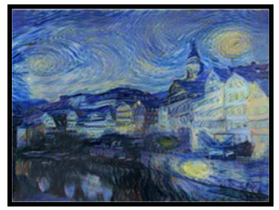
Image Editing with Optimization (part II)

Jun-Yan Zhu

16-726, Spring 2021

Image Editing with Optimization

$\hat{y}$ $\quad$ y

$|\mathrm{Gram}($  $) - \mathrm{Gram}($  $)|$

optimized output $\quad$ style image

$\hat{y}$ $\quad$ x

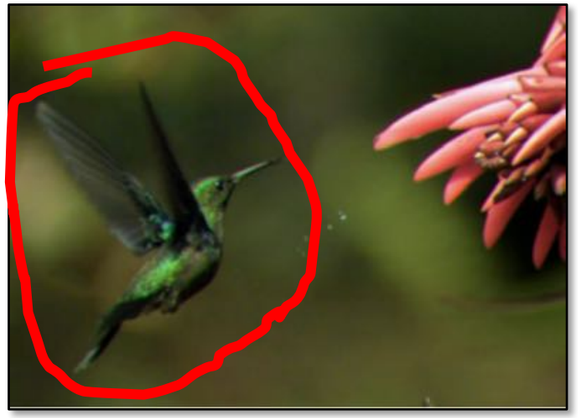
$+ \quad |\mathrm{F}($  $) - \mathrm{F}($  $)|$

optimized output $\quad$ content image

$$\arg\min_{\hat{y}} \mathcal{L}_{\text{style}}(\hat{y}, y) + \lambda \mathcal{L}_{\text{content}}(\hat{y}, x)$$

User input $\quad$ Input image

Image Editing with Optimization



input photo

Image Editing Program

User Input

result

$$\arg\min_{\hat{y}} \mathcal{L}_{\text{background\_boundary}}(\hat{y}, y) + \lambda \mathcal{L}_{\text{source\_gradient}}(\hat{y}, x)$$

result background result object

[Pérez et al. 2003], [Tao et al. 2014]

Learning Natural Image Manifold

- Deep generative models: $G(z) : z \rightarrow x$
  - Generative Adversarial Network (**GAN**)
    (e.g., DCGAN, StyleGAN2, BigGAN)
  - Variational Auto-Encoder (**VAE**)
    (e.g., VQ-VAE2)
  - Flow-based models (e.g., RealNVP, Glow)…

  …

Changing Variables

- Traditional method: Optimizing the image

user constraint

$$\hat{y}^* = \arg\min_{\hat{y}} \mathcal{L}(x, y, \hat{y})$$

input result

- New method: Optimizing the latent code

user constraint

$$z^* = \arg\min_{z} \mathcal{L}(x, y, G(z))$$

input Latent code

Generator

Projecting and Editing an Image

original photo

Project

projection on manifold

Editing UI

different degree of image manipulation

Edit Transfer

transition between the original and edited projection

Projecting and Editing an Image

original photo

Project

projection on manifold

Editing UI

different degree of image manipulation

Edit Transfer

transition between the original and edited projection

Projecting an Image into GAN Manifold

Input: real image x

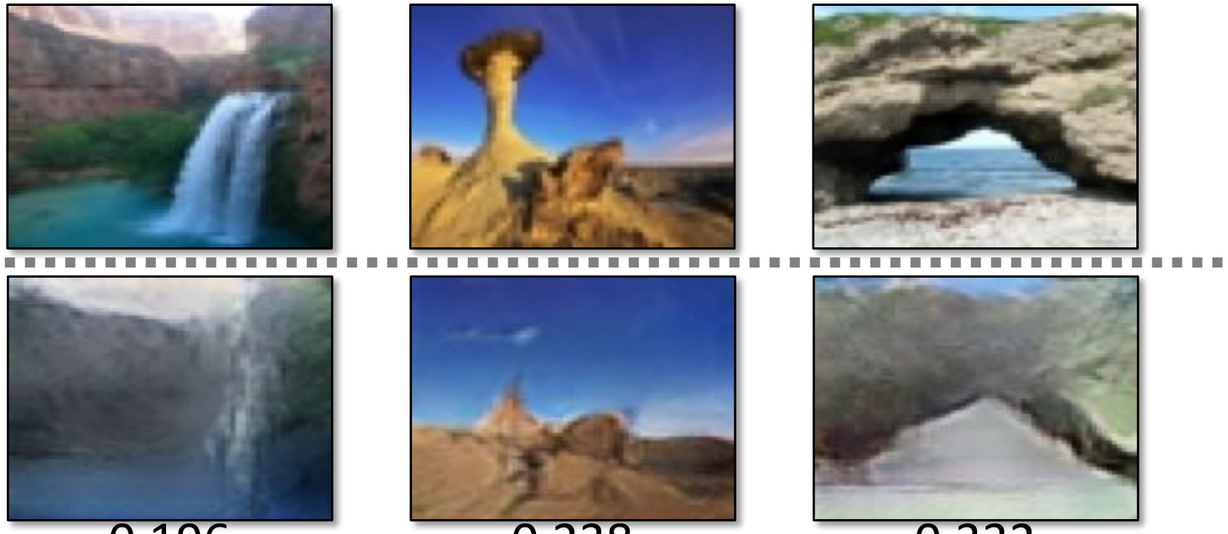
Output: latent vector z

Optimization

$$z^* = \arg\min_z \mathcal{L}(G(z), x)$$

Reconstruction loss

Generative model



0.196 0.238 0.332

Projecting an Image into GAN Manifold

Input: real image x

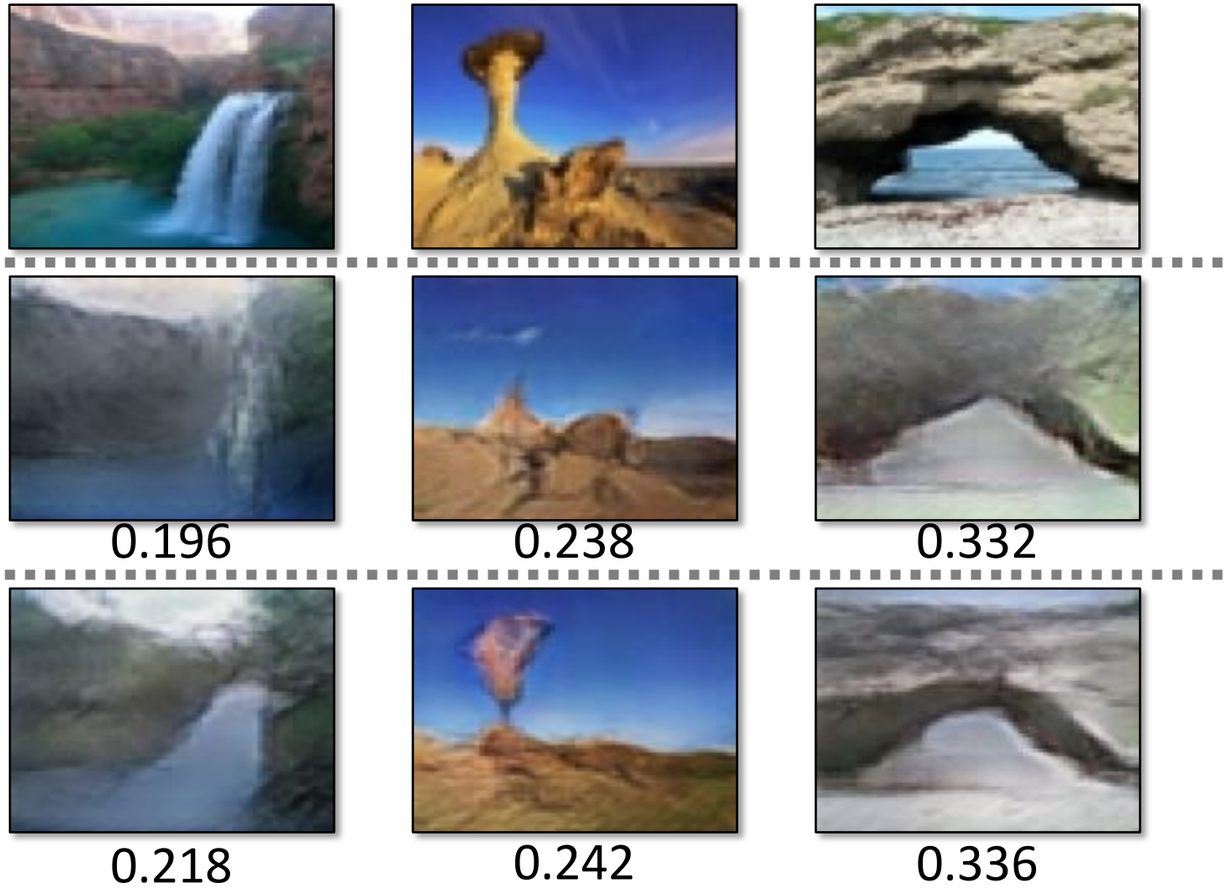
Output: latent vector z

Optimization

$$z^* = \arg\min_z \mathcal{L}(G(z), x)$$

Inverting Network $z = E(x)$

$$E = \arg\min_E \mathbb{E}_x \underbrace{\mathcal{L}(G(E(x)), x)}_{\text{Auto-encoder with a fixed decoder}}$$



also see VAE-GAN based image projection
Neural Photo Editor [Brock et al. ICLR 2017]

Projecting an Image into GAN Manifold

Input: real image x

Output: latent vector z

Optimization

$$z^* = \arg\min_z \mathcal{L}(G(z), x)$$

Inverting Network $z = E(x)$

$$E = \arg\min_E \mathbb{E}_x \mathcal{L}(G(E(x)), x)$$

Hybrid Method

Use the network as initialization for the optimization problem

Manipulating the Latent Code

original photo

Project

projection on manifold

Editing UI

different degree of image manipulation

Edit Transfer

transition between the original and edited projection

Post-Processing (optional)

original photo

Project

projection on manifold

Editing UI

different degree of image manipulation

Edit Transfer

transition between the original and edited projection

Image Editing with GANs

- Step 1: Image Projection/Reconstruction

$$z_0 = \arg\min_z \mathcal{L}(G(z), x)$$

- Step 2: Manipulating the latent code

$$z_1 = z_0 + \Delta z$$

- Step 3: Generate the edited result

$$G(z_1)$$

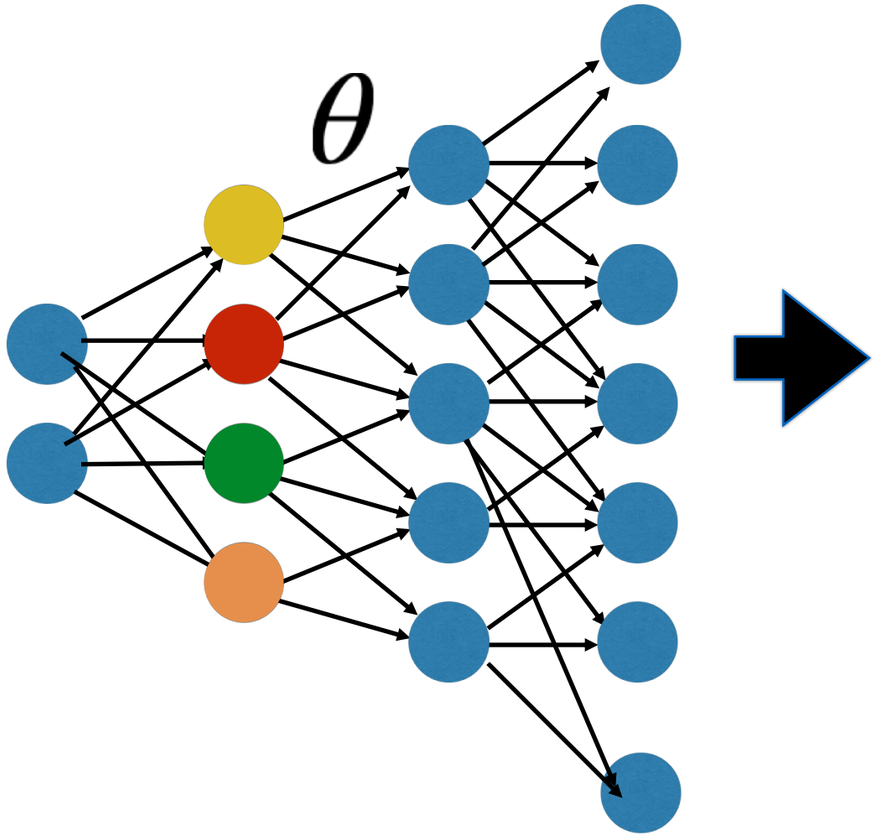
Image Projection with GANs

Image Reconstruction (high-res images, Big Models)

Original image x

Image Reconstruction (high-res images, Big Models)

Random vector z ?



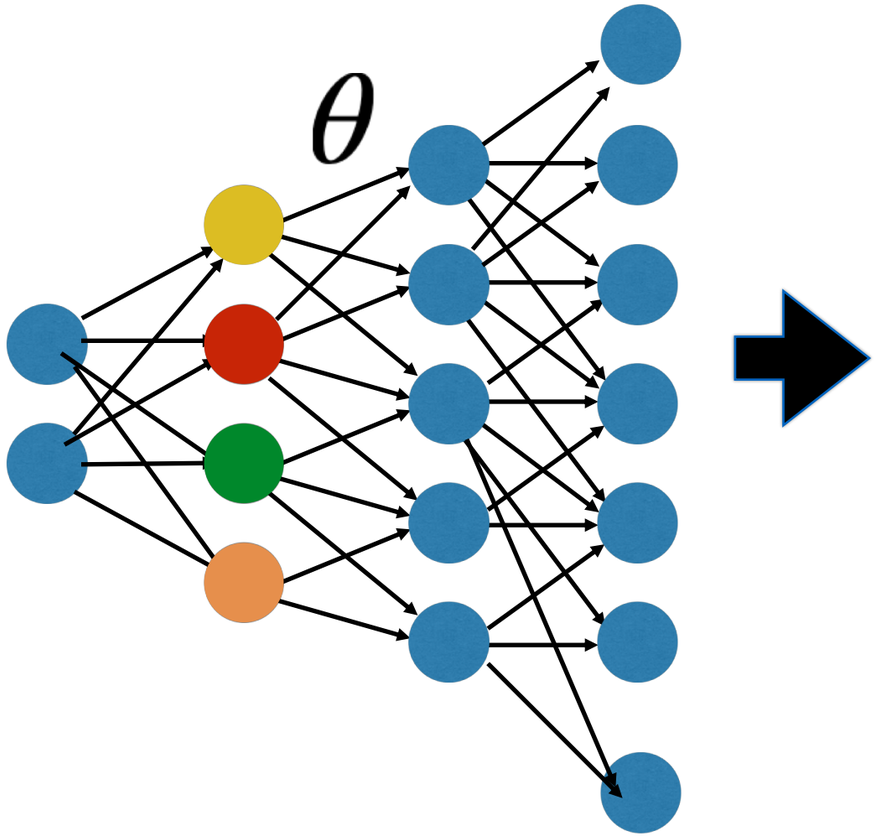
Original image x

$$z^* = \arg\min_z \mathcal{L}(G(z;\theta), x)$$

iGAN [Zhu et al. 2016]

Image Reconstruction (high-res images, Big Models)

Random vector z^*



Reconstructed image $G(z^*;\theta)$

$$z^* = \arg\min_z \mathcal{L}(G(z;\theta), x)$$

iGAN [Zhu, Krähenbühl, Shechtman, Efros]

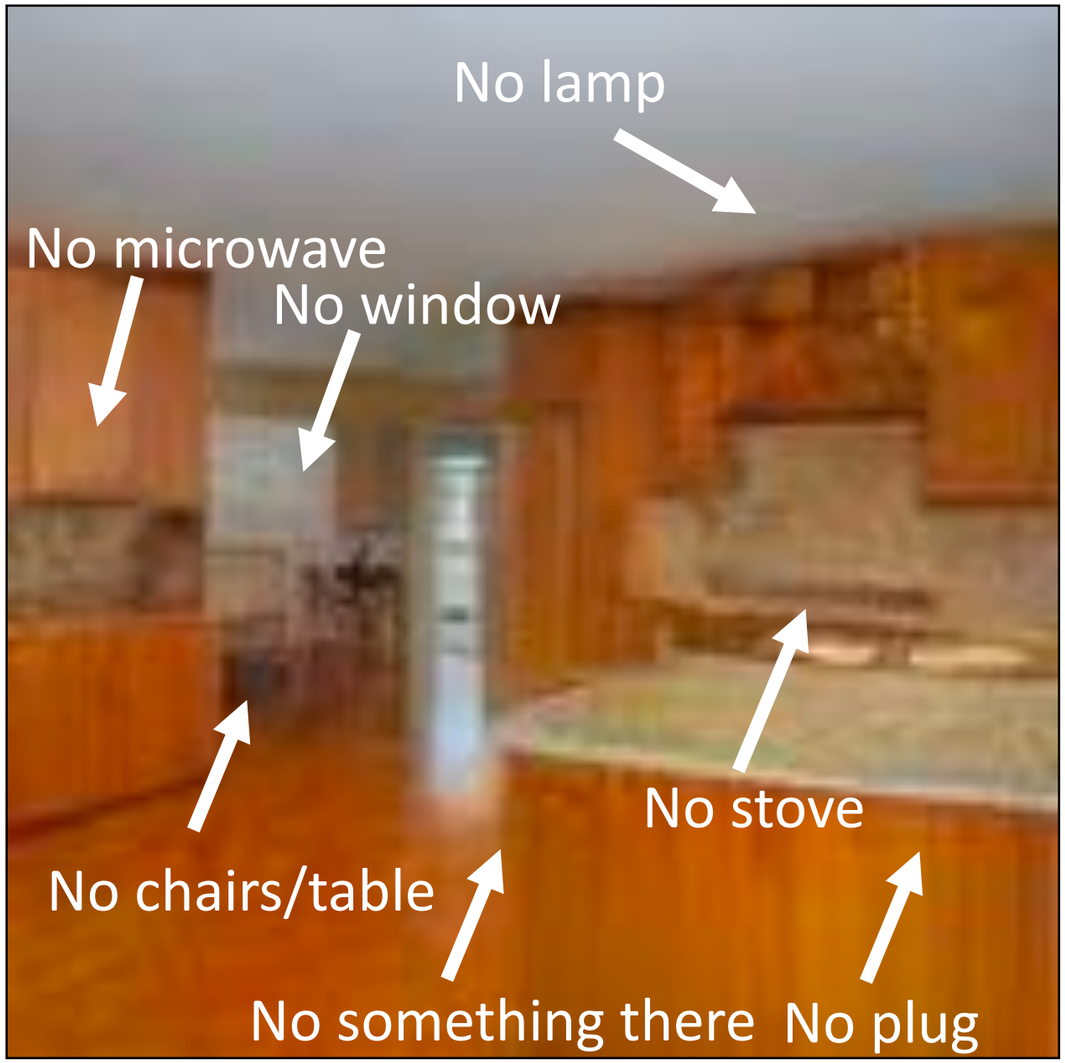
Find the Differences…

Original image

GAN reconstructed image

Find the Differences…

Original image

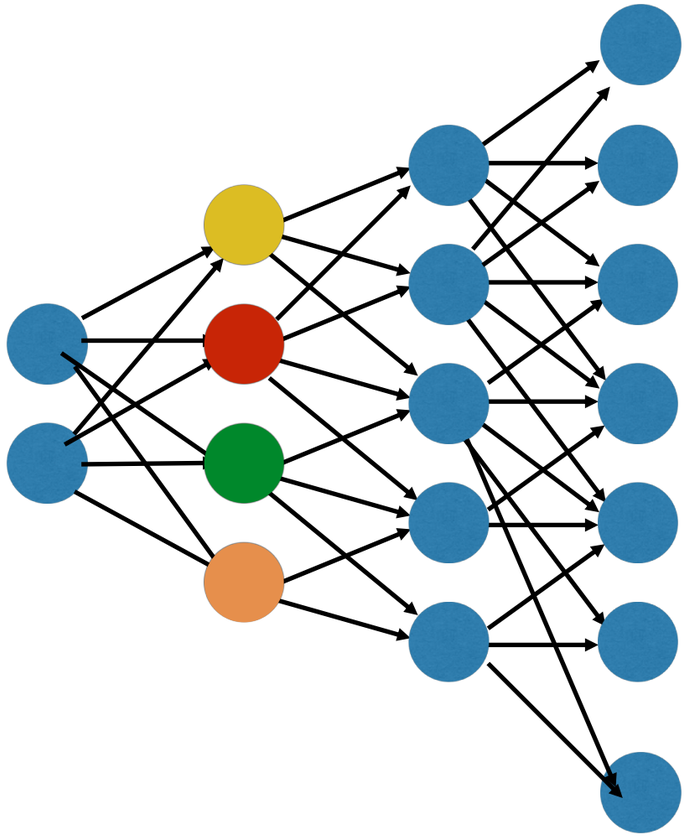


GAN reconstructed image

Reconstructing a Real Photo

Original image

Random vector z^*



Reconstructed image $G(z^*;\theta)$

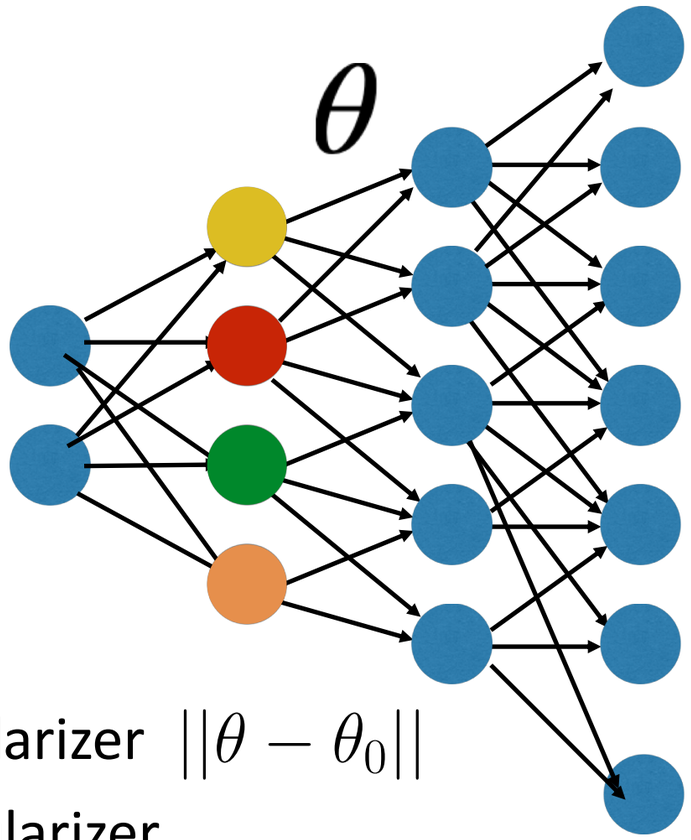
$$z^* = \arg\min_z \mathcal{L}(G(z;\theta), x)$$

iGAN [Zhu et al. 2016]

Reconstructing a Real Photo

Original image

Random vector z^*



Reconstructed image $G(z^*;\theta)$

Weight space regularizer $||\theta - \theta_0||$

Feature space regularizer

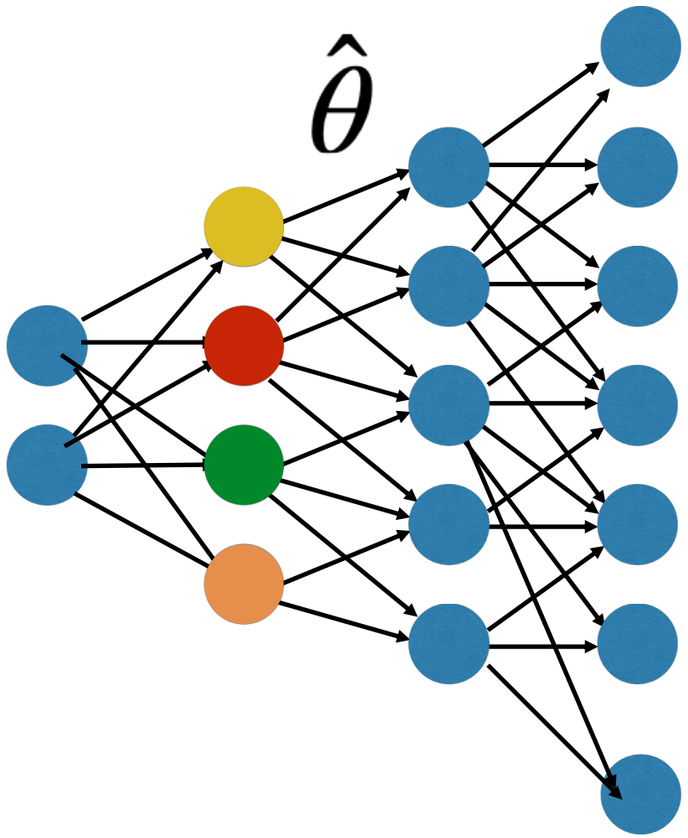
$$z^*, \theta^* = \arg\min_{z,\theta} \mathcal{L}(G(z;\theta), x)$$

← Regularizer

Reconstructing a Real Photo

Original image

Random vector z^*



Reconstructed image $G(z^*; \theta^*)$

$$z^*, \theta^* = \arg\min_{z,\theta} \mathcal{L}(G(z;\theta), x) + R(\theta)$$ ← Regularizer

Reconstructing a Real Photo

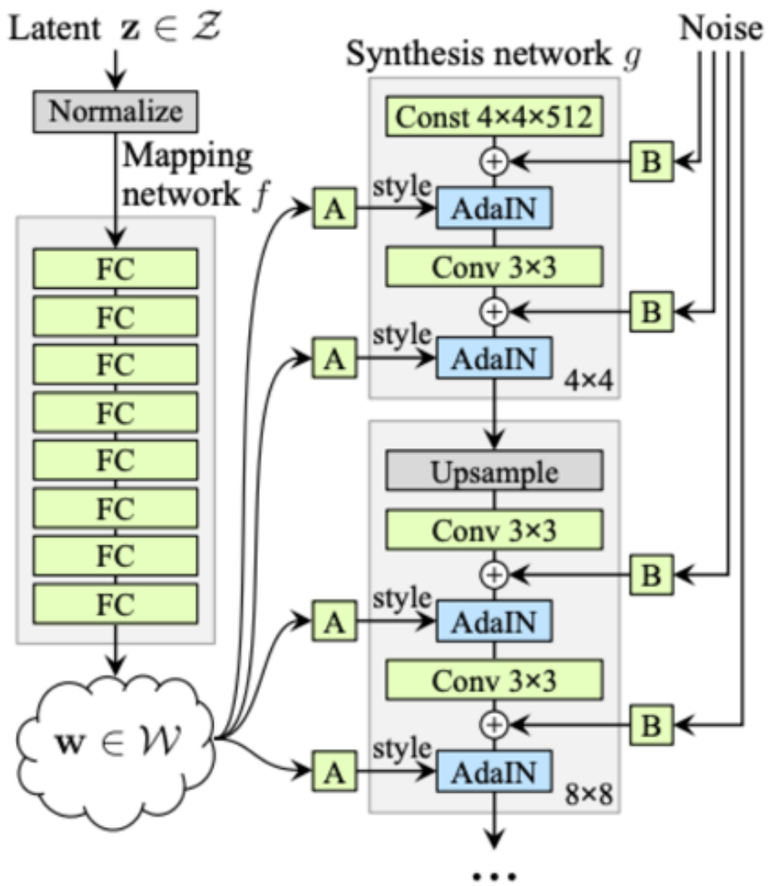
Original image

With z^*

With $z^*$and θ^*

Semantic Photo Manipulation [Bau, Strobelt, Peebles, Wulff, Zhou, Zhu, Torralba, SIGGRAPH 2019]
Inspired by Deep Image Prior [Ulyanov et al.] and Deep Internal learning [Shocher et al.]

Using Different Layers



Optimizing the latent code

$$z^* = \arg\min_z \mathcal{L}(G(z), x)$$

Optimizing the style code

$$w^* = \arg\min_w \mathcal{L}(g(w), x)$$

Optimizing the extended style code

$$w_+^* = \arg\min_{w+} \mathcal{L}(g(w_+), x)$$

Image2StyleGAN [Abdal et al., 2019], StyleGAN2 [Karras et al., 2019]

Using Different Layers: w space

StyleGAN — generated images

StyleGAN2 — generated images

StyleGAN2 [Karras et al., 2019]

Using Different Layers: w space

StyleGAN2 — real images

StyleGAN2 [Karras et al., 2019]

Using Different Layers: w+ space

All the results are reconstructed using Face Model

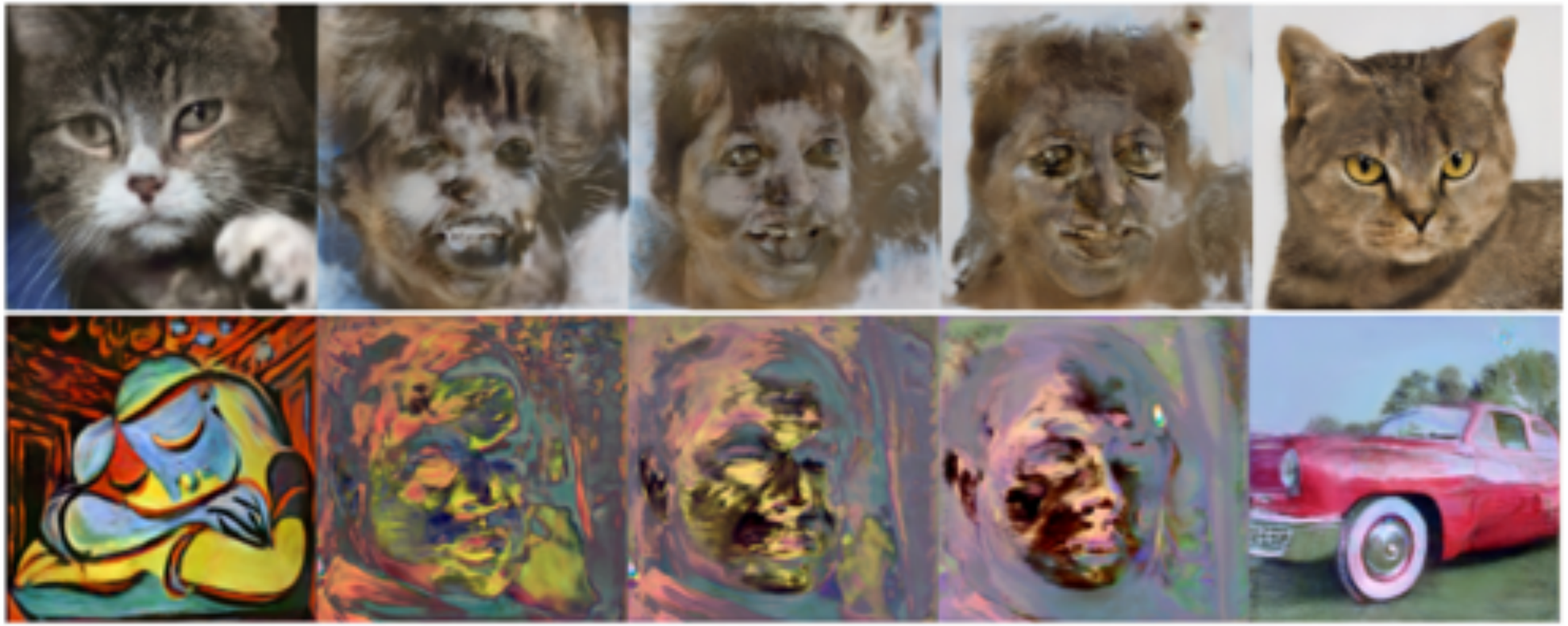
Image2StyleGAN [Abdal et al., 2019]

Reconstruction ≠ Editing

Interpolations between two images

Image2StyleGAN [Abdal et al., 2019]

Reconstruction ≠ Editing



Interpolations between two images

Image2StyleGAN [Abdal et al., 2019]

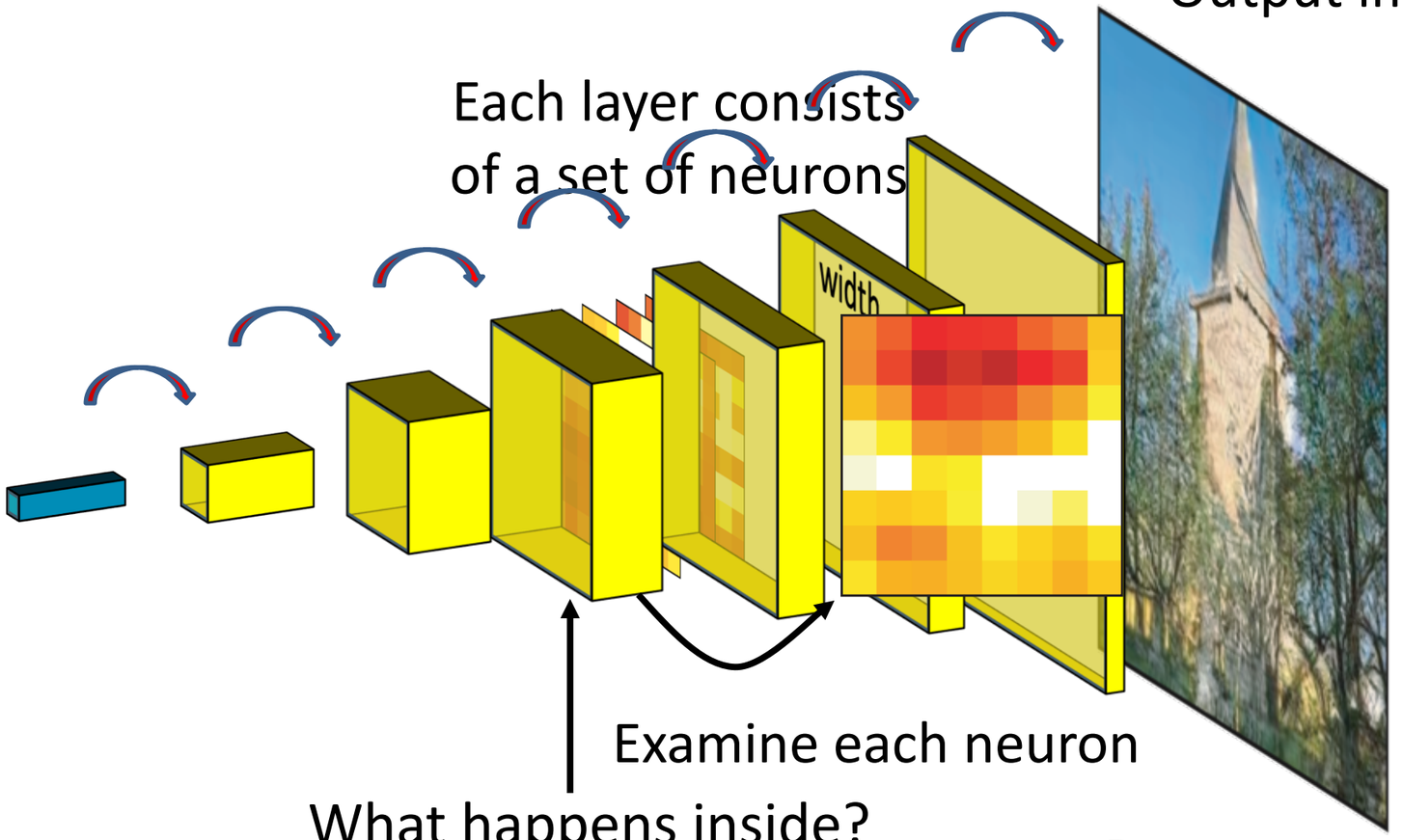
Manipulating Latent code/layer (channel analysis)

Understanding a Generator

Each step:
Increases spatial resolution

Output image

Each layer consists
of a set of neurons



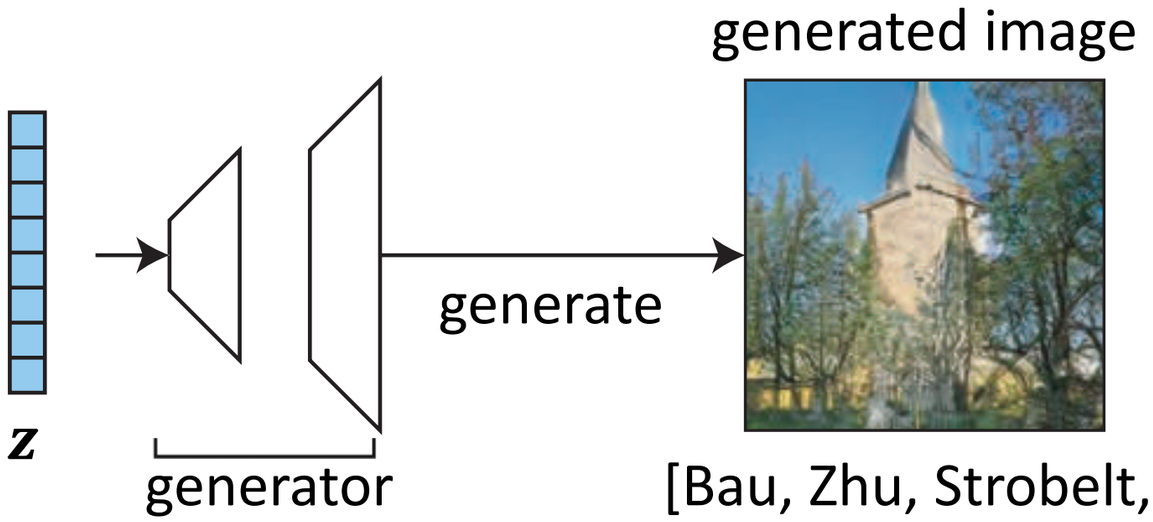
Input: random
vector ***z***

Examine each neuron

What happens inside?

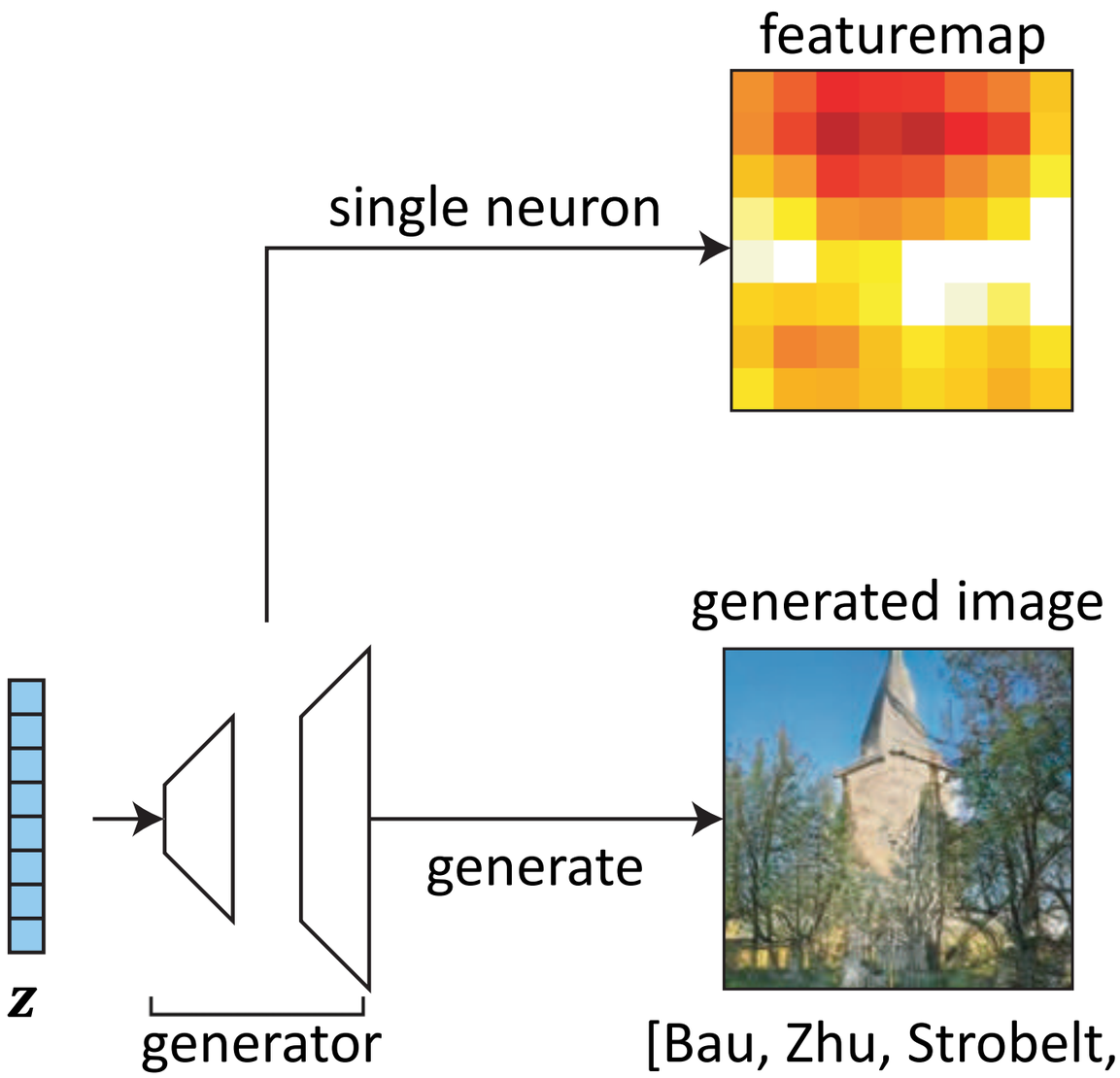
Progressive GANs [Karras et al. 2018]

Which neurons **correlate** to an object class?



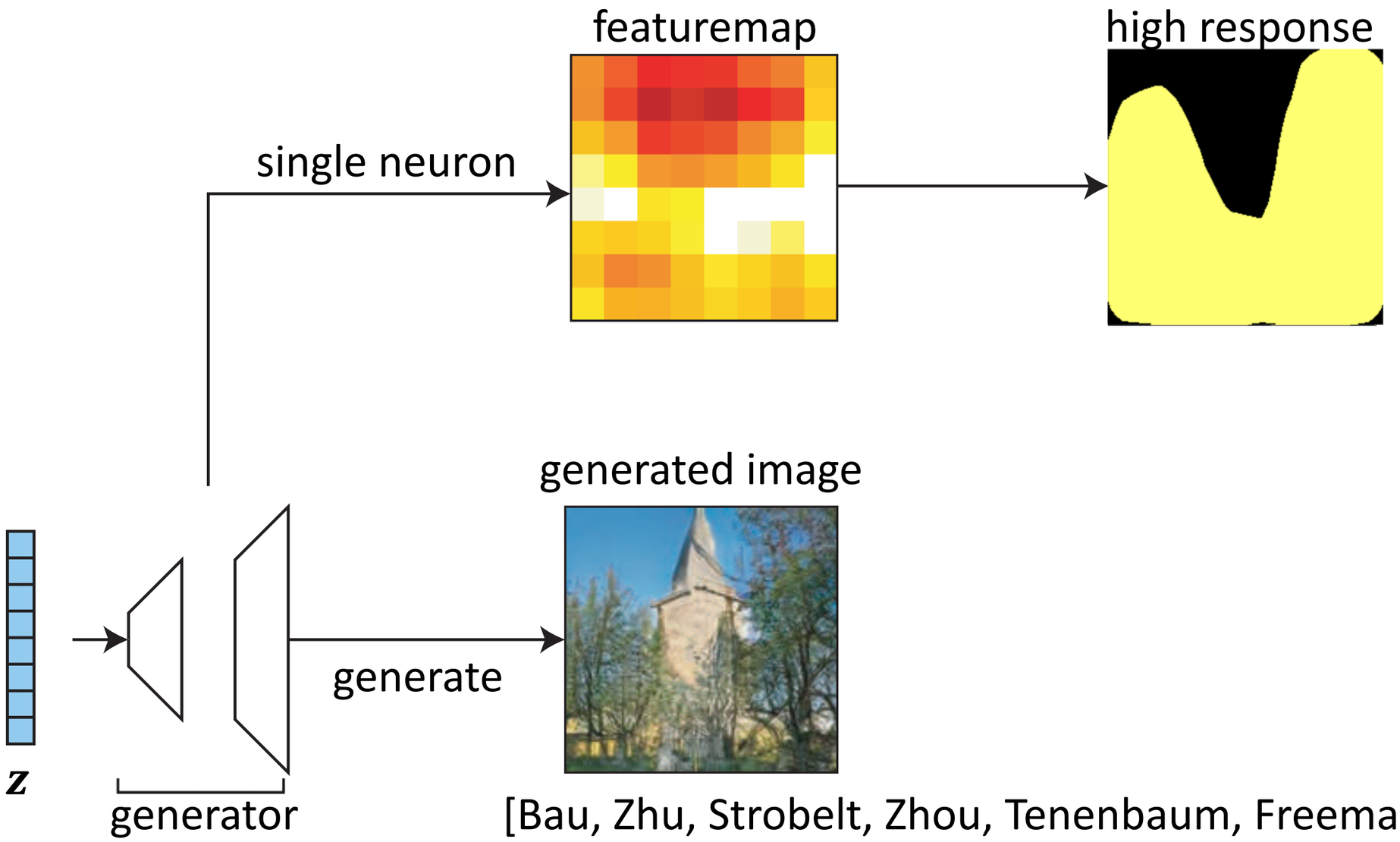
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **correlate** to an object class?



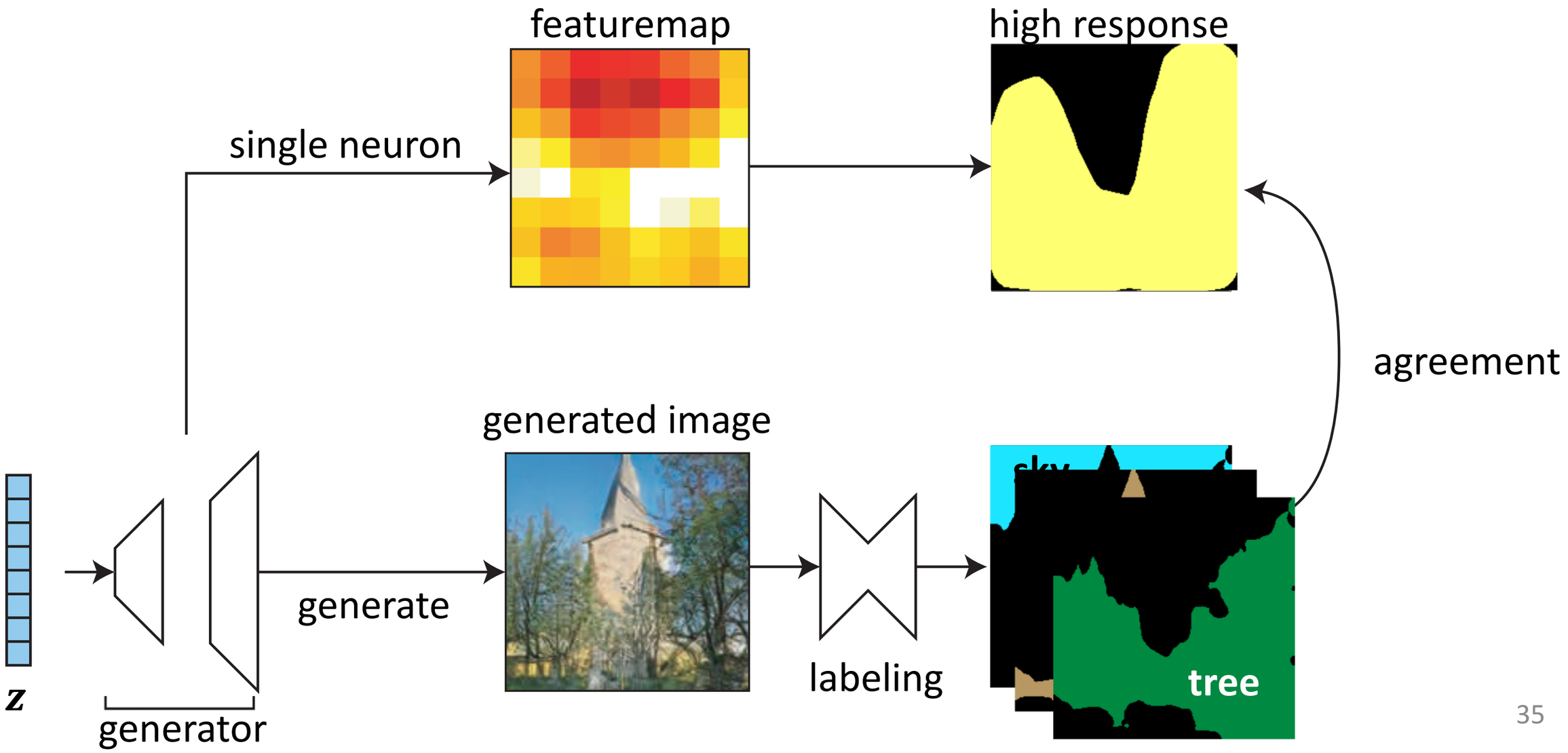
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **correlate** to an object class?



[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **correlate** to an object class?



Which neurons **correlate** to an object class?

Church samples

Tree Neuron

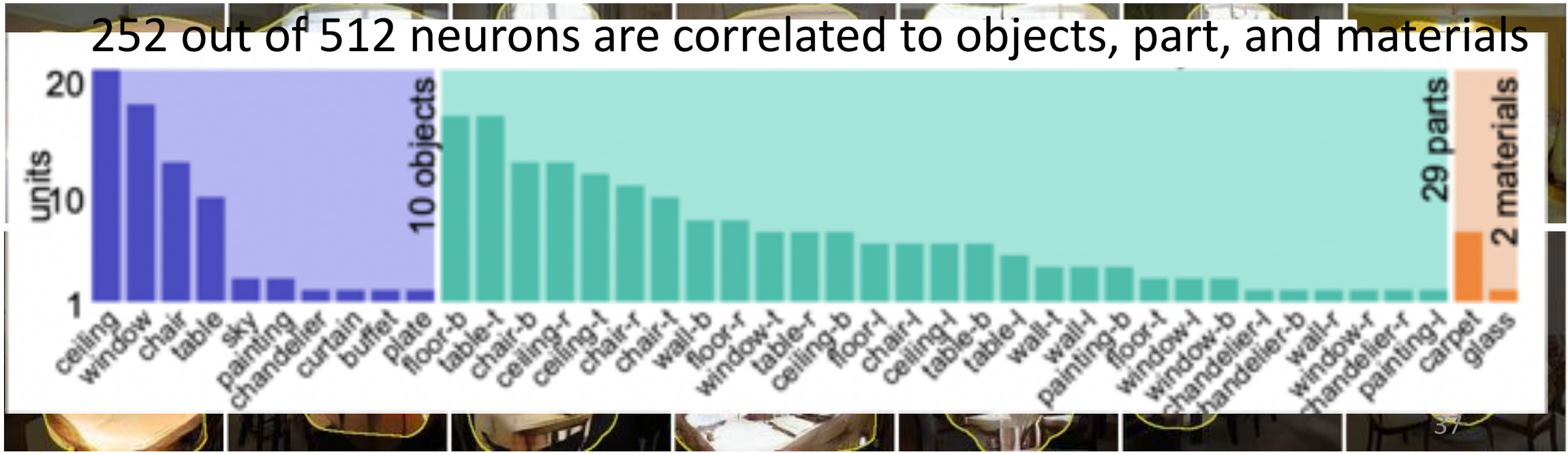
Dome Neuron

Which neurons **correlate** to an object class?

Dining room samples

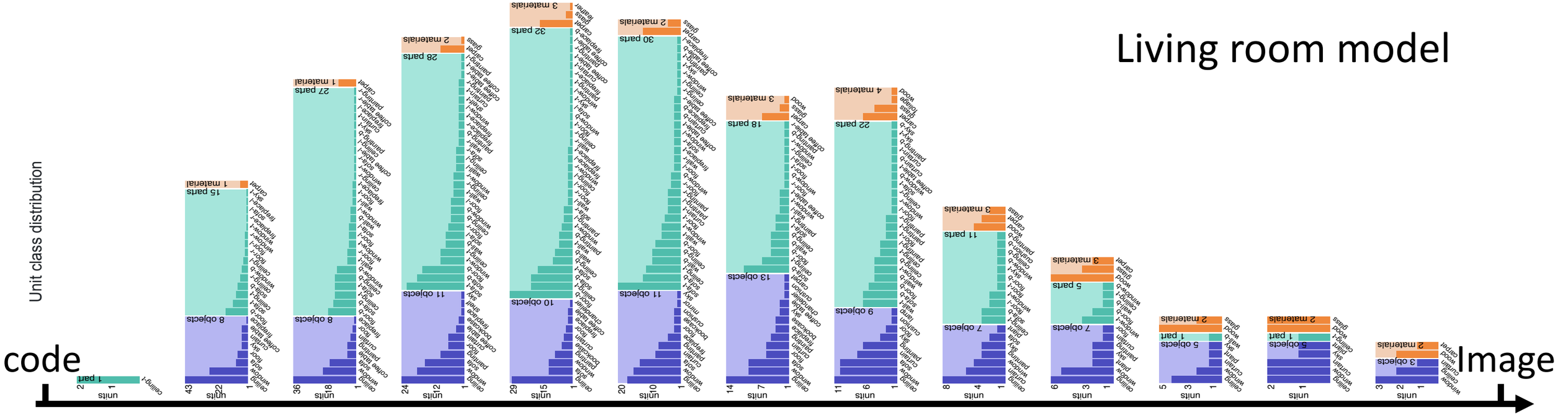
Window Neuron

Table Neuron



Which neurons **correlate** to an object class?

Living room model

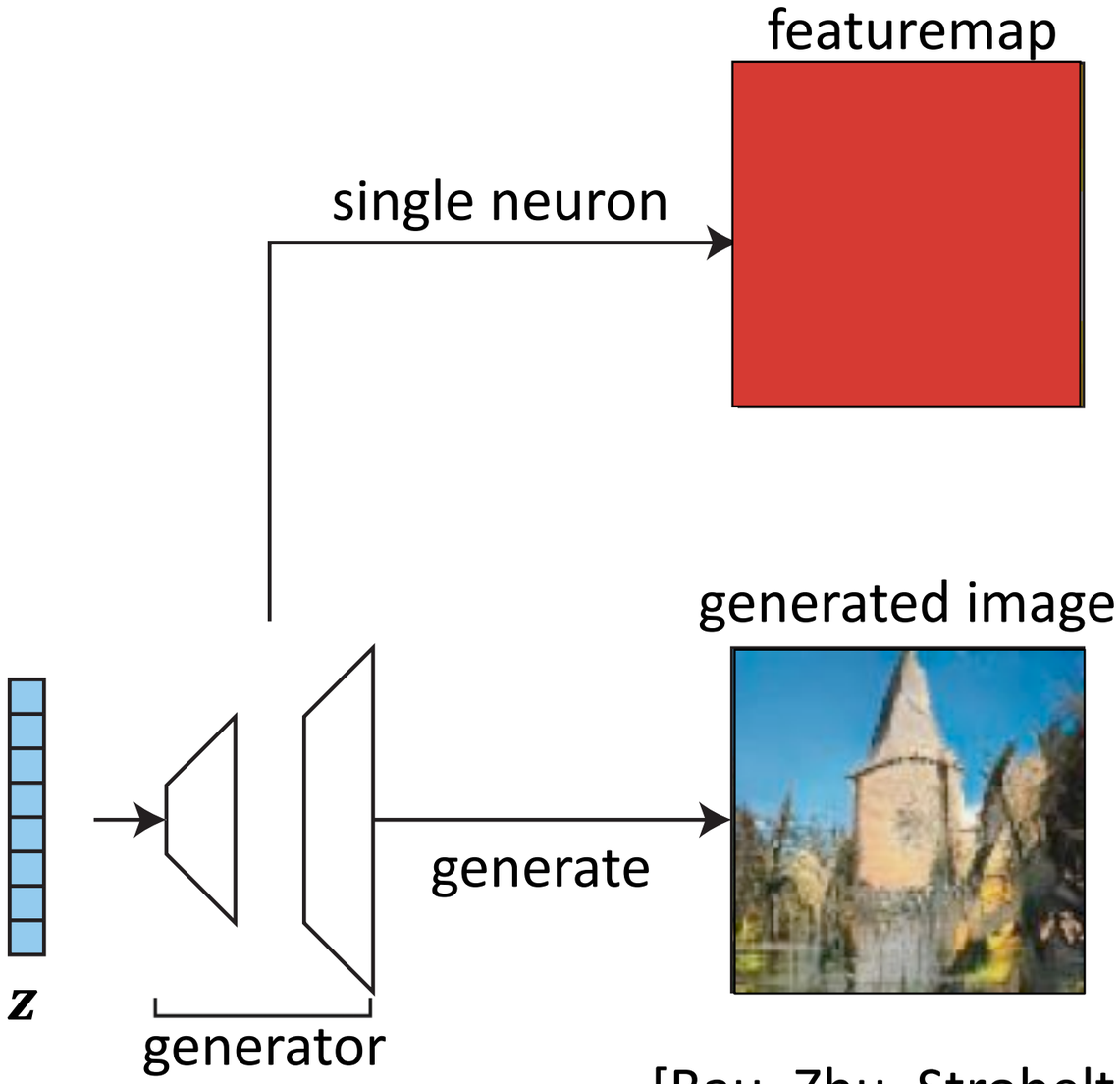


Layout

Object and parts

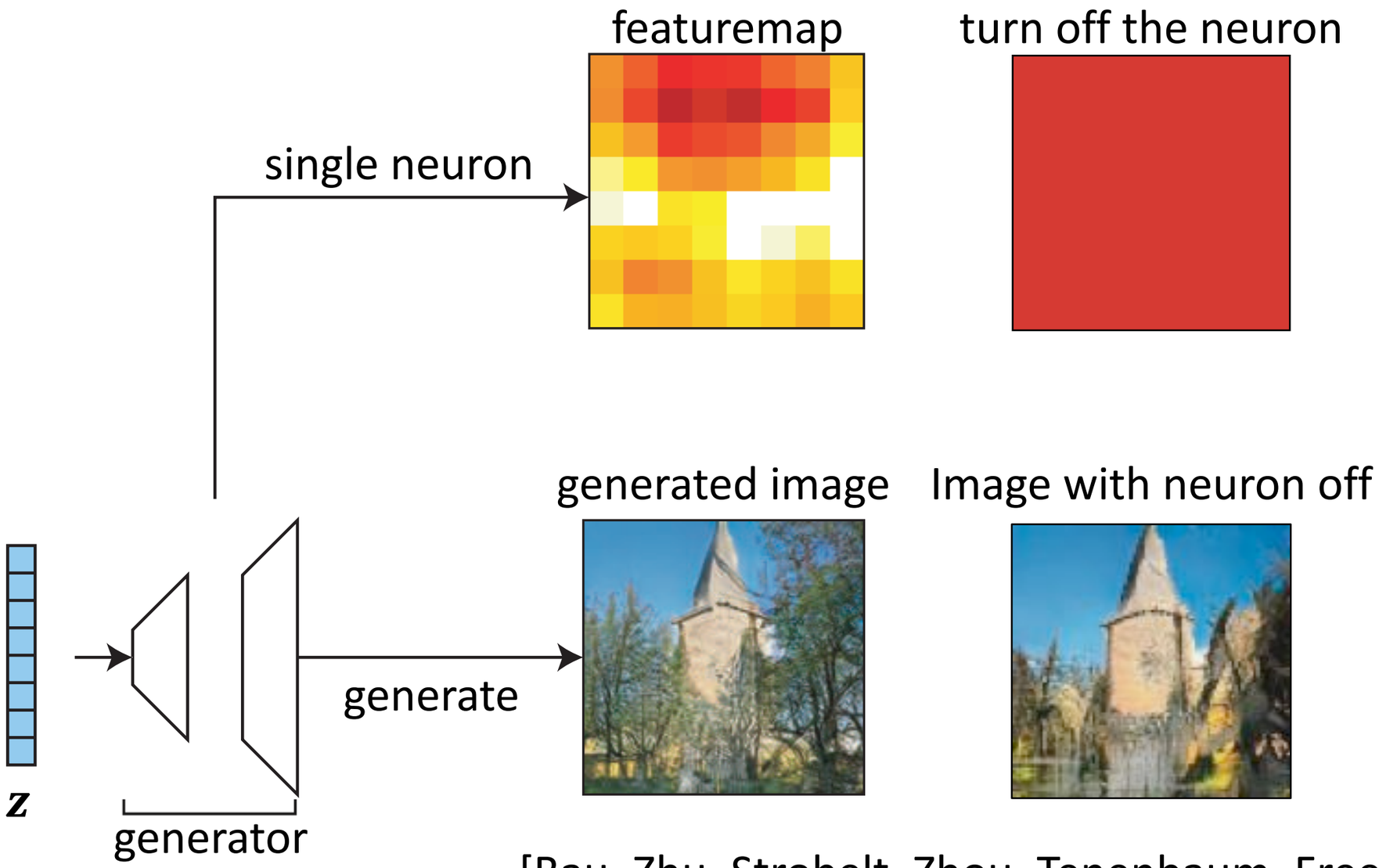
Edges, textures, local structure

Which neurons **cause** an object class?



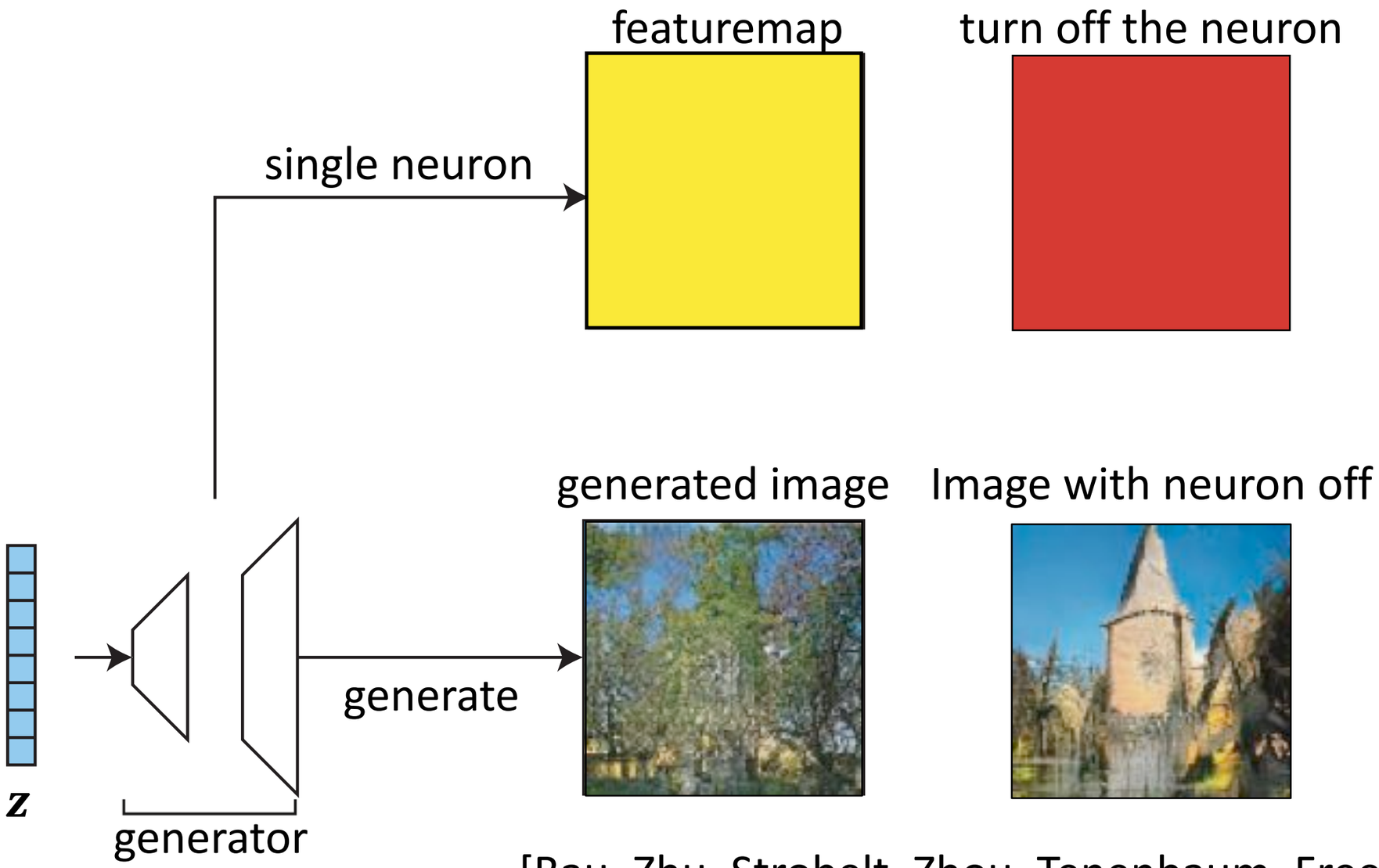
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **cause** an object class?



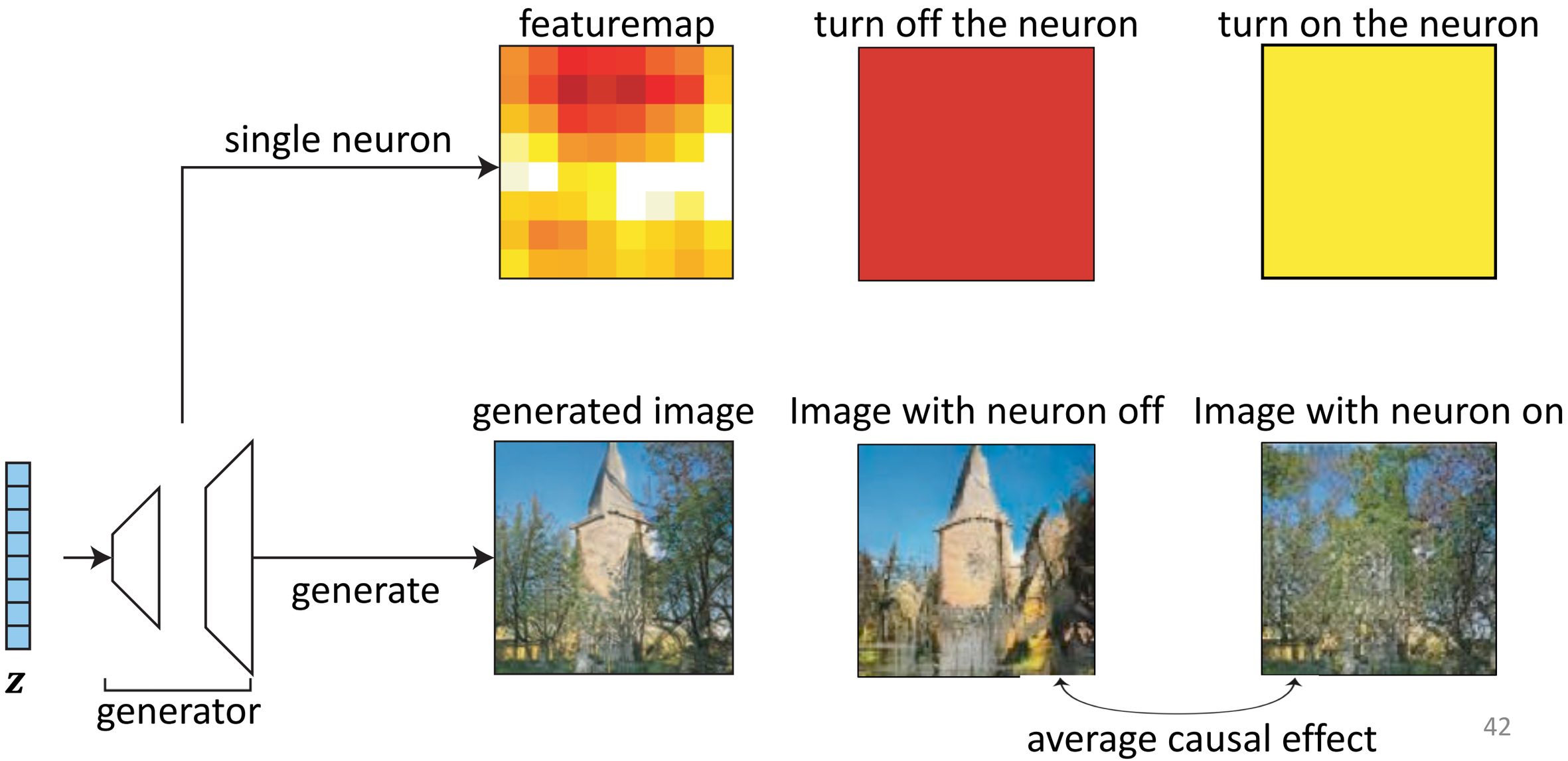
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **cause** an object class?



[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **cause** an object class?



Which neurons **cause** an object class?

Object-Scene Relationships

Turn off **person** neurons

Object-Scene Relationships

Turn off **window** neurons

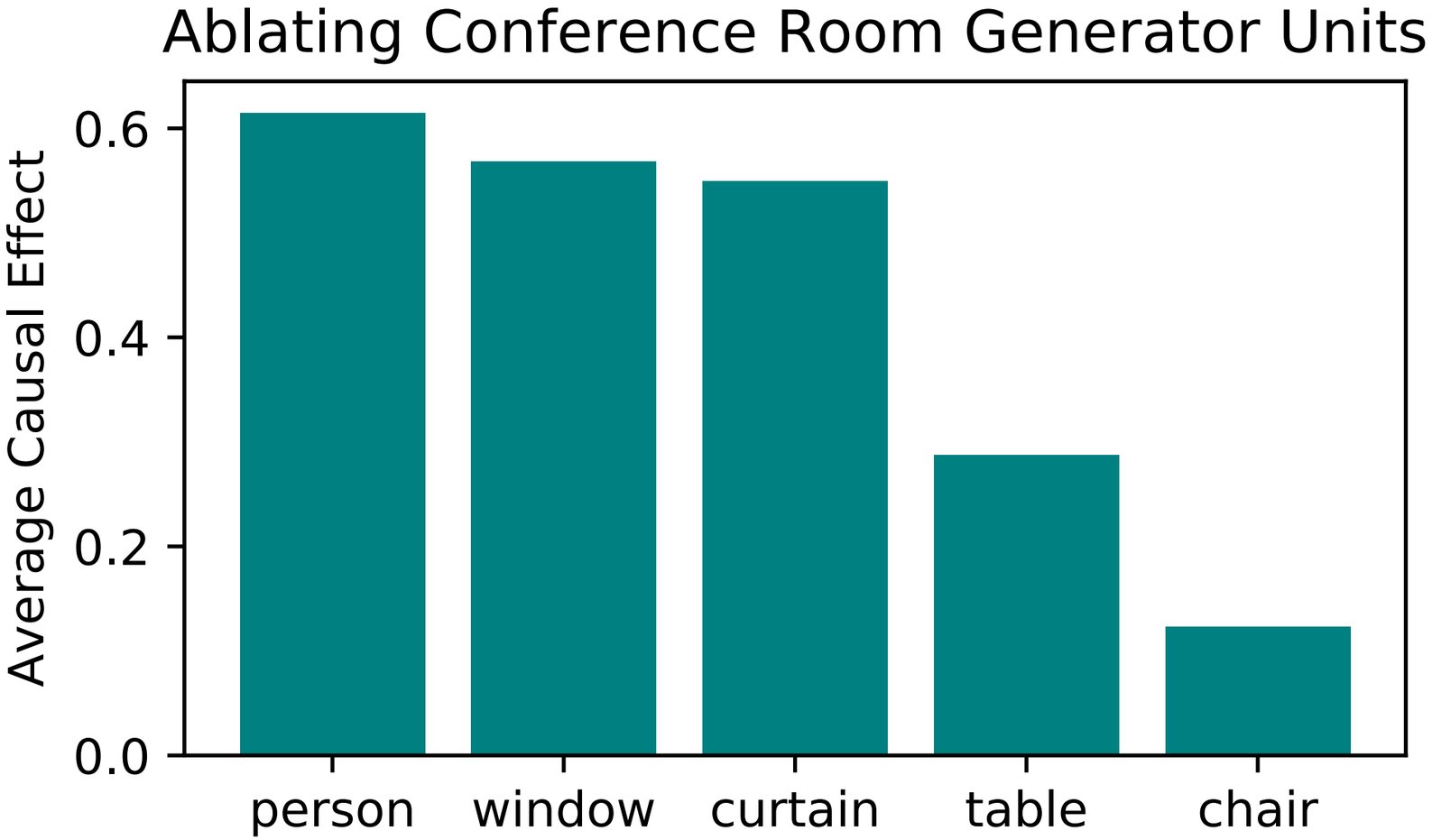
Object-Scene Relationships

Turn off **table** neurons

Object-Scene Relationships

Turn off **chair** neurons

Object-Scene Relationships



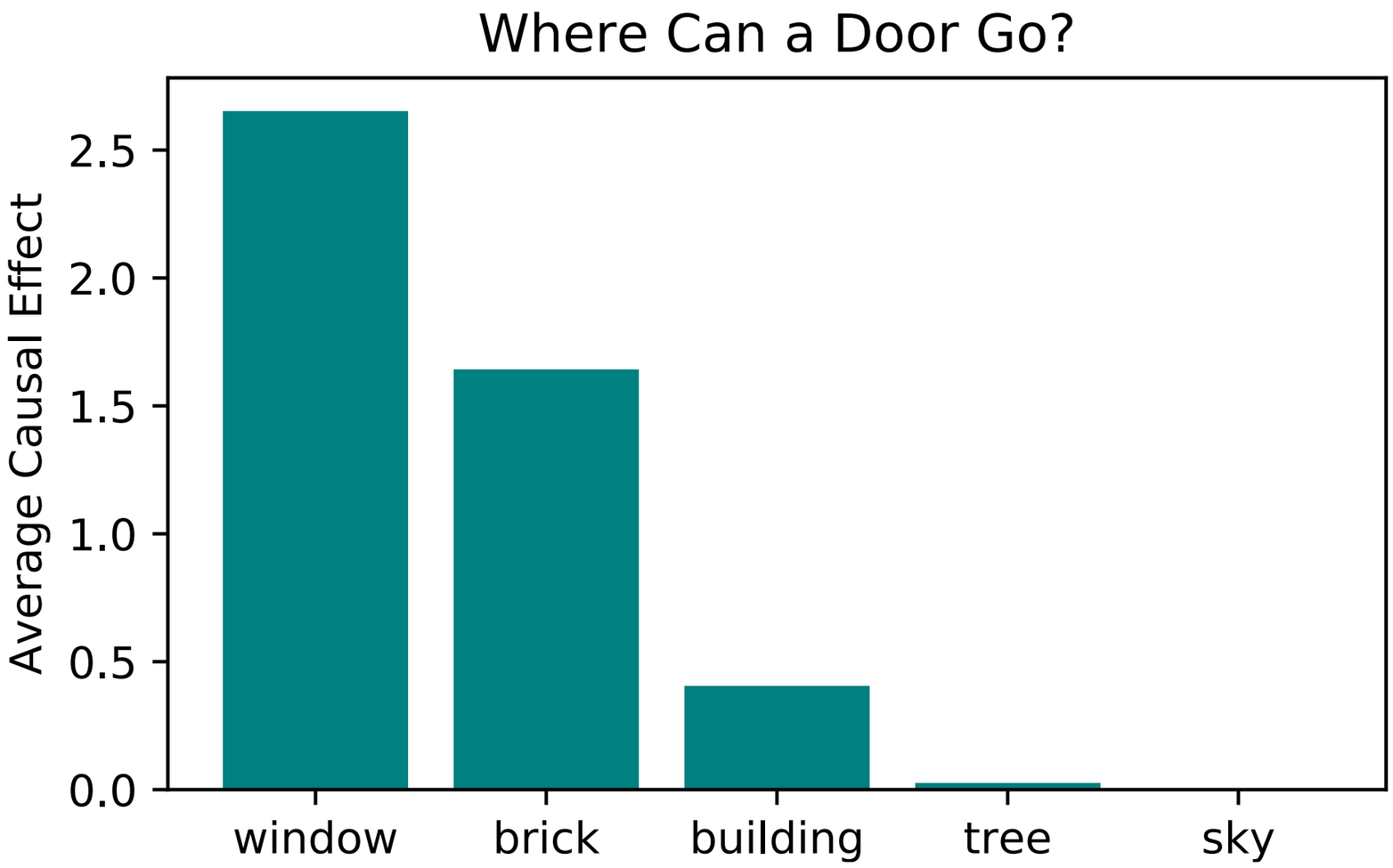
Object-Scene Relationships

Yellow box: highlight every location where we can insert doors

Object-Scene Relationships

Yellow box: highlight every location where we can insert doors

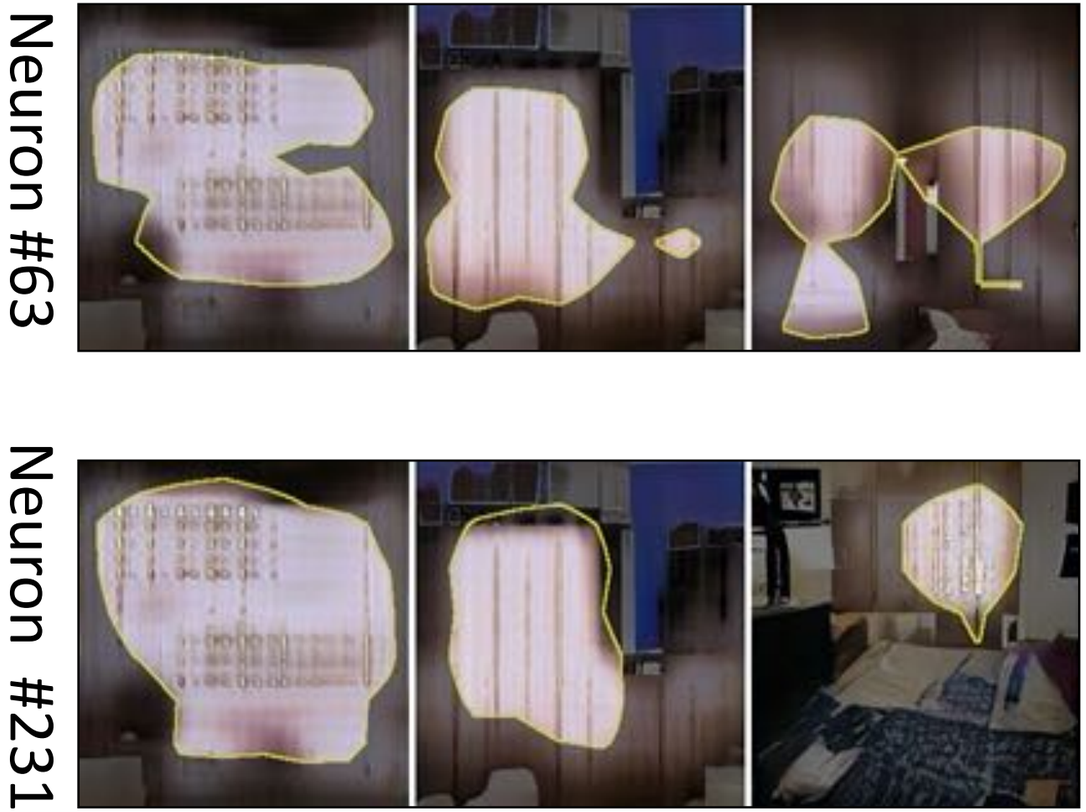
Object-Scene Relationships



Debugging and Improving Models

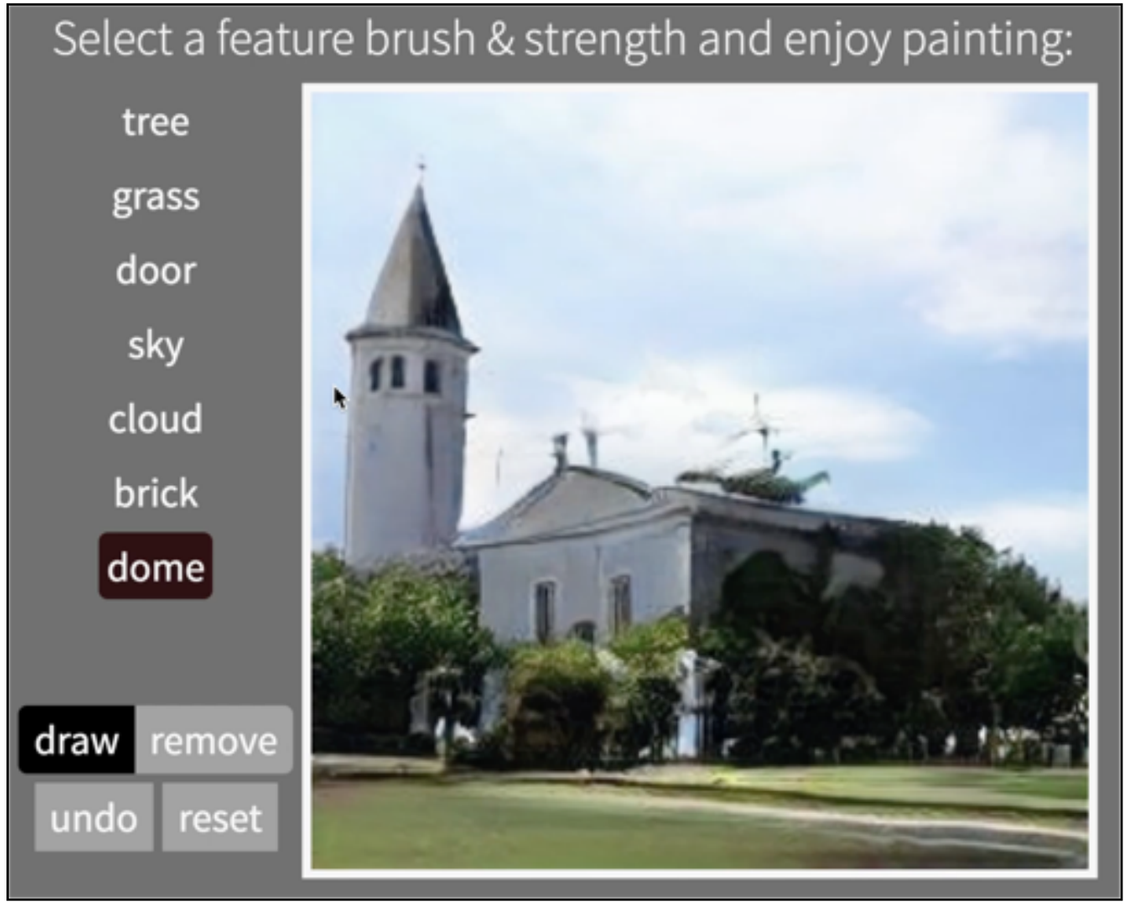
Turning off neurons with artifacts

Images with artifacts



Example artifact-causing neurons

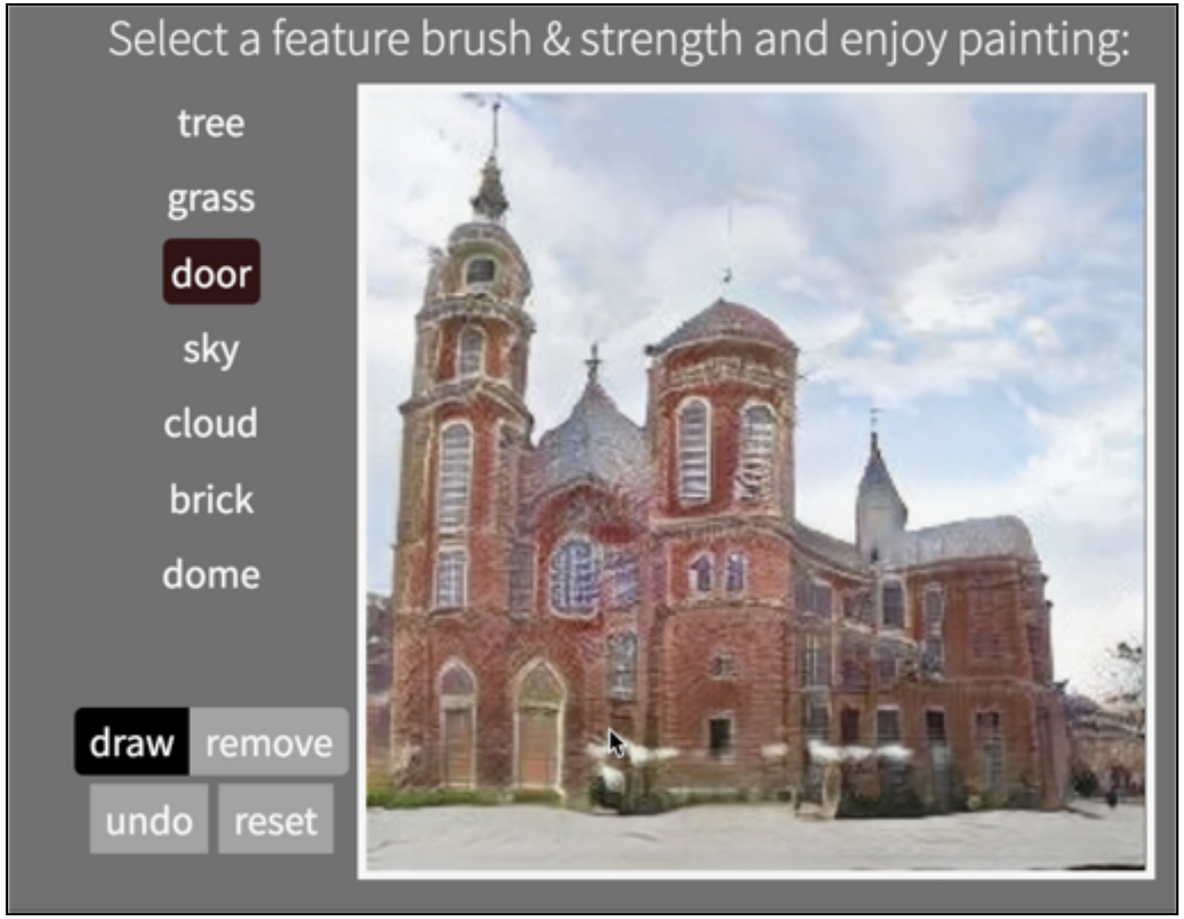
Interactive Painting



Online Demo

http://bit.ly/ganpaint

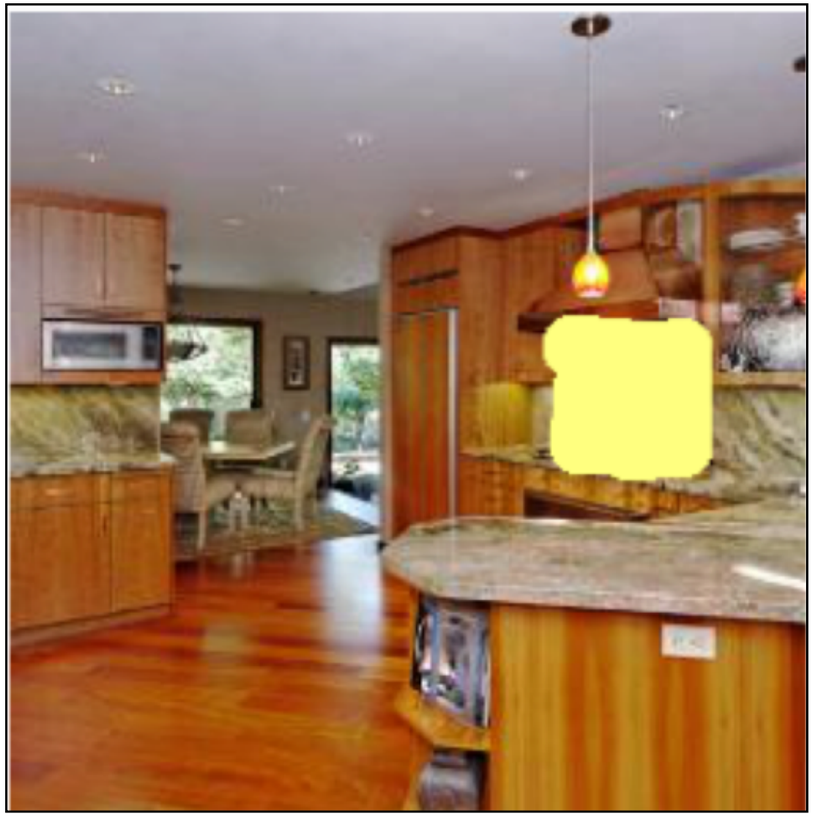
Interactive Painting



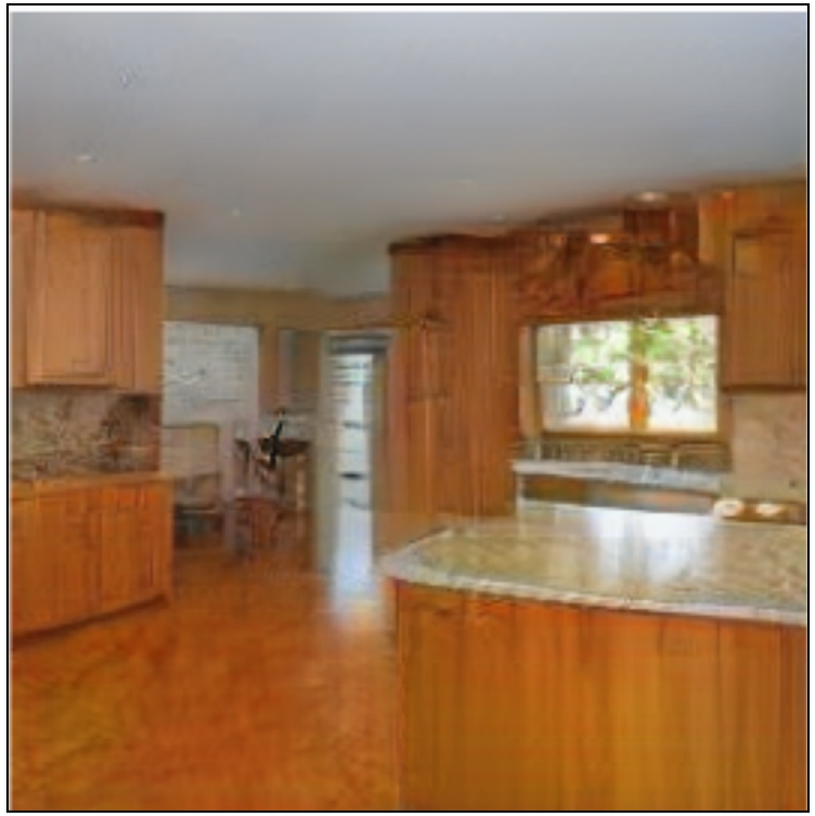
Online Demo

http://bit.ly/ganpaint

Manipulating a Real Photo



Original image + edits



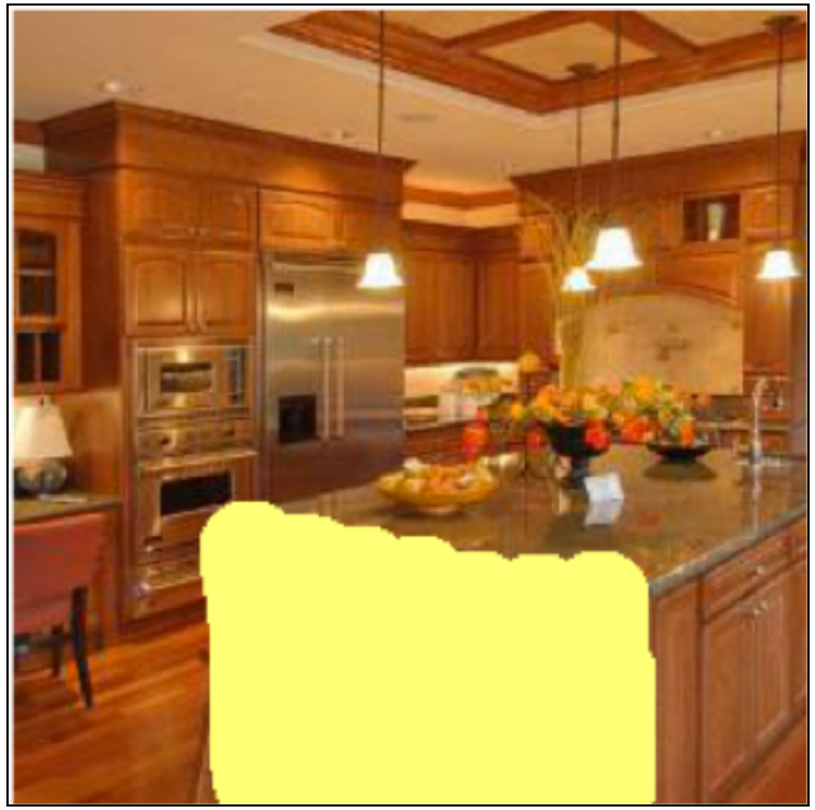
Editing with $\hat{\boldsymbol{z}}$

Editing with $\hat{\boldsymbol{z}}$ and $\widehat{\boldsymbol{\theta}}$

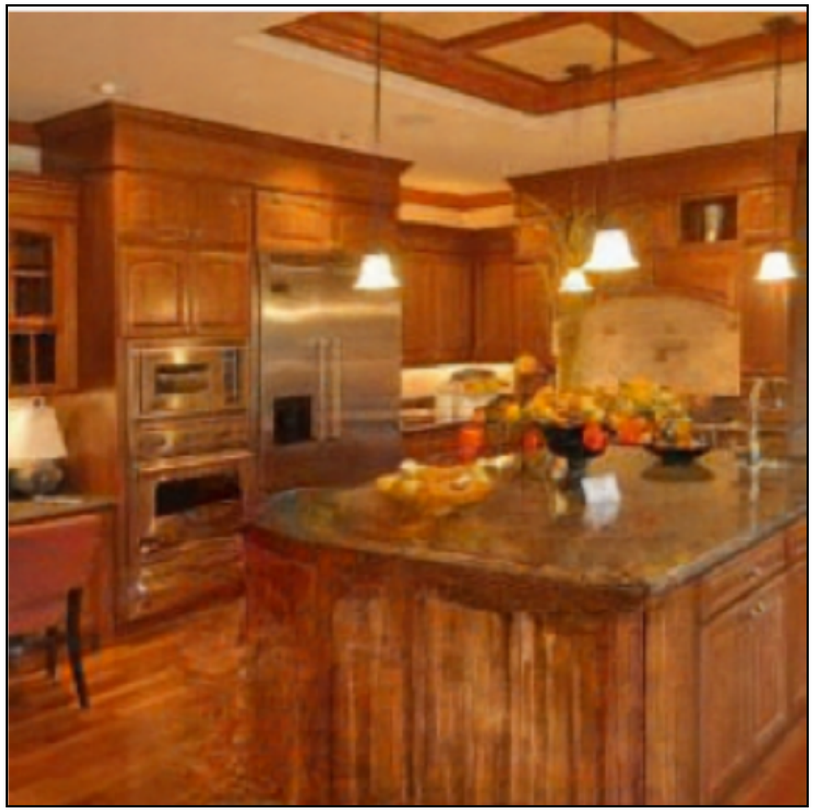
Semantic Photo Manipulation [Bau, Strobelt, Peebles, Wulff, Zhou, Zhu, Torralba, **SIGGRAPH** 2019]

Manipulating a Real Photo

Input image



Remove chairs



Output result

Semantic Photo Manipulation [Bau, Strobelt, Peebles, Wulff, Zhou, Zhu, Torralba, **SIGGRAPH** 2019]

Manipulating a Real Photo

Input image

Add windows

Output result

Semantic Photo Manipulation [Bau, Strobelt, Peebles, Wulff, Zhou, Zhu, Torralba, **SIGGRAPH** 2019]

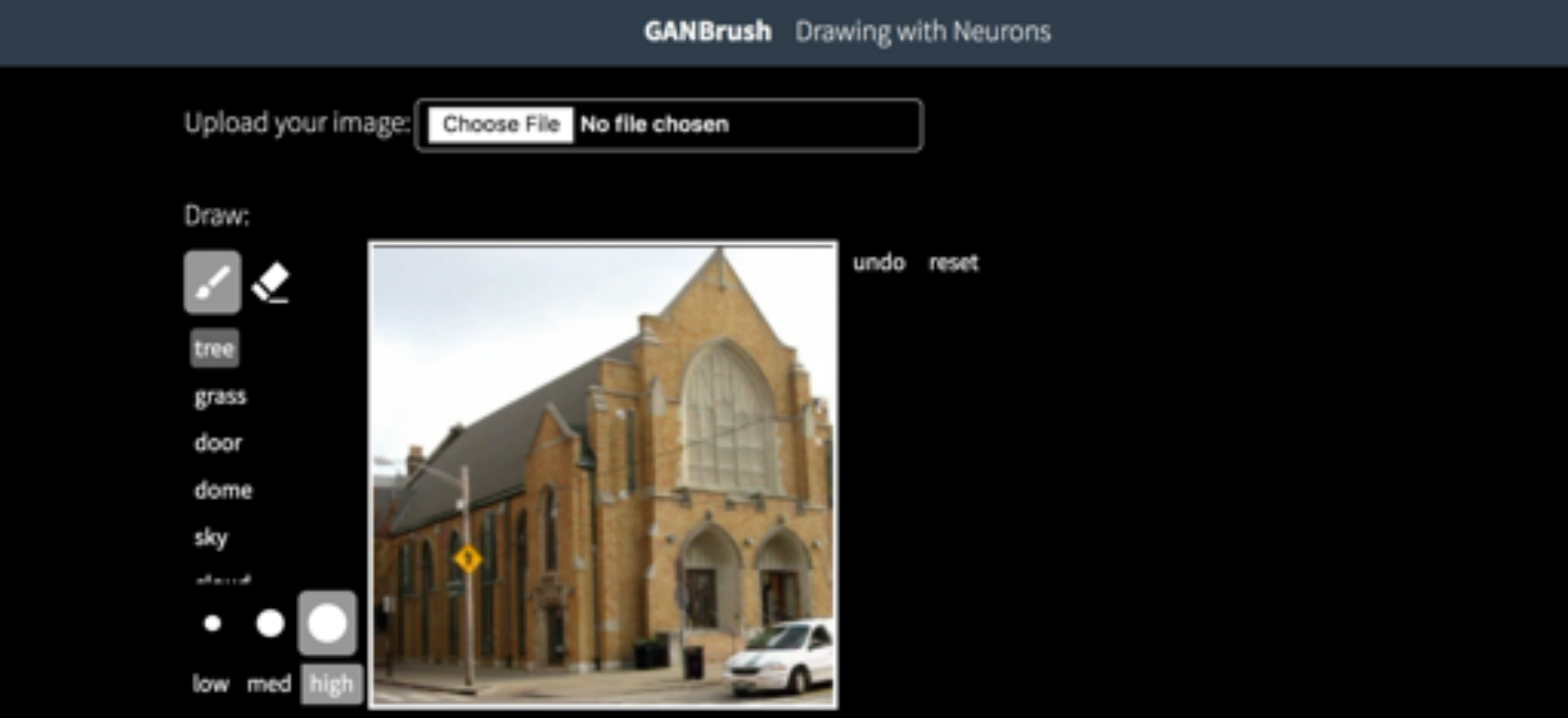
Manipulating a Real Photo via GAN Dissection

Input image

Restyle trees for spring

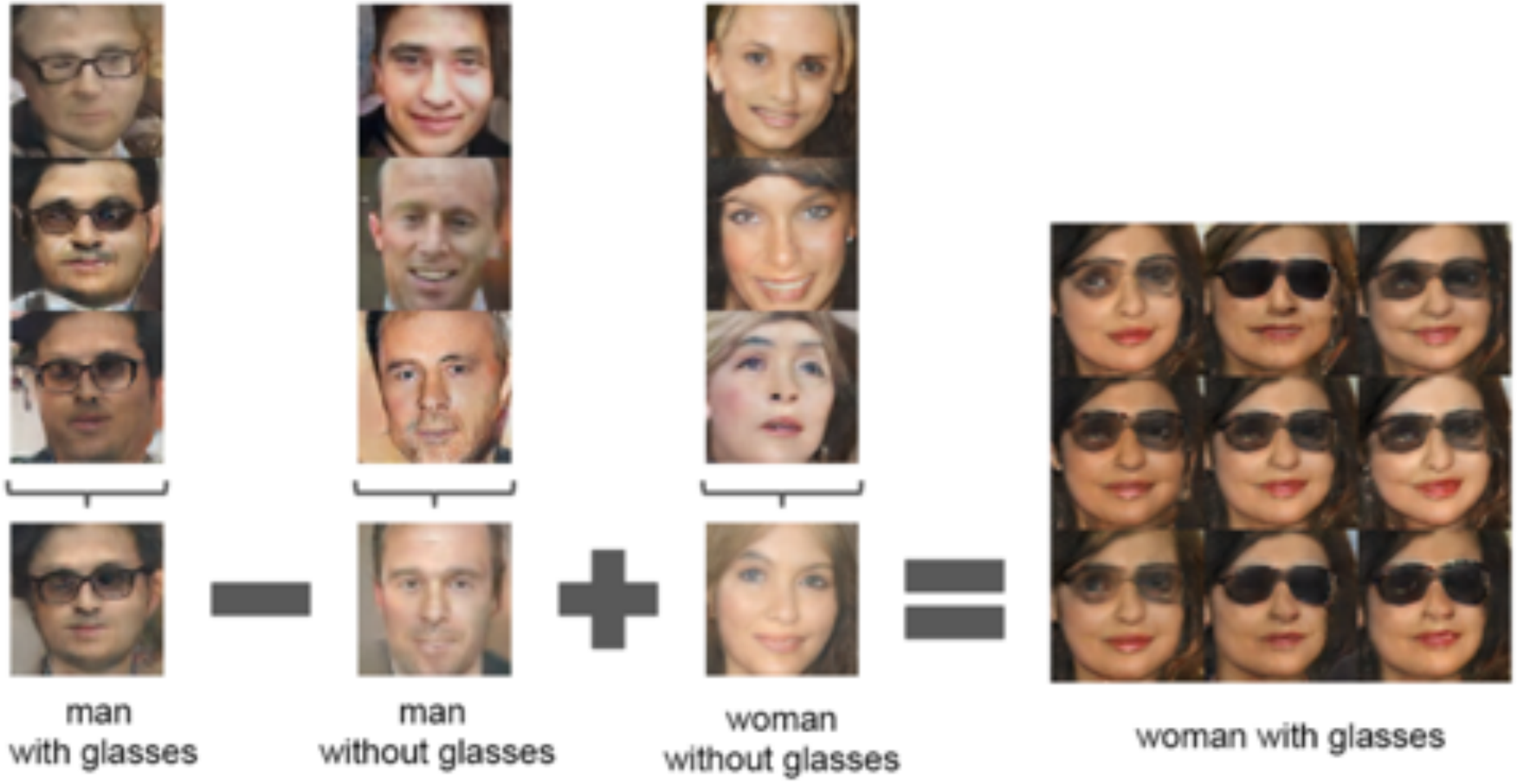
Restyle trees for autumn

Semantic Photo Manipulation [Bau, Strobelt, Peebles, Wulff, Zhou, Zhu, Torralba, **SIGGRAPH** 2019]


GANBrush Drawing with Neurons
Upload your image:
Choose File
No file chosen
Draw:
tree
grass
door
dome
sky
low
med
high
undo
reset

Manipulating Latent code/layer (computing directions offline)

Compute Δz



First annotate images, then compute directions

DCGAN [Radford et al. 2016]

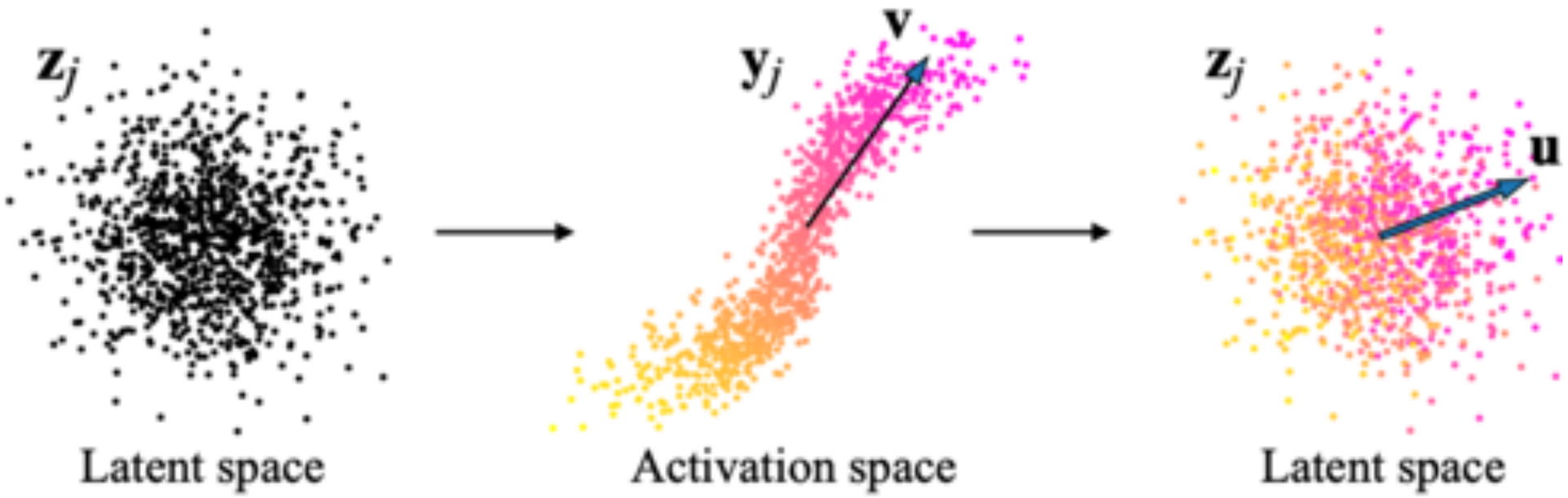
Manipulating Latent code/layer (PCA directions)

GANSpace: Discovering PCA directions

First compute potential directions (PCA), then annotate directions

GANspace [Härkönen et al. 2020]

GANSpace: Discovering PCA directions



Also see “Editing in Style: Uncovering the Local Semantics of GANs”, Collins et al., CVPR 2020
“Closed-Form Factorization of Latent Semantics in GANs”, Shen and Zhou. CVPR 2021

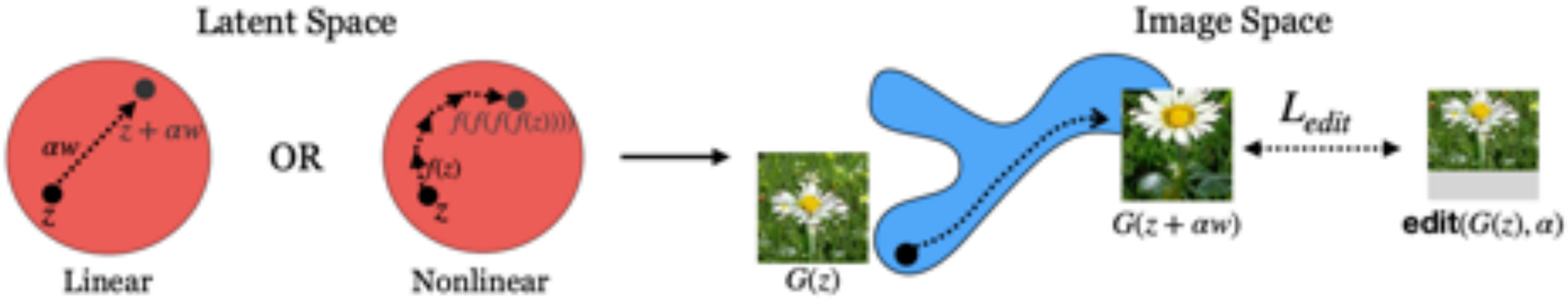
GANspace [Härkönen et al., 2020]

GANSpace: Discovering PCA directions

GANspace [Härkönen et al., 2020]

Manipulating Latent code/layer (offline optimization)

Offline optimization



Given a pre-defined function **edit** and a pre-trained generator **G**

Linear case:
(w is a vector)

$$\arg\min_{w} \mathbb{E}_{z,\alpha}[\mathcal{L}(G(z+\alpha w), \texttt{edit}(G(z), \alpha))]$$

Non-linear case:
(f is a function)
apply it n times

$$\arg\min_{f} \mathbb{E}_{z,n}[||G(f^n(z)) - \texttt{edit}(G(z), n\epsilon))||],$$

[Jahanian*, Chai*, Isola. 2020]

Offline optimization

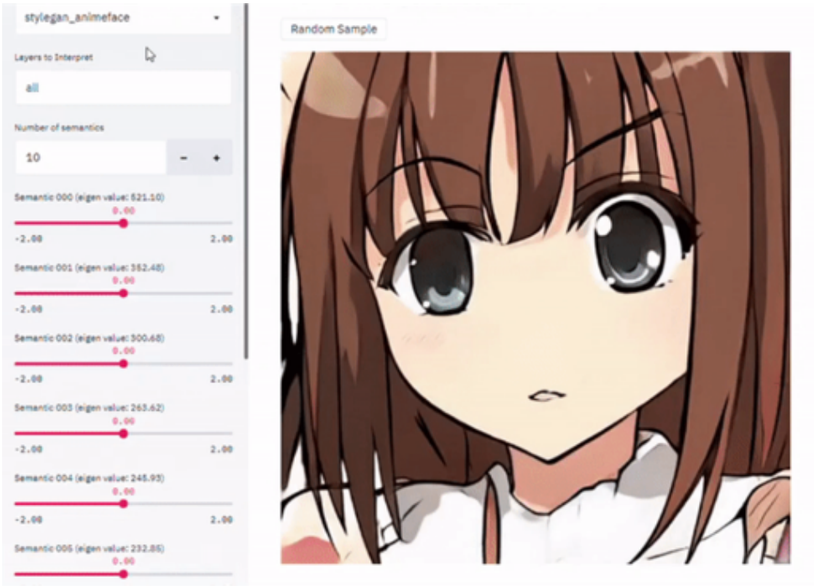
Requirement**:** A known **edit** function

[Jahanian*, Chai*, Isola. 2020]

Different Ways of Using Networks

- Train a network to produce images (instead of hand-crafted filters)
  - Image-to-Image Translation, Fast Neural Style Transfer, Image Super-resolution
- Define a Loss function based on a network (instead of pixel loss)
  - Perceptual Loss, Adversarial Loss, Contrastive Learning loss
- Using networks' features (instead of pixels, edges, or wavelets)
  - Gram matrix, Deep Image Analogy
- Optimizing the latent code of a generative model (instead of raw pixels)
  - GAN Projection (iGAN, Image2StyleGAN), Latent vector editing
- Optimizing the weights of a network
  - Deep Image Prior, GANPaint

Thank You!



16-726, Spring 2021

https://learning-image-synthesis.github.io/

video © [Shen and Zhou, CVPR 2021]