Image Editing with Optimization (part II)

Jun-Yan Zhu

16-726, Spring 2023

Image Editing with Optimization

input photo

Image Editing Program

User Input

result

Desired output:

- stay close to the input.
- satisfy user's constraint.

Photo: [Tao et al. 2014]

Image Editing with Optimization

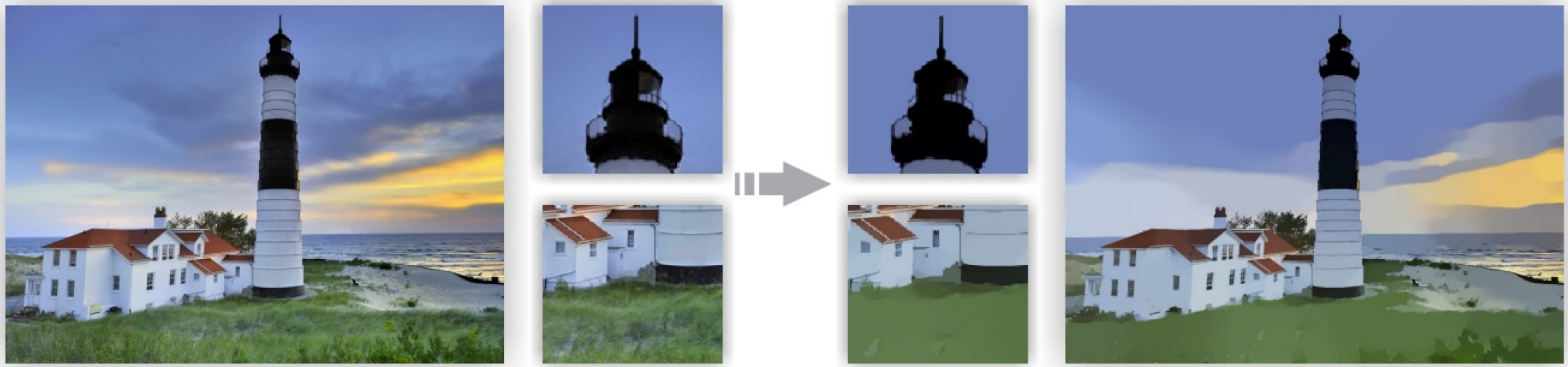


Image Smoothing via L0 Gradient Minimization [Xu et al., SIGGRAPH Asia 2011]

$$\arg\min_{\hat{y}}\{||\underbrace{\hat{y}}_{\text{output}} - \underbrace{x}_{\text{input}}|| + \lambda \underbrace{C}_{\text{L0 norm on image gradients}}(\hat{y})\}$$

(the total number of nonzero elements)

Things can get really bad

Image Warping

Image Composition

The lack of “safety wheels”

Adding the "safety wheels"

Input Photo

Image Editing Program

User Input

Output Result

Natural Image Manifold

A desired output:

- stay close to the input.
- satisfy user's constraint.
- Lie on the natural image manifold

Changing Variables

- Traditional method: Optimizing the image

$$\hat{y}^* = \arg\min_{\hat{y}} \mathcal{L}(\underset{\text{input}}{x}, \overset{\text{user constraint}}{y}, \underset{\text{output}}{\hat{y}})$$

- New method: Optimizing the latent code

$$z^* = \arg\min_{z} \mathcal{L}(\underset{\text{input}}{x}, \overset{\text{user constraint}}{y}, \underset{\text{Generator}}{G}(\underset{\text{Latent code}}{z}))$$

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

Baseline

- Baseline: Optimizing the latent code

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

z^* and z_0 are used interchangeably

Find the Differences…

Original image

GAN reconstructed image

Baseline

- Baseline: Optimizing the latent code

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

How to Improve GANs Projection

- Baseline: Optimizing the latent code

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

- Generator fine-tuning:

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

Generator Fine-tuning (Progressive GANs)

Original image

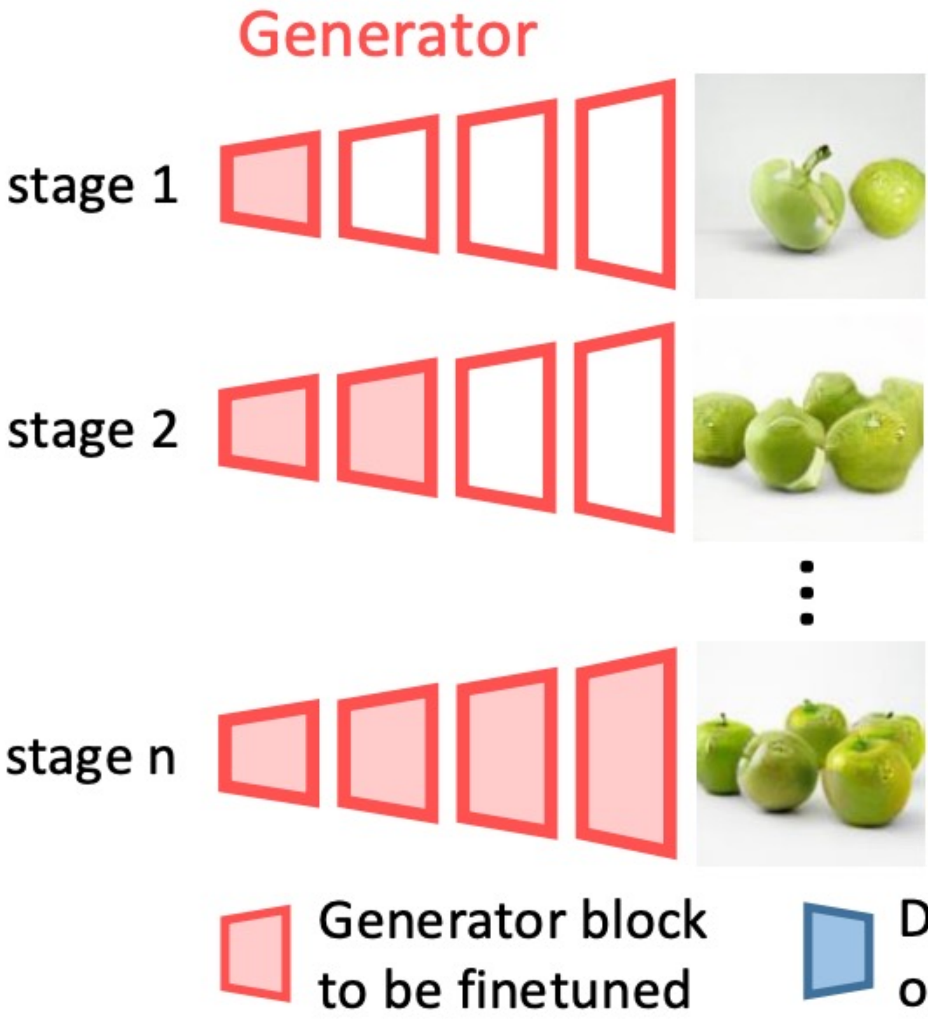
With z^*

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.]

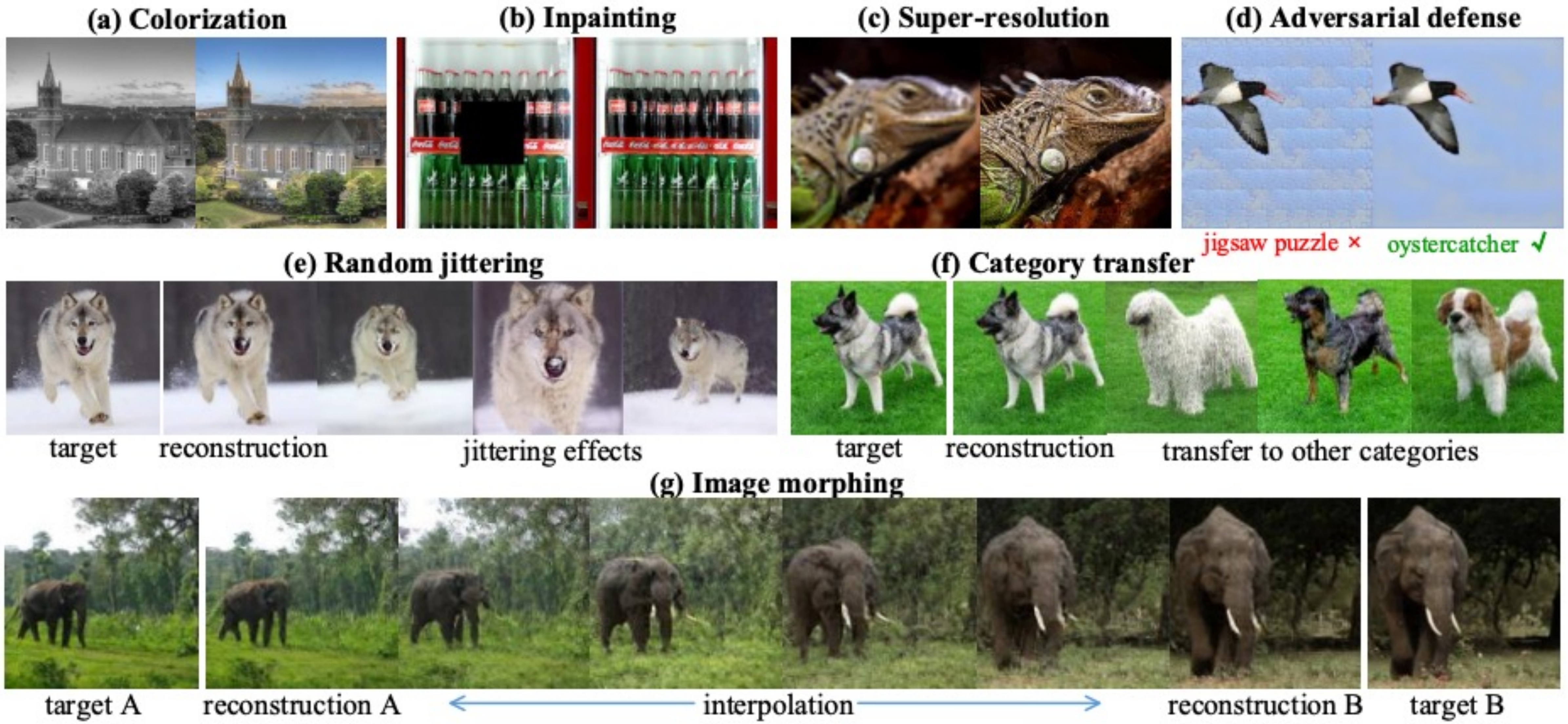
Generator Fine-tuning (BigGAN)



Progressive Reconstruction
- First match semantics
- Then match color and textures

Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation
[Pan et al., ECCV 2020]

Generator Fine-tuning (BigGAN)



Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation
[Pan et al., ECCV 2020]

How to Improve GANs Projection

- Baseline: Optimizing the latent code

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

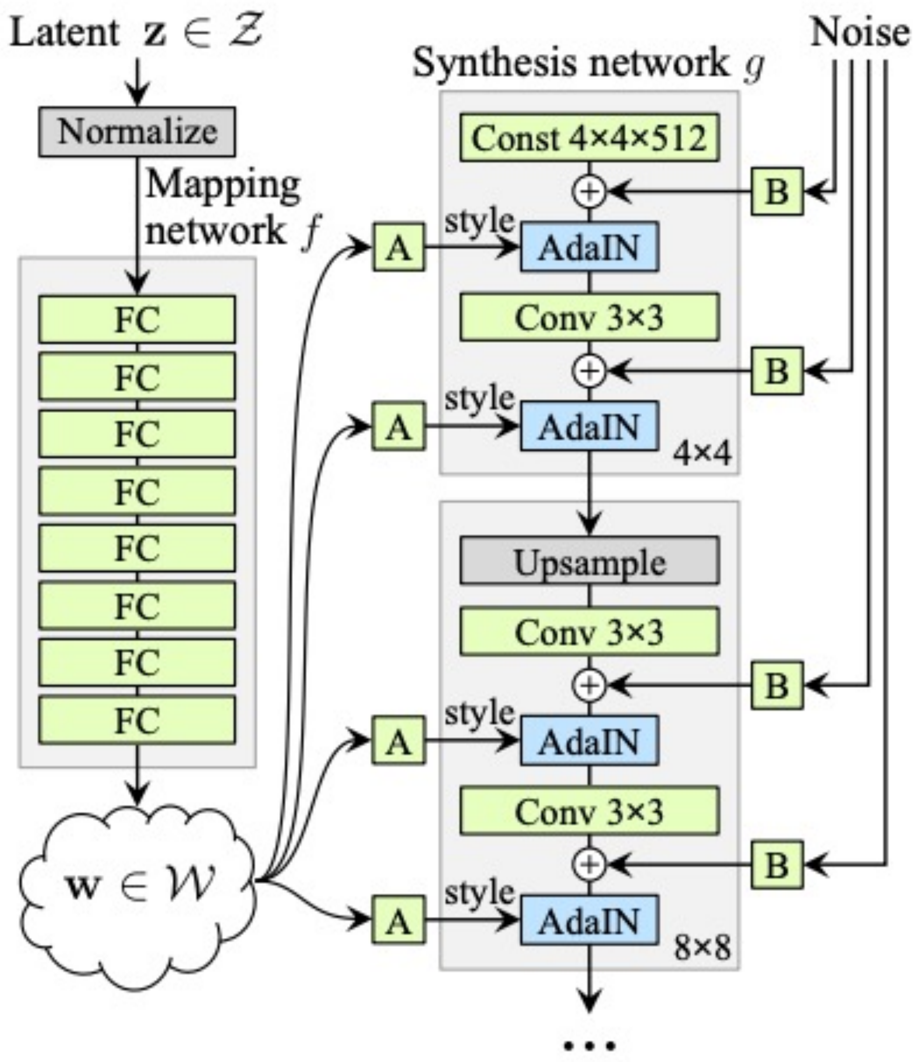
- Generator fine-tuning:

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

- Optimizing intermediate features

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

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

Input

Reconstruction

StyleGAN2 [Karras et al., 2019]

Using Different Layers: w+ Space

All the results are reconstructed via the StyleGAN Face model.

Image2StyleGAN [Abdal et al., 2019]

How to Improve GANs Projection

- Baseline: Optimizing the latent code

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

- Generator fine-tuning:

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

- Optimizing intermediate features

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

Used together

Generator Fine-tuning with w+ Space

Pivotal Tuning for Latent-based Editing of Real Images [Roich et al., 2021]

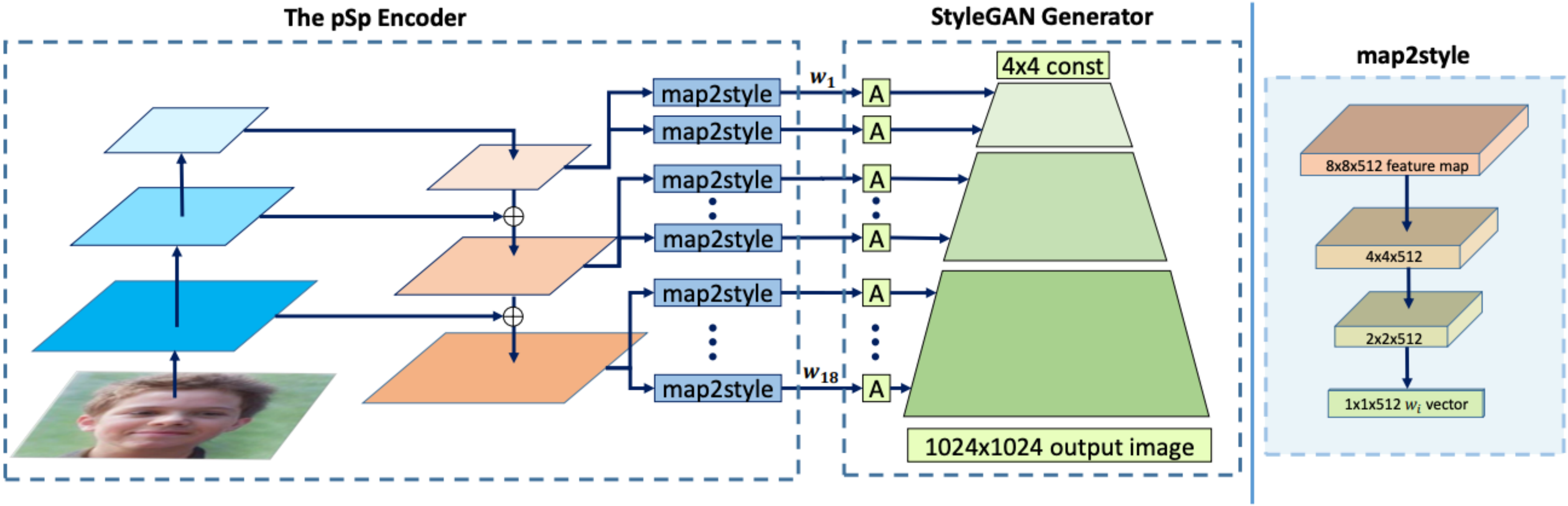
How to Improve GANs Projection

- Baseline: Optimizing the latent code

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

- Training an encoder E(x). Advantages?
  - Faster inference
  - More reliable initialization
- Encoder design depends on
  - Generator architecture.
  - Which latent space: z, w, w+.
  - Pre-trained network weights.

Example: An StyleGAN Encoder



Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation
[Richardson et al., CVPR 2021]

Example: An StyleGAN Encoder

Input $\mathcal{W}$ Naive $\mathcal{W}+$ pSp

Debugging GANs Projection (HW5)

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

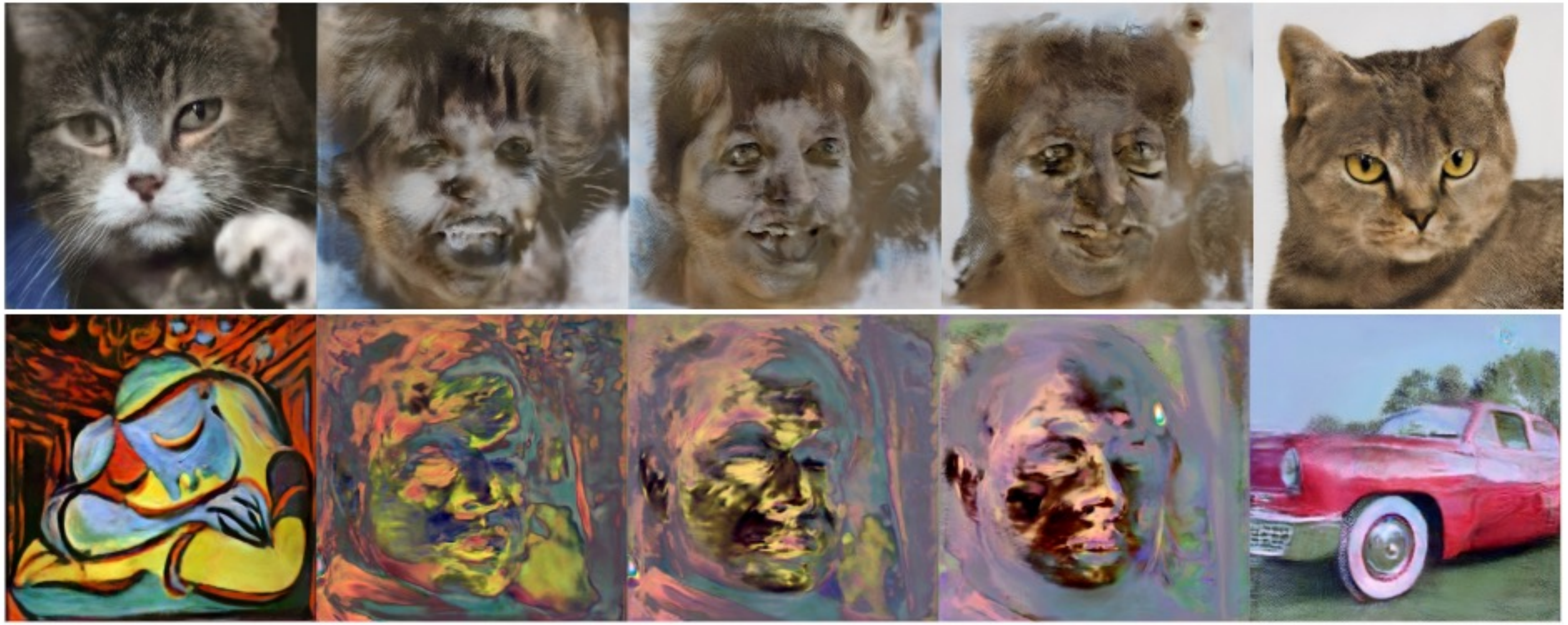
- What can go wrong?
  - Generator: G (cannot generate the image or too deep)
  - Reconstruction loss: L (not a good image distance)
  - Optimization method: SGD, ADAM (local minimum)

    (1) use a more advanced solver: e.g., L-BFGS (Quasi-Newton)

    (2) train an encoder to initialize the latent code. E(x)

- Debugging steps:
  - Reconstruct a generated image
  - Reconstruct a training set real image
  - Reconstruct a validation/test set real image
  - Reconstruct an in-the-wild image (e.g., Internet photo, camera roll)

Reconstruction ≠ Editing



Interpolations between two images

Image2StyleGAN [Abdal et al., 2019]

Image Editing with GANs

- Step 1: Image Projection/Reconstruction

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

- Step 2: Manipulating the latent code

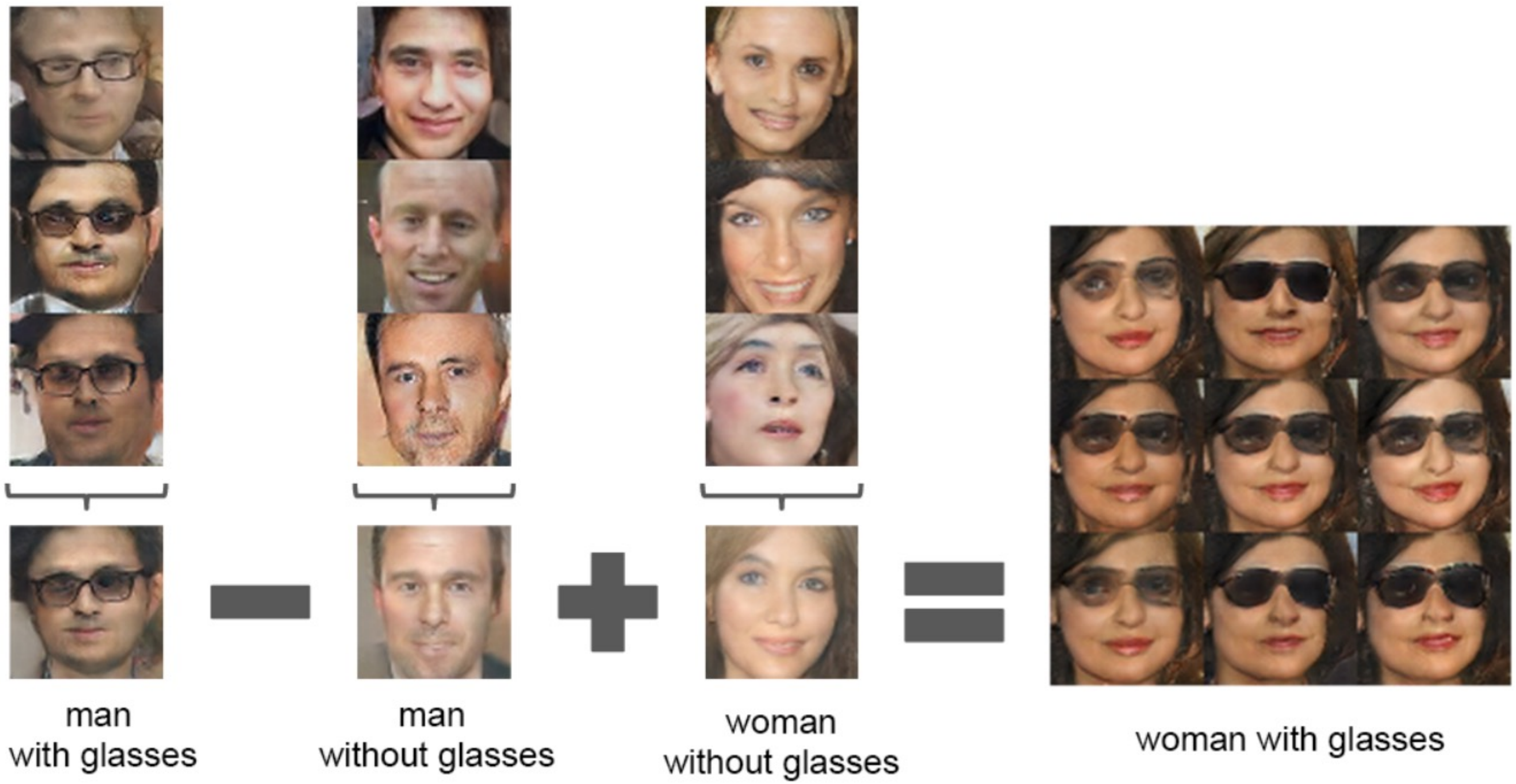
$$z_1 = z_0 + \Delta z$$

- Step 3: Generate the edited result

$$G(z_1)$$

Manipulating Latent code/layer (computing directions offline)

Compute Δz



Step 1: annotate images (manually or via a pre-trained classifier)

Step 2: compute directions

DCGAN [Radford et al. 2016]

Manipulating Latent code/layer (PCA directions)

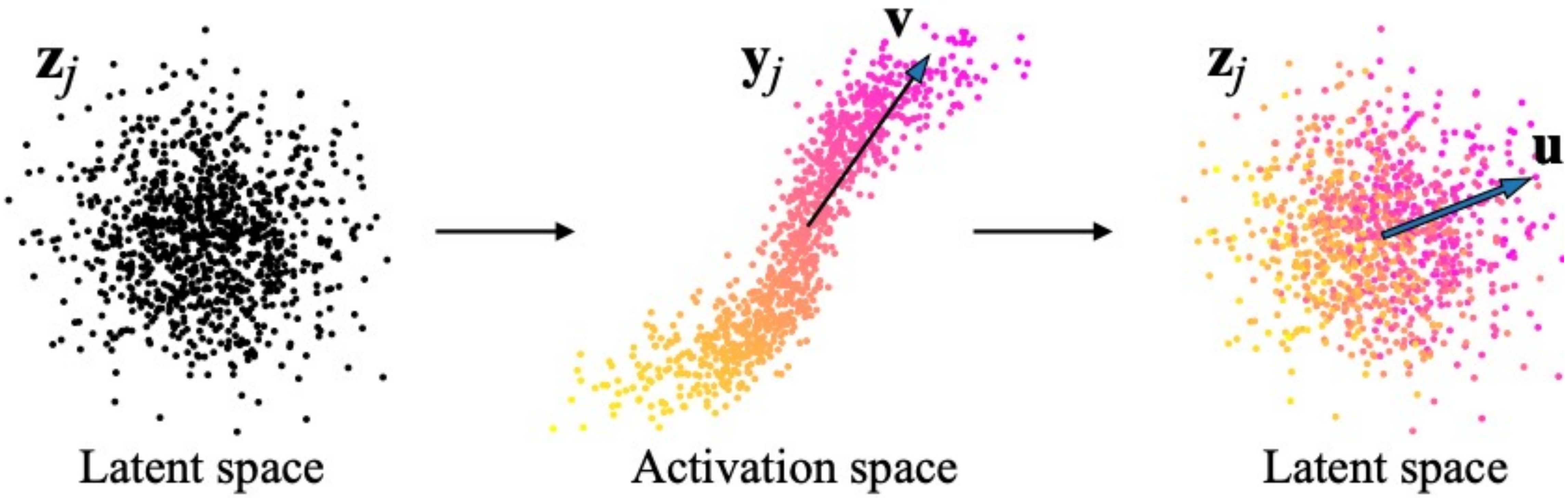
GANSpace: Discovering PCA directions

First find compute potential directions (PCA), then name them

GANspace [Härkönen et al. 2020]

GANSpace: Discovering PCA directions

z: latent codes. y: intermediate features.
v: PCA direction in feature space , u: PCA direction in latent space



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

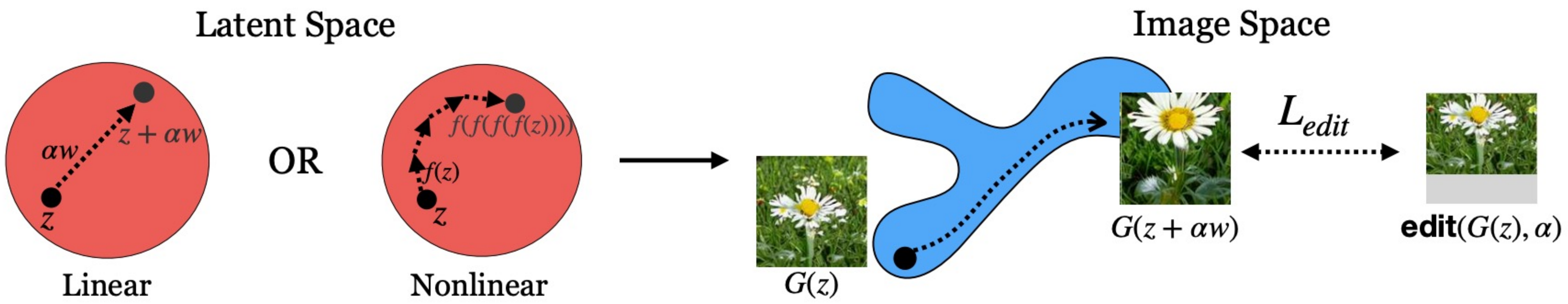
GANspace [Härkönen et al., 2020]

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)

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

strength

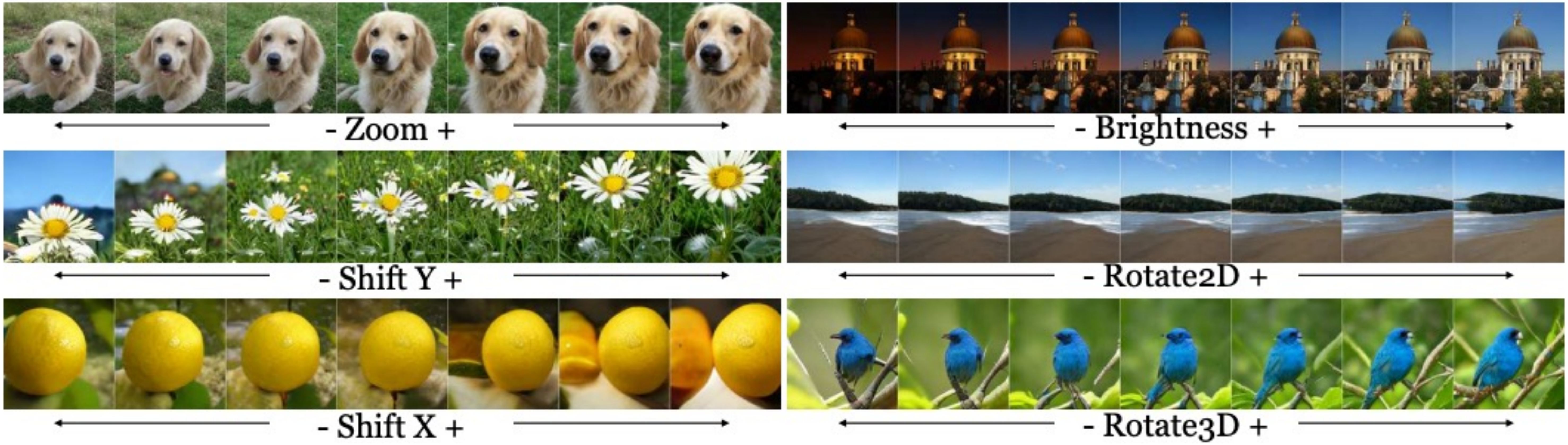
Non-linear case:
(f is a function)
apply it n times

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

strength

[Jahanian*, Chai*, Isola. 2020]

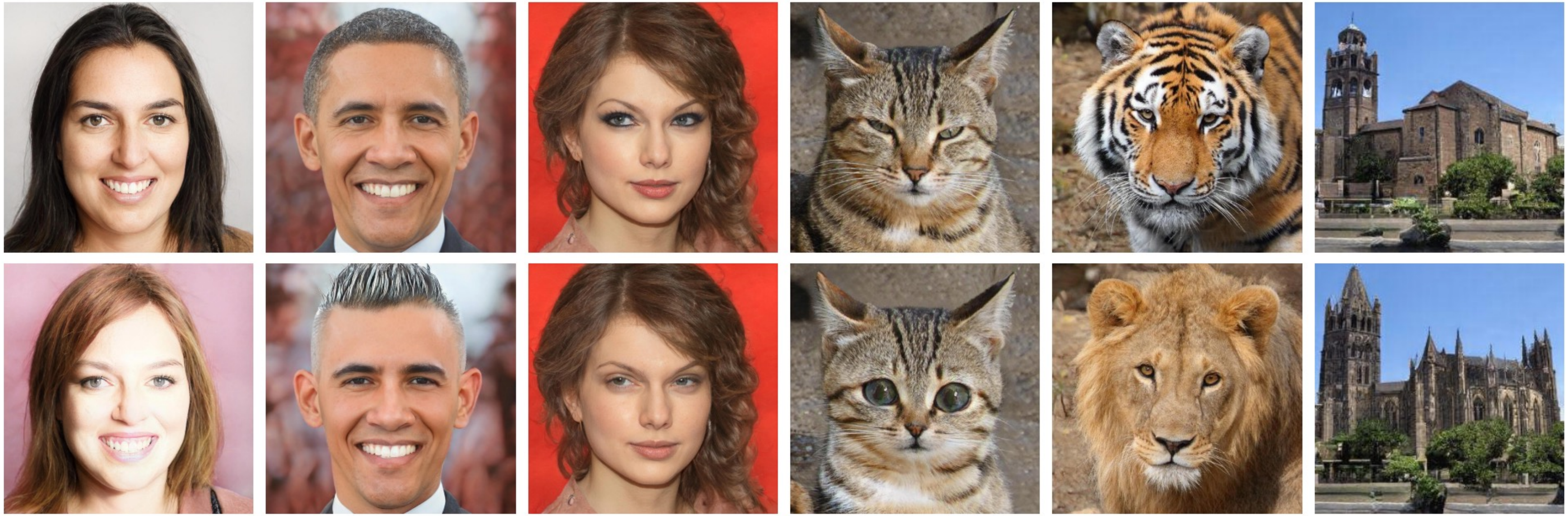
Offline optimization



Requirement: A known **edit** function
(e.g., shift, zoom, rotate)

[Jahanian*, Chai*, Isola. 2020]

CLIP-guided Directions



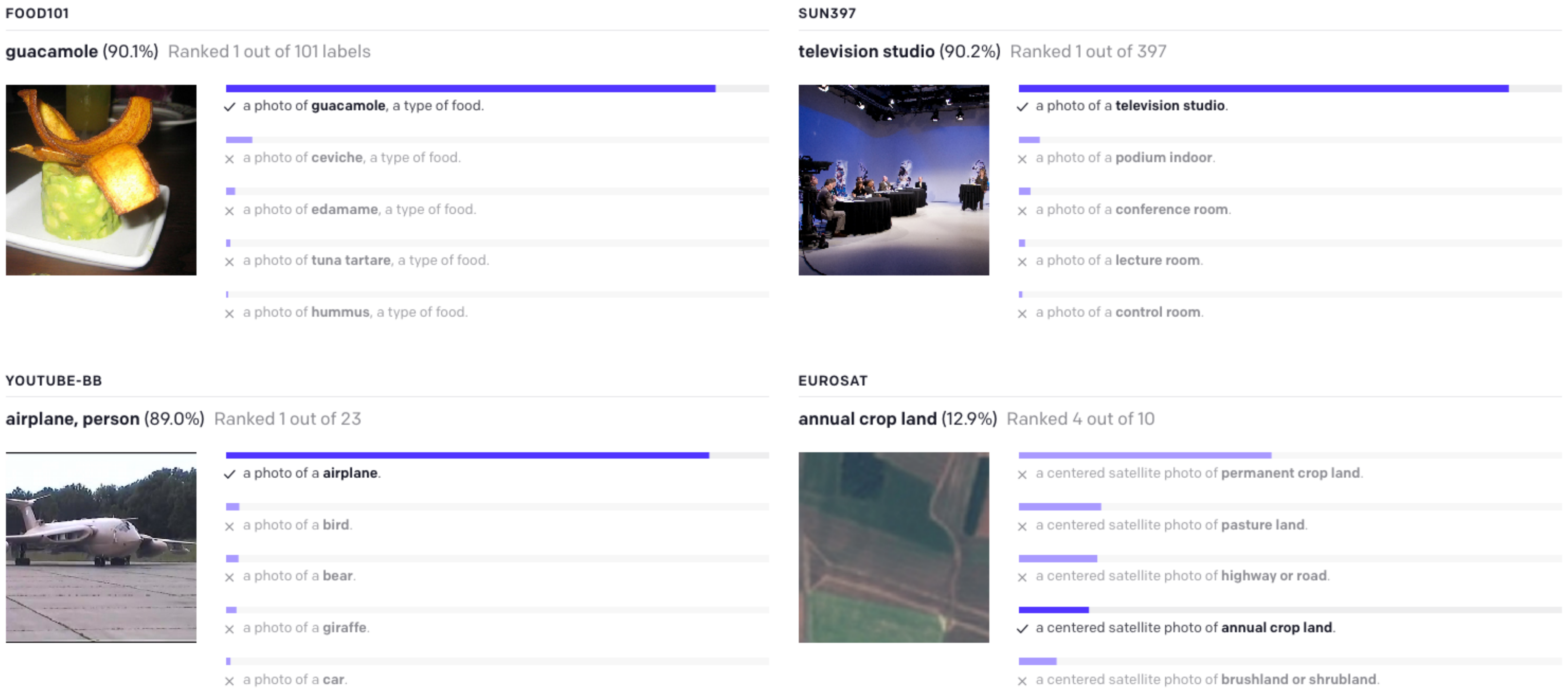
"Emma Stone" "Mohawk hairstyle" "Without makeup" "Cute cat" "Lion" "Gothic church"

$$\arg\min_{w \in \mathcal{W}+} D_{\text{CLIP}}(G(w), t) + \lambda_{\text{L2}} \|w - w_s\|_2 + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}}(w)$$

Output is close to the text Close to the original latent Output is close to input

StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery [Patashnik et al., ICCV 2021]

CLIP: Connecting Text and Images



<u>Input</u>: an image and a caption.

<u>Output</u>: similarity between the text embedding and the image embedding

StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery [Patashnik et al., ICCV 2021]

CLIP-guided Directions

$$\underset{w \in \mathcal{W}+}{\arg\min}\, D_{\text{CLIP}}(G(w), t) + \lambda_{\text{L2}} \left\| w - w_s \right\|_2 + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}}(w)$$

Output is close to the text   Close to the original latent   Output is close to input

StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery [Patashnik et al., ICCV 2021]

Manipulating network weights

Image Editing with GANs

- Step 1: Image Projection/Reconstruction

$$z_0, \theta_0 = \arg\min_{z,\theta} \mathcal{L}(G(z;\theta), x)$$

- Step 2: Manipulating the network weights

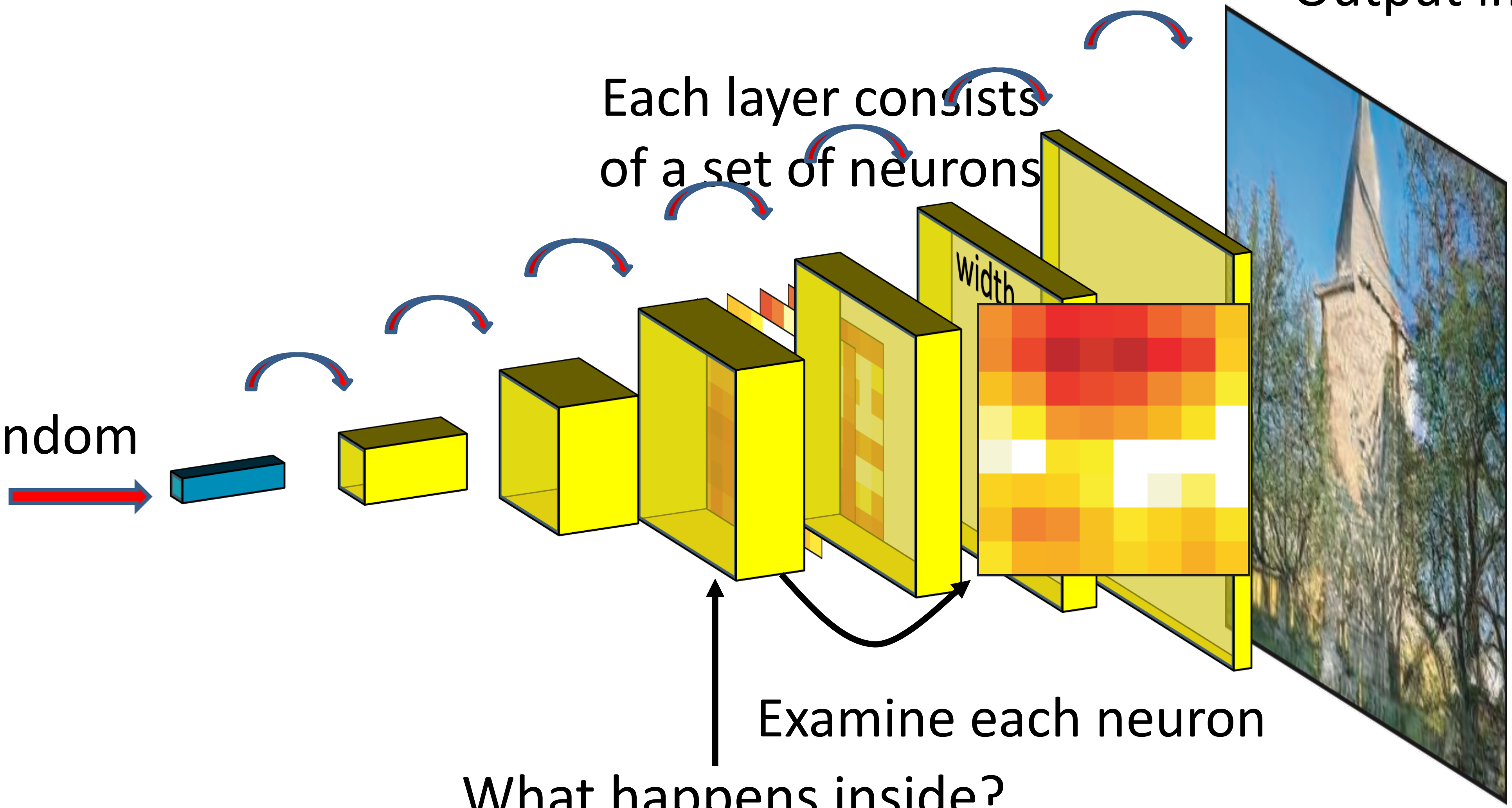
$$\theta_1 = \theta_0 + \Delta\theta$$

- Step 3: Generate the edited result

$$G(z_0; \theta_1)$$

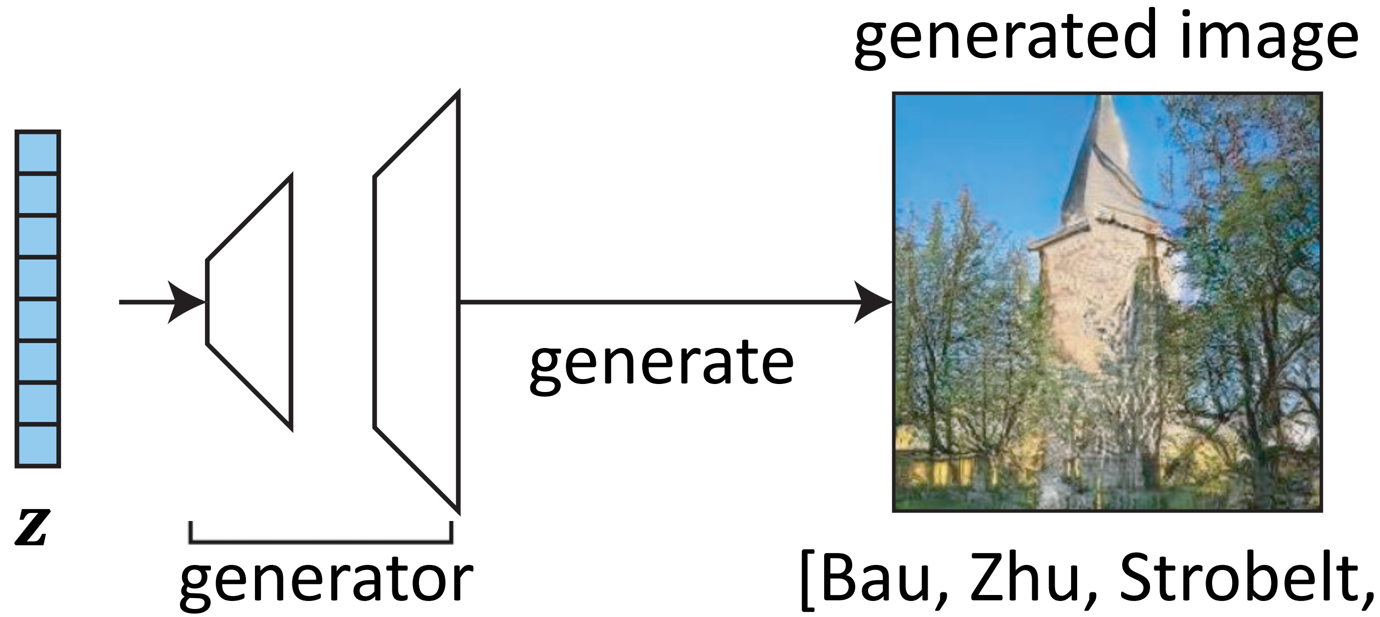
Understanding a Generator

Each step:
Increases spatial resolution



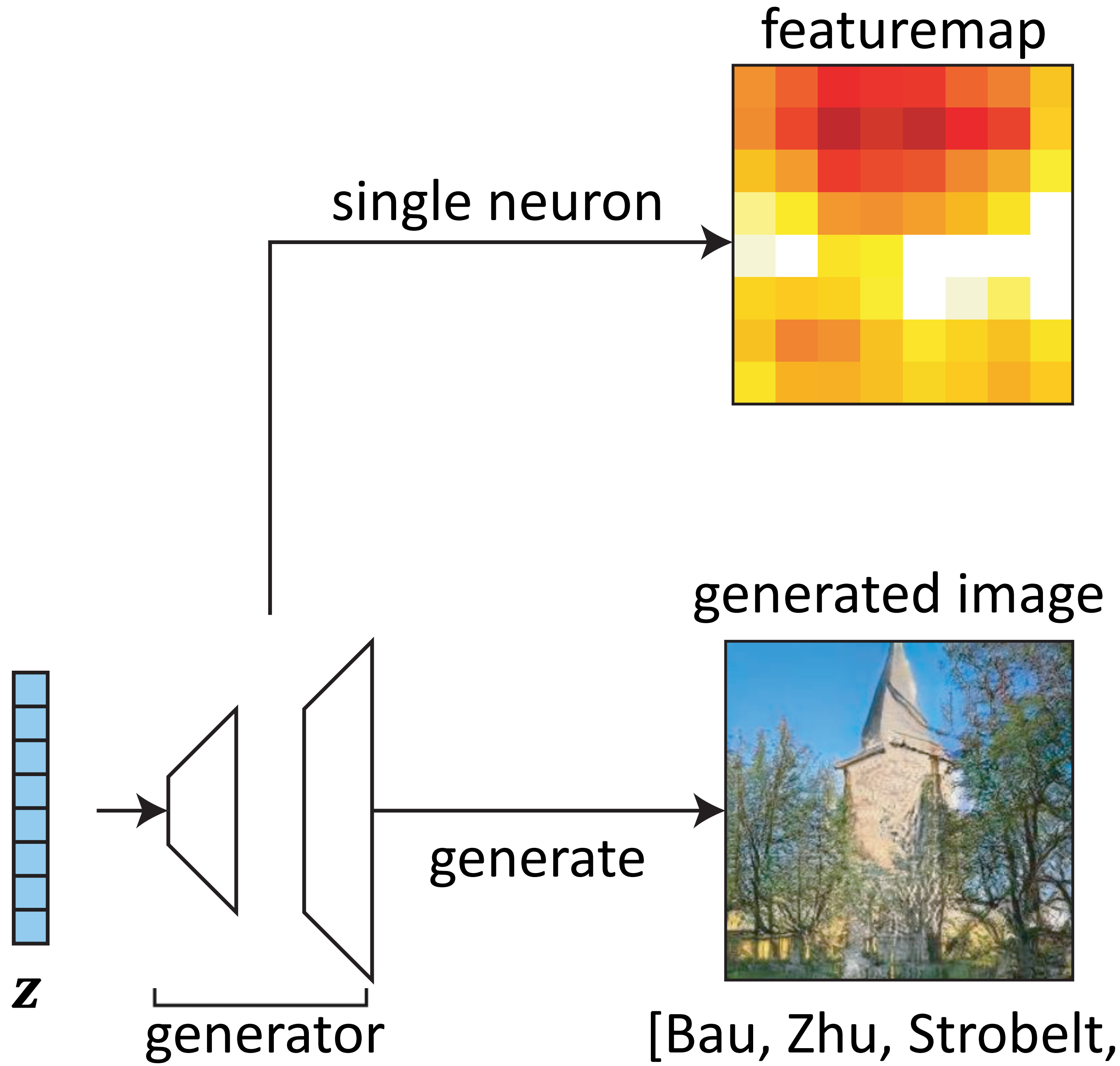
Progressive GANs [Karras et al. 2018]

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



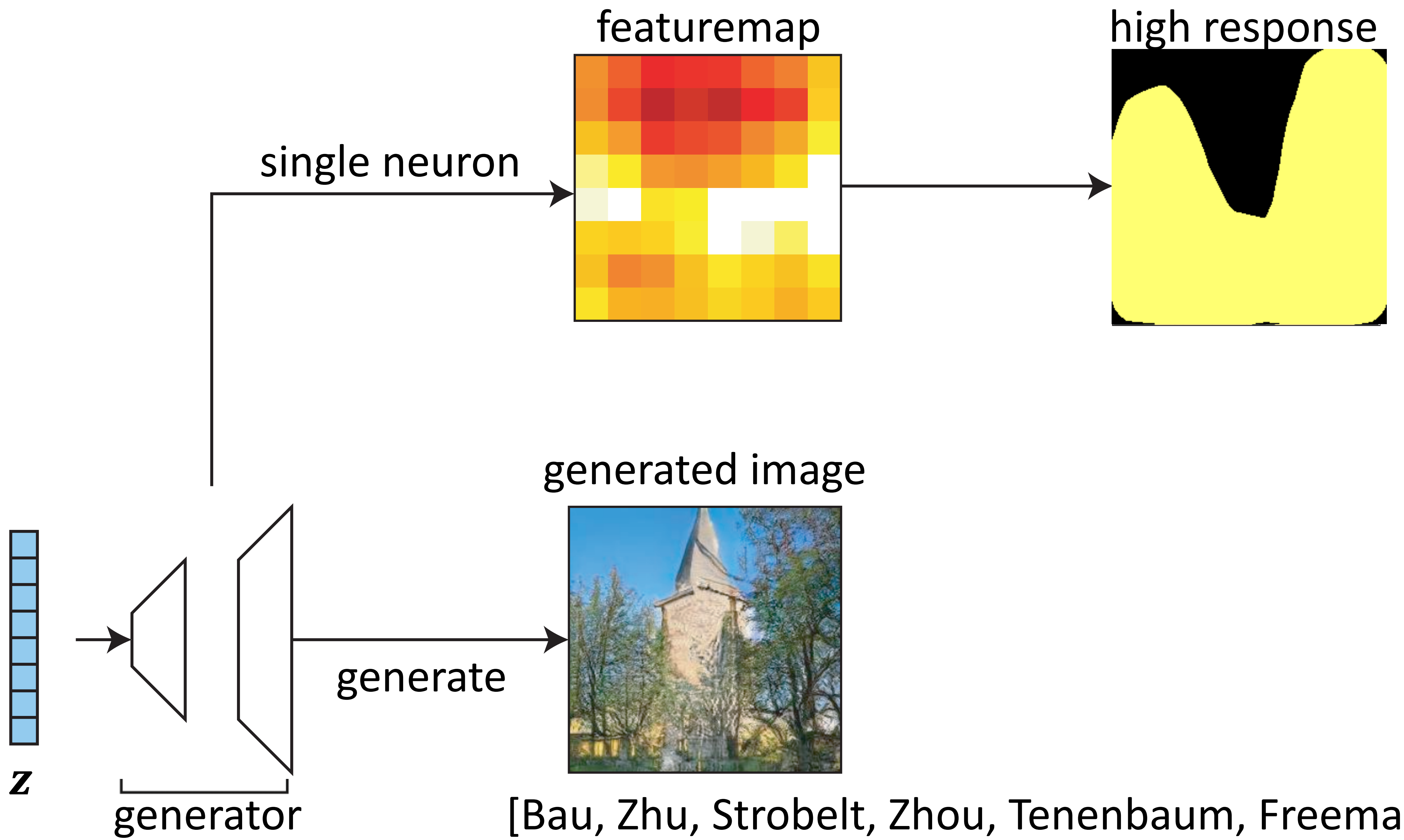
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

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



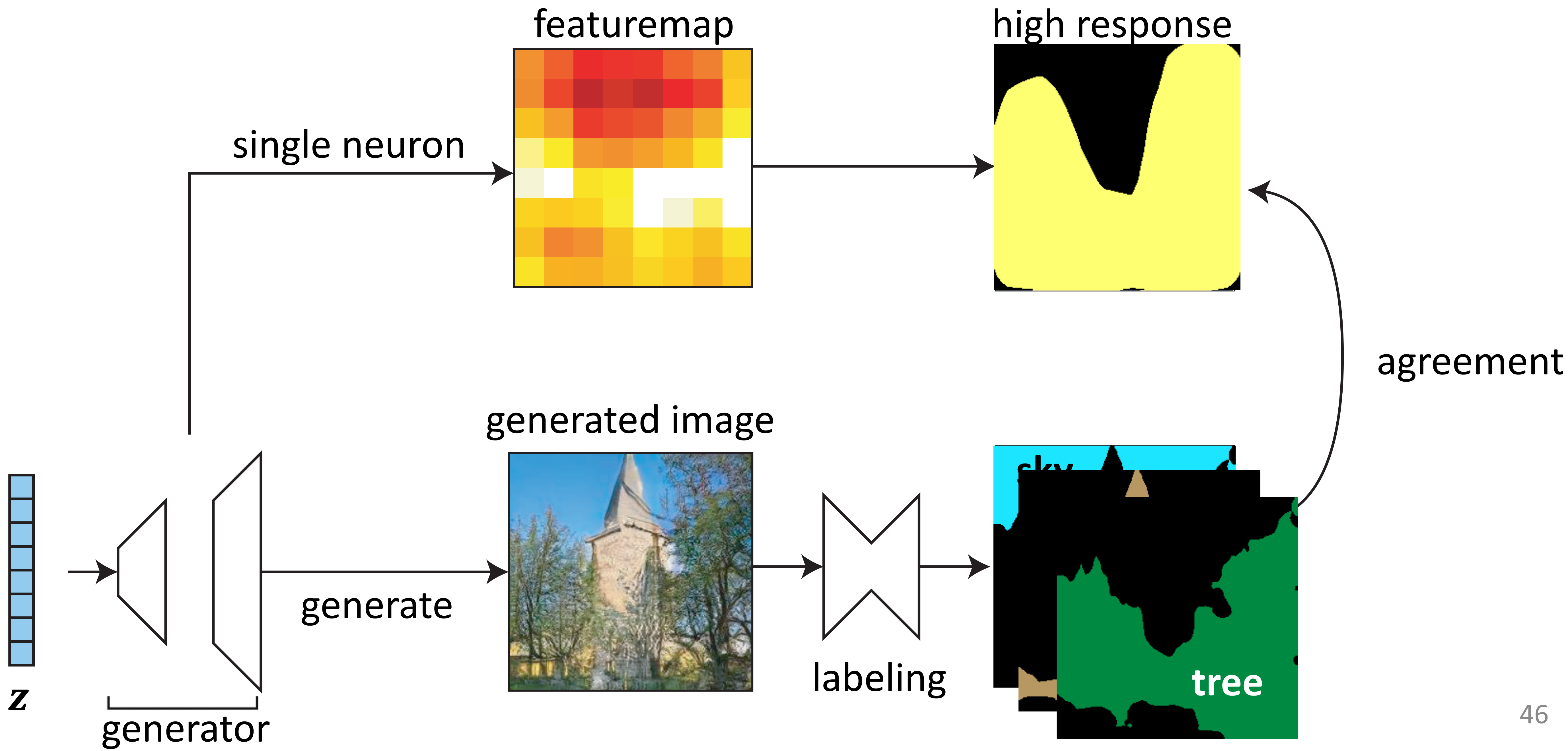
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

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

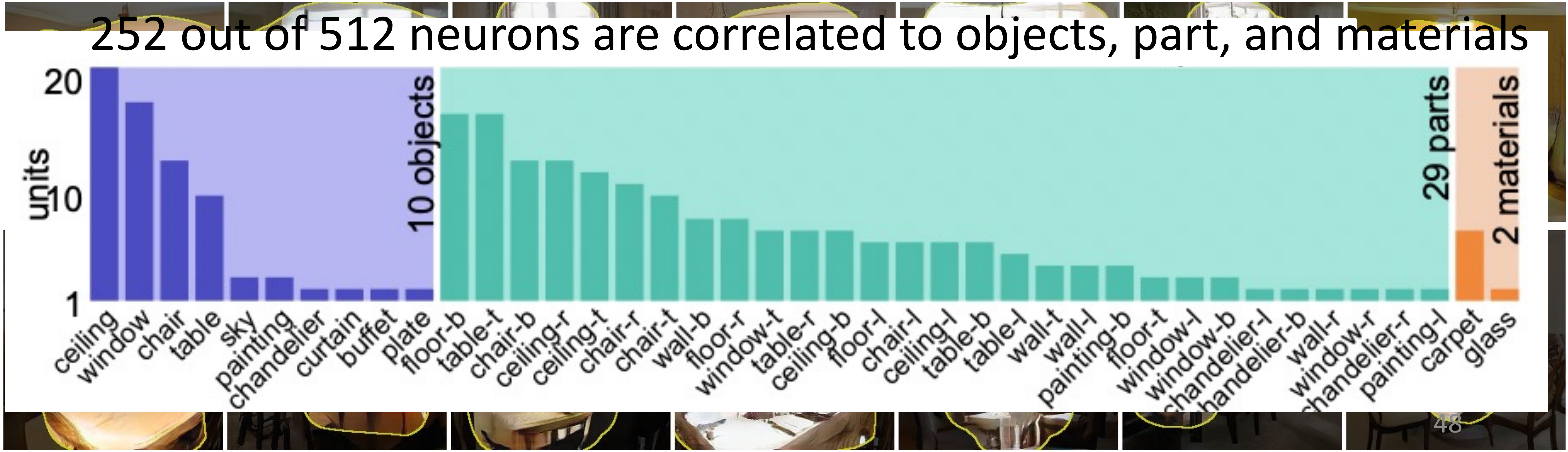
Dome Neuron

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

Dining room samples

Window Neuron

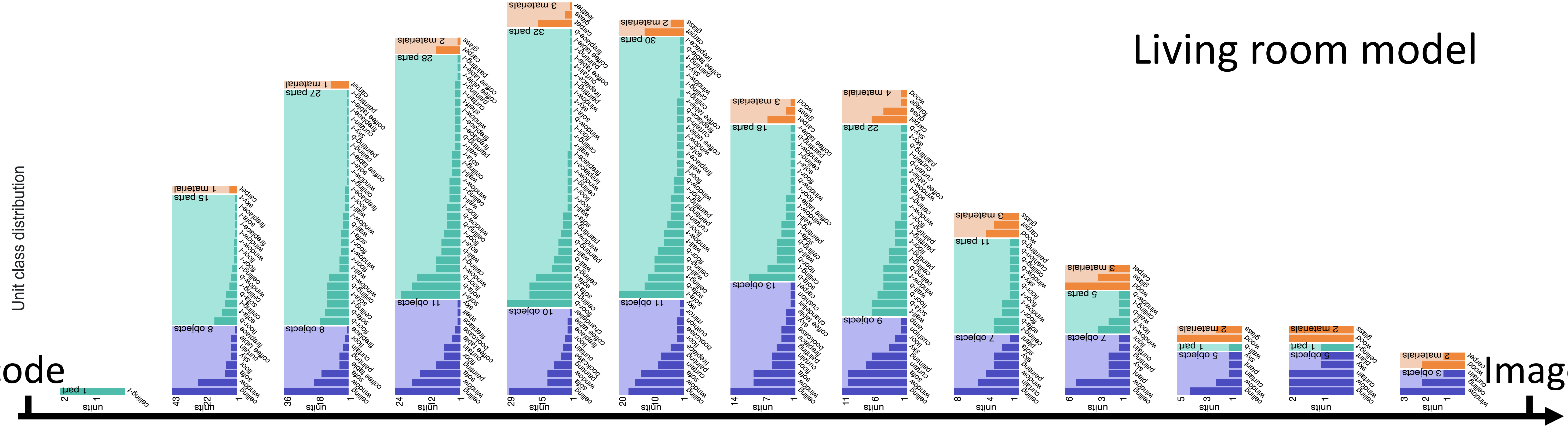
Table Neuron



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

Living room model

code



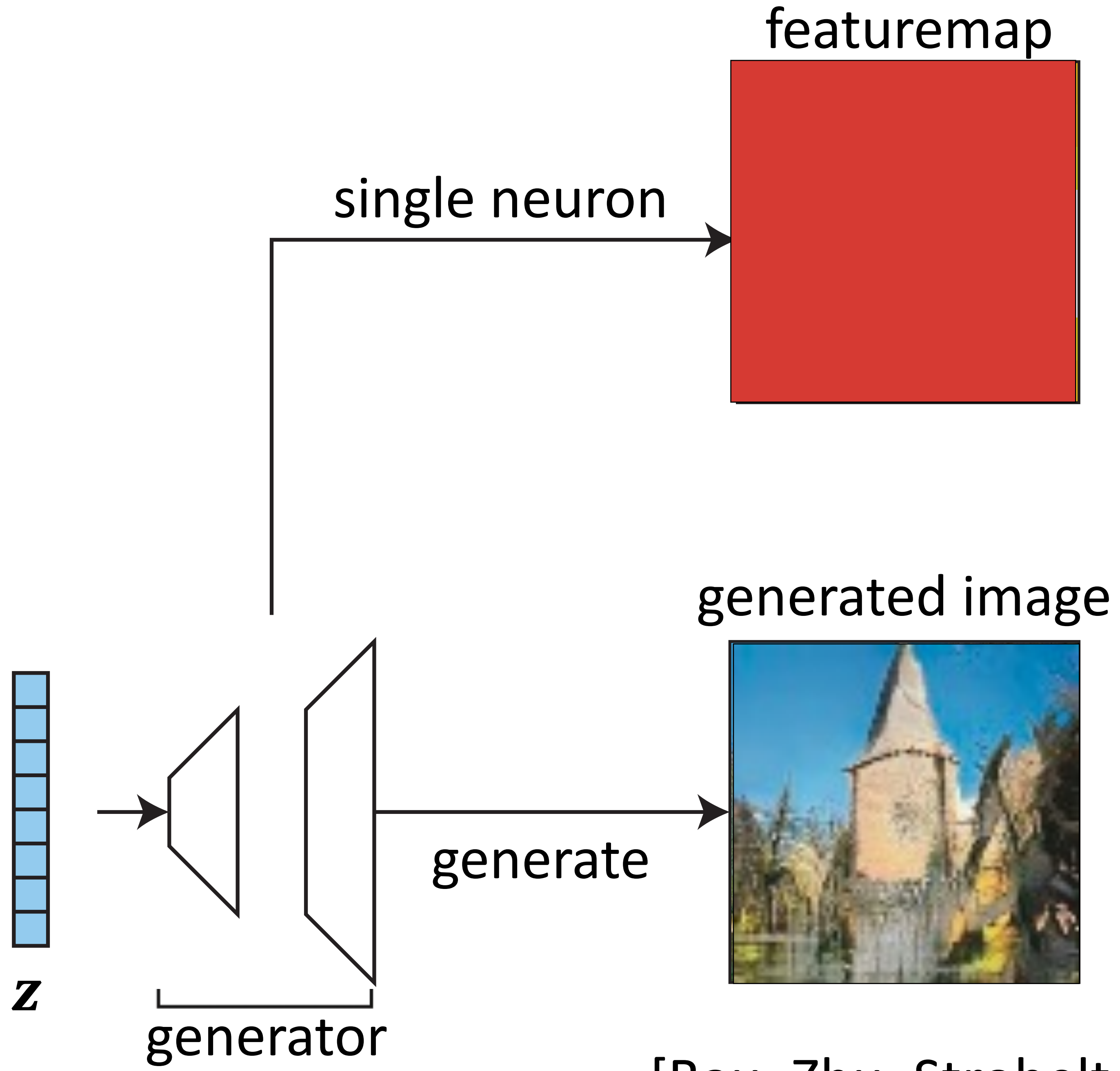
Image

Layout

Object and parts

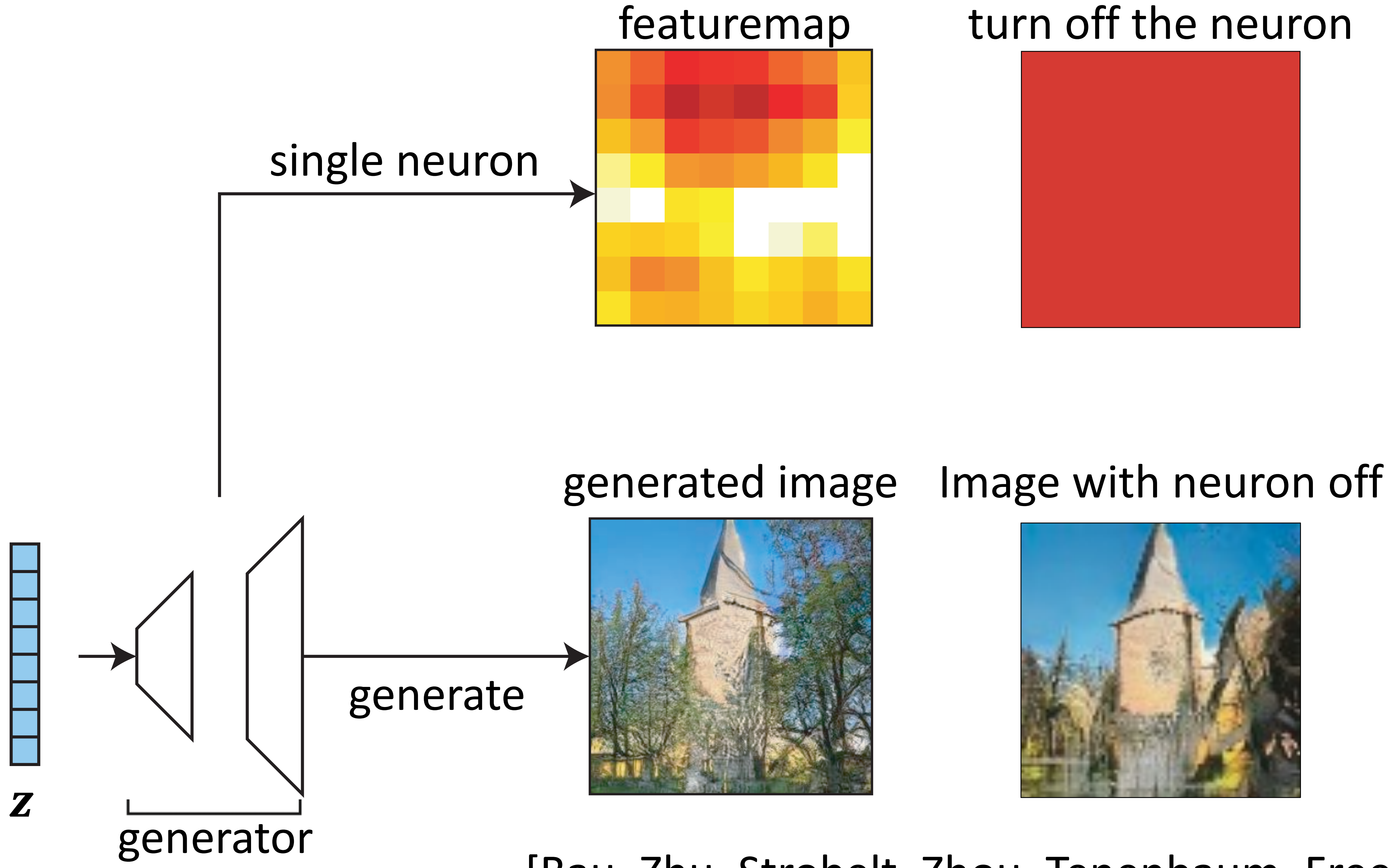
Edges, textures, local structure

Which neurons **cause** an object class?



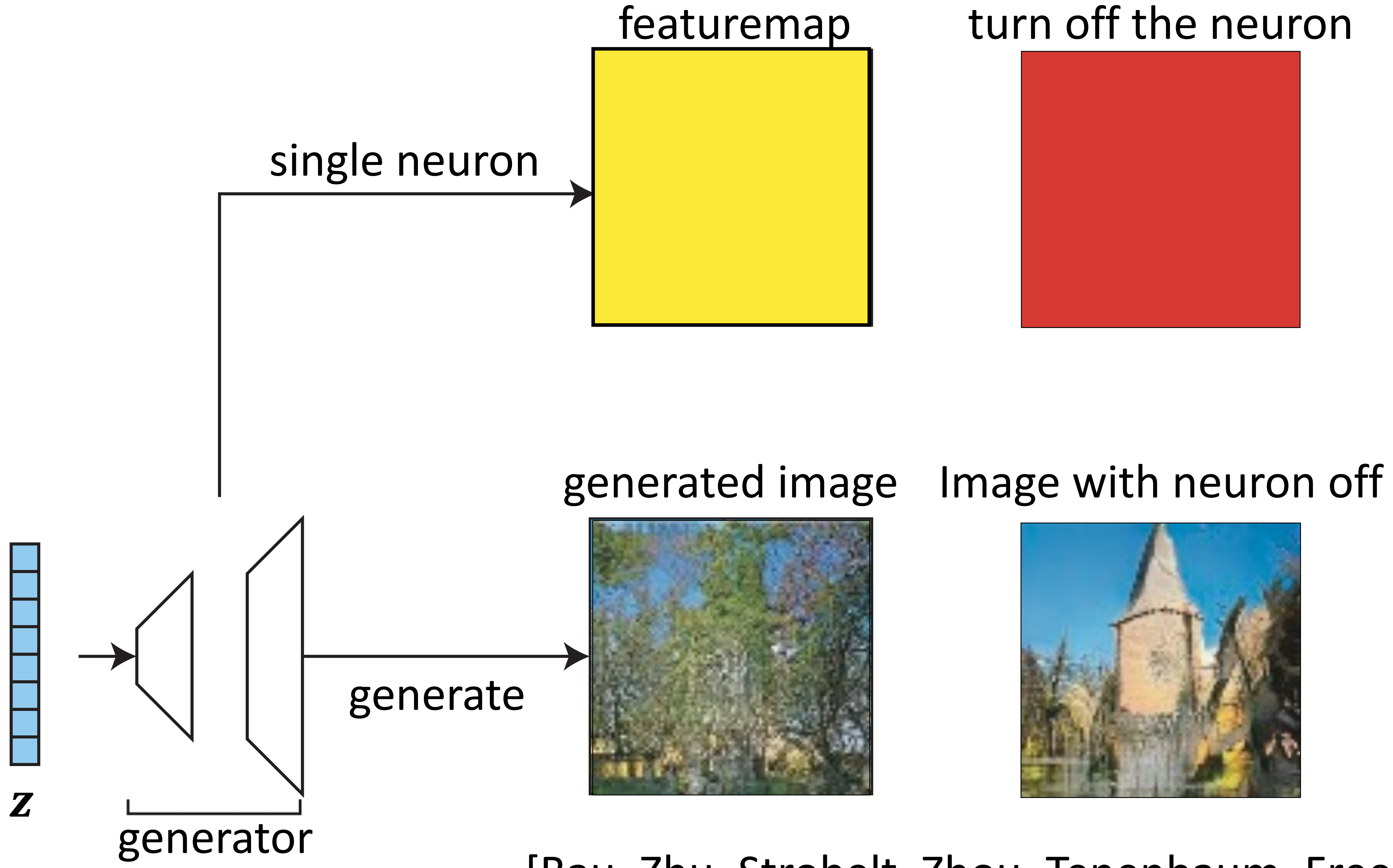
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **cause** an object class?



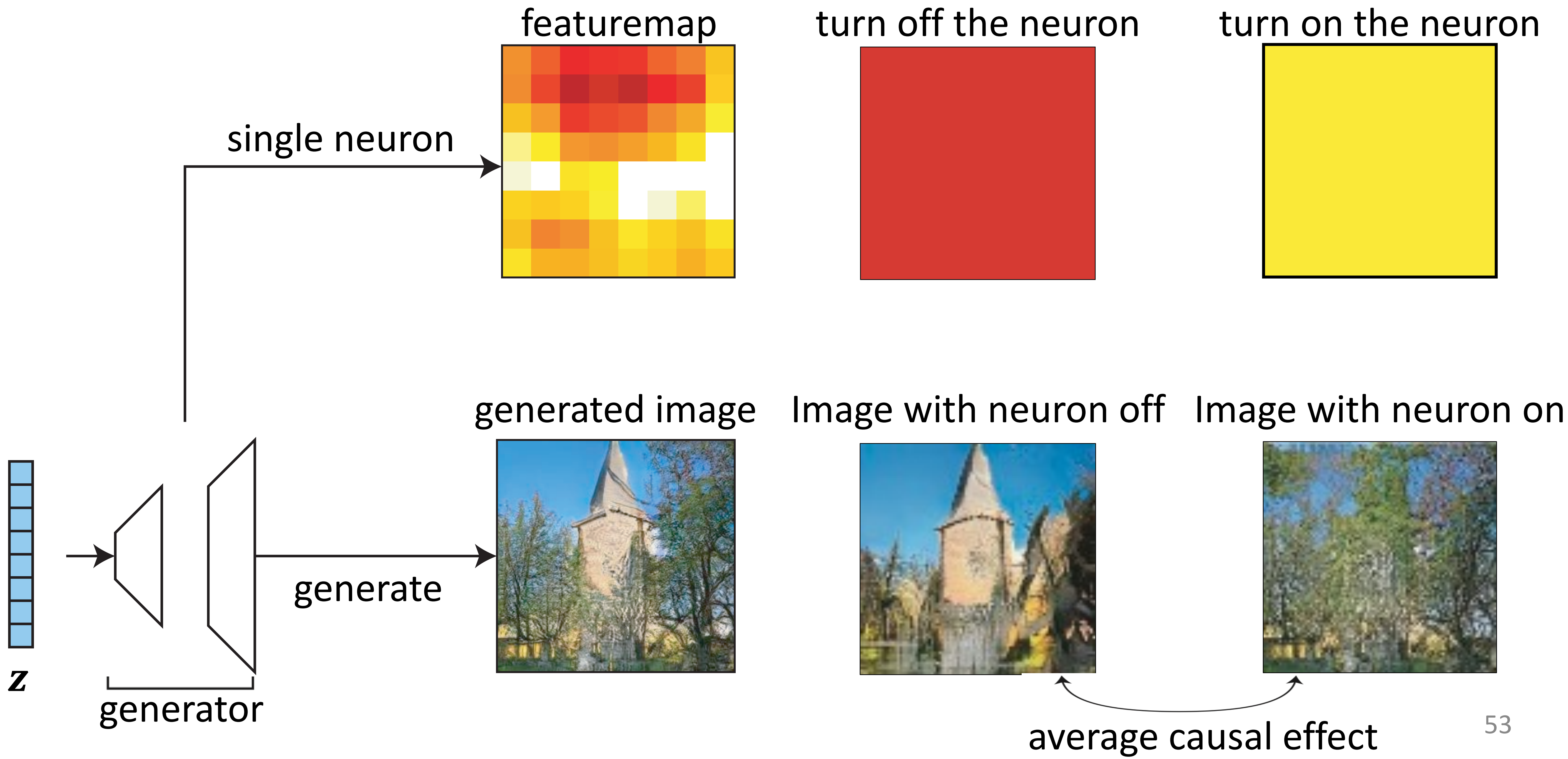
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **cause** an object class?



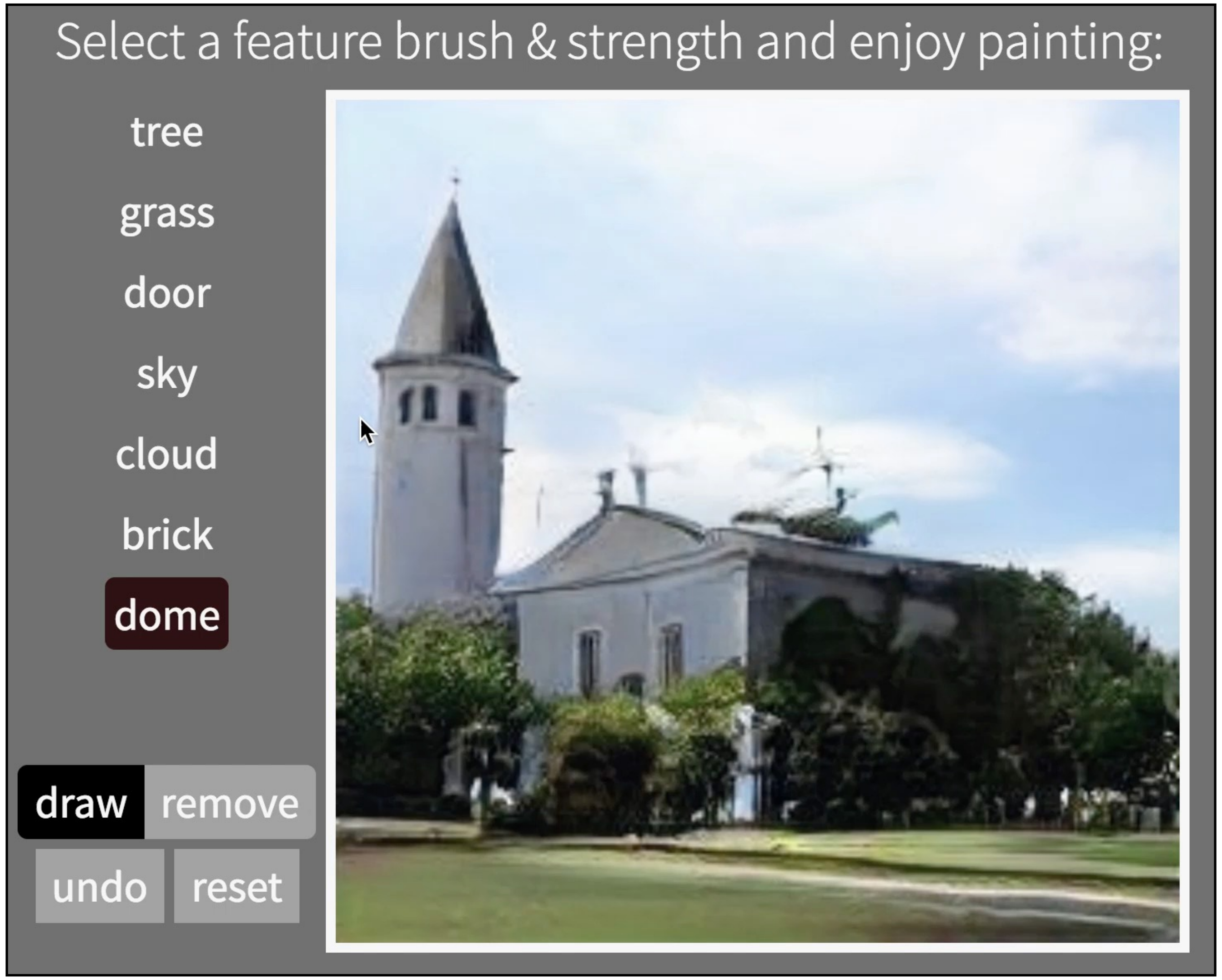
[Bau, Zhu, Strobelt, Zhou, Tenenbaum, Freeman, Torralba. ICLR 2019]

Which neurons **cause** an object class?



Which neurons **cause** an object class?

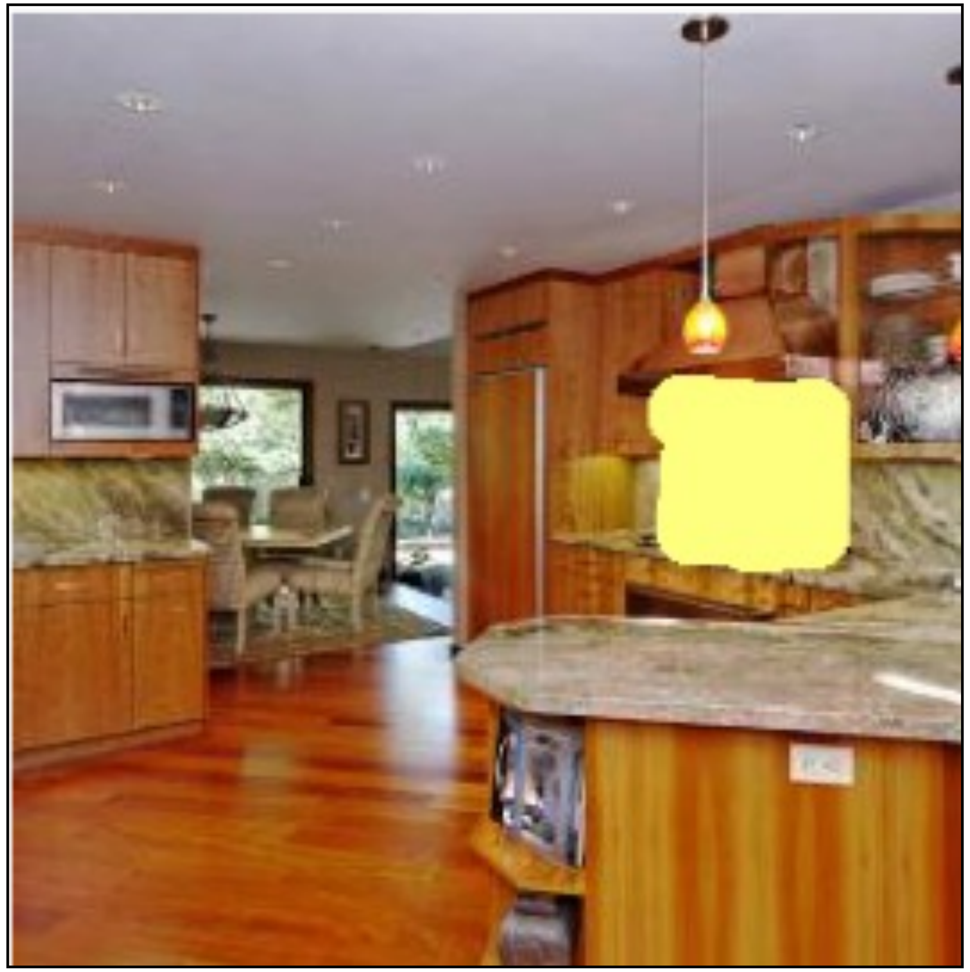
Interactive Painting



Online Demo

http://bit.ly/ganpaint

Manipulating a Real Photo



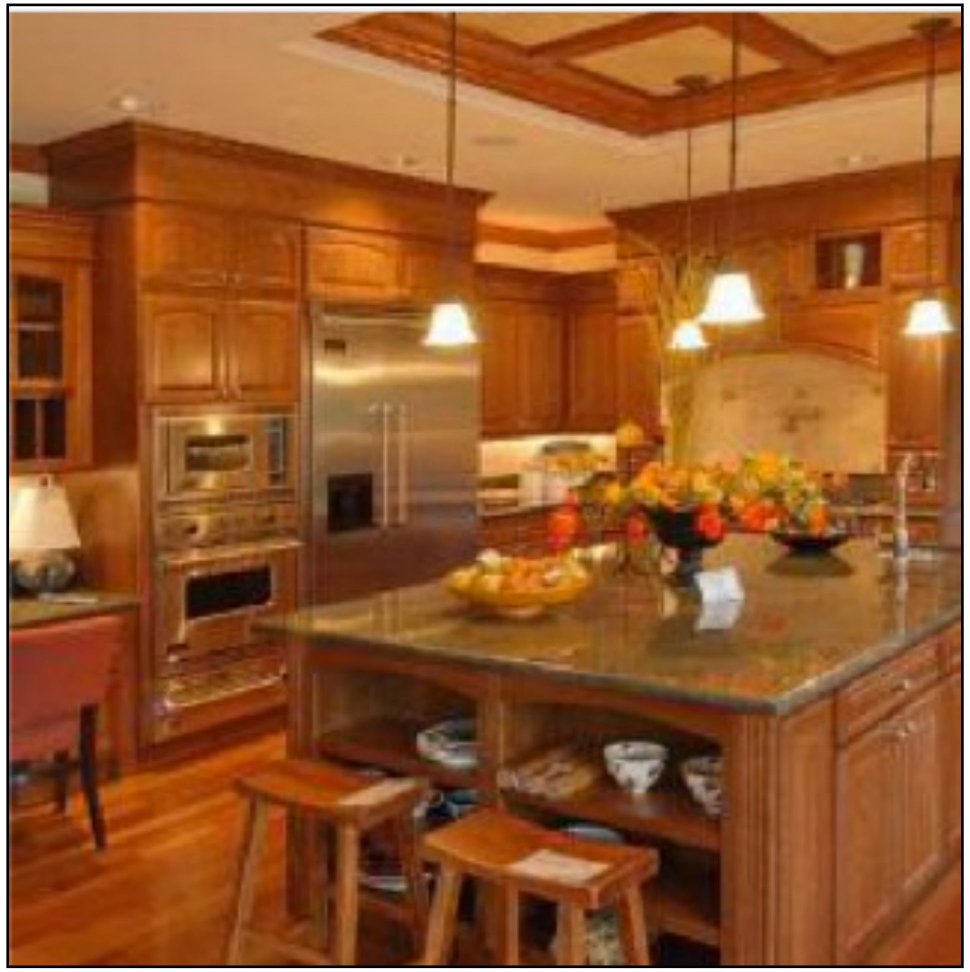
Original image + edits

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

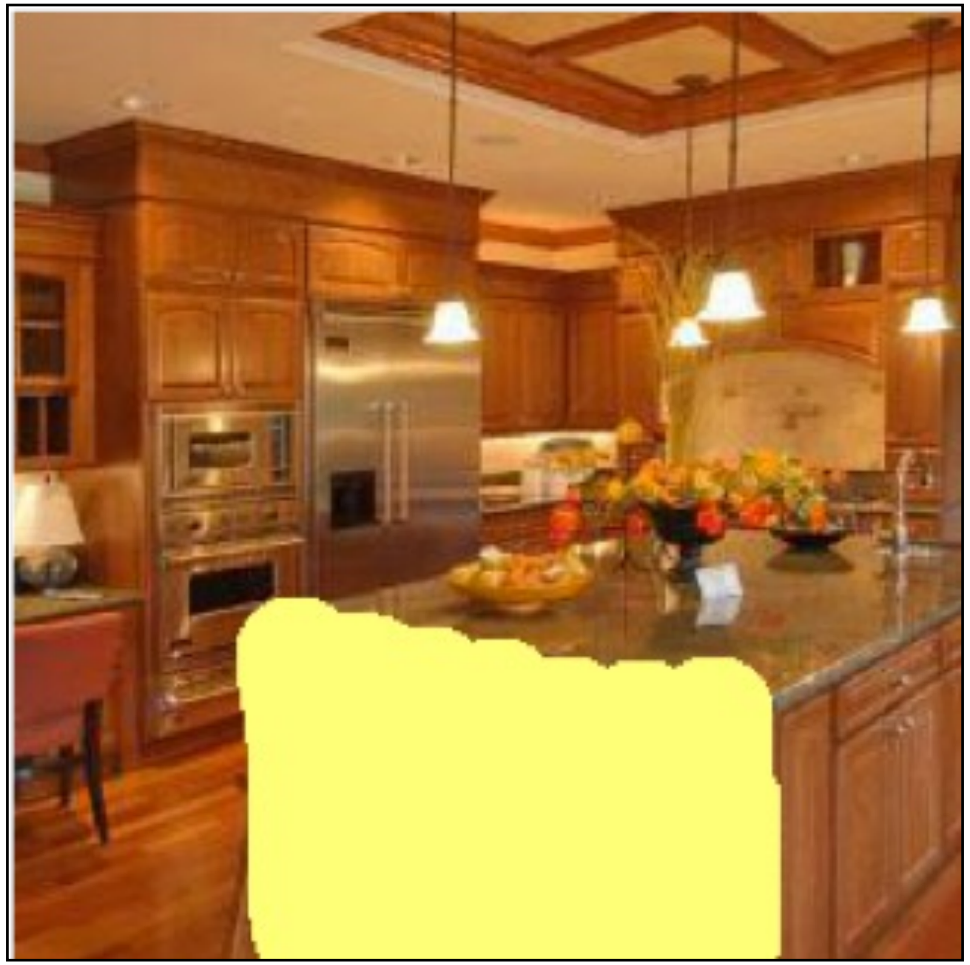
Editing with $\hat{\boldsymbol{z}}$ and $\widehat{\boldsymbol{\theta}}$

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

Manipulating a Real Photo



Input image



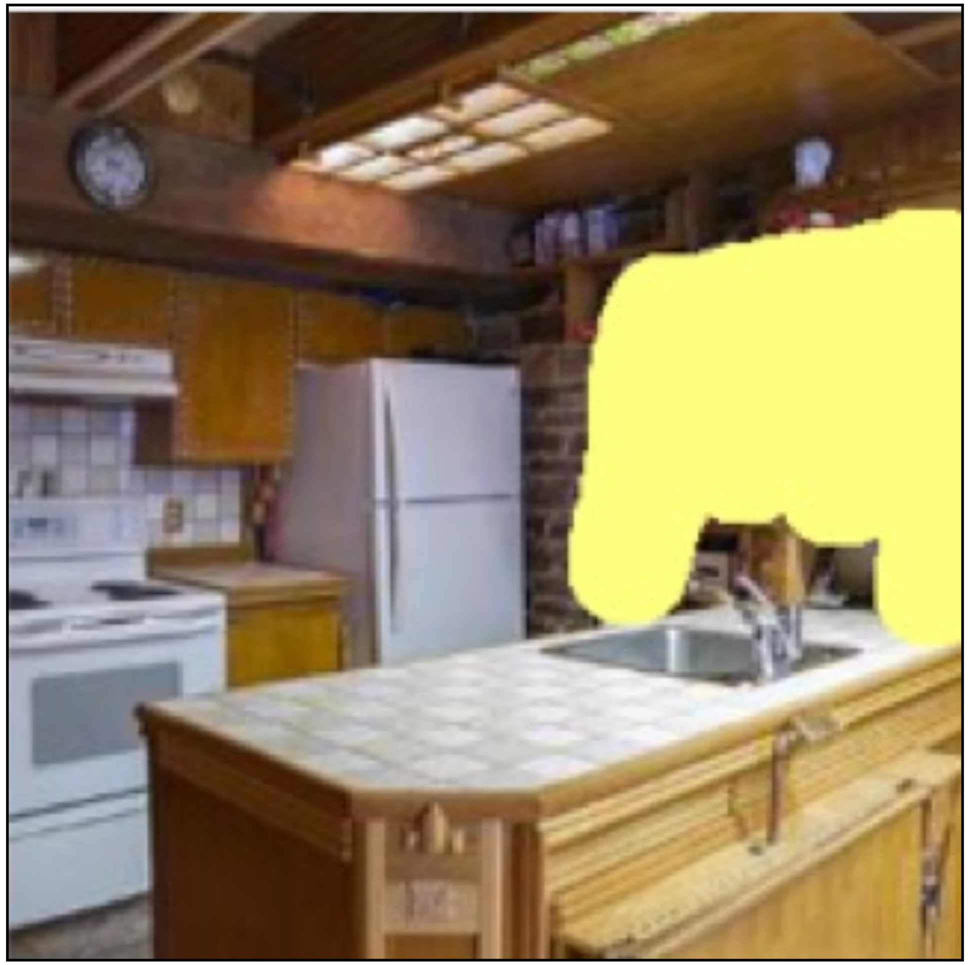
Remove chairs

Output result

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

Manipulating a Real Photo

Input image



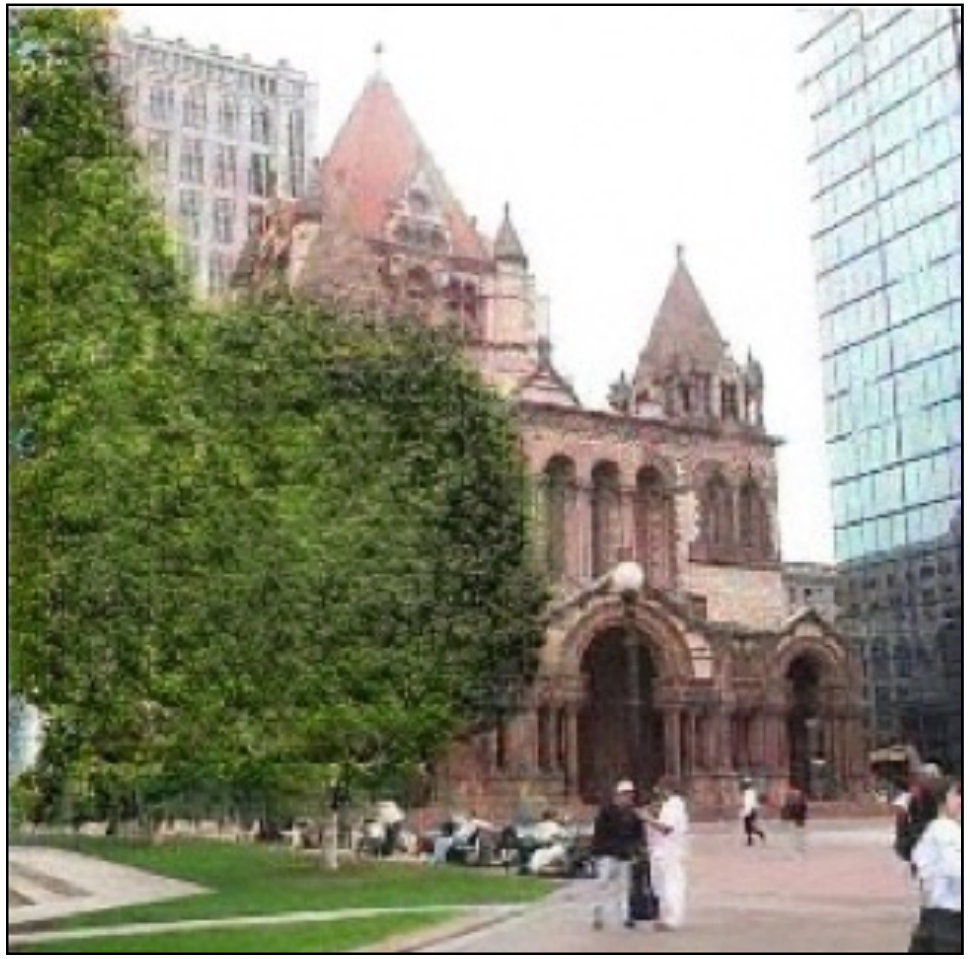
Add windows

Output result

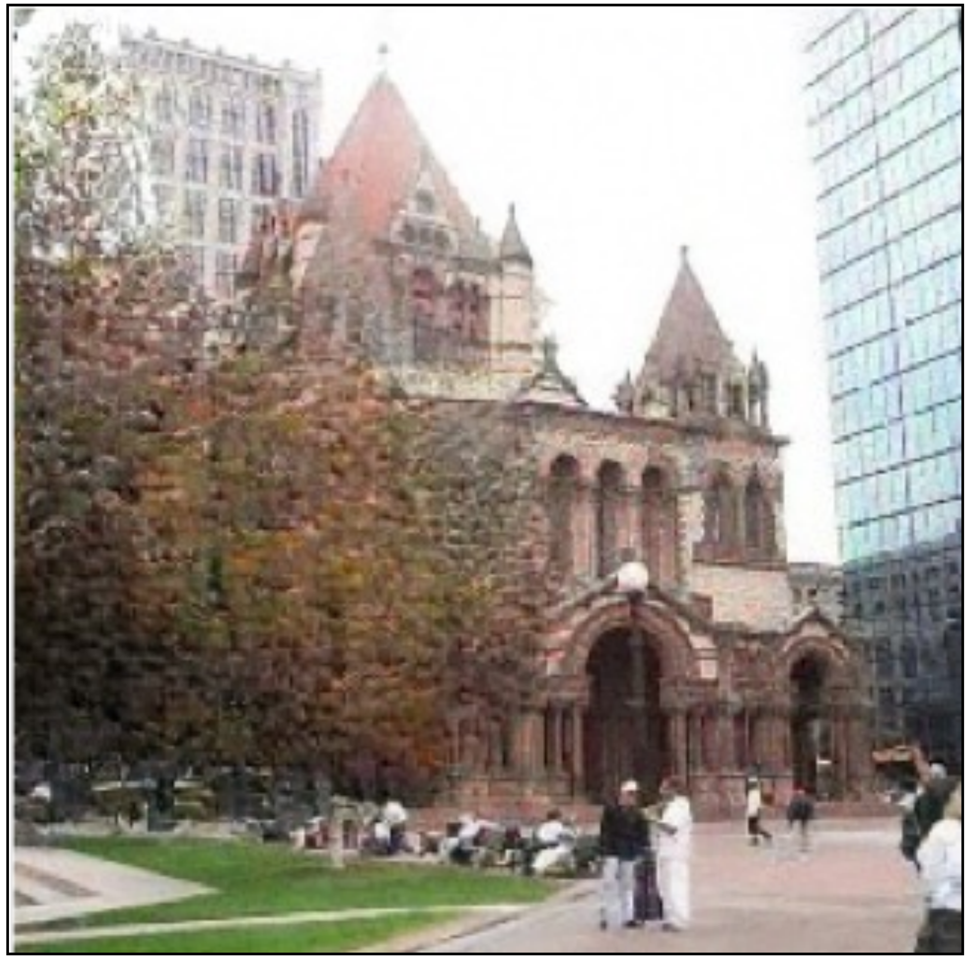
Semantic Photo Manipulation [Bau, Strobelt, Peebles, Wulff, Zhou, Zhu, Torralba, **SIGGRAPH** 2019]

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

Optimization
with Text-to-Image Diffusion Models

Text-to-image isn't perfect…

Stable Diffusion

Photo of a moongate

Text-to-image isn’t perfect…

Stable Diffusion

Photo of a moongate

Customization

Actual moongate images

Stable Diffusion

`Photo of a moongate`

Customization

Custom Diffusion

Actual moongate images

`Photo of a moongate`

Customization

Actual moongate images

Custom Diffusion

`Photo of a moongate`

Unseen contexts

Custom Diffusion

Actual moongate images

Moongate in the middle of highway

Unseen contexts

Actual moongate images

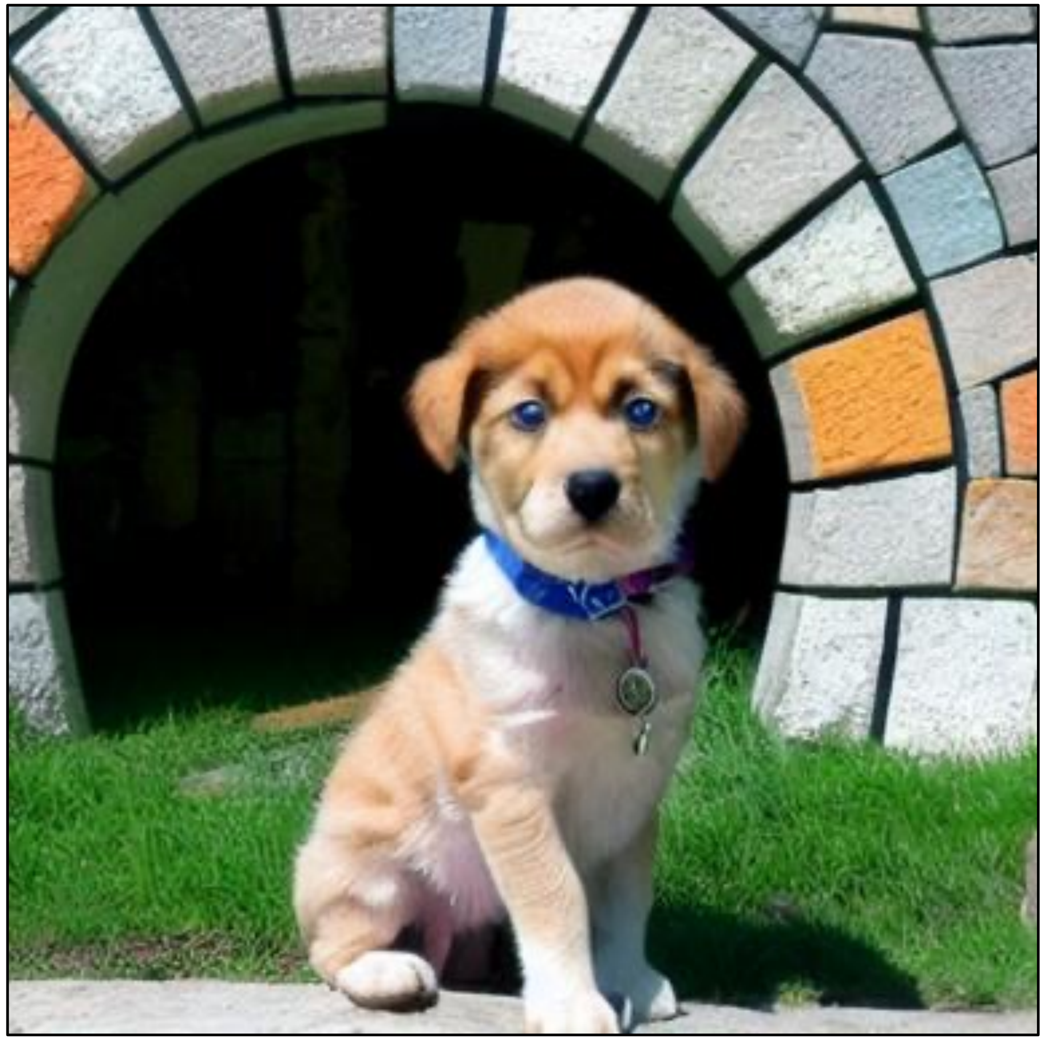
Custom Diffusion

`Moongate in snowy ice`

Unseen contexts

Custom Diffusion

Actual moongate images

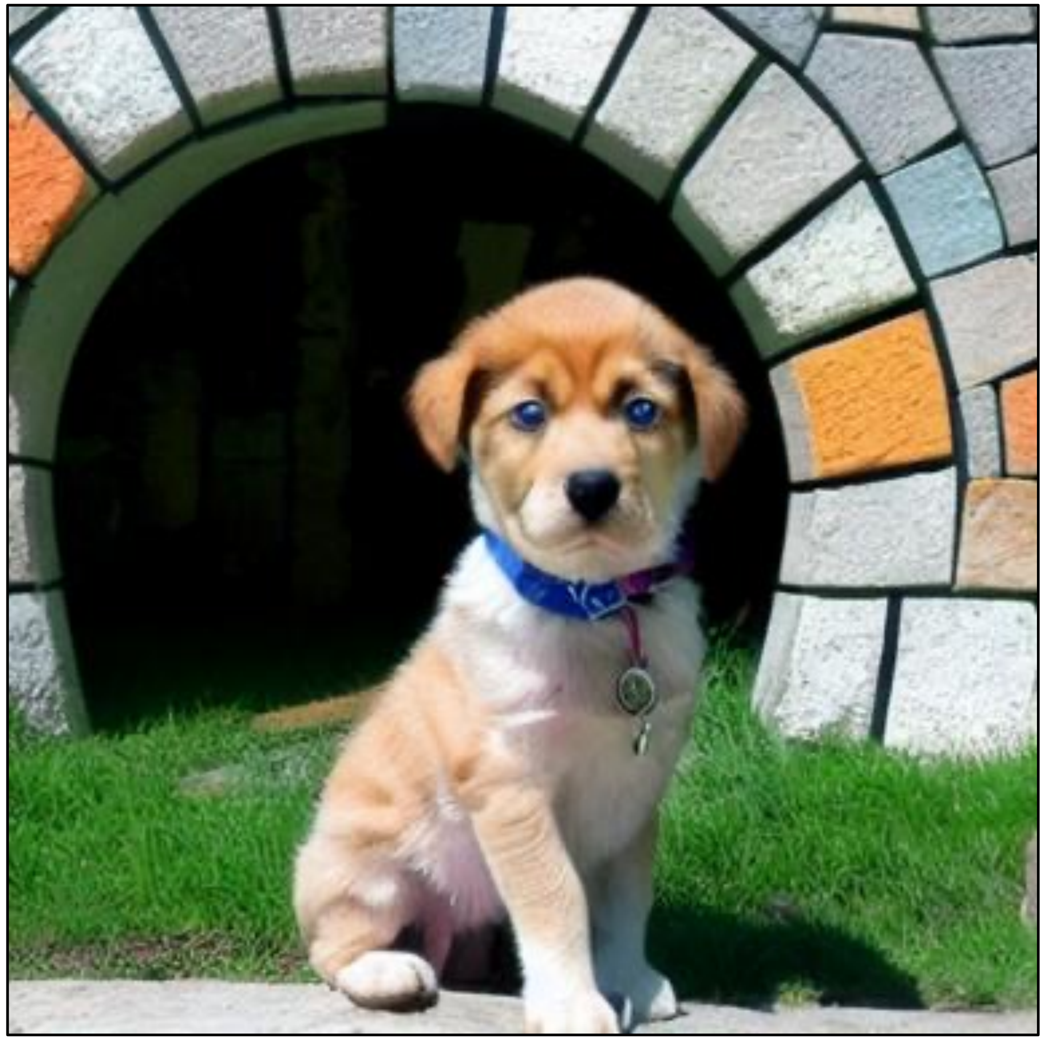


A puppy in front of Moongate

Multiple concepts

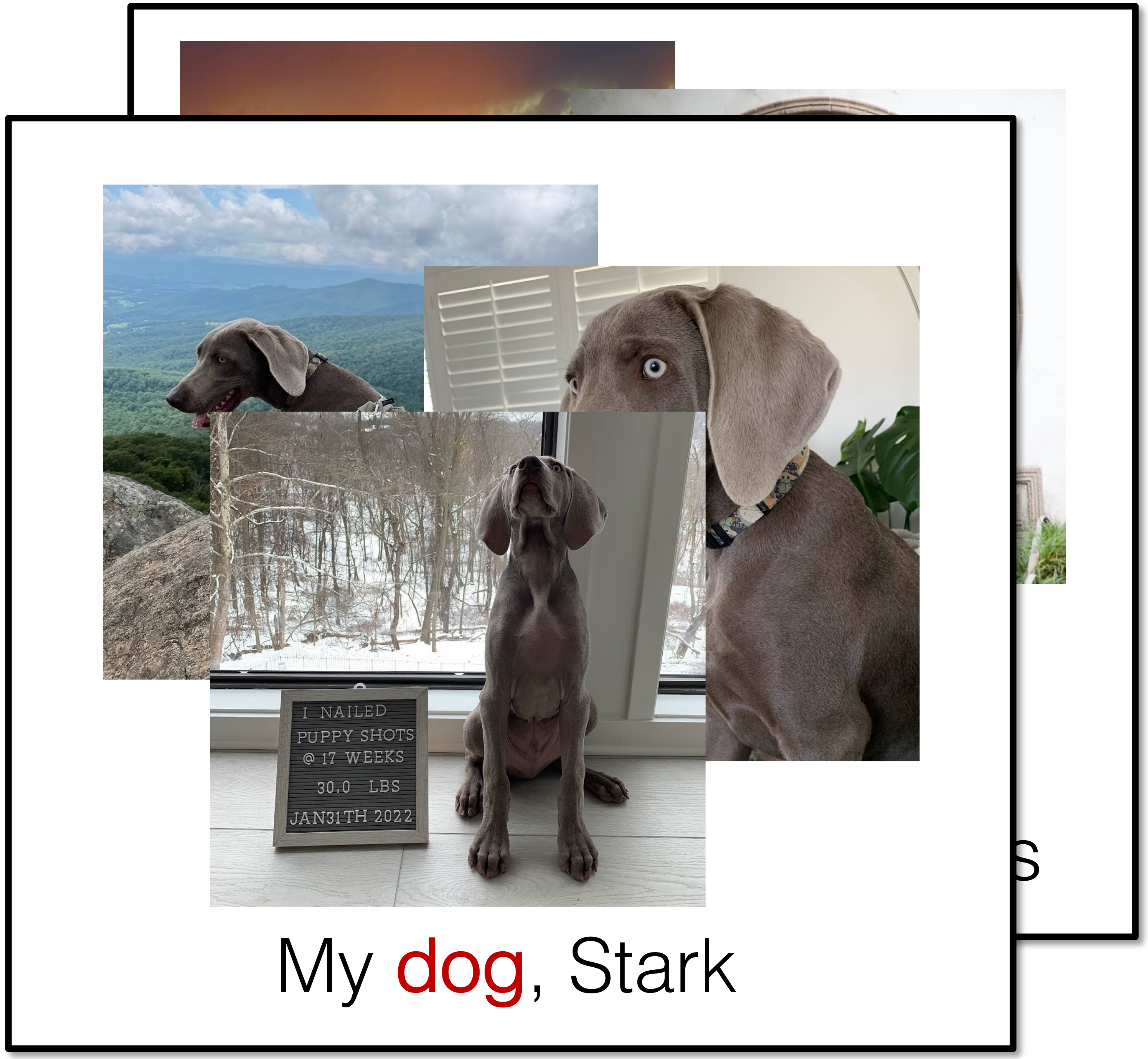
My dog, Stark

Custom
Diffusion



A puppy in front of Moongate

Multiple concepts



Custom Diffusion

`V* dog wearing sunglasses in front of moongate`

Textual Inversion

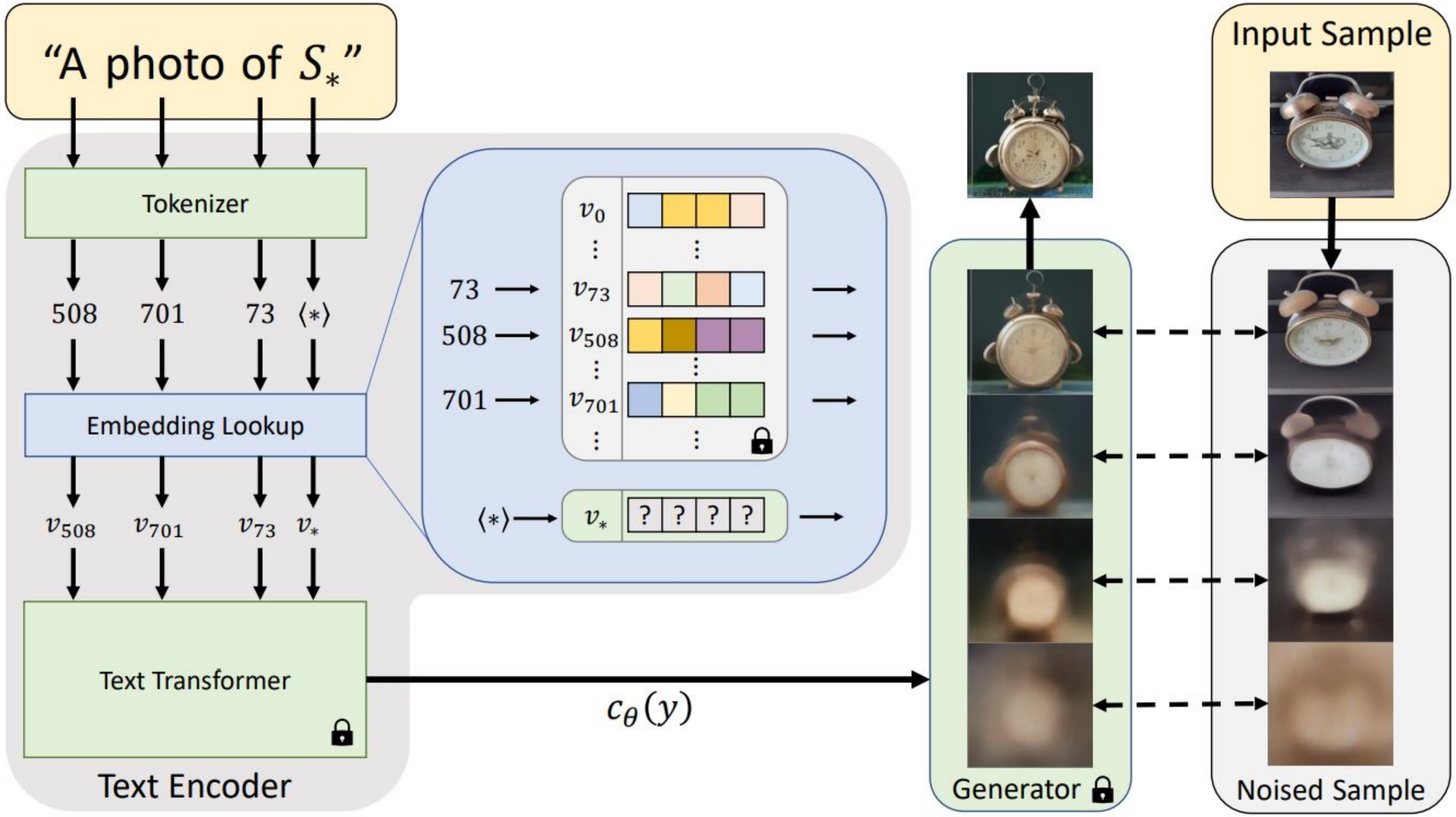
Input samples $\xrightarrow{invert}$ "S_*" $\rightarrow$ "An oil painting of S_*" "App icon of S_*" "Elmo sitting in the same pose as S_*" "Crochet S_*"

[Rinon Gal et al., arXiv 2022]

Textual Inversion

[Rinon Gal et al., arXiv 2022]

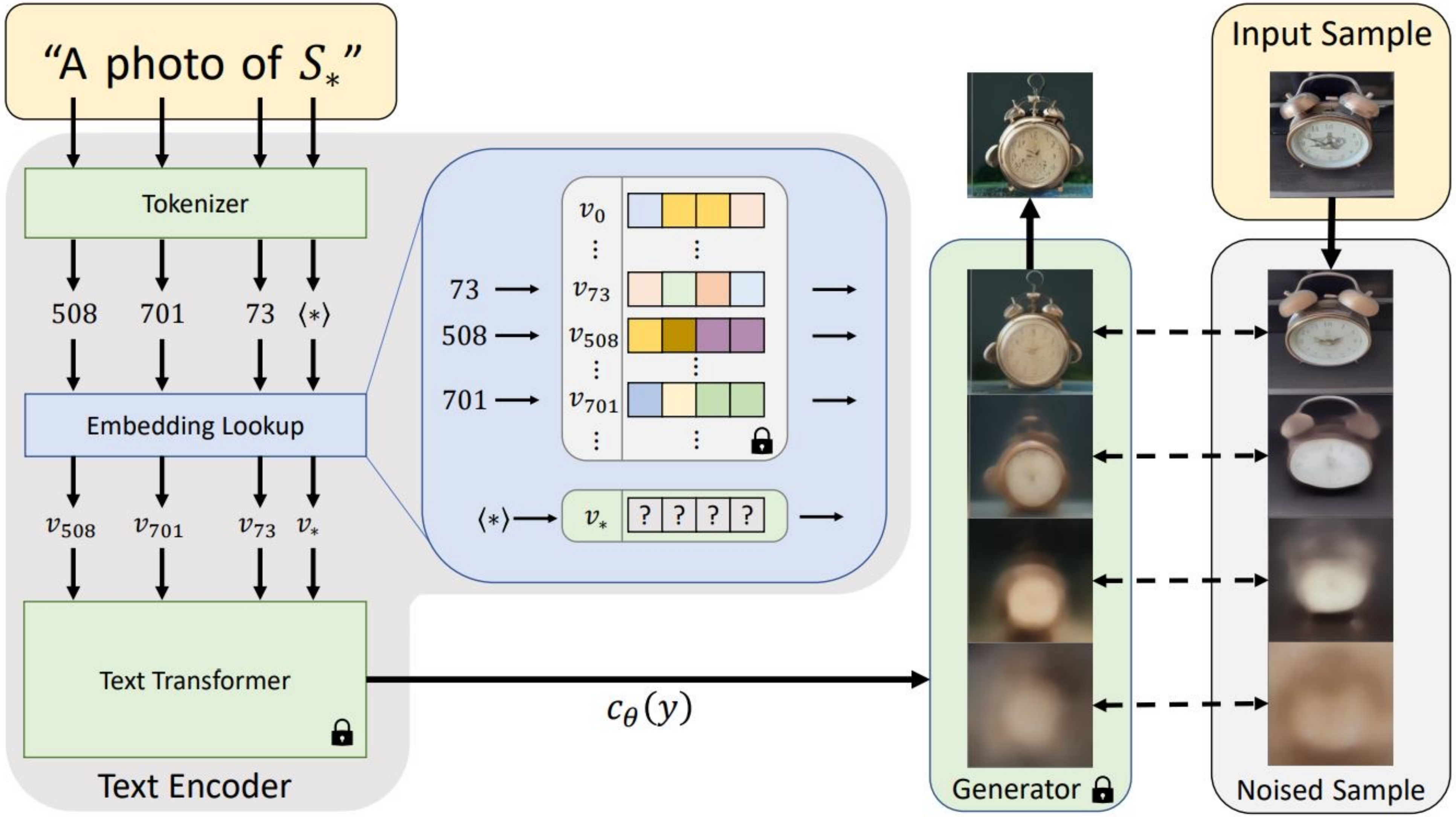
Textual Inversion



$$v_* = \arg\min_v \mathbb{E}_{z\sim\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2\right]$$

[Rinon Gal et al., arXiv 2022]

Textual Inversion



$$v_* = \arg\min_v \mathbb{E}_{z\sim\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2\right]$$

[Rinon Gal et al., arXiv 2022]

Works well for artistic styles

Input samples → "The streets of Paris in the style of S_*" "Adorable corgi in the style of S_*" "Painting of a black hole in the style of S_*" "Times square in the style of S_*"

[Rinon Gal et al., arXiv 2022]

Compositional Ability

Only works for style

[Rinon Gal et al., arXiv 2022]

Reconstruction quality

Target images

S* cat swimming in a pool

[Rinon Gal et al., arXiv 2022]

How to improve reconstruction quality?

Optimization space

(model weights, weight subsets, extended embedding space)

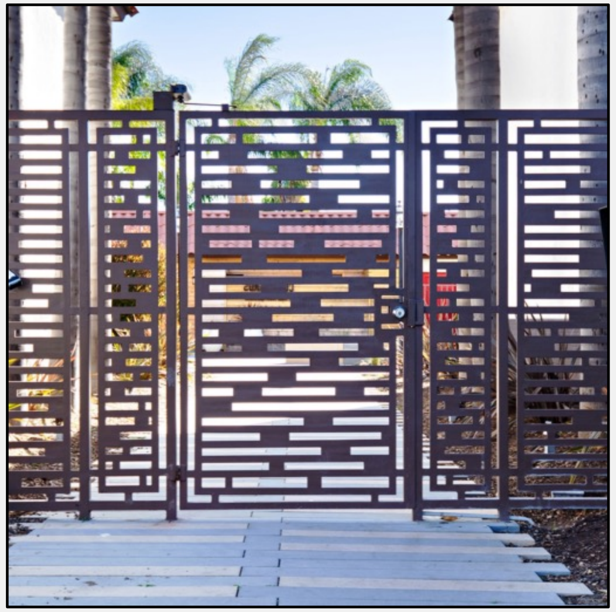
Model Training

Generate/Retrieve images with similar captions

Moongate

sky full of stars and the moon

Blood moon



Apartment gates

...

Model Training

Training dataset

Photo of a moongate

Photo of a moongate

...

Target images

+

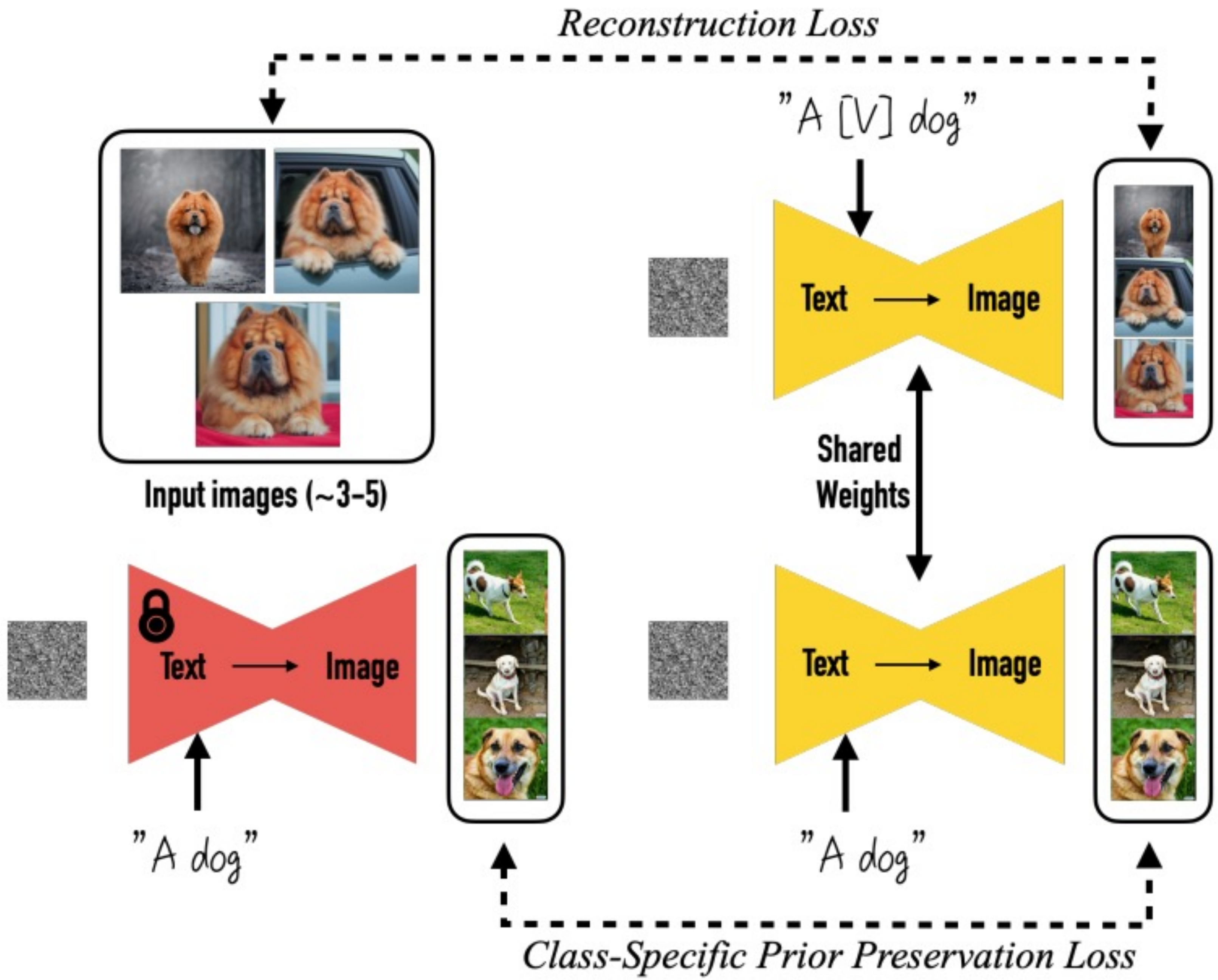
sky full of stars and the moon

Blood moon

...

Regularization images

DreamBooth Training (Fine-tuning all the weights)



[Nataniel Ruiz et al., CVPR 2023]

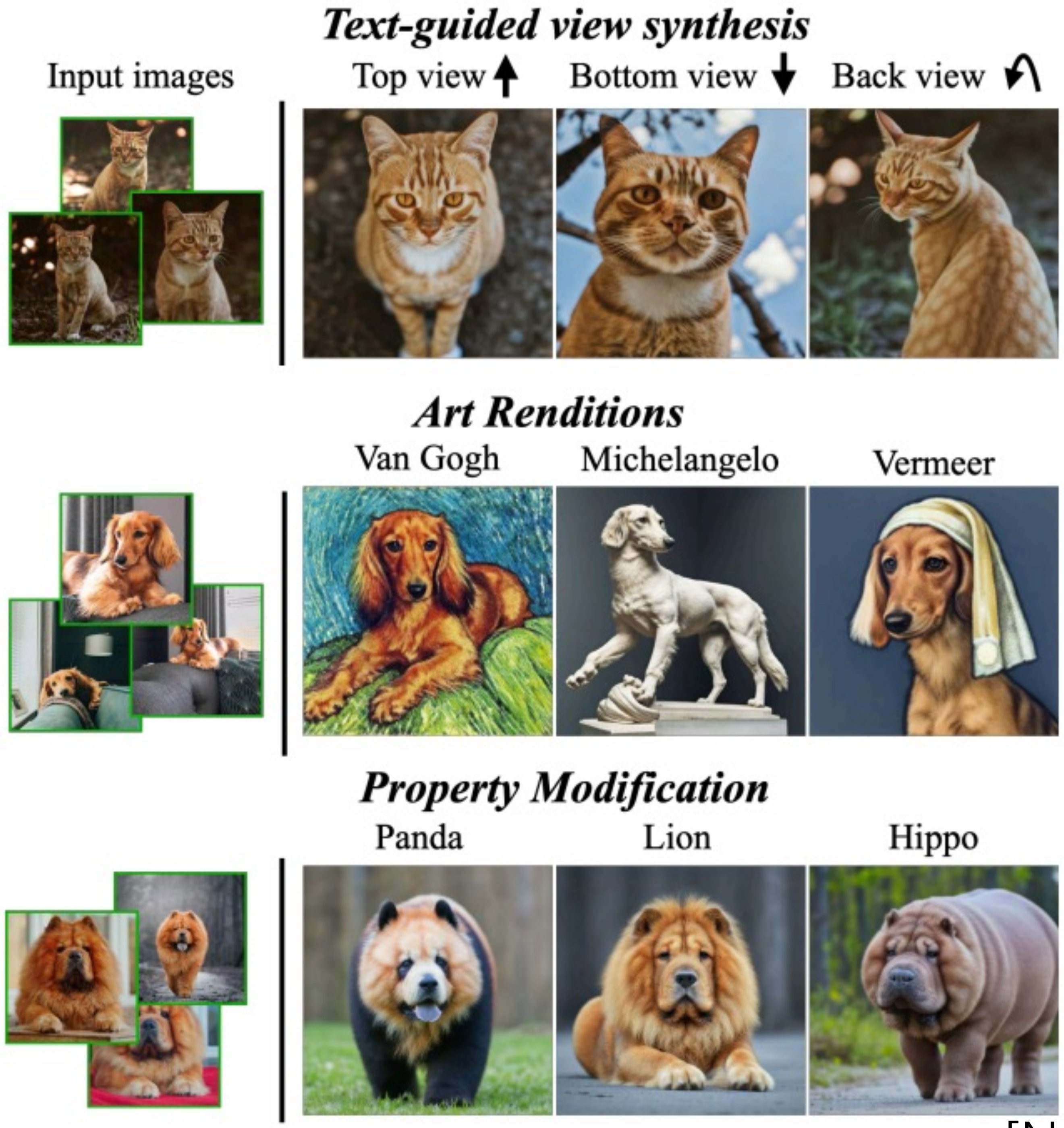
DreamBooth Results

[Nataniel Ruiz et al., CVPR 2023]

DreamBooth Results

[Nataniel Ruiz et al., CVPR 2023]

DreamBooth Applications



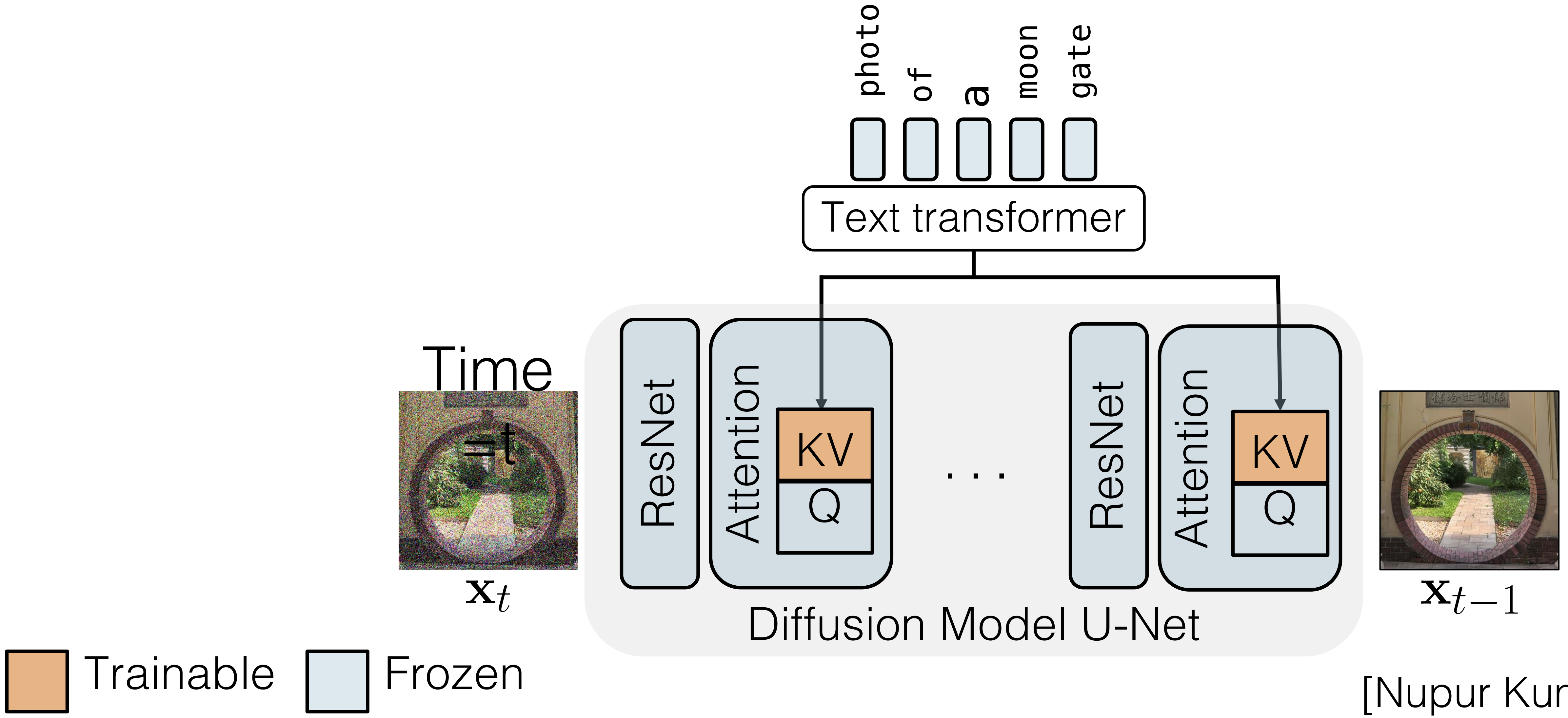
[Nataniel Ruiz et al., CVPR 2023]

practical concerns

1. training time (30-60 min)
2. storage per concept (3-4 GB)

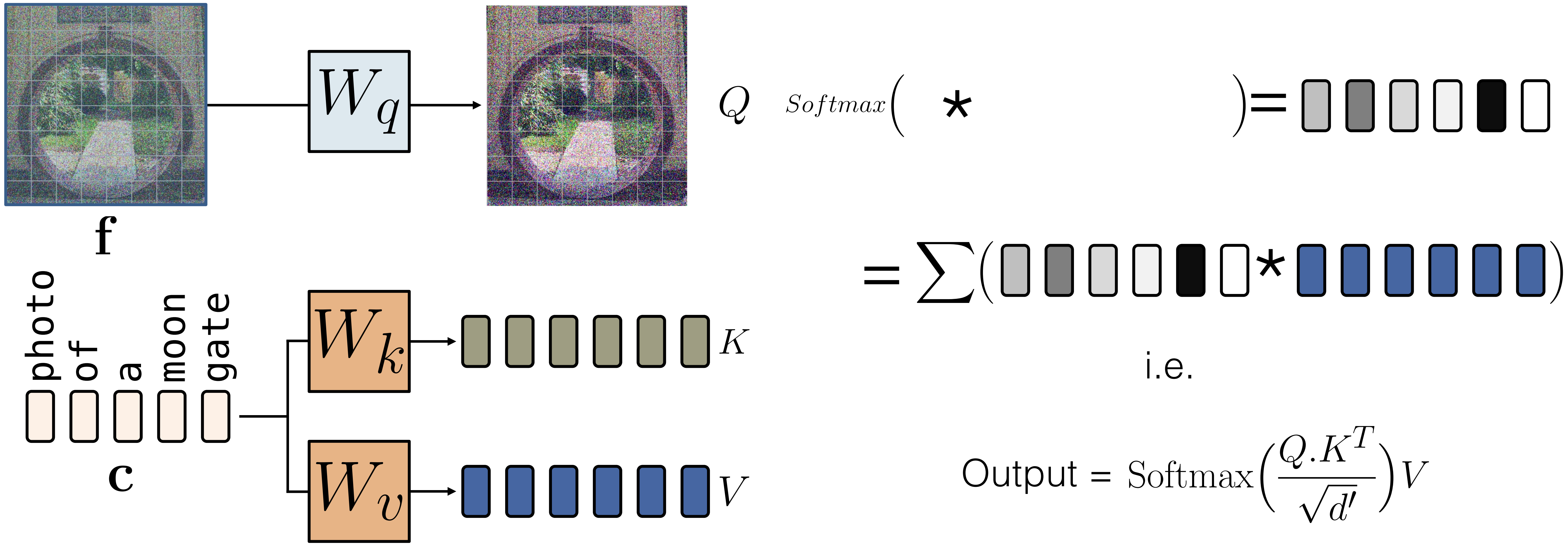
Efficient Custom Diffusion Training

Fine-tune key, value projection matrices in cross-attention layers



[Nupur Kumari et al., CVPR 2023]

Text-image Cross-Attention

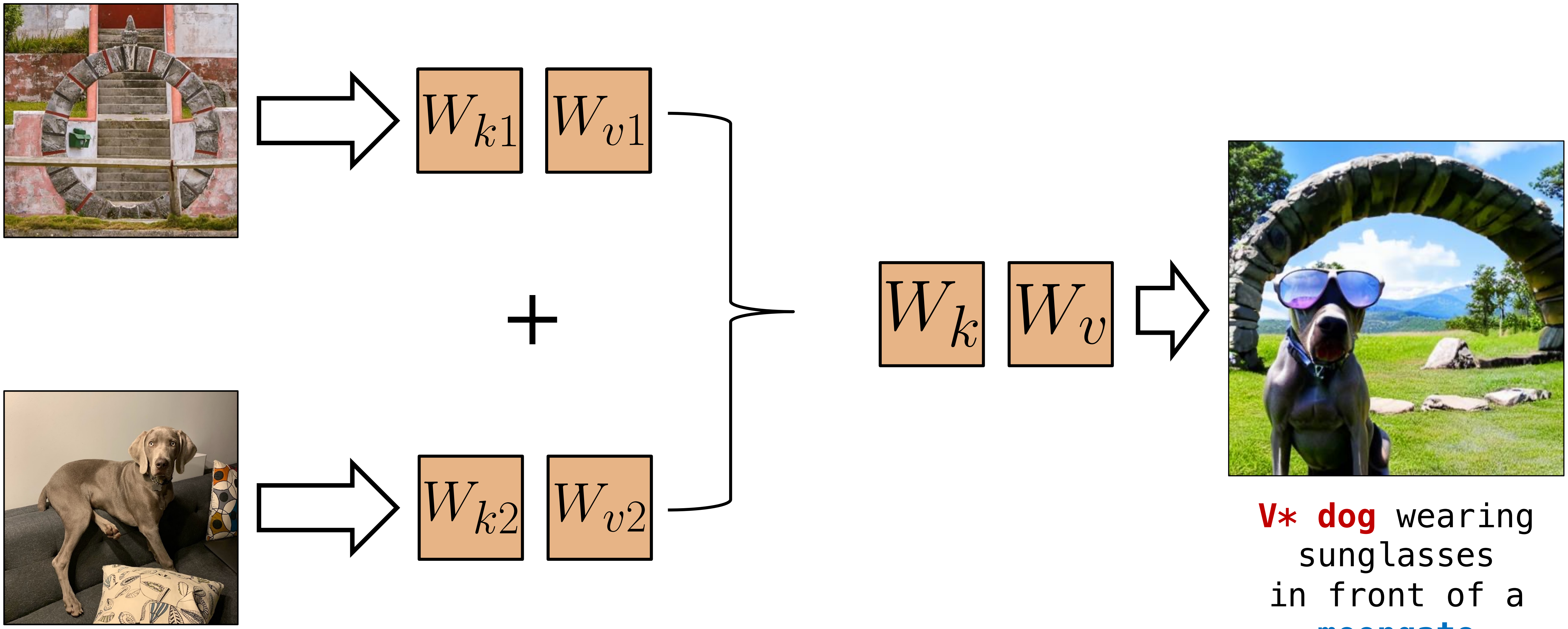


Trainable Frozen

[Nupur Kumari et al., CVPR 2023]

Custom Diffusion for multiple-concepts

Merging two concepts



V* dog wearing sunglasses in front of a moongate

Custom Diffusion for multiple-concepts

Merging two concepts

$$\hat{W} = \arg\min_{W} ||WC_{\text{reg}}^{\top} - W_0 C_{\text{reg}}^{\top}||_F$$

$$\text{s.t. } WC^{\top} = V, \text{ where } C = [\mathbf{c}_1 \cdots \mathbf{c}_N]^{\top}$$

$$\text{and } V = [W_1 \mathbf{c}_1^{\top} \cdots W_N \mathbf{c}_N^{\top}]^{\top}.$$

C_{reg} : a collection of random text prompts.

C : target prompts.

[Nupur Kumari et al., CVPR 2023]

Custom Diffusion for multiple-concepts

Merging two concepts

$$\hat{W} = W_0 + \mathbf{v}^\top \mathbf{d}, \textbf{ where } \mathbf{d} = C(C_{\text{reg}}^\top C_{\text{reg}})^{-1}$$
$$\textbf{and } \mathbf{v}^\top = (V - W_0 C^\top)(\mathbf{d} C^\top)^{-1}.$$

C_{reg} : a collection of random text prompts.
C : target prompts.

[Nupur Kumari et al., CVPR 2023]

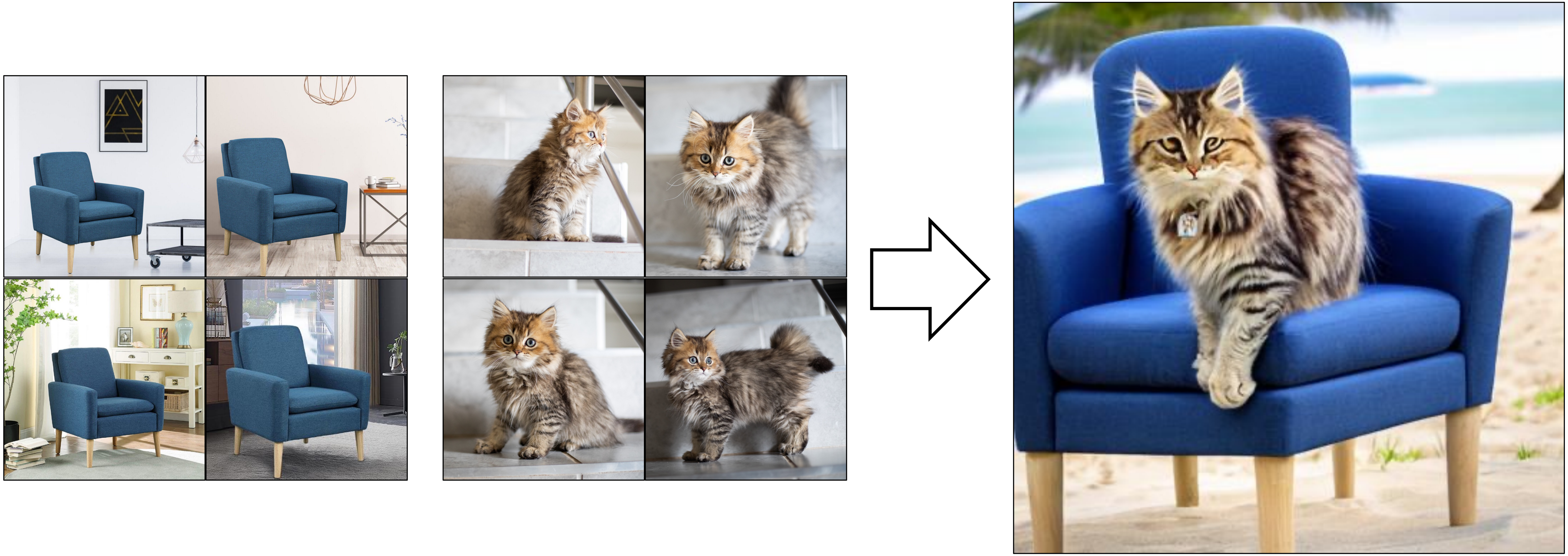
More examples

V_1* chair with the V_2* cat sitting on it near a beach

More examples

V_1^* chair with the V_2^* cat sitting on it near a beach

More examples



V_1* chair with the V_2* cat sitting on it near a beach

More examples

The V_1^* cat is sitting inside a V_2^* wooden pot and looking up

More examples

The V_1* cat is sitting inside a V_2* wooden pot and looking up

More examples

The V_1* cat is sitting inside a V_2* wooden pot and looking up

More examples

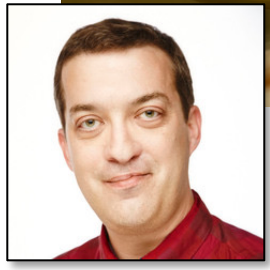
V_1* flower in the V_2* wooden pot on a table

More examples

V_1^* flower in the V_2^* wooden pot on a table

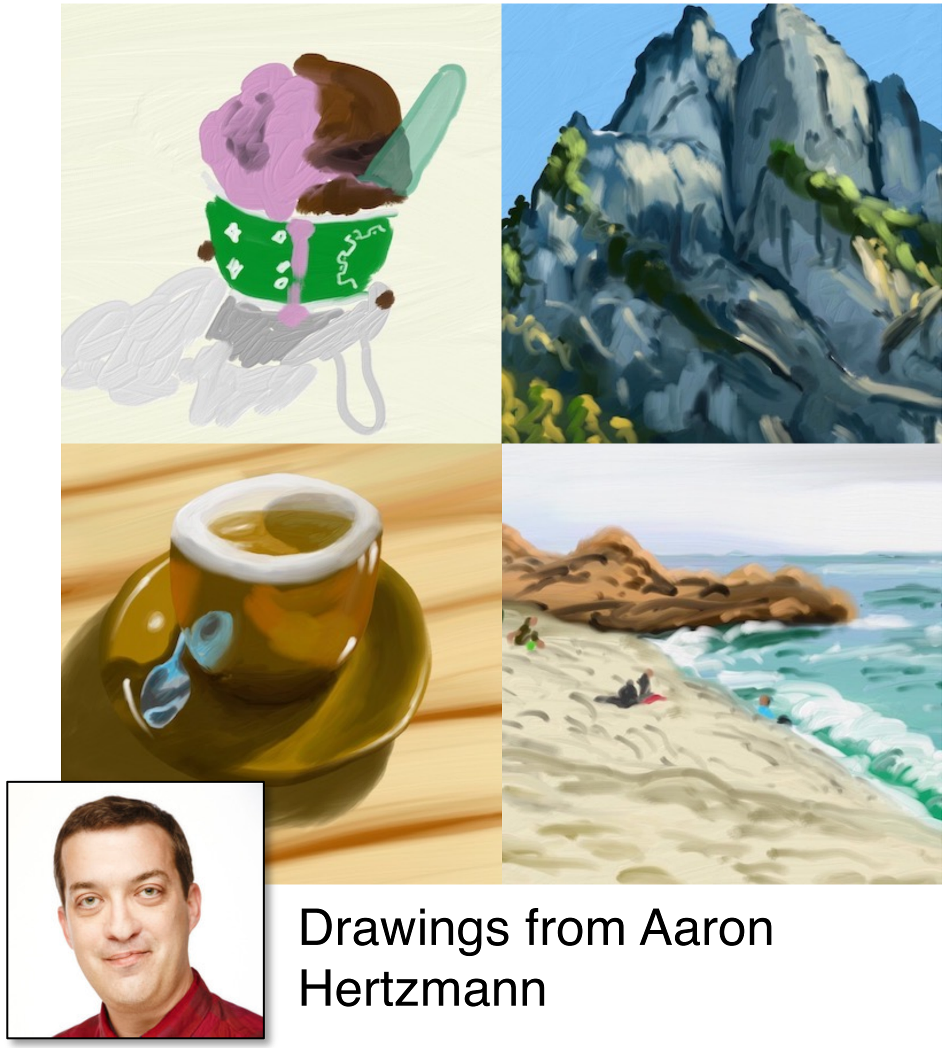
More examples

V_1* flower in the V_2* wooden pot on a table



Drawings from Aaron Hertzmann

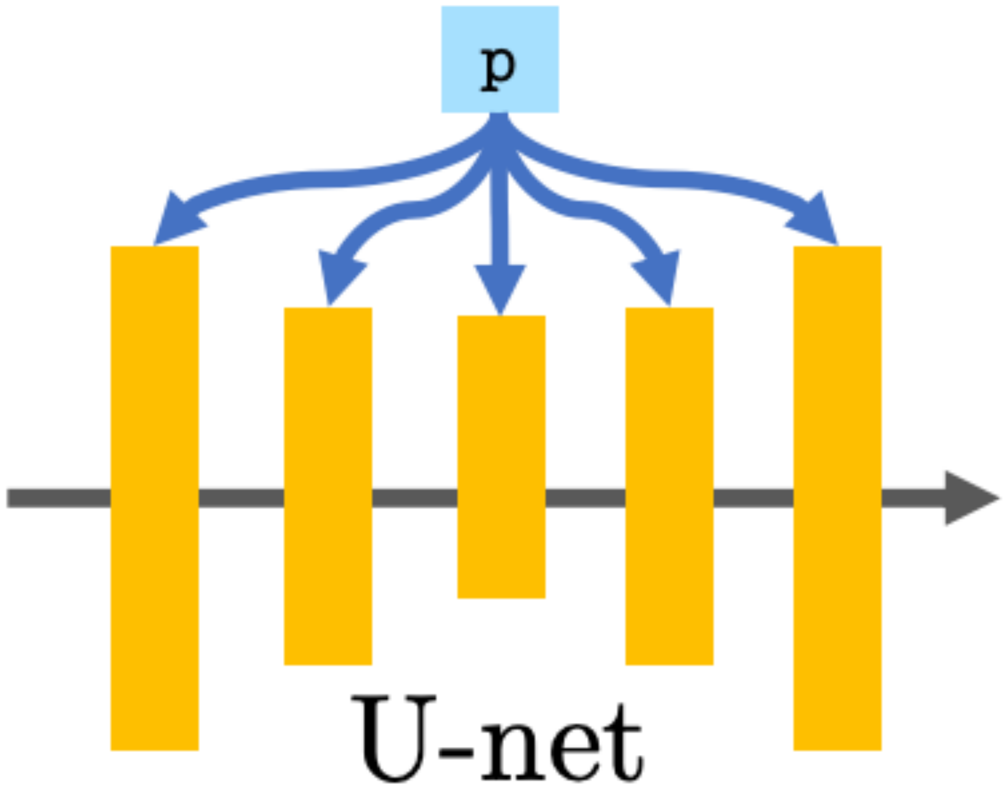
Plant painting in style of V* art

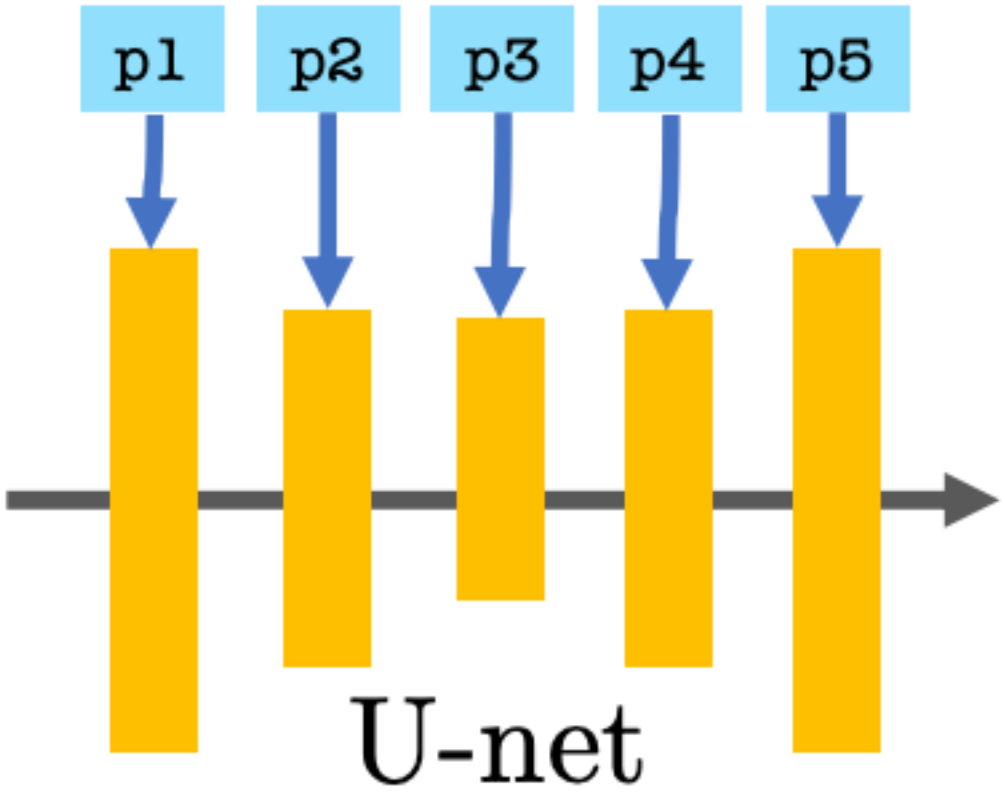


Drawings from Aaron Hertzmann

Painting of dog in style of V* art

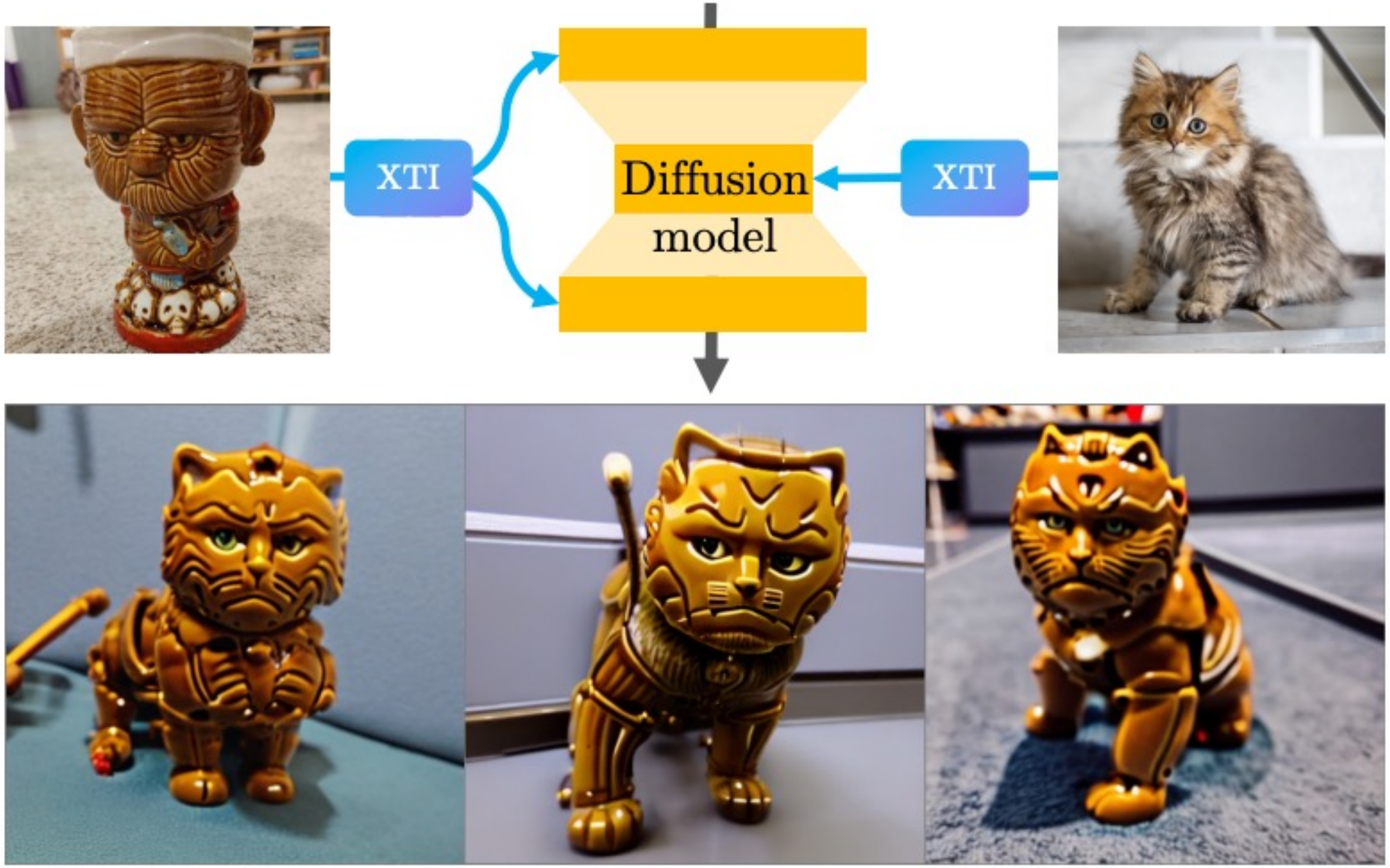
Extended Textual Inversion





[Andrey Voynov et al., arXiv 2023]

Shape-Style Mixing



[Andrey Voynov et al., arXiv 2023]

Extended Textual Inversion

Real | Textual Inversion | Extended Textual Inversion

<teddy bear> in Times Square

<cat> wearing sunglasses

[Andrey Voynov et al., arXiv 2023]