CARETS: A Consistency And Robustness Evaluative Test Suite for VQA

Carlos E. Jimenez **Olga Russakovsky** **Karthik Narasimhan**
Princeton University
{carlosej,olgarus,karthikn}@princeton.edu

Abstract

We introduce CARETS, a systematic test suite to measure consistency and robustness of modern VQA models through a series of six fine-grained capability tests. In contrast to existing VQA test sets, CARETS features balanced question generation to create *pairs* of instances to test models, with each pair focusing on a specific capability such as rephrasing, logical symmetry or image obfuscation. We evaluate six modern VQA systems on CARETS and identify several actionable weaknesses in model comprehension, especially with concepts such as negation, disjunction, or hypernym invariance. Interestingly, even the most sophisticated models are sensitive to aspects such as swapping the order of terms in a conjunction or changing the number of answer choices mentioned in the question. We release CARETS to be used as an extensible tool for evaluating multi-modal model robustness.[1]

1 Introduction

The task of visual question answering integrates the domains of computer vision and NLP by probing models' understanding of images through natural language queries. After the introduction of the Visual Question Answering (VQA) benchmark (Antol et al., 2015), subsequent work identified the presence of several superficial correlations and other weaknesses latent in the VQA question gathering process (Goyal et al., 2017; Agrawal et al., 2018), which lead to potentially optimistic evaluations when considering accuracy alone. More recently developed benchmarks (Hudson and Manning, 2019; Goyal et al., 2017; Agrawal et al., 2018) explicitly avoid these weaknesses by introducing question, answer, and image balancing, or distributional shifts. While these efforts provide more difficult benchmarks, a thorough evaluation of model capabilities requires a deeper and more detailed approach.

To this end, we introduce CARETS – a Consistency And Robustness Evaluative Test Suite for visual question answering. Inspired by recent work in NLP that generates 'unit tests' for models (Ribeiro et al., 2020b), CARETS contains systematically generated VQA tests that evaluate six different capabilities that any VQA model should handle – robustness to question rephrasings, ontological reasoning, symmetrical logic, visual perturbations, question negation, and attribute antonymy. Each test point in CARETS consists of a *pair* of instances which are small but strategic variations of each other either visually or in the question's text. This allows us to conduct fine-grained capability evaluations beyond just measuring high-level accuracy scores.

Across tests, we generate over 190,000 instance pairs in total using a programmatic approach that fills in templates (from nearly 200 templates in total) using ground-truth scene graphs (Krishna et al., 2017) from the GQA (Hudson and Manning, 2019) validation split. We then evaluate six modern VQA models on each test using several metrics: overall accuracy, *self-consistency* and *comprehensive accuracy*. Self-consistency measures models' ability to maintain their answer across question variations, while comprehensive accuracy estimates their ability to answer all instance variants correctly.

Our experiments reveal several interesting findings: (1) most modern VQA systems achieve only middling self-consistency (∼60-80%) which is further not always correlated with their accuracy, (2) all models struggle to comprehend the concept of negation (self-consistency of 18-28% and comprehensive accuracy <17%), and (3) even simple perturbations like changing the order of choices in the question text can induce a substantial drop

[1] Source code, data, and additional resources may be found at https://github.com/princeton-nlp/CARETS

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6392 - 6405
May 22-27, 2022 ©2022 Association for Computational Linguistics

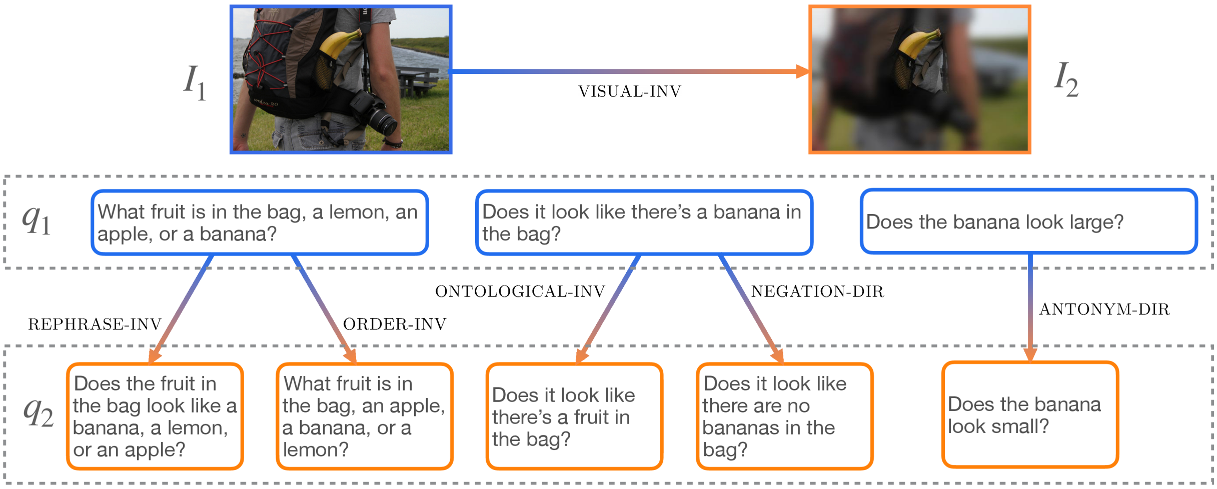

I_1
VISUAL-INV
I_2
q_1
What fruit is in the bag, a lemon, an apple, or a banana?
Does it look like there's a banana in the bag?
Does the banana look large?
REPHRASE-INV
ORDER-INV
ONTOLOGICAL-INV
NEGATION-DIR
ANTONYM-DIR
q_2
Does the fruit in the bag look like a banana, a lemon, or an apple?
What fruit is in the bag, an apple, a banana, or a lemon?
Does it look like there's a fruit in the bag?
Does it look like there are no bananas in the bag?
Does the banana look small?

Figure 1: Our consistency and robustness test suite (CARETS) consists of six tests, corresponding to six identified phenomena that VQA models should be robust to. Tests evaluate models' predictions between pairs of instances (changing $q_1 \to q_2$ or changing $I_1 \to I_2$). REPHRASE-INV for robustness to simple rephrasings; ORDER-INV for robustness to changed argument order in lists, conjunctions, and disjunctions; ONTOLOGICAL-INV for understanding of ontology; VISUAL-INV for robustness to visual context perturbations; NEGATION-DIR for robustness to negative clauses; ANTONYM-DIR for understanding of mutually exclusive attributes.

(10-15%) in performance. Moreover, even state-of-the-art models like LXMERT (Tan and Bansal, 2020) are highly sensitive to the type of questions (binary vs multi-choice) and the number of choices provided. These results reveal several shortcomings in modern VQA systems and hint at potential areas for future improvement. Going beyond our current discoveries, CARETS is an extensible framework and may be easily extended by adding new capability tests for fine-grained evaluation of future models.

2 Related Work

VQA evaluation. The textual, visual, and answer biases discovered in the VQA dataset (Antol et al., 2015) spurred on recent work seeking to improve model evaluation for the task by eliminating these biases (Goyal et al., 2017), introducing distributional shifts (Agrawal et al., 2018), requiring model explanations (Li et al., 2018), thoroughly analyzing biases in datasets and models (Manjunatha et al., 2019), or evaluating on different recognition subtasks (Kafle and Kanan, 2017). While debiased and challenging benchmarks are important, their focus on accuracy as the sole evaluation metric leaves much to be desired (Ribeiro et al., 2020a; Kervadec et al., 2020). In contrast, our testbed provides question or image *pairs* that compares models' predictions *between* questions; measuring their accuracy, consistency, and robustness to a variety of text and image perturbations.

Synthetic Dataset Generation for VQA. One way in which we can generate diverse and balanced datasets is to generate them synthetically, as is done by (Johnson et al., 2015; Zhang et al., 2016; Hudson and Manning, 2019). Synthetically generating questions, images, or both, allows fine control over the distribution of questions, answers, and images. Additionally for our case, synthetic generation allows us to control not just the particular semantics of one question, but also how one question relates to another question in a precisely defined way (e.g. one question is a negation of another) while also remaining relevant and grounded in the image. As both the CLEVR (Johnson et al., 2015) and GQA (Hudson and Manning, 2019) datasets use image scene graphs for question and label generation, they contain questions combining a variety of required capabilities, including compositionality. We feature real-world images with synthetically generated questions as well, but in contrast to GQA, our evaluation has instance *pairs* to systematically test a focused set of capabilities, showing that models may still struggle with simpler, non-compositional questions.

Consistency as Model Comprehension. Some recent work has sought to evaluate models using consistency and other metrics (Hudson and Manning, 2019; Shah et al., 2019; Ribeiro et al., 2020a; Selvaraju et al., 2020; Bitton et al., 2021; Mouselinos et al., 2022). These evaluations often evaluate consistency through question entailment and implication, or simply contrasting examples in the case of (Bitton et al., 2021). The concurrent work (Mouselinos et al., 2022) takes a unique approach, performing discrete visual perturbations while preserving the semantic integrity of a question; their work complements our own as it focuses exclusively on visual perturbations in a synthetic image setting. While we consider these previous methods important for evaluating model comprehension, they often combine question types and capabilities, changing the kind of expected answer, or evaluating consistency on a tree or set of entailed questions. Though ideally models would be consistent and robust for these more complex types of tests, our approach reveals that models can fail even on simple implications.

3 Fine-grained capability tests

Our goal is to provide a testbed for fine-grained evaluation of VQA models' capabilities. To do so, we generate multiple tests, each corresponding to a specific model capability. In contrast to standard VQA benchmarks (Antol et al., 2015; Goyal et al., 2017; Hudson and Manning, 2019), our test sets consist of a *pair* of *original* and *perturbed* instances $\langle(I_1, q_1, a_1), (I_2, q_2, a_2)\rangle$, each with an image, a question and an answer. The two instances within a pair differ from each other in a minimal yet carefully constructed way to hone in on a particular capability, similar to BLiMP (Warstadt et al., 2020). A model is then evaluated on its overall *accuracy*, *self-consistency* (ability to produce consistent, even if wrong, answers to both instances within a pair), and *comprehensive accuracy* (ability to answer consistently and correctly for an instance pair).

Overall, we create six tests corresponding to key capabilities. We provide a high-level description of each test here and describe generation details in Section 4. First, we create four *invariance* tests[2] that use variations of the question phrasing and expect the model to produce the *same* answer to both questions within an instance pair:

1. **Rephrasing invariance** (REPHRASE-INV) measures the model's understanding of minor, meaning-preserving textual changes, e.g.: "What color is the bottle on the shelf, white or blue?" and "*Does the* color of the bottle on the shelf *seem more* white or blue?"
2. **Ontological invariance** (ONTOLOGICAL-INV) measures understanding of ontology, e.g. changing a hyponym in: "Do you see a green jacket?" to a hypernym "Do you see any green *clothes*?"
3. **Order invariance** (ORDER-INV) measures understanding of logically equivalent questions containing different argument orders, e.g.: "Is the black vehicle a van or a truck?" and "Is the black vehicle a *truck* or a *van*?"
4. **Visual obfuscation invariance** (VISUAL-INV) measures the model's answering ability when parts of the image not directly relevant to the visual question are removed. Specifically, we explore blurring, masking and cropping techniques to modify the image.

We also create *directional expectation* tests to measure model behavior on instance pairs where the answer is expected to *change*:

5. **Attribute antonym directional expectation** (ANTONYM-DIR) measures the model's understanding of antonyms, e.g., "Do you think that the wood table is short?" and "Do you think that the wood table is *long*?"
6. **Negation directional expectation** (NEGATION-DIR) measures a model's grasp of negation, e.g., " Are there any apples in this picture?" and "Are there *no* apples in this picture?

4 Dataset generation

Each of the six test datasets start with the generation of 'original', unperturbed instances (I_1, q_1, a_1) (Section 4.1). Then, for each such instance, we generate a variation (I_2, q_2, a_2) by either perturbing the original question q_1 or image I_1 (Section 4.2). Further, each test set is composed of a diverse set of questions. These may broadly be grouped into *verification* (or *binary*) questions, with expected answers being either *yes* or *no*, and *multi-choice* questions, with expected answers derived from a list of objects or attributes provided in the question.

4.1 Original instance generation

Questions for each test are generated from question templates (examples for each are provided in Appendix A.1) grouped into the following types.

Q1: **Object verification** (54 templates): e.g., "Is there a `<obj>` in the image?"

[2] Reusing terminology from Ribeiro et al. (2020b).

Q2: **Conjunctive verification** (18 templates): e.g., "Is there a `<obj1>` and a `<obj2>` in the image?"

Q3: **Disjunctive verification** (18 templates): e.g., "Is there a `<obj1>` or a `<obj2>` in the image?"

Q4: **Attribute verification** (25 templates): e.g., "Is the `<obj>` in the image `<attr>`?"

Q5: **Object multi-choice** (25 templates): e.g., "Is the `<obj-class>`, `<choices>`?"

Q6: **Attribute multi-choice** (39 templates): e.g., "What sort of `<attr-class>` is the `<obj>`, `<choices>`?"

Q7: **Action multi-choice** (28 templates): e.g., "What is the `<action-class>` that the `<obj>` doing, `<choices>`?"

Words in `typewriter` font represent template arguments. Generally, each `<obj>` argument can be filled by a singular object ("cup") or an attribute+object ("red cup") while `<attr>` arguments are filled with singular attributes ("shiny"). For object verification (Q1), attribute verification (Q4), attribute multi-choice (Q6), and action multi-choice (Q7) questions, some templates let `<obj>` arguments be filled with an object related to another object (e.g. "red cup *on the* table"); this type is excluded from conjunctive verification (Q2) and disjunctive verification (Q3) questions to prevent the generation of verbose questions.

For the **multi-choice** questions (Q5, Q6, Q7), `<choices>` are replaced with a list of 2 or 3 *singular* objects, attributes, or actions respectively (e.g. "cow or horse" or "wood, metal, or plastic"). The `<obj-class>` argument is filled with a hypernym of all object choices and always appears with either an attribute or a related object ("black animal", "animal *eating* grass"). The `<attr-class>` argument is filled with the attribute category of all attribute choices (e.g. "material"). Finally, the `<action-class>` argument is filled with the action category of all action choices (e.g. "sport").

Question argument generation. The question arguments are generated using images from the validation split of the GQA dataset (Hudson and Manning, 2019) which contains 10,696 images manually annotated with 1,536 different objects and 603 different object attributes.

To generate questions, we sample objects and attributes directly from an image's scene graph annotation to populate a question type's arguments. For *binary* question types this results in questions with solely affirmative answers. To produce an answer balanced dataset, we run a second stage of question argument generation for *binary* questions to generate *plausible* negative questions with *false* objects or attributes. We sample *false* objects from a distribution conditioned on an image's objects, and optionally sample object attributes from a distribution conditioned on the chosen object.

For `<choices>` arguments, *false* choices are again generated from a distribution conditioned on the object's hypernym for Q5 questions, the attribute category for Q6 questions, or the action category for the Q7 questions. We additionally ensure that the generated choices are mutually exclusive (e.g. "tan or beige" would be an invalid generation). To get more diverse multi-choice questions, we first generate a large pool of question candidates, and then select only a small number of questions sampled from this pool with sample probabilities inversely proportional to the count of the questions' hypernym, attribute class, or action class, and the count of the generated answer.

Question argument refinement. To improve the reliability of generated questions, we apply a variety of checks and constraints. For example, when sampling *false* objects from the conditional distribution, we filter out all objects (and their hypernyms[3]) present in the scene graph in order to guarantee that the sampled object is truly not present. We also filter out question arguments that are not included in the image scene graph but are sub-parts of objects that are annotated (e.g., "tire" when a "car" is annotated). Finally, we enforce various logical constraints on question arguments to prevent trivial or malformed questions. For example, for conjunctive and disjunctive questions (Q2, Q3), we apply a *hypernym exclusion* constraint to prevent questions like "Is there a `black cat` and a `black animal` in the image?".

4.2 Perturbed pair generation

We now describe our procedure for creating perturbed instances (I_2, q_2, a_2) for the six tests. In all tests except visual obfuscation, the image remains unchanged, i.e. $I_2 = I_1$.

[3] We use an ontology with hypernym paths generated with WordNet (Miller, 1995). We manually review and revise the default synset annotations from Visual Genome (Krishna et al., 2017) for the entire object vocabulary, and compare to a sample of annotated images to ensure general validity.

(a) Rephrasing invariance. Since each original question q_1 was generated using a text template, we simply use a different template of the same type to generate a valid rephrasing q_2. The image and answer remain the same, i.e. $I_1 = I_2, a_1 = a_2$ and the model is expected to be invariant to this rephrasing. We apply this to Q1, Q2, Q3, Q5, Q6 and Q7.

(b) Ontological invariance. Here, we use object verification questions (Q1) only and perform two types of transformations. For positive questions (i.e. a_1 = *yes*), we filter question arguments to only include objects that are valid hyponyms (using WordNet again) and use those to generate a perturbed question q_2 by changing the hyponym to a hypernym. For example, q_1 = "Do you see a jogging *woman*?" with a_1 = *yes* is paired with q_2 = "Do you see a jogging *person*?" containing a hypernym. Similarly, for negative questions (a_1 = *no*), we filter question arguments to only include valid hypernyms and generate a q_2: thus for example, q_1 = "Do you see a jogging *person*" with a_1 = *no* is paired with q_2 = "Do you see a jogging *woman*?" containing a hyponym, a_2 = *no* also.

(c) Order invariance. Order invariance tests apply to conjunctive verification, disjunctive verification, and all multi-choice question types; models are expected to be invariant to the logical order of arguments. We perturb conjunctive verification and disjunctive verification questions by swapping the questions' first and second arguments (`<obj1>, <obj2>`). For multi-choice question types, we perturb instances by generating the `<choices>` argument with different orders. The answer remains the same in both cases by construction.

(d) Visual obfuscation invariance. For this test, we let $q_1 = q_2$ and $a_1 = a_2$ but generate a perturbed image I_2 by obscuring parts of I_1 that are irrelevant to the question at hand using bounding box annotations from Visual Genome (Krishna et al., 2017). For all *true* objects in a question, we consider the bounding boxes around these object(s) to be the *foreground* and all other pixels in the image to be the *background*. For negative verification questions asking about object(s) not present in the image, we select one (or two) random object bounding box(es) as the foreground and consider everything else to be the background, since focusing on any image region should not affect the model's answer.[4] We then apply five types of perturbations to obscure the background: (i-iii) **Gaussian blurring** using the soft masking method of (Yang et al., 2021) with light ($\sigma = 3$), medium ($\sigma = 6$), or heavy ($\sigma = 9$) blurring, (iv) **Masking** with the channel-wise averaged pixel value from the GQA (Hudson and Manning, 2019) training dataset, entirely obscuring the context, and (v) **Cropping** to the smallest rectangular region including the foreground. Example images are shown in Appendix A.2.

(e) Negation directional expectation. For the negation directional test, we use object verification, conjunctive verification, and disjunctive verification questions. Each question q_1 is perturbed by substituting the original's text template with a paired negated text template to create q_2. Since each perturbed question represents the negation of the original, the expected answers $a_1 \neq a_2$.

(f) Attribute antonym directional expectation. We perturb the generated attribute verification questions by changing the `<attr>` question argument to its antonym using WordNet. All attribute antonym relations are manually curated to remove unintuitive examples; questions with arguments without a valid antonym are discarded. The original and perturbed questions of a pair have opposite answers $a_1 \neq a_2$.

5 Experimental Setup

5.1 Human baseline

To assess the quality, difficulty and validity of the generated tests, we sample 100 question *pairs* (200 questions) from each question type for the 6 tests and procure 5 annotations per question from workers on Amazon Mechanical Turk. Workers are vetted for a 97% approval rating and minimum of 500 completed HITs. Workers take $\sim$ 2 minutes per task on average and are compensated \$0.50 per task and thus $\sim$ \$15 per hour. Each HIT include 24 questions total, including 4 verification questions[5], and typically include a variety of question types from each of our tests.

[4] We choose 32×32 as a minimum bounding box size, shown to be reliably recognized by humans (Torralba et al., 2008).

[5] Tasks are also interspersed with binary or multi-choice "gold-standard" questions with perfect annotator agreement from the VQA dataset (Antol et al., 2015) which are required to be answered correctly before a HIT can be submitted. Workers are given the opportunity to correct answers before submitting if a "gold-standard" question has been answered incorrectly.

Human agreement. In addition to "yes" and "no" for binary questions and the appropriate choices for multi-choice questions, all questions offer an *ambiguous* option. Human answers are the majority vote among the 5 workers; questions failing to reach majority or with *ambiguous* as the majority are always counted against accuracy. This process is inspired by the human evaluations of implied question pairs in Ribeiro et al. (2020a). We report both human and model performance in Section 6.

5.2 Evaluated models

We evaluate six recent models on our tests, and compare them to human accuracy. Models are trained on the GQA (Hudson and Manning, 2019) balanced training split (using hyperparameters suggested from the original papers). All models, except LXMERT[6], are trained and finetuned using the MMF (Singh et al., 2020) library and region of interest (RoI) features from Faster R-CNN (Ren et al., 2015) with a ResNeXt-152 (Xie et al., 2017) backbone pre-trained on the Visual Genome (Krishna et al., 2017) dataset for object-based models. More details are provided in Appendix A.3.

Model initialization and pre-training. Of the six models evaluated, a defining characteristic of each model relates to their initializations, image-encoding choice, and the use of multi-modal pre-training. Our most basic model (CNN+LSTM) is randomly initialized and uses no pre-trained components; however, GloVe (Pennington et al., 2014) word embeddings are used for representing tokens. Another class of models use pre-trained image encoders to extract object features from images. Of our models, BAN (Kim et al., 2018) is randomly initialized prior to training but ingests pre-trained Faster R-CNN features which should provide the model with enhanced visual capabilities over the CNN+LSTM model. MultiModal BiTransformer (MMBT) (Kiela et al., 2019), uses similar pre-trained image features as BAN but is further initialized with pre-trained BERT (Devlin et al., 2019) weights prior to training on GQA. The last class of models are multi-modal pre-trained models; those that use pre-trained image features and are pre-trained on multi-modal tasks, such as image-based masked language modeling. Models in this class include LXMERT (Tan and Bansal, 2020), ViLBERT (Lu et al., 2019), and VisualBERT (Li et al., 2019). Similar to MMBT, ViLBERT and VisualBERT are also initialized with pre-trained weights from BERT.

5.3 Metrics

Accuracy (ACC). On our test datasets with K paired instances, we define accuracy as:

$$\frac{1}{2K}\sum_{i=1}^{K} \mathbb{1}[\hat{a}_1^i = a_1^i] + \mathbb{1}[\hat{a}_2^i = a_2^i]$$

where the model answers $\hat{a}_a^i, \hat{a}_2^i$ on the original and perturbed questions respectively are compared to the ground truth answers a_1^i, a_2^i.

Self-consistency (CONS). We measure self-consistency of the model predictions across the original and perturbed questions as

$$\frac{1}{K}\sum_{i=1}^{K} \begin{cases} \mathbb{1}[\hat{a}_1^i = \hat{a}_2^i] & \text{on invariance tests} \\ \mathbb{1}[\hat{a}_1^i \neq \hat{a}_2^i] & \text{on directional exp. tests} \end{cases}$$

Note that these metrics only measure the internal consistency of the model and do not include the ground truth answers a_1^i, a_2^i.

Comprehensive accuracy (C-ACC). We define comprehensive accuracy as:

$$\frac{1}{K}\sum_{i=1}^{K} \mathbb{1}[\hat{a}_1^i = a_1^i \wedge \hat{a}_2^i = a_2^i]$$

measuring whether model predictions are both accurate and self-consistent across perturbations.

6 Results

(R1) Modern VQA models are not robust to invariance and directionality tests. Fig 2 details the performance of various models under our suite of tests. Each bar in the figure shows both ACC and C-ACC for the model, with the arrow representing the gap between the two. We first observe that all models achieve significantly lower performance (at least a 8% drop) compared to humans (grey). Even simple tests such as REPHRASE-INV (Figure 2(a)) prove to be quite challenging, with models managing $< 68\%$ ACC compared to humans' 86%. On tests like NEGATION-DIR (Figure 2(e)), models only get about 50% accuracy, substantially worse than human scores of 78%.

Moreover, C-ACC is considerably lower than ACC across the board, with as much as a 35% gap on NEGATION-DIR and 14.5% on ANTONYM-DIR tests, even for a state-of-the-art model like LXMERT.[7] Even though this gap is smaller on

[6] For LXMERT we use the authors' open source repository at https://github.com/airsplay/lxmert

[7] Other modern systems like ViLBERT and VisualBERT

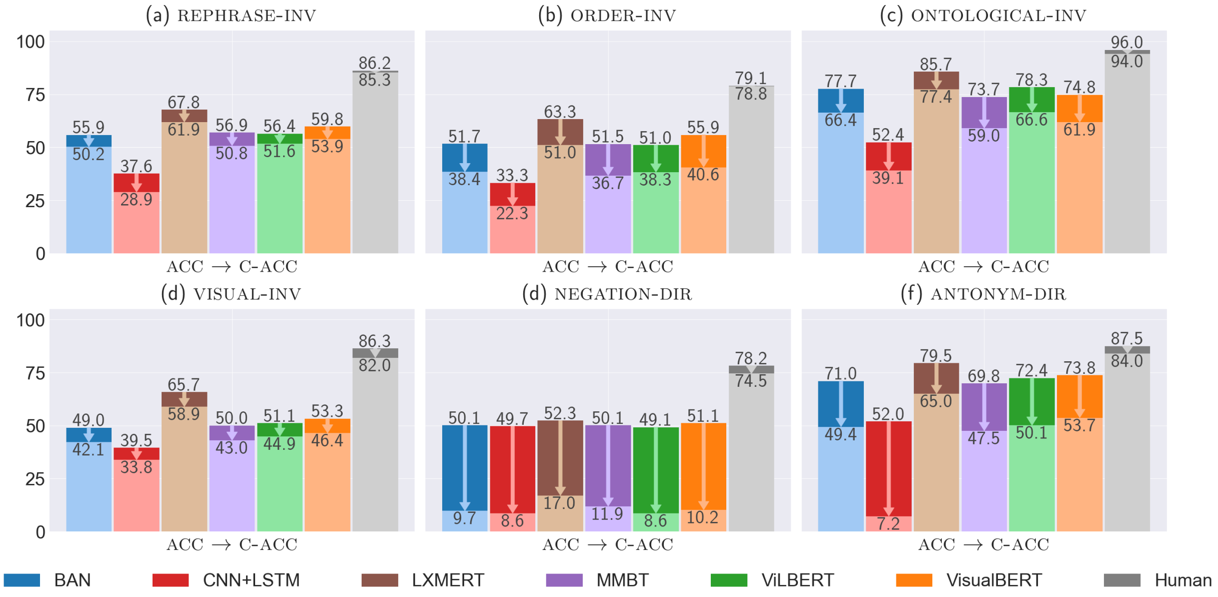


Figure 2: ACC and C-ACC across all six tests. All models perform worse than humans and exhibit consistent drops in C-ACC from ACC, especially struggling on NEGATION-DIR and ANTONYM-DIR tests. Best viewed in color.

	CONS					
	Reph.	Order	Onto.	Vis.	Neg.	Attr.
BAN	82.3	65.2	77.4	82.0	56.8	19.3
CNN+LSTM	68.9	62.6	73.4	80.0	10.5	17.8
LXMERT	83.8	**71.0**	**83.5**	78.6	**71.1**	**28.1**
MMBT	81.7	58.6	70.5	81.4	54.6	23.1
ViLBERT	**86.4**	66.2	76.5	83.4	55.3	18.7
VisualBERT	83.6	60.8	74.1	**83.8**	59.8	18.3
Human	96.6	98.8	94.0	89.0	89.0	88.5

Table 1: Self-consistency scores for all tests. Models only achieve middling self-consistency, compared to >89% human self-consistency.

	Orig.	Pert.	Hyper-	Hypo-
BAN	**79**	76	75	**79**
CNN+LSTM	49	**56**	**75**	72
LXMERT	**89**	82	80	**87**
MMBT	**82**	66	61	**80**
ViLBERT	**85**	72	69	**84**
Visual BERT	**81**	69	68	**80**
Human	**96**	**96**	91	**96**

Table 2: Ontological invariance results breakdown: original vs perturbed accuracy, and hyponym vs hypernym self-consistency.

other tests like REPHRASE-INV or VISUAL-INV, the performance drop is still at least 6-7% in most cases. This means that models are not invariant to textual rephrasings of questions and do not have a strong grasp of concepts like attributes and negation, despite negation of attributes appearing in the GQA training dataset.

(R2) VQA systems are not self-consistent in their predictions. Table 1 shows the self-consistency scores for all models under our different tests. While humans achieve CONS > 88% in all the tests, VQA models are much worse – at least 6% lower CONS in all cases, with the best performing model (LXMERT) still 26% lower than human performance on average across tests and models. Scores are especially low on the directional tests (antonym and negation), which means that models are confused in their decisions simply with the addition of negation words – this hints at issues of overfitting to spurious feature without understanding the presence or absence of specific concepts, corroborating the findings of (Bitton et al., 2021). Interestingly, the best performing model (LXMERT) is not always the most consistent. Furthermore, there is no single model that is the most self-consistent, with LXMERT, ViLBERT and VisualBERT each returning the highest consistency scores on different tests.

(R3) Models are more robust to hyponym than hypernym variations. Breaking out the results on the ontological invariance test (Figure 2 (c)) in the last two columns of Table 2, we see that self-consistency is higher on the hyponym perturbations (on negative answer questions) than on hypernym perturbations (positive questions); this

have even worse C-ACC.

	Conjunction				Disjunction			
	ACC	Y	N	O	ACC	Y	N	O
BAN	52	53	47	0	52	**71**	28	0
CNN+LSTM	39	53	47	0	35	**65**	35	0
LXMERT	78	49	51	0	56	**59**	32	9
MMBT	56	50	50	0	55	**63**	34	2
ViLBERT	58	54	46	0	56	**79**	21	0
VisualBERT	59	49	51	0	57	**70**	30	0

Table 3: Conjunctive (Con) vs disjunctive (Dis) comprehensive accuracy on ORDER-INV, along with a breakdown of response rates for yes (Y), no (N) and other than yes/no (O).

	ACC	CONS	C-ACC	Human ACC
Binary	**74.5**	**89.3**	**69.8**	82.8
Multi-choice	59.0	69.7	47.3	83.6
2-choice	62.4	70.8	49.9	83.4
3-choice	55.4	68.4	44.5	83.9

Table 4: Comparison of LXMERT's behavior for questions with binary answer types and multi-choice answers. Metrics are reported as averages over rephrasing invariance and order invariance tests. Human ACC reports human performance across test sets.

effect is particularly noticeable for MMBT and ViLBERT with a 19% and 15% difference, respectively. Thus, when an object is not detected in an image its hyponym elicits a negative response as expected; however when an object (like "steak") is detected, the hypernym question ("Is there any *meat* in the image?") may trip the model to generate a negative response. This points to the need for more structured, hierarchical grounding of concepts in these models.

(R4) Models perform better on conjunctive rather than disjunctive tests. From Table 3, we note that models generally have higher accuracy on conjunctive rather than disjunctive tests, with the largest discrepancy for LXMERT at 81% accuracy on conjunctive tests vs only 62% on disjunctive. Many models seem to exhibit a strong positive bias for disjunctive questions, suggesting they may just be short-cutting to answering 'yes' for disjunctive questions. LXMERT also seems to frequently confuse disjunctive questions for an open-ended or multi-choice question.

(R5) Models are sensitive to answer types and the number of choices in a question. Table 4 provides a breakdown of LXMERT's scores for binary and multi-choice questions. It is evident that multi-choice questions are harder for the model, with self-consistency dropping by 16% between binary and multi-choice questions, and C-ACC dropping by 33%. This is surprising since the multi-choice questions only include two or three choices and hence are quite similar to the binary (yes/no) questions. This may indicate a bias in the models towards binary questions with simple answers. Furthermore, Table 4 also shows that models consistently perform worse on 3-choice questions than 2-choice ones, with even the top-performing LXMERT having a 7% drop from 62% to 55%. This hints that there may be some effect of randomness in the way these models pick their answers. In contrast and as expected, humans are robust to the number of choices.

	LXMERT (Augmented)		
	ACC	CONS	C-ACC
REPHRASE-INV	84.0 ↑ 16.2	94.1 ↑ 10.3	81.5 ↑ 19.6
ORDER-INV	82.1 ↑ 18.8	92.3 ↑ 21.3	78.8 ↑ 27.8
ONTOLOGICAL-INV	95.3 ↑ 9.6	94.0 ↑ 10.5	92.4 ↑ 14.9
NEGATION-DIR	88.3 ↑ 36.0	90.5 ↑ 62.4	83.6 ↑ 66.6
ANTONYM-DIR	65.0 ↑ 5.5	83.5 ↑ 12.4	76.7 ↑ 11.6

Table 5: We augment the LXMERT model by adding 95,000 questions (9.9% of the training dataset) to the training dataset and keep all other training procedures the same. The added questions are generated using the same methodology as CARETS but using images from the GQA training set. GQA validation accuracy stays comparable at 70.60% for LXMERT (Augmented) vs. 70.67% for LXMERT (GQA).

(R6) Visual perturbations are easier for models to deal with. From Figure 2 and Table 1, we notice that the models are slightly more robust to visual perturbations on average compared to the lexical ones. All models only have a drop of 4-8% from ACC to C-ACC, while self-consistency of all models is also 78% or higher. Appendix A.2 provides a more detailed breakdown of all the different visual perturbation tests we performed.

(R7) Direct data augmentation improves CARETS evaluation. We show the feasibility of high performance on CARETS through data augmentation. We add 95,000 questions generated from CARETS question templates, and using a similar distribution of question types, to the original training split of GQA and re-train the LXMERT model. Table 5 shows that this dramatically improves the model on all three metrics (ACC, self-consistency and C-ACC), with the LXMERT(Augmented) model achieving near human performance on tests like ORDER-INV and ANTONYM-DIR. Since CARETS is designed to be an evaluation suite, these numbers show that CARETS questions should generally be within the capabilities of existing SOTA models, provided that they are able to generalize appropriately.

7 Conclusion & Future Work

In this work, we have developed CARETS – a new test suite for capability-focused robustness testing and comprehension evaluation of visual question answering (VQA) models. CARETS consists of six different tests that use instance *pairs* to evaluate models on their understanding of various linguistic and visual concepts. Using this test suite, we evaluated six modern VQA systems to reveal several inconsistencies in state-of-the-art models and provide a fine-grained view of their comprehension of different visuo-linguistic concepts. Quite surprisingly, we find that even state-of-the-art models struggle with concepts like negation, disjunction, order invariance and multi-choice questions. CARETS can also support the addition of more tests in the future and we view it as a platform for continually stress testing and improving VQA models.

CARETS emulates previous work in using text templates to generate questions and their textual perturbations (Hudson and Manning, 2019; Johnson et al., 2017; Ribeiro et al., 2020b). The use of templates to generate perturbations is motivated by the desire to maintain the grounded integrity of generations, ensuring that they remain relevant and that the generated label is true in the context of the subject image. While we have sought to generate a diverse language set through using a large number of templates (nearly 200 in total), there are some limitations to this approach. An improvement to our approach may be able to generate more sophisticated questions and perturbations through conditional text generation (Schick and Schütze, 2021; Madaan et al., 2021; Wu et al., 2021) while also preserving our other motivations, such as atomicity and grounded relevancy.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2107048. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We would also like to thank Tianyu Gao, Austin W. Hanjie and Alexander Wettig for their valuable feedback and advice.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

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

Yonatan Bitton, Gabriel Stanovsky, Roy Schwartz, and Michael Elhadad. 2021. Automatic generation of contrast sets from scene graphs: Probing the compositional consistency of gqa. *ArXiv*, abs/2103.09591.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT.*

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Justin Johnson, Li Fei-Fei, Bharath Hariharan, C. Lawrence Zitnick, Laurens Van Der Maaten, and Ross Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Justin Johnson, Ranjay Krishna, Michael Stark, Li Jia Li, David A. Shamma, Michael S. Bernstein, and Fei Fei Li. 2015. Image retrieval using scene graphs. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In

IEEE International Conference on Computer Vision (ICCV).

Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. 2020. Roses are red, violets are blue... but should vqa expect them to?

Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. 2019. Supervised multimodal bitransformers for classifying images and text.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, and et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language.

Qing Li, Qingyi Tao, Shafiq Joty, Jianfei Cai, and Jiebo Luo. 2018. Vqa-e: Explaining, elaborating, and enhancing your answers for visual questions. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *AAAI*.

Varun Manjunatha, Nirat Saini, and Larry S. Davis. 2019. Explicit bias discovery in visual question answering models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Spyridon Mouselinos, Henryk Michalewski, and Mateusz Malinowski. 2022. Measuring CLEVRness: Black-box testing of visual reasoning models. In *International Conference on Learning Representations*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2020a. Are red roses red? Evaluating consistency of question-answering models. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020b. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *EMNLP*.

Ramprasaath R. Selvaraju, Purva Tendulkar, Devi Parikh, E. Horvitz, Marco Túlio Ribeiro, Besmira Nushi, and Ece Kamar. 2020. Squinting at vqa models: Introspecting vqa models with sub-questions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. 2019. Cycle-consistency for robust visual question answering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2020. Mmf: A multimodal framework for vision and language research. `https://github.com/facebookresearch/mmf`.

Hao Tan and Mohit Bansal. 2020. LXMert: Learning cross-modality encoder representations from transformers. *Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

A. Torralba, R. Fergus, and W. Freeman. 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics (TACL).*

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. 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)*, pages 6707–6723, Online. Association for Computational Linguistics.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Kaiyu Yang, Jacqueline Yau, Li Fei-Fei, Jia Deng, and Olga Russakovsky. 2021. A study of face obfuscation in imagenet.

P. Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and yang: Balancing and answering binary visual questions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

A Appendix

A.1 Test Dataset Examples

We provide examples of templates for each question type here. In Section 4.1, we simplified the template arguments for readability. Here, we show template examples with their arguments as they are actually represented for generation.

Q1: **Object verification** (54 templates)

```
"Does it look like there [1:Is] [1:DET] <attrs1> <obj1> anywhere?"
"[1:Is] there [1:DET] <attrs1> <obj1> anywhere?"
```

Q2: **Conjunctive verification** (18 templates):

```
"[1:Is] there both [1:DET] <attrs1> <obj1> and [2:DET] <attrs2> <obj2>?"
"Do you see both [1:DET] <attrs1> <obj1> and [2:DET] <attrs2> <obj2> in this photo?"
```

Q3: **Disjunctive verification** (18 templates):

```
"Do you see either [1:DET] <attrs1> <obj1> or [2:DET] <attrs2> <obj2>?"
"Does it look like there is either [1:DET] <attrs1> <obj1> or [2:DET] <attrs2> <obj2> anywhere"
```

Q4: **Attribute verification** (25 templates):

```
"Does it seem that the <obj1> that [1:Is] <2rel1> the <attrs2> <obj2> [1:Is] <attrs1>?"
"Would you say that the <obj1> <2rel1> the <attrs2> <obj2> [1:Is] <attrs1>?"
```

Q5: **Object multi-choice** (25 templates):

```
"What would you call the <obj-category1> that [1:Is] <2rel1> the <attrs2> <obj2>, <obj-category-options1>?"
"Do you think the <attrs1> <obj-category1> [1:Is] <obj-category-options1>?"
```

Q6: **Attribute multi-choice** (39 templates):

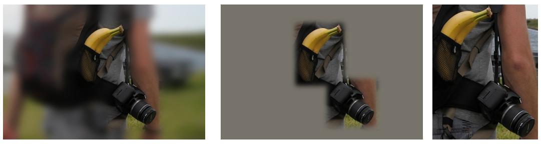
```
"Which sort of <category1> [1:Is] the <attrs1> <obj1>, <category-options1>?"
"What type of <category1> [1:Is] the <attrs1> <obj1>, <category-options1>?"
```

Q7: **Action multi-choice** (28 templates):

```
"What kind of <category1> [1:Is] the <attrs1> <obj1> doing, <category-options1>?"
"What would you call the sort of <category1> that the <attrs1> <obj1> [1:Is] doing, <category-options1>?"
```

A.2 Visual Obfuscation Invariance Details

Table 3 shows examples of context blurring, masking, and cropping. Five perturbations are done in total, blurring (with $\sigma \in \{3, 6, 9\}$), masking context by replacing pixel values with the channel-wise average computed from the GQA training data, and cropping around the tightest bounding box containing the question's objects.



Gaussian blur ($\sigma = 9$) Masking Cropping

Figure 3: Visual obfuscation for the question "Is there both a banana and a black camera in this photo?"

A.3 Training Details

We provide greater detail on training environments and hyperparameter choices.

All models are trained using the GQA (Hudson and Manning, 2019) balanced training set and validated on the balanced validation set (with minimal parameter tuning, aiming to stay faithful to

Test Dataset	Binary	Multi-choice	Total
REPHRASE-INV	10,000	9,412	19,412
ORDER-INV	5,000	9,412	14,412
ONTOLOGICAL-INV	13,952	-	13,952
VISUAL-INV	18,000	8,272	26,272
NEGATION-DIR	10,000	-	10,000
ANTONYM-DIR	5,000	-	5,000

Table 6: Test dataset statistics: answer distributions for *original* (non-perturbed) questions in each test dataset.

	O	M	C	G3	G6	G9
BAN	48.83	49.38	47.70	49.10	49.29	49.42
CNN+LSTM	40.28	39.29	40.14	39.45	39.18	39.35
LXMERT	69.64	66.33	61.82	66.59	66.89	67.06
MMBT	50.37	50.00	49.17	50.44	50.27	50.18
ViLBERT	51.60	51.12	49.99	51.69	51.36	51.22
VisualBERT	53.81	53.25	52.53	53.67	53.39	53.44

Table 7: Visual Accuracy

the original implementation). All non-LXMERT models are trained and finetuned withing the MMF (Singh et al., 2020) framework, trained using binary cross-entropy loss for a set maximum number of epochs, taking the epoch checkpoint that best performs on the validation set, and using features from Faster R-CNN (Ren et al., 2015) with a ResNext-152 (Xie et al., 2017) backbone pre-trained on the Visual Genome (Krishna et al., 2017) dataset for object-based models.

BAN. The Bilinear Attention Network (Kim et al., 2018) (BAN) uses pre-trained Faster R-CNN features (Ren et al., 2015) and GloVe (Pennington et al., 2014) embeddings with an attention model and early bilinear fusion mechanism. We train "four glimpse" (with 4 attention heads) BAN (Kim et al., 2018) (BAN) for 13 epochs using the Adamax (Kingma and Ba, 2015) optimizer with an initial learning rate of $1e^{-3}$ and batch size of 256, decaying the learning rate at epochs 11 and 13. As in the original hyperparameter configuration, we perform gradient clipping at 0.25.

LXMERT. LXMERT (Tan and Bansal, 2020) is a transformer-based architecture (Vaswani et al., 2017) with pre-trained Faster R-CNN features. It undergoes an extended pre-training procedure using 5 different pre-training tasks, including image question answering. We first pre-train a version of LXMERT from scratch with all GQA validation instances removed from the pre-training data to prevent direct leakage. We use all the default hyperparameters used in the author's GitHub repository.[8] We then finetune LXMERT base on the GQA training dataset for 4 epochs with the same hyperparameter configuration as the original implementation, using a batch size of 32, and initial learning rate of $1e^{-5}$. LXMERT base is pre-trained using the MS COCO (Lin et al., 2014) and Visual Genome datasets. LXMERT also uses a Faster R-CNN (Ren et al., 2015) with a ResNet-101 (He et al., 2016) backbone.

VisualBERT. VisualBERT (Li et al., 2019) is similar in architecture and pre-training method to BERT (Devlin et al., 2019). It performs an early fusion of text and image features immediately before several transformer layers. It uses Faster R-CNN features, is initialized using weights from BERT, and pre-trained on 2 different tasks using the MS COCO dataset. We finetune an MS COCO pre-trained version of VisualBERT using the same hyper parameters and training scheme of the original implementation for the VQA task. We use the Adam W optimizer with an initial learning rate of $2e^{-5}$ and a batch size of 64 for a maximum of 20 epochs.

[8] github.com/airsplay/lxmert

CNN+LSTM. This model uses a 6 layer CNN module and a bidirectional LSTM module before concatenating the output of each module and passing the combined output to a FC classifier. The LSTM module uses GloVe word embeddings (Pennington et al., 2014). Model weights are randomly initialized. We train the model for 25 epochs using the Adam W optimizer with an initial learning rate of $1e^{-4}$ and batch size of 256. This model uses a 6 layer CNN module and a bidirectional LSTM module with a hidden size of 128, and concatenates the output of each module before passing the combined output to 2 layer MLP classifier with a ReLU activation. The LSTM module uses GloVe word embeddings (Pennington et al., 2014) to represent questions.

MMBT. The MultiModal BiTransformer (MMBT) (Kiela et al., 2019) is an early fusion model, which uses Faster R-CNN features projected to a common space and concatenated with contextual BERT embeddings before being passed to transformer layers. MMBT uses pre-trained Faster R-CNN features and is initialized with BERT pre-trained weights. We finetune MMBT with the Adam W (Kingma and Ba, 2015) optimizer with an initial learning rate of $5e^{-5}$ with a batch size of 64, for a maximum of 15 epochs.

ViLBERT. ViLBERT (Lu et al., 2019) uses two parallel transformer "streams" for vision and language separately. These streams interact using multi-modal co-attentional transformer blocks. It uses Faster R-CNN features and is initialized using some weights from BERT. ViLBERT is pre-trained using 2 different tasks, using the MS COCO dataset. We finetune the MS COCO pre-trained version of ViLBERT in a similar manner to the finetuning scheme used for the VQA task of the original implementation. We use the Adam W (Kingma and Ba, 2015) optimizer with an initial learning rate of $4e^{-5}$ and batch size 64, for a maximum of 20 epochs.

A.4 Test Results

We provide fuller test results for each test dataset, including accuracy on the original instances and perturbed instances in tables 8, 9, 10, 11, and 12. These results supplement the primary results reported in Section 6.

	ACC	Original ACC	Perturbed ACC	CONS	C-ACC
BAN	55.86	55.90	55.83	82.32	50.20
CNN+LSTM	37.58	37.61	37.56	68.93	28.90
LXMERT	67.80	67.70	67.90	83.83	61.89
MMBT	56.91	56.79	57.02	81.72	50.76
ViLBERT	56.44	56.54	56.35	86.44	51.63
VisualBERT	59.81	60.13	59.48	83.62	53.87
Human	86.15	86.74	85.56	96.58	85.28

Table 8: Full results for REPHRASE-INV.

	ACC	Original ACC	Perturbed ACC	CONS	C-ACC
BAN	51.69	51.81	51.57	65.22	38.43
CNN+LSTM	33.25	33.24	33.26	62.56	22.29
LXMERT	63.26	63.22	63.30	71.02	51.01
MMBT	51.52	51.30	51.75	58.62	36.65
ViLBERT	51.02	51.01	51.04	66.21	38.27
VisualBERT	55.87	55.80	55.93	60.75	40.60
Human	79.10	79.38	78.83	98.82	78.76

Table 9: Full results for ORDER-INV.

A.5 Model comparison results

We provide additional model comparison results for each test dataset in Tables 13, 14, 15, 16, 17.

	ACC	Original ACC	Perturbed ACC	CONS	C-ACC
BAN	77.67	79.25	76.10	77.36	66.36
CNN+LSTM	52.37	48.70	56.04	73.37	39.06
LXMERT	85.69	88.96	82.42	83.48	77.43
MMBT	73.73	81.84	65.62	70.46	58.97
ViLBERT	78.30	84.96	71.65	76.48	66.56
VisualBERT	74.82	80.53	69.10	74.13	61.88
Human	96.00	96.00	96.00	94.00	94.00

Table 10: Full results for ONTOLOGICAL-INV.

	ACC	Original ACC	Perturbed ACC	CONS	C-ACC
BAN	71.02	81.70	60.34	56.80	49.42
CNN+LSTM	51.98	70.64	33.32	10.50	7.24
LXMERT	79.47	85.14	73.80	71.12	65.04
MMBT	69.83	82.74	56.92	54.58	47.54
ViLBERT	72.41	88.94	55.88	55.30	50.06
VisualBERT	73.76	82.16	65.36	59.84	53.68
Human	87.50	86.00	89.00	89.00	84.00

Table 11: Full results for ANTONYM-DIR.

	ACC	Original ACC	Perturbed ACC	CONS	C-ACC
BAN	50.10	64.64	35.57	19.30	9.73
CNN+LSTM	49.67	48.51	50.82	17.85	8.59
LXMERT	52.31	76.08	28.55	28.11	17.00
MMBT	50.13	67.41	32.86	23.07	11.90
ViLBERT	49.15	67.58	30.72	18.73	8.61
VisualBERT	51.10	68.67	33.53	18.26	10.23
Human	78.25	82.50	74.00	88.50	74.50

Table 12: Full results for NEGATION-DIR.

	B	C+L	L	M	ViL	Vis
BAN	-	75.70	69.42	75.58	76.88	73.97
CNN+LSTM	48.58	-	44.02	49.11	49.55	47.43
LXMERT	83.84	80.09	-	83.23	84.40	83.01
MMBT	76.53	76.22	69.77	-	79.47	77.58
ViLBERT	77.00	76.27	70.12	78.74	-	76.87
VisualBERT	78.95	76.93	73.39	81.89	81.80	-

Table 13: REPHRASE-INV model comparison: Entries ($C_{i,j}$) show the proportion of model j's correct predictions that are also predicted correctly by model i. E.g. here we see that BAN correctly predicts 65.39% of the instance pairs correctly predicted by LXMERT, but LXMERT correctly predicts 79.85% of the correct instance pairs captured by BAN.

	B	C+L	L	M	ViL	Vis
BAN	-	76.20	67.90	75.41	76.28	72.81
CNN+LSTM	46.63	-	41.31	47.71	47.94	45.95
LXMERT	82.49	78.80	-	82.10	82.86	81.14
MMBT	74.95	75.90	67.06	-	78.11	76.10
ViLBERT	74.88	75.72	67.02	77.31	-	74.11
VisualBERT	78.42	78.36	71.91	82.56	81.12	-

Table 14: Model coverage results for ORDER-INV.

	B	C+L	L	M	ViL	Vis
BAN	-	81.71	81.91	84.95	83.92	83.85
CNN+LSTM	55.09	-	53.05	54.57	53.64	53.70
LXMERT	90.36	86.81	-	91.20	91.81	90.89
MMBT	80.64	76.83	78.48	-	82.13	83.12
ViLBERT	84.60	80.21	83.90	87.22	-	88.14
VisualBERT	80.77	76.71	79.36	84.34	84.22	-

Table 15: Model coverage results for ONTOLOGICAL-INV.

	B	C+L	L	M	ViL	Vis
BAN	-	79.06	78.60	82.55	82.89	82.21
CNN+LSTM	57.94	-	55.13	59.96	59.20	57.48
LXMERT	87.98	84.30	-	87.19	88.06	87.63
MMBT	81.07	80.03	76.56	-	83.04	82.15
ViLBERT	84.48	82.22	80.20	86.21	-	84.84
VisualBERT	85.38	81.47	81.32	86.88	86.45	-

Table 16: Model coverage results for ANTONYM-DIR.

	B	C+L	L	M	ViL	Vis
BAN	-	55.26	66.06	68.95	71.67	68.69
CNN+LSTM	54.73	-	52.40	54.11	54.66	53.57
LXMERT	68.91	55.16	-	70.95	72.17	70.60
MMBT	68.99	54.64	68.08	-	73.13	71.83
ViLBERT	70.24	54.08	67.76	71.69	-	73.48
VisualBERT	70.04	55.16	68.96	73.19	76.43	-

Table 17: Model coverage results for NEGATION-DIR.