An Adversarial Framework for Generating Unseen Images by Activation Maximization

Yang Zhang*[1], Wang Zhou*[2]†, Gaoyuan Zhang[1], David Cox[1], Shiyu Chang[3]

[1]MIT-IBM Watson AI Lab, Cambridge, MA, USA [2]Meta AI, New York, NY, USA
[3]University of California at Santa Barbara, USA
{yang.zhang2, gaoyuan.zhang, david.d.cox}@ibm.com, wangzhou@fb.com, chang87@ucsb.edu

Abstract

Activation maximization (AM) refers to the task of generating input examples that maximize the activation of a target class of a classifier, which can be used for class-conditional image generation and model interpretation. A popular class of AM method, GAN-based AM, introduces a GAN pre-trained on a large image set, and performs AM over its input random seed or style embeddings, so that the generated images are natural and adversarial attacks are prevented. Most of these methods would require the image set to contain some images of the target class to be visualized. Otherwise they tend to generate other seen class images that most maximizes the target class activation. In this paper, we aim to tackle the case where information about the target class is completely removed from the image set. This would ensure that the generated images truly reflect the target class information residing in the classifier, not the target class information in the image set, which contributes to a more faithful interpretation technique. To this end, we propose PROBEGAN, a GAN-based AM algorithm capable of generating image classes unseen in the image set. Rather than using a pre-trained GAN, PROBEGAN trains a new GAN with AM explicitly included in its training objective. PROBEGAN consists of a class-conditional generator, a seen-class discriminator, and an all-class unconditional discriminator. It can be shown that such a framework can generate images with the features of the unseen target class, while retaining the naturalness as depicted in the image set. Experiments have shown that PROBEGAN can generate unseen-class images with much higher quality than the baselines. We also explore using PROBEGAN as a model interpretation tool.[1]

1 Introduction

Activation maximization (AM) refers to the technique of generating input examples, such as images and audios, that maximize the activation of a classifier, so that the generated examples conform to the class characteristics as depicted by the classifier. There are two primary uses of AM. First, it can be used to perform class-conditional generation, *e.g.* [37]. Second and more importantly, it can be used as a global model interpretation technique that shows what features the classifier is utilizing to make classification decisions [4]. Although AM can be applied to different domains, this paper will primarily focus on the task of image generation.

To prevent AM from adversarially attacking the classifier, existing works would impose different regularization techniques to ensure that the generated images look like 'natural' images free of adversarial patterns [26, 45, 48, 38]. Recently, GAN has emerged as a popular regularization technique to enforce a distributional match between generated images and a given natural image set [24, 33, 34, 4]. In most cases, the image set is chosen to be general enough that it contains some images of whatever target class whose activation is being maximized.

In this paper, we would like to tackle a much more challenging setting, where images of the target class are completely removed from the image set, a constraint we call *the class absence constraint*. For example, when generating the class of 'truck', no truck image would be seen in the image set. The GAN has to completely resort to the information residing in the image classifier. The motivation for studying this problem is twofold. First, it constitutes a pioneering scientific exploration of how much information about a class can be elicited from a neural classifier, which we will refer to as the classifier's memory. Since the target class information in the generated images completely comes from the classifier, the more information the classifier memorizes, the greater details the generated images would contain, and thus the better these images would resemble the target class. The second motivation of studying this setting is that when AM is applied to interpreting neural classifiers, only by enforcing the class absence constraint can we ensure the generated images are faithfully reflecting the memory of the classifier, not 'stealing' the information from the image set.

So far, no algorithms can generate reasonable images if the target absence constraint is strictly enforced. For example, the most successful paradigm so far is to pre-train a GAN on the image set, and then perform activation maximization over the random seed or style embedding [24, 33, 34, 4]. However, the output space of the pre-trained GAN does not necessarily include the target class images. It is pointed out [4, 24] that if the target class is completely unseen, such framework would tend to use seen image classes to maximize the unseen class activation.

*Equal contribution.

†Work done while the author worked at IBM Research, USA.

[1] Our code is at https://github.com/csmiler/ProbeGAN/

Motivated by this, we propose PROBEGAN, a GAN-based activation maximization algorithm that can generate intelligible images for a class that is completely absent from the image set. Rather than using a pre-trained GAN, PROBEGAN trains a new GAN when the unseen target class is chosen, explicitly including the activation maximization into the training objective. PROBEGAN consists of three components, a class-conditional generator, which generates images for both seen and unseen classes, a seen-class conditional discriminator, which only sees the seen-class images, and an all-class unconditional discriminator, which sees all the fake images. It can be shown that such a framework can generate images with the features of the unseen target class, while retaining the naturalness as depicted in the image set.

The contribution of this paper is summarized as follows.

- PROBEGAN is among the first GAN-based activation maximization algorithms that can generate reasonable unseen class images.
- Experiments show that PROBEGAN can generate images with much higher quality than the baselines do, and reveal that neural classifiers can memorize much greater details than one may expect.
- We also explore the use of PROBEGAN as a model interpretation tool, which can convincingly and faithfully identify the features memorized by the classifier.

2 Related Work

Feature visualization Feature visualization has been broadly explored to uncover the information retained by a network. To visualize a feature, images are often synthesized by maximizing the activation (AM) of certain neurons/filters/layers of the classifier [11, 32]. However, this maximization approach can lead to unrecognizable outputs [35] or adversarial examples [46]. Heuristic regulations [26, 45, 48, 38] can be applied to improve the image quality. BigGAN-AM [24] and its variants [8, 34, 33] utilize a generator trained on 1000-class ImageNet [40] as a learned prior and can generate almost natural visualizations. However, it is ambiguous whether the features generated come from the model being visualized or from the prior. A robust classifier [44] helps to filter adversarial generations but it requires seed distribution computed from real data. The generation quality and diversity of such approaches are worse than GANs [20, 5].

Instance-based interpretation Given an input image, a saliency map created with gradient ascend [45] or deconvolution [50] can provide visual guidance on where the model focuses to make the decision. Visualization of activation of internal filters [39, 6, 50] also provides a view of the model's abstraction on the input. While instance-based interpretation may explain the classifier's decision on individual images, it cannot provide comprehensive knowledge of the classifier.

GANs GANs [14] specialize in generating undistinguishable data that follow the distribution of the training dataset. To mitigate the challenges of training instability and mode collapse [42, 1], different objective functions [27, 42, 3, 2, 43, 25, 36] and regulations [21, 15, 30, 51, 28] have been proposed. ProGAN [18] trains GANs to generate high-resolution output by incrementally adding layers to the network. StyleGAN [19, 20] maps the latent space to a feature space to separate high-level attributes and stochastic effects. Conditional GANs [29] feed additional class information to both the generator and discriminator. AC-GAN [37, 13] train auxiliary classifiers to classify the generated images, and encourage the generated images to maximize the classification accuracy. Class information can also be passed to batch normalization layers [7, 9, 52] or class embeddings [31]. BigGAN [5] generates photo-realistic images after properly scaling up the training of SAGAN [52].

3 The PROBEGAN Algorithm

In this section, we are going to introduce our problem setup as well as PROBEGAN. We will use upper-cased letters, $\boldsymbol{X}$ and X, to denote random vectors (bolded) and variables (non-bolded); lower-cased letters, $\boldsymbol{x}$ and x, to denote deterministic vectors (bolded) and scalars (non-bolded). $p_{Y|\boldsymbol{X}}(y|\boldsymbol{x})$ denotes the probability of $Y = y$ conditional on $\boldsymbol{X} = \boldsymbol{x}$. To concretize our explanation, we use an image classification task as an example, but the algorithm can be generalized to other domains.

3.1 The Problem Setup

Denote $\boldsymbol{X}$ as the random variable of images. Denote Y as the class label ranging from 0 to $C - 1$, where C is the total number of classes. The problem we are interested in can be formulated as follows. Suppose we have two pieces of information

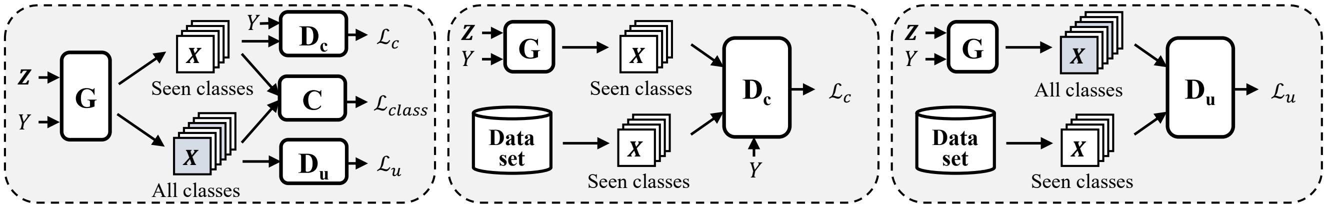
1. An image classifier, whose output is $\hat{p}_{Y|\boldsymbol{X}}(y|\boldsymbol{x})$;
2. An generic image set.

Our goal is to probe what features the classifier memorizes for a given target class, denoted as y^*. To do this, we propose the following steps:

1. Remove all the target-class images from the image set;
2. Design a GAN framework to generate images that:
   (a) maximize the classifier's activation of the target class, $\hat{p}_{Y|\boldsymbol{X}}(y|\boldsymbol{x})$;
   (b) conforms to the image set in terms of naturalness.

After the images are generated, we can then inspect if a certain feature we are interested in appears in the pictures. If the feature appears, it indicates that it is memorized by the classifier. We can also gauge the overall quality of the generated images. The higher the quality, the richer information the classifier can memorize for the target class. Below are some further explanations about the rationale behind this.

Why impose a naturalness requirement? By making the images natural, it makes it easier for humans to subjectively perceive what features are memorized by the classier. More importantly, without this regularization, the generated image can easily degenerate into adversarial samples [46], *i.e.* images that can be classified as the target class but do not look like the target class images at all.



(a) Training the generator (b) Training the conditional discriminator (c) Training the unconditional discriminator

Figure 1: PROBEGAN framework and the data flow when training different modules. G represents the generator; D_c represents the conditional discriminator; D_u represents the unconditional discriminator; C represents the classifier. Class dilution (mixing the target class into other classes) is performed on the input to the unconditional discriminator.

Why excluding the target-class images from the image set? As can be in from the steps above, we require that the image set is that it *must not* contain the target class images, a constraint referred to as the *target absence condition*. This is because the training of the image generation network is informed by two information sources, the classifier and the image set. If both sources contain information about the target class, then it would be unclear whether the generated images reflect the classifier's memory or simply the image set. Only by removing the target class from the image set can we ensure that every class-specific feature that appears in the generated images comes from the classifier's memory.

3.2 The Challenges of Target Absence Constraint

Although the target absence constraint is the key to probing the information from the classifier, it creates challenges of incorporating the image set information, particularly for GAN-based approaches. Since the images to be generated are naturally different from the real images in terms of class, the discriminator can easily tell fake from real by looking at the class differences, instead of other distributional discrepancies such as naturalness, which creates an incorrect gradient signal for the generator.

Formally, let us represent the image $\boldsymbol{X}$ as a concatenation of two feature vectors, $\boldsymbol{X} = f(\boldsymbol{F}_s, \boldsymbol{F}_c)$, where $\boldsymbol{F}_s$ is called *class-specific features*, whose distribution varies across different classes. $\boldsymbol{F}_c$ is called *common features*, whose distribution is the same across all the classes. Since $\boldsymbol{F}_c$ does not reflect any class information, it would not be captured or utilized by the image classifier. However, $\boldsymbol{F}_c$ is very important in distinguishing natural images from others, and that is why the image set is needed to inform the distribution of $\boldsymbol{F}_c$. Ideally we would want the classifier to guide the distribution of $\boldsymbol{F}_s$, and the image set to guide the distribution of $\boldsymbol{F}_c$.

However, in practice, the discriminator of any GAN-based algorithm is prone to focusing on the distributional discrepancies in $\boldsymbol{F}_s$, rather than $\boldsymbol{F}_c$, between the fake and real images, because the differences in the former are much more distinct due to the target absence constraint. As a result, the distribution of $\boldsymbol{F}_c$ is left uninformed.

3.3 PROBEGAN

The key to solving the problem is to make the discriminator less sensitive to the differences in $\boldsymbol{F}_s$, and more to the differences in $\boldsymbol{F}_c$. To this end, we propose PROBEGAN, which can achieve this purpose by mixing the fake images of the target class into the fake images of the seen classes, *i.e.* classes that are present in the image set, without supplying class labels, a step we call *class dilution*.

Modules As shown in Figure 1, PROBEGAN consists of three modules. The first module is the conditional generator, denoted as $G(\boldsymbol{Z}, Y)$, which, given a class label Y, generates images of that class from a random vector $\boldsymbol{Z}$. The second module is the conditional discriminator, denoted as $D_c(\boldsymbol{X}, Y)$, which discriminates fake images from real images given the class labels. The third module is the unconditional discriminator, denoted as $D_u(\boldsymbol{X})$, which discriminates fake images without access to the class labels. The purpose of the conditional discriminator is to guide the image generation of the seen class, and that of the unconditional discriminator is to guide the generation of the target class.

Objectives The objective of the generator is to fool the conditional discriminator with the generated seen class images, and to fool the unconditional discriminator with the generated images from *all the classes*, while maximizing the classifier's logit of the corresponding classes (as shown in Figure 1(a)); whereas the objective of the discriminators is to discriminate fake images from the real images in the image set (as shown in Figures 1(b) and (c)). Formally, the general objective can be written as follows

$$\min_{G(\cdot)} \max_{D_c(\cdot,\cdot), D_u(\cdot)} \sum_{y\in\mathcal{Y}\backslash y^*} \mathcal{L}_c(y) + \mathcal{L}_u + \lambda_g \sum_{y\in\mathcal{Y}} \mathcal{L}_{class}(y), \tag{1}$$

where $\mathcal{Y}$ denotes the indices of all the classes. $\mathcal{Y}\backslash y^*$ denotes all the classes but the target class; $\mathcal{L}_c(y)$ is the conditional adversarial loss for class y, which is the same hinge loss as in BigGAN:

$$\begin{aligned}\mathcal{L}_c(y) =& \mathbb{E}_{\boldsymbol{X}|Y=y}[\min(0, -1 + D_c(X, y))] \\ &- \mathbb{E}_{\boldsymbol{Z}}[\min(0, -1 - D_c(G(\boldsymbol{Z}, y), y))].\end{aligned} \tag{2}$$

$\mathcal{L}_u$ is the unconditional adversarial loss, which takes the form of W-GAN loss with gradient penalty [15]. The unconditional adversarial loss involves fake images of all the classes, versus real images of only the seen classes ($Y \neq y^*$), *i.e.*

$$\mathcal{L}_u = \mathbb{E}_{\boldsymbol{X}|Y\neq y^*}[D_u(X)] - \mathbb{E}_{\boldsymbol{Z},Y}[D_u(G(\boldsymbol{Z}, Y))] + \lambda_{gp}\mathcal{L}_{gp}, \tag{3}$$

where $\mathcal{L}_{gp}$ is the gradient penalty loss defined in [15]. Finally, $\mathcal{L}_{class}(y)$ is the classification loss for class y, *i.e.*

$$\mathcal{L}_{class}(y) = -\mathbb{E}_{\boldsymbol{Z}}[\log \hat{p}_{Y|\boldsymbol{X}}(y|G(\boldsymbol{Z}, y))]. \tag{4}$$

Class Dilution It is worth emphasizing that although the primary purpose of the unconditional discriminator is to guide the generation of target class images, the fake images inputted to the unconditional discriminator involves not only the target class, but also the seen classes, an operation we refer to as class dilution. As can be seen in Figure 1(c), there exists a class difference in the real and fake images inputted to the unconditional discriminator, and the difference would be even more noticeable if the fake images only involve the target class. To understand the intuition behind class dilution, consider a simple example where there are four classes, $\{A, B, C, D\}$, with D being the target class. If fake images are from D but the real images are from $\{A, B, C\}$, the fake images are easily identifiable by the class difference. On the other hand, if the fake images are from $\{A, B, C, D\}$ but the real images are from $\{A, B, C\}$, that would greatly dilute the class differences.

4 Experiments

In this section, we will present two sets of results. The first set (Section 4.2) demonstrates the quality of the generated images and how much information PROBEGAN can recover from the classifiers compared with the baselines. The second set of results (Sections 4.3-4.6) shows how PROBEGAN can be used to interpret image classifiers.

4.1 Configurations

Datasets To evaluate the performance of our approach, we conduct experiments on two image datasets, CIFAR-10 [22] and Waterbird dataset [41], and an audio dataset [12, 47]. We randomly select one class as the unseen class.

Baselines The following two baselines are implemented.

- BIGGAN-AM [24]: synthesizing images from a classifier by using a pre-trained GAN network from ImageNet as a strong prior and searching for embeddings that can be mapped to the target class.
- NAIVE: PROBEGAN without the conditional discriminator and class dilution. There is only the unconditional discriminator distinguishing fake images of target class from real images of seen classes. It is expected to suffer from the challenges mentioned in Section 3.2.

Each algorithm will be trained with a regular classifier and a robust classifier.

Evaluation metrics Class-wise Fréchet Inception Distance (FID) [16], *i.e.*, the intra-FID score [31], is used for quantitative evaluation on image classifiers. FID score calculates the Wasserstein-2 distance of the feature vectors of an Inception-v3 network between the generated and real images, and the lower FID score indicates the more similar the two image sets are. Sample images are included to qualitatively illustrate the performance. In addition, we employ Amazon Mechanical Turk (MTurk) to categorize the generated samples, and report the percentage of correctly recognized samples of the new class. Higher recognition rates indicate better resemblance to the target class thus better interpretation. Further details can be found in Appendix A.3.

Implementation The architecture of PROBEGAN mostly follows BigGAN [5], although other architectures can also work with minor modifications. The generator and conditional discriminator are exactly the same as BigGAN. The unconditional discriminator is added by creating a linear layer branch before the last layer of the conditional discriminator. In other words, the two discriminators have shared parameters in all but the last layer. Since only the last layer of the BigGAN discriminator involves class information, such parameter sharing will not introduce class information to the unconditional discriminator. The parameter sharing can improve training stability. Otherwise, it would hard to synchronize the convergence rate of the three modules. For NAIVE approach, we remove the class information from the generator by feeding a constant label and the discriminator by removing the class-conditional branch while the classifier remains. Results of BIGGAN-AM is produced with the code[2] provided by the author. Our Pytorch implementation will be made publicly available. More experiment details can be found in Appendix A.

4.2 Main Results

Table 1 and Figure 2 present the FID results and some example generated images on CIFAR-10, respectively. For each case, we select one of the classes as the target class, and the remaining nine classes are seen classes. The results for the target classes "horse" and "truck" are reported, and the rest of the classes are listed in Appendix B. In Figure 6, the evaluation results by MTurk are listed.

Among the methods that use the robust classifier, PROBEGAN-robust achieves the best intra-FID score (45.40 for "horse" and 68.33 for "truck"). On the other hand, the NAIVE methods either generate adversarial examples with the regular classifier or images that are morphed from other classes (e.g., auto-like horse head and truck), which is clear evidence that the discriminator is overfitting class differences instead of naturalness. As a result, the generated images are forced to contain features from the seen classes. Finally, the images generated by BIGGAN-AM are far from natural, which validates previous observation that these methods are only capable of generating features that are present in the image set but they cannot generalize to unseen features. These observations are also evident in Figure 6: PROBEGAN achieves the best human recognition rate, while samples generated by BIGGAN-AM are barely recognizable.

To better appreciate the quality of the images generated by PROBEGAN, Figure 3 shows sample images of all the cases where each of the 10 CIFAR-10 classes serves as the target classes respectively. As can be seen, PROBEGAN is able to generate images that are very faithful to reality, which reveals the rather surprising abundance of information that the classifier can memorize. For example, the generated images mostly contain coherent backgrounds – trucks and cars on road, planes in the sky, horses on grass, etc. Each class contains great diversities in terms of color, orientation, breed/sub-types, etc. Some horse images even show humans

[2] https://github.com/qilimk/biggan-am

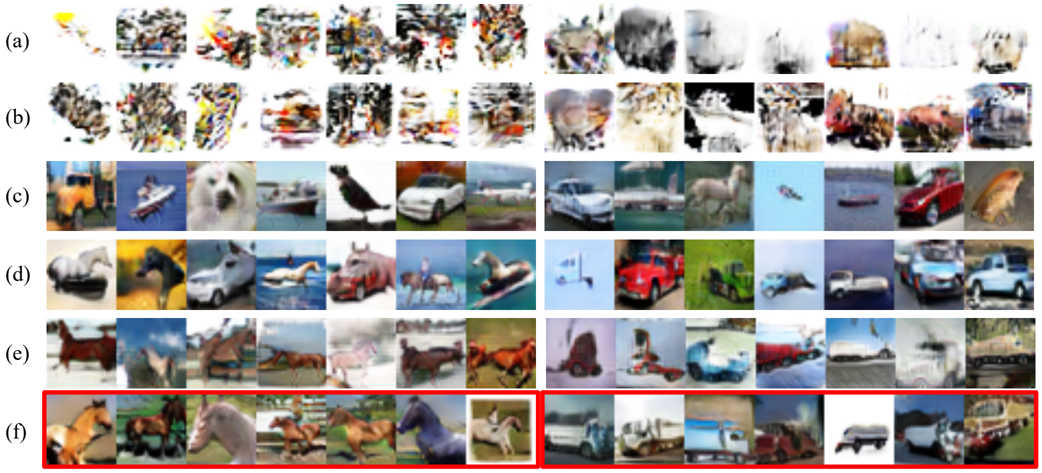


Figure 2: Sample generated images of "horse" (left) and "truck" (right) by (a) BigGAN-AM-regular, (b) BigGAN-AM-robust, (c) Naive-regular, (d) Naive-robust, (e) ProbeGAN-regular, and (f) ProbeGAN-robust (marked red).

Table 1: FID results on CIFAR-10. Gray background indicates the unseen class. Results for BigGAN is from our reimplementation, which is better than that is reported in [5].

Dataset	Network	FID ↓	intra-FID ↓									
			plane	auto	bird	cat	deer	dog	frog	horse	ship	truck
CIFAR10	BigGAN* [5]	7.99	29.05	13.14	26.73	24.23	16.25	26.25	24.08	14.20	14.64	17.37
	ProbeGAN-oracle	6.67	26.41	12.93	27.12	26.39	15.18	22.42	18.00	13.69	15.95	14.03
w/o horse	BigGAN-AM-regular	168.4	-	-	-	-	-	-	-	227.3	-	-
	Naive-regular	31.70	-	-	-	-	-	-	-	101.7	-	-
	ProbeGAN-regular	8.99	23.38	11.60	23.98	26.45	14.24	23.44	16.69	91.35	13.31	13.56
	BigGAN-AM-robust	161.6	-	-	-	-	-	-	-	223.3	-	-
	Naive-robust	48.92	-	-	-	-	-	-	-	76.02	-	-
	ProbeGAN-robust	8.39	26.13	12.60	24.28	27.08	15.63	24.53	17.69	**45.40**	14.83	14.02
w/o truck	BigGAN-AM-regular	114.1	-	-	-	-	-	-	-	-	-	179.3
	Naive-regular	33.36	-	-	-	-	-	-	-	-	-	118.5
	ProbeGAN-regular	8.71	24.30	12.64	23.89	25.45	13.31	22.30	16.75	13.88	14.12	105.99
	BigGAN-AM-robust	99.20	-	-	-	-	-	-	-	-	-	161.5
	Naive-robust	56.47	-	-	-	-	-	-	-	-	-	84.21
	ProbeGAN-robust	8.80	27.70	14.63	25.62	27.02	14.99	23.27	17.89	15.08	15.54	**68.33**

riding on the horses. These results indicate that the classifier may contain more information than previously expected.

The results also indicate that the classifier does not memorize all the information. Some images for "cat" and "dog" capture the fur or the head, and distort the body shapes, suggesting that the classifier only focuses on parts of the subject. It is worth mentioning that the classification accuracy of the model on "cat" and "dog" is the lowest. Interestingly, the generated images of "truck" often have large blocks of smooth color and lack finer details of texture. This indicates that the classifier relies more on outline shapes to recognize trucks instead of fine textures.

4.3 Importance of Target Absence Constraint

To illustrate the importance of the target absence constraint, we use all the images of CIFAR-10 including the target class ("truck" as an example) to train ProbeGAN, but manually inject an artificial marker to the unseen class. Specifically, for the unseen class "truck", we add 8×8 red blocks at the top-left and top-right corners of all the images, whereas no blocks are added to other classes. As shown in Figure 4, all the generated images of "truck" contain the artificial red blocks just like the altered training images. However, the classifier does not contain this information since it was trained on the original CIFAR-10 dataset. This proves that it is important to remove the samples of the target class from the dataset, otherwise, it is ambiguous whether the generated feature comes from the classifier or the dataset itself. We also demonstrate in Appendix B.1 that if an artificial feature is included in the training of the classifier instead, ProbeGAN can recover the artificial feature which is not preset when training ProbeGAN, suggesting that ProbeGAN can probe the memory of neural classifiers.

4.4 Inspection of Spurious Features

It has been observed in [41] that on the task of classifying between waterbirds and landbirds, image classifiers trained with empirical risk minimization (ERM) are undesirably sensitive to the biased background, but those trained with distributionally robust optimization (DRO) with regulariza-

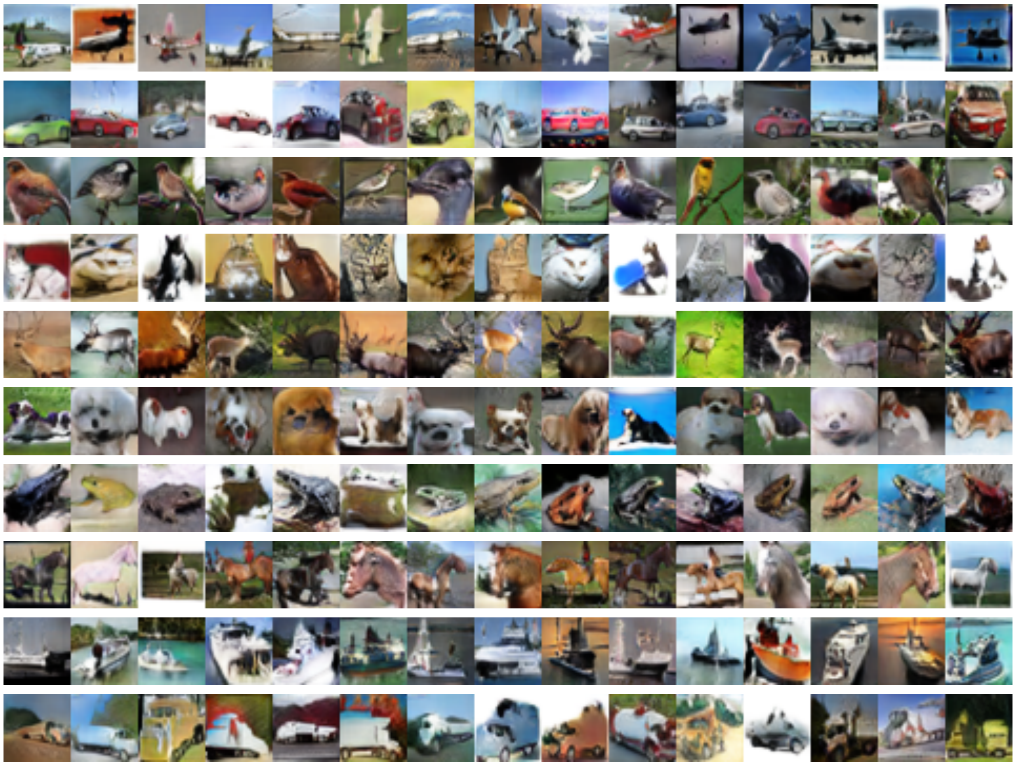


Figure 3: Sample images of CIFAR-10 classes generated by PROBEGAN-robust. Each class is taken as the “unseen” class, respectively. Each row corresponds to one unseen class setting, which are, from top to bottom, plane, auto, bird, cat, deer, dog, frog, horse, ship, and truck.

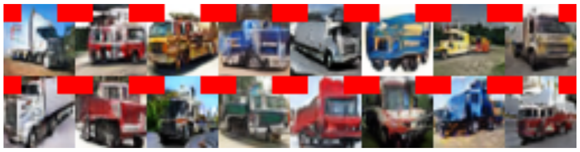


Figure 4: Sample generated images of “truck” when images of “truck” with artificial red blocks are present while other classes remain unchanged.

tions are more robust to the background. With PROBEGAN, we can investigate whether and to what extent these classifiers memorize the background features.

We use PROBEGAN to visualize two classifiers, ERM and DRO with an ℓ_2 penalty, trained with the same settings as in [41] (model accuracy is listed in Appendix A). When the target class is waterbird, we only include images of landbirds on *land background* in the image set, which guarantees no water background are included, and vice versa. Figure 5 shows the visualization results. As can be seen, when waterbirds are generated based on images of landbirds on land background, the results for DRO show the birds on top of land background which is inherited from the training images. This indicates that DRO does not memorize the biased background at all, and thus PROBEGAN copies whatever background that is in the image set. On the contrary, the results with ERM contain water bodies as background, which proves that the classifier remembers “water” as part of waterbirds. This is consistent with the classifier accuracy on waterbirds with land background, 79.7% for DRO and 30.8% for ERM (Appendix A), as the ERM classifier overfits some coherent background features. When generating landbirds with images of waterbirds on water background, both DRO and ERM perform inadequately, with features of trees and bamboos (land background in Waterbird dataset) added to the images even though the training images only contain water background. This suggests that both methods overfit to the land background.

4.5 Robust v.s. Regular Classifiers

With PROBEGAN, we are also able to answer some questions about different classifiers, the first question being how robust classifiers are different from regular classifiers. Thus we also implemented systems with the robust classifier replaced with a regular classifier (the algorithm names are appended “-regular”). According to Table 1, the intra-FID of PROBEGAN-regular significantly degrades compared to that of PROBEGAN-robust. By comparing Figures 2(e) and (f), we can see that PROBEGAN-regular generates much less visually distinct features. It is also obvious in Figure 6 that samples generated with robust models reproduce the target class much better than those with regular models. These results suggest that regular classifiers tend to over-emphasize features that are visually imperceptible, whereas robust classifiers would only focus on the visually salient

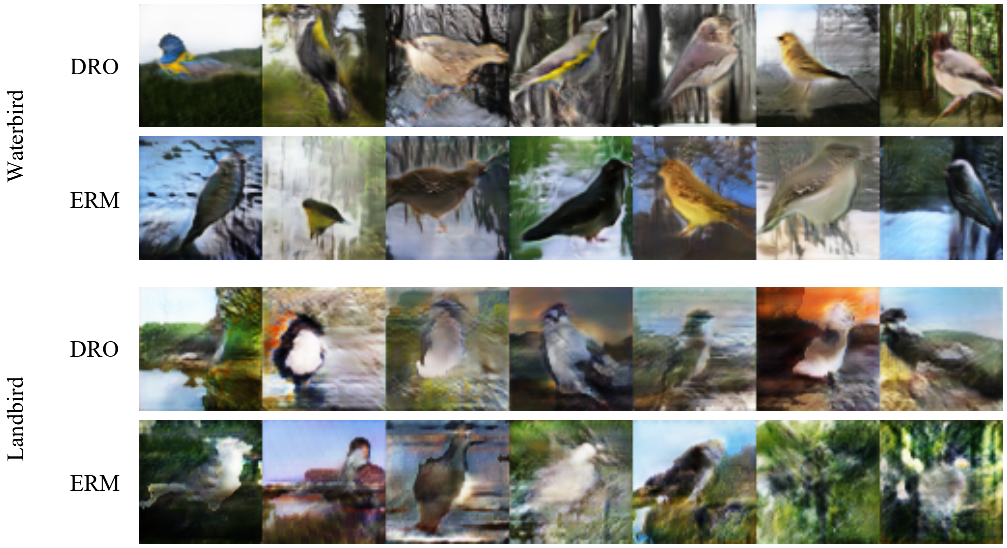


Figure 5: Samples of Waterbird and Landbird with classifier trained using DRO or ERM, respectively. When generating images of waterbirds, only images of landbirds on land background are used to avoid information leak, and vice versa.

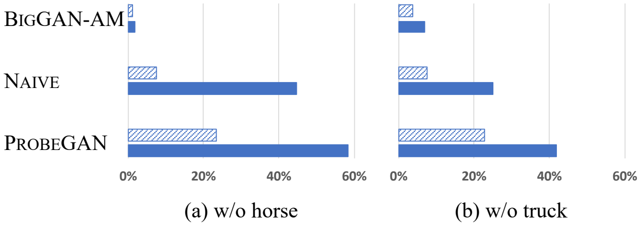


Figure 6: Human recognition rate by MTurk. The dashed bars represent the results using a regular classifier, while the solid bars with a robust classifier.

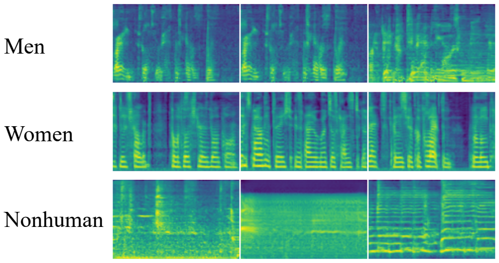


Figure 7: Sample mel-spectrograms of generated audio clips when each of the classes, men, women, and nonhuman, is taken as the "unseen" class.

ones. This is consistent with the conclusions in [17, 44]. The effect of classifier performance is also discussed in Appendix B.

4.6 Application to the Audio Modality

To demonstrate the generalizability on other modalities, we use PROBEGAN to elicit the memory of a classifier that recognizes men's voice, women's voice, and nonhuman sounds. Figure 7 shows sample mel-spectrograms generated by PROBEGAN . Our MTurk evaluation reports that PROBEGAN can generate perceptually-convincing audios for the unseen classes, with human recognition rates of 49% / 64% / 61% for men / women / nonhuman, respectively, even though no real data of the target class is provided. We encourage readers to listen to our audio clips online[3]. Additional details and results can be found in the appendix.

5 Conclusions and Limitations

We study the problem of model interpretation by feature visualization. While existing methods are able to generate plausible features by using learned priors from large datasets, they also mix the features from the classifier and the prior, making interpretation impossible. We propose a PROBEGAN framework that excludes the data of target classes and generates samples conditional on unseen classes with information from the classifier and generic natural data of classes other than the target classes. Experiments on both image and audio datasets demonstrate that PROBEGAN can generate natural samples of the target classes even if no real data of these classes are provided. By doing so, PROBEGAN offers a way to interpret neural classifiers.

On the other hand, there are some limitations in PROBEGAN. First, one would need to train a different PROBEGAN for each target class, leading to low efficiency. Generating multiple target classes would be an interesting future direction. Second, we observed that PROBEGAN suffers from mode collapse, so the generated images may over emphasize certain features. Finally, since PROBEGAN seeks to generate natural images, it would not probe the behavior on OOD images, which may hide issues of the classifier. As a result, PROBEGAN is not guaranteed to exhaust all the features that the classifier memorizes, so the absence of a certain feature in the generated images should not be interpreted as a proof that the classifier ignores this feature.

[3] https://anonymous0203.github.io/

References

[1] Arjovsky, M.; and Bottou, L. 2017. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*.

[2] Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.

[3] Bellemare, M. G.; Danihelka, I.; Dabney, W.; Mohamed, S.; Lakshminarayanan, B.; Hoyer, S.; and Munos, R. 2017. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*.

[4] Booth, S.; Zhou, Y.; Shah, A.; and Shah, J. 2021. Bayes-TrEx: a Bayesian Sampling Approach to Model Transparency by Example. In *the AAAI Conference on Artificial Intelligence*.

[5] Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*.

[6] Carter, S.; Armstrong, Z.; Schubert, L.; Johnson, I.; and Olah, C. 2019. Activation atlas. *Distill*, 4: e15.

[7] De Vries, H.; Strub, F.; Mary, J.; Larochelle, H.; Pietquin, O.; and Courville, A. C. 2017. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, 6594–6604.

[8] Dosovitskiy, A.; and Brox, T. 2016. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, 658–666.

[9] Dumoulin, V.; Shlens, J.; and Kudlur, M. 2017. A learned representation for artistic style. In *International Conference on Learning Representations*.

[10] Engstrom, L.; Ilyas, A.; Santurkar, S.; and Tsipras, D. 2019. Robustness (Python Library). https://github.com/MadryLab/robustness.

[11] Erhan, D.; Bengio, Y.; Courville, A.; and Vincent, P. 2009. Visualizing higher-layer features of a deep network. *University of Montreal Tech Report*.

[12] Gemmeke, J. F.; Ellis, D. P. W.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R. C.; Plakal, M.; and Ritter, M. 2017. Audio Set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*. New Orleans, LA.

[13] Gong, M.; Xu, Y.; Li, C.; Zhang, K.; and Batmanghelich, K. 2019. Twin auxilary classifiers GAN. In *Advances in Neural Information Processing Systems*, 1328–1337.

[14] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2672–2680.

[15] Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, 5767–5777.

[16] Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 6626–6637.

[17] Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, 125–136.

[18] Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

[19] Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4401–4410.

[20] Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition*, 8110–8119.

[21] Kodali, N.; Abernethy, J.; Hays, J.; and Kira, Z. 2017. On convergence and stability of GANs. *arXiv preprint arXiv:1705.07215*.

[22] Krizhevsky, A.; Nair, V.; and Hinton, G. 2009. The cifar-10 dataset. https://www.cs.toronto.edu/~kriz/cifar.html.

[23] Kumar, K.; Kumar, R.; de Boissiere, T.; Gestin, L.; Teoh, W. Z.; Sotelo, J.; de Brébisson, A.; Bengio, Y.; and Courville, A. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*.

[24] Li, Q.; Mai, L.; Alcorn, M. A.; and Nguyen, A. 2020. In *Asian Conference on Computer Vision*.

[25] Lim, J. H.; and Ye, J. C. 2017. Geometric GAN. *arXiv preprint arXiv:1705.02894*.

[26] Mahendran, A.; and Vedaldi, A. 2016. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 233–255.

[27] Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; and Smolley, S. P. 2017. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision*, 2794–2802.

[28] Mescheder, L.; Geiger, A.; and Nowozin, S. 2018. Which training methods for GANs do actually converge? *arXiv preprint arXiv:1801.04406*.

[29] Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[30] Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

[31] Miyato, T.; and Koyama, M. 2018. cGANs with projection discriminator. In *International Conference on Learning Representations*.

[32] Mordvintsev, A.; Olah, C.; and Tyka, M. 2015. Inceptionism: Going deeper into neural networks. *Google Research Blog*.

[33] Nguyen, A.; Clune, J.; Bengio, Y.; Dosovitskiy, A.; and Yosinski, J. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *IEEE conference on Computer Vision and Pattern Recognition*.

[34] Nguyen, A.; Dosovitskiy, A.; Yosinski, J.; Brox, T.; and Clune, J. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, 3387–3395.

[35] Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE conference on Computer Vision and Pattern Recognition*.

[36] Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, 271–279.

[37] Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier GANs. In *34th International Conference on Machine Learning*, 2642–2651.

[38] Olah, C.; Mordvintsev, A.; and Schubert, L. 2017. Feature visualization. *Distill*.

[39] Olah, C.; Satyanarayan, A.; Johnson, I.; Carter, S.; Schubert, L.; Ye, K.; and Mordvintsev, A. 2018. The building blocks of interpretability. *Distill*, 3: e10.

[40] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A.; and Li, F.-F. 2015. Imagenet large scale visual recognition challenge. In *International Journal of Computer Vision*, 211–252.

[41] Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*.

[42] Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2234–2242.

[43] Salimans, T.; Zhang, H.; Radford, A.; and Metaxas, D. 2018. Improving GANs using optimal transport. In *International Conference on Learning Representations*.

[44] Santurkar, S.; Ilyas, A.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Image synthesis with a single (robust) classifier. In *Advances in Neural Information Processing Systems*, 1260–1271.

[45] Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. *ICLR workshop*.

[46] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

[47] Veaux, C.; Yamagishi, J.; MacDonald, K.; et al. 2016. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.

[48] Wei, D.; Zhou, B.; Torrabla, A.; and Freeman, W. 2015. Understanding intra-class knowledge inside cnn. *arXiv preprint arXiv:1507.02379*.

[49] Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*.

[50] Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833.

[51] Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2018. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*.

[52] Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2019. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, 7354–7363.

[53] Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40: 1452–1464.