Abstract

Recent Visual Question Answering (VQA) models have shown impressive performance on the VQA benchmark but remain sensitive to small linguistic variations in input questions. Existing approaches address this by augmenting the dataset with question paraphrases from visual question generation models or adversarial perturbations. These approaches use the combined data to learn an answer classifier by minimizing the standard cross-entropy loss. To more effectively leverage the augmented data, we build on the recent success in contrastive learning. We propose a novel training paradigm (ConCAT) that alternately optimizes cross-entropy and contrastive losses. The contrastive loss encourages representations to be robust to linguistic variations in questions while the cross-entropy loss preserves the discriminative power of the representations for answer classification. We find that alternately optimizing both losses is key to effective training. VQA models trained with ConCAT achieve higher consensus scores on the VQA-Rephrasings dataset as well as higher VQA accuracy on the VQA 2.0 dataset compared to existing approaches across a variety of data augmentation strategies.

1 Introduction

Visual Question Answering (VQA) refers to the task of automatically answering free-form natural language questions about an image. For VQA systems to work reliably when deployed in the wild for applications such as assisting visually impaired users, they need to be robust to different ways a user might ask the same question. For example, VQA models should produce the same answer for two paraphrased questions—“What is in the basket?” and “What is contained in the basket?”—since their semantic meaning is the same. While significant progress has been made towards building more accurate VQA systems, these models are still sensitive to minor linguistic variations in the input question.

To make VQA systems more robust, existing approaches (Shah et al. 2019; Tang et al. 2020) train state-of-the-art VQA systems (Jiang et al. 2018) on augmented data which includes different variations of the input question. For instance, VQA-CC (Shah et al. 2019) use a visual question generation (VQG) model to generate paraphrased question given an image and an answer. These models fuse image and question features into a joint vision and language (V+L) representation followed by a standard softmax classifier to produce answer probabilities. The models are optimized by minimizing the standard cross-entropy loss. Cross-entropy loss treats every image-question pair independently and fails to exploit the information that some questions in the augmented dataset are paraphrases of each other.

We overcome this limitation by using a contrastive loss InfoNCE (Oord, Li, and Vinyals 2018) that encourages joint V+L (Vision and Language) representations obtained from samples whose questions are paraphrases of each other to be closer while pulling apart the V+L representations of samples with different answers. Since we operate in a supervised setting, we choose Supervised Contrastive Loss (SCL) (Khosla et al. 2020) which extends InfoNCE to utilize label information by bringing samples from the same class together. We introduce a variant of the SCL loss which emphasizes on rephrased image-question pairs over pairs that are entirely different but have the same answer. Our pro-
posed training paradigm (ConCAT) alternately minimizes the SCL loss with cross-entropy loss to learn better vision and language representations as shown in Fig[1]. Minimizing the contrastive loss encourages representations to be robust to linguistic variations in questions while the cross-entropy loss preserves the discriminative power of the representations for answer classification. Instead of pretraining with SCL loss, then fine-tuning with cross-entropy loss as in {Khosla et al. 2020}, we find that minimizing the two losses alternately by constructing loss-specific mini-batches helps learn better representations. For the contrastive loss, we carefully curate these loss-specific mini-batches by sampling various types of negatives and positives given a reference.

We show the efficacy of our training paradigm across two rephrasing or data-augmentation strategies. Using rephrasings obtained from a VQG model proposed in (Shah et al. 2019), our approach outperforms a baseline that simply treats these rephrasings as additional samples and ignores the link between question and its paraphrases. We noticed that VQG model fails to produce a diverse set of rephrasings for a question and experiment with using back-translation to obtain question rephrasings. Back-translation (Edunov et al. 2018) involves translating an input sentence from one language to another and then translating it back into the original language using a pair of machine translation models (e.g. en-fr and fr-en). Back-translation preserves the semantic meaning of the question but generates syntactically diverse question paraphrases. Utilizing the publicly available collection of neural machine translation models in HuggingFace (Wolf et al. 2019), we generate 88 rephrasings of a question using MT model pairs for 88 different languages. We use a sentence similarity model (Reimers and Gurevych 2019) to filter irrelevant rephrasings and obtain 3 rephrasings per original question of VQA v2.0 dataset without any manual supervision.

We perform ablations to show that optimizing the contrastive and cross-entropy loss alternately helps compared to jointly optimizing them or pretraining with contrastive loss and then finetuning with cross-entropy approach in {Khosla et al. 2020}. We also show that curating batches with our proposed negative sampling strategy further boosts performance. On the VQA Rephrasings benchmark which measures the model’s answer consistency across several rephrasings of a question, ConCAT improves Consensus Score (Shah et al. 2019) by 1.63% over an improved baseline. In addition, on the standard VQA 2.0 benchmark, we improve the VQA accuracy by 0.78% overall. It is also worth noting that under our training paradigm (ConCAT), VQA models show better performance than existing approaches across both the aforementioned data-augmentation strategies.

2 Related Work

Models for VQA. Several models have been proposed for Visual Question Answering which fuse CNN grid features and LSTM features with different forms of attention (Lu et al. 2016; Yang et al. 2013; Fukui et al. 2016; Jiang et al. 2020). Bottom-Up and Top-Down (Anderson et al. 2017) proposed to learn attention over object regions obtained from a pretrained object detector and subsequent works (Kim, Jun, and Zhang 2018; Yu et al. 2018; Jiang et al. 2018) came up with various ways to fuse image and language representations. Recent works (Lu et al. 2019; Lu et al. 2020; Su et al. 2019; Li et al. 2020; Su et al. 2019; Tan and Bansal 2019; Chen et al. 2019) use multi-modal transformers to learn visuo-linguistic representations from object detector features and BERT question features (Devlin et al. 2018). We borrow the multi-modal transformer architecture from UNITER (Chen et al. 2019) for all our experiments.

Robustness of VQA Models. Robustness of VQA models with respect to multi-modal vision and language input has been studied in great detail. (Goyal et al. 2016; Zhang et al. 2015) proposed balanced datasets to build models that are sensitive to image modality and don’t overfit to language modality for answering visual questions. C-VQA (Agrawal et al. 2017) and VQA-CP (Agrawal et al. 2017a) datasets were proposed to test robustness against changing question-answer distributions. SQUINT (Selvaraju et al. 2020) encouraged consistency between reasoning questions and associated sub-questions. Our work focuses on robustness to question paraphrases introduced in VQA-Rephrasings (Shah et al. 2019) dataset collected from human annotators. VQA-CC (Shah et al. 2019) trained a Visual Question Generation (VQG) model to generate paraphrases of questions to augment the training dataset while VQA-Aug (Fang et al. 2020) augmented the training dataset by generating paraphrases of questions via back-translation. We show that these data augmentation techniques can be better utilized via our proposed training paradigm to build more robust and accurate VQA models.

Paraphrase Generation in NLP. There has been significant work in the area of Natural Language Processing (NLP) for generating paraphrases of a sentence using LSTM networks (Prakash et al. 2016). Deep Reinforcement Learning (Li et al. 2017), Variational Autoencoders (Gupta et al. 2017) and Transformers (Wang et al. 2019). However, these works require supervision in form of paraphrase pairs. In order to mitigate this limitation of labelled data, Neural Machine Translation (NMT) models have been used to generate paraphrases in a self-supervised fashion via Back-Translation (Mallinson, Sennrich, and Lapata 2017; Wieting, Mallinson, and Gimpel 2017). We build on top of these works and use state-of-the-art NMT models from HuggingFace (Wolf et al. 2019) to generate paraphrases for visual questions without any supervision.

Contrastive Learning. There has been recent surge in the use of Contrastive Learning for learning visual representations in a self-supervised manner (Wu et al. 2018; Heäaff et al. 2019; He et al. 2020; He et al. 2020a; Chen et al. 2020a; Chen et al. 2020b). Recently, contrastive learning has also been used for phase grounding (Gupta et al. 2020). (Gupta et al. 2020) used InfoNCE loss (Oord, Li, and Wierstra 2018) to learn a compatibility function between a set of region features from an image and contextualized word representations. In contrast, we want to learn representations which are robust to linguistic variations in the question.
for VQA. To utilize label information in contrastive losses, (Khosla et al. 2020) proposed Supervised Contrastive Learning (SCL) loss for learning visual representations. We use a variant of the SCL loss which scales the contributions from augmented positive samples (rephrasings in our case) over intra-class positive samples (that have the same answer) using a scaling factor. Moreover, our training paradigm alternately optimizes the two losses, whereas (Khosla et al. 2020) follow the traditional pretrain-finetune training regimen. Furthermore, (Khosla et al. 2020) randomly sample positive and negative pairs based on label information for contrastive learning whereas we carefully curate loss specific batches by sampling hard-negatives from the dataset. We show how these differences affect performance through a series of ablations in the experiments section.

3 Preliminaries

In this section, we introduce the VQA task and the standard cross entropy training of VQA models. We then recap contrastive methods for learning representations (Chen et al. 2020a) and the recently proposed Supervised Contrastive Learning (SCL) (Khosla et al. 2020) setup. We describe our proposed approach in the next section.

VQA. The task of Visual Question Answering (VQA) (Agrawal et al. 2015; Goyal et al. 2016) involves predicting an answer a for a question q about an image v. An instance of this problem in the VQA Dataset D is represented via a tuple x = (v, q, a), ∀x ∈ D. Recent VQA models (Jiang et al. 2018; Anderson et al. 2017; Chen et al. 2019) take image and question as input and outputs a joint vision and language (V+L) representation h ∈ R^{d_x} using a multi-modal network f:

\[ h = f(v, q) \]

The V+L representation h is then used to predict a probability distribution over the answer space A with a softmax classifier \( f^a(h) \) learned by minimizing the cross-entropy:

\[
\mathcal{L}_{CE} = -\log \frac{\exp(f^a(h)[a])}{\sum_{a' \in A} \exp(f^a(h)[a'])} \tag{1}
\]

where \( f^a(h)[a] \) is the logit corresponding to the answer a.

Contrastive Learning. Recent works in vision (Chen et al. 2020a) have used contrastive losses to bring representations of two augmented views of the same image (called positives) closer together while pulling apart the representations of two different images (called negatives). In this, the representation h obtained from an image encoder is projected into a \( d_z \)-dimensional hyper-sphere using a projection network g such that \( z = g(h) \in R^{d_z} \). Given a mini-batch of size K, the image representation h is learned by minimizing the InfoNCE (Oord, Li, and Vinyals 2018) loss which operates on a pair of positives \((z_i, z_p)\) and \( K - 1 \) negative pairs \((z_i, z_k)\) such that \( i, p, k \in [1, K], k \neq i \) as follows:

\[
\mathcal{L}_{NCE}^i = -\log \frac{\exp(\Phi(z_i, z_p)/\tau)}{\sum_{k=1}^{K} \mathbb{1}_{k \neq i} \exp(\Phi(z_i, z_k)/\tau)}, \tag{2}
\]

where \( \Phi(u, v) = u^\top v / \|u\|\|v\| \) computes similarity between u and v and \( \tau > 0 \) is a scalar temperature parameter.

A generalization of InfoNCE loss to handle more than one positive-pair was proposed by (Khosla et al. 2020) called Supervised Contrastive Loss (SCL). Given a reference sample \( x \), SCL uses class-label information to form a set of positives \( \mathcal{X}^+(x) \) that contains samples with the same label as \( x \). \( \mathcal{X}^+(x) \) also contains augmented views of the sample because they share the same label as \( x \). For a minibatch with \( K \) samples, SCL is defined as:

\[
\mathcal{L}_{SC}^i = -\sum_{p=1}^{\lvert \mathcal{X}^+(x_i) \rvert} \log \frac{\exp(\Phi(z_i, z_p)/\tau)}{\sum_{k=1}^{K} \mathbb{1}_{k \neq i} \exp(\Phi(z_i, z_k)/\tau)} \tag{3}
\]

Overall, \( \mathcal{L}_{SC}^i \) tries to bring the representation of samples in \( \mathcal{X}^+(x_i) \) closer together compared to representations of samples with a different ground-truth label.

4 Approach

We now describe our proposed training paradigm, ConCAT, that uses alternate contrastive and cross-entropy training to learn VQA models robust to question paraphrases.

4.1 Augmented Dataset with Back Translation

We first augment the training dataset with question paraphrases. We use 88 different MarianNMT (Junczys-Dowmunt et al. 2018) back translation model pairs released by HuggingFace Transformers (Wolf et al. 2019) to generate question paraphrases. We use Sentence-BERT (Reimers and Gurevych 2019) to filter out paraphrases that have \( \geq 0.95 \) cosine similarity with the original question and choose three unique paraphrases randomly from the filtered set.

For a sample \( x = (v, q, a) \in D \), let’s denote a set of paraphrases for question q by \( \mathcal{Q}(q) \) and the corresponding set of VQA triplets as:

\[
\mathcal{X}^+_{\text{para}}(x) = \{(v, q', a) \mid q' \in \mathcal{Q}(q)\} \tag{4}
\]

As shown in Figure 2(a), we augment the VQA dataset \( D \) with multiple paraphrased samples of a given question and denote the augmented dataset \( D^{\text{aug}} \) as:

\[
D^{\text{aug}} = D \bigcup_{x \in D} \mathcal{X}^+_{\text{para}}(x) \tag{5}
\]

4.2 Scaled Contrastive Loss for VQA

We would like our VQA model to produce the same and correct answer for a question and its paraphrase given an input image. This motivates us to map joint vision and language (V+L) representations of an original and paraphrased sample closer to each other. Moreover, since we operate in a supervised setting, following SCL (Khosla et al. 2020) we also pull the joint representations for the questions with the same answer (intra-class positives) closer together while pulling apart the representations of questions with different answers.
Figure 2: **Overview of ConCAT.** (a) We augment the VQA dataset by paraphrasing every question via back-translation. (b) We carefully curate a contrastive batch by sampling different types of positives and negatives to learn joint V+L representations by minimizing scaled supervised contrastive loss $L_{SSC}$. (c) Cross Entropy loss $L_{CE}$ is alternately optimized with $L_{SSC}$.

We define the set of all samples with the same ground truth answer as $x$ by:

$$X^+(x) = \{(\hat{v}, \hat{q}, \hat{a}) \in D_{aug} \mid \hat{a} = a\}$$

(6)

Note that $X^+_{para}(x) \subset X^+(x)$ as all question paraphrases have the same answer for a given image but not all questions with the same answer are paraphrases. We refer to samples in set $X^+_{cls}(x) = X^+(x) - X^+_{para}(x)$ as *intra-class positives* and set $X^+_{para}(x)$ as *paraphrased positives* w.r.t. $x$ as depicted in Figure 2(b).

Following Eq. (3), all the samples in $X^+(x_i)$ in $L_{SSC}$ are treated the same. That is, representations from both the paraphrased positives and intra-class positives are brought closer together. To emphasize on the link between question and its paraphrase, we propose a variant of the SCL loss in Eq. (7) which assigns higher weight to paraphrased positives $X^+_{para}(x_i)$ over intra-class positives $X^+_{cls}(x_i)$. We introduce a scaling factor $\alpha_{ip}$ in the SCL loss (Eq. (3)) for a sample $x_i$ as follows:

$$L_{SSC} = - \sum_{p=1}^{K} \alpha_{ip} \cdot \log \left(\frac{\exp(\Phi(z_i, z_p)/\tau)}{\sum_{k=1}^{K} \delta_{k \neq i} \cdot \exp(\Phi(z_i, z_k)/\tau)}\right)$$

(7)

$$L_{SSC} = \sum_{i=1}^{K} \frac{L_{SSC}}{\sum_{p} \alpha_{ip}}$$

(8)

The scaling factor $\alpha_{ip}$ assigns a higher weight $s > 1$ to positive samples corresponding to question paraphrases compared to other intra-class positives. Intuitively, because of the higher weight, the loss will penalize the model strongly if it fails to bring the representations of a question and its paraphrase closer. We define $\alpha_{ip}$ as:

$$\alpha_{ip} = \begin{cases} s & \text{if } x_p \in X^+_{para}(x_i), \\ 1 & \text{otherwise} \end{cases}$$

(9)

### 4.3 Alternate Training

We train our model by alternating between contrastive $L_{SSC}$ and the cross-entropy $L_{CE}$ iterations. Our training paradigm is summarized in Algorithm 1. Specifically, given $N$ total training iterations, we update our model with $L_{SSC}$ after every $N_{ce} - 1$ updates with $L_{CE}$, where $N_{ce}$ is a hyperparameter. Our overall loss for the $n^{th}$ iteration $\forall n \in [1, N]$ can be written as:

$$L_n = \mathbb{I}_{n \mod N_{ce}=0} \cdot L_{SSC} + \mathbb{I}_{n \mod N_{ce}\neq0} \cdot L_{CE}$$

(10)

Alternately training the model with the two losses simplifies the optimization procedure compared to two-stage training (pretraining-then-finetuning as in (Khosla et al. 2020)) which requires double the hyper-parameters and training iterations. We also experiment with joint-training using a linear combination of the $L_{SSC}$ and $L_{CE}$ losses. The key drawback of this paradigm is that it does not allow for loss-specific curation of batches. Due to this, $L_{CE}$ is forced to operate with batches created for $L_{SSC}$ (built with positives and negatives). Empirically, we show that alternate training works better than both joint-training as well as pretraining-then-finetuning approach. Figure 2 (b),(c) parts depict our training strategy (ConCAT).
Algorithm 1 ConCAT with $L_{SSC}$ and $L_{CE}$

**input:** steps $N$; constant $N_{ce}$; data $D^{aug}$; networks $f, g$

**for all** $i \in \{1, \ldots, N\}$

$B = \emptyset$

if $(i \mod N_{ce}) = 0$

# sscl iteration

$B = \text{CURATE}(N_{r}, D^{aug}, w); L = L_{SSC}$

else do

# ce iteration

$B \sim D^{aug}; L = L_{CE}$

update $f(.)$, $g(.)$ networks to minimize $L$ over $B$

return network $f(.)$; throw away $g(.)$

Algorithm 2 Batch Curation Strategy for $L_{SSC}$

**input:** number of references $N_{r}$; data $D$; weights $w$

**function** CURATE($N_{r}, D, w$)

$B = \emptyset, B_{r} = \emptyset$

# initialize batches

**for all** $i \in \{1, \ldots, N_{r}\}$

$x_{i} \sim D$

$\bar{x}_{i} \sim \mathcal{X}_{c_{neg}}^{+}(x_{i})$ # intra-class positive

t $\sim \text{Cat}(T|w)$ # negative type

$\bar{x}_{i} \sim \mathcal{X}_{t_{neg}}^{-}(x_{i})$ # negative

append $B = B \cup \{x_{i}, \bar{x}_{i}, \bar{t}_{i}\}$

**for all** $i \in \{1, \ldots, |B|\}$

$x'_{i} \sim \mathcal{X}_{para}^{-}(x_{i})$ # paraphrased positive

append $B_{r} = B_{r} \cup \{x'_{i}\}$

return $B \cup B_{r}$

4.4 Negative Types and Batch Creation

Any contrastive loss such as SCL operates with multiple negative samples. For a given reference sample $x = (v, q, a) \in D^{aug}$, we define a corresponding set of negatives as samples with ground truth different than the reference $x$:

$$
\mathcal{X}^{-}(x) = \{(\hat{v}, \hat{q}, \hat{a}) \in D^{aug} | \hat{a} \neq a\}
$$

We carefully curate batches for $L_{SSC}$ by sampling different types of negatives. Given $\hat{x} = (\hat{v}, \hat{q}, \hat{a}) \in \mathcal{X}^{-}(x)$, we classify a negative pair $(x, \hat{x})$ into one of three negative categories from $T = \{\text{que}, \text{img}, \text{rand}\}$ via a mapping function $\Omega : D \times D \rightarrow T$ such that:

$$
\Omega(x, \hat{x}) = \begin{cases} 
\text{img} & \text{if } v = \hat{v} \\
\text{que} & \text{if } \text{sim}(q, \hat{q}) > \epsilon \\
\text{rand} & \text{else}
\end{cases}
$$

(11)

where $\epsilon$ is a similarity threshold. We partition $\mathcal{X}^{-}(x)$ in three mutually exclusive subsets $\mathcal{X}^{-}_{t}(x), t \in T$ defined as:

$$
\mathcal{X}^{-}_{t}(x) = \{\hat{x} \in \mathcal{X}^{-}(x) | \Omega(x, \hat{x}) = t\}
$$

(12)

- **Image Negatives**, $\mathcal{X}^{-}_{img}(x)$: Image negatives are samples that have the same image as the reference $(x)$ but different answer. Since VQA dataset has multiple questions ($\sim 5.4$) per image, finding image negatives is trivial.

- **Question Negatives**, $\mathcal{X}^{-}_{que}(x)$: Question negatives are samples with similiar questions and different answers. We measure the similarity between the questions by computing their cosine distance in the vector space of the Sentence-BERT (Reimers and Gurevych 2019) model.

- **Random Negatives**, $\mathcal{X}^{-}_{rand}(x)$: Random negatives are samples that do not fall under either Image or Question negative categories i.e. any image and question pair that has a different answer than the reference.

We hypothesize that discriminating between joint V+L representations of above negatives and the reference would lead to more robust V+L representations as it requires the model to preserve relevant information from both modalities in the learnt representation. Negative samples belonging to each of the above types are depicted in Figure 2b).

**Batch Curation.** To create mini-batches for $L_{SSC}$, as described in Algorithm [2] we start by filling our batch with triplets of reference $x_{i}$, a intra-class positive $\bar{x}_{i}$ and a negative sample $\bar{t}_{i}$ of type $t$. The negative type $t$ is sampled from a categorical distribution $\text{Cat}(T|w)$ where $w = (w_{img}, w_{que}, w_{rand})$ are the probability weights of selecting different types. This procedure is repeated for specified number of times $N_{r}$ to create a batch $B$. Finally, for every sample in $B$ we add a corresponding paraphrased positive $x'_{i}$ sample. When sampling all the three types of negatives we use $w = (w_{img}, w_{que}, w_{rand}) = (0.25, 0.25, 0.5)$. For $L_{CE}$, we randomly sample mini-batch from the dataset $D^{aug}$.

**Importance of Scaling Factor.** VQA Dataset has a skewed distribution of answer labels and since we sample references for SCL minibatch independently of each other (see Algorithm [2]) quite often we end up with multiple intra-class positives but only a single paraphrased positive for given a reference in a minibatch. To balance this trade-off we choose to scale the loss corresponding to paraphrased positive sample, we call this loss Scaled Supervised Contrastive Loss ($L_{SSC}$).

5 Experiments

5.1 Datasets and Metrics

We use the VQA v2.0 (Goyal et al. 2016) and the VQA-Rephrasings (Shah et al. 2019) datasets for experiments. VQA contains nearly 443K train, 214K val and 453K test instances. VQA-Rephrasings was collected to evaluate the robustness of VQA models towards human rephrased questions. Specifically, the authors collected 3 human-provided rephrasings for 40k image-question pairs from the VQA v2.0 validation dataset.

(Shah et al. 2019) also introduced Consensus Score (CS) as an evaluation metric to quantify the agreement of VQA models across multiple rephrasings of the same question. Amongst all subsets of paraphrased questions of size $k$, the consensus score $\text{CS(s)}(k)$ measures the fraction of subsets in which all the answers have non-zero VQA-Score. For a set
of paraphrases $Q$, the consensus score $CS(k)$ is defined as:

$$CS(k) = \sum_{Q' \subseteq Q; |Q'| = k} \frac{S(Q')}{C_k}$$  \hspace{1cm} (13)

$$S(Q') = \begin{cases} 1 & \text{if } \forall q \in Q', \text{ VQA-Score}(q) > 0, \\ 0 & \text{else} \end{cases}$$  \hspace{1cm} (14)

Where $C_k$ is the number of subsets of size $k$ sampled from a set of size $n$. $CS(k)$ is zero for a group of questions $Q$ when the model answers at least $k$ questions correctly.

When reporting results on the val split and VQA-Rephrasings, we train on the VQA 2.0 train split and when reporting results on the VQA 2.0 test-dev and test-std we train on both VQA 2.0 train and val splits. The VQA Rephrasings dataset (Shah et al. 2019) is never used for training and used only for evaluation.

### 5.2 Baselines and Training Details

**VQA Model.** For $f$, we use a multimodal transformer (MMT) inspired from (Chen et al. 2019), with 6 layers and 768-dim embeddings. It takes as input two different modalities. The question tokens are encoded using a pre-trained three layer BERT (Devlin et al. 2018) encoder which is fine-tuned along with the multimodal transformer. Object regions are encoded by extracting features from a frozen ResNeXt-152 (Xie et al. 2017) based Faster R-CNN model (Ren et al. 2015). The projection module $g$ consists of two linear layers and a L-2 normalization function.

**Question Paraphrases using VQG.** Apart from training with question paraphrases generated via back-translation, we also experiment with generating question paraphrases using the VQG module from (Shah et al. 2019). We input the VQG module with 88 random noise vectors to keep the generation comparable with Back-translation approach. For filtering, we use the gating mechanism used by the authors and sentence similarity score of $\geq 0.85$ and keep a maximum of 3 unique rephrasings for each question.

**Training Details.** We train our models using Adam optimizer (Kingma and Ba 2014) with a linear warmup and with a learning rate of 1e-4 and a staircase learning rate schedule, where we multiply the learning rate by 0.2 at 10.6K and at 15K iterations. We train for 5 epochs of train + augmented dataset on 4 NVIDIA Titan XP GPUs and use a batch-size of 420 when using $L_{ssc}$ and $L_{ce}$ both and 210 otherwise. We use the PyTorch (Paszke et al. 2019) for all the experiments. We set number of references $N_r = 70$, the scaling factor $s = 20$, the similarity threshold $c = 0.95$ and $N_{ce} = 4$.

**Existing state-of-the-art methods.** Previous work (Shah et al. 2019) in VQA-Rephrasings trained a VQA model using a cycle-consistent training scheme along with the VQA model. The approach involved generating questions by a VQG model such that the answer for the original and the generated question are consistent with each other. For their experiments, they build on top of Pythia (Jiang et al. 2018) and BAN (Anderson et al. 2017) as base VQA models. We treat these approaches as baselines for our experiments.

### 6 Results

In this section, we carefully analyze the importance of each component of our approach, and compare results with existing approaches (Pythia+CC, BAN+CC). In Table 1, we report evaluation on VQA-Rephrasings and VQA 2.0 dataset.

| Model                        | DA       | Scaling | N-Type | Consensus Score(k=1) | Consensus Score(k=2) | Consensus Score(k=3) | Consensus Score(k=4) | VQA Score val | VQA Score test-dev | VQA Score test-std |
|------------------------------|----------|---------|--------|-----------------------|-----------------------|-----------------------|-----------------------|---------------|---------------------|---------------------|
| Pythia (2018)                |          |         |        | 63.43                 | 52.03                 | 45.94                 | 39.49                 | 65.78         | 68.43               | -                   |
| BAN (2018)                   |          |         |        | 64.88                 | 53.08                 | 47.45                 | 39.87                 | 66.04         | 69.64               | -                   |
| Pythia + CC (2019)           |          |         |        | 64.36                 | 55.45                 | 50.92                 | 44.30                 | 66.03         | 68.88               | -                   |
| BAN + CC (2019)              |          |         |        | 65.77                 | 56.94                 | 51.76                 | 48.18                 | 66.77         | 69.87               | -                   |
| MMT + CE                     |          |         |        | 67.74                 | 59.82                 | 55.10                 | 51.82                 | 66.46         | -                   | -                   |
| MMT + CE VQG                 |          |         |        | 66.53                 | 59.26                 | 54.92                 | 51.85                 | 64.50         | -                   | -                   |
| MMT + ConCAT VQG             | ✓        |         | RQI    | 66.49                 | 59.55                 | 55.33                 | 52.31                 | 64.74         | -                   | -                   |
| MMT + CE BT                  |          |         |        | 67.58                 | 60.04                 | 55.53                 | 52.36                 | 66.31         | 69.51               | 69.22               |
| MMT + (SCL→CE) BT            |          |         | R      | 65.34                 | 57.39                 | 52.63                 | 49.20                 | 64.21         | -                   | -                   |
| MMT + (CE + SCL) BT          |          |         | R      | 66.95                 | 59.70                 | 55.32                 | 52.20                 | 65.10         | -                   | -                   |
| MMT + ConCAT BT              |          |         | R      | 68.35                 | 60.97                 | 56.49                 | 53.30                 | 66.73         | -                   | -                   |
| MMT + ConCAT BT              | ✓        |         | R      | 68.19                 | 60.92                 | 56.53                 | 53.42                 | 66.62         | -                   | -                   |
| MMT + ConCAT BT              | ✓        |         | RQ     | 68.41                 | 61.24                 | 56.88                 | 53.77                 | 66.97         | -                   | -                   |
| MMT + ConCAT BT              | ✓        |         | RI     | 68.47                 | 61.28                 | 56.91                 | 53.79                 | 66.93         | -                   | -                   |
| MMT + ConCAT BT              | ✓        |         | RQI    | 68.20                 | 60.90                 | 56.49                 | 53.36                 | 66.60         | -                   | -                   |
| MMT + ConCAT BT              | ✓        |         | RQI    | 68.62                 | 61.42                 | 57.08                 | 53.99                 | 66.98         | 69.80               | 70.00               |

Table 1: Evaluation on VQA-Rephrasings and VQA 2.0 dataset. DA denotes the source of augmented data from either Back Translation (BT) or Visual Question Generation (VQG). N-Type defines the type of negatives used from Image (1), Question (2) and Random (3). Scaling denotes whether scaling factor $\alpha$ (defined in Eq. 9) was used. When reporting on test-dev and test-std we train our model on train-val set of VQA 2.0 dataset.
Figure 3: Qualitative Examples. Predictions of ConCAT and MMT+CE baseline on several image-question pairs and their corresponding rephrased questions. Average Consensus Scores (k=1-4) are also shown at the bottom (higher the better).

port the Consensus Score (CS(k)) for k = [1, 4] on VQA-Rephrasings (Shah et al. 2019) and VQA Accuracy on VQA 2.0 (Goyal et al. 2016) datasets.

Alternate Training. We find that alternate training (ConCAT) with contrastive loss (LSC) and cross-entropy (LCE) (Row 11) performs better on both Consensus Scores scores and VQA Accuracy compared to training with just LCE (Row 8). Following the approach taken in (Khosla et al. 2020), we try pre-training the model with LSC and then fine-tuning the model on LCE (Row 9) and find that training alternately (ConCAT) works better. Furthermore, joint-training of both the losses (LSC + LCE) together (Row 10) performs worse than just using LCE (Row 8).

Scaled Supervised Contrastive Loss (LSSC). Compared to Supervised Contrastive Loss LSC (Khosla et al. 2020) (Row 15), we also see improvement on both VQA Accuracy and Consensus Scores when using our proposed variant Scaled Supervised Contrastive Loss LSSC (Row 16). We find that using LSSC is more effective when used with contrastive batches (Row 16 vs Row 12).

Negative Sampling Strategy. Furthermore, we find that our proposed negative sampling strategy (Algorithm 2) where we carefully curate batches for LSSC loss (Row 16) helps improve consensus score over random-sampling (Row 12). We also find that adding either que-type negatives (Row 13) or img-type negatives (Row 14) improves consensus scores over random negative sampling (Row 12).

ConCAT with paraphrases from VQG model. Similar to improvements seen by using ConCAT with BT data, we see improvements when using the VQG model from (Shah et al. 2019) to generate paraphrases. Although, we find the quality and diversity of paraphrases generated by the VQG module to be poor, we show that using ConCAT with VQG rephrasings (Row 7) leads to gains on both VQA and Consensus Scores over using data-augmentation with LCE (Row 6).

Comparison with existing methods. We compare with existing state-of-the-art approaches from (Shah et al. 2019) on VQA rephrasings dataset and find that ConCAT (Row 16) outperforms state-of-the-art approaches Pythia+CC (Row 3) and BAN+CC (Row 4) on all Consensus Scores by large margins while showing on-par performance on VQA scores.

Qualitative Analysis. We qualitatively visualize few samples in Figure 3. We compare our final approach (Row 16) with MMT + CE (Row 8). As evident from samples, ConCAT improves the consistency in answers across the rephrasings. (2, 2) shows an interesting example where ConCAT yields a singular answer for one question paraphrase and produces the original plural answer for other paraphrased question. In (2,3), baseline incorrectly answers the original VQA question but correctly answers some of the rephrasings whereas our approach gets all the questions right. (2,4) illustrates a failure case where both the approaches fail to answer all the paraphrased questions correctly.

7 Conclusion

To summarize, we have three main contributions. First, we propose a novel training paradigm (ConCAT) that involves
alternate training of Contrastive and Cross-entropy losses to learn joint vision and language representations that are robust to question paraphrases. Minimizing the contrastive loss encourages representations to be robust to linguistic variations in questions while the Cross-entropy loss preserves the discriminative power of the representations for answer classification. Second, we introduce Scaled Supervised Contrastive Loss (SSCL), that assigns higher weight to positive samples associated with question paraphrases over samples that have the same answer boosting the performance further. Finally, we propose a negative sampling strategy to curate loss-specific batches which improves performance over random sampling strategy. Compared to previous approaches, VQA models trained with ConCAT achieve higher consistency scores on the VQA-Rephrasings dataset as well as higher VQA accuracy on the VQA 2.0 dataset across a variety of data augmentation strategies.

8 Acknowledgements

We thank Abhishek Das, Prithvijit Chattopadhyay and Arjun Majumdar for their feedback. The Georgia Tech effort was supported in part by NSF, AFRL, DARPA, ONR YIPs, ARO PECASE, Amazon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

References

[Agrawal et al. 2015] Agrawal, A.; Lu, J.; Antol, S.; Mitchell, M.; Zitnick, C. L.; Batra, D.; and Parikh, D. 2015. Vqa: Visual question answering.

[Agrawal et al. 2017a] Agrawal, A.; Batra, D.; Parikh, D.; and Kembhavi, A. 2017a. Don’t just assume; look and answer: Overcoming priors for visual question answering.

[Agrawal et al. 2017b] Agrawal, A.; Kembhavi, A.; Batra, D.; and Parikh, D. 2017b. C-vqa: A compositional split of the visual question answering (vqa) v1.0 dataset.

[Anderson et al. 2017] Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2017. Bottom-up and top-down attention for image captioning and visual question answering.

[Chen et al. 2019] Chen, Y.-C.; Li, L.; Yu, L.; Kholy, A. E.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2019. UniTER: Learning universal image-text representations. arXiv preprint arXiv:1909.11740.

[Chen et al. 2020a] Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709.

[Chen et al. 2020b] Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. 2020b. Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029.

[Chen et al. 2020c] Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020c. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297.

[Devlin et al. 2018] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[Edunov et al. 2018] Edunov, S.; Ott, M.; Auli, M.; and Grangier, D. 2018. Understanding back-translation at scale.

[Fukui et al. 2016] Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multi-modal compact bilinear pooling for visual question answering and visual grounding. arXiv preprint arXiv:1606.01847.

[Goyal et al. 2016] 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.

[Gupta et al. 2017] Gupta, A.; Agarwal, A.; Singh, P.; and Rai, P. 2017. A deep generative framework for paraphrase generation. arXiv preprint arXiv:1709.05074.

[Gupta et al. 2020] Gupta, T.; Vahdat, A.; Chechik, G.; Yang, X.; Kautz, J.; and Hoiem, D. 2020. Contrastive learning for weakly supervised phrase grounding.

[He et al. 2020] He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9729–9738.

[Hénaff et al. 2019] Hénaff, O. J.; Srinivas, A.; De Fauw, J.; Razavi, A.; Doersch, C.; Eslami, S.; and Oord, A. v. d. 2019. Data-efficient image recognition with contrastive predictive coding. arXiv preprint arXiv:1905.09272.

[Jiang et al. 2018] Jiang, Y.; Natarajan, V.; Chen, X.; Rohrbach, M.; Batra, D.; and Parikh, D. 2018. Pythia v0.1: the winning entry to the vqa challenge 2018.

[Jiang et al. 2020] 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 et al. 2018] 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.

[Khosla et al. 2020] Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. ArXiv abs/2004.11362.

[Kim, Jun, and Zhang 2018] Kim, J.-H.; Jun, J.; and Zhang, B.-T. 2018. Bilinear attention networks. In Advances in Neural Information Processing Systems, 1564–1574.

[Kingma and Ba 2014] Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization.

[Li et al. 2017] Li, Z.; Jiang, X.; Shang, L.; and Li, H. 2017. Paraphrase generation with deep reinforcement learning. arXiv preprint arXiv:1711.00279.

[Li et al. 2020] Li, G.; Duan, N.; Fang, Y.; Gong, M.; Jiang, D.; and Zhou, M. 2020. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In AAAI, 11336–11344.

[Lu et al. 2016] Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical question-image co-attention for visual question answering.

[Lu et al. 2019] Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019.
Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In Advances in Neural Information Processing Systems, 13–23.

[Lu et al. 2020] Lu, J.; Goswami, V.; Rohrbach, M.; Parikh, D.; and Lee, S. 2020. 12-in-1: Multi-task vision and language representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10437–10446.

[Mallinson, Sennrich, and Lapata 2017] Mallinson, J.; Sennrich, R.; and Lapata, M. 2017. Paraphrasing revisited with neural machine translation. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 881–893.

[Oord et al. 2018] Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. ArXiv abs/1807.03748.

[Paszke et al. 2019] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. Pytorch: An imperative style, high-performance deep learning library. ArXiv abs/1912.01703.

[Prakash et al. 2016] Prakash, A.; Hasan, S. A.; Lee, K.; Datla, V.; Qadir, A.; Liu, J.; and Farri, O. 2016. Neural paraphrase generation with stacked residual lstm networks. arXiv preprint arXiv:1610.01847.

[Reimers and Gurevych 2019] Reimers, N., and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

[Shah et al. 2019] Shah, M.; Chen, X.; Rohrbach, M.; and Parikh, D. 2019. Cycle-consistency for robust visual question answering. In Proceedings of the IEEE conference on computer vision and pattern recognition, 6649–6658.

[Tang et al. 2019] Tang, R.; Ma, C.; Zhang, W. E.; Wu, Q.; and Yang, X. 2020. Semantic equivalent adversarial data augmentation for visual question answering.

[Wang et al. 2019] Wang, S.; Gupta, R.; Chang, N.; and Baldrige, J. 2019. A task in a suit and a tie: paraphrase generation with semantic augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 7176–7183.

[Wieting, Mallinson, and Gimpel 2017] Wieting, J.; Mallinson, J.; and Gimpel, K. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. arXiv preprint arXiv:1706.01847.

[Wolf et al. 2019] 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.
Appendix

A Ablations with Joint Training

In the joint training experiment (Table 1, Row 10) we use a weighing parameter ($\beta$) to combine the $L_{SC}$ and $L_{CE}$ losses. We ablate on the choice of weight ($\beta$) used, and we represent the overall loss in this experiment as:

$$L_{joint} = \beta L_{SC} + (1 - \beta) L_{CE}$$

We find that increasing $\beta$ leads to decreasing gains on Consensus Scores as shown in Table A. We also find that the VQA-Accuracy hits a sweet-spot at $\beta = 0.5$ and we use this configuration as our baseline.

| Model         | $\beta$ | CS(4)     | VQA Score val |
|---------------|---------|-----------|--------------|
| 1 MMT + CE    | 0.25    | 52.97     | 66.14        |
| 2 MMT + CE    | 0.50    | 52.36     | 66.31        |
| 3 MMT + CE    | 0.75    | 48.53     | 61.34        |
| 4 MMT + CE    | 0.90    | 40.68     | 51.03        |
| 5 MMT + Concat| -       | 53.99     | 66.98        |

Table A: Ablations on the choice of our hyper-parameter $\beta$ for joint training.

B Gradient Surgery of $L_{SC}$ and $L_{CE}$

To know whether the gradients of both the losses ($L_{SC}$ and $L_{CE}$) are aligned with each other during training, we follow the gradient surgery setup of (Yu et al. 2020) for multi-task learning. During joint-training, we take the dot-products of gradients from both the losses and plot them to see how well they are aligned i.e., whether the dot product is positive or negative. In Figure A we plot the un-normalized dot product between the gradients corresponding to $L_{CE}$ and $L_{SSC}$ losses. We find that except for initial few steps the gradients of both the losses are aligned (dot product is positive) and thus the updates are complementary with respect to each other.

C Training Details

All the models have $\sim$100M trainable parameters. We train our models using Adam optimizer (Kingma and Ba 2014) with a linear warmup and with a learning rate of 1e-4 and a staircase learning rate schedule, where we multiply the learning rate by $0.2$ at 10.6K and at 15K iterations. We train for 5 epochs of augmented train dataset $D_{aug}$ on 4 NVIDIA Titan XP GPUs and use a batch-size of 420 when using $L_{SC}$ and $L_{CE}$ both and 210 otherwise. We use the PyTorch (Paszke et al. 2019) for all the experiments. The hyperparameters are summarized in Table B.

D Augmented Data

Back Translation: We use 88 different Marian-NMT (Junczys-Dowmunt et al. 2018) back translation model pairs released by Hugging Face (Wolf et al. 2019) to generate question paraphrases. We use Sentence-BERT (Reimers and Gurevych 2019) to filter out paraphrases that cosine similarity of $\geq 0.95$ with the original question and choose three unique paraphrases randomly from the filtered set. After filtering duplicates we end up with 2.89 paraphrases per original question on average.

VQG: We use the VQG model introduced by previous work (Shah et al. 2019) that takes as input the image and answer to generate a paraphrased question. We input the VQG module with 88 random noise vectors to keep the generation comparable with Back-translation approach. For filtering, we use the gating mechanism used by the authors and sentence similarity score of $\geq 0.85$ and keep a maximum of 3 unique rephrasings for each question. After filtering duplicates we end up with only 0.96 paraphrases per original question on average. We generally find the quality of VQG paraphrases worse as compared to Back-Translated ones.

Evaluation: During training we evaluate our models using the back-translated rephrasings on a subset of questions from validation set which do not overlap with VQA-Rephrasings (Shah et al. 2019) dataset.

E Code and Result Files

We share the code for running the baseline and the best experiments (Table 1, Rows 8, 16). Please find the released code at: https://www.github.com/yashkant/concat-vqa
### Table B: Hyperparameter choices for models.

| #  | Hyperparameters                  | Value  | #  | Hyperparameters                  | Value  |
|----|----------------------------------|--------|----|----------------------------------|--------|
| 1  | Maximum question tokens          | 23     | 2  | Maximum object tokens            | 101    |
| 3  | $\mathcal{L}_{CE}:\mathcal{L}_{SSC}$ iterations ratio | 3:1    | 4  | Number of TextBert layers        | 3      |
| 5  | Embedding size                   | 768    | 6  | Number of Multimodal layers      | 6      |
| 7  | Multimodal layer intermediate size | 3072   | 8  | Number of attention heads        | 12     |
| 9  | Negative type weights ($\omega$)  | (0.25, 0.25, 0.5) | 10 | Multimodal layer dropout         | 0.1    |
| 11 | Similarity Threshold ($s$)        | 0.95   | 12 | Optimizer                        | Adam   |
| 13 | Batch size                       | 210/420| 14 | Base Learning rate               | 2e-4   |
| 15 | Warm-up learning rate factor     | 0.1    | 16 | Warm-up iterations               | 4266   |
| 17 | Vocabulary size                  | 3129   | 18 | Gradient clipping (L-2 Norm)     | 0.25   |
| 19 | Number of epochs                 | 5/20   | 20 | Learning rate decay              | 0.2    |
| 21 | Learning rate decay steps        | 10665, 14931 | 22 | Number of iterations             | 25000  |
| 23 | Projection Dimension ($R^{d_z}$) | 128    | 24 | Hidden Dimension ($R^{d_h}$)      | 3129   |

### F Qualitative Samples

Figures [B][C][D][E] show many more qualitative samples comparing the baseline and ConCAT.
Figure B: Qualitative Examples. Predictions of ConCAT and MMT+CE baseline on several image-question pairs and their corresponding rephrased questions. Average Consensus Scores (k=1-4) are also shown at the bottom (higher the better).
Figure C: Qualitative Examples. Predictions of ConCAT and MMT+CE baseline on several image-question pairs and their corresponding rephrased questions. Average Consensus Scores (k=1-4) are also shown at the bottom (higher the better).
Figure D: Qualitative Examples. Predictions of ConCAT and MMT+CE baseline on several image-question pairs and their corresponding rephrased questions. Average Consensus Scores (k=1-4) are also shown at the bottom (higher the better).
Figure E: Qualitative Examples. Predictions of ConCAT and MMT+CE baseline on several image-question pairs and their corresponding rephrased questions. Average Consensus Scores (k=1-4) are also shown at the bottom (higher the better).