

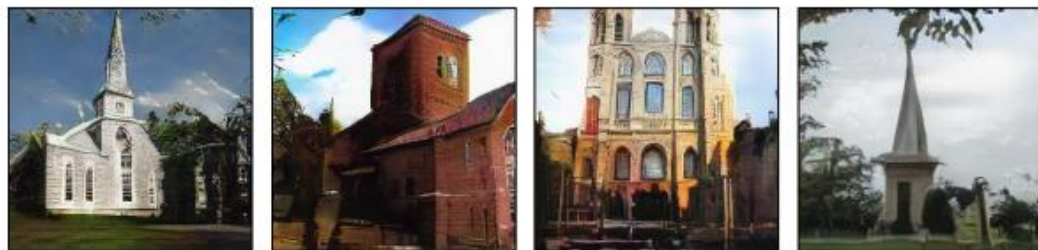
GAN DISSECTION: VISUALIZING AND UNDERSTANDING GENERATIVE ADVERSARIAL NETWORKS



(a) Generate images of churches



(b) Identify GAN units that match trees



(c) Ablating units removes trees



(d) Activating units adds trees



(e) Identify GAN units that cause artifacts



(e) Identify GAN units that cause artifacts



(f) Bedroom images with artifacts



(g) Ablating "artifact" units improves results

Описание метода.

$G: \mathbf{z} \rightarrow \mathbf{x}$ - генератор, где $\mathbf{z} \in \mathbb{R}^{|\mathbf{z}|}$ из какого-то распределения
 $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ - сгенерированная картинка $H \times W$

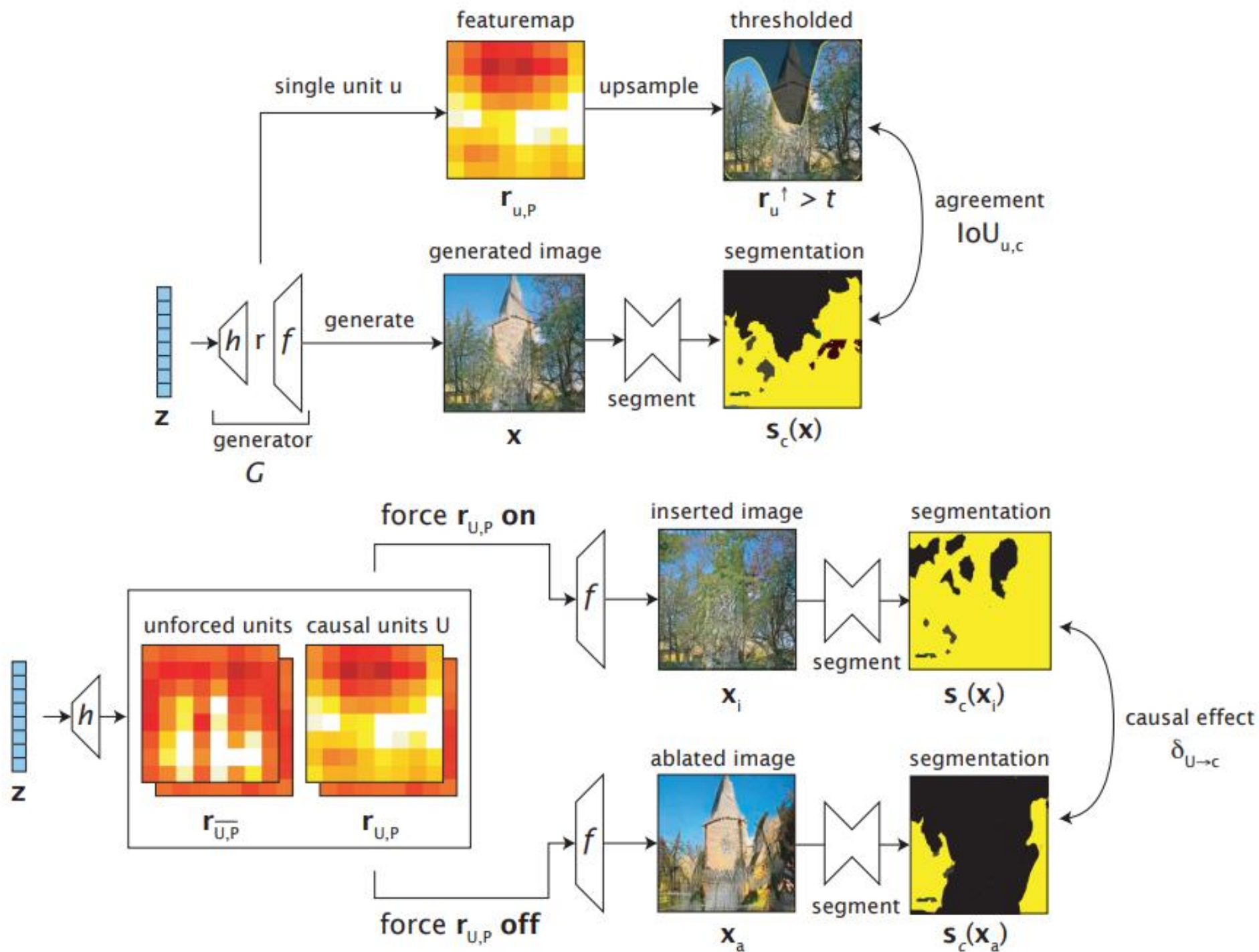
- $\mathbf{r} = h(\mathbf{z})$ - выход промежуточного слоя генератора, т.е. $\mathbf{x} = f(\mathbf{r}) = f(h(\mathbf{z})) = G(\mathbf{z})$
- Так как $\mathbf{x} = f(\mathbf{r})$, очевидно, \mathbf{r} содержит информацию о классе \mathbf{c} .
- Мы хотим узнать, **как** именно в \mathbf{r} закодирована информация о классе \mathbf{c} .

Описание метода.

- Делаем разбиение в локации P $\mathbf{r}_{U,P} = (\mathbf{r}_{U,P}, \mathbf{r}_{\bar{U},P})$, где класс c зависит юнитов $\mathbf{r}_{U,P}$ и не зависит от юнитов $\mathbf{r}_{\bar{U},P}$
- Юнит - каждый канал в нашей featuremap в r .
- U - множество интересных нам юнитов
- \bar{U} - множество всех юнитов
- \mathbb{P} - пиксели featuremap в r .

Этапы:

1. Dissection - берём большое множество классов и определяем, какие из них имеют точное представление (в виде юнитов) в r .
2. Intervention - определяем множество юнитов, которые отвечают за генерацию объектов класса (включаем и выключаем юниты)



Первый этап.

- Напомню, $\mathbf{r}_{u,\mathbb{P}}$ - одноканальная (юнит u) featuremap размером $h \times w$ (обычно меньше картинки на выходе)
- Хотим узнать, кодирует ли $\mathbf{r}_{u,\mathbb{P}}$ какой-то семантический класс, например "дерево".
- Выбираем такой набор классов \mathcal{C} , для каждого из которых мы имеем семантическую сегментацию $\mathbf{s}_c(\mathbf{x})$

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

$\mathbf{r}_{u,\mathbb{P}}^{\uparrow}$ - upsampled $\mathbf{r}_{u,\mathbb{P}}$ до $H \times W$



Thresholding unit #65 layer 3 of a dining room generator matches 'table' segmentations with $\text{IoU}=0.34$.



Thresholding unit #37 layer 4 of a living room generator matches 'sofa' segmentations with $\text{IoU}=0.29$.

Второй этап.

Original image :

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

Image with U ablated at pixels P :

$$\mathbf{x}_a = f(\mathbf{0}, \mathbf{r}_{\overline{U,P}})$$

Image with U inserted at pixels P :

$$\mathbf{x}_i = f(\mathbf{k}, \mathbf{r}_{\overline{U,P}})$$

Присутствие класса c на картинке проверяем с помощью сравнения картинок с обнуленными U в P (\mathbf{x}_a) и, наоборот, U в P заполненными значениями \mathbf{k} (\mathbf{x}_i).
Усредняем по всем картинкам и локациям.

Average Causal Effect (ACE) юнитов U на генерацию класса c :

$$\delta_{U \rightarrow c} \equiv \mathbb{E}_{\mathbf{z}, P}[s_c(\mathbf{x}_i)] - \mathbb{E}_{\mathbf{z}, P}[s_c(\mathbf{x}_a)]$$

Второй этап.

Пусть в r имеем d юнитов, тогда чтобы найти оптимальное сочетание юнитов с фиксированным размером $|U|$, понадобится исследовать $\binom{d}{|U|}$ подмножеств.

Вместо этого мы ищем оптимальный вектор $\alpha \in [0, 1]^d$, где α_u - степень влияния юнита u на генерацию c . Задача - максимизировать ACE $\delta_{\alpha \rightarrow c}$:

Image with partial ablation at pixels P : $\mathbf{x}'_a = f((\mathbf{1} - \alpha) \odot \mathbf{r}_{U,P}, \mathbf{r}_{U,\bar{P}})$

Image with partial insertion at pixels P : $\mathbf{x}'_i = f(\alpha \odot \mathbf{k} + (\mathbf{1} - \alpha) \odot \mathbf{r}_{U,P}, \mathbf{r}_{U,\bar{P}})$

Objective : $\delta_{\alpha \rightarrow c} = \mathbb{E}_{\mathbf{z}, P} [s_c(\mathbf{x}'_i)] - \mathbb{E}_{\mathbf{z}, P} [s_c(\mathbf{x}'_a)]$,

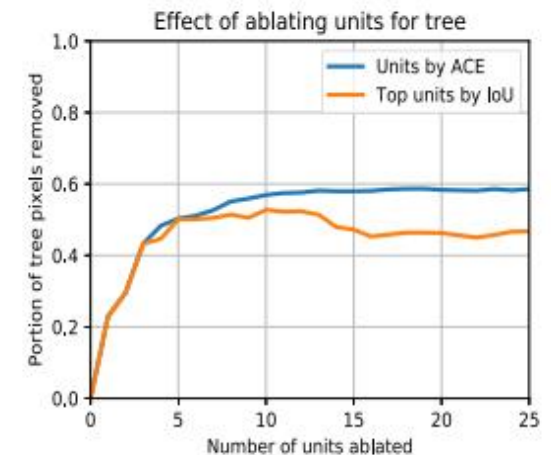
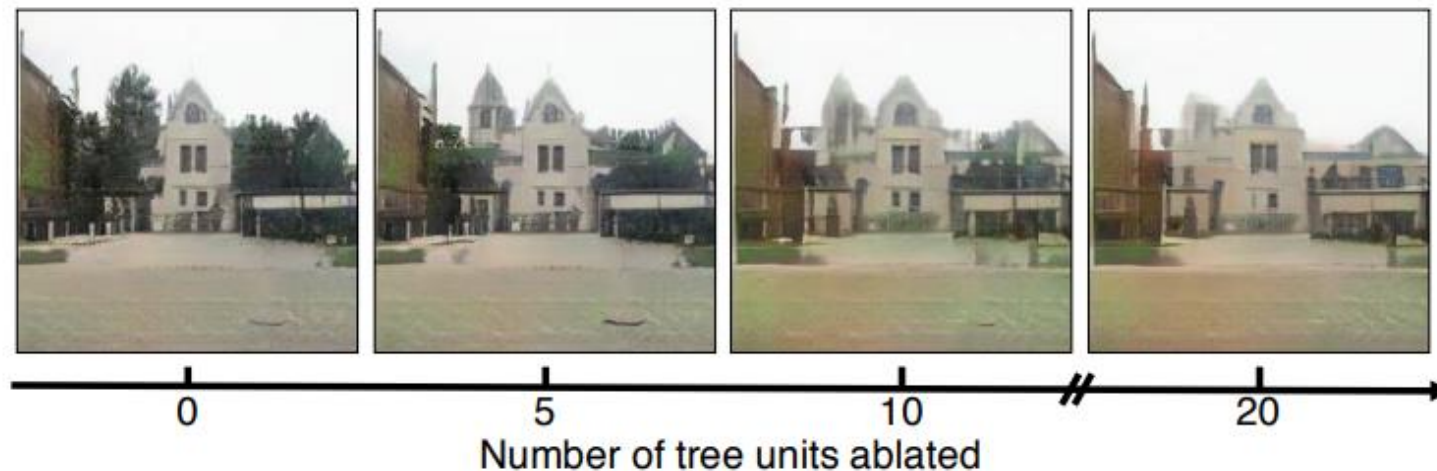
Второй этап.

Инициализируем: $\alpha_u = \frac{\text{IoU}_{u,c}}{\max_v \text{IoU}_{v,c}}$

$$\alpha^* = \arg \min_{\alpha} (-\delta_{\alpha \rightarrow c} + \lambda \|\alpha\|_2)$$

для минимального
множества юнитов

Ранжируем юниты по α_u^* , чтобы добиться максимального эффекта при удалении объектов этого класса. Чем больше юнитов удалим, тем меньше деревьев останется на картинке.



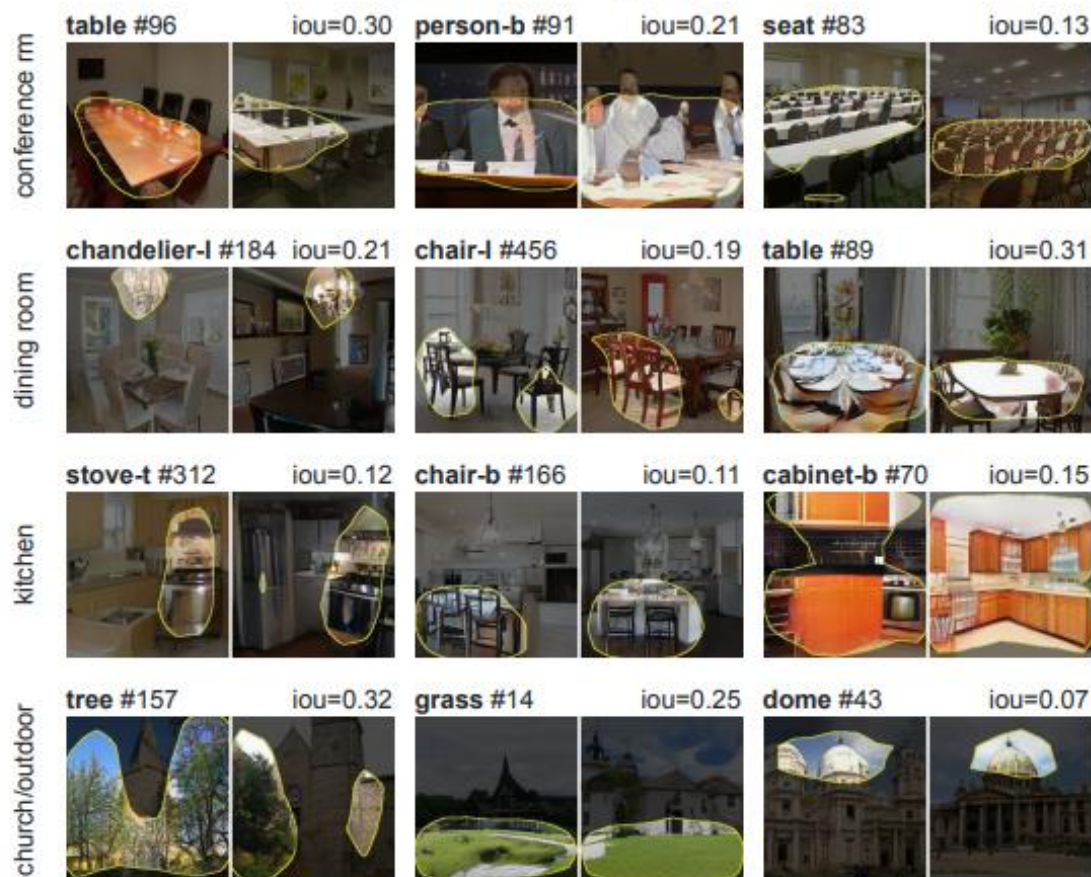
Результаты.

- Изучали три варианта ProgressiveGANa на LSUN scene dataset. Модель сегментации обучали на ADE20K scene dataset.
- Модель может сегментировать 336 классов, 29 частей объектов и 25 материалов.
- Так же объекты классов были поделены на более мелкие классы (части этих объектов по расположению) для более подробного изучения юнитов

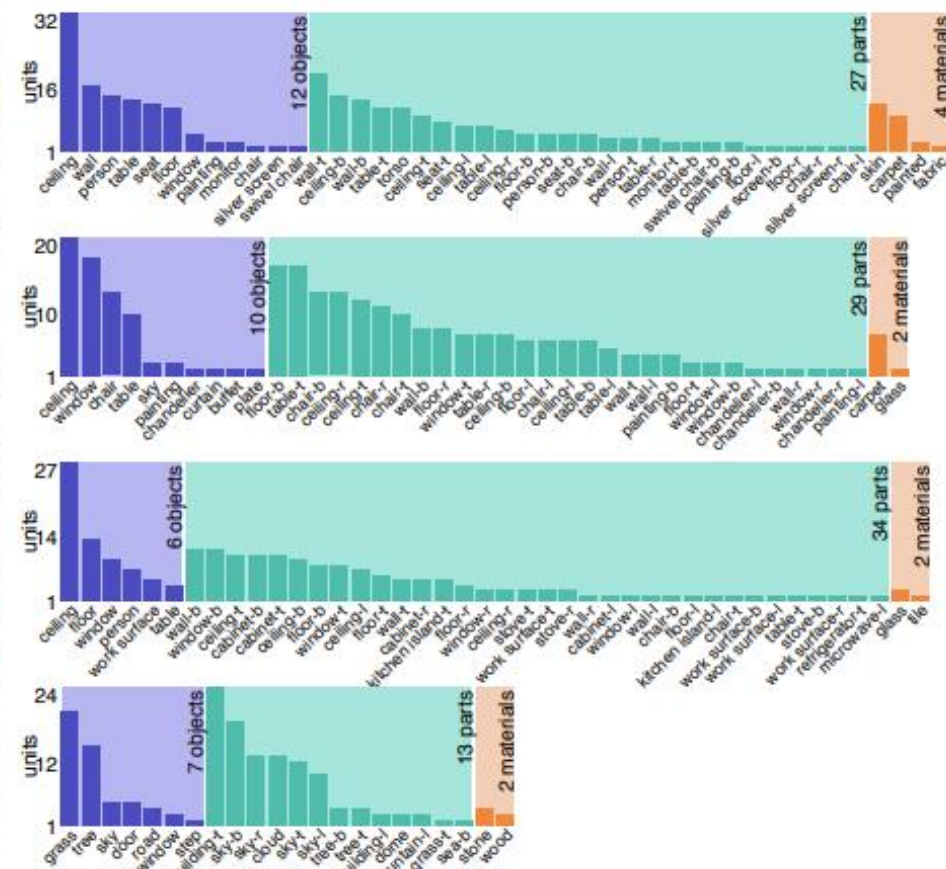
Результаты. Обучение на разных сценах.

Обратим внимание на то, что есть юниты, отвечающие за отдельные части объекта. На картинке представлены юниты 4-го слоя. Юнит считается, если $\text{IoU} > 0.05$ и $\text{pixel_accuracy} > 0.75$ по сравнению с сегментацией.

Units in scene generator



Unit class distribution



Результаты. Сравнение по слоям.

layer1

512 units total

0 object units

2 part units

0 material units



Unit class distribution



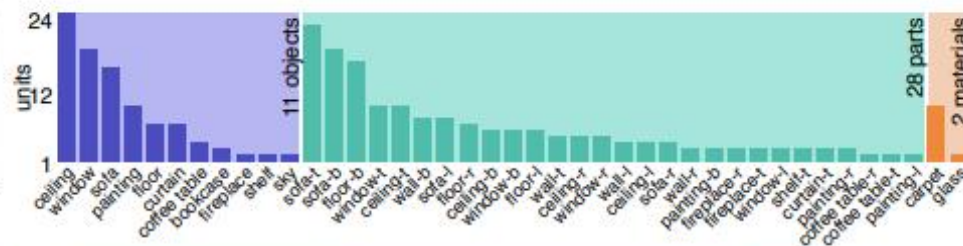
layer4

512 units total

86 object units

149 part units

10 material units



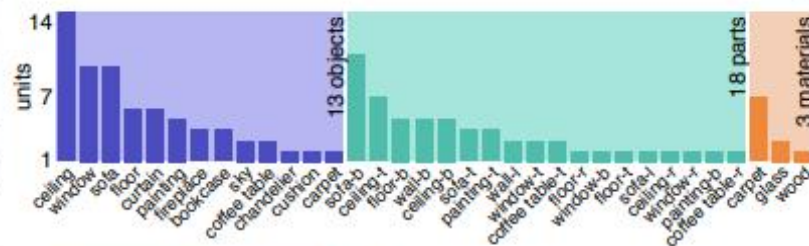
layer7

256 units total

59 object units

48 part units

9 material units



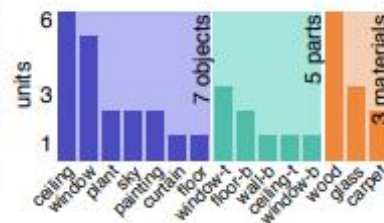
layer10

128 units total



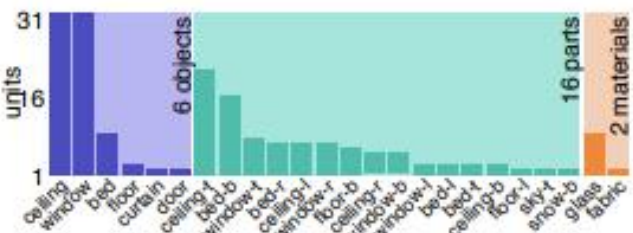


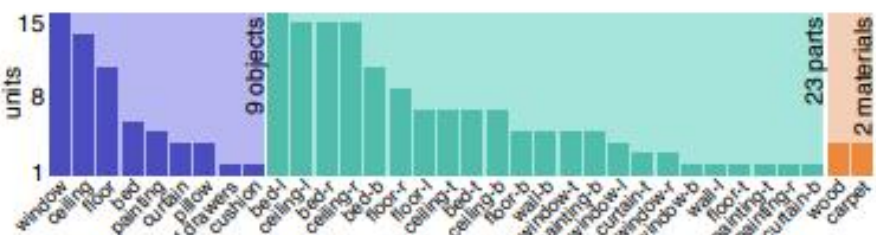


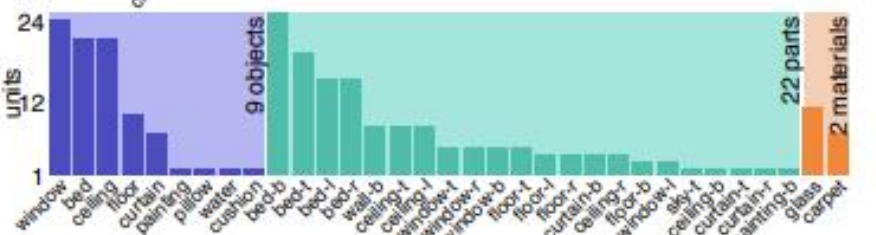
19 object units

8 part units

11 material units

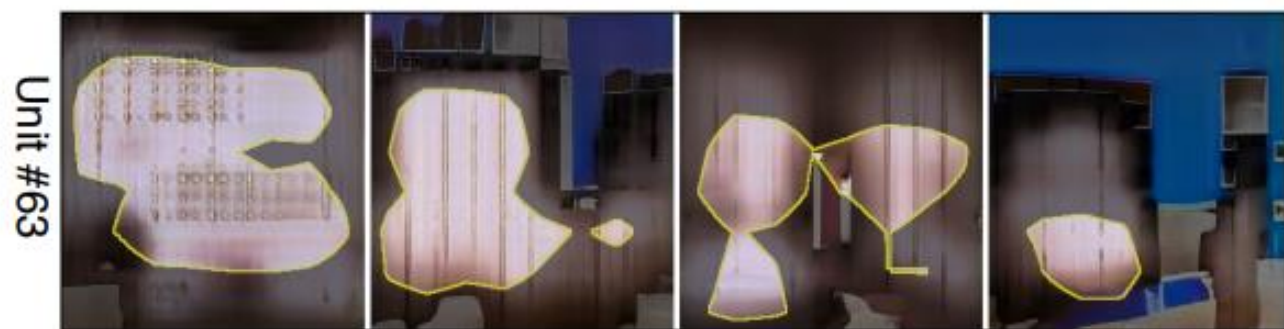


Результаты. Сравнение архитектур.

interpretable units	SWD	Best "bed" unit	Best "window" unit	Unit class distribution
base prog GAN 512 units total 74 object units 84 part units 9 material units		bed layer4 #253 iou=0.18 	window layer4 #142 iou=0.19 	
+batch stddev 512 units total 55 object units 128 part units 6 material units		bed layer4 #88 iou=0.11 	window layer4 #422 iou=0.25 	
+pixelwise norm 512 units total 82 object units 128 part units 16 material units		bed layer4 #129 iou=0.29 	window layer4 #494 iou=0.26 	

SWD - Sliced Wasserstein Distance. Чем меньше, тем реалистичнее картинка. Чем выше качество GANa, тем больше интерпретируемых юнитов.

Результаты. Удаление дефектов генерации.



(b) Bedroom images with artifacts



(a) Example artifact-causing units



(c) Ablating “artifact” units improves results

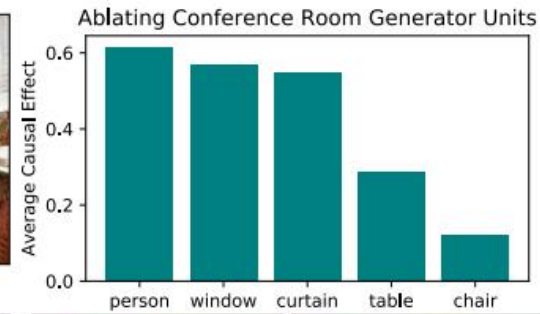
Результаты. Удаление объектов.



ablate person units



ablate curtain units



ablate window units



ablate table units



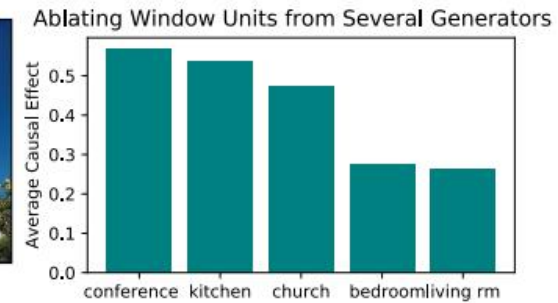
ablate chair units



conference room



church



kitchen



living room



bedroom

Результаты. Вставка объектов.



(a)



(b)



(c)



(d)



(e)