Supplemental material to "Fooling Vision and Language Models Despite Localization and Attention Mechanism"

A. Implementation details of improved attacks to VQA models

In this appendix, we detail our attack strategy against VQA models, which is briefly sketched in Section 3.3. We first provide in A.1 the full background setup and some terminology, as well as a formal definition of targeted adversarial examples for VQA models. Then, we present the attack details in A.2 and A.3 using the terminology defined in A.1. In A.4, we will explain other implementation details, such as hyperparameter values, used in our evaluation. In the end, we provide some proofs to the theoretical analysis behind our technique in A.5.

A.1. Targeted adversarial examples for VQA

We denote a VQA model as $f_\theta(I, Q)$, where θ is the parameters of the model, I is the input image, and Q is the input question. The output $f_\theta(I, Q)$ is the predicted answer to the question Q given the image I.

Most existing VQA models consider this task as a classification problem. That is, they choose the most probable answer among the *top-K most frequent answers* in the training set (or both training and testing set). Typically, state-of-the-art VQA models use $K = 3000$.

We consider that the target to a VQA model f_θ is a question-answer pair $(Q^{\mathbf{target}}, A^{\mathbf{target}})$, and a targeted adversarial example is an image $I^{\mathbf{adv}}$ such that

$$f_\theta(I^{\mathbf{adv}}, Q^{\mathbf{target}}) = A^{\mathbf{target}} \text{ s.t. } d(I^{\mathbf{adv}}, I^{\mathbf{ori}}) \leq B$$

where we refer to $I^{\mathbf{ori}}$ as the *benign image*.

A neural network-based VQA model f_θ can also be represented as $f_\theta(I, Q) = \mathbf{argmax}_i J_\theta(I, Q)$, where $J_\theta(I, Q)$ outputs a K-dimensional vector in which each dimension indicates the probability of corresponding choice to be the predicted answer. Therefore, we can generate adversarial examples by solving the following optimization problem

$$\mathbf{argmin}_{I^{\text{adv}}} \mathcal{L}(J_\theta(I^{\mathbf{adv}}, Q^{\mathbf{target}}), A^{\mathbf{target}}) \quad (5)$$

$$\text{s.t. } d(I^{\mathbf{adv}}, I^{\mathbf{ori}}) \leq B \quad (6)$$

where $I^{\mathbf{ori}}$ is a benign image, and the goal is to find an adversarial example $I^{\mathbf{adv}}$ that is close to $I^{\mathbf{ori}}$. Typically, $\mathcal{L}$ is chosen as the same loss function for training the model, but other alternatives which are monotonic to the training loss can also be used. In particular, Carlini *et al*. show that the choice of different loss functions has a significant impact on the attack success rate [10], when the attacks are evaluated on MNIST dataset [39]. In this work, we consider $\mathcal{L}$ to be the cross-entropy loss, which is equivalent to the best loss function used in [10].

A.2. Solving the optimization problem

In our attack method, we approximate the optimization problem using an alternative objective function (4). We restate it below using the notation defined in A.1:

$$\begin{aligned}\xi(A^{\mathbf{predict}}) \quad = \quad & \mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{target}}) \\ & +\lambda_1 \cdot \mathbf{1}(A^{\mathbf{target}} \neq A^{\mathbf{predict}}) \\ & \cdot(\tau - \mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{predict}})) \\ & +\lambda_2 \cdot \text{ReLU}(d(x, I^{\mathbf{ori}}) - B + \epsilon) \quad (7)\end{aligned}$$

In this formula, we use x to represent the image. Thus the adversarial example is the value of x that minimizes the objective (7). This objective has three components. The first component, $\mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{target}})$, is the same as objective (5). The latter two are the innovations in this work, and we elaborate their design in the following.

The second component. The second component is

$$\lambda_1 \cdot \mathbf{1}(A^{\mathbf{target}} \neq A^{\mathbf{predict}}) \cdot (\tau - \mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{predict}}))$$

Here the hyperparameter λ_1 is used to balance this component and others, and $A^{\mathbf{predict}}$ is the prediction of the original image. The value of $A^{\mathbf{predict}}$ is set dynamically during the iterative optimization process, so that each iteration may choose a different value of $A^{\mathbf{predict}}$. We will explain this process in more details in the next subsection. We set τ to be a constant, e.g., $\mathbf{log}(K)$ when $\mathcal{L}$ is chosen as the cross-entropy loss. This constant guarantees the second term is always non-negative, especially when $\mathbf{1}(A^{\mathbf{target}} \neq A^{\mathbf{predict}})$. In fact, we have the following theorem:

Theorem 1. *Assuming $\tau = \log K$, where K is the number of output classes, $\mathcal{L}$ is the cross-entropy loss, i.e., $\mathcal{L}(u, i) = -\log u_i$, the last layer of J is a softmax operator, and $A^{\mathbf{predict}}$ is the prediction of the model over input image x and question $Q^{\mathbf{target}}$, i.e., $\mathbf{argmax}_i J_\theta(x, Q^{\mathbf{target}})$, then we have*

$$\begin{aligned}&\mathbf{1}(A^{\mathbf{target}} \neq A^{\mathbf{predict}}) \\ &\cdot(\tau - \mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{predict}})) \quad \geq \quad 0 \quad (8)\end{aligned}$$

To understand how this component works, we consider two possible cases. First, in the case $A^{\mathbf{predict}} = A^{\mathbf{target}}$, the image generated in the last iteration is already an adversarial example, and thus this component is 0 since $\mathbf{1}(A^{\mathbf{target}} \neq A^{\mathbf{predict}}) = 0$. In this case, optimizing objective (7) is equivalent to maximizing the probability of predicting the target answer $A^{\mathbf{target}}$.

Second, when $A^{\mathbf{predict}} \neq A^{\mathbf{target}}$, minimizing the second component is essentially maximizing

Algorithm 2 Targeted Adversarial Generation Algorithm

Input: $\theta, I^{\mathbf{ori}}, Q^{\mathbf{target}}, A^{\mathbf{target}}, B, \epsilon, \lambda_1, \lambda_2, \eta, \mathbf{maxitr}$
Output: $I^{\mathbf{adv}}$

1 $I^1 \leftarrow I^{\mathbf{ori}} + \delta$ for δ sampled from a uniform distribution between $[-B, B]$;
2 **for** $i = 1 \rightarrow \mathbf{maxitr}$ **do**
3 $A^{\mathbf{predict}} = f_\theta(I^i, Q^{\mathbf{target}})$;
4 **if** $A^{\mathbf{predict}} = A^{\mathbf{target}}$ and $i > 50$ **then**
5 **return** I^i
6 $I^{i+1} \leftarrow \mathbf{update}(I^i, \eta, \nabla_x \xi(A^{\mathbf{predict}}))$;
7 **return** $I^{\mathbf{maxitr}+1}$

$\mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{predict}})$, which is equivalent to minimizing the probability of the model to predict $A^{\mathbf{predict}}$, which is different from the target answer $A^{\mathbf{target}}$. As for the value of the hyperparameter λ_1, which is used to balance between this component and others, we find that setting $\lambda_1 = 1$ works well in most cases. Notice that in this case, jointly optimizing the first and the second component is equivalent to optimizing the best loss function used in Carlini's attack.

The third component. The third component is set to enforce the constraint (6). In particular, $\mathrm{ReLU}(x) = \mathbf{max}(0, x)$ is the rectifier function, and ϵ is a small positive hyper-parameter that we will explain later. When $d(I^{\mathbf{adv}}, I^{\mathbf{ori}}) \leq B - \epsilon < B$, i.e., constraint (6) is satisfied, the third component is 0, and thus has no effective on the objective. On the other hand, if an adversarial example $I^{\mathbf{adv}}$ does not satisfy constraint (6), we show that it is never optimal for (7) when $\lambda_2 \epsilon$ is large enough. We have the following theorem:

Theorem 2. *When $\lambda_2 \epsilon > \mathcal{L}(f_\theta(I^{\mathbf{ori}}, Q^{\mathbf{target}}), A^{\mathbf{target}}) + \lambda_1 \tau$, the solution $I^{\mathbf{adv}}$ minimizing the objective (7) satisfies constraint (6) as well.*

In practice, we can set ϵ to be a small value (e.g., 2), and set λ_2 to be a large value (e.g., 10), then the generated adversarial examples end up not activating the ReLU function (i.e., the output of the function is 0). Even when the ReLU function is activated, its value is not larger than ϵ, and thus the constraint (6) is still satisfied.

Notice that in most previous iterative optimization-based approaches [10, 43], optimizing (5) while satisfying constraint (6) is converted into a joint optimization problem of $\mathcal{L}(...) + \lambda d(...)$, which minimizes both the lost function (5) and the distance function $d(I^{\mathbf{adv}}, I^{\mathbf{ori}})$. The most prominent difference is that our approach does not minimize this distance as long as it is within the bound B.

A.3. Putting everything together

The overall adversarial generation method is presented in Algorithm 1 (see the main paper). We restate the algorithm in Algorithm 2 using the notation defined in A.1, and explain the details below.

This algorithm takes the hyper-parameters defined above, along with η, representing the learning rate, and $\mathbf{maxitr}$, representing the maximal number iterations that the algorithm runs. In the algorithm, I^1 is initialized with a random starting point satisfying constraint (6) (line 1). Then the algorithm iteratively updates I^i (lines 2-6). In each iteration, the prediction $A^{\mathbf{predict}}$ is first computed (line 3). If this prediction already matches the target, and the algorithm has run for at least 50 iterations, the algorithm stops and returns I^i as a successful adversarial example (lines 4-5). Here, 50 is a hyperparameter that can be further tuned. In this work, we fix it to be 50 in all experiments. On the other hand, if the algorithm does not stop at line 5, then I^{i+1} will be updated based on the gradient $\nabla_x \xi(A^{\mathbf{predict}})$ and the learning rate η (line 6). Here, **update** can be any optimization algorithm. We evaluated the algorithm's performance by using SGD, Adam, or RMSProp, and found that Adam always yields the best attack success rate. Therefore, we use Adam as the **update** function through out this work. In the end, if it does not return at line 5 during some iteration, then the algorithm fails at finding an adversarial example, and it returns $I^{\mathbf{maxitr}+1}$ as a result. In our evaluation, we set $\eta = 1.0$ and $\mathbf{maxitr} = 1000$ for evaluation.

Note that Carlini *et al.* [10] also suggest running the optimization algorithm multiple times with different random starting points (i.e., line 1) to avoid local optima. We employ the same trick and pick the best adversarial example generated among different executions of Algorithm 1 as the final result.

A.4. Adversarial example generation algorithms details

In our evaluation, we examine both attack methods, i.e. Carlini *et al.* [10] and our proposed algorithm. For Carlini's attack, we choose to minimize the loss function:

$$\mathrm{ReLU}(\mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{predict}}) - \mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{target}})) + \lambda d(x, I^{\mathbf{ori}})$$

where $x = 255 \times (\mathbf{tanh}(\delta) + 1)/2$ to simulate the boxed constraint that each pixel value can only take value from $[0, 255]$. This approach is demonstrated to be the most effective one in [10]. Here λ is chosen to be 0.1 by a grid search.

For our approach, as we discussed in Section 3, we choose the values of hyper-parameters as follows: $\epsilon = 2, \lambda_1 = 1, \lambda_2 = 10, \eta = 1.0, \mathbf{maxitr} = 1000$. Note that these hyper-parameters are set based on each image being represented as a vector of pixel values from $[0, 255]$.

When we generate adversarial examples, we employ the RMSE distance function as used in [43]. In particular, assuming there are two N-dimensional vectors x_1, x_2, then

the RMSE between the two vectors is computed as

$$RMSE(x_1, x_2) = \sqrt{||x_1 - x_2||_2^2/N} = ||x_1 - x_2||_2/\sqrt{N}$$

where $||\cdot||_2$ denotes the L2-norm of a vector. Further, in all experiments, the bound on the distance $B = 20$. In our experiments, the average distance for generated adversarial examples is below 10. We demonstrate several adversarial examples in Section D.1 to illustrate that the generated adversarial examples are visually similar to their benign counterparts.

A.5. Proofs to the theorems

We now present the formal proofs to Theorem 1 and Theorem A.2 presented in A.2.

Proof of Theorem 1. We consider two cases between the relationship between $A^{\mathbf{target}}$ and $A^{\mathbf{predict}}$. First, when $A^{\mathbf{target}} = A^{\mathbf{predict}}$, the left-hand side of (8) is 0, and thus (8) is trivially true.

Second, when $A^{\mathbf{target}} \neq A^{\mathbf{predict}}$, then the left-hand side of (8) becomes

$$\tau - \mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{predict}})$$

Thus proving (8) is equivalent to prove

$$\mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{predict}}) \leq \tau = \log K$$

We prove this by contradiction. Assuming $\mathcal{L}(J_\theta(x, Q^{\mathbf{target}}), A^{\mathbf{predict}}) > \log K$, then we have

$$-\log J_\theta(x, Q^{\mathbf{target}})_{A^{\mathbf{predict}}} > \log K$$

and thus

$$J_\theta(x, Q^{\mathbf{target}})_{A^{\mathbf{predict}}} < 1/K$$

By definition, we have

$$A^{\mathbf{predict}} = \mathbf{argmax}_i J_\theta(x, Q^{\mathbf{target}})_i$$

thus we know

$$\forall i \in \{1, ..., K\}. J_\theta(x, Q^{\mathbf{target}})_i < 1/K$$

Therefore, we know that

$$\sum_{i=1}^{K} J_\theta(x, Q^{\mathbf{target}})_i < K \times (1/K) = 1$$

However, we assume the last layer of J is a softmax layer, and thus we have

$$\sum_{i=1}^{K} J_\theta(x, Q^{\mathbf{target}})_i = 1$$

which is a contradiction. Therefore, we conclude that Theorem 1 is true. □

Proof of Theorem A.2. We prove this by contradiction. We assume an adversarial example $I^{\mathbf{adv}} = I^\star$ does not satisfy (6), but optimizes (7). In this case, $d(I^{\mathbf{adv}}, I^{\mathbf{ori}}) > B > B - \epsilon$, and thus the ReLU function is activated and its output must be greater than ϵ. Thus, the third component is at least $\lambda_2\epsilon$. Since the other two components are also non-negative, therefore, the objective of (7) is at least $\lambda_2\epsilon$ as well. On the other hand, we can set $I^{\mathbf{adv}} = I^{\mathbf{ori}}$, so that the value of objective (7) is at most $\mathcal{L}(f_\theta(I^{\mathbf{ori}}, Q^{\mathbf{target}}), A^{\mathbf{target}}) + \lambda_1\tau$. Since $\lambda_2\epsilon > \mathcal{L}(f_\theta(I^{\mathbf{ori}}, Q^{\mathbf{target}}), A^{\mathbf{target}}) + \lambda_1\tau$, we have that setting $I^{\mathbf{adv}} = I^{\mathbf{ori}}$ results in a lower value of objective (7) than $I^{\mathbf{adv}} = I^\star$, which contradicts the assumption! □

B. Further evaluation on DenseCap

We present the top-K accuracy results for Caption B in Figure 7. The 17 failed adversarial examples of Caption A are presented in Figure 9. We omit the 148 failed adversarial examples of Caption B due to size limitation.

We also report the top-K accuracy results for our Gold set in Figure 8.

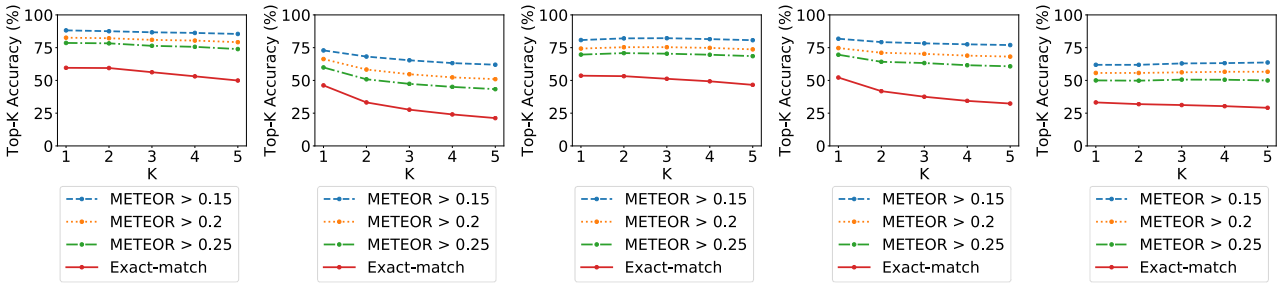
C. VQA attack dataset construction details

We construct multiple attack datasets in our experiments. Each dataset contains a set of (I, Q, A) triples, where I is a benign image, and (Q, A) is a target question-answer pair. We explain how these triples are selected in different datasets below.

Gold. For this dataset, we manually create triples where the target question is meaningful to the image, and the target answer is incorrect to the question and image pairs. To achieve this goal, we randomly select 100 images. For each of them, we manually choose questions that are meaningful to the image, while both MCB and N2NMN models can answer correctly the questions based on the image. If none of such questions exist for an image, we replace it with another randomly selected image. We repeat this process until we get 100 question-image pairs where both models predict correct answers. Then, for each question-image pair, we manually choose an answer that makes sense for the question but is incorrect in the context of the image. In the end, we have 100 (I, Q, A) triples that constitute the Gold set.

VQA-A. This dataset is designed to be a combination of two sub-datasets, i.e., **Popular-QA** and **Rare-QA**. These two aim at evaluating the resilience of the two VQA models against adversarial examples with different target question-answer pairs.

For the **Popular-QA** dataset, we select 3,000 *popular* question-answer pairs. In particular, we first remove all answers appearing less than 3 times along with their questions in the VQA training set. This is because we observe that among these least frequent answers, many are simply typos,



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4 (e) Image 5

Figure 7: Top-K accuracy on the Caption B dataset averaged across 5 images generated with each target caption

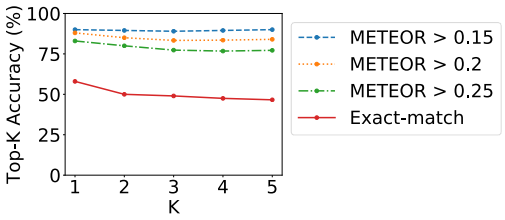


Figure 8: The result for adversarial attack against DenseCap model on the Gold set.

(e.g., spelling "kitchen" as "kitten"). Therefore, we remove them from consideration.

We consider the top-1000 most frequent questions in the remaining set as *popular*. Further, for each popular question, we choose its top-3 most frequent answers and consider each corresponding question-answer pair as *popular*. To ensure each question has at least 3 answers, we also remove all questions with less than 3 answers before selecting the top-1000 most frequent questions.

We are interested in the popular question-answer pairs, because they appear more frequently in the training set, and thus the models may more likely remember these question-answer pairs. Therefore, we hypothesize that it is more likely to successfully generate an adversarial example with such a target for an irrelevant image. We create this dataset to test this hypothesis.

We also randomly select 5 images, which are provided in the top raw of Figure 11. For each question-answer pair (Q, A) and each image I, we add the triple (I, Q, A) to the Popular-QA set. In the end, there are 15,000 triples in this dataset.

The second dataset, i.e., **Rare-QA**, is similar to Popular-QA, but the question-answer pairs are *rare*. In particular, we filter out the answers appearing less than 3 times, and all questions with less than 3 remaining answers in the same way as during construction of Popular-QA.

Among the remaining questions, we select the top-1000 least frequent ones, and for each of them, we select the three least frequent answers. We consider the question-answer pairs selected by such criteria as *rare*, and in the end, we have 3,000 rare question-answer pairs. We use the same 5 images as in Popular-QA, and generate a triple using each question-answer pair and each image to construct 15,000 triples which constitute Rare-QA.

Popular question-answer pairs		
QA1	What room is this?	bathroom
QA2	What sport is this?	baseball
QA3	What is the man doing?	skateboarding
QA4	What is the man holding?	frisbee
QA5	Is it raining?	no
Rare question-answer pairs		
QA1	What vegetable can be seen?	carrot
QA2	What is the fence covered with?	net
QA3	What does the blue signs represent?	handicap
QA4	Why is the girl standing in the middle of the room with an object in each hand?	playing wii
QA5	Who manufactured this plane?	japan

Table 3: The question-answer pairs used in **Scale-Image**, popular (top) and rare (bottom).

In doing so, we can evaluate the resilience of the two VQA models against adversarial examples on both popular question-answer pairs and rare question-answer pairs.

VQA-B. This dataset is similar to VQA-A, but is designed to evaluate the adversarial generation algorithm's performance across different benign images. To this end, we randomly select five popular question-answer pairs and five rare question-answer pairs, listed in Table 3, as well as 5,000 images to construct 50,000 triples in total. These triples constitute Scale-Image.

D. More Results on Experiments with VQA

We present the CDF curves of adversarial examples generated using both our approach and Carlini's approach

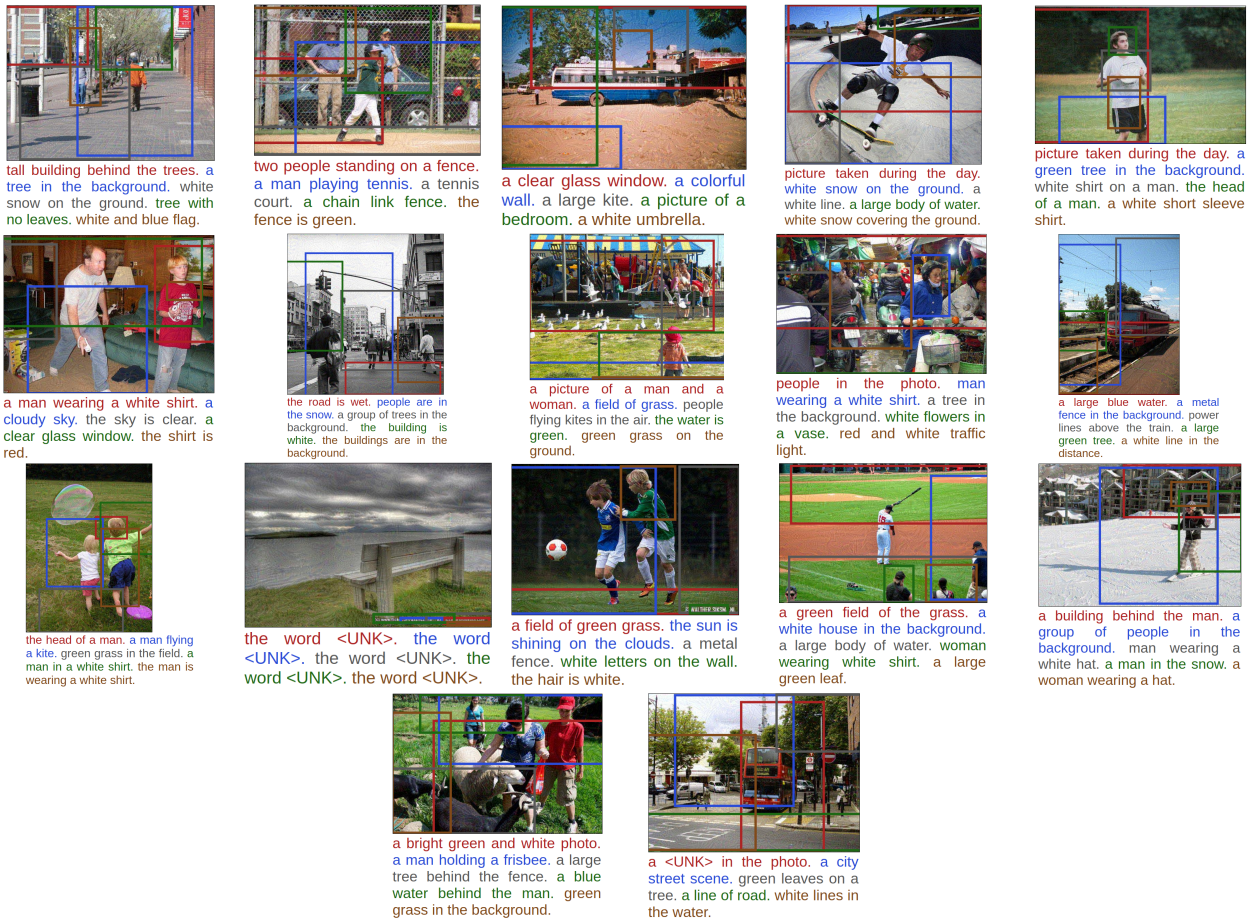


Figure 9: 17 failing adversarial examples generated from Caption A. For all these examples, the target caption is "white clouds in blue sky".

against MCB and N2NMN from the five images used in VQA-A, but we separately plot the analysis for Popular-QA and Rare-QA. For each combination of attack-model-dataset, we plot five curves on the same figure. The results are presented in Figure 10. The results show that for the same specification, the CDF curves for different images are close to each other. This shows that the attack performance is less dependent on images and more on the QA targets. We also see that the Rare-QA targets are more difficult than Popular-QA targets, and that our attack achieves higher probabilities that the Carlini's attack.

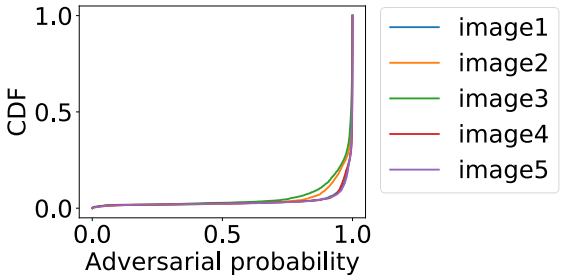
D.1. Qualitative study

Figure 11 presents some qualitative examples from our experiments on the Rare-QA pairs. We provide both benign images and adversarial examples generated against MCB and N2NMN. We observe that it is hard to distinguish the benign images from adversarial ones visually.

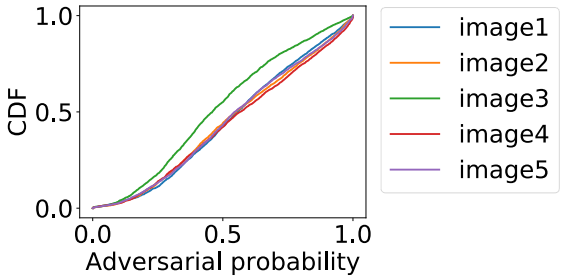
We show the highest predictions of both VQA models on the benign images (top) and on the adversarial examples generated for the target QA pairs (bottom). We show targets in "[]" and highlight the failed attacks in *italics*. First, we note that even for the questions irrelevant to the images, initially, both VQA models can make reasonable predictions. We then review the models' behavior on the adversarial examples. We observe that the MCB model is more frequently fooled by the adversaries than the N2NMN model, for instance in the case of the first question. For the second question both models predict "left" instead of the target "to left", so essentially the attack succeeds, but it is counted as a failure case. Therefore, our quantitative results provide an over-conservative estimation on the attack success rate. Finally, for the third question all the attacks fail, and top predictions such as "yes" indicate the models' confusion. Interestingly, N2NMN model predicts "military" instead of "navy" for Image 2, which can also be counted as a success.

D.2. Transferability Discussion

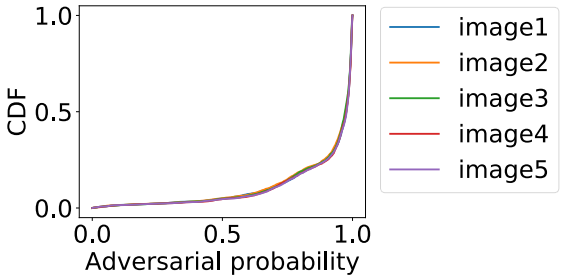
In this work, we focus on *white-box* adversarial examples, which means that the generation of these adversar-



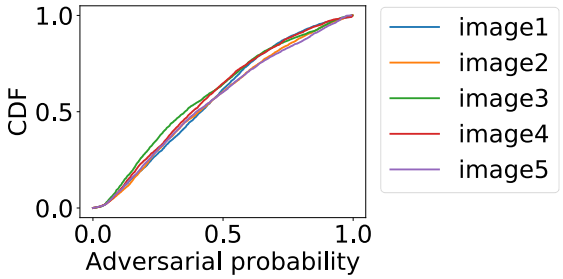
(a) CDF on adversarial probability of adversarial examples generated by our approach against MCB on Popular-QA.



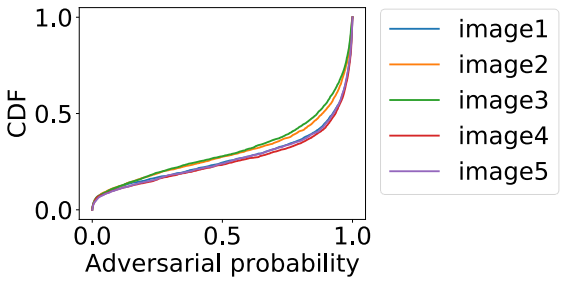
(b) CDF on adversarial probability of adversarial examples generated by Carlini's approach against MCB on Popular-QA.



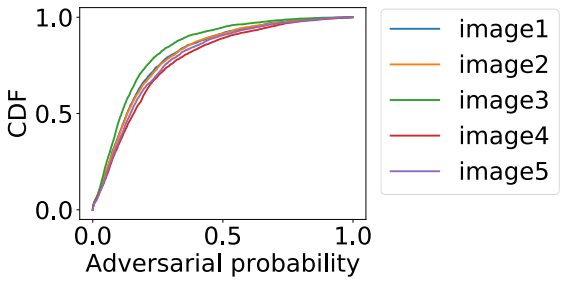
(c) CDF on adversarial probability of adversarial examples generated by our approach against N2NMN on Popular-QA.



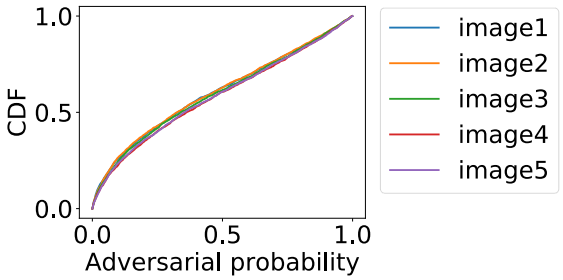
(d) CDF on adversarial probability of adversarial examples generated by Carlini's approach against N2NMN on Popular-QA.



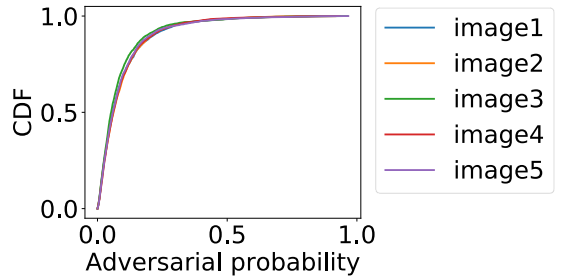
(e) CDF on adversarial probability of adversarial examples generated by our approach against MCB on Rare-QA.



(f) CDF on adversarial probability of adversarial examples generated by Carlini's approach against MCB on Rare-QA.



(g) CDF on adversarial probability of adversarial examples generated by our approach against N2NMN on Rare-QA.



(h) CDF on adversarial probability of adversarial examples generated by Carlini's approach against N2NMN on Rare-QA.

Figure 10: More CDF figures

ial examples requires full knowledge of the model architectures. However, we also demonstrate that an adversary could likely generate *black-box* adversarial examples without such knowledge. This is possible due to the *transferability* of adversarial examples, i.e., their ability to *transfer* between different network architectures [43, 53, 56, 66].

Previous work demonstrates transferability between: (1) two models with the same architecture trained on different

Predictions on the benign images





Image 1	Image 2	Image 3	Image 4	Image 5
		What are the people there for?		
MCB: baseball	flying kites	airplane	tennis	elephants
N2NMN: baseball	kites	flying	tennis	parade
		Where is the boy's shadow?		
MCB: ground	ground	plane	court	ground
N2NMN: ground	kite	sky	tennis court	ground
		Why is the man wearing a head covering?		
MCB: protection	safety	safety	tennis	protection
N2NMN: protection	flying kite	safety	sweat	shade

Predictions on the adversarial examples

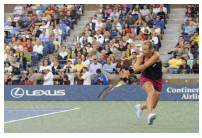
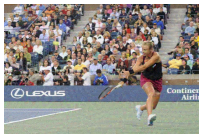
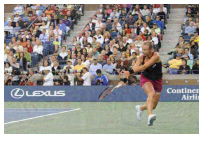
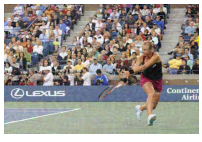
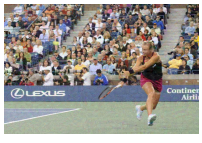
		What are the people there for? [festival]		
MCB: festival	festival	festival	festival	festival
				
N2NMN: *parade*	*parade*	*parade*	*parade*	festival
				
		Where is the boy's shadow? [to left]		
MCB: *left*	*left*	*left*	*left*	*left*
				
N2NMN: *left*	*left*	*left*	*left*	*left*
				
		Why is the man wearing a head covering? [navy]		
MCB: *yes*	*yes*	*safety*	*yes*	*costume*
				
N2NMN: *yes*	*military*	*yes*	*yes*	*yes*
				

Figure 11: MCB and N2NMN's predictions on benign and adversarial images and QA pairs from **Rare-QA**. The target answer is provided in "[]" along with the question. The *text in italics* indicates that the targeted adversarial examples do not mislead the model to produce the exact target answer.

training data; (2) two models with different architectures trained on the same training data; and (3) even a neural network model and a non-neural network model (e.g., kNN, SVM). Most previous work demonstrates transferability of non-targeted adversarial examples. In [43], Liu *et al*. further demonstrate that almost none of targeted adversarial examples generated for one model transfer to another one, and developed a novel approach to generate targeted adversarial examples for an ensemble of multiple state-of-the-art classification models to achieve better transferability.

The transferability of adversarial examples enables the adversary to generate black-box adversarial examples from a white-box adversary. To do so, the adversary can simply generate adversarial examples against a white-box model that performs the same task as the black-box model, and these adversarial examples would transfer to the black-box model with a high probability. Papernot *et al*. show that they can effectively generate non-targeted black-box adversarial examples against black-box online machine learning systems hosted by Amazon, Google and MetaMind [57, 56]. Further, Liu *et al*. demonstrate successful non-targeted and targeted black-box adversarial examples against Clarifai.com, which is a commercial company providing state-of-the-art image classification services [43].

Again, all these previous work only study image classification models. In this work, we are interested in the transferability of targeted adversarial examples between vision-language models, which we show below.

Experiment Results. We test the transferability of the generated adversarial examples between MCB and N2NMN. We use the **Gold** set to generate adversarial examples for this evaluation. We find that 79 out of 100 adversarial examples generated for the MCB model can transfer to N2NMN, while the number is 60 in the other direction. This shows that adversarial examples on VQA models can transfer well, and thus opens the door for black-box attacks.

Notice that in existing work [43], Liu *et al*. demonstrate that it is non-trivial to generate transferable targeted adversarial examples from a single image classification model. We note that both MCB and N2NMN employ the same pre-trained ResNet-152 features [22] as their image representation. Thus, we attribute the good transferability results to the use of ResNet-152 in both models.

E. Analysis on Hard Targets for Generating Adversarial Examples

While we observe that adversarial examples can be generated for most target question-answer pairs, in some cases the adversarial generation algorithm fails. We notice that whether the attack will succeed or not depends on the target question-answer pair rather than on the benign image. In this section, we investigate the failure cases and provide some insights into why some targeted attacks may be hard.

E.1. The effectiveness of language priors

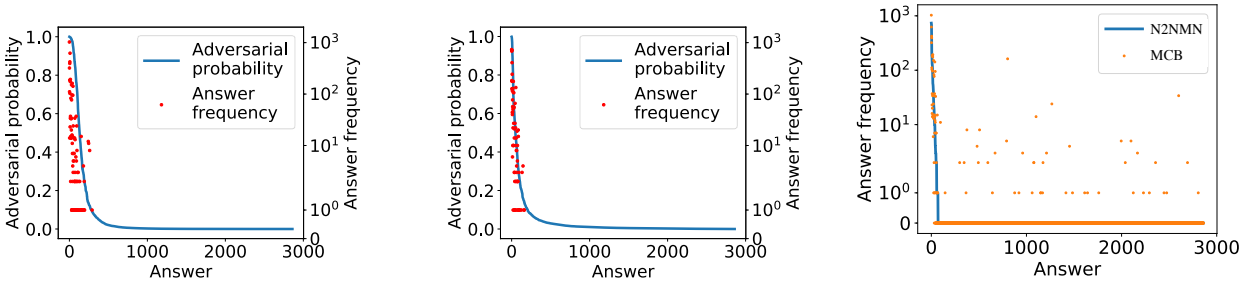
As we have observed in the experimental results described in Section 5 (in the main paper), whether a question-answer pair is a hard target depends more on the question-answer pair itself and less on the image. Therefore, we hypothesize that the language component in the VQA models may prevent adversarial examples to fool the models with certain targets. This phenomenon can be considered the *language prior* of VQA models. That is, given a question, if the model is less likely to predict a certain answer, we are also less likely to successfully generate targeted adversarial examples using it as the target answer.

In this section, we evaluate this phenomenon to verify our hypothesis. In particular, we choose a question, "What sport is this?". We first evaluate the **answer frequency** as follows. We run the VQA model on each of the 5,000 images in the VQA validation set and the selected question to get 5,000 answers. We compute the frequency of each answer in this set.

Intuitively, the answer frequency is a Monte-Carlo simulation of the answer distribution of the VQA model, and our goal is to examine the relationship between the answer distribution and the success of using an answer as the target to generate adversarial examples. In particular, we want to show that the answer frequency is positively correlated with the adversarial probability for each answer. To this end, we sequentially set each answer as the target answer, while setting the question chosen above (i.e., "what sport is this") as the target question, and Image 1 in Figure 11 as the benign image. Then we compute the adversarial probability of each answer. We sort all the answers in the descending order of their adversarial probabilities, and jointly plot the adversarial probabilities and the answer frequencies. Figures 12a and 12b show the corresponding plots for MCB and N2NMN. In these plots, each point in the x-axis indicates a label of an answer, so that the answer with the highest adversarial probability is labeled as 0, and so on. The blue line plots the adversarial probability of all answers, while the red dots plot the answer frequency. We only plot the answers whose frequency is at least 1, namely the answers must appear in the model's prediction set.

From both figures, we can observe a clear relationship between the answer frequency and the adversarial probability. That is, all answers with a frequency of 1 and higher can be predicted with a large probability (e.g., > 0.1), and all these answers can be used as targets to generate adversarial examples. Further, we observe that the answer frequency loosely aligns with the adversarial probability. This observation supports our hypothesis that the answer frequency is positively correlated with the adversarial probability.

Further, we observe that N2NMN has fewer answers



(a) MCB

(b) N2NMN

(c) Answer frequency of MCB and N2NMN

Figure 12: Answer frequency versus adversarial probability. Figure 12a and Figure 12b show that the answer frequency is positively correlated with the adversarial probability on MCB and N2NMN respectively. Figure 12c shows the answer frequency of the MCB model and the N2NMN model.

with a positive frequency. We illustrate this phenomenon in Figure 12c. In this figure, we sort all answers in the descending order of their frequencies based on the N2NMN model, and the x-axis corresponds to their rank. The blue plot shows the distribution of the answer frequency computed based on the N2NMN model, while the orange dots are each answer's frequency computed based on the MCB model. We can observe that many answers have a large frequency based on the MCB model, but their frequency based on the N2NMN model is 0. Therefore, combined with the observation above, this demonstrates that the N2NMN model has a smaller range of answers that can be used as the target to generate adversarial examples than the MCB model.

Notice that all these answers are generated based on the same questions. We investigate the results, and find that many of the answers predicted by the MCB model are irrelevant to the question used in this evaluation. This shows that, since N2NMN composes the network modules according to the input question, it is more effective at constructing corresponding filter modules, which can eliminate the answers irrelevant to the question. On the other hand, the MCB model does not have this functionality, since its architecture is identical throughout all questions. Therefore, when an image is less relevant to the question, the MCB model may predict answers considering the image more than the question. In this sense, the answer set of N2NMN is smaller than the one of MCB, since the former only includes answers relevant to the question. This also indicates that N2NMN has a stronger language prior than MCB, which partially explains why N2NMN behaves slightly more resilient than MCB in our previous experiments.

E.2. Meaningless question-answer targets

We further evaluate the effect of language prior by constructing a dataset of *meaningless* question-answer targets. We select 100 questions from 5 categories starting with (1) "What color"; (2) "What animal"; (3) "Is"; (4) "How many"; and (5) "Where". Then we construct the set of meaningful answers to each type of questions: for example, "silver" is a meaningful answer to a "what color" question. In doing so, the answer assigned to one type of question is guaranteed to be meaningless to the questions in another type. Thus we choose a meaningless answer for each of the 100 questions. We use them as targets and the 5 images used in Popular-QA and Rare-QA as the benign images to generate the adversarial examples. In the end, we observe that the attack success rates using our approach against MCB and N2NMN are only 7.8% and 4.6% respectively; the corresponding numbers for Carlini's attack are 6.8% and 3.8% respectively. This experiment further confirms the significance of the language prior and again demonstrates that N2NMN is more resilient against adversarial examples than MCB.