MUTANT: A Training Paradigm for Out-of-Distribution Generalization in Visual Question Answering

Tejas Gokhale * and **Pratyay Banerjee** * and **Chitta Baral** and **Yezhou Yang**
Arizona State University
tgokhale,pbanerj6,chitta,yz.yang@asu.edu

Abstract

While progress has been made on the visual question answering leaderboards, models often utilize spurious correlations and priors in datasets under the i.i.d. setting. As such, evaluation on out-of-distribution (OOD) test samples has emerged as a proxy for generalization. In this paper, we present *MUTANT*, a training paradigm that exposes the model to perceptually similar, yet semantically distinct *mutations* of the input, to improve OOD generalization, such as the VQA-CP challenge. Under this paradigm, models utilize a consistency-constrained training objective to understand the effect of semantic changes in input (question-image pair) on the output (answer). Unlike existing methods on VQA-CP, *MUTANT* does not rely on the knowledge about the nature of train and test answer distributions. *MUTANT* establishes a new state-of-the-art accuracy on VQA-CP with a 10.57% improvement. Our work opens up avenues for the use of semantic input mutations for OOD generalization in question answering.

1 Introduction

Availability of large-scale datasets has enabled the use of statistical machine learning in vision and language understanding, and has lead to significant advances. However, the commonly used evaluation criterion is the performance of models on test-samples drawn from the same distribution as the training dataset, which cannot be a measure of generalization. Training under this "independent and identically distributed" (i.i.d.) setting can drive decision making to be highly influenced by dataset biases and spurious correlations as shown in both natural language inference (Kaushik and Lipton, 2018; Poliak et al., 2018; McCoy et al., 2019) and visual question answering (Goyal et al., 2017; Agrawal et al., 2018a; Selvaraju et al., 2020). As such, evaluation on out-of-distribution (OOD) samples has emerged as a metric for generalization.

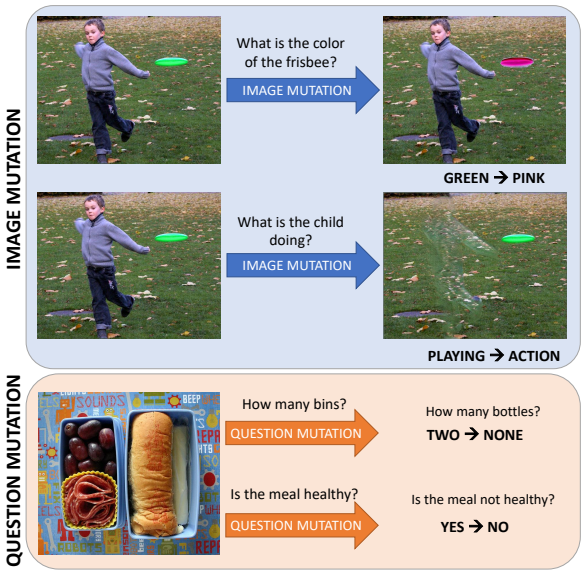

Figure 1: Illustration of the mutant samples. The input mutation, either by manipulating the image or the question, results in a change in the answer.

Visual question answering (VQA) (Antol et al., 2015) is a task at the crucial intersection of vision and language. The aim of VQA models is to provide an answer, given an input image and a question about it. Large datasets (Antol et al., 2015) have been extensively used for developing VQA models. However over-reliance on datasets can cause models to learn spurious correlations such as linguistic priors (Agrawal et al., 2018a) that are specific to certain datasets and do not generalize to "Out-of-Distribution" (OOD) samples, as shown in Figure 1. While learning patterns in the data is important, learning dataset-specific spurious correlations is not a feature of robust VQA models. Developing robust models has thus become a key pursuit for recent work in visual question answer-

*Equal Contribution

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 878–892,
November 16–20, 2020.

ing through data augmentation (Goyal et al., 2017), reorganization (Agrawal et al., 2018a).

Every dataset contains biases; indeed inductive bias is *necessary* for machine learning algorithms to work. Mitchell (1980) states that an unbiased learner's ability to classify is no better than a look-up from memory. However this bias has a component which is useful for generalization (positive bias), and a component due to spurious correlations (negative bias). We use the term "positive bias" to denote the correlations that are necessary to perform a task — for instance, the answer to a "What sport is ..." question is correlated to a name of a sport. The term "negative bias" is used for spurious correlations tat may be learned from the data — for instance, always predicting "tennis" as the answer to "What sport..." questions. The goal of OOD generalization is to mitigate negative bias while learning to perform the task. However existing methods such as LMH (Clark et al., 2019) try to remove all biases between question-answer pairs, by penalizing examples that can be answered without looking at the image; we believe this to be counter-productive. The analogy of antibiotics which are designed to remove pathogen bacteria, but also end up removing useful gut microbiome (Willing et al., 2011) is useful to understand this phenomenon.

We present a method that focuses on increasing positive bias and mitigating negative bias, to address the problem of OOD generalization in visual question answering. Our approach is to enable the **mutation** of inputs (questions and images) in order to expose the VQA model to perceptually similar yet semantically dissimilar samples. The intuition is to implicitly allow the model to understand the critical changes in the input which lead to a change in the answer. This concept of mutations is illustrated in Figure 1. If the color of the frisbee is changed, or the child removed, i.e. *when an image-mutation is performed*, the answer to the question changes. Similarly, if a word is substituted by an adversarial word (bins→bottles), an antonym, or negation (healthy→not healthy), i.e. *when a question-mutation is performed*, the answer also changes. Notice that both mutations do not significantly change the input, most of the pixels in the image and words in the question are unchanged, and the type of reasoning required to answer the question is unchanged. However the mutation significantly changes the answer.

In this work, we use this concept of mutations to enable models to focus on parts of the input that are critical to the answering process, by training our models to produce answers that are consistent with such mutations. We present a question-type exposure framework which teaches the model that although such linguistic priors may exist in training data (such as the dominant answer "tennis" to "What sport is ..." questions), other sports can also be answers to such questions, thus mitigating negative bias. This is in contrast to Chen et al. (2020a) who focus on using data augmentation as a means for mitigating language bias.

Our method uses a pair-wise training protocol to ensure consistency between answer predictions for the original sample and the mutant sample. Our model includes a projection layer, which projects cross-modal features and true answers to a learned manifold and uses Noise-Contrastive Estimation Loss (Gutmann and Hyvärinen, 2010) for minimizing the distance between these two vectors. Our results establish a new state-of-the-art accuracy of 69.52% on the VQA-CP-v2 benchmark outperforming the current best models by 10.57%. At the same time, our models achieves the best accuracy (70.24%) on VQA-VQA-v2 among models designed for the VQA-CP task.

This work takes a step away from explicit debiasing as a method for OOD generalization and instead proposes amplification of positive bias and implicit attenuation of spurious correlations as the objective. Our contributions are as follow.

- We introduce the Mutant paradigm for training VQA models and the sample-generation mechanism which takes advantage of semantic transformations of the input image or question, for the goal of OOD generalization.
- In addition to the conventional classification task, we formulate a novel training objective using Noise Contrastive Estimation over the projections of cross-modal features and answer embeddings on a shared projection manifold, to predict the correct answer.
- Our pairwise consistency loss acts as a regularization that seeks to bring the distance between ground-truth answer vectors closer to the distance between predicted answer vectors for a pair of original and mutant inputs.
- Extensive experiments and analyses demonstrate advantages of our method on the VQA-CP dataset, and establish a new state-of-the-art of $\mathbf{69.52}\%$, an improvement of $\mathbf{10.57}\%$.

2 MUTANT

We consider the open-ended VQA problem as a multi-class classification problem. The VQA dataset $\mathcal{D} = \{Q_i, I_i, a_i\}_{i=1}^{N}$ consists of questions $Q_i \in \mathcal{Q}$ and images $I_i \in I$, and answers $a_i \in \mathcal{A}$. Many contemporary VQA models such as Up-Dn (Anderson et al., 2018) and LXMERT (Tan and Bansal, 2019) first extract cross-modal features from the image and question using attention layers, and then use these features as inputs to a neural network answering module which predicts the answer classes. In this section we define our Mutant paradigm under this formulation of the VQA task.

2.1 Concept of Mutations

Let $X = (Q, I)$ denote an input to the VQA system with true answer a. A *mutant* input X^* is created by a small transformation in the image (Q, I^*) or in the question (Q^*, I) such that this transformation leads to a new answer a^*, as shown in Figure 1. There are three categories of transformation T that create the mutant input $X^* = T(X)$, addition, removal, or substitution. For image mutations, these correspond to addition or removal of objects, and morphing the attributes of the objects, such as color, texture, and lighting conditions. For instance addition or removal of a person from the image in Figure 3 changes the answer to the question "How many persons are pictured". Question mutations can be performed by addition of a negative word ("no", "not", etc.) to the question, masking critical words in the question, and substituting an object-word with an antonym or adversarial word. Thus for each sample in the VQA dataset, we can obtain a mutant sample and use it for training.

2.2 Training with Mutants

Our method of training with mutant samples relies on three key concepts that supplement the conventional VQA classification task.

Answer Projection: The traditional learning strategy of VQA models optimizes for a standard classification task using softmax cross-entropy:

$$\mathcal{L}_{VQA} = \frac{-1}{N} \sum_{i=1}^{N} log(softmax\big(f_{VQA}(X_i), a_i\big)). \quad (1)$$

QA as a classification task is popular since the answer vocabulary follows a long-tailed distribution over the dataset. However this formulation is problematic since it does not consider the meaning of the answer while making a decision, but instead learns a correlation between the one-hot vector of the answer-class and input features. Thus to answer the question "What is the color of the banana", models learn a strong correlation between the question features and the answer-class for "yellow", but do not encode the notion of *yellowness* or *greenness* of bananas. This key drawback negatively impacts the generalizability of these models to raw green or over-ripe black bananas at test-time.

To mitigate this, in addition to the classification task, we propose a training objective that operates in the space of answer embeddings. The key idea is to map inputs (image-question pairs) and outputs (answers) to a shared manifold in order to establish a metric of similarity on that manifold. We train a projection layer that learns to project features and answers to the manifold as shown in Figure 2. We then use Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010) as a loss function to minimize the distance between the projection of cross modal features z and projection of glove vector v for ground-truth answer a, given by:

$$\mathcal{L}_{NCE} = -log\Big(\frac{e^{cos(z_{feat},\ z_a)}}{\sum_{a_i \in \mathcal{A}} e^{cos(z_{feat},\ z_a^i)}}\Big), \quad (2)$$

where $z_{feat} = f_{proj}(z)$ and $z_a = f_{proj}(glove(a))$, and $\mathcal{A}$ is the set of all possible answers in our training dataset. It is important to note that this similarity metric is not between the true answer and the predicted answer, but between the projection of input features and the projection of answers, to incorporate context in the answering task.

Type Exposure: Linguistic priors in datasets have led models to learn spurious correlations between question and answers. For instance, in VQA, the most common answer for "What sport ..." questions is "tennis", and for "How many ..." questions is "two". Our aim is to remove this negative bias from the models. Instead of removing *all bias* from these models, we teach models to identify the question type, and learn which answers can be valid for a particular question type, irrespective of their frequency of occurrence in the dataset. For instance, the answer to "How many ..." can be all numbers, answers to "What color ..." can be all colors, and answers to questions such as "Is the / Are there ..." questions is either yes or no. We call this *Type Exposure* since it instructs the model that although a strong correlation may exist between a question-answer pair, there are other answers which are also

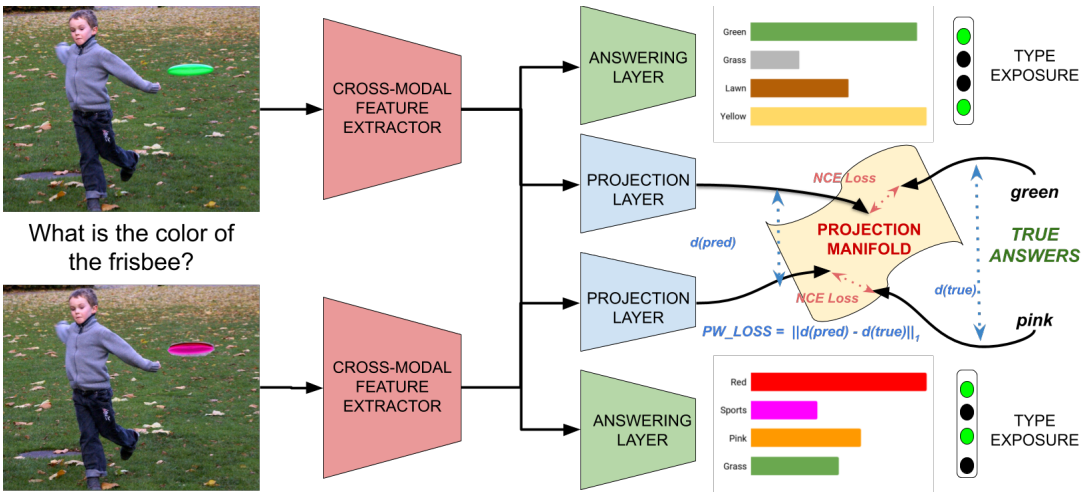

Figure 2: Overall architecture of the Mutant Method includes a cross-modal feature extractor, answer projection layer, answering layer and type exposure model

valid for the specific type of question. Our Type Exposure model uses a feedforward network to predict question type and to create a binary mask over answer candidates that correspond to this type.

Pairwise-Consistency: The final component of Mutant is pairwise consistency. We jointly train our models with the original and mutant sample pair, with a loss function that ensures that the distance between two predicted answer vectors is close to the distance between two ground-truth answer vectors. The pairwise consistency loss is given below, where z_a is the vector for answer a, m, GT denote mutant sample and ground-truth respectively.

$$\mathcal{L}_{PW} = ||cos(z_{a_{GT}}, z^m_{a_{GT}}) - cos(z_{a_{pred}}, z^m_{a_{pred}})||_1.$$

This pairwise consistency is designed as a regularization that incorporates the notion of semantic shift in answer space as a consequence of a mutation. For instance, consider the image mutation in Figure 3 which changes the ground-truth answer from "two" to "one". This shift in answer-space should be reflected by the predictor.

3 Generating Input Mutations for VQA

In order to train VQA models under the mutant paradigm, we need a mechanism to create mutant samples. Mutations are transformations that act on semantic entities in either the image or the question, in ways that can reliably lead to a new answer. For the question, semantic entities are words, while for images, semantic entities are objects. It is important to note that our mutation process is automated and does not use the knowledge about the test set

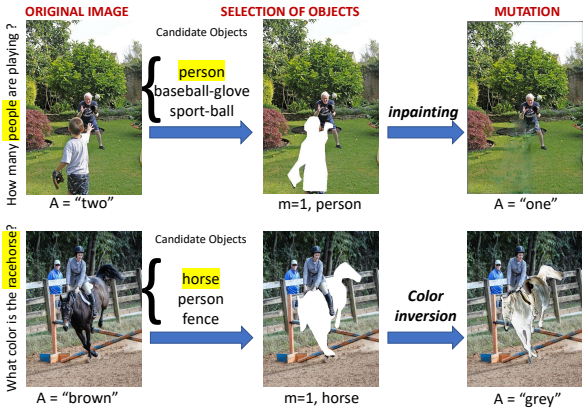

Figure 3: Figure illustrating our dataset creation pipeline for image mutations. m object instances of "critical" object are identified from the question and image, and mutation performed either by removal or color inversion. A represents the answer to the question.

distribution in order to create new samples. In this section, we delineate our automated generation process for both image and question-mutation.

3.1 Image Mutations

For image mutation, we first identify critical objects from the image that results in a change in the answer, and either remove instances of these objects (removal) or morph their color (substitution).

Removing Object Instances: Removing an instance of an object class can be either critical to the question (i.e. the answer to the question changes) or non-critical (i.e. answer is unchanged). If an object (or it's synonym or hypernym) is mentioned in the question, we deem it to be critical to the question, otherwise it is deemed non-critical. For

Mutation Type	Question	Answer
Original	Is the lady holding the baby?	Yes
Substitution (Negation)	Is the lady not holding the baby?	No
Substitution (Adversarial)	Is the cat holding the baby?	No
Original	How many people are there?	Three
Deletion (Masking)	How many [MASK] are there?	"Number"
Original	What is the color of the man's shirt?	Blue
Substitution (Negation)	What is not the color of the man's shirt?	Magenta
Deletion (Masking)	Is the [MASK] holding the baby?	Can't say
Original	What color is the umbrella ?	Pink
Deletion (Masking)	What color is the [MASK]?	"color"

Table 1: Examples of our question mutation. The image is shown on the left, and the original question is in the first row of the table. Examples of the two types of mutation are shown in the table.

each object with M instances in the image, we randomly remove m instances from the image s.t. $m \in \{0, \ldots, M\}$ using polygon annotations from the COCO (Lin et al., 2014) dataset. Thus for each image, we get multiple masked images, with pixels inside the instance bounding-box removed, as shown in Figure 3. These masked images are fed to a GAN-based inpainting network (Yu et al., 2018) that makes the mutant image photo-realistic, and also prevents the model from getting cues from the shape of the mask. In the case of numeric questions, if m critical objects are removed, the answer to for the mutant image changes from n to $n - m$. For yes-no questions, removal of all critical objects ($m = n$) will flip the answer from "yes" to "no", while removing $m < n$ critical objects will not. Note that $m = 0$ corresponds to the original image and does not result in a change in the answer.

Color Inversion: For mutations that involve a change in color, we use samples with questions about the color of objects in the image, and change the color of critical objects by pixel-level color inversion in RGB-space. The true answer is replaced with the new color of the critical objects. To get objects with new colors, we do not use the knowledge about colors of objects in the world. In some cases, the new colors of the object may not correspond to real-world scenes, thus forcing the model to actually identifying colors, and not answer from language priors, such as "bananas are yellow".

3.2 Question Mutations

We use three types of question mutations as shown in the example in Table 1. We first identify the critical object and then apply template-based question operators similar to (Gokhale et al., 2020). The first operator is negation for yes-no questions, which is achieved by a template based procedure that negates the question by adding a "no" or "not" before a verb, preposition or noun phrase. The second is the use of antonyms or adversarial object-words to substitute critical words. The third mutation masks words in the question and thus introduces ambiguity in the question. Questions for which the new answer cannot be deterministically identified are annotated with a broad category label such as *color, location, fruit* instead of the exact answers such as *red, library, apple* which the model cannot be expected to answer since some words have been masked or replaced with adversarial words. Yet, we want the model to be able to identify this broad category of answers even under partially occluded inputs. The answer remains unchanged for mutations with non-critical objects or words.

Mutation Category	Number of Samples
Object Removal	159,899
Color Change	30,759
Negation	237,611
Adversarial Substitution	146,814
Word Masking	104,666

Table 2: Distribution of generated mutant samples by category of mutation

3.3 Mutant Statistics:

We use the training set of VQA-CP-v2 (Agrawal et al., 2018a) to generate mutant samples. For each original sample, we generate 1.5 mutant samples on average, thus obtaining a total of 679k samples. Table 2 shows the distribution of our generated mutations with respect to the type of mutation. Addition of mutant samples does not change the distribution of samples per question-type.[1]

[1]More details about mutant samples are in Supp. material.

Model	VQA-CP v2 test (%) ↑				VQA-v2 val (%) ↑				Gap (%)
	All	Yes/No	Num	Other	All	Yes/No	Num	Other	
GVQA (Agrawal et al., 2018b)	31.30	57.99	13.68	22.14	48.24	72.03	31.17	34.65	16.94
AReg (Ramakrishnan et al., 2018)	41.17	65.49	15.48	35.48	62.75	79.84	42.35	55.16	21.58
RUBi (Cadene et al., 2019)	47.11	68.65	20.28	43.18	63.10	-	-	-	14.05
SCR (Wu and Mooney, 2019)	48.47	70.41	10.42	47.29	62.30	77.40	40.90	56.50	13.83
LMH (Clark et al., 2019)	52.45	69.81	44.46	45.54	61.64	77.85	40.03	55.04	9.19
CSS (Chen et al., 2020a)	58.95	84.37	49.42	48.21	59.91	73.25	39.77	55.11	0.96
UpDn (Anderson et al., 2018)	39.74	42.27	11.93	46.05	63.48	81.18	42.14	55.66	23.74
UpDn + Ours	61.72	88.90	49.68	50.78	62.56	82.07	42.52	53.28	0.84
LXMERT (Tan and Bansal, 2019)	46.23	42.84	18.91	55.51	**74.16**	**89.31**	**56.85**	**65.14**	27.97
LXMERT + Ours	**69.52**	**93.15**	**67.17**	**57.78**	70.24	89.01	54.21	59.96	**0.72**

Table 3: Accuracies on VQA-CP v2 test and VQA-v2 validation set, along with Percentage gap between overall accuracies on these two datasets. *"Ours"* represents the final model with Answer Projection, Type Exposure and Pairwise Consistency. Overall best scores are **bold**, our best are underlined.

4 Experiments

4.1 Setting

Datasets: We train and evaluate our models on VQA-CP-v2. This is a natural choice for evaluating OOD generalization since VQA-CP is a non-i.i.d. reorganization of the VQA dataset, and was created in order to evaluate VQA models in a setting where language priors cannot be relied upon for a correct prediction. This is because for every question type (65 types according to the question prefix), the prior distribution of answers is different in train and test splits of VQA-CP. We also train and evaluate our models on the VQA-v2 (Goyal et al., 2017) validation set, and compare the gap between the imbalanced and non-i.i.d. setting of VQA-CP against the balanced i.i.d. setting of VQA.

Hyperparameters: All of our models are trained on two NVIDIA Tesla V100 16GB GPUs for 10 epochs with batch size of 32 and learning rate $1e$–5. Each epoch takes approximately three hours for UpDn and four hours for LXMERT.

4.2 Baseline Models

We compare our method with GVQA (Agrawal et al., 2018b), RUBI (Cadene et al., 2019), SCR (Wu and Mooney, 2019), LMH (Clark et al., 2019), CSS (Chen et al., 2020a) as our baselines. Since most of these methods are built with UpDn (Anderson et al., 2018) as the backbone, we investigate the efficacy of UpDn under the mutant paradigm. On the other hand, LXMERT (Tan and Bansal, 2019) has emerged as a powerful transformer-based cross-modal feature extractor, and is pre-trained on tasks such as masked language modeling and cross-modality matching, inspired by BERT (Devlin et al., 2019). LXMERT is a top performing single-model on multiple vision-and-language tasks such as VQA, GQA (Hudson and Manning, 2019), ViZWiz (Bigham et al., 2010), and NLVR2 (Suhr et al., 2019). We therefore use is as a strong baseline for our experiments. LXMERT is representative of the recent trend towards using BERT-like pre-trained models (Lu et al., 2019; Su et al., 2019; Li et al., 2020; Chen et al., 2019) and fine-tuning them on multiple downstream vision and language tasks. Note that we do not use ensemble models for our experiments and focus only on single-model baselines.

4.3 Results on VQA-CP-v2 and VQA-v2

Performance on two benchmarks VQA-CP-v2 and VQA-v2 is shown in Table 3. We compare existing models against UpDn and LXMERT incorporated into our Mutant method. For the VQA-CP benchmark, our method improves the performance of LXMERT by 23.29%, thus establishing a new state of the art on VQA-CP, beating the previous best by 10.57%. Our method shows improvements across all categories, with 8.78% on the Yes-No category, 17.75% on Number-based questions, and 9.57% on other questions. We use negation as one of the question mutation operators on yes-no questions, but such questions are not present in the test set. However our model takes advantage of this mutation and improves substantially on yes-no questions. The Mutant method also consistently improves the performance of the UpDn model by 21.98% overall. Note that baseline models AReg, RUBI, SCR, LMH, and CSS all modify UpDn by

Model	Data	VQA-CP v2 test ↑ (%)			
		All	Yes/No	Num	Other
UpDn	VQA-CP	39.74	42.27	11.93	46.05
UpDn	VQA-CP + Mutant	50.16	61.45	35.87	50.14
Increase in Accuracy		*10.42*	*19.18*	*23.94*	*4.09*
LXMERT	VQA-CP	46.23	42.84	18.91	55.51
LXMERT	VQA-CP + Mutant	59.69	73.19	32.85	59.29
Increase in Accuracy		*13.46*	*30.35*	*13.94*	*3.78*
LXM + Ours	VQA-CP + Img. Mut.	64.85	85.68	66.44	53.80
LXM + Ours	VQA-CP + Que. Mut.	67.92	91.64	65.73	56.09
LXM + Ours	VQA-CP + Both Mut.	**69.52**	**93.15**	**67.17**	**57.78**

Table 4: Top section: Comparison of UpDn and LXMERT when trained on VQA-CP and augmented with mutant samples, and the increase in accuracy due to mutant samples. Bottom section: Comparison of LXMERT when using image or text mutations, or both.

adding de-biasing techniques. We show our de-biasing method improves on two SOTA models and outperforms all of the above baselines, unlike previous work which only modifies UpDn. This empirically shows Mutant to be model-agnostic.

When trained and evaluated on the balanced i.i.d. VQA-v2 dataset, our method achieves the best performance amongst methods designed specifically for OOD generalization, with an accuracy of 70.24%. This is closest among baselines to the SOTA established by LXMERT, which is trained explicitly for the balanced, i.i.d. setting. To make this point clear, we report the *gap* between the overall scores for VQA-CP and VQA-v2, following the protocol from Chen et al. (2020a) in Table 3.

Results on VQA-v2 without re-training:
Additionally, we use our best model trained on VQA-CP and evaluate it on the VQA test standard set without re-training on VQA-v2 data. The objective here is to evaluate whether models trained on biased data (VQA-CP) and mutant data is able to generalize to VQA-v2 which uses an i.i.d. train-test split. This gives us an overall accuracy of 67.63% comprising with 88.56% on yes-no questions, 50.76% on number-based questions, and 54.56% on other questions. This is better than all existing VQA-CP models that are explicitly trained on VQA-v2 (reported in Table 3), and thus demonstrates the generalizability of our approach.

4.4 Analysis

Effect of Training with Mutant Samples:
In this analysis we measure the effect of augmenting the training data with mutant samples on UpDn and LXMERT without any architectural changes.

Model	VQA-CP v2 test ↑ (%)			
	All	Yes/No	Num	Other
UpDn	50.16	61.45	35.87	50.14
UpDn + AP	54.51	88.35	41.01	32.89
UpDn + TE	56.32	80.56	46.14	46.41
UpDn + AP + TE	55.76	90.25	43.78	41.40
UpDn + AP + PW	57.54	91.59	49.17	41.93
UpDn + TE + PW	60.32	86.10	50.23	49.58
UpDn + AP + TE + PW	61.72	88.90	49.68	50.78
LXM	59.69	73.19	32.85	59.29
LXM + AP	60.45	88.46	43.24	50.49
LXM + TE	63.36	77.10	46.50	61.27
LXM + AP + TE	64.73	85.34	47.23	58.71
LXM + AP + PW	67.14	90.49	65.52	55.34
LXM + TE + PW	64.17	94.71	35.19	48.80
LXM + AP + TE + PW	**69.52**	**93.15**	**67.17**	**57.78**

Table 5: Ablation study to investigate the effect of each component of our method: Answer Projection (AP), Type Exposure (TE), Pairwise Consistency (PW), and independent effect of image and question mutations.

The results are reported in Table 4. Both models improve when exposed to the mutant samples, UpDn by 10.42% and LXMERT by 13.46%. There is a markedly significant jump in performance for both models for the yes-no and number categories. UpDn especially benefits from Mutant samples in terms of the accuracy on numeric questions (a boost of 23.94%).

We also compare our final model when trained only with image mutations and only with question mutations in Table 4. While this is worse than training with both types of mutations, it can be seen that question mutations are better than image mutations in the case of yes-no and other questions, while image mutations are better on numeric questions.

Ablation Study:
We conduct ablation studies to evaluate the efficacy of each component of our method, namely Answer Projection, Type Exposure and Pairwise Consistency, on both baselines, as shown in Table 5. Introduction of Answer Projection significantly improves yes-no performance, while Type Exposure improves performance on other questions. We also observe that the pairwise consistency loss significantly boosts performance on numeric questions and yes-no questions. Note that there is a minor difference between the original and the mutant sample, and the model needs to understand this difference, which in turn can enable the model to reason about the question and predict the new answer. For instance the pairwise consistency loss allows the

Model	Method	VQA-CP v2 test ↑ (%)			
		All	Yes/No	Num	Other
UpDn + Ours	Base	61.72	88.90	49.68	50.78
UpDn + Ours	LMH	55.38	90.99	39.74	40.99
Drop in Accuracy		*6.34*	*-2.09*	*9.95*	*9.80*
LXMERT + Ours	Base	69.52	93.16	67.17	57.78
LXMERT + Ours	LMH	63.85	88.34	48.23	55.28
Drop in Accuracy		*5.67*	*4.82*	*18.86*	*2.50*

Table 6: Effect of combining LMH de-biasing with the Mutant paradigm, measured as drop in accuracy (%)

model to learn the correlation between one missing object and a change in answer from "two" to "one" in Figure 3, resulting in an improvement in the counting ability of our VQA model. Similarly, the pairwise consistency allows the model to improve on yes-no questions for which the answer changes when a critical object is removed.

Effect of LMH Debiasing on Mutant:
We compare the results of our model when trained with or without the explicit de-biasing method LMH (Clark et al., 2019). LMH is an ensemble-based method trained for *avoiding* dataset biases, and is the most effective among all de-biasing strategies developed for the VQA-CP challenge. LMH implements a learned mixing strategy, by using the main model in combination with a bias-only model trained only with the question, without the image. The learned mixing strategy uses the bias-only model to remove biases from the main model. It can be seen from Table 6 that LMH leads to a drop in performance when used in combination with Mutant. This is potentially because in the process of debiasing, LMH ends up attenuating positive bias introduced by Mutant that is useful for generalization. Kervadec et al. (2020) have concurrently shown that de-biasing methods such as LMH indeed result in a decrease in performance on out-of-distribution (OOD) test samples in the GQA (Hudson and Manning, 2019) dataset, mirroring our analysis on VQA-CP shown in Table 6.

5 Related Work

De-biasing of VQA datasets: The VQA-v1 dataset (Antol et al., 2015) contained imbalances and language priors between question- answer pairs. This was mitigated by VQA-v2 (Goyal et al., 2017) which balanced the data by collecting complementary images such that each question was associated with two images leading to two different answers. Identifying that the distribution of answers in the VQA dataset led models to learn superficial correlations, Agrawal et al. (2018a) proposed the VQA-CP dataset by re-organizing the train and test splits such that the the distribution of answers per question-type was significantly different for each split.

Robustness in VQA: Ongoing efforts seek to build robust VQA models for VQA for various aspects of robustness. Shah et al. (2019) propose a model that uses cycle-consistency to not only answer the question, but also generate a complimentary question with the same answer, in order to increase the linguistic diversity of questions. In constrast, our work generates questions with a different answer. Selvaraju et al. (2020) provide a dataset which contains perception-related sub-questions for each VQA question. Antonym-consistency has been tackled in Ray et al. (2019). Inspired by invariant risk minimization (Arjovsky et al., 2019) which links out-of-distribution generalization to invariance and causality, Teney et al. (2020b) provide a method to identify invariant correlations in the training set and train models to ignore spurious correlations. Asai and Hajishirzi (2020); Gokhale et al. (2020) explore robustness to logical transformation of questions using first-order logic connectives *and* ($\wedge$), *or* ($\vee$), *not* ($\neg$). Removal of bias has been a focus of Ramakrishnan et al. (2018); Clark et al. (2019) for the VQA-CP task. We distinguish our work from these by amplifying positive bias and attenuating negative bias.

Data Augmentation: It is important to note that the above work on data de-biasing and robust models focuses on the language priors in VQA, but not much attention has been given to visual priors. Within the last year, there has been interest in augmenting VQA training data with counterfactual images (Agarwal et al., 2020; Chen et al., 2020a). Independently, Teney et al. (2020a) have also demonstrated that counterfactual images obtained via minimal editing such as masking or inpainting can lead to improved OOD generalization of VQA models, when trained with a pairwise gradient-based regularization. Self-supervised data augmentation has been explored in recent work (Lewis et al., 2019; Fabbri et al., 2020; Banerjee and Baral, 2020) in the domain of text-based question answering. The mutant paradigm presented in this work is one of the first enable the generation of VQA samples that

result in different answers, coupled with a novel architecture and a consistency loss between original and mutant samples as a training objective.

Answer Embeddings: In one of the early works on VQA, Teney and Hengel (2016) use a combination of image and question representations and answer embeddings to predict the final answer. Hu et al. (2018) learn two embedding functions that transform image-question pair and answers to a shared latent space. Our method is different from this since we use a combination of classification and NCE Loss on the projection of answer vectors, as opposed to a single training objective. This means that although the predicted answer is obtained as the most probable answer from a set of candidate answers, the NCE Loss in the answer-space embeds the notion of semantic similarity between the answer. Our Type Exposure model is in principal similar to Kafle and Kanan (2016) who use the predicted answer-type probabilities in a Bayesian framework, while we use it as an additional constraint, i.e. as a regularization for a maximum likelihood objective.

6 Discussion and Conclusion

In this paper, we present a method that uses input mutations to train VQA models with the goal of Out-of-Distribution generalization. Our novel answer projection module trained for minimizing distance between answer and input projections complements the canonical VQA classification task. Our Type Exposure model allows our network to consider all valid answers per question type as equally probable answer candidates, thus moving away from the negative question-answer linguistic priors. Coupled with pairwise consistency, these modules achieve a new state-of-the-art accuracy on the VQA-CP-v2 dataset and reduce the gap between model performance on VQA-v2 data.

We differentiate our work from methods using random adversarial perturbations for robust learning (Madry et al., 2018). Instead we view input mutations as *structured perturbations* which lead to a semantic change in the input space and a deterministic change in the output space. We envision that the concept of input mutations can be extended to other vision and language tasks for robustness. Concurrent work in the domain of image classification shows that carefully designed perturbations or manipulations of the input can benefit generalization and lead to performance improvements (Chen et al., 2020b; Hendrycks et al., 2019). While perception is a cornerstone of understanding, the ability to imagine changes in the scene or language query, and predict outputs for that *imagined* input allows models to supplement "what" decision making (based on observed inputs) with "what if" decision making (based on imagined inputs). The Mutant paradigm is an effort towards "what if" decision making. Code is available here.

Acknowledgements

The authors acknowledge support from the NSF Robust Intelligence Program project #1816039, the DARPA KAIROS program (LESTAT project), the DARPA SAIL-ON program, and ONR award N00014-20-1-2332.

References

Vedika Agarwal, Rakshith Shetty, and Mario Fritz. 2020. 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.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018a. Don't just assume; look and answer: Overcoming priors for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018b. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*.

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

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

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. In *ACL*.

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. In *EMNLP*.

Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342.

Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. 2019. Rubi: Reducing unimodal biases in visual question answering. In *NeurIPS*.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations. In *European Conference on Computer Vision*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4060–4073.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Alexander R Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. In *ACL*.

Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2020. Vqa-lol: Visual question answering under the lens of logic. In *European conference on computer vision*. Springer.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2019. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1).

Hexiang Hu, Wei-Lun Chao, and Fei Sha. 2018. Learning answer embeddings for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5428–5436.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6700–6709.

Brad Jascob. v0.2.1 (February 22, 2020). Lemminflect. a python module for english word lemmatization and inflection. https://github.com/bjascob/LemmInflect.

Kushal Kafle and Christopher Kanan. 2016. Answer-type prediction for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4976–4984.

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015.

Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. 2020. Roses are red, violets are blue... but should vqa expect them to? *arXiv preprint arXiv:2006.05121*.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. 2020. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, pages 11336–11344.

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

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

Tom M Mitchell. 1980. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ....

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

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. 2018. Overcoming language priors in visual question answering with adversarial regularization. In *Advances in Neural Information Processing Systems*, pages 1541–1551.

Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. 2019. Sunny and dark outside?! improving answer consistency in vqa through entailed question generation. 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 5863–5868.

Ramprasaath R. Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Ribeiro, Besmira Nushi, and Ece Kamar. 2020. Squinting at vqa models: Interrogating vqa models with sub-questions. In *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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6649–6658.

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

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Hao Tan and Mohit Bansal. 2019. 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.

Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. 2020a. Learning what makes a difference from counterfactual examples and gradient supervision. In *European conference on computer vision*. Springer.

Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. 2020b. Unshuffling data for improved generalization. *arXiv preprint arXiv:2002.11894*.

Damien Teney and Anton van den Hengel. 2016. Zero-shot visual question answering. *arXiv preprint arXiv:1611.05546*.

Benjamin P Willing, Shannon L Russell, and B Brett Finlay. 2011. Shifting the balance: antibiotic effects on host–microbiota mutualism. *Nature Reviews Microbiology*, 9(4):233–243.

Jialin Wu and Raymond J Mooney. 2019. Self-critical reasoning for robust visual question answering. In *NeurIPS*.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514.

Appendix

A Datasets

A.1 VQA-CP

VQA-CP (Visual Question Answering under Changing Priors) (Agrawal et al., 2018a) is a re-organization of the VQA dataset (Antol et al., 2015; Goyal et al., 2017). The aim of VQA-CP is to have a different distribution of answers per question type is different in test and train splits. There are 65 question types based on the prefix of the questions such as *"how many", "what color", "what sport", "is there", "what is the", "which"*. In VQA-v2, samples are drawn at randomly and independently and assigned either to train or test, thus resulting in the same distribution for both splits.

$$P^{VQA}_{train}(A|Q,I) = P^{VQA}_{test}(A|Q,I).$$

In VQA-CP however, samples are assigned using a greedy re-splitting algorithm, either to train or test, in a way that makes sure that questions with the same type an same answer are not shared by train and test. It is important to note that there is no leakage between train and test splits compared to the original VQA splits.

$$P^{VQA-CP}_{train}(A|Q,I) \neq P^{VQA-CP}_{test}(A|Q,I).$$

The train set for VQA-CP-v2 contains 121k images, 245k questions and 2.5M answers, while the test set contains 98k images, 220k questions and 2.2M answers.

A.2 COCO

The source of images in both VQA and VQA-CP is the MS-COCO dataset (Lin et al., 2014). COCO contains natural images representing complex, real-world scenes containing common objects of 91 categories such as *"person", "chair", "fork", "horse", "sports-ball"*, etc. For each image, COCO provides 5 captions along with bounding boxes and polygon annotations for each object instance in the image.

B Image Mutant Generation Process

In this section we provide additional details about our process for generating mutant samples from original question-image-answer triplets (Q-I-A) in the VQA-CP dataset. For all linguistic operations we use a combination of SpaCy (Honnibal and Montani, 2017) and the LemmInflect library (Jascob, v0.2.1 (February 22, 2020) for lemmatization and inflection.

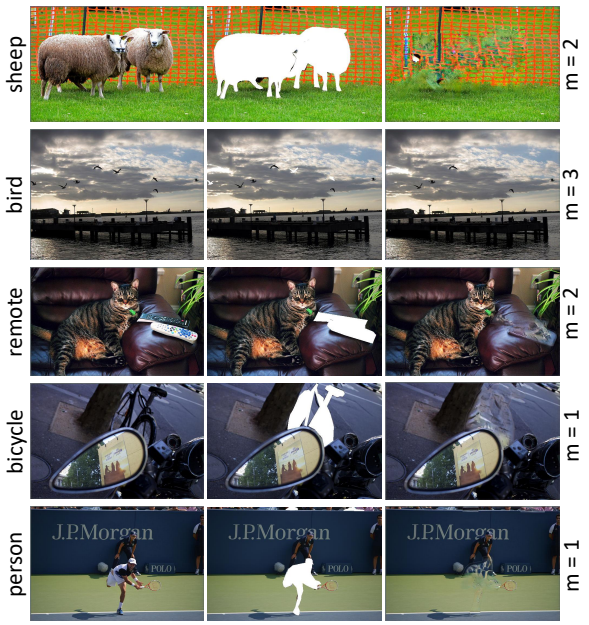

Figure 4: Illustration of COCO bounding box and polygon annotations for m instances of an object, and the inpainting results after removal

B.1 Selection of Objects

For each VQA sample, a list of words W is created, which contains words from the ground-truth answers and the question. All nouns in W are converted to their singular form. For yes-no questions, numeric questions, and questions about colors of objects, a list of objects O is obtained from COCO. Background and crowd objects are filtered out from O. From O critical objects O_C and and non-critical objects O_{NC} are obtained. Critical objects are those objects in the image that when manipulated or removed, may change the answer to the question being asked. For this we follow a simple heuristic that states that if an object-word or it's synonym or hyponym is present in W, then it is a critical object. Then a critical object $o \in O$ is chosen at random, and m instances of this object are chosen at random. The polygon annotations (a polygon border) for this object are obtained from the COCO dataset as shown in Figure 4. Using these annotations, either a removal or color-inversion operation is applied to create the mutant image.

B.2 Object Removal and In-painting

After the object instance is selected, it is removed from the image by replacing all pixel values by 1 (white). This masked image is then input to a GAN-based image inpainting network (Yu et al., 2018) that fills up this pixels in the mask. This makes the

Figure 5: Illustration of color inversion procedure

image photorealistic. This network is one of the best available off-the-shelf blind image inpainting models, and is trained on the ImageNet (Deng et al., 2009). The masked image could also be used as the mutant image however we prefer to use photorealistic images for two main reasons. First, masked images do not lie in the same distribution as natural images, and secondly, the mask boundary may give clues to the network about the the shape or outline of the missing object.

B.3 Color Inversion Process

For mutation that involves a change in the color of the object, we perform a simple pixel-wise color inversion operation on each pixel in the mask to get the mutant image as shown in Figure 5. This is to ensure that we do not use any prior knowledge about valid colors of a specific object. For instance, bananas can typically be yellow, green, or black. However, if we only change the color or a banana to one of these three colors, we would be using domain knowledge and inadvertently introducing answers from the test set, defeating the purpose of OOD generalization. Although the simple inversion process can introduce unnatural colors like blue bananas, it forces the model to understand colors in the image to answer the question instead of simply answering from linguistic priors (such as the memorized knowledge that bananas can be green, yellow, or black).

B.4 Answer Generation

The new answers are generated based on the type of question. For **yes-no questions**, if all instances of the object are removed then the answer changes from yes to no. If only some instances are removed or if the object is non-critical, the answer remains the same. For **number questions**, if m instances of a critical object are removed, the answer changes from n to $n-m$, else the answer remains the same. For **color-based questions** we convert the answer color to their HEX value using Webcolors [2], invert the value, and find the color in CSS-21 colors closest to this value to generate the new answer.

C Question Mutant Generation Process

For generating question mutants, we use three operators: negation, substitution by antonyms or adversarial words, and masking critical words.

C.1 Negation

For yes-no questions and color-based questions, we use a template-based negation technique that puts a negative word such as *"not"* or *"no"* before a preposition, noun phrase, or verb. For instance "Is this chair broken?" is negated to "Is this chair not broken?". We show examples of negation in Table 7. Negation simply flips the answer from yes to no or no to yes.

C.2 Adversarial Words and Masking

Another form of question mutation is substituting object-words with their adversarial words. To do so, we create a list of all object words and their synonyms and use BERT (Devlin et al., 2019) similarity to rank the most similar words. To replace a word, we chooser the most similar word which is not present in the image. The third type of mutation is masking, where a critical object word is removed from the question and replaced with the token "MASK".

For both these types of mutations, determining the correct answer in some cases is not possible as can be seen from examples in Table 7. Thus we use the broad category as the answer. For instance, when a question such as *"How big is the book"* is replaced with either *"How big is the plane"* or *"How big is the [MASK]"*, it is clear that the question is about the size of an object. Thus we annotate this question with this broad category "size" as the answer. In other cases, where even a broad category cannot be ascertained, the answer is replaced with "can't say" or "don't know".

[2] https://pypi.org/project/webcolors/

Mutation	Q	A	Q_{mutant}	A_{mutant}
Negation	Is this bread?	yes	Is this not bread	no
	What is the color of the woman's shirt?	black	What is not the color of the woman's shirt?	white
	Are there deciduous trees?	no	Are there no deciduous trees?	yes
	Is there a boy?	no	Is the no boy?	yes
Adversarial	Who is riding the boat?	man	Who is riding the desk	*"can't say"*
	How big is the plane?	large	How big is the book?	*"size"*
	How many pillows are on the bed?	four	How many pillows are on the table?	*"number"*
Masking	What type of drink is being displayed?	wine	What type of [MASK] is being displayed?	*"beverage"*
	How many bins?	two	How many [MASK] ?	*"number"*
	What is the green stuff on the sandwich?	lettuce	What is the green stuff on the [MASK]?	*"food"*

Table 7: Examples of three types of question mutation with new answers

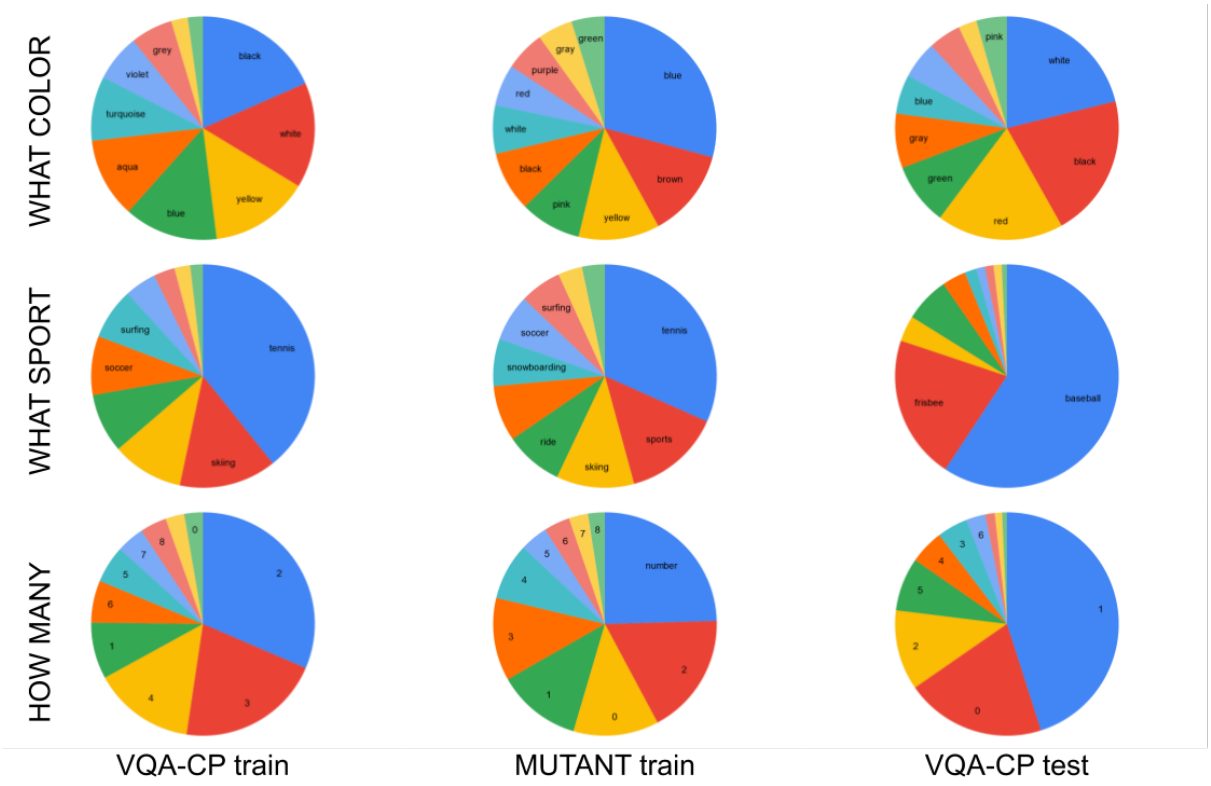

Figure 6: The distribution of answers by question types for VQA train and Mutant compared with VQA-test

To generate answer clusters and representative answer categories, we extract Glove (Pennington et al., 2014) word vectors for each answer phrase/word using Spacy. We use k-means clustering (Lloyd, 1982) with Euclidean distance metric and with varying number of K. We manually tune the number of clusters till we observe a clear set of categories appear at $K = 50$. We then manually annotate the category names.

D Dataset Analysis

Here we provide dataset analysis in terms of distribution of answers by question-type, number of samples for each type of mutation, and the final distribution of the dataset in terms of answer-type.

Category	VQA-CP (%)	Mutant (%)
Yes/No	41.86	47.88
Number	11.91	13.64
Other	46.23	38.48

Table 9: Distribution of samples in the dataset by answer type

D.1 Distribution by Question Type

We show the distribution of answers per question type in Figure 6 for three categories *"How many"*, *"What sport", and "What color"* for the top-10 answers. It can be seen that the distribution is distinct

Mutation Category	Number of Samples
Object Removal	159,899
Color Change	30,759
Negation	237,611
Adversarial Substitution	146,814
Word Masking	104,666

Table 10: Distribution of generated mutant samples by category of mutation

from the test data and close to the VQA-CP train data apart from the introduction of categorical answers such as *"number" and "sports"* during question mutation. Our mutation method does not leak information about answers from test set to train set.

D.2 Distribution by Mutation Type

Table 10 shows the number of samples generated by each type of mutation.

D.3 Distribution by Answer Type

There are three answer types in both VQA-CP and Mutant datasets: *yes/no, number, and other*. Creation of mutant samples leads to a small change in the distribution as shown in Table 9.