Greedy Gradient Ensemble for Robust Visual Question Answering

Xinzhe Han Shuhui Wang Chi Su Qingming Huang Qi Tian

This supplementary document is organized as follows:

- Section A introduces A.1 GGE for Sigmoid+BCE loss (Section 4.1); A.2 GGE for Softmax+CE loss (Section 5.4); A.3 algorithm for GGE-iter and GGE-tog (Section 4.1).

- Section B provides more detailed settings for CGR, CGW, and CGD (Section 5.1).

- Section C provides C.1 implementation details for the base model; C.2 and ablations for ensemble strategy, SUM-DQ and LMH+RUBi (Section 5.3).

- Section D provides D.1 ablation studies for base model S-MRL and BAN (Section 5.3); D.2 comparison between Self-Ensemble fashion GGE and RUBi; D.3 additional experimental results (Section 3.2 and 5.1), including Accuracy on VQA v2 and CGR/CGD for all implemented methods.

- Section E provides more quantitative examples and failure cases from GGE-DQ (Section 5.4).

A. Implementation Details for GGE

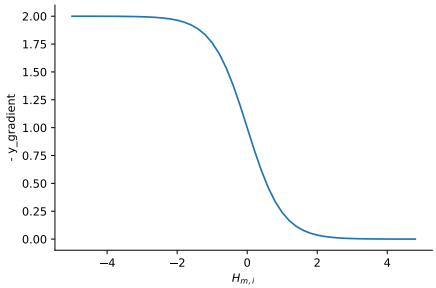
A.1. Sigmoid+BCE

For classification problem with BCE loss, the negative gradient is shown in Eq. 7 in the main paper

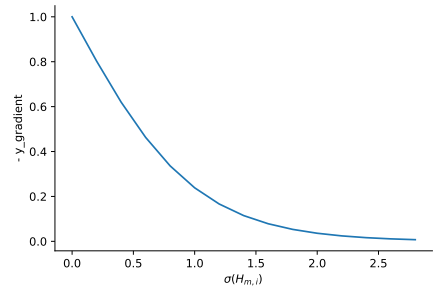
$$-\nabla\mathcal{L}(\mathcal{H}_{m,i}) = 2y_{m,i}\sigma\left(-2y_{m,i}\mathcal{H}_{m,i}\right). \quad (1)$$

If $y_{m,i} = 0$, the gradient will always be 0. If the label $y_{m,i} = 1$, we plot the change of negative gradient versus prediction $\mathcal{H}_{m,i}$.

As shown in Figure 1, the gradient will continuously decrease when biased models can predict the right answers with higher confidence. This means the base model will pay more attention to samples that are hard to solve by biased models.



(a) $-\nabla\mathcal{L}(\mathcal{H}_{m,i})$ vs. $\mathcal{H}_{m,i}$



(b) $-\nabla\mathcal{L}(\mathcal{H}_{m,i})$ vs. $\sigma(\mathcal{H}_{m,i})$

Figure 1. **Negative Gradient versus Predictions**

In practice, we clip $\sigma(\mathcal{H}_{m,i}) > 0$ with Sigmoid function, to make the range of $-\nabla\mathcal{L}(\mathcal{H}_{m,i})$ consistent with the label space [0,1] of BCE loss. B_d is a statistic answer distribution of the training set, which satisfies $B_d > 0$. Therefore, we do not need to add Sigmoid function on distribution bias in Eq.11-15 in the main paper.

However, clipping the gradient does not directly increase the scale of hard samples but only lowers the scale of easy ones, resulting in performance degradation on VQA v2. Actually, for the hard samples, the gradient can be up to 2.0 without clip

operation. If we can design a new classification loss with label space $[0, 2]$ in place of BCE, it may be an alternative approach to deal with this problem.

A.2. Softmax+CE

We provide GGE optimized with Softmax+CE loss in Section 5.4. The loss function can be written as

$$\mathcal{L}(Z, Y) = -\sum_{i=1}^{C} y_i \log(\sigma_i), \tag{2}$$

with

$$\sigma_i = \frac{e^{z_i}}{\sum_{j=1}^{C} z_j}, \tag{3}$$

where $Z = \{z_i\}_{i=1}^{C}$ is the predicted logits, C is the number of classes, and $y_i \in [0, 1]$ is the ground truth labels. The negative gradient of loss function is

$$-\nabla\mathcal{L}(z_i) = y_i - \sigma_i. \tag{4}$$

Similar to implementation of Sigmoid+BCE, we directly clip the $\nabla L(z_i)$ to the label space [0,1]

$$-\nabla\hat{\mathcal{L}}(z_i) = \begin{cases} y_i - \sigma_i & y_i > 0 \\ 0 & y_i = 0 \end{cases}. \tag{5}$$

As a result, if $y_i = 0$ the pseudo label $\hat{\mathcal{L}}(z_i)$ will still be 0, otherwise, it will decrease when biased models can predict the right answer with higher confidence. The optimization process is the same with that in Sigmoid+BCE. Additionally, since the statistical distribution $B_d \in (0, 1)$, we treat $\sigma_i = B_{d_i}$ when calculate the gradient in GGE-D and GGE-DQ.

A.3. GGE-Iter and GGE-tog

In Section 4.1 we provide two optimization schemes GGE-iteration and GGE-together. The detailed implementation is shown in Algorithm 1 and 2. Two variants of implementation do not show an obvious performance gap in most experiments.

Algorithm 1: GGE-iteration

Input: Observations X, Lables Y,
Biased features Observations $\mathcal{B} = \{B_m\}_{m=1}^{M}$,
Base function $f(.|\theta) : X \to \mathbb{R}^{|Y|}$,
Bias functions $\{h_m(.|\phi_i) : B_i \to \mathbb{R}^{|Y|}\}_{m=1}^{M}$
Initialize: $\mathcal{H}_0 = 0$;
for *Batch* $t = 1 \dots T$ **do**
  for $m = 1 \dots M$ **do**
    $L_m(\phi_m) \leftarrow$
    $\mathcal{L}'\left(h_m(B_m; \phi_m), -\nabla\mathcal{L}(H_{m-1}, Y)\right)$
    Update $\phi_m \leftarrow \phi_m - \alpha\nabla_{\phi_m} L_m(\phi_m)$
  end
  $L_{M+1}(\theta) \leftarrow$
  $\mathcal{L}'\left(f(X; \theta), -\nabla\mathcal{L}(H_M, Y)\right)$
  Update $\theta \leftarrow \theta - \alpha\nabla_{\theta} L_{M+1}(\theta)$
end
return $Y = f(X; \theta)$

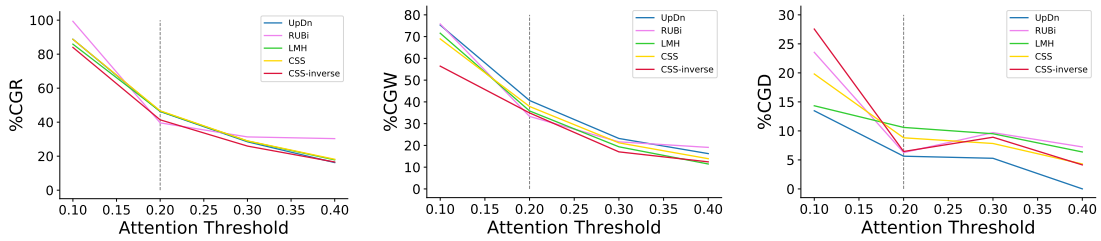
Algorithm 2: GGE-together

Input: Observations X, Lables Y,
Biased features Observations $\mathcal{B} = \{B_m\}_{m=1}^{M}$,
Base function $f(.|\theta) : X \to \mathbb{R}^{|Y|}$,
Bias functions $\{h_m(.|\phi_i) : B_i \to \mathbb{R}^{|Y|}\}_{m=1}^{M}$
Initialize: $\mathcal{H}_0 = 0$;
for *Batch* $t = 1 \dots T$ **do**
  for $m = 1 \dots M$ **do**
    $L_m(\phi_m) \leftarrow$
    $\mathcal{L}'\left(h_m(B_m; \phi_m), -\nabla\mathcal{L}(H_{m-1}, Y)\right)$
  end
  $L_{M+1}(\theta) \leftarrow$
  $\mathcal{L}'\left(f(X; \theta), -\nabla\mathcal{L}(H_M, Y)\right)$
  $L(\Theta) \leftarrow \sum_{m=1}^{M+1} L_m$
  Update $\Theta \leftarrow \Theta - \alpha\nabla_{\Theta} L(\Theta)$
end
return $Y = f(X; \theta)$

B. Details for CGD

First, we should stress that CGD *only* evaluates whether the visual information is taken for answer prediction, which is *parallel* with Accuracy and different from metrics in Referring Expression and Visual Grounding tasks. It is proposed to help quantitatively evaluate models' grounding ability.

CGD considers the top-*N* most sensitive visual region. In this paper, we evaluate the sensitivity via attention. In Figure 2, we plot change of CGR, CGW and CGD with different threshold for prevailing methods UpDn [1], RUBi [2], LMH [4] C-



(a) CGR (b) CGW (c) CGD

Figure 2. **CGR, CGW and CGD versus attention threshold for prevailing methods**.

SS [3] and CSS-V$_{inv-hat}$. We set attention threshold $t \in \{0.1, 0.2, 0.3, 0.4\}$, which indicates that top-*N* is no more than $\{9, 4, 3, 2\}$.

We choose to consider top-4 ($t = 0.2$) objects for CGD, since many questions need to consider multiple objects and $t = 0.2$ is the most discriminative threshold as shown in Figure 2(c). Apart from attention, Grad-CAM [9] can be an alternative for grounding evaluation.

C. Implementation Details for Experiments

C.1. Base Model

We use the publicly available reimplementation of UpDn[1] [1] for our baseline architecture, data preprocess and optimization.

Image Encoder. Following the popular bottom-up attention mechanism [1], we use a Faster R-CNN [8] based framework to extract visual features. We select the top-36 region proposals for each image $\mathbf{v} \in \mathbb{R}^{36\times2048}$.

Question Encoder. Each word is first initialized by 300-dim GloVe word embeddings [7], then fed into a GRU with 1024-d hidden vector. The question representation is the last state of GRU $h_T \in \mathbb{R}^{1024}$.

Multi-modal Fusion. We use traditional linear attention between h_T and $\mathbf{v}$ for visual representation. and the final representation for classification is the Hadamard product of vision and question representation.

Question-only Classifier. The question-only classifier is implemented as two fully-connected layers with ReLU activations. The input question representation is shared with that in VQA base model.

Question types. We use 65 question types annotated in VQA v2 and VQA-CP, according to the first few words of the question (e.g., "What color is"). To save the training time, we simply use statistic answer distribution conditioned by question type in the train set as the prediction of distribution bias.

Optimization. Following UpDn [1], all the experiments are conducted with the Adamax optimizer for 20 epochs with learning rate initialized as 0.001. We train all models on a single RTX 3090 GUP with PyTorch 1.7 [6] and batch size 512.

Data Preprocessing. Following previous works, we filter the answers that appear less than 9 times in the train set. For each instance with 10 annotated answers, we set the scores for labels that appear 1/2/3 times as 0.3/0.6/0.9, more than 3 times as 1.0.

C.2. Ablations for Ensemble

SUM-DQ. SUM-DQ ablation is to verify if GGE can learn biased data with biased models. The loss for the whole model is

$$L = \mathcal{L}(B_d + \sigma(B_q) + \sigma(\tilde{A}), A). \quad (6)$$

LMH+RUBi. LMH [4] and RUBi [2] are methods that can only reduce a single type of bias. LMH+RUBi is a direct combination of LMH and RUBi. It reduces distribution bias with LMH and shortcut bias with RUBi step by step. The loss for RUBi is written as

$$L_{rubi}(\tilde{A}, A) = \mathcal{L}(\tilde{A} \odot \sigma(G_q), \tilde{A}) + \mathcal{L}(c_q(G_q), A), \quad (7)$$

[1]https://github.com/hengyuan-hu/bottom-up-attention-vqa

where $G_q = g(e_q(q_i))$, $g(.)Q \rightarrow \mathbb{R}^C$. Combining with LMH, we compose A as

$$F(A, B, M) = \log A + g(M) \log B, \quad (8)$$

where M and B are the fused feature and the bias in LMH. The combined loss function is

$$L = L_{rubi}(F(A, B, M), \tilde{A}) + wH(g(M) \log B), \quad (9)$$

where $H(.)$ is the entropy and w is a hyper-parameter.

D. Supplementary Experimental Results

D.1. Ablations of Base Models

We do experiments on other base models BAN [5] and S-MRL [2]. The models are re-implemented based on officially released codes. For BAN, we set the number of Bilinear Attention blocks as 3. We choose the last bi-linear attention map of BAN and sum up along the question axis, which is referred to as the object attention for CGR and CGW. Although Accuracy of our reproduced S-MRL is a litter lower than that in [2], GGE-DQ can improve the Accuracy over 10% and surpass most of the existing methods. As shown in the table, GGE is a model-agnostic de-bias method, which can improve all three base models UpDn [1], S-MRL[2] and BAN [5] by a large margin.

D.2. Self-Ensemble Comparison

We provide an additional experiment for RUBi [2] with Self-Ensemble fashion. The input of the question-only branch is replaced by the joint representation from the base model. As shown in Table 2, RUBi-SF is even worse than baseline UpDn on both VQA-CP v2 test and VQA v2 val. On the contrary, Accuracy of GGE-SF is comparable to GGE-Q, which further demonstrates the generalization of GGE.

D.3. Additional Experimental Results

We provide detailed CGR, CGW, and results on VQA-CP and VQA v2 for all re-implemented methods in Section 3 and Section 5.

As shown Table 2, GGE-DQ largely improves more challenging "Others" question type [11]. This means that GGE-DQ really focuses on images largely rather than only relying on "inverse language bias" for higher Accuracy. Moreover, Inverse-Supervision strategy does not improve GGE-DQ-tog (GGE-DQ-tog$_{is}$ in Table 2), which also demonstrates that GGD-DQ better reduces distribution bias compared with other methods.

There are still some issues about language bias that deserves further consideration. First, both GGE-D$_{sxce}$ and GGE-Q$_{sxce}$ are robust on VQA v2 but GGE-DQ$_{sxce}$ drops a lot. We think the softmax function will amplify the gradient of biased models and over-estimate the dataset biases. Second, LMH+RUBi performs much better than both LMH and RUBi on VQA v2. This can bring further research into the relationship between distribution bias and shortcut bias. Third, UpDn$_{is}$ does not degrade a lot in VQA v2, which indicates some entanglement between entropy regularization and Inverse-Supervision strategy.

Moreover, we find that GGE also suffers from degradation on in-distribution data (VQA v2) similar to previous ensemble-based methods. This indicates that the model may over-estimate the bias for some instances. We speculate that it is due to too small scale of the gradient for some samples easy to fit by distribution bias or shortcut bias. How to control the over-fitting "degree" of biased models and scale up pseudo labels are potential research directions in the future.

E. Additional Qualitative Results

In this section, we provide more examples from GGE-DQ in Figure 3 and some failure cases in Figure 4.

Table 1. **Ablations of base model BAN and S-MRL**.

Method	VQA-CP test						
	All	Y/N	Num.	Others	↑CGR	↓CGW	↑CGD
S-MRL [2]	37.90	43.68	12.04	41.97	**41.94**	27.32	14.62
+GGE-DQ-tog	**54.62**	76.11	18.04	**47.70**	35.61	**18.17**	**17.44**
+GGE-DQ-iter	54.03	**79.66**	**20.77**	46.72	38.10	22.42	15.68
BAN [5]	35.94	40.39	12.24	40.51	5.33	5.19	0.14
+GGE-DQ-tog	**51.91**	**81.37**	21.85	45.46	**36.93**	27.10	**9.83**
+GGE-DQ-iter	50.75	74.56	20.59	**46.54**	20.87	**16.85**	4.98

Table 2. **Extra experimental results for Section 3 and Section 5**.

Method	VQA-CP test							VQA v2 val			
	All	Y/N	Num.	Others	CGR	CGW	CGD	All	Y/N	Num.	Others
UpDn [1]	39.89	43.01	12.07	45.82	44.27	40.63	3.91	**63.79**	80.94	**42.51**	**55.78**
HINT [10]	47.50	67.21	10.67	46.80	45.21	34.87	10.34	63.38	**81.18**	42.14	55.66
RUBi [2]	45.42	63.03	11.91	44.33	39.60	33.33	6.27	55.19	61.04	41.00	54.43
LM [4]	48.78	70.37	14.24	46.42	**47.30**	35.97	11.33	63.26	81.16	42.22	55.22
LMH [4]	52.73	72.95	31.90	47.79	46.44	35.84	10.60	56.35	65.06	37.63	54.69
CSS-V [3]	57.91	80.36	50.45	47.83	42.72	31.28	11.44	53.94	57.48	55.37	38.39
CSS [3]	58.11	83.65	40.73	48.14	46.70	37.89	8.81	53.15	61.20	37.65	53.36
HINT$_{inv}$	47.20	67.23	13.21	46.15	42.01	39.11	2.90	60.33	74.36	40.31	55.12
CSS-V$_{inv}$	58.05	79.84	52.24	47.23	41.38	34.93	6.45	54.39	58.73	38.81	55.23
UpDn$_{is}$	42.12	45.81	12.98	47.02	44.52	39.59	4.93	62.85	80.34	42.00	55.08
RUBi$_{is}$	48.16	72.34	12.69	45.22	47.55	33.73	13.83	59.10	76.67	41.09	50.50
LMH$_{is}$	**58.12**	79.73	**53.41**	48.01	39.51	30.82	8.69	43.29	33.22	34.14	53.40
GGE-DQ-tog$_{is}$	54.64	85.47	23.43	47.64	40.47	25.81	14.66	57.16	70.43	38.00	52.13
SUM-DQ	35.46	42.66	12.38	38.01	41.28	38.18	3.91	56.85	81.09	38.55	43.25
LMH+RUBi	51.54	74.55	22.65	47.41	46.67	40.55	6.12	60.68	77.91	39.10	53.15
GGE-D	48.27	70.75	13.42	47.53	38.79	**24.48**	14.31	62.79	79.24	42.31	55.71
GGE-Q-iter	43.72	48.17	14.24	48.78	43.74	37.04	6.70	61.23	78.28	41.42	53.50
GGE-Q-tog	44.62	47.64	14.34	48.89	45.19	38.56	6.63	62.14	78.64	40.72	54.21
GGE-DQ-iter	57.12	**87.35**	26.16	**49.77**	44.35	27.91	**16.44**	59.30	73.63	40.30	54.29
GGE-DQ-tog	57.32	87.04	27.75	49.59	42.74	27.47	15.27	59.11	73.27	39.99	54.39
RUBi-SF	37.53	43.27	14.11	41.07	39.30	32.66	7.14	55.06	70.85	30.97	49.44
GGE-SF-iter	44.53	50.98	18.24	48.90	45.07	38.99	6.08	60.66	74.93	41.14	52.95
GGE-SF-tog	43.10	49.90	17.74	47.33	42.40	35.85	6.55	59.00	73.71	41.14	52.54
GGE-D-SF-iter	56.33	86.43	23.37	49.32	43.77	29.30	14.47	62.03	80.73	41.79	53.14
GGE-D-SF-tog	52.86	76.25	20.56	49.46	42.48	30.25	12.23	59.00	73.71	41.14	52.54
UpDn$_{sxce}$	41.37	45.96	12.46	46.90	42.81	40.90	1.91	63.38	81.26	43.13	55.14
GGE$_{sxce}$-D	53.98	86.06	15.09	47.85	37.45	30.52	6.93	62.34	79.17	41.50	55.06
GGE$_{sxce}$-Q-iter	52.98	82.27	14.97	48.06	40.64	31.55	9.09	61.76	78.57	42.01	54.20
GGE$_{sxce}$-Q-tog	52.99	81.86	16.11	47.97	41.01	32.62	8.39	61.38	77.53	42.30	54.14
GGE$_{sxce}$-DQ-iter	56.25	85.08	24.78	48.56	43.13	29.52	13.61	52.38	54.51	39.93	54.07
GGE$_{sxce}$-DQ-tog	55.84	84.47	26.96	48.76	41.41	31.02	10.39	52.17	54.17	40.10	53.85

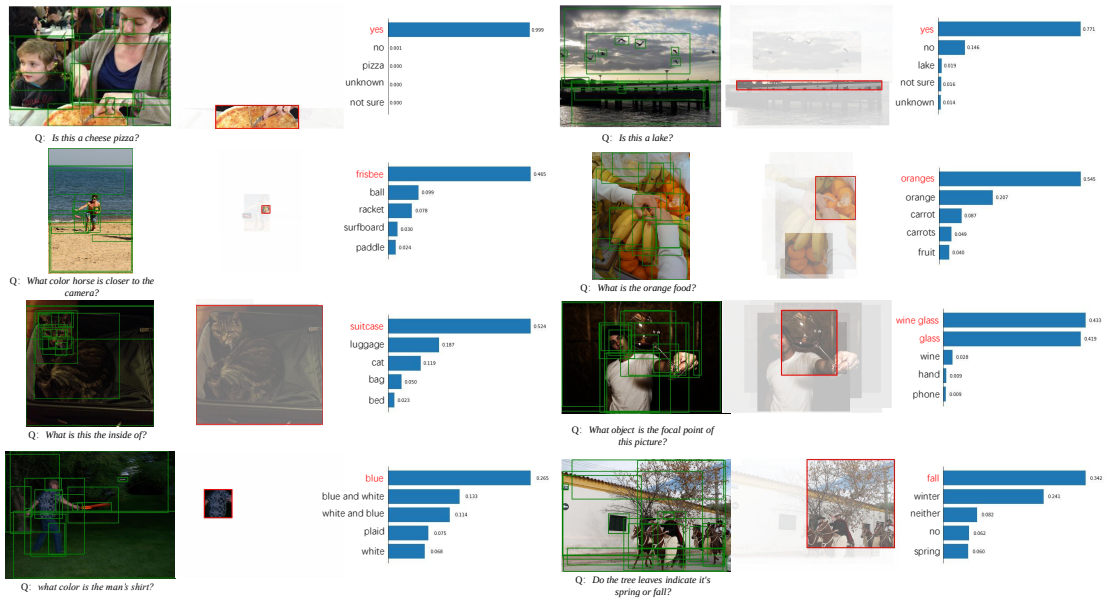


Figure 3. **More examples from GGE-DQ**. The model can successfully provide the right prediction with right evidences.

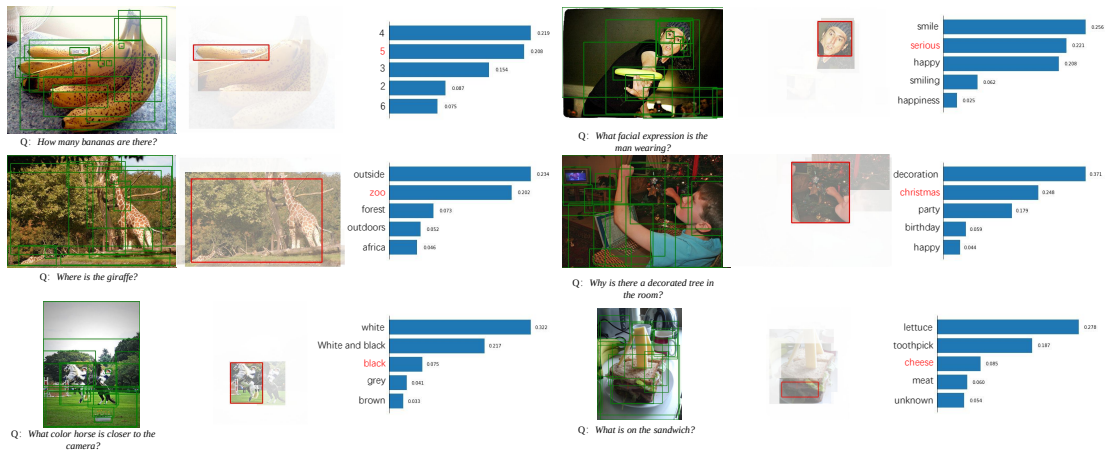


Figure 4. **Failure Cases**. Most of the failure cases still match their visual explanations (Wrong predictions with corresponding wrong evidences). The model is still weak in counting problem and questions that hardly appear in the train set (upper row). Some failure case are due to missing annotation in the dataset, since "outside" and "decoration" can also be regarded as the right answers (middle row). The last row shows that answers for failure cases are still consistent with visual explanations rather than language bias, which is identified by low CGW and indicates GGE-DQ really has better visual-grounding ability.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018. 2, 3, 4, 5

[2] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in neural information processing systems*, pages 841–852, 2019. 2, 3, 4, 5

[3] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10800–10809, 2020. 3, 5

[4] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Dont take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4060–4073, 2019. 2, 3, 5

[5] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574, 2018. 4, 5

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019. 3

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, pages 1532–1543, 2014. 3

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[9] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[10] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2591–2600, 2019. 5

[11] Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. On the value of out-of-distribution testing: An example of goodhart's law. *arXiv preprint arXiv:2005.09241*, 2020. 4