WHAT DO DEEP NETS LEARN? CLASS-WISE PATTERNS REVEALED IN THE INPUT SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) have been widely adopted in different applications to achieve state-of-the-art performance. However, they are often applied as a black box with limited understanding of *what the model has learned from the data*. In this paper, we focus on image classification and propose a method to visualize and understand the class-wise patterns learned by DNNs trained under three different settings including natural, backdoored and adversarial. Different from existing class-wise deep representation visualizations, our method searches for a single predictive pattern in the input (i.e. pixel) space for each class. Based on the proposed method, we show that DNNs trained on natural (clean) data learn abstract shapes along with some texture, and backdoored models learn a meaningless but highly predictive pattern for the backdoor target class. Interestingly, the existence of class-wise predictive patterns in the input space indicates that even DNNs trained on clean data can have backdoors, and the class-wise patterns identified by our method can be readily applied to "backdoor" attack the model. In the adversarial setting, we show that adversarially trained models learn simplified shape patterns and intentionally-perturbed "robust" datasets increase model robustness by sacrificing its ability to represent shapes. Our method can serve as an effective tool for both industrial and research communities to better understand DNNs trained on different datasets under different settings.

1 INTRODUCTION

Deep neural networks (DNNs) are a family of powerful models that have demonstrated superior learning capabilities in a wide range of applications such as image classification, object detection and natural language processing. However, DNNs are often applied as a black box with limited understanding of *what the model has learned from the data*. Existing understandings about DNNs have mostly been developed in the deep representation space or using the attention map. DNNs are known to be able to learn high quality representations (Donahue et al., 2014), and the representations are well associated with the attention map of the model on the inputs (Zhou et al., 2016; Selvaraju et al., 2016). It has also been found that DNNs trained on high resolution images like ImageNet are biased towards texture (Geirhos et al., 2019). While these works have significantly contributed to the understanding of DNNs, a method that can intuitively visualize what DNNs learn for *each class* in the *input space* (rather than the deep representation space) is still missing.

Recently, the above understandings have been challenged by the vulnerabilities of DNNs to backdoor (Szegedy et al., 2014; Goodfellow et al., 2015) and adversarial attacks (Gu et al., 2017; Chen et al., 2017). The vulnerabilities against Backdoors are believed to be caused by the preference of learning high frequency patterns (Chen et al., 2017; Liu et al., 2020; Wang et al., 2020). Nevertheless, no existing method is able to reliably reveal the backdoor patterns, even though it has been well learned by the backdoored model. One recent explanation for the adversarial vulnerability is that, besides robust features, DNNs also learn useful (to the prediction) yet non-robust features which are sensitive to small perturbations (Ilyas et al., 2019). Adversarial training, one state-of-the-art adversarial defense method, has shown to train DNNs to learn sample-wise robust features (Madry et al., 2018; Ilyas et al., 2019). However, it is still not clear if adversarially trained DNNs can learn a robust pattern for each class.

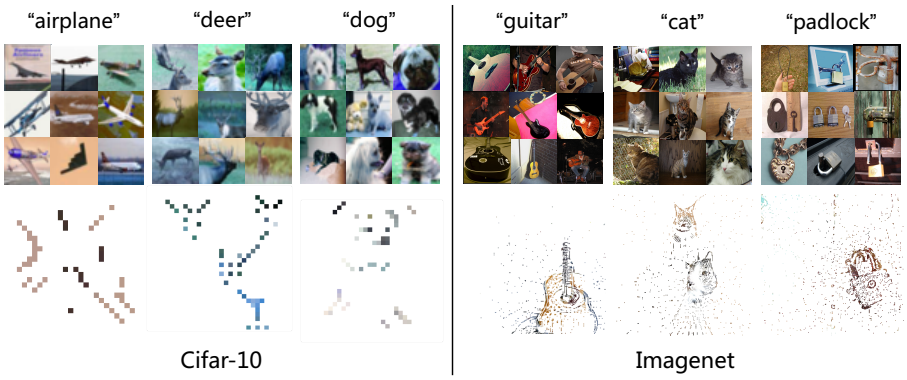

Figure 1: Example images (top row) and the class-wise patterns (bottom row) learned by a ResNet18 on CIFAR-10 (left three columns) and ResNet50 on ImageNet (right three columns) and revealed by our method. The 3 classes in ImageNet are "n02676566" ("guitar"), "n02123045"("cat"), and "n03874599" ("padlock"). The pattern size is 5% of the input size.

In this paper, we focus on image classification tasks and propose a visualization method that can reveal the patterns learned by DNNs for *each class* in the *input space*. Different from sample-wise explanation methods like attention maps, we aim to reveal the knowledge (or pattern) learned by DNNs for each class. Moreover, we reveal these patterns in the input space rather than the deep representation space. On the one hand, input space patterns are arguably easier to interpret. On the other hand, we are interested in a visualization method that can also provide new insights about the backdoor and adversarial vulnerabilities of DNNs, and such vulnerabilities are in the input space (eg. backdoored training examples and adversarial examples).

In particular, given a target class, a canvas image, and a subset of images from the nontarget classes, our method searches for a pattern (a set of pixels) from the canvas image that is highly predictive of the target class. In other words, when the pattern is attached to images from any other (i.e. nontarget) classes, the model will consistently predict them as the target class. Figure 1 illustrates a few examples of the class-wise patterns revealed by our method for DNNs trained on natural CIFAR-10 (Krizhevsky, 2009) and ImageNet (Deng et al., 2009) datasets.

In summary, our main contributions are: 1) We propose a visualization method that reveals the class-wise patterns learned by DNNs in the input space. 2) With the proposed visualization method, we show that DNNs trained on natural datasets can learn a consistent and predictive pattern for each class, and the pattern contains abstract shapes along with some texture. This sheds new lights on the current texture bias understanding of DNNs. 3) When applied on backdoored DNNs, our method can reveal the trigger patterns learned by the model from the poisoned dataset. Our method can serve as an effective tool to assist the detection of backdoored models. 4) The existence of class-wise predictive patterns in the input space indicates that even DNNs trained on clean data can have backdoors, and the class-wise patterns identified by our method can be readily applied to "backdoor" attack the model. 5) By comparing the patterns learned by DNNs trained on two intentionally-perturbed robust as well as adversarially trained DNNs, we find that adversarially trained models learn simplified shape patterns, while an intentionally-perturbed robust dataset improves model robustness by sacrificing its ability to represent shapes.

2 Related Work

General Understandings of DNNs DNNs are known to learn more complex and higher quality representations than traditional models. Features learned at intermediate layers of AlexNet have been found to contain both simple patterns like lines and corners and high level shapes (Donahue et al., 2014). These features have been found crucial for the superior performance of DNNs (He et al., 2015). The exceptional representation learning capability of DNNs has also been found related to structures of the networks like depth and width (Safran & Shamir, 2017; Telgarsky, 2016). One recent work found that ImageNet-trained DNNs are biased towards texture features (Geirhos et al.,

2019). Attention maps have also been used to develop better understandings of the decisions made by DNNs on a given input (Simonyan et al., 2014; Springenberg et al., 2015; Zeiler & Fergus, 2014; Gan et al., 2015). The Grad-CAM technique proposed by Selvaraju et al. (2016) utilizes input gradients to produce intuitive attention maps. Whilst these works mostly focus on deep representations or sample-wise attention, an understanding and visualization of what DNNs learn for each class in the input space is still missing from the current literature.

Understanding Vulnerabilities of DNNs Recent works have found that DNNs are vulnerable to backdoor and adversarial attacks. A backdoor attack implants a backdoor trigger into a victim model by injecting the trigger into a small proportion of training data (Gu et al., 2017; Liu et al., 2018). The model trained on poisoned dataset will learn a strong correlation between the trigger and a target label. A backdoored model behaves normally on clean test data, yet consistently predict a target (incorrect) label whenever the trigger appears in a test example (Zhao et al., 2020; Yao et al., 2019; Liu et al., 2020). This is believed to be caused by the fact that DNNs tend to learn more high frequency (eg. backdoor) patterns (Chen et al., 2017; Liu et al., 2020; Wang et al., 2020). However, it is still unclear whether DNNs can learn such patterns from natural (clean) data. Moreover, despite a few attempts (Wang et al., 2019; Qiao et al., 2019), the trigger pattern still can not be reliably revealed, even though it has been well learned by the backdoored model. DNNs can also be easily fooled by small, imperceptible adversarial noise into making incorrect predictions (Szegedy et al., 2014; Goodfellow et al., 2016). This has been found to be caused by learning useful (to prediction) but nonrobust (to adversarial perturbation) features (Ilyas et al., 2019). Meanwhile, adversarial training has been shown to learn more robust features and deliver effective defenses (Madry et al., 2018). However, existing understandings of adversarial training are established based on sample-wise attention (Ilyas et al., 2019). It still unclear, from the class-wise perspective, what robust or nonrobust input patterns look like. In this paper, we will propose a method to reveal the patterns (e.g. backdoor or adversarially robust/nonrobust) learned by DNNs for each class.

3 Proposed Visualization Method

In this section, we first define the input space class-wise pattern searching problem, then introduce our proposed searching method.

Motivation and Intuition. We focus on image classification with deep neural networks. Given a DNN model f trained on a K-class image dataset and a target class $y \in \{1, \cdots, K\}$, our goal is to find an input space pattern, i.e, a small set of pixels, that are extremely predictive of the target class. A highly predictive pattern of a class can largely capture the knowledge the model learned for the class. Intuitively, a predictive pattern of a target class should be able to make the model consistently predict the target class whenever it is attached to images from any other (eg. nontarget) classes.

Class-wise Pattern Searching. For a target class y, our method searches for a predictive pattern $\boldsymbol{p}_y$ from a canvas image $\boldsymbol{x}_c$, based on an additional small subset $\boldsymbol{X}_s$ of images from the nontarget classes. The canvas image is the image where the pattern (a set of pixels) is extracted. The search is done via an optimization process based on a mixed input between the canvas image $\boldsymbol{x}_c$ and an image $\boldsymbol{x}_s \in \boldsymbol{X}_s$. The mixed input $\tilde{\boldsymbol{x}}$ is defined as follows:

$$\tilde{\boldsymbol{x}} = \boldsymbol{m} * \boldsymbol{x}_c + (1 - \boldsymbol{m}) * \boldsymbol{x}_s, \tag{1}$$

where $\boldsymbol{m}$ is a mask that has the same size as either $\boldsymbol{x}_c$ or $\boldsymbol{x}_s$, and $\boldsymbol{m}_{ij} \geq 0$. The mixed input image is labeled as the target class regardless of its original class. This mixing strategy is reminiscent of the mixup (Zhang et al., 2018) data augmentation algorithm. However, we do not mix the class labels and our purpose is for pattern optimization rather than data augmentation.

During the searching process, the mask is iteratively updated to minimize the following loss:

$$\mathcal{L} = -\log f_y(\tilde{\boldsymbol{x}}) + \alpha \frac{1}{n} \left\| \boldsymbol{m} \right\|_1, \tag{2}$$

where, f_y is network's probability output with respect to target class y, $\|\cdot\|_1$ is the L_1 norm, α is a parameter that balances the two loss terms, and n is the size of the input image as well as the mask. The first loss term is the commonly used cross entropy loss. The second term increases the sparsity of mask as we are interested in simple patterns with a small number of highly predictive pixels.

During the search process, we pair the canvas image $\boldsymbol{x}_c$ randomly with images from $\boldsymbol{X}_s$, and iteratively update the mask $\boldsymbol{m}$ using standard Stochastic Gradient Decent (SGD) while keeping the model parameters fixed. At each iteration, the mask $\boldsymbol{m}$ will also be clipped into $[0, 1]$. Once a mask is learned, we further clip the values in the mask that are smaller than γ to zero, larger than γ to one. We denote this clipped mask by $\boldsymbol{m}_\gamma$. We then extract the pattern from the canvas image by $\boldsymbol{p}_y = \boldsymbol{m}_\gamma * \boldsymbol{x}_c$. The γ parameter can be flexibly determined in different applications. A large γ may lead to less predictive pattern while a small γ will produce more of a sample-wise pattern that overfits to the canvas image.

The above search method is repeatedly applied to N canvas images to generate N patterns for each class. We then select the pattern that has the lowest loss value as the final pattern of the class. This additional step is to find the most predictive pattern by exploring different canvases. The complete procedure of our method is described in Algorithm 1 in Appendix A.

Canvas Sampling. We propose four different sampling strategies for the selection of the N canvas images: positive sampling, negative sampling, random sampling and white canvas.

Positive sampling selects the top-N confident images from the target class according to the logits output of model f. Negative sampling selects the top-N most confidently misclassified images from any nontarget class into the target class. The random sampling randomly chooses N images from the target class y. The white canvas simply uses an image with all white pixels as the canvas.

Both positive and the negative sampling aim to find the most well-learned examples by the model, but from different perspectives: well-learned correctly (eg. positive) vs. well-learned incorrectly (eg. negative). The white canvas is interesting since the pattern found from the white canvas will have the texture "removed", which is useful for scenarios where only the shape features are of interest. The patterns found based on different canvases are compared in Figure 4. After applying our method on each class, we can obtain a set of class-wise patterns: $\mathcal{P} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_K\}$. This set of predictive patterns can revel the knowledge learned by model f for each class from a unique perspective.

Why is it Class-wise? At first sight, one might wonder if the discovered pattern could be sample-wise, rather than class-wise, given the use of the canvas sample. Note that, however, even though we are using a single sample as a canvas, the pattern found by the optimization algorithm is dependent on how the model has learnt the entire class, in terms of its loss. This is particularly evident in the case of the all white canvas, which bears no relation to any input sample. Hence our designation of the pattern as being "class-wise". While our method can find consistent and predictive class-wise patterns in the experiments, it might still be extendable. For example, using multiple positive canvas images at the same time, using noise rather than the non-target images, or a paired search with nontarget-class images could be replaced by a class-wise universal adversarial perturbation (Moosavi-Dezfooli et al., 2017) towards the target class on only the white canvas. We leave these explorations for future investigation.

4 Experiments

In this section, we will apply our method to reveal the class-wise patterns learned by DNNs trained on natural data, backdoored data, adversarial settings with intentionally-perturbed robust or nonrobust datasets, and adversarially trained DNNs. We also conduct comparisons with attention maps.

Experimental Setting We consider ResNet50 (He et al., 2016) on two benchmark image datasets: CIFAR-10 (Krizhevsky, 2009) and ImageNet (Deng et al., 2009). The models are trained on the training set of the datasets using standard training strategies, except the adversarial experiments in Section 4.4. We then apply our method to search for the class-wise patterns based on canvas images and nontarget-class images selected from the test set of the datasets. Here, we set the nontarget-class subset to 20% of those in the test set (more analysis is in Appendix C). The sparsity regularization parameter α is slightly adjusted around 0.2 for different classes and datasets. The clipping parameter γ is set to discover a pattern of specified size (eg. 5% or 10% of the input size), which will be explicitly stated in each experiment. The searching process is run for 5 epochs with respect to the nontarget class images. All experiments are run on GeForce GTX 1080Ti GPUs.

Predictive Power We define the predictive power of the pattern $\boldsymbol{p}_y$ for target class y as follows: $PW = \frac{ACC(f(\boldsymbol{x}_n + \boldsymbol{p}_y), y)}{ACC(f(\boldsymbol{x}), y)}$, where $ACC(f(\boldsymbol{x}_n + \boldsymbol{p}_y), y)$ is the model's accuracy on nontarget images

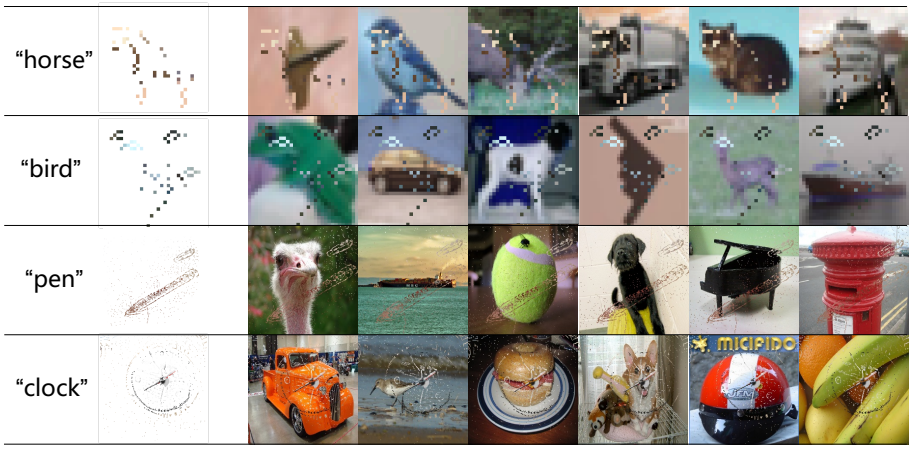

Figure 2: Class-wise patterns (second column) learned by a ResNet50 on natural CIFAR-10 (top two rows) and ImageNet (bottom two rows). Class names are shown in the first column, while nontarget-class images with the pattern attached and also being misclassified as the target class are shown in columns 3-8. The 2 classes in ImageNet are "n02783161"("pen"), "n02708093"("clock"). Pattern size is 5% of the input size.

$\boldsymbol{x}_n$ when attached with the target-class pattern $\boldsymbol{p}_y$ (via the addition operation) with respect to label y, and $ACC(f(\boldsymbol{x}), y)$ is the model's original accuracy on the target class images. Intuitively, the PW metric reflects the degree to which the pattern can determine the model's prediction towards the target class, relative to true target-class images.

4.1 Class-wise Patterns Learned by DNNs from Natural Data

We first show the class-wise patterns learned by ResNet50 on natural CIFAR-10 and ImageNet datasets. Here, we use positive canvases sampled using the positive sampling (see Section 3). The patterns revealed at 5% input size for two CIFAR-10 and ImageNet classes are illustrated in Figure 2. The predictive power of different sizes (0% - 10%) of patterns are shown in Figure 3.

Patterns Revealed by Positive Canvas As shown in Figure 2, the class-wise pattern learned from natural data contains shapes that are well associated with the object class, although some shapes are rather abstract especially for CIFAR-10 classes. Interestingly, the model can learn more than one shape for a given class, for example the "cat" class in Figure 1 and "pen" class in Figure 2. And the shapes are not necessarily at the center of the input (eg. "lock" in Figure 1). We suspect this is a result of the data augmentations (eg. random rotation, crop and shift) used in model training. According to the predictive power shown in Figure 3 at pattern size 5%, these patterns demonstrate roughly 60% (on average) of the predictive information learned by the model. These patterns indicate that DNNs can learn to capture some common shapes of the class, even though the training images may quite different from each other.

The abstract shapes are not the whole story here as there are also textures in the pattern. Shapes are more class-wise features as our method iteratively searches for the optimal positions of the pixels that are predictive to the entire class. By contrast, the textures are more sample-wise features carried over from the canvas image by the $\boldsymbol{p}_y = \boldsymbol{m}_\gamma * \boldsymbol{x}_c$ operation. The use of a white-canvas can remove sample-wise textures while keeping only class-wise shapes (see Figure 4). In relation to existing texture bias understanding of DNNs trained on ImageNet (Geirhos et al., 2019), our result implies that shape also plays an important role in DNN learning and prediction. Note that, besides shape and texture, DNNs also learn a certain amount of background noise, especially on ImageNet.

Patterns Revealed by Different Canvases The patterns found by using different types of canvases are illustrated in Figure 4. As can be observed, the abstract shapes found on different canvases are quite consistent, even on the white canvas. An interesting observation on the "hook" class (right subfigure) is that there are more than one hook, and they appear in different locations. This indicates

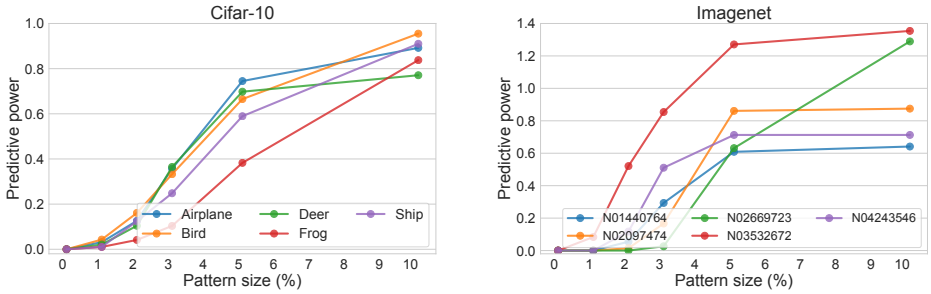

Figure 3: The predictive power of the class-wise patterns of different sizes, for 5 randomly selected CIFAR-10 (left) and ImageNet (right) classes respectively. The 5 classes and their original accuracies are: "airplane" (95.3%), "bird" (93.2%), "deer" (96.0%), "frog" (96.7%), "ship" (96.7%) for CIFAR-10, and "n01440764" (92%), "n02097474" (72%), "n02669723" (38%), "n03532672" (48%), "n04243546" (94%) for ImageNet.

that, for the same class, DNNs learn repetitive shapes at different locations of the input space. Note that the patterns identified on the white canvas contain no texture information.

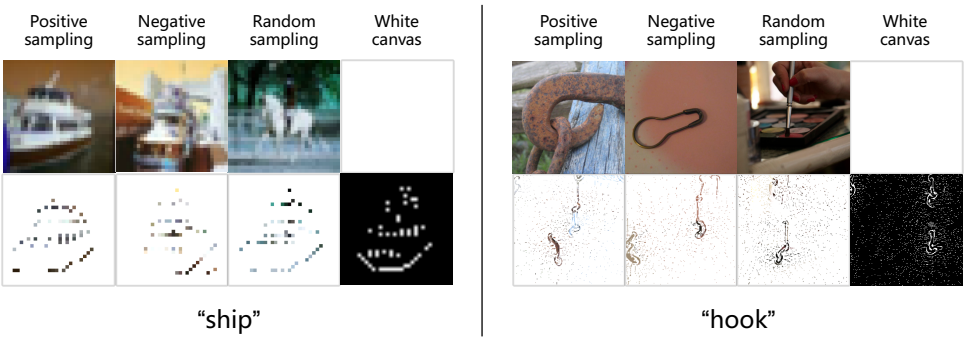

Figure 4: Patterns (second row) found by our method on different types of canvases (the first row) for CIFAR-10 "ship" (left) class and ImageNet "hook"("n03532672") (right) class.

4.2 Comparison to Attention Maps

Attention maps have been widely used as an intuitive tool to explain the decisions made by DNNs. Here, we compare the predictive power of the class-wise patterns found by our methods to the key areas found by the attention map. This is somewhat not a fair comparison since attention maps were designed for sample-wise explanations. However, we believe it is interesting to see if high attention areas are also predictive of the class. For each of the 5 ImageNet classes, we select the most confident test image and extract the highest attention area (eg. 10%) of the image, using Grad-CAM (Selvaraju et al., 2016). We then attach this pattern to other classes to evaluate its predictive power. The predictive power and a few examples are shown in Figure 5. We find that the predictive power of the attention patterns are extremely low. This indicates that sample-wise attention maps may not be a good representation of the class-wise knowledge learned by the network. As the attention visualization shown in Appendix B, the patterns identified by our method are indeed class-universal patterns and can consistently shift the network's attention towards the patterns when they are attached to different images.

4.3 Class-wise Patterns Learned by Backdoored DNNs

Here, we apply our method to reveal the patterns learned by a backdoored ResNet50 model by BadNets (Gu et al., 2017) on CIFAR-10. The patterns revealed on different canvases for the backdoored class "airplane" are illustrated in Figure 6. Here, we set the pattern size to be small (eg. 1%) as here we are interested in examining whether the model has learned a small, extremely predictive but meaningless pattern. As can be observed, our method can reliably reveal the trigger pattern,

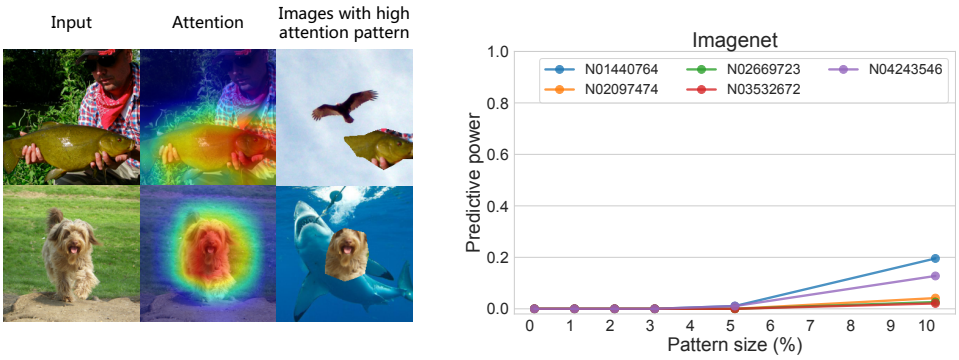

Figure 5: Left grid: input image (first column), attention map (second column) and images (third column) attached with high attention patterns extracted from the input image. Right plot: the average predictive power of high attention patterns for 5 ImageNet classes on the test set of other classes. The attention pattern size is 10% of the input size.

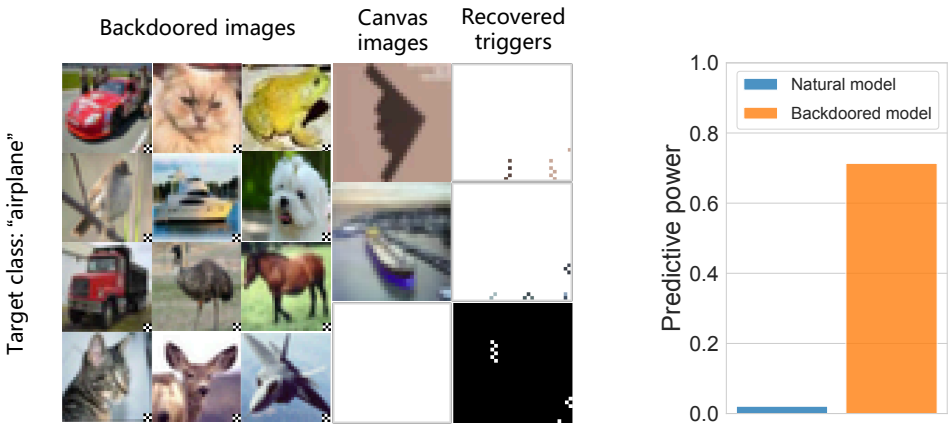

Figure 6: Backdoor trigger patterns revealed by our method from a backdoored ResNet50 model using different canvas images. The backdoor attack is the BadNets (Gu et al., 2017), the target class is "airplane" of CIFAR-10, and the trigger pattern is a black-white checkerboard at the bottom right corner of the image. The backdoor pattern was injected into 10% training data, and the attack success rate is 99.94%. The bar plot on the right shows the average predictive power of the identified pattern on natural model vs backdoored model. Pattern size is 1% image size.

especially on the white canvas. We then test the average predictive power of the three patterns on natural versus the backdoored model. As shown in the right figure, the patterns are highly predictive on the backdoored model, yet are meaningless on the natural model. We also tested the same size of pattern identified from the natural model, and find that such a small pattern is also not predictive on the natural model. The existence of small but extremely predictive patterns can be used to assist the detection of backdoored models. An interesting observation is that, although the trigger pattern was injected into a fixed location of the training images, the model learns the pattern repetitively at different locations. We suspect this is also caused by data augmentation. This also implies that the trigger pattern does not need to be fixed to a specific location for backdoor attack to succeed. This result not only provides a positive answer to the above question, but also confirms that our method can indeed reveal the pattern learned by DNNs.

Revisiting the patterns revealed in Section 4.3 on natural data, we find that DNNs trained on natural (clean) data can also have backdoors, since those class-wise patterns can be readily applied to "backdoor" attack the model. The images shown in columns 3-8 in Figure 2 are the backdoor attack images. The predictive powers shown in Figure 3 indicates that the attack success rate will be high, although not as high as the state-of-the-art (Chen et al., 2017; Liu et al., 2020) and stealthiness is not guaranteed. Interestingly, these results indicate that DNNs appear to memorize a fixed pattern

for each class, which explains why backdoor attacks can easily fool a DNN model to remember the trigger pattern.

4.4 Class-wise Patterns learned by DNNs in Adversarial Settings

Here we apply our method to identify the class-wise patterns learned by ResNet50 on three different versions of CIFAR-10 dataset: natural ($\mathcal{D}$), robust ($\mathcal{D}_R$) and nonrobust ($\mathcal{D}_{NR}$). The $\mathcal{D}_R$ and $\mathcal{D}_{NR}$ versions were generated by previous work (Ilyas et al., 2019) by perturbing the training images of the original dataset $\mathcal{D}$ to have only robust or nonrobust features in the adversarial setting. In other words, $\mathcal{D}_R$ contains only the robust features learned by a robust (adversarially trained) model, while $\mathcal{D}_{NR}$ contains only the nonrobust features learned by a nonrobust model. It was shown that DNNs exhibit moderate adversarial robustness when trained on $\mathcal{D}_R$ using even standard training (Ilyas et al., 2019). We train the ResNet50 model independently using standard training on $\mathcal{D}$, $\mathcal{D}_{NR}$ and $\mathcal{D}_R$, and using adversarial training on $\mathcal{D}$.

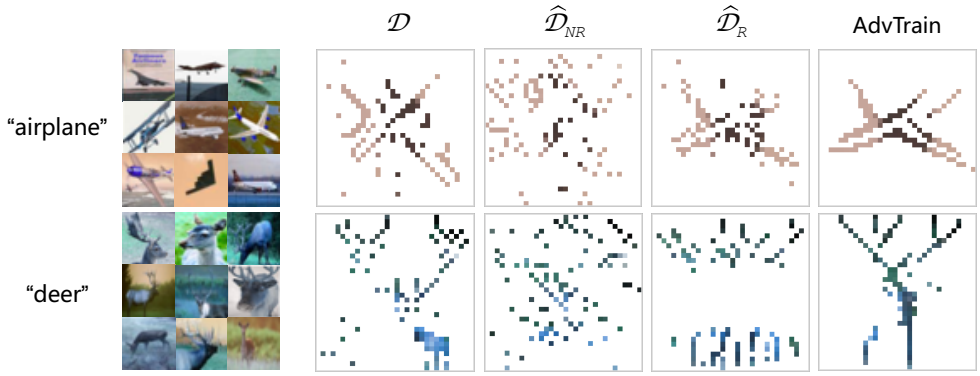

Figure 7: Examples of class-wise patterns learned by ResNet50 and revealed by our method on natural ($\mathcal{D}$), nonrobust ($\mathcal{D}_{NR}$) and robust ($\mathcal{D}_R$) data on CIFAR-10, as well as adversarially trained model (AdvTrain) on CIFAR-10. Pattern size is 10% of the input size.

The class-wise patterns learned under different settings are illustrated in Figure 7. Compared to $\mathcal{D}$, the patterns learned on $\mathcal{D}_{NR}$ are mostly background noise. However, patterns learned by the adversarially trained model contains clear shapes and almost no background noise. Compared to that learned on $\mathcal{D}$, the shapes learned by adversarially trained DNNs are more simplified and have more connected regions. This provides an interesting visualization of the appearance of robust class-wise patterns learned by adversarially trained DNNs in the input space. The robust dataset $\mathcal{D}_R$ is interesting as it is a dataset that is intentionally perturbed to be robust. The patterns learned on $\mathcal{D}_R$ indicate that the "robust" perturbation can remove much of the background noise (eg. the "airplane" class). However, it may also destroy the shapes of the object class (eg. the "deer" class). This indicates that intentionally-perturbed dataset improves model robustness by sacrificing its ability to represent shapes, possibly reducing the model's generalization performance in real-world scenarios.

5 Conclusion

In this paper, we propose a method to identify the *class-wise* patterns learned by DNNs and reveal the patterns in the *input space*. The patterns revealed by our method indicate that DNNs can learn abstract but consistent shape patterns that are associated with the target class. By further examining the patterns learned by backdoored DNNs, we find that our method can reveal the trigger pattern, and more interestingly, DNNs trained on clean data may also have "backdoors". Patterns revealed in adversarial settings indicate that adversarially trained DNNs can indeed learn more robust shape patterns, however, internationally-perturbed robust datasets can lead to highly distorted shapes. Our method is effective for communities to better understand DNN models trained on different datasets under different settings.

REFERENCES

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2014.

Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alexander G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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. In *International Conference on Learning Representations (ICLR)*, 2019.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (ICCV)*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium (NDSS)*, 2018.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. *European Conference on Computer Vision (ECCV)*, 2020.

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 (ICLR)*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Ximing Qiao, Yukun Yang, and Hai Li. Defending neural backdoors via generative distribution modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Itay Safran and Ohad Shamir. Depth-width tradeoffs in approximating natural functions with neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations (ICLR)*, 2014.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations (ICLR)*, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*, 2014.

Matus Telgarsky. benefits of depth in neural networks. In *Conference on Learning Theory (COLT)*, 2016.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Symposium on Security and Privacy (SP)*, 2019.

Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Latent backdoor attacks on deep neural networks. In *Conference on Computer and Communications Security (CCS)*, 2019.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

A INPUT SPACE CLASS-WISE PATTERN SEARCHING ALGORITHM

The proposed input space class-wise pattern searching method is described in Algorithm 1. We use the test set of the dataset to sample the canvas images $\boldsymbol{x}_c$ and the nontarget-class subset $\boldsymbol{X}_s$. The InitializeMergeMap function in line 1 initializes the mask $\boldsymbol{m}$ to the matrix with all 0.5 values. The loss term balancing parameter α and the clipping threshold γ can be flexibly determined and tuned in different application scenarios.

Algorithm 1 Input Space Class-wise Pattern (ISCP) Searching Method

Require:
A DNN model $\boldsymbol{f}$, canvas image $\boldsymbol{x}_c$, nontarget-classes subset $\boldsymbol{X}_s$, target class y, regularization balancing parameter α, clipping threshold γ, input size n, epochs N, batch size B, step size s.

Ensure: Target pattern $\boldsymbol{p}$

1: $\boldsymbol{m}$ = InitializeMergeMap()
2: **for** i in $\mathrm{range}(N)$ **do**
3: **for** $\{\boldsymbol{x}_s^{(j)}\}_{j=1}^{B}$ in $\boldsymbol{X}_s$ **do**
4: $\tilde{\boldsymbol{x}}^{(j)} = \boldsymbol{m} * \boldsymbol{x}_c + (1 - \boldsymbol{m}) * \boldsymbol{x}_s^{(j)}$
5: $\boldsymbol{L} = \sum_{j=1}^{B}\{-\log \boldsymbol{f}_y(\tilde{\boldsymbol{x}}^{(j)}) + \alpha \frac{1}{n} \|\boldsymbol{m}\|_1\}$
6: $\boldsymbol{\delta} = \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{m}}$
7: $\boldsymbol{m} = \boldsymbol{m} - s * \mathrm{sign}(\boldsymbol{\delta})$
8: Constraint all the values in $\boldsymbol{m}$ to [0, 1]
9: **end for**
10: **end for**
11: Clip the values in $\boldsymbol{m}$ that are smaller than γ to 0, larger than γ to 1, and get $\boldsymbol{m}_\gamma$
12: $\boldsymbol{p} = \boldsymbol{m}_\gamma * \boldsymbol{x}_c$
13: return $\boldsymbol{p}$

B ATTENTION SHIFT WHEN A CLASS-WISE PATTERN IS ATTACHED

We use the attention map to check the network's attention shift when the class-wise pattern found by out method is attached to an image. As shown in Figure 7, we find that class-wise pattern has very strong attention around the areas that contain clear shapes, and the attention of the network is significantly shifted towards those areas when the pattern is attached to different images. This confirms that the patterns identified by our method are indeed universal patterns that carry consistent information.

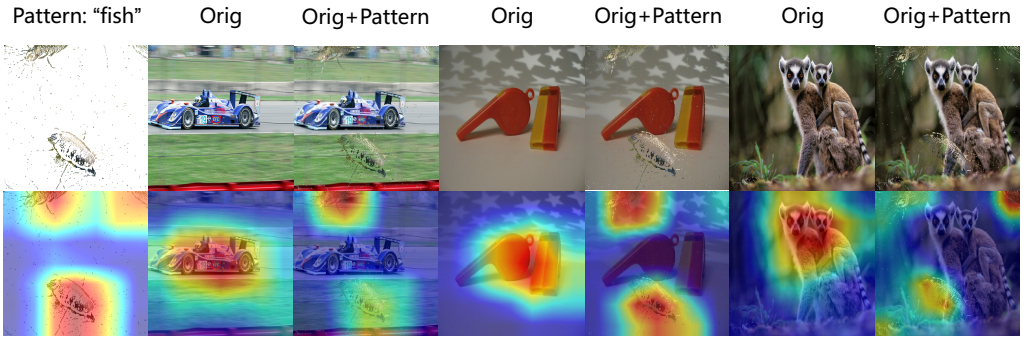

Figure 8: The attention (visualized by Grad-CAM) shift of a ResNet50 model (on ImageNet) when our class-wise pattern is attached to different images. The pattern is from "n01440764"("fish")

C PATTERNS REVEALED WITH DIFFERENT SIZES OF THE NONTARGET-CLASS SUBSET

Here, we show the patterns revealed by our method with different sizes of the nontarget-class subset selected from the original test set. We test subset sizes from 20% to 100% of the test set. This experiment is conducted on a ResNet50 model trained on natural CIFAR-10 dataset. The class-wise patterns and their predictive powers are illustrated in Figure 9. As the results show, the patterns are fairly consistent under different subset sizes (left figure), and 20% of the test set is sufficient for our method to find highly predictive patterns (right figure). Compared to the patterns found using only 20%, the patterns found with more data exhibit more local (rather than the global shape) variations. This is cause by the iterative adjustment of the pixel positions to explore predictive patterns for more non-target class images. This also causes the variation of the prediction power.

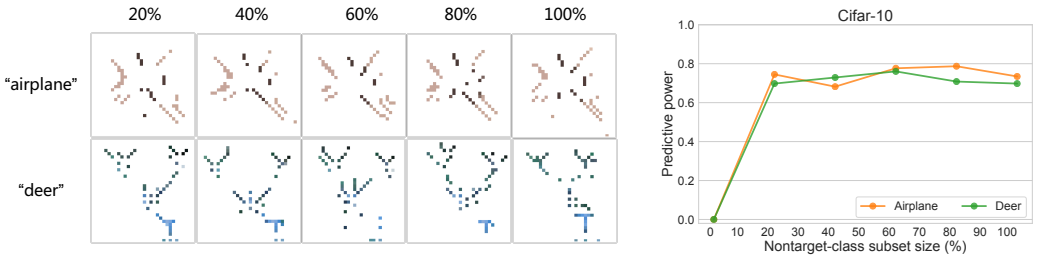

Figure 9: Left grid: Patterns found by our methods under different nontarget-class subset size (% to the original test set), for "airplane" and "deer" classes in CIFAR-10. Right plot: The predictive power of the class-wise patterns shown in the left grid.