ONE THING TO FOOL THEM ALL: GENERATING INTERPRETABLE, UNIVERSAL, AND PHYSICALLY-REALIZABLE ADVERSARIAL FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

It is well understood that modern deep networks are vulnerable to adversarial attacks. However, conventional methods fail to produce adversarial perturbations that are intelligible to humans, and they pose limited threats in the physical world. To study feature-class associations in networks and better understand the real-world threats they face, we develop feature-level adversarial perturbations using deep image generators and a novel optimization objective. We term these *feature-fool* attacks. We show that they are versatile and use them to generate targeted feature-level attacks at the ImageNet scale that are simultaneously interpretable, universal to any source image, and physically-realizable. These attacks can also reveal spurious, semantically-describable feature/class associations, and we use them to guide the design of "copy/paste" adversaries in which one natural image is pasted into another to cause a targeted misclassification.

1 INTRODUCTION

State-of-the-art neural networks are vulnerable to adversarial inputs, which cause the network to fail yet only differ from benign inputs in subtle ways. Adversaries for visual classifiers conventionally take the form of a small-norm perturbation to a benign source image that causes misclassification (Szegedy et al., 2013; Goodfellow et al., 2014). These are effective, but to a human, these perturbations typically appear as random or mildly-textured noise. As such, analyzing these adversaries does not reveal information about the network relevant to how it will function–and how it may fail–when presented with human-interpretable features. Another limitation with conventional adversaries is that they do not tend to be physically-realizable. While they can retain some effectiveness when printed and photographed in a controlled setting (Kurakin et al., 2016), they are generally ineffective in less controlled ones such as those experienced by autonomous vehicles (Kong et al., 2020).

Several works discussed in Section 2 have aimed to produce adversarial modifications that are universal to any source image, interpretable, *or* physically-realizable. But to the best of our knowledge, none exist for accomplishing all three at once. As a means of better understanding networks and what threats they face in the real world, we set out to create adversarial examples with all of these desiderata. Fig. 1 gives an example of one of our attacks in which a universal adversarial patch depicting a crane is physically placed near sunglasses to fool a network into classifying the image as a pufferfish.

Because pixel-space optimization produces non-interpretable perturbations, the ability to manipulate images at a higher level is needed. We take inspiration from recent advancements in generative modeling (e.g. Brock et al. (2018)) at the ImageNet (Russakovsky et al., 2015) scale. Instead of pixels, we perturb the latent representations inside of a deep generator to manipulate an image in feature-space. In doing so, we produce adversarial features which are inserted into source images either by directly modifying the latents used to generate them or by inserting a generated patch into a natural image. We combine this with an optimization loss that uses an external discriminator and classifier to regularize the adversarial feature into both appearing interpretable and not resembling the attack's target class.

We use this strategy to produce what we term as *feature-fool* attacks in which a feature-level manipulation causes a misclassification yet appears to a human as some interpretable object that does

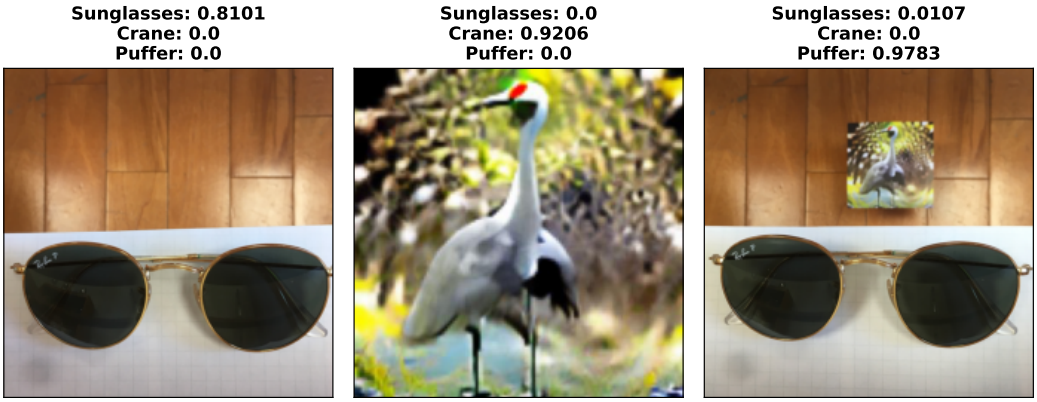

Figure 1: An example of an interpretable, universal, and physically realizable feature-fool attack against a ResNet50. The patch depicts a crane, however, when printed, physically inserted into a scene with sunglasses, and photographed, it causes a misclassification as a pufferfish. The patch was created by perturbing the latent of a generator to manipulate the image in a feature-space and training with a loss that jointly optimizes for fooling the classifier and resembling some non-target disguise class.

not resemble the target class. We show that our universal attacks are more interpretable and better disguised than analogous pixel-space attacks while being able to successfully transfer to the physical world. To further demonstrate their potential for interpretable and physically-realizable attacks, we also use these adversaries to guide the design of *copy/paste* attacks in which one natural image is pasted into another to induce an unrelated misclassification. Based on these findings, we emphasize the importance of cautious deployment for vision networks and their fortification against feature-level adversarial attacks. The following sections contain related work, methods, experiments, and a discussion. For a jargon-free summary of our work for readers who are less familiar with research in deep learning, see Appendix A.6. We are making code for generating feature-fool adversaries openly available.

2 Related Work

Conventional adversaries (Szegedy et al., 2013; Goodfellow et al., 2014) tend to be non-interpretable pixel-level perturbations that do not transfer robustly to the physical world. Here, we contextualize our approach with other work and natural examples related to overcoming these challenges.

Inspiration from Nature: Mimicry is common in nature, and in some instances, rather than holistically imitating another species' appearance, a mimic will only exhibit particular features. For example, peacocks, some butterflies, and many other animals use adversarial eyespots to stun or confuse predators (Stevens & Ruxton, 2014). Another example is the mimic octopus which imitates the patterning, but not the shape, of a banded sea snake. Peacock and ringlet butterfly are both ImageNet classes, so one cannot meaningfully test how networks trained on ImageNet respond to them. However, we show in Figure 2 using a photo of a mimic octopus from Norman et al. (2001) that a ResNet50 classifies it as a sea snake.

Generative Modeling: In contrast to pixel-space attacks, our method hinges on using a generator to manipulate images at a feature-level. One similar approach has been to train a generator or autoencoder to produce adversarial perturbations that are subsequently applied to natural inputs. This has been done by Hayes & Danezis (2018); Mopuri et al. (2018a;b); Poursaeed et al. (2018); Xiao et al. (2018); Hashemi et al. (2020); Wong & Kolter (2020) to synthesize attacks that are transferable, universal, or efficient to produce. Unlike these, however, we also explicitly focus on physical-realizability and human-interpretability. Additionally, rather than training an adversary generator, ours and other related works have skipped this step and simply trained adversarial latent perturbations to pretrained models. Liu et al. (2018) did this with a differentiable image renderer. Song et al. (2018) and Joshi et al. (2019) used deep generative networks, as well as Wang et al.

(a) (b)

Figure 2: Natural examples of adversarial features. (a) A peacock and butterfly with adversarial eyespots. (b) A mimic octopus from Norman et al. (2001) is classified as a sea snake by a ResNet50.

(2020) who aimed to create more semantically-understandable attacks by training an autoencoder with a "disentangled" embedding space. However, these works focus on small classifiers trained on simple datasets (MNIST (LeCun et al., 2010), SVHN (Netzer et al., 2011), CelebA (Liu et al., 2015) and BDD (Yu et al., 2018)). In contrast, we work at the ImageNet (Russakovsky et al., 2015) scale. Finally, compared to all of the above works, ours is also unique in that we directly regularize adversaries for interpretability and disguise with our training objective rather than relying on perturbations in latent space alone.

Physically-Realizable Attacks: We develop attacks meant to fool a classifier even when printed and photographed. This directly relates to the work of Kurakin et al. (2016) who found that conventional pixel-space adversaries could do this to a limited extent in controlled settings. More recently, Sharif et al. (2016); Brown et al. (2017); Eykholt et al. (2018); Athalye et al. (2018); Liu et al. (2019); Kong et al. (2020); Komkov & Petiushko (2021) used optimization under transformation to create adversarial clothing, stickers, patches, or objects for fooling vision systems. In contrast to each of these, we generate attacks that are not only physically-realizable but also inconspicuous in the sense that they are both interpretable and disguised.

Interpretable Adversaries: In addition to fooling models, our adversaries provide a method for discovering semantically-describable feature/class associations learned by a network. This relates to work by Geirhos et al. (2018) and Leclerc et al. (2021) who debug networks using rendering and style transfer to make a zero-order search over features, transformations, and textural changes in images that cause misclassification. More similar to our work are Carter et al. (2019) and Mu & Andreas (2020) who develop interpretations of networks using feature visualization (Olah et al., 2017) and network dissection (Bau et al., 2017) respectively. Both find cases in which their interpretations suggest a "copy/paste" attack in which a natural image of one object is pasted inside another natural image to cause a misclassification as a third object. We add to this work with a new method to identify adversarial features for copy/paste attacks, and unlike either previous approach, ours naturally does so in a context-conditional fashion.

3 METHODS

3.1 THREAT MODEL

We adopt the "unrestricted" adversary paradigm of Song et al. (2018), which requires the network's classification of an adversarial example to differ from some oracle (e.g. a human). The adversary's goal is to produce a feature that will cause a targeted misclassification without resembling the target class. In particular, we focus on attacks that are universal across a distribution of source images and physically-realizable. We assume that the adversary has access to a differentiable image generator, a corresponding discriminator, an additional auxiliary classifier (optionally), and images from the target classifier's training distribution. We also require white-box access to the target classifier, though we present black-box attacks based on transfer from an ensemble to a held-out classifier in Appendix A.2. The adversary is limited in that it can only change a certain portion of either the latent or the image, depending on the type of attack we use.

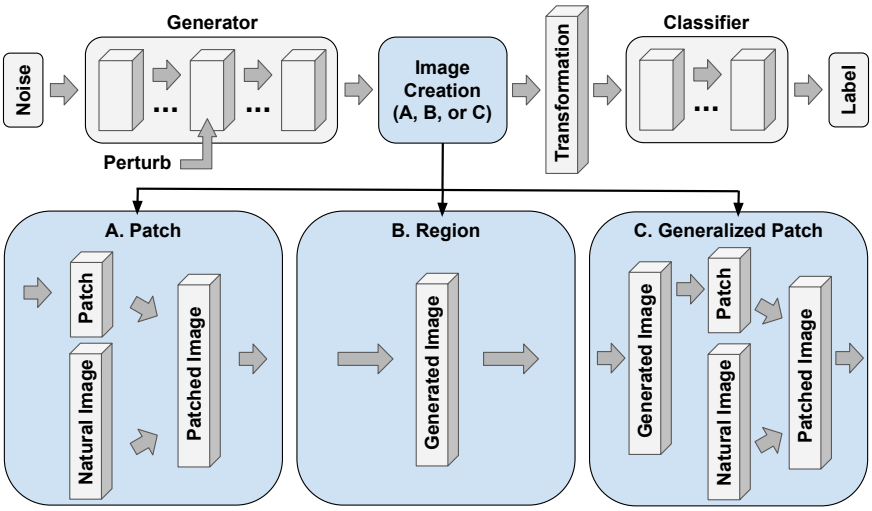

Figure 3: Our end-to-end differentiable pipeline for creating *patch*, *region*, and *generalized patch* feature-fool attacks.

3.2 Training Process

Our attacks involve manipulating the latent representation inside of a single generator layer to produce an adversarial feature. Fig. 3 outlines our overall approach. We produce three kinds of adversarial attacks, *patch*, *region*, and *generalized patch*:

Patch: We use the generator to produce a square patch that is differentiably resized and inserted into a natural image.

Region: We randomly select a square portion of the latent representation in a generator layer spanning the channel dimension but not the height or width dimensions and replace it with a learned insertion. This is analogous to a patch of the image in its pixel representation. The modified latent is then passed through the rest of the generator, producing the adversarial image.

Generalized Patch: This method produces a patch that can be of any shape, hence the name "generalized" patch. First, we generate an image in the same way that we do for region attacks. Second, we extract a generalized patch. We do this by (1) taking the absolute-valued pixel-level difference between the original and adversarial image, (2) applying a Gaussian filter for smoothing, and (3) creating a binary mask from the top decile of these pixel differences. We then apply this mask to the generated image to isolate the region of the image that the perturbation altered. We can then treat this as a patch and overlay it onto an image in any location.

Objective: For all attacks, we train a perturbation δ to the latent of the generator to minimize a loss that optimizes for both fooling the classifier and appearing as an interpretable, disguised feature:

$$\arg\min_{\delta} \mathbb{E}_{x\sim\mathcal{X},t\sim\mathcal{T},l\sim\mathcal{L}} \left[L_{\text{x-ent}}(C(A(x,\delta,t,l)), y_{\text{targ}}) + L_{\text{reg}}(A(x,\delta,t,l))\right]$$

with $\mathcal{X}$ a distribution over images (e.g. the validation set or generation distribution), $\mathcal{T}$ a distribution over transformations, $\mathcal{L}$ a distribution over insertion locations (which only applies for patch and generalized patch adversaries), C is the target classifier, A an image-generating function, $L_{\text{x-ent}}$ a targeted crossentropy loss for fooling the classifier, y_{targ} the target class, and L_{reg} a regularization loss for interpretability and disguise.

L_{reg} contains several terms. Our goal is produce features that are interpretable and disguised to a human, but absent the ability to scalably or differentiably have a human in the loop, we instead use L_{reg} as a proxy. All terms in L_{reg} for each type of attack are listed in the following section, but most crucially, it includes ones calculated using a discriminator and an auxiliary classifier. For all

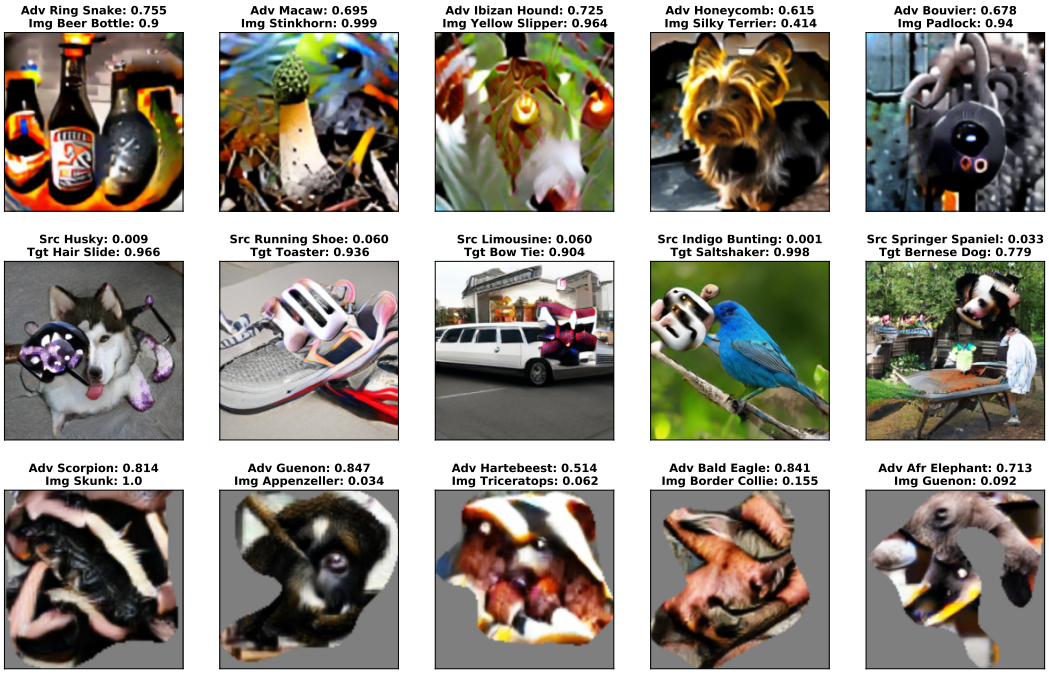

Figure 4: Examples of universal patch (top), region (middle), and generalized patch (bottom) feature-fool attacks. Each patch and generalized patch is labeled with its mean fooling confidence under insertion for random locations and source images (labeled 'Adv') and the confidence with which it is classified as the disguise class (labeled 'Img'). Numbers in each patch and generalized patch subfigure title come from different inputs, so they can add to be > 1. The region attacks are labeled with their confidence as their source class ('Src') and the target class ('Tgt').

three types of attack, we differentiably resize the patch or the extracted generalized patch and pass it through the discriminator and auxiliary classifier. We then add weighted terms to the regularization loss based on (1) the discriminator's (D) logistic loss for classifying the input as fake, (2) the softmax entropy of a classifier's (C') output, and (3) the negative of the classifier's crossentropy loss for classifying the input as the attack's target class. Thus, we have:

$$L_{\text{reg}}(a) = L_{\text{logistic}}[D(P(a))] + H[C'(P(a))] - L_{\text{x-ent}}[C'(P(a), y_{\text{targ}})] + \dots$$

Where $P(a)$ returns the extracted and resized patch from adversarial image a. These terms encourage the adversarial feature to (1) look real, and (2) look like some specific class, but (3) not the target class of the attack. The choice of what disguise class to use is left entirely to the algorithm.

4 Experiments

4.1 Attack Details

We use BigGAN generators off the shelf from Brock et al. (2018) using the implementation from Wolf (2018), and perturb the post-ReLU activations of the internal 'GenBlocks'. Notably, due to self attention inside of the BigGAN architecture, for region attacks, the change to the output image is not square even though the perturbation to the latent is. By default, we attacked a ResNet50 (He et al., 2016), and we restrict patch attacks to attacking 1/16 of the image and region and generalized patch attacks to 1/8. We found that performing our crossentropy and entropy regularization on our patches using adversarially-trained auxiliary classifiers produced subjectively more interpretable results. Presumably, this relates to how adversarially-trained networks tend to learn more interpretable representations (Engstrom et al., 2019b; Salman et al., 2020) and better approximate the human visual system (Dapello et al., 2020). So for crossentropy and entropy regularization, we used a 2-network ensemble of an $\epsilon = 4$ L_2 and $\epsilon = 3$ L_∞ robust ResNet50s from Engstrom et al.

(2019a) for regularization. For discriminator regularization, we use the BigGAN class-conditional discriminator with a uniform class vector input (as opposed to a one-hot vector). For patch adversaries, we train under colorjitter, Gaussian blur, Gaussian noise, random rotation, and random perspective transformations to simulate changes that a physically-realizable adversary would need to be robust to. For region and generalized patch ones, we only use Gaussian blurring and horizontal flipping. Also for region and generalized patch adversaries, we promote subtlety by penalizing the difference from the original image using the LPIPs perceptual distance (Zhang et al., 2018; So & Durnopianov, 2019). Finally, for all adversaries, we apply a penalty on the total variation of the patch or change induced on the image. All experiments were implemented with PyTorch (Paszke et al., 2019).

Figure 4 shows examples of universal feature level patch, region, and generalized patch attacks. In particular, the patches on the top row are effective at resembling a disguise class to the network. (We also subjectively find that they resemble the disguise class to us as well.) However, when shrunk to the size of a patch and inserted into another image, the network sees them as the target class. This suggests size biases in how networks process features. And to the extent that humans also find these patches to resemble the target, this may suggest similar properties in the human visual system. However, it is key to recognize the framing effects when analyzing these images: recognizing target-class features given the target class versus given no information are different tasks (Hullman & Diakopoulos, 2011). Analyzing human perception of feature-level adversaries may be an interesting direction for future work.

4.2 Interpretable, Universal, Physically-Realizable Attacks

To demonstrate that feature-fool adversaries are interpretable and versatile, we generate adversarial patches which appear as one object to a human, cause a targeted misclassification by the network as another, do so universally regardless of the source image, and are physically-realizable. We generated feature-fool patches using both our approach and pixel-space controls based on the methodology of Brown et al. (2017). The controls differed from the feature-fool attacks in that they were trained in pixel-space and were not optimized with any regularization for interpretability and disguise as described in Section 3. All else was kept identical including training under transformation and initializing the patch as an output from the generator. This initialization allowed for the controls to be disguised and was the same as the approach for generating disguised pixel-space patch attacks in Brown et al. (2017).

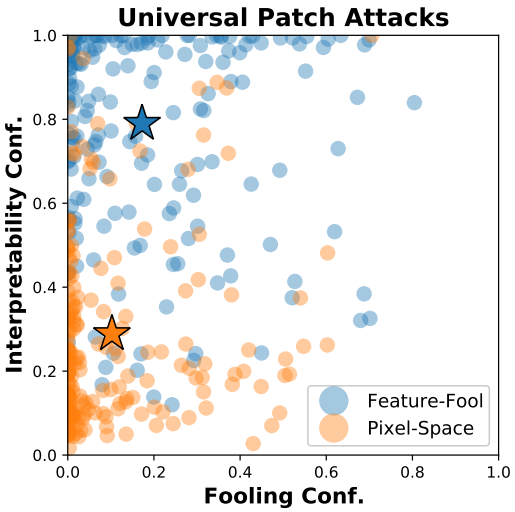

Figure 5: Feature-fool patch attacks produce more deceptive and interpretable targeted universal attacks than pixel-space controls (Brown et al., 2017). Fooling conf. shows the target class confidence. Interpretability conf. shows the auxiliary network's label class confidence when shown only the patch. Attacks further up and right are better. Centroids for both sets are shown as stars.

In Silico: Before testing in the physical world, we did so in silico with 200 feature-fool and 200 pixel-space attacks with random target classes. Fig. 5 plots the results. On the x axis are target class fooling confidences. On the y axis are the labeling confidences from the auxiliary classifier which we use as a proxy for human evaluation. For both types of attacks, crafting small patches to be targeted, universal attacks has variable success for both fooling and interpretability. The centroids for both types of attacks are denoted with stars and suggest that the feature-fools are both better at fooling the classifier and appearing interpretable on average.

That the feature-fool adversaries are more interpretable (i.e. appear as some class by the auxiliary classifier) is unremarkable because they were specifically trained to do so. More notably though,

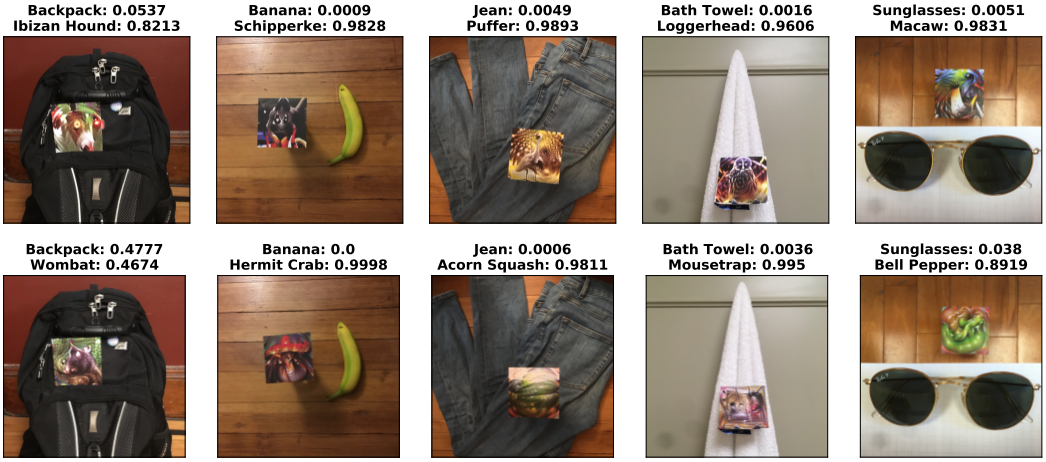

Figure 6: Successful examples of universal, physically-realizable feature-fool attacks (top) and pixel-space attacks (bottom). See Appendix A.5 for full-sized versions of the patches.

this accompanied an *increase* in mean fooling confidence. We also subjectively find the feature-fool patches to be more interpretable than the pixel-space ones. See Appendix A.5 for examples of feature-fool and pixel-space adversarial patches with high fooling confidences. Because they were initialized from generator outputs, some pixel-space patches have a veneer-like resemblance to non-target class features, but nonetheless, we find it clear from inspection that they contain higher-frequency patterns and are poorly disguised in comparison to our feature-fool attacks

In the Physical World: Next, we generated 100 additional feature-fool and pixel-space adversarial patches, selected the 10 with the best mean fooling confidence, printed them, and photographed them next to 9 different ImageNet classes of common household items.[1] We confirmed that photographs of each object with no patch were correctly classified and analyzed the outputs of the classifier when the adversarial patches were added in the physical scene.

Figure 6 shows successful examples of these physically-realizable feature-fool and pixel-space patch attacks. Meanwhile, resizable and printable versions of all 10 feature-fool and pixel-space patches are in Appendix A.5. The mean and standard deviation of the fooling confidence for the feature-fool attacks in the physical world were 0.312 and 0.318 respectively ($n = 90$) while for the pixel-space attacks, they were 0.474 and 0.423 ($n = 90$). However, we do not attempt any hypothesis tests due to nonindependence between the results across classes due to the same set of patches being used for each class. These tests in the physical world show that the feature-fool attacks were often effective but that there is high variability in this effectiveness. The comparisons to pixel-space attacks provide some evidence that unlike our results in silico, the feature-fool attacks may be less reliably successful in the real world than the controls. Nonetheless the overall level of fooling success between both groups was comparable.

4.3 Interpretability and Copy/Paste Attacks

Using adversarial examples to better interpret networks has been proposed by Dong et al. (2017) and Tomsett et al. (2018). Unlike conventional adversaries, feature-level adversaries reveal feature-class associations, which are potentially of greater practical interest. For experiments with interpreting a classifier, we use versions of our attacks in without the regularization terms described in Section 3.2. We find that inspecting the resulting adversarial features suggest both good and bad feature-class associations. As a simple demonstration, Fig. 7 shows two simple examples in which the barbershop class is desirably associated with barber-pole-stripe-like features and in which the bikini class is undesirably associated with caucasian-colored skin.

[1] Backpack, banana, bath towel, lemon, jeans, spatula, sunglasses, toilet tissue, and toaster.

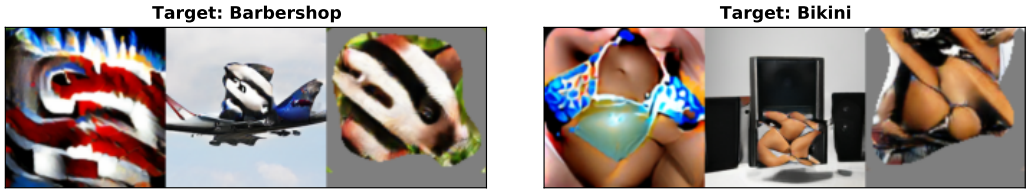

Figure 7: Examples of good and bad features class associations in which barber pole stripes are associated with a barbershop and caucasian-colored skin is associated with a bikini. Patch adversaries are on the left, region adversaries in the middle, and generalized patch adversaries on the right.

Copy/Paste Attacks: A copy-paste attack is one in which a natural image is inserted into another in order to cause an unexpected misclassification. They are more restricted than the attacks in Section 4.2 because the features pasted into an image must be natural objects rather than ones whose synthesis can be controlled. As a result, they are of high interest for developing physically-realizable attacks because they suggest combinations of real objects that could yield unexpected classifications. They also have precedent in the real world. For example, feature insertions into pornographic images have been used to evade NSFW content detectors (Yuan et al., 2019).

To develop copy/paste attacks, we select a source and target class, develop class-universal adversarial features, and analyze them for common motifs that resemble natural objects. Then we paste images of these objects into natural images and pass them through the classifier. Two other works have previously developed copy/paste attacks, also via interpretability tools that discover feature-class associations: Carter et al. (2019) and Mu & Andreas (2020). However, compared to prior approaches, our technique may be uniquely equipped to produce germane fooling features. Rather than simply producing features associated with the target class, our adversaries generate fooling features *conditional* on the distribution, $\mathcal{X}$, over source images (i.e. the source class) with which the adversaries are trained. This method allows for any source/target classes to be selected, but we find the clearest success in generating copy/paste attacks when they are somewhat related (e.g. bee and fly).

Fig. 8 gives three illustrative examples. For each attack, we show two example images for each of the patch, region, and generalized patch adversaries. Below these are the copy/paste adversaries with average target class confidence before and after feature insertion for the 6 (out of 50) images for the source class in the ImageNet validation set for which the insertion resulted in the highest target confidence. Overall, little work has been done on copy/paste adversaries, and thus far, methods have always involved a human in the loop. This makes objective comparisons between methods difficult. However we provide examples of a feature-visualization based tool inspired by Carter et al. (2019) in Appendix A.3 to compare with ours.

5 Discussion

By using a generative model to synthesize adversarial features, we contribute to a more pragmatic understanding of deep networks and their vulnerabilities. As an attack method, our approach is simple and versatile. Across experiments here and in the Appendix, we show that it can be used to produce targeted, interpretable, disguised, universal, physically-realizable, black-box, and copy/paste attacks at the ImageNet level. To the best of our knowledge, we are the first to introduce a method with *all* of these capabilities. As an interpretability method, this approach is also effective as a targeted means of searching for good, bad, and adversarially exploitable feature-class associations.

Conventional adversaries reveal intriguing properties of the learned representations in deep neural networks. However, as a means of attacking real systems, they pose limited threats outside of the digital domain (Kurakin et al., 2016). Given our results and related work, a focus on adversarial features and robust, physically-realizable attacks will be key to understanding practical threats. Importantly, even if a deep network is adversarially trained to be robust to one class of perturbations, this does not guarantee robustness to others that may be used to attack it in deployment. For better or for worse, feature-fool attacks are effective and easy to make using pretrained models. Conse-

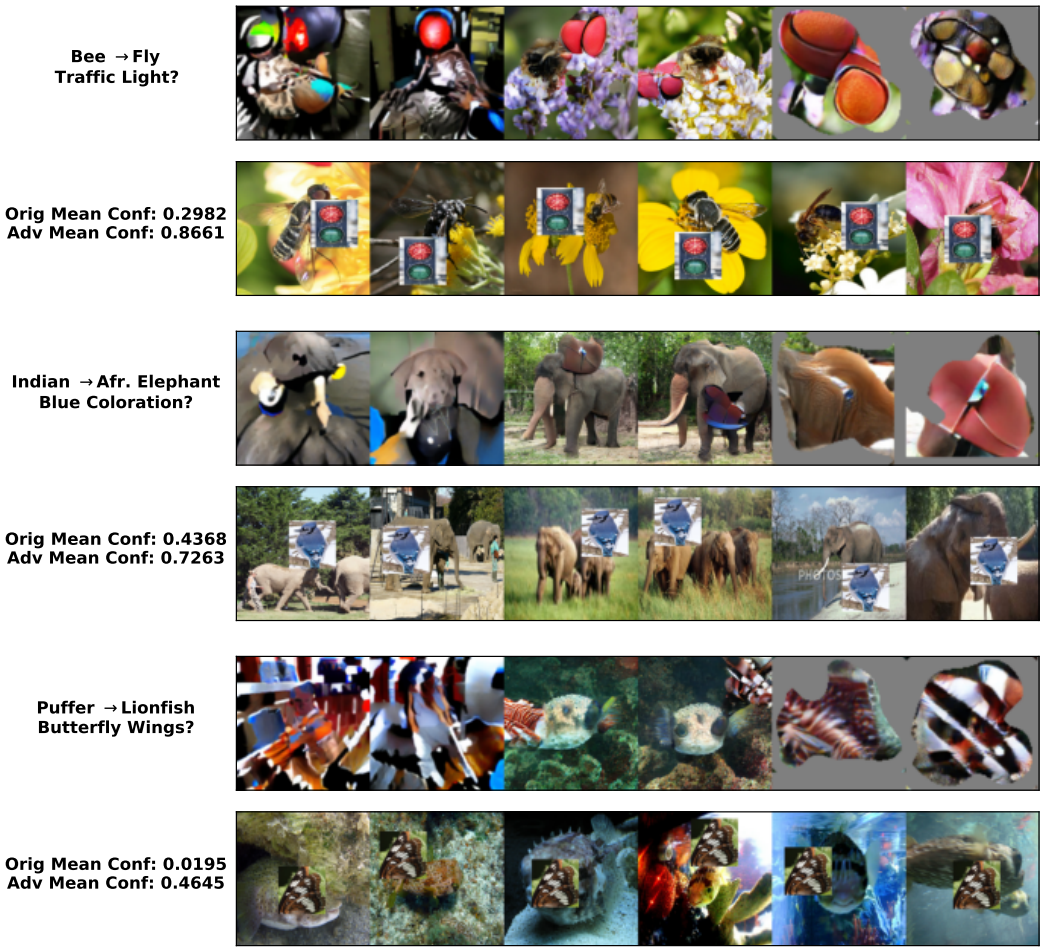

Figure 8: Patch, region, and generalized patch adversaries being used to guide three class-universal copy/paste adversarial attacks. Patch adversary example pairs are on the left, region adversaries in the middle, and generalized patch adversaries on the right of each odd row. Six successful attack examples are on each even row.

quently, we argue for focusing on pragmatic threats, training robust models (e.g. Engstrom et al. (2019a); Dapello et al. (2020)), and the use of caution with deep networks in the real world. As a promising sign, we show in Appendix A.4 that adversarial training is useful against our attacks.

A limitation is that when more constraints are applied to the adversarial generation process (e.g. universality + physical-realizability + disguise), attacks are generally less successful, and more screening is required to find good ones. They also take more time to generate which could be a bottleneck to using them for adversarial training. Further still, while we develop disguised adversarial features, we do not generally find them to be innocuous in that they often have somewhat unnatural forms typical of generated images. In this sense, our disguised attacks may nonetheless be detectable. Ultimately, this type of attack is limited by the efficiency and quality of the generator.

Future work should leverage new advances in generative modeling. One possibly useful technique could be to develop fooling features adversarially against a discriminator which is trained to recognize them from natural features. We also believe that studying human responses to feature-level adversaries and the links between interpretable representations, robustness, and similarity to the primate visual system (Dapello et al., 2020) are promising directions for better understanding both networks and biological brains. While more work remains to be done in grasping the inner representations of deep networks and ensuring that they are robust, we nonetheless believe that these findings make significant progress in understanding deep networks and the practical threats they face.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pp. 284–293. PMLR, 2018.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. Unrestricted adversarial examples via semantic manipulation. *arXiv preprint arXiv:1904.06347*, 2019.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 4(3):e15, 2019.

Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David D Cox, and James J DiCarlo. Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *BioRxiv*, 2020.

Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019a. URL `https://github.com/MadryLab/robustness`.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019b.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1625–1634, 2018.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Atiye Sadat Hashemi, Andreas Bär, Saeed Mozaffari, and Tim Fingscheidt. Transferable universal adversarial perturbations using generative models. *arXiv preprint arXiv:2010.14919*, 2020.

Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 43–49. IEEE, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jessica Hullman and Nick Diakopoulos. Visualization rhetoric: Framing effects in narrative visualization. *IEEE transactions on visualization and computer graphics*, 17(12):2231–2240, 2011.

Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. Semantic adversarial attacks: Parametric transformations that fool deep classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4773–4783, 2019.

Lim Kiat. Lucent. `https://github.com/greentfrapp/lucent`, 2019.

Stepan Komkov and Aleksandr Petiushko. Advhat: Real-world adversarial attack on arcface face id system. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 819–826. IEEE, 2021.

Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14254–14263, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

Guillaume Leclerc, Hadi Salman, Andrew Ilyas, Sai Vemprala, Logan Engstrom, Vibhav Vineet, Kai Xiao, Pengchuan Zhang, Shibani Santurkar, Greg Yang, et al. 3db: A framework for debugging computer vision models. *arXiv preprint arXiv:2106.03805*, 2021.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive gan for generating adversarial patches. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1028–1035, 2019.

Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. *arXiv preprint arXiv:1808.02651*, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Luke Melas. Pytorch-pretrained-vit. `https://github.com/lukemelas/PyTorch-Pretrained-ViT`, 2020.

Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 742–751, 2018a.

Konda Reddy Mopuri, Phani Krishna Uppala, and R Venkatesh Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 19–34, 2018b.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *arXiv preprint arXiv:2006.14032*, 2020.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Mark D Norman, Julian Finn, and Tom Tregenza. Dynamic mimicry in an indo–malayan octopus. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1478):1755–1758, 2001.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4422–4431, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *ArXiv preprint arXiv:2007.08489*, 2020.

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*, pp. 1528–1540, 2016.

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

Uchida So and Ihor Durnopianov. Lpips pytorch. https://github.com/S-aiueo32/lpips-pytorch, 2019.

Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. *arXiv preprint arXiv:1805.07894*, 2018.

Martin Stevens and Graeme D Ruxton. Do animal eyespots really mimic eyes? *Current Zoology*, 60(1), 2014.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Richard Tomsett, Amy Widdicombe, Tianwei Xing, Supriyo Chakraborty, Simon Julier, Prudhvi Gurram, Raghuveer Rao, and Mani Srivastava. Why the failure? how adversarial examples can provide insights for interpretable machine learning. In *2018 21st International Conference on Information Fusion (FUSION)*, pp. 838–845. IEEE, 2018.

Shuo Wang, Shangyu Chen, Tianle Chen, Surya Nepal, Carsten Rudolph, and Marthie Grobler. Generating semantic adversarial examples via feature manipulation. *arXiv preprint arXiv:2001.02297*, 2020.

Thomas Wolf. Pytorch pretrained biggan. https://github.com/huggingface/pytorch-pretrained-BigGAN, 2018.

Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.

Kan Yuan, Di Tang, Xiaojing Liao, XiaoFeng Wang, Xuan Feng, Yi Chen, Menghan Sun, Haoran Lu, and Kehuan Zhang. Stealthy porn: Understanding real-world adversarial images for illicit online promotion. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 952–966. IEEE, 2019.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

A APPENDIX

A.1 CHANNEL ATTACKS

In contrast to the region attacks presented in the main paper, we also experiment with *channel* attacks. For region attacks, we optimize an insertion to the latent activations of a generator's layer which spans the channel dimension but not the height and width. This is analogous to a patch attack in pixel-space. For channel attacks, we optimize an insertion which spans the height and width dimensions but only involves a certain proportion of the channels. This is analogous to an attack that only modifies the R, G, or B channel of an image in pixel-space. Unlike the attacks in Section, 4.1, we found that it was difficult to create universal channel attacks (single-image attacks, however, were very easy). Instead, we relaxed this goal and created class-universal ones which are meant to cause any generated example from one random source class to be misclassified as a target. We also manipulate $1/4^{\text{th}}$ of the latent instead of $1/8^{\text{th}}$ as we do for region attacks. Mean fooling rates and examples from the top 5 attacks out of 16 are shown in Fig. 9. They to induce textural changes somewhat like adversaries crafted by Geirhos et al. (2018) and Bhattad et al. (2019)).

Figure 9: Examples of original images (top) alongside class-universal channel adversaries (bottom). Each image is labeled with the source and target class confidence.

A.2 BLACK-BOX ATTACKS

Adversaries are often created using first order optimization on an input to a network which requires that the network's parameters are known. However, adversaries are often transferrable between models (Papernot et al., 2016), and one method for developing black-box attacks is to train against a different model and then transfer to the intended target. We do this for our adversarial patches and generalized patches by attacking a large ensemble of AlexNet (Krizhevsky et al., 2012), VGG19 (Simonyan & Zisserman, 2014), Inception-v3 (Szegedy et al., 2016), DenseNet121 (Huang et al., 2017), ViT(Dosovitskiy et al., 2020; Melas, 2020), and two robust ResNet50s Engstrom et al. (2019a), and then transferring to ResNet50 He et al. (2016). Otherwise, these attacks were identical to the ones we used in Section 4.1. These attacks were crafted with a random-source/target class and optimization for disguise. Many were unsuccessful, but a sizable fraction were able to fool the ResNet50 with a mean confidence of over 0.1 for randomly sampled images. The top 5 out of 64 of these attacks for patch and generalized patch adversaries are shown in Fig. 10.

A.3 COPY/PASTE ATTACKS WITH CLASS IMPRESSIONS

Mu & Andreas (2020) and Carter et al. (2019) both used interpretability methods to guide the development of copy/paste adversaries. Mu & Andreas (2020), used an interpretability method known as network dissection (Bau et al., 2017) to develop interpretations of neurons and fit a semantic description in compositional logic over the network using those interpretations and the network's weight magnitudes. This allowed them to identify cases in which the networks learned undesirable

Figure 10: Black-box adversarial patches (top) and generalized patches (bottom) created using transfer from an ensemble. Patches are displayed alongside their target class and mean fooling confidence.

feature-class associations. However, this approach cannot be used to make a targeted search for copy/paste attacks that will cause a given source class to be misclassified as a given target.

More similar to our work is Carter et al. (2019) who found inspiration for successful copy/paste adversaries by creating a dataset of visual features and comparing the differences between ones which the network assigned the source versus target label. We use inspiration from this approach to create a baseline against which to compare our method of designing copy/paste attacks in Section 4.3. Given a source and target class such as a bee and a fly, we optimize a set of inputs to a network for each class in order to maximize the activation of the output node for that class. Mopuri et al. (2018b) refers to these as *class impressions*. We train these inputs under transformations and with a decorrelated frequency-space parameterization of the input pixels using the Lucent (Kiat, 2019) package. We do this for the same three class pairs as in Fig. 8 and display 6 per class in Fig. 11. In each subfigure, the top row gives class impressions of the source class, and the bottom gives them for the target. Each class impression is labeled with the network's confidences for the source and target class. In analyzing these images and comparing to the ones from Fig. 11, we find no evidence of more blue coloration in the African elephant class impressions than the Indian ones. However, we find it plausible that some of the features in the fly class impression may resemble traffic lights and that those for the Lionfish may resemble an admiral Butterfly's wings. Nonetheless, these visualizations are certainly different in appearance from our adversarial ones.

These class impressions seem comparable but nonredundant with our adversarial method from Section 4.3. However, our adversarial approach may have an advantage over the use of class impressions in that it is equipped to design features that look like the target class *conditional* on the rest of the image being of the source class. Contrastingly, a class impression is only meant to visualize features typical of the target class. It is possible that this is why our adversarial attacks were able to show that inserting a blue object into an image of an Indian elephant can cause a misclassification as an African elephant – two very similar classes – while the class impressions for the two appear very similar and suggest nothing of the sort.

A.4 Defense via Adversarial Training

Adversarial training is a common and broadly effective means for improving robustness. Here, to test how effective it is for our attacks, for 5 pairs of similar classes, we generate datasets of 1024 images evenly split between each class and images with/without adversarial perturbations. We do this separately for channel, region, and patch adversaries before treating the victim network as a binary classifier and training on the examples. We report the post-training minus pre-training accuracies in Tbl. A.4 and find that across the class pairs and attack methods, the adversarial training improves binary classification accuracy by a mean of 42%.

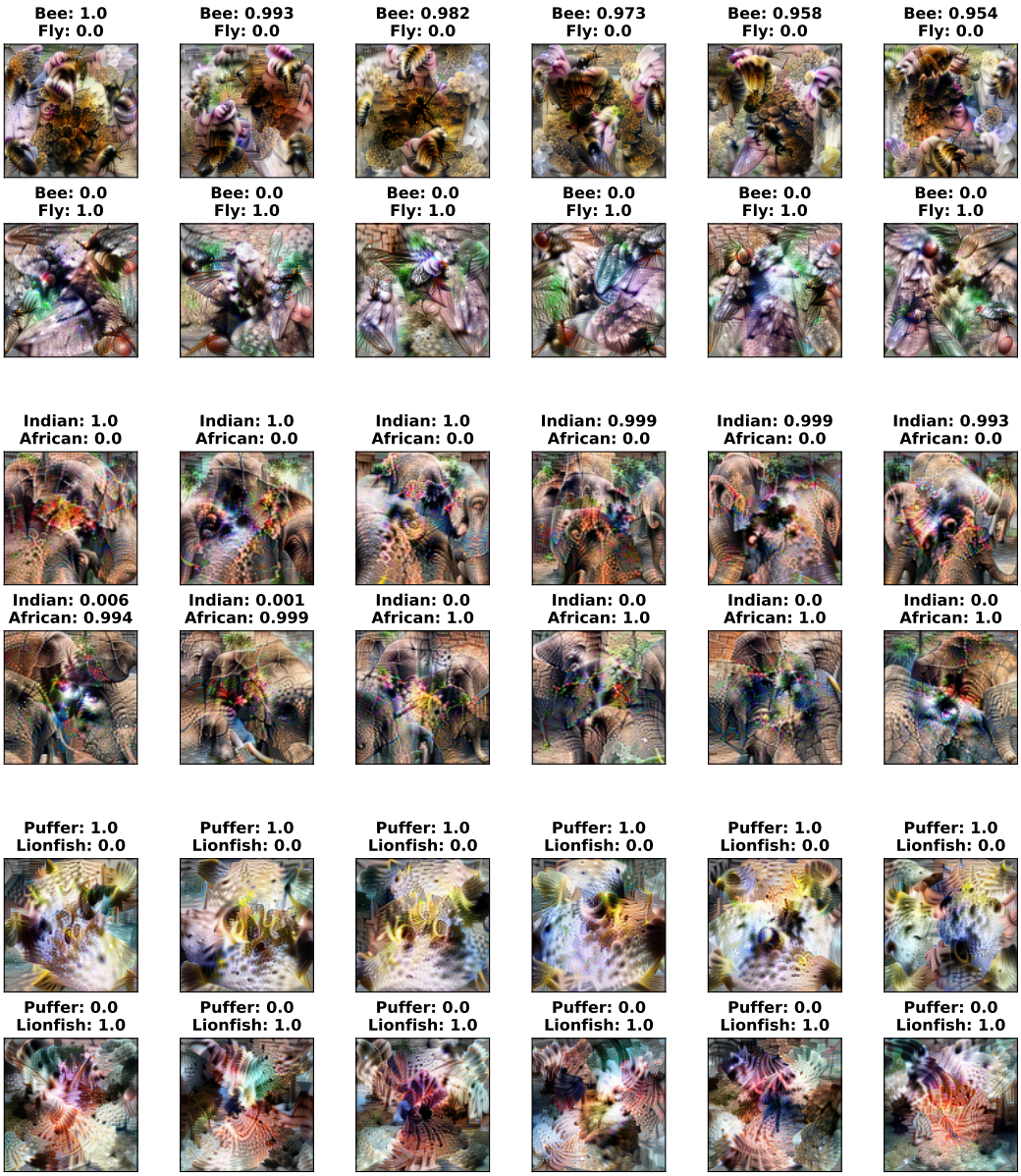

Figure 11: Class impressions for three pairs of classes. These could be used for providing insight about copy/paste attacks in the same way as the examples from Fig. 8. Each subfigure is labeled with the network's output confidence for both classes.

	Channel	Region	Patch	Mean
Great White/Grey Whale	0.49	0.29	0.38	0.39
Alligator/Crocodile	0.13	0.29	0.60	0.34
Lion/Tiger	0.29	0.28	0.63	0.40
Frying Pan/Wok	0.32	0.39	0.68	0.47
Scuba Diver/Snorkel	0.42	0.36	0.69	0.49
Mean	0.33	0.32	0.60	**0.42**

Table 1: Binary classification accuracy improvements from adversarial training for channel, region, and patch adversaries across 5 class pairs.

A.5 RESIZABLE, PRINTABLE PATCHES

See Figs. 12 and 13 for feature-fool and control adversarial images respectively. We encourage readers to experiment with these images which were optimized to fool a ResNet50. In doing so, one might find a mobile app to be convenient. We used Photo Classifier.[2]

A.6 JARGON-FREE SUMMARY

AI and related fields are making rapid progress, but there exists a communication gap between researchers and the public which too-often serves as a barrier to the spread of information outside the field. For readers who may not know all of the field's technical concepts and jargon, we provide a more readable summary here.

Historically, it has proved difficult to write conventional computer programs that accurately classify real-world images. But this task has seen revolutionary success by neural networks which can now classify images into hundreds or thousands of categories sometimes with higher accuracy than humans. Despite the impressive performance, we still don't fully understand the features that these networks use to classify images, and we cannot be confident that they will always do so correctly. In fact, past research has demonstrated that it is usually very easy to take an image that the network classifies correctly and perturb its pixel values by a tiny amount – often imperceptibly to a human – but in such a way that the network will misclassify it with high confidence as whatever target class the attacker desires. For example, we can take an elephant, make minute changes in a few pixels, and make the network believe that it is a dog. Researchers have also discovered perturbations that can be added onto a wide range of images to cause them to be misclassified, making those perturbations "universal". In general, this process of designing an image that the network will misclassify is called an "adversarial attack" on the network.

Unfortunately, conventional adversarial attacks tend to produce perturbations that are not interpretable. To a human, they usually just appear as pixelated noise. As a result, they do not help us to understand how networks will process sensible inputs, and they do not reveal weaknesses that could be exploited by adversarial features in the real world. To make progress toward solving these problems, we focus on developing interpretable adversarial features. In one sense, this is not a new idea. Quite the opposite, in fact – there are already examples in the animal kingdom. Figure 2 shows examples of adversarial eyespots on a peacock and butterfly and adversarial patterns on a mimic octopus.

To generate interpretable adversarial features, we introduce a method that uses "generative" modeling. In addition to classifying images, networks can also be used to great effect for learning to generate them. These networks are often trained with the goal of producing images that are so realistic that one cannot tell whether they came from the training set or not, and modern generation methods are moving closer to this goal. Typically, these networks take in a random input and form it into an image. Inside this process are intermediate "latent" representations of the image at each "layer" inside the generator that gradually shift from abstract, low-dimensional representations of high-level features of the image (e.g. the original input) to the actual pixel values of the final image.

In order to create interpretable adversarial images, we take images created by the generator and use them for adversarial attacks. In the simplest possible procedure, one can generate an image and then modify the generator's representations to change the generation of the image in such a way that the classifier is fooled by it. We also found that optimizing under transformations to our images (like blurring, cropping, flipping, rotating, etc.) and adding in some additional terms into our optimization objective to encourage more interpretable and better disguised images greatly improved results. Ultimately, we produce adversarial images that differ from normal ones in higher-level, more intelligible ways than conventional adversaries.

These adversaries are useful and informative in two main ways. First, they allow us to create patches that are simultaneously "disguised", "physically-realizable", and "universal" at the same time. By "disguised", we mean that they look like one thing to a human but cause an unrelated misclassification. By "physically-realizable", we mean that these images can be printed and physically placed in a scene with some other object, causing a photo of the scene to be misclassified. And by "universal",

[2] https://apps.apple.com/us/app/photo-classifier/id1296922017

we mean that these images can cause photos of a wide range of objects to be misclassified as the target class. As an example, Fig. 1 shows a patch of a crane that can be physically inserted next to any real world object (such as sunglasses) in order to cause a misclassification as a pufferfish.

Second, these adversaries allow us to interpret networks by revealing feature-class associations. We even find that this can be used to create an additional type of attack. We show that this process can guide the creation of "copy/paste" attacks in which one natural image is pasted as a patch into another in order to cause a particular misclassification. Some of these are unexpected. For example, in Fig. 8, we find that a traffic light can make a bee look like a fly. These copy/paste attacks also have implications for physically-realizable attacks because they suggest combinations of real objects that could yield unexpected classifications.

Together, our findings offer potential for better understanding network representations and better predicting the ways that they may fail. We join others in the AI community in calling for caution and adversarial robustness when deploying networks in the real world.

A.7 EPITAPH

One thing to fool them all,
One class to assign them,
One thing to see it all,
And in the real world find them.[3]

[3] Adapted from J.R.R. Tolkein's *Ring Verse*.

Figure 12: Printable examples of the disguised, transformation-robust, physically-realizable feature-fool adversarial patches from Section 4.2. Patches can be resized before printing.

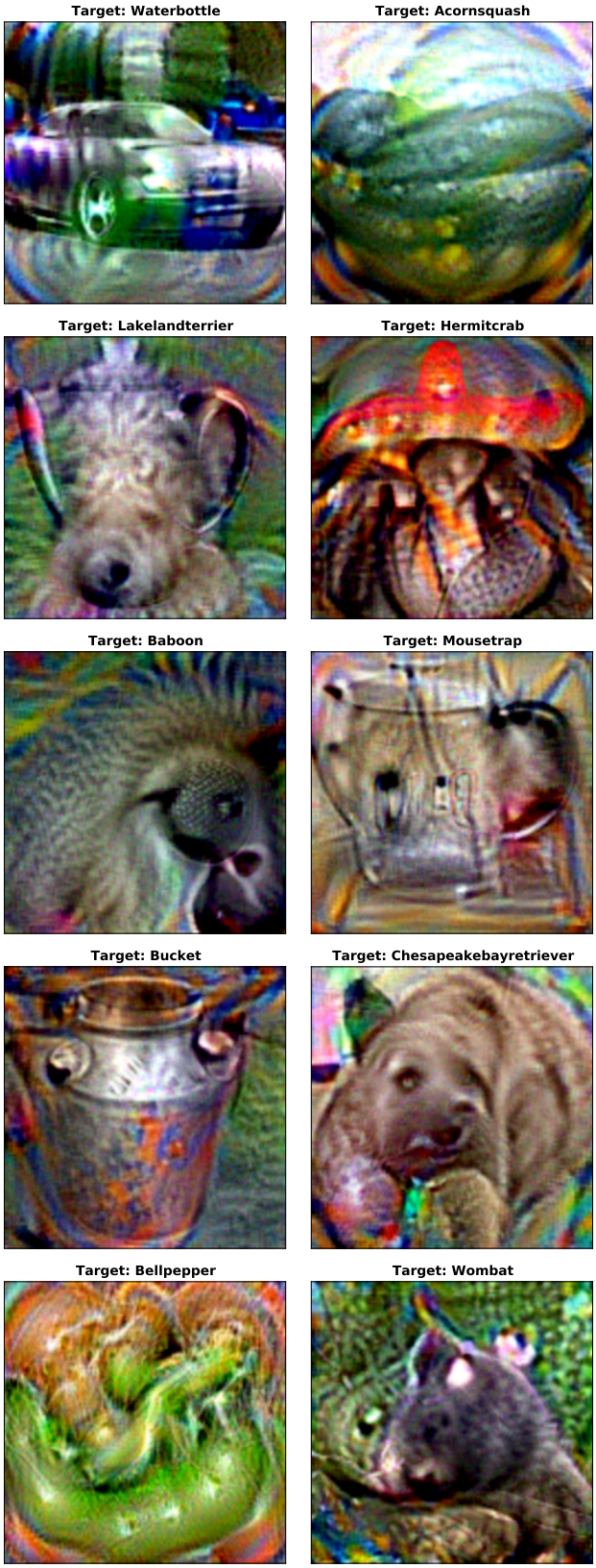

Figure 13: Printable examples of transformation-robust, physically-realizable pixel-space adversarial patches from Section 4.2. Patches can be resized before printing.