Improving VQA and its Explanations by Comparing Competing Explanations

Jialin Wu
Department of Computer Science
University of Texas at Austin
jialinwu@cs.utexas.edu

Liyan Chen
Department of Computer Science
University of Texas at Austin
liyanc@cs.utexas.edu

Raymond J. Mooney
Department of Computer Science
University of Texas at Austin
mooney@cs.utexas.edu

Abstract

Most recent state-of-the-art Visual Question Answering (VQA) systems are opaque black boxes that are only trained to fit the answer distribution given the question and visual content. As a result, these systems frequently take shortcuts, focusing on simple visual concepts or question priors. This phenomenon becomes more problematic as the questions become complex that requires more reasoning and commonsense knowledge. To address this issue, we present a novel framework that uses explanations for competing answers to help VQA systems select the correct answer. By training on human textual explanations, our framework builds better representations for the questions and visual content, and then reweights confidences in the answer candidates using either generated or retrieved explanations from the training set. We evaluate our framework on the VQA-X dataset, which has more difficult questions with human explanations, achieving new state-of-the-art results on both VQA and its explanations.

1 Introduction

Recently, Visual Question Answering (VQA) (Antol et al., 2015; Hudson and Manning, 2019; Singh et al., 2019; Marino et al., 2019) has emerged as a challenging task that requires artificial intelligence systems to predict answers by jointly analyzing both natural language questions and visual content. Most state-of-the-art VQA systems (Anderson et al., 2018; Kim et al., 2018; Ben-Younes et al., 2017; Jiang et al., 2018; Cadene et al., 2019; Lu et al., 2019; Liu et al., 2019; Tan and Bansal, 2019) are trained to simply fit the answer distribution using question and visual features and achieve high performance on simple visual questions. However, these systems often exhibit poor explanatory capabilities and take shortcuts by only focusing on simple visual concepts or question priors instead of finding the right answer for the right reasons (Ross et al., 2017; Selvaraju et al., 2019). This problem becomes increasingly severe when the questions require more complex reasoning and commonsense knowledge.

For more complex questions, VQA systems need to be right for the right reasons in order to generalize well to test problems. Two ways to provide these reasons are to crowdsource human visual explanations (Das et al., 2017) or textual explanations (Park et al., 2018). While visual explanations only annotate which parts of an image contribute most to the answer, textual explanations encode richer information such as detailed attributes, relationships, or commonsense knowledge that is not necessarily directly found in the image. Therefore, we adopt textual explanations to guide VQA systems.

Recent research that utilizes textual explanations adopts a multi-task learning strategy that jointly trains an answer predictor and an explanation generator (Li et al., 2018; Park et al., 2018). However, this approach only considers explanations for the one chosen answer. Our approach considers explanations for multiple competing answers, comparing these explanations when choosing a final answer, as shown in Figure 1.

Our framework is end-to-end trainable and therefore can be applied to differentiable VQA systems. We perform empirical analysis and show improvements of our method combined with Up-Down (Anderson et al., 2018) and LXMERT (Tan and Bansal, 2019) on the VQA-X dataset (Park et al., 2018). VQA-X is the dataset of choice for it requires more cognitive maturity, where the questions are selected such that the cognitive ability of a 9-year old is the minimum competency requirement of answering such questions with reasonable textual explanations. In the empirical analysis, we also show that our approach learns better representations for the questions and visual content by training to re-

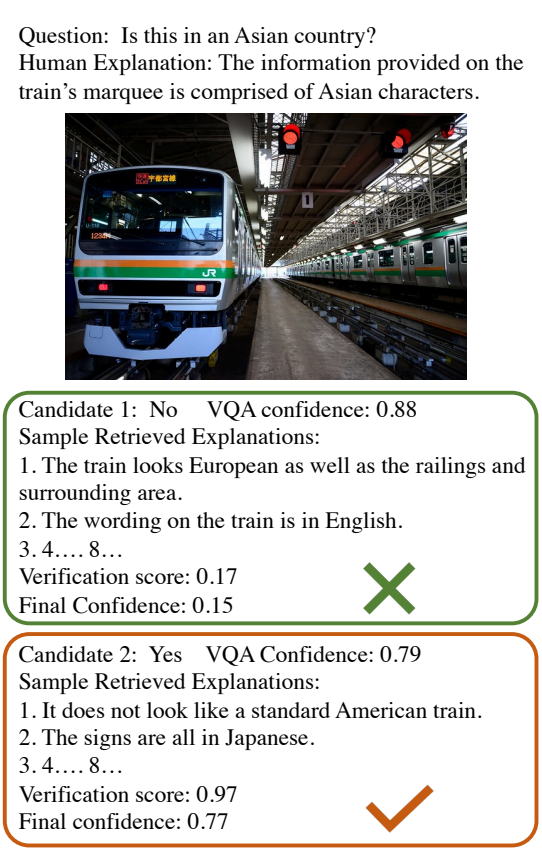

Figure 1: An example of utilizing retrieved explanations to correct the original VQA prediction. Though the original VQA confidence of the correct answer "Yes" is lower than that of the incorrect answer "No", the retrieved explanations for "Yes" support their answer better, resulting in a higher verification score and then the final correct decision.

trieve explanations, and achieves state-of-the-art results by further jointly considering competing explanations.

We also developed a new explanation generation approach for VQA by utilizing competing explanations. This approach uses retrieved explanations for each competing answer to help generate improved explanations during testing. Using both automated metrics comparing to human explanations and human evaluation, we show that these explanations are also improved, beating the current state-of-the-art method presented by Wu and Mooney (2019a).

2 Related Work

2.1 VQA with Human Visual Explanations

In order to train a VQA system to be right for the right reason, recent research first collects human visual attention (Das et al., 2017; Gan et al., 2017) that highlights the image regions that most contribute to the answer. Specifically, two popular ways are to have crowdsourced human workers to deblur the image (Das et al., 2017) or select segmented objects from the image (Gan et al., 2017). Then, the VQA systems try to align either the VQA system's attention (Qiao et al., 2018; Zhang et al., 2019) or the gradient-based visual explanation (Selvaraju et al., 2019; Wu and Mooney, 2019b) to the human attentions. These approaches help the systems focus on the right places, and improve VQA performance when the training and test distributions are very different, such as in the VQA-CP dataset (Agrawal et al., 2018).

2.2 Human Textual Explanations

While human visual explanations can help VQA systems know *where* to attend, human textual explanations can also provide information on *how* the attended image regions contribute to the answer. There are two large textual explanation datasets, VQA-E (Li et al., 2018) and VQA-X (Park et al., 2018). Explanations in VQA-E are automatically refined versions of the most relevant captions from the COCO dataset (Chen et al., 2015), which have a larger scale but are of less quality than human-written ones. Therefore, we adopt VQA-X, where crowdsourced human workers were directly asked to provide textual explanations for the questions that are judged to require children older than 9 years to answer.

2.3 Generating Textual Explanations

In order to automatically generate textual explanations, Park et al. (2018) present a single-layer LSTM network trained on image, question, and answer features to mimic crowdsourced human explanations. Wu and Mooney (2019a) uses question-attended segmentation features from the original VQA system as input, and try to generate explanations that are more faithful to the actual VQA process at the object level.

2.4 VQA with Human Textual Explanations

When visual questions become more complex and require more reasoning steps and general knowledge, visual attention, which only shows the important regions, is less helpful to improving performance. However, fewer projects are focusing on using textual explanations to help VQA. We are aware of two papers in this thread. Li et al. (2018) trains a VQA system to jointly predict the answer

and generate an explanation. However, the generated explanations may not faithfully reflect the actual VQA process, but rather hallucinate visual content (Rohrbach et al., 2018), and therefore, are not guaranteed to supervise the underlying VQA system properly. Wu and Mooney (2019b) only use textual explanations to extract a set of important visual objects, but ignore other critical richer content, $e.g.$, attributes, relationships, commonsense knowledge, etc. In contrast, our approach trains a system to distinguish correct human explanations from all the competing explanations that support incorrect answer candidates.

3 Preliminaries

This section introduces the two baseline VQA models used in our work, Bottom-up Top-down (UpDn) (Anderson et al., 2018) and LXMERT (Tan and Bansal, 2019).

A large number of previous VQA systems (Fukui et al., 2016; Ben-Younes et al., 2017; Ramakrishnan et al., 2018) utilize a trainable top-down attention mechanism over convolutional features to recognize relevant image regions. Anderson et al. (2018) introduced complementary bottom-up attention that first detects common objects and attributes so that the top-down attention can directly model the contribution of higher-level concepts. This Up-Down (UpDn) approach is widely used in recent work (Selvaraju et al., 2019; Jiang et al., 2018; Lu et al., 2019; Tan and Bansal, 2019) and significantly improves VQA performance.

UpDn systems first extract a visual feature set $\mathcal{V} = \{\mathbf{v}_i, ..., \mathbf{v}_{|\mathcal{V}|}\}$ for each image whose element $\mathbf{v}_i$ is a feature vector for the i-th detected object. On the language side, UpDn systems sequentially encode each question Q to produce a question vector $\mathbf{q}$. Let f denote the answer prediction operator that takes both visual features and question features as input and predicts the confidence for each answer a in the answer candidate set $\mathcal{A}$, $i.e.$ $P(a|\mathcal{V}, Q) = f(\mathcal{V}, \mathbf{q})$. The VQA task is framed as a multi-label regression problem with the gold-standard soft scores as targets in order to be consistent with the evaluation metric. In particular, the standard binary cross entropy loss with soft score is used to supervise the sigmoid-normalized outputs.

We briefly introduce two variants of this approach adopted in our experiments:

UpDn. This is the original UpDn system, which uses a single layer GRU to encode questions. The question vector is then used to compute a single-stage attention over the detected objects to produce attended visual features. Finally, a two-layer feed-forward network is used to compute the answer probability.

LXMERT. In order to learn richer representations for both questions and visual content, LXMERT uses transformers (Vaswani et al., 2017; Devlin et al., 2018) that learn multiple layers of attention over the input. In particular, it first learns 9 layers over the input question and 5 layers over detected objects, then finally learns another 5 layers of attention across the two modalities to produce the final joint representation.

4 Approach

This section presents our approach to utilizing competing explanations to aid VQA. As shown in Figure 2, after the base VQA system computes the top ten answers, our approach retrieves the most supportive explanations for each answer from the training set to construct the set of competing explanations. Then, these explanations are used to help generate explanations for the current question. Next, we learn to predict verification scores that indicates how well the retrieved or generated explanations support the predictions given the input question and visual content. The final answer is determined by jointly considering the original answer probabilities and these verification scores.

4.1 Retrieving Explanations

This section presents our approach to retrieving the most supportive human textual explanation from the training set for each answer candidate. Ideally, we should dynamically retrieve explanations for each answer at each iteration. However, it will be very computational costly because the question and visual features have to be computed for each image from the training set. Therefore, we adopt the below relaxation for computational efficiency that only needs to compute the features once.

In particular, we first pretrain the VQA system, and extract the question and visual embeddings, $\mathbf{q}$ and $\mathbf{v}$,for each $Q\mathcal{V}$ pair in the training set. For UpDn, we use the attended visual features and the question GRU's last hidden state as the visual and question embeddings. For LXMERT, we use the last cross-modal attention layer's visual and question output as the embeddings.

Then, for each $Q\mathcal{V}$ pair, we only compute the

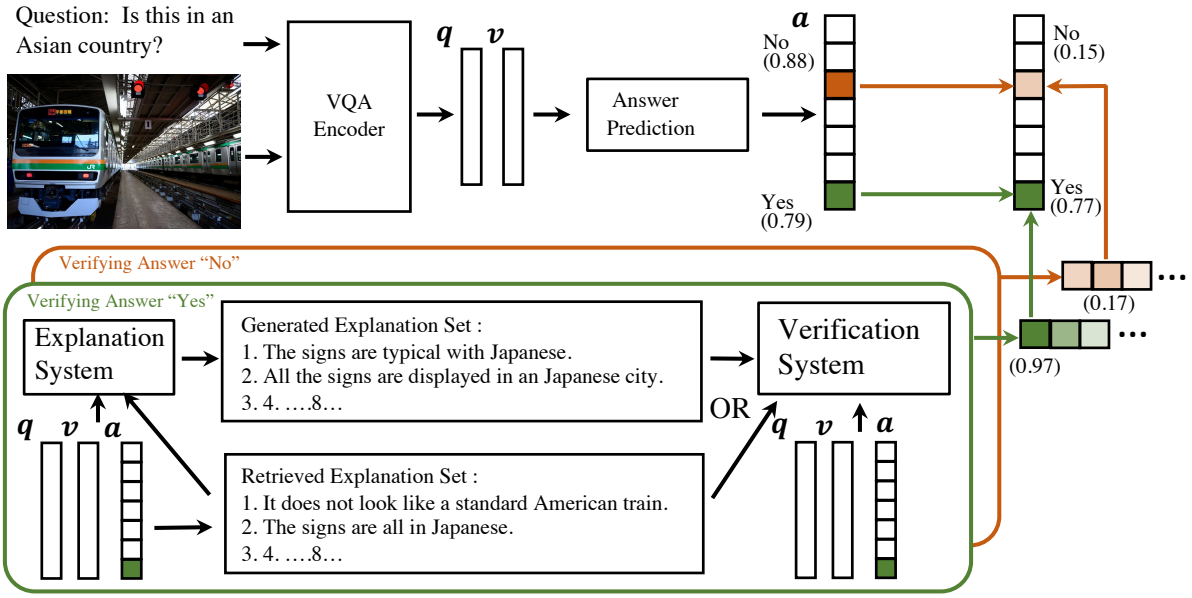

Figure 2: Our approach first predicts a set of answer candidates and retrieves explanations for each one of them based on the answer, question, and visual content. The retrieved explanations are then used to generate improved explanations for this example. Finally, either retrieved or generated explanations are employed to predict verification scores that are used to reweight the original predictions and compute the final answer.

top-10 answer candidates since the top-10 answers have already achieved high recall. After that, for each answer candidate a, we extract explanations from the training set that have the same ground truth answer [1] as the current candidate. We then sort these explanations by the L2 distance between the explanations' $\mathcal{QV}$ embeddings, $\mathbf{q} \odot \mathbf{v}$, and the example's and pick the closest 8 explanations as the competing explanations set denoted as $\mathcal{X}_a$.

4.2 Generating Explanations

This section presents how we use the retrieved explanations for similar VQA examples from the training set to help generate better explanations.

We adopt the explainer from Wu and Mooney (2019a), a two-layer LSTM network similar to the UpDn captioner (Anderson et al., 2018), as our baseline. Since the current VQA systems are built upon detected objects, we use them as the visual inputs instead of segmentations.

The baseline explainer first computes a set of question-attended visual features, $\mathcal{U}$, and an average pooled version, $\bar{\mathbf{u}}$. The explainer then uses $\bar{\mathbf{u}}$ and $\mathcal{U}$ together with question and answer embeddings as inputs to produce explanations.

Our approach simply replaces the average pooled question-attended visual features $\bar{\mathbf{u}}$ with the retrieved explanations' features, $\mathbf{x}$. We use a single-layer GRU to encode all of the retrieved explanations for the correct answer, and then max pool the last hidden states among these explanations to compute $\mathbf{x}$. We sample 8 explanations for each answer candidate to construct the generated explanation set.

4.3 Learning Verification Scores

This section presents how we train a verification system to score how well a generated or retrieved explanation supports a corresponding answer candidate given the question and visual content. The verification system takes four inputs: the visual, question, answer and its explanation features; and outputs the verification score as shown in Eq. 1:

$$S(Q, \mathcal{V}, a, x) = \sigma(f_2(f(\mathbf{q}), f(\mathbf{v}), f(\mathbf{a}), f(\phi(x)))) \quad (1)$$

where $\mathbf{a}$ is the one-hot embedding of the answer, and $\phi(x)$ is the feature vector for the explanation, x, encoded using GRU (Cho et al., 2014), ϕ. We use f_n to denote n consecutive feed-forward layers (for simplicity n is omitted when $n = 1$). We use σ to denote the sigmoid function. The verification system is similar to the answer predictor in architecture except for the number of outputs, $i.e.$ 1 for the verification system and $3,129$ for the answer predictor.

[1] More specifically, the soft score of the answer candidate in the retrieved explanation's example is over 0.6

Given the VQA examples with their explanations in the VQA-X dataset, we use binary cross-entropy loss to maximize the verification score for the matching explanations as shown in Eq. 2:

$$\mathcal{L}_m = -\log(S(Q, \mathcal{V}, a, x)) \tag{2}$$

Intuitively, we want the verification score S to be high only when the explanation is matched to the VQA example, *i.e.* replacement of any of the four input sources should lower the score. Therefore, we design five kinds of replacements for constructing negative examples below.

Replacement of Visual and Question Features. Ideally, we should replace the visual and question features with the complementary features (Antol et al., 2015) that lead to the opposite answer. For example, for the question "Is this a vegetarian pizza?", with an image of a vegetarian pizza, we should replace the image with one of a meat pizza. However, such replacement requires retrieving and computing the visual features $\mathbf{v}$ for the meat pizza, which is computationally inefficient. Therefore, we simply randomly choose a Q' or $\mathcal{V}'$ replacement from the current batch and minimize the binary cross entropy loss for the verification scores as shown in Eq. 3 and 4:

$$\mathcal{L}_r^q = -\log(1 - S(Q', \mathcal{V}, a, x)) \tag{3}$$
$$\mathcal{L}_r^v = -\log(1 - S(Q, \mathcal{V}', a, x)) \tag{4}$$

Replacement of Answer Features. We sample the answer for replacement according to the current VQA's predicted incorrect probabilities. At each step, we try to minimize the expectation of binary cross-entropy loss for the incorrect predictions as shown in Eq. 5:

$$\mathcal{L}_r^a = \mathbb{E}_{a' \sim p(a'|QV),\ s(a')<0.6}\big[-\log(1 - S(Q, \mathcal{V}, a', x))\big] \tag{5}$$

where $s(a')$ denotes the human VQA soft score for answer a'. In practice, we only sample one incorrect answer during training.

Replacement of Explanation Features. We try to replace the matched human explanations with the most supportive explanations for the sampled incorrect answer and train our verification system to disproof that explanation as shown in Eq. 6. In particular, given the sampled incorrect answer a' from the previous section, we compute the verification score for all the retrieved or generated explanation x' from the set $\mathcal{X}_{a'}$ for that wrong answer and regard the one with maximum verification score as the most supportive one.

$$\mathcal{L}_r^a = \max_{x' \in \mathcal{X}_{a'}} \big[-\log(1 - S(Q, \mathcal{V}, a, x'))\big] \tag{6}$$

Replacement of Answer and Explanation Features. To further prevent the system from being falsely confident in the sampled incorrect answer a', we also minimize the verification score for its most supportive explanation for the incorrect answer a' as shown in Eq. 7:

$$\mathcal{L}_r^{ax} = -\log\big(\max_{x' \in \mathcal{X}_{a'}} (1 - S(Q, \mathcal{V}, a', x'))\big) \tag{7}$$

To sum up, the verification loss is the sum of the aforementioned 6 losses as shown in Eq. 8:

$$\mathcal{L}_{verification} = 10\mathcal{L}_m + \mathcal{L}_r^q + \mathcal{L}_r^v + \mathcal{L}_r^a + \mathcal{L}_r^x + \mathcal{L}_r^{ax} \tag{8}$$

Because we have more negative examples, we assign a larger (*i.e.* 10) loss weight for the only positive example.

4.4 Using Verification Scores

This section presents the approach to employing the verification scores to reweight the original VQA predictions using the chain rule, where we regard the score $S(Q, \mathcal{V}, a, x)$ as the probability of the explanation x given the question Q, answer a, and visual content $\mathcal{V}$, *i.e.* $P(x|a, Q, \mathcal{V})$.

The original VQA system provides the answer probabilities conditioned on the question and visual content, *i.e.* $P(a|Q, \mathcal{V})$. Our approach tries to model the joint probability of the answer and its human explanation x_a conditioned on the question and visual content, *i.e.* $P(a, x|Q, \mathcal{V})$ using the chain rule as shown in Eq. 9:

$$P(a, x_a|Q, \mathcal{V}) = P(a|Q, \mathcal{V}) \times P(x_a|a, Q, \mathcal{V}) \tag{9}$$

Since we try to select the correct answer with its explanation, the conditional joint probability $P(a, x|Q, \mathcal{V})$ should only be high when the answer a is correct and the explanation x supports a, which is enforced using the loss in Eq. 10:

$$\mathcal{L}_{vqae} = -\log(P(a, x_a|Q, \mathcal{V})) + -\log(1 - \max_{x' \in \mathcal{X}'_a} P(a', x'|Q, \mathcal{V})) \tag{10}$$

During testing, we first extract the top 10 answer candidates $\mathcal{A}$, and then select the explanation for the answer candidate with the highest verification score. Then, we compute the joint probability for each answer candidate with its most supportive explanation to determine the final answer $a^\star$ as shown in Eq. 11:

$$a^\star = \arg\max_{a \in \mathcal{A}} \max_{x \in \mathcal{X}_a} P(a, x|Q, \mathcal{V}) \quad (11)$$

4.5 Implementation and Training Details

The section presents the implementation details and training procedure for our system.

Training Details. We first pre-train our base VQA system (Up-Down or LXMERT) on either the entire VQA v2 training set for 20 epochs or only the VQA-X training set for 30 epochs with the standard VQA loss (binary cross-entropy loss with soft scores as supervision) and the Adam optimizer (Kingma and Ba, 2015). As the VQA-X validation and test set are both from the VQA v2 validation set that is covered in the LXMERT pretraining, we do not use the officially released LXMERT parameters. The learning rate is fixed to 5e-4 for UpDn and 5e-5 for LXMERT, with a batch size of 384 during the pre-training process. For answer prediction part, We use $1,280$ hidden units in UpDn and 768 hidden units in LXMERT, and for verification part, we use $1,280$ hidden units in both systems.

We fine-tune our system using the verification loss and VQA loss $\mathcal{L}_{verification} + 0.1\mathcal{L}_{vqae}$ on the VQA-X training set for another 40 epochs. The initial learning rate for VQA system is set to be same as the pretraining if the system is pretrained on VQA-X training set, and $0.1\times$ if the system is pretrained on VQA v2 training set. For the verification systems, the initial learning rate is set to 0.0005. The learning rate for every parameters are decayed by 0.8 every 5 epochs.

During test, we consider the top-10 answer candidates for the VQA systems and use the explanation-reweighted prediction as the final answer.

Implementation. We implemented our approach on top of the original UpDn and LXMERT. Both base systems utilize a Faster R-CNN head (Ren et al., 2015) in conjunction with a ResNet-101 base network (He et al., 2016) as the object detection module. The detection head is pre-trained on the Visual Genome dataset (Krishna et al., 2017) and is capable of detecting $1,600$ objects categories and 400 attributes. Both base systems take the final detection outputs and perform non-maximum suppression (NMS) for each object category using an IoU threshold of 0.7. Convolutional features for the top 36 objects are then extracted for each image as the visual features, $i.e.$ a $2,048$ dimensional vector for each object. For question embedding, following (Anderson et al., 2018), we perform standard text pre-processing and tokenization for UpDn. In particular, questions are first converted to lower case, trimmed to a maximum of 14 words, and tokenized by white spaces. A single layer GRU (Cho et al., 2014) is used to sequentially process the word vectors and produce a sentential representation for the pre-processed question. We also use Glove vectors (Pennington et al., 2014) to initialize the word embedding matrix when embedding the questions. For LXMERT, we also follow the original BERT word-level sentence embedding strategy that first splits the sentence into words $w_1, ..., w_n$ with length of n by the same WordPiece tokenizer (Wu et al., 2016) in Devlin et al. (2018). Next, the word and its index ($i.e.$ absolute position in the sentence) are projected to vectors by embedding sub-layers, and then added to the index-aware word embeddings. We use a single-layer GRU and three-layer GRU to encode the generated or retrieved explanation in the verification system when using UpDn and LXMERT as base system, respectively.

5 Experimental Results

First, we perform empirical analysis on the VQA perform with the VQA-X (Park et al., 2018) dataset where the questions require more cognitive maturity than the original VQA-v2 dataset does. We combine the validation set (1,459 examples) and test set (1,968 examples) of the VQA-X dataset as our larger test set (3,427 examples) for more stable results because the both of them are reletively small. We compare our system's VQA performance against two corresponding base systems with the standard protocol. In addition, we examine the quality of explanations by comparing our system against a baseline model as well as human oracles. Finally, we perform ablation studies to show improved feature representation and the reweighting operation are key components of the improvements.

5.1 Results on VQA performance

Table 1 reports the results of our competing explanation approach. Our approach combined with

	VQA-X Pretrain		VQA v2 Pretrain	
	Gen. Expl.	Re. Expl.	Gen. Expl.	Re. Expl.
UpDn (Anderson et al., 2018)	74.2	74.2	83.6	83.6
UpDn+E (ours)	78.0	78.7	85.1	85.4
LXMERT (Tan and Bansal, 2019)	76.8	76.8	83.7	83.7
LXMERT+E (ours)	77.3	78.0	84.1	84.7

Table 1: Experimental results comparison on VQA-X dataset. The first and second half reports the VQA performance using UpDn and LXMERT as base system, respectively. "+E" denotes using our competing explanations approach. "Gen. Expl." and "Re. Expl." denote generated and retrieved explanations, respectively.

	Automatic Evaluation					AMT %
	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	
Faith. Expl. (Wu and Mooney, 2019a)	25.0	20.0	47.1	91.1	18.6	49.5
Faith. Expl. + E (ours)	26.4	20.4	48.5	95.3	18.7	55.6

Table 2: Automatic and human evaluation of our generated explanations. "AMT %" denotes the human evaluation scores in percentage. We use beam search with a beam size of 2.

UpDn pretrained on the entire VQA v2 dataset achieves the best results. When training only on the VQA-X training set, we improve the original UpDn and LXMERT by 4.5 % and 1.2 %, respectively. UpDn benefits more from using competing explanations than LXMERT, but both improve. By using transformers, LXMERT already creates better, but less flexible, representations which are harder to improve upon by using explanations.

5.2 Results on Explanation Generation

We evaluate the generated explanations using both automatic evaluation metrics ($i.e.$ BLEU-4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016)) and human evaluation using Amazon Mechanical Turk (AMT). The explanation generator uses the features from the UpDn VQA system pretrained on VQA v2. We use beam search with a beam size of 2 when generating the explanations.

In the AMT human evaluation, we compare our system against a baseline model as well as human oracles with judgements from AMT turkers. In order to measure the difference in explanation quality relative to standard oracles, we perform two groups of comparisons: our system v.s. oracles and baseline v.s. oracles, where each group contains 500 randomly sampled sets of comparisons. Each set of comparison consists of a (question, image, answer) triplet and two randomly ordered candidate explanations for such triplet. Three turkers were asked to give judgements for each set in the form of picking the better one or recognizing a tie. Finally, we quantitatively aggregate turker votes by assigning 2 points to the winning one, 0 to the losing one, and 1 to each in the case of tie, such that each comparison is zero-sum and robust to noise to some extent. Assuming human oracles as a standard proxy to enable direct comparisons between our system and the baseline, we present the aggregated results in the form of normalized scores, where the raw scores are inversely scaled by the corresponding raw oracle scores such that oracles have a unit score and normalized scores show how much advantage the system wins over oracles.

As reported in Table 2, by additionally conditioning on human explanations for similar visual questions, the generated explanations achieve better automatic scores and higher human ratings.

5.3 Generated vs. Retrieved Explanations

This section compares using retrieved explanations vs. generated explanations. We report the overall score using UpDn pretrained on VQA v2 train set as our base system in Table 3.

Our system with retrieved explanations performs slightly better than the one with generated explanations. This is probably because there are no guarantees that the generated explanations will support the answers, which they conditioned on, unlike the retrieved explanations. As a result, since all the explanation training examples are for correct

	VQA-X
UpDn (Anderson et al., 2018)	83.6
UpDn with generated explanations	85.1
UpDn with retrieved explanations	85.4

Table 3: Comparison of the performance using generated and retrieved explanations.

answers, the explanation generation system tends to support the ground truth answer whatever the answer candidates are. Also, the generated explanations sometimes ignore or hallucinate (Rohrbach et al., 2018) visual content when explaining the answer. Therefore, although intuitively, generated explanations could work better than retrieved ones, they are currently less helpful to the VQA performance due to their imperfections.

5.4 Human v.s. Retrieved Explanations

We compare our system based on UpDn pretrained on VQA-X using retrieved explanations to one using human oracle explanations, and report the results in Table 4. In particular, we present two settings that use human explanations during test. In the first setting, denoted by $\mathcal{RR}$, we only $\mathcal{R}$eplace the retrieved explanation for the $\mathcal{R}$ight answer with the corresponding human ones, and still use retrieved explanation for the incorrect answer. This setting shows how much retrieved explanations for correct answers degenerates the results. The second setting, denoted by $\mathcal{RA}$, assumes that human explanations are used to $\mathcal{R}$eplace the retrieved explanations for $\mathcal{A}$ll the potential answer candidates. This setting provides an upper bound on our approach that uses textual explanations.

	VQA-X
UpDn (Anderson et al., 2018)	74.2
UpDn with retrieved explanations	78.7
UpDn with human explanations ($\mathcal{RR}$)	79.3
UpDn with human explanations ($\mathcal{RA}$)	80.2

Table 4: Comparison of the performance using retrieved and oracle human explanations.

Not surprisingly, the human oracle explanations help the VQA system more than the retrieved ones. It indicates that our approach could achieve even better performance with more informative explanations, which could be achieved by either developing a better explanation generator, or enlarging the explanation training set from which human explanations are retrieved. Our system's results using retrieved explanations is only 0.6% lower than with human oracle explanations for the correct answers. This indicates that the retrieved explanations (for related questions) are a reasonable approximation to oracle ones (human explanations for the exact question).

5.5 Evaluating Representation Improvement

This section presents an ablation investigating how our approach improves the learned representations. The "w/o reweighting" ablation still uses the fine-tuned representation trained using explanations, but it does not reweight the final predictions, therefore it tests the improvement due to better representations alone. The "fixed VQA" ablation uses reweighting, but does not fine-tune the VQA parameters during verification-score training ($i.e.$ only the verification parameters are trained).

	VQA-X
UpDn (Anderson et al., 2018)	74.2
UpDn+E (w/o reweighting)	77.8
UpDn+E (fixed VQA)	75.3
UpDn+E	78.7

Table 5: Evaluating representation improvement. "w/o reweighting" denotes removing the last reweighting technique, and "fixed VQA" denotes keeping VQA system's parameters fixed when training the verification system.

Table 5 reports the results of the UpDn system pretrained on VQA-X dataset. Using explanations as additional supervision helps the VQA systems build better representations for the question and answer, improving performance by 3.6%. This is because minimizing the verification loss $\mathcal{L}_{verification}$ prevent the VQA system from taking shortcuts. First, it force the VQA system to produce visual and question features whose mapping can match the explanation features by $\mathcal{L}_m$. Second, by minimizing $\mathcal{L}_r^v$ and $\mathcal{L}_r^v$, the system is forced not to only focus on question or visual priors. Finally, our full system gains 1.1% improvement due to reweighting, and achieves our best results.

6 Conclusion and Future Work

In this work, we have explored how to improve VQA performance by comparing competing explanations for each answer candidate. We present two

sets of competing explanations, generated and retrieved explanations. Our approach first helps the system learn better visual and question representations, and also reweight the original answer predictions based on the competing explanations. As a result, our VQA system avoids taking shortcuts and is able to handle difficult visual questions better on the VQA-X dataset. We also show that our approach generates better textual explanations by additionally conditioning on the retrieved explanations for similar questions. In the future, we would like to combine different sorts of explanations (e.g. both generated and retrieved ones) together to better train VQA systems.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. In *CVPR*.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic Propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and VQA. In *CVPR*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *ICCV*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *ICCV*.

Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. 2019. Murel: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1989–1998.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*.

Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. 2017. Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? In *Computer Vision and Image Understanding*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. *EMNLP*.

Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. 2017. Vqs: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: a new dataset for compositional question answering over real-world images. *arXiv preprint arXiv:1902.09506*.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0. 1: the Winning Entry to the VQA Challenge 2018. *arXiv preprint arXiv:1807.09956*.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear Attention Networks. In *NeurIPS*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*.

Qing Li, Qingyi Tao, Shafiq Joty, Jianfei Cai, and Jiebo Luo. 2018. VQA-E: Explaining, Elaborating, and Enhancing Your Answers for Visual Questions. *ECCV*.

Chin-Yew Lin. 2004. Rouge: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*.

Bei Liu, Zhicheng Huang, Zhaoyang Zeng, Zheyu Chen, and Jianlong Fu. 2019. Learning rich image region representation for visual question answering. *arXiv preprint arXiv:1910.13077*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3195–3204.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *CVPR*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.

Tingting Qiao, Jianfeng Dong, and Duanqing Xu. 2018. Exploring Human-Like Attention Supervision in Visual Question Answering. In *AAAI*.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. 2018. Overcoming Language Priors in Visual Question Answering with Adversarial Regularization. In *NeurIPS*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *NIPS*.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. 2017. Right for the Right Reasons: Training Differentiable Models by Constraining Their Explanations. In *IJCAI*.

Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Dhruv Batra, and Devi Parikh. 2019. Taking a HINT: Leveraging Explanations to Make Vision and Language Models More Grounded. In *ICCV*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-Based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575.

Jialin Wu and Raymond J Mooney. 2019a. Faithful Multimodal Explanation for Visual Question Answering. In *ACL BlackboxNLP Workshop*.

Jialin Wu and Raymond J Mooney. 2019b. Self-Critical Reasoning for Robust Visual Question Answering. *arXiv preprint arXiv:1905.09998*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. 2019. Interpretable Visual Question Answering by Visual Grounding from Attention Supervision Mining. In *WACV*.