

Mimic and Fool: A Task Agnostic Adversarial Attack

Akshay Chaturvedi*, and Utpal Garain, *Member, IEEE*

arXiv:1906.04606v1 [cs.CV] 11 Jun 2019

Abstract—At present, adversarial attacks are designed in a task-specific fashion. However, for downstream computer vision tasks such as image captioning, image segmentation etc., the current deep learning systems use an image classifier like VGG16, ResNet50, Inception-v3 etc. as a feature extractor. Keeping this in mind, we propose Mimic and Fool, a task agnostic adversarial attack. Given a feature extractor, the proposed attack finds an adversarial image which can mimic the image feature of the original image. This ensures that the two images give the same (or similar) output regardless of the task. We randomly select 1000 MSCOCO validation images for experimentation. We perform experiments on two image captioning models, Show and Tell, Show Attend and Tell and one VQA model, namely, end-to-end neural module network (N2NNM). The proposed attack achieves success rate of 74.0%, 81.0% and 89.6% for Show and Tell, Show Attend and Tell and N2NNM respectively. We also propose a slight modification to our attack to generate natural-looking adversarial images. In addition, it is shown that the proposed attack also works for invertible architecture. Since Mimic and Fool only requires information about the feature extractor of the model, it can be considered as a gray-box attack.

Index Terms—Adversarial Attack, Task agnostic method, Vision and Language Systems, Deep Learning

I. INTRODUCTION

Adversarial attacks have shed light on the vulnerability of several state-of-the-art deep learning systems across varied tasks such as image classification, object detection, image segmentation etc. [1]–[4]. Recently, adversarial attacks were also proposed for multimodal tasks involving vision and language like image captioning and visual question answering (VQA) [5], [6]. Usually, these attacks fall under two categories: white-box and black-box. In white-box attack, the adversary has complete information about the model and its parameters. Whereas in black-box attack, the adversary has no information about the model that it wants to attack. Black-box attacks [7] are possible due to the *transferability* phenomenon of adversarial examples. Liu *et al.* [8] show that the adversarial examples designed for one image classification model can be transferred successfully to other classification models as well. Similarly, Xu *et al.* [6] show transferability of adversarial images between two state-of-the-art VQA models. However, the present-day adversarial attacks are task-specific in nature since a task-specific adversarial loss function is optimized to generate adversarial examples.

On the other hand, the current deep learning systems use output from intermediate layers of convolutional neural network (CNN) based image classification models (e.g. ResNet50



Fig. 1. Examples of Mimic and Fool. The first two rows show the original and adversarial images along with the predicted captions by Show and Tell and Show Attend and Tell respectively. The last row shows original and adversarial image for N2NNM (Q, P denote the question and the predicted answer respectively).

[9], VGG16 [10], Inception-v3 [11] etc.) as a feature for the input image. The rationale behind this approach is that the discriminative features learned by these classifiers are useful for other vision tasks as well. Hence, it is more beneficial to use these features instead of learning them *from scratch*. As a result, the aforementioned image classifiers function as *feature extractors*. Some deep learning systems also *fine-tune* the parameters of the *feature extractors* during training to make the image feature more suitable for the task in

* Corresponding author. Email: akshay91.isi@gmail.com

Akshay Chaturvedi, and Utpal Garain are with Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, India.

hand. However, fine-tuning is usually done if large amount of training data is available. Although using deep CNN-based image features give significant advantage to the present-day models, they have their own set of drawbacks. CNN-based feature extractors are known to be *non-invertible* [12], [13]. Mahendran and Vedaldi [13] show that AlexNet [14] maps multiple images to the same 1000-dimensional logits. These images are thus *indistinguishable* from the viewpoint of the last fully connected layer of AlexNet.

In this paper, we propose *Mimic and Fool*, a task agnostic adversarial attack, which exploits the non-invertibility of CNN-based *feature extractors* to attack the downstream model. Given a model and its feature extractor, the proposed attack is based on the simple hypothesis that if two images are *indistinguishable* for the *feature extractor* then they will be *indistinguishable* for the model as well. In other words, attacking the feature extractor by finding two indistinguishable images is equivalent to *hacking the eyes* of the model. As an example, consider an encoder-decoder architecture like Show and Tell [15], if we can successfully find two images which are mapped to the same feature by the encoder, then the two images will generate same (or similar) caption regardless of the decoder architecture. Thus to attack any model, attacking its feature extractor suffices. Based on this insight, *Mimic and Fool* finds an adversarial image which can *mimic* the feature of the original image thereby *fooling* the model. Figure 1 shows examples of *Mimic and Fool* on two captioning models: Show and Tell [15], Show Attend and Tell [16] and one VQA model: end-to-end neural module network (N2NMN) [17].

Since *Mimic and Fool* only requires the *fine-tuned* weights of the feature extractor to attack the model, it can be thought of as a *gray-box* attack. In fact, if a model does not *fine-tune* its feature extractor, *Mimic and Fool* can function as a black-box attack. This is because the number of possible feature extractors is limited. Hence, an adversary can generate an adversarial image per feature extractor knowing that one of these images is bound to fool the model. Furthermore, *Mimic and Fool* is extremely fast and requires less computing resources since only the feature extractor needs to be loaded in the memory instead of the model.

We perform experiments on two tasks: image captioning and visual question answering (VQA). We randomly choose 1000 MSCOCO [18] validation images and study the proposed attack on three models: Show and Tell, Show Attend and Tell, and N2NMN. We get 5208 image-question pairs from VQA v2.0 dataset [19] for the 1000 selected images. We choose these three models since they use different feature extractors. Show and Tell uses fully connected features from Inception-v3, Show Attend and Tell uses convolutional layer features from VGG16 and N2NMN uses features from a residual network [9]. Thus the three feature extractors vary from shallow to very deep helping us to validate our proposed attack for different types of feature extractors. We consider our attack successful if the model gives the same output for original and adversarial image.

A. Contributions of this work

The contributions of this paper are as follows: (i) We introduce the notion of a task agnostic attack. The proposed task agnostic attack, *Mimic and Fool*, achieves success rate of 74.0%, 81.0% and 89.6% for Show and Tell, Show Attend and Tell, N2NMN respectively. The high success rate for the three models validates our hypothesis that attacking the feature extractor suffices and also shows that the proposed attack works for different feature extractors. For image captioning models, we also compute the BLEU [20] and METEOR [21] score for the failure cases to show that even though the original and adversarial captions do not match exactly for these cases, they are very similar to each other. (ii) We also propose a modified version of our attack to generate *natural-looking* adversarial images. In the modified version, we start with a fixed natural image and restrict the amount of noise that can be added to the image. We use the same starting image for both the captioning models. Even in this restricted setting, the proposed attack achieves decent success rate. This shows that, by adding imperceptible noise to the fixed image, it is possible to find an adversarial image which can mimic image feature of any arbitrary image. This result is intriguing as it suggests that the feature extractors are very chaotic in nature. (iii) Since *Mimic and Fool* is task agnostic, while attacking a VQA model like N2NMN we need to run the attack for every image instead of every image-question pair. This is a huge advantage in terms of time saved for the adversary. The same will hold true for any future tasks which take multiple modalities as input with image being one of the modalities. (iv) At first glance, it seems that an invertible feature extractor will be resistant to the proposed attack. However, we show that the proposed attack also works for invertible architecture [22]. This shows that such architectures, despite being invertible, assign similar features to dissimilar images. Hence, invertibility is not a sufficient condition to safeguard the models against the proposed attack.

The rest of the paper is organized as follows. In Section II, we explain the proposed attack in detail and then we specify the training details and the hyperparameters used for the present work. In Section III, we analyze the results of *Mimic and Fool* on the three models. We also analyze the results of the proposed variant of *Mimic and Fool* on the two captioning models; Show and Tell and Show Attend and Tell. Furthermore, we study the proposed variant on an invertible architecture. Finally in Section IV, we give conclusions and related future work.

II. METHOD

A. Proposed Attack

In this section, we describe the proposed attack, *Mimic and Fool*, and *One Image Many Outputs* (OIMO) which is able to generate *natural looking* adversarial images. Since both the attacks are task agnostic, we describe the attack in terms of the feature extractor instead of the model.

1) ***Mimic and Fool***: Let $f : \mathbb{R}^{m \times n \times 3} \rightarrow \mathbb{R}^d$ denote the feature extractor of the model. Hence, d will be $14 \times 14 \times 1024$ if we extract *conv4* features from ResNet101 and d will be

2048 if we use output of average pooling layer of Inception-v3 as image feature.

Let $I_{org} \in [0, 255]^{m \times n \times 3}$ denote the original image. Given I_{org} and a feature extractor f , our goal is to find an adversarial image $I_{adv} \in [0, 255]^{m \times n \times 3}$ which can *mimic* the image features of I_{org} . We model this task as a simple optimization problem given by

$$\min_I \frac{\|f(\text{trunc}(I)) - f(I_{org})\|_2^2}{d} \quad (1)$$

where $\|\cdot\|_2$ denotes ℓ_2 -norm and *trunc* is truncating function which ensures that the intensity values lie in the range $[0, 255]$. Although $I = I_{org}$ is a solution to the above optimization problem, it is highly unlikely that the algorithm will converge to this solution. This is because convolutional neural networks discard significant amount of spatial information as we go from lower to higher layers. Mahendran and Vedaldi [13] show that the amount of invariance increases from lower to higher layer of AlexNet and regularizers like *total variation* (TV) are needed to reconstruct the original image from higher layer features of AlexNet. We start with a *zero-image* and run the proposed attack for max_{iter} iterations and return the final truncated image $\text{trunc}(I)$ as I_{adv} .

Some feature extractors such as Inception-v3 require the intensity values of the input image to be in the range $[-1, 1]$. In such a case, let $I'_{org} \in [-1, 1]^{m \times n \times 3}$ be the scaled original image i.e.

$$I'_{org} = 2(I_{org}/255) - 1 \quad (2)$$

For this case, we modify the optimization problem defined in Equation 1 as follows

$$\min_I \frac{\|f(\tanh(I)) - f(I'_{org})\|_2^2}{d} \quad (3)$$

where *tanh* ensures that the input to feature extractor lies within the required range. We run the attack for max_{iter} iterations and rescale the final image $\tanh(I)$ to get I_{adv} i.e.

$$I_{adv} = 255 \left(\frac{\tanh(I) + 1}{2} \right) \quad (4)$$

2) ***One Image Many Outputs***: In *One Image Many Outputs* (OIMO), we start with an image $I_{start} \in [0, 255]^{m \times n \times 3}$ instead of starting with *zero-image*. The image I_{start} is kept fixed throughout the experiment. In OIMO, our goal is to modify I_{start} so as to *mimic* the feature of I_{org} . Equation 1 is modified as follows

$$\min_\delta \frac{\|f(\text{trunc}(I_{start} + \delta)) - f(I_{org})\|_2^2}{d} \quad (5)$$

Similar to Chen *et al.* [5], we modify the Equation 3 as follows

$$\min_\delta \frac{\|f(\tanh(I''_{start} + \delta)) - f(I'_{org})\|_2^2}{d} \quad (6)$$

where $I''_{start} = \text{arctanh}(\lambda I'_{start})$, $I'_{start} \in [-1, 1]^{m \times n \times 3}$ is the scaled starting image, λ is set to 0.9999 to ensure invertibility of *tanh*, $\delta \in \mathbb{R}^{m \times n \times 3}$ is the learnable parameter.

For this attack, we reduce the value of max_{iter} and initial learning rate to ensure that I_{adv} looks very similar to I_{start} .

Similar to *Mimic and Fool*, after running the attack for max_{iter} iterations, I_{adv} for Equation 5 is $\text{trunc}(I_{start} + \delta)$. For Equation 6, I_{adv} is given by the following equation

$$I_{adv} = 255 \left(\frac{\tanh(I''_{start} + \delta) + 1}{2} \right) \quad (7)$$

We name the proposed attack *One Image Many Outputs* since all the adversarial images look very similar to I_{start} .

B. Implementation Details

As stated earlier, we study the proposed attack for two image captioning models; Show and Tell, Show Attend and Tell and one VQA model, namely, N2NMN. We train the N2NMN model on VQA v2.0 dataset for 95K iterations with expert policy followed by 65K iterations in policy search after cloning stage using the original source code¹. The trained N2NMN has 61.72% accuracy on VQAv2 test-dev set. For Show and Tell and Show Attend and Tell, we use already available trained models^{2,3}.

Show and Tell uses 2048-dimensional feature from Inception-v3, Show Attend and Tell uses $14 \times 14 \times 512$ feature map from VGG16, N2NMN uses output of *res5c* layer from ResNet-152 as image feature. The input images are of size $299 \times 299 \times 3$, $224 \times 224 \times 3$, $448 \times 448 \times 3$ for Inception-v3, VGG16 and ResNet-152 respectively. The trained Show and Tell, Show Attend and Tell *fine-tune* their respective feature extractors whereas N2NMN does not use *fine-tuning*.

For *Mimic and Fool*, we set max_{iter} to 1000, 1000 and 2000 for Inception-v3, VGG16 and ResNet-152 respectively. The initial learning rate is set to 0.025, 0.025 and 0.0125 for Inception-v3, VGG16 and ResNet-152 respectively. For *One Image Many Outputs*, we set max_{iter} to 300, 500 and set the initial learning rate to 0.0125 for Inception-v3 and VGG16. We use Adam [23] as the optimizer and Keras [24] for implementing the proposed attacks. All experiments are done on a single 11 GB GeForce GTX 1080 Ti GPU.

III. RESULTS

For studying the two proposed attacks, 1000 MSCOCO validation images are randomly selected. For the 1000 selected images, there are 5208 image-question pairs in VQA v2.0 dataset. The proposed attack is considered to be *successful* if the model gives the same output for the original and the adversarial image. Hence for image captioning, the two captions need to be exactly the same for the attack to be successful. In the following subsections, we analyze the behavior of the two proposed attacks on the three models: N2NMN, Show and Tell and Show Attend and Tell.

¹<https://github.com/ronghanghu/n2nmn>

²<https://github.com/KranthiGV/Pretrained-Show-and-Tell-model>

³https://github.com/DeepRNN/image_captioning

Task	Model	Feature Extractor	Success Rate	Average Time for 1000 iterations
Image Captioning	Show and Tell	Inception-v3	74.0 %	25.35 sec
	Show Attend and Tell	VGG16	81.0 %	15.56 sec
VQA	N2NMN	ResNet-152	89.6 %	72.98 sec

TABLE I
SUCCESS RATE OF *Mimic and Fool*

A. Results for *Mimic and Fool*

Table I shows the success rate of *Mimic and Fool* for the three models. For N2NMN, out of 5208 image question pairs, *Mimic and Fool* is successful for 4665 image question pairs. This yields success rate of 89.6%. The high success rate shows that it is possible to *mimic* features extracted from a very deep network like ResNet-152 as well. Since *Mimic and Fool* is *task-agnostic*, we need to run the proposed attack at image level instead of image-question pair level. Hence, we only need to run the attack 1000 times instead of 5208 times. This is a huge advantage since it results in a drastic reduction in time. The advantage will be even more pronounced for any future tasks which have multiple modalities as input with image (or video) being one of the modalities. Figure 2 shows the predicted answer by N2NMN for different image-question pairs. From Figure 2, we can see that a single adversarial image suffices for three image-question pairs.

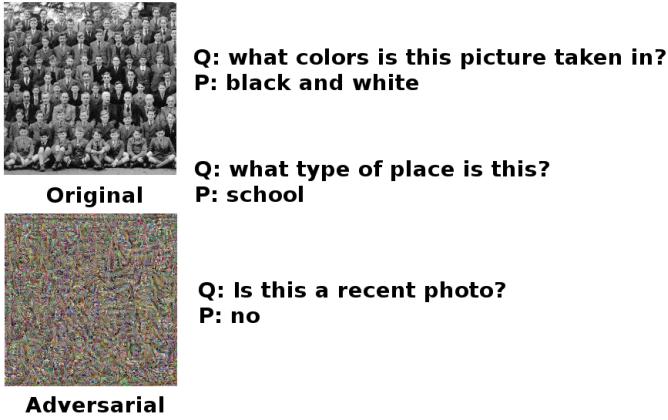


Fig. 2. Example of *Mimic and Fool* for N2NMN. Single adversarial image suffices for three image-question pairs. Q and P denote the question and the predicted answer respectively.

As we can see from Table I, *Mimic and Fool* is very fast. The attack only takes around 25 seconds for generating adversarial images for Show and Tell. The time taken for Show, Attend and Tell is even less since VGG16 is a shallower network. The proposed attack achieves success rate of 74.0% and 81.0% for Show and Tell and Show Attend and Tell respectively. This is especially encouraging result since generating exactly the same caption for an adversarial image is a very challenging task. This is because, as observed by Chen *et al.* [5], the number of possible captions are infinite which makes a captioning system harder to attack than an image classifier. Our results show that in order to generate the same caption, it suffices to attack just the encoder of the captioning model. This validates our initial hypothesis that in order to attack any model, attacking

its feature extractor suffices. For the unsuccessful cases, the predicted captions for original and adversarial images are very similar. Figure 4 shows two successful and one unsuccessful examples of *Mimic and Fool* for Show and Tell and Show Attend and Tell. As we can see from Figure 4 that for the unsuccessful cases, the predicted captions for the original and adversarial images have a large amount of overlap. We also calculate the BLEU and METEOR score, using the pipeline provided by Sharma *et al.* [25], for unsuccessful adversarial cases as shown in Table II. We use the predicted caption for the original image as reference while calculating these metrics.

B. Results for One Image Many Outputs

The main idea behind *One Image Many Outputs* is to generate natural-looking adversarial images. We randomly choose an image from MSCOCO training set as the starting image. Figure 3 shows the starting image (I_{start}) for *One Image Many Outputs* along with the predicted captions of Show And Tell and Show Attend and Tell.



Fig. 3. I_{start} for *One Image Many Outputs* and the predicted captions.

In *One Image Many Outputs*, we reduce the value of max_{iter} and the initial learning rate to ensure that the adversarial image I_{adv} looks very similar to I_{start} . Reduction in max_{iter} results in even faster running time than *Mimic and Fool*. For both the captioning models, *One Image Many Outputs* takes under 10 seconds per image.

Table III shows the success rate of *One Image Many Outputs* for Show and Tell and Show Attend and Tell. As we can see from Table I and Table III, the success rate reduces for *One Image Many Outputs* in comparison to *Mimic and Fool*. This is intuitive since in *One Image Many Outputs*, the reduced value of max_{iter} and initial learning rate allows for less adversarial noise.

Considering this reduction and the fact that the attack is successful only when there is an exact match of captions, the success rate of *One Image Many Outputs* is impressive. Similar to *Mimic and Fool*, we find that for the unsuccessful cases of *One Image Many Outputs*, the captions predicted by the model for the adversarial and original images are very similar to each

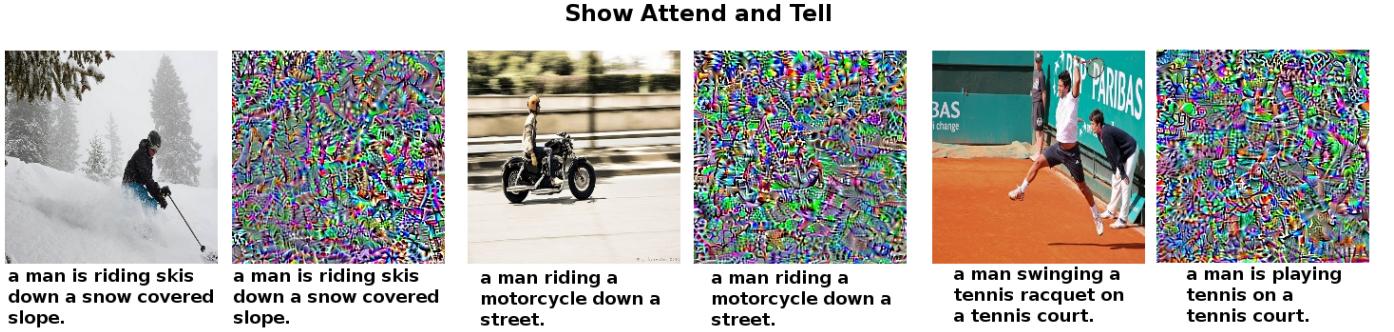
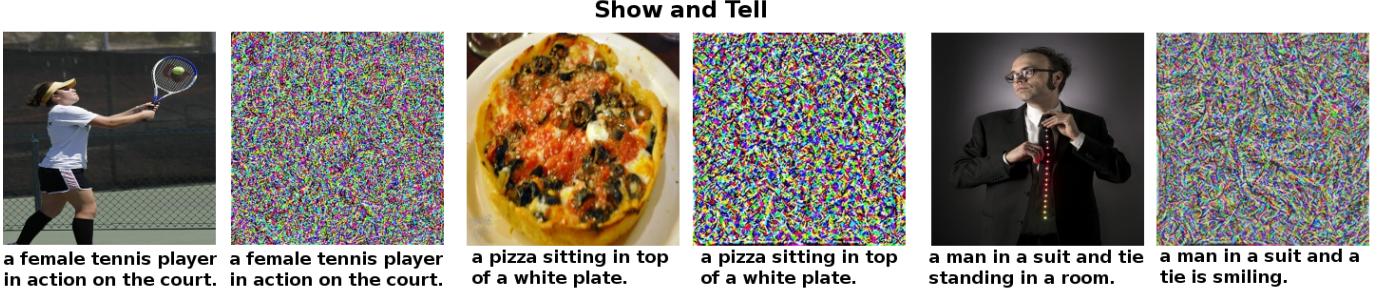


Fig. 4. Examples of *Mimic and Fool*. For both the captioning models, the figure shows two successful and one unsuccessful original and adversarial image along with the predicted captions.



Fig. 5. Examples of *One Image Many Outputs*. For both the captioning models, the figure shows two successful and one unsuccessful original and adversarial image along with the predicted captions. For adversarial images, the captions within brackets are predicted by the other captioning model i.e. for Show and Tell, the captions within brackets are captions predicted by Show Attend and Tell and vice-versa.

other. Table II shows the BLEU and METEOR score for the unsuccessful cases of *One Image Many Outputs*. This result

shows that even when I_{adv} is very similar to I_{start} , it can mimic features of an arbitrary image. This shows that CNN-

Model	Attack	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Show and Tell	Show-and-Fool	0.560	0.394	0.266	0.205	0.301
	Mimic and Fool	0.597	0.464	0.348	0.264	0.320
	OIMO	0.593	0.459	0.350	0.270	0.322
Show Attend and Tell	Mimic and Fool	0.639	0.530	0.421	0.333	0.368
	OIMO	0.594	0.468	0.359	0.284	0.336

TABLE II
BLEU AND METEOR SCORES FOR UNSUCCESSFUL CASES. OIMO REFERS TO *One Image Many Outputs*.

Model	Success Rate
Show and Tell	56.9 %
Show Attend and Tell	50.3 %

TABLE III
SUCCESS RATE OF *One Image Many Outputs*

based feature extractors are chaotic in nature.

Figure 5 shows two successful and one unsuccessful examples of *One Image Many Outputs* for Show and Tell and Show Attend and Tell. For the adversarial images in Figure 5, the captions within brackets are the predicted captions by the other captioning model i.e. the captioning model which is not under attack. As we can see from Figure 5, all the six adversarial images are very similar to the starting image, I_{start} . Also for the unsuccessful cases, the original and adversarial captions have a large amount of overlap and are semantically similar.

In Figure 5, we see that for Show and Tell, the captions within brackets, i.e. the captions predicted by Show Attend and Tell, for the three adversarial images are the same. Similarly for Show Attend and Tell, although the captions within the brackets (captions predicted by Show and Tell) are different, they are semantically similar. Moreover, for both the captioning models, all the captions within brackets are relevant captions for the starting image, I_{start} . In fact, we find that when the 1000 adversarial images for Show And Tell are given as input to Show Attend and Tell, there are only 15 unique captions. All these 15 captions are relevant captions for I_{start} . Similarly, when the 1000 adversarial images for Show Attend and Tell are given as input to Show and Tell, there are only 82 unique captions, most of which are relevant to I_{start} . We find that Show and Tell generates irrelevant captions for I_{start} only for 32 out of 1000 adversarial images. Since the two captioning models use different feature extractors, this result shows that the proposed attack is very dependent on the feature extractor. In other words, ensuring that the two images are *indistinguishable* for one feature extractor does not ensure that they will be *indistinguishable* for another feature extractor. More examples of the two proposed attacks can be seen at <https://www.isical.ac.in/~utpal/resources.php>.

C. Comparison with task specific attack

In this section, we compare our proposed attack, OIMO, with a white-box attack for Show and Tell, namely Show-and-Fool [5]. Similar to OIMO, we start with I_{start} and run the attack, using the official implementation⁴, in order to generate

adversarial captions. Show-and-Fool achieves a success rate of 95.1% and takes around 190 seconds per image. In contrast, OIMO takes around 8 seconds per image for Show and Tell. Although our proposed attack achieves a lower success rate, we can see from Table II that for unsuccessful cases, OIMO achieves higher BLEU and METEOR score than Show-and-Fool. We find that for the adversarial images generated by Show-and-Fool, Show Attend and Tell generates only 11 unique captions, all of which are relevant captions for I_{start} . Chen *et al.* [5] study the transferability of Show-and-Fool between the captioning models, however in their study, the two captioning models use the same feature extractor.

D. OIMO for invertible architecture

Recently, Jacobsen *et al.* [22] propose a deep invertible architecture, i-RevNet which learns a one-to-one mapping between image and its feature. These networks achieve impressive accuracy on ILSVRC-2012 [26]. For experimentation, we choose bijective i-RevNet which takes images of size $224 \times 224 \times 3$ as input and the corresponding feature is of size $3072 \times 7 \times 7$. We use the pretrained i-RevNet provided in the official implementation⁵ to test our proposed attack, *One Image Many Outputs*. We randomly choose 100 correctly classified images belonging to 41 different classes from the validation set of ILSVRC-2012. Furthermore, we choose a starting image, I_{start} , belonging to a different class. We also restrict the search space for adversarial images using the clipping function $Clip_{I_{start}, \epsilon}$ (i.e. $\epsilon \ell_\infty$ -neighborhood of I_{start}). Starting with $I_{start} \in [0, 255]^{224 \times 224 \times 3}$, we run the proposed attack, OIMO, in order to *mimic* the feature for 100 images. Table IV shows the success rate for different values of ϵ . The high success rate shows that the proposed attack can be applied for invertible architecture like i-RevNet as well. This is because i-RevNet, despite being invertible, assigns similar features to dissimilar images. Figure 6 shows one such successful adversarial example.

ϵ	Success Rate
2	86.0 %
5	99.0 %
10	100.0 %

TABLE IV
SUCCESS RATE OF *One Image Many Outputs* FOR I-REVN

⁴<https://github.com/IBM/Image-Captioning-Attack>

⁵<https://github.com/jhjacobsen/pytorch-i-revnet>



Fig. 6. Both the images are classified as *ice bear* by bijective i-RevNet.

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed a task agnostic adversarial attack, *Mimic and Fool*. The proposed attack exploits the non-invertibility of CNN-based feature extractors and is based on the hypothesis that if two images are *indistinguishable* for the *feature extractor* then they will be *indistinguishable* for the model as well. The high success rate of *Mimic and Fool* for three models across two tasks validates this hypothesis. Since the three models use feature extractors varying from shallow to very deep, the high success rate also shows that the proposed attack works regardless of the depth of the feature extractor. Also due to the task-agnostic nature, we need to run the attack only at image-level which is a huge advantage in terms of time saved for tasks involving multiple modalities as input. We further propose a variant of *Mimic and Fool*, named *One Image Many Outputs*, which generates *natural-looking* adversarial images. The results for this variant of the attack show that it is possible to *mimic* features of an arbitrary image by making imperceptible changes to a fixed image as well. This is an important insight into the nature of CNN-based feature extractors. We also demonstrate the applicability of the proposed attack for invertible architectures like iRevNet.

In the present version of *Mimic and Fool*, the adversary needs information about the feature extractor of the model and also needs access to the *fine-tuned* weights of the feature extractor. We found that using *pretrained* instead of *fine-tuned* weights leads to drop in success rate of the proposed attack. This makes the proposed attack a *gray-box* attack. As part of future work, one can explore different attack strategies which will work successfully with just the *pretrained* weights.

REFERENCES

- [1] S. Chen, C. Cornelius, J. Martin, and D. H. Chau, “Robust physical adversarial attack on faster R-CNN object detector,” *CoRR*, vol. abs/1804.05810, 2018.
- [2] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *International Conference on Learning Representations (ICLR)*, 2015.
- [3] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in *ICLR Workshop*, 2017.
- [4] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. L. Yuille, “Adversarial examples for semantic segmentation and object detection,” *IEEE International Conference on Computer Vision (ICCV)*, pp. 1378–1387, 2017.
- [5] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh, “Attacking visual language grounding with adversarial examples: A case study on neural image captioning,” in *Proc. of 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 2587–2597.
- [6] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darrell, and D. Song, “Fooling vision and language models despite localization and attention mechanism,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [7] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proc. of ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS, 2017.
- [8] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” in *Proc. of 5th International Conference on Learning Representations (ICLR)*, 2017.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016, pp. 770–778.
- [10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.
- [11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [12] A. Dosovitskiy and T. Brox, “Inverting visual representations with convolutional networks,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016, pp. 4829–4837.
- [13] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5188–5196.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3156–3164.
- [16] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proc. of 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 2048–2057.
- [17] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, “Learning to reason: End-to-end module networks for visual question answering,” in *IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2017, pp. 804–813.
- [18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Proc. European Conference on Computer Vision (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., 2014, pp. 740–755.
- [19] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, “Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.
- [21] S. Banerjee and A. Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments,” in *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005, pp. 65–72.
- [22] J.-H. Jacobsen, A. W. M. Smeulders, and E. Oyallon, “i-revnet: Deep invertible networks,” in *International Conference on Learning Representations (ICLR)*, 2018.
- [23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015.
- [24] F. Chollet *et al.*, “Keras,” <https://keras.io>, 2015.
- [25] S. Sharma, L. El Asri, H. Schulz, and J. Zumer, “Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation,” *CoRR*, vol. abs/1706.09799, 2017.
- [26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.