# GAN Dissection: Visualizing and Understanding Generative Adversarial Networks(Empirical Analysis)

David Bau[1]
davidbau@csail.mit.edu
Jun-Yan Zhu[1]
junyanz@csail.mit.edu
Joshua B. Tenenbaum[1]
jbt@csail.mit.edu
William T. Freeman[1]
billf@csail.mit.edu
Hendrik Strobelt[2]
hendrik.strobelt@ibm.com
Bolei Zhou[3]
bzhou@ie.cuhk.edu.hk

[1] Massachusetts Institute of Technology, Boston
[2] IBM Research, Cambridge MA
[3] The Chinese University of Hong Kong

**Abstract**

Generative Adversarial Networks(GANs) are getting quite good at rendering scenes which gets difficult even for human eyes to discriminate the authenticity of the image. People will often confuse these generated images with original training set images. One can imagine to generate such beautiful images like these by either learning about objects and compose the scene out of objects it knows or by memorizing a scene that it saw in the training data and render one just like it. The question, what does the GAN's actually know?. Do they memorize the objects or do they actually generate the real looking image of the object, these questions became the motivation for the authors. They identified a group of interpretable units that are closely related to object concepts using a segmentation-based network dissection method. Then, they quantified the causal effect of interpretable units by measuring the ability of interventions to control objects in the output. This work has examined the contextual relationship between these units and their surroundings by inserting the discovered object concepts into new images. They have shown several practical applications that are enabled by their framework, from comparing internal representations across different layers, models, and datasets, to improving GANs by locating and removing artifact-causing units, to interactively manipulating objects in a scene.

## 1 Introduction

To a human observer, a well-trained GAN appears to have learned facts about the objects in the image, for example, a door can appear on a building but not on a tree. We wish to

understand how a GAN represents such a structure. Do the objects emerge as pure pixel patterns without any explicit representation of objects such as doors and trees, or does the GAN contain internal variables that correspond to the objects that humans perceive? If the GAN does contain variables for doors and trees, do those variables cause the generation of those objects, or do they merely correlate? How are relationships between objects represented?. One can imagine two general ways that a computer algorithm could generate a beautiful realistic looking scene, either It could learn about objects and compose the scene out of objects it knows or it could memorize a scene that it saw in the training data and render one just like it. State-of-the-art GANs are still not perfect. They sometimes produce images that are quite unrealistic. Here, in this paper, the progressive GAN trained on 3,033,042 bedroom images, the largest LSUN data set, and it still outputs images with visible artifacts. They appear about 5% of the time as quoted by the authors in this work. We want to know, what causes these mistakes?. This question is the motivation behind this paper "GAN Dissection".

## 2 Literature Review

The quality and diversity of results from GANs [2] has continued to improve, from generating simple digits and faces [2], to synthesizing natural scene images [1][5], to generating 1k photorealistic portraits [3], to producing one thousand object classes [9]. In addition to image generation, GANs have also enabled many applications such as visual recognition , image manipulation and video generation [8]. Despite the huge success, little work has been done to visualize what GANs have learned. Prior work [5] manipulates latent vectors and observes how the results change accordingly.
Visualizing deep neural networks. Various methods have been developed to understand the internal representations of networks, such as visualizations for CNNs and RNNs [3]. We can visualize a CNN by locating and reconstructing salient image features or by mining patches that maximize hidden layers' activations , or we can synthesize input images to invert a feature layer . Alternately, we can identify the semantics of each unit [9] by measuring agreement between unit activations and object segmentation masks. Visualization of an RNN has also revealed interpretable units that track long-range dependencies[3]. Most previous work on network visualization has focused on networks trained for classification; our work explores deep generative models trained for image generation. Explaining the decisions of deep neural networks. We can explain individual network decisions using informative heatmaps or modified back-propagation [6][6][7]. The heatmaps highlight which regions contribute most to the categorical prediction given by the networks. Recent work has also studied the contribution of feature vectors[9] or individual channels to the final prediction. [4] has examined the effect of individual units by ablating them. Those methods explain discriminative classifiers. Our method aims to explain how an image can be generated by a network, which is much less explored.

## 3 The Model

The basis for the study is three variants of progressive GANs trained on LSUN scene datasets. To understand what's going on inside these GANs, the authors developed a technique involving a combination of dissection and intervention. Dissection and Intervention have been used to understand the internal functioning of GAN's. Given a trained segmentation model (i.e.,

a model that can map pixels in an image to one of a set of predefined object classes), we can dissect the intermediate layers of the GAN to identify the level of agreement between individual units and each object class. The segmentation model used in the paper was trained on the ADE20K scene dataset and can segment an input image into 336 object classes, 29 parts of large objects, and 25 materials. Dissection can reveal units that correlate with the appearance of objects of certain classes. Two different types of intervention help us to understand this relation better. First, we can ablate those units (switch them off) and see if the correlated objects disappear from an image in which they were previously present. Second, we can force the units on and see if the correlated objects appear in an image in which they were previously absent.

For dissection, an upsampled and thresholded feature map of a unit is taken and compared it to the segmentation map of a given object class. The extent of agreement is captured using an IoU (intersection-over-union) measure. They took the intersection of the thresholded image and the pixels defined as belonging to the segment class and divided it by their union. The result shows what fraction of the combined pixels are correlated with the class. To find causal relationships through the intervention, We can say that a given hidden unit causes the generation of object(s) of a given class if ablating that unit causes the object to disappear and activating it causes the object to appear. Averaging effects over all locations and images provides the ACE (average causal effect) of a unit on the generation of a given class. While these measures can be applied to a single unit, we have found that objects tend to depend on more than one unit. Thus we need to identify a set of units U that maximize the average causal effect for an object class c. While these measures can be applied to a single unit, we have found that objects tend to depend on more than one unit. Thus we need to identify a set of units U that maximize the average causal effect for an object class c This set is found by optimizing an objective that looks for a maximum class difference between images with partial ablation and images with partial insertion, using a parameter that controls the contribution of each unit. Here you can see the effects of increasing larger sets of hidden units, in this case identified as being associated with the class tree. This set is found by optimizing an objective that looks for a maximum class difference between images with partial ablation and images with partial insertion, using a parameter that controls the contribution of each unit. Here you can see the effects of increasing larger sets of hidden units, in this case identified as being associated with the class tree.

For dissection, we take an upsampled and thresholded feature map of a unit and compare it to the segmentation map of a given object class. The extent of the agreement is captured using an IoU (intersection-over-union) measure. We take the intersection of the thresholded image and the pixels defined as belonging to the segment class and divide it by their union. The result shows what fraction of the combined pixels are correlated with the class. To know "which convolutional units correlate to an object class that humans would recognize?", they have started it by sampling a bunch of random z and run the generator. Since the motivation inspires to focus on the internal neurons, they split the generator at a specific layer and examined the intermediate outputs of that layer. The paper uses "r" to denote the representation (the intermediate output) of the layer of interest. Completing the generator produces a synthetic image. The representation r will contain the output of every neuron of a layer. Then we can test causality by forcing all 20 of those units off. We literally reach into the network and just force the output of those neurons to zero, ignoring what they normally output, and then we continue the rest of the computation and generate the image. We get a synthesized image like this. They have called a convolutional channel a "unit". One unit of the representation forms a heat map over the image.

# 4 Data Description

The images used in this paper are all from LSUN dataset. The dataset sizes are pretty big and contains 126,227 outdoor church images, 1,315,802 living room images and 626,331 restaurant images. To generating intervention examples, churchoutdoor images are seeded by 495,586,279,700

# 5 Code Implementation

The authors have provided one demo .ipynb file to visualise the outputs, I have edited it a little to obtain results for all the other examples like kitchen, bedroom and trees. The authors gave example for church image set. Along with that, I have shown some additional results of GAN trying to generate some images in the form of application by adding or ablating the objects.

# 6 Results

Below are the results showing 1st Figure generated by the GAN and 2nd Figure shows the application made proposed by a methods which can ablate or add the tree or dome in the image just like we do in MS paint.

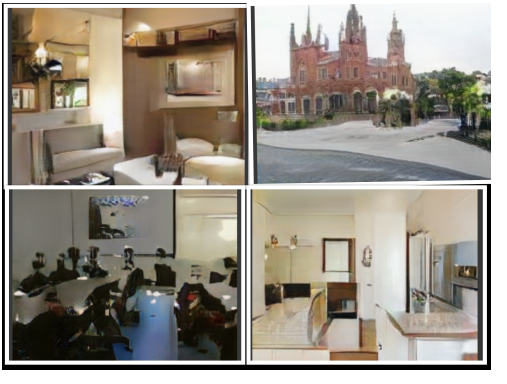

Figure 1: Generated Images

# References

[1] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks, 2015.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[3] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks, 2015.


Figure 2: GAN Paint

[4] Ari S. Morcos, David G. T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization, 2018.

[5] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

[6] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

[7] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[8] R. Swaminathan and S. Nayar. Nonmetric calibration of wide-angle lenses and polycameras. *IEEE T-PAMI*, 22(10):1172–1178, 2000.

[9] Z. Zhang. On the epipolar geometry between two images with lens distortion. In *Proc. ICPR*, pages 407–411, 1996.