MANGO: Enhancing the Robustness of VQA Models via Adversarial Noise Generation

Anonymous ACL submission

Abstract

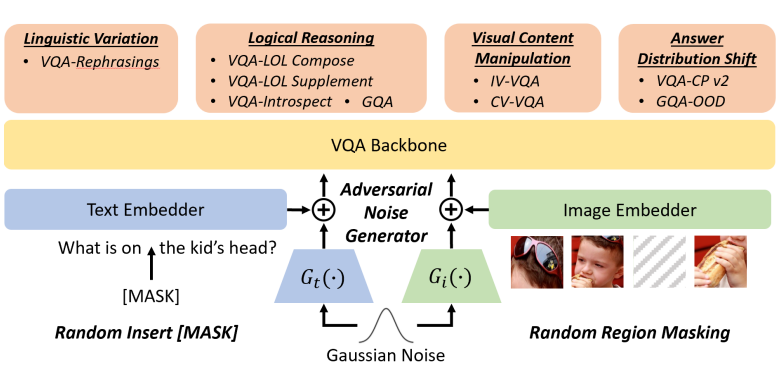
Large-scale pre-trained vision-and-language (V+L) transformers have propelled the state of the art (SOTA) on Visual Question Answering (VQA) task. Despite impressive performance on the standard VQA benchmark, it remains unclear how robust these models are. To investigate, we conduct a host of evaluations over 4 different types of robust VQA datasets: (i) Linguistic Variation; (ii) Logical Reasoning; (iii) Visual Content Manipulation; and (iv) Answer Distribution Shift. Experiments show that pre-trained V+L models already exhibit better robustness than many task-specific SOTA methods via standard model finetuning. To further enhance model robustness, we propose MANGO, a generic and efficient approach that learns a **M**ultimodal **A**dversarial **N**oise **G**enerat**O**r in the embedding space to fool V+L models. Differing from previous studies focused on one specific type of robustness, MANGO is agnostic to robustness types, and enables universal performance lift for both task-specific and pre-trained models over diverse robust VQA datasets designed to evaluate broad aspects of robustness. Comprehensive experiments demonstrate that MANGO outperforms previous task-specific SOTAs on 7 out of 9 robustness benchmarks.

1 Introduction

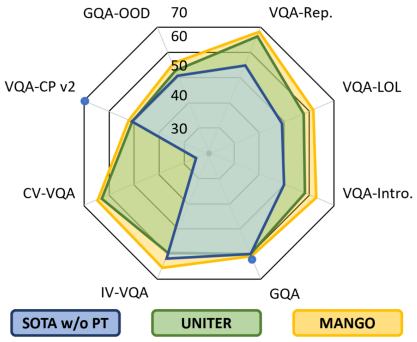
Large-scale multimodal pre-training has taken innovative strides in the realm of vision-and-language (V+L) research (Lin et al., 2020; Lu et al., 2020; Sun et al., 2019; Li et al., 2020b). Pre-trained models (Su et al., 2020; Li et al., 2020a, 2019; Huang et al., 2020) such as ViLBERT (Lu et al., 2019), LXMERT (Tan and Bansal, 2019) and UNITER (Chen et al., 2020b) have demonstrated great generalizability over diverse V+L tasks (Zellers et al., 2019; Yu et al., 2016), especially on the most popular Visual Question Answering (VQA) (Antol et al., 2015) task. However, the standard VQA evaluation benchmarks (Antol et al., 2015; Goyal et al., 2017) usually possess similar data distribution between training and test sets, with little-to-none linguistic variation in textual queries, and use only clean natural images without any visual content manipulation. Although effective for benchmarking model improvements, these standard benchmarks lack the ability to explicitly evaluate model *robustness*.

To conduct a full dissection on model robustness, we launch a comprehensive investigation with systematic evaluations of VQA models over 4 generic types of robustness: (i) robustness against *linguistic variation*; (ii) robustness against *logical reasoning*; (iii) robustness against *visual content manipulation*; and (iv) robustness against *answer distribution shift* between training and test splits. Given the abundance of diverse datasets and splits on the popular VQA task, we take VQA as the focal point of our investigation, and compile an assemblage of 9 diverse VQA datasets that cover each type of model robustness: (i) VQA-Rephrasings (Shah et al., 2019) for *linguistic variation*; (ii) VQA-LOL (Compose and Supplement) (Gokhale et al., 2020b), VQA-Introspect (Selvaraju et al., 2020) and GQA (Hudson and Manning, 2019a) for *logical reasoning*; (iii) IV-VQA and CV-VQA (Agarwal et al., 2020) for *visual content manipulation*; and (iv) VQA-CP v2 (Agrawal et al., 2018) and GQA-OOD (Kervadec et al., 2020) for *answer distribution shift*. Interestingly, analysis on several pre-trained VQA models reveals that by standard finetuning, pre-trained models already exhibit better robustness than many task-specific state-of-the-art methods. However, the achieved robustness is still limited, and far from human performance.

Recently, adversarial training (AT) (Tramèr et al., 2017; Shafahi et al., 2019; Xie et al., 2020) has shown success on *standard* V+L tasks (Gan et al., 2020; Tang et al., 2020). Inspired by this, we investigate whether AT can also serve as an effective conduit to improve performance on robustness



(a) Overview of MANGO, which trains adversarial noise generators to add perturbations at embedding level. Random masking on image and text inputs are designed to promote more diverse adversarial embeddings.



(b) Comparison between MANGO, UNITER and task-specific methods. The blue dots represent methods exploiting additional task-specific information.[1]

Figure 1: Illustration of the proposed MANGO framework and performance comparison between MANGO and SOTA.

benchmarks aforementioned. Our evaluation of VILLA (Gan et al., 2020) (AT-enhanced pre-trained model) shows that by injecting adversarial perturbation to multimodal embeddings, PGD-based (Projected Gradient Descent) AT (Madry et al., 2017; Zhu et al., 2020) can help the model adapt to linguistic variation and visual content manipulation, yielding better model robustness; but with only limited effect (sometimes even hurting model performance) on datasets that exhibit salient data distribution gap between training and test sets (*e.g.*, VQA-CP v2, GQA-OOD).

To achieve better robustness across all aspects, we propose MANGO (**M**ultimodal **A**dversarial **N**oise **G**enerat**O**r), a generic and efficient approach that introduces adversarial noise to multimodal embedding space for robustness enhancement. As shown in Figure 1a, instead of relying on PGD to generate adversarial perturbation, MANGO learns an adversarial noise generator in the form of a trained neural network to fool the model. Following Gan et al. (2020), perturbation is added to the embedding space for all modalities, as our goal is the *end results* of AT, rather than crafting actual adversarial examples. MANGO is lightweight, does not require repetitive gradient calculations on a deep model as in PGD-based approach.

To enable diverse adversarial embeddings, we further propose to randomly mask image regions and randomly insert `[MASK]` tokens when adding adversarial noise to image and word embeddings. Empirical results show that MANGO significantly improves model robustness across all tasks considered, compared to PGD-based methods.

Our main contributions are summarized as follows. (*i*) We show that V+L pre-training can greatly lift the robustness of VQA models across four different robustness types, suggesting stronger baselines for future studies on robust VQA benchmarks. (*ii*) We propose MANGO, a generic and lightweight adversarial noise generator to enhance VQA model robustness. (*iii*) As summarized in Figure 1b, MANGO improves over UNITER and outperforms previous task-specific SOTAs on 7 out of 9 robustness benchmarks.

2 Robust VQA

Terminology We start with definition of the terminology we use throughout the paper. We follow VQA literature (Cadene et al., 2019; Wu and Mooney, 2019; Teney et al., 2020c; Gokhale et al., 2020a; Kervadec et al., 2020) to unify different forms of challenging bias and out-of-distribution generalization as *robustness*, different from its definition in adversarial machine learning. Robustness does not always mean "adversarial robustness" in literature, *e.g.*, it can also refer to model robustness towards common image corruptions (Rusak et al., 2020; Zhang, 2019; Hendrycks and Dietterich, 2019). In the language of adversarial machine learning, our definition of robustness here can be understood as the "generalization" performance on the challenging robust VQA benchmarks.

Existing Benchmarks There has been a few independent studies on V+L robustness, mostly focusing on variations of the popular VQA task. VQA-CP (Agrawal et al., 2018), drawn from VQA v2 dataset (Goyal et al., 2017), is the first benchmark proposed to evaluate (and reduce) question-

[1] LMH (Tramèr et al., 2017) and MMN (Chen et al., 2021) on VQA-CP v2 and GQA are used to plot the SOTA polygon for fair comparison. For CV-VQA and IV-VQA, performance is computed as 100−#flips and 100 − 5×#flips, respectively. VQA-LOL performance is the average of accuracies on VQA-LOL Compose and VQA-LOL Supplement.

Type	Benchmark	Metric	Q Type	Train			Val		Test	
				Source	#IQ	len(Q)	#IQ	len(Q)	#IQ	len(Q)
Lingual	VQA-Rephrasings	Acc.	All	VQA v2 train	444K	6.20	162K	7.15	-	-
Reason	VQA-LOL Compose	Acc.	Y/N	VQA v2 train	444K	6.20	43K	12.09	291K	12.12
	VQA-LOL Supplement	Acc.	Y/N	VQA v2 train	444K	6.20	9K	15.15	669K	15.19
	VQA-Introspect	M✓S✓	All	VQA v1 train	248K	6.21	-	-	95K	6.36
	GQA	Acc.	All	-	943K	8.76	132K	8.77	13K	8.51
Visual	IV-VQA	#flips	All	VQA v2 train	444K	6.20	120K	5.85	-	-
	CV-VQA	#flips	Num.	VQA v2 train	444K	6.20	4K	5.83	-	-
Answer	VQA-CP v2	Acc.	All	-	438K	6.14	-	-	220K	6.31
	GQA-OOD	Acc.	All	GQA train	943K	8.76	51K	8.09	3K	7.70

Table 1: Detailed descriptions of each downstream benchmark, including robustness type, evaluation metric, question type, training data source and statistics on train, val, test data in terms of number of Image-Question pairs (#IQ) and average question length (len(Q)). We use the training data provided with the benchmark unless specified otherwise. Results on val split are reported when test split is not available. Acc. is short for Accuracy. M✓S✓ is a consistency measure between main questions and sub-questions in VQA-Introspect. #flips is the number of predictions mismatched before and after visual content manipulation.

oriented language bias in VQA models. Considerable effort (KV and Mittal, 2020; Cadene et al., 2019; Selvaraju et al., 2019; Abbasnejad et al., 2020) has been invested on VQA-CP along 3 dimensions: (i) compensating for question-answer distribution patterns through a regularizer based on an auxiliary model (Niu et al., 2020; Clark et al., 2019; Teney et al., 2020b; Grand and Belinkov, 2019; Jing et al., 2020); (ii) taking advantage of additional supervision from human-generated attention maps (Wu and Mooney, 2019; Gokhale et al., 2020a); and (iii) synthesizing counterfactual examples to augment training set (Chen et al., 2020a; Teney et al., 2020a). Recent work (Teney et al., 2020c) shows that simple methods such as generating answers at random can already surpass state of the art on some question types. The recent GQA-OOD (Kervadec et al., 2020), another robustness-focused task, is designed based on a fine-grained reorganization of the original GQA dataset (Hudson and Manning, 2019a).

Other types of VQA model robustness are also studied: VQA-Rephrasings (Shah et al., 2019) proposes cyclic consistency to improve robustness against linguistic variations in questions; Ray et al. (2019) tackles antonym consistency; Agarwal et al. (2020) studies robustness against automated semantic image manipulations, and tests for prediction consistency to questions on clean images and corresponding manipulated images.

Further studies investigate robustness against logical reasoning. For instance, Selvaraju et al. (2020) provides a dataset containing perception-related sub-questions per question for a new reasoning split of VQA dataset. VQA-LOL (Gokhale et al., 2020b) perform logical compositions and linguistic transformations to VQA questions to examine model ability in logical reasoning. Moreover, large-scale rule-based questions in GQA (Hudson and Manning, 2019a) can also support analysis on different reasoning skills of VQA models.

Despite the continuous effort in enhancing robustness of VQA models, these works mostly focus on either task-specific models or a single type of robustness. To provide a comprehensive study on pretrained V+L models for VQA robustness, we compile a list of existing datasets, and group them into four robustness types: *Lingual*, *Visual*, *Reason*, and *Answer* (Table 1). Covering various respects of a 'stress test', from *linguistic* to *visual* variations, from *reasoning complexity* to *answer distribution*, this compilation can serve as a unified yardstick for evaluating V+L model robustness and a guidance for future study on robust model design. As a start, we introduce a generic and effective approach that can lift model performance over all types of VQA robustness indiscriminately.

3 MANGO Framework

In this section, we briefly review VQA backbone, introduce a simple baseline that injects Gaussian noise, and explain the proposed MANGO approach.

3.1 VQA Backbone

Given an image-question pair $(\boldsymbol{v}, \boldsymbol{w})$ in dataset $\mathcal{D}$, the goal is to predict an answer that best matches ground-truth answer $\boldsymbol{y}$. The image input is usually projected into a set of region-level features (Anderson et al., 2018) or image patch embeddings (Kim et al., 2021) $\boldsymbol{v} = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_K\}$ ($\boldsymbol{v}_i \in \mathbb{R}^{d_v}$). The text input is tokenized and projected into high-dimensional feature vectors $\boldsymbol{w} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_L\}$ through a learnable word embedding layer ($\boldsymbol{w}_i \in \mathbb{R}^{d_w}$). These embeddings from the paired image-question inputs are then fed into a VQA model $f_{\boldsymbol{\theta}}(\boldsymbol{v}, \boldsymbol{w})$ to predict an answer, where $\boldsymbol{\theta}$ denotes

all the trainable parameters. Binary Cross Entropy (BCE) loss is used to supervise model training. The training process can be formulated as:

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{v},\boldsymbol{w},\boldsymbol{y})\sim\mathcal{D}} [\mathcal{L}_{\text{BCE}}(f_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{w}),\boldsymbol{y})]\,. \tag{1}$$

3.2 Gaussian Noise Augmentation

Randomized smoothing (Duchi et al., 2012) advocates the addition of random perturbations to model inputs, which can often yield better model performance. Recent study (Rusak et al., 2020) also shows that perturbing clean images with Gaussian noise is effective in improving model robustness against image corruptions for image classification. Inspired by this, we use Gaussian noise augmentation as a simple baseline to investigate model robustness under V+L setting. Instead of adding noise to raw image pixels as in Rusak et al. (2020), we add perturbations directly to the embeddings:

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{v},\boldsymbol{w},\boldsymbol{y})\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\delta}_v\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{1})} [\mathcal{L}_{\text{BCE}}(f_{\boldsymbol{\theta}}(\boldsymbol{v}+\boldsymbol{\delta}_v,\boldsymbol{w}),\boldsymbol{y})]\,, \tag{2}$$

where σ is the standard deviation of Gaussian noise. Similarly, we add Gaussian noise to the word embeddings:

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{v},\boldsymbol{w},\boldsymbol{y})\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\delta}_w\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{1})} [\mathcal{L}_{\text{BCE}}(f_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{w}+\boldsymbol{\delta}_w),\boldsymbol{y})]\,.$$

3.3 Adversarial Noise Generator

Adding Gaussian noise to clean image-text pairs can augment training examples to a certain level. However, as the training continues, the model can gradually adapt to the perturbations which are sampled from the *same* Gaussian noise distribution. To produce *harder* perturbations that can fool the backbone network, we propose to actively learn an adversarial noise generator. Specifically, we aim to discover an adversarial noise distribution, from which the sampled noises, when added to the multimodal embeddings, can maximally confuse the backbone network. Note that our goal is not to model the explicit density form of such a distribution, as we only care about the noise samples drawn from the distribution. To achieve this, the adversarial noise generator takes in Gaussian noise samples as input, and produces adversarial noise samples through a learned neural network.

Take image modality as an example. Let $g_{\boldsymbol{\phi}_v} : \mathbb{R}^{d_v} \rightarrow \mathbb{R}^{d_v}$ denote the adversarial noise generator. The adversarial noise $\boldsymbol{\delta}_v$ is generated by $\boldsymbol{\delta}_v = g_{\boldsymbol{\phi}_v}(\boldsymbol{\alpha}), \boldsymbol{\alpha} \in \mathcal{N}(\mathbf{0},\mathbf{1})$. Intuitively, to maximally fool the backbone network, we want to maximize prediction errors on these adversarially perturbed samples. In the meantime, we want the model to possess less confidence in its predictions on perturbed samples than clean samples, to promote harder adversarial examples Therefore, the objective of the adversarial noise generator is to *maximize* the sum of two losses: (i) task-specific loss (*e.g.*, BCE loss for VQA task); (ii) adversarial loss (*e.g.*, BCE loss for adversarial data, and the Kullback-Leibler (KL) divergence loss between the predicted answer distribution of perturbed samples and that of clean samples). On the other hand, the trained model aims to minimize both losses by taking adversarial embeddings as data augmentation. Formally, the min-max game can be defined as:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}_v} \mathop{\mathbb{E}}_{(\boldsymbol{v},\boldsymbol{w},\boldsymbol{y})\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\alpha}\in\mathcal{N}(\mathbf{0},\mathbf{1})} [\mathcal{L}_{std}(\boldsymbol{\theta},\boldsymbol{\phi}_v) + \beta\mathcal{R}_{at}(\boldsymbol{\theta},\boldsymbol{\phi}_v)]\,,$$

where β is a hyper-parameter, and

$$\mathcal{L}_{std}(\boldsymbol{\theta},\boldsymbol{\phi}_v) = \mathcal{L}_{\text{BCE}}(f_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{w}),\boldsymbol{y})\,, \tag{3}$$

$$\begin{aligned}\mathcal{R}_{at}(\boldsymbol{\theta},\boldsymbol{\phi}_v) = {} & \mathcal{L}_{\text{BCE}}(f_{\boldsymbol{\theta}}(\boldsymbol{v}+g_{\boldsymbol{\phi}_v}(\boldsymbol{\alpha}),\boldsymbol{w}),\boldsymbol{y}) \\ & + \mathcal{L}_{kl}(f_{\boldsymbol{\theta}}(\boldsymbol{v}+g_{\boldsymbol{\phi}_v}(\boldsymbol{\alpha}),\boldsymbol{w}), f_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{w})),\end{aligned} \tag{4}$$

where $\mathcal{R}_{kl}(p,q) = \text{KL}(p||q) + \text{KL}(q||p)$, p, q denote two probability distributions. The first term in $\mathcal{R}_{at}(\boldsymbol{\theta},\boldsymbol{\phi}_v)$ promotes label-preserving adversarial perturbations; while the second term advocates more fine-grained label preservation, meaning that the probability distribution across all answers is used as *soft* label, instead of using the ground truth answer index as hard label. Similarly, we can learn an adversarial noise generator (with parameters $g_{\boldsymbol{\phi}_w}$) that corresponds to the text modality.[2]

During training, we alternate between an outer loop of the backbone network update and an inner loop of generator update. We constrain the noise samples $\boldsymbol{\delta}_v$ and $\boldsymbol{\delta}_w$ to be within the sphere $||\boldsymbol{\delta}_v||_2 = ||\boldsymbol{\delta}_w||_2 = \epsilon$, by scaling the generator output with a scalar. ϵ is set as $\{1, 2, 5, 10\}$ in our experiments. For better efficiency, we also accumulate the gradients of adversarial noise generator, and only update the generator's parameters every T times ($T = \{20, 40\}$) of backbone update.

The proposed adversarial noise generator is lightweight, consisting of only a few linear layers. To avoid such a light trapping in local minimum when competing with a deep backbone network, at regular intervals, we replace the learned noise generator with a new one trained from scratch. Each time, the new generator is trained against the latest learned parameters of the backbone.

[2]The corresponding equations are omitted for simplicity.

Random Masking Adversarial noise generator, although produces more challenging and more diverse noise perturbations, does not alter the intrinsic statistics of training examples, such as the distribution of question lengths and image regions. In practice, we observe significant mismatch in these statistics between training and test splits of robustness benchmarks. For example, the average length of questions in VQA-LOL (Gokhale et al., 2020b) test split is 2-3 times longer than that in VQA v2 (Goyal et al., 2017) training split. The region distribution of images in IV-VQA and CV-VQA (Agarwal et al., 2020) is very different from VQA v2 training split, due to visual content manipulation. To compensate for such statistic mismatch, we propose to randomly mask image regions (by zeroing out corresponding feature vectors) as well as randomly insert `[MASK]` tokens when adding adversarial noise to image and word embeddings. Empirically, this simple technique is effective in further boosting model robustness.

Comparison with PGD-based AT Although MANGO is similar to VILLA (Gan et al., 2020) in terms of learning adversarial perturbations, they are different in the sense that MANGO learns an adversarial noise generator to generate adversarial perturbations, instead of relying on PGD as in VILLA. This makes MANGO more efficient, as computing gradients of a generic lightweight noise generator is less time-consuming. Empirically, MANGO also achieves better performance. The comparison on model performance and training time difference is provided in Experiments. A detailed literature review on AT is provided in Appendix A.

Comparison with ANT In ANT (Rusak et al., 2020), a similar noise generator is proposed to make neural networks robust against diverse image corruptions. However, there are two key distinctions. First, we mainly focus on transformer models for VQA task, whereas Rusak et al. (2020) focuses on convolutional networks for image classification. Second, we propose to generate adversarial noise over the embeddings of images and words, while Rusak et al. (2020) adds adversarial noise directly on image pixels.

4 Experiments

We experiment on BUTD (Anderson et al., 2018), pre-trained UNITER (Chen et al., 2020b) and VILLA (Gan et al., 2020) over all 9 robust VQA datasets (Sec. 2), plus a standard VQA-v2 dataset. UNITER is a one-stream model based on object detection to extract visual features. We also experiment on LXMERT (a two-stream model instead) (Tan and Bansal, 2019), and ViLT (directly taking image patches and word tokens as model inputs) (Kim et al., 2021) for generalizability test.

4.1 Experimental Setting

We follow the original papers to test model robustness under the most challenging setting (shown in Table 1), which is to evaluate models trained on the VQA training split for VQA-Rephrasings, VQA-LOL, VQA-Introspect, IV-VQA and CV-VQA. Detailed description of all benchmarks are provided in Appendix D. For thorough evaluation, we compare model performance against the following methods:

- **SOTA w/o PT** (*task-specific* models without pre-training): Cycle Consistency+ BAN (Shah et al., 2019) for VQA-Rephrasings, LOL (Gokhale et al., 2020b) for VQA-LOL Compose and Supplement, Pythia (Selvaraju et al., 2020; Jiang et al., 2018) for VQA-Introspect, NSM (Hudson and Manning, 2019b) for GQA, SAAA (Agarwal et al., 2020; Kazemi and Elqursh, 2017) for CV-VQA and IV-VQA, MUTANT (Gokhale et al., 2020a) for VQA-CP v2, MMN (Chen et al., 2021; Kervadec et al., 2020) for GQA-OOD;
- **BUTD** and **MANGO$_{\text{BUTD}}$**: task-specific VQA model and its enhanced version with MANGO[3];
- **UNITER$_{\text{B}}$** and **UNITER$_{\text{L}}$**: standard finetuning of pre-trained UNITER base and large model;
- **VILLA$_{\text{B}}$** and **VILLA$_{\text{L}}$**: adversarial pre-trained and finetuned UNITER base and large model;
- **MANGO$_{\text{B}}$** and **MANGO$_{\text{L}}$**: applying adversarial noise generator on pre-trained UNITER, base and large size;
- **MANGO$_{\text{VB}}$** and **MANGO$_{\text{VL}}$**: applying adversarial noise generator on adversarial pre-trained UNITER model (provided in the VILLA paper (Gan et al., 2020)) with base and large size.

4.2 Experimental Results

Table 2 presents the results of BUTD, UNITER, VILLA and MANGO on all robustness benchmarks.

[3]In practice, we remove random inserting `[MASK]` token for BUTD backbone, as it is not included in its provided vocabularies.

	Model		Lingual	Reason				Visual		Answer		
			VQA-Rep.	VQA-LOL Comp.	VQA-LOL Supp.	VQA-Intro.	GQA	IV-VQA	CV-VQA	VQA-CP v2	GQA-OOD	VQA v2
		Meta-Ave. ↑	Acc. ↑	Acc. ↑	Acc. ↑	M✓ S✓ ↑	Acc. ↑	#flips ↓	#flips ↓	Acc. ↑	Acc. ↑	Acc. ↑
1	SOTA w/o PT	-	56.59	48.99	50.54	50.05	**63.17**	7.53	78.44	**69.52**	52.70	-
2	BUTD	35.03	56.88	53.04	52.53	46.17	55.41	9.83	62.23	40.39	50.29	67.60
3	MANGO$_{\text{BUTD}}$	36.49	57.84	54.98	54.83	47.58	56.50	9.01	58.10	40.60	51.50	68.18
4	UNITER$_{\text{B}}$	40.98	64.56	54.54	50.00	56.80	59.99	8.47	40.67	46.93	53.43	72.70
5	MANGO$_{\text{B}}$	42.80	65.80	56.22	56.49	58.33	60.65	7.32	38.11	47.52	55.15	73.24
6	VILLA$_{\text{B}}$	42.37	65.35	54.90	56.17	58.29	60.26	7.07	38.28	46.39	54.11	73.59
7	MANGO$_{\text{VB}}$	43.08	65.91	55.44	57.58	58.94	60.73	7.43	38.25	48.63	55.79	73.45
8	UNITER$_{\text{L}}$	43.37	67.64	58.60	55.95	57.64	60.30	8.20	36.66	50.98	53.65	73.82
9	MANGO$_{\text{L}}$	45.27	**68.33**	59.45	**60.50**	62.14	61.10	**6.69**	**35.52**	52.76	**56.40**	74.26
8	VILLA$_{\text{L}}$	44.33	68.16	58.66	58.29	62.00	61.38	6.70	37.55	49.10	55.26	**74.69**
9	MANGO$_{\text{VL}}$	**45.31**	68.27	**61.49**	58.83	**62.60**	61.41	6.73	35.64	52.55	56.08	74.20

Table 2: Comparison to task-specific state-of-the-art (SOTA), UNITER, VILLA on 9 robustness downstream benchmarks and a standard VQA benchmark. Results are reported on val split of VQA-Rephrasings (VQA-Rep.), VQA-LOL Compose (Comp.) and Supplement (Supp.), VQA-Introspect (VQA-Intro.), IV-VQA, CV-VQA, VQA-CP v2 and test-dev split of GQA, GQA-OOD and VQA v2. ↑ (↓) indicate the higher (lower) the better.

Meta-Ave (average of scores across all benchmarks) is used as the global metric.[4] We compare task-specific models in L2-3 and pre-trained models with base size (12 layers) in L4-7.

Task-specific VQA model BUTD (L2) establishes a weak baseline across all robustness benchmarks, with a Meta-Ave of 35.03. With pre-training and deeper model architecture, UNITER$_{\text{B}}$ (L4) achieves much stronger performance, with a Meta-Ave of 40.98. MANGO$_{\text{BUTD}}$ (L3) and MANGO$_{\text{B}}$ (L5) achieve across-the-board performance lift on all robustness benchmarks over the corresponding baselines, harnessing an absolute gain of +1.46 and +1.82 on Meta-Ave.

MANGO *vs.* VILLA VILLA$_{\text{B}}$ (L6) improves over the strong baseline UNITER$_{\text{B}}$ by +1.39 Meta-Ave (42.37) via PGD-based adversarial training. As VILLA$_{\text{B}}$ performs adversarial training on both pre-training and finetuning stages, we apply our method to their adversarial pre-trained model for fair comparison. MANGO$_{\text{VB}}$ (L5) outperforms VILLA$_{\text{B}}$ on 7 out of 9 robustness benchmarks, with an absolute gain +0.71 on Meta-Ave. MANGO$_{\text{VB}}$ is particularly effective on reasoning (+1.41 on VQA-LOL Supp.) and OOD benchmarks (+2.24 on VQA-CP, +1.68 on GQA-OOD). We also compare the training speed of Mango$_{\text{VB}}$ and VILLA$_{\text{B}}$ under the same experimental setting. Our experiments show that MANGO$_{VB}$ *is 25% faster than* VILLA$_{B}$ *(1.44* vs. *1.92 second per gradient update step).*[5] We contribute the better efficiency to the use of global noise generator in MANGO instead of iterative PGD steps as in VILLA. More comparisons between MANGO and VILLA are included in Appendix B.

Scaling Up to Large Model Size (24 Layers) Compared to base models (L2&L4), large models (L6&L8) have more advantage on Meta-Ave (UNITER: 43.37(L) vs. 40.98(B); VILLA: 44.33(L) vs. 42.37(B)), which is consistent with the observations on standard V+L tasks in (Chen et al., 2020b; Gan et al., 2020). When applying adversarial noise to large backbone models (L7&L9), MANGO further pushes the margins of performance gain across all benchmarks: an absolute gain of +1.90 over UNITER$_{\text{L}}$ and +0.98 over VILLA$_{\text{L}}$ on Meta-Ave.

End-to-end Comparison with SOTA MANGO achieves new state of the art on 7 out of 9 benchmarks, except VQA-CP v2 and GQA. SOTA methods on these two benchmarks exploit additional task-specific information. Specifically, MUTANT (Gokhale et al., 2020a) for VQA-CP v2 is trained with excessive additional image-question pairs designed to promote positive bias; while NSM (Hudson and Manning, 2019b) for GQA takes advantage of additional scene graph annotations, which are only provided in GQA. As the goal of our proposed method is to bring universal performance lift on all robustness benchmarks, we do not exploit these additional task-specific information introduced by MUTANT and GQA.

4.3 A Closer Look into Robustness

We conduct an in-depth autopsy to examine the robustness of competing methods over each robustness type. For simplicity, we focus our discussions on UNITER$_{\text{B}}$, VILLA$_{\text{B}}$ and MANGO$_{\text{VB}}$.

[4] For IV-VQA and CV-VQA, we take the negative of the number of flips for calculating Meta-Ave.

[5] The speed comparison is conducted during finetuning experiments for both models with the same batch size, gradient accumulation steps and GPUs.

Robustness against Linguistic Variation As shown in Table 2 ('Lingual' column), UNITER$_B$ has shown its advantage of defending model robustness against linguistic variation. We contribute the performance lift from UNITER$_B$ to excessive variations of textual inputs seen during pre-training. Comparing AT-enhanced methods, MANGO$_{VB}$ improves over VILLA$_B$, even though VILLA$_B$ has already shown significant improvement over UNITER$_B$. We attribute the improvement from MANGO to not only the adversarial data augmentation during training, but also the random masking introduced from the text modality (more detailed analyses in Table 3).

Robustness against Logical Reasoning We compare model performance on 4 benchmarks under the 'Reason' column in Table 2. Different from VQA-LOL Compose, VQA-LOL Supplement dataset consists of questions generated by heuristic rules. Semantically-close questions with different answers are included to make the task more challenging. The close-to-random performance on VQA-LOL Supplement dataset indicates that UNITER$_B$ severely suffers from these challenging semantically-close questions.

VILLA$_B$ brings performance lift on all 4 reasoning benchmarks. Not surprisingly, VILLA$_B$ exhibits more robustness than UNITER$_B$ on semantically-close questions in VQA-LOL Supplement. Our hypothesis is that the adversarial embeddings learned during VILLA$_B$ training can mimic the effect of adding semantically-close questions as training data, and the generated adversarial perturbations are also constrained to be small to preserve the semantic meaning of the clean text embeddings.

MANGO$_{VB}$ outperforms VILLA$_B$ on all reasoning benchmarks. Similar to VQA-Rephrasings, MANGO$_{VB}$ has more advantages over VQA-LOL Compose and VQA-LOL Supplement, whose average question length is much longer than VQA v2. By randomly inserting `[MASK]` tokens, MANGO$_B$ effectively augments training data with questions of similar lengths to the test split.

Robustness against Visual Content Manipulation UNITER$_B$ performs on par to SOTA model on IV-VQA, and significantly improves over SOTA on CV-VQA (Table 2 'Visual' column). This is due to that during pre-training, UNITER$_B$ has already be trained on diverse images, and the pre-training task of masked region modeling can also prevent UNITER$_B$ from overfitting to visual biases.

Modality	Method	VQA-Rep. Acc. ↑	VQA-LOL Comp. Acc. ↑	VQA-LOL Supp. Acc. ↑	IV-VQA #flips ↓	VQA-CP v2 Acc. ↑
None	1 None	64.56	54.54	50.00	8.47	47.29
Image	2 GN	65.17	54.46	50.68	8.45	47.29
	3 AN	65.42	54.59	52.54	7.52	47.38
	4 MANGO	65.51	**56.67**	55.20	7.39	47.51
Text	5 GN	64.73	53.66	54.59	8.46	46.59
	6 AN	65.36	54.12	52.95	7.99	47.09
	7 MANGO	65.63	55.79	**56.54**	7.53	47.45
Both	8 MANGO	**65.80**	56.22	56.49	**7.32**	**47.52**

Table 3: Ablation studies on adding noise to different modalities and on different types of noise. UNITER$_B$ is used as the backbone. GN (AN) stands for Gaussian (Adversarial) Noise.

VILLA$_B$ improves model robustness against visual content manipulation, and MANGO$_{VB}$ performs on par with VILLA$_B$. Our hypothesis is that by injecting adversarial perturbations at pre-training stage, the model is exposed to even more diverse images, hence easier to recover from visual biases.

Robustness against Answer Distribution Shift On out-of-distribution (OOD) benchmarks, UNITER$_B$ performs poorly on VQA-CP v2, while improving over SOTA model on GQA-OOD (Table 2 'Answer' column). MUTANT is a very task-specific method, which augments VQA-CP v2 training with excessive rule-based image-question pairs to counter the training split bias. Hence, it is difficult to generalize to other robustness cases. Additional manual effort is required to generalize to other rule-based datasets such as VQA-LOL, GQA, IV-VQA and CV-VQA. Interestingly, VILLA$_B$ improves over UNITER$_B$ on GQA-OOD, but not on VQA-CP v2, while MANGO$_B$ (MANGO$_{VB}$) significantly outperforms UNITER$_B$ (VILLA$_B$) on both benchmarks. These results suggest that MANGO are more generalizable than VILLA to challenging OOD datasets.

4.4 Ablation Study

Noise Generation and Random Masking We select one dataset from each robustness type as a representative benchmark for ablation studies: VQA-CP v2, VQA-Rephrasings, VQA-LOL (Compose and Supplement), and IV-VQA. Results are summarized in Table 3. First, we compare with the baseline that simply adds Gaussian noise to either image or text modality.[6] Different from observations in (Rusak et al., 2020), comparing L2/L5 with L1 indicates that adding simple Gaussian noise to multimodal embeddings is not always helpful. Es-

[6] In our experiments, we set standard deviation to 0.5, and only perturb 50% of training data via Gaussian noise within each minibatch.

Method	VQA-Rep.	VQA-LOL Comp.	VQA-LOL Supp.	GQA	GQA-OOD	VQA v2
	Acc. ↑	Acc. ↑	Acc. ↑	Acc. ↑	Acc. ↑	Acc. ↑
LXMERT	67.20	49.34	47.33	59.78	53.86	72.31
Ours	**68.61**	**53.83**	**53.54**	**60.06**	**54.94**	**72.70**
ViLT	61.87	49.57	50.24	56.60	49.43	70.84
Ours	**62.20**	**51.16**	**52.95**	**57.41**	**49.57**	**71.24**

Table 4: Results of MANGO with LXMERT, ViLT and VinVL as the backbone. VinVL results are reported on base model.

Model	NLVR^2	RefCOCO	RefCOCOg	VE
UNITER$_B$	77.52	80.55	74.41	78.44
MANGO$_B$	**78.36**	**80.95**	**75.37**	**78.87**

Table 5: Results on other V+L tasks, we report the average of performance across different splits of each task for simplicity.

pecially, adding Gaussian noise on text modality brings unstable performance.

Second, we experiment with adding adversarial noise alone, without random masking. Results on L3/L6 show that universal performance improvements over Gaussian noise (L2/L5). Intuitively, adversarial noise is *harder* than Gaussian noise, as the adversarial noise generator learns to fool the backbone network. Such hard training examples helps to boost model robustness.

Third, we show that by using random masking (L4/L7), which encourages more diverse adversarial embeddings, MANGO is better than using adversarial noise alone (L2/L5). Randomly inserting `[MASK]` tokens (L7) also shifts the distribution of question lengths that the model is exposed to during training. Hence, we observe more gains on benchmarks with severe mismatches in question length between training and test sets. For example, in VQA-LOL, the testing questions are significantly longer than training questions on average.

Lastly, we observe that adding adversarial noise on one modality is already gaining significant improvement (L4/L7). Empirically, adding adversarial noise on both modalities (L8) only performs slightly better or on par with MANGO on text or image modality alone. More ablation results on model architecture are included in Appendix B.

Results on Other VQA Backbones We also apply MANGO to other V+L backbones, LXMERT (Tan and Bansal, 2019) and ViLT (Kim et al., 2021), for generalizability test. When comparing both baselines with their MANGO-enhanced versions ("ours" in Table 4), we observe universal performance lift from MANGO across all benchmarks considered. [7]

[7] IV-VQA, CV-VQA and VQA-CP v2 are excluded in this study as the performance on these benchmarks is based on examples in VQA v2 val split, which is used to supervise LXMERT pre-training.

Figure 2: Visualization of model predictions, comparing MANGO(M) against UNITER(U) and VILLA(V). Correct answers are highlighted in green and wrong ones are in red.

Results on other V+L tasks MANGO is task-agnostic, thereby can also be applied to other standard V+L tasks. Table 5 shows that MANGO$_B$ surpasses UNITER$_B$ on 4 popular V+L tasks, including NLVR^2 (Suhr et al., 2019), RefCOCO (Yu et al., 2016), RefCOCOg (Yu et al., 2016) and Visual Entailment (VE) (Xie et al., 2019). We leave thorough investigation of the effectiveness of MANGO on other standard V+L tasks as future study.

Qualitative Analysis Figure 2 visualizes predictions from UNITER, VILLA and MANGO on 4 benchmarks (one for each robustness type). These visualizations illustrate MANGO's consistently accurate performance when facing challenges of: (a) uninformative leading phrase added to the question; (b) removal of irrelevant object in the image; (c) over-length logical combination of questions; and (d) imbalanced answer distribution ('white' appears 3 times as many as 'blue' in training set).

5 Conclusion

We provide a systematic study on the robustness of VQA models over a wide range of robust VQA benchmarks. The comprehensive evaluation shows V+L pre-training can effectively defend model performance under various types of robust tests. We further propose MANGO, a simple yet effective method to enhance model robustness, which advances the state of the art on 7 out of 9 robustness benchmarks. We hope this set of results can be used as baseline for future research.

References

Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. 2020. Counterfactual vision and language learning. In *CVPR*.

Vedika Agarwal, Rakshith Shetty, and Mario Fritz. 2020. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *CVPR*.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. 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 *CVPR*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *ICCV*.

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. 2019. Rubi: Reducing unimodal biases for 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 *CVPR*.

Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. 2021. Meta module network for compositional visual reasoning. *WACV*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020b. Uniter: Universal image-text representation learning. In *ECCV*.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *ICLR*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *EMNLP*.

Corentin Dancette, Remi Cadene, Damien Teney, and Matthieu Cord. 2021. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. In *ICCV*.

Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Zicheng Liu, Michael Zeng, et al. 2021. An empirical study of training end-to-end vision-and-language transformers. *arXiv preprint arXiv:2111.02387*.

John C Duchi, Peter L Bartlett, and Martin J Wainwright. 2012. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. *NeurIPs*.

Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2020a. Mutant: A training paradigm for out-of-distribution generalization in visual question answering. *EMNLP*.

Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2020b. Vqa-lol: Visual question answering under the lens of logic. *ECCV*.

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 *CVPR*.

Gabriel Grand and Yonatan Belinkov. 2019. Adversarial regularization for visual question answering: Strengths, shortcomings, and side effects. *ACL workshop*.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*.

Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. 2021. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *CVPR*.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*.

Drew A Hudson and Christopher D Manning. 2019a. Gqa: a new dataset for compositional question answering over real-world images. In *CVPR*.

Drew A. Hudson and Christopher D. Manning. 2019b. Learning by abstraction: The neural state machine. *NeurIPS*.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*.

Chenchen Jing, Yuwei Wu, Xiaoxun Zhang, Yunde Jia, and Qi Wu. 2020. Overcoming language priors in vqa via decomposed linguistic representations. In *AAAI*.

Vahid Kazemi and Ali Elqursh. 2017. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*.

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

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICLR*.

Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. 2020. Flag: Adversarial data augmentation for graph neural networks. *arXiv preprint arXiv:2010.09891*.

Gouthaman KV and Anurag Mittal. 2020. Reducing language biases in visual question answering with visually-grounded question encoder. *ECCV*.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *ECCV*.

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

Junnan Li, Ramprasaath R Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. 2021a. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020b. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*.

Linjie Li, Jie Lei, Zhe Gan, and Jingjing Liu. 2021b. Adversarial vqa: A new benchmark for evaluating the robustness of vqa models. In *ICCV*.

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

Liunian Harold Li, Haoxuan You, Zhecan Wang, Alireza Zareian, Shih-Fu Chang, and Kai-Wei Chang. 2020c. Weakly-supervised visualbert: Pre-training without parallel images and captions. *arXiv preprint arXiv:2010.12831*.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020d. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

Junyang Lin, An Yang, Yichang Zhang, Jie Liu, Jingren Zhou, and Hongxia Yang. 2020. Interbert: Vision-and-language interaction for multi-modal pretraining. *arXiv preprint arXiv:2003.13198*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

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

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *CVPR*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2020. Counterfactual vqa: A cause-effect look at language bias.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*.

Dong Huk Park, Trevor Darrell, and Anna Rohrbach. 2019. Robust change captioning. In *ICCV*.

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 *EMNLP*.

Evgenia Rusak, Lukas Schott, Roland S. Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. 2020. A simple way to make neural networks robust against diverse image corruptions. *ACL*.

Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. 2019. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *ICCV*.

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. *CVPR*.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! In *NeurIPS*.

M Shah, X Chen, M Rohrbach, and D Parikh. 2019. Cycle-consistency for robust visual question answering. In *CVPR*.

Sasha Sheng, Amanpreet Singh, Vedanuj Goswami, Jose Alberto Lopez Magana, Wojciech Galuba, Devi Parikh, and Douwe Kiela. 2021. Human-adversarial visual question answering.

Rakshith R Shetty, Mario Fritz, and Bernt Schiele. 2018. Adversarial scene editing: Automatic object removal from weak supervision. In *NeurIPs*.

Lei Shi, Kai Shuang, Shijie Geng, Peng Su, Zhengkai Jiang, Peng Gao, Zuohui Fu, Gerard de Melo, and Sen Su. 2020. Contrastive visual-linguistic pretraining. *arXiv preprint arXiv:2007.13135*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*.

Alane Suhr, Stephanie Zhou, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *ACL*.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *ICCV*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*.

Ruixue Tang, Chao Ma, Wei Emma Zhang, Qi Wu, and Xiaokang Yang. 2020. Semantic equivalent adversarial data augmentation for visual question answering. *arXiv preprint arXiv:2007.09592*.

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

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

Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. 2020c. On the value of out-of-distribution testing: An example of goodhart's law. *NeurIPS*.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

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

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. 2020. Adversarial examples improve image recognition. In *CVPR*.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*.

Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. 2021. Probing inter-modality: Visual parsing with self-attention for vision-language pre-training. In *NeurIPS*.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. *arXiv preprint arXiv:2006.16934*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *ECCV*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Making visual representations matter in vision-language models. In *CVPR*.

Richard Zhang. 2019. Making convolutional networks shift-invariant again. In *ICML*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *AAAI*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *ICLR*.

A Detailed Related Work

Multimodal Pre-training Early approaches to vision-and-language pre-training (Lu et al., 2019; Tan and Bansal, 2019) adopt a two-stream architecture. Later on, single-stream architecture gains popularity (Zhou et al., 2020; Su et al., 2020; Li et al., 2020a; Chen et al., 2020b). To enhance the model performance, there have been efforts in designing different training strategies (Lu et al., 2020; Gan et al., 2020; Shi et al., 2020; Li et al., 2020c; Cho et al., 2021) and injecting external knowledge (Li et al., 2020d; Yu et al., 2020) as additional model inputs. While most of these methods reply on offline extracted region-level features (Anderson et al., 2018; Zhang et al., 2021), there has been growing interests in end-to-end learning directly from image pixels (Huang et al., 2020, 2021; Kim et al., 2021; Xue et al., 2021; Li et al., 2021a; Dou et al., 2021).

Distinct from these efforts on improving performance over standard benchmarks,[8] we focus on a different direction, evaluating and enhancing the *robustness* of pre-trained models. This helps us better understand how well multimodal pre-training truly advances this field, and guides us to design more robust models.

Adversarial Training As one of the most effective strategies of defending against adversarial attacks (Szegedy et al., 2013), adversarial training (AT) has been widely studied for enhancing adversarial robustness of neural networks (Tramèr et al., 2017; Shafahi et al., 2019; Xie et al., 2020), using adversarial examples as effective data augmentation. Recent studies show that, by injecting adversarial perturbations into feature space, AT can further improve model generalization on language understanding (Zhu et al., 2020), visual question answering (Gan et al., 2020; Tang et al., 2020), and graph neural networks (Kong et al., 2020).

In our work, we investigate the use of an adversarial noise generator for robustness enhancement, inspired by (Rusak et al., 2020), which proposes a similar noise generator to make neural networks robust against diverse image corruptions.

Robust V+L Datasets There have been continuous efforts in examining the robustness of VQA models. Some recent attempts include, (i) generating adversarial VQA questions with human-and-model-in-the-loop (Li et al., 2021b; Sheng et al., 2021) and (ii) removing multimodal shortcuts, that involve both questions and images, from existing VQA datasets (Dancette et al., 2021).

In addition to robust VQA datasets, CLEVR-Change (Park et al., 2019) has been introduced to study robust Change Captioning, where the model needs to identify an important scene change and using language to describe the change. We hope our work can encourage future works to explore various stress tests on diverse V+L tasks to provide a full dissection of model robustness for pre-trained V+L models.

B More Results

We report model evaluation of prediction consistency on VQA-Rephrasings (Shah et al., 2019), VQA-introspect (Selvaraju et al., 2020), IV-VQA (Agarwal et al., 2020), CV-VQA (Agarwal et al., 2020) and GQA-OOD (Kervadec et al., 2020). We also include more detailed results on VQA v2 (Goyal et al., 2017), and additional ablation experiments on model architecture.

Evaluation on Consistency In addition to accuracy, many benchmarks consider consistency as an additional measure for evaluating model robustness. Here, we take VQA-Rephrasings and VQA-Introspect as examples to demonstrate that MANGO can also help boost consistency in model predictions. Results are summarized in Table 6.

On VQA-Rephrasings, we investigate consistency in model predictions on different variants of semantically equivalent questions. Consistency is measured by a Consensus Score $CS(k)$.[9] MANGO achieves universal performance lift across all consistency measures, compared to each baseline model. The best results are achieved by MANGO_{L}, surpassing SOTA by *+9.43*, *+12.27*, *+13.62*, *+14.40* on $CS(k), k = 1, 2, 3, 4$, respectively.

On VQA-Introspect, we examine consistency between the main reasoning questions and perceptual sub-questions, measured by 5 metrics. Similarly, MANGO brings universal consistency improvements across all baseline models. The best performance is achieved by MANGO_{VL}, surpassing

[8] Examples of standard benchmarks include VQA (Antol et al., 2015), VCR (Zellers et al., 2019), NLVR^2 (Suhr et al., 2019), Image-Text Retrieval (Lee et al., 2018), and Referring Expressions (Yu et al., 2016).

[9] Consensus Score is the ratio of the number of subsets where all the answers are correct and the total number of subsets of size k. For every group Q with n rephrasings, all subsets of size k are sampled. The answer to a question is considered correct if it has a non-zero VQA accuracy.

Model	VQA-Rephrasings				VQA-Reas.	VQA-Introspect				
	CS(1) ↑	CS(2) ↑	CS(3) ↑	CS(4) ↑	Acc. ↑	M✓ S✓ ↑	M✓ S× ↓	M× S✓ ↓	M× S× ↓	S✓|M✓ ↑
SOTA	65.77	56.94	51.76	48.18	69.61	50.05	19.73	17.40	12.83	71.73
BUTD	63.73	54.52	49.13	45.42	65.19	46.17	19.01	20.61	14.20	70.82
MANGO$_{\text{BUTD}}$	64.55	55.60	50.34	46.73	65.76	47.58	18.18	20.64	13.60	72.35
UNITER$_{\text{B}}$	71.29	63.95	59.48	56.31	73.33	56.80	16.53	16.93	9.74	77.46
MANGO$_{\text{B}}$	72.66	66.03	61.92	58.95	74.20	58.33	15.88	16.76	9.04	78.60
VILLA$_{\text{B}}$	72.18	65.28	60.99	57.93	73.63	58.29	15.34	17.08	9.30	79.17
MANGO$_{\text{VB}}$	72.78	65.97	61.70	58.59	74.41	58.94	15.47	16.59	9.00	79.20
UNITER$_{\text{L}}$	74.44	67.93	63.85	60.86	72.99	57.64	15.35	17.54	9.47	79.01
MANGO$_{\text{L}}$	**75.20**	**69.21**	**65.38**	**62.58**	76.91	62.14	14.71	15.40	7.74	80.86
VILLA$_{\text{L}}$	74.93	68.65	64.61	61.61	76.18	62.00	**14.19**	15.72	8.10	**81.38**
MANGO$_{\text{VL}}$	75.17	69.01	65.07	62.16	**77.20**	**62.60**	14.60	**15.13**	**7.67**	81.09

Table 6: Results of consistency evaluations on VQA-Rephrasings and VQA-Introspect. VQA-Reasoning (VQA-Reas.) is a split of VQA-Introspect, containing only the main reasoning questions (M). S stands for sub-questions. ✓ or × indicate a correct or wrong prediction.

Model	VQA	IV-VQA					VQA Num.	CV-VQA				
	Acc.↑	Acc.↑	# of flips ↓	p2n ↓	n2p ↓	n2n ↓	Acc.↑	Acc.↑	# of flips ↓	p2n ↓	n2p ↓	n2n ↓
SOTA	70.26	-	7.85	3.47	2.79	1.58	49.90	-	78.44	31.66	25.38	21.40
BUTD	63.92	73.73	9.83	4.29	3.42	2.12	44.14	50.16	62.23	25.53	21.56	15.14
MANGO$_{\text{BUTD}}$	64.63	74.53	9.01	4.08	3.27	1.66	44.99	54.29	58.10	23.01	20.72	14.37
UNITER$_{\text{B}}$	70.34	83.35	8.47	3.89	2.60	1.97	53.82	63.22	40.67	23.21	10.72	6.74
MANGO$_{\text{B}}$	71.17	82.69	7.32	3.55	2.27	1.49	54.86	64.21	38.11	22.22	9.97	5.92
VILLA$_{\text{B}}$	71.27	82.87	7.07	3.48	2.16	1.44	55.02	65.06	38.28	22.17	10.60	5.51
MANGO$_{\text{VB}}$	71.47	82.84	7.43	3.57	2.34	1.52	55.27	65.66	38.25	22.10	9.78	6.38
UNITER$_{\text{L}}$	72.60	**85.86**	8.20	3.96	2.37	1.88	56.61	67.13	36.66	22.05	9.80	4.81
MANGO$_{\text{L}}$	73.06	84.05	**6.69**	**3.34**	2.00	1.34	57.44	67.30	**35.52**	**21.59**	**8.55**	5.39
VILLA$_{\text{L}}$	**73.20**	84.79	6.70	3.45	1.95	**1.29**	57.43	65.54	37.55	24.05	8.94	**4.56**
MANGO$_{\text{VL}}$	72.96	84.70	6.73	3.42	**1.89**	1.42	**57.53**	**67.86**	35.64	22.07	8.79	4.78

Table 7: Detailed Results on IV-VQA and CV-VQA. ↑ (↓) indicate the higher (lower) the better. We compare with UNITER, VILLA and task-specific SOTA method (Kazemi and Elqursh, 2017).

SOTA by *+12.55*, *+5.54*, *+2.27*, *+5.16*, *+10.10* on M✓S✓, M✓S×, M×S✓, M×S×, and S✓|M✓, respectively.

On IV-VQA and CV-VQA, we decouple the inconsistency in model predictions on edited images (measured by #flips) into 3 categories: (*i*) p2n: answer predicted on the edited image was wrong, but the prediction on the corresponding real image was correct; (*ii*) n2p: model makes a correct prediction on the edited image, while predicting a wrong answer on real image; (*iii*) n2n: different answers were predicted on edited and real images and both are wrong. These metrics may expose that there is brittleness even when the model makes correct predictions, indicating that models often exploit spurious correlations while making predictions. We follow (Agarwal et al., 2020) to report accuracy on VQA v2 val split to serve as reference for IV-VQA, and performance on counting questions in VQA v2 val split for CV-VQA.

Similar conclusions are drawn from the results presented in Table 7. First, MANGO brings consistent performance improvements across all metrics on both benchmarks, compared to BUTD and UNITER. Second, MANGO significantly improves over SOTA. We also observe significant improvements from MANGO over VILLA on CV-VQA. These results suggest that for *challenging* questions such as counting problems in CV-VQA, MANGO is more robust than VILLA.

On GQA-OOD, except for the accuracy over all GQA-OOD samples ('All' in Table 8), three additional metrics are considered: (*i*) the accuracy on OOD samples, which are the samples of the tail of the answer class distribution ('Tail'); (*ii*) the accuracy on the head of distribution ('Head'); and (*iii*) Δ(head, tail) = (head - tail)/tail to illustrate how much the error prediction is imbalanced between frequent and rare answers ('Δ'). More details on the statistics of head and tail examples can be found in (Kervadec et al., 2020). MANGO achieves universal performance lift across all accuracy measures, compared to each baseline model. However, better accuracy does not indicate better-

Model	All ↑	Tail ↑	Head ↑	Δ ↓
SOTA (best All)	52.70	48.00	55.50	15.60
SOTA (best Δ)	50.20	47.20	51.90	**9.90**
BUTD	50.29	44.31	53.38	20.40
MANGO$_{\text{BUTD}}$	51.50	47.13	54.36	15.34
UNITER$_{\text{B}}$	53.43	48.45	56.49	16.59
MANGO$_{\text{B}}$	54.47	50.24	57.07	13.59
VILLA$_{\text{B}}$	54.11	49.86	56.72	13.76
MANGO$_{\text{VB}}$	55.79	50.89	58.74	15.43
UNITER$_{\text{L}}$	53.65	48.82	56.61	15.96
MANGO$_{\text{L}}$	**56.40**	**51.27**	**59.55**	16.15
VILLA$_{\text{L}}$	55.26	50.80	58.05	14.27
MANGO$_{\text{VL}}$	56.08	**51.27**	59.03	15.14

Table 8: Detailed Results on GQA-OOD. ↑ (↓) indicate the higher (lower) the better. We compare with both SOTA (best ALL) (Chen et al., 2021) and SOTA (best Δ) (Kim et al., 2018).

Method	All	Y/N	Num	Other
UNITER$_{\text{B}}$	72.70	88.97	55.67	62.81
MANGO$_{\text{B}}$	**73.24**	**89.27**	**56.48**	**63.34**

Table 9: Detailed results of UNITER$_{\text{B}}$ and MANGO$_{\text{B}}$ on VQA v2.

balanced predictions between tail and head splits. We observe that there are more performance improvements on head split than tail split. When compared to SOTA, MANGO$_{\text{B}}$ surpasses MMN (Chen et al., 2021) (SOTA with the best All) across all metrics. BAN (Kim et al., 2018) is the SOTA method with the best Δ; however, it suffers on all accuracy measures.

On VQA v2, we use MANGO$_{\text{B}}$ and UNITER$_{\text{B}}$ as examples to show that our method can provide universal performance lift for each question type. This is also consistent with our observations on various robust vqa benchmarks, as they focus on different question types by design. For examples, IV-VQA speficially desgined for counting questions, VQA-LOL only includes yes/no questions.

Method	VQA-Rep.	GQA	VQA v2	VQA-CP v2	GQA-OOD
Human	**98.91**	**89.30**	**80.78**	**80.78**†	**89.30**†
MANGO	68.33	61.41	74.20	52.76	56.40

Table 10: Comparison between human performance and results from MANGO. † we use human performance on VQA v2 and GQA as estimation of human performance as VQA-CP v2 and GQA-OOD.

Comparison to Human Performance We compare the human performance made available by the original authors with the best performance achieved by MANGO in Table 10. The large gap between MANGO and human performance suggest that there are still room to improve model robustness. Note that for VQA-CP v2 and GQA-OOD, which are re-distribution of VQA v2 and GQA, we can use human performance on the original datasets as reference. The gaps between human and SOTA methods are even larger on these two OOD datasets.

Additional Ablations Table 11, we make a direct comparison between adversarial noise generator (MANGO) and PGD-based AT (VILLA) during finetuning stage, when both models are initialized with pre-trained UNITER$_{\text{B}}$ weights. As results shows, MANGO is on par with VILLA on standard VQA v2 dataset and IV-VQA, but more competitive on all other robust VQA benchmarks. Note that the design of MANGO is not specific for finetuning experiments only. Similar to VILLA, it can be naturally extended to pre-training stage, which is an interesting direction for future works to explore.

We conduct additional ablation studies to validate several model design choices of MANGO, including KL-divergence Loss, retraining noise generator every T steps (retrain NG), the architecture of NG (multiple linear layers with nonlinear activation) and the effectiveness of masking on VILLA. Results are reported in Table 12.

A few key observations are summarized here: (i) KL divergence loss contributes to performance improvements in MANGO. (ii) Without resetting generator parameters and retraining generator periodically renders inferior performance. As explained in Section 3.3, the lightweight generator may be trapped in a local optima. In addition, we explore randomly initializing the adversarial noise generator and freeze the generator parameters, which results in even worse performance. (iii) Replacing our noise generator with a single linear layer also hurts the performance. Note that applying linear layers to a Gaussian noise only changes its mean and variance, still results in a Gaussian noise. (iv) VILLA$_{\text{B}}$ + Masking renders weaker performance than MANGO$_{\text{VB}}$. This observation is consistent with comparison of VILLA$_{\text{B}}$ in Table 2 and "AN" in Table 4, which can be considered as "MANGO$_{\text{B}}$ - Masking".

Moreover, we conduct a comparison between simple Gaussian Noise (GN) with MANGO, where noises/perturbations are added to both image and text modalities. Results in the bottom part of Table 12 show that adding simple Gaussian noise to embeddings from both modalities underperforms the proposed MANGO method. When compared with L2 (GN on image modality) and L5 (GN on text modality) of Table 3, we observe that adding

Model		Lingual	Reason				Visual		Answer		
		VQA-Rep.	VQA-LOL Comp.	VQA-LOL Supp.	VQA-Intro.	GQA	IV-VQA	CV-VQA	VQA-CP v2	GQA-OOD	VQA v2
	Meta-Ave. ↑	Acc. ↑	Acc. ↑	Acc. ↑	M✓ S✓ ↑	Acc. ↑	#flips ↓	#flips ↓	Acc. ↑	Acc. ↑	Acc. ↑
MANGO$_{B}$	42.80	**65.80**	**56.22**	56.49	**58.33**	**60.65**	7.32	**38.11**	**47.52**	**55.15**	73.24
VILLA$_{B}$ (fine. only)	41.83	65.02	55.66	53.48	57.24	60.36	**7.20**	40.50	45.96	55.01	**73.29**

Table 11: Direct comparison between adversarial noise generator (MANGO) and PGD-based AT (VILLA (fine. only)) during finetuning stage. Both models are initialized with pre-trained UNITER$_{B}$ weights.

Method	VQA-Rep.	VQA-LOL
MANGO$_{B}$	**65.80**	**56.61**
$-L_{kl}$	65.01	54.55
− retrain NG	65.48	54.57
random init., no training	65.14	54.16
w/ 1-layer linear NG	65.54	53.14
VILLA$_{B}$ + Masking	65.46	55.96
MANGO$_{VB}$	**65.91**	**56.55**
MANGO$_{B}$ (Both)	**65.80**	**56.36**
GN (Both)	64.72	53.62

Table 12: Additional ablation results.

Gaussian noise to both modalities does not yield better performance.

C Implementation Details

Our models are implemented based on PyTorch.[10] To speed up training, we use Nvidia Apex[11] for mixed precision training. Gradient accumulation (Ott et al., 2018) is applied to reduce multi-GPU communication overheads. All experiments are run on Nvidia V100 GPUs (32GB VRAM; NVLink connection). We use AadmW (Loshchilov and Hutter, 2019) with β_1=0.9, β_2=0.98 and an L2 weight decay of 0.01 to optimize model training. Throughout the training, the learning rate is scheduled to warmup over the first 10% training steps followed by linear decay to 0. The peak learning rate is set to be 8e-5 and 5e-5 for base and large models, respectively. Additional hyper-parameters used to train our adversarial noise generators are listed in Table 13. Empirically, we found that model training is sensitive to adversarial noise retrain steps, $p^{\text{img}}_{\text{mask}}$ and $p^{\text{txt}}_{\text{mask}}$.

D Downstream Benchmarks

In addition to dataset statistics summarized in Table 1, we provide an overview of each robustness benchmark as follows.

VQA-Rephrasings (Shah et al., 2019) is based on VQA v2 (Goyal et al., 2017). It contains 3 human-provided rephrasings for 40K questions on 40K images from VQA v2 val split. In addition to accuracy, consistency in model predictions to different semantically-equivalent questions is also used to measure the robustness of VQA models against linguistic variations. We follow (Shah et al., 2019) to evaluate models trained with VQA v2 train split.

VQA-LOL (Gokhale et al., 2020b) is introduced to examine the logical reasoning ability of a VQA model through questions containing logical compositions and linguistic transformations (negation, disjunction, conjunction, and antonyms). It consists of two datasets: VQA-LOL *Compose* (logical combinations of multiple closed binary questions about the same image in VQA v2) and VQA-LOL *Supplement* (logical combinations of additional questions based on external object and caption annotations about the images from COCO (Chen et al., 2015)). Both datasets share the same train/val images as VQA v2. In total, 757K/42.5K/291K and 1.61M/91.8K/669K image-question pairs are generated for train/val/test splits of VQA-LOL Compose and VQA-LOL Supplement, respectively. In our experiments, we follow (Gokhale et al., 2020b) to evaluate models trained with VQA v2 train split on test split of both datasets.

VQA-Introspect (Selvaraju et al., 2020) is created to investigate the consistency in model predictions of a VQA model between reasoning questions and their associated low-level perception questions. It first introduces a new Reasoning split of the VQA v2 dataset and collects 238K new perception questions. These questions correspond to the set of perceptual tasks needed to effectively answer complex reasoning questions in the Reasoning split. In total, VQA-Introspect contains 167K sub-questions for 56K reasoning questions in VQA v2 train, and 72K sub-questions for 22K reasoning questions in VQA v2 val. In our experiments, we follow (Selvaraju et al., 2020) to evaluate models trained with VQA v1 (Antol et al., 2015) train split on VQA-Introspect val split.

GQA (Hudson and Manning, 2019a) contains 22M

[10] https://pytorch.org/
[11] https://github.com/NVIDIA/apex

Task	Model	Training Steps	$p^{\text{img}}_{\text{mask}}$	$p^{\text{txt}}_{\text{mask}}$	Adv. Noise Lr.	kl-div loss weight β	Adv. Noise Retrain steps	Adv. Noise Retrain Lr.
VQA-Rephrasings	MANGO$_{\text{B}}$	4000	0.15	0.15	1e-5	1.0	400	1e-4
	MANGO$_{\text{L}}$	3000	0.15	0.30	5e-6	1.0	400	5e-5
VQA-LOL	MANGO$_{\text{B}}$	4000	0.15	0.45	1e-5	1.0	400	1e-4
	MANGO$_{\text{L}}$	4000	0	0.45	5e-6	1.0	400	5e-5
VQA-Introspect	MANGO$_{\text{B}}$	2000	0	0.15	1e-5	1.0	400	1e-4
	MANGO$_{\text{L}}$	3000	0	0.45	5e-6	1.0	400	5e-5
GQA	MANGO$_{\text{B}}$	4000	0.15	0.15	1e-5	1.0	800	1e-4
	MANGO$_{\text{L}}$	4000	0.15	0.15	1e-5	1.0	800	1e-4
VQA CP v2	MANGO$_{\text{B}}$	3000	0.15	0.6	1e-5	0	400	1e-4
	MANGO$_{\text{L}}$	3000	0.15	0.6	1e-6	0	400	1e-5
GQA-OOD	MANGO$_{\text{B}}$	4000	0.15	0.15	1e-5	1.0	800	1e-4
	MANGO$_{\text{L}}$	2000	0.15	0.15	1e-5	1.0	800	1e-4
IV-VQA &CV-VQA	MANGO$_{\text{B}}$	4000	0.15	0.45	1e-5	1.0	400	1e-4
	MANGO$_{\text{L}}$	3000	0.15	0.15	5e-6	1.0	400	5e-5
VQA v2	MANGO$_{\text{B}}$	6000	0.15	0.45	1e-5	1.0	400	1e-4
	MANGO$_{\text{L}}$	5000	0.15	0.45	1e-5	1.0	400	1e-4

Table 13: Hyper-parameter values used in our experiments. We use batch size of 5120 (3072) and gradient accumulation steps of 5 (8) for base (large) model experiments.

automatically generated questions based on ground-truth image scene graphs. The questions are constructed via a set of heuristic rules, which are designed to evaluate a VQA model in terms of different types of reasoning skills (*e.g.*, spatial understanding and multi-step inference). We follow (Hudson and Manning, 2019b) to use the balanced version of GQA, which has been designed to reduce biases in answer distribution. In the balanced version, 1.7M questions are split into 70%/10%/10% for training, validation and test sets, respectively. In our experiments, models are trained on GQA train split and we report performance on test-dev split.

IV-VQA & CV-VQA (Agarwal et al., 2020) are two synthetic datasets, created by removing objects in the real VQA images. In IV-VQA, irrelevant objects are erased and model predictions before and after image manipulations are expected to be invariant. In CV-VQA, which focuses on counting questions, one relevant object is removed from the given image and model predictions on the quantity of such object are expected to be subtracted by 1. Objects of choice are based on heuristic rules and removed via inpainter-GAN (Shetty et al., 2018). In total, 376K and 13K image-question pairs are generated for IV-VQA and CV-VQA, respectively. The detailed splits can be found in Table 1. In our experiments, we follow (Agarwal et al., 2020) to evaluate models trained with VQA v2 train split on IV-VQA/CV-VQA val split.

VQA-CP v2 (Agrawal et al., 2018) is an out-of-distribution (OOD) reorganization of VQA v2. It was created to examine the robustness of a VQA model in a setting where language priors cannot be relied upon for a correct prediction. The questions in VQA v2 are first assigned to one of 65 question types according to their prefix (first few words). For every question type, the prior distribution of answers is shuffled to be different in train and test splits of VQA-CP v2. Our models are trained on VQA-CP v2 train split and evaluated on test split, following (Agrawal et al., 2018).

GQA-OOD (Kervadec et al., 2020) is also an OOD benchmark, created by re-organization of the GQA dataset. By utilizing fine-grained question generation templates in GQA, GQA-OOD divides questions into 37K local groups, and shifts answer distribution by selecting a subset of answer classes for each question group, according to their frequencies. Unlike VQA-CP v2, GQA-OOD features distribution shifts for both validation and test, allowing to validate models under OOD conditions. In our experiments, we follow (Kervadec et al., 2020) to evaluate models trained with GQA train split on GQA-OOD test-dev split.

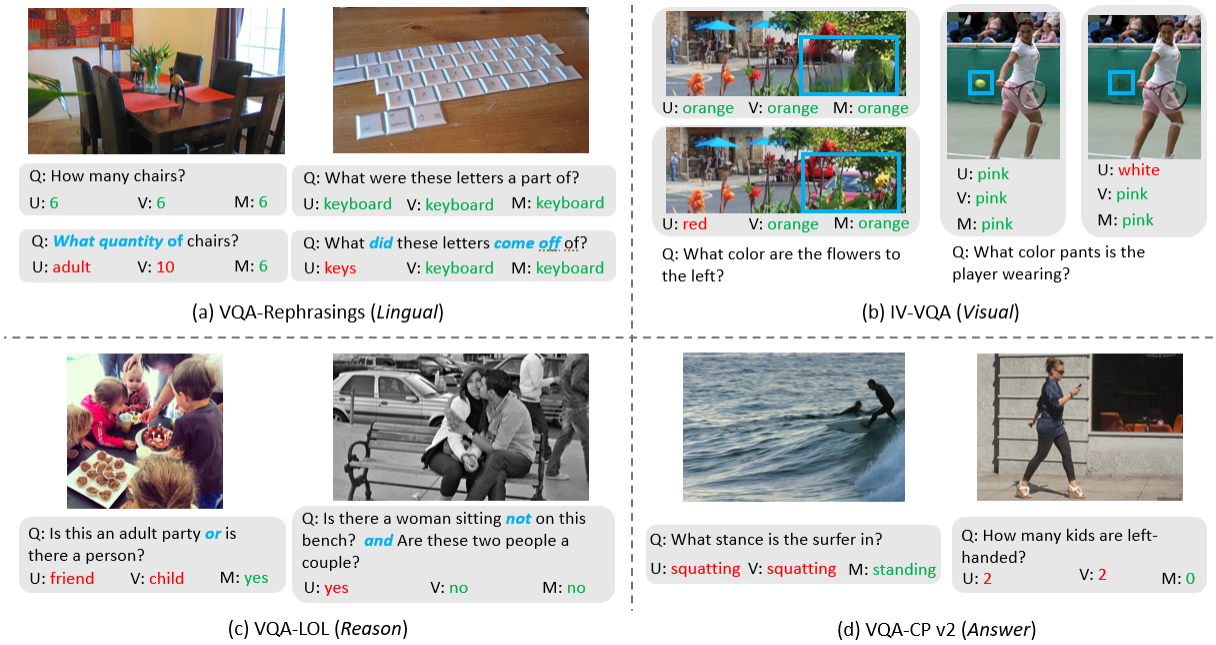


Figure 3: More visualization of model predictions, comparing MANGO (M) against UNITER (U) and VILLA (V). Correct answers are highlighted in green and wrong ones are in red.

E More Visualizations

We provide additional visualization of model predictions in Figure 3. MANGO consistently provides accurate predictions for each robustness type.

F Limitation and Broader Impact

A truly robust VQA system offers the possibility to be applied to real-life scenarios such as a chatbot that assists visually impaired people. In this paper, we aim to improve the robustness of VQA models, specifically the model performance on 9 robust VQA benchmarks. While our method outperforms the previous state-of-the-arts, the model does not always guarantee a perfect prediction. Like any other data-driven system, our method is sensitive to the distribution of training data, therefor may fail when encountering VQA examples in the wild.