This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Perception Matters: Detecting Perception Failures of VQA Models Using Metamorphic Testing

Yuanyuan Yuan[1] Shuai Wang[1*] Mingyue Jiang[2] Tsong Yueh Chen[3]
[1]HKUST, [2]Zhejiang Sci-Tech University, [3]Swinburne University of Technology
{yyuanaq, shuaiw}@cse.ust.hk, mjiang@stu.edu.cn, tychen@swin.edu.au

Abstract

Visual question answering (VQA) takes an image and a natural-language question as input and returns a natural-language answer. To date, VQA models are primarily assessed by their accuracy on high-level reasoning questions. Nevertheless, Given that perception tasks (e.g., recognizing objects) are the building blocks in the compositional process required by high-level reasoning, there is a demanding need to gain insights into how much of a problem low-level perception is. Inspired by the principles of software metamorphic testing, we introduce `MetaVQA`*, a model-agnostic framework for benchmarking perception capability of VQA models. Given an image i,* `MetaVQA` *is able to synthesize a low-level perception question q. It then jointly transforms (i, q) to one or a set of sub-questions and sub-images.* `MetaVQA` *checks whether the answer to (i, q) satisfies metamorphic relationships (MRs), denoting perception consistency, with the composed answers of transformed questions and images. Violating MRs denotes a failure of answering perception questions.* `MetaVQA` *successfully detects over 4.9 million perception failures made by popular VQA models with metamorphic testing. The state-of-the-art VQA models (e.g., the champion of VQA 2020 Challenge) suffer from perception consistency problems. In contrast, the Oscar VQA models, by using anchor points to align questions and images, show generally better consistency in perception tasks. We hope* `MetaVQA` *will revitalize interest in enhancing the low-level perceptual abilities of VQA models, a cornerstone of high-level reasoning.*

1. Introduction

Deep learning techniques have been applied to a variety of question-answering tasks. In particular, visual question answering (VQA) models take an image and a natural-language question as input and return a natural-language answer as output [6]. At present, the standard paradigm is to use hold-out validation, based on a train–validation–test dataset split, to estimate the accuracy of VQA models. Recent works have also shown that high-level logic reasoning consistency might not be preserved by VQA models and provide enhancements accordingly [31, 33, 14].

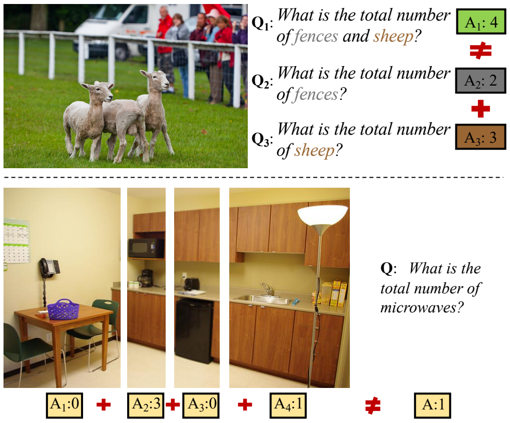

Q1: What is the total number of fences and sheep?
A1: 4
≠
Q2: What is the total number of fences?
A2: 2
+
Q3: What is the total number of sheep?
A3: 3
Q: What is the total number of microwaves?
A1:0
+
A2:3
+
A3:0
+
A4:1
≠
A:1

Figure 1: Perception task failures detected by `MetaVQA`.

Human cognition is believed to be a compositional process [33]: high-level reasoning and understanding requires to first perform multiple perception tasks. For instance, to answer the question "is the microwave occupied?" (the second image in Fig. 1), a VQA model should first detect the microwave, extract its associated properties, identify other objects (e.g., human beings) in the context, understand the question, and try to reason about if the microwave is being used by someone. In other words, perception tasks (e.g., identifying the microwave) serve as the cornerstone for VQA models to answer high-level reasoning questions. Despite the overall optimistic views on VQA models' accuracy of performing perception tasks, Fig. 1 presents two failed perception tasks of a popular VQA model. While it is generally hard to check if the answer A_1 to Q_1 is incorrect (unless involving human efforts), the sum of A_2 and A_3 is inconsistent with A_1, indicating that the VQA model obviously failed to answer at least one perception question. Similarly for the second case, the sum of A_{1-4} is not equal

[*] Corresonding Author

to A, indicating that objects are not correctly recognized in the original image or at least in one cut. In contrast to the overall optimistic view on VQA models' perception ability and recent thrusts on checking high-level reasoning correctness [31, 33, 14], preliminary observations shed light on our key incentive of this research:

> "Is answering perception questions, as the keystone of high-level reasoning, really a solved task for VQA models?"

Inspired by principles of software metamorphic testing (MT) [10], this research proposes MetaVQA as a model-agnostic approach to testing VQA models on its understanding of perception tasks. In particular, given an image i, MetaVQA synthesizes a question q focusing on the objects and properties detected in i by object detectors. MetaVQA then performs a set of transformations on q and i to generate transformed $q' \in \mathcal{Q}$ and $i' \in \mathcal{I}$. MetaVQA checks whether the answers proposed by a VQA model to $\mathcal{Q}$ and $\mathcal{I}$ satisfy metamorphic relations (MRs), denoting perception consistency, with those produced by the same model to q and i. Violating MRs denotes a failure of answering perception questions for VQA models.

This work aims to study VQA models in a realistic setting and to more clearly delineate the perception ability of VQA models. MetaVQA is effective and shows promising results when evaluating popular VQA models based on different model architectures. Our approach detects over 4.9 million erroneous answers produced by VQA models that have been extensively trained and have shown high leaderboard performance [2]. In particular, MetaVQA reveals surprising results that the state-of-the-art VQA model, GridFeat+MoVie which won the 1st place of the VQA 2020 challenge, has higher error rates in perception tasks than its competitors. We provide a detailed investigation of errors detected by MetaVQA. MetaVQA could facilitate model debugging and serve as assessment criteria in addition to standard leaderboard benchmark. In summary, this work makes the following main contributions:

- At the conceptual level, we advocate assessing VQA models' perception ability, the trust base of high-level logic reasoning. Our model-agnostic approach effectively tests VQA models without any knowledge of their internal structure and without a requirement for manually labeled answers.
- At the technical level, we design MetaVQA as a metamorphic testing framework that implements a set of question- and image-oriented metamorphic relations (MRs). Each MR asserts one or several perception abilities of the VQA models.
- At the empirical level, we use MetaVQA to test ten popular VQA models, including recent years' VQA challenge champions. MetaVQA successfully exposes millions of erroneous answers to perception questions. VQA models of different architectures manifest distinct accuracy in answering perception questions. We give further discussions and studies to explore potential enhancement of perception tasks.

We have released MetaVQA for results verification and benchmarking VQA models [1].

2. Metamorphic Testing (MT)

Determining the correctness of answers produced by VQA models for arbitrary question-and-image pairs is tedious and requires considerable manual effort. Inspired by the principles of MT and its major success in automatically detecting bugs and assessing quality of software, AI models, and Big Data sectors [9, 32, 10, 41, 43, 44], we use MT to assess VQA model accuracy. The overall strength of MT lies in its ability to assert model correctness via **metamorphic relations** (**MRs**), without the need to specify the ground truth. Each MR denotes a necessary and usually invariant property of the model. For instance, to test the implementation of $sin(x)$, instead of knowing the expected output of arbitrary floating-point input x (which is rarely possible), we can assert that the MR $sin(x) = sin(\pi - x)$ always holds when transforming x. A bug in $sin(x)$ is detected when the outputs of $sin(x)$ and $sin(\pi - x)$ differ. A properly defined MR obviates the need for manually labeling answers, thus making VQA model assessment much easier and more flexible without any manual effort.

MetaVQA implements a set of MRs to transform inputs, including both questions and images, and assert whether the VQA answers to the transformed inputs exhibit *perception consistency* with answers to the original inputs. Our evaluation shows that the VQA models are prone to generating perception failures, indicating the effectiveness of MT.

3. Related Work

VQA Challenge. A number of works have constructed datasets for the VQA task over the last several years, among which the most famous is the VQA dataset [7] and its follow-up VQA 2.0 dataset [15]. Four VQA challenges have been launched based on the VQA 2.0 dataset. Table 1 reviews popular VQA models, and we also present a model structure hierarchy in Fig. 2. This research evaluates VQA challenge champions of recent three years (see Sec. 6). In addition, we also evaluate recently released BERT-like and Multi-Task VQA models (e.g., Oscar [22]). Despite their highly impressive leaderboard performance [2], MetaVQA successfully detects millions of erroneous answers to perception questions.

Benchmarking Computer Vision and Language Models. Various techniques for testing conventional software

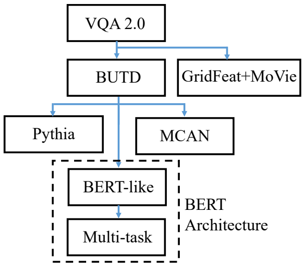


Figure 2: Model hierarchy.

Model (Family) Name	Backbone	Image Feature	VQA Challenge
BUTD [5]	Faster R-CNN	Region Feature	2017 champion
Pythia [18]	Faster R-CNN	Region Feature	2018 champion
MCAN [42]	MCAN	Region Feature	2019 champion
BERT-like [24, 35, 21, 36]	BERT	Region Feature	
Multi-task [25, 22, 11]	BERT	Region Feature	
GridFeat+MoVie [27, 17]	MCAN	Grid Feature	2020 champion

Table 1: VQA model structures.

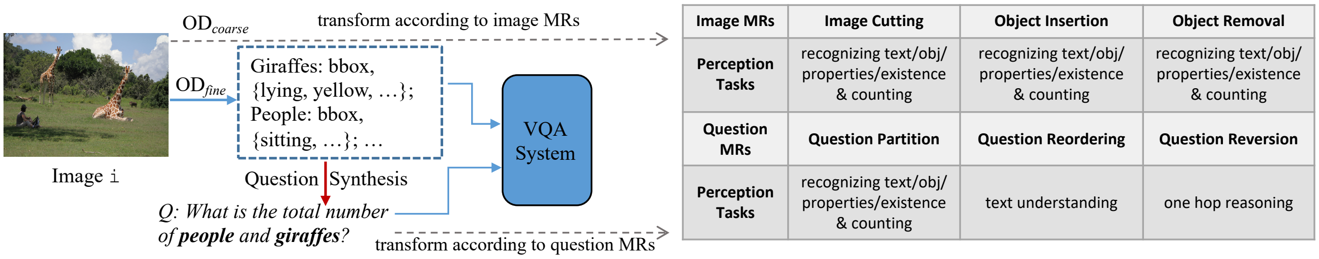


Image MRs	Image Cutting	Object Insertion	Object Removal
Perception Tasks	recognizing text/obj/ properties/existence & counting	recognizing text/obj/ properties/existence & counting	recognizing text/obj/ properties/existence & counting
Question MRs	Question Partition	Question Reordering	Question Reversion
Perception Tasks	recognizing text/obj/ properties/existence & counting	text understanding	one hop reasoning

(a) VQA Models and Perception Consistent Transformations

(b) MRs and Checked Perception Tasks

Figure 3: High-level workflow and how MRs can expose perception failures.

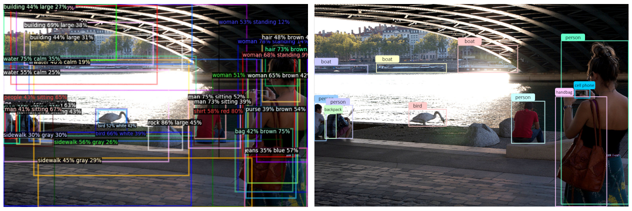
have been recently applied to deep learning-based image-analysis models [29, 39, 12, 40] and NLP models [13, 38, 26, 16]. In contrast to previous works, this research focuses on assessing VQA models, which typically develop a *joint* understanding of free-form, open-ended, natural-language questions and images that can be used to generate a comprehensive and flexible output [6].

Recent research works have also studied the robustness of VQA models, particularly on its high-level reasoning capability [31, 33, 14, 19, 8]. Some existing works leverage heavy-weight gradient-based or GAN-based methods to synthesize counter-factual examples [34, 4, 3, 37, 30]. They could also require manual efforts [31, 33] or expensive image manipulation models (e.g., inpainting) that are not scalable [14]. MetaVQA proposes a comprehensive set of practical and model-agnostic metamorphic testing schemes that particularly focus on checking low-level perception ability. It treats each VQA model as a "black box" and mutates image/question inputs without requiring any labeled answers.

4. Approach

A VQA model V is typically trained on a dataset of (i, q, a) triplets where i denotes an image, q denotes a question to i, and a denotes the correct answer. During prediction, let a target image be i and a raised question be q, V will yield the corresponding answer $a = V(q, i)$. Conventional methods use train–validation–test dataset splits to estimate the accuracy of VQA models, which requires a labeled answer a to represent ground truth. As shown in Fig. 3(a), MetaVQA instead conducts perception-consistent transformations on questions and images, and therefore, the launched testing is fully automated without any manually defined ground truth or labels.

To systematically explore the perception ability of VQA models, this work designs MRs toward both natural language questions and images. To this end, MetaVQA assesses all basic VQA perception abilities listed in [33] except spatial relationships which seem arguable to be a "perception task." Fig. 3(b) illustrates how each MR asserts one or several perception abilities of VQA models. The rest of this section introduces question-oriented MRs (Sec. 4.1) and image-oriented MRs (Sec. 4.2), respectively.

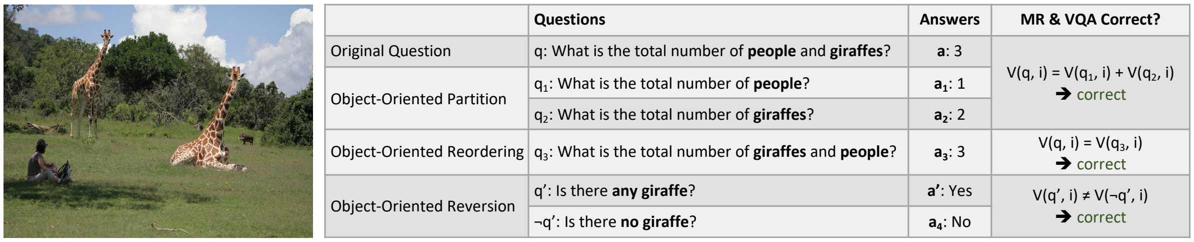


(a) Fine-Grained Objects and Attributes Detection (b) Coarse-Grained Objects Detection

Figure 4: Objects and properties detection.

4.1. Question-Oriented MRs

Objects/Properties Detection. Fig. 4 depicts the pre-process of question transformation, in which we use object detection to recognize and localize various kinds of object instances from an image i. We first use a fine-grained object detector OD_{fine} to extract objects and their associated properties (see Fig. 4(a)). OD_{fine}, derived from



	Questions	Answers	MR & VQA Correct?
Original Question	q: What is the total number of **people** and **giraffes**?	**a**: 3	V(q, i) = V(q$_1$, i) + V(q$_2$, i) ➔ correct
Object-Oriented Partition	q$_1$: What is the total number of **people**?	**a$_1$**: 1	
	q$_2$: What is the total number of **giraffes**?	**a$_2$**: 2	
Object-Oriented Reordering	q$_3$: What is the total number of **giraffes** and **people**?	**a$_3$**: 3	V(q, i) = V(q$_3$, i) ➔ correct
Object-Oriented Reversion	q': Is there **any giraffe**?	**a'**: Yes	V(q', i) ≠ V(¬q', i) ➔ correct
	¬q': Is there **no giraffe**?	**a$_4$**: No	

(a) Image `i`

(b) MRs and Corresponding Transformation Strategies

Figure 5: Natural language question-oriented MRs and transformations.

BUTD [5], is trained on densely annotated images from the Visual Genome dataset [20], in which each image is annotated with objects, properties and relations. OD$_{fine}$ can thus detect objects of thousands of classes with properties of hundreds of classes. For instance, in Fig. 4(a), OD$_{fine}$ can extract each person and his associated properties (e.g., clothes, gesture). These object- and property-related features are used to synthesize questions based on different question-oriented MRs.

In addition to question transformation, image transformation (e.g., cutting) is also built on the basis of object bounding box localization (Sec. 4.2). However, since OD$_{fine}$ may "segment" the image into large components (e.g., "sidewalk" in Fig. 4(a)) which overlap with others, we instead use a coarse-grained object detector OD$_{coarse}$ to localize object instances in i (see Fig. 4(b)). OD$_{coarse}$ detects relatively small number of objects. Bounding boxes, which are usually not overlapped, will be used to guide images transformation (Sec. 4.2).

4.1.1 Object-/Property-Oriented Partitioning

Given the object- and property-recognition results for an image i, we first synthesize number-counting questions to assert whether VQA models can give *perceptually consistent answers*. To do so, we design a number-counting question q to be "partitionable", in the sense that it counts the total number of multiple object instances or objects with multiple properties. We then conduct object-/property-oriented partition, by dividing q into sub-questions $q' \in Q_{par}$, where each q' counts the number of individual objects or objects with particular properties. The corresponding MR is $V(q,i) = \sum_{q' \in Q_{par}} V(q',i)$, meaning that the composed answers to $q' \in Q_{par}$ and i must be equivalent to the original answer. A perception error in VQA, which presumably indicates failures of text/object/property recognition or counting (see "Question Partition" in Fig. 3(b)), is detected when this MR is violated.

While the object- and property-oriented transformation can incorporate an arbitrary number of objects/properties when synthesizing the question q, `MetaVQA` is implemented to only use *two* objects or properties when synthesizing q (therefore $|Q_{par}| = 2$). VQA models already show low accuracy in preserving the perception consistency for this setting. We now introduce two strategies to concretize the above MR by focusing on objects or properties.

Object-Oriented Partitioning. We first propose object-oriented partitioning, which decomposes a number-counting question q of two objects into sub-questions q_1, q_2 that depend on each individual object O_1 and O_2, respectively. Hence, to synthesize q, we require image i to have at least two objects. Consider Fig. 5, where we first synthesize q to count the total number of people and giraffes in i. We then generate two sub-questions q_1 and q_2 to count the number of people and giraffes, respectively, in i. The VQA model V passes our test if the MR, $V(q,i) = V(q_1,i) + V(q_2,i)$, is satisfied. When synthesizing q, we may randomly use O_1 or O_2 that do not appear in i. While counting the number of such non-existent objects should equal to zero and does not interfere with the MR, we find certain cases where the MR is violated, indicating that VQA models somehow "see" such non-existent objects in i and yield an erroneous answer.

Property-Oriented Partitioning. We further test VQA models in differentiating instances of the same object but with different properties. To do so, we synthesize and then decompose a question q counting an object of two properties into sub-questions counting individual properties. Considering the following question to the image i in Fig. 5(a):

> "How many standing giraffes and lying giraffes are in the image?"

where we synthesize q to count the number of giraffes with different gestures in i. We then generate two sub-questions $q_{standing}$ and q_{lying} as follows:

> "How many standing giraffes are in the image?"
> "How many lying giraffes are in the image?"

The VQA model V passes our test if the MR, $V(q,i) = V(q_{standing},i) + V(q_{lying},i)$, is not violated.

Recall fine-grained object detector OD$_{fine}$ can identify objects and their associated properties. We construct each q by randomly picking an object and two of its associated properties. Similar to object-oriented partitioning, we would pick certain properties that do not appear in i. In particular, we divide all found objects in i into two categories:

human objects (e.g., man, woman) and non-human objects. For human objects, we randomly pick a pair of human actions (e.g., “walking” and “reading”) from an action pool P_{human} for splitting. For non-human objects, we randomly pick a pair of colors from a color pool P_{color}. See Supp. Material for the definition of P_{human} and P_{color}.

Discussions. VQA models may express quantity for large amounts with quantifiers like *many* or *a lot*. The “sum” of such quantifiers are defined as consistent with those quantifier themselves (e.g., *many* = *many* + *a lot*). We also want to emphasize that MRs based on question partitions are *not* used to check whether VQA can accurately count numbers. Instead, by partitioning object-/property-oriented questions, it reveals whether the model can really differentiate low-level perception targets (e.g., objects, properties) by checking the consistency of composed answers with the original answers. Indeed, our evaluation will show that the champion of the 2020 VQA challenge, GridFeat+MoVie [27, 17], which has notably enhanced the number counting accuracy with MoVie [27], still shows very high error rates regarding our partitioning MR.

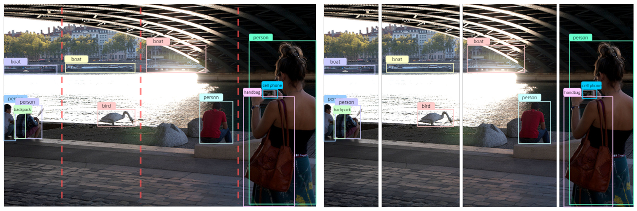
4.1.2 Object- and Property-Oriented Reordering

Consistent with the question-partitioning scheme introduced in Sec. 4.1.1, the reordering transformation scheme also concerns number-counting questions. We therefore reuse the multiple object/property question q synthesized in Sec. 4.1.1. However, in contrast to partitioning question q, we create a set of transformed questions $q \in Q_{reorder}$ by reordering the objects or properties in q and check whether the answers are consistent. Reordering any two pairs of n objects or properties in q leads to $\binom{n}{2}$ permutations; as in Sec. 4.1.1, we reduce the difficulty by using only two objects or properties in generating q. Hence, a given q will have only one transformed q' produced by reordering two objects or properties, from which we check whether MR $V(q,i) = V(q',i)$ holds. Violating this MR, similar to question partitioning (Sec. 4.1.1), indicates failures of object/property recognition or counting.

Considering the sample transformation in Fig. 5(b), where we produce q_3 by reordering the two objects (“giraffes” and “people”) in q. We can then check the consistency of the answers to q_3 and q. As mentioned in Sec. 4.1.1, we may pick non-existent objects/properties to form q. Reordering non-existent objects/properties should also preserve perception consistency.

4.1.3 Object- or Property-Oriented Reversion

The question-partitioning and reordering transformations assert the perception consistency related to number counting. We further propose a question-reversion transformation, which constructs a pair of questions $q, \neg q$ to check for the existence and non-existence of certain objects and properties in i. In this case, the answers to the two questions must be contradictory. Considering the sample questions in Fig. 5(b), which check for the existence and nonexistence of any giraffe in the image with q' and $\neg q'$, respectively. A VQA model passes the test if it produces “yes” as an answer to one of these two questions and “no” to the other. That is, we check whether MR $V(q',i) \neq V(\neg q',i)$ is satisfied. This scheme primarily checks the correctness of simple one hop reasoning, which is also considered into “perception task” by previous work [33].



(a) Original image

(b) Project bounding box to x-axis and cut according to median

Figure 6: Image cutting according to identified median lines parallel to bounding boxes.

4.2. Image-Oriented MRs

This section further proposes three MRs for image transformation. In particular, we propose MRs to (1) cut i into $i_{cut} \in I$, (2) insert new objects into i, and (3) remove existing objects from i. From a holistic perspective, these three schemes assert the perception ability of object and property recognition by stressing VQA models from different aspects. Modern VQA models typically aim to achieve a joint understanding of questions and images, and our approach transforms both dimensions, thus systematically exposing potential perception errors in VQA models. Since images/objects are manipulated using OD$_{coarse}$, at this step we limit objects used in synthesizing questions (Sec. 4.1) to objects detected by OD$_{coarse}$.

Table 2: Extended MRs for image cuts.

Question from MRs	**Extended MRs when images are cut**
Partition & Reorder	$V(q,i) = \sum_{i_{cut} \in I} V(q, i_{cut})$
Reversion	$V(q,i) = V(q,i_1) \vee V(q,i_2) \vee \ldots$
Reversion	$V(\neg q,i) = V(\neg q,i_1) \wedge V(\neg q,i_2) \wedge \ldots$

4.2.1 Image Cutting

As previously mentioned, we use coarse-grained object detector OD$_{coarse}$ to localize objects in i to support image cutting. At this step, we cut i into the maximum number of cuts wherein each cut has at least one recognized object and no bounding box is split into pieces. Fig. 6 illustrates our approach, in which we project the bounding box ranges to the x-axis and cut according to the median of the uncovered ranges on the x-axis.

At this step we reuse questions synthesized from partition & reordering (Sec. 4.1.1) and reversion (Sec. 4.1.3). Once the image i is cut into a set $\mathcal{I}$, we aggregate the answers generated for each cut $i_{cut} \in \mathcal{I}$, and check whether the composition of answers matches the answer to the original i and q. Table 2 lists the corresponding MRs. Answers to number-counting questions on image cuts are summed to check for consistency with the answers provided to the original question and image. For question reversion (Sec. 4.1.3), which synthesizes and transforms "yes/no" question q, we join the answers for each i_{cut} using logical or ($\vee$) or logic and ($\wedge$), respectively.

Table 3: Extended MRs for object insertion/removal.

Question from MRs	**Extended MRs when objects are inserted/removed**
Partition & Reorder	$V(q, i) = V(q, i_{remove/insert})$
Reversion	$V(q, i) = V(q, i_{remove/insert})$
Reversion	$V(\neg q, i) = V(\neg q, i_{remove/insert})$
Partition & Reorder (**Removal**$^{+}$)	$V(q, i) = V(q, i_{remove}) + 1$

4.2.2 Image-Object Removal and Insertion

We further manipulate images by randomly removing or inserting an object. We only remove an object in i in case its bounding box has no overlapping with other objects. To do so, we place a white patch over its bounding box. To insert a new object in i, we rely on the bounding boxes of the existing objects in i to prevent bounding box overlapping, which may cause image-comprehension challenges even for human eyes. An object instance is inserted only if instances of that object already exist in i. This would prevent unrealistic insertion, e.g., inserting an elephant in an indoor scene photo. To reasonably control the complexity, we only remove or insert one object instance to generate a mutated image i' each time. The additional objects can be prepared by using instance-segmentation methods or reusing masks over object instances annotated for image datasets.

When the inserted or removed objects do *not* appear in the questions posed by MetaVQA, we use the first three MRs listed in Table 3 to check perception consistency. However, our tentative study shows that certain VQA models might exploit biases by answering "0" to most number-counting questions (see Sec. 6). Considering merely transforming q does not expose this subtle issue (since $0 = 0{+}0$), we further design the **Removal**$^{+}$ scheme, by removing an object instance which is referred in the question q from the image. Accordingly, we extend the MRs by adding one (see the last MR of Table 3). For example, removing the human being from the image i of Fig. 5 generates image i'. We then check MR $V(q, i) = V(q, i') + 1$ where q is already defined in Fig. 5. This would detect VQA models that exploit dataset bias by always answering "0", since $0 \neq 0{+}1$.

5. Evaluation Setup

The evaluated VQA models are listed in Table 4. MetaVQA treats each VQA model as a "black box." Our testing schemes are orthogonal to particular model structures, and are thus generally applicable for benchmarking VQA models of different architectures.

Most VQA models are evaluated on the test split of the VQA 2.0 dataset [15] and are ranked by their hold-out accuracy on the VQA Challenge leaderboard [2]. To get the best performance, these models are typically trained with both train and validation splits of VQA 2.0 (referred as "Model^{+}" in Table 4). Some authors also released models trained only using the train split; we also evaluated such models. We only test the models released by the authors without any fine-tuning or retraining.

As aforementioned, OD$_{fine}$ is implemented using BUTD [5] whose training data are densely annotated images from the Visual Genome dataset [20]. Note that real-scene images in VQA 2.0, which are from the COCO dataset [23], have already been annotated with bounding boxes. Hence, to implement OD$_{coarse}$ for image cutting (Sec. 4.2.1), we directly reuse those bound boxes. Similarly, images from COCO are also annotated with masks over each object instance; we reuse those masks to extract object instances and support object insertion/removal (Sec. 4.2.2).

We construct test inputs from the *validation split* of VQA 2.0. Hence, testing models trained with only training split represents an "in-the-wild" setting, where non-training data are used by third-party analysts or even adversaries to assess the released VQA models. In contrast, testing models whose training data already subsumes the validation split checks if the models have really captured the low-level perceptions in the training data. As will be shown in Sec. 6, VQA models can still make considerable errors in answering perception questions derived from its training data.

6. Evaluation

Table 4 reports the testing results. MetaVQA successfully exposes over 4.9 million erroneous answers to perception questions. To ease the presentation, for each MR, we highlight VQA models with the top-five highest error rates.

6.1. Cross-Model Comparison

Sec. 3 has mentioned that the tested models in Table 4, including three recent champions of the VQA challenge, have distinct model architectures. However, Table 4 shows that VQA perception errors are a *general concern* regardless of the underlying design. MetaVQA identifies considerable numbers of errors in all of these models. While such low-level perception tasks may be considered "solved" based on standard hold-out accuracy results, MetaVQA highlights that basic perception skills (e.g., counting and recognition)

Table 4: Error rates overview. Highest top-five error rates for each MR are marked. The total number of test inputs is also given, e.g., for **Partition** we synthesize 142,213 questions and use them for partitioning and model testing. For the question transformations (2nd–4th columns), we do not change images. For image transformations (5th–8th columns), we report the total results by transforming images and questions with different schemes.

VQA Models	**Partition**	**Reversion**	**Reordering**	**Cutting**	**Insetion**	**Removal**	**Removal**$^+$
	142,213	53,221	142,213	315,796	836,574	806,872	105,847
GridFeat+MoViE$^+$	94.97%	68.77%	24.13%	36.99%	11.76%	10.83%	88.12%
Oscar$^+_{\text{large}}$	84.91%	34.94%	23.66%	35.15%	5.74%	3.91%	91.63%
Oscar$_{\text{base}}$	85.13%	36.71%	34.35%	45.44%	6.59%	4.49%	89.86%
MCAN$^+_{\text{large}}$	92.25%	47.94%	37.21%	45.78%	13.92%	6.30%	90.90%
MCAN$^+_{\text{small}}$	94.48%	51.07%	31.72%	48.16%	11.89%	5.58%	90.14%
VisualBERT$^+$	35.17%	82.44%	18.02%	6.44%	1.19%	0.52%	99.68%
VisualBERT	30.34%	98.61%	10.20%	4.50%	1.02%	0.30%	99.56%
ViLBERT	23.93%	98.70%	8.78%	1.82%	0.61%	0.20%	99.74%
Pythia$^+$	97.92%	73.15%	45.77%	74.33%	9.17%	7.40%	97.31%
Pythia	92.04%	71.25%	37.83%	62.37%	10.16%	8.87%	98.26%
Total	1,039,811	353,155	386,366	1,139,966	606,235	390,489	1,000,456

demands principled improvement.

The 2020 VQA champion, GridFeat+MoViE$^+$, exhibits high error rates regarding the the question partitioning and object insertion/removal MRs. These results illustrate the benefits of `MetaVQA` in addition to standard leaderboard evaluation. In contrast, BERT-like models seemingly manifest low error rates for most MRs. However, BERT-like models indeed yields *zero* for most number-counting related perception questions. On the other hand, **Removal**$^+$ (see Sec. 4.2.2) demystifies the true capability of BERT-like models, by revealing a very high error rate (over 99%). See Sec. 6.2 for further discussion on this matter.

Oscar models generally outperform others with low error rates, e.g., Oscar$^+_{\text{large}}$ has no "top-five" error rate cases. Overall, Oscar uses object tags detected in images as anchor points to align with the paired question. Holistically, this alignment promotes a joint comprehension of the question and the image. Consistent with [22], such alignment and and its promoted joint understanding should generally improve the perception ability of VQA models.

As mentioned in Sec. 5, some models, referred to as "Model$^+$", may have been trained using both training and validation splits of VQA 2.0. While it has been generally illustrated that using both training and validation split can enhance the hold-out accuracy, Table 4 shows that using more training data might not necessarily reduce the error rates, e.g., comparing Pythia$^+$ with Pythia. Overall, it might not be inaccurate to interpret that enhancing the perception ability requires more principled model design or training set enhancement rather than simply adding more training data.

6.2. Cross-Method Comparison

The performance of the models against different MRs fluctuates, indicating the challenge of answering different perception questions. Particularly, the models perform poorly on partitioning (with less than 15% of answers passing our tests). We give further study relevant to this matter in Sec. 6.3. Similarly, question reversion, by asserting answers to q and $\neg q$ must be contradictory, also induces relatively high error rate. This indicates that VQA models might not properly manifest consistency for even "one hop reasoning" (cf. Fig. 3(b)). Object-/property-oriented reordering shows low error rates. This indicates that text understanding is relatively easy. Nevertheless, we still observe several cases (see Supp. Material) where the models mostly answering "yes" (or "no") to "yes/no" questions.

All three image transformation methods expose considerable numbers of perception errors, indicating that object/property recognition, as a basic perception ability, is not "solved" yet. Fig. 7 presents cases to illustrate how perception errors are detected via object insertion/removal. We emphasize that image cutting or object insertion schemes do *not* fragment or overlap existing objects in the image, thus preserving the perception consistency (Sec. 4.2.1). Nevertheless, manual study shows that image cutting might place the same objects into different region proposals or grids, thus stressing the perception task of object/property recognition. Similarly, many defects are exposed when objects are inserted *close to* existing objects (the first case in Fig. 7). By inserting object close to existing objects, the region proposal or grids used for object localization could be disturbed, thus causing recognition failures.

Removal$^+$. The **Removal** column in Table 4 shows high accuracy of BERT-like models. However, BERT-like models indeed perform much worse than others regarding the object removal related transformations (as can be revealed from the **Removal**$^+$). Manual study shows that BERT-like models frequently exploit the image biases by answering "0" when counting most object instances regardless whether the instances exist or not. In contrast, **Removal**$^+$, by transforming images (see Sec. 4.2.2), successfully eliminates false positives incurred by the BERT-like models. Also, it is clear that the 2020 VQA champion, GridFeat+MoViE$^+$, shows the lowest error rate in the **Removal**$^+$ column. This result is consistent with its model design: the MoViE model [28] is primarily designed to enhance the visual

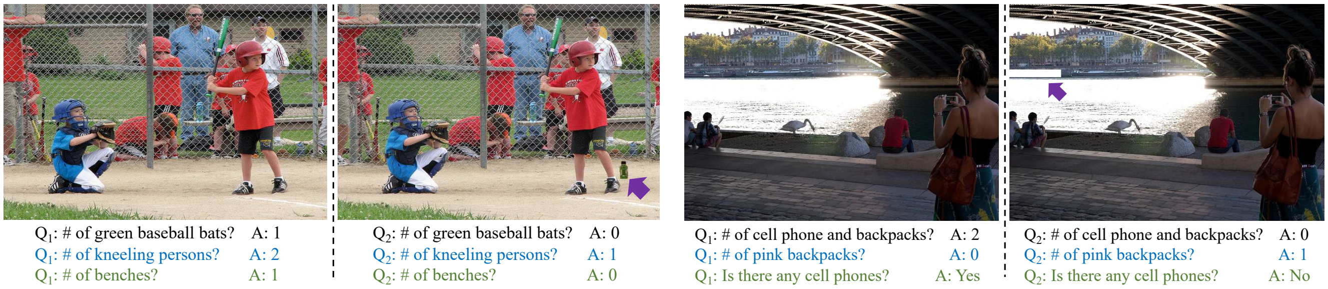

Q1: # of green baseball bats? A: 1
Q1: # of kneeling persons? A: 2
Q1: # of benches? A: 1
Q2: # of green baseball bats? A: 0
Q2: # of kneeling persons? A: 1
Q2: # of benches? A: 0
Q1: # of cell phone and backpacks? A: 2
Q1: # of pink backpacks? A: 0
Q1: Is there any cell phones? A: Yes
Q2: # of cell phone and backpacks? A: 0
Q2: # of pink backpacks? A: 1
Q2: Is there any cell phones? A: No

Figure 7: Case study of perception failures. The inserted/removed objects are pinpointed in the figures. Due to the limited space, we replace phrase "*what is the total number*" with "#".

counting capability. However, the 2020 champion still shows low accuracy for other number counting-related schemes (e.g., **Partitioning**).

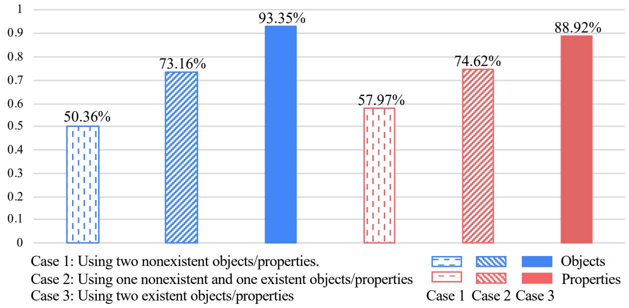


Figure 8: Exploring partition failures.

6.3. Exploring Partition Failures

Table 4 shows that the average error rate of **Partitioning** is 91.7% (excluding BERT models which mostly yield "0"). $\text{Oscar}^{+}_{\text{large}}$ performs slightly better than others. This section explores the partition errors of $\text{Oscar}^{+}_{\text{large}}$.

Recall when synthesizing question q for partitioning, we randomly add objects/properties which do not exist in the image (Sec. 4.1.1). Fig. 8 presents error rates in terms of three cases focusing on objects or properties. Case_3 (for both objects and properties) manifests a much higher error rate compared with the others. This is intuitive: to answer questions in case_1 whose MR is $0 = 0 + 0$, the VQA model only need to decide whether the target objects/properties *exist* in the image. Holistically, in addition to decide existence, case_2 and case_3 require to further localize, differentiate, and align target objects with questions in order to satisfy the MRs, which, as shown in Fig. 8, impose much higher challenge (around 90% error rates). This again indicates that common perception tasks impose different levels of challenges to VQA models. Nevertheless, even for the relatively easier perception tasks of recognizing the existence, Oscar models can still make over half errors.

In sum, we interpret that the relatively lower error rates of Oscar models in Table 4 suggests the importance of enhancing the joint and aligned understanding of questions and images. Nevertheless, `MetaVQA` indicates that common perception tasks, even for recognizing the existence of objects/properties, still require major improvement. We hope the evaluation results could re-advocate the attention of enhancing basic and common perception ability of VQA models, which forms the keystone of high-level reasoning.

7. Discussion and Future Work

Metamorphic Assessment Criteria. The model-agnostic design of `MetaVQA` helps to benchmark and assess VQA models, by efficiently exposing their error rates in passing certain MRs. This suggests an important usage scenario for `MetaVQA`: to expose the differences and *preferences* of VQA models, thereby enabling users to select the most appropriate VQA model for their particular scenarios. For instance, users particularly concerned with low-level perception questions may select VQA models based on the Oscar frameworks, while GridFeat+MoVie, which has demonstrated highly impressive accuracy in the leaderboard [2], may be more desirable for scenarios requiring frequent high-level logic reasoning.

Usage Scenario of `MetaVQA`. `MetaVQA` can be used to detect VQA errors during the system-testing stage (e.g., during VQA competitions). Moreover, it is also possible to integrate `MetaVQA` into the VQA development (model-training) stage, where developers can use `MetaVQA` as a complement to the standard model validation procedure, to better understand their VQA models in solving perception tasks. We also envision a potential opportunity to augment model-training datasets with error-triggering inputs identified by `MetaVQA`. We leave this as one future research.

Acknowledgments. We thank the CVPR anonymous reviewers and AC for their valuable feedback. Mingyue Jiang is supported by NSFC Grant 61802349 and Tsong Yueh Chen is supported by Australian Research Council (DP210102447).

References

[1] MetaVQA Codebase. https://github.com/MetaVQA/MetaVQA, 2020.

[2] VQA Challenge Leaderboard. https://visualqa.org/roe.html, 2020.

[3] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10044–10054, 2020.

[4] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal VQA: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9690–9698, 2020.

[5] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[7] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

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

[9] Tsong Y Chen, Shing C Cheung, and Shiu Ming Yiu. Metamorphic testing: a new approach for generating next test cases. Technical report, Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong . . . , 1998.

[10] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, TH Tse, and Zhi Quan Zhou. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys (CSUR)*, 51(1):1–27, 2018.

[11] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal image-text representation learning. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020.

[12] Anurag Dwarakanath, Manish Ahuja, Samarth Sikand, Raghotham M. Rao, R. P. Jagadeesh Chandra Bose, Neville Dubash, and Sanjay Podder. Identifying implementation bugs in machine learning based image classifiers using metamorphic testing. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2018, pages 118–128, 2018.

[13] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. Fairness testing: testing software for discrimination. In *ACM ESEC/FSE*, pages 498–510. ACM, 2017.

[14] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. MUTANT: A training paradigm for out-of-distribution generalization in visual question answering. *arXiv preprint arXiv:2009.08566*, 2020.

[15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] Pinjia He, Clara Meister, and Zhendong Su. Structure-invariant testing for machine translation. In *Proceedings of the 42nd International Conference on Software Engineering*, ICSE '20, 2020.

[17] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. *arXiv preprint arXiv:2001.03615*, 2020.

[18] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the VQA challenge 2018. *CoRR*, abs/1807.09956, 2018.

[19] Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–170, 2020.

[20] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[21] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[22] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[24] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[25] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020.

[26] Pingchuan Ma, Shuai Wang, and Jin Liu. Metamorphic testing and certified mitigation of fairness violations in nlp models. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, IJCAI, pages 458–465, 2020.

[27] Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. MoVie: Revisiting modulated convolutions for visual counting and beyond, 2020.

[28] Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. Revisiting modulated convolutions for visual counting and beyond. *arXiv preprint arXiv:2004.11883*, 2020.

[29] Augustus Odena and Ian Goodfellow. Tensorfuzz: Debugging neural networks with coverage-guided fuzzing. *arXiv preprint arXiv:1807.10875*, 2018.

[30] Jingjing Pan, Yash Goyal, and Stefan Lee. Question-conditioned counterfactual image generation for vqa. *arXiv preprint arXiv:1911.06352*, 2019.

[31] Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, 2019.

[32] Sergio Segura, Gordon Fraser, Ana B Sanchez, and Antonio Ruiz-Cortés. A survey on metamorphic testing. *IEEE Transactions on software engineering*, 42(9):805–824, 2016.

[33] Ramprasaath R Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Besmira Nushi, and Ece Kamar. SQuINTing at VQA Models: Introspecting VQA Models With Sub-Questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10003–10011, 2020.

[34] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6649–6658, 2019.

[35] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2019.

[36] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114, 2019.

[37] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. *arXiv preprint arXiv:2004.09034*, 2020.

[38] Sakshi Udeshi, Pryanshu Arora, and Sudipta Chattopadhyay. Automated directed fairness testing. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, ASE 2018, pages 98–108, 2018.

[39] Jingyi Wang, Guoliang Dong, Jun Sun, Xinyu Wang, and Peixin Zhang. Adversarial sample detection for deep neural network through model mutation testing. In *Proceedings of the 41st International Conference on Software Engineering*, ICSE '19, pages 1245–1256, 2019.

[40] Shuai Wang and Zhendong Su. Metamorphic object insertion for testing object detection systems. In *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1053–1065. IEEE, 2020.

[41] Xiaoyuan Xie, Joshua WK Ho, Christian Murphy, Gail Kaiser, Baowen Xu, and Tsong Yueh Chen. Testing and validating machine learning classifiers by metamorphic testing. *Journal of Systems and Software*, 84(4):544–558, 2011.

[42] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6281–6290, 2019.

[43] Zhi Quan Zhou and Liqun Sun. Metamorphic testing of driverless cars. *Communications of the ACM*, 62(3):61–67, 2019.

[44] Zhi Quan Zhou, Liqun Sun, Tsong Yueh Chen, and Dave Towey. Metamorphic relations for enhancing system understanding and use. *IEEE Transactions on Software Engineering*, 2018.