Counterfactual Samples Synthesizing for Robust Visual Question Answering

Long Chen[1*]   Xin Yan[1*]   Jun Xiao[1†]   Hanwang Zhang[2]   Shiliang Pu[3]   Yueting Zhuang[1]

[1]DCD Lab, College of Computer Science, Zhejiang University

[2]MReaL Lab, Nanyang Technological University   [3]Hikvision Research Institute

Abstract

Despite Visual Question Answering (VQA) has realized impressive progress over the last few years, today's VQA models tend to capture superficial linguistic correlations in the train set and fail to generalize to the test set with different QA distributions. To reduce the language biases, several recent works introduce an auxiliary question-only model to regularize the training of targeted VQA model, and achieve dominating performance on VQA-CP. However, since the complexity of design, current methods are unable to equip the ensemble-based models with two indispensable characteristics of an ideal VQA model: 1) visual-explainable: the model should rely on the right visual regions when making decisions. 2) question-sensitive: the model should be sensitive to the linguistic variations in question. To this end, we propose a model-agnostic Counterfactual Samples Synthesizing (CSS) training scheme. The CSS generates numerous counterfactual training samples by masking critical objects in images or words in questions, and assigning different ground-truth answers. After training with the complementary samples (i.e., the original and generated samples), the VQA models are forced to focus on all critical objects and words, which significantly improves both visual-explainable and question-sensitive abilities. In return, the performance of these models is further boosted. Extensive ablations have shown the effectiveness of CSS. Particularly, by building on top of the model LMH [14], we achieve a record-breaking performance of 58.95% on VQA-CP v2, with 6.5% gains.[1]

1. Introduction

Visual Question Answering (VQA), *i.e.*, answering natural language questions about the visual content, is one of the core techniques towards complete AI. With the release of multiple large scale VQA datasets (*e.g.*, VQA v1 [6] and v2 [17]), VQA has received unprecedented attention and

---

[*]Long Chen and Xin Yan are co-first authors with equal contributions.
[†]Corresponding author.
[1]Codes: https://github.com/yanxinzju/CSS-VQA

(a) Visual-Explainable Ability
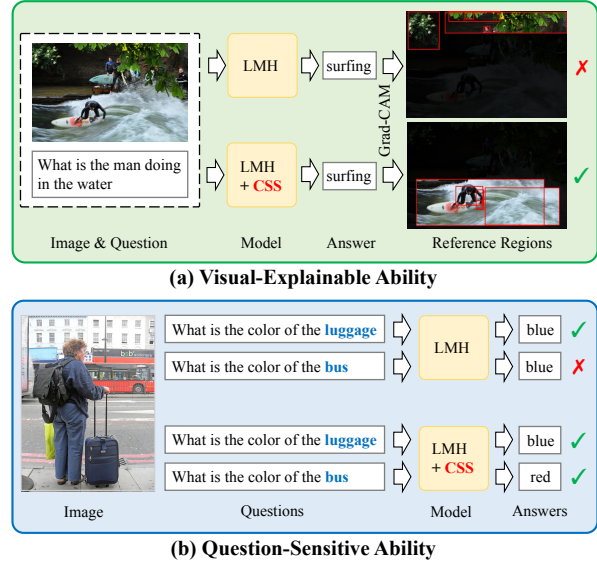

(b) Question-Sensitive Ability

Figure 1: The two indispensable characteristics of an ideal VQA model. (a) **visual-explainable ability**: the model not only needs to predict correct answer (*e.g.*, "surfing"), but also relies on the right reference regions when making this prediction. (b) **question-sensitive ability**: the model should be sensitive to the linguistic variations, *e.g.*, after replacing the critical word "luggage" with "bus", the predicted answers of two questions should be different.

hundreds of models have been developed. However, since the inevitable annotation artifacts in the real image datasets, today's VQA models always over-rely on superficial linguistic correlations (*i.e.*, language biases) [2, 42, 23, 17]. For example, a model answering "2" for all "*how many X*" questions can still get satisfactory performance regardless of the *X*. Recently, to disentangle the bias factors and clearly monitor the progress of VQA research, a diagnostic benchmark VQA-CP (VQA under Changing Priors) [3] has been proposed. The VQA-CP deliberately has different question-answer distributions in the train and test splits. The performance of many state-of-the-art VQA models [5, 15, 40, 4] drop significantly on VQA-CP compared to other datasets.

Currently, the prevailing solutions to mitigate the bias issues are **ensemble-based** methods: they introduce an aux-

Figure 2: (a): A training sample from the VQA-CP. (b): The synthesized training sample by V-CSS. It masks ciritcal objects (*e.g.*, "tie") in image and assigns different ground-truth answers ("not green"). (c): The synthesized training sample by Q-CSS. It replaces critical words (*e.g.*, "tie") with special token "[MASK]" in question and assigns different ground-truth answers ("not green").

iliary question-only model to regularize the training of targeted VQA model. Specifically, these methods can further be grouped into two sub-types: 1) *adversary-based* [33, 18, 7]: they train two models in an adversarial manner [16, 12], *i.e.*, minimizing the loss of VQA model while maximizing the loss of question-only model. Since the two models are designed to share the same question encoder, the adversary-based methods aim to reduce the language biases by learning a bias-neutral question representation. Unfortunately, the adversarial training scheme brings significant noise into gradients and results in an unstable training process [18]. 2) *fusion-based* [10, 14, 27]: they late fuse the predicted answer distributions of the two models, and derive the training gradients based on the fused answer distributions. The design philosophy of the fusion-based methods, is to let the targeted VQA model focuses more on the samples, which cannot be answered correctly by the question-only model.

Although the ensemble-based methods have dominated the performance on VQA-CP, it is worth noting that current methods fail to equip them with two indispensable characteristics of an ideal VQA model: 1) **visual-explainable**: the model should rely on the right visual regions when making decisions, *i.e.*, right for the right reasons [34]. As shown in Figure 1 (a), although both two models can predict the correct answer "surfing", they actually refer to totally different reference regions when making this answer prediction. 2) **question-sensitive**: the model should be sensitive to the linguistic variations in question. As shown in Figure 1 (b), for two questions with similar sentence structure (*e.g.*, only replacing word "luggage" with "bus"), if the meanings of two questions are different, the model should perceive the discrepancy and make corresponding predictions.

In this paper, we propose a novel model-agnostic Coun-

terfactual Samples Synthesizing (CSS) training scheme. The CSS serves as a plug-and-play component to improve the VQA models' visual-explainable and question-sensitive abilities, even for complex ensemble-based methods. As shown in Figure 2, CSS consists of two different types of samples synthesizing mechanisms: V-CSS and Q-CSS. For V-CSS, it synthesizes a counterfactual image by masking critical objects in the original image. By "critical", we mean that these objects are important in answering a certain question (*e.g.*, object ▮ for the question "what color is the man's tie"). Then, the counterfactual image and original question compose a new image-question (VQ) pair. For Q-CSS, it synthesizes a counterfactual question by replacing critical words in the original question with a special token "[MASK]". Similarly, the counterfactual querstion and original image compose a new VQ pair. Given a VQ pair (from V-CSS or Q-CSS), a standard VQA training sample triplet still needs the corresponding ground-truth answers. To avoid the expensive manual annotations, we design a dynamic answer assigning mechanism to approximate ground-truth answers for all synthesized VQ pairs (*e.g.*, "not green" in Figure 2). Then, we train the VQA models with all original and synthesized samples. After training with numerous complementary samples, the VQA models are forced to focus on critical objects and words.

Extensive ablations including both qualitative and quantitative results have demonstrated the effectiveness of CSS. The CSS can be seamlessly incorporated into the ensemble-based methods, which not only improves their both visual-explainable and question-sensitive abilities, but also consistently boosts the performance on VQA-CP. Particularly, by building of top on model LMH [14], we achieve a new record-breaking performance of 58.95% on VQA-CP v2.

2. Related Work

Language Biases in VQA. Despite VQA is a multi-modal task, a large body of research [21, 2, 42, 17] has shown the existence of language biases in VQA. There are two main solutions to reduce the language biases:
1. Balancing Datasets to Reduce Biases. The most straight-forward solution is to create more balanced datasets. For example, Zhang *et al.* [42] collected complementary abstract scenes with opposite answers for all binary questions. And Goyal *et al.* [17] extended this idea into real images and all types of questions. Although these "balanced" datasets have reduced biases to some extent, the statistical biases from questions still can be leveraged [3]. As shown in the benchmark VQA-CP, the performance of numerous models drop significantly compared to these "balanced" datasets. In this paper, we follow the same spirit of dataset balancing and train VQA models with more complementary samples. Especially, CSS doesn't need any extra manual annotations.
2. Designing Models to Reduce Biases. Another solution is

to design specific debiasing models. So far, the most effective debiasing models for VQA are ensemble-based methods [33, 18, 7, 10, 14, 27]. In this paper, we propose a novel CSS training scheme, which can be seamlessly incorporated into the ensemble-based models to further reduce the biases.

Visual-Explainable Ability in VQA Models. To improve visual-explainable ability, early works [32, 26, 43] directly apply human attention as supervision to guide the models' attention maps. However, since the existence of strong biases, even with appropriate attention maps, the remaining layers of network may still disregard the visual signal [36]. Thus, some recent works [36, 39] utilize Grad-CAM [35] to obtain private contribution of each object to correct answers, and encourage the rank of all object contributions to be consistent with human annotations. Unfortunately, these models have two drawbacks: 1) They need extra human annotations. 2) The training is not end-to-end.

Question-Sensitive Ability in VQA Models. If VQA systems really "understand" the question, they should be sensitive to the linguistic variations in question. Surprisingly, to the best of our knowledge, there is only one work [37] has studied the influence of linguistic variations in VQA. Specifically, it designs a cycle-consistent loss between two dual tasks, and utilizes sampled noises to generate diverse questions. However, Shah *et al.* [37] only considers the robustness to different rephrasings of questions. In contrast, we also encourage the model to perceive the difference of questions when changing some critical words.

Counterfactual Training Samples for VQA. Some concurrent works [1, 30] also try to synthesize counterfactual samples for VQA. Different from these works that all resort to GAN [16] to generate images, CSS only mask critical objects or words, which is easier and more adoptable.

3. Approach

We consider the common formulation of VQA task as a multi-class classification problem. Without loss of generality, given a dataset $\mathcal{D} = \{I_i, Q_i, a_i\}_i^N$ consisting of triplets of images $I_i \in \mathcal{I}$, questions $Q_i \in \mathcal{Q}$ and answers $a_i \in \mathcal{A}$, VQA task learns a mapping $f_{vqa} : \mathcal{I} \times \mathcal{Q} \to [0, 1]^{|\mathcal{A}|}$, which produces an answer distribution given image-question pair. For simplicity, we omit subscript i in the following sections.

In this section, we first introduce the base bottom-up top-down model [4], and the ensemble-based methods for debiasing in Section 3.1. Then, we introduce the details of the Counterfactual Samples Synthesizing (CSS) in Section 3.2.

3.1. Preliminaries

Bottom-Up Top-Down (UpDn) Model. For each image I, the UpDn uses an image encoder e_v to output a set of object features: $\boldsymbol{V} = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_{n_v}\}$, where $\boldsymbol{v}_i$ is i-th object feature. For each question Q, the UpDn uses a question encoder e_q to output a set of word features: $\boldsymbol{Q} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_{n_q}\}$, where

---

Algorithm 1 Ensemble-based Model (fusion-based)

1: **function** $\mathcal{VQA}(I, Q, a, cond)$
2:     $\boldsymbol{V} \leftarrow e_v(I)$
3:     $\boldsymbol{Q} \leftarrow e_q(Q)$
4:     $P_{vqa}(\boldsymbol{a}) \leftarrow f_{vqa}(\boldsymbol{V}, \boldsymbol{Q})$
5:     $P_q(\boldsymbol{a}) \leftarrow f_q(\boldsymbol{Q})$ ▷ question-only model
6:     $\hat{P}_{vqa}(\boldsymbol{a}) \leftarrow M(P_{vqa}(\boldsymbol{a}), P_q(\boldsymbol{a}))$
7:     $Loss \leftarrow \text{XE}(\hat{P}_{vqa}(\boldsymbol{a}), a)$ ▷ update parameters
8:     **if** $cond$ **then**
9:         **return** $\boldsymbol{V}, \boldsymbol{Q}, P_{vqa}(\boldsymbol{a})$
10:     **end if**
11: **end function**

$\boldsymbol{w}_j$ is j-th word feature. Then both $\boldsymbol{V}$ and $\boldsymbol{Q}$ are fed into the model f_{vqa} to predict answer distributions:

$$P_{vqa}(\boldsymbol{a}|I, Q) = f_{vqa}(\boldsymbol{V}, \boldsymbol{Q}). \qquad (1)$$

Model f_{vqa} typically contains an attention mechanism [13, 29, 41], and it is trained with cross-entropy loss [38, 11].

Ensemble-Based Models. As we discussed in Section 1, the ensemble-based models can be grouped into two sub-types: adversary-based and fusion-based. Since adversary-based models [33, 18, 7] suffer severe unstable training and relatively worse performance, in this section, we only introduce the fusion-based models [10, 14, 27]. As shown in Algorithm 1, they introduce an auxiliary question-only model f_q which takes $\boldsymbol{Q}$ as input and predicts answer distribution:

$$P_q(\boldsymbol{a}|Q) = f_q(\boldsymbol{Q}). \qquad (2)$$

Then, they combine the two answer distributions and obtain a new answer distribution $\hat{P}_{vqa}(\boldsymbol{a})$ by a function M:

$$\hat{P}_{vqa}(\boldsymbol{a}|I, Q) = M(P_{vqa}(\boldsymbol{a}|I, Q), P_q(\boldsymbol{a}|Q)). \qquad (3)$$

In the training stage, the XE loss is computed based on the fused answer distribution $\hat{P}_{vqa}(\boldsymbol{a})$ and the training gradients are backpropagated through both f_{vqa} and f_q. In test stage, only model f_{vqa} is used as the plain VQA models.

3.2. Counterfactual Samples Synthesizing (CSS)

The overall structure of CSS training scheme is shown in Algorithm 2. Specifically, for any $\mathcal{VQA}$ model, given a training sample (I, Q, a), CSS consists of three main steps:

1. Training $\mathcal{VQA}$ model with original sample (I, Q, a);
2. Synthesizing a counterfactual sample (I^-, Q, a^-) by V-CSS or (I, Q^-, a^-) by Q-CSS;
3. Training $\mathcal{VQA}$ model with the counterfactual sample.

In the following, we introduce the details of V-CSS and Q-CSS (*i.e.*, the second step). As shown in Algorithm 2, for each training sample, we only use one certain synthesizing mechanism, and δ is the trade-off weight (See Figure 4 (c) for more details about the influence of different δ).

Algorithm 2 Counterfactual Samples Synthesizing

1: **function** $\mathcal{CSS}(I, Q, a)$
2:    $\boldsymbol{V}, \boldsymbol{Q}, P_{vqa}(\boldsymbol{a}) \leftarrow \mathcal{VQA}(I, Q, a, \text{True})$
3:    $cond \sim U[0, 1]$
4:    **if** $cond \geq \delta$ **then**   ▷ execute V-CSS
5:       $\mathcal{I} \leftarrow \text{IO\_SEL}(I, Q)$
6:       $s(a, \boldsymbol{v}_i) \leftarrow \mathcal{S}(P_{vqa}(a), \boldsymbol{v}_i)$
7:       $I^+, I^- \leftarrow \text{CO\_SEL}(\mathcal{I}, \{s(a, \boldsymbol{v}_i)\})$
8:       $a^- \leftarrow \text{DA\_ASS}(I^+, Q, \mathcal{VQA}, a)$
9:       $\mathcal{VQA}(I^-, Q, a^-, \text{False})$
10:    **else**   ▷ execute Q-CSS
11:       $s(a, \boldsymbol{w}_i) \leftarrow \mathcal{S}(P_{vqa}(a), \boldsymbol{w}_i)$
12:       $Q^+, Q^- \leftarrow \text{CW\_SEL}(\{s(a, \boldsymbol{w}_i)\})$
13:       $a^- \leftarrow \text{DA\_ASS}(I, Q^+, \mathcal{VQA}, a)$
14:       $\mathcal{VQA}(I, Q^-, a^-, \text{False})$
15:    **end if**
16: **end function**

3.2.1 V-CSS

We sequentially introduce all steps of V-CSS following its execution path (line 5 to 8 in Algorithm 2), which consists of four main steps: initial objects selection (IO_SEL), object local contributions calculation, critical objects selection (CO_SEL), and dynamic answer assigning (DA_ASS).

1. Initial Objects Selection (IO_SEL). In general, for any specific QA pair (Q, a), only a few objects in image I are related. To narrow the scope of critical objects selection, we first construct a smaller object set $\mathcal{I}$, and assume all objects in $\mathcal{I}$ are possibly important in answering this question. Since we lack annotations about the critical objects for each sample, we followed [39] to extract the objects which are highly related with the QA. Specifically, we first assign POS tags to each word in the QA using the spaCy POS tagger [19] and extract nouns in QA. Then, we calculate the cosine similarity between the GloVe [31] embedding of object categories and the extracted nouns, the similarity scores between all objects in I and the QA are denoted as $\mathcal{SIM}$. We select $|\mathcal{I}|$ objects with the highest $\mathcal{SIM}$ scores as $\mathcal{I}$.

2. Object Local Contributions Calculation. After obtaining the object set $\mathcal{I}$, we start to calculate the local contribution of each object to the predicted probability of ground-truth answer. Following recent works [22, 36, 39] which utilize the modified Grad-CAM [35] to derive the local contribution of each participant, we calculate the contribution of i-th object feature to the ground-truth answer a as:

$$s(a, \boldsymbol{v}_i) = \mathcal{S}(P_{vqa}(a), \boldsymbol{v}_i) \coloneqq (\nabla_{\boldsymbol{v}_i} P_{vqa}(a))^T \mathbf{1}, \quad (4)$$

where $P_{vqa}(a)$ is the predicted answer probability of ground truth answer a, $\boldsymbol{v}_i$ is i-th object feature, and $\mathbf{1}$ is an all-ones vector. Obviously, if the score $s(a, \boldsymbol{v}_i)$ is higher, the contributions of object $\boldsymbol{v}_i$ to answer a is larger.

Algorithm 3 Dynamic Answer Assigning

1: **function** $\text{DA\_ASS}(I^+, Q^+, \mathcal{VQA}, a)$
2:    $\mathcal{VQA}.\text{eval}()$ ▷ don't update parameters
3:    $\_, \_, P^+_{vqa}(\boldsymbol{a}) \leftarrow \mathcal{VQA}(I^+, Q^+, a, \text{True})$
4:    $a^+ \leftarrow \text{top-N}(\text{argsort}_{a_i \in \mathcal{A}}(P^+_{vqa}(a_i)))$
5:    $a^- \coloneqq \{a_i | a_i \in a, a_i \notin a^+\}$ ▷ a is gt answer set
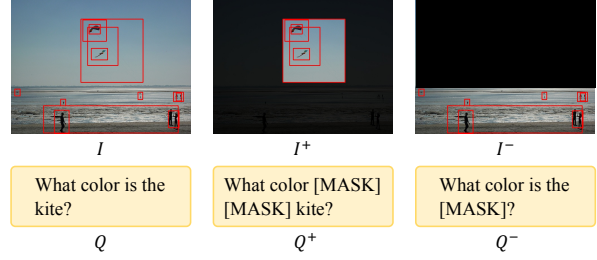6:   **return** a^-
7: **end function**

Figure 3: An informal illustration example of the I^+, I^-, Q^+, and Q^- in $\mathcal{CSS}$. For I^+ and I^-, they are two mutual exclusive object sets. For Q^+ and Q^-, we show the example when word "kite" is selected as critical word.

3. Critical Objects Selection (CO_SEL). After obtaining the private contribution scores $s(a, \boldsymbol{v}_i)$ for all objects in $\mathcal{I}$, we select the top-K objects with highest scores as the critical object set I^+. The K is a dynamic number for each image, which is the smallest number meets Eq. (5):

$$\sum_{\boldsymbol{v}_i \in I^+} \exp(s(a, \boldsymbol{v}_i)) / \sum_{\boldsymbol{v}_j \in \mathcal{I}} \exp(s(a, \boldsymbol{v}_j)) > \eta, \quad (5)$$

where η is a constant, we set $\eta = 0.65$ in all experiments (See Figure 4 for more details about the dynamic K setting).

Then, the counterfactual visual input I^- is the absolute complement of set I^+ in set I, *i.e.*, $I^- = I \backslash I^+$. We show an example of I, I^+, and I^- in Figure 3.

4. Dynamic Answer Assigning (DA_ASS). Given the counterfactual visual input I^- and original question Q, we compose a new VQ pair (I^-, Q). To assign ground truth answers for VQ pair (I^-, Q), we design a dynamic answer assigning (DA_ASS) mechanism. The details of DA_ASS are shown in Algorithm 3. Specifically, we first feed another VQ pair (I^+, Q) into the $\mathcal{VQA}$ model, and obtain the predicted answer distribution $P^+_{vqa}(\boldsymbol{a})$. Based on $P^+_{vqa}(\boldsymbol{a})$, we select the top-N answers with highest predicted probabilities as a^+. Then we define $a^- \coloneqq \{a_i | a_i \in a, a_i \notin a^+\}$. In an extreme case, if the model predicts all ground truth answer correctly for VQ pair (I^+, Q), *i.e.*, $a \subset a^+$, then a^- is a $\emptyset$, *i.e.*, zero for all answer candidates. The basic motivation is that if current model can predict ground truth answer for (I^+, Q) (*i.e.*, I^+ contains critial objects and I^- not), the ground truth for (I^-, Q) should not contain original ground truth answers anymore, *e.g.*, "not green" in Figure 2.

3.2.2 Q-CSS

All steps in Q-CSS are similar to V-CSS. Following its execution path (line 11 to 13 in Algorithm 2), it consists of word local contribution calculation, critical words selection (CW_SEL), and dynamic answer assigning (DA_ASS).

1. Word Local Contribution Calculation. Similar with the V-CSS (cf. Eq. (4)), we calculate the contribution of i-th word feature to the ground-truth answer a as:

$$s(a, \boldsymbol{w}_i) = \mathcal{S}(P_{vqa}(a), \boldsymbol{w}_i) := (\nabla_{\boldsymbol{w}_i} P_{vqa}(a))^T \mathbf{1}. \quad (6)$$

2. Critical Words Selection (CW_SEL.) In this step, we first extract question-type words for each question Q^2 (*e.g.*, "what color" in Figure 3). Then, we select top-K words with highest scores from the remaining sentence (except the question-type words) as critical words. The counterfactual question Q^- is the sentence by replacing all critical words in Q with a special token "[MASK]". Meanwhile, the Q^+ is the sentence by replacing all other words (except question-type and critical words) with "[MASK]". We show an example of Q, Q^+, and Q^- in Figure 3.

3. Dynamic Answer Assigning (DA_ASS.) This step is identical to the DA_ASS in V-CSS, *i.e.*, Algorithm 3. For Q-CSS, the input for DA_ASS is the VQ pair (I, Q^+).

4. Experiments

Settings. We evaluated the proposed CSS for VQA mainly on the VQA-CP test set [3]. We also presented experimental results on the VQA v2 validation set [17] for completeness. For model accuracies, we followed the standard VQA evaluation metric [6]. For fair comparisons, we did all the same data preprocessing steps with the widely-used UpDn model [4] using the publicly available reimplementation[3].

4.1. Ablative Studies

4.1.1 Hyperparameters of V-CSS and Q-CSS

We run a number of ablations to analyze the influence of different hyperparameters of V-CSS and Q-CSS. Specifically, we conducted all ablations by building on top of ensemble-based model LMH [14]. Results are illustrated in Figure 4.
The size of $\mathcal{I}$ in V-CSS. The influence of different size of $\mathcal{I}$ is shown in Figure 4 (a). We can observe that the model's performance gradually decreases with the increase of $|\mathcal{I}|$.
The size of critical objects in V-CSS. The influence of masking different numbers of critical objects is shown in Figure 4 (a). We compared the dynamic K (Eq. (5)) with some fixed constants (*e.g.*, 1, 3, 5). From the results, we can observe that the dynamic K achieves the best performance.
The size of critical words in Q-CSS. The influence of replacing different sizes of critical words is shown in Figure 4

		Model	All	Y/N	Num	Other
Plain Models	UpDn [4]	Baseline	39.74	42.27	11.93	46.05
		Baseline†	39.68	41.93	12.68	45.91
		+Q-CSS	40.05	42.16	12.30	46.56
		+V-CSS	40.98	43.12	12.28	46.86
		+$\mathcal{CSS}$	**41.16**	**43.96**	**12.78**	**47.48**
Ensemble-Based Models	PoE [14, 27]	Baseline	39.93	–	–	–
		Baseline†	39.86	41.96	12.59	46.25
		+Q-CSS	40.73	42.99	12.49	**47.28**
		+V-CSS	**49.65**	**74.98**	**16.41**	45.50
		+$\mathcal{CSS}$	48.32	70.44	13.84	46.20
	RUBi [10]	Baseline	44.23	–	–	–
		Baseline†	45.23	64.85	11.83	44.11
		+Q-CSS	46.31	**68.70**	**12.15**	43.95
		+V-CSS	46.00	62.08	11.84	**46.95**
		+$\mathcal{CSS}$	**46.67**	67.26	11.62	45.13
	LMH [14]	Baseline	52.05	–	–	–
		Baseline†	52.45	69.81	44.46	45.54
		+Q-CSS	56.66	80.82	45.83	46.98
		+V-CSS	58.23	80.53	**52.48**	48.13
		+$\mathcal{CSS}$	**58.95**	**84.37**	49.42	**48.21**

Table 1: Accuracies (%) on VQA-CP v2 test set of different VQA architectures. $\mathcal{CSS}$ denotes the model with both V-CSS and Q-CSS.† represents these results are based on our reimplementation.

(b). From the results, we can observe that replacing only one word (*i.e.*, top-1) achieves the best performance.
The proportion δ of V-CSS and Q-CSS. The influence of different δ is shown in Figure 4 (c). From the results, we can observe that the performance is best when $\delta = 0.5$.

4.1.2 Architecture Agnostic

Settings. Since the proposed CSS is a model-agnostic training scheme, which can be seamlessly incorporated into different VQA architectures. To evaluate the effectiveness of CSS to boost the debiasing performance of different backbones, we incorporated the CSS into multiple architectures including: **UpDn** [4], **PoE** (Product of Experts) [14, 27], **RUBi** [10], **LMH** [14]. Especially, PoE, RUBi, LMH are ensemble-based methods. All results are shown in Table 1.
Results. Compared to these baseline models, the CSS can consistently improve the performance for all architectures. The improvement is more significant in the ensemble-based models (*e.g.*, 6.50% and 9.79% absolute performance gains in LMH and PoE). Furthermore, when both two types of CSS are used, models often achieve the best performance.

4.2. Comparisons with State-of-the-Arts

4.2.1 Performance on VQA-CP v2 and VQA v2

Settings. We incorporated the CSS into model LMH [14], which is dubbed as **LMH-CSS**, and compared it with the state-of-the-art models on both VQA-CP v2 and VQA v2. According to the backbone of these models, we group them into: 1) **AReg** [33], **MuRel** [9], **GRL** [18], **RUBi** [10],
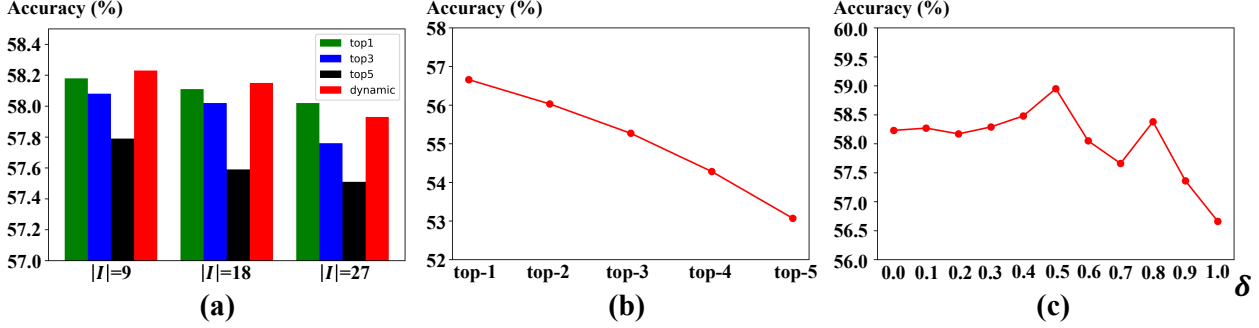
Figure 4: **Ablations**. Accuracies (%) on VQA-CP v2 test set of different hyperparameters settings of V-CSS or Q-CSS. (a) The results of different size of $\mathcal{I}$ and critical objects in V-CSS. All results come from model LMH+V-CSS. (b) The results of different size of critical words in Q-CSS. All results come from model LMH+Q-CSS. (c) The results of different δ. All results come from model LMH+V-CSS+Q-CSS.

SCR [39], **LMH** [14], **HINT** [36]. These models utilize the **UpDn** [4] as their backbone. 2) **HAN** [28], **GVQA** [3], **ReGAT** [25], **NSM** [20]. These models utilize other different backbones, *e.g.*, BLOCK [8], BAN [24] *etc*. Especially, the AReg, GRL, RUBi, LMH are ensemble-based models.
Results. The results are reported in Table 3. When trained and tested on the VQA-CP v2 dataset (*i.e.*, left side of Table 3), the LMH-CSS achieves a new state-of-the-art performance over all question categories. Particularly, CSS improves the performance of LMH with a 6.50% absolution performance gains (58.95% vs. 52.45%). When trained and tested on the VQA v2 dataset (*i.e.*, middle side of Table 3), the CSS results in a minor drop in the performance by 1.74% for LMH. For completeness, we further compared the performance drop between the two benchmarks. Different from previous models that suffer severe performance drops (*e.g.*, 23.74% in UpDn, and 9.19% in LMH), the LMH-CSS can significantly decrease the performance drop into 0.96%, which demonstrate that the effectiveness of CSS to further reduce the language biases in VQA.

4.2.2 Performance on VQA-CP v1

Settings. We further compared the LMH-CSS with state-of-the-art models on VQA-CP v1. Similarly, we group these baseline models into: 1) **GVQA** with SAN [40] backbone, 2) **AReg**, **GRL**, **RUBi**, and **LMH** with UpDn backbone.
Results. Results are reported in Table 2. Compared to these baseline models, the LMH-CSS achieves a new state-of-the-art performance on VQA-CP v1. Particularly, the CSS improves the performance of LMH with a 5.68% absolution performance gains (60.95% vs. 55.27%).

4.3. Improving Visual-Explainable Ability

We will validate the effectiveness of CSS to improve the visual-explainable ability by answering the following questions: **Q1**: Can existing visual-explainable models be incorporated into the ensemble-based framework? **Q2**: How

Model	All	Yes/No	Num	Other
GVQA [3]	39.23	64.72	11.87	24.86
UpDn [4]	39.74	42.27	11.93	**46.05**
+AReg† [33]	41.17	65.49	15.48	35.48
+GRL† [18]	45.69	77.64	13.21	26.97
+RUBi$^{\dagger *}$ [10]	50.90	80.83	13.84	36.02
+LMH$^{\dagger *}$ [14]	55.27	76.47	26.66	45.68
+LMH-CSS	**60.95**	**85.60**	**40.57**	44.62

Table 2: Accuracies (%) on VQA-CP v1 test set of state-of-the-art models. † represents the ensemble-based methods. * indicates the results from our reimplementation using offical released codes.

does CSS improve the model's visual-explainable ability?

4.3.1 CSS vs. SCR (Q1)

Settings. We equipped the existing state-of-the-art visual-explainable model SCR [39] into the LMH framework, and compared it with CSS. Results are reported in Table 4 (a).
Results. Since the training of all SOTA visual-explainable models (*e.g.*, SCR, HINT) are not end-to-end, for fair comparisons, we used a well-trained LMH (*i.e.*, 52.45% accuracies on VQA-CP v2) as the initial model. However, we observe that its performance continues to decrease from the start, which shows that the existing visual-explainable models can not be easily incorporated into the ensemble-based framework. In contrast, CSS can improve the performance.

4.3.2 Evaluations of Visual-Explainable Ability (Q2)

Settings. We evaluate the effectiveness of CSS to improve the visual-explainable ability on both quantitative and qualitative results. For quantitative results, since we lack human annotations about the critical objects for each question, we regard the $\mathcal{SIM}$ score (Section 3.2.1 IO_SEL) as pseudo ground truth. Thus, we design a new metric *Average Importance* ($\mathcal{AI}$): the average $\mathcal{SIM}$ score of the top-K objects with highest $|s(a, \boldsymbol{v})|$. The results are shown in Table 4 (b).

Model	Venue	Expl.	VQA-CP v2 test ↑				VQA v2 val ↑				GapΔ↓	
			All	Yes/No	Num	Other	All	Yes/No	Num	Other	All	Other
HAN [28]	*ECCV'18*		28.65	52.25	13.79	20.33	–	–	–	–	–	–
GVQA [3]	*CVPR'18*		31.30	57.99	13.68	22.14	48.24	72.03	31.17	34.65	16.94	12.51
ReGAT [25]	*ICCV'19*		40.42	–	–	–	67.18	–	–	–	26.76	–
RUBi [10]	*NeurIPS'19*		47.11	68.65	20.28	43.18	61.16	–	–	–	14.05	–
NSM [20]	*NeurIPS'19*		45.80	–	–	–	–	–	–	–	–	–
UpDn [4]	*CVPR'18*		39.74	42.27	11.93	46.05	63.48	81.18	42.14	55.66	23.74	9.61
+AReg† [33]	*NeurIPS'18*		41.17	65.49	15.48	35.48	62.75	79.84	42.35	55.16	21.58	19.68
+MuRel [9]	*CVPR'19*		39.54	42.85	13.17	45.04	–	–	–	–	–	–
+GRL† [18]	*ACL'19*		42.33	59.74	14.78	40.76	51.92	–	–	–	9.59	–
+RUBi†* [10]	*NeurIPS'19*		45.23	64.85	11.83	44.11	50.56	49.45	41.02	53.95	5.33	9.84
+SCR [39]	*NeurIPS'19*		48.47	70.41	10.42	47.29	62.30	77.40	40.90	56.50	13.83	9.21
+LMH†* [14]	*EMNLP'19*		52.45	69.81	44.46	45.54	61.64	77.85	40.03	55.04	9.19	9.50
+LMH-CSS	*CVPR'20*		**58.95**	**84.37**	**49.42**	**48.21**	59.91	73.25	39.77	55.11	**0.96**	**6.90**
+HINT [36]	*ICCV'19*	HAT	47.70	70.04	10.68	46.31	62.35	80.49	41.75	54.01	14.65	7.70
+SCR [39]	*NeurIPS'19*	HAT	49.17	71.55	10.72	47.49	62.20	78.90	41.40	54.30	13.03	6.81
+SCR [39]	*NeurIPS'19*	VQA-X	49.45	72.36	10.93	48.02	62.20	78.80	41.60	54.40	12.75	6.38

Table 3: Accuracies (%) on VQA-CP v2 test set and VQA v2 val set of state-of-the-art models. The gap represents the accuracy difference between VQA v2 and VQA-CP v2. † represents the *ensemble-based* methods. *Expl.* denotes the model has used extra human annotations, *e.g.*, human attention (HAT) or explanations (VQA-X). * indicates the results from our reimplementation using official released codes.

Model	All	Yes/No	Num	Other
SCR	48.47	70.41	10.42	47.29
LMH	52.45	69.81	44.46	45.54
LMH+SCR	continued decrease			
LMH+$\mathcal{CSS}$	58.95	84.37	49.42	48.21

(a) Accuracies (%) on VQA-CP v2 test set.

Model	Top-1	Top-2	Top-3
UpDn	22.70	21.58	20.89
SCR	27.58	26.29	25.38
LMH	29.67	28.06	27.04
LMH+V-CSS	30.24	28.53	27.51
LMH+$\mathcal{CSS}$	**33.43**	**31.27**	**29.86**

(b) $\mathcal{AI}$ score (%) on VQA-CP v2 test set.

Model	k=1	k=2	k=3	k=4	$\mathcal{CI}$
UpDn	49.94	38.80	31.55	28.08	6.01
LMH	51.68	39.84	33.38	29.11	7.44
LMH+Q-CSS	54.83	42.34	35.48	31.02	9.02
LMH+$\mathcal{CSS}$	**55.04**	**42.78**	**35.63**	**31.17**	**9.03**

(c) **Left**: $CS(k)$ (%) on VQA-CP-Rephrasing; **Right**: $\mathcal{CI}$ score (%) on VQA-CP v2 test set.

Table 4: Quantitative results about the evaluation of the VQA models' visual-explainable and question-sensitive abilities.

For qualitative results, we illustrated in Figure 5 (a).
Results. From Table 4 (b), we can observe that CSS dramatically improves the $\mathcal{AI}$ scores, which means the actually influential objects are more related to the QA pair. From Figure 5 (a), we can find that the CSS helps the model to make predictions based on critical objects (*i.e.*, green boxes), and suppress the influence of irrelevant objects (*i.e.*, red boxes).

4.4. Improving Question-Sensitive Ability

We will validate the effectiveness of CSS to improve the question-sensitive ability by answering the following questions: **Q3**: Does CSS helps to improve the robustness to diverse rephrasings of questions? **Q4**: How does CSS improve the model's question-sensitive abilities?

4.4.1 Robustness to Rephrasings of Questions (Q3)

Settings. As discussed in previous work [37], being robust to diverse rephrasing of questions is one of key behaviors of a question-sensitive model. To more accurately evaluate the robustness, we re-splited the existing dataset VQA-Rephrasings [37] with the same splits as VQA-CP, and denoted it as VQA-CP-Rephrasings. For evaluation, we used

the standard metric *Consensus Score $CS(k)$*. Results are reported in Table 4 (c) (left). We refer readers to [37] for more details about the VQA-Rephrasings and metric $CS(k)$.
Results. From Table 4 (c), we can observe that Q-CSS dramatically improves the robustness to diverse rephrasings of questions. Furthermore, V-CSS can help to further improve the robustness, *i.e.*, $\mathcal{CSS}$ achieves the best performance.

4.4.2 Evaluations of Question-Sensitive Ability (Q4)

Settings. We evaluate the effectiveness of CSS to improve the question-sensitive ability on both quantitative and qualitative results. For quantitative results, since there is no standard evaluation metric, we design a new metric *Confidence Improvement* ($\mathcal{CI}$): Given a test sample (I, Q, a), we remove a critical noun in question Q, and obtain a new test sample (I, Q^*, a)[4]. Then we feed both two samples into evaluated model, and calcluate the confidence decreses of the ground-truth answer. We formally define $\mathcal{CI}$ in Eq. 7:

$$\mathcal{CI} = \frac{\sum_{(I,Q)}(P_{vqa}(a|I,Q) - P_{vqa}(a|I,Q^*)) \cdot \mathbf{1}(a = \hat{a})}{\sum_{(I,Q)} 1} \quad (7)$$

---

[4]The auxiliary test set is released in: github.com/yanxinzju/CSS-VQA
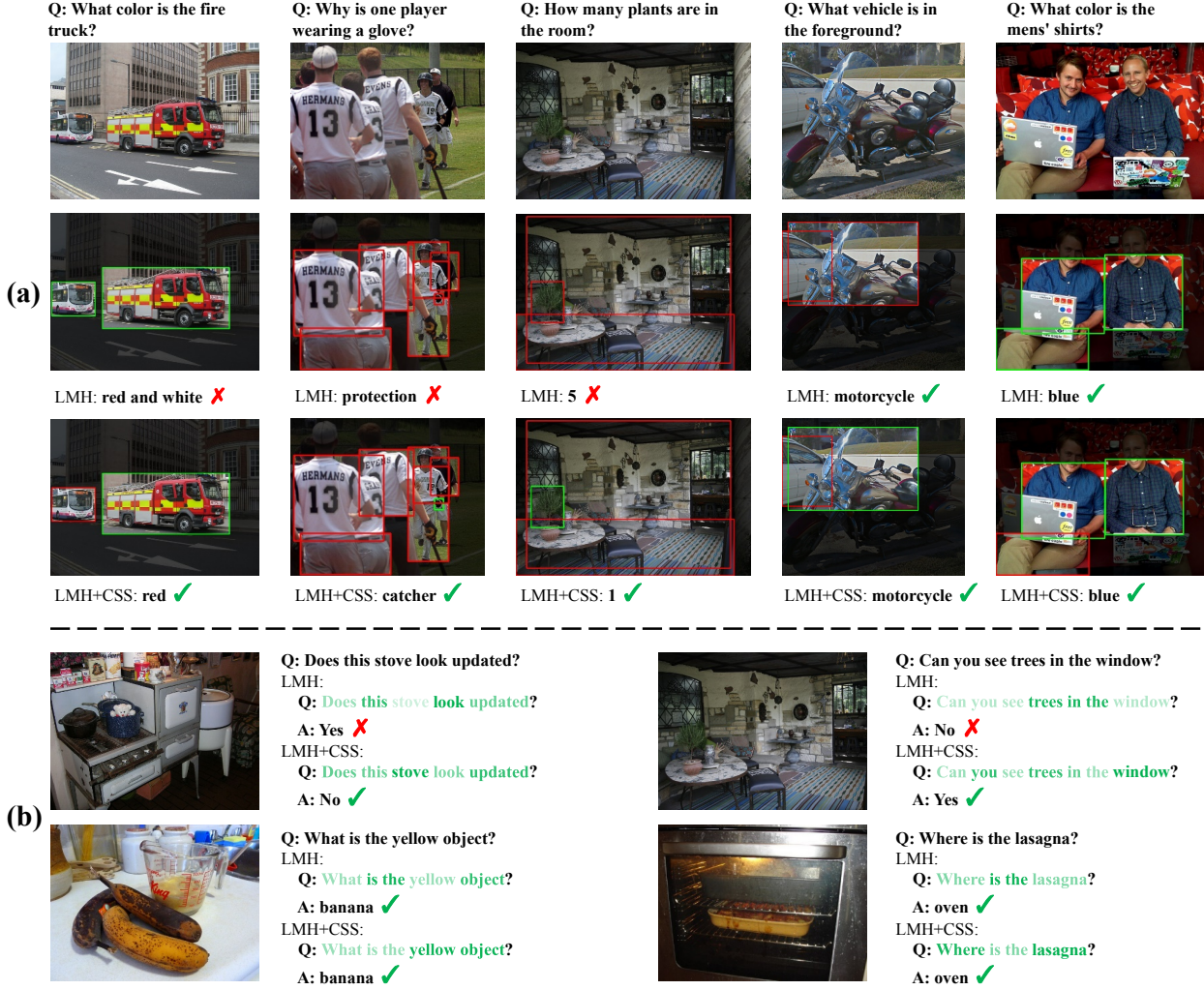
Figure 5: (a) **visual-explainable ability**: The **green** boxes denote their scores $s(\hat{a}, \boldsymbol{v})>0$, *i.e.*, positive contributions to final predictions; The **red** boxes denote their scores $s(\hat{a}, \boldsymbol{v})<0$, *i.e.*, negative contributions to final predictions. Only objects which are highly related to the QA pair are shown (*i.e.*, $\mathcal{SIM} \geq 0.6$). (b) **question-sensitive ability**: The different shades of green color in the question denotes the relative values of $s(\hat{a}, \boldsymbol{w})$. Thus, the word with darker green denotes the word has larger contribution to final predictions.

where $\hat{a}$ is the model predicted answer for sample (I, Q), $\mathbf{1}$ is an indicator function. The results are reported in Table 4 (c). For qualitative results, we illustrated in Figure 5 (b).

Results. From Table 4 (c), we can observe that CSS helps the model to benefit more from the critical words, *i.e.*, removing critical words results in more confidence drops for the ground-truth answers. From Figure 5 (b), we can find that CSS helps the model to make predictions based on critical words (*e.g.*, "stove" or "lasagna"), *i.e.*, forcing model to understand the whole questions before making predictions.

5. Conclusion

In this paper, we proposed a model-agnostic Counterfactual Samples Synthesizing (CSS) training scheme to im-prove the model's visual-explainable and question-sensitive abilities. The CSS generates counterfactual training samples by masking critical objects or words. Meanwhile, the CSS can consistently boost the performance of different VQA models. We validate the effectiveness of CSS through extensive comparative and ablative experiments. Moving forward, we are going to 1) extend CSS to other visual-language tasks that suffer severe language biases; 2) design a specific VQA backbone to benefits from CSS.

References

[1] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *arXiv*, 2019. 3

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

[3] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018. 1, 2, 5, 6, 7

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 1, 3, 5, 6, 7

[5] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016. 1

[6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 1, 5

[7] Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *ACL*, 2019. 2, 3

[8] Hedi Ben-Younes, Rémi Cadene, Nicolas Thome, and Matthieu Cord. Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *AAAI*, 2019. 6

[9] Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. Murel: Multimodal relational reasoning for visual question answering. In *CVPR*, 2019. 5, 7

[10] Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases in visual question answering. In *NeurIPS*, 2019. 2, 3, 5, 6, 7

[11] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multiagent training for scene graph generation. In *ICCV*, 2019. 3

[12] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, 2018. 2

[13] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 2017. 3

[14] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *EMNLP*, 2019. 1, 2, 3, 5, 6, 7

[15] 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 *EMNLP*, 2016. 1

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2, 3

[17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 1, 2, 5

[18] Gabriel Grand and Yonatan Belinkov. Adversarial regularization for visual question answering: Strengths, shortcomings, and side effects. In *ACL workshop*, 2019. 2, 3, 5, 6, 7

[19] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017. 4

[20] Drew A Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. In *NeurIPS*, 2019. 6, 7

[21] Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. Revisiting visual question answering baselines. In *ECCV*, 2016. 2

[22] Sarthak Jain and Byron C Wallace. Attention is not explanation. In *NAACL*, 2019. 4

[23] 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 *CVPR*, 2017. 1

[24] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *NeurIPS*, 2018. 6

[25] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *ICCV*, 2019. 6, 7

[26] Chenxi Liu, Junhua Mao, Fei Sha, and Alan Yuille. Attention correctness in neural image captioning. In *AAAI*, 2017. 3

[27] Rabeeh Karimi Mahabadi and James Henderson. Simple but effective techniques to reduce biases. In *arXiv*, 2019. 2, 3, 5

[28] Mateusz Malinowski, Carl Doersch, Adam Santoro, and Peter Battaglia. Learning visual question answering by bootstrapping hard attention. In *ECCV*, 2018. 6, 7

[29] Yulei Niu, Hanwang Zhang, Manli Zhang, Jianhong Zhang, Zhiwu Lu, and Ji-Rong Wen. Recursive visual attention in visual dialog. In *CVPR*, 2019. 3

[30] Jingjing Pan, Yash Goyal, and Stefan Lee. Question-conditioned counterfactual image generation for vqa. In *arXiv*, 2019. 3

[31] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 4

[32] Tingting Qiao, Jianfeng Dong, and Duanqing Xu. Exploring human-like attention supervision in visual question answering. In *AAAI*, 2018. 3

[33] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *NeurIPS*, 2018. 2, 3, 5, 6, 7

[34] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *IJCAI*, 2017. 2

[35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 3, 4

[36] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *ICCV*, 2019. 3, 4, 6, 7

[37] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *CVPR*, 2019. 3, 7

[38] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019. 3

[39] Jialin Wu and Raymond J Mooney. Self-critical reasoning for robust visual question answering. In *NeurIPS*, 2019. 3, 4, 6, 7

[40] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 6

[41] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. Video question answering via attribute-augmented attention network learning. In *SIGIR*, 2017. 3

[42] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *CVPR*, 2016. 1, 2

[43] Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. Interpretable visual question answering by visual grounding from attention supervision mining. In *WACV*, 2019. 3