

# Counterfactual Samples Synthesizing and Training for Robust Visual Question Answering

Long Chen\*, Yuhang Zheng\*, Yulei Niu, Hanwang Zhang, and Jun Xiao†

**Abstract**—Today’s VQA models still tend to capture superficial linguistic correlations in the training set and fail to generalize to the test set with different QA distributions. To reduce these language biases, recent VQA works introduce an auxiliary question-only model to regularize the training of targeted VQA model, and achieve dominating performance on diagnostic benchmarks for out-of-distribution testing. However, due to complex model design, these ensemble-based methods are unable to equip themselves 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 questions. To this end, we propose a novel model-agnostic Counterfactual Samples Synthesizing and Training (CSST) strategy. After training with CSST, VQA models are forced to focus on all critical objects and words, which significantly improves both visual-explainable and question-sensitive abilities. Specifically, CSST is composed of two parts: *Counterfactual Samples Synthesizing* (CSS) and *Counterfactual Samples Training* (CST). CSS generates counterfactual samples by carefully masking critical objects in images or words in questions and assigning pseudo ground-truth answers. CST not only trains the VQA models with both complementary samples to predict respective ground-truth answers, but also urges the VQA models to further distinguish the original samples and superficially similar counterfactual ones. To facilitate the CST training, we propose two variants of supervised contrastive loss for VQA, and design an effective positive and negative sample selection mechanism based on CSS. Extensive experiments have shown the effectiveness of CSST. Particularly, by building on top of model LMH+SAR [1], [2], we achieve record-breaking performance on all out-of-distribution benchmarks (e.g., VQA-CP v2, v1, and GQA-OOD).

**Index Terms**—Visual Question Answering, Counterfactual Thinking, Language Biases, Data Augmentation, Contrastive Learning.

## 1 INTRODUCTION

VISUAL Question Answering (VQA), *i.e.*, answering natural language questions about the given visual content, is one of the essential abilities of advanced AI agents. With the release of multiple large-scale VQA datasets (e.g., VQA v1 [3], VQA v2 [4], CLEVR [5], and GQA [6]), VQA has received unprecedented attentions and hundreds of VQA 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 between the questions and answers (*a.k.a.*, language biases) [4], [5], [7], [8]. For example, a model naively answering “2” for all “How many X?” questions can still get satisfactory performance regardless of “X”. To disentangle the bias factors and clearly monitor the progress of VQA research, several diagnostic benchmarks have been proposed, such as VQA-CP [9] and GQA-OOD [10]. These benchmarks deliberately keep different question-answer distributions in training and test sets. Moreover, the performance of many state-of-the-art VQA models [11], [12], [13], [14], [15] all consistently drop significantly on these diagnostic benchmarks compared to their counterparts (*i.e.*, VQA v2/v1 and GQA).

Currently, the most prevalent solutions to mitigate the

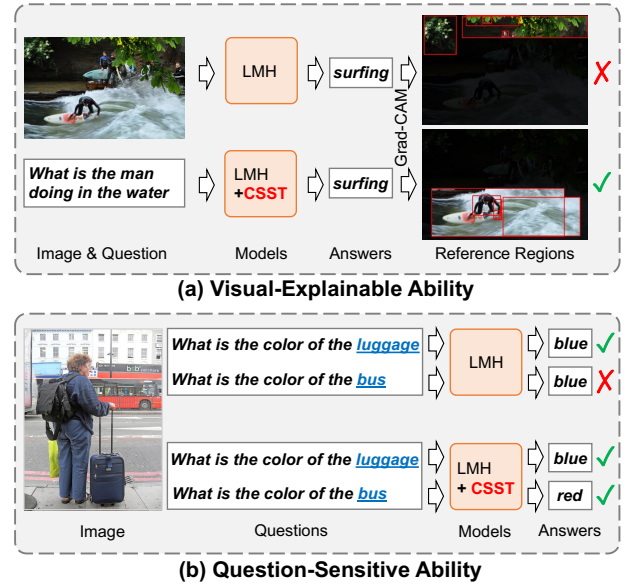


Fig. 1: Two indispensable characteristics of an ideal VQA model. (a) **Visual-explainable ability**: The model can not only predict correct answer (e.g., “surfing”), but also rely on the right visual reference regions when making this prediction. (b) **Question-sensitive ability**: The model should be sensitive to the linguistic variations. For example, after replacing the critical word “luggage” with “bus”, the predicted answers of the two questions should be different. LMH [1] is a SOTA VQA model.

language bias issues are **ensemble-based** methods: they introduce an auxiliary question-only model to regularize the training of targeted VQA model. Specifically, these methods can further be categorized into two sub-types: 1) *Adversary-*

- \* These authors contributed equally, and † denotes corresponding author.
- L. Chen is with the Department of Electrical Engineering, Columbia University, New York, NY, USA, 10027. Part of this work was done when L. Chen at Zhejiang University. E-mail: zjuchenlong@gmail.com.
- Y. Zheng and J. Xiao are with the College of Computer Science, Zhejiang University, Hangzhou, China, 310027. E-mails: itemzheng@zju.edu.cn, junx@zju.edu.cn.
- Y. Niu and H. Zhang are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, 639798. E-mails: yn.yuleiniu@gmail.com, hanwangzhang@ntu.edu.sg.

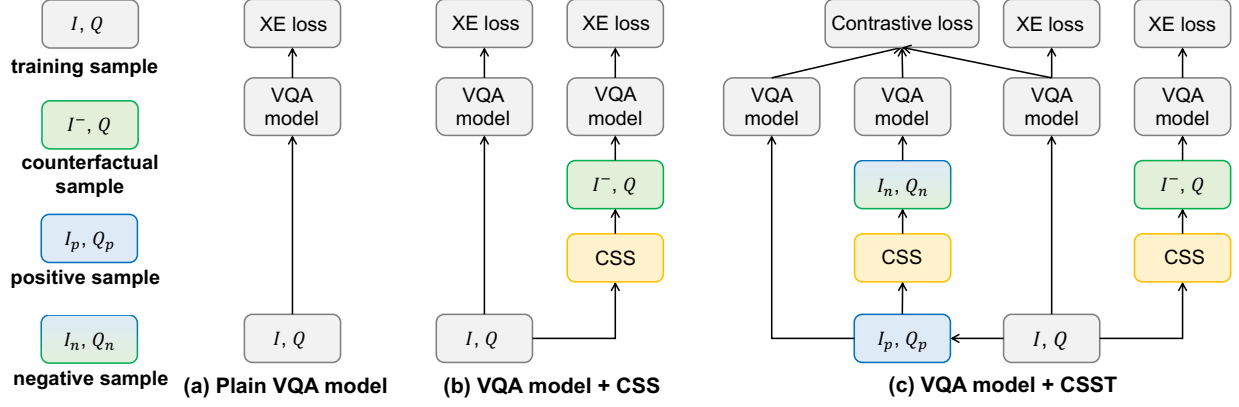


Fig. 2: Comparisons on three different VQA frameworks, and the weights of all VQA models are shared in each framework. (a) The most common VQA framework, *i.e.*, a plain VQA model with XE loss using original samples. (b) The framework of previous CSS work [16]: Using CSS to generate counterfactual samples, and training the VQA model using XE loss for both original samples and counterfactual samples. (c) The framework of the proposed CSST: Besides the XE training objectives, we use CSS to further generate positive-negative sample pairs, and train the VQA model with an extra contrastive loss. For conciseness, we only show one (type of) counterfactual sample and negative sample in each figure, and we skip the answer sets for all samples in each figure.

*based models* [17], [18], [19]: They train two models (*i.e.*, the question-only and targeted VQA model) in an adversarial manner [20], [21], *i.e.*, minimizing the loss of VQA model while maximizing the loss of question-only model. Since these two models are typically designed to share the same question encoder, these adversary-based methods aim to reduce the language biases by learning bias-neutral question representations. Unfortunately, this adversarial training scheme brings significant noise into gradient calculations, which results in an unstable training process [18]. 2) *Fusion-based models* [1], [22], [23], [24], [25]: They late fuse the predicted answer distributions of the two models, and derive training gradients based on fused answer distribution. The design philosophy of these fusion-based models, is to let the targeted VQA model focuses more on samples, which can't be easily answered correctly by the question-only model.

Although the ensemble-based methods have dominated the performance on these diagnostic benchmarks, it is worth noting that current methods fail to equip themselves with two indispensable characteristics of an ideal VQA model: 1) **Visual-explainable ability**: The VQA model should rely on the right visual regions when making decisions, *i.e.*, right for the right reasons [26]. As shown in Fig. 1 (a), although both two models can predict the correct answer “surfing”, they actually refer to different visual reference regions when making their respective answer predictions. 2) **Question-sensitive ability**: The VQA model should be sensitive to the linguistic variations in questions. As shown in Fig. 1 (b), for two questions with similar sentence structure (*e.g.*, only replacing the word “luggage” with “bus”), if the meanings of the two questions are different, the model should perceive the discrepancy and make corresponding predictions.

In this paper, we propose a novel model-agnostic *Counterfactual Samples Synthesizing and Training* (CSST) strategy. CSST serves as a plug-and-play component to improve VQA models’ visual-explainable and question-sensitive abilities, even for complex ensemble-based models. Specifically, CSST is composed of two parts: Counterfactual Samples Synthesizing (CSS) and Counterfactual Samples Training (CST).

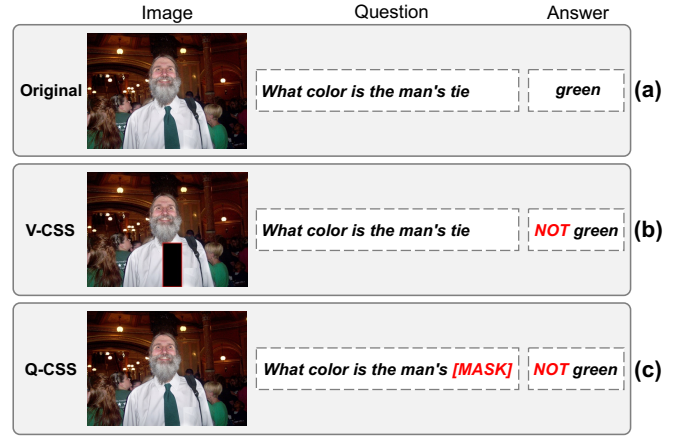




Fig. 3: (a): A training sample from VQA-CP. (b): The synthesized training sample by V-CSS. It masks critical objects (*e.g.*, ) in image and assigns different GT answers (*e.g.*, “not green”). (c): The synthesized training sample by Q-CSS. It replaces critical words (*e.g.*, “tie”) with special token “[MASK]” in the question and assigns different GT answers (*e.g.*, “not green”).

For each original VQA training sample, CSS can generate a corresponding counterfactual sample. As shown in Fig. 3, CSS consists of two different types of sample synthesizing mechanisms: V-CSS and Q-CSS. For V-CSS, it synthesizes counterfactual images by masking critical objects in the original image. By “critical”, we mean that these objects are important for answering a certain question. For example, object  (*i.e.*, green tie) is a critical object for 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 counterfactual questions by replacing critical words in original question with a special token “[MASK]”. Similarly, the counterfactual question and original image compose a new VQ pair. Meanwhile, to avoid expensive manual annotations, we design a dynamic answer assigning mechanism to approximate ground-truth answers for all synthesized VQ pairs (*e.g.*, “not green”).

Based on our general sample synthesizing mechanism



munity, especially for self-supervised representation learning [45], [46], [47]. The core idea of contrastive learning is to maximize the mutual information between the input samples and positive samples, and minimize the one between negative samples. Currently, there are several VQA works [28], [43], [48] also utilize contrastive training to distinguish the original training samples and superficially similar counterparts. Different from all existing works, we equip our CSS mechanism into positive and negative selection for contrastive training, which not only increases the sample diversity, but also meets the semantic requirements.

### 3 APPROACH

We follow the common formulation and regard the VQA task as a multi-label 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 answer sets  $a_i \in \{a_i^j | a_i^j \in \mathcal{A}\}$ , the VQA task learns a mapping  $f_{vqa} : \mathcal{I} \times \mathcal{Q} \rightarrow [0, 1]^{|A|}$ , which produces an answer distribution given image-question pair.  $\mathcal{I}$ ,  $\mathcal{Q}$ , and  $\mathcal{A}$  denote the set of images, questions, and answers, respectively. For each ground-truth answer  $a_i^j \in a_i$ , it has a soft target score  $t_i^j$ , and we use the same ground-truth soft target scores as in prior works [14]. For simplicity, we omit subscript  $i$  in the following sections.

In this section, we first introduce the UpDn baseline [14] and a typical ensemble-based framework in Sec. 3.1. Then, we introduce the details of Counterfactual Samples Synthesizing (CSS) in Sec. 3.2. Last, we introduce the details of Counterfactual Samples Training (CST) in Sec. 3.3.

#### 3.1 Preliminaries

**Bottom-Up Top-Down (UpDn) Model [14].** For each image  $I$ , the UpDn uses an image encoder  $e_v$  to output a set of object features:  $V = \{v_1, \dots, v_{n_v}\}$ , where  $v_i$  is  $i$ -th object feature. For each question  $Q$ , the UpDn uses a question encoder  $e_q$  to output a set of word features:  $Q = \{w_1, \dots, w_{n_q}\}$ , where  $w_j$  is  $j$ -th word feature. Then, both  $V$  and  $Q$  are fed into the model  $f_{vqa}$  to predict answer distributions:

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

Typically, model  $f_{vqa}$  contains a soft attention mechanism, and it is trained with a (binary) cross-entropy (XE) loss.

**Ensemble-Based Models.** As we discussed in Sec. 1, the ensemble-based models can be grouped into two sub-types: *adversary-based* and *fusion-based*. Since adversary-based models [17], [18], [19] always suffer severe unstable training and relatively worse performance, in this section, we only introduce the typical fusion-based framework [1], [22], [23]. As shown in Algorithm 1, they introduce an auxiliary question-only model  $f_q$  which takes  $Q$  as input and predicts answers:

$$P_q(a|Q) = f_q(Q). \quad (2)$$

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

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

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

#### Algorithm 1 Ensemble-based Model (fusion-based)

**Inputs:** original training sample  $(I, Q, a)$ , *update\_cond* is a control condition for parameter updates.

**Outputs:** images features  $V$ , question features  $Q$ , and predicted answer distribution  $P_{vqa}(a)$ .

```

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

```

#### 3.2 Counterfactual Samples Synthesizing (CSS)

The overall structure of the CSS training scheme is shown in Algorithm 2. Specifically, for any VQA model, given an original training sample  $(I, Q, a)$ , CSS can generate two types of counterfactual samples:  $(I^-, Q, a^-)$  by V-CSS or  $(I, Q^-, a^-)$  by Q-CSS. In the following, we mainly introduce the details of V-CSS and Q-CSS. As shown in Algorithm 2, for each specific training sample, we only use one certain synthesizing mechanism, and  $\delta$  is the trade-off weight (see Fig. 5 (c) for more details about the influence of different  $\delta$ ).

##### 3.2.1 V-CSS

We sequentially introduce all steps of V-CSS following its execution path (*i.e.*, 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 soft answer assigning (DSA\_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 specific question. Since we lack manual annotations about the critical objects for each sample, we followed [37] to extract objects which are highly related to the QA. Specifically, we first assign POS tags to each word in the QA using the spaCy POS tagger [49] and extract nouns in QA. Then, we calculate the cosine similarity between the GloVe [50] embeddings of object categories<sup>1</sup> and the extracted nouns, the similarity scores between all objects in  $I$  and the QA are denoted as  $STM$ . We select  $|\mathcal{I}|$  objects with the highest  $STM$  scores as the initial object set  $\mathcal{I}$ .

**2. Object Local Contributions Calculation.** After obtaining the initial 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 [36], [37], [51] which utilize modified Grad-CAM [38] to derive the local

<sup>1</sup>For the VQA-CP dataset, we followed prior works [37] to utilize these object category annotations, and the influence of different sizes of initial objects are also illustrated in Fig. 5 (a). For the GQA-OOD dataset, for fair comparisons with others, we skip this step for all experiments.



**Algorithm 2** Counterfactual Samples Synthesizing (CSS)

**Inputs:** original training sample  $(I, Q, a)$ , and a typical VQA model  $\mathbf{VQA}$ .

**Outputs:** counterfactual training sample  $(I^-, Q, a^-)$  from V-CSS or  $(I, Q^-, a^-)$  from Q-CSS.

```

1: function CSS( $I, Q, a$ )
2:    $\mathbf{V}, \mathbf{Q}, P_{vqa}(a) \leftarrow \mathbf{VQA}(I, Q, a, \text{False})$ 
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:      $s(a, \mathbf{v}_i) \leftarrow \mathcal{S}(P_{vqa}(a), \mathbf{v}_i)$ 
7:      $I^+, I^- \leftarrow \text{CO\_SEL}(\mathcal{I}, \{s(a, \mathbf{v}_i)\})$ 
8:      $a^- \leftarrow \text{DSA\_ASS}(I^+, Q, \mathbf{VQA}, a)$ 
9:     return  $(I^-, Q, a^-)$ 
10:  else ▷ execute Q-CSS
11:     $s(a, \mathbf{w}_i) \leftarrow \mathcal{S}(P_{vqa}(a), \mathbf{w}_i)$ 
12:     $Q^+, Q^- \leftarrow \text{CW\_SEL}(\{s(a, \mathbf{w}_i)\})$ 
13:     $a^- \leftarrow \text{DSA\_ASS}(I, Q^+, \mathbf{VQA}, a)$ 
14:    return  $(I, Q^-, a^-)$ 
15:  end if
16: end function

```

contribution of each participant, we calculate the contribution of  $i$ -th object feature to the ground-truth answer  $a$  as:

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

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

**3. Critical Objects Selection (CO\_SEL).** After obtaining the private contribution scores  $s(a, \mathbf{v}_i)$  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):

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

where  $\eta$  is a constant, we set  $\eta = 0.65$  in all experiments (See Fig. 5 for more details about the influence of dynamic  $K$  setting). Since the visual features  $\mathbf{V} = \{\mathbf{v}_i\}$  are extracted from a pretrained Faster R-CNN, *i.e.*, they always have a lot of repetitions due to overlapped image regions. To avoid visual clue leakage from the other object visual features [52], we also regard the object that overlaps any critical objects by 60% IoU or more as critical objects. Then, the counterfactual input  $I^-$  is the absolute complement of set  $I^+$  in set  $I$ , *i.e.*,  $I^- = I \setminus I^+$ . We show an example of  $I, I^+$  and  $I^-$  in Fig. 4.

**4. Dynamic Soft Answer Assigning (DSA\_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 the VQ pair  $(I^-, Q)$ , we design a new dynamic soft answer assigning (DSA\_ASS) mechanism. The details of DSA\_ASS are shown in Algorithm 3. Specifically, we first feed another VQ pair  $(I^+, Q)$  into the  $\mathbf{VQA}$  model, and obtain the predicted answer distribution  $P_{vqa}^+(a)$ . We directly use the same set of ground-truth answer categories as the pseudo ground-truth for the corresponding counterfactual sample. Meanwhile, we re-assign the ground-truth

**Algorithm 3** Dynamic Soft Answer Assigning (DSA\_ASS)

**Inputs:** VQ pair  $(I^+, Q^+)$ , a typical VQA model  $\mathbf{VQA}$ , and original ground-truth answer set  $a$ .

**Outputs:** automatically assigned pseudo gt answer set  $a^-$ , and corresponding ground-truth soft target scores  $\{t^{i-}\}$ .

```

1: function DA_ASS( $I^+, Q^+, \mathbf{VQA}, a$ )
2:    $\_, \_, P_{vqa}^+(a) \leftarrow \mathbf{VQA}(I^+, Q^+, a, \text{False})$ 
3:    $a^- := a$  ▷ same gt answer set
4:   for  $a^{i-}$  in  $a^-$  do
5:      $t^{i-} := 1 - P_{vqa}^+(a^{i-})$  ▷ soft answer
6:   end for
7:   return  $a^-$ 
8: end function

```

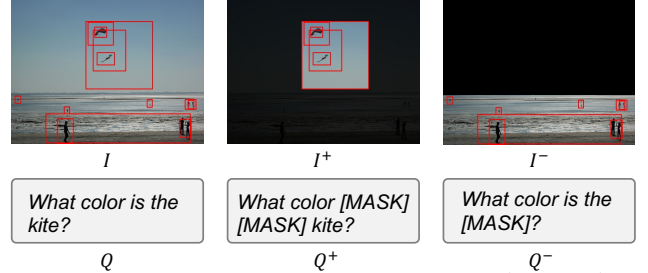


Fig. 4: An informal illustration example of the  $I^+, I^-, Q^+$ , and  $Q^-$  in CSS. For  $I^+$  and  $I^-$ , they are two mutual exclusive object sets. For  $Q^+$  and  $Q^-$ , we show the example when word “kite” is selected as critical word. For clarity, we omit some objects.

probability for each answer category of  $a^-$ , *e.g.*, the  $i$ -th answer category  $a^{i-}$  is set to  $P^{gt}(a^{i-}) = 1 - P_{vqa}^+(a^{i-})$ . In an extreme case, if the model predicts all ground-truth answers correctly for the VQ pair  $(I^+, Q)$  with 100% probabilities, then  $a^-$  is a  $\emptyset$ , *i.e.*, zero for all answer candidates. The basic motivation is that if the current model can predict ground-truth answers for  $(I^+, Q)$  (*i.e.*,  $I^+$  contains critical objects and  $I^-$  not), the ground-truth for  $(I^-, Q)$  should not contain original ground-truth answers anymore, *e.g.*, “not green” in Fig 2. Similarly, the higher confidence for  $(I^+, Q)$ , the less probability for this answer as the ground-truth for  $(I^-, Q)$ .

### 3.2.2 Q-CSS

All steps in Q-CSS are similar to V-CSS. Following its execution path (*i.e.*, line 11 to 13 in Algorithm 2), it consists of word local contribution calculation, critical words selection (CW\_SEL), and dynamic soft answer assigning (DSA\_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:

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

**2. Critical Words Selection (CW\_SEL).** In this step, we first extract question-type<sup>2</sup> words for each question  $Q$  (*e.g.*, “what color” in Fig. 4 is the question-type of question “What color is the kite?”). Then, we select top- $K$  words with the highest scores from the remaining sentence

<sup>2</sup>We slightly abuse “question-type” here. For VQA-CP, the question-type denotes the default question-type annotations in the original dataset. For GQA-OOD, the question-type denotes the annotated local groups, which can be easily mapped back to each question.

(except 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 Fig. 4.

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

### 3.2.3 Highlights of Two Improvements on CSS [16]

Compared to the initial CSS in the conference version [16], we make two important modifications and improvements:

- 1) **Object Overlapping in CO\_SEL.** Different from the words features in questions, which are relatively “independent”, object features from the same image always have a lot of repetitions due to their overlapped image regions. We avoid this visual clue leakage by considering the IoU overlaps between all objects. This strategy helps models rely more on the calculated critical objects.
- 2) **“Soft” Answer Assigning in DSA\_Ass.** In initial CSS work [16], we select top-K predictions based on  $P_{vqa}^+$  (cf. Algorithm 3), and assign ground-truth answers based on these top predictions. Compared to this “hard” answer selection, DSA\_Ass is more robust to the selection of hyperparameter  $K$ , and is more sensitive to perceive the changes of answer predictions (*i.e.*,  $P_{vqa}^+$ ).

To distinguish these two types of CSS, we denote the current improved version as CSS<sup>+</sup> in the following sections (See results in Sec. 4 for comparisons between CSS & CSS<sup>+</sup>).

## 3.3 Counterfactual Samples Training (CST)

To further benefit from the counterfactual samples generated by CSS, we propose a CST strategy for model training (cf. Algorithm 4). Specifically, it consists of a cross-entropy (XE) training loss and a contrastive (CR) training loss.

### 3.3.1 XE Training on Counterfactual Samples

For XE training, we directly follow other state-of-the-art VQA models (*e.g.*, UpDn [14]) and use binary cross-entropy loss as training objective, *i.e.*,

$$L_{XE} = - \sum_i^M \left[ t^i \log \sigma(\hat{P}_{vqa}(a|I, Q)) + (1 - t^i) \log(1 - \sigma(\hat{P}_{vqa}(a|I, Q))) \right], \quad (7)$$

where  $\sigma$  denotes the sigmoid activation function, and  $t^i$  is the soft target score of  $i$ -th ground-truth answer for this training sample (cf. line 5 in Algorithm 3). Different from the most prevalent VQA framework which only uses original samples in the XE training (cf. Fig. 2 (a)), in our XE training, we feed both original training samples and their corresponding counterfactual samples into the same VQA model (*i.e.*, line 3 to 5 in Algorithm 4).

### 3.3.2 Contrastive Training on Counterfactual Samples

For contrastive training, we propose a novel and effective positive and negative samples selection mechanism based on our CSS, and two variants of contrastive loss for VQA.

## Algorithm 4 Counterfactual Samples Training (CST)

**Inputs:** original training sample  $(I, Q, a)$ , and a typical VQA model VQA, and CSS.

```

1: function CST( $I, Q, a$ )
2:   # XE training
3:    $(I^-, Q, a^-) \leftarrow \text{CSS}(I, Q, a)$   $\triangleright$  V-CSS for example
4:    $-, -, P_a \leftarrow \text{VQA}(I, Q, a, \text{True})$   $\triangleright$  original sples.
5:    $-, -, - \leftarrow \text{VQA}(I^-, Q, a^-, \text{True})$   $\triangleright$  counterfact. sples.
6:
7:   # samples selection
8:    $(I_p, Q_p, a) \leftarrow \text{POS\_SEL}(I, Q, a)$   $\triangleright$  positive sples.
9:    $\{(I_n^i, Q_n^i, -)\} \leftarrow \text{NEG\_SEL}(I, Q, a)$   $\triangleright$  negative sples.
10:
11:  # contrastive training
12:   $-, -, P_p \leftarrow \text{VQA}(I_p, Q_p, \text{False})$   $\triangleright$  for pos. sples.
13:   $-, -, P_n^i \leftarrow \text{VQA}(I_n^i, Q_n^i, \text{False})$   $\triangleright$  for all neg. sples.
14:   $L_{CR} \leftarrow \text{CR}(P_a, P_p, \{P_n^i\})$   $\triangleright$  CR training
15: end function

```

**Positive & Negative Samples Selection Strategies.** Without loss of generality, we regard each original training sample as the *anchor* sample, and we regard all samples with the same question-type<sup>2</sup> and ground-truth answer categories as the candidate positive set<sup>3</sup>. In each training step, we randomly sample one sample from the candidate set as the *positive* sample (*i.e.*, POS\_SEL in Algorithm 4). For each positive sample, we compose four different types of *negative* samples, which can be categorized into two groups based on their sampling strategies. The first group of negative samples is the counterfactual samples of the positive samples corresponding to V-CSS and Q-CSS, respectively. The second group of negative samples is randomly sampled samples: including 1) a random sampled sample with the same question-type<sup>2</sup> but different ground-truth answer; and 2) a composed sample by replacing the original image with a randomly sampled image from the same batch. All four types of negative sample selection strategies constitute the NEG\_SEL step in Algorithm 4. Detailed quantitative results of different samples are reported in TABLE 2.

*Advantages over Existing Positive/Negative Sampling Solutions.* Currently, there are two types of positive and negative sample selection strategies. The first type also builds on top of our CSS [28]. For each sample  $(I, Q)$ , they directly use the  $(I^-, Q)$  and  $(I, Q^-)$  in our CSS (cf. Algorithm 2) as negative samples. The second type is directly randomly replaced an image or question [43] to compose negative samples (*e.g.*,  $(I_{\text{rand}}, Q)$  or  $(I, Q_{\text{rand}})$ ). Instead, we benefit from the ability of CSS, and generate counterfactual samples for positive samples (our first group), *i.e.*, negative samples consists of  $(I_{\text{pos}}^-, Q_{\text{pos}})$  and  $(I_{\text{pos}}, Q_{\text{pos}}^-)$ . Compared to the first solution, our strategy significantly increases the *diversity* of visual contents and questions in positive and negative samples, *i.e.*, in each training epoch, we can use total different counterfactual samples as negative samples. Compared to the second solution, our strategy based on CSS can help models to focus more on fine-grained differences (the masked critical objects

<sup>3</sup>For those samples do not have any sample meets the two requirements, we directly regard the original sample as the candidate positive set, and the proportions are quite small, *e.g.*,  $\approx 1\%$  for VQA-CP v2.

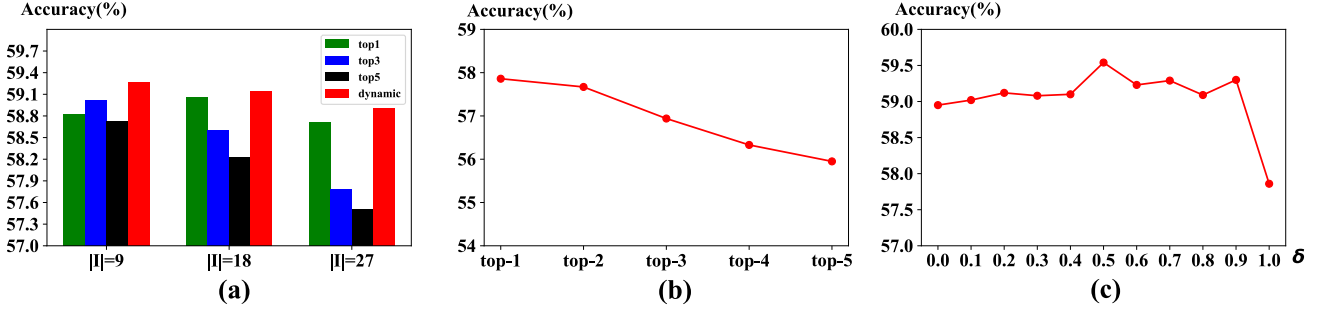


Fig. 5: **Ablations.** Accuracies (%) on VQA-CP v2 test set of different hyperparameters settings of V-CSS<sup>+</sup> or Q-CSS<sup>+</sup>. (a) The results of different size of  $\mathcal{I}$  and critical objects in V-CSS<sup>+</sup>. All results come from model LMH+V-CSS<sup>+</sup>. (b) The results of different size of critical words in Q-CSS<sup>+</sup>. All results come from model LMH+Q-CSS<sup>+</sup>. (c) The results of different proportion  $\delta$  between V-CSS<sup>+</sup> and Q-CSS<sup>+</sup>. All results come from model LMH+V-CSS<sup>+</sup>+Q-CSS<sup>+</sup>.

and words) under the contrastive training.

**Two Variants of Contrastive Loss for VQA.** In this paper, we propose two different types of contrastive loss for VQA: a global version which calculates the similarity (or distance) between anchor samples and positive/negative samples on the whole answer probabilities (dubbed as CR-G), and a local version which calculates the distance based on only ground-truth answer probabilities (dubbed as CR-L).

*Global Contrastive Loss (CR-G).* Given the predicted answer distributions (before sigmoid activation function) of anchor sample, positive sample, and negative samples (denoted as  $P_a$ ,  $P_p$  and  $\{P_n^i\}$  respectively), the CR-G loss is:

$$L_{CR-G} = -\log \left( \frac{e^{s(P_a, P_p)/\tau}}{e^{s(P_a, P_p)/\tau} + \sum_i e^{s(P_a, P_n^i)/\tau}} \right), \quad (8)$$

where  $s(P_i, P_j)$  is the cosine similarity of answer distributions  $P_i$  and  $P_j$ , and  $\tau$  is the hyperparameter temperature.

*Local Contrastive Loss (CR-L).* Instead of calculating similarity based on the whole answer distributions, there is another alternative choice which only focuses on probabilities of ground-truth answers. Thus, the CR-L loss is:

$$L_{CR-L} = -\log \left( \frac{e^{\tilde{P}_a(m)/\tau}}{e^{\tilde{P}_a(m)/\tau} + \sum_i \tilde{P}_n^i(m) \times e^{\tilde{P}_n^i(m)/\tau}} \right), \quad (9)$$

where  $m$  is the answer index of ground-truth answer with the highest target score, and  $\tilde{P}_*(\cdot)$  is the answer probabilities after sigmoid activation function (*i.e.*,  $\tilde{P}_*(\cdot) = \sigma(P_*(\cdot))$ ). In addition, we add  $\tilde{P}_n^i(m)$  as weights which helps models to focus more on hard negative samples with larger  $\tilde{P}_n^i(m)$ .

*Differences between CR-G and CR-L.* Since CR-L only optimize models along a single ground-truth answer direction, it can directly suppress models output over-high scores for negative samples on original ground-truth answers, which is important for vanilla VQA models (*e.g.*, UpDn). But for ensemble-based VQA models (*e.g.*, LMH), since we overlook the accuracy of fused answer distributions (*cf.*  $\hat{P}_{vqa}(a)$  in Eq. (3)), it would be better to use CR-G, which calculates the sample similarity over the global answer distributions.

## 4 EXPERIMENTS

### 4.1 Datasets and Evaluation Metrics

**Datasets.** We evaluated CSST on two challenging diagnostic VQA benchmarks: 1) **VQA-CP** [9]. It is a re-organization of

the training and test sets of widely-used VQA v1 [3] and VQA v2 [4], where the answer distribution of each question type in the training set is made explicitly different from the one in the test set. There are two VQA-CP datasets: VQA-CP v2 and VQA-CP v1. We followed the official splits [9] for both two datasets. We also reported our results on the VQA v2 validation set for more complete comparisons. 2) **GQA-OOD** [10]. It is a fine-grained re-organization of GQA [6]. They share the same training set, but GQA-OOD introduces fine-grained shifts into both the validation and test sets.

**Evaluation Metrics.** For model accuracies on VQA-CP, we followed standard VQA evaluation metric [3]. Similarly, we reported accuracy on all test samples (All) and three different categories separately: Yes/No (Y/N), number counting (Num), and other (Other) categories. For model accuracies on GQA-OOD, there are four metrics: 1) **Acc-A** (all): overall accuracy over all test samples. 2) **Acc-T** (tail): the accuracy on OOD samples (*i.e.*, samples of the tail of the answer class distribution). 3) **Acc-H** (head): the accuracy on in-domain samples (*i.e.*, samples of the head of each local group). 4)  $\Delta = (\text{Acc-H} - \text{Acc-T}) / \text{Acc-T}$ : the error prediction imbalance between frequent (in-domain) and rare (OOD) answers.

### 4.2 Experimental Settings

**Data Preprocessing Details.** For fair comparisons, we did all the same data preprocessing steps with the widely-used UpDn model [14] using the publicly available reimplementation<sup>4</sup>. Specifically, for image feature extraction, we used Faster R-CNN [53] pretrained on VG to extract the top- $K$  objects. Following the convention of prior works, we set  $K = 36$  for VQA-CP and VQA, and  $K = 100$  for GQA-OOD. For question feature extraction, there are slight differences for the models with different backbones. For the UpDn-based models, we set the maximum length of questions as 14, and the word embeddings are initialized with GloVe [50] vectors with the dimension of 300. A single-layer GRU is used to obtain question embedding vectors with the dimension of 1,024. For the LXMERT-based models, we used the official tokenizer in LXMERT to segment each question into word tokens. A pre-trained LXMERT model is used to obtain question features with the dimension of 768. **Training Details and Hyperparameters.** We trained UpDn-based models or LXMERT-based models for 30 epochs with

<sup>4</sup><https://github.com/hengyuan-hu/bottom-up-attention-vqa>



TABLE 1: Accuracies (%) of different VQA architectures on VQA-CP v2 test set and GQA-OOD testdev set. CSST-G and CSST-L denote the CSST with CR-G and CR-L losses, respectively. \* denotes the results from our reimplementation using official codes. Q-CSS<sup>+</sup>/V-CSS<sup>+</sup>/CSS<sup>+</sup> denote the results of this improved version of CSS, and Q-CSS/V-CSS/CSS denote the results from [16]. (a) Ablation studies of different CSST variants and components on UpDn [14] base model. (b) Ablation studies of different CSST variants and components on RUBi [22] base model. (c) Ablation studies of different CSST variants and components on LMH [1] base model.

VQA-CP v2				
Models	All	Y/N	Num	Other
UpDn [14]	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
+CSS	41.16	43.96	12.78	47.48
+CSS <sup>+</sup>	40.84	43.09	12.74	47.37
+CSST-G	41.68	45.70	14.01	47.16
+CSST-L	<b>56.55</b>	<b>80.45</b>	<b>36.29</b>	<b>49.58</b>

GQA-OOD				
Models	Acc-A	Acc-T	Acc-H	$\Delta\downarrow$
UpDn [14]	46.4	42.1	49.1	16.6
Baseline*	46.96	42.90	49.45	15.3
+Q-CSS <sup>+</sup>	46.42	43.37	48.30	11.4
+V-CSS <sup>+</sup>	46.64	43.56	48.53	11.4
+CSS <sup>+</sup>	46.17	<b>44.40</b>	47.26	<b>6.4</b>
+CSST-G	<b>47.78</b>	44.21	<b>49.97</b>	13.0
+CSST-L	47.25	43.37	49.62	14.4

VQA-CP v2				
Models	All	Y/N	Num	Other
RUBi [22]	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	<b>46.95</b>
+CSS	46.67	67.26	11.62	45.13
+CSS <sup>+</sup>	47.46	<b>69.42</b>	12.18	45.63
+CSST-G	47.06	65.91	12.61	46.64
+CSST-L	<b>47.65</b>	62.06	<b>33.46</b>	43.98

GQA-OOD				
Models	Acc-A	Acc-T	Acc-H	$\Delta\downarrow$
RUBi [22]	38.8	35.7	40.8	14.3
Baseline*	45.85	43.37	47.37	<b>9.2</b>
+Q-CSS <sup>+</sup>	47.32	43.18	49.86	15.5
+V-CSS <sup>+</sup>	45.46	41.86	47.66	13.9
+CSS <sup>+</sup>	46.75	42.90	49.11	14.5
+CSST-G	47.64	43.18	50.38	16.7
+CSST-L	<b>48.39</b>	<b>44.31</b>	<b>50.89</b>	14.8

VQA-CP v2				
Models	All	Y/N	Num	Other
LMH [1]	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
+CSS	58.95	84.37	49.42	48.21
+CSS <sup>+</sup>	59.54	83.37	52.57	48.97
+CSST-G	<b>61.66</b>	<b>90.20</b>	<b>54.42</b>	48.69
+CSST-L	55.38	68.01	52.89	<b>49.45</b>

GQA-OOD				
Models	Acc-A	Acc-T	Acc-H	$\Delta\downarrow$
LMH [1]	—	—	—	—
Baseline*	43.96	40.73	45.93	12.8
+Q-CSS <sup>+</sup>	43.45	41.20	44.84	<b>8.8</b>
+V-CSS <sup>+</sup>	42.63	41.02	43.62	6.3
+CSS <sup>+</sup>	44.24	41.20	46.11	11.9
+CSST-G	45.42	42.90	46.97	9.5
+CSST-L	<b>47.93</b>	<b>44.31</b>	<b>50.14</b>	13.2

TABLE 2: Ablation studies of the influence of different positive and negative samples selection strategies on VQA-CP v2. “CSS” denotes the model using only two negative samples from V-CSS<sup>+</sup> and Q-CSS<sup>+</sup> in CR training. “Rand” denotes the model using only the last two randomly composed negative samples in CR training. \* denotes the results from our reimplementation.

Models	CSS	Rand	All	Y/N	Num	Other
Baseline (LMH)			52.45	69.81	44.46	45.54
LMH-CSS			58.95	84.37	49.42	48.21
+CL [28]			59.18	86.99	49.89	47.16
LMH-CSS <sup>+</sup>			59.54	83.37	52.57	48.97
+CL* [28]			59.78	83.85	53.12	<b>49.00</b>
+CR		✓	60.74	89.49	52.05	48.06
+CR	✓		61.13	90.09	53.48	48.06
+CR (CSST-G)	✓	✓	<b>61.66</b>	<b>90.20</b>	<b>54.42</b>	48.69

batch size 512 on all datasets. We used the Adamax algorithm as the optimizer following the public reimplementation<sup>4</sup>. All parameters were initialized from scratch and the random seed was set to 0. Loss weights of  $L_{XE}$ ,  $L_{CR-G}$ , and  $L_{CR-L}$  were set to 1, 1, and 8 in most of experiments.  $\tau$  in the contrastive loss was set to 1 in all experiments. Results about the influence of hyperparameters are reported in Sec. 4.3.1

**Adapting Ensemble-based Methods to GQA-OOD Benchmark.** In original GQA-OOD paper [10], the authors claim none of the typical ensemble-based models (e.g., RUBi [22] and LMH [1]) can improve the Acc-T performance on GQA-OOD, and they even deteriorate Acc-H performance. In this paper, we argue that it is not suitable to apply these debiasing methods on all samples, because “biases” only reside in *imbalanced local groups* [10]. Therefore, in our experiments, we only utilized these debiasing methods on imbalanced local groups. Since the local groups of samples are different in the training and test sets, in the training stage, we only selected the imbalanced local groups from the training set based on the same Shannon entropy and threshold, i.e., we don’t use any extra information about test set in advance.

**Adapting LXMERT Backbone for VQA.** LXMERT [15] itself is a multimodal BERT model, which can be easily used

for VQA. Similar to other multimodal BERT models, after pretraining, it typically utilizes the output of the [CLS] token as a multimodal fused feature, and trains a linear classifier for answer prediction. Unfortunately, this setting is slightly different from the UpDn backbone, whose inputs are both visual and question features. To benefit from the pretrained weights and seamlessly equipped LXMERT into other existing ensemble-based methods with UpDn backbone, in this paper, we utilized the outputs of visual and language embedding tokens as the visual and question features, respectively (i.e., replacing original inputs of ensemble-based methods). The pretrained weights of LXMERT keep fixed. (See results of models with LXMERT in following sections.)

**Adapting SAR [2] strategy to LMH-CSST.** In our experiments, we also equipped the model-agnostic SAR [2] to our LMH-CSST to further improve performance. Specifically, for the Select And Rerank (SAR) module, we used SAR to refer to SAR+LMH (i.e., incorporate LMH into SAR). We chose LMH-CSST as Candidate Answer Selector (CAS), and used top-20 answers as candidates. We utilized the “R  $\rightarrow$  C” strategy to combine question and answer into a synthetic dense caption. We trained SAR for 10 epochs, and batch size was set to 32. For Question Type Discriminator, we selected 1 or 2 candidates for Y/N questions and 12 candidate answers for non-Y/N questions when testing on VQA-CP. Meanwhile, Since we lack manual question type annotations for samples on GQA-OOD, we removed Question Type Discriminator when testing on GQA-OOD, and selected 12 candidates for all questions. We refer readers to SAR [2] for more details.

### 4.3 Ablative Studies

#### 4.3.1 Influence of Different Hyperparameters of CSS

We run a number of ablation studies to analyze the influence of different hyperparameters of CSS<sup>+</sup> (i.e., V-CSS<sup>+</sup> and Q-CSS<sup>+</sup>). Specifically, we conducted all ablations by building

<sup>4</sup>V-CSS<sup>+</sup> and Q-CSS<sup>+</sup> denotes the improved version of V-CSS and Q-CSS, respectively (cf. Sec. 3.2.3).



TABLE 3: Accuracies (%) on VQA-CP v2 test set and VQA v2 val set of state-of-the-art models. The gap (other) represents the accuracy difference between the other category of VQA v2 and VQA-CP v2. “Pre.” represents “Pretraining”, *i.e.*, these models use extra datasets at their pretraining stage (*e.g.*, LXMERT backbone [15] or SAR [2]). “Ann.” represents “Annotation”, *i.e.*, these models rely on extra manual annotations. <sup>†</sup> represents the *ensemble-based* methods, \* indicates the results from our reimplementation using official released codes, <sup>‡</sup> denotes these models use the same adapting strategy for LXMERT backbone mentioned in Sec. 4.2.

Models	Base	Pre.	Ann.	VQA-CP v2 test <sup>†</sup>				VQA v2 val <sup>†</sup>				Gap $\downarrow$ Other
				All	Yes/No	Num	Other	All	Yes/No	Num	Other	
UpDn [14] <sub>CVPR'18</sub>	UpDn			39.74	42.27	11.93	46.05	63.48	81.18	42.14	55.66	9.61
AReg <sup>†</sup> [17] <sub>NeurIPS'18</sub>	UpDn			41.17	65.49	15.48	35.48	62.75	79.84	42.35	55.16	19.58
MuRel [54] <sub>CVPR'19</sub>	UpDn			39.54	42.85	13.17	45.04	—	—	—	—	—
GRL <sup>†</sup> [18] <sub>ACL'19</sub>	UpDn			42.33	59.74	14.78	40.76	51.92	—	—	—	—
RUBi <sup>†</sup> * [22] <sub>NeurIPS'19</sub>	UpDn			45.23	64.85	11.83	44.11	50.56	49.45	41.02	53.95	9.84
SCR [37] <sub>NeurIPS'19</sub>	UpDn			48.47	70.41	10.42	47.29	62.30	77.40	40.90	56.50	9.21
LMH <sup>†</sup> * [1] <sub>EMNLP'19</sub>	UpDn			52.45	69.81	44.46	45.54	61.64	77.85	40.03	55.04	9.50
CVL [55] <sub>CVPR'20</sub>	UpDn			42.12	45.72	12.45	48.34	—	—	—	—	—
Unshuffling [56] <sub>arXiv'20</sub>	UpDn			42.39	47.72	14.43	47.24	61.08	78.32	42.16	52.81	5.57
RandImg [57] <sub>NeurIPS'20</sub>	UpDn			55.37	83.89	41.60	44.20	57.24	76.53	33.87	48.57	4.37
SSL [48] <sub>IJCAI'20</sub>	UpDn			57.59	86.53	29.87	50.03	63.73	—	—	—	—
CSS+CL <sup>†</sup> [28] <sub>EMNLP'20</sub>	UpDn			59.18	86.99	49.89	47.16	57.29	67.27	38.40	54.71	7.55
CF-VQA <sup>†</sup> [24] <sub>CVPR'21</sub>	UpDn			53.55	<b>91.15</b>	13.03	44.97	63.54	82.51	43.96	54.30	9.33
GGE-DQ <sup>†</sup> [25] <sub>ICCV'21</sub>	UpDn			57.32	87.04	27.75	49.59	59.11	73.27	39.99	54.39	4.80
LMH+SAR <sup>†</sup> * [2] <sub>ACL'21</sub>	UpDn	✓		62.51	76.40	<b>59.40</b>	56.09	65.79	77.26	52.71	60.52	4.43
LMH-CSS <sub>CVPR'20</sub>	UpDn			58.95	84.37	49.42	48.21	59.91	73.25	39.77	55.11	6.90
LMH-CSS <sup>+</sup>	UpDn			59.54	83.37	52.57	48.97	59.96	73.69	40.18	54.77	5.80
LMH-CSST	UpDn			61.66	90.20	54.42	48.69	62.37	80.05	39.24	55.04	6.35
LMH-CSST+SAR	UpDn	✓		<b>66.49</b>	86.97	57.95	<b>58.53</b>	<b>69.31</b>	<b>85.85</b>	<b>52.87</b>	<b>61.08</b>	<b>2.55</b>
UpDn <sup>†</sup> * [14] <sub>CVPR'18</sub>	LXMERT	✓		44.14	43.12	17.07	51.66	67.69	83.83	50.80	59.89	8.23
LMH <sup>†</sup> * [1] <sub>EMNLP'19</sub>	LXMERT	✓		59.66	73.41	57.72	52.99	59.57	64.17	47.51	59.27	6.28
LMH-CSS <sup>†</sup>	LXMERT	✓		63.63	84.70	62.12	53.00	58.01	60.37	47.02	59.14	6.14
LMH-CSST <sup>†</sup>	LXMERT	✓		65.71	<b>90.10</b>	<b>63.70</b>	53.48	65.71	80.61	48.26	58.99	5.51
LMH-CSST+SAR <sup>†</sup>	LXMERT	✓		<b>67.49</b>	88.06	56.57	<b>59.71</b>	<b>69.32</b>	<b>84.11</b>	<b>54.90</b>	<b>61.88</b>	<b>2.17</b>
MUTANT [42] <sub>EMNLP'20</sub>	UpDn		✓	61.72	88.90	49.68	50.78	62.56	82.07	42.52	53.28	2.50
MUTANT [42] <sub>EMNLP'20</sub>	LXMERT	✓	✓	69.52	93.15	67.17	57.78	70.24	89.01	54.21	59.96	2.18

TABLE 4: Accuracies (%) on VQA-CP v1 test set of state-of-the-art models. <sup>†</sup> denotes ensemble-based methods. \* indicates the results from our reimplementation. <sup>‡</sup> denotes models use the adapting strategy for LXMERT backbone mentioned in Sec. 4.2.

Models	Base	All	Y/N	Num	Other
GVQA [9] <sub>CVPR'18</sub>	SAN	39.23	64.72	11.87	24.86
UpDn [14] <sub>CVPR'18</sub>	UpDn	39.74	42.27	11.93	46.05
AReg <sup>†</sup> [17] <sub>NeurIPS'18</sub>	UpDn	41.17	65.49	15.48	35.48
GRL <sup>†</sup> [18] <sub>ACL'19</sub>	UpDn	45.69	77.64	13.21	26.97
RUBi <sup>†</sup> * [22] <sub>NeurIPS'19</sub>	UpDn	50.90	80.83	13.84	36.02
LMH <sup>†</sup> * [1] <sub>EMNLP'19</sub>	UpDn	55.27	76.47	26.66	45.68
LMH-CSS <sub>CVPR'20</sub>	UpDn	60.95	85.60	40.57	44.62
LMH-CSS <sup>+</sup>	UpDn	61.66	84.97	41.65	46.51
LMH-CSST	UpDn	63.16	90.58	39.16	45.53
LMH-CSST+SAR	UpDn	<b>69.57</b>	<b>90.81</b>	<b>49.29</b>	<b>56.59</b>
LMH-CSS <sup>†</sup>	LXMERT	66.13	88.89	42.74	52.88
LMH-CSST <sup>†</sup>	LXMERT	67.27	92.04	43.23	52.30
LMH-CSST+SAR <sup>†</sup>	LXMERT	<b>69.75</b>	<b>92.27</b>	<b>44.05</b>	<b>57.69</b>

TABLE 5: Accuracies (%) on GQA-OOD testdev set of state-of-the-art models. \* indicates the results from our reimplementation using official released codes. <sup>‡</sup> denotes models use the adapting strategy for LXMERT backbone mentioned in Sec. 4.2.

Models	Base	Acc-A	Acc-T	Acc-H	$\Delta\downarrow$
MCAN [58] <sub>CVPR'19</sub>	MCAN	50.8	46.5	53.4	14.8
BAN4 [59] <sub>NeurIPS'18</sub>	BAN4	50.2	47.2	51.9	9.9
UpDn <sup>†</sup> * [14] <sub>CVPR'18</sub>	LXMERT	49.61	46.10	51.76	12.3
LMH <sup>†</sup> * [1] <sub>EMNLP'19</sub>	LXMERT	47.78	45.44	49.22	<b>8.3</b>
LMH-CSS <sup>†</sup>	LXMERT	49.21	46.28	51.01	10.2
LMH-CSST <sup>†</sup>	LXMERT	49.75	46.38	51.82	11.7
LMH-CSST+SAR <sup>†</sup>	LXMERT	<b>51.36</b>	<b>48.07</b>	<b>53.38</b>	11.0
UpDn <sup>†</sup> * [14] <sub>CVPR'18</sub>	UpDn	46.96	42.90	49.45	15.3
RUBi <sup>†</sup> * [22] <sub>NeurIPS'19</sub>	UpDn	45.85	43.37	47.37	<b>9.2</b>
LMH <sup>†</sup> * [1] <sub>EMNLP'19</sub>	UpDn	43.96	40.73	45.93	12.8
LMH-CSS <sup>+</sup>	UpDn	44.24	41.20	46.11	11.9
LMH-CSST	UpDn	45.42	42.90	46.97	9.5
LMH-CSST+SAR	UpDn	<b>51.07</b>	<b>47.98</b>	<b>52.97</b>	10.4

on top of a typical ensemble-based VQA model LMH [1]. To disentangle the influence of our contrastive training, we only use the XE loss as training objective and both original samples and counterfactual samples as the inputs (cf. Fig. 2 (b), similar with [16]). All results are illustrated in Fig. 5.

**Size of  $\mathcal{I}$  in V-CSS<sup>+</sup>.** The influence of different size of  $\mathcal{I}$  is shown in Fig. 5 (a). We can observe that the model’s performance gradually decreases with the increase of  $|\mathcal{I}|$ .

**Size of critical objects in V-CSS<sup>+</sup>.** The influence of masking different numbers of critical objects is shown in Fig. 5 (a). We compared dynamic K (cf. Eq. (5)) with some fixed constants

(*e.g.*, 1, 3, 5). From the results, we can observe that the model with dynamic K achieves the best performance.

**Size of critical words in Q-CSS<sup>+</sup>.** The influence of replacing different sizes of critical words is shown in Fig. 5 (b). From the results, we can observe that replacing only one word (*i.e.*, top-1) achieves the best performance.

**Proportion  $\delta$  between V-CSS<sup>+</sup> and Q-CSS<sup>+</sup>.** The influence of different  $\delta$  is shown in Fig. 5 (c). From the results, we can observe that the performance is best when  $\delta = 0.5$ .

In all the following experiments (including experiments on different benchmarks, models with different backbones,

TABLE 6: Quantitative results about the evaluation of the VQA models’ visual-explainable and question-sensitive abilities.

(a) Accuracies (%) on VQA-CP v2 test set.					(b) $AI$ score (%) on VQA-CP v2 test set.				(c) <b>Left:</b> $CS(k)$ (%) on VQA-CP-Rephrasing; <b>Right:</b> $CI$ score (%) on VQA-CP v2 test set.					
Models	All	Y/N	Num	Other	Models	Top-1	Top-2	Top-3	Models	K=1	K=2	K=3	K=4	$CI$
<i>Models with UpDn backbone</i>					<i>Models with UpDn backbone</i>				<i>Models with UpDn backbone</i>					
SCR	48.47	70.41	10.42	47.29	SCR	15.55	13.40	11.97	UpDn	49.94	38.80	31.55	28.08	33.70
LMH	52.45	69.81	44.46	45.54	LMH	17.81	15.19	13.52	LMH	51.68	39.84	33.38	29.11	40.26
+SCR	continued decrease				+V-CSS <sup>+</sup>	<b>20.66</b>	<b>16.71</b>	14.34	+Q-CSS <sup>+</sup>	55.69	43.95	37.22	32.68	57.08
+CSS <sup>+</sup>	59.54	83.37	52.57	<b>48.97</b>	+CSS <sup>+</sup>	18.81	15.73	13.84	+CSS <sup>+</sup>	56.31	44.08	37.00	32.29	51.87
+CSST	<b>61.66</b>	<b>90.20</b>	<b>54.42</b>	48.69	+CSST	19.54	16.37	<b>14.41</b>	+CSST	<b>59.97</b>	<b>49.49</b>	<b>43.37</b>	<b>39.24</b>	<b>58.45</b>
<i>Models with LXMERT backbone</i>					<i>Models with LXMERT backbone</i>				<i>Models with LXMERT backbone</i>					
LMH	59.66	73.41	57.72	52.99	LMH	17.16	15.06	13.60	LMH	61.21	50.33	43.94	39.52	55.42
+CSS <sup>+</sup>	63.63	84.70	62.12	53.00	+CSS <sup>+</sup>	18.90	16.28	14.43	+CSS <sup>+</sup>	61.75	50.62	43.92	39.27	55.44
+CSST	<b>65.71</b>	<b>90.10</b>	<b>63.70</b>	<b>53.48</b>	+CSST	<b>22.68</b>	<b>18.99</b>	<b>16.48</b>	+CSST	<b>63.87</b>	<b>53.72</b>	<b>47.70</b>	<b>43.58</b>	<b>58.22</b>

and models with contrastive training), we used the same best hyperparameter settings for both V-CSS<sup>+</sup> and Q-CSS<sup>+</sup>.

#### 4.3.2 Architecture Generalization of CSST

**Settings.** Since our CSST is a model-agnostic training strategy, which can be seamlessly incorporated into any different VQA architecture. To evaluate the effectiveness of CSST to boost the debiasing performance of different backbones, we incorporated the CSST into multiple architectures including: UpDn [14], RUBi [22], LMH [1]. Especially, RUBi and LMH are ensemble-based methods. For VQA-CP v2, we followed the same settings as prior works. For GQA-OOD, we used the adapting strategies mentioned in Sec. 4.2. All results are shown in TABLE 1. For more clear comparisons, we used superscript <sup>+</sup> to distinguish this improved version of CSS (Q-CSS/V-CSS) with their respective initial counterparts [16].

**Results.** Compared to these baseline models, our CSST (*i.e.*, both CSST-G and CSST-L) can consistently improve the performance for all architectures. For different architectures, the behaviors of CSST-G and CSST-L are slightly different: CSST-G is more suitable for complex ensemble-based models (*e.g.*, 9.21% absolute gains in LMH on VQA-CP v2), and CSST-L is more suitable for vanilla VQA models (*e.g.*, 16.87% absolute gains in UpDn on VQA-CP v2). Meanwhile, when both two types of CSS are used (*i.e.*, Q-CSS<sup>+</sup> and V-CSS<sup>+</sup>), models often achieve better performance than single type CSS. Furthermore, the model with CSST (*i.e.*, CSST-G or CSST-L) always achieves the best performance.

#### 4.3.3 Influence of Positive & Negative Sample Selection

**Settings.** To demonstrate the effectiveness of our proposed samples selection strategy, we compared CSST with a strong baseline CL [28], which has shown effectiveness under our CSS mechanism. Meanwhile, to further show the importance of *counterfactual samples* in the sample selection, we separate these four negative samples into two groups: 1) “CSS”: the first two negative samples from V-CSS<sup>+</sup> and Q-CSS<sup>+</sup>; 2) “Random” (Rand): the last two randomly composed negative samples (cf. NEG\_SEL in Algorithm 4). All results are reported in TABLE 2.

**Results.** Compared to the CL baseline, model with CSST can achieve much better performance (*e.g.*, 61.66% vs. 59.78%), which shows the importance of training sample diversity in contrastive training. Compared to the model with “Rand” negative samples, model with “CSS” negative samples also achieve better performance (*e.g.*, 61.13% vs. 60.74%), which

shows the importance of fine-grained differences in the samples for contrastive training. Meanwhile, when all negative samples are used, the model achieves the best performance.

## 4.4 Comparisons with State-of-the-Arts Models

### 4.4.1 Performance on VQA-CP v2 and VQA v2

**Settings.** We incorporated the CSST-G into model LMH [1], which is dubbed as **LMH-CSST**, and compared it with the state-of-the-art models on both VQA-CP v2 and VQA v2. According to the model framework, we group them into: 1) AReg [17], GRL [18], RUBi [22], LMH [1], CSS+CL [28], CF-VQA [24], GGE-DQ [25], and LMH+SAR [2]. They are all ensemble-based models. 2) UpDn [14], MuRel [54], SCR [37], CVL [55], Unshuffling [56], RandImg [57], and SSL [48]. These models are vanilla VQA models. For complete comparisons, we reported results on two backbones: UpDn [14] and LXMERT [15]. For the UpDn backbone, we followed same settings as prior works [1]. For the LXMERT backbone, we used adapting strategies mentioned in Sec. 4.2. Since SAR [2] is another model-agnostic model, which is orthogonal to our contributions, *i.e.*, we can also equip LMH-CSST with SAR (dubbed as **LMH-CSST+SAR**) to further boost performance. Moreover, following suggestions from [57], we also reported the performance gap on the *Other* category.

**Results.** The results are reported in TABLE 3. When trained and tested on the VQA-CP v2 dataset (*i.e.*, the left side of TABLE 3), LMH-CSST+SAR achieves a new SOTA performance on both UpDn and LXMERT backbones, with 66.49% and 67.49% accuracies, respectively. Compared to all models without extra pretraining or annotations (*i.e.*, “Pre.” and “Ann.” in TABLE 3), LMH-CSST still achieves SOTA performance with 61.66% accuracies. Particularly, CSST improves the performance of LMH with 9.21% (61.66% vs. 52.45%) and 6.05% (65.71% vs. 59.66%) absolute gains on UpDn and LXMERT backbones, respectively. When trained and tested on the VQA v2 dataset (*i.e.*, the right side of TABLE 3), CSST consistently improves the performance of LMH with 0.73% (62.37% vs. 61.64%) and 6.14% (65.71% vs. 59.57%) absolute gains on UpDn and LXMERT backbones, respectively. Different from previous SOTA models that suffer severe performance drops between VQA-CP v2 and VQA v2 (*e.g.*, 9.50% and 6.28% on LMH), LMH-CSST+SAR can significantly decrease the performance gap into 2.55% and 2.17%, which demonstrates that the effectiveness of CSST to reduce language biases.














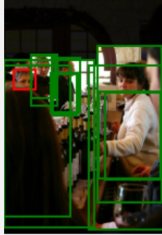


Input Question	Input Image	LMH	LMH-CSS	LMH-CSST
Q: What color is the tail of the plane?		 gray ✗	 gray ✗	 blue ✓
Q: What are the people doing?		 walking ✗	 walking ✗	 horseback riding ✓
Q: How many buses are there?		 2 ✗	 1 ✓	 1 ✓
Q: What color shirt is the woman on the right wearing?		 tan ✗	 white ✓	 white ✓

Fig. 6: Examples of visual-explainable ability from VQA-CP v2 test set. The **green** boxes denote their scores  $s(\hat{a}, v) > 0$ , *i.e.*, positive contributions to final predictions. The **red** boxes denote their scores  $s(\hat{a}, v) < 0$ , *i.e.*, negative contributions to final predictions. Only objects which are highly related to the QA pair are shown (*i.e.*,  $STM \geq 0.6$ ). LMH-CSS is the model with only XE loss [16]

#### 4.4.2 Performance on VQA-CP v1

**Settings.** We also compared the LMH-CSST with state-of-the-art models on the VQA-CP v1. Similarly, we group these models into: 1) AReg [17], GRL [18], RUBi [22] and LMH [1] are all ensemble-based models. 2) GVQA [9] and UpDn are vanilla VQA models, and GVQA is with SAN [13] backbone. All settings are same as the ones on VQA-CP v2.

**Results.** The results are reported in TABLE 4. Compared to baseline models, both LMH-CSST and LMH-CSST+SAR achieve new state-of-the-art performance on two different backbones over all metrics. Particularly, the CSST improves the performance of LMH with a 7.89% (63.16% vs. 55.27%) absolute performance gains on UpDn backbones.

#### 4.4.3 Performance on GQA-OOD

**Settings.** We further compared LMH-CSST with state-of-the-art models on GQA-OOD. Similarly, we group these models into 1) RUBi [22] and LMH [1] are ensemble-based models. 2) MCAN [58], BAN4 [59], and UpDn are vanilla VQA models. All settings are same as experiments on VQA-CP. To adapt these ensemble-based methods to GQA-OOD, we used the adapting strategy mentioned in Sec. 4.2.

**Results.** The results are reported in TABLE 5. For fair comparisons, the baselines (UpDn and LMH) with the LXMERT backbones were reimplemented by using the same adapting

strategy. Compared to baselines, LMH-CSST+SAR achieves new state-of-the-art performance on most of the metrics on both two backbones. Especially for the most important metrics Acc-T [10], our achieve the best performance over others (*e.g.*, 47.98% vs. 43.37% and 48.07% vs. 47.2%).

### 4.5 Improving Visual-Explainable Ability

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

#### 4.5.1 CSST vs. SCR (Q1)

**Settings.** We equipped the existing state-of-the-art visual-explainable model SCR [37] into the LMH framework, and compared it with CSST. Results are reported in TABLE 6 (a).

**Results.** Since the training of all SOTA visual-explainable models (*e.g.*, SCR [37], HINT [36]) 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, the proposed CSST can consistently improve performance on different backbones.







	Q: What is the name of the computer company? A: dell LMH + CSS	Q: What is the name of the computer company? A: dell LMH
	Q: What is the name of the computer company? A: dell LMH + CSS	Q: What is the name of the computer company? A: apple LMH + CSST
	Q: Is the rider a child or an adult? A: child LMH + CSS	Q: Is the rider a child or an adult? A: child LMH
	Q: Is the rider a child or an adult? A: child LMH + CSS	Q: Is the rider a child or an adult? A: adult LMH + CSST
	Q: How many buses are shown? A: 1 LMH + CSS	Q: How many buses are shown? A: 2 LMH
	Q: How many buses are shown? A: 1 LMH + CSS	Q: How many buses are shown? A: 1 LMH + CSST
	Q: What is this object? A: book LMH + CSS	Q: What is this object? A: laptop LMH
	Q: What is this object? A: book LMH + CSS	Q: What is this object? A: book LMH + CSST

Fig. 7: Visualization examples of question-sensitive ability from VQA-CP v2 test set. Different shades of green color in the question denotes the relative values of  $s(\hat{a}, w)$ , i.e., the word with darker green denotes the word has larger contribution to final predictions.

#### 4.5.2 Evaluations of Visual-Explainable Ability (Q2)

**Settings.** We evaluated 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  $STM$  scores (cf. IO\_SEL in Sec. 3.2.1) as pseudo ground-truths. Thus, we design a new metric *Average Importance* ( $\mathcal{AI}$ ): the average  $STM$  score of the top-K objects with highest  $|s(a, v)|$ . We formally define  $\mathcal{AI}$  as:

$$\mathcal{AI} = \frac{\sum_{(I, Q)} [\mathbf{1}(a = \hat{a}) \cdot \sum_k STM_k^{(I, Q)}]}{\sum_{(I, Q)} 1} \quad (9)$$

where  $k$  is the index of top-K objects,  $STM_k^{(I, Q)}$  is the  $STM$  score of k-th object for the sample  $(I, Q)$ , and  $\mathbf{1}$  is an indicator function. The results are shown in TABLE 6 (b). We further demonstrate some qualitative results in Fig. 6.

**Results.** From TABLE 6 (b), we can observe that CSST dramatically improves  $\mathcal{AI}$  scores, which means the influential objects for predictions are more related to the QA pairs. From Fig. 6, we can find that both CSS (LMH vs. LMH-CSS) and CST (LMH-CSS vs. LMH-CSST) help the model to make predictions based on critical objects (i.e., green boxes) and suppress the influence of irrelevant objects (red boxes).

## 4.6 Improving Question-Sensitive Ability

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

#### 4.6.1 Robustness to Rephrasings of Questions (Q3)

**Settings.** As discussed in previous work [39], 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-splitted the existing dataset VQA-Rephrasings [39] 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 6 (c) (left). We refer readers to [39] for more details about the VQA-Rephrasings and metric  $CS(k)$ . **Results.** From TABLE 6 (c), we can observe that Q-CSS<sup>+</sup> dramatically improves the robustness to diverse rephrasings of questions. Furthermore, V-CSS<sup>+</sup> can help to further improve the robustness, i.e., CSS<sup>+</sup> achieves better performance.

#### 4.6.2 Evaluations of Question-Sensitive Ability (Q4)

**Settings.** We evaluated the effectiveness of CSST to improve the question-sensitive ability on 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  $(I, Q^*, a)$ <sup>6</sup>. Then we fed both two samples into the evaluated model, and calculated the confidence decreases of the ground-truth answer. We formally define  $\mathcal{CI}$  as:

$$\mathcal{CI} = \frac{\sum_{(I, Q)} \mathbf{1}(P_{vqa}(a|I, Q) > P_{vqa}(a|I, Q^*)) \cdot \mathbf{1}(a = \hat{a})}{\sum_{(I, Q)} 1} \quad (9)$$

<sup>6</sup>Test set is released at: <https://github.com/yanxinzju/CSS-VQA>.



where  $\hat{a}$  is the predicted answer for sample  $(I, Q)$ ,  $\mathbf{1}$  is an indicator function. The results are reported in TABLE 6 (c). We further demonstrate some qualitative results in Fig. 7.

**Results.** From TABLE 6 (c), we can observe that CSST 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 Fig. 7, we can find that both CSS (LMH vs. LMH-CSS) and CST (LMH-CSS vs. LMH-CSST) help the model to make predictions based on critical words (*e.g.*, “bus” or “objects”), *i.e.*, forcing the model to understand the whole questions before making predictions.

## 5 CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a model-agnostic Counterfactual Samples Synthesizing and Training (CSST) strategy to improve the VQA model’s visual-explainable and question-sensitive abilities. The CSST is composed of two main components: CSS and CST, where CSS generates counterfactual training samples by masking critical objects or words, and CST consists of a XE training loss and a contrastive training loss. The CSST (*i.e.*, CSS and CST) can consistently boost the performance of different VQA models. We validate the effectiveness of CSST through extensive comparative and ablative studies on three benchmarks and multiple different backbones. Moving forward, we are going to 1) extend CSST to other visual-language tasks that suffer severe language biases or other types of biases; 2) design a specific VQA backbone that benefits from the design of CSST.

## REFERENCES

- [1] C. Clark, M. Yatskar, and L. Zettlemoyer, “Don’t take the easy way out: Ensemble based methods for avoiding known dataset biases,” in *Conference on Empirical Methods in Natural Language Processing*, 2019.
- [2] Q. Si, Z. Lin, M. Zheng, P. Fu, and W. Wang, “Check it again: Progressive visual question answering via visual entailment,” in *Annual Meeting of the Association for Computational Linguistics*, 2021.
- [3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, “Vqa: Visual question answering,” in *IEEE International Conference on Computer Vision*, 2015.
- [4] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [5] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “Clevr: A diagnostic dataset for compositional language and elementary visual reasoning,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [6] D. A. Hudson and C. D. Manning, “Gqa: A new dataset for real-world visual reasoning and compositional question answering,” in *IEEE conference on computer vision and pattern recognition*, 2019.
- [7] A. Agrawal, D. Batra, and D. Parikh, “Analyzing the behavior of visual question answering models,” in *Conference on Empirical Methods in Natural Language Processing*, 2016.
- [8] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, “Yin and yang: Balancing and answering binary visual questions,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [9] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, “Don’t just assume; look and answer: Overcoming priors for visual question answering,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [10] C. Kervadec, G. Antipov, M. Baccouche, and C. Wolf, “Roses are red, violets are blue... but should vqa expect them to?” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.
- [11] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Neural module networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [12] 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 *Conference on Empirical Methods in Natural Language Processing*, 2016.
- [13] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [14] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [15] H. Tan and M. Bansal, “Lxmert: Learning cross-modality encoder representations from transformers,” in *2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [16] L. Chen, X. Yan, J. Xiao, H. Zhang, S. Pu, and Y. Zhuang, “Counterfactual samples synthesizing for robust visual question answering,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [17] S. Ramakrishnan, A. Agrawal, and S. Lee, “Overcoming language priors in visual question answering with adversarial regularization,” in *Conference on Neural Information Processing Systems*, 2018.
- [18] G. Grand and Y. Belinkov, “Adversarial regularization for visual question answering: Strengths, shortcomings, and side effects,” in *Annual Meeting of the Association for Computational Linguistics workshop*, 2019.
- [19] Y. Belinkov, A. Poliak, S. M. Shieber, B. Van Durme, and A. M. Rush, “Don’t take the premise for granted: Mitigating artifacts in natural language inference,” in *Annual Meeting of the Association for Computational Linguistics*, 2019.
- [20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Conference on Neural Information Processing Systems*, 2014.
- [21] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang, “Zero-shot visual recognition using semantics-preserving adversarial embedding networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [22] R. Cadene, C. Dancette, H. Ben-younes, M. Cord, and D. Parikh, “Rubi: Reducing unimodal biases in visual question answering,” in *Conference on Neural Information Processing Systems*, 2019.
- [23] R. K. Mahabadi and J. Henderson, “Simple but effective techniques to reduce biases,” in *arXiv*, 2019.
- [24] Y. Niu, K. Tang, H. Zhang, Z. Lu, X.-S. Hua, and J.-R. Wen, “Counterfactual vqa: A cause-effect look at language bias,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.
- [25] X. Han, S. Wang, C. Su, Q. Huang, and Q. Tian, “Greedy gradient ensemble for robust visual question answering,” in *IEEE International Conference on Computer Vision*, 2021.
- [26] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, “Right for the right reasons: Training differentiable models by constraining their explanations,” in *International Joint Conference on Artificial Intelligence*, 2017.
- [27] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” in *arXiv*, 2020.
- [28] Z. Liang, W. Jiang, H. Hu, and J. Zhu, “Learning to contrast the counterfactual samples for robust visual question answering,” in *2020 Conference on Empirical Methods in Natural Language Processing*, 2020.
- [29] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, “Vlbert: Pre-training of generic visual-linguistic representations,” in *International Conference on Learning Representations*, 2020.
- [30] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, “Oscar: Object-semantics aligned pre-training for vision-language tasks,” in *European Conference on Computer Vision*, 2020.
- [31] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, “Uniter: Universal image-text representation learning,” in *European Conference on Computer Vision*, 2020.
- [32] A. Jabri, A. Joulin, and L. Van Der Maaten, “Revisiting visual question answering baselines,” in *European Conference on Computer Vision*, 2016.
- [33] T. Qiao, J. Dong, and D. Xu, “Exploring human-like attention supervision in visual question answering,” in *AAAI Conference on Artificial Intelligence*, 2018.
- [34] C. Liu, J. Mao, F. Sha, and A. Yuille, “Attention correctness in neural image captioning,” in *AAAI Conference on Artificial Intelligence*, 2017.

- [35] Y. Zhang, J. C. Niebles, and A. Soto, "Interpretable visual question answering by visual grounding from attention supervision mining," in *Winter Conference on Applications of Computer Vision*, 2019.
- [36] R. R. Selvaraju, S. Lee, Y. Shen, H. Jin, D. Batra, and D. Parikh, "Taking a hint: Leveraging explanations to make vision and language models more grounded," in *IEEE International Conference on Computer Vision*, 2019.
- [37] J. Wu and R. J. Mooney, "Self-critical reasoning for robust visual question answering," in *Conference on Neural Information Processing Systems*, 2019.
- [38] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *IEEE International Conference on Computer Vision*, 2017.
- [39] M. Shah, X. Chen, M. Rohrbach, and D. Parikh, "Cycle-consistency for robust visual question answering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [40] V. Agarwal, R. Shetty, and M. Fritz, "Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing," in *arXiv*, 2019.
- [41] J. Pan, Y. Goyal, and S. Lee, "Question-conditioned counterfactual image generation for vqa," in *arXiv*, 2019.
- [42] T. Gokhale, P. Banerjee, C. Baral, and Y. Yang, "Mutant: A training paradigm for out-of-distribution generalization in visual question answering," in *Conference on Empirical Methods in Natural Language Processing*, 2020.
- [43] Y. Kant, A. Moudgil, D. Batra, D. Parikh, and H. Agrawal, "Contrast and classify: Training robust vqa models," in *IEEE International Conference on Computer Vision*, 2021.
- [44] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," in *arXiv*, 2018.
- [45] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," in *arXiv*, 2018.
- [46] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [47] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*, 2020.
- [48] X. Zhu, Z. Mao, C. Liu, P. Zhang, B. Wang, and Y. Zhang, "Overcoming language priors with self-supervised learning for visual question answering," in *International Joint Conference on Artificial Intelligence*, 2020.
- [49] M. Honnibal and I. Montani, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *To appear*, 2017.
- [50] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing*, 2014.
- [51] S. Jain and B. C. Wallace, "Attention is not explanation," in *NA Annual Meeting of the Association for Computational Linguistics*, 2019.
- [52] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee, "12-in-1: Multi-task vision and language representation learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [53] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015.
- [54] R. Cadene, H. Ben-Younes, M. Cord, and N. Thome, "Murel: Multimodal relational reasoning for visual question answering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [55] E. Abbasnejad, D. Teney, A. Parvaneh, J. Shi, and A. v. d. Hengel, "Counterfactual vision and language learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [56] D. Teney, E. Abbasnejad, and A. v. d. Hengel, "Unshuffling data for improved generalization," in *arXiv*, 2020.
- [57] D. Teney, K. Kafle, R. Shrestha, E. Abbasnejad, C. Kanan, and A. v. d. Hengel, "On the value of out-of-distribution testing: An example of goodhart's law," in *Conference on Neural Information Processing Systems*, 2020.
- [58] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [59] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *Conference on Neural Information Processing Systems*, 2018.