X-GGM: Graph Generative Modeling for Out-of-Distribution Generalization in Visual Question Answering

Jingjing Jiang, Ziyi Liu, Yifan Liu, Zhixiong Nan, Nanning Zheng *
Institute of Artificial Intelligence and Robotics, Xi’an Jiaotong University, Shannxi, China
jingjingjiang2017@gmail.com, {liuziyi, lyf666}@stu.xjtu.edu.cn, {nzx2018, nnzheng}@mail.xjtu.edu.cn

ABSTRACT

Encouraging progress has been made towards Visual Question Answering (VQA) in recent years, but it is still challenging to enable VQA models to adaptively generalize to out-of-distribution (OOD) samples. Intuitively, recompositions of existing visual concepts (i.e., attributes and objects) can generate unseen compositions in the training set, which will promote VQA models to generalize to OOD samples. In this paper, we formulate OOD generalization in VQA as a compositional generalization problem and propose a graph generative modeling-based training scheme (X-GGM) to implicitly model the problem. X-GGM leverages graph generative modeling to iteratively generate a relation matrix and node representations for the predefined graph that utilizes attribute-object pairs as nodes. Furthermore, to alleviate the unstable training issue in graph generative modeling, we propose a gradient distribution consistency loss to constrain the data distribution with adversarial perturbations and the generated distribution. The baseline VQA model (LXMERT) trained with the X-GGM scheme achieves state-of-the-art OOD performance on two standard VQA OOD benchmarks, i.e., VQA-CP v2 and QQA-OOD. Extensive ablation studies demonstrate the effectiveness of X-GGM components. Code is available at https://github.com/jingjing12110/x-ggm.

KEYWORDS

graph generative modeling, OOD generalization, VQA

1 INTRODUCTION

Visual Question Answering (VQA) [4], answering questions conditioned on understanding the given image and question, is a task toward general AI [62]. Although considerable progress has been made in VQA, most methods usually process similar data distributions between training and test sets. However, a good VQA model should be robust against answer distribution shift [35] and perform well on out-of-distribution (OOD) testing [61], i.e., evaluating the ability of the model to generalize beyond dataset-specific language biases [1, 2, 20, 75]. This paper focuses on OOD generalization in VQA, aiming to improve the OOD generalization ability of baseline VQA models while preserving their in-domain (ID) performance.

OOD generalization in VQA is challenging. To generalize to out-of-distribution samples adaptively, the VQA model should own two capabilities: (1) overcoming negative language biases and (2) producing out-of-distribution answers by learning rules entailed in in-domain data. The prevailing OOD generalization methods [10, 11, 18, 66] focus on enhancing the first capability, which achieves OOD generalization by explicitly mitigating the language biases. While the second capability, which directly endues VQA models

Figure 1: Compositional generalization implied in graph generative modeling. New compositions (e.g., brown broccoli) of attributes and objects can be generated by raw compositions (i.e., green broccoli and brown food).

the potentiality to generalize to out-of-distribution (i.e., unseen or rare) samples, has not been well explored.

Benefiting from good compositional generalization capabilities, humans can easily achieve OOD generalization [42]. Motivated by this, we formulate OOD generalization in VQA as a compositional generalization problem and leverage in-domain data to generate new/raw compositions of existing visual concepts (i.e., attributes and objects) to make VQA models produce out-of-distribution answers. As shown in Figure 1, given any two attribute-object pairs, such as green broccoli and brown food in Figure 1 (a), new compositions (e.g., brown broccoli in Figure 1 (b)) can be generated by graph generative modeling. The abundant recompositions probably contain unseen or rare samples in the training set, promoting the VQA models to generalize to out-of-distribution samples. To represent relationships and generate new compositions of existing visual concepts, Graph Generative Model [65] is the natural preference, which has been proven to have the potential to improve the generalization ability of models [12, 77].

In this paper, we propose X-GGM, a graph generative modeling-based training scheme, to improve the OOD generalization performance of existing VQA models while preserving their representation ability. As shown in Figure 2, X-GGM consists of two key modules, i.e., graph relation generative modeling (R-GGM) and graph representation generative modeling (N-GGM). R-GGM aims to update relationships between nodes to indirectly improve the OOD generalization ability. It first introduces adversarial perturbations (i.e., Gaussian noise) to the cross-modality representation (x) to initialize the relation matrix. After that, it utilizes Graph Encoder to aggregate information of different nodes and to update the relation matrix in an iterative manner. N-GGM targets perturbing node representations to directly enhance the OOD generalization ability. For N-GGM, adversarial perturbations are first injected into the cross-modality representation to initialize node
cross-modality representation to describe the original data distribution of relation matrices or node representations, leading to an inaccurate prior distribution in graph generative modeling. The inaccurate distribution affects the stability of graph adversarial learning. To alleviate the unstable training issue, we propose a gradient distribution consistency loss. The proposed loss can effectively constrain the gradient consistency between the data distribution injected into adversarial perturbations and the generated distribution (i.e., the distribution of generated relation matrices or node representations). Therefore, we replace the reconstruction loss in graph adversarial learning with the proposed gradient distribution consistency loss to circumvent the inaccurate prior distribution issue.

**Contribution.** Our main contributions are summarized as follows:

1. We propose a graph generative modeling-based training scheme, X-GGM, which can markedly improve the OOD generalization ability of existing VQA models while preserving their ID performance.
2. We propose a gradient distribution consistency loss, which can effectively alleviate the unstable gradient issue in graph adversarial learning by constraining the gradient consistency of the data distribution with adversarial perturbations and the generated distribution.
3. The baseline VQA model (LXMERT) trained with the proposed X-GGM scheme achieves state-of-the-art OOD performance while preserving the ID performance on two standard VQA OOD benchmarks, i.e., VQA-CP v2 and GQA-OOD.

## 2 RELATED WORK

### 2.1 OOD Generalization in VQA

Training and testing under the independent and identically distributed (i.i.d) setting have resulted in the performance of most VQA models being highly affected by superficial correlations (i.e., language biases and dataset biases) [1, 2, 20, 75]. Recently, evaluation on the out-of-distribution (OOD) setting [18, 24, 35, 61] has thus become an increasing concern for VQA. To improve the OOD generalization performance of VQA models, the prevailing methods target eliminating the language bias. Accordingly, current debiasing methods to VQA can be broadly divided into two groups: Known Bias-based [7, 10, 46] and Unknown Bias-based [11, 18, 59].

Known Bias-based methods consider the prior knowledge of language biases and design specific debiasing methods to reduce the biases explicitly existing in training set. Unknown Bias-based methods aim to remove the language bias or the answer distribution shift without the bias to be known in advance, which are more practical. In addition to considering extra human annotations [18, 53, 66] to balance datasets to reduce biases, most Unknown Bias-based methods leverage the adversarial training strategy [15, 18, 21, 44, 58, 61]. Particularly, VILLA [15] introduces adversarial perturbations into embedding space and utilizes large-scale adversarial training to improve the generalization ability. Similarly, the proposed X-GGM scheme injects adversarial perturbations into representation space and utilizes the adversarial learning strategy based on graph generative modeling to achieve OOD generalization and to mitigate language biases without prior knowledge of these biases.

### 2.2 Graph Generative Models

Graph Generative Modeling (GGM), aiming to learn to generate new graphs with certain desirable properties by observing existing graphs, has emerged in biomedical [13], social science [49], and graph representation learning [51, 65, 73]. Recent advances in deep generative models, especially VAE [37] and GAN [19], have developed towards the graph domain, inspiring many graph generative models, such as GAE [39], GraphVAE [56], GraphGAN [65], and Graphite [22]. These graph generative models follow the uniform framework that includes generative and discriminative models, limiting the scale of generated graphs. In order to handle the challenge of scalability, deep autoregressive models are utilized for graph generation, such as GraphRNN [71], GRANs [47], and GraphAF [54]. Furthermore, GRAM [34] introduces graph attention mechanism to enable the graph generative model scalability.

Recent approaches consider improving the generalization ability of graph generative models. You et al. [72] propose GraphCL, a contrastive learning-based framework, to learn robust graph representations. Pan et al. [51] encode the topological structure and node content in a graph to obtain compact graph representations by a novel adversarial graph embedding framework. Dai et al. [12] propose ANE, an Adversarial Network Embedding framework, to leverage the adversarial learning principle to regularize the representation learning, contributing to the learning stable and robust graph representation. Zheng et al. [77] propose DB-GAN, which implicitly bridges the graph and feature spaces by prototype learning, to learn more robust graph representation. The proposed X-GGM scheme considers generating both relationships between nodes and node representations during graph generative modeling to enhance the generalization ability of graph generative models.

### 2.3 Graph Construction in V+L Tasks

Graphs are non-Euclidean structured data, which can effectively represent relationships between nodes. Some recent works construct graphs for visual or linguistic elements in V+L tasks, such as VQA [16, 27, 43], VideoQA [28, 30, 79], Image Captioning [23, 70, 76], and Visual Grounding [31, 48, 69], to reveal relationships between these elements and obtain fine-grained semantic representations. These constructed graphs can be broadly grouped into three types: visual graphs between image objects/regions (e.g., [70]),...
linguistic graphs between sentence elements/tokens (e.g., [33]), and
crossmodal graphs among visual and linguistic elements (e.g., [48]).
In this work, we construct the visual graph for X-GGM.

3 PRELIMINARY
Taking the open-ended VQA task as a multi-class classification
problem, the VQA model requires identifying the correct answer from a predefined set \( \{a \in A\} \) of possible answers based on understanding the related question \( q \) and image \( I \). Commonly, a VQA model \((\mathcal{VQA})\) can be divided into two modules: Cross-modality
Representation \((\mathcal{VQA}_R)\) and Answer Classifier \((\mathcal{VQA}_A)\), which can be formulated by \[
\hat{a} = \mathcal{VQA}_C(a|x), \quad x = \mathcal{VQA}_R(q, I),
\]
where \(\mathcal{VQA}_A\) is the crucial module of a VQA model, determining the representation
ability of the VQA model. Therefore, the method aiming to achieve OOD generalization in VQA needs to improve the
generalization performance while maintaining the representation
ability of \(x\). In this paper, we consider two prevailing VQA models, i.e., transformer-based model (LXMERT [57]) and plain model (UpDn [3]), as our baseline VQA models for evaluating the OOD performance of the proposed X-GGM scheme.

**LXMERT.** LXMERT [57] is a classic transformer-based on cross-modality pre-training, consisting of two single-modality encoders for visual and language modalities, respectively, and one cross-modality encoder for aligning entities across the two modalities. It can output a visual feature sequence \(\{o_n|o_n \in \mathbb{R}^d, 1 \leq n \leq N_o\}\), a language feature sequence, and the cross-modal feature vector \(x \in \mathbb{R}^d\), where, \(N_o\) is the number of objects in one image, and \(o_n\) is the \(n\)-th object feature vector. The visual feature sequence \(\{o_n\}\) and the cross-modal feature vector \(x\) will be passed into the X-GGM scheme to improve the OOD generalization ability of \(x\) by generatively modeling new relationships between objects and new node representations.

**UpDn.** Bottom-Up Top-Down model (UpDn) [3] is a typical attention based VQA model. For each image \(I\), it utilizes an image encoder to output the sequence of object features \(\{o_n|o_n \in \mathbb{R}^d, 1 \leq n \leq N_o\}\). For each question, it utilizes a question encoder to output the sequence of language features. After that, both the object and language features are fed into an attention module to obtain the joint representation \(x \in \mathbb{R}^d\) for answer prediction. \(\{o_n\}\) and \(x\) are the inputs of the X-GGM scheme.

4 GRAPH GENERATIVE MODELING
To improve the OOD generalization ability of \(x\) while preserving its representation ability, we consider the nature attribute of compositional generalization implicit in the information aggregation of graph nodes, and propose a Graph Generative Modeling-based training scheme (X-GGM). The overall process of training a baseline VQA model with the X-GGM scheme is shown in Algorithm 1, consisting of two parts: Graph Generative Modeling (X-GGM) and Baseline Model Update (BMU). Once \(\mathcal{VQA}_R\) yields the cross-modality representation \(x\) and the sequence of visual feature vectors \(\{o_n|o_n \in \mathbb{R}^d, 1 \leq n \leq N_o\}\), X-GGM will update relationships between objects and representations of objects by

Algorithm 1 Training a VQA Model with X-GGM
1. function X-GGM(I, q, a, R_GT)
2. \{o_n|1 \leq n \leq N_o\}, x \leftarrow \mathcal{VQA}_R(I, q)
3. cond \sim U[0, 1]
4. if cond \leq \eta then \quad \triangleright \text{execute R-GGM}
5. \quad R_0 \leftarrow R_{init}(x, \sigma)
6. \quad R_g, \{\hat{o}_1, \ldots, \hat{o}_{N_g}\} \leftarrow \text{Rgen}(\{o_n\}, R_0)
7. \quad p_a \leftarrow \mathcal{VQA}_C(x, \hat{o}_{N_g})
8. \quad L \leftarrow \text{Rlearn}(R_g, R_{GT}, p_a)
9. else \quad \triangleright \text{execute N-GGM}
10. \quad \{v_i, 1 \leq i \leq N_o\}_0 \leftarrow \text{Ninit}(x, \sigma)
11. \quad V_g, \{v_1, \ldots, v_{N_g}\} \leftarrow \text{Ngen}(\{v_i\}_0, R_{GT})
12. \quad p_a \leftarrow \mathcal{VQA}_C(x, v_{N_g})
13. \quad L \leftarrow \text{Nlearn}(V_g, \{o_n\}, p_a)
14. end if
15. end function
16. function BMU(I, q, a) \quad \triangleright \text{training baseline model}
17. \quad p_a \leftarrow \mathcal{VQA}_C(I, q, a)
18. end function

Graph Relation Generative Modeling (R-GGM) and Graph Representation Generative Modeling (N-GGM) in each training update\(^1\). In the following, we first introduce the predefined object-relationship graph and then detail R-GGM and N-GGM.

4.1 Graph Construction
For the X-GGM scheme, we construct an object-relationship graph using objects in one image as graph nodes. Specifically, the object-relationship graph can be formulated as \(G = (\mathcal{V}, \mathcal{A}, R)\), where \(\mathcal{V} = \{v_1, \ldots, v_{N_o}\}\) is a set of nodes with \(|\mathcal{V}| = N_o\), \(A\) is the adjacency matrix with \(a_{ij} \in A\) marking whether there is an edge between node \(v_i\) and \(v_j\), and \(R\) is a relation matrix describing the coherence between nodes. The constructed graph \(G\) is a fully-connected graph, meaning that the adjacency matrix \(A\) is an all-ones matrix. Therefore, we omit \(A\) in \(G\) for conciseness in following sections.

To determine the ground-truth (GT) of relation matrix \(R_{GT}\) utilizing in R-GGM, we compute the cosine similarity between object-class embeddings \(\{e_n|e_n \in \mathbb{R}^d, 1 \leq n \leq N_o\}\) and object-attribute embeddings \(\{a_n|a_n \in \mathbb{R}^d, 1 \leq n \leq N_o\}\). These embeddings (i.e., \(e_n\) and \(a_n\)) are obtained by a pre-trained BERT model [14]. Practically, the similarity computations are random combinations of object-classes and object-attributes in one image, which will breed new combinations. In N-GGM, the object feature \(o_n\) yielded by \(\mathcal{VQA}_R\) serves as the GT of the \(i\)-th node representation (i.e., \(o_{GT,i} = o_n\)).

4.2 Graph Relation Generative Modeling
Graph Relation Generative Modeling (R-GGM) consists of three stages: initializing relation matrix utilizing \(x\) and Gaussian noise (Rinit), Graph Relation Matrix Generation (Rgen), and Adversarial Learning of Relation (Rlearn).

**Initializing Relation Matrix (Rinit).** The graph \(G\) is an undirected graph with \(N_o\) nodes. Therefore, the relation matrix \(R\) in

\(^1\)Since the explicit relation triplets between objects are challenging to obtain, this paper operates the relation matrix and node representations instead of the adjacency matrix to achieve implicit recompositions.
We utilize the output of the last iteration as the final generated relation matrix, that is, $R_g = R_{N_k}$. In addition, we average node representations $\{v_i, 1 \leq i \leq N_k\}$ to derive the graph representation $\hat{g}_k$ in the $k$-th iteration.

### Adversarial Learning of R-GGM (RLEARN)

When initializing the relation matrix in the stage of RINIT, we find that the process of R-GGM introducing adversarial perturbations is similar to that of Denoising AutoEncoder [64]. From the works [5, 45], we know that the optimal Denoising AutoEncoder for additive white Gaussian noise (AWGN) can be explicitly calculated, which is related to the gradient of the data distribution after adding noises. Practically, the logarithmic gradient of the data distribution with adversarial perturbations can be deduced from Eq. (5):

$$\nabla_r \log p_\sigma(\hat{r}|r) = -\frac{(\hat{r} - r)^2}{\sigma^2}. \tag{7}$$

After obtaining $\nabla_r \log p_\sigma(\hat{r}|r)$, we can compute the gradient of log-likelihood estimation, i.e., $\nabla_r \log p_\sigma(r)$, of elements in the generated relation matrix $R_g$ to constrain the gradient distribution consistency between the generated relation matrix $R_g$ and the derived similarity matrix $R_{GT}$. Therefore, the gradient distribution consistency loss can be defined as

$$\mathcal{L}_G \triangleq \mathbb{E} \left[ \left\| \nabla_r \log p_\sigma(r) - \nabla_r \log p_\sigma(\hat{r}|r) \right\|^2_2 \right]. \tag{8}$$

In addition to constraining the gradient distribution consistency, we further utilize KL-Divergence to measure the distance between the generated distribution $p_\sigma(r)$ and the raw data distribution $p_{\text{data}}(r)$, where $p_\sigma$ and $p_{\text{data}}$ denote probability distributions of elements in matrix $R_g$ and $R_{GT}$, respectively. The distance measure loss of training our R-GGM can be formulated by

$$\mathcal{L}_D = \text{KL}(p_\sigma || p_{\text{data}}) + \text{KL}(p_{\text{data}} || p_\sigma), \tag{9}$$

where, $\text{KL}(p||q) = \mathbb{E}_{r \sim p} [\log p(r) - \log q(r)]$.

Besides the adversarial losses (i.e., $\mathcal{L}_G$ and $\mathcal{L}_D$), we compute the BCE loss ($\mathcal{L}_{\text{BCE}}$) of the VQA task by adding $x$ (the original cross-modality representation) and $\hat{g}_k$ (the graph representation yielded by the last iteration in RGEN) and passing the value into $VQA_{\mathcal{L}}$ to derive the likelihood estimation of answer distribution $p_a$.

In summary, the total loss of training R-GGM can be formulated by

$$\mathcal{L} = \alpha \mathcal{L}_G + \beta \mathcal{L}_D + \mathcal{L}_{\text{BCE}}, \tag{10}$$

where, $\alpha$ and $\beta$ are weighted coefficients.
4.3 Graph Representation Generative Modeling

Graph Representation Generative Modeling (N-GGM) is similar to R-GGM, which can also be implemented in three stages: initializing node representations with $x$ and Gaussian noise (Ninit), Graph Node Representation Generation (NGen), and Adversarial Learning of Node Representation (Nlearn).

**Initializing Node Representations (Ninit).** To obtain the $N_0$ initial node representations of the graph $G$, we first span $x$ to $[x, \ldots, x] \in \mathbb{R}^{N_x \times d}$, and utilize a fully-connected layer ($M_{FC}$) to increase the variability of node representations:

$$V_x = M_{FC}([x, \ldots, x]), V_x \in \mathbb{R}^{N_x \times d}, \quad (11)$$

where, $V_x = \{v_i|x|v_x | \in \mathbb{R}^d, 1 \leq i \leq N_0\}$. After that, we inject adversarial perturbations to $V_x$, and make the sampled feature vector $\hat{\delta}_i, x$ of the $i$-th element in $V_x$ satisfy the noise prior distribution $p_{\sigma}(\hat{\delta}_i, x|v_i, x)$:

$$p_{\sigma}(\hat{\delta}_i, x|v_i, x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\hat{\delta}_i, x - v_i, x)^2}{2\sigma^2}\right). \quad (12)$$

Finally, the initial node representations of the graph $G$ can be obtained by $V_0 = V_x = \{\hat{\delta}_0, x, \ldots, \hat{\delta}_{N_0}, x\}$.

**Node Representation Generation (NGen).** The details of NGEN are shown in Figure 4. In the $k$-th iteration, the node representations $V_{k-1} = \{v_i, 1 \leq i \leq N_0\}_{k-1}$ generated by Graph Encoder in the $(k-1)$-th iteration and the predefined GT relation matrix $R_{GT}$ are first passed into the $k$-th Graph Encoder to update the node representations of each layer. After that, the final output of the $k$-th Graph Encoder $V_k$ can be obtained by Eq. (5). The architecture of Graph Encoder in N-GGM is the same as that in R-GGM. We utilize the output of the last iteration as the generated node representations, i.e., $V_f = V_{N_0}$. Besides, in each iteration, we average $N_0$ node representations and yield the graph representation $\theta_k$.

**Adversarial Learning of N-GGM (Nlearn).** The total loss $L$ of training N-GGM is also a weighted sum of $L_{v}$, $L_{D}$, and $L_{BCE}$. Specifically, the gradient distribution consistency loss $L_{v}$ is defined as the distance between the distribution gradient of the initial node representations after adding Gaussian noise ($V_x$) and the log-likelihood estimation of generated node representations ($V_f$). $L_{D}$ is the KL-Divergence loss between the distribution of generated node representations ($V_f$) and the distribution of the raw node representations ($\{v_i\}$) yielded by $VQAAR$. The VQA classification loss $L_{BCE}$ is the same as the corresponding term in Eq. (10).

## 5 EXPERIMENTS

In this section, we evaluate the effectiveness of X-GGM key components and compare the proposed X-GGM scheme with state-of-the-art approaches on two standard VQA OOD benchmarks, i.e., VQA-CP v2 [2] and GQA-OOD [35]. Table 1 reports the statistics of experimental datasets.

### Table 1: Statistics of VQA-CP v2 and GQA-OOD.

| Dataset            | Image | Question |
|--------------------|-------|----------|
|                    | Train | Val      | OOD Test | ID Test | Total    |
| VQA-CP v2          | 119,107 | 31,976 | 89,437 | 32,266 | 272,786 |
| GQA-OOD            | 72,140 | 9,406 | 330 | 365 | 82,241 |
|                    | Train | Val | OOD Test | ID Test | Total    |
| VQA-CP v2          | 388,735 | 38,802 | 179,928 | 40,000 | 647,465 |
| GQA-OOD            | 943,000 | 51,045 | 1,063 | 1,733 | 996,841 |
| Year               |       |       |          |          |          |

### Table 2: Ablation studies on R-GGM and N-GGM with different combinations of $L_D$ and $L_v$ (More results are shown in Table 6 of the Appendix).

| S/N | R-GGM | N-GGM | VQA-CP v2 | GQA-OOD |
|-----|-------|-------|-----------|---------|
|     | $L_v$ | $L_D$ | $L_v$ | $L_D$ | $OOD\uparrow$ | $ID\uparrow$ | $Gap\uparrow$ | Tail$\uparrow$ | Head$\uparrow$ | $\Delta$ |
| LX  | —     | —     |        |        | 59.98 | 65.21 | 5.23 | 49.80 | 57.70 | 15.90 |
| #1  | ✓     | ✓     |        |        | 63.84 | 66.41 | 2.57 | 51.39 | 56.93 | 10.77 |
| #2  | ✓     | ✓     |        |        | 62.90 | 65.57 | 2.67 | 51.27 | 57.30 | 11.76 |
| #3  | ✓     | ✓     |        |        | 63.75 | 66.03 | 2.28 | 51.65 | 57.13 | 10.61 |
| #4  | ✓     | ✓     |        |        | 62.73 | 65.99 | 3.26 | 51.27 | 57.19 | 11.54 |
| #5  | ✓     | ✓     |        |        | 64.33 | 66.92 | 2.59 | 52.47 | 57.51 | 9.59 |

5.1 Experimental Settings

#### 5.1.1 Evaluation Benchmarks.

**VQA-CP v2.** It is the reorganization of the training and validation sets of VQA v2.0 [55]. To evaluate the OOD generalization performance of existing VQA models, Agrawal et al. [2] construct VQA-CP v2 by ensuring that the answer distribution of per question type in the test set and the training set is different. The original VQA-CP v2 only consists of the test set (about 98K images, 220K questions, and 2.2M answers) and the training set (about 121K images, 438K questions, and 4.4M answers). That is, the benchmark does not have an official validation set.

**GQA-OOD.** It is a fine-grained restructuring of GQA [29], which obtains the validation and test sets by introducing answer distribution shifts into the validation and test sets of GQA. However, GQA-OOD shares the same training set with GQA. The GQA-OOD benchmark provides four metrics. 1) **Tail:** accuracy on OOD samples (i.e., samples of the tail of the answer class distribution), 2) **Head:** accuracy on in-domain (ID) samples (i.e., samples from the head distribution), 3) **All:** overall accuracy on all test samples, and 4) **Δ (%)** = (head - tail) / tail: a new evaluation protocol proposed by Kervadec et al. [35] to illustrate how much is the error prediction imbalanced between frequent and rare answers.

#### 5.1.2 Experimental Details.

**Data Preprocessing.** As Teney et al. [61] discussed, there are several pitfalls in evaluating VQA models under the OOD setting on VQA-CP v2, especially using the OOD test set for model selection and evaluating the ID performance of a VQA model after retraining the model on VQA v2.0 [55]. Therefore, following recent works [11, 58–60], we re-split VQA-CP v2. Specifically, we reserve 40k random samples from the train set to evaluate the in-domain (ID) performance and 40k random instances from the test set for model selection. Besides, the bottom-up attention Faster R-CNN [3]...
We conduct ablation studies on key components of X-GGM using LXMERT as the baseline VQA model\(^2\). Considering the discovery in work\([61]\), i.e., the questions with yes/no/number answers on VQA-CP v2 are easier to game, for a fairer comparison, we train all models on VQA-CP v2 "All". But, we select the best model and focus the model analysis on VQA-CP v2 "other".

5.2.1 Effectiveness of Key Components.

R-GGM / N-GGM / X-GGM vs. Baseline. To improve the OOD generalization ability of the cross-modality representation \(x\) yielded by VQA/AR, besides training the baseline VQA model using the full X-GGM scheme, R-GGM and N-GGM can also work independently. Therefore, we consider comparisons between the ablated components (\#1, \#2, \#5) and the Baseline, i.e., LXMERT (LX.) in Table 2 to evaluate the respective effects of R-GGM, N-GGM, and X-GGM. Specifically, \(\triangleright\) LX. vs. \#1 (only using R-GGM in the training process). \(\triangleright\) LX. vs. \#2 (only using N-GGM in the training process). \(\triangleright\) LX. vs. \#5 (using the full X-GGM scheme that executes R-GGM and N-GGM at a ratio of \(\eta\) in the training process). Table 2 shows the results on VQA-CP v2 "Other" and QQA-OOD. LX. vs. \#5 demonstrates the effectiveness of X-GGM. LX. vs. \#1 / \#2 illustrates that the single R-GGM / N-GGM can also effectively improve the OOD generalization performance of the baseline VQA model.

R-GGM / N-GGM vs. X-GGM. In each training iteration, X-GGM randomly executes R-GGM or N-GGM to improve the OOD generalization ability of the baseline VQA model. To explore the importance of R-GGM and N-GGM to the full training scheme (X-GGM), as shown in Table 2 (\#1, \#2, \#5), we conduct the following comparisons: \(\triangleright\) \#1 vs. \#5, and \(\triangleright\) \#2 vs. \#5. The results in Table 2 indicate that R-GGM is more effective than N-GGM for the X-GGM scheme in boosting the OOD performance of the baseline VQA model (LXMERT) on both VQA-CP v2 "Other" and QQA-OOD. This is mainly because the gradient instability caused by the noise introduction in N-GGM is larger than that in R-GGM. Specifically, the noise vector dimensions introduced into R-GGM and N-GGM are \(N_o \times (N_o - 1) / 2 = 630\) and \(N_o \times d = 27,648\), respectively.

Gradient Distribution Consistency Loss. To evaluate the effectiveness of the gradient distribution consistency loss \(L_Y\) in alleviating the unstable gradient issue, we compare the respective effects of \(L_Y\) and \(L_D\). The KL-Divergence loss \(L_D\) is also an adversarial

5.2.2 Variants of Graph Encoder.

In the proposed X-GGM scheme, Graph Encoder is a general block utilized to aggregate node information and encode relationships between nodes. That is, architecture differences between different Graph Encoders can not markedly affect the performance of X-GGM to OOD generalization. To demonstrate that the X-GGM scheme is robust for different variants of Graph Encoder, we consider three prevailing Graph Neural Networks (i.e., GCN\([38]\), GIN\([67]\), and GAT\([63]\)) and set the number of their layers \(N_l\) to 1 and 2. From the results on VQA-CP v2 "Other" and QQA-OOD shown in Table 3, we can observe that there is only a slight difference between the OOD performance of different Graph Encoder variants. However, all of these variants have achieved dramatic improvements in OOD performance compared with the baseline VQA model (LXMERT). The results demonstrate that X-GGM is agnostic to Graph Encoder and is robust for different Graph Encoders.

5.2.3 Hyperparameter.

Ratio of R-GGM and N-GGM \(\eta\). The threshold \(\eta\) in Algorithm 1 determines the ratio of R-GGM and N-GGM executed in each training iteration. To analyze the influence of \(\eta\) for X-GGM improving the OOD performance, we conduct some ablated experiments on different values of \(\eta\), i.e., \(\eta \in \{0.1, 0.2, \ldots, 0.9\}\). Results on VQA-CP v2 "Other" and QQA-OOD are shown in Figure 5 (a) and (b), respectively. We observe that the baseline VQA model (LXMERT) trained with the proposed X-GGM scheme can achieve the best OOD performance on VQA-CP v2 "Other" and QQA-OOD when \(\eta = 0.8\) and \(\eta = 0.5\), respectively. The average gains of the OOD performance on VQA-CP v2 "Other" and QQA-OOD are +3.76\(_{+0.36}\) and +1.54\(_{+0.59}\), respectively. Comparing the OOD performance gains

---

\(\triangleright\) denotes the adversarial loss to train the baseline VQA model (LXMERT) with the X-GGM scheme. \(\triangleright\) denotes the adversarial loss to train the baseline VQA model with the X-GGM scheme, and \(\triangleright\) using a weighted sum of \(L_Y\) and \(L_D\) as the adversarial loss to train the baseline VQA model with the X-GGM scheme. The comparisons between \#3, \#4, and \#5 on VQA-CP v2 "Other" and QQA-OOD in Table 2 illustrate that both \(L_Y\) and \(L_D\) are effective for training the baseline VQA model with the X-GGM scheme. However, \(L_Y\) performs better than \(L_D\), demonstrating the effectiveness of \(L_Y\) in alleviating unstable gradient updates in graph adversarial learning.

### Table 3: Comparisons between different variants of Graph Encoder (GEnc.) with \(N_l\) layers.

| GEnc. | \(N_l\) | VQA-CP v2 OOD | QQA-OOD ID | Tail | Head | \(A_\Omega\) |
|-------|--------|---------------|----------|------|------|-----------|
| LX.   | –      | 59.98         | 65.21    | 5.23 | 49.80 | 57.70     | 15.90     |
| GCN   | 1      | 64.10         | 66.70    | 2.60 | 51.65 | 57.24     | 10.83     |
|       | 2      | 64.33         | 66.92    | 2.59 | 52.47 | 57.51     | 9.59      |
| GIN   | 1      | 63.62         | 66.24    | 2.62 | 50.99 | 57.08     | 11.93     |
|       | 2      | 63.83         | 66.39    | 2.56 | 51.27 | 57.30     | 11.76     |
| GAT   | 1      | 63.94         | 66.40    | 2.46 | 51.65 | 57.71     | 11.73     |
|       | 2      | 63.94         | 66.45    | 2.51 | 50.99 | 56.90     | 11.59     |
Table 4: Comparisons with UB-based debiasing methods on VQA-CP v2 (full comparisons with state-of-the-arts are reported in Table 7 of the Appendix). \( \dagger \): methods using additional manually annotated data, \( \ast \): results from our reimplementation using the official released code, ---: not reported.

| Method               | VQA-CP v2 (OOD Test) \( \dagger \) | VQA-CP v2 (ID Test) \( \dagger \) | Gap \( \dagger \) |
|----------------------|----------------------------------------|----------------------------------------|----------------|
|                      | All Yes/No Num. Other                  | All Yes/No Num. Other                  | All Yes/No Num. Other |
| **Unshuffling [59]** | 42.39 47.72 14.43 47.24               | 42.39 47.72 14.43 47.24               | 42.39 47.72 14.43 47.24 |
| **Unshuffling+CF [58]** | 40.60 61.30 15.60 46.00               | 63.30 79.40 45.50 53.70               | 22.70 18.10 29.90 7.70 |
| **Unshuffling+CF+GS [58]** | 46.80 64.50 15.30 45.90               | 62.40 77.80 43.80 53.60               | 15.60 13.30 28.50 7.70 |
| **UpDn [3]**         | 38.82 42.98 12.18 43.95               | 64.73 79.45 49.59 55.66               | 25.91 36.47 37.32 11.71 |
| + Top answer masked [61] | 40.61 82.44 27.63 22.26               | 30.90 44.12 5.00 20.85               | 9.71 38.32 22.63 1.41 |
| + AdvReg [21]        | 36.33 59.33 14.01 30.41               | 50.63 67.39 38.81 38.37               | 14.30 8.06 24.80 7.96 |
| + GRL [21]           | 42.33 59.74 14.78 40.76               | 56.90 69.23 42.50 49.36               | 14.57 9.49 27.72 8.60 |
| + Randlm (\( \lambda = 12 \)) [61] | 55.37 83.89 41.60 44.20               | 54.24 64.22 34.40 50.46               | 1.13 19.67 7.20 6.26 |
| + Randlm (\( \lambda = 5 \)) [61] | 51.15 75.06 24.30 45.99               | 59.28 70.66 43.06 53.40               | 8.13 4.40 18.76 7.41 |
| + MUTANT\( \ast \) [18] | 61.72 88.90 49.68 50.78               | 64.47 83.12 48.26 56.91               | 21.45 40.26 20.61 4.57 |
| + X-GGM               | 45.71 43.48 27.65 52.34               | 67.16 83.74 48.26 56.91               | 11.67 10.94 13.25 5.23 |
| **LXMERT\( \ast \) [57]** | 63.90 80.45 46.58 59.98               | 75.57 91.39 59.83 65.21               | 11.67 10.94 13.25 5.23 |
| + MUTANT\( \ast \) [18] | 69.52 93.15 67.17 57.87               | 75.57 91.39 59.83 65.21               | 11.67 10.94 13.25 5.23 |
| + VILLA\( \ast \) [15] | 48.06 42.66 18.97 58.89               | 75.85 90.40 60.02 66.66               | 27.79 47.74 41.05 7.77 |
| + X-GGM               | 66.95 78.91 53.63 64.33               | 75.88 90.18 60.06 66.92               | 8.93 11.27 6.43 2.59 |

(w.r.t. the baseline VQA model) with the standard deviations of the OOD performance, we can consider the performance fluctuation is slight, demonstrating the robustness of the X-GGM scheme to the hyperparameter \( \eta \).

5.3 Comparisons with State-of-the-Arts

To evaluate the OOD and ID performance of baseline VQA models trained with the proposed X-GGM scheme, we compare them with state-of-the-art methods on VQA-CP v2 and GQA-OOD.

5.3.1 Performance on VQA-CP v2.

**Setting.** X-GGM is a training scheme for improving the OOD generalization ability of baseline VQA models while preserving the ID performance, which alleviates the language bias in VQA to some extent without the bias being known in advance. Therefore, according to whether debiasing methods need to know the bias existing on the datasets beforehand, we group them into two types. (1) Known Bias-based (KB-based) methods, such as LMH [10], LRS [25], and LMH-MFE [17]. (2) Unknown Bias-based (UB-based) methods, such as MCE [11], MUTANT [18], Unshuffling [59], and VILLA [15]. Particularly, VILLA [15] utilizes the adversarial training strategy that introduces noises into representation space to improve model generalization ability. The strategy of injecting adversarial perturbations into representation space inspired our work. Therefore, we reimplement VILLA [15] on VQA-CP v2 for a fairer comparison. On VQA-CP v2, we utilize LXMERT and UpDn as baseline VQA models and mainly compare the proposed X-GGM scheme with UB-based methods (more comparisons with state-of-the-art methods are shown in the Appendix). As with ablation study, we implement our experiments on VQA-CP v2 “All” but focus comparisons on “Other”, which can better evaluate the ability of approaches to OOD generalization than “Yes/No” and “Num.”.

**Results.** Comparisons are reported in Table 4. As shown in the table, facilitated by the proposed X-GGM scheme, the baseline VQA model (LXMERT) achieves state-of-the-art OOD and ID performance on VQA-CP v2 “Other”. Besides, compared with all methods utilized UpDn as the baseline VQA model, our method also gains the best OOD and ID performance on “Other”. These results demonstrate the X-GGM scheme is effective for a VQA model to generalize to out-of-distribution samples while preserving its ID performance.

5.3.2 Performance on GQA-OOD.

**Settings.** Except for LMH [10] and LM [10], the other methods listed in Table 5 are UB-based and can be further grouped into plain models (models in the third row of Table 5, e.g., BAN4 [36] and MMN [8]) and transformer-based models (models in the fourth row of Table 5 like MCAN [74] and VILLA [15]). We utilize LXMERT and UpDn as baseline VQA models, and compare the proposed X-GGM scheme with all existing methods on GQA-OOD. Specifically, RUBi [6], LMH [10] and LM [10] utilize UpDn as their baseline model. VILLA [15] and MANGO [44] use LXMERT as the baseline VQA model. We also reimplement VILLA on GQA-OOD.
Table 5: Comparisons with state-of-the-arts on GQA-OOD.

| Method          | Metrics (%) | All ↑ | Tail ↑ | Head ↑ | Δ ↓ |
|-----------------|-------------|-------|--------|--------|-----|
| **Plain Models**|             |       |        |        |     |
| BAN4 [36]       | 50.20       | 57.24 | 51.90  | 9.90   |     |
| MMN [8]         | 52.70       | 48.00 | 55.50  | 15.60  |     |
| UpDn [3]        | 46.40       | 42.10 | 49.10  | 16.60  |     |
| + LMH [10]      | 33.10       | 30.80 | 34.50  | 12.00  |     |
| + LM [10]       | 34.50       | 32.20 | 35.90  | 11.50  |     |
| + RUBi [6]      | 38.80       | 35.70 | 40.80  | 14.30  |     |
| + X-GGM         | 48.41       | 45.99 | 49.90  | 8.51   |     |
| **Transformer-based Models** | |       |        |        |     |
| MCAN [74]       | 50.80       | 54.60 | 53.40  | 14.80  |     |
| MANGO3 [44]     | 56.40       | 51.27 | 59.55  | 16.15  |     |
| LXMERT          | 54.60       | 49.80 | 57.70  | 15.90  |     |
| + VILLA* [15]   | 54.47       | 49.95 | 57.24  | 14.59  |     |
| + MANGO [44]    | 54.94       | —     | —      | —      |     |
| + X-GGM         | 55.59       | 52.47 | 57.51  | 9.59   |     |

Results. From the comparisons in Table 5, we can observe that benefiting from the OOD generalization ability of X-GGM, the baseline VQA model (LXMERT) trained with our scheme outperforms all other methods on GQA-OOD on the metric Tail. Although the ID performance (Head) drops by 0.19 relative to the baseline VQA model, it is far smaller than the OOD performance improvement (2.67). Besides, the baseline VQA model (UpDn) trained with the X-GGM scheme also outperforms these methods utilized UpDn as the baseline VQA model on all metrics, especially on Tail. These results demonstrate the ability of the proposed X-GGM scheme to OOD generalization and ID preserving. Unfortunately, the OOD performance improvement on GQA-OOD is not as obvious as that on VQA-CP v2. It likely because questions of GQA-OOD are semantic compositionality. The syntax-complex questions weaken the representation ability of x while X-GGM also does not explicitly leverage the structured representation of question to enhance x.

5.4 Qualitative Examples

To qualitatively illustrate the effectiveness of the X-GGM scheme in improving the OOD performance of baseline VQA models, we show some prediction examples from the OOD test sets of VQA-CP v2 and GQA-OOD. As shown in Figure 6, the Baseline (i.e., LXMERT) is prone to utilize in-domain answers (e.g., eating and sitting) to answer out-of-distribution questions. In contrast, the Baseline after being trained with the proposed X-GGM scheme can yield out-of-distribution answers, such as brushing teeth and posing in the first row of Figure 6, demonstrating the ability of X-GGM to OOD generalization. More predictions and visualization of the generated relation matrices are shown in Figure 7-8 of the Appendix.

6 CONCLUSION

In this paper, we proposed a Graph Generative Modeling-based scheme, X-GGM, to improve the OOD generalization ability of baseline VQA models while preserving the ID performance. Specifically, the X-GGM scheme randomly executes R-GGM or N-GGM to generate a new relation matrix or new node representations to enable the baseline VQA model to generalize to out-of-distribution samples. Besides, to alleviate the unstable gradient issue in graph adversarial learning, we propose a gradient distribution consistency loss to constrain the derivative of data distribution with adversarial perturbations and generated distribution. We evaluate the effectiveness of X-GGM components by extensive and ablative experiments. The baseline VQA model (LXMERT) trained with the proposed X-GGM scheme achieves state-of-the-art OOD performance on two VQA OOD benchmarks while preserving the ID performance.

ACKNOWLEDGMENTS

This work was supported by the NSFC Grants 61773312, 62088102.

REFERENCES

[1] Ashwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the Behavior of Visual Question Answering Models. In EAMSLP. 1955–1960.
[2] Ashwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don’t just assume; look and answer: Overcoming priors for visual question answering. In CVPR. 4971–4980.
[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In CVPR. 6077–6086.
[4] Stanislaw Antol, Ashwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In CVPR. 2425–2433.
[5] Siavash A Bigdeli, Geng Lin, Tiziano Portenier, L Andrea Dunbar, and Matthias Zwicker. 2020. Learning generative models using denoising density estimators. arXiv preprint arXiv:2001.02728 (2020).
[6] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. 2019. RUBi: Reducing unimodal biases for visual question answering. In NeurIPS. 841–852.
[7] Long Chen, Xin Tan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. 2020. Counterfactual samples synthesizing for robust visual question answering.
A APPENDIX

A.1 More Experimental Results

Ablation Studies. The results in Table 2 indicate that a single R-GGM or N-GGM can also effectively improve the OOD generalization ability of baseline VQA models in addition to the full X-GGM scheme, and \( \mathcal{L}_G \) is more effective than \( \mathcal{L}_D \) for alleviating the unstable gradient issue when training the baseline VQA model with the X-GGM scheme. To further illustrate the effectiveness of \( \mathcal{L}_G \) when training the Baseline with single R-GGM or N-GGM, as shown in Table 6 (#6, #7, #8, #9), we conduct more ablated experiments. Specifically, \#6 only using \( \mathcal{L}_G \) as the adversarial loss to train the baseline VQA model (LXMERT) with the R-GGM. \#7 only using \( \mathcal{L}_D \) as the adversarial loss to train the Baseline with the R-GGM. \#8 only using \( \mathcal{L}_G \) as the adversarial loss to train the Baseline with the N-GGM. \#9 only using \( \mathcal{L}_D \) as the adversarial loss to train the Baseline with the N-GGM. The results on VQA-CP v2 "Other" and GQA-OOD are shown in Table 6, which consistently illustrate that \( \mathcal{L}_G \) is more effective than \( \mathcal{L}_D \) when training the baseline VQA model with a single R-GGM or N-GGM.

Comparisons with SoTAs on VQA-CP v2. Table 7 shows the full comparisons with state-of-the-art debiasing methods on VQA-CP v2. Compared with both KB-based and UB-based methods, the baseline VQA model (LXMERT) trained with the proposed X-GGM scheme can achieve state-of-the-art OOD and ID performance on VQA-CP v2 "Other", further demonstrating the effectiveness of X-GGM to OOD generalization.

Table 6: Full ablation studies on R-GGM and N-GGM with different combinations of \( \mathcal{L}_G \) and \( \mathcal{L}_D \).

| S/N | R-GGM | N-GGM | VQA-CP v2 | GQA-OOD |
|-----|-------|-------|-----------|---------|
|     | \( \mathcal{L}_G \) | \( \mathcal{L}_D \) | \( \mathcal{L}_G \) | \( \mathcal{L}_D \) | OOD↑ | ID↑ | Gap↓ | Tail↑ | Head↑ | \( \Delta \) |
| LX. | – | | 59.98 | 65.21 | 5.23 | 49.80 | 57.70 | 15.90 |  |
| #6 | ✓ | ✓ | 63.37 | 65.71 | 2.34 | 51.36 | 56.95 | 10.88 |  |
| #7 | ✓ | ✓ | 62.67 | 65.21 | 2.54 | 50.80 | 57.13 | 12.45 |  |
| #1 | ✓ | ✓ | 63.84 | 66.41 | 2.57 | 51.39 | 56.93 | 10.77 |  |
| #8 | ✓ | ✓ | 62.77 | 65.56 | 2.79 | 51.08 | 57.93 | 13.41 |  |
| #9 | ✓ | ✓ | 62.00 | 65.14 | 3.14 | 50.99 | 57.76 | 13.28 |  |
| #3 | ✓ | ✓ | 63.75 | 66.03 | 2.28 | 51.65 | 57.13 | 10.61 |  |
| #4 | ✓ | ✓ | 62.73 | 65.99 | 3.26 | 51.27 | 57.19 | 11.54 |  |
| #5 | ✓ | ✓ | 64.33 | 66.92 | 2.59 | 52.47 | 57.51 | 9.59 |  |

Figure 7: Qualitative examples respectively provided by the Baseline (B), i.e., LXMERT, and the Baseline trained with the X-GGM scheme (X). The wrong and correct answers are highlighted in red and green, respectively.

Additional Qualitative Examples. Like Figure 6, Figure 7 shows some additional qualitative examples from the OOD test set of VQA-CP v2 and GQA-OOD provided by the Baseline (i.e., LXMERT) and the Baseline trained with the X-GGM scheme.

Visualization of Relation Matrix. To further illustrate the process of graph generative modeling, considering the direct relation generation (i.e., R-GGM), Figure 8 compares the head maps of the pre-defined relation matrix \( R_{GT} \) and the generated relation matrix \( R_g \) of images which originate from the test set of GQA-OOD in Figure 6 and Figure 7. From the comparison between the heat maps of the pre-fined relation matrix and the generated one, we observe that correlations between nodes becomes more discriminative and new correlations are yielded after R-GGM.

References:

[66] Jiaxin Wu and Raymond Mooney. 2019. Self-critical reasoning for robust visual question answering. In NeurIPS. 8604–8614.

[67] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018).

[68] Chao Yang, Su Feng, Dongsheng Li, Huawei Shen, Guoping Wang, and Bin Jiang. 2020. Learning content and context with language bias for Visual Question Answering. arXiv preprint arXiv:2012.11134 (2020).

[69] Shiwei Guan, Guanlin Li, and Yizhou Yu. 2020. Graph-structured referring expression reasoning in the wild. In CVPR. 9952–9961.

[70] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In ECCV 684–699.

[71] Jiaxuan You, Rex Ying, Xiang Ren, William L Hamilton, and Jure Leskovec. 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. In ICML. 5694–5703.

[72] Yuning You, Tianlong Chen, Yongduo Su, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. arXiv preprint arXiv:2010.13902 (2020).

[73] Wenchao Yu, Cheng Zheng, Wei Cheng, Charu C Aggarwal, Dongjin Song, Bo Zong, Haifeng Chen, and Wei Wang. 2018. Learning deep network representations with adversarially regularized autoencoders. In KDD. 2663–2671.

[74] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In CVPR. 6281–6289.

[75] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and yang: Balancing and answering binary visual questions. In CVPR. 5014–5022.

[76] Wenqiao Zhang, Haochen Shi, Siliang Tang, Jun Xiao, Qiang Yu, and Yueting Zhang. 2021. Consensus graph representation learning for better grounded image captioning. In AAAI.

[77] Shuai Zheng, Zhenfeng Zhu, Xingxing Zhang, Zhihze Liu, Jian Cheng, and Yao Zhao. 2020. Distribution-induced Bidirectional Generative Adversarial Network for Graph Representation Learning. In CVPR. 7224–7233.

[78] Xi Zhu, Zhendong Mao, Chunxiao Liu, Peng Zhang, Bin Wang, and Yongdong Zhang. 2020. Overcoming Language Priors with Self-supervised Learning for Visual Question Answering. In IJCAI. 1083–1089.

[79] Zhao Zhu, Jing Yu, Yujing Wang, Yajing Sun, Yue Hu, and Qi Wu. 2020. Mucko: Multi-Layer Cross-Modal Knowledge Reasoning for Fact-based VisualQuestion Answering. arXiv preprint arXiv:2006.09673 (2020).
Table 7: Comparisons with state-of-the-art debiasing methods on VQA-CP v2. †: methods using additional manually annotated data, ∗: results from our reimplementation using the official released code, —: not reported.

| Method          | VQA-CP v2 (OOD Test) † | VQA-CP v2 (ID Test) † | Gap † |
|-----------------|------------------------|------------------------|-------|
|                 | All Yes/No Num. Other  | All Yes/No Num. Other  | All Yes/No Num. Other |
| KB-based Methods|                        |                        |       |
| RankVQA [52]    | 43.05 42.53 13.19 51.32|                        |       |
| LM [10]         | 48.78 72.78 14.61 45.58|                        |       |
| LMH [10]        | 52.01 72.58 31.12 46.97|                        |       |
| + LRS [25]      | 53.26 72.82 48.00 44.46|                        |       |
| + CCB+VQ-CSS [68]| 59.12 89.12 51.04 45.62|                        |       |
| + MFE [17]      | 54.55 74.03 49.16 45.82|                        |       |
| + CSS+CL [46]   | 59.18 86.99 49.82 48.21|                        |       |
| + CSS+GS [46, 58]| 57.37 79.71 50.85 47.45|                        |       |
| + CSS [7]       | 58.95 84.37 49.42 47.16|                        |       |
| + MCE (LXMERT) [11]| 70.32 — — — |                        |       |
| UB-based Methods|                        |                        |       |
| GVQA [2]        | 31.30 57.99 13.68 22.14|                        |       |
| Actively Seeking [60]| 46.00 58.24 29.49 44.33|                        |       |
| Unshuffling+CF+GS [58]| 46.80 64.50 15.30 45.90| 62.40 77.80 43.80 53.60 | 15.60 13.30 28.50 7.70 |
| Unshuffling+CF [58]| 40.60 61.30 15.60 46.00| 63.30 79.40 45.50 53.70 | 22.70 18.10 29.90 7.70 |
| VGQE [41]       | 50.11 66.35 27.08 46.77|                        |       |
| Unshuffling [59]| 42.39 47.72 14.43 47.24|                        |       |
| UpDn [3]        | 38.82 42.98 12.18 43.95| 64.73 79.45 49.59 55.66 | 25.91 36.47 37.41 11.71 |
| + Top answer masked [61]| 40.61 82.44 27.63 22.26| 30.90 44.12 5.00 20.85 | 9.71 38.32 22.63 1.41 |
| + AdvReg [21]   | 36.33 59.33 14.01 30.41| 50.63 67.39 38.81 38.37 | 14.30 8.06 24.80 7.96 |
| + GRL [21]      | 42.33 59.74 14.78 40.76| 56.90 69.23 42.50 49.36 | 14.57 9.49 27.72 8.60 |
| + RUBi [6]      | 47.11 68.65 20.28 43.18|                        |       |
| + RandImg (λ=12) [61]| 55.37 83.89 41.60 44.20| 54.24 64.22 34.40 50.46 | 1.13 19.67 7.20 6.26 |
| + CF-VQA (Sum) [50]| 53.69 91.25 12.80 45.23|                        |       |
| + CF-VQA (Har.) [50]| 49.94 74.82 18.93 45.42|                        |       |
| + DLR [32]      | 48.87 70.99 18.72 45.57|                        |       |
| + RandImg (λ=5) [61]| 51.15 75.06 24.30 45.99| 59.28 70.66 43.06 53.40 | 8.13 4.40 18.76 7.41 |
| + HINT(HAT) † [53]| 47.70 70.04 10.68 46.31|                        |       |
| + SCR [HAT] † [66]| 49.17 71.55 10.72 47.49|                        |       |
| + SCR [VQA-X] † [66]| 49.45 72.36 10.93 48.02|                        |       |
| + SL [78]       | 57.59 86.53 29.87 50.03|                        |       |
| + MUTANT † [18]| 61.72 88.90 49.68 50.78|                        |       |
| + X-GGM         | 45.71 43.48 27.65 52.34| 67.16 83.74 48.26 56.91 | 21.45 40.26 20.61 4.57 |
| MANGO [44]      | 52.76 — — —            |                        |       |
| LXMERT † [57]  | 63.90 80.45 46.58 59.98| 75.57 91.39 59.83 65.21 | 11.67 10.94 13.25 5.23 |
| + MUTANT † [18]| 69.52 93.15 67.17 57.78|                        |       |
| + VILLA † [15]| 48.06 42.66 18.97 58.89| 75.85 90.40 60.02 66.66 | 27.79 47.97 41.05 7.77 |
| + MCE+Adv [11] | 66.08 — — —            |                        |       |
| + MCE [11]      | 68.44 — — —            |                        |       |
| + X-GGM         | 66.95 78.91 53.63 64.33| 75.88 90.18 60.06 66.92 | 8.93 11.27 6.43 2.59 |
Figure 8: Heat maps of relation matrices. $R_{GT}$ is the pre-defined relation matrix (i.e., the cosine similarity between object-class embeddings and object-attribute embeddings on the image), $R_g$ is the generated relation matrix during R-GGM.