Visualizing and Understanding Generative Adversarial Networks

David Bau[1], Jun-Yan Zhu[1], Hendrik Strobelt[2], Bolei Zhou[3], Joshua B. Tenenbaum[1], William T. Freeman[1], Antonio Torralba[1]
[1]Massachusetts Institute of Technology. [2]IBM Research Cambridge, MIT-IBM Watson AI Lab. [3]Chinese University of Hong Kong

The ability of generative adversarial networks to render nearly photorealistic images leads us to ask: What does a GAN know? For example, when a GAN renders a door on a building but not in a tree (Fig. 1a), we wish to understand whether such structure emerges as pure pixel patterns without explicit representation, or if the GAN contains internal variables that correspond to human-perceived objects such as doors, buildings, and trees. And when a GAN ocassionally renders an unrealistic image (Fig. 1f), we want to know if the mistake is caused by specific variables in the network.

We present a general method for visualizing and understanding GANs at different levels of abstraction, from each neuron, to each object, to the relationship between objects. Beginning with a Progressive GAN [3] trained to generate scenes (Fig. 1a), we first identify interpretable units that are related to semantic classes (Figs. 1b,2). Then, we directly intervene within the network to identify sets of units that cause a type of object to disappear (Fig. 1c) or appear (Figs. 1d,3). Finally, we study contextual relationships by observing where we can insert objects and how this intervention interacts with other objects in the image (Figs. 1d,3). This framework enables several applications: comparing internal representations across different layers, GAN variants, and datasets (Fig.2); debugging and improving GANs by locating and ablating artifact-causing units (Fig. 1e,f,g); understanding contextual relationships between objects in natural scenes (Fig.3); and manipulating images with interactive object-level control (video).

Inspired by the emergence of single-unit object detectors in deep classifiers [1], we analyze the internal GAN representations by decomposing the featuremap $\mathbf{r}$ at a layer into positions $\mathrm{P} \subset \mathbb{P}$ and unit channels $u \in \mathbb{U}$. To identify a unit u with semantic behavior, we upsample and threshold the unit (Fig. 1b), and measure how well it matches an object class c in the image $\mathbf{x}$ as identified by a supervised semantic segmentation network $\mathbf{s}_c(\mathbf{x})$ [5]

$$\mathrm{IoU}_{u,c} \equiv \frac{\mathbb{E}_{\mathbf{z}}\left|(\mathbf{r}^{\uparrow}_{u,\mathbb{P}} > t_{u,c}) \wedge \mathbf{s}_c(\mathbf{x})\right|}{\mathbb{E}_{\mathbf{z}}\left|(\mathbf{r}^{\uparrow}_{u,\mathbb{P}} > t_{u,c}) \vee \mathbf{s}_c(\mathbf{x})\right|}, \text{where } t_{u,c} = \arg\max_t \frac{\mathrm{I}(\mathbf{r}^{\uparrow}_{u,\mathbb{P}} > t; \mathbf{s}_c(\mathbf{x}))}{\mathrm{H}(\mathbf{r}^{\uparrow}_{u,\mathbb{P}} > t, \mathbf{s}_c(\mathbf{x}))}$$

The threshold $t_{u,c}$ is chosen to maximize the information quality ratio, that is, the portion of the joint entropy H which is mutual information I [4]. To identify a sets of units $\mathrm{U} \subset \mathbb{U}$ that cause semantic effects, we intervene in the network $G(\mathbf{z}) = f(h(\mathbf{z})) = f(\mathbf{r})$ by decomposing the featuremap $\mathbf{r}$ into two parts $(\mathbf{r}_{\mathrm{U,P}}, \mathbf{r}_{\overline{\mathrm{U,P}}})$, and forcing the components $\mathbf{r}_{\mathrm{U,P}}$ on and off:

Original image : $\mathbf{x} = G(\mathbf{z}) \equiv f(\mathbf{r}) \equiv f(\mathbf{r}_{\mathrm{U,P}}, \mathbf{r}_{\overline{\mathrm{U,P}}})$

Image with U ablated at pixels P : $\mathbf{x}_a = f(\mathbf{0}, \mathbf{r}_{\overline{\mathrm{U,P}}})$

Image with U inserted at pixels P : $\mathbf{x}_i = f(\mathbf{c}, \mathbf{r}_{\overline{\mathrm{U,P}}})$

We measure the average causal effect (ACE) [2] of units U on class c as:

$$\delta_{\mathrm{U}\to c} \equiv \mathbb{E}_{\mathbf{z},\mathrm{P}}[\mathbf{s}_c(\mathbf{x}_i)] - \mathbb{E}_{\mathbf{z},\mathrm{P}}[\mathbf{s}_c(\mathbf{x}_a)], \quad (1)$$

We find (Figs. 2,3) many units of GAN representations can be interpreted, not only as signals that correlate with object concepts but as variables that have a causal effect on the synthesis of semantic objects in the output. This analytic framework reveals some of the learned internal structure, enabling applications that apply direct interventions on the representations.

[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017.

[2] Paul W Holland. Causal inference, path analysis and recursive structural equations models. *ETS Research Report Series*, 1988(1):i–50, 1988.

[3] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[4] Dedy Rahman Wijaya, Riyanarto Sarno, and Enny Zulaika. Information quality ratio as a novel metric for mother wavelet selection. *Chemometrics and Intelligent Laboratory Systems*, 160:59–71, 2017.

[5] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

Figure 1: Overview: (a-d) We analyze how internal representations relate to (a) output of a Progressive GAN by identifying (b) units that correlate with semantic concepts and (c) intervening to remove and (d) add objects. (e-g) Our framework can be used to (e) identify units that (f) cause artifacts and (g) reduce artifacts when ablated. Please see our video for more results.

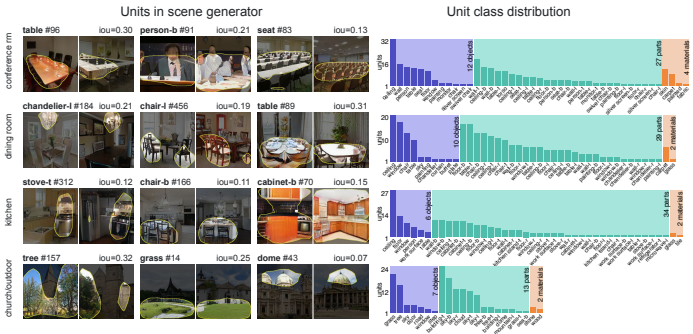

Figure 2: Comparing representations learned by progressive GANs trained on different scene types. The units that emerge match objects that commonly appear in the scene type: seats in conference rooms and stoves in kitchens. Units from `layer4` are shown. A unit is counted as a class predictor if it matches a supervised segmentation class with pixel accuracy > 0.75 and IoU > 0.05 when upsampled and thresholded. The distribution of units over classes is shown in the right column.

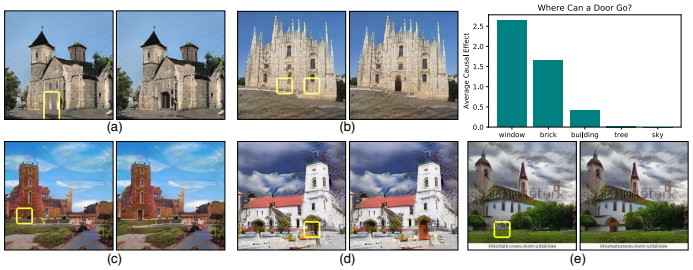

Figure 3: Inserting door units by setting 20 causal units to a fixed high value at one pixel in the representation. Whether the door units can cause the generation of doors is dependent on local context: we highlight every location that is responsive to insertions of door units on top of the original image, including two separate locations in (b) (we intervene at left). The same units are inserted in every case, but the door that appears has a size, alignment, and color appropriate to the location. One way to add door pixels is to emphasize a door that is already present: the result is a larger door (d). The chart summarizes the causal effect of inserting door units at one pixel with different context.