Semantic Equivalent Adversarial Data Augmentation for Visual Question Answering

Ruixue Tang[1], Chao Ma[1⋆], Wei Emma Zhang[2], Qi Wu[2], and Xiaokang Yang[1]

[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
{alicetang, chaoma, xkyang}@sjtu.edu.cn
[2] University of Adelaide
{wei.e.zhang, qi.wu01}@adelaide.edu.au

Abstract. Visual Question Answering (VQA) has achieved great success thanks to the fast development of deep neural networks (DNN). On the other hand, the data augmentation, as one of the major tricks for DNN, has been widely used in many computer vision tasks. However, there are few works studying the data augmentation problem for VQA and none of the existing image based augmentation schemes (such as rotation and flipping) can be directly applied to VQA due to its semantic structure – an $\langle image, question, answer\rangle$ triplet needs to be maintained correctly. For example, a direction related Question-Answer (QA) pair may not be true if the associated image is rotated or flipped. In this paper, instead of directly manipulating images and questions, we use generated adversarial examples for both images and questions as the augmented data. The augmented examples do not change the visual properties presented in the image as well as the **semantic** meaning of the question, the correctness of the $\langle image, question, answer\rangle$ is thus still maintained. We then use adversarial learning to train a classic VQA model (BUTD) with our augmented data. We find that we not only improve the overall performance on VQAv2, but also can withstand adversarial attack effectively, compared to the baseline model. The source code is available at https://github.com/zaynmi/seada-vqa.

Keywords: VQA, Data Augmentation, Adversarial Learning

1 Introduction

Both computer vision and natural language processing (NLP) have made enormous progress on many problems using deep learning in recent years. Visual question answering (VQA) is a field of study that fuses computer vision and NLP to achieve these successes. The VQA algorithm aims to predict a correct answer to the given question referring to an image. The recent benchmark study [17] demonstrates that the performance of VQA algorithms hinges on the amount of training data. Existing algorithms can always benefit greatly from more training data. This suggests that data augmentation without manual annotations is

⋆ Corresponding author.

an intuitive attempt to improve the VQA performance, just like its success on the other deep learning applications.

Existing Data augmentation approaches enlarge the training dataset size by either data warping or oversampling [37]. Data warping transforms data and keeps their labels. Typical examples include geometric and color transformations, random erasing, adversarial training, and neural style transfer. Oversampling generates synthetic instances and adds them to the training set. Data augmentation has shown to be effective in alleviating the overfitting problem of DNNs [37]. However, data augmentation in VQA is barely studied due to the challenge of maintaining an $\langle image, question, answer \rangle$ triplet semantically correct. Neither geometric transform nor randomly erasing the image could preserve the answer. For example, when asking about *What is the position of the computer?*, *Is the car to the left or right of the trash can?*, flipping or rotating images results in the opposite answers. Randomly erasing the image associated with the question *How many ...?* would miss the number of objects. Such transforms need tailored answers which are unavailable. On the textual side, it is challenging to come up with generalized rules for language transformation. Universal data augmentation techniques in NLP have not been thoroughly explored. Therefore, it is non-trivial to explore the data augmentation technique to facilitate VQA.

Previous works have generated reasonable questions based on the image content [16] and the given answer [25], namely Visual Question Generation (VQG). However, a significant portion of the generated questions either have grammatical errors or are oddly phrased. In addition, they learn from the questions and images in the same target dataset, thus the generated data are drawn from the same distribution of the original data. Since the training and test data usually do not share the same distribution, the generated data could not help to relieve the overfitting.

In this paper, we propose to generate semantic equivalent adversarial examples of both visual and textual data as augmented data. Adversarial examples are strategically modified samples that could successfully fool the deep models to make incorrect predictions. However, the modification is imperceptible that keeps the semantics of data while driving the underlying distribution of adversarial examples away from that of the original data [41]. In our method, visual adversarial examples are generated by an un-targeted gradient-based attacker [24], and textual adversarial examples are paraphrases that could fool the VQA model (predicting a wrong answer) while keeping the questions semantically equivalent. The existence of adversarial examples not only reveals the limited generalization ability of ConvNets, but also poses security threats on the real-world deployment of these models.

We adversarially train the strong baseline method Bottom-Up-Attention and Top-Down (BUTD) [2] on VQAv2 dataset [13] with clean examples and adversarial examples generated on-the-fly. We regard the adversarial training as a regularizer acting in a period of training time. Experimental results demonstrate that our proposed adversarial training framework not only better boosts the model performance on clean examples than other data augmentation techniques, but

also improves the model robustness against adversarial attacks. Although there are few works studying the data augmentation problem for VQA [18,35,33,1], they merely generate either new questions or images. To our best knowledge, our work is the first to augment both visual and textual data in VQA.

To summarize, our major contributions are threefold:

- We propose to generate visual and textual adversarial examples to augment the VQA dataset. Our generated data preserve the semantics and explore the learned decision boundary to help improve the model generalization.
- We propose an adversarial training scheme that enables VQA models to take advantage of the regularization power of adversarial examples.
- We show that the model trained with our method achieves 65.16% accuracy on the clean validation set, beating its vanilla training counterpart by 1.84%. Moreover, the adversarially trained model significantly increases accuracy on adversarial examples by 21.55%.

2 Related Work

VQA. A large number of VQA algorithms have been proposed, including spatial attention [2,44,26,6], compositional approaches [4,3,14], and bilinear pooling schemes [10,20]. Spatial attention [2] is one of the most widely used methods for both natural and synthetic image VQA. A large portion of prior arts [19,46,29,31] are built upon the bottom-up top-down (BUTD) attention method [2]. We also choose the BUTD as our baseline VQA model. Instead of developing a more sophisticated answering machine, we propose a VQA data augmentation technique that can potentially benefit existing VQA methods since data is the fuel.

Data Augmentation. Compared to vision, a few efforts have been done on augmenting text for classification problems. Wei *et al.* [40] make a comprehensive extension for text editing techniques on NLP data augmentation and achieve gains on text classification. However, our study shows that it could degrade the model performance on the VQA task (see Section 4). Other works generate paraphrases [45,28] and add noise to smooth text data [42]. There are fewer works [18,33,35,1,30] that learn data augmentation for VQA. Kafle *et al.*[18] do a pioneer work where they generate new questions by using semantic annotations on images. Work of [33] automatically generates entailed questions for a source QA pair , but it uses additional data in Visual Genome [22] to add diversity to the generated questions. Work of [35] proposes a cyclic-consistent training scheme where it generates different rephrasings of question and train the model such that the predicted answers across the generated and original questions remain consistent. The method [1] employ a GAN-based re-synthesis technique to automatically remove objects to strengthen the model robustness against semantic visual variations. Note that all of these methods augment data in a single modality (text-only or image-only) and heavily rely on complex modules to achieve slight performance gains.

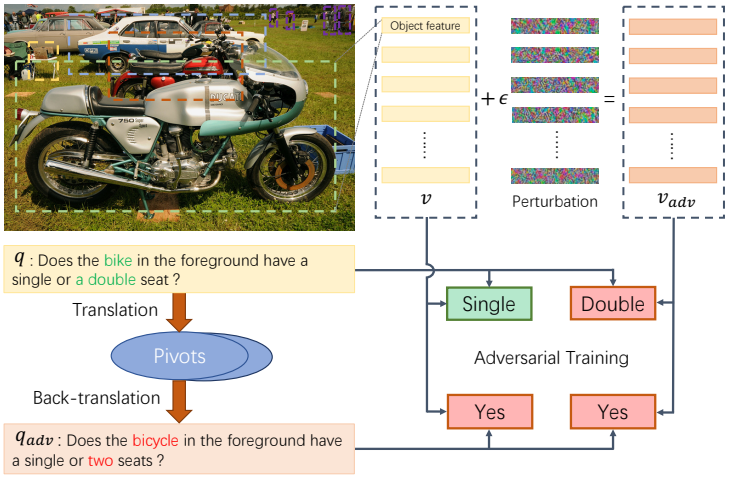

Fig. 1. Framework of the proposed data augmentation method. We generate adversarial examples of both visual and textual data as augmented data, which are passed through the VQA model to obtain incorrect answers. The augmented and original data are jointly trained using the proposed adversarial training scheme, which can boost model performance on clean data while improving model robustness against attack.

Adversarial Attack and Defense. In recent years, research works [38,12] add imperceptible perturbations to input images, named adversarial examples, to evaluate the robustness of deep neural networks against such perturbation attacks. In the NLP community, state-of-the-art textual DNN attackers [5,7,9] use a different approach from those in the visual community to generate textual adversarial examples. Our work is inspired by SCPNs [15] and SEA [34] which generate paraphrases of the sentence as textual adversarial examples. Meanwhile, previous works [12] show that training with adversarial examples can improve the model generalization on small dataset (e.g., MNIST), but degrade the performance on large datasets (e.g., ImageNet), in the fully-supervised setting. Recent notable work [41] suggests that adversarial training could boost model performance even on ImageNet with a well-designed training scheme. A number of methods [36,43] have investigated adversarial attack on the VQA task. However, they merely attack the image and do not discuss how the adversarial examples can benefit the VQA model. To summarize, how adversarial examples can facilitate VQA remains an open problem. This work sheds light on utilizing adversarial examples as augmented data for VQA.

3 Method

We now introduce our data augmentation method to train a robust VQA model. As illustrated in Fig. 1, given an $\langle image, question, answer \rangle$ triplet, we first gen-

erate the paraphrases of questions and store them, then, generate visual adversarial examples on-the-fly to obtain semantically equivalent additional training triplets, which are used in the proposed adversarial training scheme. We describe them in detail as follows.

3.1 VQA Model

Answering questions about images can be formulated as the problem of predicting an answer a given an image v and a question q according to a parametric probability measure:

$$\hat{a} = \arg\max_{a \in \mathcal{A}} p(a|v, q; \theta) \tag{1}$$

where θ represents a vector of all parameters to learn and $\mathcal{A}$ is a set of all answers. VQA requires solving several tasks at once involving both visual and textual inputs. Here we use Bottom-Up-Attention and Top-Down (BUTD) [2] as our backbone model because it has become a golden baseline in VQA. In BUTD, region-specific image features extracted by fine-tuned Faster R-CNN [11] are utilized as visual inputs. In this paper, let $v = \{\overrightarrow{v_1}, \overrightarrow{v_2}, ..., \overrightarrow{v_K}\}$ be a collection of visual features extracted from K image regions and the question is a sequence of words $q = \{q_1, q_2, ..., q_n\}$. The $\langle image, question, answer\rangle$ triplet has a strong semantic relation that neither image nor question can be easily transformed to augment the training data while preserving original content.

3.2 Data Augmentation

Due to the risk of affecting answers, we avoid manipulating the raw inputs (i.e., images and questions) directly, such as cropping the image or changing the word order. Inspired by the adversarial attack and defense, we propose to generate adversarial examples as additional training data. In this section, we present how to generate adversarial examples of images and questions while preserving the original labels and how to use them to augment the training data.

Visual Adversarial Examples Generation. Adversarial attacks are originated from the computer vision community. In general, the overarching goal is to add the least amount of perturbation to the input data to cause the desired misclassification. We employ an efficient gradient-based attacker Iterative Fast Gradient Sign Method (IFGSM)[23] to generate visual adversarial examples. Before illustrating IFGSM, we firstly introduce FGSM, as IFGSM is an extension of it. Goodfellow *et al.*[12] proposed the FGSM as a simple way to generate adversarial examples. We could apply it on visual input as:

$$v_{adv} = v + \epsilon \operatorname{sign}(\nabla_v L(\theta, v, q, a_{true})) \tag{2}$$

where v^{adv} is the adversarial example of v, θ is the set of model parameters, $L(\theta, v, q, a_{true})$ denotes the cost function used to train the VQA model, ϵ is the

size of the adversarial perturbation. The attacker backpropagates the gradient to the input visual feature to calculate $\nabla_v L(\theta, v, q, a_{true})$ while fixing the network parameters. Then, it adjusts the input by a small step in the direction (i.e. $\text{sign}(\nabla_v L(\theta, v, q, a_{true}))$) that maximize the loss. The resulting perturbed, v_{adv}, is then misclassified by the VQA model (e.g., the model predicts *Double* in Fig. 1).

A straightforward extension of FGSM is to apply it multiple times with small step size, referred to IFGSM as:

$$v_{adv}^0 = v, \quad v_{adv}^{N+1} = Clip_{v,\epsilon}\left\{v_{adv}^N + \alpha\,\text{sign}(\nabla_v L(\theta, v_{adv}^N, q, a_{true}))\right\} \tag{3}$$

where $Clip_{v,\epsilon}(A)$ denotes element-wise clipping A, with $A_{i,j}$ clipped to the range $[v_{i,j} - \epsilon, v_{i,j} + \epsilon]$, α is step size in each iteration. In this paper, we summarize gradient-based method as VAdvGen(v, q).

One-step methods of adversarial example generation generate a candidate adversarial image after computing only one gradient. Iterative methods apply many gradient updates. They typically do not rely on any approximation of the model and typically produce more harmful adversarial examples when running for more iterations. Our experimental results show that the accuracy of the BUTD vanilla trained model on visual adversarial examples generated by IFGSM is about 17%−30% for $\epsilon \in [0.3, 1.3]$. It implies that adversarial examples have different distribution to normal examples.

Semantic Equivalent Questions Generation. To generate adversarial example q_{adv} of a question, we cannot directly apply approaches from image DNN attackers since textual data is discrete. In addition, the perturbation size that measured by L_p norm in image is also inapplicable for textual data. Moreover, the small changes in texts, e.g., character or word change, would easily destroy the grammar and semantics, rendering the possibility of attack failure. Adhere to the principle of not changing the semantics of input data, inspired by [15,28], we generate semantically equivalent adversarial questions by using a sequence-to-sequence paraphrasing model.

Here we use a paraphrasing model [28] based purely on neural networks and it is an extension of the basic encoder-decoder Neural Machine Translation (NMT) framework. In the neural encoder-decoder framework, the encoder (RNN) is used to compress the meaning of the source sentence into a sequence of vectors. The decoder, a conditional RNN language model, generates a target sentence word-by-word. The encoder takes a sequence of original question words $X = \{x_1, ..., x_{T_x}\}$ as inputs, and produces a sequence of context. The decoder produces, given the source sentence, a probability distribution over the target sentence $Y = \left\{y_1, ..., y_{T_y}\right\}$ with a softmax function:

$$P(Y|X) = \prod_{t=1}^{T_y} P(y_t|y_{<t}, X) \tag{4}$$

However, in the case of paraphrasing, there is not a path from English to English, but a path from English to a pivot language to English can be used. For example, the source English sentence E_1, is translated into a single French sentence F. Next, F is translated back into English, giving a probability distribution over English sentences, E_2, which acts as paraphrase distribution:

$$P(E_2|E_1, F) = P(E_2|F) \tag{5}$$

Our paraphrasing model pivots through the set of K-best translations $\mathcal{F} = \{F_1, ..., F_K\}$ of E_1. This ensures that multiple aspects (semantic and syntactic) of the source sentence are captured. Translating multiple pivot sentences into one sentence producing a probability distribution over the target vocabulary could be formed as:

$$P(y_t = w|y_{<t}, \mathcal{F}) = \sum_{i=1}^{K} P(\mathcal{F}_i|E_1) \cdot P(y_t = w|y_{<t}, \mathcal{F}_i) \tag{6}$$

We further expand on the multi-pivot approach by pivoting over multiple sentences in multiple languages (e.g., French and Portuguese). Deriving from Eq. 6, we obtain $P(y_t = w|y_{<t}, \mathcal{F}^{Fr})$ and $P(y_t = w|y_{<t}, \mathcal{F}^{Po})$. Then averaging these two distributions, producing a multi-sentence, multi-lingual paraphrase probability:

$$P(y_t = w|y_{<t}, \mathcal{F}^{Fr}, \mathcal{F}^{Po}) = \frac{1}{2}(P(y_t = w|y_{<t}, \mathcal{F}^{Fr}) + P(y_t = w|y_{<t}, \mathcal{F}^{Po})) \tag{7}$$

which is used to obtain the probability distributions over sentences:

$$P(E_2|E_1) = \prod_{t=1}^{T_{E_2}} P(y_t|y_{<t}, \mathcal{F}^{Fr}, \mathcal{F}^{Po}) \tag{8}$$

We employ the pre-trained NMT model[3] which is trained for English↔Portuguese and English↔French to generate paraphrase candidates. A score [34] that measures the semantic similarity between paraphrase and its original text is defined as:

$$S(q, q') = \min\left(1, \frac{P(q'|q)}{P(q|q)}\right) \tag{9}$$

where $P(q'|q)$ is the probability of a paraphrase q' given original question q defined in Eq. 8, $P(q|q)$, which approximates how difficult it is to recover q, is used to normalize different distributions. We penalize those candidates with edit distance more than e or unknown words by adding a large negative number λ to the similarity score. We select the paraphrase candidates with the top-k semantic scores as our q_{adv}. The generation algorithm of q_{adv} is denoted $q_{adv} = \text{QAdvGen}(q)$.

[3] https://github.com/OpenNMT/OpenNMT-py

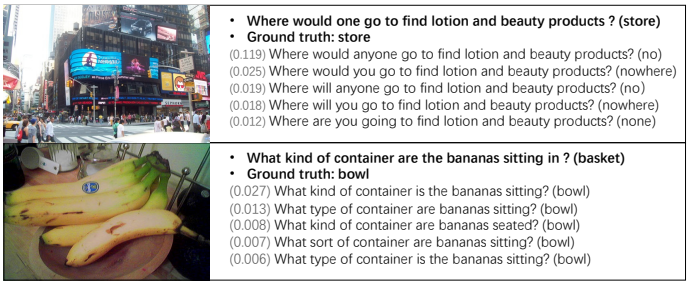

Fig. 2. Examples of our generated q_{adv}. The first question in bold in each block is the original question. The words in brackets are model predictions of the corresponding question; the numbers in brackets are the semantic score of q_{adv}.

Our paraphrases edit at least words to maintain syntax and semantics rather than exploring the linguistic variations regardless of the possibility of being perceived. We illustrate two examples of our q_{adv} in Fig. 2. They show that generated paraphrases could easily "break" the BUTD model. A predicted label is considered "flipped" if it differs from the prediction on the corresponding original question (assume that we do not attack visual data in this part). We observe that q_{adv} not only flip from positive predictions to negative ones but also correct the negative predictions to positive ones in some cases. Surprisingly, the flip rate of the vanilla trained model is 36.72% causing an absolute accuracy drop of 10%. It suggests that there is brittleness in the model decision and indicates that the model exploits spurious correlations while making its predictions.

3.3 Adversarial Training with Augmented Examples

Considering the adversarial training framework [24,41], we treat adversarial examples as additional training samples and train networks with a mixture of adversarial and clean examples. The augmented questions are model-agnostic and generated before training, while visual adversarial examples are continually generated at every step of training. There are two kinds of visual adversarial examples depending on the question inputs:

$$v_{qc} = \text{VAdvGen}(v, q), \quad v_{qadv} = \text{VAdvGen}(v, q_{adv}) \tag{10}$$

For each (v, q) pair, we have 4 additional training pairs, (v_{qc}, q), (v_{qadv}, q), (v_{qc}, q_{adv}) and (v_{qadv}, q_{adv}). All these four pairs are semantically equivalent, which means they hold the same ground truth answer. We maintain the original $\langle image, question, answer\rangle$ triplet but augment the original example at least four times, in the case of only one q_{adv} generated. We formulate a loss function that allows control of the relative weight of additional pairs in each batch:

$$\begin{aligned} Loss = L(\theta, v, q, a_{true}) + w\Big(&L(\theta, v_{qc}, q, a_{true}) + L(\theta, v_{qadv}, q, a_{true}) \\ &+ L(\theta, v_{qc}, q_{adv}, a_{true}) + L(\theta, v_{qadv}, q_{adv}, a_{true})\Big) \end{aligned} \tag{11}$$

Algorithm 1: Pseudo code of our adversarial training

Input: A set of clean visual and textual examples v, q with answers a
Output: Network parameter θ

q_{adv} = QAdvGen(q);
for *each training step* i **do**
  Sample a mini-batch of clean visual examples v^b, clean textual examples q^b, textual adversarial examples q^b_{adv} and answer a^b;
  if *i is in adversarial training period time* **then**
    Generating the corresponding mini-batch of additional training pairs (v^b_{qc}, q^b), (v^b_{qadv}, q^b), (v^b_{qc}, q^b_{adv}) and (v^b_{qadv}, q^b_{adv});
    Minimize the loss in Eq. 11 w.r.t. network parameter θ
  else
    Minimize the loss $L(\theta, v^b, q^b, a^b)$ w.r.t. network parameter θ
  end
end
return θ

where $L(\theta, v, q, a_{true})$ is a loss on a batch of v and q examples with true answer a_{true}, w is a parameter which controls the relative weight of adversarial examples in the loss. Our main goal is to improve network performance on clean images by leveraging the regularization power of adversarial examples. We empirically find that training with a mixture of adversarial and clean examples from beginning to end would not converge well on clean samples. Therefore, we mix them in a period of training time and fine-tune with clean examples in the rest epochs. Not only does this boost the performance on clean examples, but also improves the robustness of the model to adversarial examples. We present our adversarial training scheme in Algorithm 1.

4 Experiments

4.1 Experiments Setup

Dataset. We conduct experiments on the VQAv2 [13], which is improved from the previous version to emphasize visual understanding by reducing the answer bias in the dataset. VQAv2 contains 443K train, 214K validation and 453K test examples. The annotations for the test set are unavailable except for the remote evaluation servers. We provide our results on both validation and test set, and perform ablation study on the validation set.

VQA Architectures. We use a strong baseline Bottom-Up-Attention and Top-Down (BUTD) [2] which combines a bottom-up and a top-down attention mechanism to perform VQA, with the bottom-up mechanism generating

object proposals from Faster R-CNN [11], and the top-down mechanism predicting an attention distribution over those proposals. Following setting in [2,39], we use a maximum of 100 object proposals per image, which are 2048 dimensional features, as visual input. We represent question words as 300 dimensional embeddings initialized with pre-trained GloVe vectors [32], and process them with a one-layer GRU to obtain a 1024 dimensional question embedding.

Training Details. For fair comparison, we train the BUTD baseline and our framework using Adamax [21] with a batch size of 256 on the training split for a total of 25 epochs. Baseline achieves 63.32% accuracy on the validation set and we save this checkpoint to evaluate the attackers in the following. We set an initial learning rate of 0.001, and then decay it after five epochs at the rate of 0.25 for every two epochs. We inject the additional data merely in a period of epochs $(start, end)$, where *start* is the epoch when we start adversarial training and *end* is the epoch when we start standard training. We set the number of iterations n of IFGSM to 2 and the number of generated paraphrases per question to 1 for saving training time. In paraphrase generating, we set the edit distance threshold $e = 4$ and penalization score $\lambda = -10$. To avoid *label leaking* effect [24], we replace the true label in Eq. 2 and 3 with the most likely label predicted by the model when adversarial training. Our best result is achieved by using values $\epsilon = 0.3, \alpha = 0.0625, w = 50$. These hyperparameters are chosen based on grid search, and other settings are tested in the ablation studies.

4.2 Results

Overall Performance. Table 1 shows the results on VQAv2 validation, test-dev and test-std sets. We compare our method with the BUTD vanilla training setting. Our method outperforms vanilla trained baseline, making gains of 1.82%, 2.55%, 2.6% on validation, test-dev and test-std set, respectively. Furthermore, our training scheme only consumes a small amount of additional time (4 min for an epoch) while allows for a considerable increase in standard accuracy.

Comparison with Other Data Augmentation Methods. We compare our method with related VQA data augmentation method CC [35], and NLP data augmentation method EDA [40] and report the results on VQAv2 in Table 1. The model of CC is trained to predict the same answer for a question and its rephrasing, which are generated by a VQG module in their training scheme. Their outperforming validation accuracy is in contrast to the less competitive accuracy on the test-dev set. It reveals CC is less capable of generalizing on unseen data. EDA is a text editing technique boosting model performance on the text classification task. We implement it to generate three augmented questions per original question and set the percent of words in a question that are changed $\alpha = 0.1$. However, results (see Table 1) show that EDA could degrade the performance on clean data and make a 0.59% accuracy drop. It demonstrates that text editing techniques for generating question are not applicable as large

Table 1. Performance and ablation studies on VQAv2.0. All models listed here are single model, which trained on the training set to report *Val* scores and trained on training and validation sets to report *Test-dev* and *Test-std* scores. The first row represents the vanilla trained baseline model. The rows begin with + represents the data augmentation method added to the first row. EDA-3 represents that we generate three augmented questions per original questions using EDA [40]. † This method is implemented based on a stronger BUTD (see [35]) and obtains a relatively small improvement (0.48%) on validation score, even so, its test-dev score is surpassed by our method.

Method	Val	Test-dev				Test-std
		Overall	Yes/no	Number	Others	
BUTD [2]	63.32	65.23	81.82	44.21	56.05	65.67
+Noise	63.28	64.80	81.03	43.96	55.70	-
+EDA-3 [40]	62.73	-	-	-	-	-
+CC [35]†	65.53	67.55	-	-	-	-
+Ours	**65.16**	**67.78**	**84.08**	**47.55**	**58.48**	**68.27**
+Ours *w/o* Aug-Q	65.05	67.58	83.85	47.34	58.31	-
+Ours *w/o* Aug-V	64.69	67.45	83.55	46.96	58.37	-

numbers of questions are too short that could not be allowed to insert, delete or swap words. Moreover, sometimes the text editing may change the original semantic meaning of the question, which leads to noisy and even incorrect data.

Since our augmented data might be regarded as injecting noise to original data, we also set comparison by injecting random noise with a standard deviation of 0.3 (same as our ϵ in reported results) to visual data. Random noise, as well, could be regarded as a naive attacker that causes a 0.9% absolute accuracy drop on the vanilla model. However, jointly training with clean and noising data could not boost the performance on clean data, as reported in Table 1. It proves that our generated data are drawn from the proper distribution that let the model take full advantage of the regularization power of adversarial examples.

4.3 Analysis

Training Set Size Impact. Furthermore, we conduct experiments using a fraction of the available data in the training set. As overfitting tends to be more severe when training on smaller datasets, we show that our method has more significant improvements for smaller training sets. We run both vanilla training and our method for the following training set fractions (%): $\{20, 40, 60, 80\}$. Performances are shown in Table 2. The best accuracy without augmentation, 63.32%, was achieved using 100% of the training data. Our method surpasses it with 80% of the training data, achieving 64.27%.

Effect of Augmenting Time. We empirically find that the time when the adversarial examples are injected into training has an effect on accuracy. We

Table 2. Validation accuracy (%) across BUTD with and without our framework on different training set sizes.

Training set size	BUTD	+Ours
80%	62.77	64.27 (+1.50)
60%	61.55	63.11 (+1.56)
40%	59.47	61.35 (+1.88)
20%	55.45	57.39 (+**1.94**)

Table 3. Validation accuracy (%) of our method using different adversarial training periods.

$(start, end)$	Accuracy
(5,25)	63.93
(10,25)	64.08
(10,15)	**65.16**
(15,20)	64.18

demonstrate this via ablation studies in Table 3. We try several adversarial training period $(5, 25)$, $(10, 25)$, $(10, 15)$ $(15, 20)$. They respectively evaluate the effect of delaying injecting additional training data after different epochs and prove the advantage gained from fine-tuning with clean data in the last few epochs. Results show that $(10, 15)$ is the optimal adversarial training period, and it surpasses the baseline model and achieves 65.16% accuracy. One explanation is that adversarial examples have different underlying distributions to normal examples, and if boosting model performance on clean examples is our main goal, it is inappropriate to inject the perturbed examples at an early stage where the model has not warm up, and the fitting ability of model on clean examples need to be retrieved at the end of the training process.

4.4 Ablation Studies

Augmentation Decomposed. Results from ablation studies to test the contributions of our method's components are given in Table 1. The augmentation on visual and textual (question) data both make their individual contribution to improve the accuracy. We observe that visual adversarial examples are critical to our performance, and removing it causes a 0.47% accuracy drop (see Ours w/o Aug-V) on the validation set. The question augmentation also leads to comparable improvements, see the model of Ours w/o Aug-V.

Ablation on Adversarial Attackers. We now ablate the effects of attacker strength and type used in our method on network performance. To evaluate the regularization power of adversarial examples, we first compute the accuracy of the vanilla model after being attacked by the gradient-based attacker with a variety of sets of parameters. Since the visual input ranges from 0 to 83, we try perturbation size ϵ among $\{0.3, 0.5, 1, 1.3\}$, approximately following the ratio of ϵ to pixel value in [41], and step size α among $\left\{\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, 1, 2\right\}$.

Fig. 3(b) reflects the attacker strength changes with different parameter settings (accuracy declines implies strength increases) while Fig. 3(e) reflects how the model performance changes with attacker strength. We observe that the accuracy on clean data is inversely proportional to attacker strength. As weaker

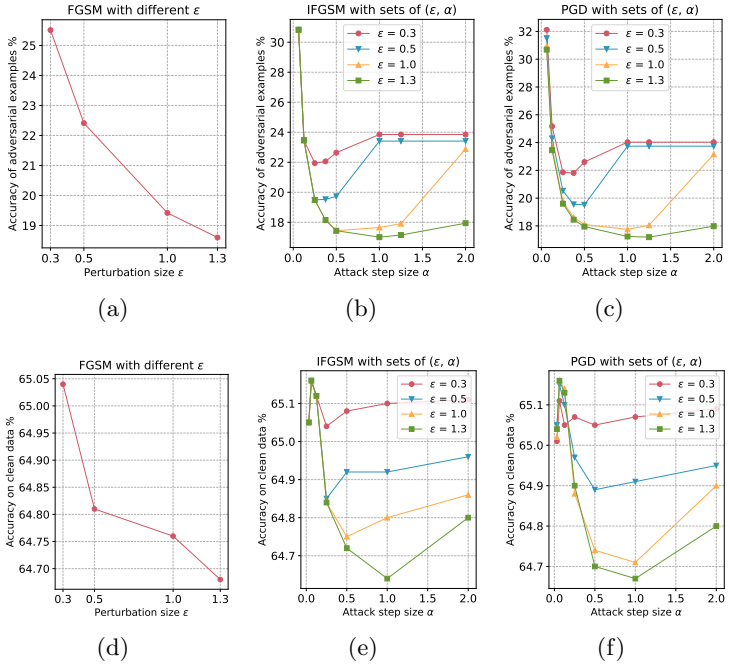

Fig. 3. Ablation on visual attacker strength and type. The top row is the accuracy of the vanilla model on adversarial examples generated by FGSM, IFGSM, and PGD, respectively. The bottom row is the standard accuracy of our model that adversarially trained with the corresponding attacker. The number of iterations is fixed to 2.

attackers push the distribution of adversarial examples less away from the distribution of clean data, the model is better at bridging domain differences. However, the extremely weak attacker (e.g., random noise, $\alpha < \frac{1}{64}$) yields negligible improvement on standard accuracy, since the generated data are drawn similar distribution with original data.

We then study the effects of applying different gradient-based attackers in our method on model performance. Specifically, we try two other attackers, FGSM and PGD [27]. Their performances are reported in Fig. 3(a), 3(d), 3(c), 3(f). We observe that all attackers substantially improve model performance over the vanilla training baseline. This result suggests that our VQA data augmentation method is not designed for a specific attacker.

4.5 Model Robustness

Improvement of model robustness against adversarial attacks is a reward of our adversarial training scheme. As shown in Table 4, we are able to significantly increase accuracy on visual adversarial examples by 13.74%, when using the training attacker at test-time. Following [8], we test a stronger PGD attacker

Table 4. Validation accuracy (%) of vanilla and adversarially trained (using IFGSM $\epsilon = 0.3, \alpha = 0.0625, n = 2$) network on clean and adversarial examples with various test-time attackers. Parap. represents the generated paraphrases in our method. Note that the IFGSM and PGD still act as the white-box attacker when testing.

	Clean	IFGSM	Parap.	IFGSM & Parap.	PGD
BUTD [2]	63.32	30.83	54.03	22.09	18.05
+Ours	**65.16**	44.57	63.18	43.64	22.64

($\epsilon = 0.5, \alpha = 0.125, n = 6$) and our model could beat the baseline by 4.59%. On the textual side, the accuracy of the vanilla model on q_{adv} is 54.03% and the flip rate (lower is better) is 36.72% while our adversarially trained model obtained an accuracy of 63.18% and a flip rate of 18.8% on q_{adv}. When attacking both visual and textual sides in test-time, our model beats the vanilla model by 21.55%. These results indicate that our model is capable of defending against both visual and textual common attackers.

4.6 Human Evaluation of Semantic Consistency

In order to show the semantic consistency of our generated paraphrases with original questions, we conduct a human study. We sampled 100 questions and their paraphrases with top1 semantic similarity score defined in Eq. 9, and asked 4 human evaluators to assign labels (e.g., positive for similar or negative for not similar). We averaged the opinions of different evaluations for each query to get a positive score of 84%. It demonstrates that the majority of paraphrases are similar to the originals.

5 Conclusion

In this paper, we propose to generate visual and textual adversarial examples as augmented data to train a robust VQA model with our adversarial training scheme. The visual adversaries are generated by gradient-based adversarial attacker and textual adversaries are paraphrases. Both of them keep modification imperceptible and maintain the semantics. Experimental results show that our method not only outperforms prior arts of VQA data augmentation, and also improves model robustness against adversarial attacks. To the best of our knowledge, this is the first work that uses both semantic equivalent visual and textual adversaries as data augmentation for the visual question answering problem.

Acknowledgements. This work was supported by National Key Research and Development Program of China (2016YFB1001003), NSFC (U19B2035, 61527804, 60906119), STCSM (18DZ1112300). C. Ma was sponsored by Shanghai Pujiang Program.

References

1. Agarwal, V., Shetty, R., Fritz, M.: Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. arXiv preprint arXiv:1912.07538 (2019)
2. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6077–6086 (2018)
3. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Learning to compose neural networks for question answering. arXiv preprint arXiv:1601.01705 (2016)
4. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 39–48 (2016)
5. Belinkov, Y., Bisk, Y.: Synthetic and natural noise both break neural machine translation. arXiv preprint arXiv:1711.02173 (2017)
6. Ben-Younes, H., Cadene, R., Cord, M., Thome, N.: Mutan: Multimodal tucker fusion for visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2612–2620 (2017)
7. Blohm, M., Jagfeld, G., Sood, E., Yu, X., Vu, N.T.: Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. arXiv preprint arXiv:1808.08744 (2018)
8. Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A.: On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705 (2019)
9. Ebrahimi, J., Lowd, D., Dou, D.: On adversarial examples for character-level neural machine translation. arXiv preprint arXiv:1806.09030 (2018)
10. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. arXiv preprint arXiv:1606.01847 (2016)
11. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
12. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
13. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6904–6913 (2017)
14. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: End-to-end module networks for visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 804–813 (2017)
15. Iyyer, M., Wieting, J., Gimpel, K., Zettlemoyer, L.: Adversarial example generation with syntactically controlled paraphrase networks. arXiv preprint arXiv:1804.06059 (2018)
16. Jain, U., Zhang, Z., Schwing, A.G.: Creativity: Generating diverse questions using variational autoencoders. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6485–6494 (2017)
17. Kafle, K., Kanan, C.: Visual question answering: Datasets, algorithms, and future challenges. Computer Vision and Image Understanding **163**, 3–20 (2017)

18. Kafle, K., Yousefhussien, M., Kanan, C.: Data augmentation for visual question answering. In: Proceedings of the 10th International Conference on Natural Language Generation. pp. 198–202 (2017)
19. Kim, J.H., Jun, J., Zhang, B.T.: Bilinear attention networks. In: Advances in Neural Information Processing Systems. pp. 1564–1574 (2018)
20. Kim, J.H., On, K.W., Lim, W., Kim, J., Ha, J.W., Zhang, B.T.: Hadamard product for low-rank bilinear pooling. arXiv preprint arXiv:1610.04325 (2016)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
22. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision **123**(1), 32–73 (2017)
23. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
24. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)
25. Li, Y., Duan, N., Zhou, B., Chu, X., Ouyang, W., Wang, X., Zhou, M.: Visual question generation as dual task of visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6116–6124 (2018)
26. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: Advances in neural information processing systems. pp. 289–297 (2016)
27. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
28. Mallinson, J., Sennrich, R., Lapata, M.: 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. pp. 881–893 (2017)
29. Norcliffe-Brown, W., Vafeias, S., Parisot, S.: Learning conditioned graph structures for interpretable visual question answering. In: Advances in Neural Information Processing Systems. pp. 8334–8343 (2018)
30. Patro, B., Namboodiri, V.P.: Differential attention for visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7680–7688 (2018)
31. Peng, G., Jiang, Z., You, H., Lu, P., Hoi, S., Wang, X., Li, H.: Dynamic fusion with intra-and inter-modality attention flow for visual question answering. arXiv preprint arXiv:1812.05252 (2018)
32. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
33. Ray, A., Sikka, K., Divakaran, A., Lee, S., Burachas, G.: Sunny and dark outside?! improving answer consistency in vqa through entailed question generation. arXiv preprint arXiv:1909.04696 (2019)
34. Ribeiro, M.T., Singh, S., Guestrin, C.: Semantically equivalent adversarial rules for debugging nlp models. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 856–865 (2018)

35. Shah, M., Chen, X., Rohrbach, M., Parikh, D.: Cycle-consistency for robust visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6649–6658 (2019)
36. Sharma, V., Vaibhav, A., Chaudhary, S., Patel, L., Morency, L.: Attend and attack: Attention guided adversarial attacks on visual question answering models (2018)
37. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1), 60 (2019)
38. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
39. Teney, D., Anderson, P., He, X., Van Den Hengel, A.: Tips and tricks for visual question answering: Learnings from the 2017 challenge. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4223–4232 (2018)
40. Wei, J.W., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
41. Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A., Le, Q.V.: Adversarial examples improve image recognition. arXiv preprint arXiv:1911.09665 (2019)
42. Xie, Z., Wang, S.I., Li, J., Lévy, D., Nie, A., Jurafsky, D., Ng, A.Y.: Data noising as smoothing in neural network language models. arXiv preprint arXiv:1703.02573 (2017)
43. Xu, X., Chen, X., Liu, C., Rohrbach, A., Darrell, T., Song, D.: Fooling vision and language models despite localization and attention mechanism. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4951–4961 (2018)
44. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 21–29 (2016)
45. Yu, A.W., Dohan, D., Luong, M.T., Zhao, R., Chen, K., Norouzi, M., Le, Q.V.: Qanet: Combining local convolution with global self-attention for reading comprehension. arXiv preprint arXiv:1804.09541 (2018)
46. Zhang, Y., Hare, J., Prügel-Bennett, A.: Learning to count objects in natural images for visual question answering. arXiv preprint arXiv:1802.05766 (2018)