Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering

Aishwarya Agrawal[1]*, Dhruv Batra[1,2], Devi Parikh[1,2], Aniruddha Kembhavi[3]
[1]Georgia Institute of Technology, [2]Facebook AI Research, [3]Allen Institute for Artificial Intelligence
{aishwarya, dbatra, parikh}@gatech.edu, anik@allenai.org

Abstract

A number of studies have found that today's Visual Question Answering (VQA) models are heavily driven by superficial correlations in the training data and lack sufficient image grounding. To encourage development of models geared towards the latter, we propose a new setting for VQA where for every question type, train and test sets have different prior distributions of answers. Specifically, we present new splits of the VQA v1 and VQA v2 datasets, which we call Visual Question Answering under Changing Priors (VQA-CP v1 and VQA-CP v2 respectively). First, we evaluate several existing VQA models under this new setting and show that their performance degrades significantly compared to the original VQA setting. Second, we propose a novel Grounded Visual Question Answering model (GVQA) that contains inductive biases and restrictions in the architecture specifically designed to prevent the model from 'cheating' by primarily relying on priors in the training data. Specifically, GVQA explicitly disentangles the recognition of visual concepts present in the image from the identification of plausible answer space for a given question, enabling the model to more robustly generalize across different distributions of answers. GVQA is built off an existing VQA model – Stacked Attention Networks (SAN). Our experiments demonstrate that GVQA significantly outperforms SAN on both VQA-CP v1 and VQA-CP v2 datasets. Interestingly, it also outperforms more powerful VQA models such as Multimodal Compact Bilinear Pooling (MCB) in several cases. GVQA offers strengths complementary to SAN when trained and evaluated on the original VQA v1 and VQA v2 datasets. Finally, GVQA is more transparent and interpretable than existing VQA models.

1. Introduction

Automatically answering questions about visual content is considered to be one of the highest goals of artificial intelligence. Visual Question Answering (VQA) poses a rich set of challenges spanning various domains such as computer vision, natural language processing, knowledge representation, and reasoning. In the last few years, VQA has received a lot of attention – a number of VQA datasets have been curated [5, 21, 41, 12, 25, 11, 29, 13, 39] and a variety of deep-learning models have been developed [5, 9, 38, 37, 16, 3, 34, 19, 24, 4, 31, 20, 10, 26, 14, 35, 36, 40, 30].

Figure 1: Existing VQA models, such as SAN [38], tend to largely rely on strong language priors in train sets, such as, the prior answer (*'white'*, *'no'*) given the question type (*'what color is the'*, *'is the person'*). Hence, they suffer significant performance degradation on test image-question pairs whose answers (*'black'*, *'yes'*) are not amongst the majority answers in train. We propose a novel model (GVQA), built off of SAN that explicitly grounds visual concepts in images, and consequently significantly outperforms SAN in a setting with mismatched priors between train and test.

However, a number of studies have found that despite recent progress, today's VQA models are heavily driven by superficial correlations in the training data and lack sufficient visual grounding [1, 39, 13, 17]. It seems that when faced with a difficult learning problem, models typically resort to latching onto the language priors in the training data to the point of ignoring the image – *e.g.*, overwhelmingly replying to *'how many X?'* questions with *'2'* (irrespective of X), *'what color is ... ?'* with *'white'*, *'is the ... ?'* with *'yes'*.

One reason for this emergent dissatisfactory behavior is the fundamentally problematic nature of IID train-test splits *in the presence of strong priors*. As a result, models that intrinsically memorize biases in the training data demonstrate acceptable performance on the test set. This is problematic for benchmarking progress in VQA because it becomes unclear what the source of the improvements is – if models have learned to ground concepts in images or they are driven by memorizing priors in training data.

To help disentangle these factors, we present new splits of the VQA v1 [5] and VQA v2 [13] datasets, called **Visual**

*Work partially done while interning at Allen Institute for AI.

Question Answering under Changing Priors (**VQA-CP v1** and **VQA-CP v2** respectively). These new splits are created by re-organizing the train and val splits of the respective VQA datasets in such a way that the distribution of answers per question type (*'how many'*, *'what color is'*, *etc*.) is by design *different* in the test split compared to the train split (Section 3). One important thing to note: we do not change the distribution of the underlying perceptual signals – the images – between train and test. Generalization across different domains of images (*e.g*. COCO images *vs*. web cam images) is an active research area and not the focus of this work. We change the distribution of *answers for each question type* between train and test. Our hypothesis is that it is reasonable to expect models that are answering questions for the 'right reasons' (image grounding) to recognize *'black'* color at test time even though *'white'* is the most popular answer for *'What color is the . . . ?'* questions in the train set Fig. 1.

To demonstrate the difficulty of our VQA-CP splits, we report the performance of several existing VQA models [23, 3, 38, 10] on these splits. Our key finding is that the performance of *all tested existing* models drops significantly when trained and evaluated on the new splits compared to the original splits (Section 4). This finding provides further confirmation and a novel insight to the growing evidence in literature on the behavior of VQA models [1, 39, 13, 17].

We also propose a novel **Grounded Visual Question Answering** (**GVQA**) model that contains inductive biases and restrictions in the architecture specifically designed to prevent it from 'cheating' by primarily relying on priors in the training data (Section 5). GVQA is motivated by the intuition that questions in VQA provide two key pieces of information:
(1) What should be recognized? Or what visual concepts in the image need to be reasoned about to answer the question (*e.g*., *'What color is the plate?'* requires looking at the plate in the image),
(2) What should be said? Or what is the space of plausible answers (*e.g*., *'What color . . . ?'* questions need to be answered with names of colors).

Our hypothesis is that models that do not explicitly differentiate between these two roles – which is the case for most existing models in literature – tend to confuse these two signals. They end up learning from question-answer pairs that a plausible color of a plate is white, and at test time, rely on this correlation more so than the specific plate in the image the question is about. GVQA explicitly disentangles the visual concept recognition from the answer space prediction.

GVQA is built off of an existing VQA model – Stacked Attention Networks (SAN) [38]. Our experiments demonstrate that GVQA significantly outperforms SAN on both VQA-CP v1 and VQA-CP v2 datasets (Section 6.1). Interestingly, it also outperforms more powerful VQA models such as Multimodal Compact Bilinear Pooling (MCB) [10] in several cases (Section 6.1). We also show that GVQA offers strengths complementary to SAN when trained and evaluated on the original VQA v1 and VQA v2 datasets (Section 6.3). Finally, GVQA is more transparent than existing VQA models, in that it produces interpretable intermediate outputs unlike most existing VQA models (Section 6.4).

2. Related Work

Countering Priors in VQA: In order to counter the language priors in the VQA v1 dataset, [13] balance every question by collecting complementary images for every question. Thus, for every question in the proposed VQA v2 dataset, there are two similar images with different answers to the question. By construction, language priors are significantly weaker in the VQA v2 dataset. However, the train and test distributions are still similar. So, leveraging priors from the train set will still benefit the model at test time. [39] balance the yes/no questions on abstract scenes from the VQA v1 dataset in a similar manner. More recently, [18] propose two new evaluation metrics that compensate for the skewed distribution of question types and for the skewed distribution of answers within each question type in the test set. As a remedy for machines using "shortcuts" to solve multiple-choice VQA, [7] describe several principles for automatic construction of good decoys (the incorrect candidate answers). [8] study cross-dataset adaptation for VQA. They propose an algorithm for adapting a VQA model trained on one dataset to apply to another dataset with different statistical distribution. All these works indicate that there is an increasing interest in the community to focus on models that are less driven by training priors and are more visually grounded.

Compositionality. Related to the ability to generalize across different answer distributions is the ability to generalize to novel compositions of known concepts learned during training. Compositionality has been studied in various forms in the vision community. Zero-shot object recognition using attributes is based on the idea of composing attributes to detect novel object categories [22, 15]. [6] have studied compositionality in the domain of image captioning by focusing on structured representations (subject-relation-object triplets). [17] study compositionality in the domain of VQA with synthetic images and questions, with limited vocabulary of objects and attributes. More recently, [2] propose a compositional split of the VQA v1 dataset, called C-VQA, that consists of real images and questions (asked by humans) to test the extent to which existing VQA models can answer compositionally novel questions. However, even in the C-VQA splits, the distribution of answers for each question type does not change much from train to test. Hence, models relying on priors, can still generalize to the test set.

[3, 4] have developed Neural Module Networks for VQA that consist of different modules each specialized for a par-

ticular task. These modules can be composed together based on the question structure to create a model architecture for the given question. We report the performance of this model [3] on our VQA-CP datasets and find that its performance degrades significantly from the original VQA setting to the proposed CP setting (Section 4).

Zero-shot VQA has also been explored in [33]. They study a setting for VQA where the test questions (the question string itself or the multiple choices) contain at least one unseen word. [28] propose answering questions about unknown objects (*e.g.*, *'Is the dog black and white?'* where *'dog'* is never seen in training questions or answers). These are orthogonal efforts to our work in that our focus is not in studying if unseen words/concepts can be recognized during testing. We are instead interested in studying the extent to which a model is visually grounded by evaluating its ability to generalize to a different answer distribution for each question type. In our splits, we ensure that concepts seen during test time are present during training to the extent possible.

3. VQA-CP : Dataset Creation and Analysis

The VQA-CP v1 and VQA-CP v2 splits are created such that the distribution of answers per question type (*'how many'*, *'what color is'*, *etc.*) is different in the test data compared to the training data. These splits are created by re-organizing the training and validation splits of the VQA v1 [5] and VQA v2 [13] datasets respectively [1], using the following procedure:

Question Grouping: Questions having the same question type (first few words of the question – *'What color is the'*, *'What room is'*, etc.) and the same ground truth answer are grouped together. For instance, {*'What color is the dog?'*, *'white'*} and {*'What color is the plate?'*, *'white'*} are grouped together whereas {*'What color is the dog?'*, *'black'*} is put in a different group. This grouping is done after merging the QA pairs from the VQA train and val splits. We use the question types provided in the VQA datasets.

Greedily Re-splitting: A greedy approach is used to redistribute data points (image, question, answer) to the VQA-CP train and test splits so as to maximize the coverage of the VQA-CP test concepts in the VQA-CP train split while making sure that questions with the same question type and the same ground truth answer are not repeated between test and train splits. In this procedure, we loop through all the groups created above, and in every iteration, we add the current group to the VQA-CP test split unless the group has already been assigned to the VQA-CP train split. We always maintain a set of concepts[2] belonging to the groups

[1]We can not use the test splits from VQA datasets because creation of VQA-CP splits requires access to answer annotations, which are not publicly available on the test sets.

[2]For a given group, concepts are the set of all unique words present in the question type and the ground truth answer belonging to that group.

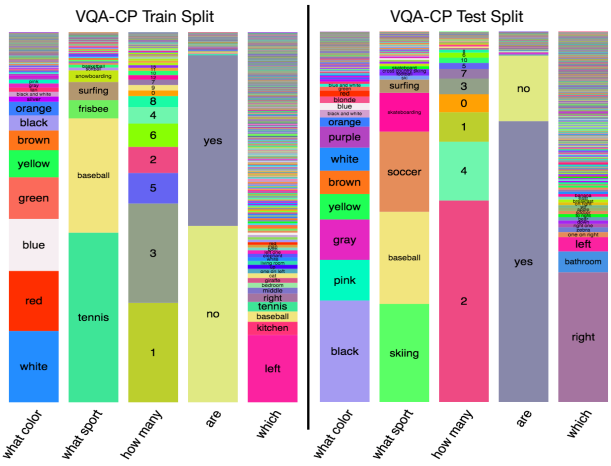

Figure 2: Distribution of answers per question type vary significantly between VQA-CP v1 train (left) and test (right) splits. For instance, *'white'* and *'red'* are commonly seen answers in train for *'What color'*, where as *'black'* is the most frequent answer in test. These have been computed for a random sample of 60K questions.

in the VQA-CP test split that have not yet been covered by the groups in the VQA-CP train split. We then pick the group that covers majority of the concepts in the set, from the groups that have not yet been assigned to either split and add that group to the VQA-CP train split. We stop when the test split has about 1/3rd the dataset and add the remaining groups (not yet assigned to either split) to the train split.

The above approach results in 98.04% coverage of test question concepts (set of all unique words in questions after removing stop words – *'is'*, *'are'*, *'the'*, *etc.*) in the train split for VQA-CP v1 (99.01% for VQA-CP v2), and 95.07% coverage of test answers by the train split's top 1000 answers for VQA-CP v1 (95.72% for VQA-CP v2). VQA-CP v1 train consists of ∼118K images, ∼245K questions and ∼2.5M answers (∼121K images, ∼438K questions and ∼4.4M answers for VQA-CP v2 train). VQA-CP v1 test consists of ∼87K images, ∼125K questions and ∼1.3M answers (∼98K images, ∼220K questions and ∼2.2M answers for VQA-CP v2 test).

Fig. 2 shows the distribution of answers for several question types such as *'what color'*, *'what sport'*, *'how many'*, etc. for the train (left) and test (right) splits of the VQA-CP v1 dataset (see supplementary material[3] for this analysis of the VQA-CP v2 dataset). We can see that the distributions of answers for a given question type is significantly different. For instance, *'tennis'* is the most frequent answer for the question type *'what sport'* in VQA-CP v1 train split whereas *'skiing'* is the most frequent answer for the same question type in VQA-CP v1 test split. However, for VQA v1 dataset, the distribution for a given question type is similar across train and val splits [5] (for instance, *'tennis'* is the most

[3]Supplementary material is available on the project website: www.cc.gatech.edu/~aagrawal307/vqa-cp/

Model	Dataset	Overall	Yes/No	Number	Other	Dataset	Overall	Yes/No	Number	Other
per Q-type prior [5]	VQA v1	35.13	71.31	31.93	08.86	VQA v2	32.06	64.42	26.95	08.76
	VQA-CP v1	08.39	14.70	08.34	02.14	VQA-CP v2	08.76	19.36	11.70	02.39
d-LSTM Q [5]	VQA v1	48.23	79.05	33.70	28.81	VQA v2	43.01	67.95	30.97	27.20
	VQA-CP v1	20.16	35.72	11.07	08.34	VQA-CP v2	15.95	35.09	11.63	07.11
d-LSTM Q + norm I [23]	VQA v1	54.40	79.82	33.87	40.54	VQA v2	51.61	73.06	34.41	39.85
	VQA-CP v1	23.51	34.53	11.40	17.42	VQA-CP v2	19.73	34.25	11.39	14.41
NMN [3]	VQA v1	54.83	80.39	33.45	41.07	VQA v2	51.62	73.38	33.23	39.93
	VQA-CP v1	29.64	38.85	11.23	27.88	VQA-CP v2	27.47	38.94	11.92	25.72
SAN [38]	VQA v1	55.86	78.54	33.46	44.51	VQA v2	52.02	68.89	34.55	43.80
	VQA-CP v1	26.88	35.34	11.34	24.70	VQA-CP v2	24.96	38.35	11.14	21.74
MCB [10]	VQA v1	60.97	81.62	34.56	52.16	VQA v2	59.71	77.91	37.47	51.76
	VQA-CP v1	34.39	37.96	11.80	39.90	VQA-CP v2	36.33	41.01	11.96	40.57

Table 1: We compare the performance of existing VQA models on VQA-CP test splits (when trained on respective VQA-CP train splits) to their performance on VQA val splits (when trained on respective VQA train splits). We find that the performance of all tested existing models degrades significantly in the new Changing Priors setting compared to the original VQA setting.

frequent answer for both the train and val splits). In the VQA-CP v1 splits, similar differences can be seen for other question types as well – *'are'*, *'which'*.

4. Benchmarking VQA Models on VQA-CP

To demonstrate the difficulty of our VQA-CP splits, we report the performance of the following baselines and existing VQA models when trained on VQA-CP v1 and VQA-CP v2 train splits and evaluated on the corresponding test splits. We compare this with their performance when trained on VQA v1 and VQA v2 train splits and evaluated on the corresponding val splits. Results are presented in Table 1.

per Q-type prior [5]: Predicting the most popular training answer for the corresponding question type (e.g., *'tennis'* for *'What sport . . . ?'* questions) [4].

Deeper LSTM Question (d-LSTM Q) [5]: Predicting the answer using question alone ("blind" model).

Deeper LSTM Question + normalized Image (d-LSTM Q + norm I) [5]: The baseline VQA model.

Neural Module Networks (NMN) [3]: The model designed to be compositional in nature.

Stacked Attention Networks (SAN) [38]: One of the widely used models for VQA.

Multimodal Compact Bilinear Pooling (MCB) [10]: The winner of the VQA Challenge (on real image) 2016.

Brief descriptions of all of these models are in the supp.

From Table 1, we can see that the performance of all tested existing VQA models drops significantly in the VQA-CP setting compared to the original VQA setting. Note that even though the NMN architecture is compositional by design, their performance degrades on the VQA-CP datasets. We posit this may be because they use an additional LSTM encoding of the question to encode priors in the dataset. Also note that the d-LSTM Q + norm I model suffers the largest drop in overall performance compared to other VQA models, perhaps because other models have more powerful visual processing (for instance, attention on images). Another interesting observation from Table 1 is that the ranking of the models based on overall performance changes from VQA to VQA-CP . For VQA , SAN outperforms NMN, whereas for VQA-CP , NMN outperforms SAN. For a brief discussion on trends for different question types, please see the supp.

[4]Note that, ideally the performance of this baseline on VQA-CP test set should be zero because the answers, given the question type, are different in test and train. But, due to some inter-human disagreement in the datasets, the performance is slightly higher (Table 1).

5. GVQA model

We now introduce our Grounded Visual Question Answering model (GVQA). While previous VQA approaches directly map Image-Question tuples (I, Q) to Answers (A), GVQA breaks down the task of VQA into two steps: **Look** - locate the object / image patch needed to answer the question and recognize the visual concepts in the patch, and **Answer** - identify the space of plausible answers from the question and return the appropriate visual concept from the set of recognized visual concepts by taking into account which concepts are plausible. For instance, when GVQA is asked *'What color is the dog?'*, it identifies that the answer should be a color name, locates the patch in the image corresponding to dog, recognizes various visual concepts such as *'black'*, *'dog'*, *'furry'*, and finally outputs the concept *'black'* because it is the recognized concept corresponding to color. Another novelty in GVQA is that it treats answering yes/no questions as a visual verification task, i.e., it verifies the visual presence/absence of the concept mentioned in the question. For instance, when GVQA is asked *'Is the person wearing*

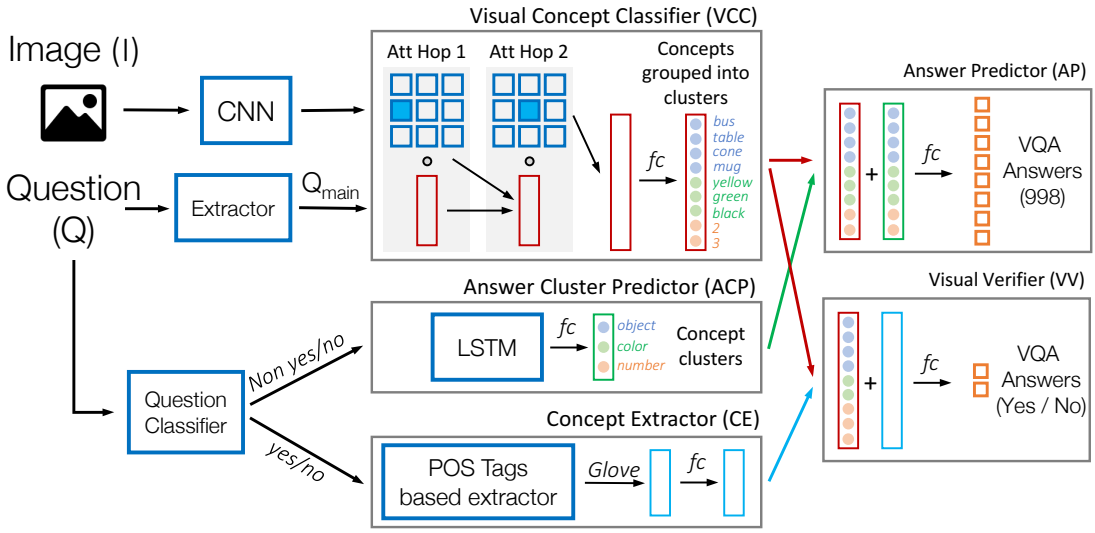

Figure 3: The proposed Grounded Visual Question Answering (GVQA) model.

shorts?', it identifies that the concept whose visual presence needs to be verified is *'shorts'* and answers *'yes'* or *'no'* depending on whether it recognizes shorts or not in the image (specifically, on the patch corresponding to *'person'*).

GVQA is depicted in Figure 3. Given a question and an image, the question first goes through the *Question Classifier* and gets classified into yes/no or non yes/no. For non yes/no questions, the GVQA components that get activated are – 1) *Visual Concept Classifier (VCC)* which takes as input the image features extracted from *CNN* and Q_{main} given by the question *Extractor*, 2) *Answer Cluster Predictor (ACP)* whose input is the entire question. The outputs of *VCC* and *ACP* are fed to the *Answer Predictor (AP)* which produces the answer. For yes/no questions, the GVQA components that get activated are – 1) *VCC* (similarly to non yes/no), 2) *Concept Extractor (CE)* whose input is the entire question. The outputs of *VCC* and *CE* are fed to the *Visual Verifier (VV)* which predicts *'yes'* or *'no'*. We present the details of each component below.

Visual Concept Classifier (VCC) is responsible for locating the image patch that is needed to answer the question, as well as producing a set of visual concepts relevant to the located patch. E.g., given *'What is the color of the bus next to the car?'*, the VCC is responsible for attending on the bus region and then outputting a set of concepts such as *'bus'* and attributes such as its color, count, etc. It consists of a 2-hop attention module based off of Stacked Attention Networks ([38]) followed by a stack of binary concept classifiers. The image is fed to the attention module in the form of activations of the last pooling layer of VGG-Net [32]. To prevent the memorization of answer priors per question type, the question is first passed through a language *Extractor*, a simple rule that outputs the string (called Q_{main}) after removing the question type substring (eg. *'What kind of'*). Q_{main} is embedded using an LSTM and then fed into the attention module. The multi hop attention produces a weighted linear combination of the image region features from VGG-Net, with weights corresponding to the degree of attention for that region. This is followed by a set of fully connected (FC) layers and a stack of ∼2000 binary concept classifiers that cover ∼95% of the concepts seen in train. VCC is trained with a binary logistic loss for every concept.

The set of VCC concepts is constructed by extracting objects and attributes, pertinent to the answer, from training QA pairs and retaining the most frequent ones. Object concepts are then grouped into a single group where as attribute concepts are clustered into multiple small groups using K-means clustering in Glove embedding space [27], for a total of C clusters.[5] Concept clustering is required for the purpose of generating negative samples required to train the concept classifiers (for a concept classifier, positive samples are those which contain that concept either in the question or the answer). Since the question does not indicate objects and attributes absent in the image, negative data is generated using the following assumptions: (1) the attended image patch required to answer a question has at most one dominant object in it (2) every object has at most one dominant attribute from each attribute category (e.g., if the color of a bus is red, it can be used as a negative example for all other colors). Given these assumptions, when a concept in a cluster is treated as positive, all other concepts in that cluster are treated as negatives. Note that only a subset of all concept clusters are activated for each question during training, and only these activated clusters contribute to the loss.

[5] We use $C = 50$ because it gives better clusters than other values. Also, agglomerative clustering results in similar performance as K-means. More details in the supplementary material.

Question Classifier classifies the input question Q into 2 categories: Yes-No and non Yes-No using a Glove embedding layer, an LSTM and FC layers. Yes-No questions feed into the CE and the rest feed into the ACP.

Answer Cluster Predictor (ACP) identifies the *type* of the expected answer (*e.g.* object name, color, number, *etc.*). It is only activated for non yes/no questions. It consists of a Glove embedding layer and an LSTM, followed by FC layers that classify questions into one of the C clusters. The clusters for ACP are created by K-means clustering on (1000) answer classes by embedding each answer in Glove space.[6]

Concept Extractor (CE) extracts question concepts from yes/no questions whose visual presence needs to be verified in the image, using a POS tag based extraction system[7]. E.g., for *'Is the cone green?'*, we extract *'green'*. The extracted concept is embedded in Glove space followed by FC layers to transform this embedding to the same space as the VCC concepts so that they can be combined by VV. Please see the description of VV below.

Answer Predictor (AP): Given a set of visual concepts predicted by the VCC, and a concept category predicted by the ACP, the AP's role is to predict the answer. ACP categories correspond to VCC concept clusters (see ACP's and VCC's output classes in Fig. 3. The colors denote the correspondence). Given this alignment, the output of the ACP can be easily mapped into a vector with the same dimensions as the VCC output by simply copying ACP dimensions into positions pertaining to the respective VCC cluster dimensions. The resulting ACP embedding is added element-wise to the VCC embedding followed by FC layers and a softmax activation, yielding a distribution over 998 VQA answer categories (top 1000 training answers minus *'yes'* and *'no'*).

Visual Verifier (VV): Given a set of visual concepts predicted by the VCC and the embedding of the concept whose visual presence needs to be verified (given by CE), the VV's role is to verify the presence/absence of the concept in VCC's predictions. Specifically, the CE embedding is added element-wise to the VCC embedding followed by FC layers and a softmax activation, yielding a distribution over two categories – *'yes'* and *'no'*.

Model Training and Testing: We first train VCC and ACP on the train split using the cluster labels (for ACP) and visual concept labels (for VCC)[8]. The inputs to Answer Predictor (and Visual Verifier) are the predictions from VCC and ACP (CE in the case of yes/no questions) on the training data. During training, we use ground truth labels for yes/no and non yes/no questions for the Question Classifier. During testing, we first run the Question Classifier to classify questions into yes/no and non yes/no. And feed the questions into their respective modules to obtain predictions on the test set. Please refer to the supp. for implementation details.

[6]We first create the clusters for ACP using the answer classes. We then create the clusters for VCC by assigning each VCC concept to one of these ACP clusters using Euclidean distance in Glove embedding space.

[7]We use NLTK POS tagger. Spacy POS tagger results in similar performance. More details in the supplementary material.

[8]Note that we do not need additional image labels to train VCC, our labels are extracted automatically from the QA pairs. Same for ACP.

Dataset	Model	Overall	Yes/No	Number	Other
VQA-CP v1	SAN [38]	26.88	35.34	11.34	24.70
	GVQA (Ours)	**39.23**	**64.72**	**11.87**	**24.86**
VQA-CP v2	SAN [38]	24.96	38.35	11.14	21.74
	GVQA (Ours)	**31.30**	**57.99**	**13.68**	**22.14**

Table 2: Performance of GVQA (our model) compared to SAN on VQA-CP datasets. GVQA consistently outperforms SAN.

6. Experimental Results

6.1. Experiments on VQA-CP v1 and VQA-CP v2

Model accuracies: Table 2 shows the performance of our GVQA model in comparison to SAN (the model which GVQA is built off of) on VQA-CP v1 and VQA-CP v2 datasets using the VQA evaluation metric [5]. Accuracies are presented broken down into Yes/No, Number and Other categories. As it can be seen from Table 2, the proposed architectural improvements (in GVQA) over SAN show a significant boost in the overall performance for both the VQA-CP v1 (12.35%) and VQA-CP v2 (6.34%) datasets. It is worth noting that owing to the modular nature of the GVQA architecture, one may easily swap in other attention modules into the VCC. Interestingly, on the VQA-CP v1 dataset, GVQA also outperforms MCB [10] and NMN [3] (Table 1) on the overall metric (mainly for yes/no questions), in spite of being built off of a relatively simpler attention module from SAN, and using relatively less powerful image features (VGG-16) as compared to ResNet-152 being used in MCB. On the VQA-CP v2 dataset, GVQA outperforms NMN in overall metric (as well as for number questions) and MCB for yes/no and number questions.

To check if our particular VQA-CP split was causing some irregularities in performance, we created four sets of VQA-CP v2 splits with different random seeds. This also led to a large portion of the dataset (84%) being covered across the test splits. The results show that GVQA consistently outperforms SAN across all four splits with average improvement being 7.14% (standard error: 1.36). Please see supp. for performance on each split.

Performance of Model Components *Question Classifier*: On the VQA-CP v1 test set, the LSTM based question classifier obtains 99.84% accuracy. *ACP*: The Top-1 test accuracy is 54.06%, with 84.25% for questions whose answers are in attribute clusters and 43.17% for questions whose answers are in object clusters. The Top-3 accuracy rises to 65.33%. Note that these accuracies are computed using the automatically created clusters. *VCC*: The weighted mean test F1 score across all classifiers is 0.53. The individual concepts

are weighted as per the number of positive samples, reflecting the coverage of that concept in the test set. Please refer to the supp. for accuracies on the VQA-CP v2 dataset.

6.2. Role of GVQA Components

In order to evaluate the role of various GVQA components, we report the experimental results (on VQA-CP v1) by replacing each component in GVQA (denoted by "- <component>") with its traditional counterpart, i.e., modules used in traditional VQA models (denoted by " + <traditional counterpart>"). For instance, GVQA - CE + LSTM represents a model where CE in GVQA has been replaced with an LSTM. The results are presented in Table 3 along with the result of the full GVQA model for reference.

GVQA - Q_{main} + Q_{full}: GVQA's performance when the entire question (Q_{full}) is fed into VCC (as opposed to after removing the question type (Q_{main})) is 33.55% (overall), which is 5.68% (absolute) less than that with Q_{main}. Note that even with feeding the entire question, GVQA outperforms SAN, thus demonstrating that removing question type information helps but isn't the main factor behind the better performance of GVQA. As an additional check, we trained a version of SAN where the input is Q_{main} instead of Q_{full}. Results on VQA-CP v2 show that this version of SAN performs 1.36% better than the original SAN, however still 4.98% worse than GVQA (with Q_{main}). Please see supp. for detailed performance of this version of SAN.

GVQA - CE + LSTM: We replace CE with an LSTM (which is trained end-to-end with the Visual Verifier (VV) using VQA loss). The overall performance drops by 11.95%, with a drop of 28.76% for yes/no questions. This is an expected result, given that Table 2 shows that GVQA significantly outperforms SAN on yes/no questions and the CE is a crucial component of the yes/no pipeline.

GVQA - ACP + LSTM: We replace ACP with an LSTM (which is trained end-to-end with the Answer Predictor (AP) using VQA loss). The overall performance is similar to GVQA. But, the presence of ACP makes GVQA transparent and interpretable (see Section 6.4).

GVQA - VCC$_{loss}$: We remove the VCC loss and treat the output layer of VCC as an intermediate layer whose activations are passed to the Answer Predictor (AP) and trained end-to-end with AP using VQA loss. The overall performance improves by 1.72% with biggest improvement in the performance on other questions (3.19%). This suggests that introducing the visual concept (semantic) loss in between the model pipeline hurts. Although removing VCC loss and training end-to-end with VQA loss achieves better performance, the model is no longer transparent (see Section 6.4). Using VCC loss or not is a design choice one would make based on the desired accuracy vs. interpretability trade off.

GVQA - VCC$_{loss}$ - ACP + LSTM: Replacing ACP with an LSTM on top of **GVQA - VCC$_{loss}$** hurts the overall performance by 2.09% with biggest drop (4.94%) for "other" questions (see **GVQA - VCC$_{loss}$** and **GVQA - VCC$_{loss}$ - ACP + LSTM** rows in Table 3). This suggests that ACP helps significantly (as compared to an LSTM) in the absence of VCC loss (and it performs similar to an LSTM in the presence of VCC loss, as seen above). In addition, ACP adds interpretability to GVQA.

Model	Overall	Yes/No	Number	Other
GVQA - Q_{main} + Q_{full}	33.55	51.64	11.51	24.43
GVQA - CE + LSTM	27.28	35.96	11.88	24.85
GVQA - ACP + LSTM	39.40	64.72	11.73	25.33
GVQA - VCC$_{loss}$	40.95	65.50	12.32	28.05
GVQA - VCC$_{loss}$ - ACP + LSTM	38.86	65.73	11.58	23.11
GVQA	39.23	64.72	11.87	24.86

Table 3: Experimental results when each component in GVQA (denoted by "- <component>") is replaced with its corresponding traditional counterpart (denoted by " + <traditional counterpart>").

	VQA v1		VQA v2	
Model	Overall	Yes/No	Overall	Yes/No
SAN	55.86	78.54	52.02	68.89
GVQA	51.12	76.90	48.24	72.03
Ensemble (SAN, SAN)	56.56	79.03	52.45	69.17
Ensemble (GVQA, SAN)	56.91	80.42	52.96	72.72
Oracle (SAN, SAN)	60.85	83.92	56.68	74.37
Oracle (GVQA, SAN)	63.77	88.98	61.96	85.65

Table 4: Results of GVQA and SAN on VQA v1 and VQA v2 when trained on the corresponding train splits.

6.3. Experiments on VQA v1 and VQA v2

We also trained and evaluated GVQA on train and val splits of the VQA v1 [5] and VQA v2 [13] datasets (results in Table 4[9]). On VQA v1, GVQA achieves 51.12% overall accuracy, which is 4.74% (absolute) less than SAN. This gap is not surprising because VQA v1 has well-established heavy language priors that existing models (including SAN) can "memorize" from train set and exploit on the test set (since test set contains same priors as train set), whereas GVQA is designed not to. As vision improves, grounded models like GVQA may show improved performance over models that leverage priors from training data. Moreover, it is important to note that the gain (GVQA acc - SAN acc) on VQA-CP v1 (12.35% absolute) is much higher than the loss (SAN acc - GVQA acc) on VQA v1 (4.74% absolute).

On VQA v2, GVQA under performs SAN by 3.78% overall, which is less than SAN acc - GVQA acc on VQA v1. And it outperforms SAN by 3.14% for yes/no questions.

[9]We present overall and yes/no accuracies only. Please refer to the supp. for performance on number and other categories.

This shows that when the priors are weaker (in VQA v2 compared to those in VQA v1), the gap between GVQA and SAN's performance decreases. We also trained and evaluated GVQA- VCC_{loss} on both the VQA v1 and VQA v2 datasets and found that it performs worse than GVQA on VQA v1 and similar to GVQA on VQA v2. So in addition to interpretability, GVQA is overall better than GVQA-VCC_{loss} on these original VQA splits.

In order to check whether GVQA has strengths complementary to SAN, we computed the oracle of SAN's and GVQA's performance – **Oracle (GVQA, SAN)**, i.e., we pick the predictions of the model with higher accuracy for each test instance. As it can be seen from Table 4, the Oracle (GVQA, SAN)'s overall performance is 7.91% higher than that of SAN for VQA v1 (9.94% for VQA v2) suggesting that GVQA and SAN have complementary strengths. Also, note that Oracle (GVQA, SAN) is higher than Oracle (SAN, SAN) for both VQA v1 and VQA v2, suggesting that GVQA's complementary strengths are more than that of another SAN model (with a different random initialization).

Inspired by this, we report the performance of the ensemble of GVQA and SAN **Ensemble (GVQA, SAN)** in Table 4, where the ensemble combines the outputs from the two models using product of confidences of each model. We can see that Ensemble (GVQA, SAN) outperforms Ensemble (SAN, SAN) by 0.35% overall for VQA v1 (and by 0.51% for VQA v2). It is especially better for yes/no questions. We also found that the ensemble of GVQA- VCC_{loss} with SAN performs worse than Ensemble (SAN, SAN) for both the VQA datasets (refer to supp. for accuracies). Hence, GVQA is a better complement of SAN than GVQA- VCC_{loss}, in addition to being more transparent.

6.4. Transparency

The architecture design of GVQA makes it more transparent than existing VQA models because it produces interpretable intermediate outputs (the outputs of VCC, ACP and the concept string extracted by the CE) unlike most existing VQA models. We show some example predictions from GVQA in Fig. 4. We can see that the intermediate outputs provide insights into why GVQA is predicting what it is predicting and hence enable a system designer to identify the causes of error. This is not easy to do in existing VQA models. Fig. 5 shows two other examples (one success and one failure) comparing and contrasting how GVQA's intermediate outputs can help explain successes and failures (and thus, enabling targeted improvements) which is not possible to do for SAN and most other existing VQA models. See the supplementary material for more such examples.

7. Conclusion

GVQA is a first step towards building models which are visually grounded by design. Future work involves developing models that can utilize the best of both worlds (visual grounding and priors), such as, answering a question based on the knowledge about the priors of the world (sky is usually blue, grass is usually green) when the model's confidence in the answer predicted as result of visual grounding is low.

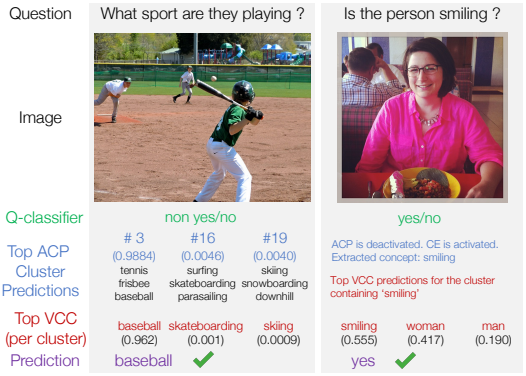

Figure 4: Qualitative examples from GVQA. **Left:** We show top three answer cluster predictions (along with random concepts from each cluster) by ACP. Corresponding to each cluster predicted by ACP, we show the top visual concept predicted by VCC. Given these ACP and VCC predictions, the Answer Predictor (AP) predicts the correct answer *'baseball'*. **Right:** Smiling is the concept extracted by the CE whose visual presence in VCC's predictions is verified by the Visual Verifier, resulting in *'yes'* as the final answer.

Figure 5: **Left**: GVQA's prediction (*'green'*) can be explained as follows – ACP predicts that the answer should be a *color*. Of the various visual concepts predicted by VCC, the only concept that is about color is *green*. Hence, GVQA's output is *'green'*. SAN incorrectly predicts *'yellow'*. SAN's architecture doesn't facilitate producing an explanation of why it predicted what it predicted, unlike GVQA. **Right**: Both GVQA and SAN incorrectly answer the question. GVQA is incorrect perhaps because VCC predicts *'black'*, instead of *'gray'*. In order to dig further into why VCC's prediction is incorrect, we can look at the attention map (in the supp.), which shows that the attention is on the pants for the right leg, but on the socks (black in color) for the left leg. So, perhaps, VCC's "black" prediction is based on the attention on the left leg.

Acknowledgements. We thank Yash Goyal for useful discussions. This work was funded in part by: NSF CAREER awards, ONR YIP awards, Google FRAs, Amazon ARAs, DARPA XAI, and ONR Grants N00014-14-1-{0679, 2713} to DB, DP, and PGA Family Foundation award to DP.

References

[1] A. Agrawal, D. Batra, and D. Parikh. Analyzing the behavior of visual question answering models. In *EMNLP*, 2016. 1, 2

[2] A. Agrawal, A. Kembhavi, D. Batra, and D. Parikh. C-vqa: A compositional split of the visual question answering (vqa) v1. 0 dataset. *arXiv preprint arXiv:1704.08243*, 2017. 2

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Deep compositional question answering with neural module networks. In *CVPR*, 2016. 1, 2, 3, 4, 6

[4] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *NAACL*, 2016. 1, 2

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015. 1, 3, 4, 6, 7

[6] Y. Atzmon, J. Berant, V. Kezami, A. Globerson, and G. Chechik. Learning to generalize to new compositions in image understanding. *arXiv preprint arXiv:1608.07639*, 2016. 2

[7] W. Chao, H. Hu, and F. Sha. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *NAACL*, 2018. 2

[8] W. Chao, H. Hu, and F. Sha. Cross-dataset adaptation for visual question answering. In *CVPR*, 2018. 2

[9] K. Chen, J. Wang, L. Chen, H. Gao, W. Xu, and R. Nevatia. ABC-CNN: an attention based convolutional neural network for visual question answering. *CoRR*, abs/1511.05960, 2015. 1

[10] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 1, 2, 4, 6

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

[12] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015. 1

[13] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 1, 2, 3, 7

[14] I. Ilievski, S. Yan, and J. Feng. A focused dynamic attention model for visual question answering. *CoRR*, abs/1604.01485, 2016. 1

[15] D. Jayaraman, F. Sha, and K. Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1629–1636, 2014. 2

[16] A. Jiang, F. Wang, F. Porikli, and Y. Li. Compositional memory for visual question answering. *CoRR*, abs/1511.05676, 2015. 1

[17] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 1, 2

[18] K. Kafle and K. Christopher. An analysis of visual question answering algorithms. In *ICCV*, 2017. 2

[19] K. Kafle and C. Kanan. Answer-type prediction for visual question answering. In *CVPR*, 2016. 1

[20] J.-H. Kim, S.-W. Lee, D.-H. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual QA. In *NIPS*, 2016. 1

[21] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 1

[22] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009. 2

[23] J. Lu, X. Lin, D. Batra, and D. Parikh. Deeper lstm and normalized cnn visual question answering model. https://github.com/VT-vision-lab/VQA_LSTM_CNN, 2015. 2, 4

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

[25] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems*, pages 1682–1690, 2014. 1

[26] H. Noh and B. Han. Training recurrent answering units with joint loss minimization for vqa. *CoRR*, abs/1606.03647, 2016. 1

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

[28] S. K. Ramakrishnan, A. Pal, G. Sharma, and A. Mittal. An empirical evaluation of visual question answering for novel objects. *arXiv preprint arXiv:1704.02516*, 2017. 3

[29] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems*, pages 2953–2961, 2015. 1

[30] K. Saito, A. Shin, Y. Ushiku, and T. Harada. Dualnet: Domain-invariant network for visual question answering. *CoRR*, abs/1606.06108, 2016. 1

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

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 5

[33] D. Teney and A. v. d. Hengel. Zero-shot visual question answering. *arXiv preprint arXiv:1611.05546*, 2016. 3

[34] P. Wang, Q. Wu, C. Shen, A. van den Hengel, and A. R. Dick. Explicit knowledge-based reasoning for visual question answering. *CoRR*, abs/1511.02570, 2015. 1

[35] Q. Wu, P. Wang, C. Shen, A. van den Hengel, and A. R. Dick. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *CVPR*, 2016. 1

[36] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016. 1

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

[38] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 2, 4, 5, 6

[39] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh. Yin and Yang: Balancing and answering binary visual questions. In *CVPR*, 2016. 1, 2

[40] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Proceedings of ACM Multimedia Systems*, 2003. 1

[41] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004, 2016. 1