

# Good, Better, Best: Textual Distractors Generation for Multiple-Choice Visual Question Answering via Reinforcement Learning

Jiaying Lu<sup>♠\*</sup>, Xin Ye<sup>♥</sup>, Yi Ren<sup>♠</sup>, Yezhou Yang<sup>♥</sup>

<sup>♠</sup>Department of Computer Science, Emory University

<sup>♥</sup>School of Computing and Augmented Intelligence, Arizona State University

<sup>♠</sup>School for the Engineering of Matter, Transport and Energy, Arizona State University

jiaying.lu@emory.edu, {xinye1, yiren, yz.yang}@asu.edu

## Abstract

Multiple-choice VQA has drawn increasing attention from researchers and end-users recently. As the demand for automatically constructing large-scale multiple-choice VQA data grows, we introduce a novel task called textual Distractors Generation for VQA (DG-VQA) focusing on generating challenging yet meaningful distractors given the context image, question, and correct answer. The DG-VQA task aims at generating distractors without ground-truth training samples since such resources are rarely available. To tackle the DG-VQA unsupervisedly, we propose GOBBET, a reinforcement learning (RL) based framework that utilizes pre-trained VQA models as an alternative knowledge base to guide the distractor generation process. In GOBBET, a pre-trained VQA model serves as the environment in RL setting to provide feedback for the input multi-modal query, while a neural distractor generator serves as the agent to take actions accordingly. We propose to use existing VQA models' performance degradation as indicators of the quality of generated distractors. On the other hand, we show the utility of generated distractors through data augmentation experiments, since robustness is more and more important when AI models apply to unpredictable open-domain scenarios or security-sensitive applications. We further conduct a manual case study on the factors why distractors generated by GOBBET can fool existing models.

## 1. Introduction

Visual Question Answering (VQA) [1, 30, 47, 49] is an emerging research problem that requires algorithms to answer arbitrary natural language questions about a given image. Recently, VQA has attracted a large number of interests across computer vision, natural language process-

\*Work was done as a visiting researcher at Arizona State University.

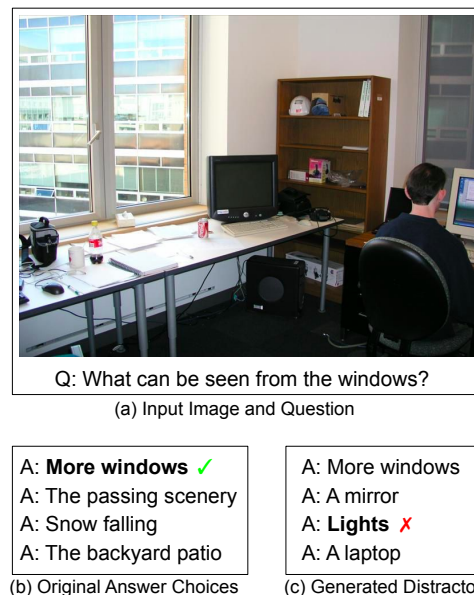


Figure 1. An example of DG-VQA task. The well-trained VQA model predicts the right answer choice for the input image and question (1a and 1b). However, it is easy to distinguish correct choice from distractor choices. The model will be fooled when encountering generated distractors (1c).

ing, and knowledge representation and reasoning communities, since these questions require AI models' capability of understanding vision, language, and even external knowledge [30, 47] to answer. In general, VQA can be divided into two specific sub-tasks according to the question forms: 1) open-ended VQA [1, 30] requiring a free-form response; and 2) multiple-choice (MC) VQA [47, 49] requiring a single answer picking from a list of given candidates. In this paper, we are particularly interested in the MC VQA task.

With the advancement of deep learning, neural network models have achieved remarkable progress to bridge the gap between human performance and state-of-the-art neu-

ral network models on the MC VQA task. However, it has been pointed out that the distractors (the wrong candidate choices) are too simple or biased [19] in several benchmark datasets, which raises doubt about the proposed models’ actual discriminative ability. For instance, in the toy example shown in Figure 1, the correct answer is “More windows”, while the distractor “The passing scenery” is not challenging because the scenery outside is stationary. Similarly, “Snow falling” is not challenging due to the sunny weather.

To facilitate multiple-choice VQA better to serve as a robust “multi-modality Turing test” [9,43] and to explore factors causing failure of existing VQA models, we introduce a novel task as generating challenging distractors, dubbed as DG-VQA: *textual Distractor Generation for VQA*. The task can be defined as follows: Given an image with a corresponding natural language question and a correct answer, generating distractors that lead to trained VQA models failing at picking the right choice from the candidate list, where the candidate list is composed of the correct answer and generated distractors. Figure 1c provides a toy example of generated distractors, where “A mirror” and “Lights” are very confusing choices for both AI models and humans. Producing such distractors provides a tool for researchers to figure out whether well-trained VQA models are vulnerable to potential attacks and determine whether they are ready for real-world deployment. Moreover, MC question answering is widely used in the education area, and manual distractor generation is hard and time-consuming. There are some previous works [8,25] focusing on automatic distractor generation (DG) to alleviate instructors’ workload. Unfortunately, none of them consider that applying multimodal materials in education becomes increasingly favorite.

One of the major technical challenges for the proposed DG-VQA task is that the training data is very limited or not available in real-world scenarios. Previous unsupervised distractor generation methods typically rely on the similarity measurements between the textual answer and generated candidates, thus ignoring the critical signals from the image input. Owing to the recent progress of pre-trained deep neural networks on large datasets, the pre-trained VQA models are capable of generating high-quality answers according to the given image and question. Whilst learning the correct answers, these models may also be storing plausible responses to the multimodal context, and may be able to generate distractors for the input. Therefore, we propose to utilize these existing VQA models as an alternative knowledge source to guide a distractor generator, which helps train the distractor generator without training samples. More specifically, we fix the pre-trained VQA models and use them to produce numerical quality judgment for the generated distractors based on input context (*i.e.* the judgment score represents which distractor is better). To propagate the non-differentiable judgment scores to the distractor

generator, we opt for the reinforcement learning (RL) techniques [24,37]. We dub the proposed framework as GOBBET (“GOod Better BEsT” for DG-VQA). In GOBBET, the distractor generator, which is regarded as an agent, receives rewards from the pre-training VQA model, which serves as the environment, based on the input context and generated distractors, which are actions taken by the agent. Therefore, the distractor generator is trained to maximize the cumulative reward from the pre-training VQA model. The choice of reward function is flexible as long as it represents the quality of generated distractors. In practice, we define the negative judgment score as the reward, and we utilize the policy gradient algorithm to optimize the generator.

In this work, an extensive suite of experiments has been conducted on the public MC VQA benchmark Visual7W [49]. Since the goal is to generate challenging distractors that lure existing models to fail, we propose to adopt performance degradation as the main measurement for generated distractors. Through experiments results on different existing VQA models, we validate distractors predicted by GOBBET outperform all baseline methods. In addition, we further demonstrate the utility of generated distractors by feeding them as augmented data into VQA models. We have observed the performance boosts on models trained with augmented data, which support the effectiveness of GOBBET from another perspective. Finally, we conduct case studies on distractors created by baselines and GOBBET, to gain a more intuitive sense of why GOBBET can generate more challenging distractors than other methods.

## 2. Related Work

**Visual Question Answering.** The open-ended answering task [20,39] and the multiple-choice task [47,49] are two typical tasks for VQA [1]. In this work, we focus on the multiple-choice task. Existing VQA models commonly combine an image encoder and a textual encoder to represent input pictures and input questions. The multimodal context embedding is fused and then fed into an answer decoder to generate the answers. Traditionally, convolutional neural networks [14] and recurrent neural networks [17] are popular choices for image encoders and textual encoders, while the answer decoder ranges from a softmax classifier [7], an RNN decoder [29] to a dot product layer [19]. More recently, Transformer-based networks [23] have shown distinguish performance as a uniform layer of both multi-modality encoders and decoders.

**Distractor Generation.** Automatic distractors generation (DG) from text is explored in-depth in the Natural Language Processing domain. At the same time, there are only a few studies in the multi-modal domain. Most prior approaches to textual DG are based on unsupervised similarity measures. These include n-gram co-occurrence likelihood [15], word/sentence embedding-based semantic simi-

larities [21], syntactic homogeneity [3] and ontology-based similarity [41]. Besides, other works utilize supervised learning algorithms for DG. Sakaguchi *et al.* [38] train a discriminative model to predict distractors, Liang *et al.* [25] apply learning to rank algorithm, and Gao *et al.* [8] use an end-to-end framework to produce distractors generatively. Although being successful, multimodality knowledge is still required to produce high-quality distractors.

**Pre-trained Models as Knowledge Bases.** Knowledge bases have shown great potential in multi-modal information retrieval setting [28, 30, 50]. Unfortunately, the construction of a large-scale multi-modal knowledge base (KB) is time-consuming, and the coverage of KB is limited. Hence, in practice, we often need to populate these KBs from raw text or other modalities, where ad-hoc complex pipelines are required and noise can easily accumulate. Recently, researchers start to explore alternative lightweight KBs. Gokhale *et al.* [10] incorporate the semantics-inverting and semantics-preserving transformations over input textual query for the robust vision-and-language model optimization, instead of explicit knowledge of the text. Petroni *et al.* [34] proposes to utilize language models pre-trained on large textual corpora as KBs storing relational linguistic knowledge, and experiments on multiple downstream tasks such as question answering and relation prediction well support their arguments. Furthermore, Wang *et al.* [44] explore constructing open knowledge base from pre-trained language models without human supervision. Our GOBBET share a similar idea to these works to use pre-trained VQA models as alternative KBs to retrieve distractors based on input multi-modal query.

**Reinforcement Learning.** Reinforcement learning (RL) [42] has been adopted in a variety of vision and language tasks, such as image captioning [37], text to image synthesis [36], VQA [6, 27] and visual dialogue [48]. Liu *et al.* [27] propose a RL-based strategy to generate visual questions. Fan *et al.* [6] enhance content and linguistic attributes of produced questions by introducing two discriminators in an RL framework. In GOBBET, we utilize the REINFORCE algorithm [45] to propagate the feedback backward from the pre-trained VQA models to the distractor generator.

### 3. Problem Definition

Textual Distractor Generation for multiple-choice VQA (DG-VQA) aims at generating challenging distractors (wrong options)  $\mathbf{D} = \{d_1, d_2, \dots, d_k\}$  based on the input image  $\mathbf{i}$ , question  $\mathbf{q}$  and answer  $\mathbf{a}$ . Both  $\mathbf{q}$ ,  $\mathbf{a}$  and  $\mathbf{d}$  are all textual sequence consist of words  $w_{1:T} = (w_1, w_2, \dots, w_T)$ . Depending on the dataset, sometimes not only the correct answer  $\mathbf{a}^{\text{COR}}$  is provided, but also several wrong answers  $\{\mathbf{a}_1^{\text{WOR}}, \dots, \mathbf{a}_m^{\text{WOR}}\}$ . In this work, we focus on the most general case that only one correct answer  $\mathbf{a}$  is provided. Fig-

ure 1 displays a toy example of proposed DG-VQA task. The generated distractors are expected to be challenging, since such distractors can better serve as an effective assessment for humans and AIs [13, 19]. However, challenging does not mean the generated distractors  $\mathbf{D}$  must be semantically equivalent to the input correct answer  $\mathbf{a}$ . Therefore, we propose pre-trained models’ performance degradation and data augmentation improvement as two indicators to evaluate the generated distractors.

## 4. GOBBET: A Reinforcement Learning Framework for DG-VQA

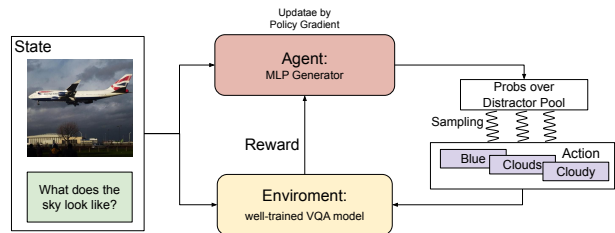


Figure 2. The proposed GOBBET Framework

To tackle the DG-VQA task without direct training samples, our key insight is to leverage the pre-trained VQA models as an alternative knowledge source. Therefore, we propose GOBBET (“GOod Better BEsT” for DG-VQA), a reinforcement learning-based framework where the agent model is trained to generate distractors based on the feedback from the environment. Figure 1 shows the overall architecture of the proposed GOBBET. We will introduce the technical details in the following subsections.

### 4.1. DG-VQA as A RL Problem

Inspired by recent progress in reinforcement learning (RL) and adversarial generation [33, 46], RL methods are promising for scarce supervision scenarios and efficient to address the inconsistency between the training objective and test metrics [6, 48]. Therefore, we adopt a policy gradient framework GOBBET to generate textual distractors for multiple-choice VQA. GOBBET has two major components: (1) the agent  $G_\theta$ , which is a distractor generator that generates high-quality distractors  $\mathbf{D}$  according to the input image  $\mathbf{i}$  and question  $\mathbf{q}$ ; (2) the environment  $J_\phi$  which is a pre-trained VQA model that produces rewards based on the generated distractors and the input context. GOBBET is somehow similar to the GAN framework [11] if we regard the agent as the generator and the environment as the discriminator, but we opt to fix the pre-trained VQA model to serve as the static external knowledge source during the GOBBET training process. The reason that we do not make the VQA models trainable is the concern of local convergence [31].

We first denote distractors generation as a sequence generation process. The distractor generator  $G_\theta$  is trained to produce a set of distractors  $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k\}$ , where each  $\mathbf{d}$  is a sequence of words  $\mathbf{d} = d_{1:T} = (d_1, d_2, \dots, d_t, \dots, d_T)$ . It is worth noting that, the bold math symbol  $\mathbf{d}_k$  denotes the  $k$ -th distractor in a set of distractors  $\mathbf{D}$ , while the regular math symbol  $d_t$  denotes the  $t$ -th word in a sequence of word. At each timestep  $t$ ,  $G_\theta$  is generate one word  $d_t$  given the input image  $\mathbf{i}$ , the question  $\mathbf{q}$ , and the generated distractor sequence until last timestep  $d_{1:t-1}$ :

$$d_t = G_\theta(\mathbf{i}, \mathbf{q}, d_{1:t-1}). \quad (1)$$

Under the RL setting, at timestep  $t$  the state  $s$  of the generator is the currently produced tokens  $d_{1:t-1}$  and the action  $a$  is the next token  $d_t$  to produce. So the state transition is deterministic once an action has been chosen. Following the notation in [42], the object of the  $G_\theta$  is to produce a sequence to minimize its negative expected reward:

$$L(\theta) = -\mathbb{E}_{d_{1:T} \sim G_\theta} [R(d_{1:T})], \quad (2)$$

where  $d_{1:T}$  is the a sampled generation from the model  $G_\theta$ .

Without the loss of generality, we adopt the REINFORCE algorithm [45] to optimize the agent  $G_\theta$  through the policy gradient, and we take judgement scores  $R(d_{1:T})$  (*i.e.* the likelihood of  $\mathbf{d}$  as the answer to the input context  $(\mathbf{i}, \mathbf{q})$ ) from the environment  $J_\phi$ . Formally, the optimization process is defined as follows:

$$\begin{aligned} d_{1:T} &= G_\theta(\mathbf{i}, \mathbf{q}; d_{1:T-1}), \\ R(d_{1:T}) &= J_\phi(\mathbf{i}, \mathbf{q}, d_{1:T}), \\ \nabla_\theta L(\theta) &= -\mathbb{E}_{d_{1:T} \sim G_\theta} [R(d_{1:T}) \nabla_\theta \log G_\theta(d_{1:T})]. \end{aligned} \quad (3)$$

It is worth mentioning that the environment  $J_\phi$  can only output a reward value from a completed sequence  $\mathbf{d} = d_{1:T}$ . However, in DG-VQA setting and under the sequence generation scenario, the model should consider the partial reward of the incompleted sequence  $R(d_{1:t}), \forall t < T$ . To tackle this challenge, we follow the common practice to use the Monte Carlo search [45] to sample the unknown last  $T - t$  tokens at intermediate timesteps. In practice, the expected gradient can be approximated using several distractors  $\mathbf{d}^s$  sampled from  $G_\theta$  for each input image, question, and correct answer triplet in a minibatch.

$$\nabla_\theta L(\theta) \approx \sum_s -R(\mathbf{d}^s) \nabla_\theta \log G_\theta(\mathbf{d}^s). \quad (4)$$

## 4.2. The Agent: A neural distractor generator

In GOBBET framework, the agent is responsible for generating textual distractors according to the rewards from the environment. Therefore, the technical choice of the agent is flexible, as long as it can select actions based on observations from the environment and can update its policy parameters. From the possible choices such as MLP(multi-Layer

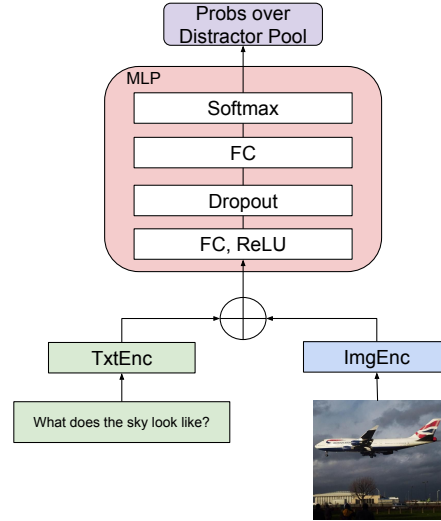


Figure 3. The model architecture for agent in GOBBET, where TxtEnc denotes text encoder, ImgEnc denotes image encoder, and FC denotes fully connected layer.

Perceptron), RNN, and Transformer, we select MLP for its advantages in training speed. Moreover, empirical results show that the simple MLP model is sufficient to generate challenging distractors under the guidance of the environment. Consequently, one complete sequence  $d_{1:T}$  can be generated for each training iteration by selecting the output distractor over a distractor candidate pool. Thus, the distractor generator can be formulated as follows:

$$\mathbf{d} = d_{1:T} = G_\theta(\mathbf{i}, \mathbf{q}). \quad (5)$$

Figure 3 depicts the details of the agent model’s architecture, which is essentially a encoder-decoder model. The encoder is a two-channel encoder that embeds the input question and input image into dense embedding vectors, while the decoder is a MLP that outputs the probabilities over the candidate distractor pool. For the encoder part, a text encoder and an image encoder are applied to the input question  $\mathbf{q}$  and image  $\mathbf{i}$ , respectively.

$$\mathbf{x}_q = \mathcal{F}_t(\mathbf{q}), \mathbf{x}_i = \mathcal{F}_i(\mathbf{i}), \quad (6)$$

where the text encoder  $\mathcal{F}_t$  is responsible to produce the dense embedding of a textual sequence  $\mathbf{q} = q_{1:T}$ , thus MLP, CNN, RNN, or Transformers can be chosen. We implement the average of the pre-trained word embeddings [32] as the representation of  $q$ . Similarly, the technical choice of image encoder  $\mathcal{F}_i$  is also flexible, and we opt to pre-trained deep CNN model [14].

After we get the embedding  $\mathbf{x}_q$  for input question and  $\mathbf{x}_i$  for input image, we fuse them together as the overall input context representation by  $\mathbf{c} = \mathbf{x}_i \oplus \mathbf{x}_q$ , as shown in Figure 3. The fuse operation  $\oplus$  can be concatenation, element-wise

summation, element-wise multiplication or bilinear pooling [7], where we choose concatenation in GOBBET implementation. The context embedding  $\mathbf{c}$  is then feed into the decoder, and we adopt a multilayer perceptron (MLP) for it:

$$\mathbf{z} = (w_2 \text{ReLU}(w_1 \mathbf{c} + b_1) + b_2), \quad (7)$$

where  $w_1, w_2, b_1, b_2$  denote the learnable parameters for the first or second layer in the MLP,  $\mathbf{z}$  denotes the predicted unnormalized distribution over the distractor pool. Finally, the output probability over the distractor pool is obtained by  $P(\mathbf{d}|\mathbf{i}, \mathbf{q}) = \text{softmax}(\mathbf{z})$ .

### 4.3. The Environment: VQA Models

In GOBBET framework, the environment  $J_\Phi$  is responsible for providing rewards according to the input context and the actions made by the agent (*i.e.* the distractors generated by the agent), as Equation 3 defines. More specifically,  $J_\Phi$  is a pre-trained multiple-choice VQA model that is fixed during the GOBBET training and testing process. In principle, we take the validity score  $J_\Phi(\mathbf{i}, \mathbf{q}, \mathbf{d})$  as the reward, as the validity score indicate to what extent the generated distractor  $\mathbf{d}$  is a “plausible” answer to the input context. Furthermore, we punish the distractor  $d$  which is semantically equivalent to the correct answer  $\mathbf{a}$  for the given context. The semantic similarity module is a BERT [5] model trained on sentence similarity tasks, and it produces a binary label that represents whether the two input sentences are semantically similar. Therefore, we define the reward function as follows:

$$R(d) = \begin{cases} -1 & \text{if IsSemEquiv}(\mathbf{d}, \mathbf{a}); \\ J_\Phi(\mathbf{i}, \mathbf{q}, \mathbf{d}) & \text{otherwise.} \end{cases} \quad (8)$$

Any multi-choice VQA model which produces a validity (sometimes also called likelihood) scores of responses for given visual questions can serve as the environment in GOBBET framework, such as TellingVQA [49], RevisitedVQA [19], MCB [7], *etc.* It is worth noting that the choice of environment is not restricted to the abovementioned models, but is generally applicable to any VQA models which can produce such scores. Moreover, Our proposed GOBBET also supports leveraging a bundle of pre-trained VQA models together as the environment to provide a combined reward.

## 5. Experiments

In this section, we evaluate our proposed GOBBET model focusing on the following research questions:

- *RQ1*: How does GOBBET perform in comparison to other methods?

- *RQ2*: Can generated distractors help build more robust VQA models?
- *RQ3*: How is the quality of generated distractors?

### 5.1. Experimental Settings

**Dataset.** We evaluate our model on *Visual7w* [49], which is a public multiple-choice visual question answering dataset. *Visual7w* consists of 47,300 images from COCO and 327,939 multiple-choice QA pairs collected on Amazon Mechanical Turk.

**Evaluation Metrics.** Traditional metrics of distractor generation for question answering [25, 26, 35], such as reliability and validity, often rely on manual evaluation, which are hard to scale. In order to enable the automatic evaluation, we define the ability of generated distractors to fool pre-trained VQA models as the metric, denoted as  $\Delta\text{Acc}$ .  $\Delta\text{Acc}$  is the difference between VQA model’s performance on the original distractors and on the generated distractors  $\text{Acc}_{\text{original}} - \text{Acc}_{\text{generated}}$ , *i.e.* the performance degradation of VQA model when presenting generated distractors instead of original distractors. The higher  $\Delta\text{Acc}$  is, the better-generated distractors are. In this work, we leverage the following popular VQA models for calculating  $\Delta\text{Acc}$ :

- **TellingVQA** [49] is a recurrent QA model with spatial attention. It first encodes the image through a pre-trained VGG-16 model [40]. Then it uses a one-layer LSTM to read the image encoding and all the question tokens. It continues to feed the answer choice tokens into LSTM, and would finally produce the validity score.
- **RevisitedVQA** [19] proposes a light architecture for MC VQA task. RevisitedVQA receives an image-question-answer triplet, encodes it, and utilizes a MLP to compute whether or not the triplet is correct.
- **MCB** [7] proposes a novel method called Multimodal Compact Bilinear pooling to efficiently and expressively combine language and vision features.

**Baseline Methods.** We compare our GOBBET with the following baseline methods:

- **Q-type prior** is a heuristic method for distractor generation. We select three most popular answers per question type as distractors.
- **Adversarial Matching** [47] forces distractors to be as relevant as possible to the context (image and question), while preventing distractors to be overly similar to the correct answer. We also employ BERT [5] to compute the relevance between the context and the distractor, and ESIM+ELMo [4, 18] to compute the similarity between the answer and the distractor. During

training, distractors are responses randomly sampled from the whole training response(answer choice) pool.

- **LSTM Q+I** [1] utilizes a two-layer LSTM to encode the input question and a VGGNet [40] to encode the input image. After a point-wise multiplication operation to fuse question embedding and image embedding, a multi-layer perceptron is employed to predict the response over a pre-defined response pool. Similar to *Adversarial Matching*, we construct the pool using all correct answers in the training set. We change the training targets from correct answers to incorrect ones. The incorrect responses are generated by pre-trained VQA models but discard the generated responses that are identical to correct answers.

## 5.2. Implementation Details

For the agent in GOBBET, we adopt a two-channel vision and language neural network that outputs probabilities over the candidate distractor pool. We set the candidate distractor frequency threshold to 20, to filter the candidate pool size  $K$  to 1516, which covers 2% of all training and validation choices. The questions are represented by 300-dim averaged word embeddings from the pre-trained fastText [2] model. We use all words in the training dataset to finetune the word embedding. In the experiment, we set the dropout rate to 0.5 in each hidden layer with a ReLU activation. We further set the maximum training epochs as 200 with the early stop strategy.

For the environment in GOBBET, we adopt RevisitedVQA model [19] for its superior efficiency and effectiveness. The pre-trained RevisitedVQA model outperforms other state-of-the-art models which are mentioned in § 4.3, as it achieves 65.8% accuracy on the Visual7W dataset. We evaluate the proposed GOBBET with two ablated versions:

- **GOBBET-base**: Model parameters are updated only through policy gradient, where the rewards are from the pre-trained VQA models as the environment.
- **GOBBET-warmup**: Reinforce algorithm is known to have a large variance. Inspired by Imitation Learning [16] and Teacher Forcing [22], we first train the agent model with correct answer choice using cross-entropy loss for a small size (80) epochs. The warmup training process is to prevent generating unstable results. Then we train the agent as in GOBBET-base.

## 5.3. Evaluate VQA Models on Generated Distractors (RQ1)

We answer RQ1: “How does GOBBET perform in comparison to other methods in terms of fooling pre-trained

VQA models” in this subsection. Table 1 shows the performance degradation of pre-trained VQA models when presenting to distractors generated by different DG-VQA models. Since the three pre-trained VQA models use different architectures, the distractor generation model requires high generalization capability to confuse all three of them. As can be seen, all baseline models yield poor quality distractors in terms of  $\Delta Acc$ . More specifically, *Q-type prior* fails to fool any VQA models. *Adversarial Matching* and *LSTM Q+I* lack generalization capability, which is only able to abate one or two VQA models’ accuracy in small margins. In contrast, our proposed GOBBET methods yield significant improvements on all three pre-trained VQA models. It is worth noting that GOBBET-base performs better on RevisitedVQA and MCB than GOBBET-warmup, while worse on TellingVQA( $\Delta Acc = -30.9\%$ ), It indicates that without the warmup process (*i.e.* fine-tuning on correct answers), the agent model is vulnerable to overfitting (*e.g.*  $Acc = 0.01\%$  for GOBBET-base when it receives rewards from RevisitedVQA and tries to fool it at the same time).

Furthermore, the larger  $\Delta Acc$  of GOBBET shows pre-trained VQA models provide an alternative knowledge source for DG-VQA, thus leading to GOBBET capable of generating high-quality distractors without any training samples. However, only receiving rewards from one specific environment is not robust. We address this issue by incorporating the warm-up process, in which case it provides a smoother beginning probability distribution over the candidate pool. Therefore, it prevents the agent from falling into the biased local minima trap.

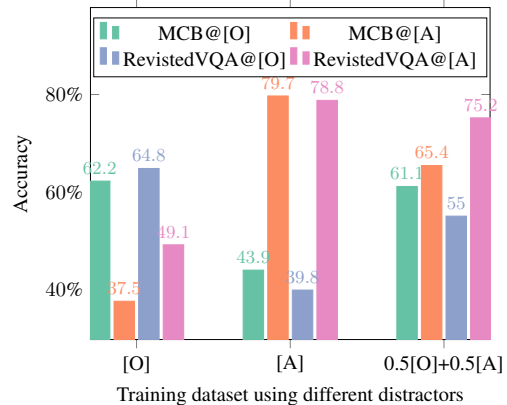


Figure 4. Data Augmentation Results

## 5.4. Augmenting VQA model with Generated Distractors (RQ2)

To answer RQ2, we utilize generated distractors as the augmented data to train more robust VQA models. In particular, we keep the correct answer of each input question image pair and swap the original distractors to the generated

Table 1. Pre-trained models performance on generated distractor for Visual7W dataset. The columns “TellingVQA”, “RevisitedVQA”, and “MCB” represent the VQA models that take the multiple-choice VQA assessment. The first row “Original distractors” indicates VQA models are presented to the original distractors, while other rows indicate the distractors are generated by corresponding DG-VQA models.  $\Delta Acc$  denotes the performance degradation of pre-trained VQA models, and higher  $\Delta Acc$  means the better generated distractors.

Model	TellingVQA [49]		RevisitedVQA [19]		MCB [7]	
	Acc	$\Delta Acc$	Acc	$\Delta Acc$	Acc	$\Delta Acc$
Original distractors	55.6%	-	64.8%	-	62.2%	-
Baselines						
<i>Q-type prior</i>	57.3%	-1.7%	68.7%	-3.9%	85.7%	-23.5%
<i>Adversarial Matching</i> [47]	54.7%	0.9%	71.7%	-6.9%	51.3%	10.9%
<i>LSTM Q+I</i> [1]	41.7%	13.9%	68.9%	-4.1%	85.7%	-23.5%
Proposed Methods						
Reward from RevisitedVQA						
- GOBBET-base	86.5%	-30.9%	<u>0.01%</u>	<b>64.7%</b>	<u>26.5%</u>	<b>35.7%</b>
- GOBBET-warmup	<u>33.7%</u>	<b>21.9%</b>	49.1%	15.8%	37.5%	24.7%

ones. MCB and RevisitedVQA are better suited for this setting since they take both correct and incorrect choices into consideration while training, while TellingVQA only takes the correct answer as input. Hence, we re-train MCB and RevisitedVQA models from scratch under two settings: 1) generated distractors alone; 2) the mixup of original and generated distractors. As a control group, we adopt the VQA models trained on the original distractors alone. All distractors are produced by our GOBBET-warmup, which has been shown the most effective DG-VQA model.

Figure 4 reports the results, where the x-axis denotes on which training set the models are trained, and the y-axis denotes the model performance in terms of accuracy.  $[O]$  and  $[A]$  refer to the original data and the augmented data respectively. And  $0.5[O] + 0.5[A]$  denotes 50% of all questions’ incorrect alternatives are replaced by the generated distractors. Different bars indicate the VQA accuracy that models can achieve on a specific testing set, e.g. the green bar denotes MCB model testing on the original distractors, while the orange bar denotes MCB models testing on the generated distractors. At first glance, we find that data augmentation training improves the models’ performance on generated distractors. However, it hurts the model performance on the original test data. We observe a similar pattern for models trained solely on original distractors. As a balance, models trained on the union of augmented and original data achieve the best performance with the minimum  $Acc@[O]$  drop of 1.1% and the highest  $Acc@[A]$  improvement by 27.9%. These results demonstrate the effectiveness of DG-VQA for training more robust VQA models.

### 5.5. Case Study (RQ3)

The case study is critical to answering RQ3, since the pre-trained VQA models’ performance degradation and the data augmentation effectiveness only indirectly validate the quality. Meanwhile, the widely used text generation mea-





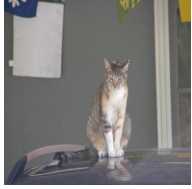
surements such as BLEU, ROUGE are not applicable either, because high quality is not necessarily related to n-gram similarity between original distractors and generate distractors. Thus, We collect textual distractor choices generated by baselines and the proposed methods, as can be seen in Table 2. We further analyze them in-depth, and have observed the following factors lead to high-quality and challenging distractors generated by GOBBET:

**Concept Similarity:** It is not surprising that GOBBET learns the strategy to replace correct answers with conceptually similar terms, as humans follow the same strategy to come up with distracting choices. As we can see, distractors generated by GOBBET and the correct answers almost belong to the same concept categories. For example, “baseball”, “soccer”, “tennis” (distractors by GOBBET-warmup of the third example) and “golf” (correct answer of the third example) are sport terms. And all distractors produced by GOBBET-warmup for the second question “how many black cows are there” are all numbers, which belong to the same category of the correct answer: “3”.

**Context Matters:** Another critical factor is input context. In the first column of Table 2, both distractors of the original dataset and augmented ones are adjectives to describe the weather. However, “cloudy” is better than “stormy” to depict the picture, compared to “hazy”, “windy” and “sunny”. Under the original choice setting, the defender can select the correct answer. But once encountered with the generated distractors, it is confusing and misleadingly pick “cloudy” as the answer. Tackling vision and language tasks needs multimodal cognitive ability. In the DG-VQA task, a system should comprehensively utilize information from both the given questions and the images.

**Attack the Weaknesses and Improve:** Our architecture is able to receive feedback from the defender (pre-trained VQA model). It is common to exploit opponents’ weaknesses to defeat them. By analyzing judgment scores of

Table 2. Excerpts from sampled original and adversarial generated distractor choices. Green choices are correct answers. Bold texts indicate options chosen by the pre-trained RevisitedVQA model in Visual7W.

				
Q:What does the sky look like?	Q:How many black cows are there?	Q:What sport are they playing?	Q:Why is there a piece missing?	Q:What two colors are in the flag directly above the cats head?
Original Choices				
A: <b>Stormy</b> ✓	A: <b>3</b> ✓	A: <b>Golf</b>	A: <b>Someone ate some</b>	A: <b>Green and yellow</b>
A: Hazy	A: 9	A: <b>Baseball</b> ✗	A: It was removed	A: <b>Blue and white</b> ✗
A: Windy	A: 8	A: Hockey	A: <b>It was put somewhere else</b> ✗	A: Black and red
A: Sunny	A: 7	A: Basketball	A: Someone took it	A: Green and black
Distractors by <i>Adversarial Matching</i>				
A: Stormy	A: 3	A: <b>Golf</b> ✓	A: Someone ate some	A: Green and yellow
A: <b>Sky</b> ✗	A: Zero	A: Volleyball	A: Wood	A: Blue and black
A: Blue	A: 5	A: Playing soccer	A: <b>Glass</b> ✗	A: Blue and red
A: Cloudy	A: <b>0</b> ✗	A: Soccer	A: To rest	A: <b>Blue and white</b> ✗
Distractors by GOBBET-base				
A: Stormy	A: 3	A: Golf	A: Someone ate some	A: Green and yellow
A: Shadows	A: <b>Shadows</b> ✗	A: Shadows	A: Shadows	A: <b>Shadows</b> ✗
A: Daylight	A: During daylight	A: <b>During daylight</b> ✗	A: <b>Daylight</b> ✗	A: Daylight
A: <b>Shadow</b> ✗	A: In the daytime	A: Daylight	A: During daylight	A: In the daytime
Distractors by GOBBET-warmup				
A: Stormy	A: 3	A: Golf	A: Someone ate some	A: <b>Green and yellow</b> ✓
A: <b>Cloudy</b> ✗	A: Two	A: <b>Baseball</b> ✗	A: <b>To eat</b> ✗	A: Blue
A: Blue	A: Four	A: Soccer	A: To cook	A: Legs
A: Clouds	A: <b>One</b> ✗	A: Tennis	A: For display	A: Orange

the alternatives, the distractor generator identifies the differences between the hard and the easy ones. Examples of this can be found in distractors generated by GOBBET-base (see the first question in Table 2). It seems that our system generates easy-to-human distractors like “shadows” or “daylight”. However, the defender is observed to be confused by them. A similar phenomenon has also been observed in [12], where neural networks are vulnerable to small perturbations on input images while humans can easily distinguish them. Our model is able to identify such tricky weaknesses of defender models and exploit them. Moreover, by considering these weaknesses for the next round of training, a model’s robustness is improved.

In summary, the case study supports that our method in fact outputs high-quality distractors by considering all together with the semantics of the correct answer, the information of the context, and the feedback from the trained VQA models as an alternative knowledge source.

## 6. Conclusion

In this work, we introduce the novel DG-VQA task. These generated “hard negative” distractors are significant since deep networks have been applied in many real-life

and safety-sensitive environments. One major challenge for DG-VQA is the sparsity of training samples. To address it, we developed the policy gradient-based GOBBET, where pre-trained VQA models serve as the alternative knowledge source to guide the distractor generation. Furthermore, the generated distractors can provide insights into factors that cause VQA models vulnerable.

Recent advances in text and image retrieval have enabled many multi-modality applications to deal with open-world, knowledge-based scenarios. Instead of relying on the established knowledge base, GOBBET paves a new pathway for future research that leverages pre-trained VQA models as an underlying multi-modal knowledge base. Whilst learning pre-training tasks, these models may also be storing latent cross-modality knowledge present in the training data. Compared to established structured KBs, pre-trained models have many advantages, such as they do not require a pre-defined schema or ad-hoc canonicalization process, thus enabling them easy to extend to different domains.

**Acknowledgement.** This work was supported by the National Science Foundation under Grant CNS-2101052, IIS-2132724 and IIS-1750082.



## References

- [1] Stanislaw Antol, Aishwarya Agrawal, Jiaseen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 1, 2, 6, 7
- [2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 6
- [3] Chia-Yin Chen, Hsien-Chin Liou, and Jason S Chang. Fast: an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 1–4. Association for Computational Linguistics, 2006. 3
- [4] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, 2017. 5
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019. 5
- [6] Zhihao Fan, Zhongyu Wei, Siyuan Wang, Yang Liu, and Xuanjing Huang. A reinforcement learning framework for natural question generation using bi-discriminators. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1763–1774, 2018. 3
- [7] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, 2016. 2, 5, 7
- [8] Yifan Gao, Lidong Bing, Piji Li, Irwin King, and Michael R Lyu. Generating distractors for reading comprehension questions from real examinations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6423–6430, 2019. 2, 3
- [9] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, page 201422953, 2015. 2
- [10] Tejas Gokhale, Abhishek Chaudhary, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Semantically distributed robust optimization for vision-and-language inference. In *Findings of the Association for Computational Linguistics*, 2022. 3
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3
- [12] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 8
- [13] Hubbard C Goodrich. Distractor efficiency in foreign language testing. *Tesol Quarterly*, pages 69–78, 1977. 3
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 4
- [15] Jennifer Hill and Rahul Simha. Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and google n-grams. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 23–30, 2016. 2
- [16] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016. 6
- [17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2
- [18] Suzana Ilic, Edison Marrese-Taylor, Jorge Balazs, and Yutaka Matsuo. Deep contextualized word representations for detecting sarcasm and irony. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–7, 2018. 5
- [19] Allan Jabri, Armand Joulin, and Laurens van der Maaten. Revisiting visual question answering baselines. In *European conference on computer vision*, pages 727–739. Springer, 2016. 2, 3, 5, 6, 7
- [20] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 2
- [21] Girish Kumar, Rafael Banchs, and Luis Fernando D’Haro. Revup: Automatic gap-fill question generation from educational texts. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 154–161, 2015. 3
- [22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. 6
- [23] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2
- [24] Yan Li and Jieping Ye. Learning adversarial networks for semi-supervised text classification via policy gradient. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1715–1723. ACM, 2018. 2
- [25] Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C Lee Giles. Distractor generation for multiple choice questions using learning to rank. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 284–290, 2018. 2, 3, 5
- [26] Chen Liang, Xiao Yang, Drew Wham, Bart Pursel, Rebecca Passonneau, and C Lee Giles. Distractor generation with

- generative adversarial nets for automatically creating fill-in-the-blank questions. In *Proceedings of the Knowledge Capture Conference*, page 33. ACM, 2017. 5
- [27] Feng Liu, Tao Xiang, Timothy M Hospedales, Wankun Yang, and Changyin Sun. ivqa: Inverse visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8611–8619, 2018. 3
- [28] Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. Weakly-supervised visual-retriever-reader for knowledge-based question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6417–6431, 2021. 3
- [29] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE international conference on computer vision*, pages 1–9, 2015. 2
- [30] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3
- [31] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018. 3
- [32] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 4
- [33] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 3
- [34] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, 2019. 3
- [35] Van-Minh Pho, Thibault André, Anne-Laure Ligozat, Brigitte Grau, Gabriel Illouz, Thomas François, et al. Multiple choice question corpus analysis for distractor characterization. In *LREC*, pages 4284–4291, 2014. 5
- [36] Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 1060–1069. JMLR.org, 2016. 3
- [37] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017. 2, 3
- [38] Keisuke Sakaguchi, Yuki Arase, and Mamoru Komachi. Discriminative approach to fill-in-the-blank quiz generation for language learners. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 238–242, 2013. 3
- [39] Shailaja Keyur Sampat, Akshay Kumar, Yezhou Yang, and Chitta Baral. Clevr-hyp: A challenge dataset and baselines for visual question answering with hypothetical actions over images. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3692–3709, 2021. 2
- [40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5, 6
- [41] Katherine Stasaski and Marti A Hearst. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312, 2017. 3
- [42] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 3, 4
- [43] Alan M Turing. Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65. Springer, 2009. 2
- [44] Chenguang Wang, Xiao Liu, and Dawn Song. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*, 2020. 3
- [45] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 3, 4
- [46] Houpu Yao, Zhe Wang, Guangyu Nie, Yassine Mazbouidi, Yezhou Yang, and Yi Ren. Improving model robustness with transformation-invariant attacks. *arXiv preprint arXiv:1901.11188*, 2019. 3
- [47] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731, 2019. 1, 2, 5, 7
- [48] Junjie Zhang, Qi Wu, Chunhua Shen, Jian Zhang, Jianfeng Lu, and Anton van den Hengel. Asking the difficult questions: Goal-oriented visual question generation via intermediate rewards. *arXiv preprint arXiv:1711.07614*, 2017. 3
- [49] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004, 2016. 1, 2, 5, 7
- [50] Yuke Zhu, Ce Zhang, Christopher Ré, and Li Fei-Fei. Building a large-scale multimodal knowledge base system for answering visual queries. *arXiv preprint arXiv:1507.05670*, 2015. 3