Fooling Network Interpretation in Image Classification

Akshayvarun Subramanya* Vipin Pillai* Hamed Pirsiavash
University of Maryland, Baltimore County
{akshayv1, vp7, hpirsiav}@umbc.edu

Abstract

Deep neural networks have been shown to be fooled rather easily using adversarial attack algorithms. Practical methods such as adversarial patches have been shown to be extremely effective in causing misclassification. However, these patches are highlighted using standard network interpretation algorithms, thus revealing the identity of the adversary. We show that it is possible to create adversarial patches which not only fool the prediction, but also change what we interpret regarding the cause of the prediction. Moreover, we introduce our attack as a controlled setting to measure the accuracy of interpretation algorithms. We show this using extensive experiments for Grad-CAM interpretation that transfers to occluding patch interpretation as well. We believe our algorithms can facilitate developing more robust network interpretation tools that truly explain the network's underlying decision making process.

1. Introduction

Deep learning has achieved great results in many domains including computer vision. However, it is still far from being deployed in many real-world applications due to reasons including:

(1) Explainable AI (XAI): Explaining the prediction of deep neural networks is a challenging task because they are complex models with large number of parameters. Recently, XAI has become a trending research area in which the goal is to develop reliable interpretation algorithms that explain the underlying decision making process. Designing such algorithms is a challenging task and considerable work [28, 35, 26] has been done to describe *local explanations* - explaining the model's output for a given input [4].

(2) Adversarial examples: It has been shown that deep neural networks are vulnerable to adversarial examples. These carefully constructed samples are created by adding imperceptible perturbations to the original input for changing the final decision of the network. This is important for two reasons: (a) Such vulnerabilities could be used by adversaries to fool AI algorithms when they are deployed in real-world applications such as Internet of Things (IoT) [24] or self-driving cars [29] (b) Studying these attacks can lead to better understanding of how deep neural networks work and also possibly better generalization.

In this paper, we design adversarial attack algorithms that not only fool the network prediction but also fool the network interpretation. Our main goal is to utilize such attacks as a tool to investigate the reliability of network interpretation algorithms. Moreover, since our attacks fool the network interpretation, they can be seen as a potential vulnerability in the applications that utilize network interpretation to understand the cause of the prediction (e.g., in health-care applications [8].)

Reliability of network interpretation: We are interested in studying the reliability of the interpretation in highlighting true cause of the prediction. To this end, we use the adversarial patch method [5] to design a *controlled* adversarial attack setting where the adversary changes the network prediction by manipulating only a small region of the image. Hence, we know that the cause of the wrong prediction should be inside the patch. We show that it is possible to optimize for an adversarial patch that attacks the prediction without being highlighted by the interpretation algorithm as the cause of the wrong prediction.

Grad-CAM [26] is one of the most well-known network interpretation algorithms that performs well on sanity check among state-of-the-art interpretation algorithms recently studied in [1]. Hence, we choose to study the correctness of Grad-CAM as a case study. Also, we show that our results even though tuned for Grad-CAM, can transfer directly to Occluding Patch [34] as another interpretation algorithm.

As an example, in Figure 1, the original image (left) is correctly classified as "French Bulldog". On the top row, a targeted adversarial patch has successfully changed the prediction to "Soccer Ball". Since the adversary is able to manipulate only the pixels inside the patch, it is expected that the interpretation algorithm (e.g, Grad-CAM) for "Soccer Ball" category should highlight some patch pixels

*Equal contribution

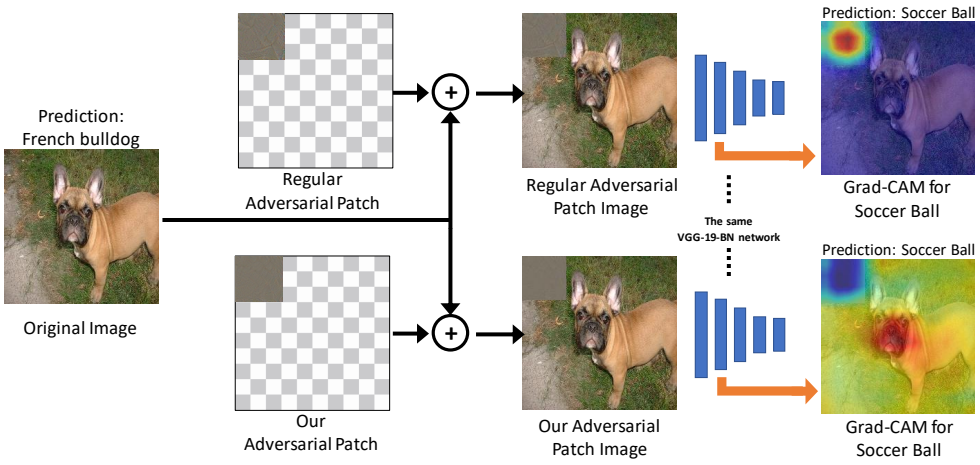

Figure 1: We show that Grad-CAM highlights the patch location in the image perturbed by regular targeted adversarial patches [5] (top row). Our modified attack algorithm goes beyond fooling the final prediction by also fooling the Grad-CAM visualization. Here, Grad-CAM is used to visualize the cause of the target category.

as the cause of the wrong prediction. This is shown in the first row, top-right image. However, in the bottom row, our adversarial patch algorithm, not only changes the prediction to "Soccer Ball", but also does it in a way that Grad-CAM does not highlight the pixels inside the patch. We argue that since the adversary can change only the patch pixels, we know that the cause of the wrong prediction should be inside the patch. So, the observation that Grad-CAM does not highlight the patch pixels reveals that Grad-CAM is not reliably highlighting the source of prediction. Note that in this setting, the target category is not chosen by the model and is randomly chosen by the adversary from all possible wrong categories (i.e., uniformly from 999 categories of ImageNet). We believe this shows that the Grad-CAM algorithm is not *necessarily* showing the true cause of the prediction.

We optimize the patch by adding a new term in the optimization of adversarial patches that suppresses Grad-CAM activation at the location of the patch while still encouraging the wrong prediction (target category). We believe our algorithms can be used as a form of evaluation for future interpretation algorithms.

Practical implications: Our attack is more practical since we are manipulating only a patch and hence is closer to real world applications. As a practical example, some applications in health-care are not only interested in the prediction, but also understanding the cause of it (e.g., what region of a medical image of a patient causes diagnosis of cancer.) We believe our attacks can be generalized beyond object classification to empower an adversary to manipulate the reasoning about some medical diagnosis. Such an attack can cause serious issues in health-care, for instance by manipulating medical records to charge insurance companies [11].

Our key contributions are summarized as follows:

(1) We introduce a novel algorithm to construct adversarial patches which fool both the classifier and the interpretation of the resulting category.

(2) With extensive experiments, we show that our method (a) generalizes from Grad-CAM to Occluding Patch [34], another interpretation method, (b) generalizes to unseen images (universal), (c) is able to fool GAIN [22], a model specifically trained with supervision on interpretation, and (d) is able to make the interpretation uniform to hide any signature of the attack.

(3) We use these attacks as a tool to assess the reliability of Grad-CAM, a popular network interpretation algorithm. This suggests that the community needs to develop more robust interpretation algorithms possibly using our tool as an evaluation method.

2. Related work

Adversarial examples: Adversarial examples were discovered by Szegedy *et al*. [32] who showed that state-of-the-art machine learning classifiers can be fooled comprehensively by simple backpropagation algorithms. Goodfellow *et al*. [13] improved this by Fast Gradient Sign Method (FGSM) that needs only one iteration of optimization. The possibility of extending these examples to the real world was shown in [20, 27] and [3] showed that adversarial examples could be robust to affine transformations as well. Madry *et al*. [23] proposed Projected Gradient Descent (PGD) which has been shown to be the best first-order adversary for fooling classifiers. Chen *et al*. [7] show that physical world adversarial examples can be created for object detection networks such as Faster R-CNN. Zając *et al*. [33] also showed that instead of modifying the image, a frame can be placed around the image to fool classifiers. Su *et al*. [31] showed that modifying one pixel using Differential Evolution (DE) is suffi-

cient to fool classifiers. Although there have been many proposed defense algorithms, most of them have been overcome by making changes to the attack algorithm as shown in [6, 2]. Training robust networks is an important problem that can lead to better understanding of neural networks and also improve their generalization capabilities.

Adversarial patches: Adversarial patches [5, 18] were introduced as a more practical version of adversarial attacks where we restrict the spatial dimensions of the perturbation, but remove the imperceptibility constraint. These patches can be printed and 'pasted' on top of an image to mislead classification networks. Recently, [30] showed that physical adversarial examples and adversarial patches can be created for object detection algorithms as well. We improve this by ensuring that the patches fool network interpretation tools that try to understand the reasoning for misclassification.

Interpretation of deep neural networks: As neural networks are getting closer towards deployment in real world applications, it is important that their results are interpretable. Doshi-Velez *et al*. [10] discuss the legal and societal implications of explainable AI and suggest that although explainable systems might possibly be sub-optimal, it is a necessity that needs to be considered under design. This becomes extremely relevant when machine learning is used in biology, where it is essential to ensure the model's decision-making process is reliable and is not due to an artifact of the data (See Discussion in [8]). So it is important to make sure that the deep neural networks can be explained using robust and reliable interpretation algorithms which can ensure transparency in the network's explanation. Researchers have proposed various algorithms in this direction. One of the earliest attempt [28] calculates the derivative of the network's outputs w.r.t the input to compute class specific saliency maps. Zhou *et al*. [34] calculates the change in the network output when a small portion of the image (11×11 pixels) is covered by a random occluder. We call this **Occluding Patch**. CAM [35] used weighted average map for each image based on their activations. The most popular one that we consider in this paper is called **Grad-CAM** [26], a gradient based method which provides visual explanations for any neural network architecture. Li *et al*. [22] recently improved upon Grad-CAM using Guided attention mechanism with state-of-the-art results on PASCAL VOC 2012 segmentation task. Although the above methods have shown great improvement in explaining the network's decision, our work highlights that it is important to ensure that they are robust enough to adversaries as well.

Attacking network interpretation: Ghorbani *et al*. [12] introduce adversarial perturbations that produce perceptively indistinguishable inputs that are assigned the *same* predicted label, yet have very *different* interpretations. However, in this setting, the adversarial image after perturbation can have image regions which correspond to stronger features for the same predicted label and as a result lead to different interpretations by dominating the prediction score. This is also noted in the discussion section in [12]. To mitigate this, we design a *controlled* setting using adversarial patches where the adversary changes the network prediction by manipulating only a small region of the image. Here, we clearly know that the interpretation for the wrong prediction should be inside the patch. Heo *et al*. [16] introduce a threat model wherein the adversary can modify the model parameters to fool the network interpretation. However, in a practical setting, the adversary might not always be able to modify the model parameters. Hence, we are interested in modifying only the pixels in a small image area without altering the model. Kindermans *et al*. [19] showed how saliency methods are unreliable by adding a constant shift to input data and checking against different saliency methods. Adebayo *et al*. [1] introduce sanity checks to evaluate existing saliency methods and show that some of them are independent of both the model and the data generating process. We believe our method can serve as an additional evaluation for future interpretation algorithms.

3. Method

We propose algorithms to learn adversarial patches that when pasted on the input image, can change the interpretation of the model's prediction. We will focus on Grad-CAM [26] in designing our algorithms and then, show that our results generalize to other interpretation algorithms as well.

Background on Grad-CAM visualization

Consider a deep network for image classification task, e.g., VGG, and an image x_0. We feed the image to the network and get the final output y where y^c is the logit or class-score for the c'th class. To interpret the network's decision for category c, we want to generate heatmap G^c for a convolutional layer, e.g, *conv5*, which when up-sampled to the size of input image, highlights the regions of the image that have significant effect in producing higher values in y^c. We denote A^k_{ij} as the activations of the k'th neuron at location (i, j) of the chosen layer. Then, as in [26], we measure the effect of each feature of the convolutional layer at the final prediction by:

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

where Z is a normalizer. Then we calculate the interpretation (heatmap) as the weighted sum of activations of the convolutional layer discarding the negative values:

$$G^c_{ij} = max(0, \sum_k \alpha^c_k A^k_{ij})$$

We then normalize the heatmap: $\hat{G}^c := \frac{G^c}{|G^c|_1}$

Background on adversarial patches

Consider an input image x and a predefined constant binary mask m that is 1 on the location of the patch (top left corner in the experiments of Figure 1) and 0 everywhere else. We want to find an adversarial patch z that changes the output of the network to category t when pasted on the image, so we solve:

$$z = \arg\min_z \ell_{ce}(x \odot (1-m) + z \odot m; t)$$

where $\ell_{ce}(.;t)$ is the cross entropy loss for the target category t and $\odot$ is the element-wise product. Note that for simplicity of the notation, we assume z has the same size as x, but only the patch location is involved in the optimization. This results in adversarial patches similar to [5].

3.1. Fooling interpretation with targeted patches

We now build upon the Grad-CAM method and adversarial patches explained in the preceding section to design our controlled setting that lets us study the reliability of network interpretation algorithms. As shown in Figure 1, when an an image is attacked by an adversarial patch, Grad-CAM of the target category (wrong prediction) can be used to investigate the cause of the misclassification. It highlights the patch very strongly revealing the cause of the attack. This is expected as the adversary is restricted to perturbing only the patch area and the patch is the cause of the final misclassification towards target category.

In order to hide the adversarial patch in the interpretation of the final prediction, we add an additional term to our loss function while optimizing the patch such that the heatmap of the Grad-CAM interpretation at the patch location m is suppressed. Hence, assuming the perturbed image $\tilde{x} = x_0 \odot (1-m) + z \odot m$, we optimize:

$$\arg\min_z \Big[\ell_{ce}(\tilde{x}; t) + \lambda \sum_{ij} \big(\hat{G}^t(\tilde{x}) \odot m\big)\Big] \qquad (1)$$

where t is the target category and λ is the hyper-parameter to trade-off the effect of two loss terms. We choose the target label randomly across all classes excluding the original prediction similar to "step rnd" method in [21].

To optimize the above loss function, we use an iterative approach similar to projected gradient decent (PGD) algorithm [23]. We initialize z randomly and iteratively update it by: $z^{n+1} = z^n - \eta Sign(\frac{\partial \ell}{\partial z})$ with learning rate η. At each iteration, we project z to the feasible region by clipping it to the dynamic range of image values.

We argue that if this method succeeds in fooling the Grad-CAM to not highlight the adversarial patch location, it means the Grad-CAM algorithm is not showing the true cause of the attack since we know the attack is limited to the patch location only.

3.2. Non-targeted patches

A similar approach can be used to develop a non-targeted attack by maximizing the cross entropy loss of the correct category. This can be considered a weaker form of attack since the adversary has no control over the final category which is predicted after adding the patch. In this case, our optimization problem becomes:

$$\arg\min_z \Big[\max(0, M - \ell_{ce}(\tilde{x}; c)) + \lambda \sum_{ij} \big(\hat{G}^a(\tilde{x}) \odot m\big)\Big] \qquad (2)$$

where c is the predicted category for the original image, $a = \arg\max_k y(k)$ is the top prediction at every iteration, and $y(k)$ is the logit for category k. Since cross entropy loss is not upper-bounded, it can dominate the optimization, so we use contrastive loss [14] to ignore cross entropy loss when the probability of c is less than the chance level, thus $M = -log(p_0)$ where p_0 is the chance probability (e.g., 0.001 for ImageNet). Note that the second term is using the interpretation of the current top category a.

3.3. Targeted regular adversarial examples

We now consider regular adversarial examples (non-patch) [32] where the ℓ_∞ norm of the perturbation is restricted to a small ϵ, (e.g. 8/255) which fools both the network prediction and the network interpretation. To this end, in Eq. 1, we expand mask m to cover the whole image and initialize z from x. For completeness, we report the results of such attacks in our experiments, but as noted in the related work section, they do not necessarily show that the interpretation method is wrong.

3.4. Universal targeted patches

Universal attack is a much stronger form of attack wherein we train a patch that generalizes across images in fooling towards a particular category. Such an attack shows that it is possible to fool an unknown test image using a patch learned using the training data. This is a more practical form of attack, since the adversary needs to train the patch just once, which would be strong enough to fool multiple unseen test images. To do so, we optimize the summation of losses for all images in our training data using mini-batch gradient descent for:

$$\arg\min_z \sum_{n=1}^{N} \Big[\ell_{ce}(\tilde{x}_n; t) + \lambda \sum_{ij} \big(\hat{G}^t(\tilde{x}_n) \odot m\big)\Big] \qquad (3)$$

4. Experiments

We perform our experiments in two different benchmarks. We use VGG19 network with batch normalization,

Method	Top-1 Acc(%)	Non-Targeted		Targeted		
		Acc (%)	Energy Ratio (%)	Acc (%)	Target Acc (%)	Energy Ratio(%)
Adversarial Patch [5]	74.24	0.06	50.87	0.02	99.98	76.26
Our Patch	74.24	0.05	**2.61**	2.95	77.88	**6.80**

Table 1: Comparison of heatmap energy within the 8% patch area for the adversarial patch [5] and our patch. We use an ImageNet pretrained VGG19-BN model on 50,000 images of the validation set of ImageNet dataset. Accuracy denotes the fraction of images that had the same final predicted label as the original image. Target Accuracy denotes the fraction of images where the final predicted label has changed to the randomly chosen target label.

ResNet-34 and DenseNet-121 as standard network architectures. We use ImageNet [9] ILSVRC2012 for these experiments.

Then to evaluate our attack in a more challenging setting, we use GAIN$_{ext}$ model from [22] which is based on VGG19 (without batch normalization), but is specifically trained with supervision on the network attention to provide more accurate interpretation. We use PASCAL VOC-2012 dataset for these experiments since GAIN$_{ext}$ uses semantic segmentation annotation and its pre-trained model is available only for this dataset.

4.1. Evaluation

We use standard classification accuracy to report the success rate of the attack and use the following metrics to measure the success of fooling interpretation:

(a) Energy Ratio: We normalize the interpretation heatmap to sum to one for each image, and then calculate the ratio of the total energy of the interpretation at the patch location to that of the whole image. We call this metric "Energy Ratio". It will be 0 if the patch is not highlighted at all and 1 if the heatmap is completely concentrated inside the patch. In the case of a uniform heatmap, the energy ratio will be 8.2% (the relative area of the patch).

(b) Histogram Intersection: To compare two different interpretations, we calculate the Grad-CAM heatmap of the original image and the adversarial image, normalize each to sum to one per image, and calculate the histogram intersection between them.

(c) Localization: We use the metric from the object localization challenge of ImageNet competition. Similar to the Grad-CAM paper, we draw a bounding box around values larger than a threshold (0.15 as used in [26]), and evaluate object localization by comparing the boxes to the ground-truth bounding boxes.

We assume input images of size 224×224 and patches of size 64×64 which occupy almost 8.2% of the image area. We place the patch on the top-left corner of the image for most experiments so that it does not overlap with the main objects of interest. We use PyTorch [25] along with NVIDIA Titan-X GPUs for all experiments.

4.2. Targeted adversarial patches

For the adversarial patch experiments described in the method section, we use 50,000 images of the validation set of ImageNet ILSVRC2012 [9]. We perform 750 iterations of optimization with $\eta = 0.005$ and $\lambda = 0.05$. We use the Energy Ratio metric for evaluation. The results in Table 1 show that our patch has significantly less energy in the patch area. However, this comes with some reduction in the targeted attack accuracy which can be attributed to the increased difficulty of the attack. Figure 2 shows the qualitative results.

4.3. Non-targeted adversarial patches

Here, we perform the non-targeted adversarial patch attack using 50,000 images of the validation set of ImageNet [9] ILSVRC2012. We perform 750 iterations with $\eta = 0.005$ and $\lambda = 0.001$. The results are shown in Table 1 and the qualitative results are included in the supplementary material.

4.4. Different networks and patch locations

Most of our experiments use models based on VGG19-BN and also place the patch on the top left-corner of the image. In this section, we evaluate our targeted adversarial patch attack algorithm on ResNet-34 [15] and DenseNet-121 [17] by placing the patch on the top-right corner of the image. Similar to VGG experiments, both models are pre-trained on ImageNet dataset. We use 5,000 random images from the ImageNet validation set to evaluate these attacks using the Energy Ratio metric which is presented in Table 2. Our patch fools the interpretation while reaching the target category in more than 90% of the images. The qualitative results for these experiments are included in the supplementary material.

4.5. Uniform heatmap patches

One may argue that our attacks may not be effective in practice to fool the manual investigation of the network output since the lower (blue) heatmap of the Grad-CAM can still be considered as a distinguishable signature (see Figure 2). We mitigate this concern by optimizing the patch to encourage higher values of Grad-CAM outside the patch area (top-right corner instead of the patch area which is at the

Method	Targeted	
	Target Acc (%)	Energy Ratio (%)
Adv. Patch (R-34)	100.0	61.9
Our Patch (R-34)	90.3	**8.2**
Adv. Patch (D-121)	99.9	71.3
Our Patch (D-121)	93.6	**5.3**

Table 2: Comparison of Grad-CAM heatmap energy within the 8% patch area (placed at the top-right corner) for different networks on 10% randomly sampled ImageNet validation images. R-34 and D-121 refer to ResNet-34 and DenseNet-121 models respectively.

top-left corner). Our results in Table 3 and Figure 3 show that our attack can still fool the interpretation by generating a more uniform pattern for the heatmap. We perform 1,000 iterations with $\eta = 0.007$ and $\lambda = 0.75$.

Method	Target Acc (%)	Energy Ratio (%)	
		Top-Left	Top-Right
Adv. Patch [5]	100	76.96	1.65
Our Patch (Top-Left)	83.5	**14.99**	**7.57**

Table 3: Comparison of heatmap energy for the uniform patches. We report the energy at both the top-left and top-right corners of the heatmap.

4.6. Targeted regular adversarial examples

For the regular adversarial examples (non-patch) described in Section 3.3 that fool both the network prediction and interpretation, we perform 150 iterations with $\epsilon = 8/255$, $\eta = 0.001$, and $\lambda = 0.05$. Since the attack is not constrained to a patch location, the Energy Ratio metric is no longer applicable in this case. We use Localization Error and Histogram Intersection as the evaluation metrics in Table 4. We compare with PGD attack [23] as a baseline. The corresponding qualitative results are included in the supplementary material. Note that in this case, we run Grad-CAM for the original predicted category.

Image	Loc. Error(%)	Histogram
Original	66.68	1.0
PGD Adv.	67.74	0.77
Grad-CAM Adv.	**76.02**	**0.64**

Table 4: Evaluation results for adversarial examples generated using our method and PGD [23] on 10% randomly sampled ImageNet validation images. Note that for histogram intersection, lower is better while for localization error, higher is better.

4.7. Targeted patch on guided attention models

To challenge our attack algorithms, we use the GAIN$_{ext}$ model [22] which is based on VGG19 and is supervised using semantic segmentation annotation to produce better Grad-CAM results. The model is pre-trained on the training set of PASCAL VOC-2012, and we use the test set for optimizing the attack. Since each image in the VOC dataset can contain more than one category, we use the least likely predicted category as the target category. We perform 750 iterations with $\eta = 0.1$ and $\lambda = 10^{-5}$. The qualitative results are shown in Figure 4 and the quantitative ones in Table 5. Interestingly, our attack can fool this model even though it is trained to provide better Grad-CAM results.

Method	Target Acc (%)	Energy Ratio (%)
Adv. Patch [5]	94.34	37.90
Our Patch	94.70	**3.2**

Table 5: Targeted adversarial patch attack on GAIN$_{ext}$ model [22]

4.8. Generalization beyond Grad-CAM

We show that our patches learned using Grad-CAM are also hidden in the visualizations generated by Occluding Patch [34] method, which is a different interpretation algorithm. In occluding patch method, we visualize the change in the final score of the model by sliding a small black box on the image. Larger decrease in the score indicates that the regions are more important and hence they contribute more to the heatmap. The results of fooling GAIN$_{ext}$ model are shown in Table 6 and Figure 5.

Method	Targeted Attack Energy Ratio (%)
Adversarial Patch [5]	80.44
Our Patch	**31.59**

Table 6: Results showing transfer of our patch trained for Grad-CAM and evaluated on Occluding Patch [34] visualization using the GAIN$_{ext}$ model for VOC dataset.

4.9. Universal targeted patches

To show that the patch can generalize across images, we learn a universal patch for a given category using the training data and evaluate it on the test data. We use GAIN$_{ext}$ model along with $\eta = 0.05$ and $\lambda = 0.09$. The results are shown in Figure 6 and Table 7. We learn 20 different patches for each class of PASCAL VOC-2012 as the target category. We observe high fooling rates for both our method and regular adversarial patch, but our method has considerably low energy focused inside the patch area.

5. Conclusion

We introduce adversarial patches (small area, ~8%, with unrestricted perturbations) which fool both the classifier and the interpretation of the resulting category. Since we know that the patch is the true cause of the wrong prediction, a reliable interpretation algorithm should definitely highlight the patch region. We successfully design an adversarial patch that does not get highlighted in the

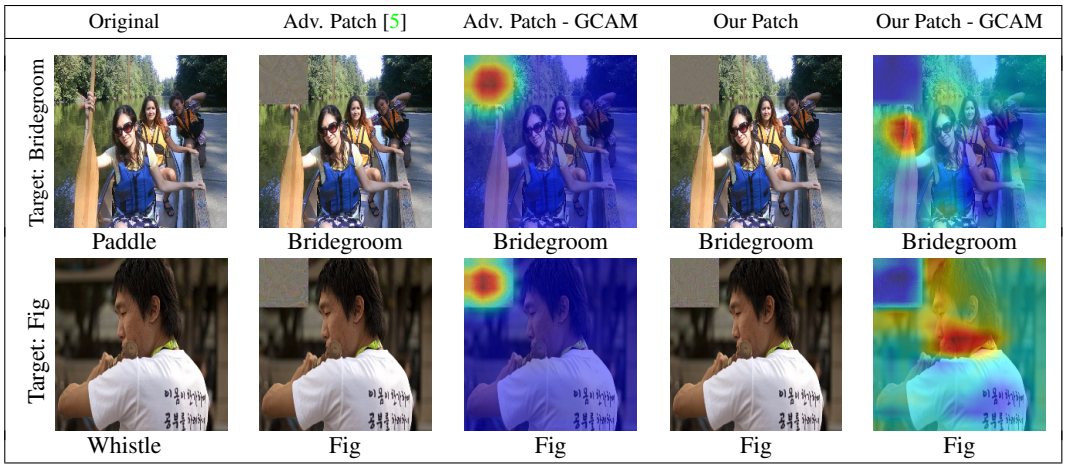

Figure 2: **Targeted patch attack:** We use an ImageNet pretrained VGG 19 BN network to compare the Grad-CAM visualization results for a random target category using our method vs Adv. Patch [5]. The predicted label is written under each image. Note that the patch is not highlighted in the last column.

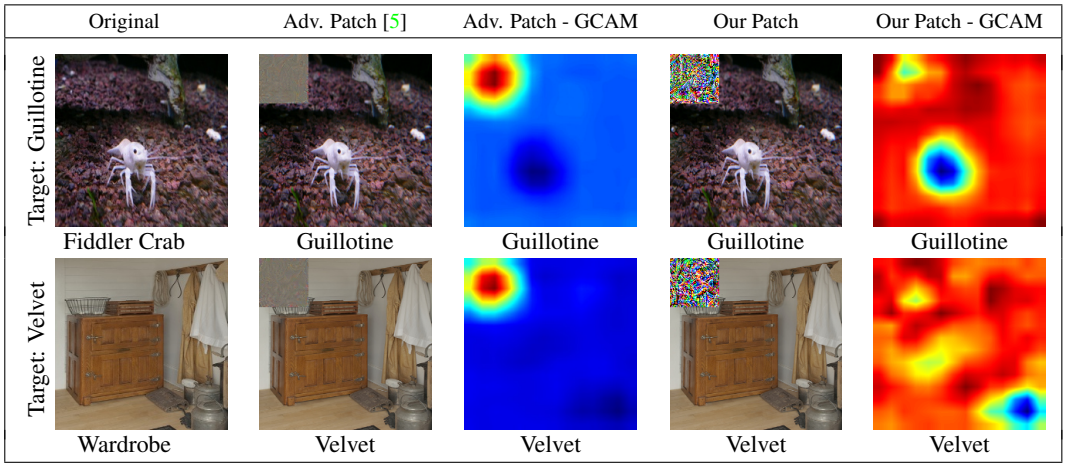

Figure 3: **Uniform patch attack:** Here, we paste our adversarial patch on the top-left corner and encourage the Grad-CAM heatmap for the target category to highlight the top-right corner. This shows that our algorithm can also be modified to hide our patch in the Grad-CAM visualization. The predicted label is written under each image. Note that the patch is not identifiable in the last column.

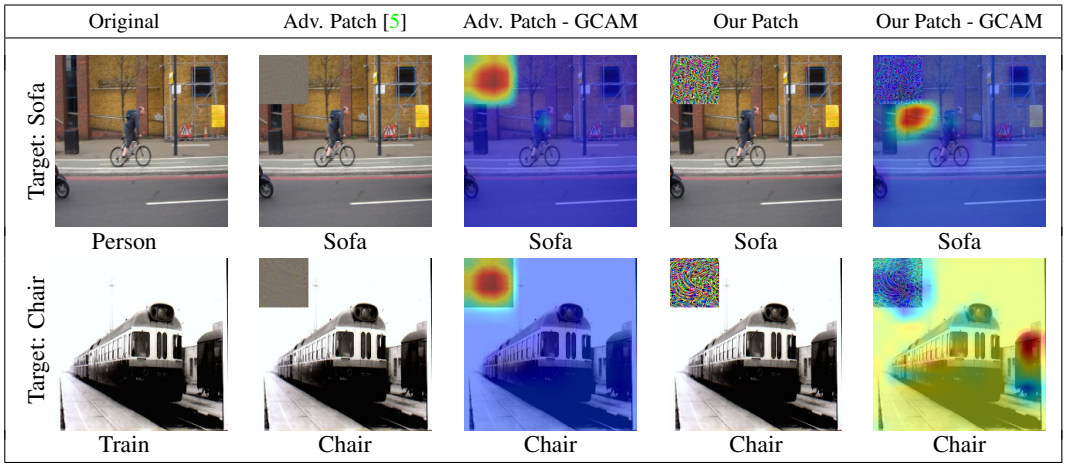

Figure 4: **Targeted attack for guided attention models:** We use GAIN$_{ext}$ [22] VGG19 model on VOC dataset to compare Grad-CAM visualization results for the least likely target category using our method vs Adv. Patch [5]. The predicted label is written under each image. GAIN$_{ext}$ is particularly designed to produce better Grad-CAM visualizations using direct supervision on the Grad-CAM output.

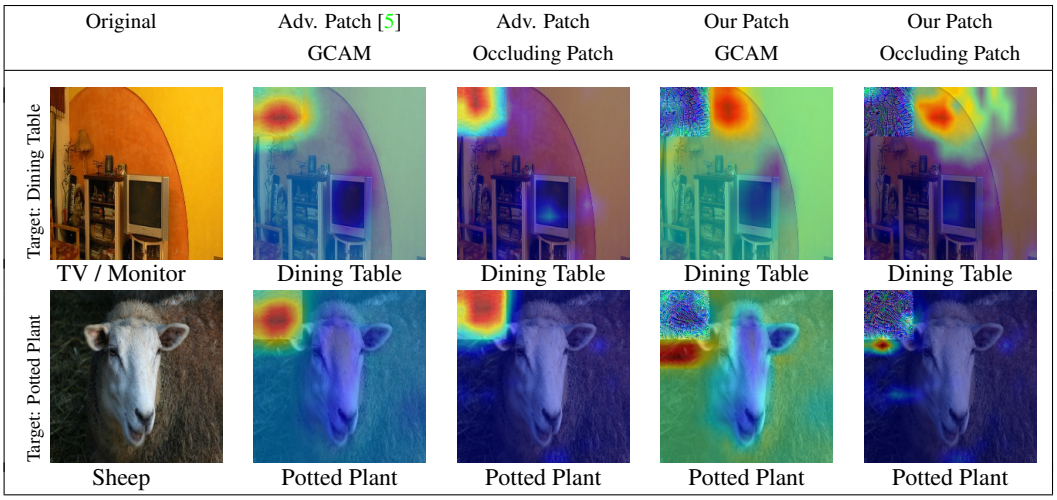

Figure 5: **Generalization beyond Grad-CAM:** Transfer of Grad-CAM visualization attack to Occluding Patch visualization. Here, we use targeted patch attacks (least likely target category) using our method vs Adv. Patch [5] on the GAIN$_{ext}$ [22] network for VOC dataset. The predicted label is written under each image. Grad-CAM and Occluding Patch visualizations are always computed for the target category. Note that the patch is hidden in both visualizations in columns 4 and 5.

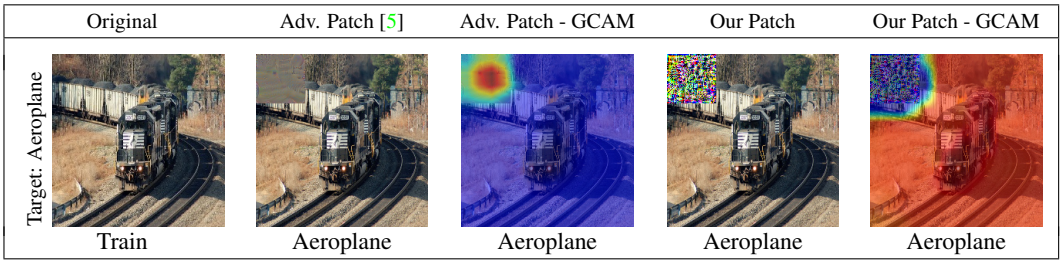

Figure 6: **Universal targeted patch:** Grad-CAM visualization results comparing our method vs Adv. Patch [5]. The top-1 predicted label is written under each image and Grad-CAM is always computed for the target category. The target category chosen was "Aeroplane". Additional results are included in the supplementary material.

Methods		aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Energy	Reg. Patch	86.3	64.8	90.3	47.7	92.8	**0.0**	78.3	48.4	84.5	94.4	78.2	90.9	16.9	85.0	78.61	1.1	86.9	85.4	83.0	98.2
	Our Patch	**0.0**	**0.8**	**0.6**	**0.3**	**0.0**	0.2	**1.4**	**0.6**	**0.6**	**2.4**	**3.7**	**0.0**	**0.0**	**1.2**	**0.8**	**0.0**	**2.7**	**2.6**	**0.1**	**0.0**
Target Acc	Reg. Patch	99.4	97.0	100	98.8	100	92.6	93.2	99.8	99.3	100	99.0	100	99.9	99.2	99.8	98.7	99.8	99.6	99.5	100
	Our Patch	94.5	97.4	99.3	84.5	94.3	99.9	99.7	99.6	98.7	34.3	87.3	94.7	98.3	99.4	99.8	99.2	99.8	93.8	99.2	99.0

Table 7: Results for the universal targeted patch attack using the GAIN$_{ext}$ [22] model on PASCAL VOC-2012 dataset using regular adversarial patch [5] and our adversarial patch. We learn universal patches for each of the 20 classes as the target category.

interpretation and hence show that popular interpretation algorithms are not highlighting the true cause of the prediction. Moreover, we show that our attack works in various settings: (1) generalizes from Grad-CAM to Occluded Patch [34], another interpretation method, (2) generalizes to unseen images (universal), (3) is able to fool GAIN [22], a model specifically trained with supervision on interpretation and (4) is able to make the interpretation uniform to hide the signature of the attack. Our work suggests that the community needs to develop more robust interpretation algorithms.

Acknowledgement: This work was performed under the following financial assistance award: 60NANB18D279 from U.S. Department of Commerce, National Institute of Standards and Technology, funding from SAP SE, and also NSF grant 1845216.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *NeurIPS*, 2018. 1, 3

[2] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 3

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*, 2018. 2

[4] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010. 1

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. In *Machine learning and Computer Security Workshop - NeurIPS*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[6] Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *AISec@CCS*, 2017. 3

[7] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Robust physical adversarial attack on faster r-cnn object detector. *arXiv preprint arXiv:1804.05810*, 2018. 2

[8] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018. 1, 3

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, pages 248–255. IEEE, 2009. 5

[10] Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Stuart Schieber, James Waldo, David Weinberger, and Alexandra Wood. Accountability of ai under the law: The role of explanation. *arXiv preprint arXiv:1711.01134*, 2017. 3

[11] Samuel G. Finlayson, John D. Bowers, Joichi Ito, Jonathan L. Zittrain, Andrew L. Beam, and Isaac S. Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019. 2

[12] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *arXiv preprint arXiv:1710.10547*, 2017. 3

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2014. 2

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 4

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[16] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. *arXiv preprint arXiv:1902.02041*, 2019. 3

[17] Forrest N. Iandola, Matthew W. Moskewicz, Sergey Karayev, Ross B. Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *CoRR*, abs/1404.1869, 2014. 5

[18] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. *arXiv preprint arXiv:1801.02608*, 2018. 3

[19] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. *arXiv preprint arXiv:1711.00867*, 2017. 3

[20] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 4

[22] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929. IEEE, 2018. 2, 3, 5, 6, 7, 8

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 2, 4, 6

[24] Arsalan Mosenia and Niraj K Jha. A comprehensive study of security of internet-of-things. *IEEE Transactions on Emerging Topics in Computing*, 5(4):586–602, 2017. 1

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[26] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 3, 5

[27] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM, 2016. 2

[28] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1, 3

[29] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*, 2018. 1

[30] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, Baltimore, MD, 2018. USENIX Association. 3

[31] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019. 2

[32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICLR*, abs/1312.6199, 2013. 2, 4

[33] Michał Zając, Konrad Żołna, Negar Rostamzadeh, and Pedro O Pinheiro. Adversarial framing for image and video classification. *arXiv preprint arXiv:1812.04599*, 2018. 2

[34] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *ICLR*, abs/1412.6856, 2015. 1, 2, 3, 6, 8

[35] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2921–2929. IEEE, 2016. 1, 3