Improving Robustness of VQA Models by Adversarial and Mixup Augmentation

Jurijs Nazarovs, Xujun Peng, Govind Thattai, Anoop Kumar, Aram Galstyan

Amazon Alexa AI

Abstract

Recent multimodal models such as VilBERT and UNITER have shown impressive performance on vision-language tasks such as Visual Question Answering (VQA), Visual Referring expressions, and others. However, those models are still not very robust to subtle variations in textual and/or visual input. To improve model robustness to linguistic variations, here we propose a novel adversarial objective function that incorporates information about the distribution of possible linguistic variations. And to improve model robustness to image manipulation, we propose a new VQA-specific mixup technique which leverages object replacement. We conduct extensive experiments on benchmark datasets and demonstrate the effectiveness of the proposed mitigation methods in improving model robustness.

Introduction

Recent development in vision-and-language (V+L) and multimodal research have shown great performance in a wide variety of V+L tasks (Zhou et al. 2020; Hu et al. 2020; Murahari et al. 2020; Wang et al. 2020; Alberti et al. 2019; Hao et al. 2020; Majumdar et al. 2020; Cao et al. 2020), such as Visual Commonsense Reasoning (VCR) (Zellers et al. 2019), Referring Expression Comprehension (Yu et al. 2016), and Visual Question Answering (VQA) (Antol et al. 2015), in which we are interested in particularly.

The main goal of VQA tasks is to answer an open-end natural language questions based on an observed image. Given a rising integration of vision-capturing devices in our daily life, including assistive devices, security cameras, medical diagnosis machines, robots and etc, VQA has a broad range of applications, which are centered around a human user asking a machine questions about images. Since VQA systems can also be incorporated in mission-critical applications, like a security cameras, it vital for a VQA system to work reliably, and have to be aware of different variations of questions asked by a user. For example, given the same image and three semantically similar questions: “How many cookies”, “How many cookies are there?”, “How many cookies are on the plate?”, VQA model should produce the same answer. In other words, VQA system has to be robust to **linguistic variations**. In addition to linguistic variations in questions, it is desirable that VQA system show good performance in generalization to different items surrounding an object of question. For example, given a question “How many cookies are on the plate?”, model should provide a right answer regarding if there are bananas, soup, broccoli or other objects around. We refer to this source of robustness as **visual manipulation**.

Despite the rapid progress in recent years in VQA systems, current methods are still far from being perfectly robust to valid linguistic variations and general visual manipulations, even given the same language. Problem is getting worse, when VQA systems is targeting to cover several languages. One of the way to improve robustness of the VQA systems (and VLM in general) to linguistic variation or visual manipulation is to create a dataset deliberately diversified with respect to the source of the robustness, with an idea to train a more generalised model. For example (Shah et al. 2019; Kant et al. 2021) introduced new datasets targeting linguistic variation of questions, by introducing questions rephrased by a human and using back-translation, respectively. Authors in (Iyyer et al. 2018) focus on manually creating an adversarial dataset by human, which can trick the VQA system. From the visual manipulation content, (Agarwal, Shetty, and Fritz 2020) introduce a new VQA dataset, generated by semantic manipulations based on in-painting GAN(Shetty, Fritz, and Schiele 2018). While generating new datasets targeting specific sources of variations/manipulations can be useful for evaluation of the robustness of VQA systems towards this source, it is less efficient for training to improve the robustness of the model. While ideally data should contain an interaction with all possible combinations of questions, natural images and answers, to generate such datasets is infeasible.

In addition to generating new datasets targeting the specific source of data variation, another direction is to focus on training methods, by introducing regularization components in the loss or new architectures (Shi et al. 2020; Gan et al. 2020). The limitation of these systems, is that the training procedure does not explicitly utilize the information about linguistic variation or visual manipulation in their training.

To overcome these limitations, we introduce two novel components during training. First, to improve robustness of VQA system to linguistic variations, we propose to use an adversarial perturbations, similar to (Gan et al. 2020), but with **(a)** explicit incorporation of possible linguistic variations to the noise generation and **(b)** regularization terms in the loss, which preserve the 'validity' of adversarial perturbations, forcing them to be similar to observed linguistic variations. Second, to improve general robustness to visual manipulations, we introduce a novel mixup image replacement technique, which is based on substituting objects between 'positive' images, i.e. images which have the questions with similar semantic meaning. In addition, we introduce the new metric, the consensus score area under curve (CS-AUC), to evaluate the robustness of VQA systems.

We evaluate our model on the VQA Rephrasings benchmark (Shah et al. 2019), which measures the model's answer consistency across several rephrasings of a question and we show that our method outperforms current SOTA in metrics measuring robustness of the model to linguistic variations and a general VQA score. We extensively ablate with different choices of explicit incorporation of linguistic variations in our model and different method for image mixup replacement.

Related work

VQA datasets to target a source of variation: To address different types of robustness one of the approaches is to generate datasets, either synthetically using machine/deep learning tools or through crowdsourcing, which explicitly include the desired variation. For example, motivating that bias in our language tend to be a simpler signal for learning than visual modalities, and to improve a *generalization* of the VQA system, authors in (Goyal et al. 2017) propose a new dataset, VQA V2.0. It balances the popular VQA dataset (Antol et al. 2015) by collecting complementary images such that every question in the balanced dataset is associated with not just a single image, but rather a pair of similar images that result in two different answers to the question. To adjust for *shift in distributions* of answers per question type (e.g., "what color", "how many") between the train and test sets and address over-fitting, (Agrawal et al. 2018) propose a new dataset VQACP (for both versions VQA V1 and V2). It is a reshuffling of the original VQA (V1, V2) dataset, such that the distribution of answers per question type (e.g., "what color", "how many") differs between the train and test sets. Authors in (Li et al. 2021; Rosenberg et al. 2021) study on the vulnerability of VQA models when under *adversarial attacks by human*. To measure the robustness of SOTA VLM, they proposed new datasets, where validation and testing sets consist only of examples, which were successful in attacking the SOTA. To incorporate *logical reasoning* in data sets, several papers introduced new datasets. For example, authors in (Gokhale et al. 2020) construct an augmentation of the VQA dataset as a benchmark, with questions containing logical compositions and linguistic transformations (negation, disjunction, conjunction, and antonyms). Similarly, new datasets were introduced in (Selvaraju et al. 2020; Hudson and Manning 2019). To highlight the *linguistic variation*, authors in (Shah et al. 2019) introduced VQA-Rephrases, which is based on VQA V2.0 validation and provide 3 rephrasing by humans for 40k questions on 40k images. From the *visual manipulation* content, (Agarwal, Shetty, and Fritz 2020) introduce a new VQA dataset, generated by semantic manipulations based on inpainting GAN(Shetty, Fritz, and Schiele 2018).

Robustness to linguistic variations: There are several ways to augment the language input. Particularly, a common way is to paraphrase the questions in a way, to preserve semantic. For example, authors in (Shah et al. 2019) propose to generate augmented questions, by training a new questions-generator as a separate model block in a cycle-consistent way, similar to cycle-GAN(Zhu et al. 2017). This allows to generate diverse, semantically similar variations of questions conditioned on the answer, which are used to improve robustness of VQA system. Although not for VQA system, but for language model, similarly, authors in (Iyyer et al. 2018) proposes syntactically controlled paraphrase networks (SCPNs) and use them to generate adversarial examples. Given a sentence and a target syntactic form (e.g., a constituency parse), SCPNs are trained to produce a paraphrase of the sentence with the desired syntax. Another example of improving robustness of the model through training is (Kant et al. 2021), where authors proposed a novel training paradigm (ConClaT) that optimizes both cross-entropy and contrastive losses, which encourages representations to be robust to linguistic variations in questions while the cross-entropy loss preserves the discriminative power of representations for answer prediction.

Robustness to visual manipulations: One of the examples to improve robustness of the model to visual manipulations is a mixup, proposed by authors in (Zhang et al. 2017). Mixup was a new augmentation scheme for image classification tasks, which is presented as a linear combination of input and one-hot representation of outputs (labels).

Adversarial perturbations: Villa Adversarial perturbations literature in general is focusing on finding an adversarial attack to break the model. For example, authors in (Xu et al. 2018) show how for to learn an adversarial image in VQA settings, such that for the same questions leads to a different answer. Authors in (Gan et al. 2020) went further and instead of learning an adversarial attack on a VQA systems, they learned an adversarial perturbations of the latent space, to improve robustness of the model.

Background & Notation

Data processing. To utilize Neural Network for VQA system, it is common to preprocess input data (Image, Question, Answer) to obtain an appropriate format for the model. Given an *image*, we apply pre-trained maskrcnn to derive bounded boxes Figure 1 of objects and use corresponding features of those boxes (p-dimensional vector) as an input in our model. Given *question* in a text form, the common strategy is to tokenize them, i.e. each word is represented by a number. To perform a tokenization, we use pre-trained (bert-base-uncased) BertTokenizer. Since questions vary in their length, it is important to get the same length for all questions. Thus, given a fixed length, we either cut the sentence or append 0 to make it a proper length. To understand where the starting and end of sentence, special tokens are attached to the beginning (token 101) and end (token 102).

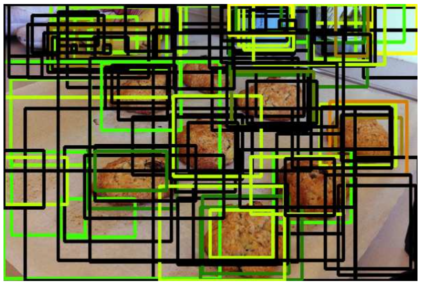


Figure 1: Example of object segmentation as a preprocessing step to create an input for VQA system

In VQA settings, *answer* is usually correspond to a single word, e.g. 'Yes', 'No', 'Salad', '5' and etc. Given the whole set of answers (in VQA-V2 is 3129 classes), answers are represented by the 'VQA-score' (mentioned above) in k of the 3129 positions, as a 'k-hot' representation.

Notation. For the rest of the text we use following notation: (Image, Question, Answer) is (v, q, y); $q_i \in Q$ is an element of a set of anchor questions (not rephrases) Q; $\{q_i^j\}_{j=0}^{M_i}$ is a set of similar questions to a question q_i, where $q_i^0 = q_i$ and M_i is a number of similar questions, which varies depending on definition of 'similarity' and question q_i (more on this in method section); z_q is a latent representation of a question q. We refer to VQA system with parameters θ as f_θ, and adversarial perturbation generator with parameters ϕ_q as g_{ϕ_q}.

Adversarial perturbation. One of the way to improve robustness of VQA system is to learn an adversarial perturbation, which is added to the latent representation. Intuitively, we can think of learning an adversarial perturbations as learning the 'distance' to the decision boundary. From one perspective, we want to learn a noise ('distance') big enough that VQA model fails in its prediction, from another we want the VQA model to predict adversarial sample as a correct one. Following (Gan et al. 2020), the optimization objective is represented as a mini-max problem:

$$\begin{aligned}\min_{\boldsymbol{\theta}} \max_{\phi_q} \underset{(v,q,y)\sim D}{E} \underset{\alpha\in N(0,1)}{E} \Big\{ & L_{\mathrm{BCE}}\left(f_\theta(\boldsymbol{v},\boldsymbol{q}),\boldsymbol{y}\right) \\ & + L_{\mathrm{BCE}}\left(f_\theta\left(\boldsymbol{v},\boldsymbol{q}+g_{\phi_q}(\boldsymbol{\alpha})\right),\boldsymbol{y}\right) \\ & + KL\left(f_\theta\left(\boldsymbol{v},\boldsymbol{q}+g_{\phi_q}(\boldsymbol{\alpha})\right),f_\theta(\boldsymbol{v},\boldsymbol{q})\right)\Big\}\end{aligned} \tag{1}$$

which has two objectives:

1. Given a generated noise $g_{\phi_q}(\alpha)$, where $\alpha \sim N(0,1)$, VQA model tries to minimize the prediction error of both original and perturbed data.
2. Given a VQA system f_θ, the noise generator model tries to create a noise, to maximize the error of 1) perturbed data classification, 2) KL divergence between predictions with perturbed and original data.

Note that we abuse notation and use $\boldsymbol{q} + g_{\phi_q}(\boldsymbol{\alpha})$ to refer that adversarial pertubation is added to latent representation of a question q.

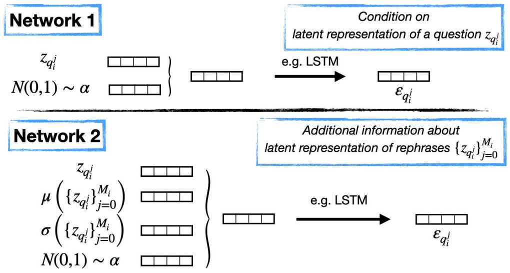

Network 1
Condition on
latent representation of a question $z_{q_i^j}$
$z_{q_i^j}$
$N(0,1) \sim \alpha$
e.g. LSTM
$\varepsilon_{q_i^j}$
Network 2
Additional information about
latent representation of rephrases $\{z_{q_i^j}\}_{j=0}^{M_i}$
$z_{q_i^j}$
$\mu\left(\{z_{q_i^j}\}_{j=0}^{M_i}\right)$
$\sigma\left(\{z_{q_i^j}\}_{j=0}^{M_i}\right)$
$N(0,1) \sim \alpha$
e.g. LSTM
$\varepsilon_{q_i^j}$

Figure 2: Adversarial perturbation generation networks. They utilize information about latent representation z_q of linguistic variations $\{q_i^j\}_{j=0}^{M_i}$ and generate adversarial perturbation $\varepsilon_{q_i}^j$ for a question q_i^j.

Mixup. Another known type of augmentation is 'mixup', originally introduced in (Zhang et al. 2017) as a way to improve robustness of image classification models in supervised settings. The primary idea of the original paper is, instead of considering the tuple of (image, class), we consider a linear combination of two tuples as image $= \lambda$ image$_A$ + $(1-\lambda)$image$_B$ and class $= \lambda$ class$_A + (1-\lambda)$class$_B$, where class is one/k-hot representation, and λ is a mixup ratio. Similar settings were explored in (Chen, Yang, and Yang 2020) but for language models, and thus instead of images the mixup of language components were used.

Method

While we propose two separate methods how to improve robustness of VQA system to both linguistic variation and visual manipulations, we start with the description of method to deal with linguistic variation first.

Adversarial perturbations to improve robustness to linguistic variations.

While the strategy of augmenting latent representation with adversarial perturbation in (1) technically sound, we realized that in current formulation there are ways of possible improvement: **(a)** current noise generation network g only depends on alpha, and can generate the same noise despite the linguistic variation, **(b)** loss does not incorporate any information regarding a linguistic variation explicitly. To address aforementioned issues we propose the following: **First**, since different questions have their own linguistic variations, we condition noise generation network on additional information provided by question (more on this below). **Second**, to incorporate a linguistic variation explicitly we introduce an additional regulation term on generated noise, KL_{lv}, which forces to generate adversarial perturbation, such that resulted augmented latent representation is not far from latent representation of elements from a set of linguistic variations of q_i^j:

$$KL_{\mathrm{lv}} = \sum_{a\in O} KL\left(\left\{q_i^j + g_{\phi_q}\left(\boldsymbol{\alpha}\right), q_i^j\right\}_{j_{j=0}}^{M_i}, \left\{q_i^j\right\}_{j=0}^{M_i}\right) \tag{2}$$

With this contributions, we incorporate new regulation KL term from Eq.2 and obtain the loss in following equation.

Figure 3: Example of different images of the Similar Batch, corresponding to the question 'How many cookies can be seen?', which correspond to different questions and answers, but with similar semantic meaning.

$$
\begin{aligned}
\min_{\boldsymbol{\theta}} \max_{\phi_q} \underset{(v,q,y)\sim D}{E} \underset{\alpha\in N(0,1)}{E} \{ & L_{\mathrm{BCE}}\left(f_\theta(\boldsymbol{v},\boldsymbol{q}),\boldsymbol{y}\right) \\
& + L_{\mathrm{BCE}}\left(f_\theta\left(\boldsymbol{v},\boldsymbol{q}+g_{\phi_q}(\boldsymbol{\alpha},q)\right),\boldsymbol{y}\right) \\
& + KL\left(f_\theta\left(\boldsymbol{v},\boldsymbol{q}+g_{\phi_q}(\boldsymbol{\alpha},q)\right), f_\theta(\boldsymbol{v},\boldsymbol{q})\right) - KL_{\mathrm{lv}} \}
\end{aligned}
$$

For our condition noise generation network, we explore several directions. As we mentioned our motivation is that adversarial noise should be generated based on information about the linguistic variation. Thus, we considered two types of networks (visualization is presented in Figure 2):

1. Network 1: noise is generated conditionally on a latent representation of a question only
2. Network 2: noise is generated conditionally on a latent representation of a question and summary statistics (sample mean and sample variance) of latent representation of either a) rephrase of a question or b) similar questions.

Since our proposition, noise generation network conditioned on distribution of linguistic variations and KL of linguistic variations, KL_{lv}, requires an access to linguistic variations of questions, next we introduce the way to construct these linguistic variations.

Generating linguistic variations.

To generate linguistic variations of questions for training, we consider two different methods: Rephrases and Similar questions. We refer to them as Rephrase Batch and Similar Batch respectively. While both methods provide a linguistic variation, preserving semantical meaning, they do it in a different degree, which we discuss below.

Rephrase Batch: Following (Shi et al. 2020), our rephrase batch is generated by augmenting the train set with question paraphrases using 88 different MarianNMT (Junczys-Dowmunt et al. 2018) back-translation model pairs released by HuggingFace (Wolf et al. 2019). To construct the data, three unique paraphrases, which have ≥ 0.95 similarity with the original question, were randomly selected. The similarity is defined as a cosine similarity Sentence-BERT (Reimers and Gurevych 2019) encoding of questions. **Note:** that for rephrased questions correspond to the same image and answer. That is, for the triplet (I, Q, A), and rephrases of Q, Q', we still have the same I and A.

Rephrase Batch	Similar Batch
How many cookies can it be seen?	How many cookies?
How many cookies can you see?	What types of cookies are in the package?
How many cookies can we see?	How many types of cookies are there?

Table 1: Examples of questions, to introduce linguistic variation in the data, constructed using Rephrase and Similar methods.

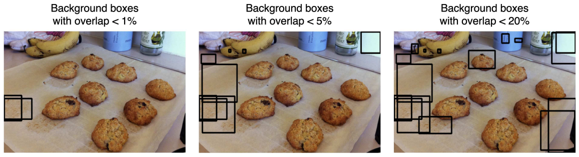

Background boxes with overlap < 1%
Background boxes with overlap < 5%
Background boxes with overlap < 20%

Figure 4: Example of selecting background boxes with different level of overlapping area with objects related to question.

Similar Batch: For the similar batch, instead of generating new questions, we looking through the training set, to find similar questions. Namely, given an anchor question, we find questions ≥ 0.95 similarity with the anchor question. Similarity score is computed same way as in rephrase batch. Because number of similar questions vary per anchor question, in our experiments we set an upper bound on how many similar questions to use. **Note:** in contrast to rephrased questions, similar questions correspond to different images and answers, since answers depends on images. In the future we refer to such images as *'similar' images*, since they correspond to similar questions. Examples of images are provided in Figure 3

Difference between Rephrase and Similar batches: While two methods generate sentence with similar semantically meaning and with linguistic variation, they are still different. Table 1 provide examples to the same original question "How many cookies can be seen?". Rephrased questions have less linguistic variations but better semantic preservation, while similar questions provide more linguistic variations, but might drift away in semantic meaning.

Mixup replacement to improve robustness to visual manipulations.

As mentioned in (Goyal et al. 2017), our language tend to carry a simpler signal for learning than visual modalities, which makes it easier for VQA system to learn to focus more on questions and pay less attention to visual components. In addition, authors in (Agarwal, Shetty, and Fritz 2020) showed that current VQA systems are brittle to semantic variations in the image, revealing the false correlation that the models exploit to predict the answer. To address this issue, they proposed for each image in the training set, generate a single copy, by removing objects not related to the question. This was implemented with in-painting GAN(Shetty, Fritz, and Schiele 2018). Motivated by this idea and mixup for image classification problem, we proposed a novel augmentation technique, *mixup replacement*. While using in-painting GAN provide images of a good quality, with objects removed, this leads to extra computational overload. in addition, for each new image, we have to re-run segmentation model to define bounded boxes for input to the network.

In contrast to replacing objects which are not related to questions, we propose to replace features of extracted bounded boxes, which are classified as background objects, between semantic similar images. The motivation behind selecting similar images is that background objects between such images have positive correlation. Thus, replacing these objects does not destroy the conceptual meaning of the image, but provides variation for model to learn to focus on object in the question. We hypothesis that if robust model knows how to handle unrelated to the question objects, then model should know how to handle the same objects in a conceptually similar images. Forcing model to see a lot of different objects unrelated to the question, should make it to pay more attention to related objects.

From implementation perspective, given a batch of similar images, we use pre-trained Faster R-CNN to extract bounded boxes, select those which are classified as background and which which have overlap area $\leq p\%$ with objects related to question, where p is a hyper-parameter, Figure 4. To find semantic similar images, we use 'Similar Batch', mentioned before, to find similar questions, and use their corresponding images to replace objects. Given a Batch 1 of related images, i.e. images with similar questions, we create a permuted version, Batch 1'. We use corresponding images (in order) from Batch 1' to replace objects in Batch 1. The implementation of the algorithm is visualized in Figure 5.

A more trivial version of mixup replacement is to replace random boxes between similar images, as demonstrate in Figure 6. While it provides a lot of variation to the image, it might also replace features of objects, related to the question.

Note that our method is different from a general mixup used as robustness tool in image classification tasks, since we do not consider a linear combination on pixel-wise level, but replacement of patches of images, namely latent features of bounded boxes.

Experiments

In our experiments we seek to demonstrate the improvement of our proposed approach in robustness of VQA systems to linguistic variations and visual manipulations. We describe used datasets and metrics below.

Datasets

In our experiments we utilize the VQA v2.0 (Goyal et al. 2016), VQA-Rephrasings (Shah et al. 2019). *VQA-V2* is a set of COCO natural images, with corresponding open-end questions and 10 answers per question (obtained from 10 different people). Since each question contains 10 answers (not necessary the same), instead of selecting one answer, data provides a score for each answer. Score is computed as $\min(\frac{\text{\# of humans provided this answer}}{3}, 1)$. The intuition is that if a model predicts as good as 3 humans, then we should get maximum score of 1. VQA-V2 contains nearly 443K train, 214K val and 453K test instances. *VQA-Rephrasings* consists of 3 human rephrasings for $\sim$40k questions on $\sim$40k images from the VQA V2 validation dataset, resulting in a total of $\sim$120k questions rephrasing by humans. The dataset was designed to evaluate the robustness of VQA models towards human rephrased questions. In addition to these datasets, we utilize two methods to generate linguistic variations, which are used for training, Rephrase Batch and Similar Batch. These are based on VQA-Rephrasings by Back-translation (Shi et al. 2020). Detailed description is provided in method section.

Metrics

To measure the performance of our model, we refer to two metrics: general VQA-score and Consensus Score (k).

VQA-score. The score is similar to a standard classification accuracy, i.e. it measures how correct the model's prediction is. However, it accounts for the specification of a VQA setup. In VQA-V2 dataset (Goyal et al. 2017) (and other datasets based on it) each question has 10 answers. Instead of selecting the best answer according to a rule, like the most frequent answer, authors proposed to report answer with the score: $\min(\frac{\text{\# of humans provided this answer}}{3}, 1)$. And thus, answers are reported as a k-hot representation, with assigned scores in rage [0,1]. Then when model predicts a class (single class), instead of selecting this class, we select the score, corresponding to it. Which is later averaged across number of samples. As you can see, if data set contain only 1 answer per question with score 1, then VQA-Score would correspond to a standard accuracy.

Consensus Score, CS(k). Intuitively, for a VQA model to be consistent across various rephrasings of the same question, the answer to all rephrasings should be the same. It is measured by a Consensus Score $CS(k)$, (Shah et al. 2019). For every group Q consisting of n rephrasings, we sample all subsets of size k. The consensus score $CS(k)$ is defined as the ratio of the number of subsets where all the answers are correct and the total number of subsets of size k. The answer to a question is considered correct if it has a non-zero VQA accuracy θ as defined in (Agrawal, Batra, and Parikh 2016). $CS(k)$ is formally defined as:

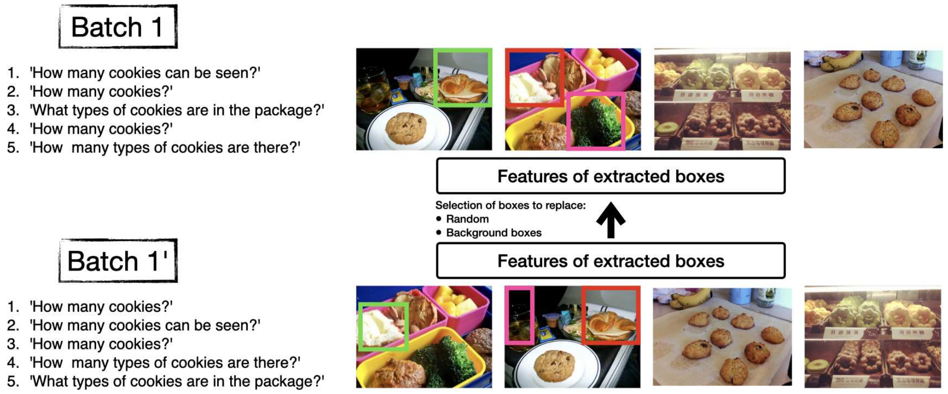
$$CS(k) = \sum_{Q' \subset Q, |Q'|=k} \frac{\mathcal{S}\left(Q'\right)}{{}^{n}C_{k}}, \tag{3}$$

where

$$\mathcal{S}\left(Q'\right) = \begin{cases} 1 & \text{if } \forall q \in Q' \quad \theta(q) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

and ${}^{n}C_{k}$ is number of subsets of size k sampled from a set of size n. As consensus score is a all-or-nothing score, to achieve a non-zero consensus score at k for a group of questions Q, the model has to answer at least k questions correctly in a group of questions Q. When $k = |Q|$ (e.g. when $k = 4$ in VQA-Rephrasings), the model needs to answer all rephrasings of a question and the original question correctly in order to get a non-zero consensus score. It is evident that a model with higher average consensus score at high values of k is quantitatively more robust to linguistic variations in questions than a model with a lower score.

Consensus Score - Area Under the Curve (CS-AUC). Given a set of measures of consensus scores for a different k, e.g. $CS(1), \ldots, CS(4)$, the comparison between different experiments might be confusing. For example $CS(1)$



1. 'How many cookies can be seen?'
2. 'How many cookies?'
3. 'What types of cookies are in the package?'
4. 'How many cookies?'
5. 'How many types of cookies are there?'

Features of extracted boxes
Selection of boxes to replace:
• Random
• Background boxes
Features of extracted boxes
Batch 1'

1. 'How many cookies?'
2. 'How many cookies can be seen?'
3. 'How many cookies?'
4. 'How many types of cookies are there?'
5. 'What types of cookies are in the package?'

Figure 5: Implementation of mixup replacement technique to improve robustness of VQA to visual manipulations. Batch 1 is generated as 'Similar Batch', mentioned before. That is, all questions are similar in semantic meaning, but correspond to different images and answers. Batch 1' is a shuffled version of Batch 1. By replacing objects between similar images, we decrease the probability of introducing out-of-distribution patches.

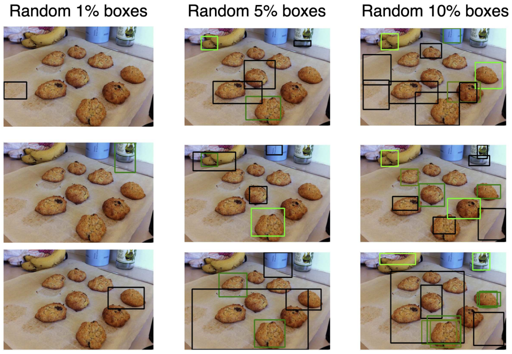


Figure 6: Example of randomly selected boxes, where each column correspond to percentage of selected boxes, and each row is a new sample. Black boxes correspond to background objects, while colored boxes to non-background boxes.

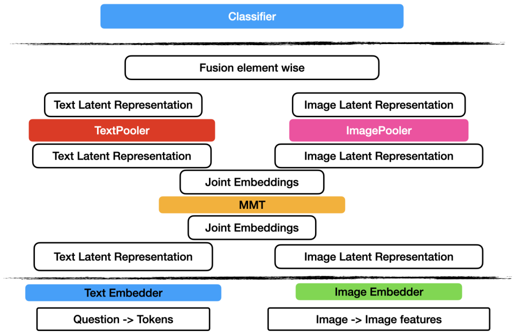

Classifier
Fusion element wise
Text Latent Representation
Image Latent Representation
TextPooler
ImagePooler
Text Latent Representation
Image Latent Representation
Joint Embeddings
MMT
Joint Embeddings
Text Latent Representation
Image Latent Representation
Text Embedder
Image Embedder
Question -> Tokens
Image -> Image features

Figure 7: Uniter-like architecture used in our experiments. It is split in 3 main blocks: First block utilizes pre-trained models to tokenize questions and extracted bounded boxes from images; Second block extract latent representation of input features and perform multimodal transformation; Third block uses extracted and aligned textual and visual features to provide an answer for VQA system. We add adversarial perubations to input of the second block.

of model A can be higher, then $CS(1)$ of model B, but it might be reverse for $CS(4)$. Thus, motivated by AUC of ROC, we propose to compute a Consensus Score Area Under the Curve (CS-AUC). Following the trapezoid rule of numerical integration, we compute CS-AUC as $\frac{1}{2}CS(1) + CS(2) + \cdots + CS(k-1) + \frac{1}{2}CS(k)$.

Model, hyper-parameter settings and hardware

Main VQA Model Following (Kant et al. 2021) we use a multimodal transformer (MMT) as a main model f in our VQA system. MMT is currently a representative of SoTA models (Jiang et al. 2020; Lu et al. 2019; Chen et al. 2020; Li et al. 2020; Fukui et al. 2016) in VQA that rely heavily on multi-modal transformer architecture. The used model was originally inspired by (Chen et al. 2020), with 6 layers and 768-dim latent representation of inputs (embedding). Being a Vision Language Model in its core, our VQA system takes as input two different modalities: Language (Question) and Visual (Image). The question tokens are encoded using a pre-trained three layer BERT (Devlin et al. 2019) encoder which is fine-tuned along with the multimodal transformer. Images are processed by detecting object regions and extracting features from a frozen ResNeXT-152 (Xie et al. 2017) based Faster R-CNN model (Ren et al. 2015).

However, notice that our method of improving the robustness of VQA relies on adversarial pertubation of latent space and mixup replacement, which are not strictly limited to the architecture of MMT selected for these experiments.

Adversarial Perturbations Generative Model Recall that input to our VQA system is multi-model, namely a question and an image. After the pre-processing steps, questions are tokenize and still preserve sequential information, while each image is represented by n bounded boxes, which in general do not contain any sequential information and can be shuffled in any order. Given that we focus on generating adversarial perturbations to improve robustness to linguistic variations, our generation network, applied to questions components, is conditioned on question itself. And

Method	VQA-Score (%)	VQA-Rephrases (%)					VQA-Rephrases BT (%)				
		CS(1)	CS(2)	CS(3)	CS(4)	CS-AUC	CS(1)	CS(2)	CS(3)	CS(4)	CS-AUC
Contrast and classify (Trained by us)	65.10	66.66	59.41	55.08	52.02	173.83	71.74	70.55	69.92	69.47	211.44
1. AP to text, no KL_{lv}	**65.97**	66.91	59.68	55.36	52.32	174.12	**72.47**	**71.43**	**70.88**	**70.50**	**213.62**
2. AP text (Network 1), and KL_{rephrase}	65.24	**67.24**	**60.45**	**56.29**	53.31	**177.00**	72.12	70.98	70.36	69.91	212.36
3. AP text (Network 2), and KL_{rephrase}	65.23	67.21	60.42	**56.29**	**53.33**	176.58	72.13	70.95	70.31	69.86	212.23
4. AP text (Network 1), KL_{similar}	65.41	66.40	58.96	54.54	51.44	172.55	72.08	70.85	70.19	69.72	211.89
5. AP text (Network 2), KL_{similar}	65.30	66.66	59.38	55.03	51.97	173.52	71.98	70.64	69.95	69.46	210.73
6. AP text (Network 2), KL_{similar}, mixup text	64.41	66.25	59.69	55.72	52.88	175.07	70.82	69.72	69.13	68.70	209.20
7. AP text (Network 2), KL_{similar}, mixup both	65.17	66.43	58.72	54.10	50.84	171.35	72.00	70.39	69.53	68.92	210.06
8. AP text (Network 2), KL_{similar}, mixup replacement random object	64.93	66.21	58.99	54.72	51.73	171.72	71.36	69.98	69.23	68.71	208.26
9. AP text (Network 2), KL_{similar}, mixup replacement overlap	62.76	64.28	56.81	52.44	49.40	165.77	69.39	67.81	66.99	66.44	202.15
10. AP text (Network 2), KL_{similar}, mixup text, mixup replacement overlap	61.88	63.63	57.06	53.14	50.43	135.42	68.25	67.08	66.46	66.02	166.55

Table 2: Notation: (a) 'AP' is Adversarial Perturbation, (b) 'Network 1' and 'Network 2' corresponds to Adversarial Perturbation Network conditioned on question only and question with summaries about linguistic variations correspondingly, Figure 2, (c) KL_{lv}, KL_{rephrase}, and KL_{similar} correspond to KL with Linguistic variations from any source, from rephrased question and from similar questions correspondingly. (d) mixup replacement corresponds to our proposed technique to replace features of bounded boxes of images, either random boxes or background boxes based on overlap.

thus, should preserve sequential information. While there are several sequential networks, e.g. Neural ODE (Chen et al. 2018), Dilated CNN (Yang et al. 2017), and etc, we chose a simple LSTM model (Hochreiter and Schmidhuber 1996). While we do not focus on adding adversarial noise to latent representation of images, it can be done with Fully Connected networks and reshaping tensors, since they do not contain order information.

The model is trained with AdamX optimizer with initial learning rate of 1e-4 and a learning rate scheduler. Such that learning rate is decayed by 0.7 at 5k and 15k iterations. We train our model for 10 epochs with batch size of 256, using 8 Tesla V100 GPUs to split the batch. The code was implemented in PyTorch.

Baselines

Recall that in our experiments we seek to demonstrate the improvement of our proposed approach in robustness of VQA systems to linguistic variations and visual manipulations. Namely, we would like to evaluate through ablation study the effect of several components: **(a)** type of condition used in generating adversarial perturbations for text; **(b)** inclusion of KL_{lv} term to regulate the generated adversarial perturbations; **(c)** significance of using different types of linguistic variation for training, namely 'Rephrase Batch' and 'Similar Batch'; **(d)** Not sure if we need it traditional mixup for text and images; **(e)** our newly proposed method mixup replacement for objects in images. In addition, our main baseline is current SOTA for robustness to linguistic variations (Shi et al. 2020), which utilizes the alternative training mixing two types of losses, contrastive and typical cross-entropy loss.

Results

We provide results of our experiments in the Table 2 and in the following text we refer to the model from the table in braces: e.g. (1-5). First, we compare adversarial perturbation (AP) to text and vary the inclusion of KL_{lv} term and conditional network (1-5). We see that including KL_{rephrase} term provides a significant improvement in consensus score for VQA-Rephrase. There is no significant difference between Network 1 and Network 2 (2-3), given inclusion of KL_{rephrase}. Given an inclusion of KL_{similar} and usage of Network 2, we compare different mixup strategies (5-10). We see that adding text mixup technique (6) provides the highest improvement in consensus score compare to other mixup strategies; however, applying mixup to both image and text, provides a higher improvement in VQA-score. Comparing mixup replacement techniques (8-10), we see that the biggest improvement in VQA-Score and CS corresponds to mixup replacement of random objects.

Conclusion

In this paper we introduce two novel training components to improve robustness of VQA systems. First, to improve robustness of VQA system to linguistic variations, we propose to use an adversarial perturbations, similar to (Gan et al. 2020), but with **(a)** explicit incorporation of possible linguistic variations to the noise generation and **(b)** regularization terms in the loss, which preserve the 'validity' of adversarial perturbations, forcing them to be similar to observed linguistic variations. Second, to improve general robustness to visual manipulations, we introduce a novel mixup image replacement technique, which is based on substituting objects between 'positive' images. Through our experiments and ablation studies we show the benefits of explicit incorporation of linguistic variation to adversarial perturbations for a latent representation to improve robustness to linguistic variations.

References

Agarwal, V.; Shetty, R.; and Fritz, M. 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*, 9690–9698.

Agrawal, A.; Batra, D.; and Parikh, D. 2016. Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356*.

Agrawal, A.; Batra, D.; Parikh, D.; and Kembhavi, A. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4971–4980.

Alberti, C.; Ling, J.; Collins, M.; and Reitter, D. 2019. Fusion of detected objects in text for visual question answering. *arXiv preprint arXiv:1908.05054*.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Cao, J.; Gan, Z.; Cheng, Y.; Yu, L.; Chen, Y.-C.; and Liu, J. 2020. Behind the Scene: Revealing the Secrets of Pre-trained Vision-and-Language Models. *arXiv preprint arXiv:2005.07310*.

Chen, J.; Yang, Z.; and Yang, D. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chen, Y.-C.; Li, L.; Yu, L.; Kholy, A. E.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2020. Uniter: Universal image-text representation learning. In *ECCV*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. .

Gan, Z.; Chen, Y.-C.; Li, L.; Zhu, C.; Cheng, Y.; and Liu, J. 2020. Large-Scale Adversarial Training for Vision-and-Language Representation Learning. *NeurIPs*.

Gokhale, T.; Banerjee, P.; Baral, C.; and Yang, Y. 2020. VQA-LOL: Visual question answering under the lens of logic. *ECCV*.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2016. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. arXiv:1612.00837.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 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*, 6904–6913.

Hao, W.; Li, C.; Li, X.; Carin, L.; and Gao, J. 2020. Towards learning a generic agent for vision-and-language navigation via pre-training. In *CVPR*.

Hochreiter, S.; and Schmidhuber, J. 1996. LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, 9.

Hu, X.; Yin, X.; Lin, K.; Wang, L.; Zhang, L.; Gao, J.; and Liu, Z. 2020. VIVO: Surpassing Human Performance in Novel Object Captioning with Visual Vocabulary Pre-Training. *arXiv preprint arXiv:2009.13682*.

Hudson, D. A.; and Manning, C. D. 2019. GQA: a new dataset for compositional question answering over real-world images. In *CVPR*.

Iyyer, M.; Wieting, J.; Gimpel, K.; and Zettlemoyer, L. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

Jiang, H.; Misra, I.; Rohrbach, M.; Learned-Miller, E.; and Chen, X. 2020. In Defense of Grid Features for Visual Question Answering. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10264–10273.

Junczys-Dowmunt, M.; Grundkiewicz, R.; Dwojak, T.; Hoang, H.; Heafield, K.; Neckermann, T.; Seide, F.; Germann, U.; Fikri Aji, A.; Bogoychev, N.; Martins, A. F. T.; and Birch, A. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, 116–121. Melbourne, Australia: Association for Computational Linguistics.

Kant, Y.; Moudgil, A.; Batra, D.; Parikh, D.; and Agrawal, H. 2021. Contrast and classify: Training robust vqa models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1604–1613.

Li, L.; Lei, J.; Gan, Z.; and Liu, J. 2021. Adversarial vqa: A new benchmark for evaluating the robustness of vqa models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2042–2051.

Li, X.; Yin, X.; Li, C.; Hu, X.; Zhang, P.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; et al. 2020. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *ECCV*.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Majumdar, A.; Shrivastava, A.; Lee, S.; Anderson, P.; Parikh, D.; and Batra, D. 2020. Improving Vision-and-Language Navigation with Image-Text Pairs from the Web. *arXiv preprint arXiv:2004.14973*.

Murahari, V.; Batra, D.; Parikh, D.; and Das, A. 2020. Large-scale Pretraining for Visual Dialog: A Simple State-of-the-Art Baseline. In *ECCV*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks.

Rosenberg, D.; Gat, I.; Feder, A.; and Reichart, R. 2021. Are VQA systems rad? measuring robustness to augmented data with focused interventions. *arXiv preprint arXiv:2106.04484*.

Selvaraju, R. R.; Tendulkar, P.; Parikh, D.; Horvitz, E.; Ribeiro, M.; Nushi, B.; and Kamar, E. 2020. Squinting at vqa models: Interrogating vqa models with sub-questions. *CVPR*.

Shah, M.; Chen, X.; Rohrbach, M.; and Parikh, D. 2019. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6649–6658.

Shetty, R. R.; Fritz, M.; and Schiele, B. 2018. Adversarial scene editing: Automatic object removal from weak supervision. In *NeurIPs*.

Shi, L.; Shuang, K.; Geng, S.; Su, P.; Jiang, Z.; Gao, P.; Fu, Z.; de Melo, G.; and Su, S. 2020. Contrastive Visual-Linguistic Pretraining. *arXiv preprint arXiv:2007.13135*.

Wang, Y.; Joty, S.; Lyu, M. R.; King, I.; Xiong, C.; and Hoi, S. C. 2020. Vd-bert: A unified vision and dialog transformer with bert. *arXiv preprint arXiv:2004.13278*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771.

Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; and He, K. 2017. Aggregated Residual Transformations for Deep Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, X.; Chen, X.; Liu, C.; Rohrbach, A.; Darrell, T.; and Song, D. 2018. Fooling vision and language models despite localization and attention mechanism. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4951–4961.

Yang, Z.; Hu, Z.; Salakhutdinov, R.; and Berg-Kirkpatrick, T. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *International conference on machine learning*, 3881–3890. PMLR.

Yu, L.; Poirson, P.; Yang, S.; Berg, A. C.; and Berg, T. L. 2016. Modeling context in referring expressions. In *ECCV*.

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhou, L.; Palangi, H.; Zhang, L.; Hu, H.; Corso, J. J.; and Gao, J. 2020. Unified vision-language pre-training for image captioning and vqa. In *AAAI*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.