Towards Hiding Adversarial Examples from Network Interpretation

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

Abstract

Deep 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 can be 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 prediction. We show that our algorithms can empower adversarial patches, by hiding them from network interpretation tools. 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 networks is a challenging task simply 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 can explain the underlying decision making process. Designing such algorithms is a challenging task and considerable research [1, 2, 3] has been done to describe *local explanations* - explaining the model's output for a given input [4]. We will be focusing on such methods in our work. Most of the these algorithms rely on studying the gradient of the output of a machine learning model with respect to its input.

(2) Adversarial examples: Many works have shown that deep networks are vulnerable to adversarial examples, which are carefully constructed samples created by adding imperceptible perturbations to the original input to change 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) [5] or self-driving cars [6]. (b) Studying these attacks can lead to better understanding of how deep networks work and possibly improve generalization on new environments.

We specifically focus on adversarial patches rather than regular adversarial examples since patches are a more practical form of attack, and also the cause of the misclassification is strictly limited to the patch area. Hence, it is not trivial for the attacker to mislead the interpretation to highlight non-patch regions without perturbing them.

Consider an example of adversarial attack using adversarial patches as seen in [7], we show that an interpretation algorithm like Grad-CAM [3] usually highlights the location of such an adversarial patch making it clear that the image patch was responsible for misclassification. We are interested in designing stronger attack algorithms that not only change the prediction but also mislead the interpretation of the model to hide the attack from investigation. As an example, assume an adversary (one in a crowd of pedestrians) is wearing a t-shirt with a printed adversarial patch on the back that fools a self-driving car leading to an accident. Now, a simple investigation with standard network interpretation methods may reveal which pedestrian in the scene was the cause of the wrong decision, and thereby identifying the adversary. However, we show that it is possible for the adversary to learn a patch without revealing their identity (patch location) and thus escape scrutiny.

We do this by encouraging the optimization to change the interpretation of the network when constructing the corresponding adversarial example. Our work highlights that using a well-designed attack algorithm, an adversary can construct sophisticated adversarial examples which not only change the prediction but also remove any trace of corruption when inspected by network interpretation algorithms.

2. Related work

Adversarial examples: Adversarial examples were dis-

*Equal contribution

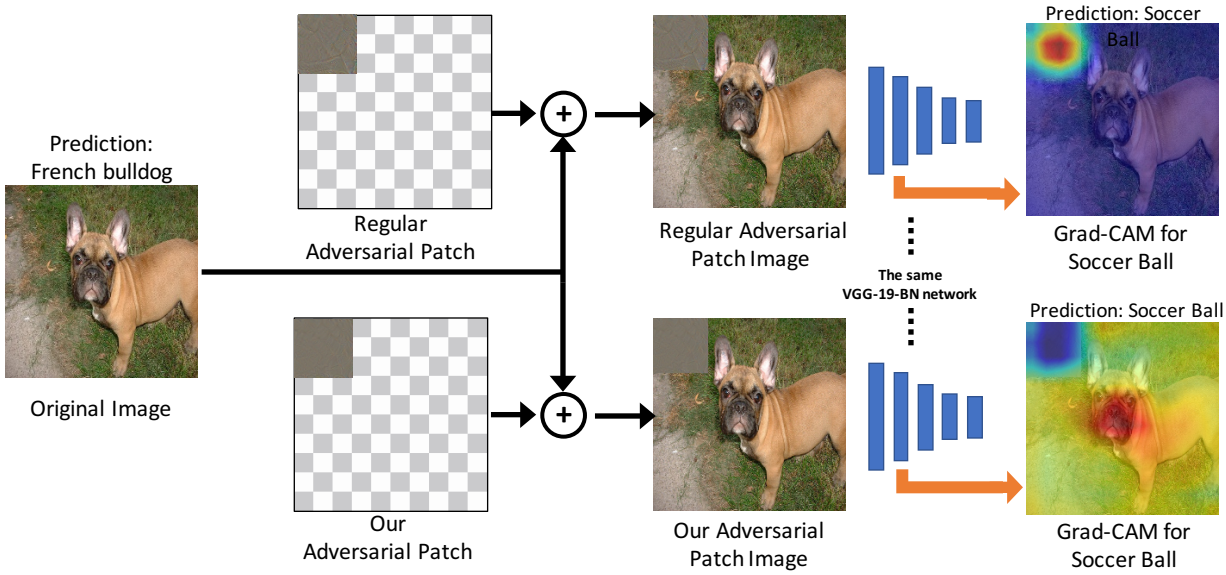

Figure 1. We show that Grad-CAM highlights the patch location in the image perturbed by regular targeted adversarial patches [7] (top row). Our modified attack algorithm goes beyond fooling the final prediction by hiding the patch in the Grad-CAM visualization, making it difficult to investigate the cause of the mistake. Note that Grad-CAM visualizes the cause of target category.

covered by Szegedy et al. [8] who showed that state-of-the-art machine learning classifiers can be fooled comprehensively by simple backpropagation algorithms. Goodfellow et al. [9] 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 [10, 11] and recently, [12] showed that adversarial examples could be robust to affine transformations. Madry et al. [13] proposed Projected Gradient Descent (PGD) which has been shown to be the best first-order adversary for fooling classifiers. Kindermans et al. [14] showed how saliency methods are unreliable by adding constant shift to input data and checking against different saliency methods. In our work, we show that it is not only possible to fool the classifier using an adversary, but also hide it from standard network interpretation algorithms.

Adversarial patches: Adversarial Patches [7, 15] were introduced recently 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. We improve this by hiding the patch location from network interpretation tools.

Interpretation of deep networks: As neural networks are getting closer towards deployment in real world applications, it is important that their results are interpretable. Researchers have proposed various algorithms in this direction. One of the earliest attempt was done in [1] where they calculate the derivative of the network's outputs w.r.t the input to compute class specific saliency maps. Zhou et al. [16] calculates the change in the network output when a small portion of the image (say 11×11 pixels) is covered by a random occluder. We call this **Occluding Patch**. CAM [2] 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** [3], a gradient based method which provides visual explanations for any neural network architecture. Kunpeng *et al*. [17] recently improved upon Grad-CAM using Guided attention mechanism with state-of-the-art results on segmentation tasks. 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.

3. Method

We propose algorithms to learn adversarial patches that when pasted on the input image, can change the interpretation of the model's final decision. We will focus on Grad-CAM [3], which is one of the popular network interpretation methods in designing our algorithms and then, will 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 con-

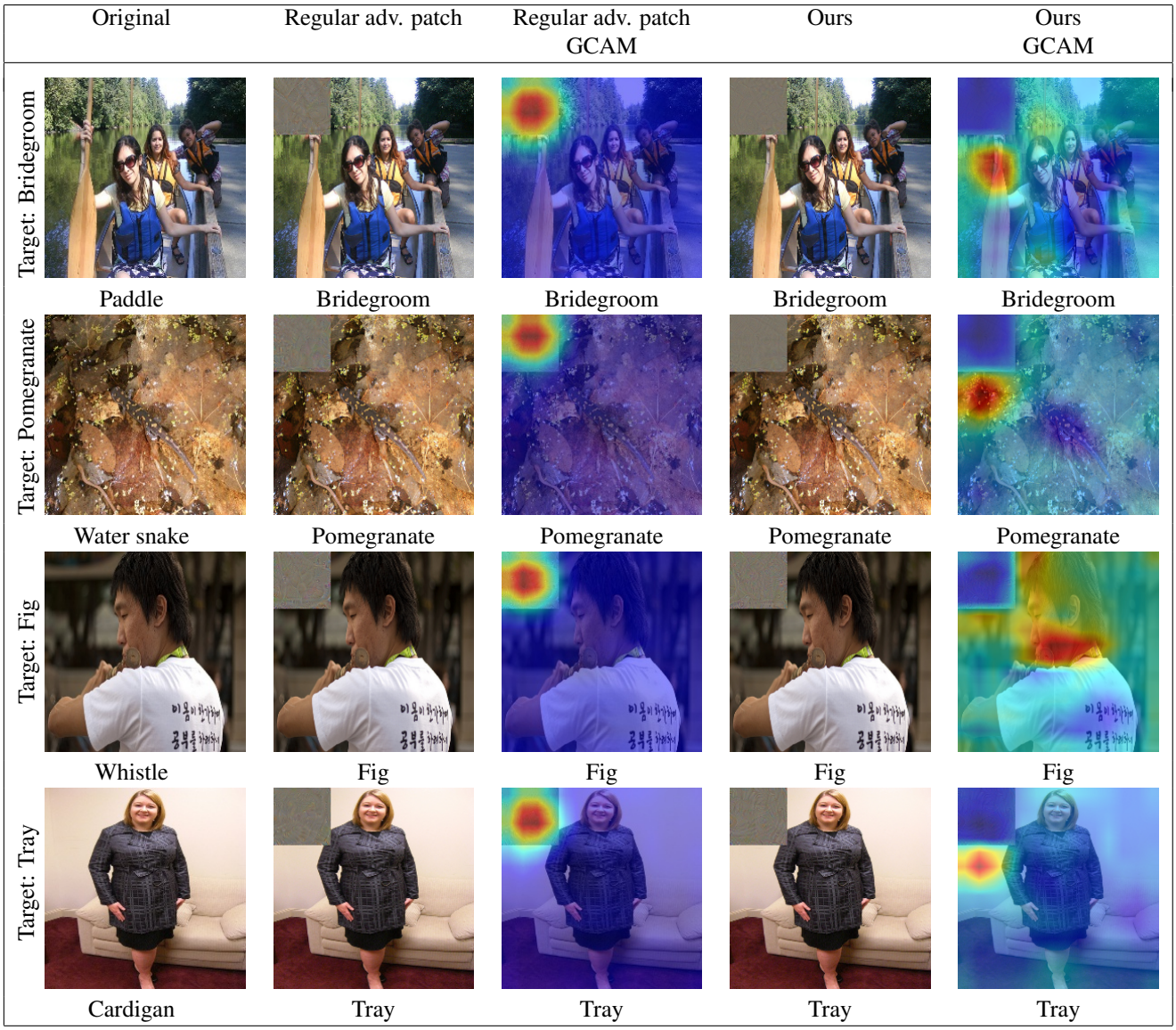

Figure 2. Comparison of Grad-CAM visualization results for targeted patch attacks using our method ('Ours') vs regular adversarial patch ('AP'). The predicted label is written under each image, the attack was successful for all images, and Grad-CAM is always computed for the target category. Note that the patch is not highlighted in the right column.

volutional 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 [3], we define:

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

where Z is a normalizer and then calculate the heatmap as follows :

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

Finally, we normalize it to get:

$$\hat{G}^c := \frac{G^c}{|G^c|_1}$$

.

Background on adversarial patches:

Consider an input image x_0 and a predefined constant binary mask m that is 1 on the location of the patch and 0 everywhere else. We want to find an adversarial patch z that changes the output of the network to catgegory t when pasted on the image, so we solve:

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

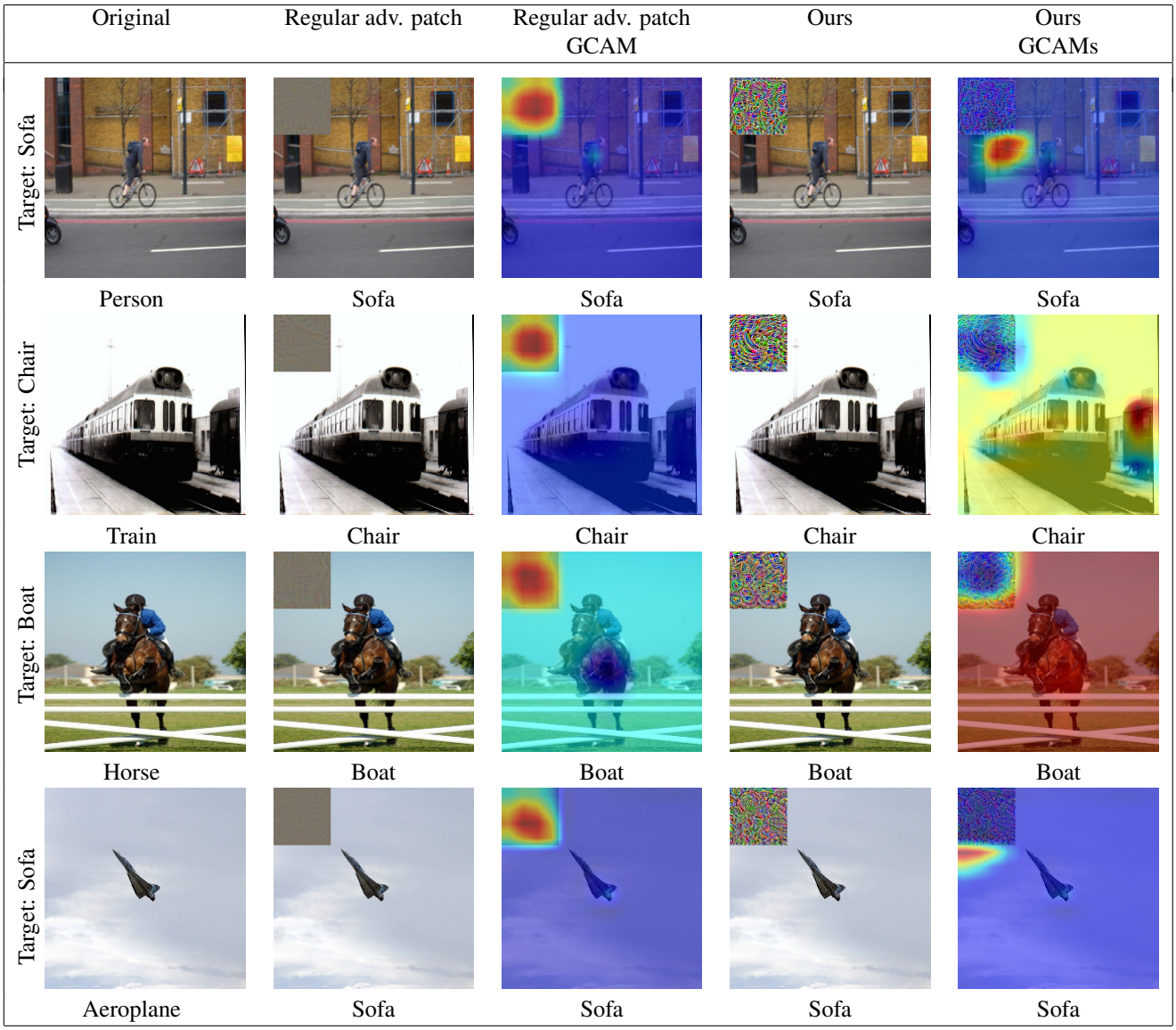

Figure 3. Comparison of Grad-CAM visualization results for targeted patch attacks for the least likely target category using our method ('Ours') vs regular adversarial patch ('Adv Patch') on the GAIN$_{ext}$ [17] network for VOC dataset. The predicted label is written under each image, the attack was successful for all images, and Grad-CAM is always computed for the target category. GAIN$_{ext}$ is particularly designed to produce better Grad-CAM visualizations using direct supervision of the Grad-CAM output.

where $\ell_{ce}(.;t)$ is the cross entropy loss for the target category t and $\odot$ is the element-wise product. This results in adversarial patches similar to [7].

3.1. Misleading interpretation in targeted mode:

As shown in Figure 2, when Grad-CAM of the target category is used to investigate the cause of misclassification , it highlights the patch very strongly revealing the cause of the attack. This is expected as the adversary is restricted to perturbing the patch area and the patch is the cause of the final misclassifcation towards target category. To fool the network's interpretation so that the adversarial patch is not highlighted at the interpretation of the final prediction, we add an additional term to our loss function in learning the patch to suppress the total activation of the visualization at the patch location m. 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 tar-

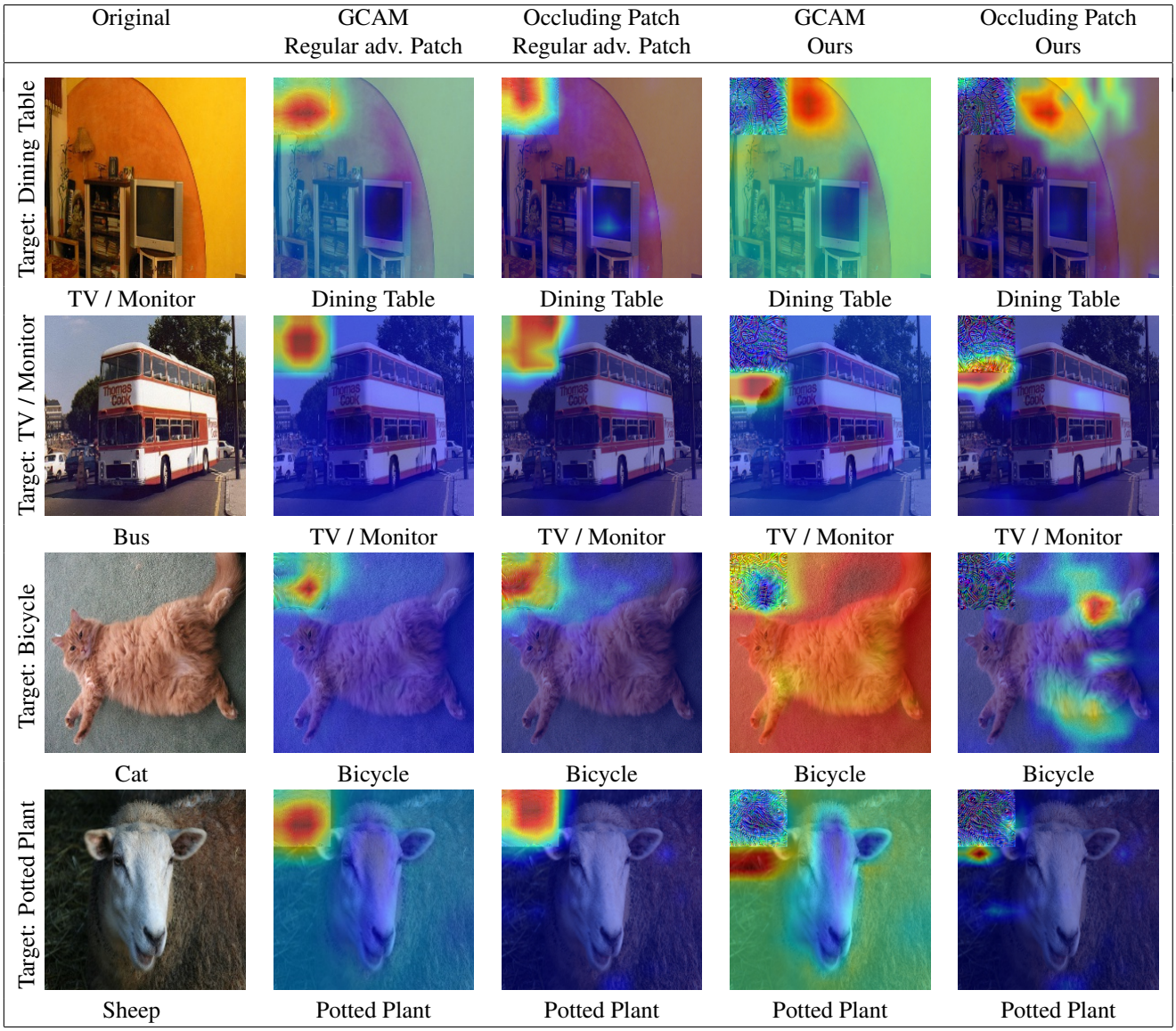

Figure 4. Transfer of Grad-CAM visualization attack to Occluding Patch visualization. Here, we use targeted patch attacks for the least likely target category using our method ('Ours') vs regular adversarial patch ('Adv Patch') on the GAIN$_{ext}$ [17] network for VOC dataset. The predicted label is written under each image, the attack was successful for all images, and 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.

get label randomly across all classes excluding the original prediction similar to "step rnd" method in [18].

3.2. Misleading interpretation in non-targeted mode

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 we have no control over the final category which is predicted after adding the patch. In this case, our optimization problem becomes:

$$\arg\min_{z} \Big[\max(0, M - \ell_{ce}(\tilde{x}; c)) + \lambda \sum_{ij} \big(\hat{G}^{a}(\tilde{x}) \odot m\big) \Big]$$
$$\text{where} \quad a = \arg\max_{k} p(k).$$

where c is the predicted category for the original image, $p(k)$ is the probability of category k, and a is the top prediction at every iteration. Since cross entropy loss is not upper-bounded, it can dominate the optimization, so we use contrastive loss [19] to ignore cross entropy when the prob-

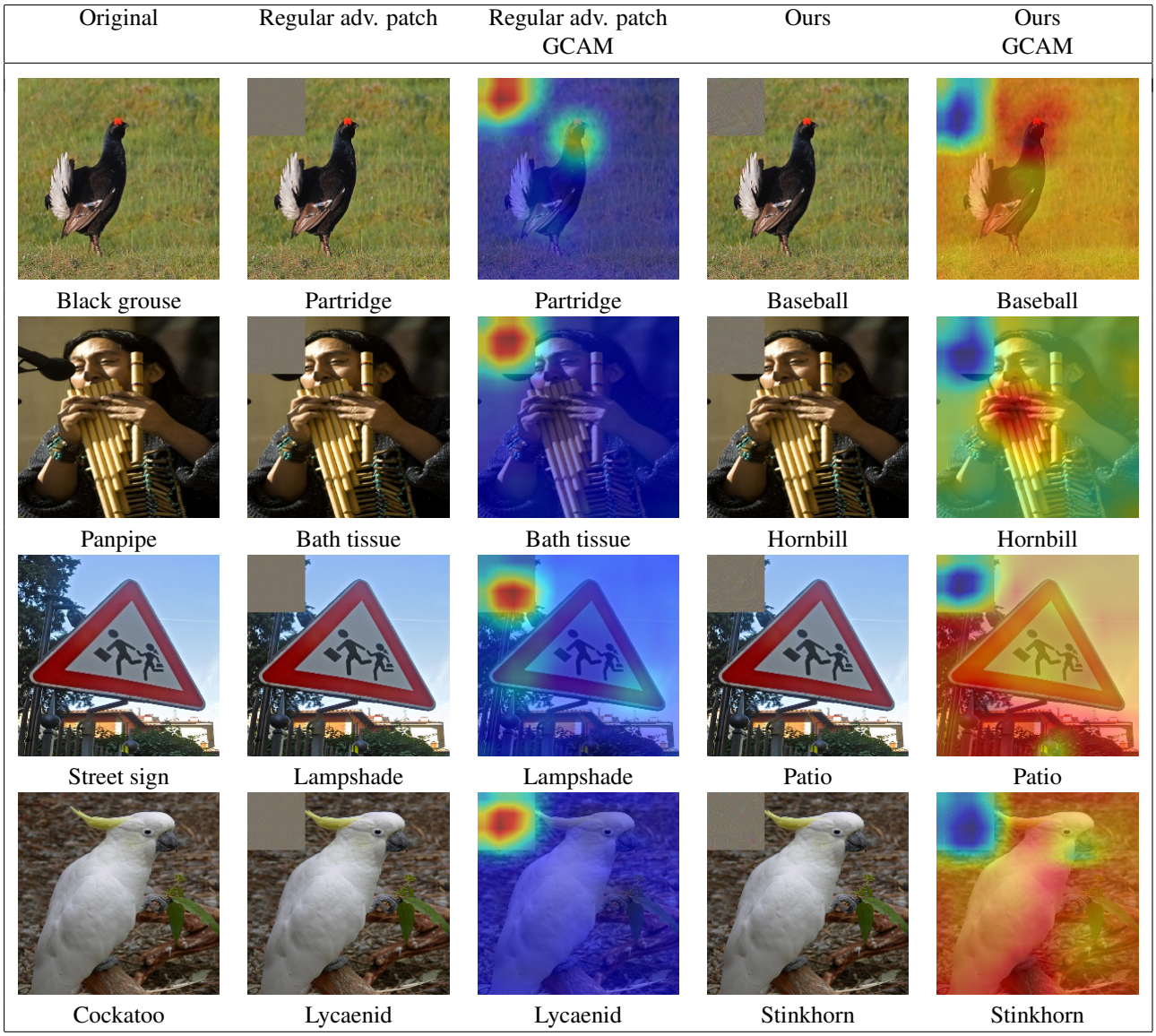

Figure 5. Comparison of Grad-CAM visualization results for non-targeted patch attacks using our method ('Ours') vs regular adversarial patch ('AP'). The predicted label is written under each image, the non-targeted attack was successful for all images, and Grad-CAM is always computed for the predicted category.

ability 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 visualization of the current top category a.

To optimize above loss functions, we use an iterative approach similar to projected gradient decent (PGD) algorithm [13]. We initialize z^0 randomly and then 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 the image values.

3.3. Misleading interpretation for guided attention models

Recently, there have been works [20], [21] which focus on improving the attention maps of the predicted objects when training a classifier. Kunpeng *et al*. [17] further improve upon this by providing supervision on the network's attention in an end-to-end way. This is done by designing loss functions that guide the network's focus on the entire area critical to the task of interest. We perform the targeted attack as described in section 3.1 on the GAIN$_{ext}$ model from [17]. Since the GAIN$_{ext}$ model was fine-tuned

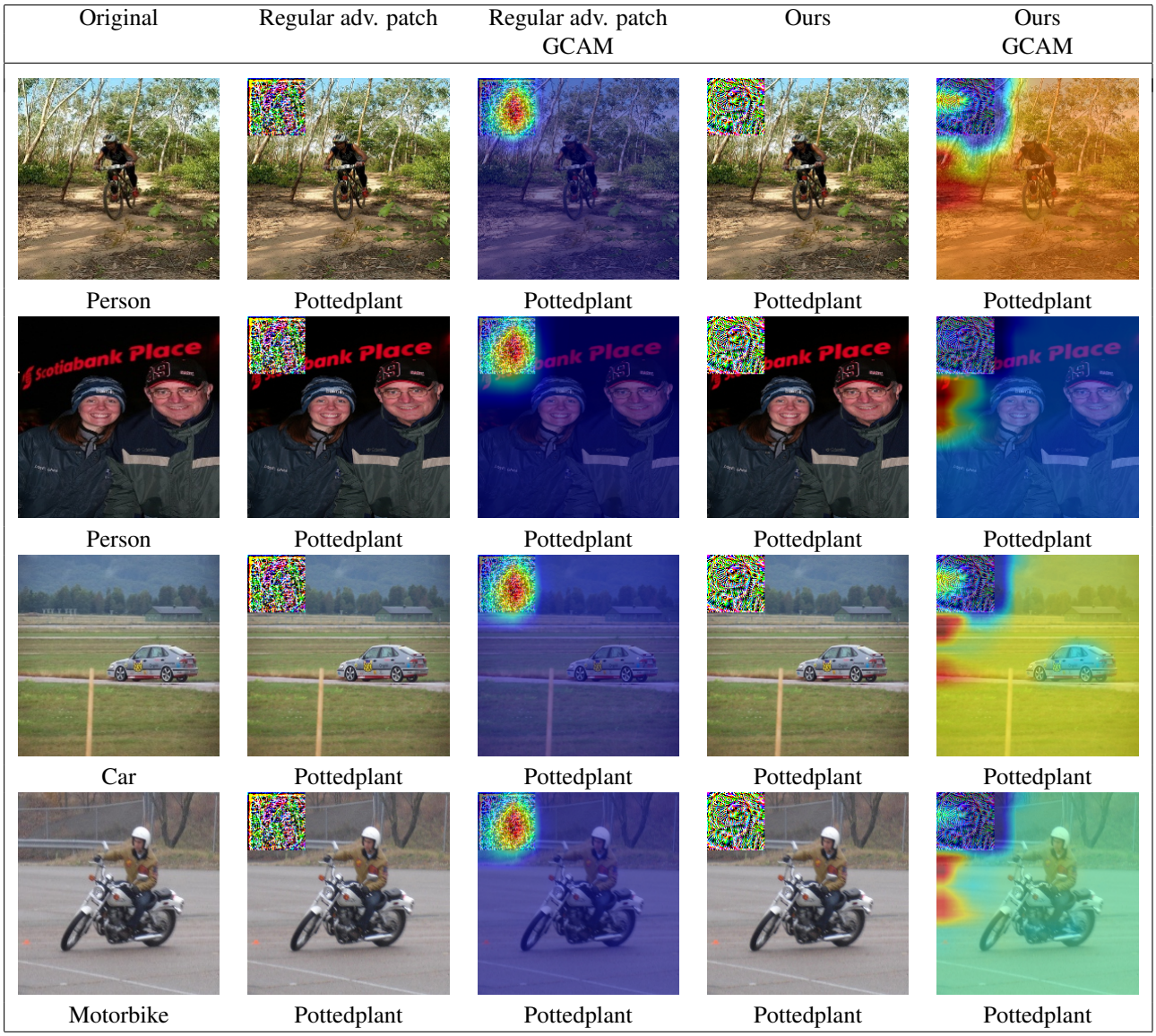

Figure 6. Comparison of Grad-CAM visualization results for universal targeted patch attacks using our method ('Ours') vs regular adversarial patch ('AP'). The top-1 predicted label is written under each image, the universal attack was successful for all VOC images, and Grad-CAM is always computed for the predicted category. The target category chosen was "Pottedplant".

Method	Top-1 Acc(%)	Non-Targeted		Targeted		
		Acc (%)	Energy Ratio (%)	Acc (%)	Target Acc (%)	Energy Ratio(%)
Adversarial Patch [7]	74.24	0.06	38.95	0.02	99.98	58.39
Our Patch	74.24	0.05	**2.00**	2.95	77.88	**5.21**

Table 1. Comparison of heatmap energy within the 8% patch area for the adversarial patch [7] and our patch. 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.

for the PASCAL VOC dataset, we perform a targeted attack to change the prediction to the least likely category from the predictions and also ensure that the Grad-CAM heatmap does not overlap with the patch. We also observe that patches created our method transfer to Occluding Patch [2] as seen in Figure 4. This shows thateven if the patch is

Method	Targeted	
	Target Acc (%)	Energy Ratio (%)
Adversarial Patch [7]	94.34	29.02
Our Patch	94.70	**2.45**

Table 2. Comparison of Grad-CAM heatmap energy within the 8% patch area for the adversarial patch [7] and our patch trained on the GAIN$_{ext}$ [17] for VOC dataset. The heatmap has far less energy in the patch area that the adversary can change.

Method	Targeted	
	Target Acc (%)	Energy Ratio (%)
Adversarial Patch [7]	99.97	61.91
Our Patch	92.28	**0.56**

Table 3. Comparison of heatmap energy for the universal attack case.

Method	Targeted Attack Energy Ratio (%)
Adversarial Patch [7]	61.59
Our Patch	**24.19**

Table 4. Comparison of Occluding Patch [16] heatmap energy within the 8% patch area for the adversarial patch [7] and our patch trained on the GAIN$_{ext}$ [17] for VOC dataset. Note that we still use Grad-CAM in training and evaluate on Occluding Patch. This shows our attack generalizes from Grad-CAM to occluding patch.

optimized to fool only one visualization, it also results in fooling other visualizations as well, which is to the advantage of the adversary.

3.4. Universal 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 learnt using the training data. Similar to 3.3, we consider the GAIN$_{ext}$ model which is fine-tuned on PASCAL VOC dataset. We fix the target category as "Pottedplant" and learn the patch as a form of targeted attack as explained in 3.1 which ensures mis-classification towards the target category along with the heatmap on the patch area being minimal. This is the most practical form of attack, since the adversary needs to train the patch just once, which would be strong enough to fool multiple test scenarios.

4. Experiments

We use pre-trained VGG-19 [22] with batch normalization implemented in PyTorch.

4.1. Misleading interpretation of adversarial patches: For the adversarial patch experiments described in the method section, we use a patch of size 64x64 on the top-left corner for 50,000 images of size 224x224 ($\sim$ 8% of the image area) in the validation set of ImageNet [23] ILSVRC2012. We do 750 iterations with $\eta = 0.001$. To evaluate how much the patch is highlighted in the visualization, we construct the visualization heatmap $\hat{G}^c$ for the mistaken category, and calculate the ratio of the total energy 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. The quantitative results are shown in Table 1.

4.2. Misleading interpretation for guided attention models: We use the GAIN$_{ext}$ model fine-tuned on the VOC dataset from [17] and VOC test set for these experiments. We use a patch of size 64x64 on the top-left corner for 4952 images of size 224x224 ($\sim$ 8% of the image area) in the test set of the PASCAL VOC dataset. We do 750 iterations with $\eta = 0.1$. Since each image in the VOC dataset can contain more than one category, we use the least likely predicted category to perform the targeted patch attack. We evaluate using the same method as described in Section 4.1. The results of the evaluation are described in Table 2.

4.3. Generalization beyond Grad-CAM: We also show that our patches learned using Grad-CAM are hidden in the visualizations generated by Occluding Patch [16] method. 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 means more important regions and hence they contribute more to the heatmap. The results are shown in Table 4 and Figure 4.

4.4. Universal Patches: For these experiments, we learn a patch of size 64x64 on the top-left but use the training set from PASCAL VOC dataset. The optimization performed is similar to the above section. We use 50 iterations per image with $\eta = 0.05$ and $\lambda = 0.09$. As described in 3.4, we choose "Pottedplant" as the target category and the evaluation was done on test set. The results for these can be found in Figure 6 and Table 3 . We observe high fooling rates for both the methods, but our method has considerably low energy focused inside the patch area. Note that only the region of the patch is perturbed and everything else is untouched.

5. Conclusion

We presented novel adversarial attack algorithms that go beyond fooling the prediction by hiding the cause of the mistake from our common interpretation tools to result in a stronger attack. We show that our attack tuned for Grad-CAM can transfer to other visualization algorithms and we also show that we can create universal patches that can generalize fooling across images. Our work shows that there is a need for more robust deep learning tools that reveal the correct cause of network's predictions.

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

References

[1] 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, 2

[2] 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, 2, 7

[3] 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. *See https://arxiv.org/abs/1610.02391 v3*, 7(8), 2016. 1, 2, 3

[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] 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

[6] 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

[7] TB Brown, D Mané, A Roy, M Abadi, and J Gilmer. Adversarial patch. arxiv e-prints (dec. 2017). *arXiv preprint cs.CV/1712.09665*, 2017. 1, 2, 4, 7, 8

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

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. 2

[10] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 2

[11] 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

[12] Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017. 2

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2, 6

[14] 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. 2

[15] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. *arXiv preprint arXiv:1801.02608*, 2018. 2

[16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 2, 8

[17] 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, 4, 5, 6, 8

[18] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. 5

[19] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *null*, pages 1735–1742. IEEE, 2006. 5

[20] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *Proceedings of the European Conference on Computer Vision*. IEEE, 2017. 6

[21] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE CVPR*, volume 1, page 3, 2017. 6

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 8

[23] 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. 8