

# Compositional Substitutivity of Visual Reasoning for Visual Question Answering

Chuanhao Li<sup>1,2\*</sup>, Zhen Li<sup>1\*</sup>, Chenchen Jing<sup>3✉</sup>, Yuwei Wu<sup>2,1✉</sup>,  
Mingliang Zhai<sup>1</sup>, and Yunde Jia<sup>2,1</sup>

<sup>1</sup> Beijing Key Laboratory of Intelligent Information Technology,  
School of Computer Science & Technology, Beijing Institute of Technology, China

<sup>2</sup> Guangdong Laboratory of Machine Perception and Intelligent Computing,  
Shenzhen MSU-BIT University, China

<sup>3</sup> School of Computer Science, Zhejiang University, Hangzhou, China  
{lichuanhao, li.zhen, wuyuwei, zhaimingliang, jiyunde}@bit.edu.cn  
jingchenchen@zju.edu.cn

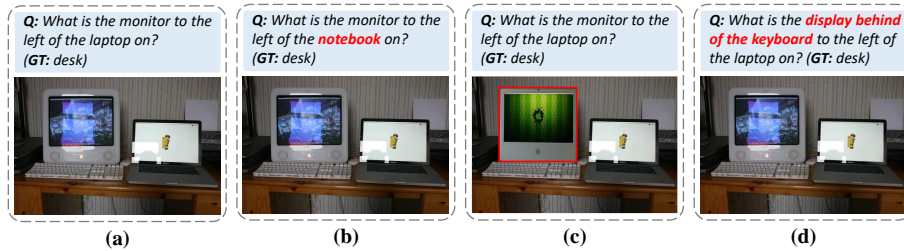
**Abstract.** Compositional generalization has received much attention in vision-and-language and visual reasoning recently. Substitutivity, the capability to generalize to novel compositions with synonymous primitives such as words and visual entities, is an essential factor in evaluating the compositional generalization ability but remains largely unexplored. In this paper, we explore the compositional substitutivity of visual reasoning in the context of visual question answering (VQA). We propose a training framework for VQA models to maintain compositional substitutivity. The basic idea is to learn invariant representations for synonymous primitives via support-sets. Specifically, for each question-image pair, we construct a support question set and a support image set, and both sets contain questions/images that share synonymous primitives with the original question/image. By enforcing a VQA model to reconstruct the original question/image with the sets, the model is able to identify which primitives are synonymous. To quantitatively evaluate the substitutivity of VQA models, we introduce two datasets: GQA-SPS and VQA-SPS v2, by performing three types of substitutions using synonymous primitives including words, visual entities, and referents. Experimental results demonstrate the effectiveness of our framework. We release GQA-SPS and VQA-SPS v2 at <https://github.com/NeverMoreLCH/CG-SPS>.

## 1 Introduction

Compositionality is one of the fundamental properties of human cognition argued by Fodor and Pylyshyn [20]. Compositional generalization, the ability of models to generalize to novel compositions, is critical to simulate the compositional properties of human cognition. Recently, compositional generalization has received much attention in vision-and-language (V&L) and visual reasoning. An essential factor in evaluating the compositional generalization ability is

---

\* equal contribution; ✉ corresponding author: Chenchen Jing and Yuwei Wu.



**Fig. 1:** Illustration of three types of synonymous primitive substitutions for VQA. (a) A sample from GQA [2]. (b) Synonymous word substitution. (c) Synonymous visual entity substitution. (d) Synonymous referent substitution.

*substitutivity*, which refers to the ability to generalize to novel compositions generated via synonymous primitive substitutions. A model with substitutivity can better generalize to novel compositions, because they are able to take advantage of the interchangeability of synonyms to understand the novel composition [8]. Nonetheless, most existing work [21–24] focuses on novel compositions systematically combined by known primitives (*systematicity*), while substitutivity remains largely unexplored.

In this paper, we explore the compositional substitutivity of visual reasoning in the context of visual question answering (VQA). Considering that VQA involves primitives from two modalities, words and visual entities, and there are referential relationships between the two modalities, we divide synonymous primitive substitutions (SPS) in VQA into three types: synonymous word substitutions, synonymous visual entity substitutions, and synonymous referent substitutions, as shown in Fig. 1. Synonymous word substitutions and synonymous visual entity substitutions can be generated by using semantic synonymous words and visual entities to replace corresponding primitives in the original sample, respectively. Synonymous referent substitutions can be generated by using the referential relationships in images to describe the referent in the question.

We present a model-agnostic training framework to maintain the substitutivity of VQA models. The basic idea of the framework is to encourage the model to identify which primitives are synonymous via support-sets. Specifically, the framework mainly consists of two parts: support-set construction and sample reconstruction. For support-set construction, we use back-translation and dataset image retrieval to obtain several support questions and images that share synonymous primitives with the original training sample. For sample reconstruction, we encourage the model to learn the feature representation that the question/image in each training sample can be reconstructed as a weighted combination of its support questions/images at the feature level. In doing so, the model learns to push the feature representation of synonymous primitives together rather than over-fitting individual samples.

To quantitatively evaluate the substitutivity of VQA models, we build two new datasets, *i.e.*, GQA-SPS and VQA-SPS v2, based on the GQA dataset [2]

and the VQA v2 dataset [42], respectively. We automatically generate synonymous words, visual entities and referential expressions for the primitives in original validation samples to construct semantic synonymous samples. Moreover, a consistency metric is introduced to measure whether a VQA model consistently makes correct answers for both the original and constructed sample. Experimental results on GQA, GQA-SPS, VQA v2, VQA-SPS v2, VQA-CP v2 [63] and VQA-Rephrasings [39] demonstrate that our framework not only improves the compositional substitutivity, but also the capability of independent and identically distributed (IID) generalization.

In summary, our contributions are as follows:

- We are the first to explore the compositional substitutivity under multiple types of synonymous primitive substitutions including words, visual entities and referents in the context of VQA, which is critical for evaluating the compositional generalization capability.
- We propose a model-agnostic training framework that improves the substitutivity of VQA models by encouraging the model to identify synonymous primitives.
- We present a GQA-SPS dataset to evaluate the substitutivity of VQA models with different types of synonymous primitive substitutions.

## 2 Related Work

### 2.1 Compositional Generalization

There is a substantial amount of research [14, 31, 32, 34, 51–53] exploring the compositional abilities of neural networks. The compositionality can be viewed from multiple perspectives, including systematicity [31, 32, 53], substitutivity [8, 14], productivity [8, 31], localism [33] and overgeneralisation [34]. In this paper, we focus on the substitutivity of visual reasoning, which remains unexplored.

### 2.2 Compositional Substitutivity

Compositional substitutivity is one of the essential factors in evaluating the compositionality of neural networks [8]. Several works in NLP have demonstrated that existing models exhibit poor capability of compositional substitutivity. For instance, Ren *et al.* [10] generated adversarial examples by word substitution, which successfully attack text classification models. Dankers *et al.* [13] found that machine translation models are hard to maintain consistent outputs after synonym substitution. In V&L, several works [17–19] evaluate models of different V&L tasks including image captioning, VQA in compositional settings, and find that all models struggle for complex tests involving substitutivity. The studies above in both NLP and V&L evaluate the compositional substitutivity of models in language modality, especially at the word level. Different from them, we explore the compositional substitutivity of visual reasoning in both language

and vision modalities, and introduce three types of synonymous primitive substitutions rather than only the synonymous word substitutions.

There are several works [9, 11, 12, 15, 16] for improving the compositional substitutivity. In NLP, Li *et al.* [9] introduced two different representations to improve compositional substitutivity. Zhou *et al.* [11] enhanced the robustness against word substitution-based perturbations using synonyms. Yang *et al.* [12] presented a triplet metric learning strategy to pull words closer to their synonyms and push away to their non-synonyms in the embedding space. In V&L, Whitehead *et al.* [15] presented the VQA P2 dataset to measure the robustness of VQA models under linguistic perturbations, and enhanced the robustness by enforcing consistency in intermediate representations and answers. Gou *et al.* [16] proposed a synonymous sentences-aware attack to deceive natural language video localization models, and defended the attack using adversarial training. The methods above focus on improving the compositional substitutivity under synonymous word substitutions in language modality. By contrast, we propose to improve the compositional substitutivity under three types of synonymous substitutions in both language and vision modalities, by learning invariant representations for synonymous primitives.

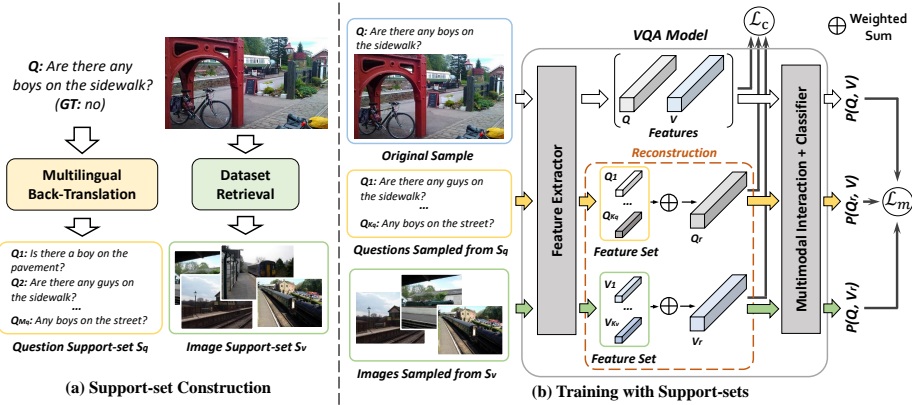
### 2.3 Consistency in VQA

Consistency in VQA can be defined as the ability of a model to generate uncontradicted answers. There are several works that evaluate the consistency of VQA models. For instance, some works [40, 41, 43] are proposed to evaluate the implication consistency, which requires VQA models to produce non-contradictory answers to a series of questions entailed in the same visual fact. Other works [28, 44, 45] measured the perception consistency, *i.e.*, the capability to correctly answer both low-level perception questions and high-level reasoning questions simultaneously. By contrast, our GQA-SPS dataset evaluates synonymous consistency—whether a VQA model correctly answers both a sample and its synonymous samples simultaneously. Shah *et al.* [39] proposed to evaluate the synonymous consistency under sentence-level synonymous rephrasing. Differently, we focus on the consistency under primitive-level synonymous substitutions to evaluate the compositional substitutivity.

## 3 Framework

### 3.1 Overview

VQA aims to provide an answer  $A$  for a natural language question  $Q$  about an image  $V$ . For a given training sample  $(Q, V)$  with the ground-truth answer  $A$ , the question can be denoted as a set of words  $Q = \{q_i\}_{i=1}^{N_q}$ , where  $q_i$  is the  $i$ -th word in the question and  $N_q$  is the number of words in the question. The image can be represented by a set of detected objects  $V = \{v_i\}_{i=1}^{N_v}$ , where  $v_i$  is the  $i$ -th object and  $N_v$  is the total number of objects in the image.



**Fig. 2:** Overview of the proposed training framework. (a) We construct a question support-set and an image-support set for each question and image in the training set by multilingual back-translation and dataset retrieval, respectively. (b) We use support-sets to reconstruct training samples for learning invariant representations of synonymous primitives during training.

The overview of the proposed framework is shown in Fig. 2. Specifically, for a training sample  $(Q, V)$ , we first construct two support sets:  $S_q = \{Q_i\}_{i=1}^{M_q}$  and  $S_v = \{V_i\}_{i=1}^{M_v}$ , where  $Q_i$  and  $V_i$  represent the  $i$ -th support question and image having synonymous primitives with  $Q$  and  $V$ , respectively.  $M_q$  and  $M_v$  denote the element number in  $S_q$  and  $S_v$ , respectively. Then we reconstruct  $Q$  and  $V$  using a weighted combination of the elements in  $S_q$  and  $S_v$  at the feature level, respectively. The question/image obtained by the weighted combination is denoted as  $Q_r/V_r$ . By encouraging the VQA model to make the same predictions for  $(Q, V)$ ,  $(Q_r, V)$  and  $(Q, V_r)$ , and learn similar features for  $Q/V$  and  $Q_r/V_r$ , the model learns to push the feature representation of synonymous primitives together, thus improving compositional substitutivity.

### 3.2 Support-Set Construction

**Question Support-Set.** To ensure that the support questions and the original question share synonymous primitives, *i.e.*, synonyms, we use a pretrained multilingual neural machine translation model mBART [4] to generate support questions that are semantically synonymous to the original sample by the back-translation mechanism [5]. We use 24 different intermediate languages, *i.e.*,  $M_q = 24$ , and use mBART to translate the original English question into each intermediate language, and then translate it back to English to obtain support questions. To improve the quality of support questions, we filter out questions with issues automatically by several rules summarized from empirical experiments. The questions will be filtered out if they (1) differ significantly in length from the original questions. (2) contain repeated substrings. (3) have simple syntax and punctuation errors. Furthermore, we randomly replicate questions in the

support-set until the question number of the support-set reaches  $M_q$ , to ensure that each question has the same number of support questions.

**Image Support-Set.** For each image, we retrieve similar images in the training set to generate support images. We use the pretrained CLIP model [6] to compute the similarity of each image to all training images except itself, and select top- $M_v$  images with the highest similarities as its image support-set. We empirically set  $M_v = 128$  for all experiments.

### 3.3 Sample Reconstruction

After pre-processing the question/image support-sets and image support-sets for all training samples, we train a VQA model using them. For each sample  $(Q, V)$  during training, we sample a sub-set  $T_q/T_v$  from its  $S_q/S_v$ , where  $T_q$  and  $T_v$  contain  $K_q$  and  $K_v$  elements respectively. We firstly use the feature extractors  $f_q(\cdot)$  and  $f_v(\cdot)$  in the training VQA model to obtain deep features for words and objects by

$$\mathbf{h}_q = f_q(Q), \mathbf{h}_v = f_v(V), \quad (1)$$

where  $\mathbf{h}_q$  denotes the word-level feature of  $Q$ , and  $\mathbf{h}_v$  denotes the object-level feature of  $V$ . Then we measure the similarities between  $Q/V$  and the elements in  $T_q/T_v$  by

$$\mathbf{u}_q = \text{softmax}(\{g(\mathbf{h}_q, f_q(e)) | e \in T_q\}), \mathbf{u}_v = \text{softmax}(\{g(\mathbf{h}_v, f_v(e)) | e \in T_v\}), \quad (2)$$

where  $g(\cdot, \cdot)$  is a function that measures the similarities of input vectors by

$$g(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}. \quad (3)$$

The obtained similarities are used as the weights to reconstruct a question/image using the sampled support-sets at the feature level by

$$\mathbf{h}_{q_r} = \sum_{i=1}^{K_q} \mathbf{u}_q^i \times f_q(T_q^i), \mathbf{h}_{v_r} = \sum_{i=1}^{K_v} \mathbf{u}_v^i \times f_v(T_v^i), \quad (4)$$

where  $\mathbf{u}_q^i/\mathbf{u}_v^i$  represents  $i$ -th similarity value in  $\mathbf{u}_q/\mathbf{u}_v$ , and  $T_q^i/T_v^i$  denotes  $i$ -th support question/image in  $T_q/T_v$ . For simplicity, we use  $Q_r$  and  $V_r$  to represent the reconstructed question and image respectively, as  $\mathbf{h}_{q_r}$  and  $\mathbf{h}_{v_r}$  can be viewed as the deep features extracted by the VQA model for them.

### 3.4 Optimization

We use two different losses to supervise the training process, including a method-specific loss  $\mathcal{L}_m$  and a contrastive learning loss  $\mathcal{L}_c$ .

**Method-Specific Loss.** The  $\mathcal{L}_m$  is determined by the selected method, since different methods use different training losses. For a training sample  $(Q, V)$  with ground-truth  $A$ , the  $\mathcal{L}_m$  is computed by

$$\mathcal{L}_m = \text{loss}(P(Q, V), A), \quad (5)$$

where  $P(Q, V)$  represents the output of the VQA model (*e.g.*, distribution vector with size of the number of categories), and  $\text{loss}(\cdot, \cdot)$  denotes the loss function used in the selected method, such as the cross-entropy loss used in UpDn [25].

As the reconstructed sample  $(Q_r, V)/(Q, V_r)$  maintains the same semantics as  $(Q, V)$ , we use the same loss and ground-truth to train  $(Q, V)/(Q_r, V)$ . Thus, the loss  $\mathcal{L}_m$  is reformulated as

$$\mathcal{L}_m = \text{loss}(P(Q, V), A) + \lambda_q \text{loss}(P(Q_r, V), A) + \lambda_v \text{loss}(P(Q, V_r), A), \quad (6)$$

where  $\lambda_q$  and  $\lambda_v$  are hyper-parameters to balance different types of samples.

**Contrastive Learning Loss.** To enforce the model to learn invariant representations for synonymous primitives, we encourage the VQA model to learn the feature representation for which the  $Q/V$  is similar to  $Q_r/V_r$  rather than other questions/images. Compared to  $Q_i/V_i$  (the  $i$ -th support question/image),  $Q_r/V_r$  has a more stable similarity with  $Q/V$  as it avoids wrong primitive feature alignment via weighted sum, as shown in Fig. 3. If the support question/image is directly used as the positive sample for contrastive learning, the wrong primitive feature alignment may lead to reduced discriminability of the learned primitive features. As a result, we use a contrastive learning loss to pull the features of  $Q/V$  and  $Q_r/V_r$  close, while pushing the features of  $Q/V$  and other questions/images away. The loss is computed by

$$\mathcal{L}_c = -\log \left( \frac{e^{g(\mathbf{h}_q, \mathbf{h}_{q_r})} + e^{g(\mathbf{h}_v, \mathbf{h}_{v_r})}}{e^{g(\mathbf{h}_q, \mathbf{h}_{q_r})} + e^{g(\mathbf{h}_v, \mathbf{h}_{v_r})} + e^{g(\mathbf{h}_q, \mathbf{h}_{q_-})} + e^{g(\mathbf{h}_v, \mathbf{h}_{v_-})}} \right), \quad (7)$$

where  $\mathbf{h}_{q_-}$  and  $\mathbf{h}_{v_-}$  denote a question feature and an image feature sampled from the current training batch, respectively.

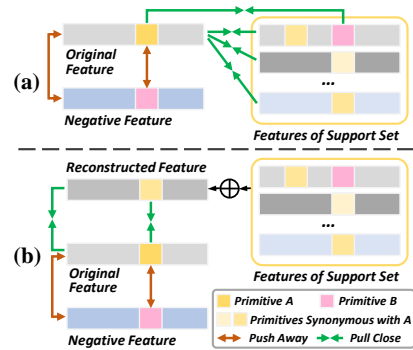
In summary, the total loss for training the VQA model can be viewed as

$$\mathcal{L} = \mathcal{L}_m + \lambda_c \mathcal{L}_c, \quad (8)$$

where  $\lambda_c$  is a hyper-parameter that balances the loss term.

## 4 GQA-SPS Dataset

In this section, we introduce the GQA-SPS dataset, which enables the quantitative evaluation of the compositional substitutivity of VQA models. For the



**Fig. 3:** (a) Contrastive learning without reconstruction. (b) Contrastive learning with reconstruction.

construction of the VQA-SPS v2 dataset, please refers to the **supplementary material**. We construct three pairs of validation splits by performing synonymous substitutions on three types of primitives, words, visual entities and referents. Each pair contains a val-A and a val-B split, the samples in the val-A split are derived from the val-all split of the GQA dataset [2], and the samples in the val-B split are obtained by performing synonymous primitive substitutions on the samples in the val-A split. The above three pairs of validation splits are called Word SPS, Visual Entity SPS and Referent SPS, respectively.

#### 4.1 Sample Generation Pipeline

In the following, we illustrate how we generate samples for the three pairs of validation splits. We first build a word vocabulary  $\mathcal{D}$  based on the train-balanced and val-balanced splits of GQA, because most reasoning models [1, 28, 29] use them for training and validation, respectively. To ensure the applicability of the GQA-SPS dataset to these methods, we first collect all samples from the val-all split of GQA, which do not contain words outside  $\mathcal{D}$ , as the initial sample set  $\mathcal{I}$ . **Word SPS.** For each word in the vocabulary, we use WordNet [30] to obtain all the word sets it belongs to, and each word set represents a unique semantic of the word in a specific context, then we get the synonyms of the word under each semantic. For example, the word “plane” belongs to different word sets. In a word set, the definition of “plane” is “an aircraft that has a fixed wing and is powered by propellers or jets”, and has a synonym “airplane”. While in another word set, “plane” represents “an unbounded two-dimensional shape”, and has a different synonym “sheet”. To ensure the rationality of synonymous word substitutions, we collect words that satisfy both conditions: (1) The word belongs to only one word set. (2) The word has at least one synonym that is not in the vocabulary. We collect samples containing words that meet the above conditions from  $\mathcal{I}$  as val-A, and replace these words with synonyms to form val-B.

**Visual Entity SPS.** We construct Visual Entity SPS based on the initial sample set  $\mathcal{I}$ . For an image  $V$ , we first collect its objects that are related with at least a question in  $\mathcal{I}$ , and remove the larger one if two objects overlap. We denote the objects as a set  $\mathcal{J}_V = \{v_i\}_{i=1}^{N_V}$ , where  $v_i$  is the  $i$ -th object and  $N_V$  is the object number. For each object, we use GLIGEN [35], a large-scale text-to-image generation method, to redraw an object synonymous with its attributes and name provided by GQA. By repeatedly using GLIGEN, we obtain a new image synonymous with  $V$ , in which  $N_V$  objects are redrawn. We discard new images of poor quality based on manual review, and use the remaining new images and their associated questions to form val-B, Val-A is derived by substituting the new images in val-B with their corresponding original images.

**Referent SPS.** For each sample  $(Q, V)$  in  $\mathcal{I}$ , GQA provides a scene graph  $G = \{(S_i, O_i, R_i)\}_{i=1}^{M_R}$ , where  $(S_i, O_i, R_i)$  represents the  $i$ -th relational triple in  $V$ .  $S_i$ ,  $O_i$  and  $R_i$  denote the subject, object and relationship, respectively. To generate the referential expressions with unique referentiality, we devise three rules to filter samples in  $\mathcal{I}$ : (1) There is at least a subject  $S_u$  that is unique in  $V$ , and the subject name appears in  $Q$ . (2) There is an object  $O_u$  that only has



a relationship  $R_u$  that is unique with  $S_u$ . (3) The object  $O_u$  is unique in  $V$ . We screen out the samples that meet the above three rules at the same time, as the val-A split. To construct the val-B split, we substitute the subject name  $S_u$  in questions with the referential expression “ $H_u + R_u +$  the  $O_u$ ” for each sample in val-A, where  $H_u$  is the hypernym of  $S_u$  obtained through manual annotation. For example, for the question “What is the apple on?” with a relational triple (apple, plate, to the left of) that meets the above three rules simultaneously, and “fruit” is the hypernym of “apple”. We substitute the word “apple” with the synonymous referential expression “fruit to the left of the plate” to generate a new question “What is the fruit to the left of the plate on?”.

## 4.2 Dataset Analysis

Compared to the 132062 samples in the validation split of the GQA dataset, we obtain 150074, 23410 and 123147 samples for Word SPS, Visual Entity SPS and Referent SPS, respectively. The sample numbers of Word SPS and Referent SPS are on the same order of magnitude as the validation split of GQA. The reason why Visual Entity SPS has a smaller sample number is that we are strict about the quality of the images generated by GLIGEN.

## 4.3 Consistency Score

An ideal VQA model should generate correct answers for not only the sample from val-A but also its paired sample from val-B. To this end, we devise a consistency metric  $Cons$ , which measures the consistency of VQA models to correctly answer paired synonymous samples.  $Cons$  is computed by

$$Cons = \left( \sum_{(s_a, s_b) \in (D_A, D_B)} E(P(s_a), P(s_b)) \right) / |D_A|, \quad (9)$$

where  $(D_A, D_B)$  denotes a pair of val-A and val-B in GQA-SPS,  $(s_a, s_b)$  denotes a pair of samples from  $D_A$  and  $D_B$  respectively,  $E(\cdot, \cdot)$  is an indicator function that outputs 1 when the two inputs are both correct and outputs 0 otherwise,  $P(\cdot)$  represents the output of the VQA model for the input question-image pair, and  $|\cdot|$  is the sample number of the input dataset.

# 5 Experiments

## 5.1 Experimental Settings

**Baselines.** We incorporate the proposed framework into five VQA models including MAC [1], LXMERT [3], ViLT [46], mPLUG [54] and BEiT-3 [55]. MAC is a popular foundational model used in compositional generalization. LXMERT is a two-stage representative transformer-based pre-trained model for vision-and-language reasoning, which inputs object-level visual features. ViLT, mPLUG and BEiT-3 are typical one-stage pre-trained models that inputs raw images

without processing. As our framework is based on SUPport Set (SUPS), the trained five models are called MAC+SUPS, LXMERT+SUPS, ViLT+SUPS, mPLUG+SUPS, BEiT-3+SUPS respectively.

**Datasets.** We evaluate the proposed framework on GQA [2], VQA-Rephrasings [39] and our GQA-SPS. GQA-SPS is proposed for testing the compositional substitutivity, while GQA is used for testing the IID generalization. The reason for choosing GQA is to evaluate the compatibility of compositional generalization and IID generalization. VQA-Rephrasings is an extension of the VQA v2 dataset [42] and is used for evaluating the consistency of VQA models to synonymous rephrasing of questions. To validate the generalizability of our framework across datasets, we also perform experiments on VQA v2 [42], VQA-CP v2 [63] and our VQA-SPS v2, which are provided in the **supplementary material**.

**Implementation Details.** For evaluation on GQA and GQA-SPS, we train or finetune the models using the train split of GQA, and select a checkpoint that performs best on the validation split of GQA. Based on the checkpoint, we report the experimental results on the test-dev split of GQA and all validation splits of GQA-SPS. For evaluation on VQA-Rephrasings, we use the train split and a subset of the validation split of VQA v2 for training and finetuning and checkpoint selection respectively, note that the subset does not contain any test samples from VQA-Rephrasings. In addition, more implementation details are provided in the **supplementary material**.

## 5.2 Evaluation of Compositional Substitutivity

We evaluate the compositional substitutivity on the proposed GQA-SPS dataset. In addition to the above five baselines, we test 11 representative VQA models including large vision-language models varies in parameters (4B to 17B), VL-T5 [38], OpenFlamingo-4B [36], BLIP-2-FlanT5<sub>XL</sub> [37], QWen-VL-7B [57], CogVLM-17B [58], LLaVa-v1.5-7B-Xtuner [60], mPLUG-Owl2-LLaMA2-7b [59], MMAlaya [61], XComposer2-7B [56] and Gemini 1.0 Pro [64]. For VL-T5, we finetune it on the train split of GQA, and select a checkpoint for evaluation in the same way as the five baselines. For other models, *i.e.*, large vision-language models, we implement them based on the VLMEvalKit toolkit [62], and evaluate them at a zero-shot paradigm.

The results on the GQA-SPS dataset are listed in Tab. 1, from which we can observe that: (1) Our framework consistently improves the performance of five baselines. (2) Our framework improves the answer accuracy and consistency simultaneously under all three types of SPS by a large margin (*e.g.*, 6.1% and 8.1% absolute accuracy and consistency gains for MAC, respectively, under word SPS). (3) Large vision-language models achieve dissatisfactory compositional substitutivity although they’ve been trained on a large amount of visual question answering samples. These observations show that the proposed support-set based training framework is effective in improving the compositional substitutivity of VQA models for different baselines.

**Table 1:** Accuracy (%) and Consistency (%) of the state-of-the-arts on GQA-SPS, where “*Acc1*” and “*Acc2*” represent the accuracy on val-A (samples sampled from the validation split of GQA) and val-B (samples by performing SPS on the samples in val-A), respectively, “*Cons*” is the consistency score mentioned in Section 4.3, and “*HM*” is the harmonic mean of consistency scores on different test pairs of GQA-SPS.

| Visual Input    | Method              | Word SPS    |             |             | Vis. Entity SPS |             |             | Referent SPS |             |             | HM          |
|-----------------|---------------------|-------------|-------------|-------------|-----------------|-------------|-------------|--------------|-------------|-------------|-------------|
|                 |                     | <i>Acc1</i> | <i>Acc2</i> | <i>Cons</i> | <i>Acc1</i>     | <i>Acc2</i> | <i>Cons</i> | <i>Acc1</i>  | <i>Acc2</i> | <i>Cons</i> | <i>Cons</i> |
| Raw Images      | OpenFlamingo [36]   | 47.8        | 48.0        | 37.4        | 70.8            | 71.0        | 64.3        | 51.0         | 49.6        | 42.7        | 45.7        |
|                 | BLIP-2 [37]         | 62.8        | 62.1        | 57.7        | 50.5            | 55.5        | 44.6        | 55.2         | 51.3        | 43.2        | 47.7        |
|                 | QWen-VL [57]        | 62.6        | 58.2        | 56.3        | 56.2            | 45.7        | 42.9        | 42.8         | 37.0        | 34.1        | 42.6        |
|                 | CogVLM [58]         | 69.2        | 65.0        | 62.2        | 78.2            | 78.4        | 75.3        | 57.1         | 48.1        | 44.1        | 57.7        |
|                 | LLaVa-v1.5 [60]     | 66.0        | 62.4        | 59.3        | 75.7            | 76.6        | 72.8        | 56.0         | 50.1        | 45.1        | 56.8        |
|                 | mPLUG-Owl2 [59]     | 61.5        | 59.1        | 55.6        | 75.3            | 75.5        | 71.8        | 51.8         | 45.9        | 41.7        | 53.7        |
|                 | MMAIaya [61]        | 58.8        | 56.9        | 52.4        | 64.8            | 63.7        | 60.0        | 47.0         | 41.6        | 35.0        | 46.6        |
|                 | XComposer2 [56]     | 54.3        | 50.2        | 46.5        | 72.0            | 72.5        | 67.6        | 47.0         | 40.8        | 36.4        | 47.0        |
|                 | Gemini 1.0 Pro [64] | 59.6        | 57.8        | 51.8        | 70.2            | 71.8        | 65.2        | 59.8         | 54.0        | 44.8        | 52.7        |
|                 | ViLT [46]           | 70.0        | 64.2        | 58.1        | 75.0            | 75.5        | 68.6        | 70.3         | 56.2        | 47.9        | 57.0        |
|                 | + <b>SUPS</b>       | 72.5        | 68.0        | 63.3        | 75.9            | 76.3        | 69.5        | 71.6         | 60.8        | 53.7        | 61.5        |
|                 | mPLUG [54]          | 71.7        | 67.6        | 62.6        | 77.1            | 77.6        | 71.0        | 67.9         | 60.7        | 52.5        | 61.1        |
|                 | + <b>SUPS</b>       | 73.0        | 68.9        | 64.6        | 78.4            | 79.4        | 72.9        | 69.5         | 62.1        | 55.5        | 63.6        |
|                 | BEiT-3 [55]         | 78.3        | 72.5        | 68.9        | 78.9            | 80.1        | 73.0        | 76.9         | 62.3        | 56.4        | 65.3        |
| + <b>SUPS</b>   | <b>79.2</b>         | <b>73.0</b> | <b>69.2</b> | <b>79.8</b> | <b>81.1</b>     | <b>73.9</b> | <b>78.2</b> | <b>63.8</b>  | <b>57.8</b> | <b>66.2</b> |             |
| Object Features | VL-T5 [38]          | 78.0        | 71.4        | 67.6        | 79.5            | 79.8        | 72.9        | 77.0         | 59.8        | 52.9        | 63.3        |
|                 | MAC [1]             | 63.7        | 56.3        | 47.5        | 65.2            | 66.5        | 57.7        | 62.7         | 49.5        | 36.9        | 45.8        |
|                 | + <b>SUPS</b>       | 68.6        | 62.4        | 55.6        | 68.2            | 69.9        | 61.5        | 66.1         | 54.5        | 44.1        | 52.7        |
|                 | LXMERT [3]          | 79.8        | 73.5        | 69.1        | 81.0            | 79.6        | 73.3        | 79.9         | 63.1        | 56.6        | 65.5        |
| + <b>SUPS</b>   | <b>80.7</b>         | <b>74.7</b> | <b>70.9</b> | <b>82.1</b> | <b>80.6</b>     | <b>74.8</b> | <b>80.5</b> | <b>64.1</b>  | <b>58.4</b> | <b>67.3</b> |             |

### 5.3 Evaluation of Synonymous Rephrasing

Unlike our GQA-SPS dataset that focuses on primitive-level synonymous substitutions to evaluate the compositional substitutivity, the VQA-Rephrasings dataset is used to evaluate the consistency of VQA models under sentence-level synonymous rephrasing. In VQA-Rephrasings, the question of test samples is generated by human rephrasing, which is significantly different in style from the questions generated by back-translation used in our framework. Even so, we observe that our framework improves LXMERT with 0.56% and 0.9% absolute gains in the mean accuracy and the strictest consistency of rephrased questions (*i.e.*, REP and CS(4)), respectively, as the experimental results listed in Tab. 2. The observations demonstrate that our framework can effectively improve VQA models not only under primitive-level synonymous substitutions, but also under sentence-level synonymous rephrasing.

**Table 2:** Accuracy (%) of the state-of-the-arts on VQA-Rephrasings [39].

| Method                           | CS(k)        |              |              |              | VQA Acc      |              |
|----------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                                  | $k=1$        | $k=2$        | $k=3$        | $k=4$        | <i>ORI</i>   | <i>REP</i>   |
| BAN [47]                         | 64.88        | 53.08        | 47.45        | 39.87        | 64.97        | 55.87        |
| BAN+CC [39]                      | 65.77        | 56.94        | 51.76        | 48.18        | 65.87        | 56.59        |
| MANGO <sub>VB</sub> [49]         | 72.78        | 65.97        | 61.70        | 58.59        | -            | -            |
| ConClaT [48]                     | 68.62        | 61.42        | 57.08        | 53.99        | -            | -            |
| BLIP-2 OPT <sub>6.7B</sub> [37]  | 50.23        | 43.86        | 40.64        | 38.59        | 46.07        | 44.31        |
| BLIP-2 FlanT5 <sub>XL</sub> [37] | 56.50        | 50.77        | 47.64        | 45.50        | 53.68        | 51.09        |
| mPLUG [54]                       | 72.66        | 67.05        | 62.68        | 60.14        | 71.71        | 65.32        |
| LXMERT [3]                       | 73.82        | 67.42        | 63.08        | 60.03        | 72.32        | 65.10        |
| + <b>SUPS (Ours)</b>             | <b>74.46</b> | <b>67.98</b> | <b>63.89</b> | <b>60.93</b> | <b>72.51</b> | <b>65.66</b> |

#### 5.4 Evaluation of IID Generalization

The experimental results on the GQA dataset are listed in Tab. 3. We observe from the table that our framework improves the accuracy of all five baselines. The reason why the performance gains of the proposed framework on GQA are limited is that we mainly focus on the compositional substitutivity of VQA models, which can be viewed as a capability of out-of-distribution (OOD) generalization, while the GQA dataset is more suitable to evaluate IID generalization. The experimental results demonstrate that the improvements of our framework are compatible with compositional generalization and IID generalization.

**Table 3:** Accuracy (%) of state-of-the-arts on the test-dev split of GQA [2]. (a) Methods that use raw images as visual input. (b) Methods that use object features as visual input.

| Method         | Acc         | Method        | Acc         |
|----------------|-------------|---------------|-------------|
| BLIP-2 [37]    | 44.7        | LCGN [29]     | 55.8        |
| MiniGPT-4 [50] | 43.5        | MMN [26]      | 60.4        |
| ViLT [46]      | 56.8        | VL-T5 [38]    | 58.4        |
| + <b>SUPS</b>  | 57.1        | MDETR [27]    | <b>63.0</b> |
| mPLUG [54]     | 59.7        | MAC [1]       | 53.1        |
| + <b>SUPS</b>  | 60.3        | + <b>SUPS</b> | 53.5        |
| BEiT-3 [55]    | 60.8        | LXMERT [3]    | 59.7        |
| + <b>SUPS</b>  | <b>61.8</b> | + <b>SUPS</b> | 60.1        |

(a)

(b)

#### 5.5 Ablation Studies

The results of ablation studies on GQA-SPS are shown in Tab. 4, in which we use LXMERT [3] as the baseline method. Firstly, we investigate whether using the support set to construct samples to perform data augmentation can improve the compositional substitutivity. We observe that the performance drops, which maybe caused by the uncontrolled sample quality. Then, we study the influences of constructing support sets using our multilingual back-translation and dataset

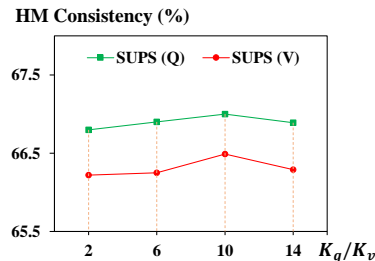
**Table 4:** Ablation studies on GQA-SPS, where “DA” means data augmentation, “OSB” means constructing support sets using Others questions/images in the Same Batch [7].

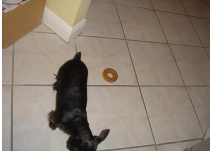




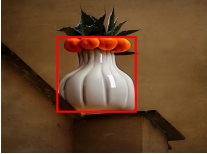
| Method           | SUPS(Q) | SUPS(V) | Consistency  |               |              |  |
|------------------|---------|---------|--------------|---------------|--------------|--|
|                  |         |         | Word         | Visual Entity | Referent     |  |
| LXMERT [3]       |         |         | 69.06        | 73.30         | 56.63        |  |
| + DA             | ✓       |         | 70.27        | 69.83         | 55.62        |  |
|                  | ✓       | ✓       | 60.90        | 70.51         | 49.21        |  |
| + OSB [7]        | ✓       |         | 64.40        | 71.04         | 51.55        |  |
|                  | ✓       | ✓       | 69.57        | 73.48         | 58.21        |  |
| + OSB [7]        | ✓       |         | 69.96        | 73.58         | 57.37        |  |
|                  | ✓       | ✓       | 69.53        | 73.56         | 57.27        |  |
| + SUPS<br>(Ours) | ✓       |         | <b>71.43</b> | 73.26         | 58.26        |  |
|                  | ✓       | ✓       | 69.89        | 74.78         | 57.35        |  |
|                  | ✓       | ✓       | 70.92        | <b>74.81</b>  | <b>58.39</b> |  |

retrieval. We use a different method that uses samples in a same training batch to build support sets [7], and observe worse performance than our framework. One possible explanation is that there are large differences between different samples in VQA, and samples in the same batch do not necessarily share synonymous primitives. Moreover, we observe that the performance improvement on a single modality is significant when only using the question/image support-set. These observations suggest that all components of our framework are effective in improving baseline methods, and components are complementary to each other.

## 5.6 Parameter Analysis

We analyze the influences of  $K_q$  and  $K_v$  on the consistency of our framework, which denote the sampled numbers of support questions and support images for each training sample, respectively. The consistency of the LXMERT+SUPS with different  $K_q$  and  $K_v$  on GQA-SPS are shown in Fig. 4, which demonstrates that the performance of LXMERT+SUPS grows as the sampled number of support questions/images grows. However, the performance drops when the sampled number exceeds 10, possibly due to the fact that it becomes increasingly difficult to learn which primitives are synonymous as the sampled number increases. As a result, we set both  $K_q$  and  $K_v$  as 10 for LXMERT+SUPS.

**Fig. 4:** Parameter analysis. For SUPS (Q), the x-coordinate variable is  $K_q$ . For SUPS (V), the x-coordinate variable is  $K_v$ .

|                   |                                                                                   |                                                                                                  |                                                                                                                |                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| Test Samples      |  |                 |                              |                              |
|                   | Q: How big is the dog?<br>(GT: small)                                             | Q: Is there a kid or a woman in this photo? (GT: yes)                                            | Q: How is the yellow clothing item called? (GT: sweater)                                                       | Q: What do you think is in the vase?<br>(GT: plant)                                                             |
|                   | Prediction (LXMERT): small ✓<br>Prediction (Ours): small ✓                        | Prediction (LXMERT): yes ✓<br>Prediction (Ours): yes ✓                                           | Prediction (LXMERT): sweater ✓<br>Prediction (Ours): sweater ✓                                                 | Prediction (LXMERT): plant ✓<br>Prediction (Ours): plant ✓                                                      |
|                   | <hr/>                                                                             |                                                                                                  |                                                                                                                |                                                                                                                 |
| Samples under SPS | <b>Word SPS</b><br>Q: How big is the <b>puppy</b> ?                               | <b>Word SPS</b><br>Q: Is there a <b>child</b> or an <b>adult female</b> in this <b>picture</b> ? | <b>Visual Entity SPS</b><br> | <b>Visual Entity SPS</b><br> |
|                   | Prediction (LXMERT): large ✗<br>Prediction (Ours): small ✓                        | Prediction (LXMERT): no ✗<br>Prediction (Ours): yes ✓                                            | Prediction (LXMERT): glove ✗<br>Prediction (Ours): sweater ✓                                                   | Prediction (LXMERT): carrot ✗<br>Prediction (Ours): plant ✓                                                     |
|                   | <b>Referent SPS</b><br>Q: How big is the <b>animal to the left of the toy</b> ?   | <b>Referent SPS</b><br>Q: Is there a kid or a <b>female talking on the phone</b> in this photo?  |                                                                                                                |                                                                                                                 |
|                   | Prediction (LXMERT): large ✗<br>Prediction (Ours): small ✓                        | Prediction (LXMERT): no ✗<br>Prediction (Ours): yes ✓                                            |                                                                                                                |                                                                                                                 |

**Fig. 5:** Qualitative comparisons between LXMERT+SUPS (Ours) and LXMERT. The red words in questions and the red boxes in images denote synonymous primitives.

## 5.7 Qualitative Analysis

Fig. 5 depicts several qualitative examples from the GQA-SPS dataset between LXMERT and LXMERT+SUPS. We observe that LXMERT cannot make correct predictions when the primitives in the sample are replaced with synonymous primitives, while LXMERT+SUPS can make predictions accurately. For instance, in the first example, LXMERT can correctly answer the question “How big is the dog?”, but fails when simply replacing the “dog” with its synonym “puppy”, indicating that LXMERT does not learn that “dog” is synonymous to “puppy”. These qualitative examples show that our framework can help LXMERT maintain consistent answers for a sample and its synonymous samples, which proves that the framework is effective for learning the synonymous relationships between primitives. More qualitative examples are given in the **supplementary material**.

## 6 Conclusion

In this paper, we explored the compositional substitutivity of visual reasoning in the context of VQA. We have presented a model-agnostic training framework to encourage VQA models to identify synonymous primitives by learning invariant representations via support-sets. The proposed framework can be seamlessly incorporated into existing VQA models to improve their compositional substitutivity. We constructed a GQA-SPS dataset and a VQA-SPS v2 dataset, which enable the quantitative evaluation of the substitutivity for VQA models. Experimental results demonstrate that our framework can improve not only the OOD generalization capability of synonymous substitutions and rephrasing, but also the IID generalization capability.

**Acknowledgments** This work was supported by the Natural Science Foundation of China (NSFC) under Grants No. 62172041 and No. 62176021, Natural Science Foundation of Shenzhen under Grant No. JCYJ20230807142703006, Key Research Platforms and Projects of the Guangdong Provincial Department of Education under Grant No.2023ZDZX1034, and the China Postdoctoral Science Foundation (No. 2023M743003).

## References

1. D. A. Hudson and C. D. Manning, “Compositional attention networks for machine reasoning,” in *Int. Conf. Learn. Represent.*, 2018.
2. —, “Gqa: A new dataset for real-world visual reasoning and compositional question answering,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 6700–6709.
3. H. Tan and M. Bansal, “Lxmert: Learning cross-modality encoder representations from transformers,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019, pp. 5100–5111.
4. Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, “Multilingual denoising pre-training for neural machine translation,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.
5. R. Sennrich, B. Haddow, and A. Birch, “Improving neural machine translation models with monolingual data,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 86–96.
6. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
7. M. Patrick, P.-Y. Huang, Y. Asano, F. Metze, A. Hauptmann, J. Henriques, and A. Vedaldi, “Support-set bottlenecks for video-text representation learning,” *arXiv preprint arXiv:2010.02824*, 2020.
8. D. Hupkes, V. Dankers, M. Mul, and E. Bruni, “Compositionality decomposed: How do neural networks generalise?” *Journal of Artificial Intelligence Research*, vol. 67, pp. 757–795, 2020.
9. Y. Li, L. Zhao, J. Wang, and J. Hestness, “Compositionality generalization for primitive substitutions,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019, pp. 4293–4302.
10. S. Ren, Y. Deng, K. He, and W. Che, “Generating natural language adversarial examples through probability weighted word saliency,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1085–1097.
11. Y. Zhou, X. Zheng, C.-J. Hsieh, K.-W. Chang, and X.-J. Huang, “Defense against synonym substitution-based adversarial attacks via dirichlet neighborhood ensemble,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 5482–5492.
12. Y. Yang, X. Wang, and K. He, “Robust textual embedding against word-level adversarial attacks,” in *Uncertainty in Artificial Intelligence*, 2022, pp. 2214–2224.
13. V. Dankers, E. Bruni, and D. Hupkes, “The paradox of the compositionality of natural language: A neural machine translation case study,” in *Proceedings of the*

- Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 4154–4175.
14. K. Kudo, Y. Aoki, T. Kuribayashi, A. Brassard, M. Yoshikawa, K. Sakaguchi, and K. Inui, “Do deep neural networks capture compositionality in arithmetic reasoning?” in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 1343–1354.
  15. S. Whitehead, H. Wu, Y. R. Fung, H. Ji, R. Feris, and K. Saenko, “Learning from lexical perturbations for consistent visual question answering,” *arXiv preprint arXiv:2011.13406*, 2020.
  16. W. Gou, W. Shi, J. Lou, L. Huang, P. Zhou, and R. Li, “Sneak: Synonymous sentences-aware adversarial attack on natural language video localization,” *arXiv preprint arXiv:2112.04154*, 2021.
  17. T. Klinger, D. Adjudah, V. Marois, J. Joseph, M. Riemer, A. Pentland, and M. Campbell, “A study of compositional generalization in neural models,” *arXiv preprint arXiv:2006.09437*, 2020.
  18. G. Pantazopoulos, A. Suglia, and A. Eshghi, “Combine to describe: Evaluating compositional generalization in image captioning,” in *Proceedings of the Association for Computational Linguistics: Student Research Workshop*, 2022, pp. 115–131.
  19. T. Stoikou, M. Lymperaïou, and G. Stamou, “Knowledge-based counterfactual queries for visual question answering,” *arXiv preprint arXiv:2303.02601*, 2023.
  20. J. A. Fodor and Z. W. Pylyshyn, “Connectionism and cognitive architecture: A critical analysis,” *Cognition*, vol. 28, no. 1-2, pp. 3–71, 1988.
  21. B. Bogin, S. Subramanian, M. Gardner, and J. Berant, “Latent compositional representations improve systematic generalization in grounded question answering,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 195–210, 2021.
  22. J. Shi, H. Zhang, and J. Li, “Explainable and explicit visual reasoning over scene graphs,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 8376–8384.
  23. A. Akula, V. Jampani, S. Changpinyo, and S.-C. Zhu, “Robust visual reasoning via language guided neural module networks,” in *Adv. Neural Inform. Process. Syst.*, 2021, pp. 11 041–11 053.
  24. J. Jiang, Z. Liu, Y. Liu, Z. Nan, and N. Zheng, “X-ggm: Graph generative modeling for out-of-distribution generalization in visual question answering,” in *ACM Int. Conf. Multimedia*, 2021, pp. 199–208.
  25. P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6077–6086.
  26. W. Chen, Z. Gan, L. Li, Y. Cheng, W. Wang, and J. Liu, “Meta module network for compositional visual reasoning,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 655–664.
  27. A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, “Mdetr-modulated detection for end-to-end multi-modal understanding,” in *Int. Conf. Comput. Vis.*, 2021, pp. 1780–1790.
  28. C. Jing, Y. Jia, Y. Wu, X. Liu, and Q. Wu, “Maintaining reasoning consistency in compositional visual question answering,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 5099–5108.
  29. R. Hu, A. Rohrbach, T. Darrell, and K. Saenko, “Language-conditioned graph networks for relational reasoning,” in *Int. Conf. Comput. Vis.*, 2019, pp. 10 294–10 303.
  30. G. A. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.



31. S. Ontanon, J. Ainslie, Z. Fisher, and V. Cvick, “Making transformers solve compositional tasks,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 3591–3607.
32. H. Zheng and M. Lapata, “Disentangled sequence to sequence learning for compositional generalization,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 4256–4268.
33. N. Saphra and A. Lopez, “Lstms compose—and learn—bottom-up,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 2797–2809.
34. V. Dankers, A. Langedijk, K. McCurdy, A. Williams, and D. Hupkes, “Generalising to german plural noun classes, from the perspective of a recurrent neural network,” in *CoNLL*, 2021, pp. 94–108.
35. Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee, “Gligen: Open-set grounded text-to-image generation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 22 511–22 521.
36. A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa *et al.*, “Openflamingo: An open-source framework for training large autoregressive vision-language models,” *arXiv preprint arXiv:2308.01390*, 2023.
37. J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” *arXiv preprint arXiv:2301.12597*, 2023.
38. J. Cho, J. Lei, H. Tan, and M. Bansal, “Unifying vision-and-language tasks via text generation,” in *Int. Conf. Mach. Learn.*, 2021, pp. 1931–1942.
39. M. Shah, X. Chen, M. Rohrbach, and D. Parikh, “Cycle-consistency for robust visual question answering. in 2019 ieee,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 6642–6651.
40. A. Ray, K. Sikka, A. Divakaran, S. Lee, and G. Burachas, “Sunny and dark outside?! improving answer consistency in vqa through entailed question generation,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019, pp. 5860–5865.
41. M. T. Ribeiro, C. Guestrin, and S. Singh, “Are red roses red? evaluating consistency of question-answering models,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6174–6184.
42. Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, “Making the v in vqa matter: Elevating the role of image understanding in visual question answering,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 6904–6913.
43. S. Tascon-Morales, P. Márquez-Neila, and R. Sznitman, “Logical implications for visual question answering consistency,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 6725–6735.
44. R. R. Selvaraju, P. Tendulkar, D. Parikh, E. Horvitz, M. T. Ribeiro, B. Nushi, and E. Kamar, “Squinting at vqa models: Introspecting vqa models with sub-questions,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 10 003–10 011.
45. Y. Yuan, S. Wang, M. Jiang, and T. Y. Chen, “Perception matters: Detecting perception failures of vqa models using metamorphic testing,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 16 908–16 917.
46. W. Kim, B. Son, and I. Kim, “Vilt: Vision-and-language transformer without convolution or region supervision,” in *Int. Conf. Mach. Learn.*, 2021, pp. 5583–5594.
47. J.-H. Kim, J. Jun, and B.-T. Zhang, “Bilinear attention networks,” *Adv. Neural Inform. Process. Syst.*, vol. 31, 2018.

48. Y. Kant, A. Moudgil, D. Batra, D. Parikh, and H. Agrawal, "Contrast and classify: Training robust vqa models," in *Int. Conf. Comput. Vis.*, 2021, pp. 1604–1613.
49. L. Li, Z. Gan, and J. Liu, "A closer look at the robustness of vision-and-language pre-trained models," *arXiv preprint arXiv:2012.08673*, 2020.
50. D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.
51. C. Li, Z. Li, C. Jing, Y. Jia, and Y. Wu, "Exploring the effect of primitives for compositional generalization in vision-and-language," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 19 092–19 101.
52. L. Yang, Q. Kong, H.-K. Yang, W. Kehl, Y. Sato, and N. Kobori, "Deco: Decomposition and reconstruction for compositional temporal grounding via coarse-to-fine contrastive ranking," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 23 130–23 140.
53. J. Li, S. Tang, L. Zhu, W. Zhang, Y. Yang, T.-S. Chua, and F. Wu, "Variational cross-graph reasoning and adaptive structured semantics learning for compositional temporal grounding," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.
54. C. Li, H. Xu, J. Tian, W. Wang, M. Yan, B. Bi, J. Ye, H. Chen, G. Xu, Z. Cao *et al.*, "mplug: Effective and efficient vision-language learning by cross-modal skip-connections," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 7241–7259.
55. W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, and F. Wei, "Image as a foreign language: BEiT pretraining for vision and vision-language tasks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.
56. X. Dong, P. Zhang, Y. Zang, Y. Cao, B. Wang, L. Ouyang, X. Wei, S. Zhang, H. Duan, M. Cao, W. Zhang, Y. Li, H. Yan, Y. Gao, X. Zhang, W. Li, J. Li, K. Chen, C. He, X. Zhang, Y. Qiao, D. Lin, and J. Wang, "Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model," *arXiv preprint arXiv:2401.16420*, 2024.
57. J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A frontier large vision-language model with versatile abilities," *arXiv preprint arXiv:2308.12966*, 2023.
58. W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song *et al.*, "Cogvlm: Visual expert for pretrained language models," *arXiv preprint arXiv:2311.03079*, 2023.
59. Q. Ye, H. Xu, J. Ye, M. Yan, A. Hu, H. Liu, Q. Qian, J. Zhang, and F. Huang, "mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024, pp. 13 040–13 051.
60. X. Contributors, "Xtuner: A toolkit for efficiently fine-tuning llm," <https://github.com/InternLM/xtuner>, 2023.
61. D. Ltd., "mmalaya," <https://github.com/DataCanvasIO/MMAIaya>, 2024.
62. O. Contributors, "Opencompass: A universal evaluation platform for foundation models," <https://github.com/open-compass/opencompass>, 2023.
63. A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, "Don't just assume; look and answer: Overcoming priors for visual question answering," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4971–4980.
64. G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.