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) Lack of Explainability: The goal of Explainable AI is to develop reliable interpretation algorithms that explain the underlying decision making process of deep networks. Such algorithms are challenging to design and considerable work [19, 23, 18] has been done to describe *local explanations* - explaining the model's output for a given input [2].

(2) Adversarial examples: Deep neural networks have been shown to be vulnerable to adversarial examples [21, 6, 15, 12], which could be used to fool AI algorithms when deployed in real-world applications [16, 20]. These have also been extended to interpretation algorithms [9, 11, 1].

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.

*Equal contribution

Reliability of network interpretation: To study the reliability of the interpretation in highlighting true cause of the prediction, we use the adversarial patch method [3] 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. In this paper, we choose to study the correctness of Grad-CAM [18], a well-known interpretation algorithm which performs well on sanity check [1] and show that our method transfers to other interpretation algorithms as well.

We show an illustration of our work in Figure 1. We learn 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). The observation that Grad-CAM does not highlight the patch pixels for our adversarial patch reveals that Grad-CAM is not reliably highlighting the source of prediction. Note that in this setting, the target category is randomly chosen by the adversary from all possible wrong categories. We believe this shows that the Grad-CAM algorithm is not *necessarily* showing the true cause of the prediction.

Ghorbani *et al*. [5] introduced adversarial perturbations that result in 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 [5]. Our work mitigates this concern by designing a *controlled* setting using adversarial patches where the adversary is restricted to a small region of the image. We believe our algorithms can be used as a form of evaluation for future interpretation algorithms.

Our key contributions are summarized as follows:

(1) We introduce a novel algorithm to construct adversarial patches which fool both the classifier and the inter-

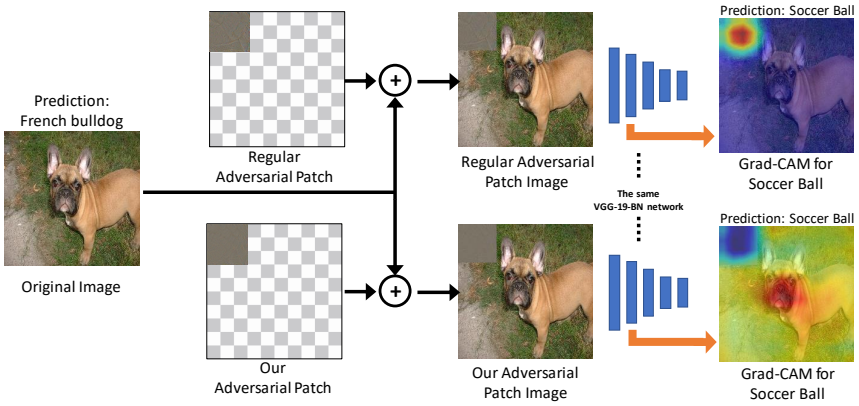

Figure 1: Our modified attack algorithm goes beyond fooling the final prediction by also fooling the Grad-CAM visualization. The original image (left) is correctly classified as "French Bulldog". On the top row, a targeted adversarial patch has successfully changed the prediction to "Soccer Ball". In 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. Here, Grad-CAM visualization is done for "Soccer Ball" category.

pretation of the resulting category.

(2) With extensive experiments, we show that our method (a) generalizes from Grad-CAM to Occluding Patch [22], another interpretation method, (b) generalizes to unseen images (universal), (c) is able to fool GAIN [14], a model specifically trained with interpretation supervision.

(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. Method

In our work we focus on Grad-CAM [18] for designing our algorithms and then, show that our results generalize to other interpretation algorithms as well.

Background on Grad-CAM visualization: For a given model (e.g. VGG) and a category c, Grad-CAM is used to highlight the image regions responsible for the model's classification decision as category c. This is done by considering the output of a convolutional layer, e.g. *conv5* and computing the derivative of the output y^c w.r.t. these activations. We then take the mean over the spatial locations to obtain the gradient-weighted importance of each filter of the convolutional layer:

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

where A_{ij}^k is the activation at filter k for spatial location (i, j) and Z is a normalizer. Then we calculate the interpretation (heatmap) as the weighted sum of activations of the convolutional layer discarding the negative values:

$$G_{ij}^c = max(0, \sum_k \alpha_k^c A_{ij}^k)$$

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 [3].

2.1. 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 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:

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

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 [13].

To optimize the above loss function, we use an iterative approach similar to projected gradient decent (PGD) algorithm [15]. 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 restricted to the patch location.

2.2. 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 is a weaker attack since the adversary has no control over the category predicted after adding the patch. In this case, our optimization problem becomes:

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

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 [7] 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. Experiments

We perform our experiments in two different benchmarks. We use VGG19 network with batch normalization and the ImageNet [4] dataset for these experiments.

Then to evaluate our attack in a more challenging setting, we use GAIN_{ext} model from [14] 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.

3.1. Evaluation

We use standard classification accuracy to report the success rate of the attack and we define a novel metric to measure the success of fooling interpretation.

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. It will be 0 if the patch is not highlighted at all and 1 if the heatmap is completely concentrated inside the patch.

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 [17] along with NVIDIA Titan-X GPUs for all experiments.

3.2. Targeted adversarial patches

For the adversarial patch experiments described in the method section, we use 50,000 images of the validation set of ImageNet [4]. 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. We also perform an ablation experiment to learn patches that result in *uniform* interpretation heatmap to mitigate the concern that the patch is clearly visible during a manual investigation, details for which can be found in the appendix.

3.3. Non-targeted adversarial patches

Here, we perform the non-targeted adversarial patch attack using 50,000 images of the validation set of ImageNet [4] ILSVRC2012. We perform 750 iterations with $\eta = 0.005$ and $\lambda = 0.001$. The results are shown in Table 1 and Figure A3 in the appendix.

3.4. Targeted patch on guided attention models

To challenge our attack algorithms, we use the GAIN_{ext} model [14] which is based on VGG19 and is supervised using semantic segmentation to produce better Grad-CAM results. The model is pre-trained on the training set of 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. We perform 750 iterations with $\eta = 0.1$ and $\lambda = 10^{-5}$. The quantitative results are shown in Table 2 and qualitative results are shown in Figure A4 of the appendix. Interestingly, our attack can fool this model even though it is trained to provide better Grad-CAM results.

3.5. Generalization beyond Grad-CAM

We show that our patches learned using Grad-CAM are also hidden in the visualizations generated by Occluding Patch [22] 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

Method	Top-1 Acc(%)	Non-Targeted		Targeted		
		Acc (%)	Energy Ratio (%)	Acc (%)	Target Acc (%)	Energy Ratio(%)
Adversarial Patch [3]	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 [3] 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.

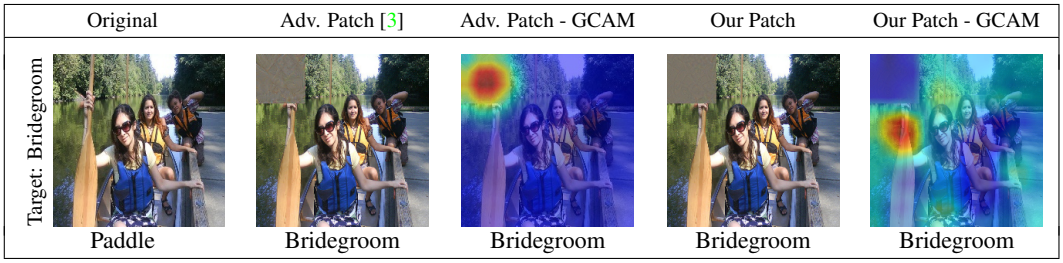

Figure 2: **Targeted patch attack:** We use an ImageNet pretrained VGG-19 BN network to compare the Grad-CAM visualization results for a random target category using our method vs Adv. Patch [3]. The predicted label is written under each image. Note that the patch is not highlighted in the last column. More results of this experiment can be found in Figure A2 of the appendix.

Method	Target Acc (%)	Energy Ratio (%)
Adv. Patch [3]	94.34	37.90
Our Patch	94.70	**3.2**

Table 2: Targeted adversarial patch attack on GAIN_{ext} model [14]

Method	Targeted Attack Energy Ratio (%)
Adversarial Patch [3]	80.44
Our Patch	**31.59**

Table 3: Results showing transfer of our patch trained for Grad-CAM and evaluated on Occluding Patch [22] visualization using the GAIN_{ext} model for VOC dataset.

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 above in Table 3 and Figure A5 of the appendix.

3.6. Universal targeted patches

Universal attack is a much stronger form of attack wherein the adversary needs to train a patch just once per target category, and is able to fool multiple unseen test images. To learn universal patches, we use Eq. 1 to sum over all training images for a given target category 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 Table 4 and qualitative results are in Figure A6 of appendix. We learn 20 different patches for each class of VOC dataset as the target. 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.

Method	Target Acc (%)	Energy Ratio (%)
Adv. Patch [3]	98.78	69.58
Our Patch	93.63	**0.9**

Table 4: Universal targeted patch attack on GAIN_{ext} model. Note that the results are averaged over 20 PASCAL VOC classes. Individual class results can be found in the Table T4 of appendix.

4. Conclusion

We introduced adversarial patches 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 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 [22], another interpretation method, (2) generalizes to unseen images (universal), and (3) is able to fool GAIN [14], a model specifically trained with supervision on interpretation. Our work suggests that the community needs to develop more robust interpretation algorithms, possibly using our method as an evaluation metric.

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

[2] 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

[3] 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, 4, 6, 7, 8, 10

[4] 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. 3

[5] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *arXiv preprint arXiv:1710.10547*, 2017. 1

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2014. 1

[7] 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. 3

[8] 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. 6

[9] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. *arXiv preprint arXiv:1902.02041*, 2019. 1

[10] 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. 6

[11] 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. 1

[12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 1

[13] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 3

[14] 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, 4, 7, 10

[15] 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. 1, 3, 6

[16] 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

[17] 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. 3

[18] 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, 2, 6

[19] 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

[20] 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

[21] 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. 1, 6

[22] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *ICLR*, abs/1412.6856, 2015. 2, 3, 4

[23] 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

5. Appendix

Targeted regular adversarial examples: We consider regular adversarial examples (non-patch) [21] 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 in the main paper, we expand mask m to cover the whole image and initialize z from x. Although this experiment does not necessarily show that the interpretation method is wrong, we report the results of such attacks for completeness. 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. We use the following two evaluation metrics:

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

(b) Localization: Similar to [18], we draw a bounding box around values larger than a threshold (0.15), and evaluate object localization from ImageNet competition. In Table T1, we compare with PGD attack [15] as a baseline. The corresponding qualitative results are shown in Fig A7. 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 T1: Evaluation results for adversarial examples generated using our method and PGD [15] on 10% randomly sampled ImageNet validation images. Note that for histogram intersection, lower is better while for localization error, higher is better.

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 top-left corner). Our results in Table T2 and Figure A1 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 $\lambda = 0.75$.

Different networks and patch locations: In this section, we evaluate our targeted adversarial patch attack algorithm on ResNet-34 [8] and DenseNet-121 [10] by placing the patch on the top-right corner of the image. Both models are pretrained on ImageNet dataset and we use 5,000 random images from the ImageNet validation set to evaluate these attacks using the *Energy Ratio* metric presented in Table T3. Our patch fools the interpretation while reaching the target category in more than 90% of the images. Figures A8 and A9 show the qualitative results for Resnet-34 and Densenet-121 networks respectively.

Method	Target Acc (%)	Energy Ratio (%)	
		Top-Left	Top-Right
Adv. Patch [3]	100	76.96	1.65
Our Patch (Top-Left)	83.5	**14.99**	**7.57**

Table T2: Comparison of heatmap energy for the uniform patches. We report the energy at both the top-left and top-right corners of the heatmap.

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 T3: Comparison of Grad-CAM heatmap energy within the top-right corner patch area for ResNet-34 (R-34) and DenseNet-121 (D-121) networks on 10% randomly sampled ImageNet validation images.

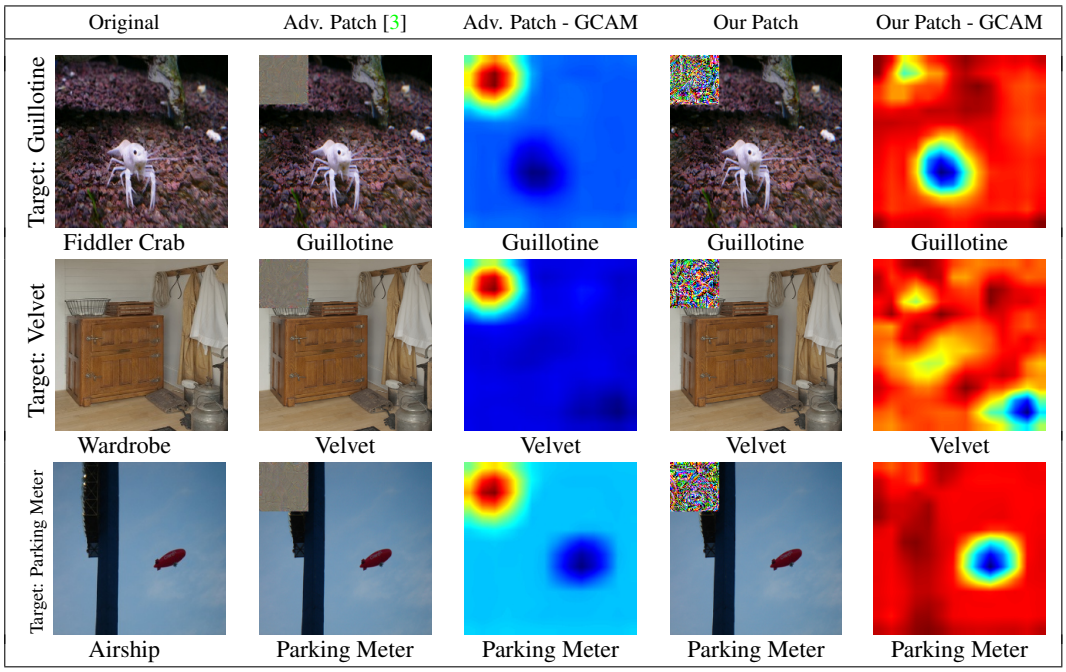

Figure A1: **Uniform patch attack:** Here, we paste our adversarial patch on the top-left corner and encourage the Grad-CAM heatmap for the target category to highlight the top-right corner. This shows that our algorithm can also be modified to hide our patch in the Grad-CAM visualization. The predicted label is written under each image. Note that the patch is not identifiable in the last column.

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 T4: Results for the universal targeted patch attack using the GAIN$_{ext}$ [14] model on PASCAL VOC-2012 dataset using regular adversarial patch [3] and our adversarial patch. We learn universal patches for each of the 20 classes as the target category.

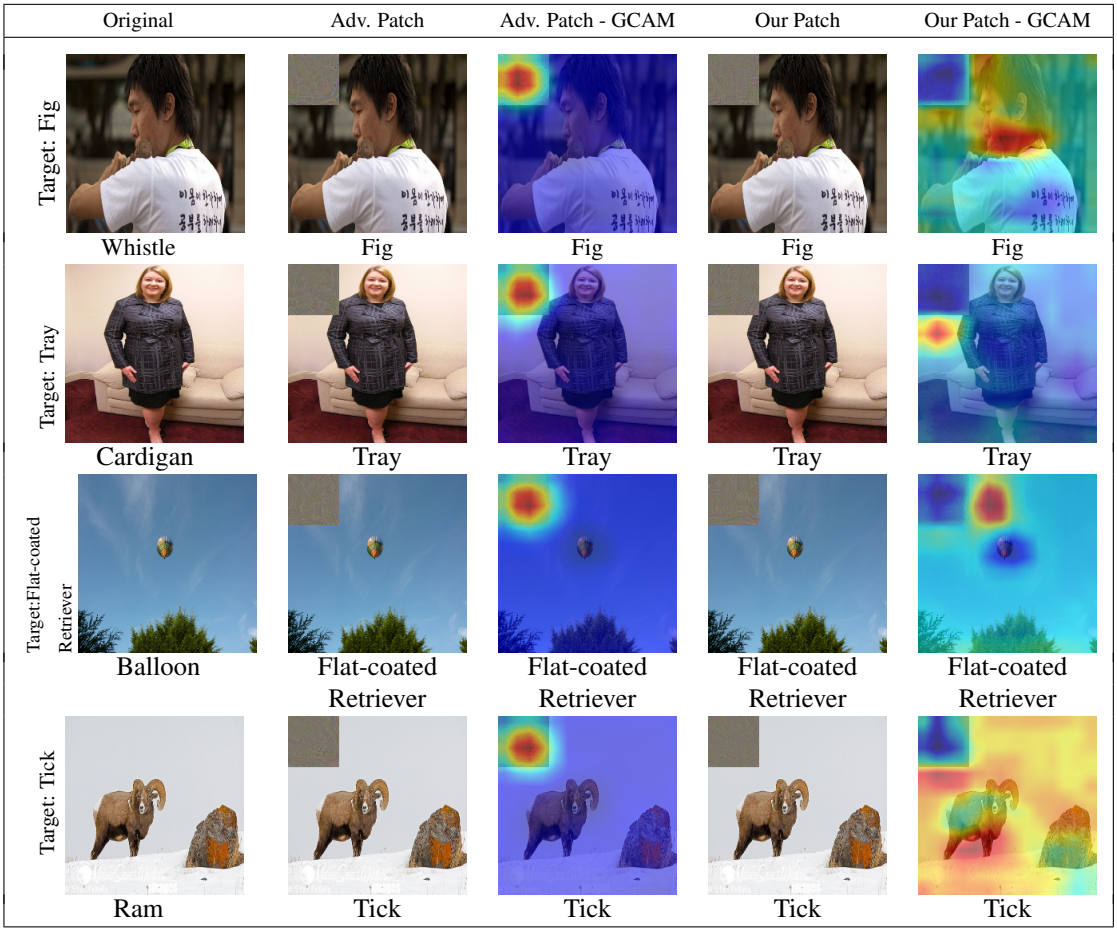

Figure A2: **Targeted patch attack:** We use an ImageNet pretrained VGG 19 BN network to compare the Grad-CAM visualization results for a random target category using our method vs Adv. Patch [3]. The predicted label is written under each image. Note that the patch is not highlighted in the last column.

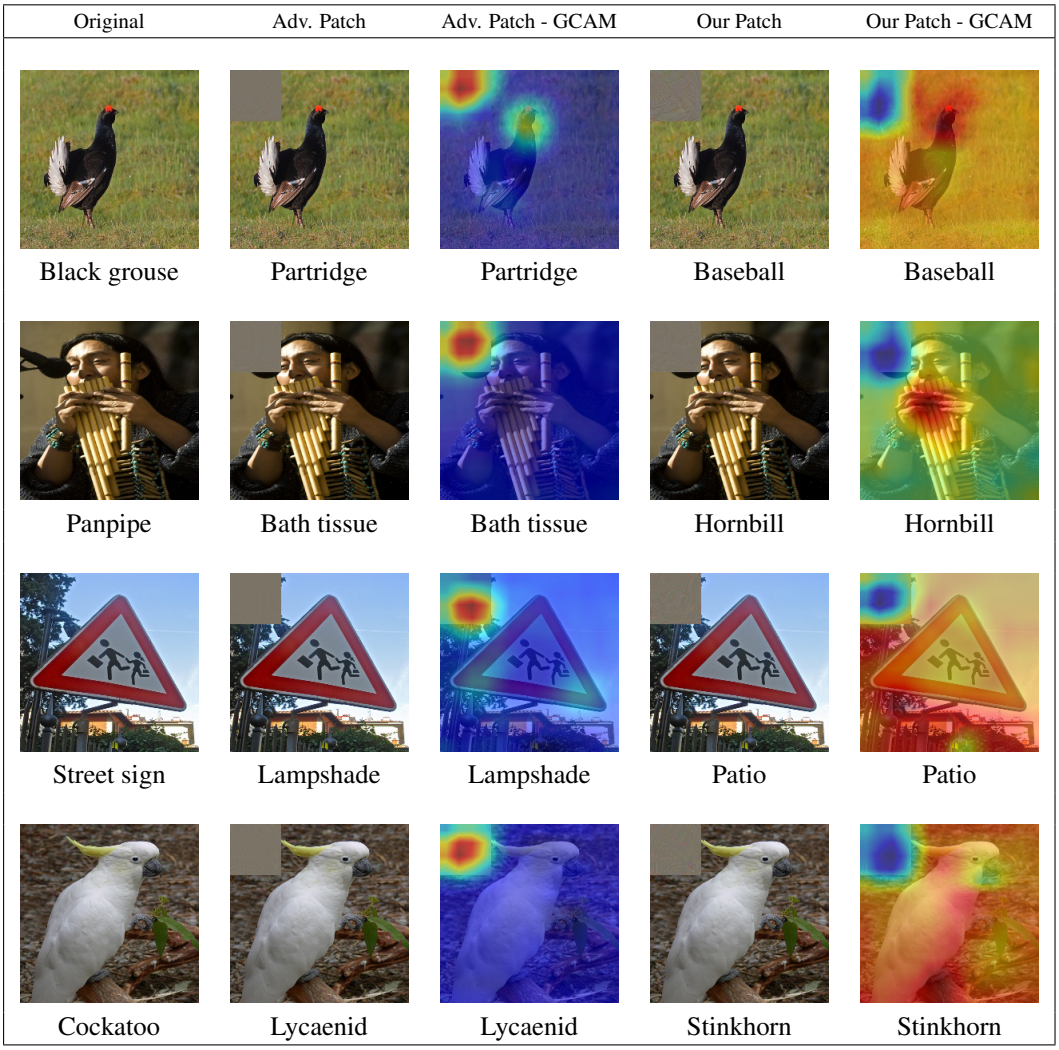

Figure A3: **Non-targeted Patch Attack:** Comparison of Grad-CAM results for non-targeted patch attacks using our method vs regular adversarial patch. We use ImageNet pre-trained VGG19-BN. The predicted label is written under each image, the non-targeted attack was successful for all images, and Grad-CAM is always computed for the predicted category. Images come from ImageNet validation set.

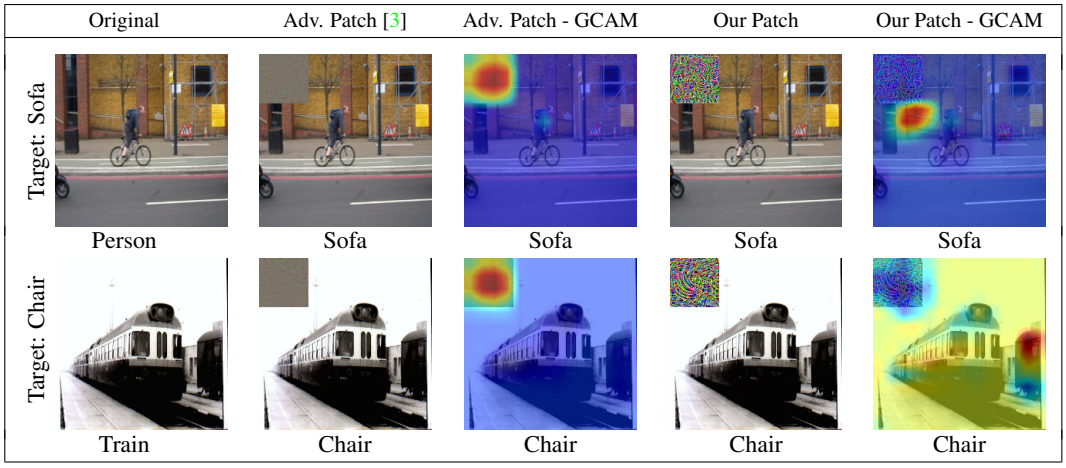

Figure A4: **Targeted attack for guided attention models:** We use GAIN$_{ext}$ [14] VGG19 model on VOC dataset to compare Grad-CAM visualization results for the least likely target category using our method vs Adv. Patch [3]. The predicted label is written under each image. GAIN$_{ext}$ is particularly designed to produce better Grad-CAM visualizations using direct supervision on the Grad-CAM output.

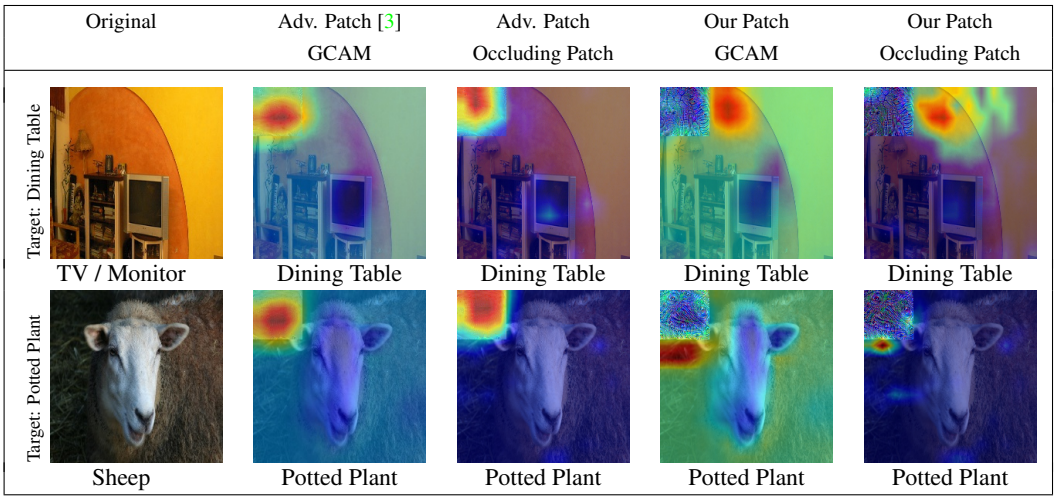

Figure A5: **Generalization beyond Grad-CAM:** Transfer of Grad-CAM visualization attack to Occluding Patch visualization. Here, we use targeted patch attacks (least likely target category) using our method vs Adv. Patch [3] on the GAIN$_{ext}$ [14] network for VOC dataset. The predicted label is written 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.

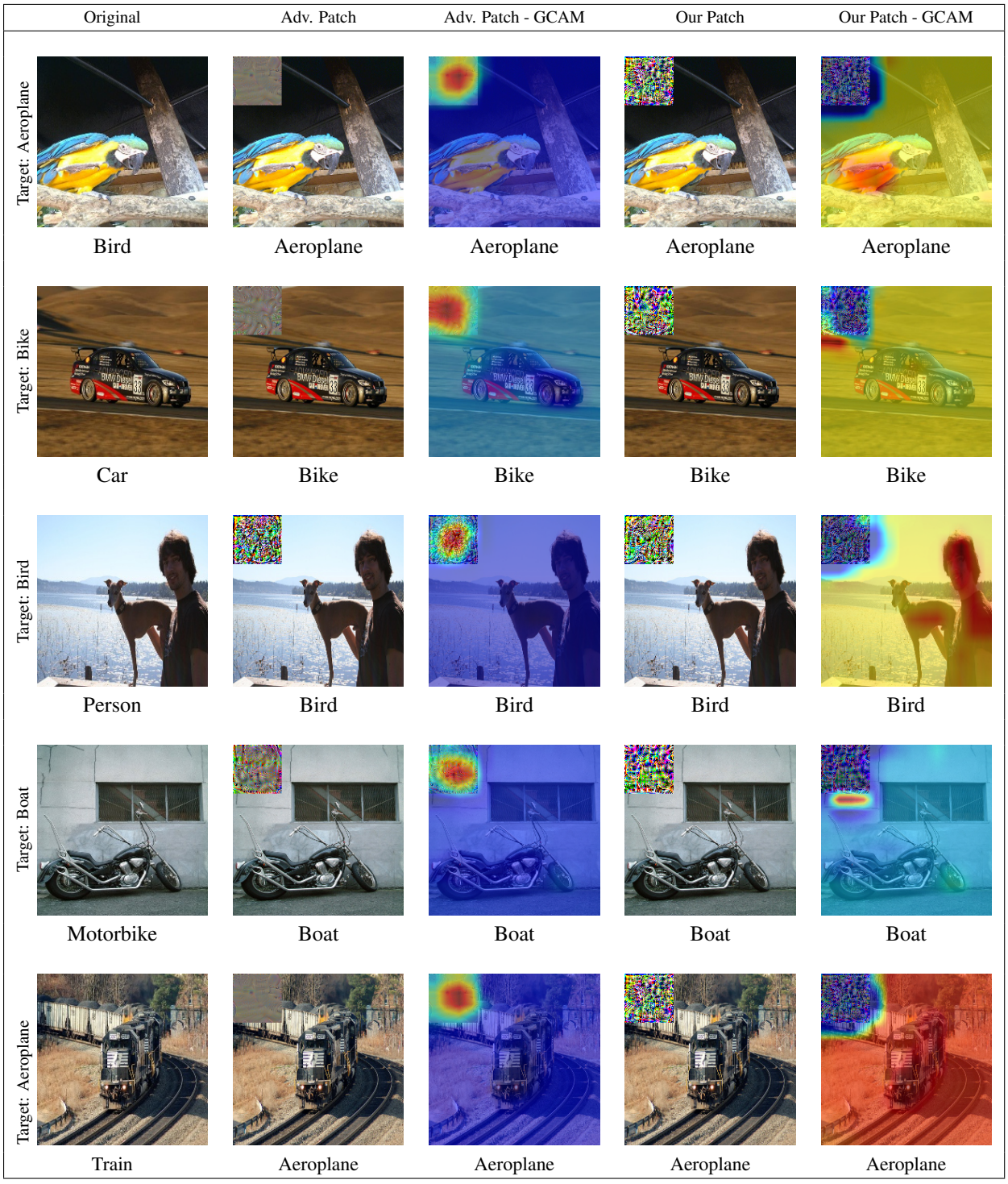

Figure A6: **Universal targeted patch attack:** As described in **Section 3.6** of the main paper, we compare Grad-CAM of our universal attack on GAIN$_{ext}$ with the regular adversarial patch. The predicted label is written under each image, the targeted attack was successful for all images, and Grad-CAM is always computed for the target category. Note that each row shows the result for a different category chosen as the universal target. Images are from PASCAL VOC-2012 validation set. The quantitative results are in **Table T4**.

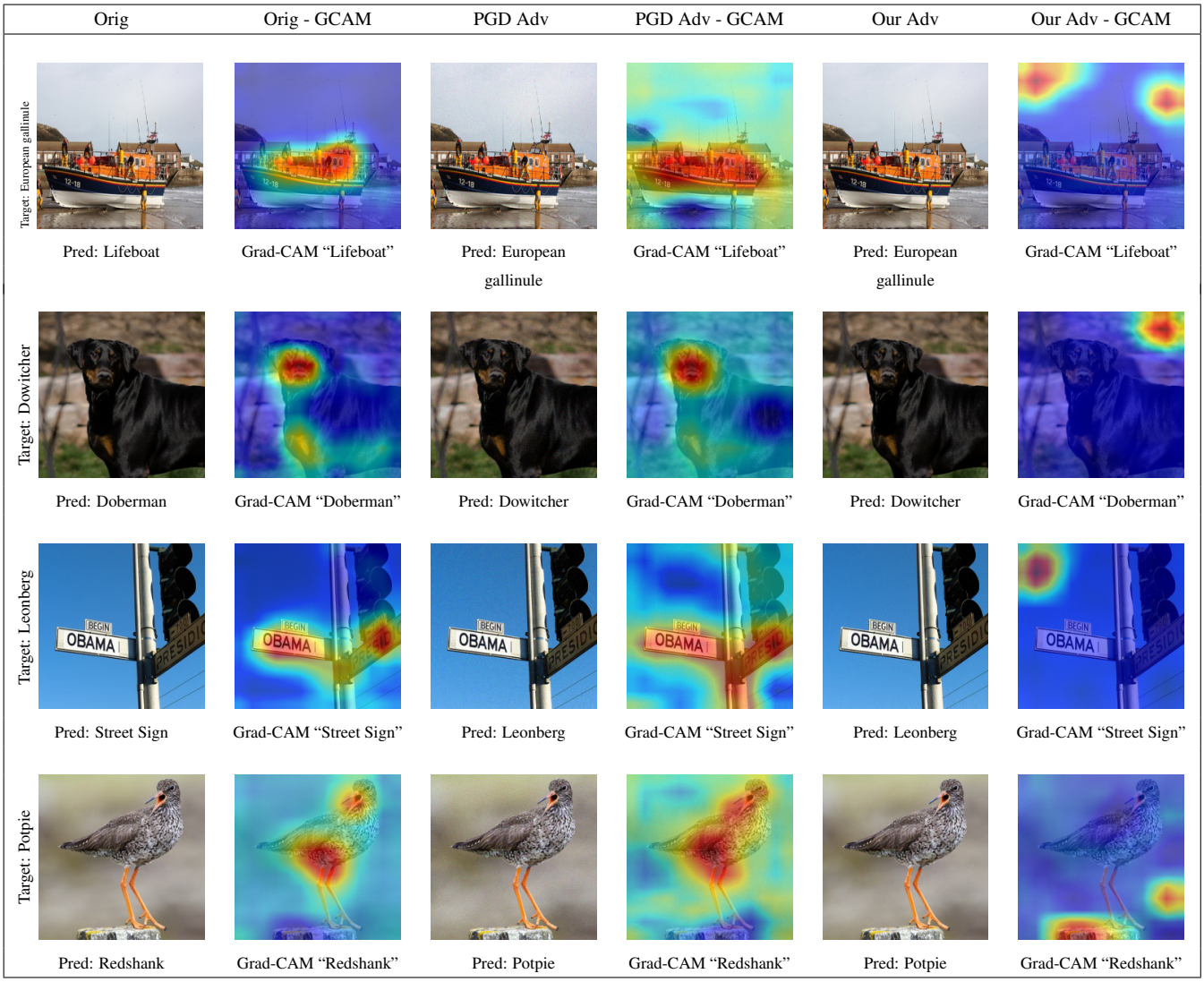

Figure A7: **Targeted regular adversarial examples:** As described in Section 5 of the main paper, we use an ImageNet pretrained VGG 19-BN network to perform a targeted attack using our method as well as using standard PGD method. Note that in this case, unlike other experiments, we compare Grad-CAM for the *original* category and not the target one. The predicted label is written under each image. The attack was successful for all images. Note that compared to the original image and the PGD adversarial image, the Grad-CAM for our adversarial image fires less on the object. This attack not only reduces the probability of the original category, but also changes its interpretation. Images are from ImageNet validation set.

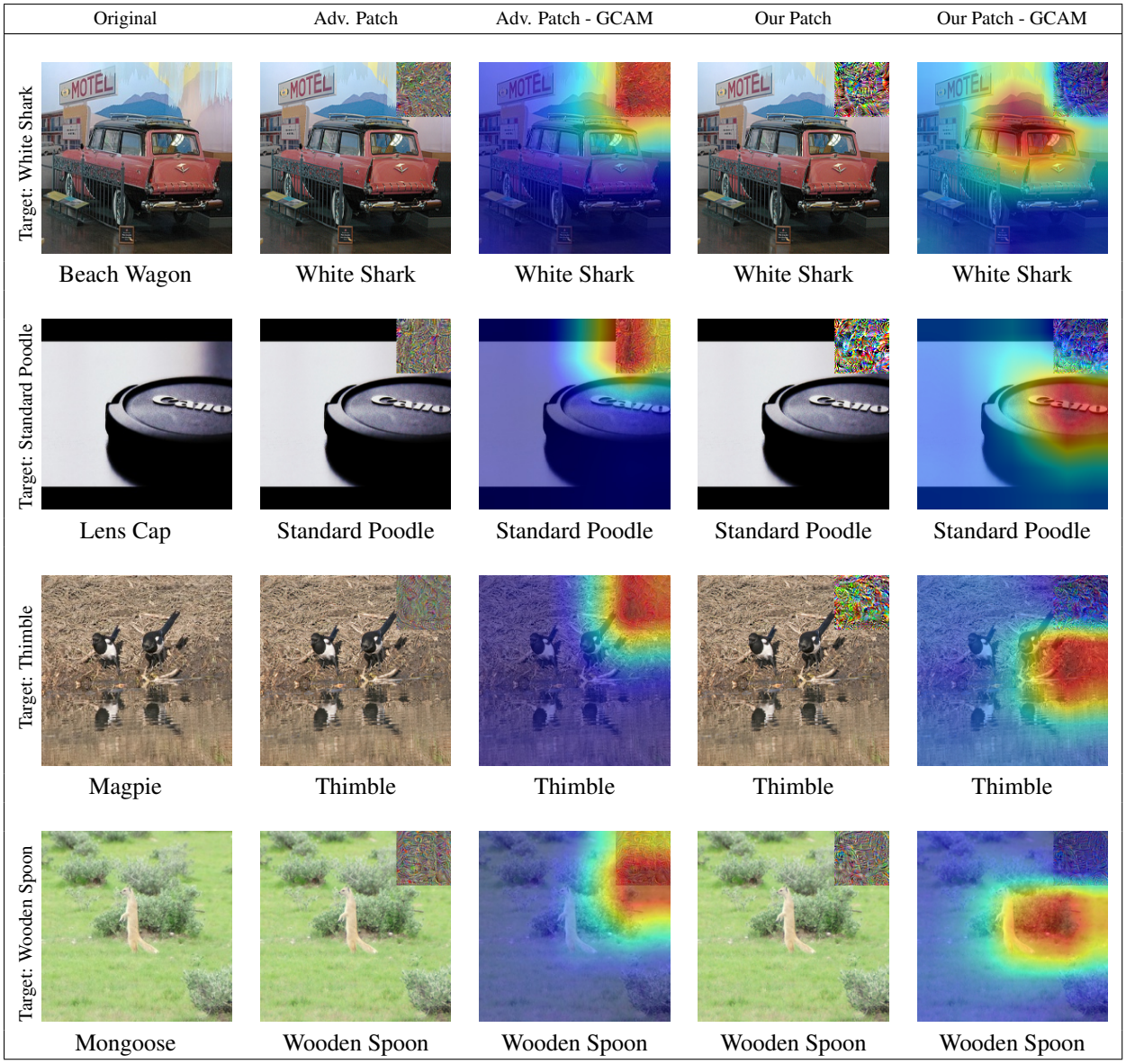

Figure A8: **Different networks and patch locations:** Comparison of Grad-CAM visualization results for our targeted patch attack vs regular adversarial patch. It uses ImageNet pretrained **ResNet-34** network with the patch on the top right corner. The predicted label is written under each image, the targeted attack was successful for all images in this figure, and Grad-CAM is always computed for the target category. Images are from ImageNet validation set.

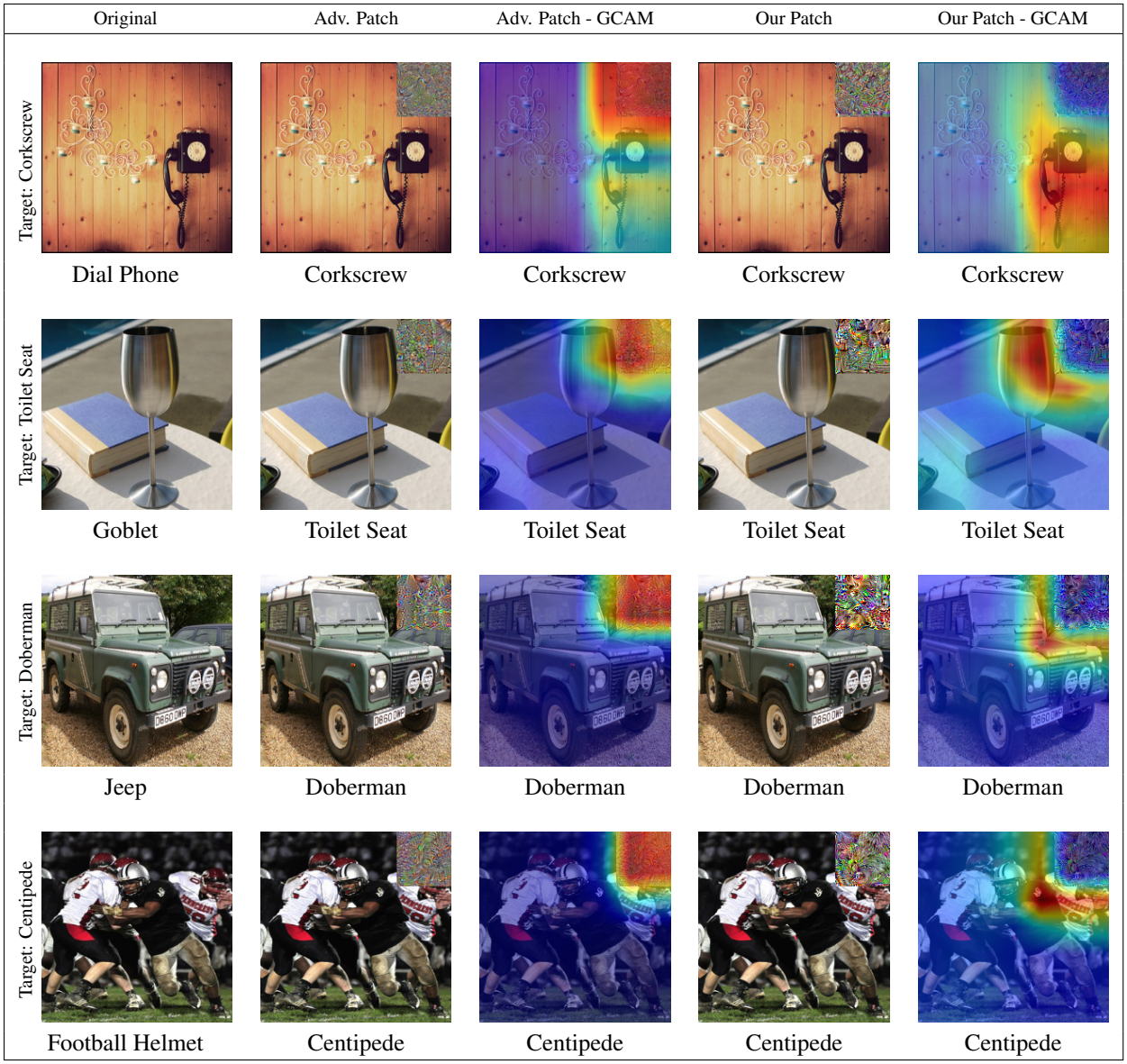

Figure A9: Similar to Figure A8, but for **DenseNet-121** network.