This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Learning Answer Embeddings for Visual Question Answering

Hexiang Hu*
U. of Southern California
Los Angeles, CA
hexiang.frank.hu@gmail.com

Wei-Lun Chao*
U. of Southern California
Los Angeles, CA
weilunchao760414@gmail.com

Fei Sha
U. of Southern California
Los Angeles, CA
feisha@usc.edu

Abstract

We propose a novel probabilistic model for visual question answering (Visual QA). The key idea is to infer two sets of embeddings: one for the image and the question jointly and the other for the answers. The learning objective is to learn the best parameterization of those embeddings such that the correct answer has higher likelihood among all possible answers. In contrast to several existing approaches of treating Visual QA as multi-way classification, the proposed approach takes the semantic relationships (as characterized by the embeddings) among answers into consideration, instead of viewing them as independent ordinal numbers. Thus, the learned embedded function can be used to embed unseen answers (in the training dataset). These properties make the approach particularly appealing for transfer learning for open-ended Visual QA, where the source dataset on which the model is learned has limited overlapping with the target dataset in the space of answers. We have also developed large-scale optimization techniques for applying the model to datasets with a large number of answers, where the challenge is to properly normalize the proposed probabilistic models. We validate our approach on several Visual QA datasets and investigate its utility for transferring models across datasets. The empirical results have shown that the approach performs well not only on in-domain learning but also on transfer learning.

1. Introduction

Visual question answering (Visual QA) has made significant progress in the last few years. More than 10 datasets have been released for the task [10, 14, 25], together with a number of learning models that have been narrowing the gap between the human's performance and the machine's [28, 7, 18, 15, 27].

In this task, the machine is presented with an image and a related question and needs to output a correct answer. There are several ways of "outputting", though. One way is to ask the machine to generate a piece of free-form texts [8]. However, this often requires humans to decide whether the answer is correct or not. Thus, scaling this type of evaluation to assess a large amount of data (on a large number of models) is challenging.

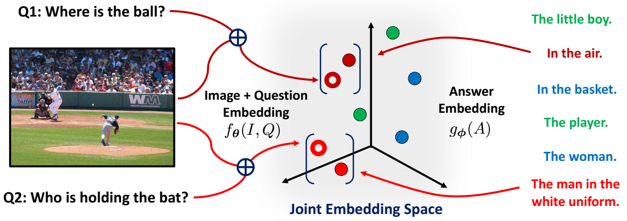

Figure 1. Conceptual diagram of our approach. We learn two embedding functions to transform image question pair (i, q) and (possible) answer a into a joint embedding space. The distance (by inner products) between the embedded (i, q) and a is then measured and the closest a (in red) would be selected as the output answer.

Automatic evaluation procedures have the advantage of scaling up. There are two major paradigms. One is to use multiple-choice based Visual QA [32, 1, 22]. In this setup, for each pair of image and question, a correct answer is mixed with a set of incorrect answers and the learner optimizes to select the correct one. While popular, it is difficult to design good incorrect answers without bias such that learners are not able to exploit [5]. Several recent papers [5, 13] have shown that even when the image or the question is missing, the correct answer can still be identified (using the incidental statistics, *i.e*., bias, in the data).

The other paradigm that is amenable to automatic evaluation revises the pool of possible answers to be the same for any pair of image and question [9, 3], *i.e*., open-ended Visual QA. In particular, the pool is composed of most frequent K answers in the training dataset. This has the advantage of framing the task as a multi-way classifier that outputs one of the K categories, with the image and the question as the input to the classifier.

However, while alleviating the bias of introducing incorrect answers that are image and question specific, the open-end Visual QA approaches also suffer from several prob-

[*] Equal contributions

lems. First, treating the answers as independent categories (as entailed by the multi-way classification) removes the semantic relationship between answers. For example, the answers of "running" and "jogging" (to the question "what is the woman in the picture doing?") are semantically close, so one would naturally infer the corresponding images are visually similar. However, treating "running" and "jogging" as independent categories "choice i" and "choice j" would not automatically regularize the learner to ensure the classifier's outputs of visually similar images and semantically similar questions to be semantically close. In other words, we would desire the outputs of the Visual QA model express semantic proximities aligned with visual and semantic proximities at the inputs. Such alignment will put a strong prior on what the models can learn and prevent them from exploiting biases in the datasets, thus become more robust.

Secondly, Visual QA models learned on one dataset do not transfer to another dataset unless the two datasets share the same space of top K answers—if there is a difference between the two spaces (for example, as "trivial" as changing the frequency order of the answers), the classifier will make a substantial number of errors. This is particularly alarming unless we construct a system a prior to map one set of answers to another set, we are likely to have very poor transfer across datasets and would have to train a new Visual QA model whenever we encounter a new dataset. In fact, for two popular Visual QA datasets, about 10% answers are shared and of top-K answers (where $K < 10,000$), only 50% answers are shared. We refer readers to section 4.5 and Table 6 for more results.

In this paper, we propose a new learning model to address these challenges. Our main idea is to learn also an embedding of the answers. Together with the (joint embedding) features of image and question in some spaces, the answer embeddings parameterize a probabilistic model describing how the answers are similar to the image and question pair. We learn the embeddings for the answers as well as the images and the questions to maximize the correct answers' likelihood. The learned model thus aligns the semantic similarity of answers with the visual/semantic similarity of the image and question pair. Furthermore, the learned model can also embed any unseen answers, thus can generalize from one dataset to another one. Fig. 1 illustrates the main idea of our approach.

Our method needs to learn embeddings of hundreds and thousands of answers. Thus to optimize our probabilistic model, we overcome the challenge by introducing a computationally efficient way of adaptively sampling negative examples in a minibatch.

Our model also has the computational advantage that for each pair of image and question, we only need to compute the joint embedding of image and question for once, irrespective of how many candidate answers one has to examine. On the other end, models such as [13, 7] learn a joint embedding of the triplet (image, question and answer) needs to compute embeddings at the linear order of the number of candidate answers. When the number of candidate answers need to be large (to obtain better coverage), such models do not scale up easily.

While our approach is motivated by addressing challenges in open-end Visual QA, the proposed approach trivially includes multiple-choice based Visual QA as a special case and is thus equally applicable. We extensively evaluated our approach on several existing datasets, including Visual7W [32], VQA2 [9], and Visual Genome [16]. We show the gain in performance by our approach over the existing approaches that are based on multi-way classification. We also show the effectiveness of our approach in transferring models trained on one dataset to another. To our best knowledge, we are likely the first to examine the challenging issue of transferability in the open-end Visual QA task[1].

The rest of the paper is organized as follows. Section 3.1 introduces the notation and problem setup. Section 3.2 presents our proposed methods. Section 4 shows our empirical results on multiple Visual QA datasets.

2. Related Work

2.1. Visual QA

In open-end Visual QA, one popular framework of algorithms is to learn a joint image-question embedding and perform multi-way classification (for predicting top-frequency answers) on top [29, 2, 4, 7, 28, 18]. Though such methods naturally limited themselves to answer questions within a fixed (usually small) vocabulary, this framework has been shown to outperform other methods that dedicate for free-form answer generation [25, 14]. Different from this line of research, in the multiple-choice setting, algorithms are usually designed to learn a scoring function with the image, question, and answer triplets [13, 7, 24]. Such methods can take the advantage of answer semantics but fail to scale up inferencing along the increasing number of answer candidates. Comparing to all previous approaches, our proposed framework leverages the advantages of both worlds, capable of taking the answer semantic into account while remaining efficient. Please refer to section 3.6 for detailed discussion.

2.2. Learning Aligned Embeddings

The idea of learning and aligning embeddings has been explored in visual recognition [6, 20], in which the image and label embeddings are learned. Our work extends it to Visual QA[2] for parameterizing and learning a novel proba-

[1] Our work focuses on the transferability across datasets with different question and answer spaces. We leave visual transferability (e.g., by domain adaptation) as future work.

[2] We replace the image embedding with a joint one for image-question pairs, and investigate more complicated models for the answer embedding.

bilistic model. We further propose an efficient optimization technique to handle a large number of candidate answers (e.g., more than 201,000 in Visual Genome [16]), a situation rarely encountered in visual recognition.

3. Methods

In what follows, we describe our approach in detail. We start by describing a general setup for Visual QA and introducing necessary notations. We then introduce the main idea, followed by detailed descriptions of the method and important steps to scale the method to handling hundreds of thousands negative samples.

3.1. Setup and Notations

In the Visual QA task, the machine is given an image i and a question q, and is asked to generate an answer a. In this work, we focus on the open-ended setting where a is a member of a set $\mathcal{A}$. This set of candidate answers is intuitively "the universe of all possible answers". However, in practice, it is approximated by the top K most frequent correct answers in a training set [18, 7, 28], plus all the incorrect answers in the dataset (if any). Another popular setting is multiple-choice based. For each pair of (i, q), the set $\mathcal{A}$ is different (this set is either automatically generated [5] or manually generated [32, 1]). Without loss of generality, however, we use $\mathcal{A}$ to represent both. Whenever necessary, we clarify the special handling we would need for (i, q) specific candidate set.

We distinguish two subsets in $\mathcal{A}$ with respect to a pair (i, q): $\mathcal{C}$ and $\mathcal{D} = \mathcal{A} - \mathcal{C}$. The set $\mathcal{C}$ contains all the correct answers for (i, q)—it could be a singleton or in some cases, contains multiple *semantically similar* answers to the correct answer (e.g., "policeman" to "police officer"), depending on the datasets. The set $\mathcal{D}$ contains all the incorrect (or undesired) answers.

A training dataset is thus denoted by a set of N distinctive triplets $D = \{(i_n, q_n, \mathcal{C}_n)\}$ when only the correct answers are given, or $D = \{(i_n, q_n, \mathcal{A}_n = \mathcal{C}_n \cup \mathcal{D}_n)\}$ when both the correct and incorrect answers are given.

Note that by i, q or a, we refers to their "raw" formats (an image in pixel values, and a question or an answer in its textual forms).

3.2. Main Idea

Our main idea is motivated by two deficiencies in the current approaches for open-ended Visual QA [1]. In those methods, it is common to construct a K-way classifier so that for each (i, q), the classifier outputs k that corresponds to the correct answer (*i.e.*, the k-th element in $\mathcal{A}$ is the correct answer).

However, this classification paradigm cannot capture all the information encoded in the dataset for us to derive better models. First, by equating two different answers a_k and a_l with the ordinal numbers k and l, we lose the semantic kinship between the two. If there are two triplets $(i_m, q_m, a_k \in \mathcal{C}_m)$ and $(i_n, q_n, a_l \in \mathcal{C}_n)$ having similar visual appearance between i_m and i_n and similar semantic meaning between q_m and q_n, we would expect a_k and a_l to have some degrees of semantic similarity. In a classification framework, such expectation cannot be fulfilled as the assignment of ordinal numbers k or l to either a_k or a_l can be arbitrary such that the difference between k and l does not preserve the similarity between a_k and a_l. However, observing such similarity at both the inputs to the classifier and the outputs of the classifier is beneficial and adds robustness to learning.

The second flaw with the multi-way classification framework is that it does not lend itself to generalize across two datasets with little or no overlapping in the candidate answer sets $\mathcal{A}$. Unless there is a prior defined mapping between the two sets, the classifier trained on one dataset is not applicable to the other dataset.

We propose a new approach to overcome those deficiencies. The key idea is to learn embeddings of all the data. The embedding functions, when properly parameterized and learned, will preserve similarity and will generalize to unseen answers (in the training data).

Embeddings We first define a joint embedding function $f_{\boldsymbol{\theta}}(i, q)$ to generate the joint embedding of the pair i and q. We also define an embedding function $g_{\boldsymbol{\phi}}(a)$ to generate the embedding of an answer a. We will postpone to later to explain why we do not learn a function that generates the joint embedding of the triplet.

The embedding functions are parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. In this work, we use deep learning models such as multi-layer perceptron (MLP) and Stacked Attention Network (SAN) [28, 15] (after removing the classifier at the last layer). In principle, any representation network can be used—our focus is on how to use the embeddings.

Probabilistic Model of Compatibility (PMC) Given a triplet $(i_n, q_n, a \in \mathcal{C}_n)$ where a is a correct answer, we define the following probabilistic model

$$p(a|i_n, q_n) = \frac{\exp(f_{\boldsymbol{\theta}}(i_n, q_n)^\top g_{\boldsymbol{\phi}}(a))}{\sum_{a' \in \mathcal{A}} \exp(f_{\boldsymbol{\theta}}(i_n, q_n)^\top g_{\boldsymbol{\phi}}(a'))} \quad (1)$$

Discriminative Learning with Weighted Likelihood Given the probabilistic model, it is natural to learn the parameters to maximize its likelihood. In our work, we have found the following *weighted* likelihood is more effective

$$\ell = -\sum_{n}^{N} \sum_{a \in \mathcal{C}_n} \sum_{d \in \mathcal{A}} \alpha(a, d) \log P(d|i_n, q_n), \quad (2)$$

where the weighting function $\alpha(a, d)$ measures how much the answer d could contribute to the objective function. A nature design is

$$\alpha(a, d) = \mathbb{I}[a = d], \tag{3}$$

where $\mathbb{I}[\cdot]$ is the binary indicator function, taking value of 1 if the condition is true and 0 if false. In this case, the objective function reduces to the standard cross-entropy loss if $\mathcal{C}_n$ is a singleton. However, in section 3.4, we discuss several different designs.

3.3. Large-scale Stochastic Optimization

The optimization of eq. (2) is very challenging on real Visual QA datasets. There, the size of $\mathcal{A}$ can be as large as hundreds of thousands[3]. Thus computing the normalization term of the probability model is a daunting task.

We use a minibatch based stochastic gradient descent procedure to optimize the weighted likelihood. Specifically, we choose B triplets randomly from D (the training dataset defined in section 3.1) and compute the gradient of the weighted likelihood.

Within a minibatch $(i_b, q_b, \mathcal{C}_b)$ or $(i_b, q_b, \mathcal{C}_b \cup \mathcal{D}_b)$ for $b = 1, 2, \cdots B$, we construct a minibatched-universe

$$\mathcal{A}_B = \bigcup_{b=1}^{N} (\mathcal{C}_b \bigcup \mathcal{D}_b) \tag{4}$$

Namely, all the possible answers in the minibatch are used.

However, this "mini-universe" might not be a representative sampling of the true "universe" $\mathcal{A}$. Thus, we augment it with *negative sampling*. First we compute the set

$$\bar{\mathcal{A}}_B = \mathcal{A} - \mathcal{A}_B \tag{5}$$

and sample M samples from this set. These samples (denoted as $\mathcal{A}_o$) are mixed with $\mathcal{A}_B$ to increase the exposure to incorrect answers (*i.e.* negative samples) encountered by the triplets in a minibatch. In short, we use $\mathcal{A}_0 \bigcup \mathcal{A}_B$ in lieu of $\mathcal{A}$ in computing the posterior probability $p(a|i, q)$ and the likelihood.

3.4. Defining the Weighting Function α

We can take advantage of the weighting function $\alpha(a, d)$ to incorporate external or prior semantic knowledge. For example, $\alpha(a, d)$ can depend on semantic similiarity scores between a and d. Using the WUPS score [26, 19], we define the following rule

$$\alpha(a, d) = \begin{cases} 1 & \text{if } \textbf{WUPS}(a, d) > \lambda, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where λ is a threshold (e.g., 0.9 as in [19]). $\alpha(a, d)$ can also be used to scale triplets with a lot of semantic similar answers in $\mathcal{C}$ (for instance, "apple", "green apple", "small apple" or "big apple" are good answers to "what is on the table?"):

$$\alpha(a, d) = \frac{\mathbb{I}[a = d]}{|\mathcal{C}|} \tag{7}$$

such that each of these similar answers only contributes to a fraction of the likelihood to the objective function. The idea of eq. (7) has been exploited in several recent work [30, 12, 15] to boost the performance on VQA [3] and VQA2 [9].

3.5. Prediction

During testing, given the learned $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\phi}}$, for the **open-ended setting** we can apply the following decision rule

$$a^* = \arg\max_{a \in \mathcal{A}} f_{\boldsymbol{\theta}}(i, q)^\top g_{\boldsymbol{\phi}}(a), \tag{8}$$

to identify the answer to the pair (i, q).

Note that we have the freedom to choose $\mathcal{A}$ again: it can be the same as the "universe of answers" constructed for the training (*i.e.*, the collection of most frequent answers), or a union with all the answers in the validation or testing set. The flexibility is afforded here by using the embedding function $g_{\boldsymbol{\phi}}$ to embed any texts. Note that in existing open-ended Visual QA, the set $\mathcal{A}$ is constrained to the most frequent answers, reflecting the limitation of using multi-way classification as a framework for Visual QA tasks.

This decision rule readily extends to the **multiple-choice setting**, where we just need to set $\mathcal{A}$ to include the correct answer and the incorrect answers in each testing triplet.

3.6. Comparison to Existing Algorithms

Most existing Visual QA algorithms (most working on the open-ended setting on VQA [3] and VQA2 [9]) train a multi-way classifier on top of the $f_{\boldsymbol{\theta}}$ embedding. The number of classes are set to 1,000 for VQA [7] and around 3,000 for VQA2 [7, 30, 15] of the top-frequency correct answers. These top-frequent answers cover over 90% of the training and 88% of the training and validation examples. Those training examples whose correct answers are not in the top-K frequent ones are simply disregarded during training.

There are some algorithms also learning a tri-variable compatability function $h(i, q, a)$ [13, 7, 24]. And the correct answer is inferred by identify a^* such that $h(i, q, a^*)$ is the highest. This type of learning is particularly suitable for multiple-choice based Visual QA. Since the number of candidate answers is small, enumerating all possible a is feasible. However, for open-ended Visual QA tasks, the number of possible answers is very large—computing the function $h()$ for every one of them is costly.

[3] In the Visual Genome dataset [16], for example, we have more than 201,000 possible answers.

Table 1. Summary statistics of Visual QA datasets.

Dataset	# of Images			# of $(i, q, \mathcal{C})$ triplets			$(\|\mathcal{C}\|, \|\mathcal{D}\|)$
Name	train	val	test	train	val	test	per tuple
VQA2 [9]	83K	41K	81K	443K	214K	447K	$(10, 0)$
Visual7W [32]	14K	5K	8K	69K	28K	42K	$(1, 3)$
V7W [5]	14K	5K	8K	69K	28K	42K	$(1, 6)$
qaVG [5]	49K	19K	29K	727K	283K	433K	$(1, 6)$

Note that our decision rule relies on computing $f_{\boldsymbol{\theta}}(i, q)^\top g_{\boldsymbol{\phi}}(a)$, a factorized form of the more generic function $h(i, q, a)$. However, precisely due to this factorization, we only need to compute $f_{\boldsymbol{\theta}}(i, q)$ just once for every pair (i, q). For $g_{\boldsymbol{\phi}}(a)$, as long as the model is sufficiently simple, enumerating over many possible a is less demanding than what a generic (and more complex) function $h(i, q, a)$ requires. Indeed, in practice we only need to compute $g_{\boldsymbol{\phi}}(a)$ once for any possible a[4]. See section 4.6 for details.

4. Experiments

We validate our approach on several Visual QA datasets. We start by describing these datasets and the empirical setups. We then report our results. The proposed approach performs very well. It outperforms the corresponding multi-way classification-based approaches where the answers are modeled as independent ordinal numbers. Moreover, it outperforms those approaches in transferring models learned on one dataset to another one.

4.1. Datasets

We apply the proposed approach to four datasets. Table 1 summarizes their characteristics.

VQA2 [9]. The dataset uses images from MSCOCO [17] with the same training/validation/testing splits and constructs triplets $(i_n, q_n, \mathcal{C}_n)$ of image (i_n), question (q_n), and correct answers $(\mathcal{C}_n)$ respectively. On average, 6 questions are generated for each image, and each (i_n, q_n) pair is answered by 10 human annotators (i.e, $|\mathcal{C}_n| = 10$). The most frequent one is selected as the single correct answer t_n.

Visual7W Telling (Visual7W) [32] and V7W [5]. Visual7W uses 47,300 images from MSCOCO [17] and contains 139,868 $(i_n, q_n, \mathcal{C}_n, \mathcal{D}_n)$ tuples. The set of correct answers $\mathcal{C}_n$ is a singleton, containing only one answer. Each has 3 incorrect answers generated by humans (i.e., $|\mathcal{D}_n| = 3$). Humans are encouraged to start questions with the 6W words; i.e., "who", "where", "how", "when", "why", and "what". V7W is a revised version of Visual7W, which has a more carefully designed set of incorrect answers to prevent machines from ignoring the image, or question or both to exploit the bias in the datasets [5]. In this dataset, each $(i_n, q_n, \mathcal{C}_n)$ triplet is associated with 6 auto-generated incorrect answers.

Table 2. The answer coverage of each dataset.

	# of unique answers	triplets covered by top $K =$		
Dataset	train/val/test/All	1,000	3,000	5,000
VQA2	22K/13K/ - /29K	88%	93%	96%
Visual7W	63K/31K/43K/108K	57%	68%	71%
VG	119K/57K/79K/201K	61%	72%	76%

Visual Genome (VG) [16] and qaVG [5]. qaVG [5] is a multiple-choice Visual QA dataset derived from VG [16]. VG contains 101,174 images from MSCOCO [17] and has 1,445,322 $(i_n, q_n, \mathcal{C}_n)$ triplets. The set of correct answers $\mathcal{C}_n$ is a singleton. On average an image is coupled with 14 question-answer pairs. qaVG augments each $(i_n, q_n, \mathcal{C}_n)$ triplet with 6 auto-generated incorrect answers. The dataset is divided into 50%, 20%, and 30% for training, validation, and testing—each portion is a "superset" of the corresponding one in Visual7W or V7W. We train our model on VG [16] and evaluate it on qaVG [5].

Answer Coverage within Each Dataset. In Table 2, We show the number of unique answers in each dataset on each split, together with the portions of question and answer pairs covered by the top-K frequent correct answers from the training set. We observe that the qaVG contains the largest number of answers, followed by Visual7W and VQA2. In terms of coverage, we see that the distribution of answers on VQA2 is the most skewed: over 88% of training and validation triplets are covered by the top-1000 frequent answers. On the other hand, Visual7W and qaVG needs more than top-5000 frequent answers to achieve a similar coverage.

Thus, a prior, Visual7W and qaVG are "harder" datasets, where a multi-way classification-based open-ended Visual QA model will not perform well unless the number of categories is significantly higher (say $\gg 5000$) in order to be able to encounter less frequent answers in the test portion of the dataset—the answers just have a long-tail distribution.

4.2. Experimental Setup

Our Model. We use two different models to parameterize the embedding function $f_{\boldsymbol{\theta}}(i, q)$ in our experiments—Multi-layer Perceptron [13, 5] (MLP) and Stacked Attention Network [28, 15] (SAN). For both models, we first represent each token in the question by the 300-dimensional GloVe vector [21], and use the ResNet-152 [11] to extract the visual features following the exact setting of [15]. Detailed specifications of each model are as follows.

- Multi-layer Perceptron (MLP): We represent an image by the 2,048-dimensional vector form the top layer of the

[4] The answer embedding $g(a)$ for all possible answers (say 100,000) can be pre-computed. At inference we only need to compute the embedding $f(i, q)$ once for an (i, q) pair and perform 100,000 inner products. In contrast, methods like [13, 7, 24] need to compute $h(i, q, a)$ for 100,000 times. Even if such a function is parameterized with a simple MLP, the computation is much more intensive than an inner product when one has to perform 100,000 times.

ResNet-152 pre-trained on ImageNet [23], and a question by the average of the GloVe vectors after a linear transformation followed by tanh non-linearity and dropout. We then concatenate the two features (in total 2,348 dimension), and feed them into a one-layer MLP (4,096 hidden nodes and intermediate dropout), with the output dimensionality of 1,024.

- Stacked Attention Network (SAN): We represent an image by the $14 \times 14 \times 2048$-dimensional tensor, extracted from the second last layer of the ResNet-152 pre-trained on ImageNet [23]. See [18] for details. On the other hand, we represent a question by a one layer bidirectional LSTM over GloVe word embeddings. Image and question features are then inputed into the SAN structure for fusion. Specifically, we follow a very similar network architecture presented in [15], with the output dimensionality of 1,024.

For parameterizing the answering embedding function $g_{\boldsymbol{\phi}}(a)$, we adopt two architectures: **1)** Utilizing a one-layer MLP on *average* GloVe embeddings of answer sequences, with the output dimensionality of 1,024. **2)** Utilizing a two-layer bidirectional LSTM (bi-LSTM) on top of GloVE embeddings of answer sequences. We use MLP for computing answer embedding by default. We denote method with bi-LSTM answer embedding with a postfix $\star$ (*e.g.* **SAN$\star$**). Please refer to our Suppl. Material for more details about architectures and optimization.

In the following, we denote our factorized model applying PMC for optimization as fPMC (cf. eq (1)). We consider variants of fPMC with different architectures (*e.g.* MLP, SAN) for computing $f_{\boldsymbol{\theta}}(i, q)$ and $g_{\boldsymbol{\phi}}(a)$, named as fPMC(MLP), fPMC(SAN) and fPMC(SAN$\star$).

Competing Methods. We compare our model to multi-way classification-based (CLS) models which take either MLP or SAN as $f_{\boldsymbol{\theta}}$. We denote them as CLS(MLP) or CLS(SAN). We set the number of output classes for CLS model to be top-3,000 frequent training answers for VQA2, and top-5,000 for Visual7W and VG. This is a common setup for open-ended Visual QA [1].

Meanwhile, we also re-implement approaches that learn a scoring function $h(i, q, a)$ with its input as $(i_n, q_n, \mathcal{C}_n)$ triplets [13, 5]. As such methods are initially designed for multiple-choice datasets, the calibration between positive and negative samples needs to be carefully tuned. It is challenging to adapt to 'open-end' settings where the number of negative answers scaled up [5]. Therefore, we adapt them to also utilize our PMC framework for training, which optimize stochastic multi-class cross-entropy with negative answers sampling. We name such methods as uPMC (unfactorized PMC) and call its variants as uPMC(MLP) and uPMC(SAN). We also compare to reported results from other state-of-the-art methods.

Table 3. Results (%) on Visual QA with different settings: open-ended (Top-K) and multiple-choice (MC) based for different datasets. The omitted ones are due to their missing in the corresponding work.

Method	Visual7W MC [32]	V7W MC [5]	VQA2 Top-3k [9]	qaVG MC [5]
LSTM [32]	55.6	-	-	-
MLP [5]	65.7	52.0	-	58.5
MLP [13]	67.1	-	-	-
C+LSTM [9]	-	-	54.1	-
MCB [9]	62.2	-	62.3	-
MFB [31]	-	-	65.0	-
BUTD [2]	-	-	65.6	-
MFH [30]	-	-	66.8	-
Multi-way Classification Based Model (CLS)				
CLS(MLP)	51.6	40.9	53.5	46.9
CLS(SAN)	53.7	43.6	62.4	53.0
Our Probabilistic Model of Compatibility (PMC)				
uPMC(MLP)	62.4	51.6	51.4	54.5
uPMC(SAN)	65.3	55.2	56.0	61.3
fPMC(MLP)	63.1	52.4	59.3	57.7
fPMC(SAN)	65.6	55.4	63.2	62.6
fPMC(SAN$\star$)	66.0	55.5	63.9	63.4

Evaluation Metrics The evaluation metric for each dataset is different. For VQA2, the standard metric is to compare the selected answer a^* of a (i, q) pair to the ten corresponding human annotated answers $\mathcal{C} = \{s_1, \cdots, s_{10}\}$. The performance on such an (i, q) pair is set as follows

$$\text{acc}(a^*, \mathcal{C}) = \max\left\{1, \frac{\sum_l \mathbb{I}[a^* = s_l]}{3}\right\}. \tag{9}$$

We report the average performance over examples in the validation split and test split.

For Visaul7W (or V7W), the performance is measured by the portion of correct answers selected by the Visual QA model from the candidate answer set. The chance for random guess is 25% (or 14.3%). For VG, we focus on the multiple choice evaluation (on qaVG). We follow the settings proposed by [5] and measure multiple choice accuracy. The chance for random guess is 14.3%.

4.3. Results on Individual Visual QA Datasets

Table 3 gives a comprehensive evaluation for most state-of-the-art approaches on four different settings over VQA2(test-dev), Visual7W, V7W and qaVG[6]. Among all those settings, our proposed fPMC model outperform the

[5] See the Suppl. Material for details

[6] The omitted ones are due to their missing in the corresponding work. In fact, most existing work only focuses on one or two datasets.

Table 4. The effect of negative sampling ($M = 3,000$) on fPMC. The number is the accuracy in each question type on VQA2 (val).

Method	Mini-Universe	Y/N	Number	Other	All
MLP	$\mathcal{A}_B$	70.1	33.0	38.7	49.8
SAN		78.2	37.1	45.7	56.7
MLP	$\mathcal{A}_o \bigcup \mathcal{A}_B$	76.6	36.1	43.9	55.2
SAN		79.0	38.0	51.3	60.0

corresponding classification model by a noticeable margin. Meanwhile, fPMC outperforms uPMC over all settings. Comparing to other state-of-the-art methods, we show competitive performance against most of them.

In Table 3, note that there are differences in the experimental setups in many of the comparison to state-of-the-art methods. For instance, MLP [13, 5] used either better text embedding or more advanced visual feature, which benefits their result on Visual7W significantly. Under the same configuration, our model has obtained improvement. Besides, most of the state-of-the-art methods on VQA2 fall into the category of classification model that accommodates specific Visual QA settings. They usually explore better architectures for extracting rich visual information [32, 2], or better fusion mechanisms across multiple modalities [9, 31, 30]. We notice that our proposed PMC model is orthogonal to all those recent advances in multi-modal fusion and neural architectures. More advanced deep learning models can be adapted into our framework as $f_{\boldsymbol{\theta}}(i, q)$ (*e.g.* fPMC(MFH)) to achieve superior performance across different settings. This is particularly exemplified by the dominance of SAN over the vanilla MLP model. We leave this for future work.

4.4. Ablation Studies

Importance of Negative Sampling Our approach is probabilistic, demanding to compute a proper probability over the space of all possible answers. (In contrast, classification-based models limit their output spaces to a pre-determined number, at the risk of not being able to handle unseen answers).

In section 3.3, we describe a large-scale optimization technique that allows us to approximate the likelihood by performing negative sampling. Within each mini-batch, we create a mini-universe of all possible answers as the union of all the correct answers (*i.e.*, $\mathcal{A}_B$). Additionally, we randomly sample M answers from the union of all answers outside of the mini-batch, creating "an other world" of all possible answers $\mathcal{A}_o$. The $\mathcal{A}_o$ provides richer negative samples to $\mathcal{A}_B$ and is important to the performance of our model, as shown in Table 4.

We further conducted detailed analysis on the effects of negative sample sizes as shown in Fig. 2. With the number of negative samples increasing from 0 to 3,000 for each mini-batch, we observe a increasing trend from the validation accuracy. A significant performance boost is obtained comparing methods with a small number of negative samples to no additional negative samples. The gain then becomes marginal after $\mathcal{A}_o$ is greater than 2,000.

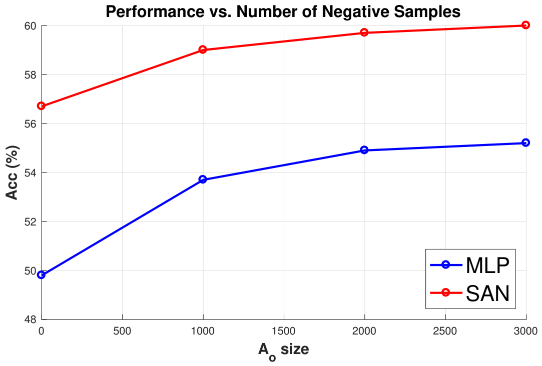

Figure 2. Detailed analysis on the size of negative sampling to fPMC(MLP) and fPMC(SAN) at each mini-batch. The reported number is the accuracy on VQA2 (val).

The Effect of Incorporating Semantic Knowledge in Weighted Likelihood In section 3.2, we have introduced the weighting function $\alpha(a, d)$ to measure how much an incorrect answer d should contribute to the overall objective function. In particular, this weighting function can be used to incorporate prior semantic knowledge about the relationship between a correct answer a and an incorrect answer d. We report the details in the Suppl. Material.

4.5. Transfer Learning Across Datasets

One important advantage of our method is to be able to cope with unseen answers in the training dataset. This is in stark contrast to multi-way classification based models which will have to skip on those answers as the output categories are selected as top-K most frequent answers from the training dataset.

Thus, classification based models for Visual QA are not amenable to transfer across datasets where there is a large gap between different spaces of answers. Table 6 illustrates the severity by computing the number of common answers across datasets. On average, about 7% to 10% of the unique answers are shared across datasets. If we restrict the number of answers to consider to top 1,000, about 50% to 65% answers are shared. However, top 1000 most frequent answers are in general not enough to cover all the questions in any dataset. Hence, we arrive at the unexciting observation—we can transfer but we can only answer a few questions!

In Table 5, we report our results of transferring learned Visual QA model from one dataset (row) to another one (column). For VQA2, we evaluate the open-end accuracy using top-3000 frequent answer candidates on validation set. We evaluate multiple-choice accuracy on the test set of Visual7W and qaVG.

Table 5. Results of cross-dataset transfer using either classification-based models or our models (PMC) for Visual QA. (f_θ = SAN)

	Visual7W				VQA2				qaVG			
	CLS	uPMC	fPMC	fPMC⋆	CLS	uPMC	fPMC	fPMC⋆	CLS	fPMC	fPMC	fPMC⋆
Visual7W	53.7	65.3	65.6	**66.0** ↑	19.1	18.5	**19.8** ↑	19.1	42.8	52.2	**54.8** ↑	54.3
VQA2	45.8	56.8	60.2	**61.7** ↑	59.4	56.0	60.0	**60.9** ↑	37.6	51.5	54.8	**56.8** ↑
qaVG	58.9	66.0	68.4	**69.5** ↑	25.6	23.6	25.8	**26.4** ↑	53.0	61.2	62.6	**63.4** ↑

Table 6. The # of common answers across datasets (training set)

	Top-K most frequent answers					Total # of
Dataset	1K	3K	5K	10K	all	unique answers
VQA2, Visual7W	451	1,262	2,015	3,585	10K	137K
VQA2, qaVG	495	1,328	2,057	3,643	11K	149K
Visual7W, qaVG	657	1,890	3,070	5,683	27K	201K

The classification models (CLS) clearly fall behind the performance of our method (uPMC and fPMC)—the red upper arrows signify improvement. In some pairs the improvement is significant (e.g., from 42.8% to 54.8% when transferring from Visual7W to qaVG). Furthermore, we noticed that fPMC outperforms uPMC in all transfer settings.

However, VQA2 seems a particular difficult dataset to be transferred to, from either V7W or qaVG. The improvement from CLS to fPMC is generally small. This is because VQA2 contains a large number of Yes/No answers. For such answers, learning embeddings is not advantageous as there are little semantic meanings to extract from them.

Table 7. Transferring is improved on the VQA2 dataset without Yes/No answers (and the corresponding questions) (f_θ = SAN).

Dataset	CLS	uPMC	fPMC	fPMC⋆
Visual7W	31.7	29.5	33.1 ↑	32.0
qaVG	42.6	39.3	43.0	43.4 ↑

We perform another study by removing those answers (and associated questions) from VQA2 and report the transfer learning results in Table 7. In general, both CLS and fPMC transfer better. Moreover, fPMC improves over CLS by a larger margin than that in Table 5.

To gain a deeper understanding towards which component brings the advantage in transfer learning, we performed additional experiments to analyze the difference on seen/unseen answers. At the same time, we include a t-SNE visualization to access the quality of our answer embeddings. We conclude that learned answer embeddings can better capture semantic and syntactic similarities among answers. See the Suppl. Material for details on both analysis.

4.6. Inference Efficiency

Next we study the inference efficiency of the proposed fPMC, uPMC (i.e., triplet based approaches [13, 7, 24] with PMC) models with the CLS model. For fair comparison, we use the one-hidden-layer MLP model for all approaches, keep $|\mathcal{C}| = 1000$ and mini-batch size to be 128 (uPMC based approach is memory consuming. More candidates require reducing the mini-batch size). We evaluate models on the VQA2 validation set (∼2200 mini-batches) and report the (average) mini-batch inference time. Fig. 3 and Table 8 show that fPMC(MLP) obtains similar performance to CLS(MLP), with at least 10 times faster than uPMC(MLP).

Table 8. Efficiency study among CLS(MLP), uPMC(MLP) and fPMC(MLP). The reported numbers are the average inference time of a mini-batch of 128 ($|\mathcal{C}|$ = 1000).

Method	CLS(MLP)	uPMC(MLP)	fPMC(MLP)
Time (ms)	22.01	367.62	22.14

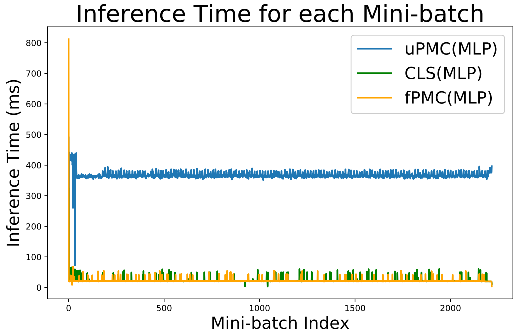

Figure 3. **Inference time Vs. Mini-batch index.** fPMC(MLP) and CLS(MLP) model are 10x faster than uPMC(MLP) (use PyTorch v0.2.0 + Titan XP + Cuda 8 + Cudnnv5).

5. Discussion

We propose a novel approach of learning answer embeddings for the visual question answering (Visual QA) task. The main idea is to learn embedding functions to capture the semantic relationship among answers, instead of treating them as independent categories as in multi-way classification-based models. Besides improving Visual QA results on single datasets, another significant advantage of our approach is to enable better model transfer. The empirical studies on several datasets have validated our approach.

Our approach is also "modular" in the sense that it can exploit any joint modeling of images and texts (in this case, the questions). An important future direction is to discover stronger multi-modal modeling for this purpose.

Acknowledgment This work is partially supported by USC Graduate Fellowship, NSF IIS-1065243, 1451412, 1513966/1632803, 1208500, CCF-1139148, a Google Research Award, an Alfred. P. Sloan Research Fellowship and ARO# W911NF-12-1-0241 and W911NF-15-1-0484.

References

[1] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. Lawrence Zitnick, D. Parikh, and D. Batra. Vqa: Visual question answering. *IJCV*, 2016. 1, 3, 6

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and vqa. In *CVPR*, 2018. 2, 6, 7

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 1, 4

[4] H. Ben-younes, R. Cadene, M. Cord, and N. Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, 2017. 2

[5] W.-L. Chao, H. Hu, and F. Sha. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *NAACL*, 2018. 1, 3, 5, 6, 7

[6] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 2

[7] 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. In *EMNLP*, 2016. 1, 2, 3, 4, 5, 8

[8] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question. In *NIPS*, pages 2296–2304, 2015. 1

[9] 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. In *CVPR*, 2017. 1, 2, 4, 5, 6, 7

[10] A. K. Gupta. Survey of visual question answering: Datasets and techniques. *arXiv preprint arXiv:1705.03865*, 2017. 1

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[12] I. Ilievski and J. Feng. A simple loss function for improving the convergence and accuracy of visual question answering models. In *CVPR Workshop*, 2017. 4

[13] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. In *ECCV*, 2016. 1, 2, 4, 5, 6, 7, 8

[14] K. Kafle and C. Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20, 2017. 1, 2

[15] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017. 1, 3, 4, 5, 6

[16] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 2, 3, 4, 5

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[18] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, pages 289–297, 2016. 1, 2, 3, 6

[19] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014. 4

[20] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014. 2

[21] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 5

[22] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015. 1

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6

[24] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016. 2, 4, 5, 8

[25] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. v. d. Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017. 1, 2

[26] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL*, 1994. 4

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

[28] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 2, 3, 5

[29] 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. 2

[30] Y. Zhou, Y. Jun, X. Chenchao, F. Jianping, and T. Dacheng. Beyond bilinear: Generalized multi-modal factorized high-order pooling for visual question answering. *arXiv preprint arXiv:1708.03619*, 2017. 4, 6, 7

[31] Y. Zhou, Y. Jun, F. Jianping, and T. Dacheng. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, 2017. 6, 7

[32] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 1, 2, 3, 5, 6, 7