline, which uses TransE as knowledge encoding, visual attentive feature $\mathbf{u}$ plus knowledge embedding $\mathbf{e}$ to build the keys and values in VKMN, and the triple replicated memory-blocks as discussed in Section 3.5 to avoid the ambiguity from knowledge extraction from question.

Model	test-dev					test-standard				
	Open-Ended				MC	Open-Ended				MC
	All	Y/N	Num	Other	All	All	Y/N	Num	Other	All
iBOWIMG[49]	55.7	76.5	35.0	42.6	-	55.9	76.78	35.0	42.6	62.0
DPPnet[31]	57.2	80.7	37.2	41.7	-	57.4	80.23	36.9	42.2	62.7
SMem[44]	58.0	80.9	37.3	43.1	-	58.2	80.9	37.5	43.5	-
AMA[42]	59.2	81.0	38.4	45.2	-	59.4	81.1	37.1	45.8	-
DMN+[43]	60.3	80.5	36.8	48.3	-	-	-	-	-	-
MRN[19]	61.6	82.2	38.8	49.2	-	61.8	82.4	38.2	49.4	66.3
HieCoAtt[26]	61.8	79.7	38.7	51.7	65.8	62.1	80.0	38.2	52.0	66.1
RAU[31]	63.3	81.9	39.0	53.0	67.7	63.2	81.7	38.2	52.8	67.3
DAN[30]	64.3	83.0	39.1	53.9	**69.1**	64.2	82.8	38.1	54.0	69.0
MCB+Att[14]	64.2	82.2	37.7	54.8	-	-	-	-	-	-
MCB+Att+GloVe[14]	64.7	82.5	37.6	56.6	-	-	-	-	-	-
MLB [20]	65.1	84.1	38.2	54.9	-	-	-	-	-	-
MFB+CoAtt+GloVe [48]	65.9	84.0	**39.8**	56.2	-	65.8	83.8	**38.9**	56.3	-
SVA [50]	**66.0**	**84.3**	39.3	56.4	-	65.9	**84.4**	**38.9**	55.9	-
VKMN(ours)	**66.0**	83.7	37.9	**57.0**	**69.1**	**66.1**	84.1	38.1	**56.9**	**69.1**

Table 2: Results comparison with state-of-the-art methods on the VQA v1.0 dataset. "-" indicates the result is not available. "Att" indicates some attention mechanism is used. "CoAtt" means co-attention. "GloVec" indicates that the word embedding method [32] is adopted. Here all the reported results are obtained with a **single model** without model ensemble. For each answer-type, the best result is **bolded**.

We evaluate the contribution of each component by either removing it or replacing it with another commonly used component, with all the other components fixed.

(1) We replace TransE based knowledge encoding in our design with BOW encoding [32]. Results show that TransE performs better than BOW, especially on the "other" answer-type (57.0 vs 56.1).

(2) We remove the visual input and design a *blind model* without any visual cues, so that key/value are only text features, i.e., replacing Eq.2 with

$$\mathbf{x} = \Psi(\mathbf{e}, \mathbf{t}) = \sigma(W_e \Phi(\mathbf{e})) \odot \sigma(W_t \mathbf{t}),$$

and directly using $\mathbf{t}$ as the query. This comparison verifies the importance of visual attentions in VKMN.

(3) We directly use the joint embedding $\mathbf{q} = \mathbf{t} \odot \mathbf{u}$ for answer prediction without using memory blocks. This is actually the result by MLB. It is obvious that VKMN outperforms MLB, especially on the "other" answer-type (57.0 vs 54.9). The difference is *significant* according to the paired t-test. This study verifies the effectiveness of the proposed memory network module.

(4) We disable the triple replication mechanism in memory networks, while only use s and r to build the key and t as the value. This study shows that the triple replication mechanism is important to avoid the ambiguity in visual-question pair, especially on the "other" answer-type (57.0 vs 53.9).

Table 1 lists the detailed results of the four cases in comparison to our designed model (baseline) on different question categories. This ablation study verifies the effectiveness of our design choice.

4.5. Benchmarking Results

We further list the full benchmark results on both test-dev and test-standard dataset of the VQA v1.0 dataset in Table 2. For easy and fair comparison, we also list the results by state-of-the-art methods with single model. It shows that VKMN outperforms state-of-the-art results on both *Open-Ended* and *Multiple-choice* tasks, and especially for the "other" answer-type, which proves that VKMN is effective on incorporating external knowledge for answering 6W questions *(what, where, when, who, why, and how)*. Figure 5 further illustrates some quantitative examples, in comparison to the state-of-the-art method MLB. Below each example, we also show the confidence score of top-5 knowledge triples according to Eq.5. It is obvious that the VKMN model could attend to highly relevant knowledge triples. Note that although some top-1 triple is relevant but not quite accurate (due to appearance frequency in training-set), the final decision is based on softmax classifier (Eq.7) with weight averaging knowledge representation (Eq.6), which tends to produce right answers. Besides, we show some failure cases in Figure 6 along with the MLB attention maps. These cases are related to spatial relationship reasoning, in which MLB does not get correct attention regions. Problem may be alleviated when resorting to some advanced attention mechanism such structure attention [50].

We further evaluated our method on VQA v2.0 dataset, which is about twice larger than VQA v1.0 on question number, and much more difficult than VQA v1.0. Table 3 listed the comparison results on test-standard set. Single

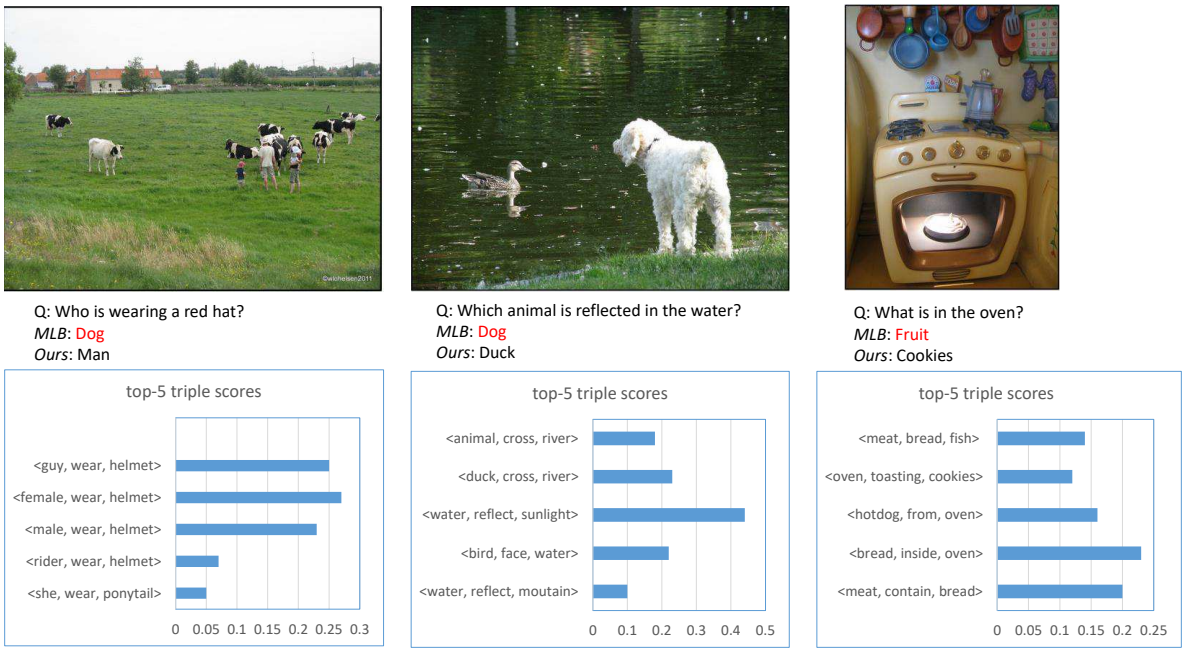

Figure 5: Examples of the predicted answers and top-5 knowledge triple scores for given query images and questions. The predicted answers by MLB (used as our attention module) is also given as comparison.

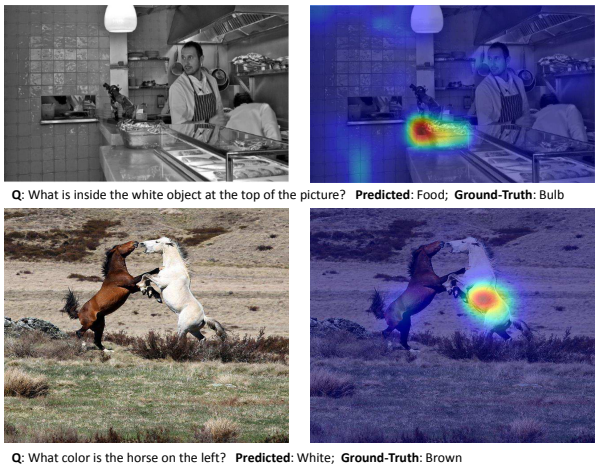

Figure 6: Some failure cases with attention maps. Both cases are due to wrong attention regions.

VKMN model achieves 64.36 overall accuracy, which is much better than MCB and MLB (ours retrained model and result submitted by the DCD-ZJU team), especially on the "other" answer-type. Our ensemble result is based on the simplest snapshot ensemble [18] of our VKMN model and the SVA model [50] as we observed that VKMN and SVA are complementary on different answer-types. Although the top-3 solutions in the VQA v2.0 challenge show better accuracy than our VKMN, they heavily rely on model ensemble, while their best results are obtained by ensembling dozens or even hundreds of models. Even for the first place solution [37], their best single model result is still much worse than that of our single VKMN model, especially on the "other" answer-type, when leaving out the exhaustive bottom-up attention from the object detection results.

Model	All	Y/N	Num	Other
Adelaide-Teney-MSR (30 ensemble)[37]	69.13	85.54	47.45	59.82
Adelaide-Teney-MSR (best single + bottom-up)[37]	65.67	82.20	43.90	56.26
Adelaide-Teney-MSR (best single)[37]	62.27	79.32	39.77	52.59
DLAIT [1]	68.22	83.17	46.66	60.15
HDU-USYD-UNCC [1]	68.09	84.5	45.39	59.01
MCB	62.27	78.82	38.28	53.36
MLB (by DCD-ZJU)[1]	62.54	79.85	38.64	52.95
MLB (our retrained single model, +VG)	63.50	77.73	39.12	56.98
SVA (our retrained single model) [50]	64.55	80.76	42.59	55.73
VKMN (Single)	64.36	83.70	37.90	57.79
VKMN (Ensemble)	66.67	82.88	43.17	57.95

Table 3: Results on VQA v2.0 test-standard set. The first group of results are from top-3 solutions in the VQA v2.0 challenge. The second group of results are from some attention based methods related to us. For our retrained MLB model on VQA v2.0, we include the external visual genome (VG) data. The final group of results are from our VKMN.

5. Conclusion

In this paper, we present the Visual Knowledge Memory Network (VKMN) method as an efficient way to leverage pre-built visual knowledge base for accurate visual question answering. Experiments show that VKMN achieves promising results on VQA v1.0 and v2.0 benchmarks, and outperforms state-of-the-art methods on the knowledge-reasoning related questions (i.e., the "other" answer-type in both benchmarks).

Acknowledgements Thanks Zhiqiang Shen for the help of preparing some illustrations for our early submissions.

References

[1] Vqa 2.0 challenge leaderboard, 2017. http://www.visualqa.org/roe_2017.html. 8

[2] A. Agrawal, J. Lu, S. Antol, M. Mitchell, L. Zitnick, et al. VQA: Visual Question Answering. *IJCV*, 2016. 2, 5, 6

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *CVPR*, 2016. 2

[4] S. Auer, C. Bizer, G. Kobilarov, et al. Dbpedia: A nucleus for a web of open data. *The semantic web*, 2007. 3, 6

[5] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[6] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, 2013. 2

[7] K. Bollacker, C. Evans, P. Paritosh, et al. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, 2008. 2, 3

[8] A. Bordes, S. Chopra, and J. Weston. Question answering with subgraph embeddings. In *EMNLP*, 2014. 4

[9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013. 4, 5, 6

[10] D. Chen and C. D. Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *EMNLP*, 2014. 6

[11] C. N. dos Santos, M. Tan, B. Xiang, and B. Zhou. Attentive pooling networks. *CoRR, abs/1602.03609*, 2016. 1

[12] A. Fader, L. Zettlemoyer, and O. Etzioni. Open question answering over curated and extracted knowledge bases. In *SIGKDD*, 2014. 2

[13] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, et al. From captions to visual concepts and back. In *CVPR*, 2015. 2

[14] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, et al. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 1, 2, 4, 6, 7

[15] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12), 2015. 1

[16] Y. Goyal, T. Khot, D. Summers-Stay, et al. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, 2017. 5, 6

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[18] G. Huang, Y. Li, G. Pleiss, Z. Liu, et al. Snapshot ensembles: Train 1, get m for free. In *ICLR*, 2017. 8

[19] J.-H. Kim, S.-W. Lee, D.-H. Kwak, et al. Multimodal Residual Learning for Visual QA. In *NIPS*, 2016. 1, 7

[20] J.-H. Kim, K.-W. On, J. Kim, J.-W. Ha, and B.-T. Zhang. Hadamard product for low-rank bilinear pooling. In *ICLR*, 2017. 1, 2, 4, 6, 7

[21] O. Kolomiyets and M.-F. Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181, 2011. 2

[22] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*, 2017. 2, 6

[23] N. Léonard, S. Waghmare, Y. Wang, and J.-H. Kim. RNN : Recurrent Library for Torch. *arXiv preprint arXiv:1511.07889*, 2015. 6

[24] T. Y. Lin, M. Maire, S. Belongie, J. Hays, et al. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2, 6

[25] H. Liu and P. Singh. Conceptnet – a practical commonsense reasoning toolkit. *BT technology journal*, 22(4), 2004. 3, 6

[26] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 1, 2, 7

[27] M. Malinowski and M. Fritz. Towards a visual turing challenge. *arXiv preprint arXiv:1410.8027*, 2014. 1

[28] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 1

[29] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston. Key-Value Memory Networks for Directly Reading Documents. In *EMNLP*, 2016. 2, 3, 4, 5

[30] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2017. 7

[31] H. Noh, P. H. Seo, and B. Han. Image Question Answering using Convolutional Neural Network with Dynamic Parameter Prediction. In *CVPR*, 2016. 7

[32] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 6, 7

[33] T. Rocktäschel, E. Grefenstette, et al. Reasoning about entailment with neural attention. In *ICLR*, 2016. 1

[34] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y.-G. Jiang, and X. Xue. Weakly supervised dense video captioning. In *CVPR*, 2017. 2

[35] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016. 1, 2

[36] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *NIPS*, 2015. 2, 3, 4

[37] D. Teney, P. Anderson, X. He, and A. v. d. Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *arXiv preprint arXiv:1708.02711*, 2017. 8

[38] E. M. Voorhees and D. M. Tice. Building a question answering test collection. In *SIGIR*, 2000. 2

[39] P. Wang, Q. Wu, et al. Explicit knowledge-based reasoning for visual question answering. In *IJCAI*, 2017. 2, 3

[40] P. Wang, Q. Wu, C. Shen, et al. Fvqa: Fact-based visual question answering. *IEEE TPAMI*, 2017. 2, 3

[41] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. 2, 3, 4

[42] Q. Wu, P. Wang, C. Shen, A. Dick, et al. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *CVPR*, 2016. 2, 3, 7

[43] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016. 3, 7

[44] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 1, 2, 3, 7

[45] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, et al. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[46] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 2

[47] W. Yin, H. Schütze, B. Xiang, and B. Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*, 2015. 1

[48] Z. Yu, J. Yu, J. Fan, and D. Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, 2017. 1, 2, 7

[49] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple Baseline for Visual Question Answering. *arXiv preprint arXiv:1512.02167*, 2015. 1, 7

[50] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma. Structured attentions for visual question answering. In *ICCV*, 2017. 1, 2, 7, 8

[51] Y. Zhu et al. Building a large-scale multimodal knowledge base system for answering visual queries. *arXiv preprint arXiv:1507.05670*, 2015. 3