Counterfactual Samples Synthesizing for Robust Visual Question Answering

Long Chen1*  Xin Yan1*  Jun Xiao1†  Hanwang Zhang2  Shiliang Pu3  Yueting Zhuang1
1DCD Lab, College of Computer Science, Zhejiang University  2MReaL Lab, Nanyang Technological University  3Hikvision 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 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-

1*Long Chen and Xin Yan are co-first authors with equal contributions.
†Corresponding author.
1Codes: https://github.com/yanxinzju/CSS-VQA
Figure 2: (a): A training sample from the VQA-CP. (b): The synthesized training sample by V-CSS. It masks critical 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”).

In this paper, we propose a novel model-agnostic Counterfactual 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 question 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 straightforward 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 \( \mathcal{D} = \{I_i, Q_i, \alpha_i\}_i^N \) consisting of triplets of images \( I_i \in \mathcal{I} \), questions \( Q_i \in \mathcal{Q} \) and answers \( \alpha_i \in \mathcal{A} \), VQA task learns a mapping \( f_{vqa} : \mathcal{I} \times \mathcal{Q} \rightarrow \{0, 1\}^{|\mathcal{A}|} \), 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: \( V = \{v_1, ..., v_{n_v}\} \), 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: \( Q = \{w_1, ..., w_{n_q}\} \), where \( w_j \) is \( j \)-th word feature. Then both \( V \) and \( Q \) are fed into the model \( f_{vqa} \) to predict answer distributions:

\[
P_{vqa}(\alpha|I, Q) = f_{vqa}(V, Q).
\]

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 subtypes: 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 \( Q \) as input and predicts answer distribution:

\[
P_q(\alpha|Q) = f_q(Q).
\]

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

\[
\hat{P}_{vqa}(\alpha|I, Q) = M(P_{vqa}(\alpha|I, Q), P_q(\alpha|Q)).
\]

In the training stage, the XE loss is computed based on the fused answer distribution \( \hat{P}_{vqa}(\alpha) \) 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 \( VQA \) model, given a training sample \( (I, Q, \alpha) \), CSS consists of three main steps:

1. Training \( VQA \) model with original sample \( (I, Q, \alpha) \);
2. Synthesizing a counterfactual sample \( (I^-, Q, a^-) \) by V-CSS or \( (I, Q^-, a^-) \) by Q-CSS;
3. Training \( VQA \) 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 \( \delta \) is the trade-off weight (See Figure 4 (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 (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 \(C\), and assume all objects in \(C\) 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 \(Sim\). We select \(|C|\) objects with the highest \(Sim\) scores as \(C\).

2. Object Local Contributions Calculation. After obtaining the object set \(C\), 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:

\[
s(a, v_i) = S(P_{vqa}(a), v_i) := (\nabla_{v_i} P_{vqa}(a))^T 1,
\]

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

Algorithm 2 Counterfactual Samples Synthesizing

1: function CSS(I, Q, a)
2: \(V, Q, P_{vqa}(a) \leftarrow VQA(I, Q, a, True)\)
3: \(cond \sim U[0, 1]\)
4: if \(cond \geq \delta\) then \(\triangleright\) execute V-CSS
5: \(I \leftarrow IO_SEL(I, Q)\)
6: \(i, v_i \leftarrow S(P_{vqa}(a), v_i)\)
7: \(I^+, I^- \leftarrow CO_SEL(I, \{s(a, v_i)\})\)
8: \(a^- \leftarrow DA_ASS(I^+, Q, VQA(a))\)
9: \(VQA(I^-, Q, a^-, False)\)
10: else \(\triangleright\) execute Q-CSS
11: \(s(a, w_i) \leftarrow S(P_{vqa}(a), w_i)\)
12: \(Q^+, Q^- \leftarrow CW_SEL(\{s(a, w_i)\})\)
13: \(a^- \leftarrow DA_ASS(I, Q^+, VQA(a))\)
14: \(VQA(I, Q^-, a^-, False)\)
15: end if
16: end function

Algorithm 3 Dynamic Answer Assigning

1: function DA_ASS(I^+, Q, VQA, a)
2: \(VQA.eval()\) \(\triangleright\) don’t update parameters
3: \(a^+ \leftarrow VQA(I^+, Q^+, a, True)\)
4: \(a^- \leftarrow \text{top-N}(\text{argsort}_{a\in A}(P_{vqa}(a)))\)
5: \(a^- := \{a_i | a_i \in a, a_i \notin a^+\} \triangleright a\) is gt answer set
6: return \(a^-\)
7: end function

Figure 3: An informal illustration example of the \(I^+, I^-, Q^+,\) and \(Q^-\) in 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 \(s(a, v_i)\) for all objects in \(C\), 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):

\[
\sum_{v_i \in I^+} \exp(s(a, v_i))/\sum_{v_j \notin I} \exp(s(a, v_j)) > \eta,
\]

where \(\eta\) 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., \(I^- = I \setminus I^+\). 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 \(VQA\) model, and obtain the predicted answer distribution \(P_{vqa}(a)\). Based on \(P_{vqa}(a)\), we select the top-N answers with highest predicted probabilities as \(a^+\). Then we define \(a^- := \{a_i | a_i \in a, a_i \notin a^+\}\).

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 an empty set, 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 critical 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 as:

$$s(a, \omega_i) = S(P_{vqa}(a), \omega_i) := (\nabla \omega P_{vqa}(a))^T 1. \quad (6)$$

2. Critical Words Selection (CW_SEL). In this step, we first extract question-type words for each question $Q$ (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 $I$ in V-CSS. The influence of different size of $I$ is shown in Figure 4 (a). We can observe that the model’s performance gradually decreases with the increase of $|I|$. 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 (b). From the results, we can observe that replacing only one word (i.e., top-1) achieves the best performance.

The proportion $\delta$ of V-CSS and Q-CSS. The influence of different $\delta$ 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. CSS denotes the model with both V-CSS and Q-CSS. It represents these results are based on our reimplementation.

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], and compared it with the
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% absolute 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% absolute 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 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 $S_{AI}$, $M$ score (Section 3.2.1 IO, Sel) as pseudo ground truth. Thus, we design a new metric Average Importance ($A_{E}$): the average $S_{AI}$, $M$ score of the top-K objects with highest $|s(a, v)|$. The results are shown in Table 4 (b).
For qualitative results, we illustrated in Figure 5 (a).

**Results.** From Table 4 (b), we can observe that CSS dramatically improves the AT 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., 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 (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 samples into the evaluated model, and calculate the confidence decreases of the ground-truth answer. We formally define $CI$ in Eq. 7:

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

(7)

---

4The auxiliary test set is released in: [github.com/yanxinzju/CSS-VQA](https://github.com/yanxinzju/CSS-VQA)
where \( \hat{a} \) is the model predicted answer for sample \((I, Q)\), 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 improve 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.

**Acknowledgement** This work was supported by National Key Research & Development Project of China (No.2018YFA0101900), National Natural Science Foundation of China (U19B2043, 61976185), Zhejiang Natural Science Foundation (LR19F020002, LZ17F020001), Fundamental Research Funds for the Central Universities and Chinese Knowledge Center for Engineering Sciences and Technology. Long Chen was supported by 2018 ZJU Academic Award for Outstanding Doctoral Candidates.
References


[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.

[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


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


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