Dissecting neural nets

Antonio Torralba

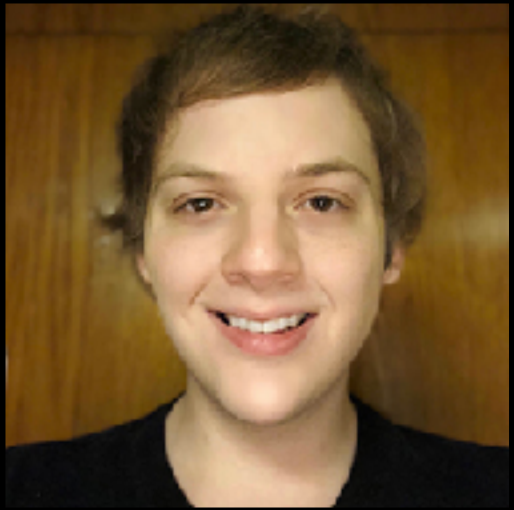
David Bau, Hendrik Strobelt, William Peebles
Jonas Wulff, Bolei Zhou, Jun-Yan Zhu

MIT Quest for Intelligence

CSAIL

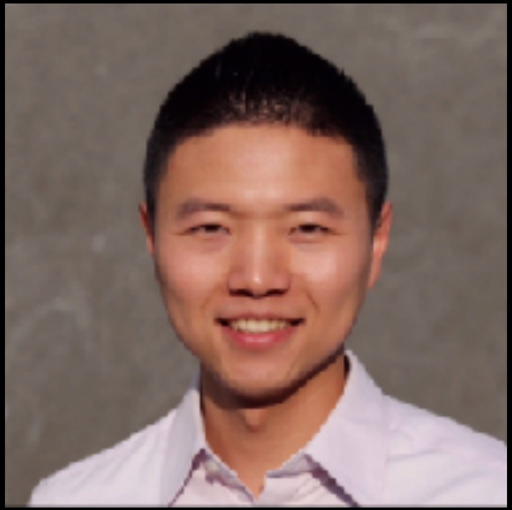
David Bau

Hendrik Strobelt



William Peebles

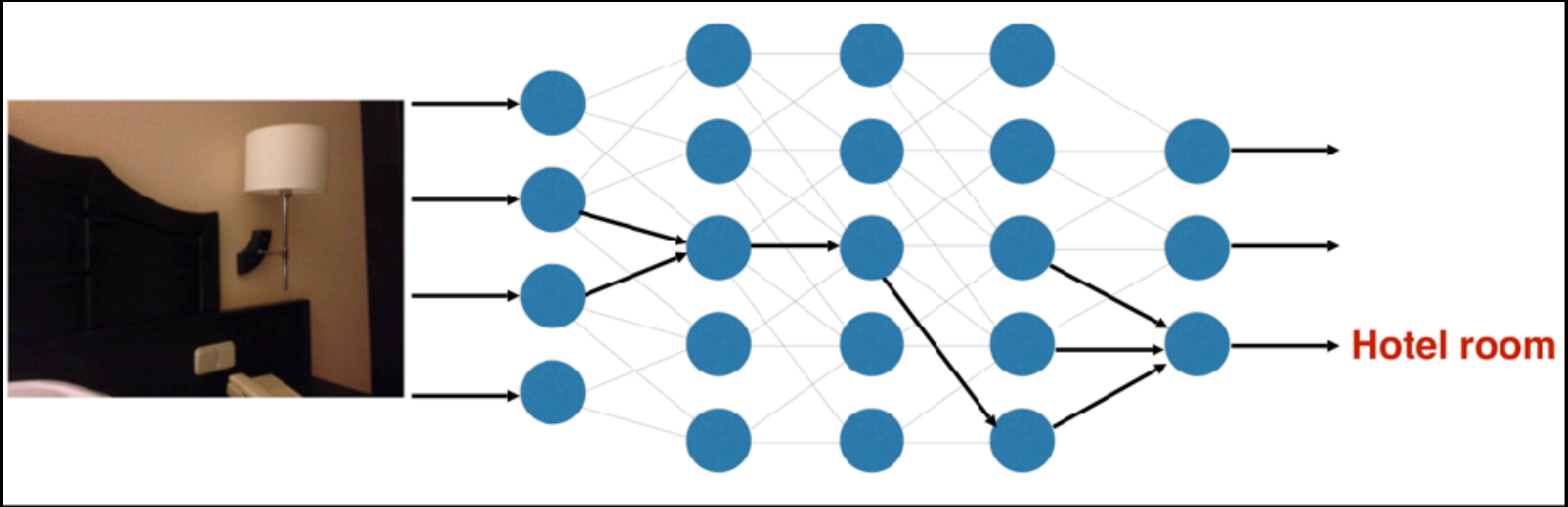
Jonas Wulff



Bolei Zhou

Jun-Yan Zhu

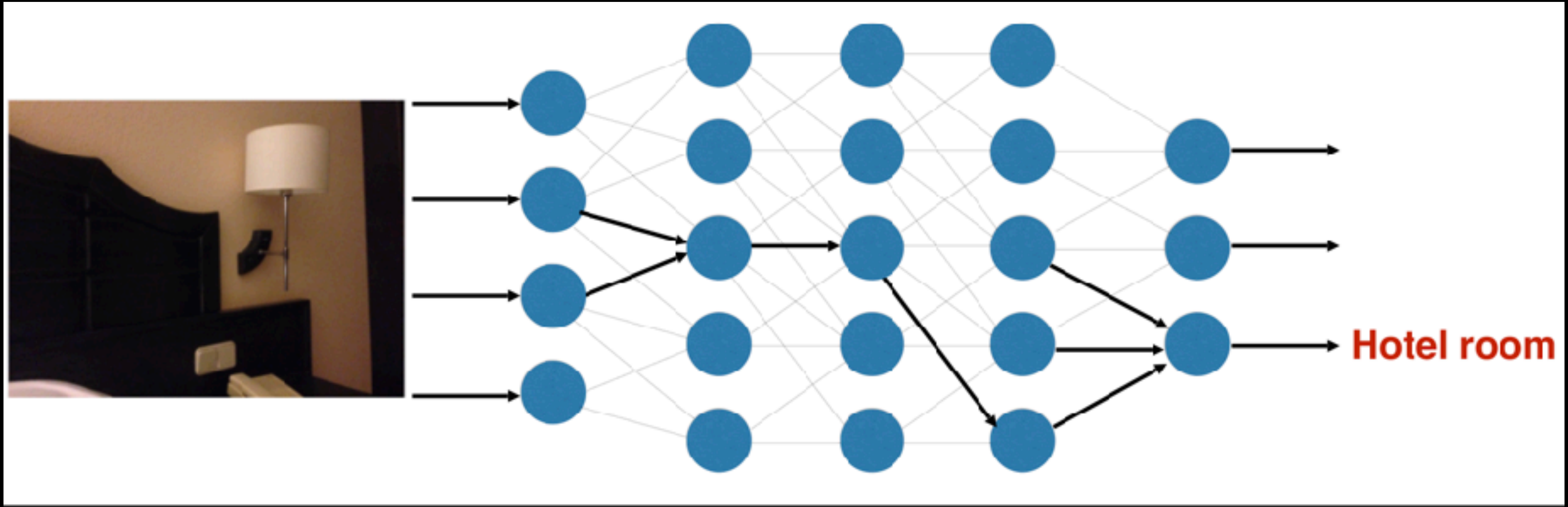
Neural nets



Neural nets

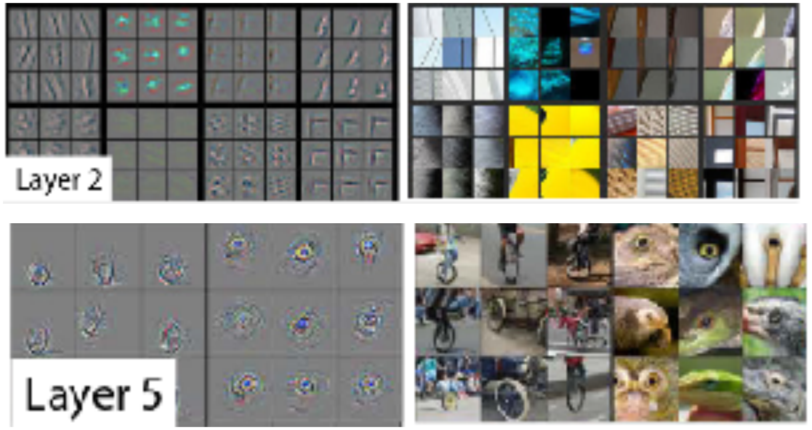

Take/Choose a photo
Predictions:
type: indoor
semantic categories:
coffee_shop:0.47, restaurant:0.17,
cafeteria:0.08, food_court:0.06,
Take/Choose a photo
Predictions:
type: indoor
semantic categories:
supermarket:0.96,
Take/Choose a photo
Predictions:
type: indoor
semantic categories:
conference_center:0.51,
auditorium:0.12, office:0.08,

Neural nets



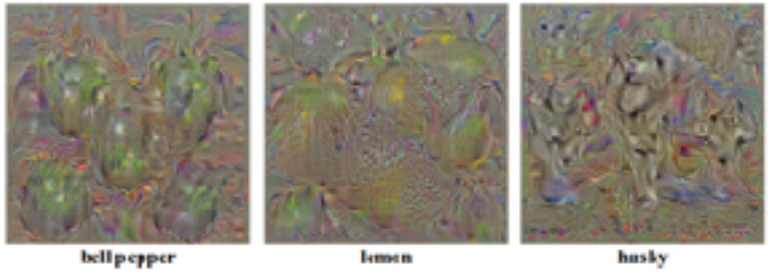
Visualizing the learned representation

Deconvolution



Zeiler, M. et al., ECCV 2014.

Back-propagation



Simonyan, K. et al.. ICLR’15

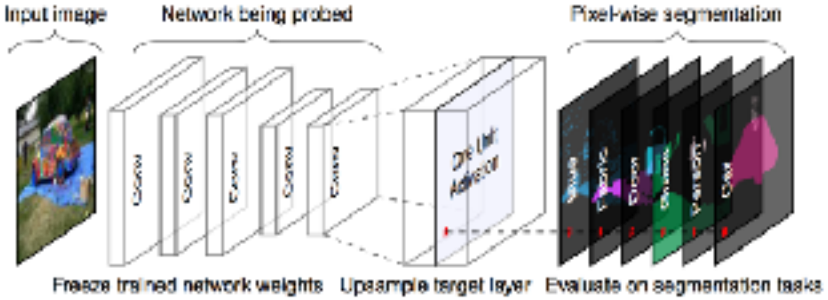
Inceptionism. Google Blog. June 2015

Top activated images

Girshick, R., et al. CVPR 2014

A. Mahendran and A. Vedaldi, CVPR 2015

Network dissection



D. Bau et al. CVPR 2017

Understanding the representation

MNIST

IMAGENET

brambling

terrier

Places

bedroom

mountain

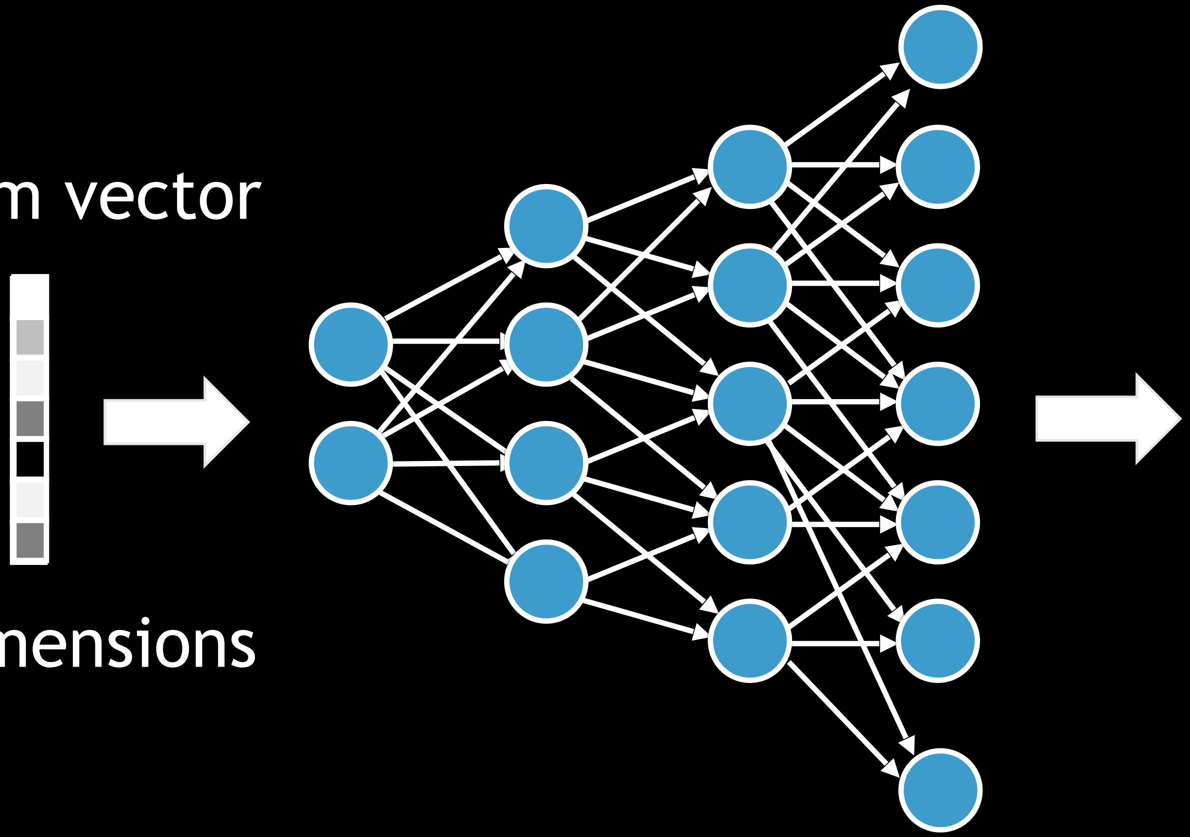
Action recognition

COCO
Common Objects in Context

A horse carrying a large load of hay and two people sitting on it.

Generative Adversarial Network (GAN)

Random vector



512 dimensions

Randomly generated image

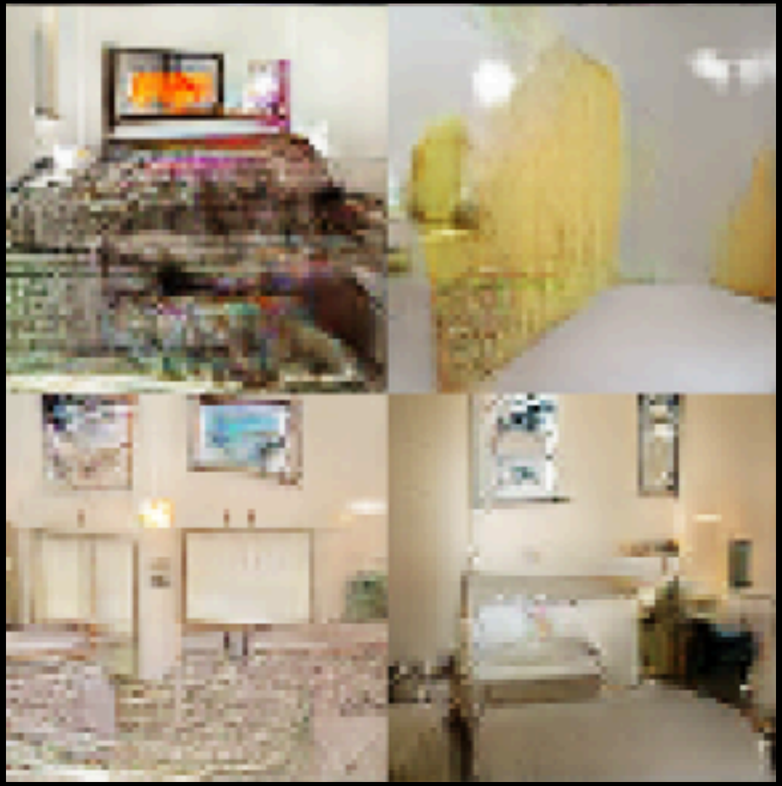
GANs [Goodfellow et al.]

Generative Adversarial Network (GAN)

2014

GANs [Goodfellow et al.]

2016



DCGAN [Radford et al.]

2018

…

ProgGAN [Karras et al.]

To render a scene,
What does a GAN need to know?

Progressive GANs [Karras et al., ICLR 2018]

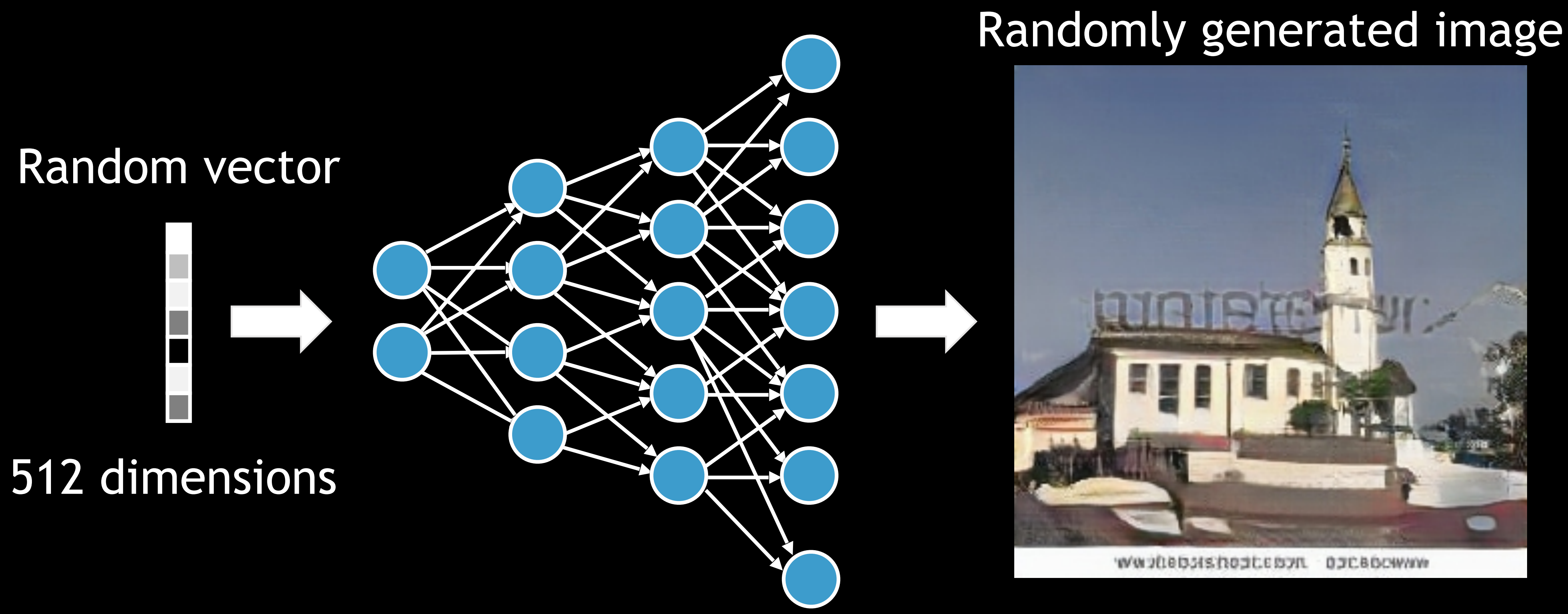
Progressive GANs [Karras et al., ICLR 2018]





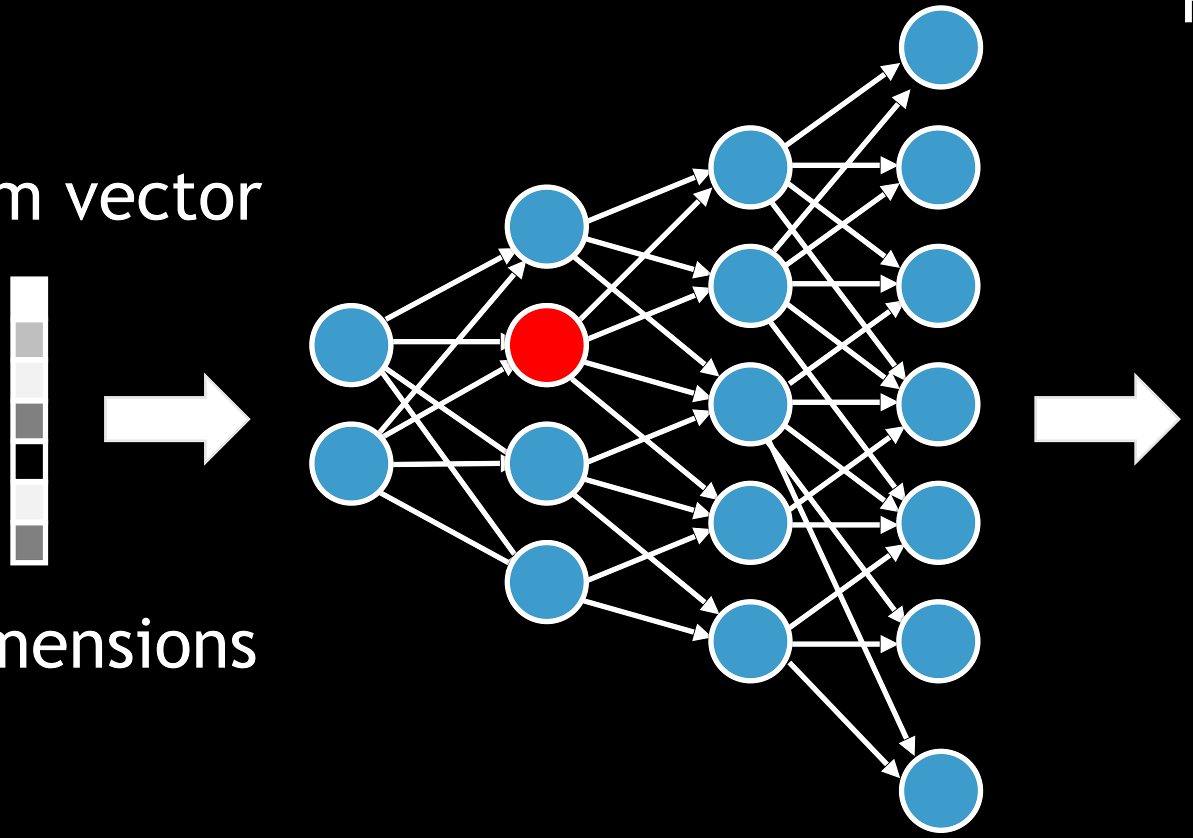



Are there watermark neurons?



Are there watermark neurons?

Random vector



512 dimensions

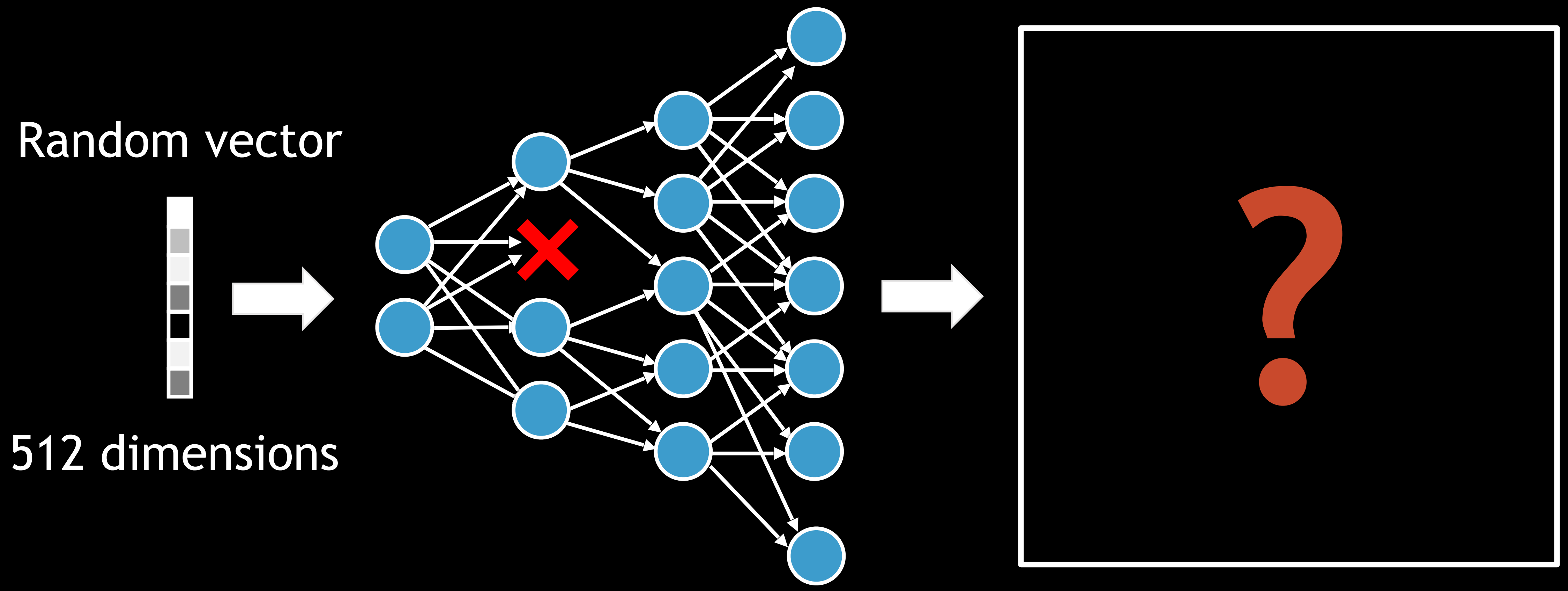
Randomly generated image

Layer 4, Neuron 201

Layer 4, Neuron 445

… and five more neurons

What if we turn off these neurons?







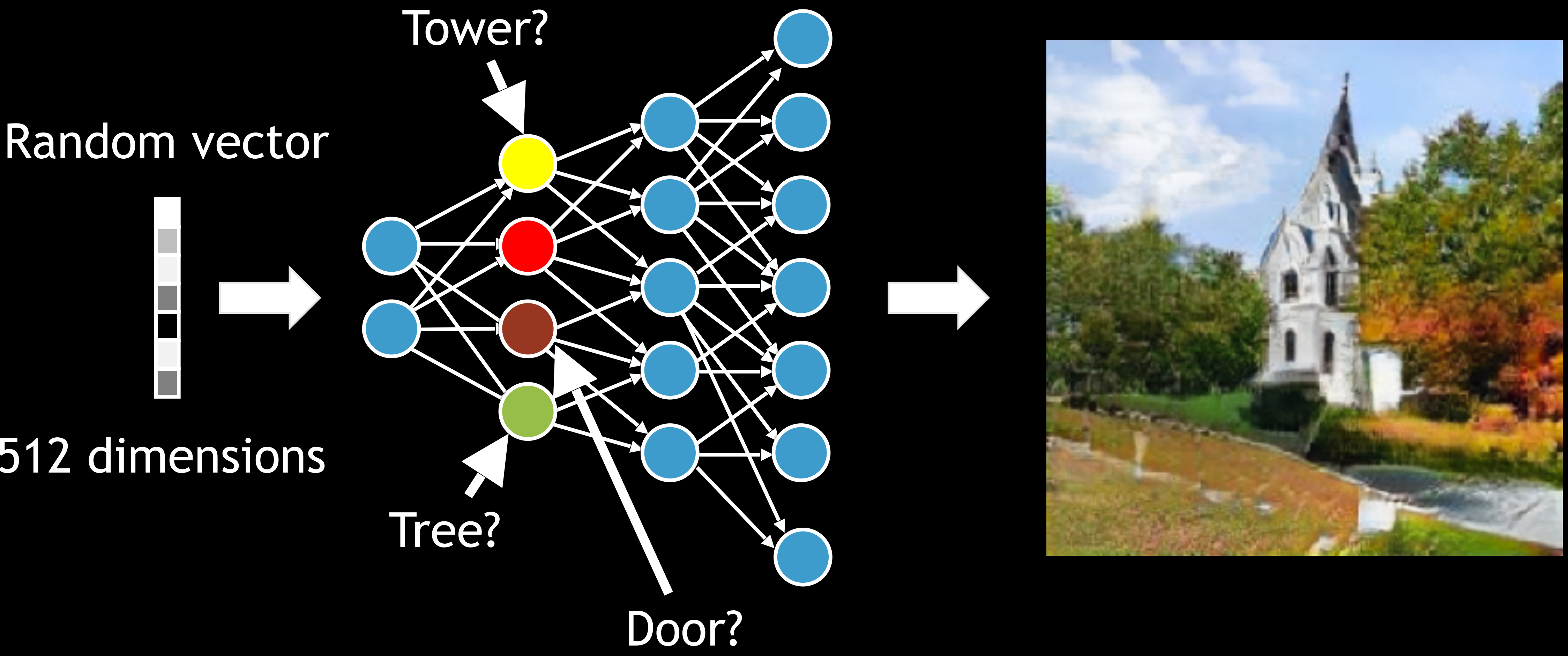



Deactivating banner neurons in layer 4

Deactivating watermak neurons

Deactivating 31 units in layer 4

Are there other objects?

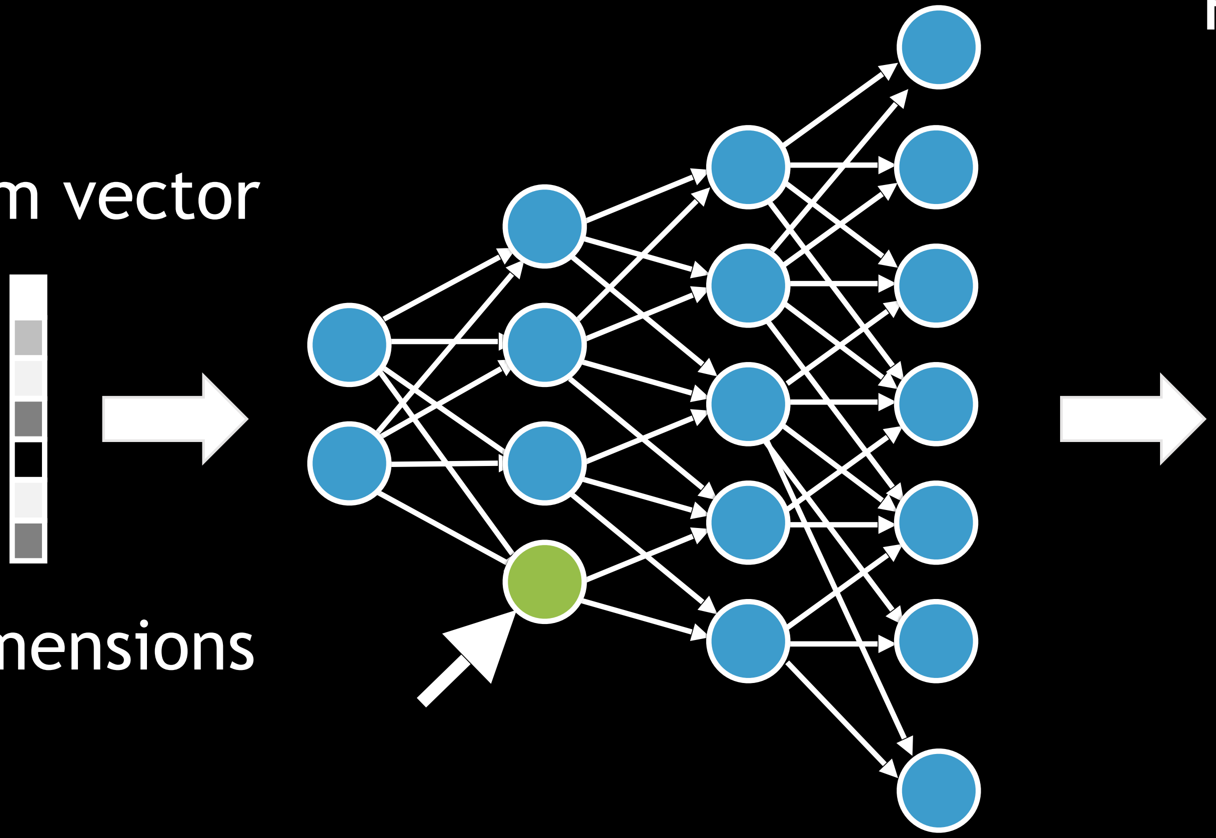


GAN Dissection [Bau et al., ICLR 2019]

Are there other objects?

Random vector

512 dimensions



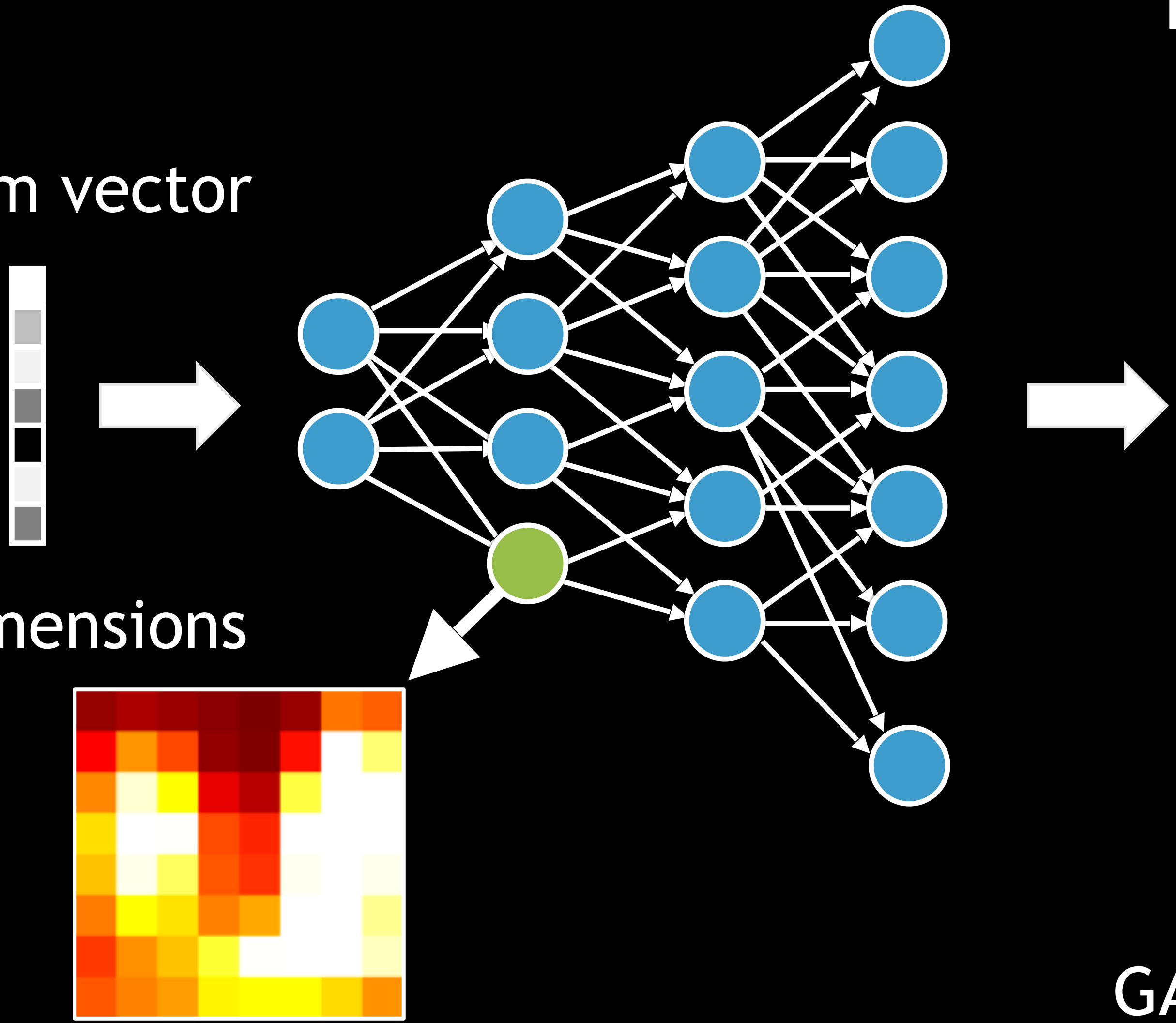
Randomly generated image

GAN Dissection [Bau et al., ICLR 2019]

Are there other objects?

Random vector

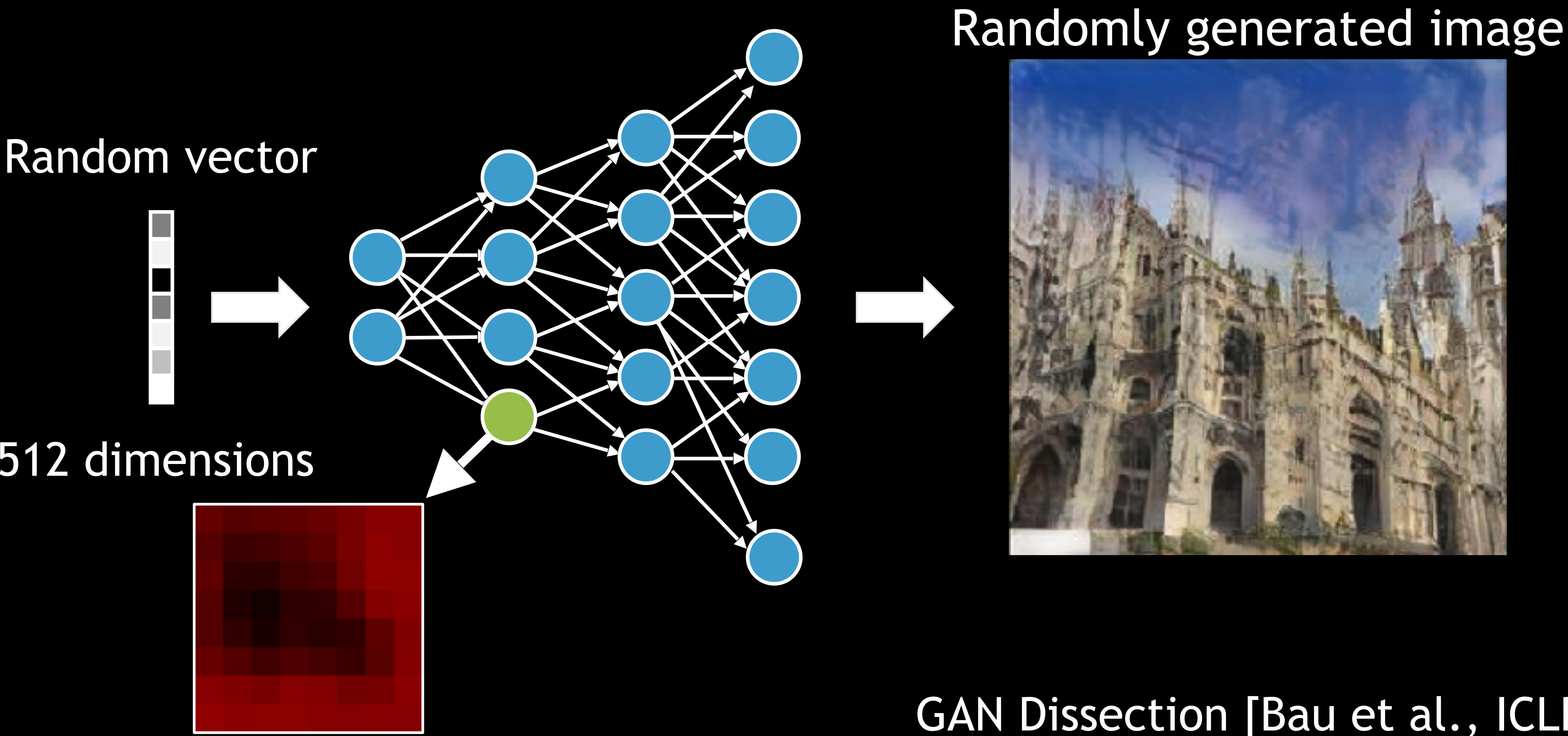
512 dimensions



Randomly generated image

GAN Dissection [Bau et al., ICLR 2019]

Are there other objects?

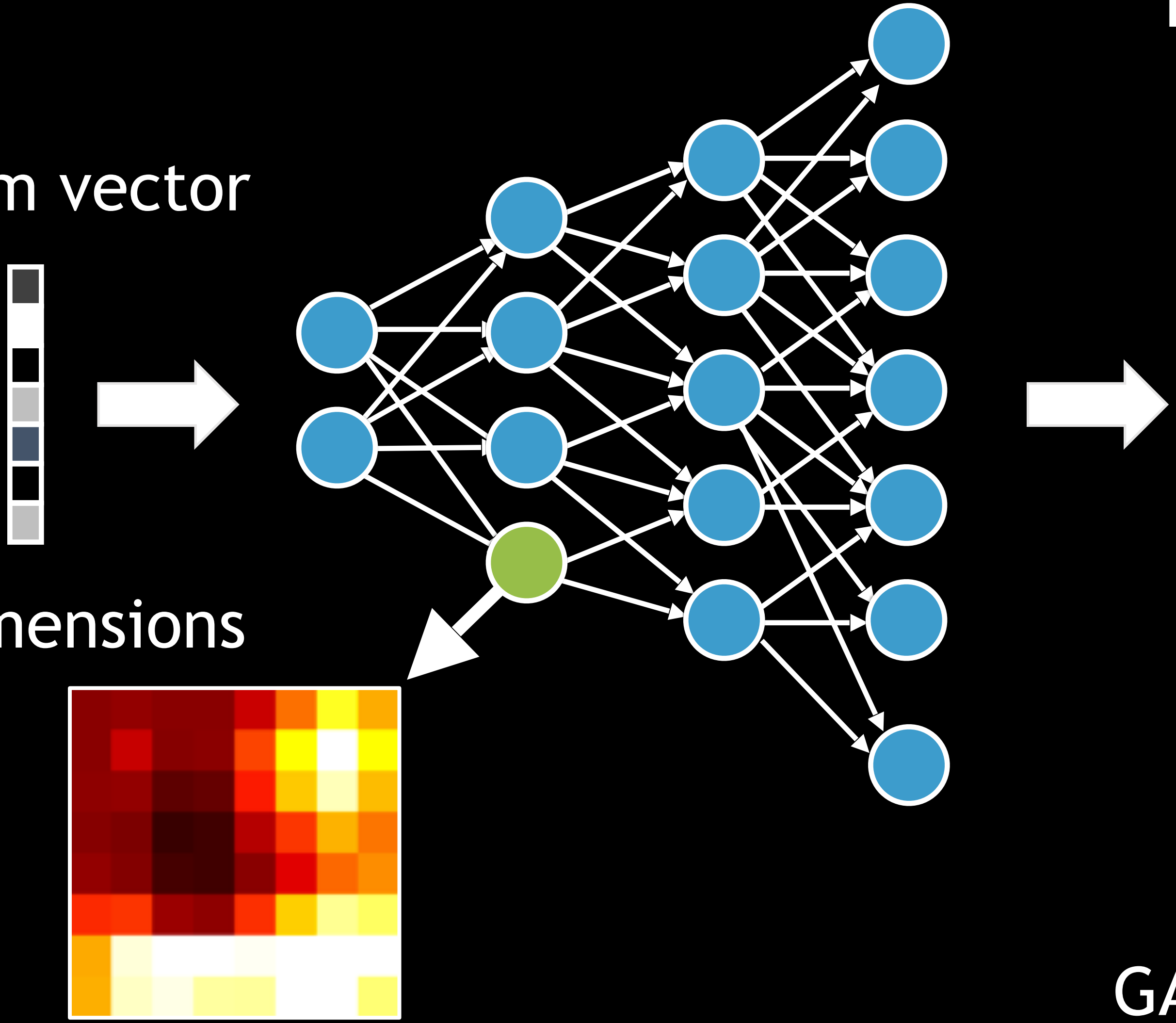


GAN Dissection [Bau et al., ICLR 2019]

Are there other objects?

Random vector

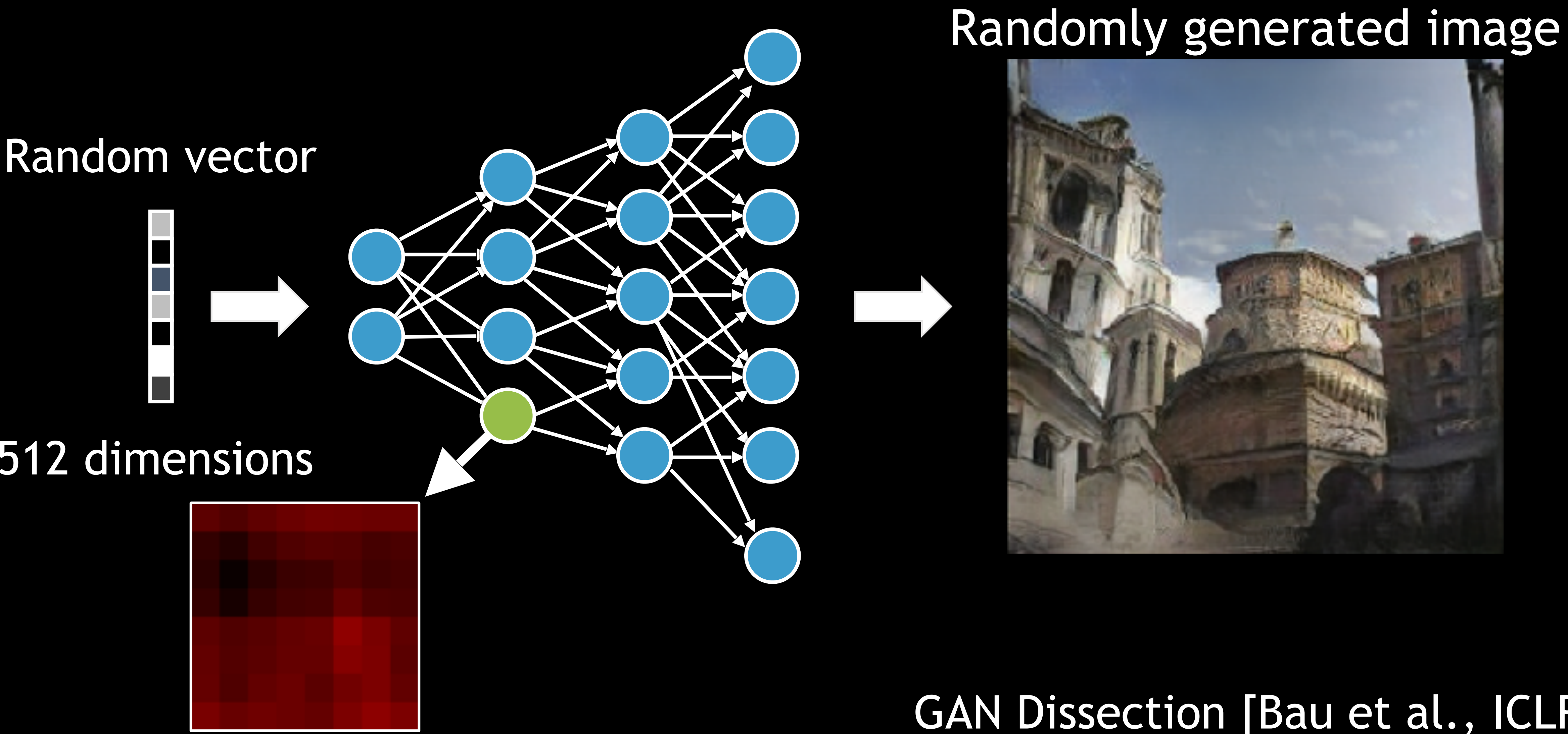
512 dimensions



Randomly generated image

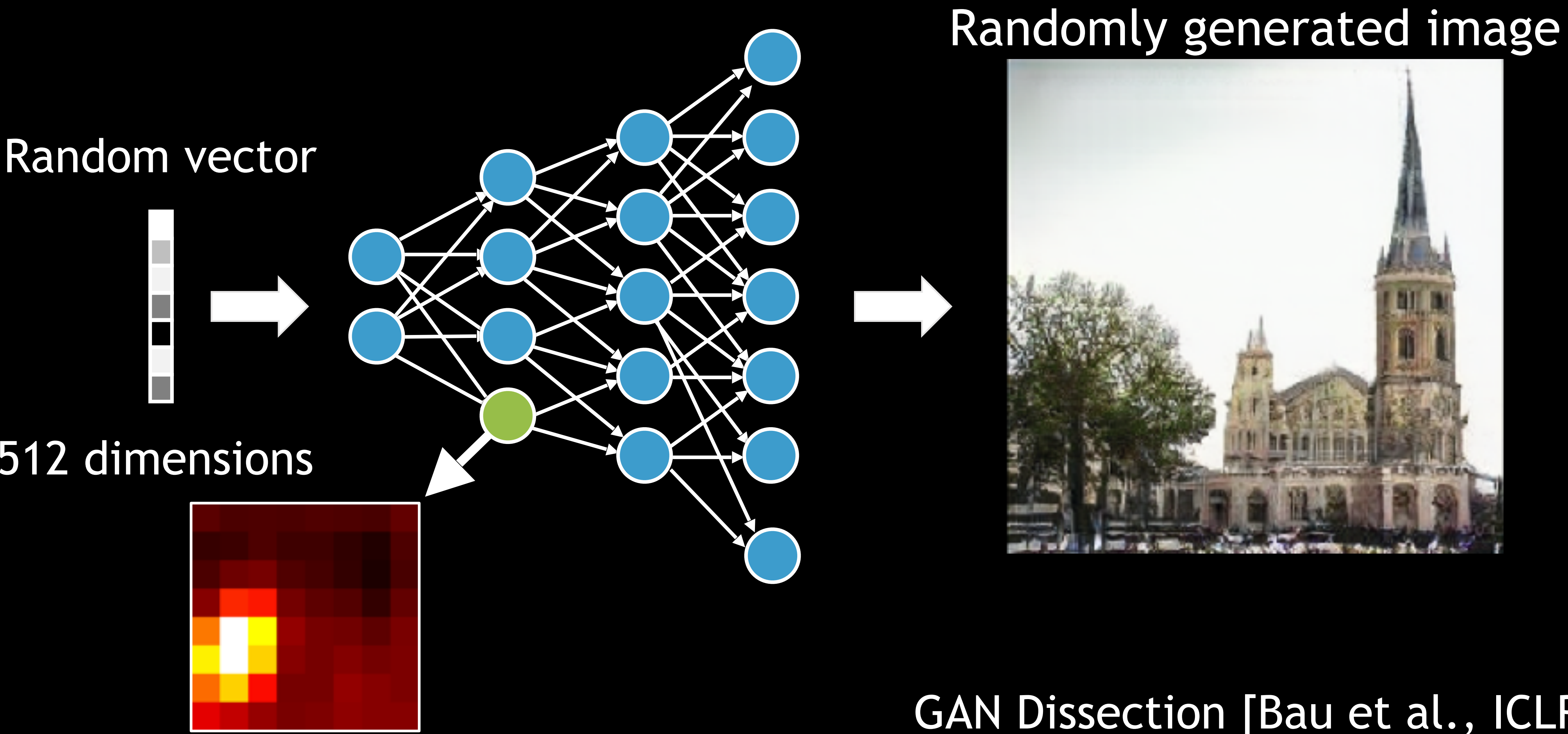
GAN Dissection [Bau et al., ICLR 2019]

Are there other objects?



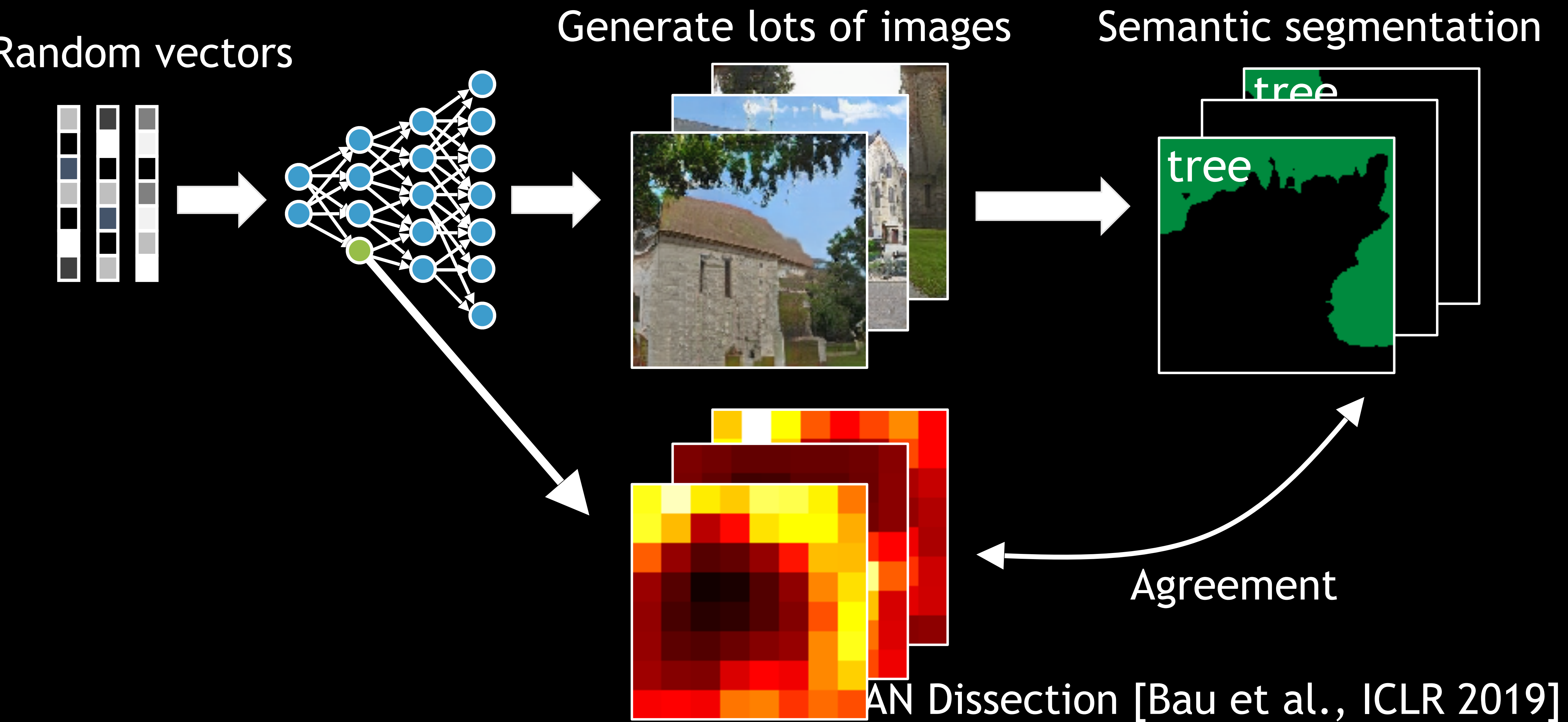
GAN Dissection [Bau et al., ICLR 2019]

Are there other objects?



GAN Dissection [Bau et al., ICLR 2019]

Dissecting a GAN



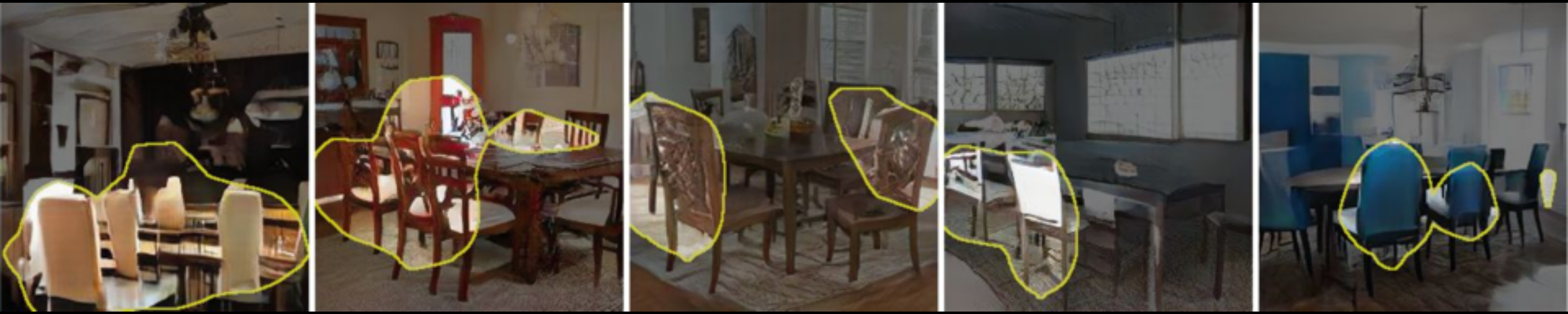
GAN Dissection [Bau et al., ICLR 2019]

Layer 4, Neuron 119: tree

Layer 4, Neuron 43 : dome

Layer 4, Neuron 84: window

Layer 4, Neuron 315: chair



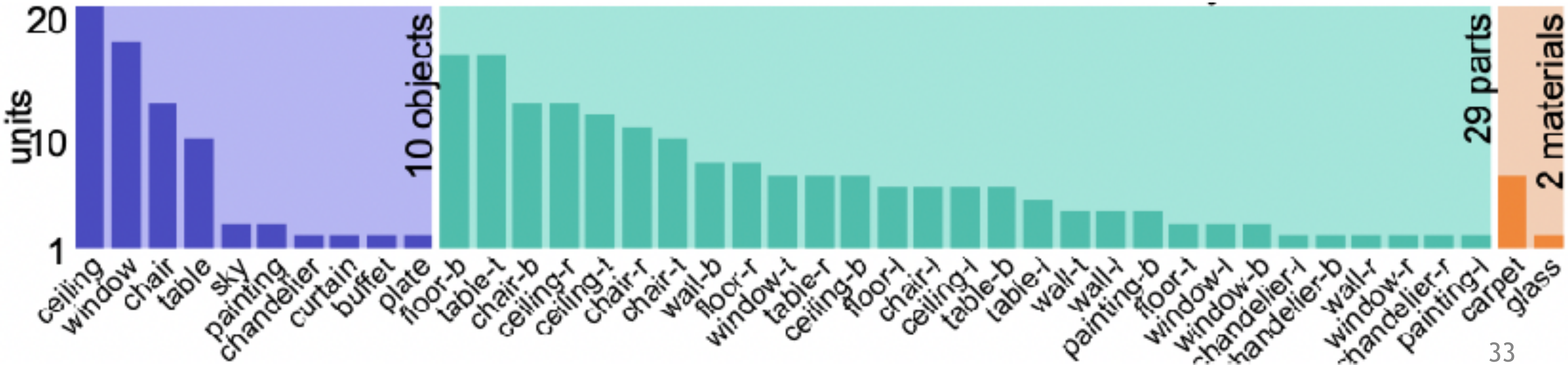
Which units **correlate** to an object class?

Dining room samples

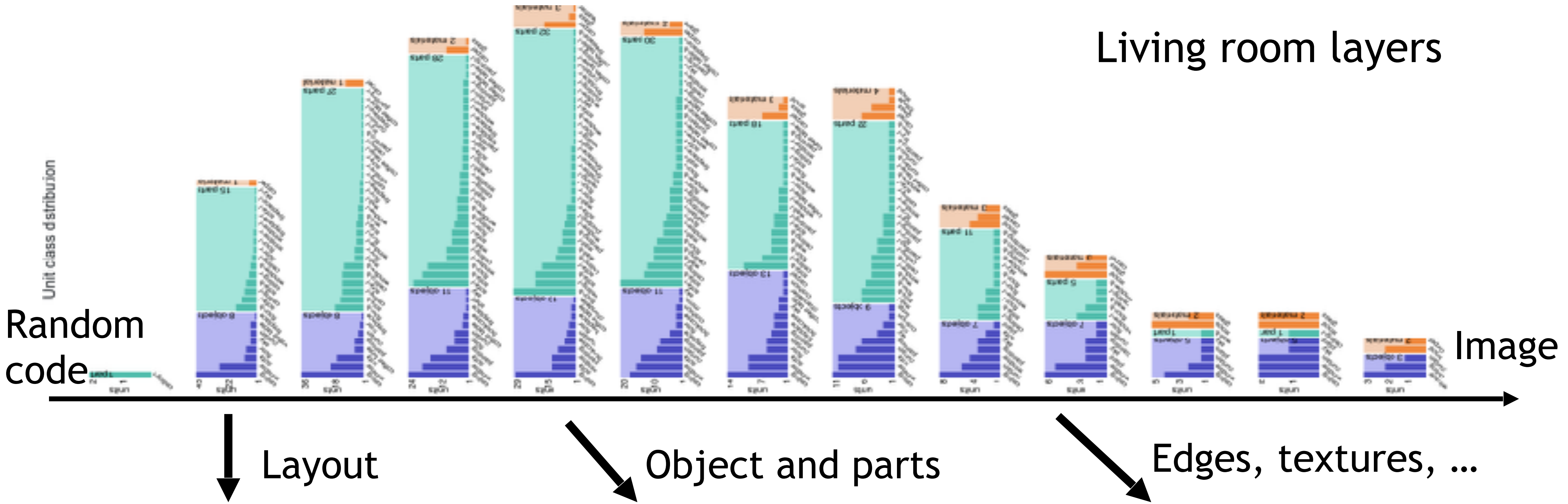
Which units **correlate** to an object class?

Dining room samples

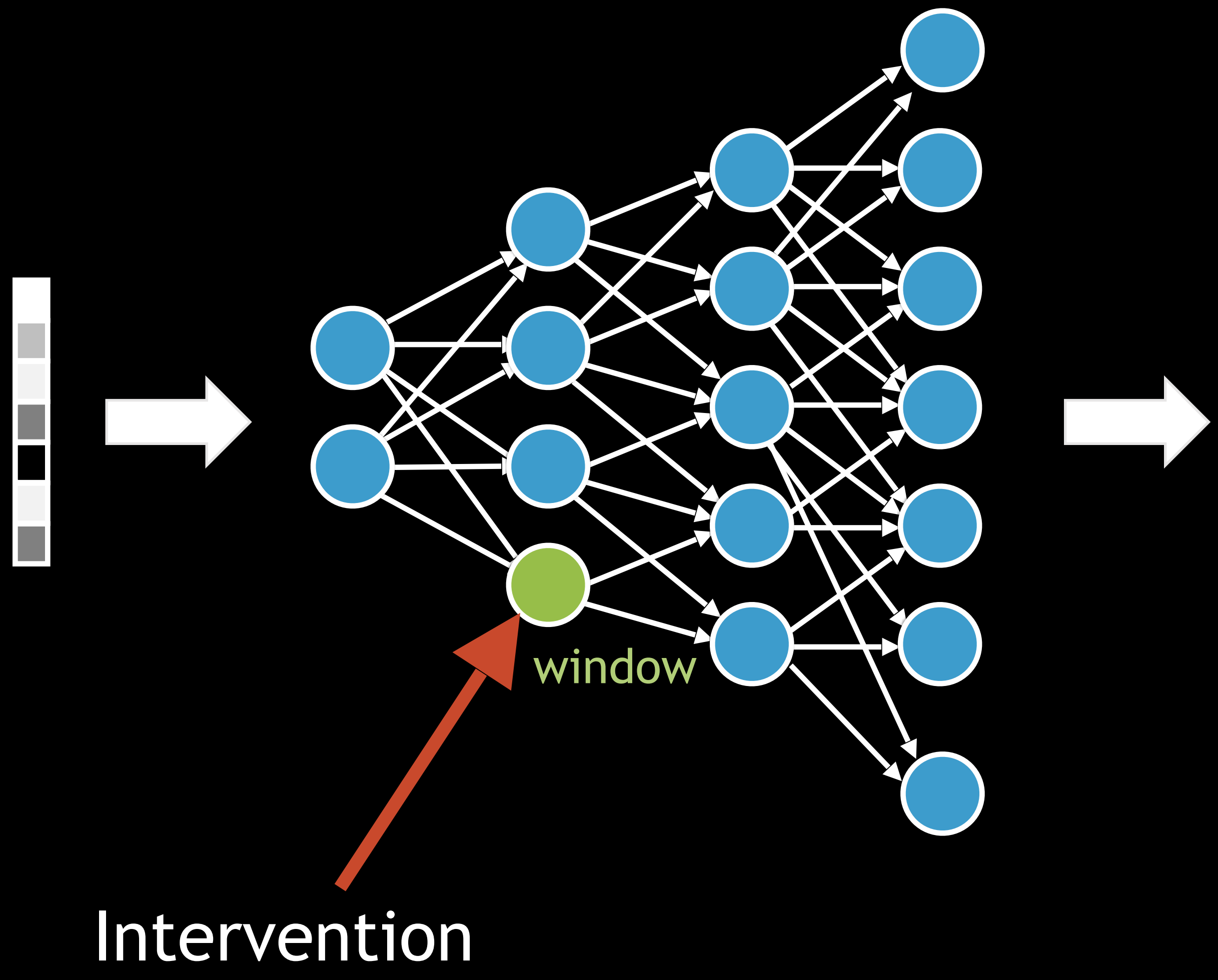
252 out of 512 units are correlated to objects, part, and materials



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



Manipulating units



Turning off window neurons

Turning off window neurons

Turning off window neurons

Turning off people neurons

Turning off people neurons

Turning on grass neurons

Turning on grass neurons

We can generate new compositions outside the training set

Generating and discovering new styles

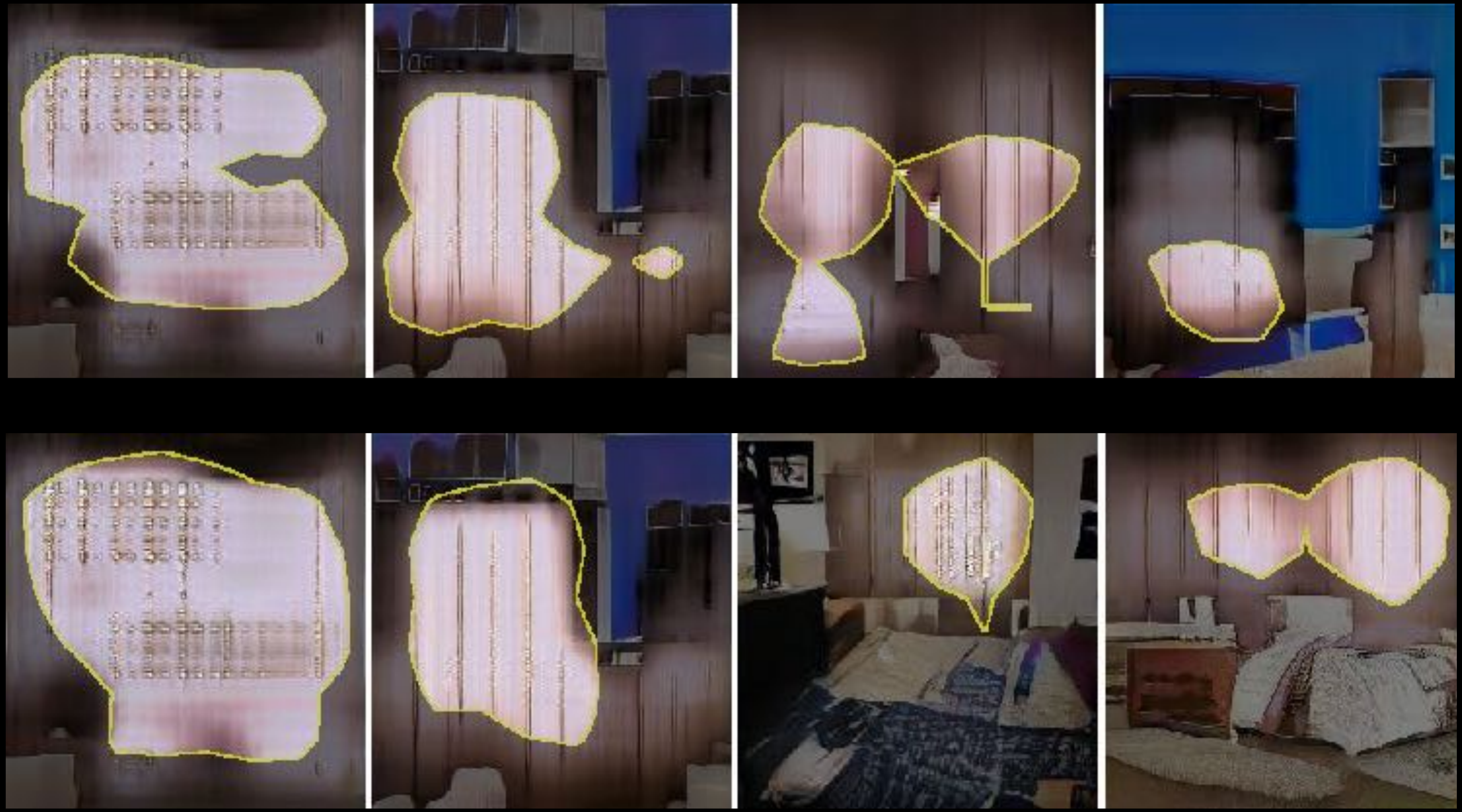
Improving GAN results

Bedroom images with artifacts

Are there artifact-causing units?

Unit #63

Unit #231



Improving GAN results

Deactivating problematic units

Improving GAN results

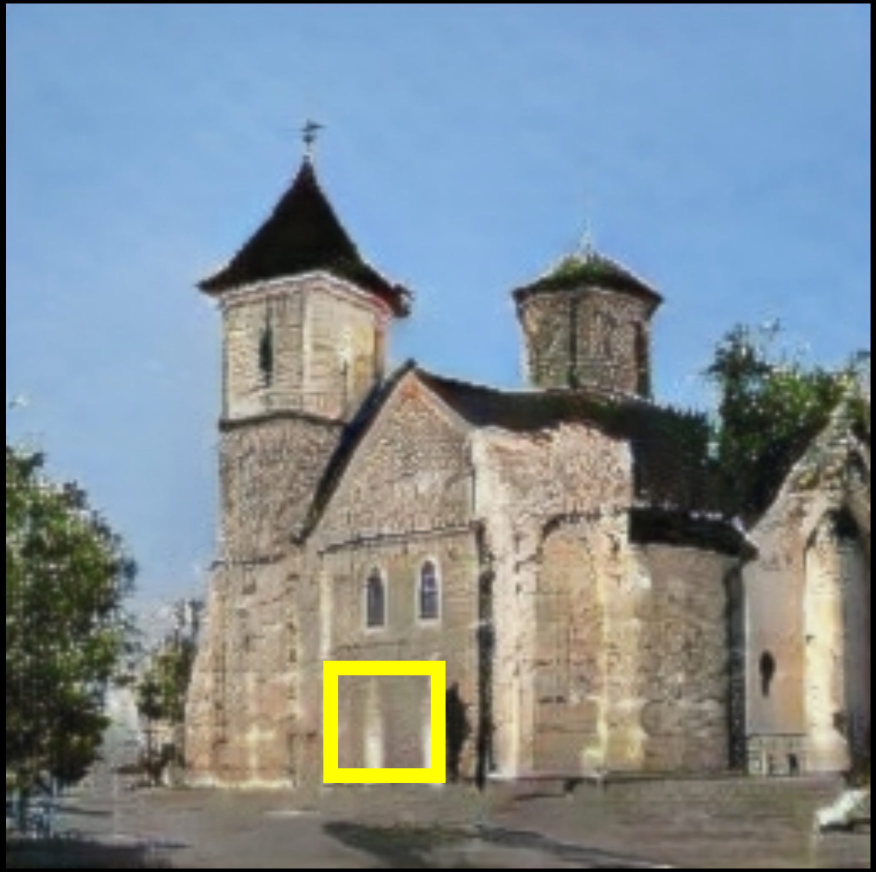
Ablating “artifact” units improves results

Activating **window** units in a location

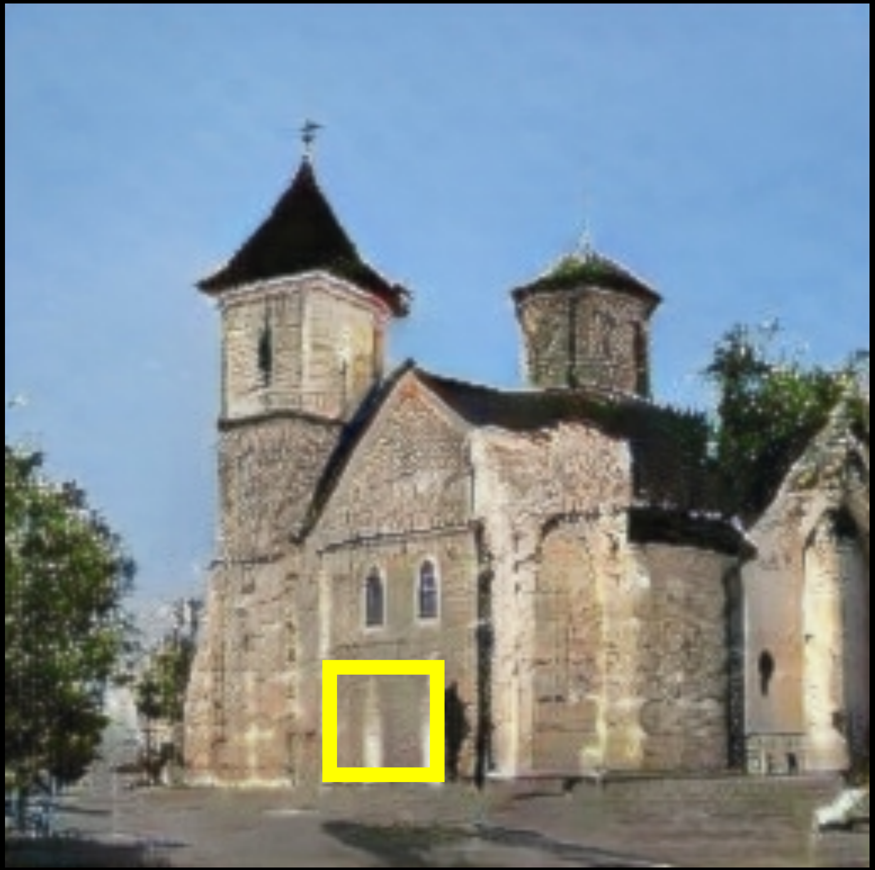
Activating **window** units in a location

Activating **window** units in a location

Turning on door neurons



Turning on door neurons



Turning on door neurons

Turning on door neurons

Turning on door neurons

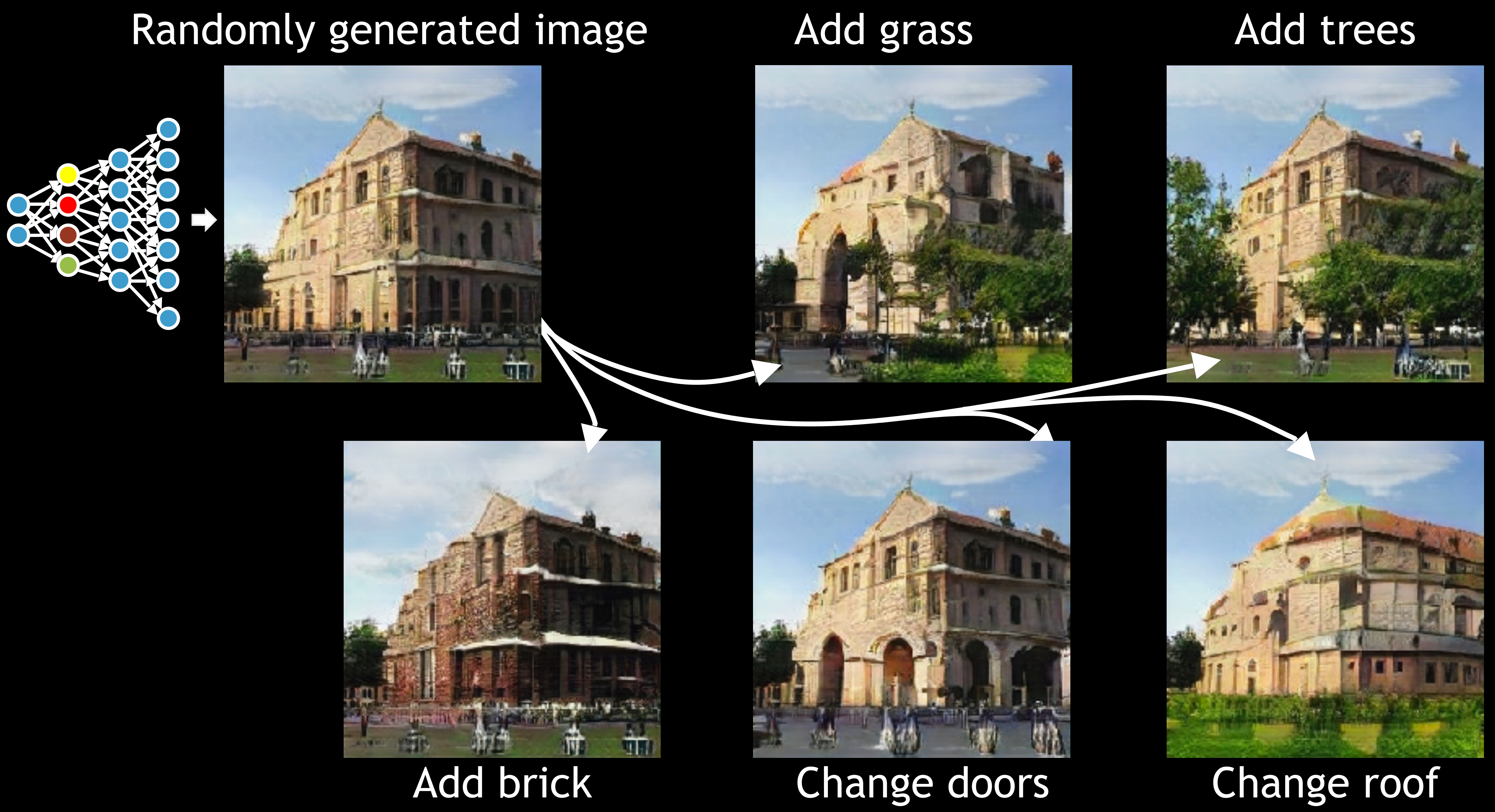
Turning on door neurons

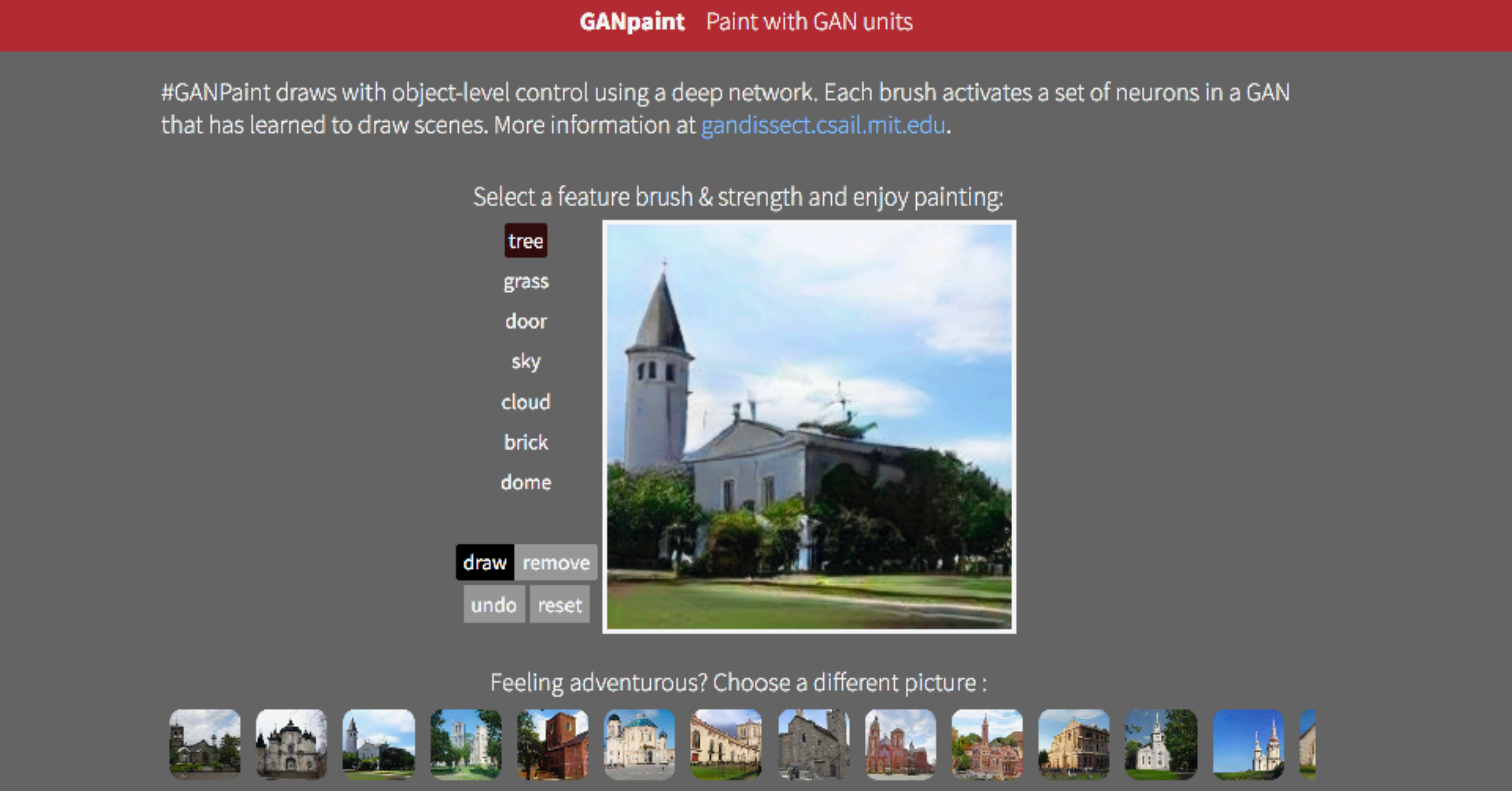
Where can you put a door?

Original image

Effect of activating layer4 door units

Bau, at al ICLR 2017 “GAN Dissection” supplemental (page 17)


Randomly generated image
Add grass
Add trees
Add brick
Change doors
Change roof



Online Demo: http://bit.ly/ganpaint

How to edit my own photo?

GAN-Synthesized Kitchen

My Kitchen Photo

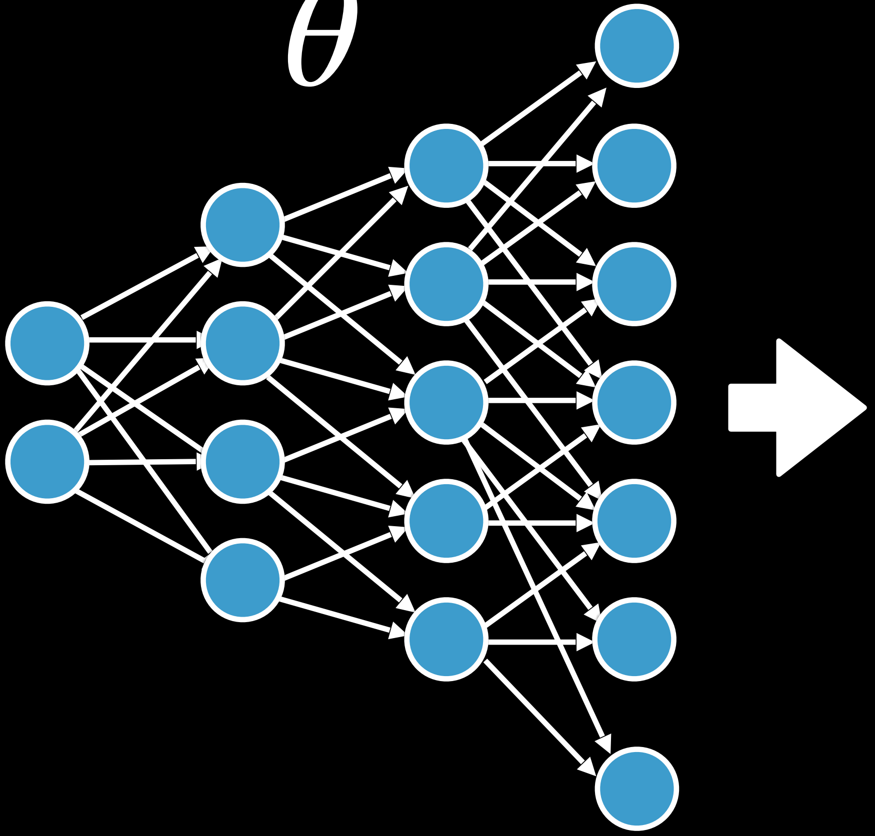
How to edit my own photo?

My Kitchen Photo

How to edit my own photo?

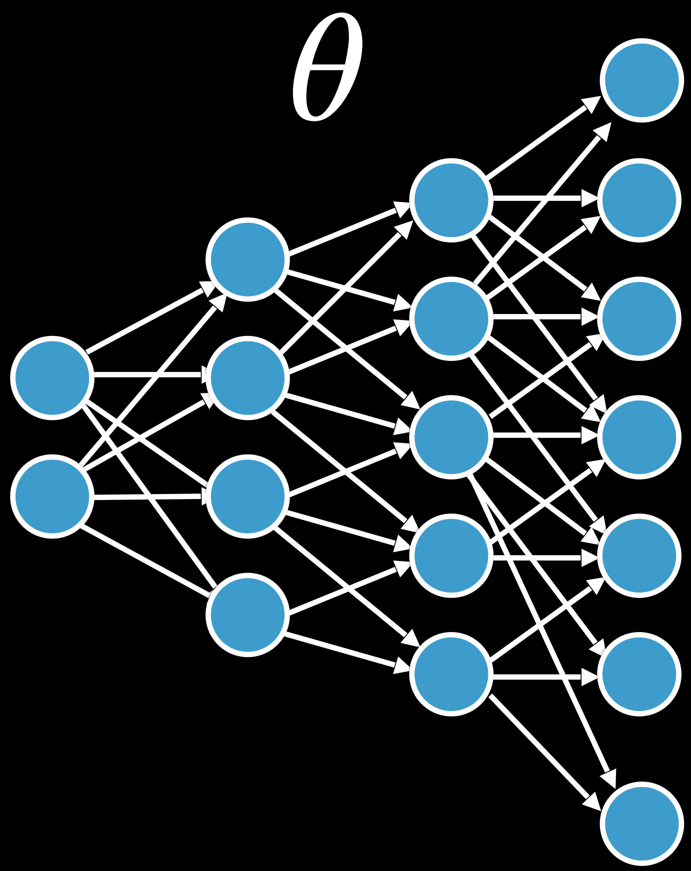
θ

Random vector z?



My Kitchen Photo

How to edit my own photo?



My Kitchen Photo

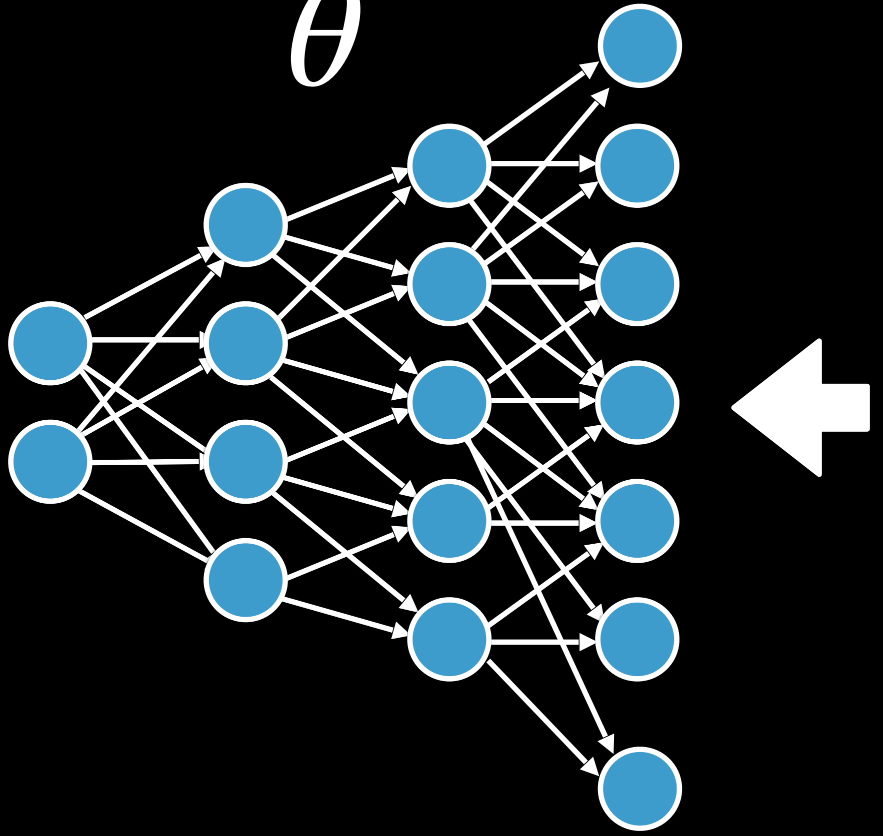
$$\hat{z} = \arg\min_{z} L_{rec}(I, G(z, \theta))$$

[Zhu et al., 2016]
[Dosovitskiy and Brox., 2016]

How to edit my own photo?

θ

Optimized vector $\hat{z}$

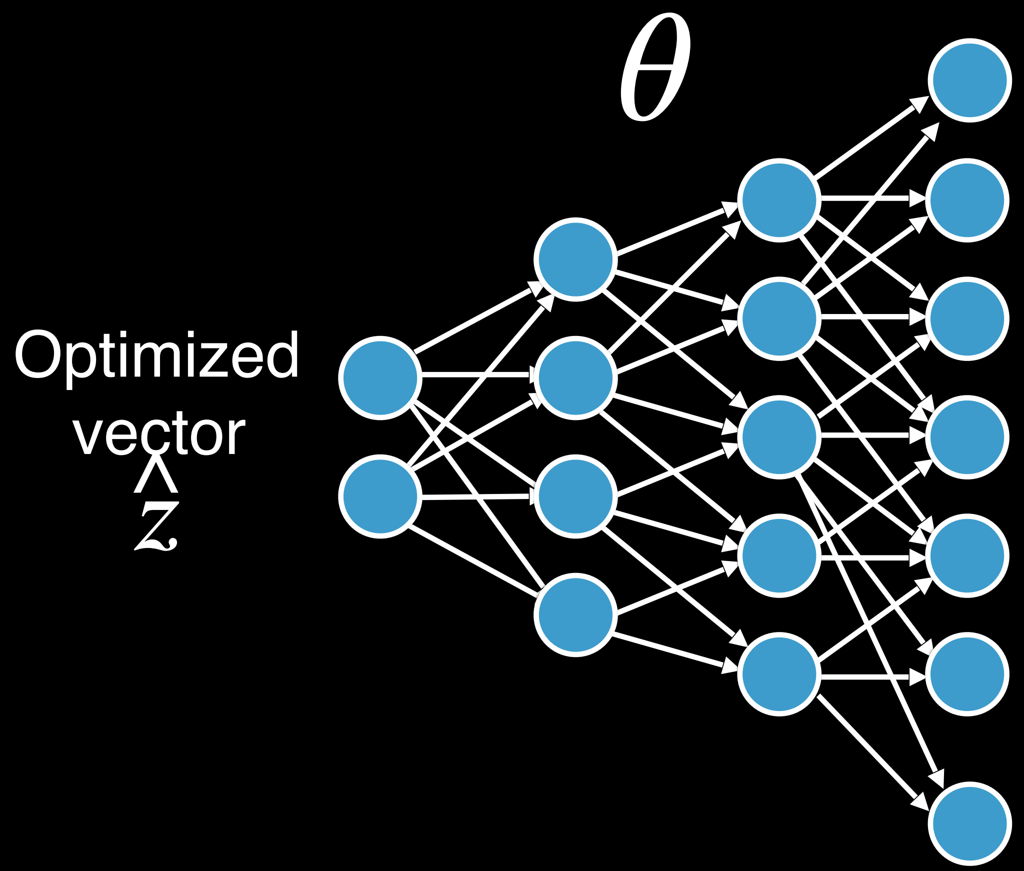


My image

$$\hat{z} = \arg\min_{z} L_{rec}(I, G(z, \theta))$$

[Zhu et al., 2016]
[Dosovitskiy and Brox., 2016]

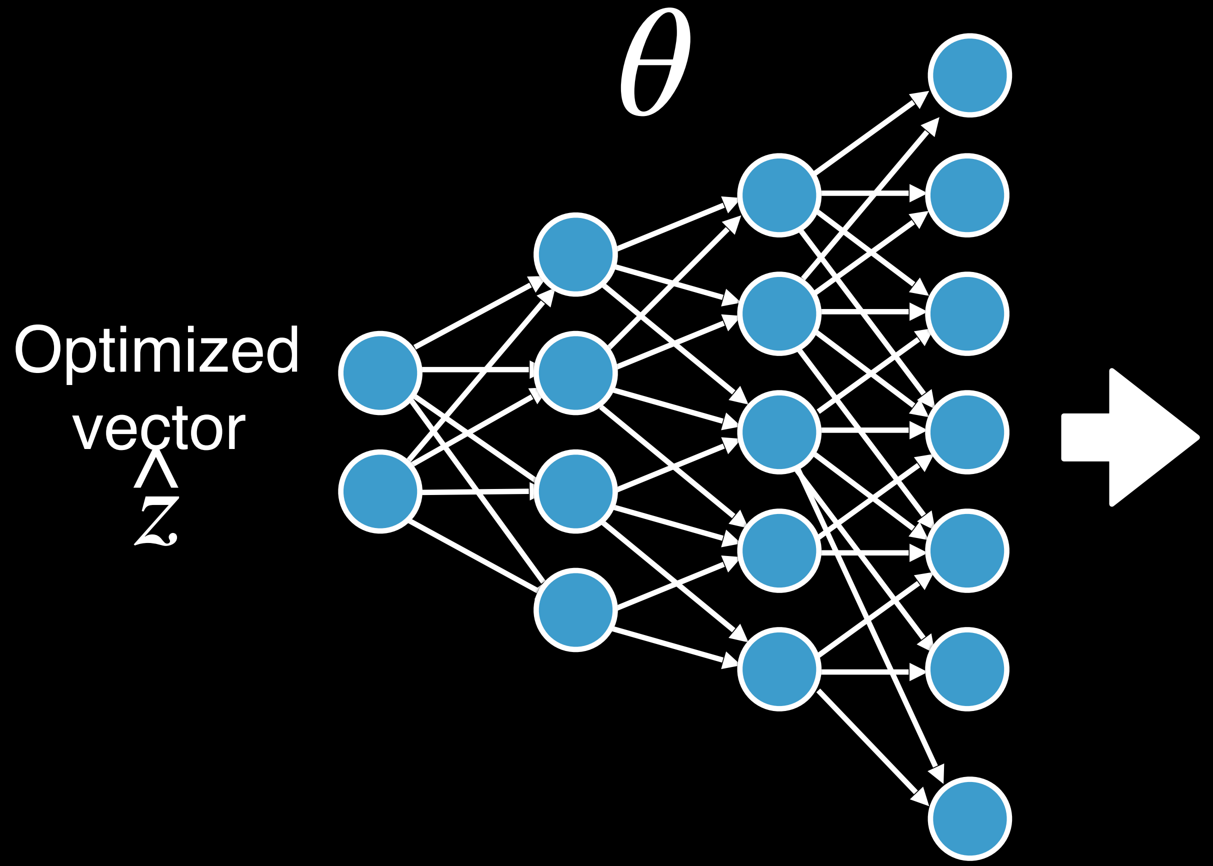
How to edit my own photo?



$$\hat{z} = \underset{z}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta))$$

[Zhu et al., 2016]
[Dosovitskiy and Brox., 2016]

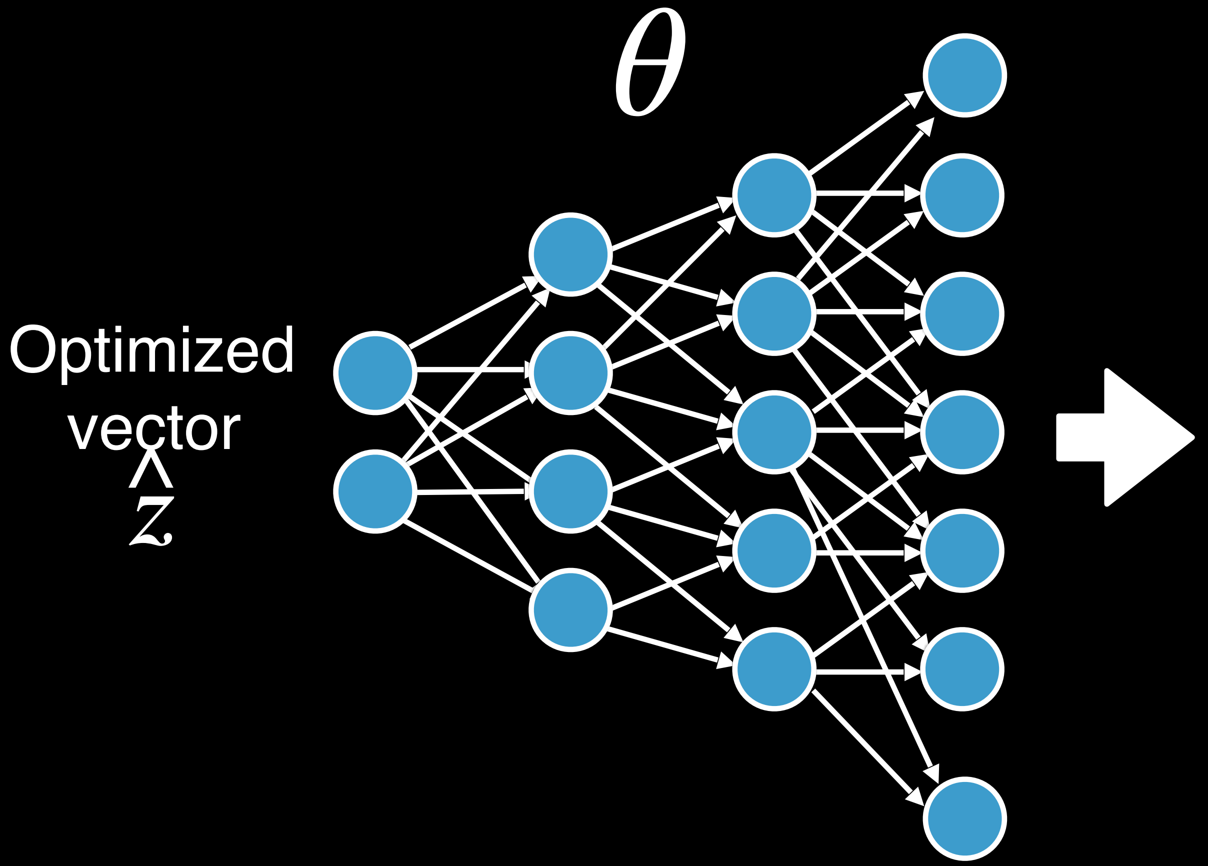
How to edit my own photo?



$$\hat{z} = \underset{z}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta))$$

[Zhu et al., 2016]
[Dosovitskiy and Brox., 2016]

How to edit my own photo?



Reconstructed image

$$\hat{z} = \underset{z}{\operatorname{argmin}} L_{rec}(I, G(z, \theta))$$

[Zhu et al., 2016]
[Dosovitskiy and Brox., 2016]

Find the differences…

Original image

Find the differences…

Original image

GAN reconstructed image

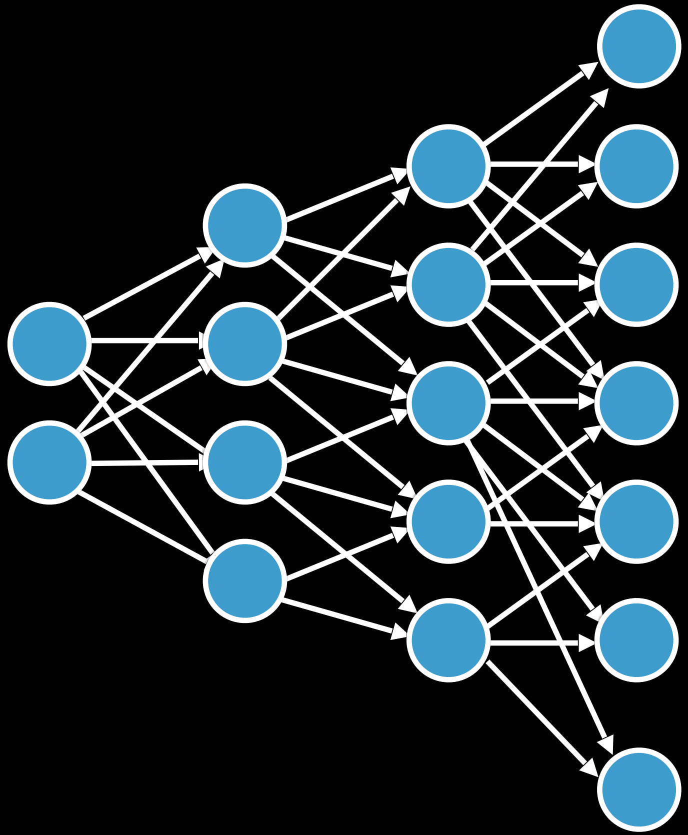
Find the differences…

Original image

GAN reconstructed image

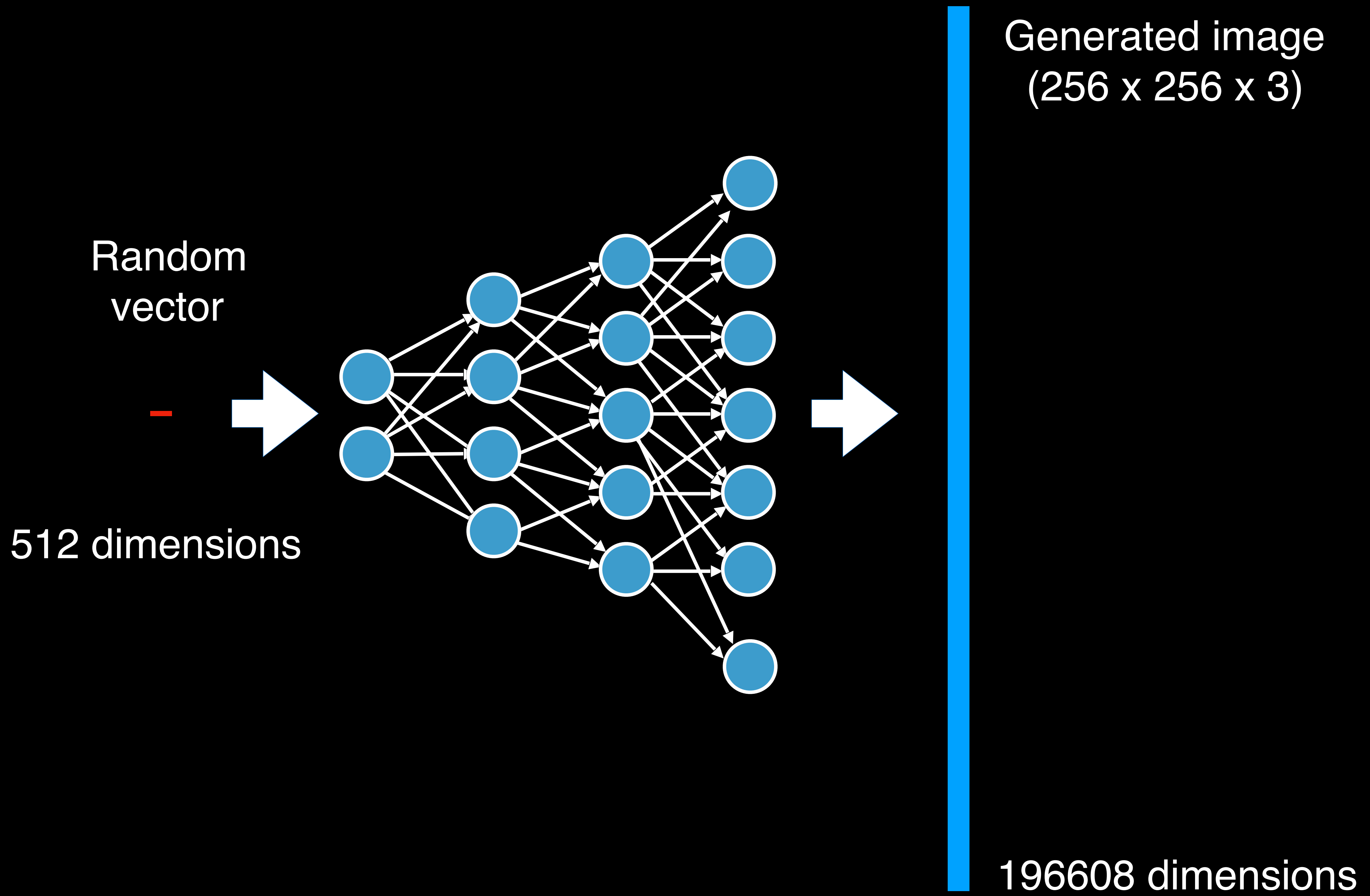
Random vector

512 dimensions



Generated image (256 x 256 x 3)

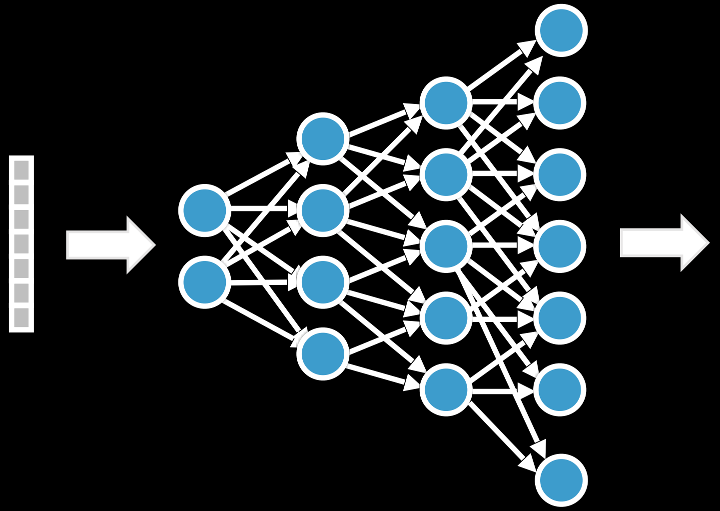
196608 dimensions


Random
vector
512 dimensions
Generated image
(256 x 256 x 3)
196608 dimensions

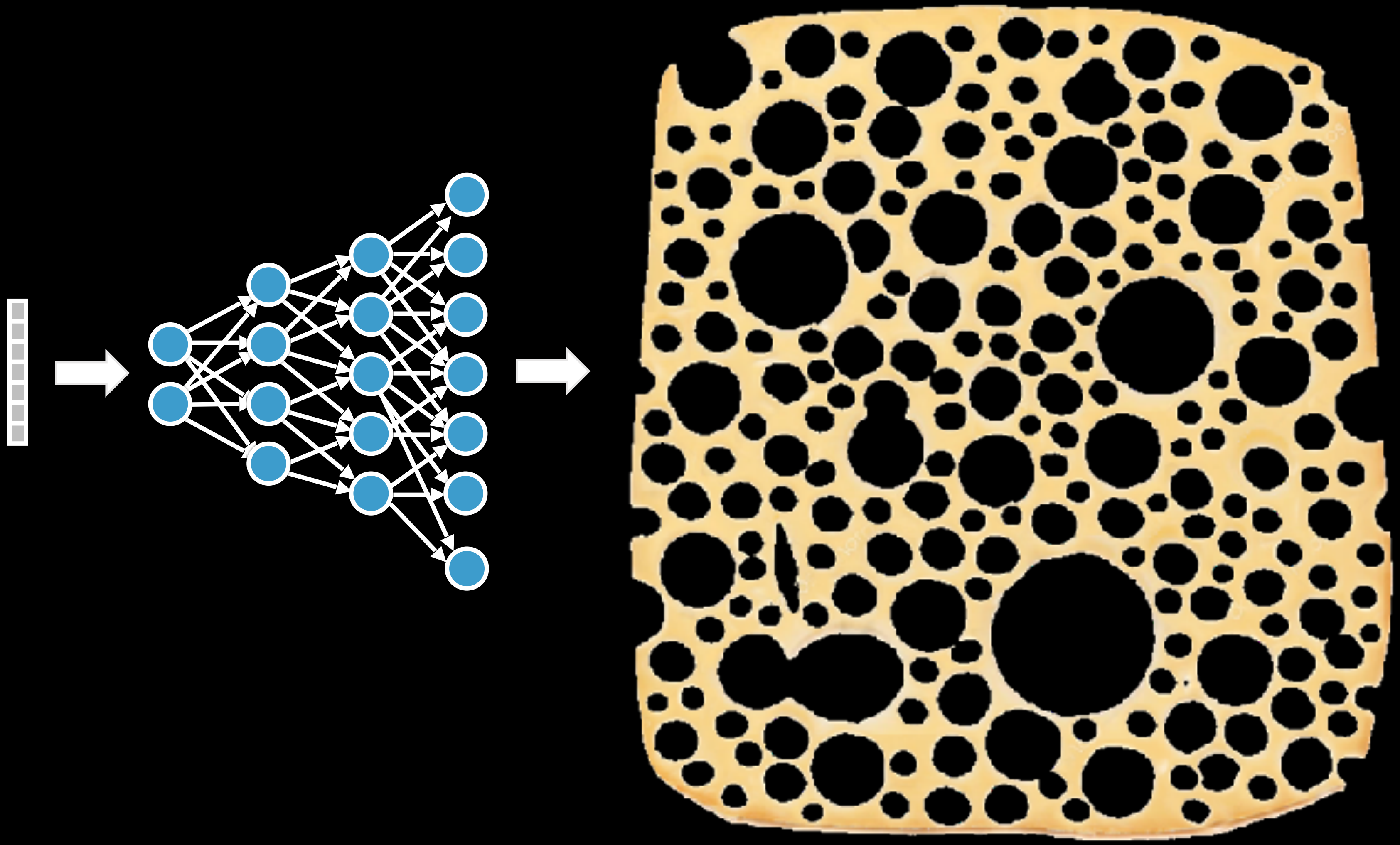
The manifold of natural images

The manifold of natural images

The manifold of natural images

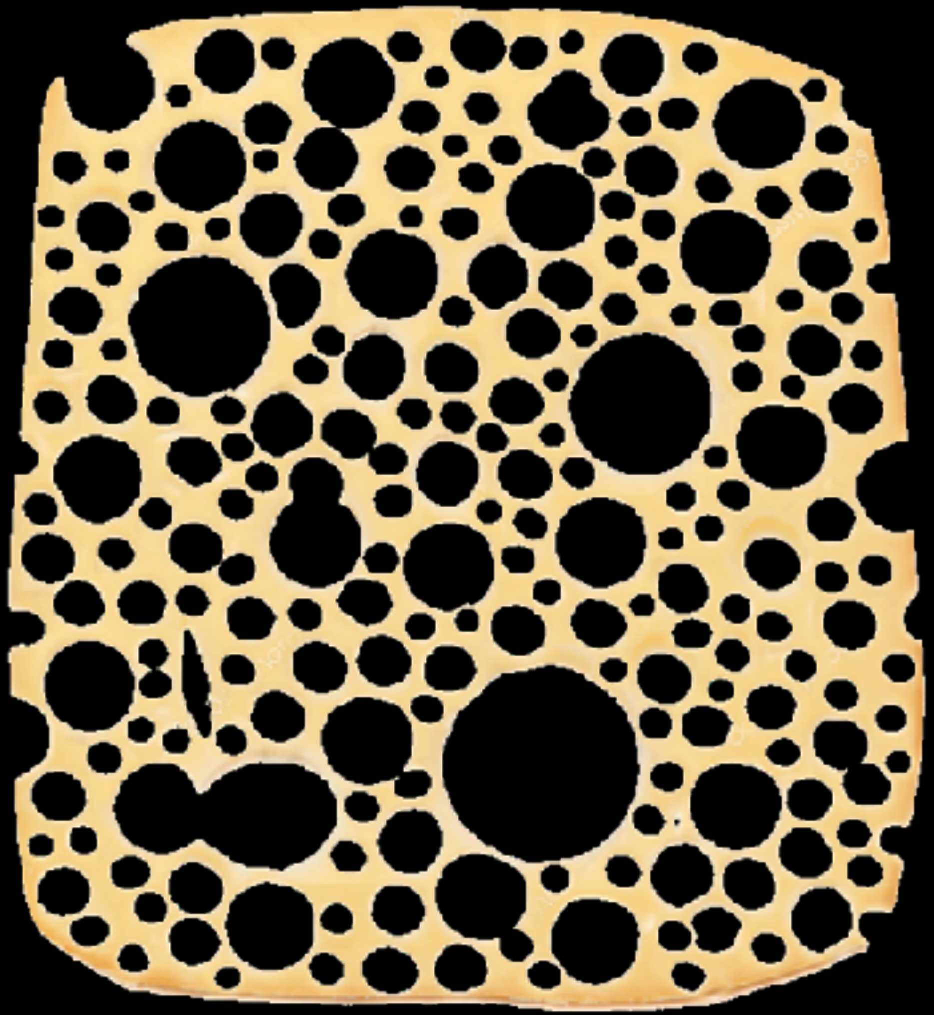


The manifold of natural images



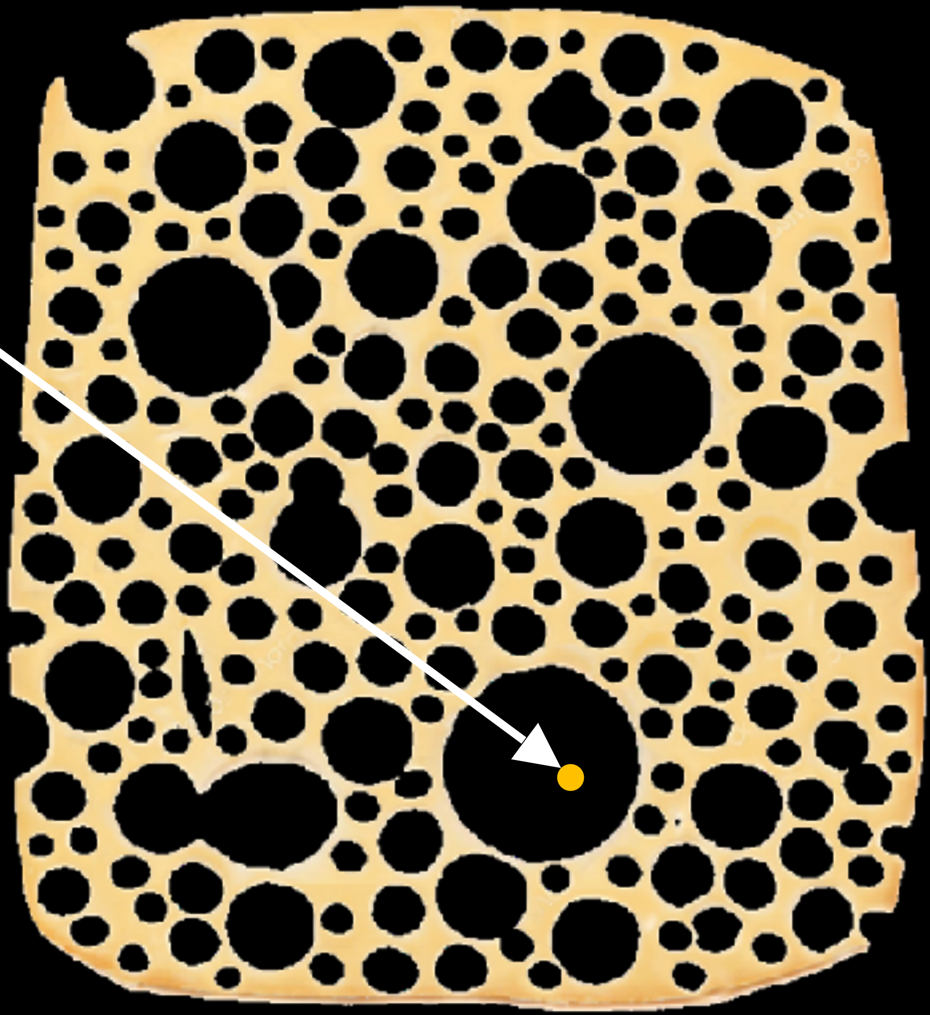
The manifold of natural images

Original image

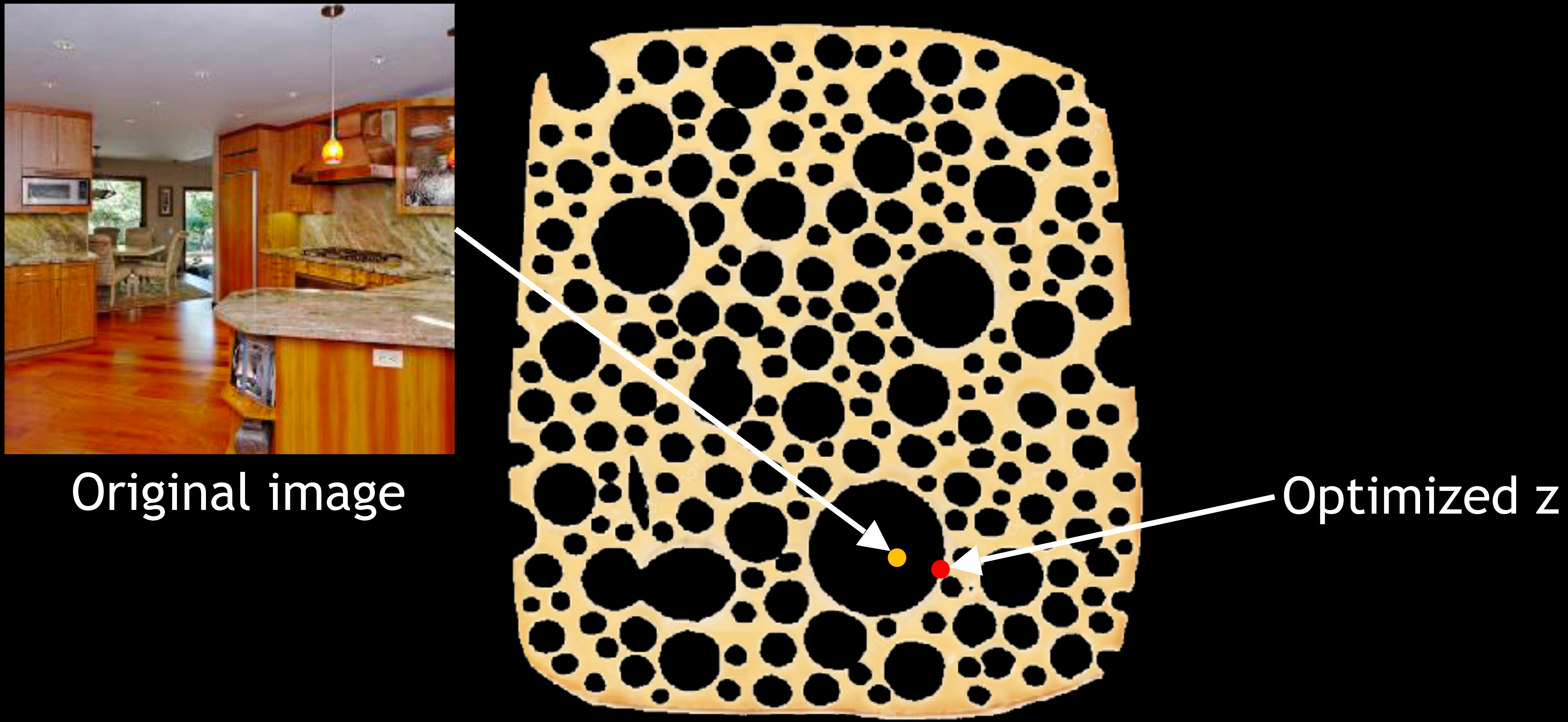


The manifold of natural images

Original image

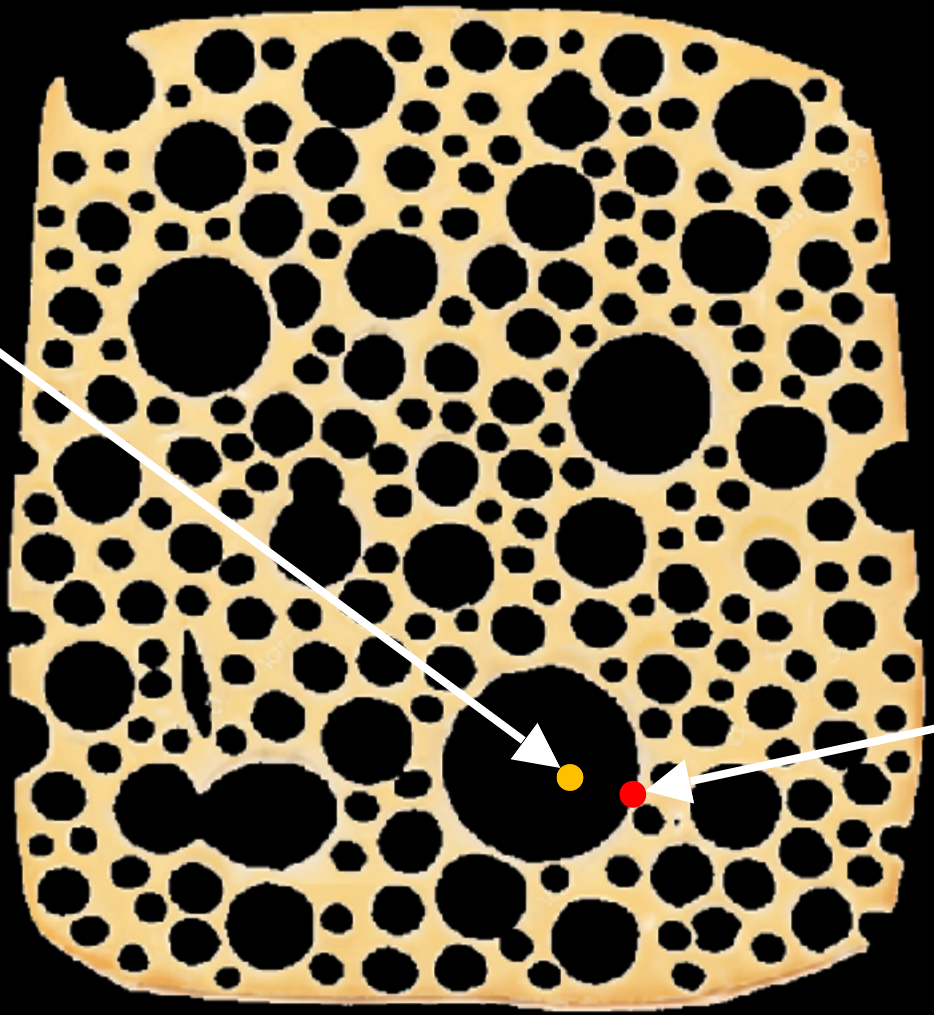


The manifold of natural images



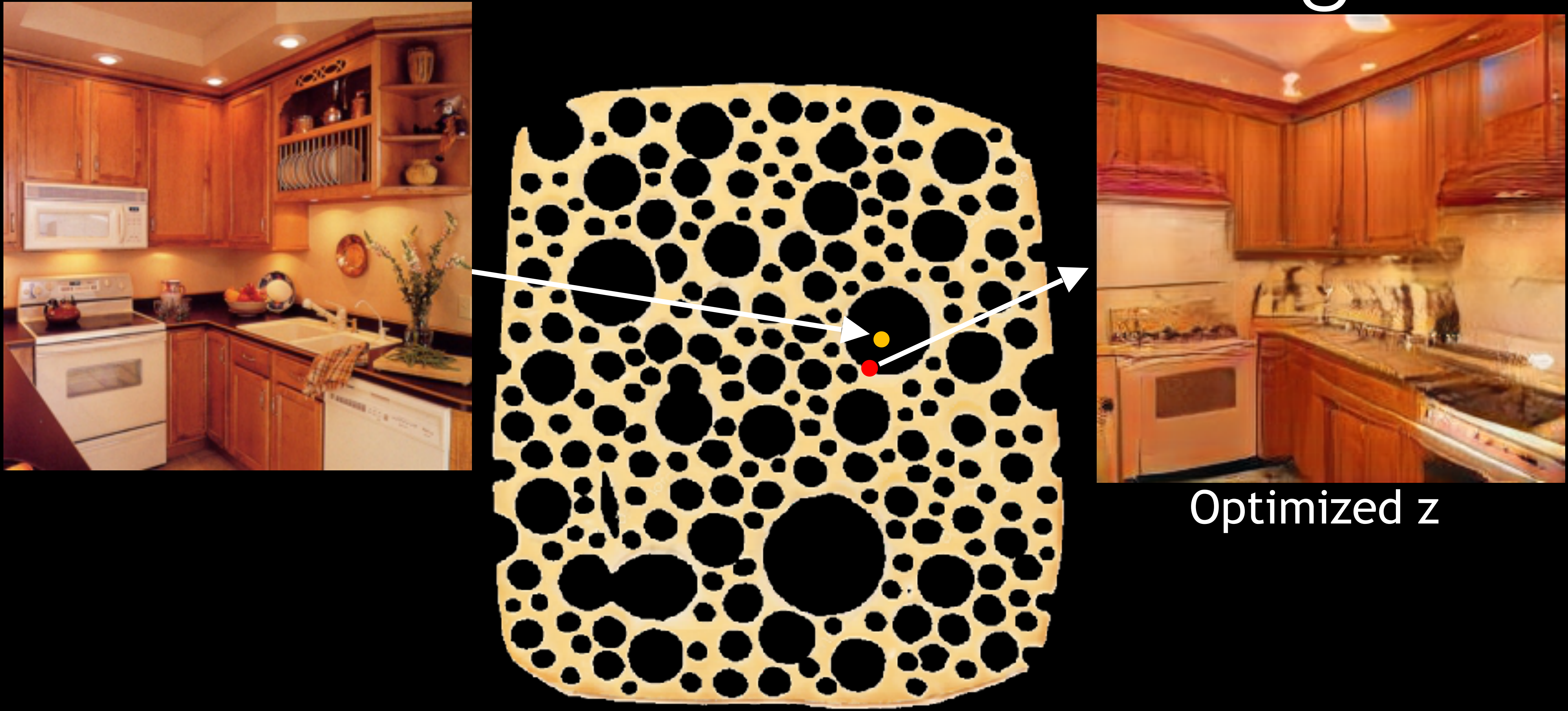
The manifold of natural images

Original image



Optimized z

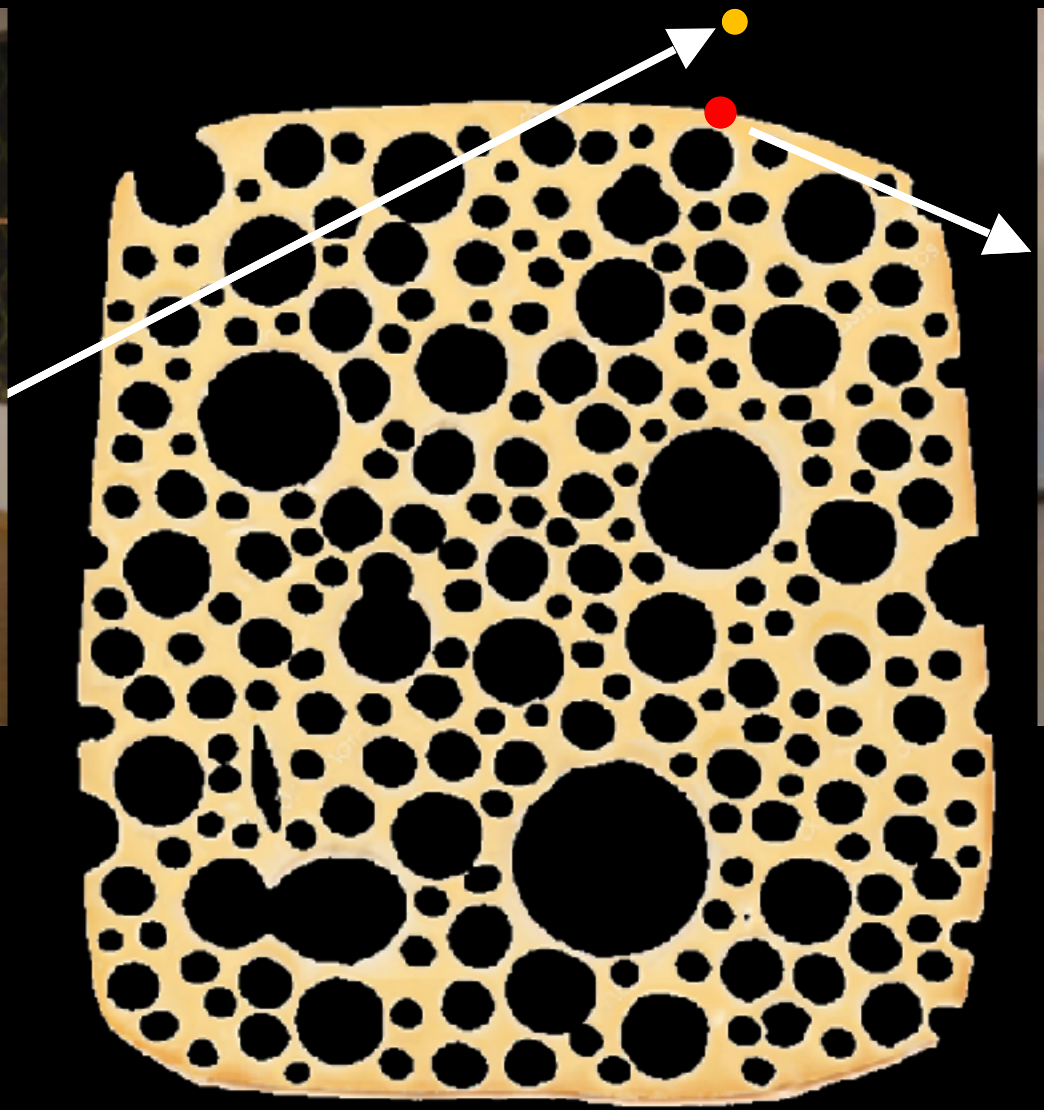
The manifold of natural images



Optimized z

The manifold of natural images

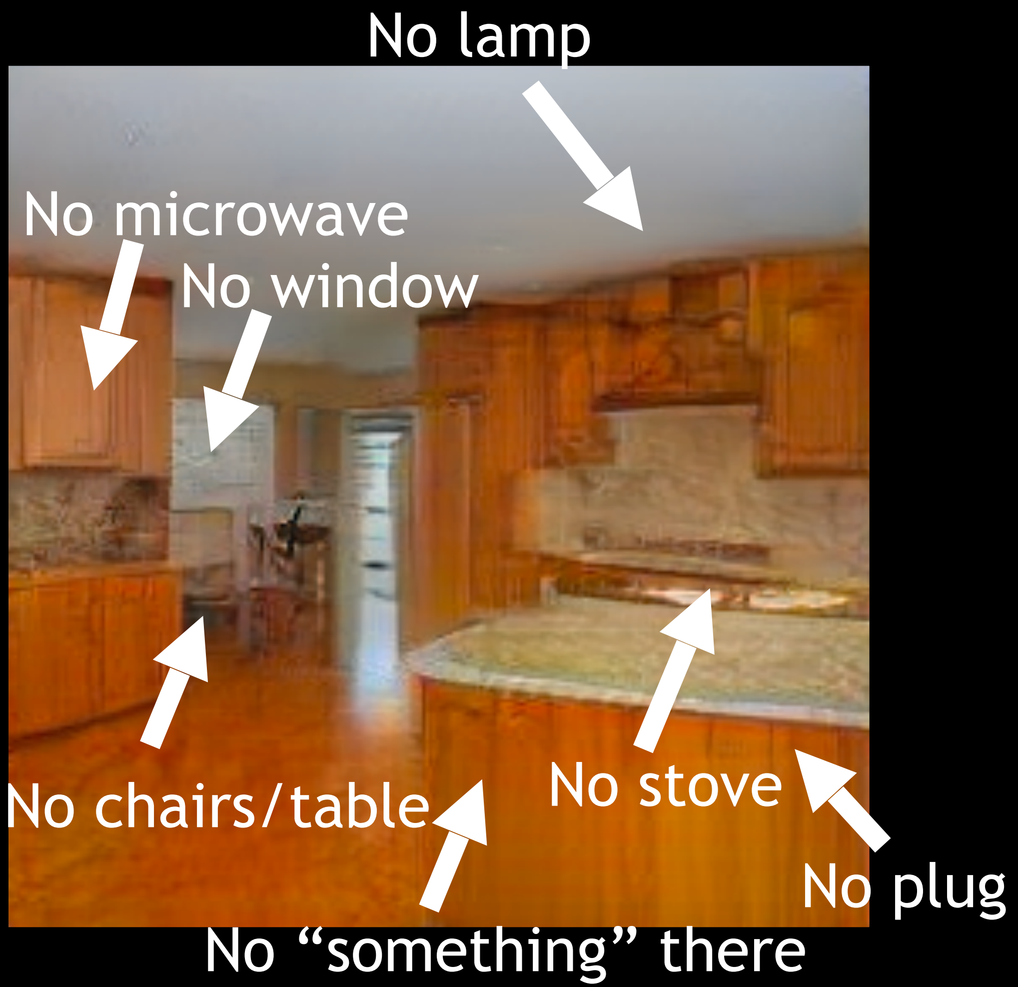
Original image



Optimized z

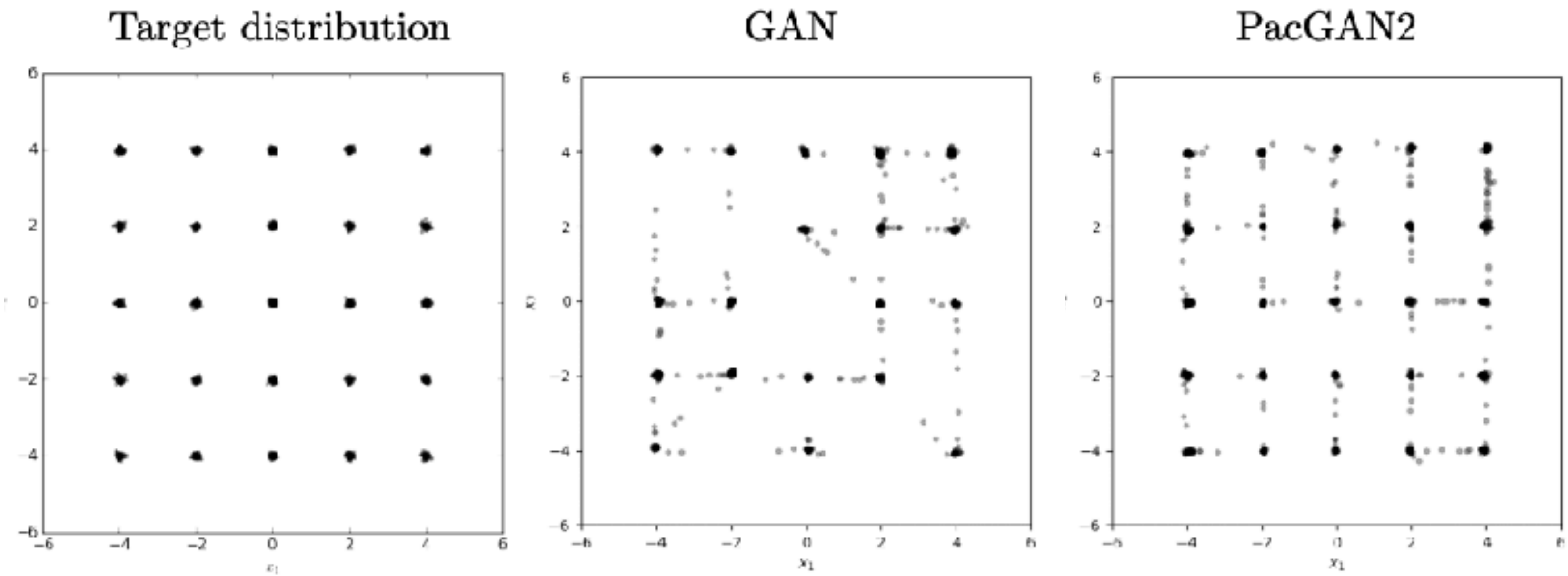
What can a GAN see?

Original image



GAN reconstructed image

GANs drop modes, even on small problems



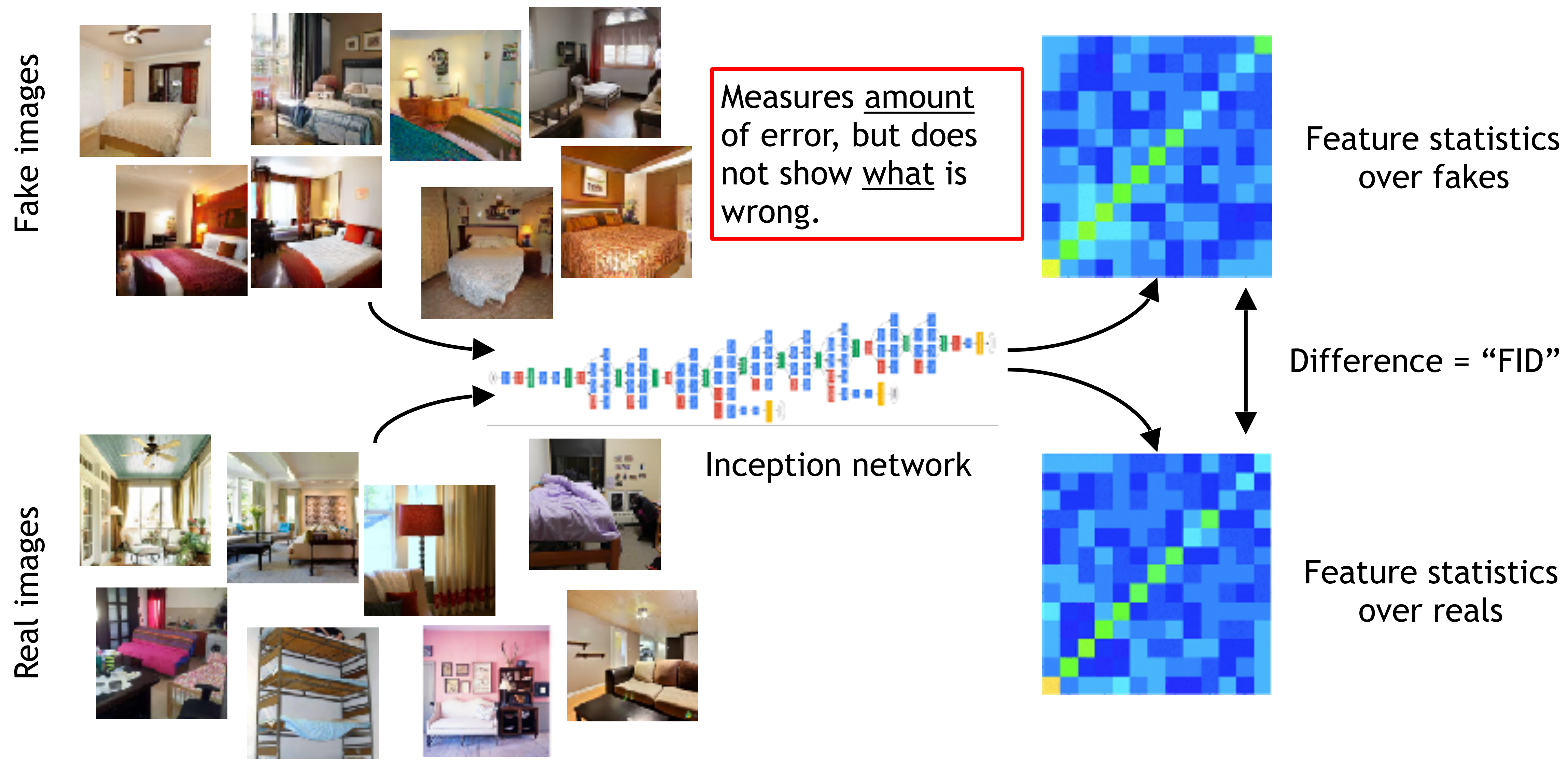
On a simple 2d problem, the whole distribution can be seen directly as a scatterplot.

On these simple problems, GANs can be seen to drop many modes.

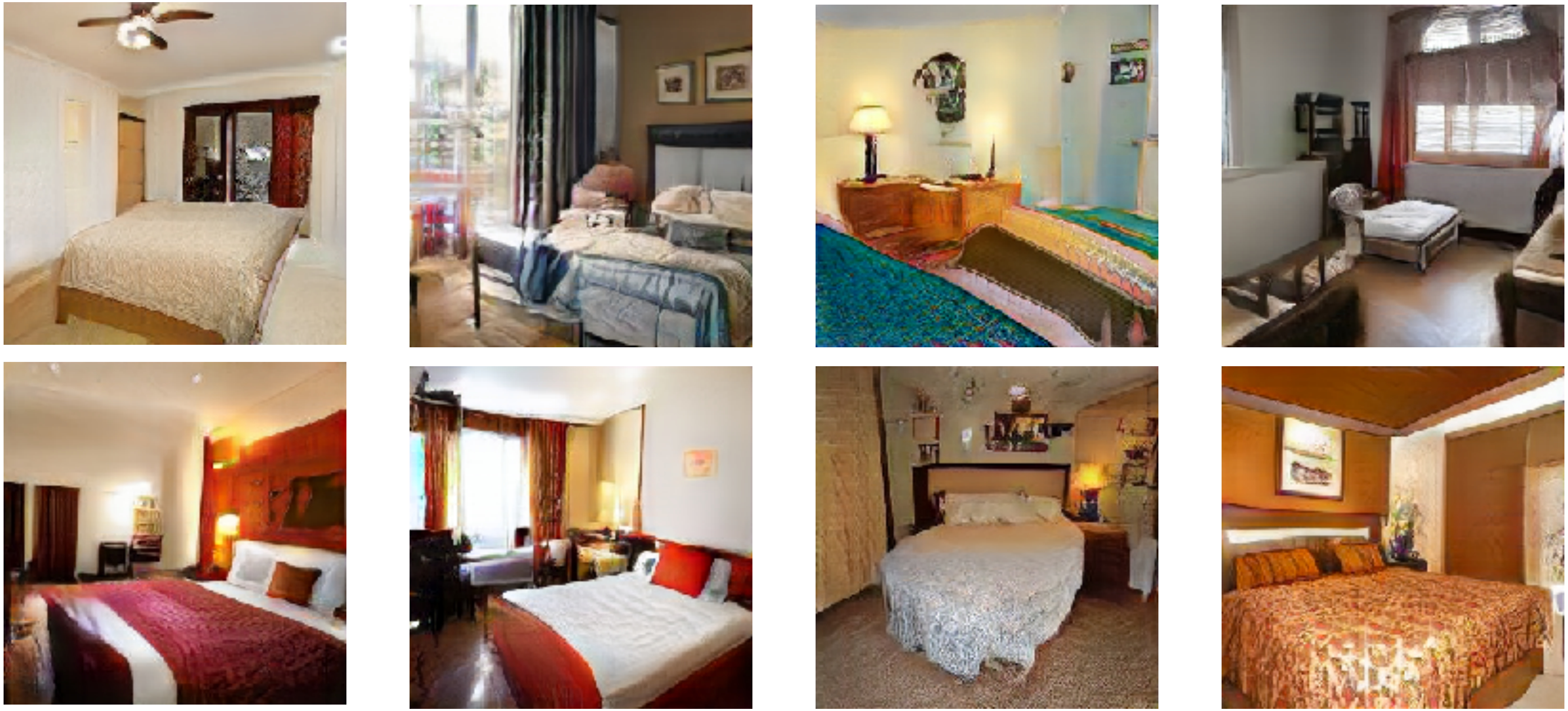
Reducing mode-dropping is an open and active area of research.

Lin, et al “PacGAN” 2018

The current way to benchmark GAN output



The current way to see a GAN's capabilities

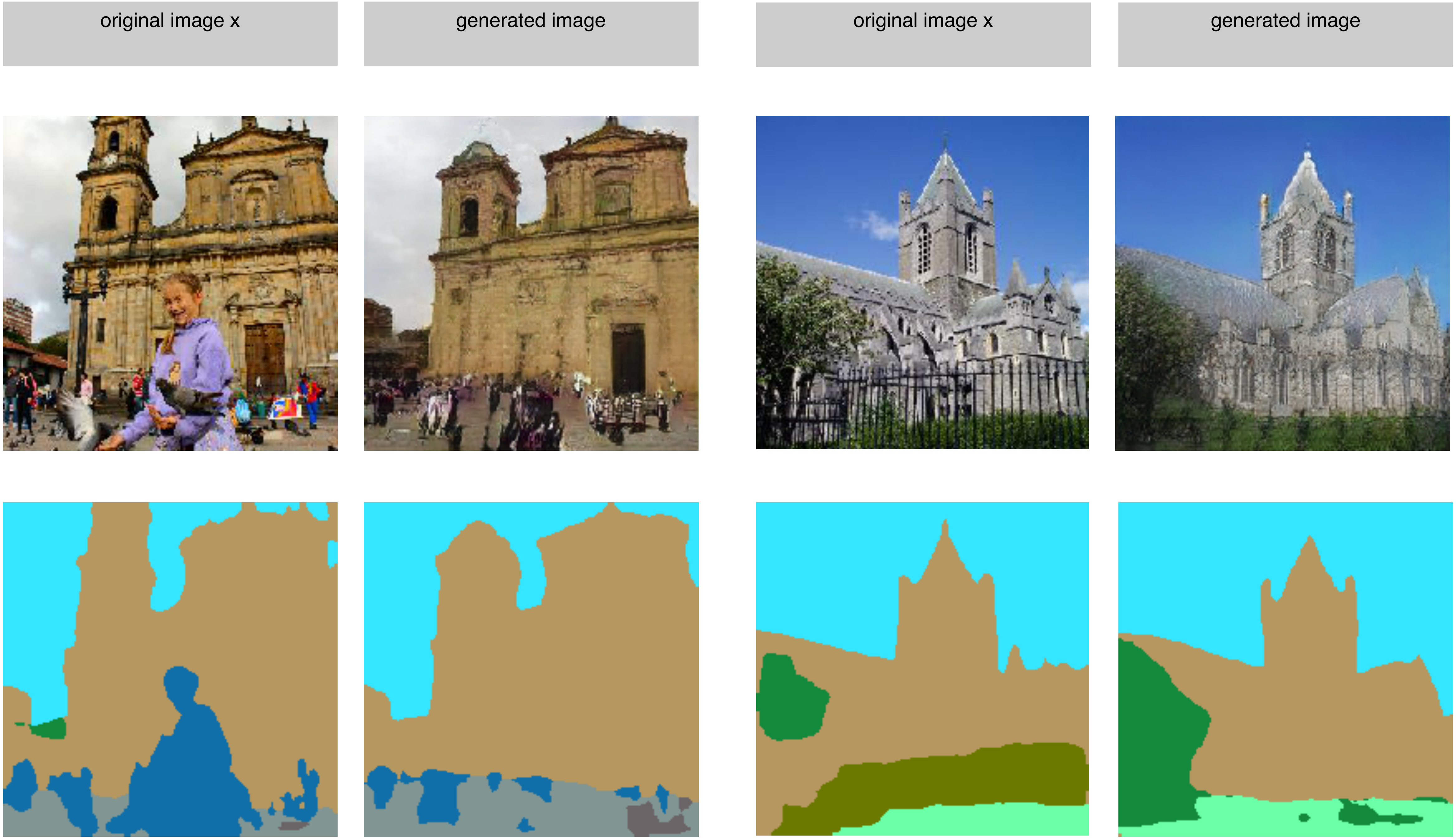


Just generate a sample, and see what it can do

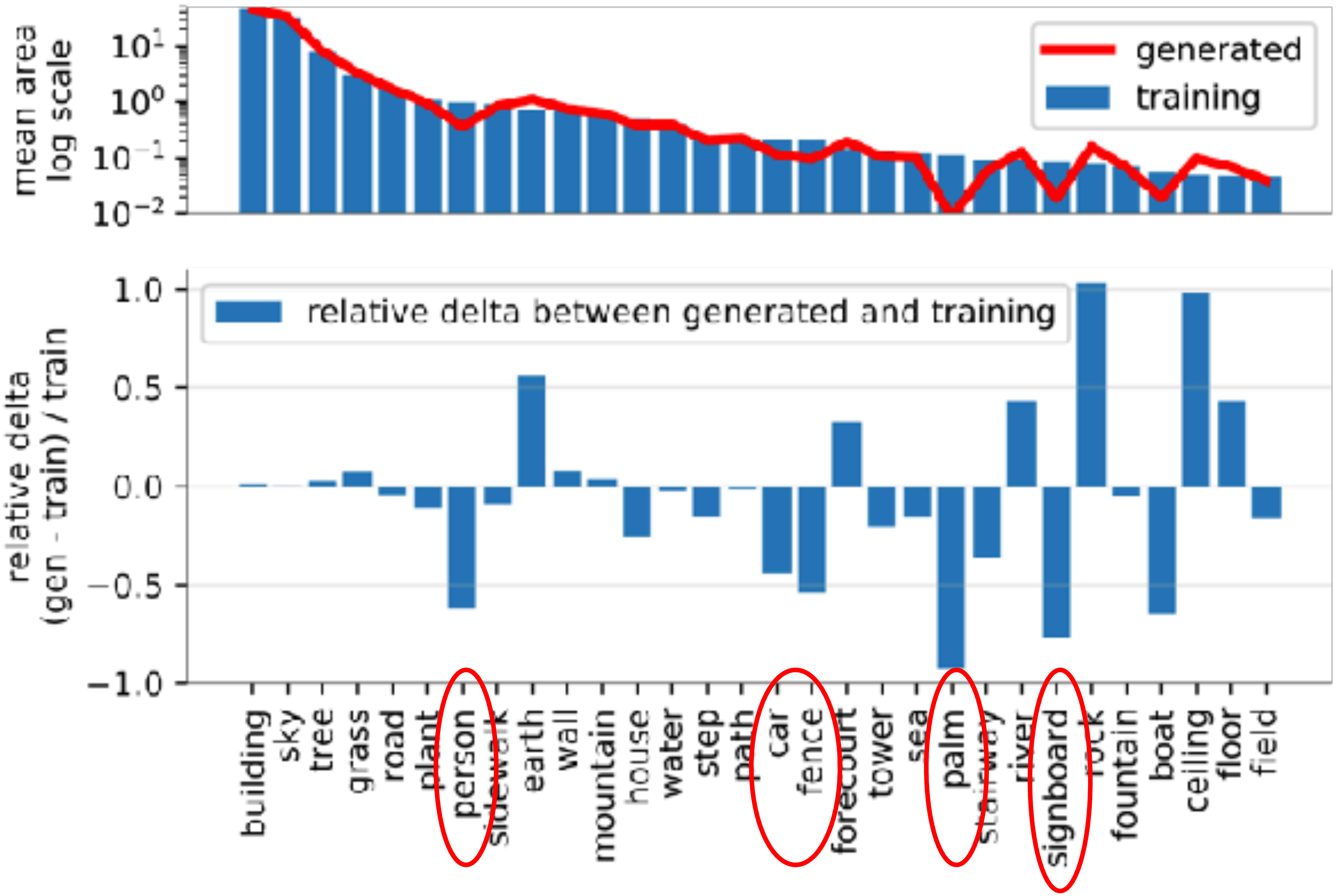
Karras, et al "Progressive GAN" Bedrooms, 2017 original image x

original image x


original image x
generated image
original image x
generated image


original image x
generated image
original image x
generated image

Some models omit very specific categories



The GAN tends to omit people

Skipping people

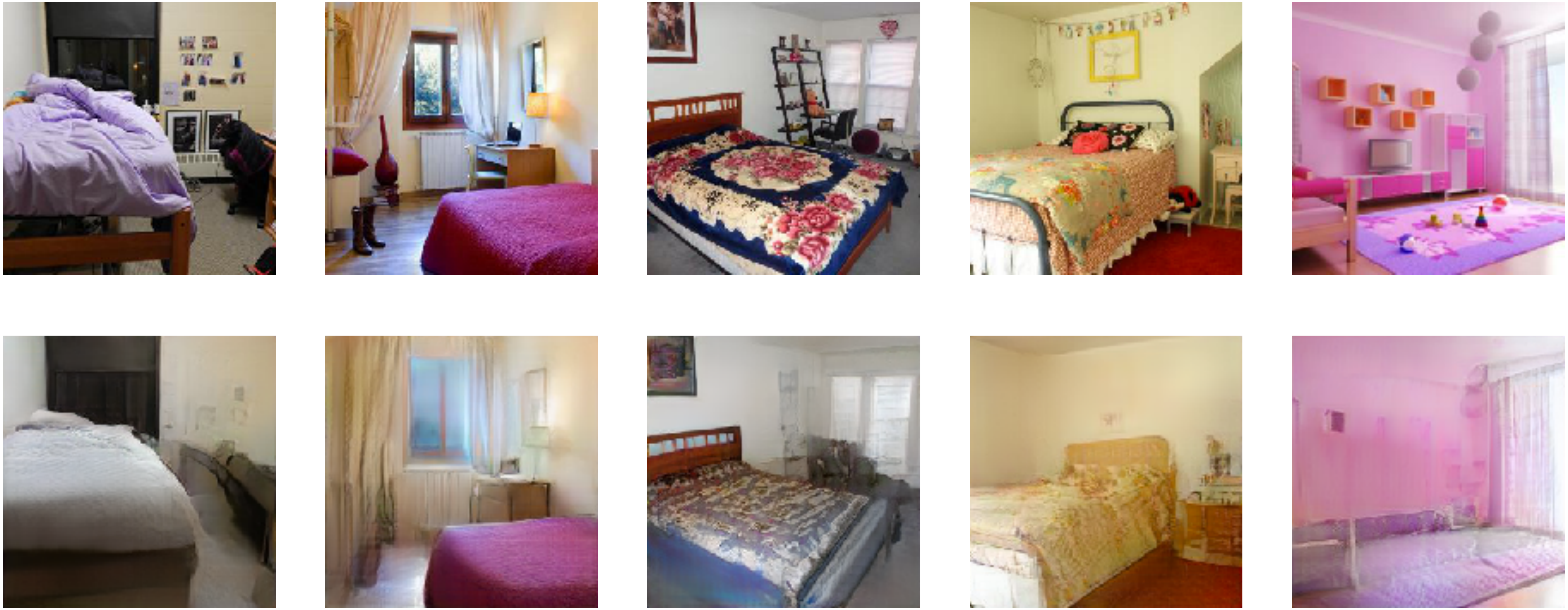
Not everyone…

Skipping vehicles

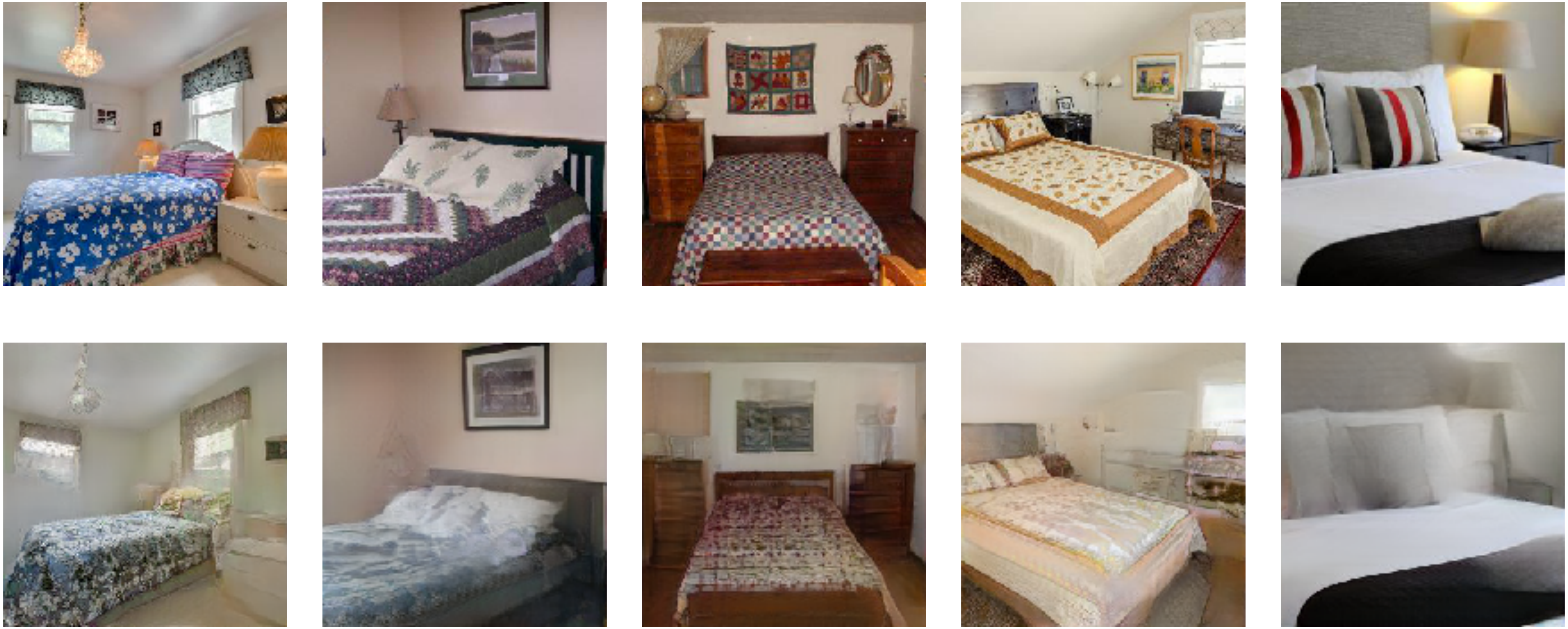
Skipping signage and text

Skipping monuments

Bedroom skips lots of idiosyncratic detail



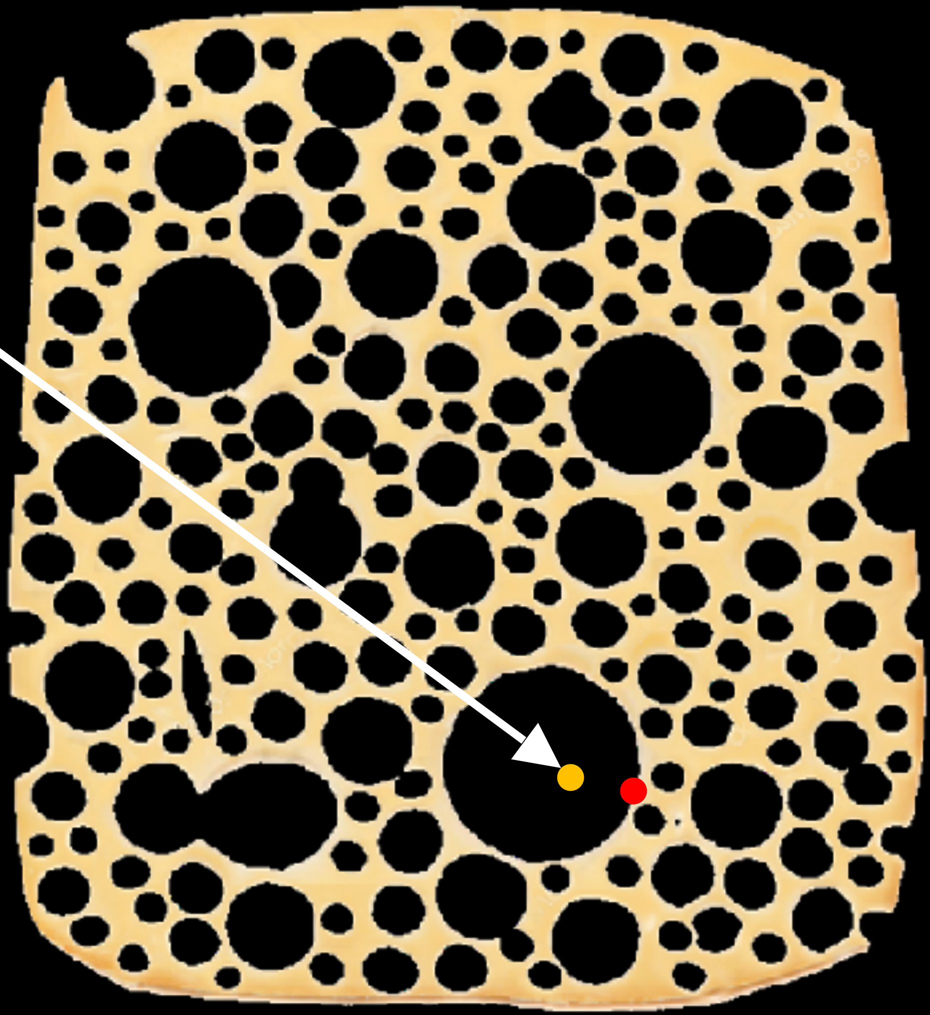
Different textures are available



)

What do we do?

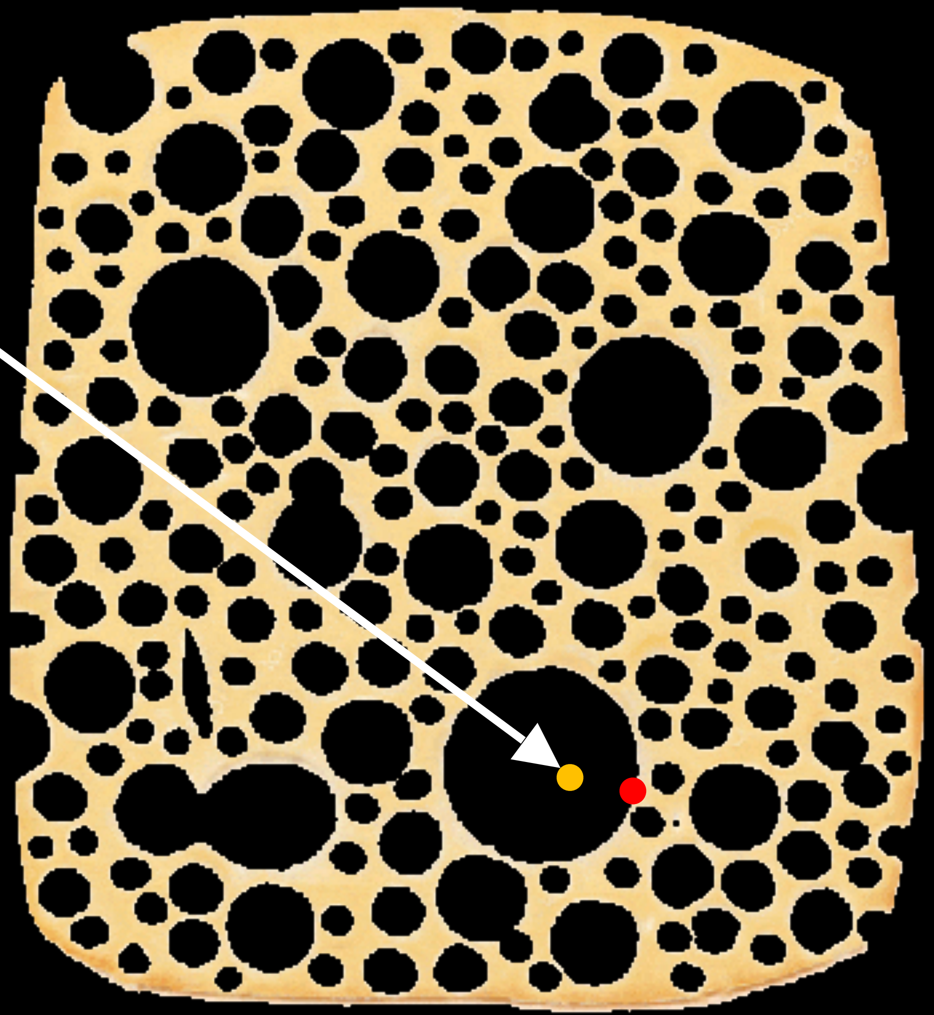
Original image



What do we do?

Adapted cheese

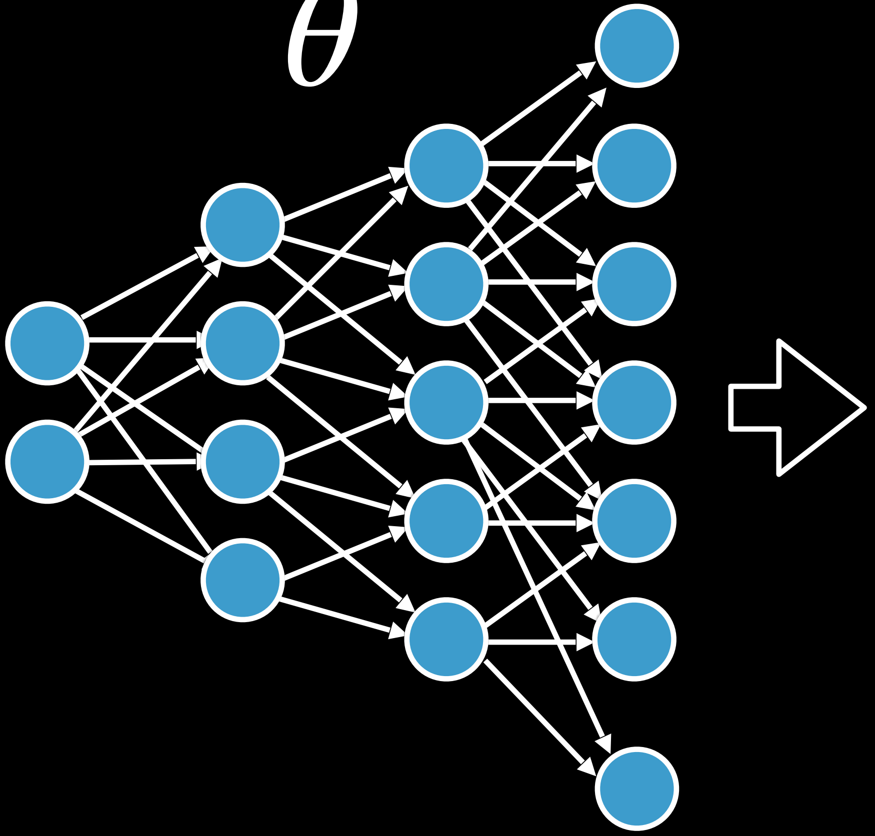
Original image



Reconstructing my own photo

Optimized vector $\hat{z}$

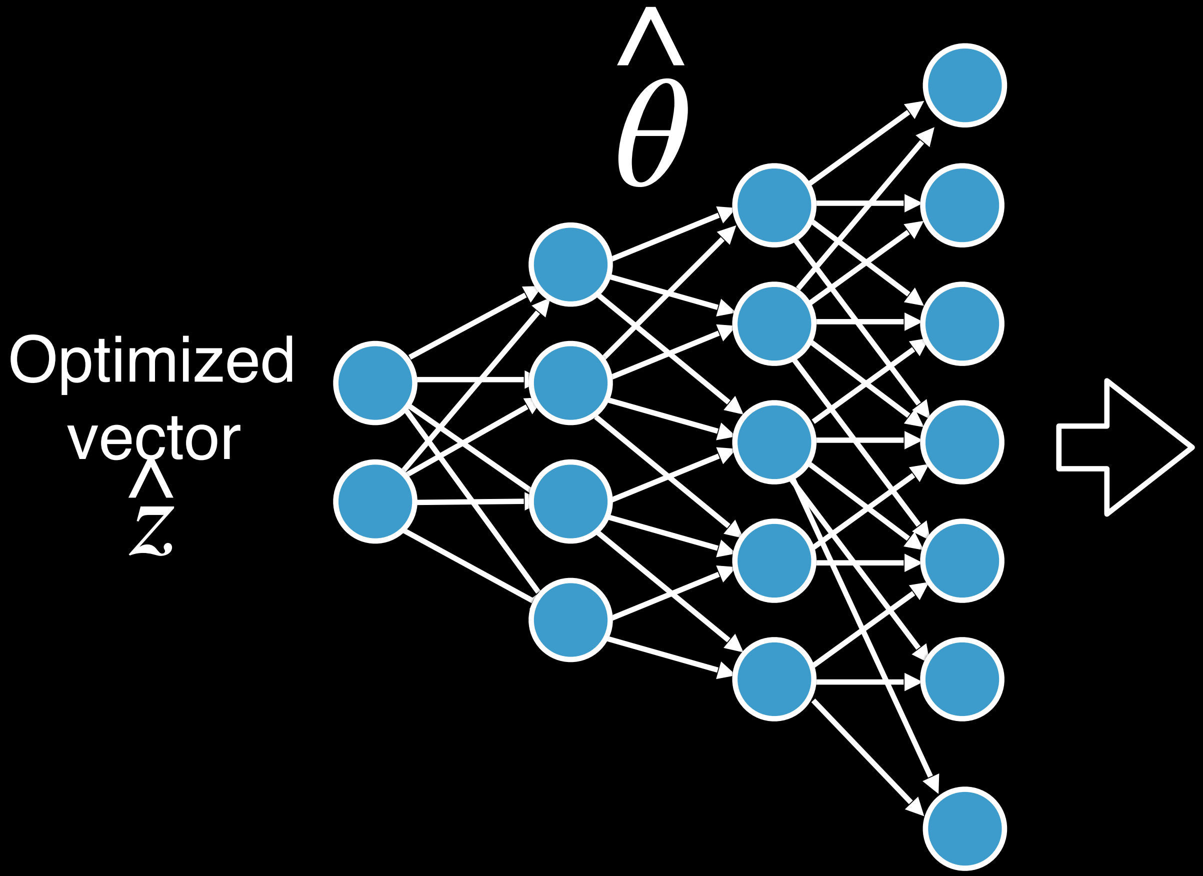
θ



Reconstructed image

$$\hat{z} = \underset{z}{\mathrm{argmin}} L_{rec}(I, G(z,\ \theta))$$

Reconstructing my own photo

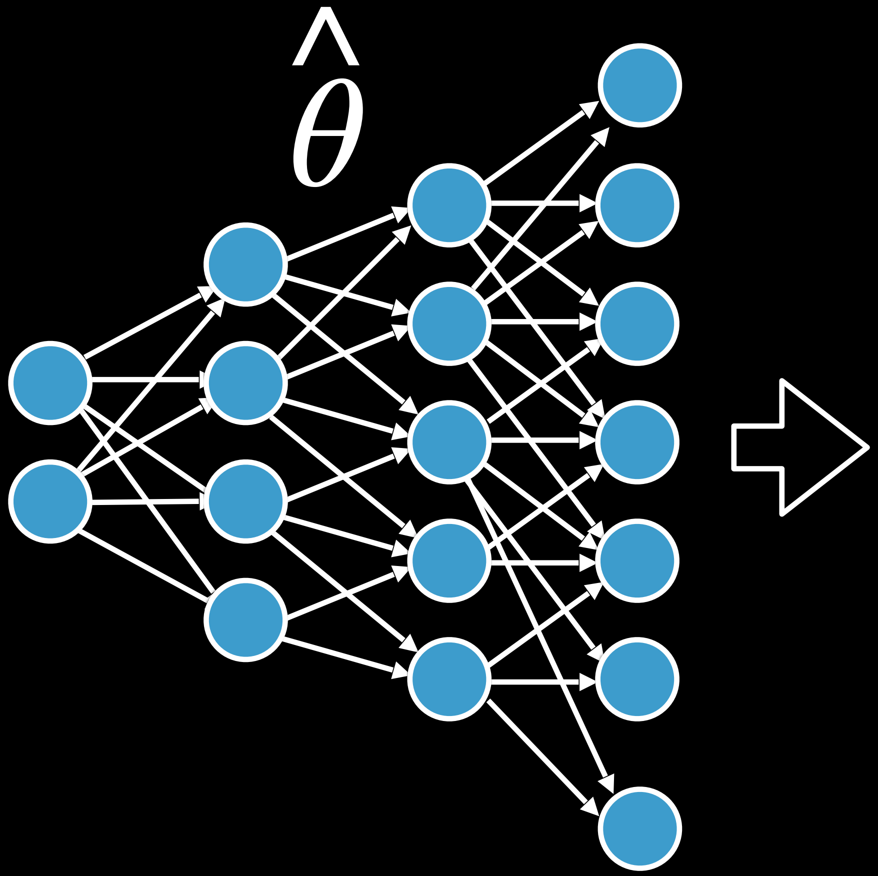


Reconstructed image

$$\hat{z}, \hat{\theta} = \underset{z,\theta}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta))$$

Reconstructing my own photo

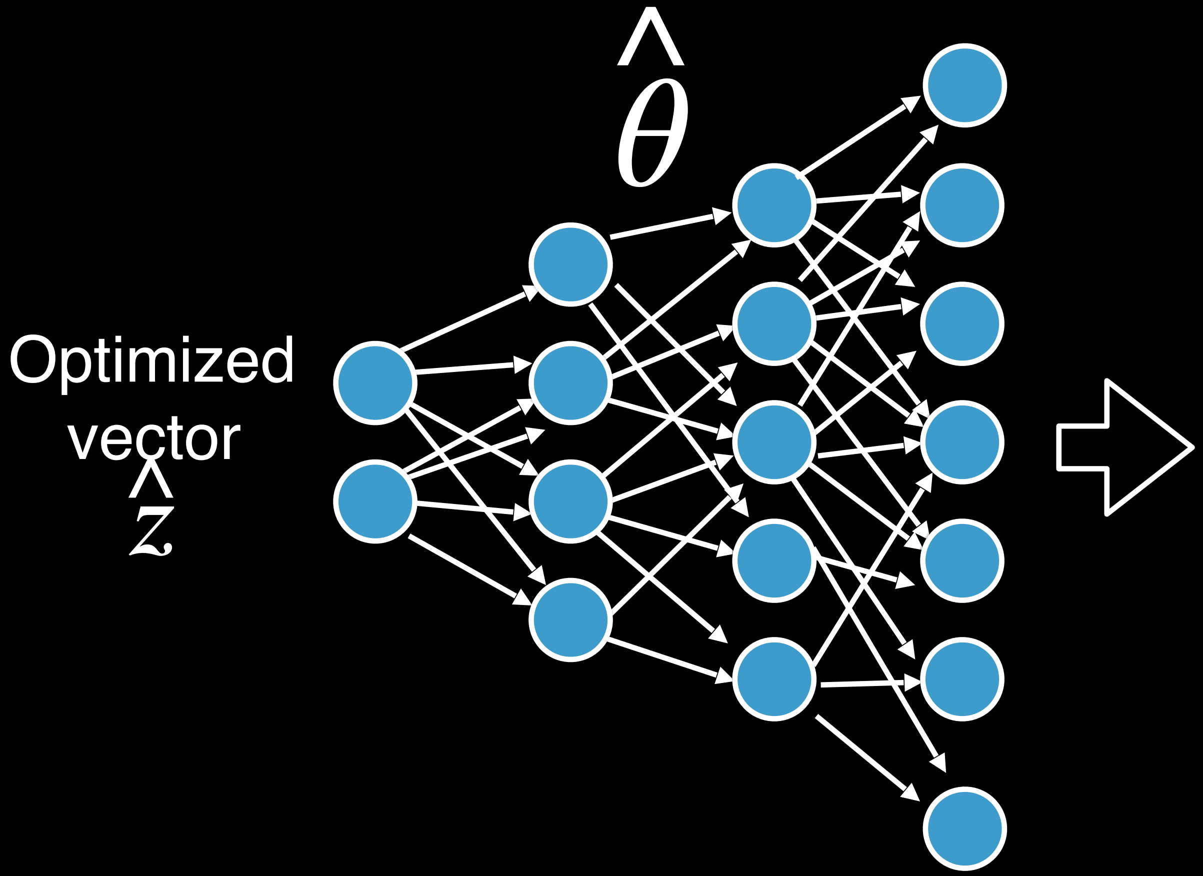
Optimized vector $\hat{z}$



Reconstructed image

$$\hat{z}, \hat{\theta} = \underset{z,\theta}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta) + R(\theta)$$

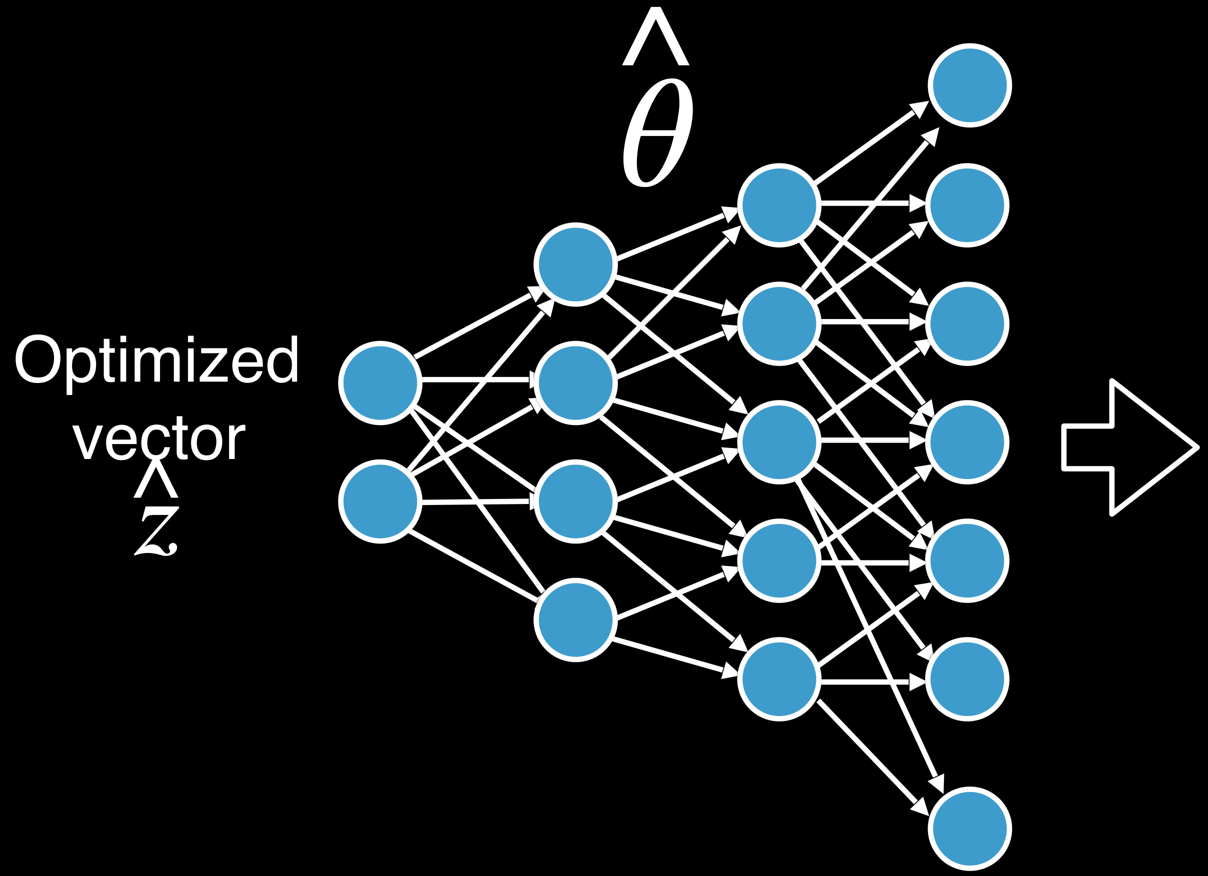
Reconstructing my own photo



Reconstructed image

$$\hat{z}, \hat{\theta} = \underset{z,\theta}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta) + R(\theta)$$

Reconstructing my own photo

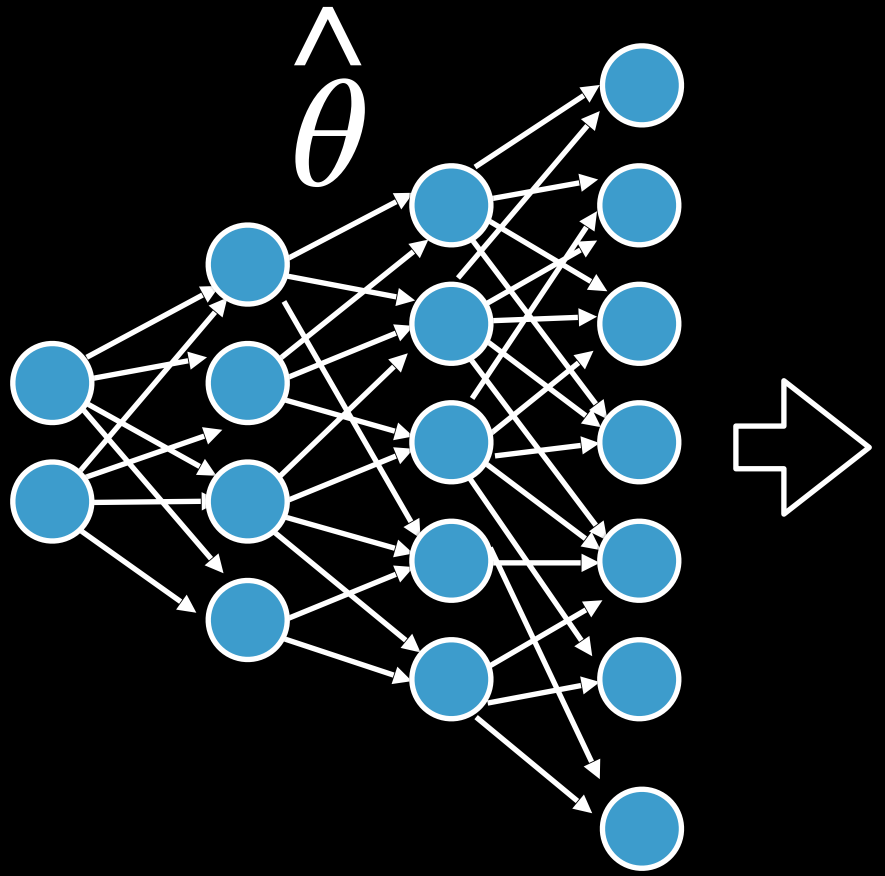


Reconstructed image

$$\hat{z}, \hat{\theta} = \underset{z,\theta}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta) + R(\theta)$$

Reconstructing my own photo

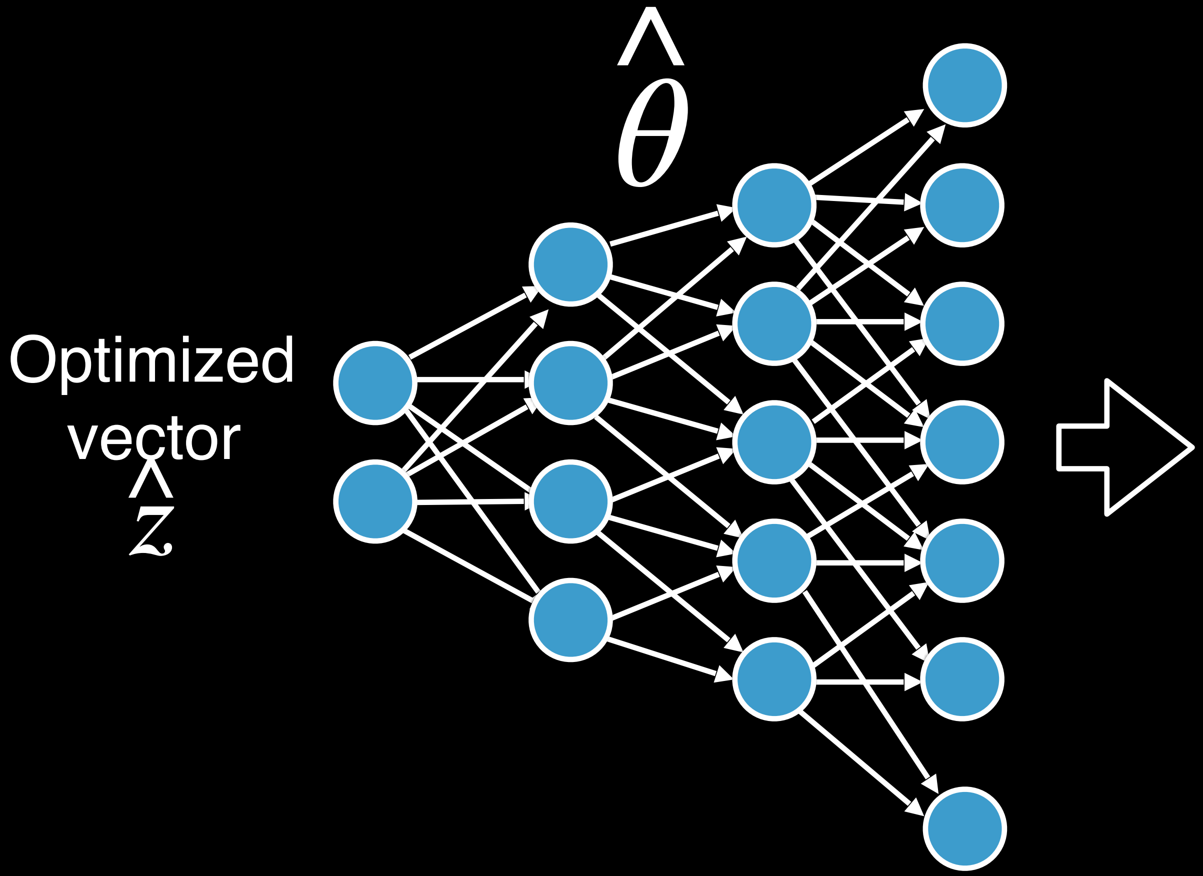
Optimized vector $\hat{z}$



Reconstructed image

$$\hat{z}, \hat{\theta} = \underset{z,\theta}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta) + R(\theta))$$

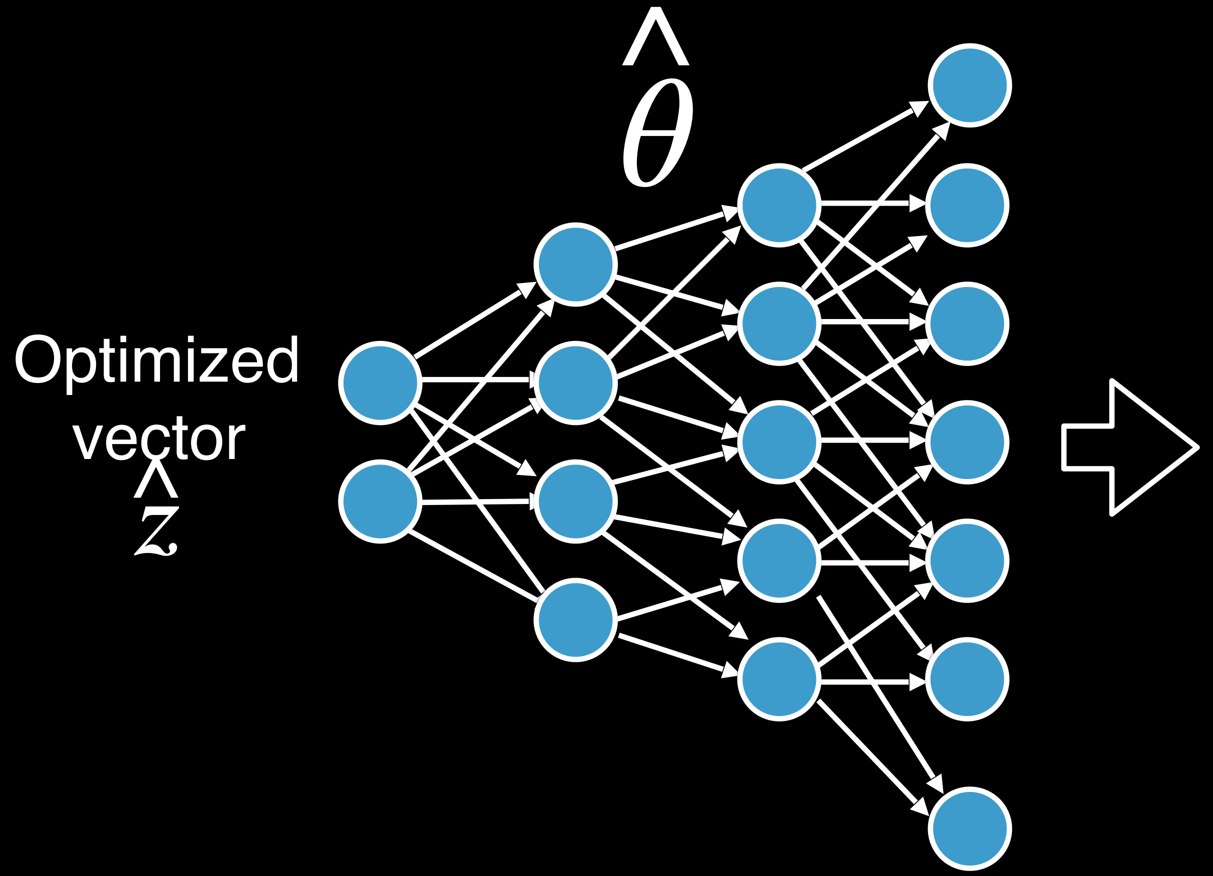
Reconstructing my own photo



Reconstructed image

$$\hat{z}, \hat{\theta} = \underset{z,\theta}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta) + R(\theta)$$

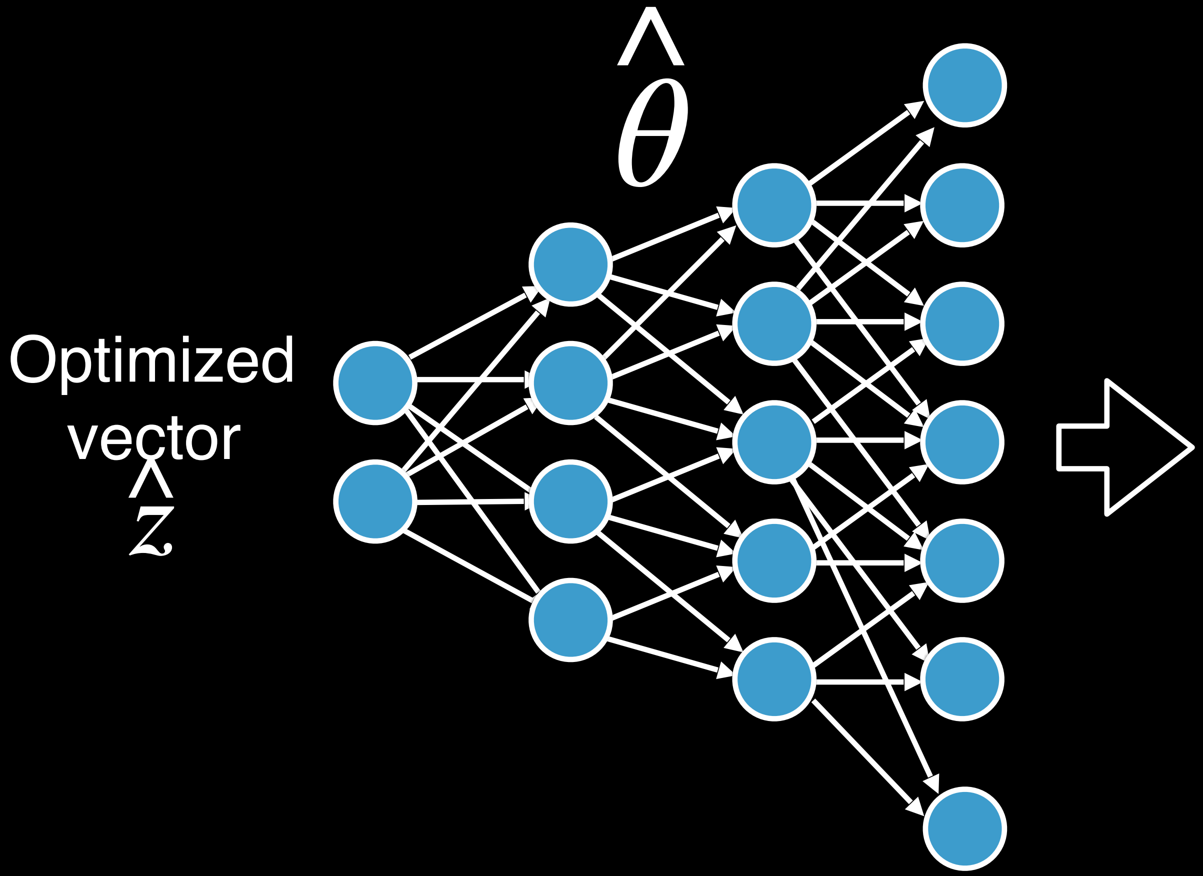
Reconstructing my own photo



Reconstructed image

$$\hat{z}, \hat{\theta} = \underset{z,\theta}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta) + R(\theta)$$

Reconstructing my own photo



Reconstructed image

$$\hat{z}, \hat{\theta} = \underset{z,\theta}{\operatorname{argmin}} L_{rec}(I, G(z,\ \theta) + R(\theta)$$

Reconstructing my own photo

Original image

Optimized z

Optimized z and adapted network

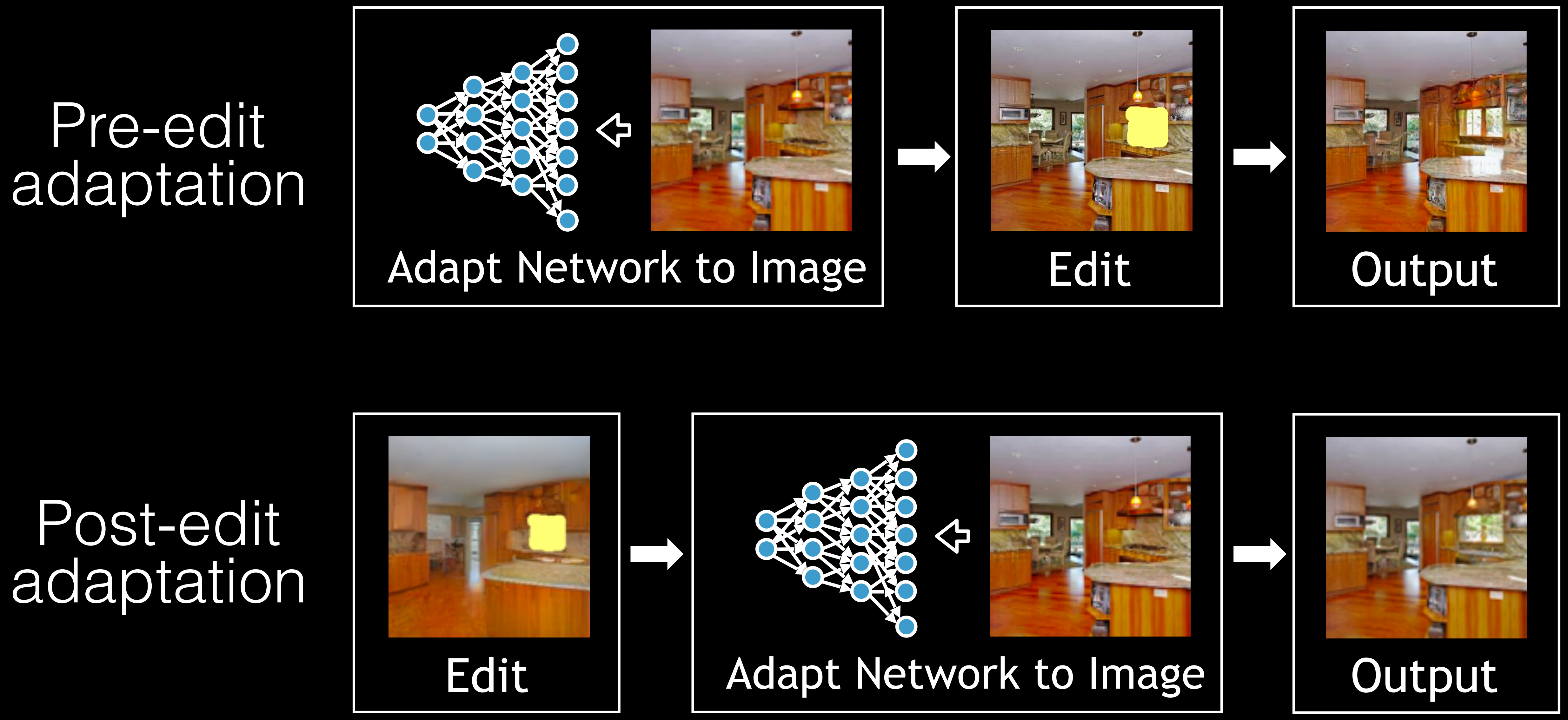
Inspired by Deep Image Prior [Ulyanov et al., 2018] and [Shocher et al., 2017]

Will image editing work?

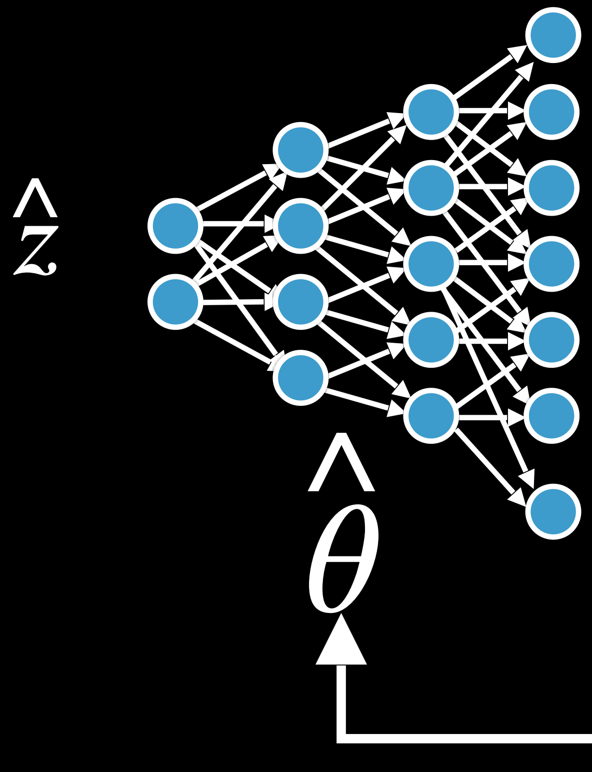
Will image editing work?

Original image and edit area

Edited result with adapted network


Pre-edit adaptation
Adapt Network to Image
Edit
Output
Post-edit adaptation
Edit
Adapt Network to Image
Output

Pre-edit adaptation



Whole Image Objective:

$$L_{rec}(I, G(z, \theta))$$

Details of pre-edit adaptation

Two ways to adjust network:

Original image

Reconstruction by unmodified G

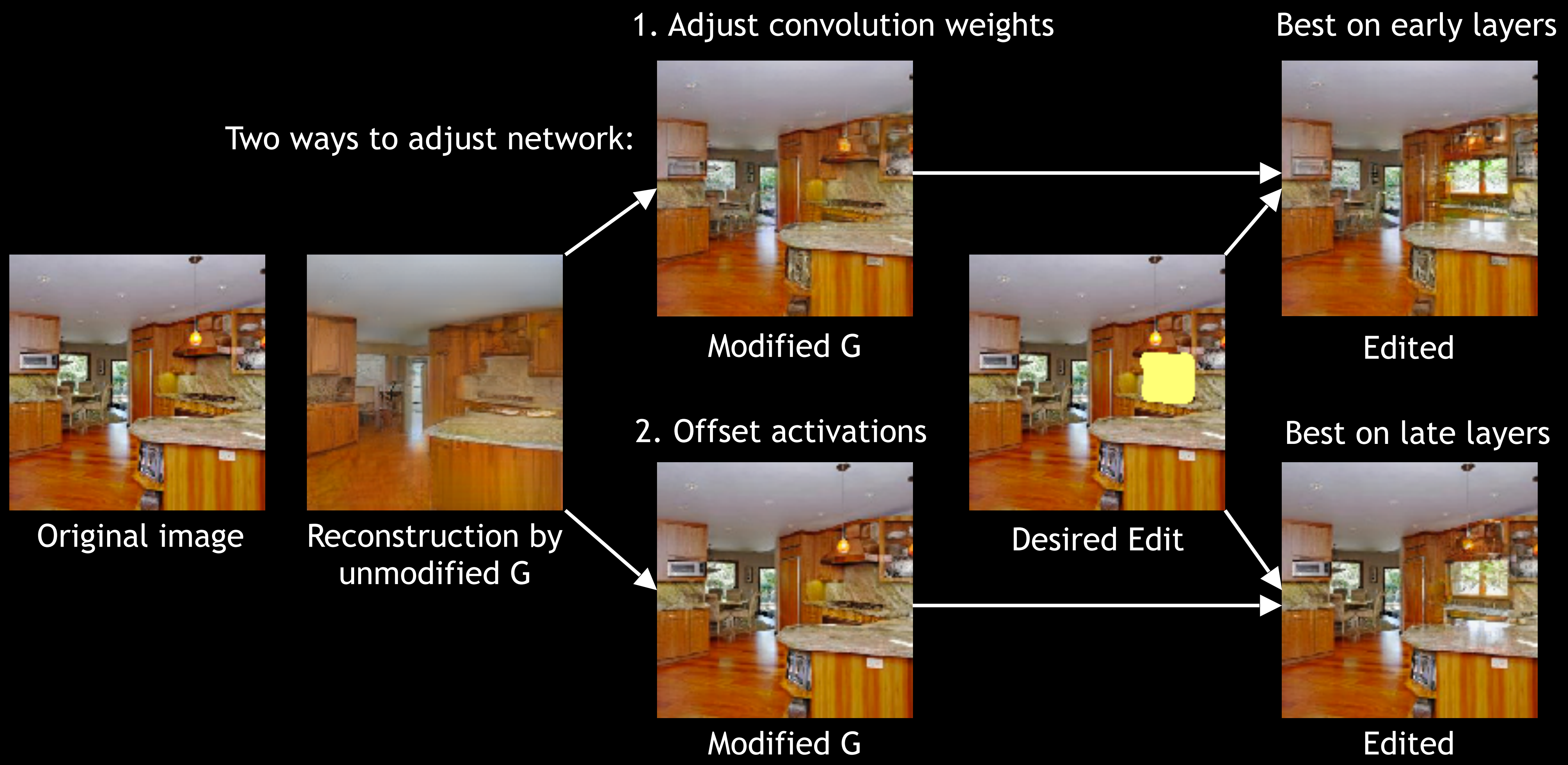
1. Adjust convolution weights

Modified G

2. Offset activations

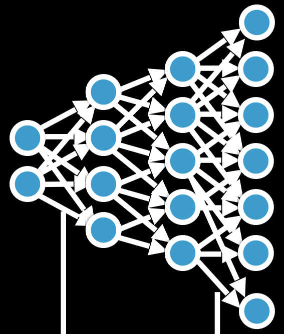
Modified G

Details of pre-edit adaptation



Details of pre-edit adaptation

Combining all the steps together…



1. Invert the original network

2. Adjust the network parameters

Uploaded image x

1a. Encode to z

Encoder net:
$z_0 = E(x)$
$z_4 = g_4(..(g_1(z_0)))$

1b. Encode to layer

Optimize z_4 :
$z_4^* \approx z_4$
$g_{15}(...(g_5(z_4^*))) \approx x$

2a. Coarse layer

Optimize g_6:
$g_6^* \approx g_6$
$g_{15}(...(g_6^*(...))) \approx x$

2b. Fine layer

Optimize d_{12}:
$d_{12}^* \approx 0$
$g_{15}(...(d_{12}+g_{12}(...))) \approx x$

Manipulating a real photo

Input image

Add windows

Output result

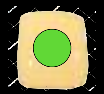
Manipulating a real photo

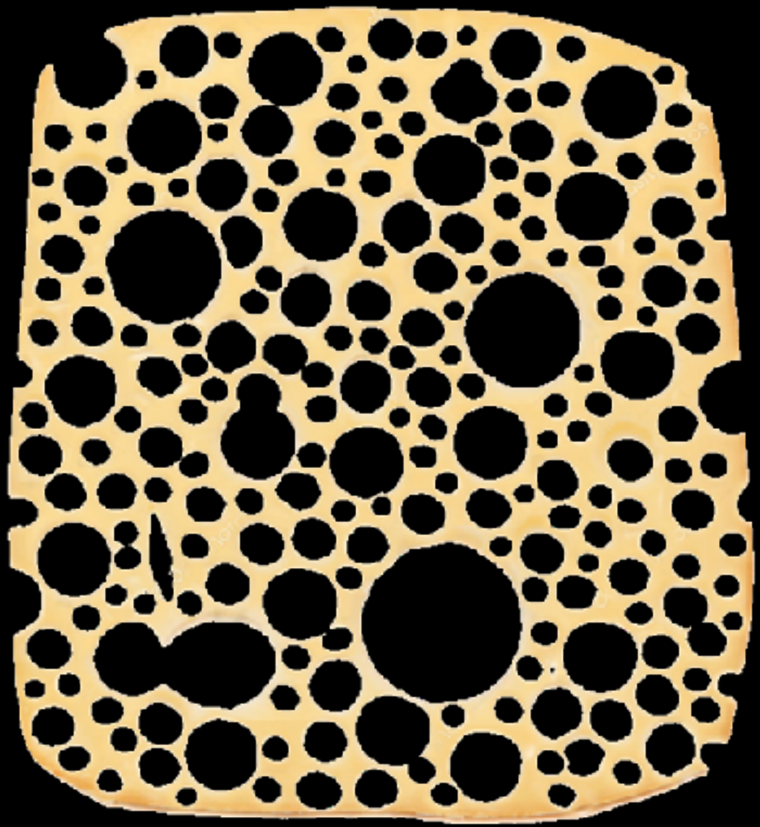
Input image

Remove chairs

Output result

Manipulating a photo outside the training set





Manipulating a photo outside the training set

Realtime editing

Online demo: **ganpaint.io**

Limitations

GAN resolution and fidelity

256px GAN

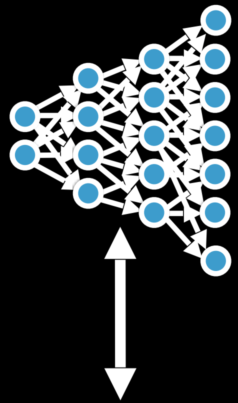
1024px original

Undesired global effects

Distorted lamp

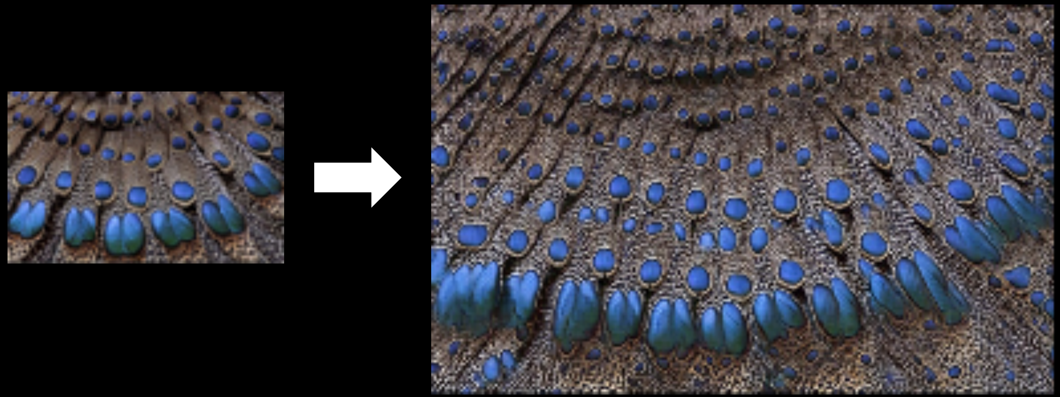
Original lamp

Adaptation Speed

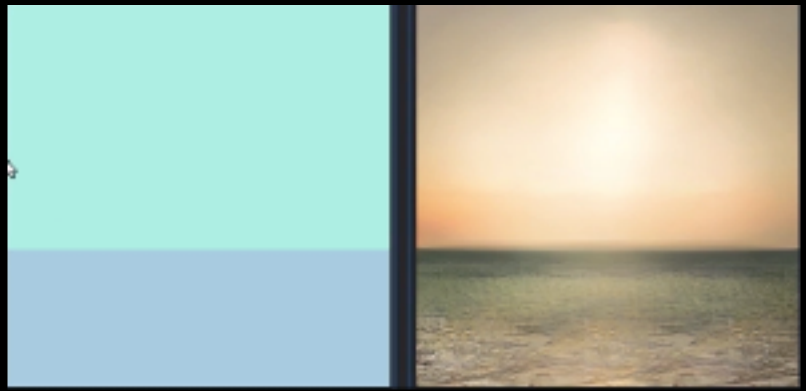


Adaptation time

Synthesis



Texture synthesis [Zhou et al.], colorization [Iizuka et al.]
superresolution [Ledig et al.], etc.



SPADE, pix2pix, pix2pixHD, Scribbler,
UNIT, MUNIT, CycleGAN, PairedCycleGAN

Low-level — High-level

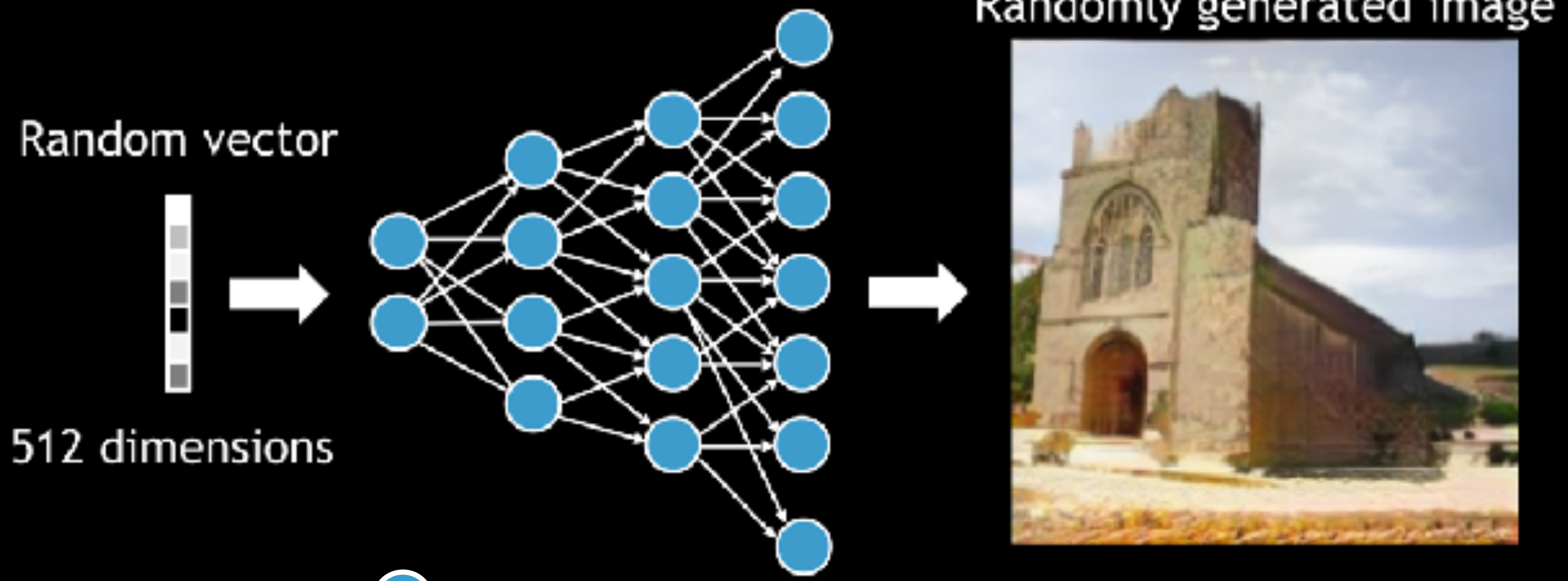
iGAN [Zhu et al.], Neural Photo Editor [Brock et al.]
IcGAN [Perarnau et al.]

GANPaint (**Our work**)
See domain-specific work: [Portenier et al.]

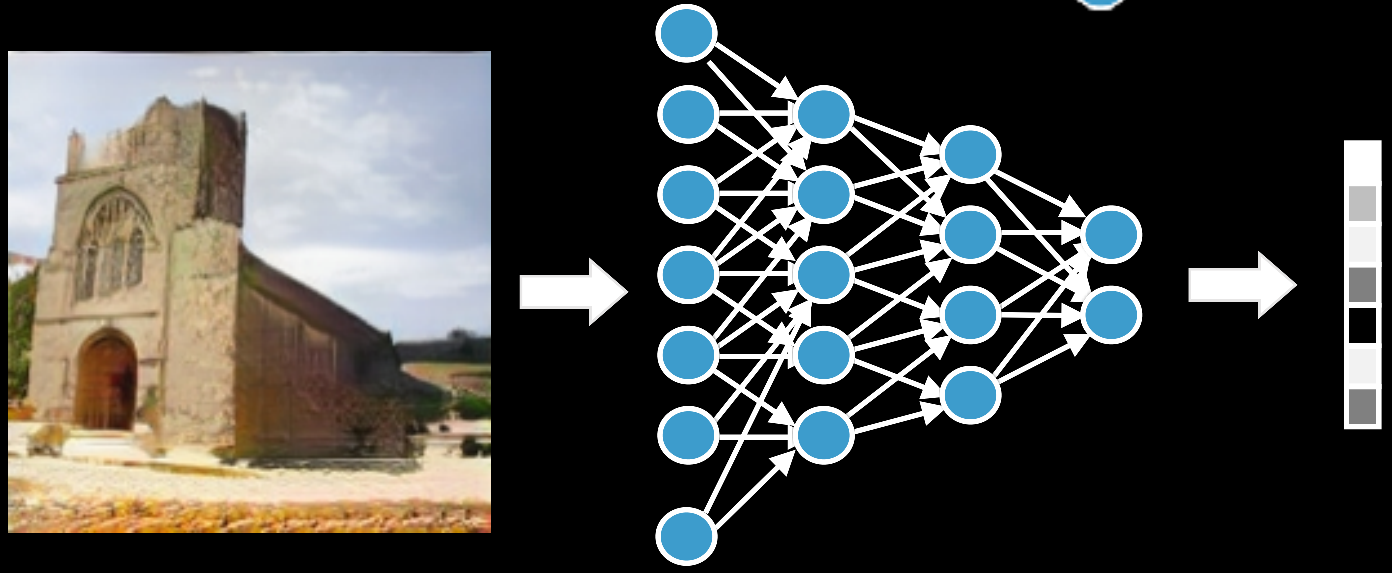
Manipulation

Dissecting neural networks

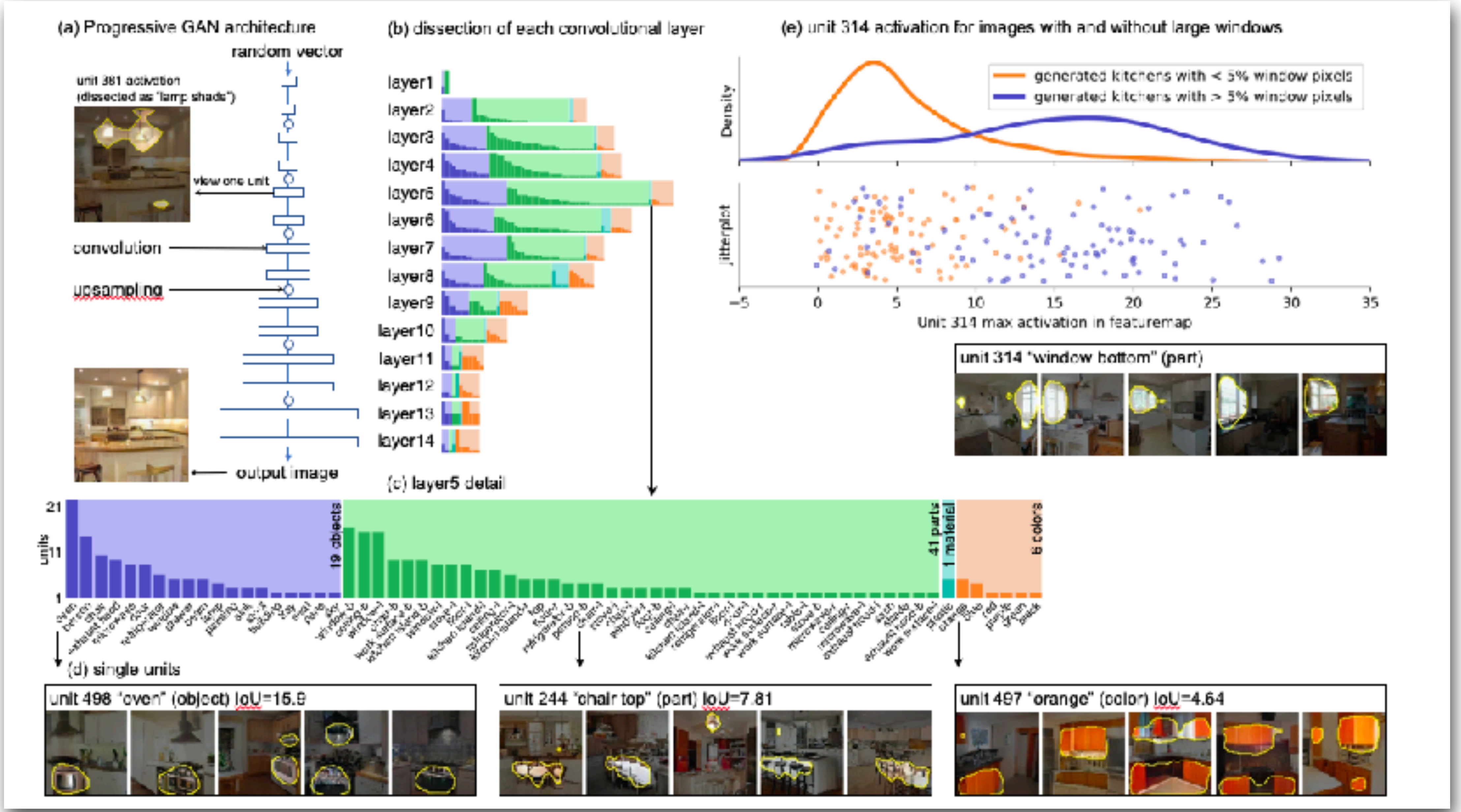
Generative network



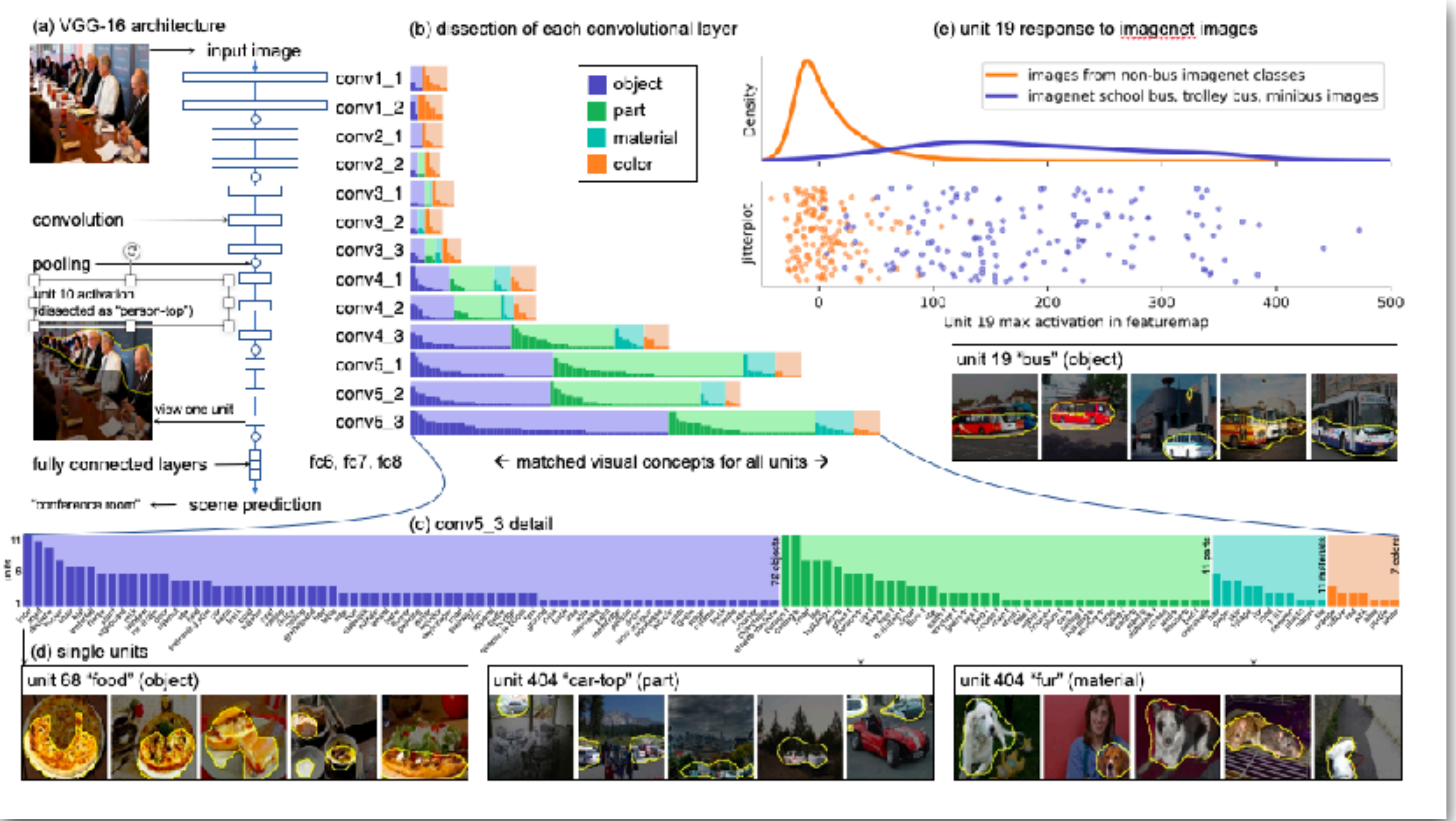
Discriminative network



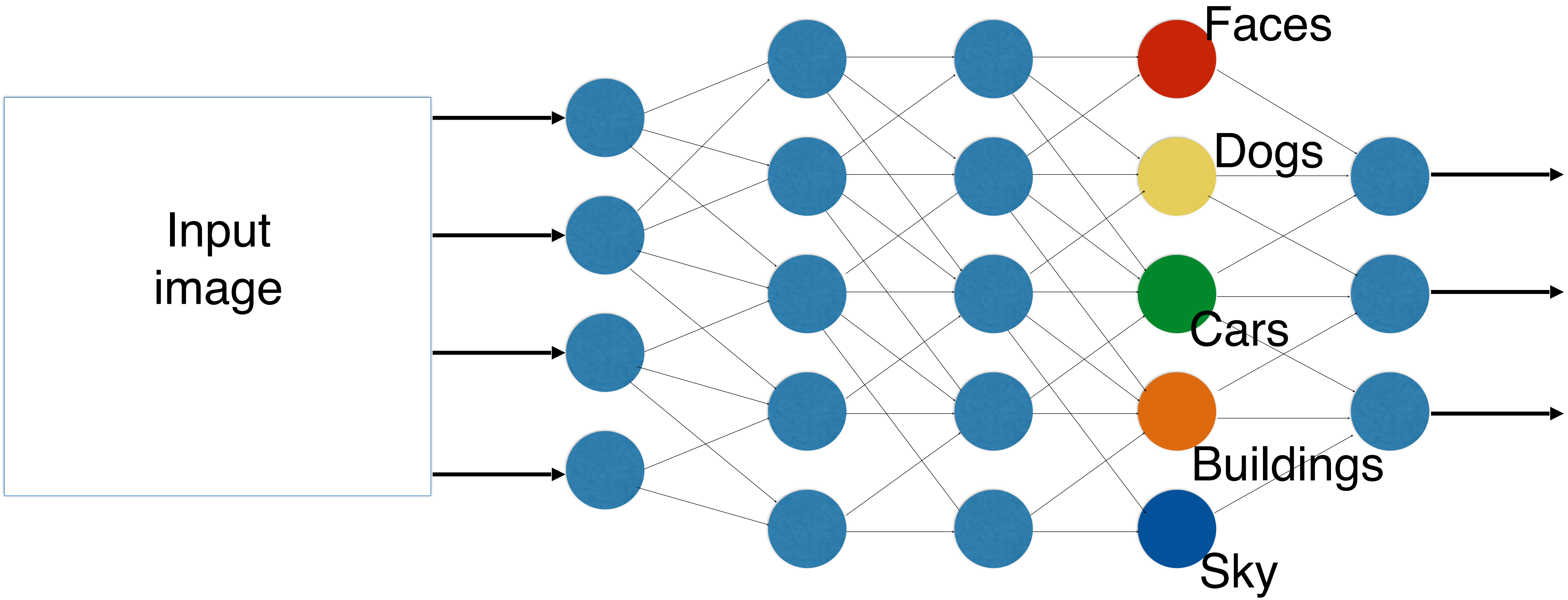
Dissection of a GAN



Dissection of a ConvNet



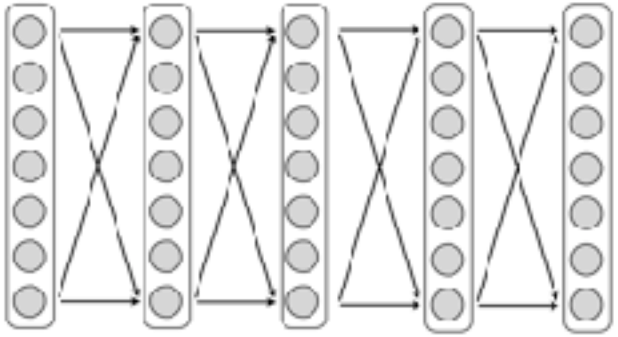
What is the network doing?



Object Detectors Emerge in Deep Scene CNNs. ICLR 2015.
Network Dissection: Quantifying Interpretability of Deep Visual Representations. CVPR 2017.

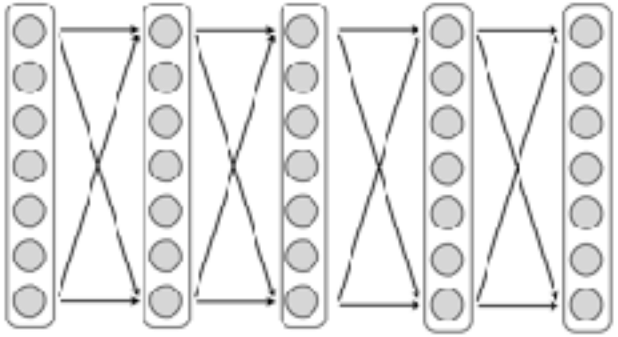
Network dissection

1. Train your network to solve your task



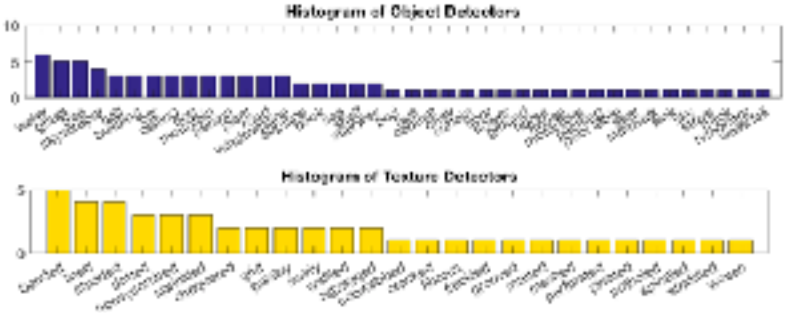
Living room
Kitchen
Coast
Theater
...

2. Run network dissection to interpret the internal representation

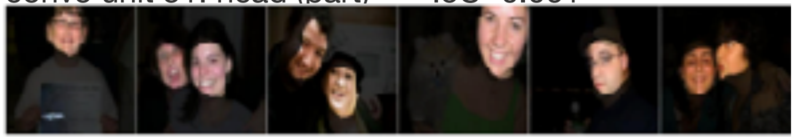


Network Dissection

http://netdissect.csail.mit.edu/



conv5 unit 51: head (part) IoU=0.061



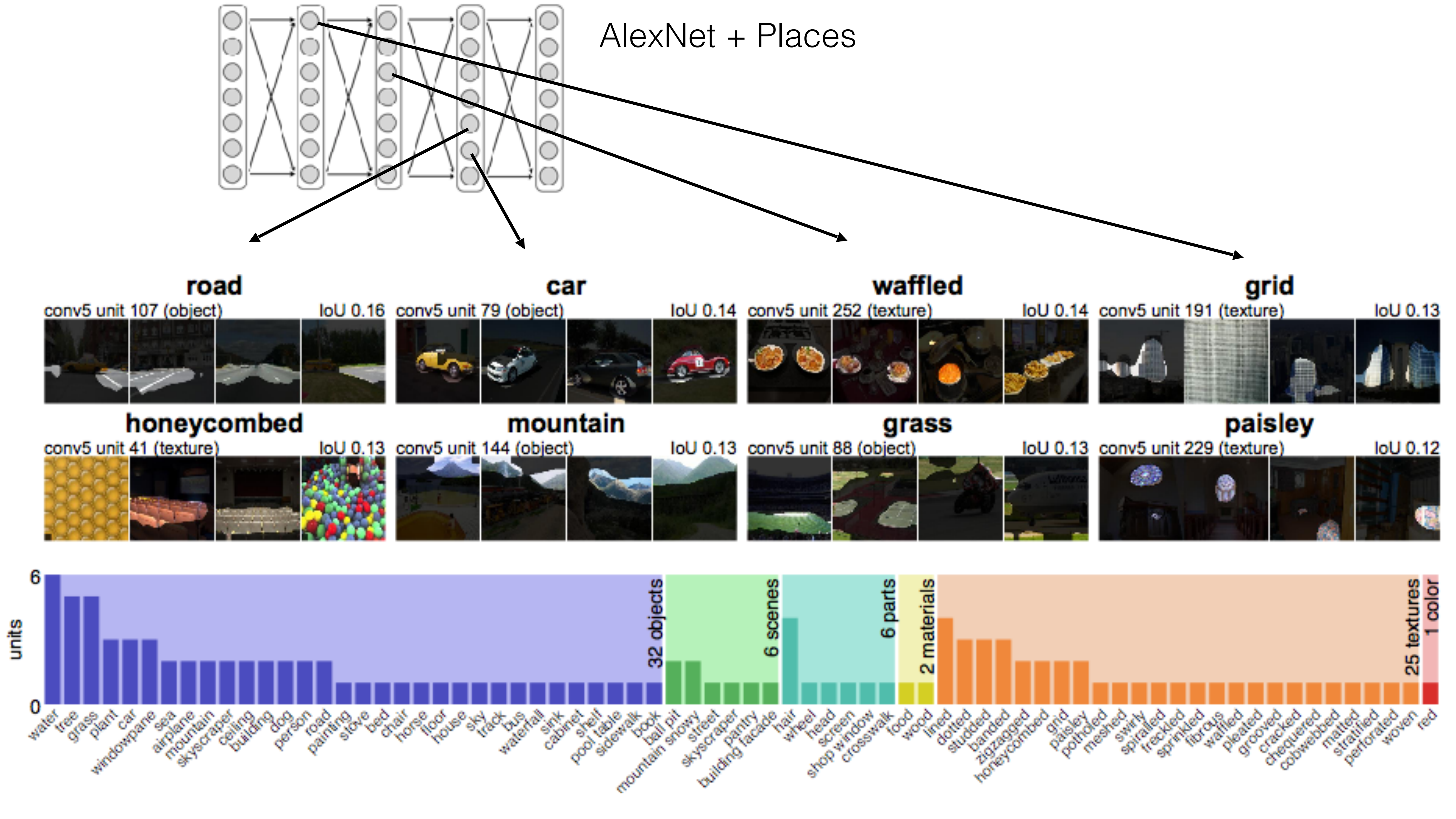
conv5 unit 124: creek (scene) IoU=0.031

conv5 unit 79 car (object) IoU=0.13

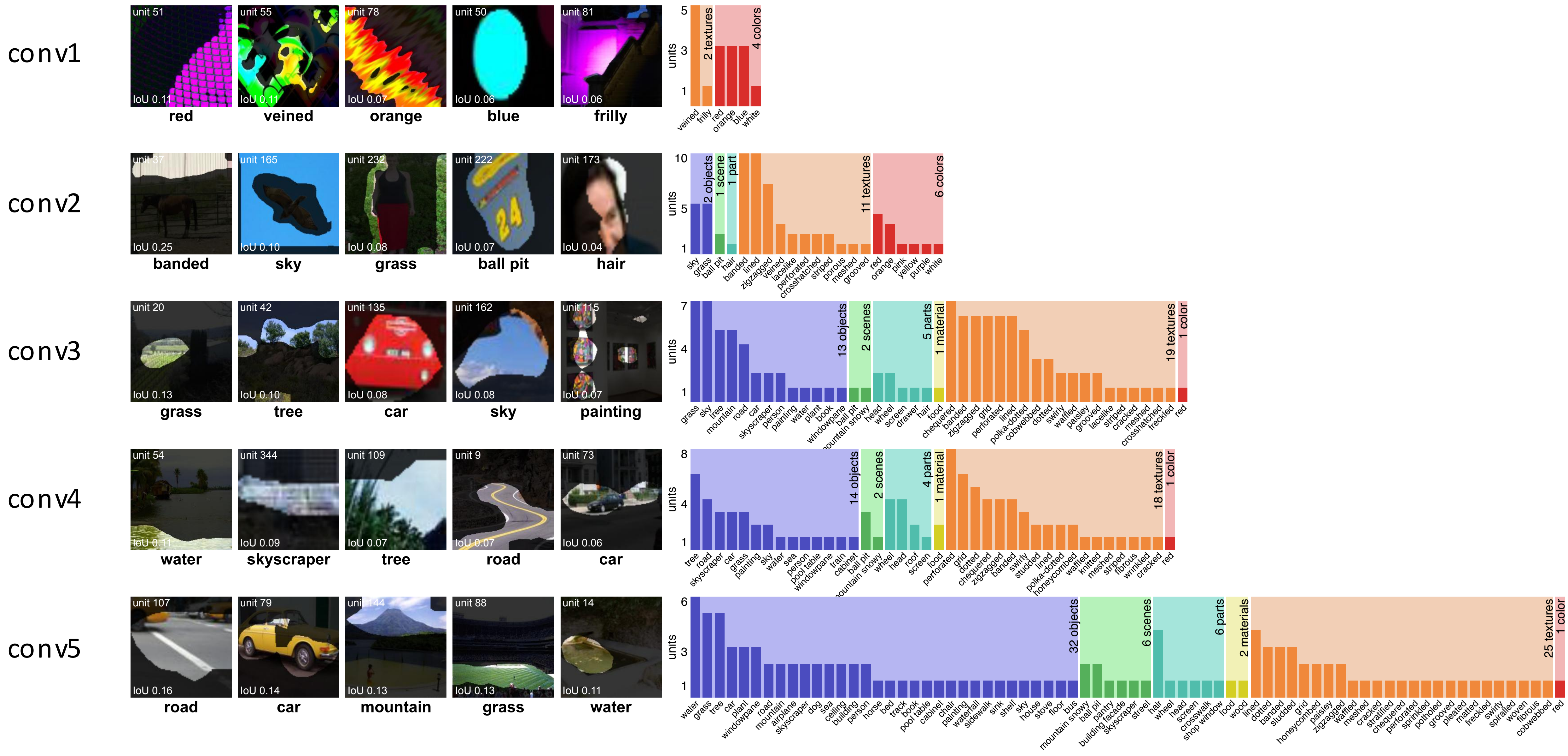
conv5 unit 107 road (object) IoU=0.15

conv5 unit 144 mountain (object) IoU=0.13

http://netdissect.csail.mit.edu/


AlexNet + Places
road
conv5 unit 107 (object)
IoU 0.16
car
conv5 unit 79 (object)
IoU 0.14
waffled
conv5 unit 252 (texture)
IoU 0.14
grid
conv5 unit 191 (texture)
IoU 0.13
honeycombed
conv5 unit 41 (texture)
IoU 0.13
mountain
conv5 unit 144 (object)
IoU 0.13
grass
conv5 unit 88 (object)
IoU 0.13
paisley
conv5 unit 229 (texture)
IoU 0.12
units
6
0
32 objects
6 scenes
6 parts
2 materials
25 textures
1 color
water tree grass plant car windowpane sea airplane mountain skyscraper ceiling building dog person road painting stove bed chair horse floor house sky track bus waterfall sink cabinet shelf pool table sidewalk book
ball pit mountain snowy street skyscraper pantry building facade
hair wheel head screen shop window crosswalk
food wood
lined dotted studded banded zigzagged honeycombed grid paisley potholed meshed swirly spiralled freckled sprinkled fibrous waffled pleated grooved cracked chequered cobwebbed matted stratified perforated woven
red

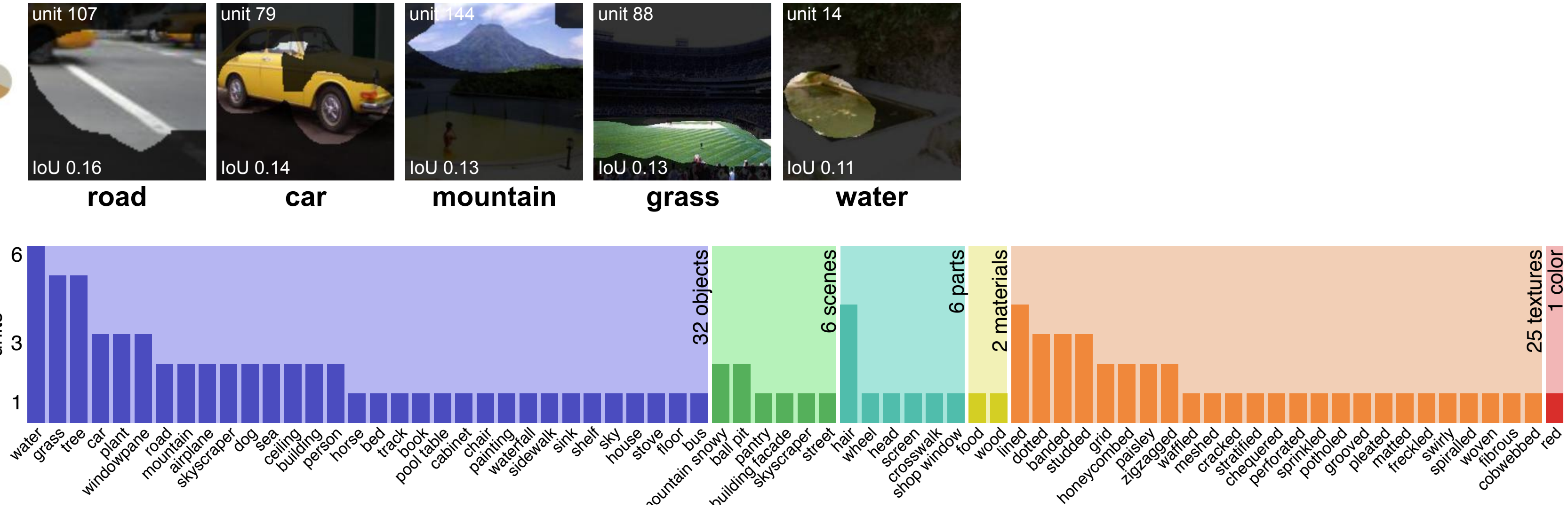
AlexNet + Places



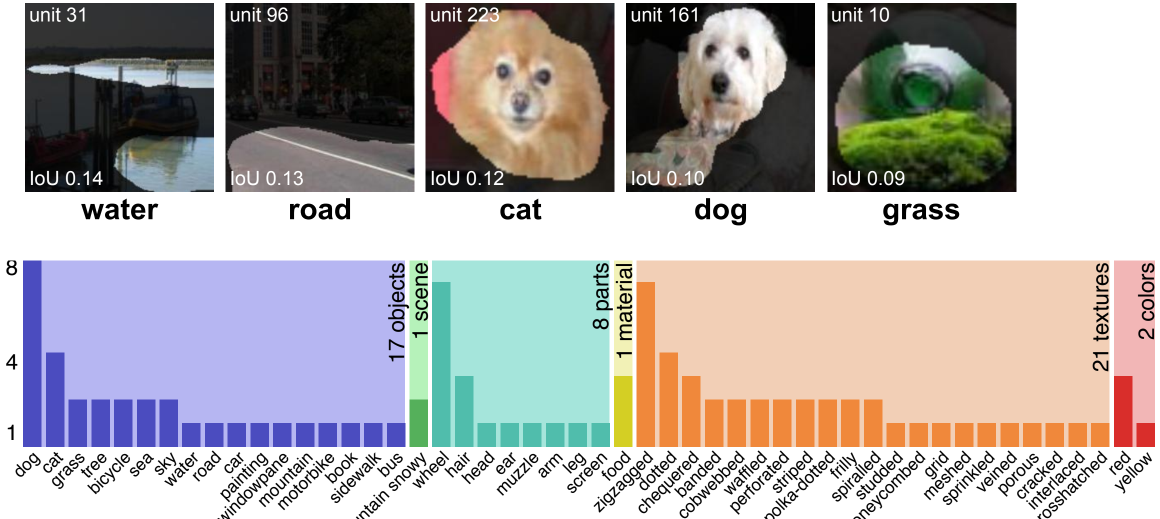
Alexnet

Architecture: AlexNet

places



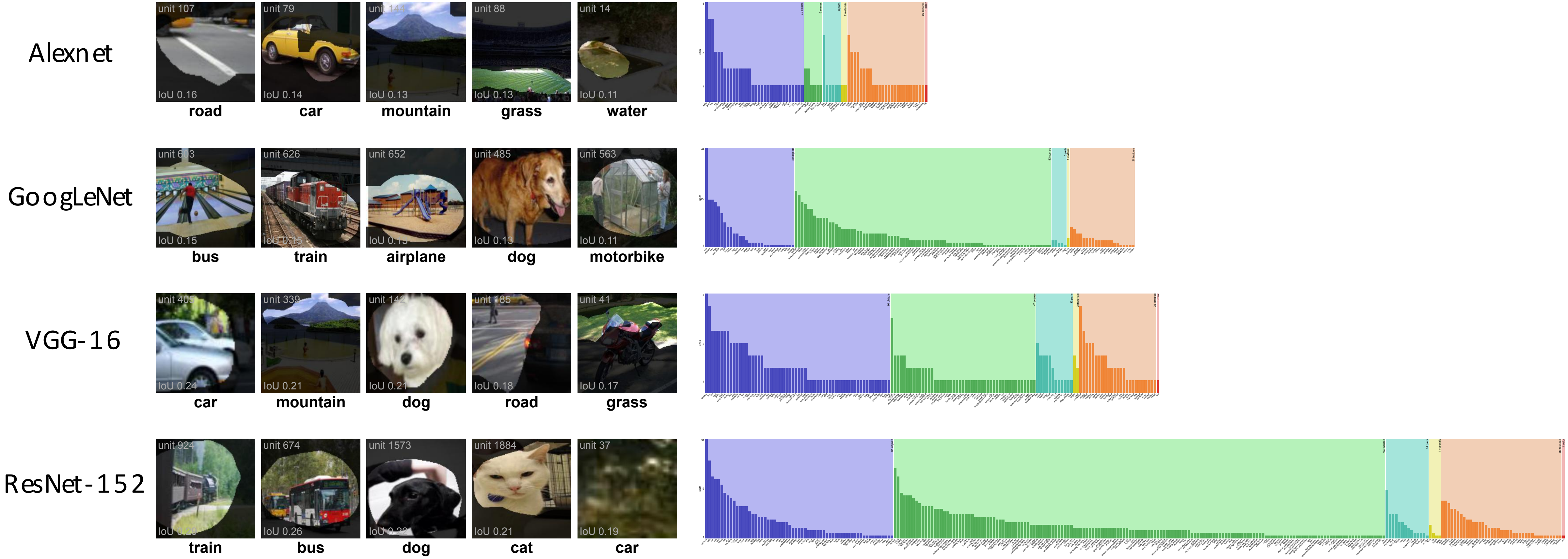
IMAGENET



Top layer shown (conv 5)

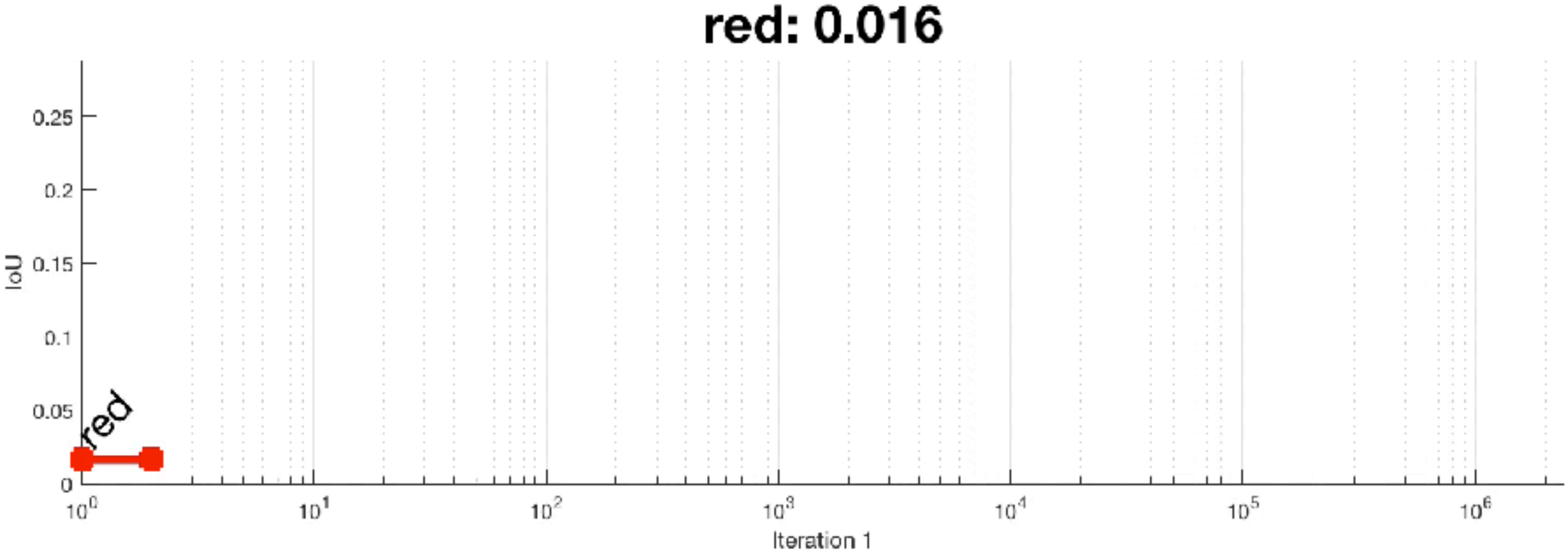
Comparing different architectures

Task: Places classification



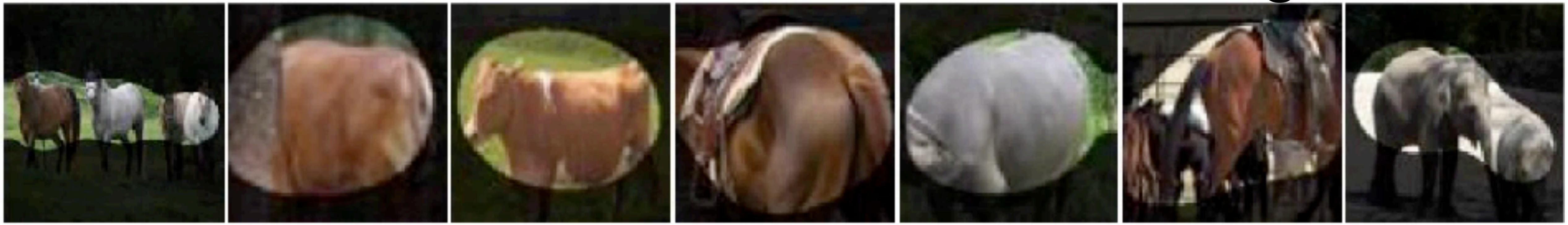
Top layer shown

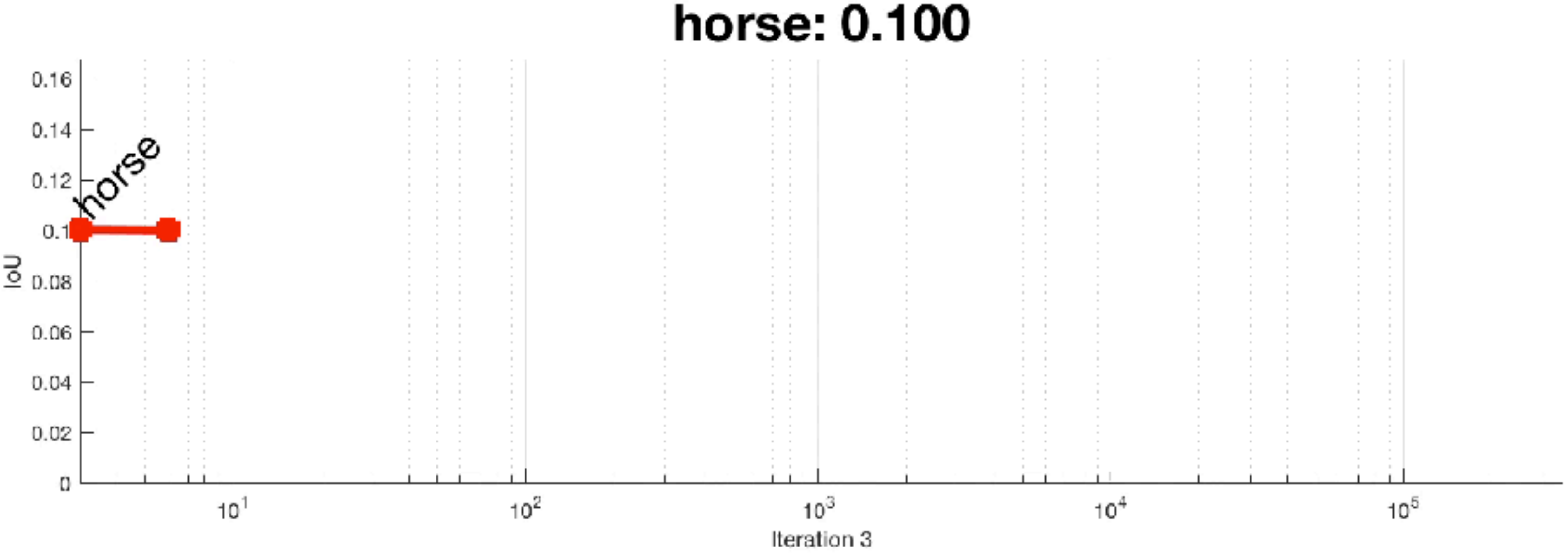
Evolution of a unit during training



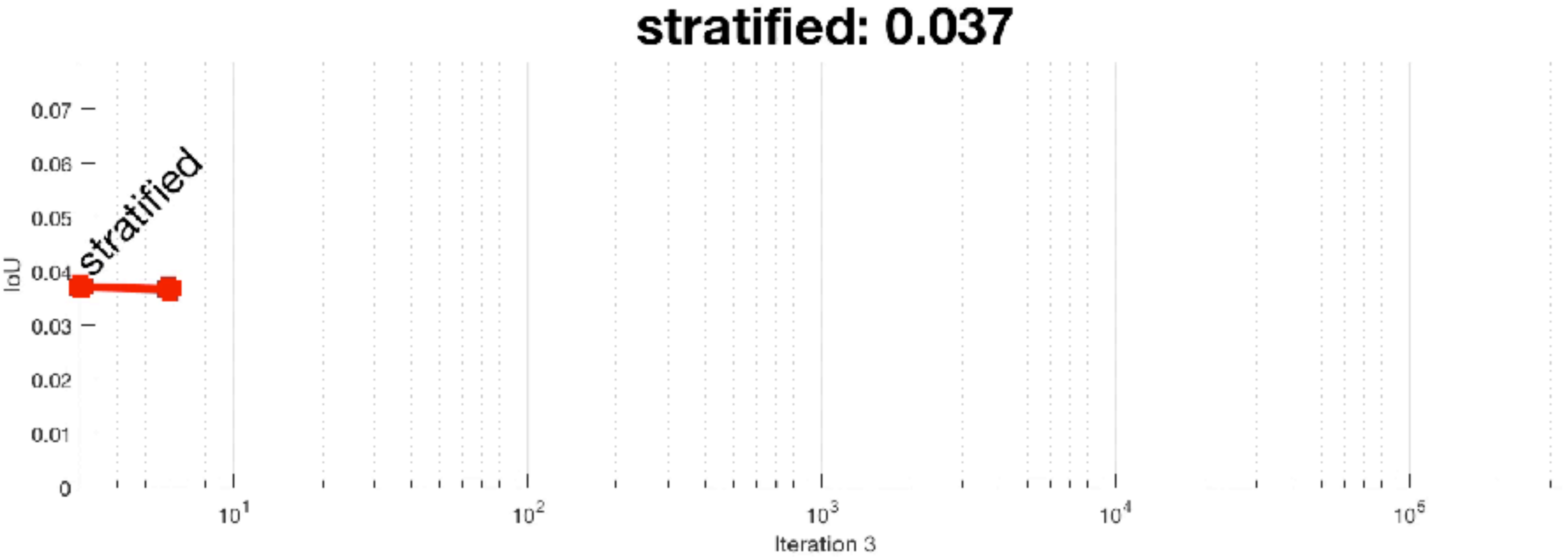
Fine-tune network from Places to ImageNet

Fine-tune network from Places to ImageNet

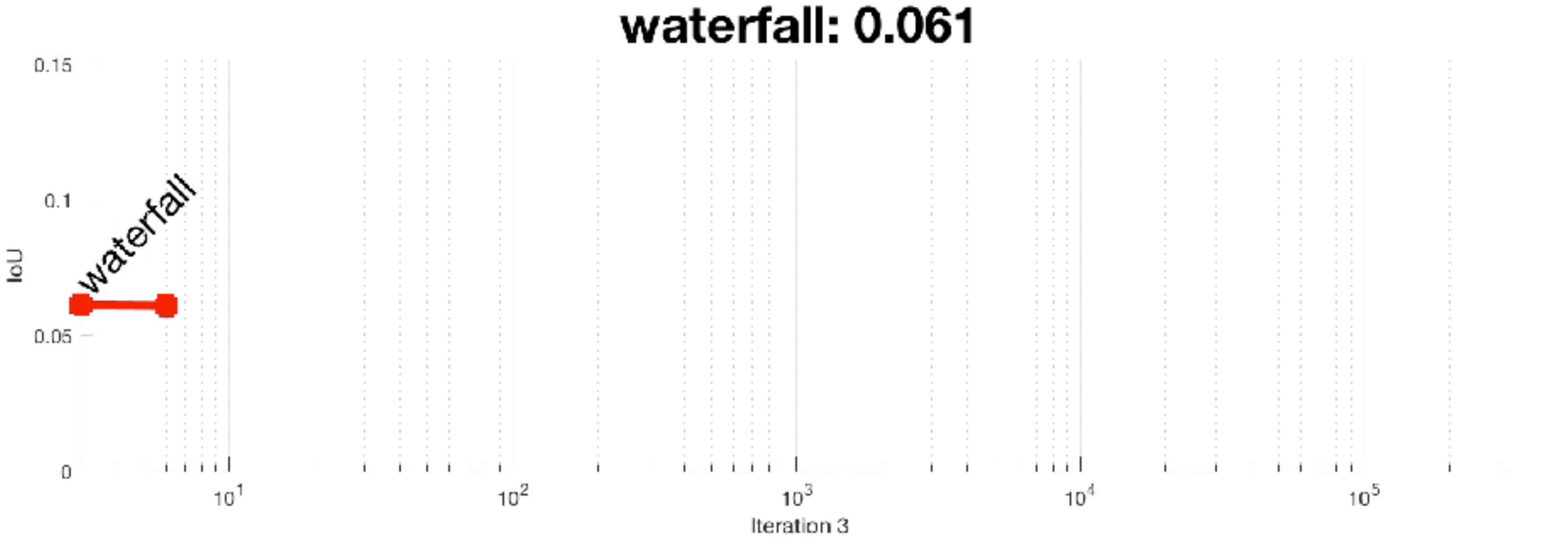




Fine-tune network from Places to ImageNet



Fine-tune network from Places to ImageNet



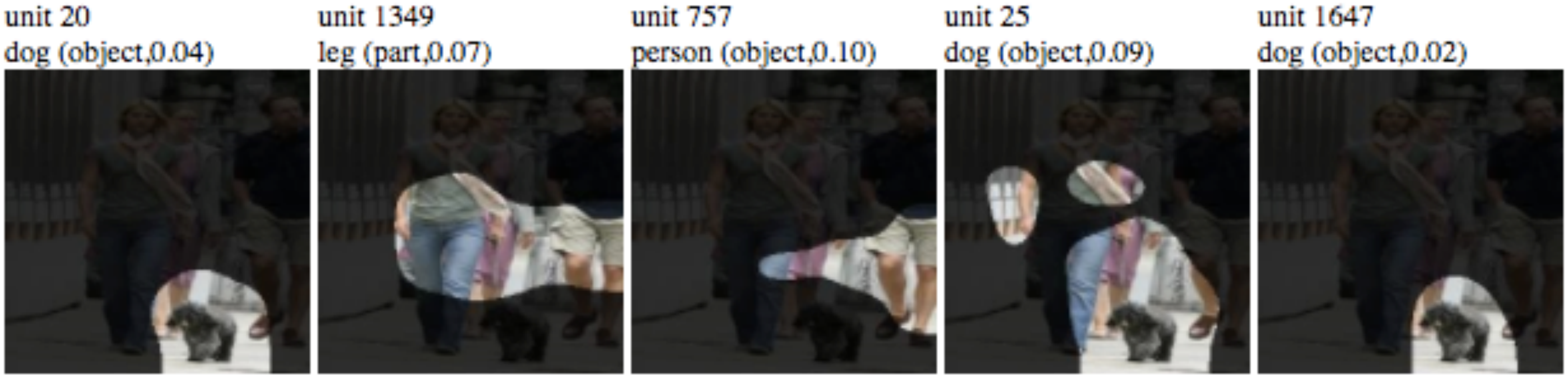
Explaining the output

Walking the dog

Explaining the output

Output: Walking the dog

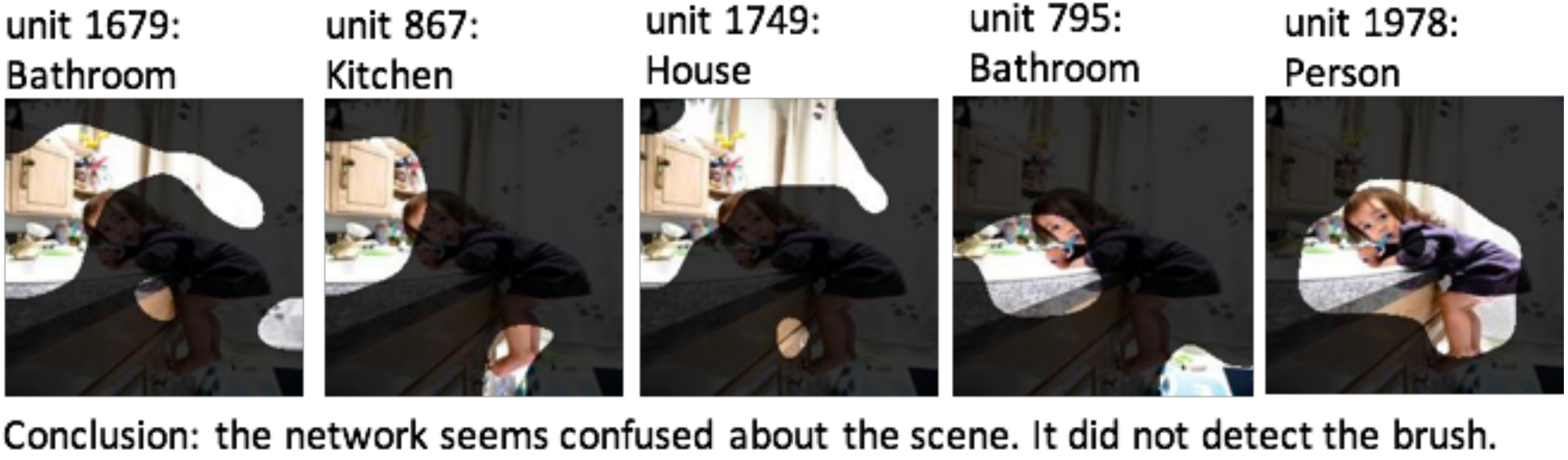
Explanation, *I saw the following evidence:*



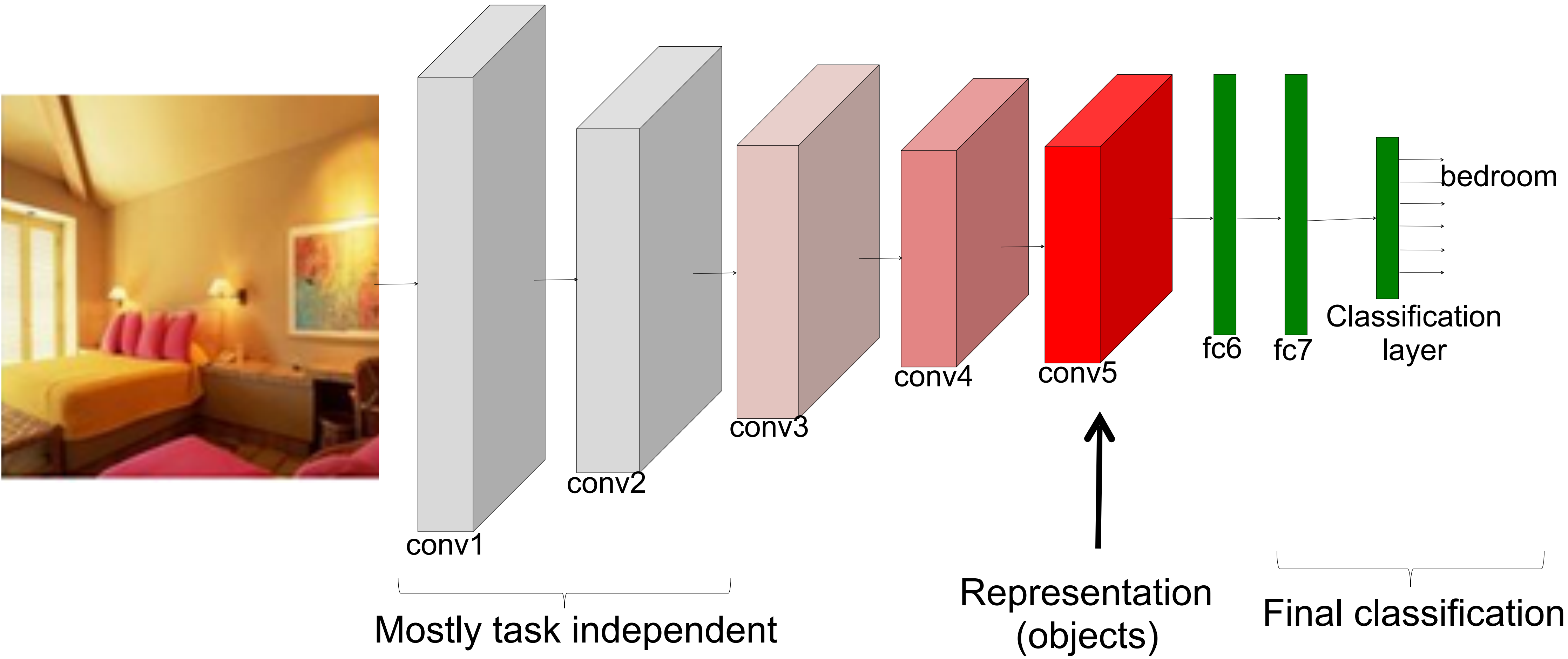
Explaining the output

Output: washing dishes.

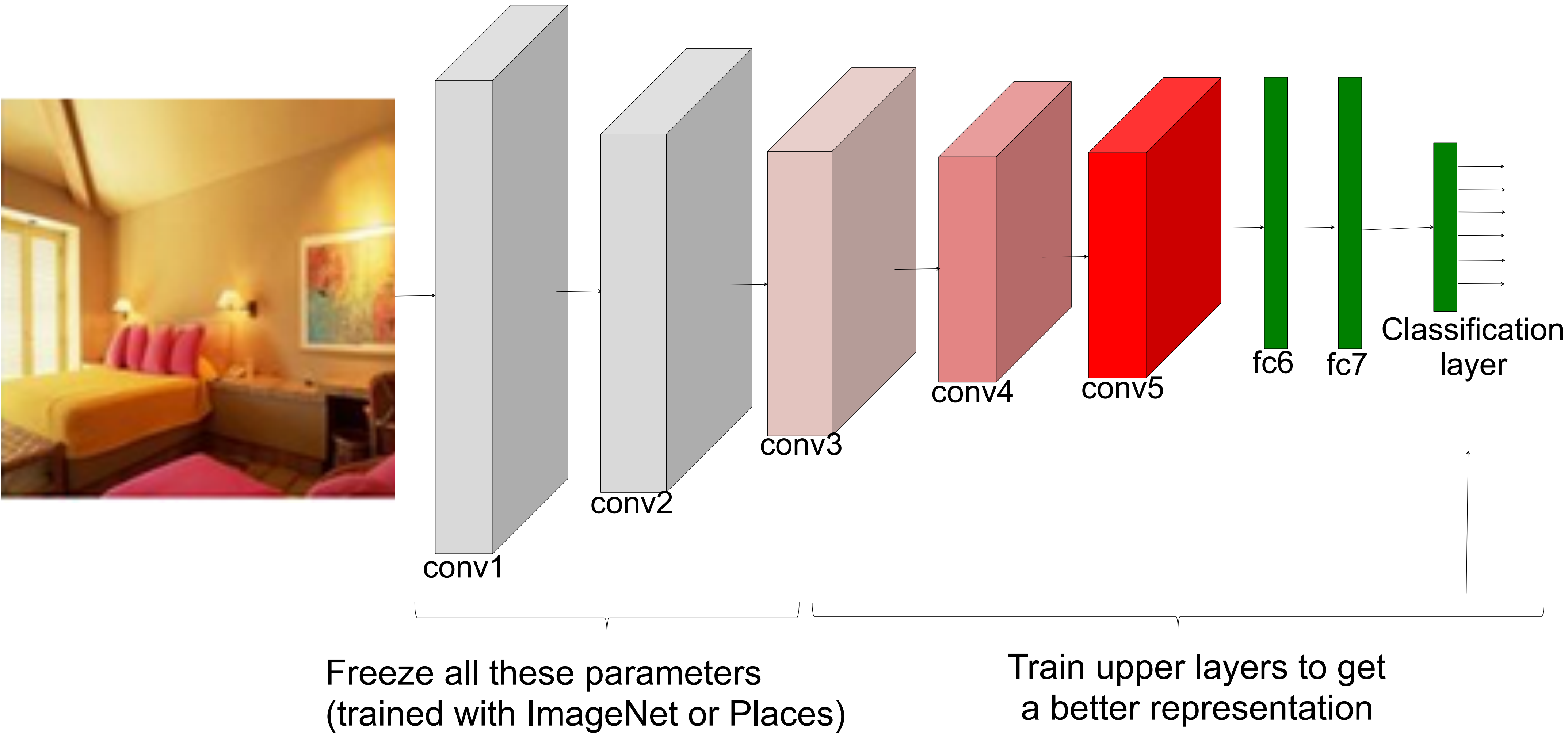
Explanation, *I saw the following evidence:*



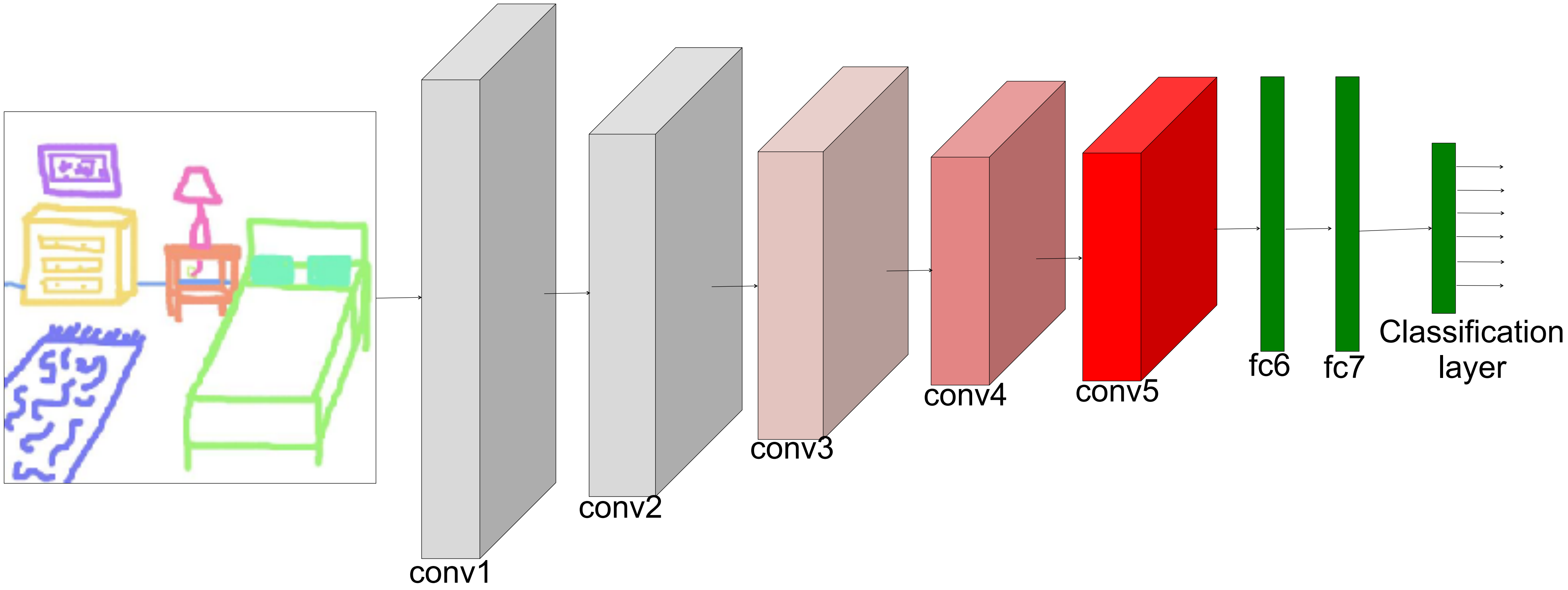
in AlexNet…


conv1
conv2
conv3
conv4
conv5
fc6
fc7
Classification layer
bedroom
Mostly task independent
Representation (objects)
Final classification

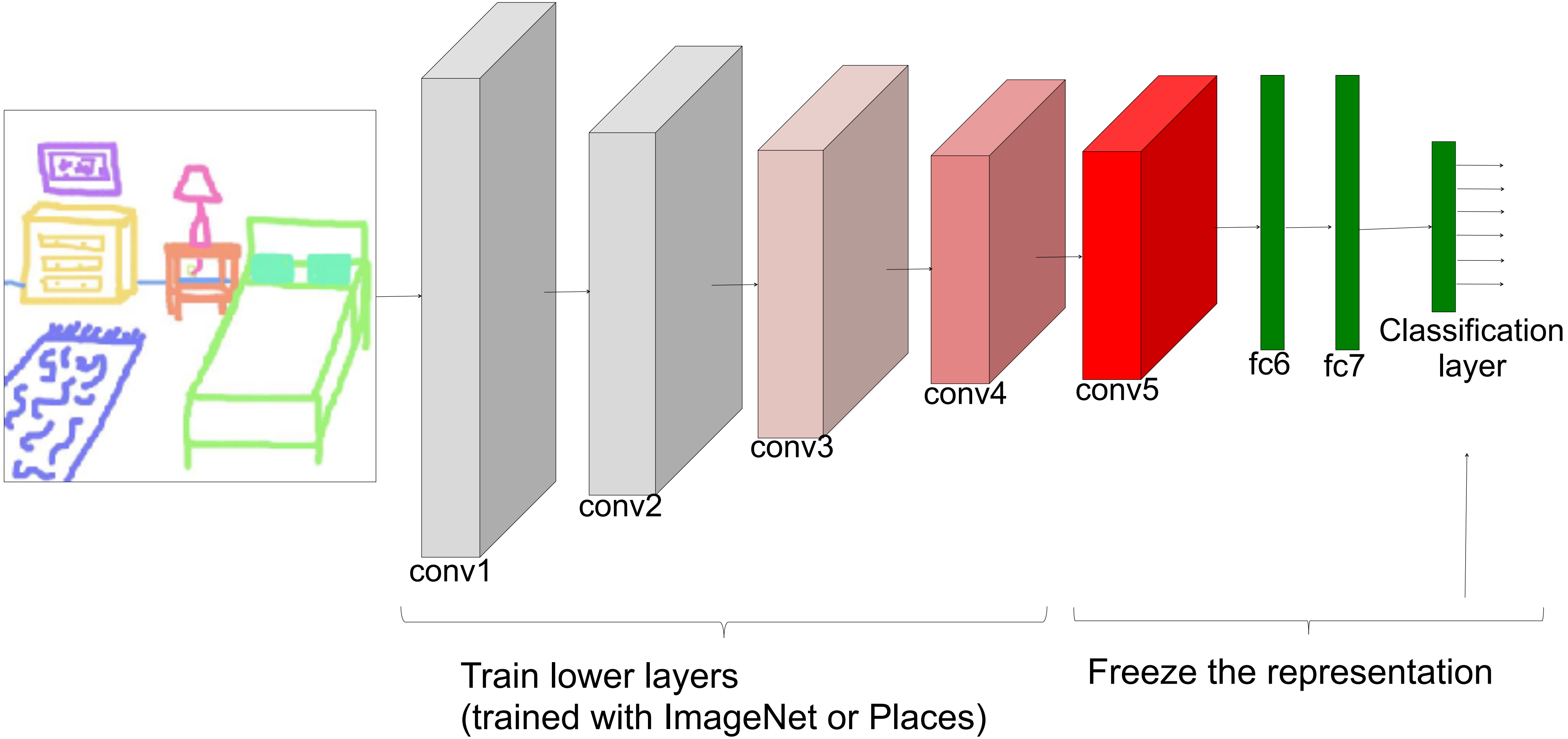
Strategies for training for new tasks



What happens when the input changes?



Strategies for training for new tasks



Line drawings

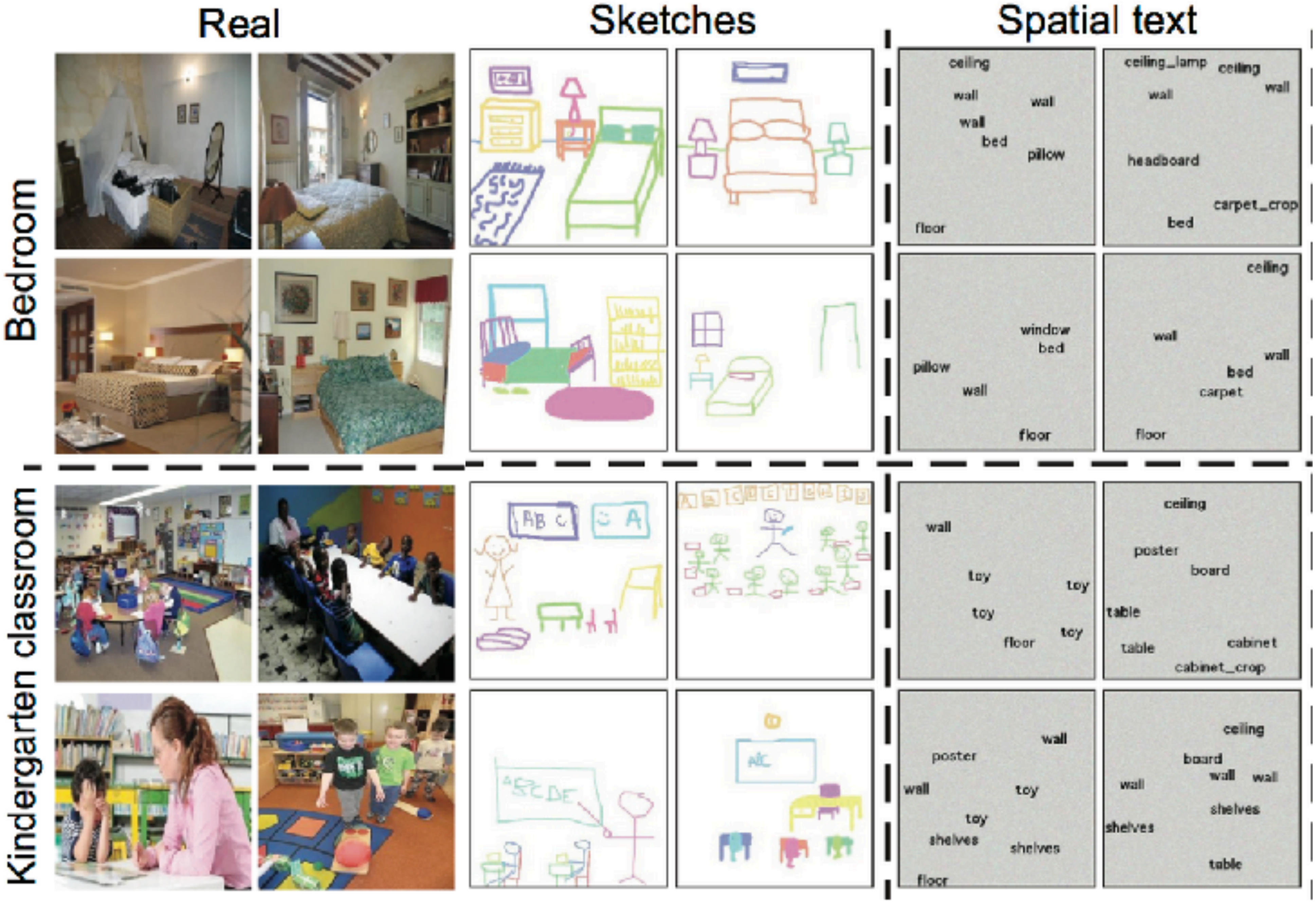
Learning Aligned Cross-Modal Representations from Weakly Aligned Data. Ll. Castrejón*, Y. Aytar*, C. Vondrick, H. Pirsiavash and A. Torralba. CVPR 2016

Localized words

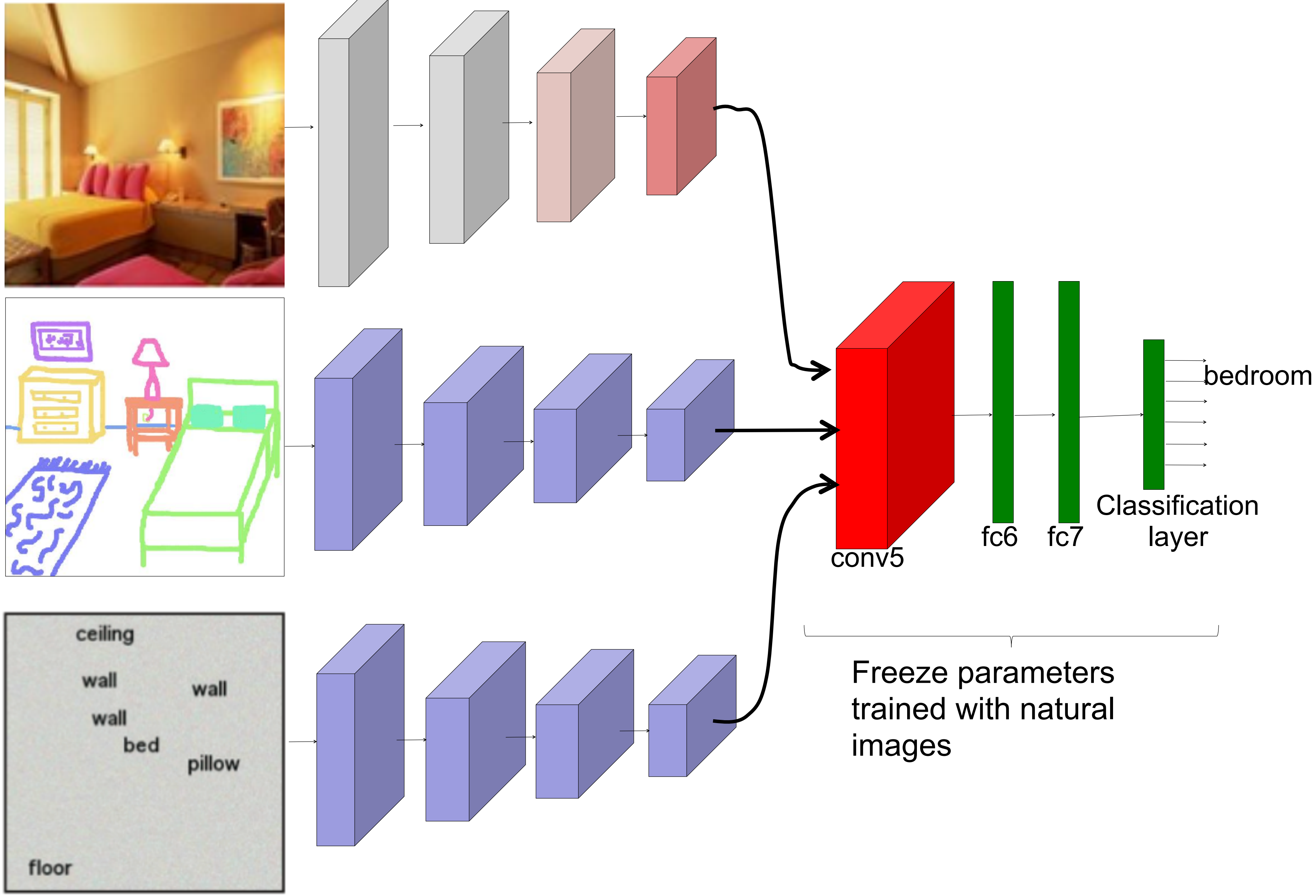
Highway

Bedroom

Learning Aligned Cross-Modal Representations from Weakly Aligned Data. Ll. Castrejón*, Y. Aytar*, C. Vondrick, H. Pirsiavash and A. Torralba. CVPR 2016

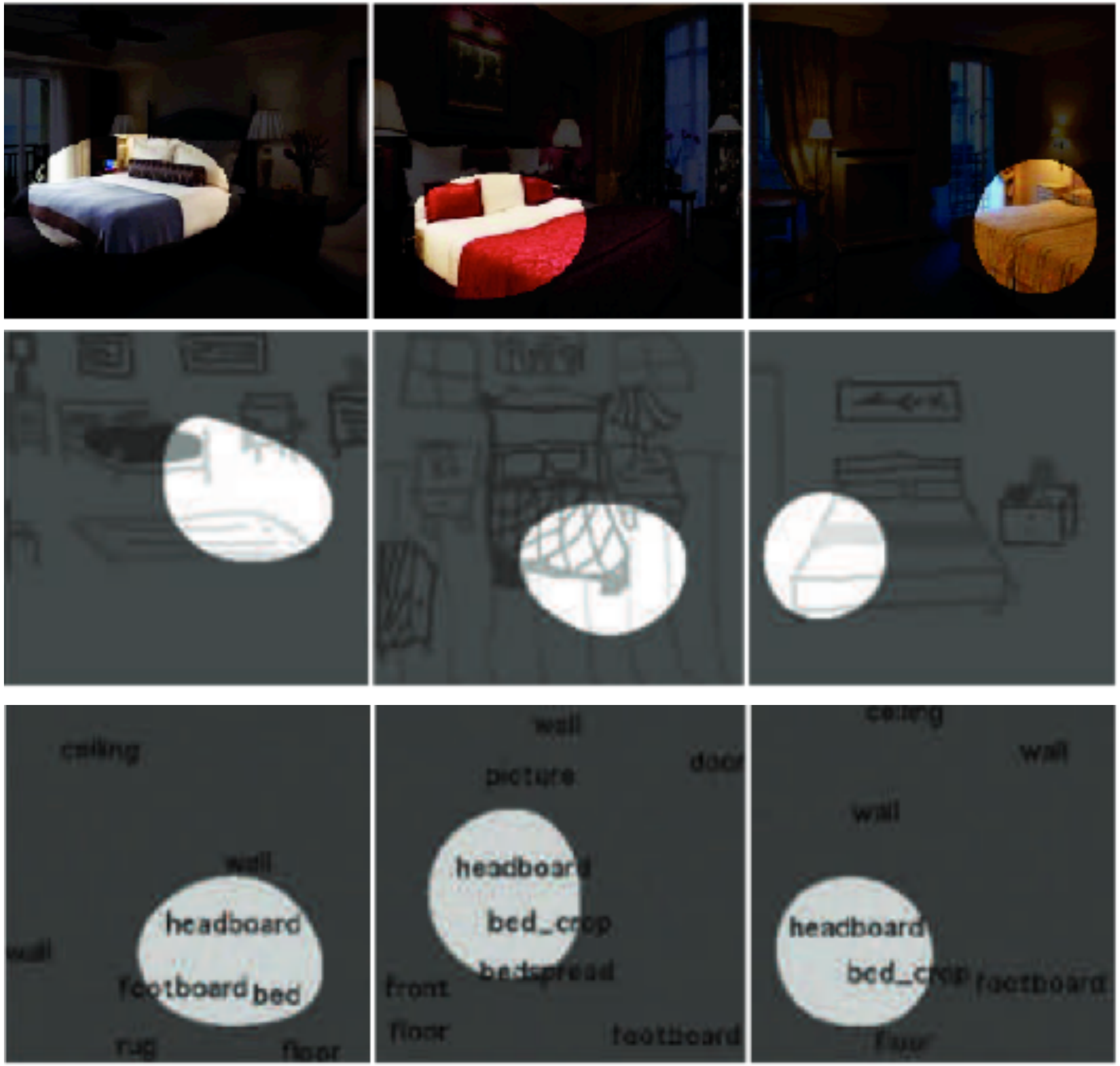


http://cmplaces.csail.mit.edu/



Learning Aligned Cross-Modal Representations from Weakly Aligned Data. Ll. Castrejón*, Y. Aytar*, C. Vondrick, H. Pirsiavash and A. Torralba. CVPR 2016

Unit 115
(Bed)



. . .

. . .

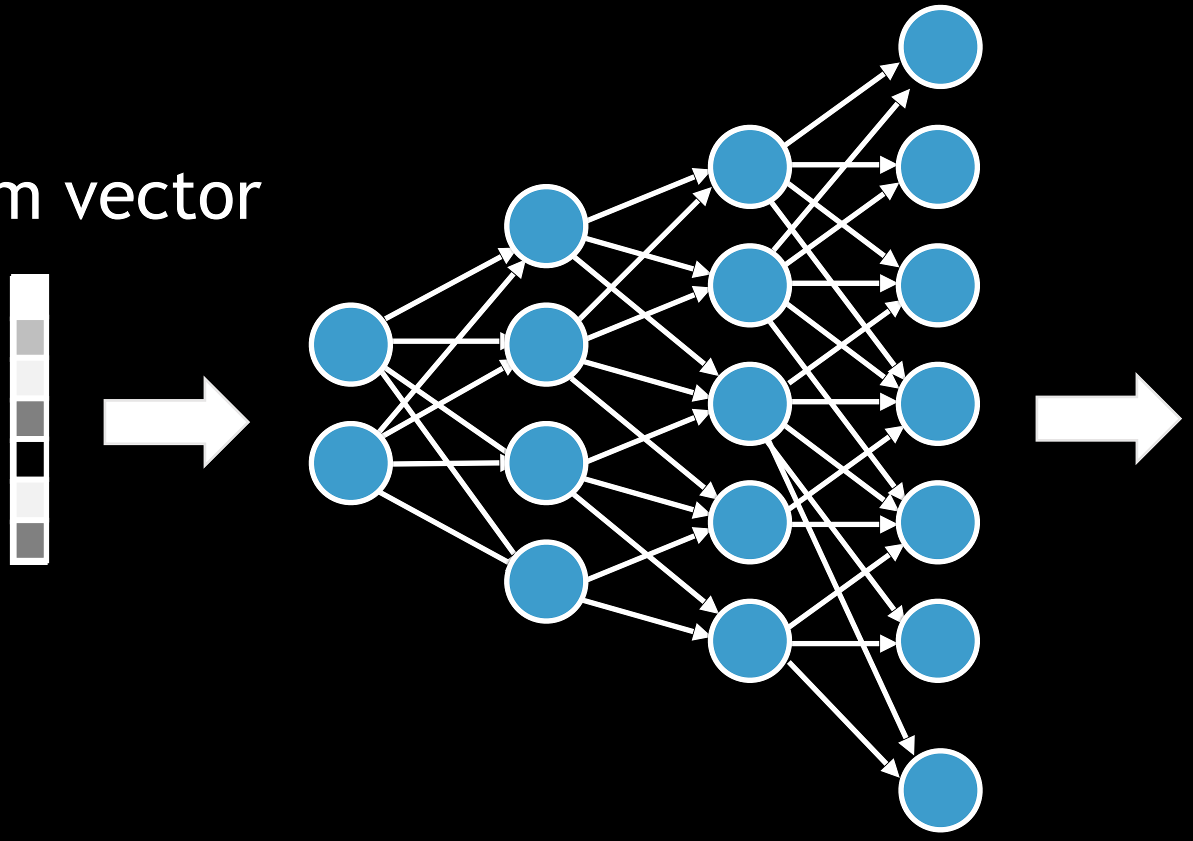
. . .

Learning Aligned Cross-Modal Representations from Weakly Aligned Data. Ll. Castrejón*, Y. Aytar*, C. Vondrick, H. Pirsiavash and A. Torralba. CVPR 2016

Datasets are usually a closed set of images…

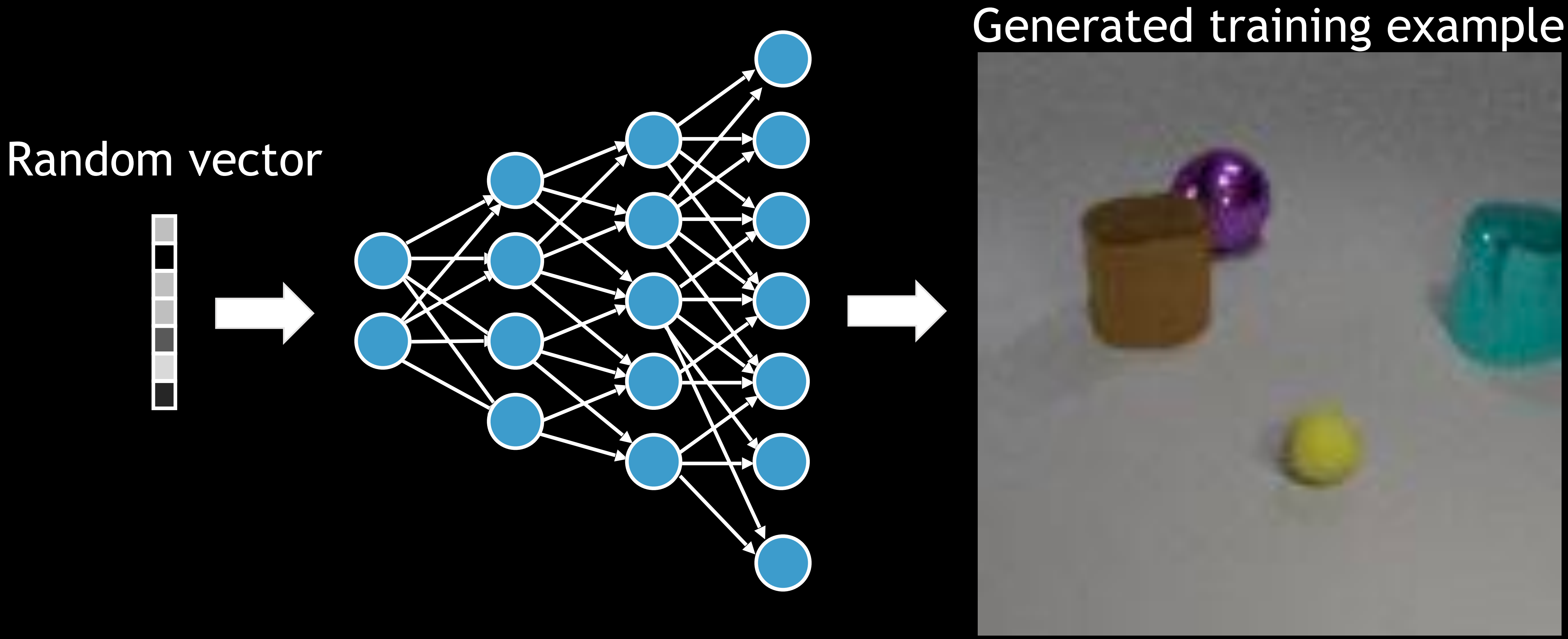
Generating our training data

Random vector

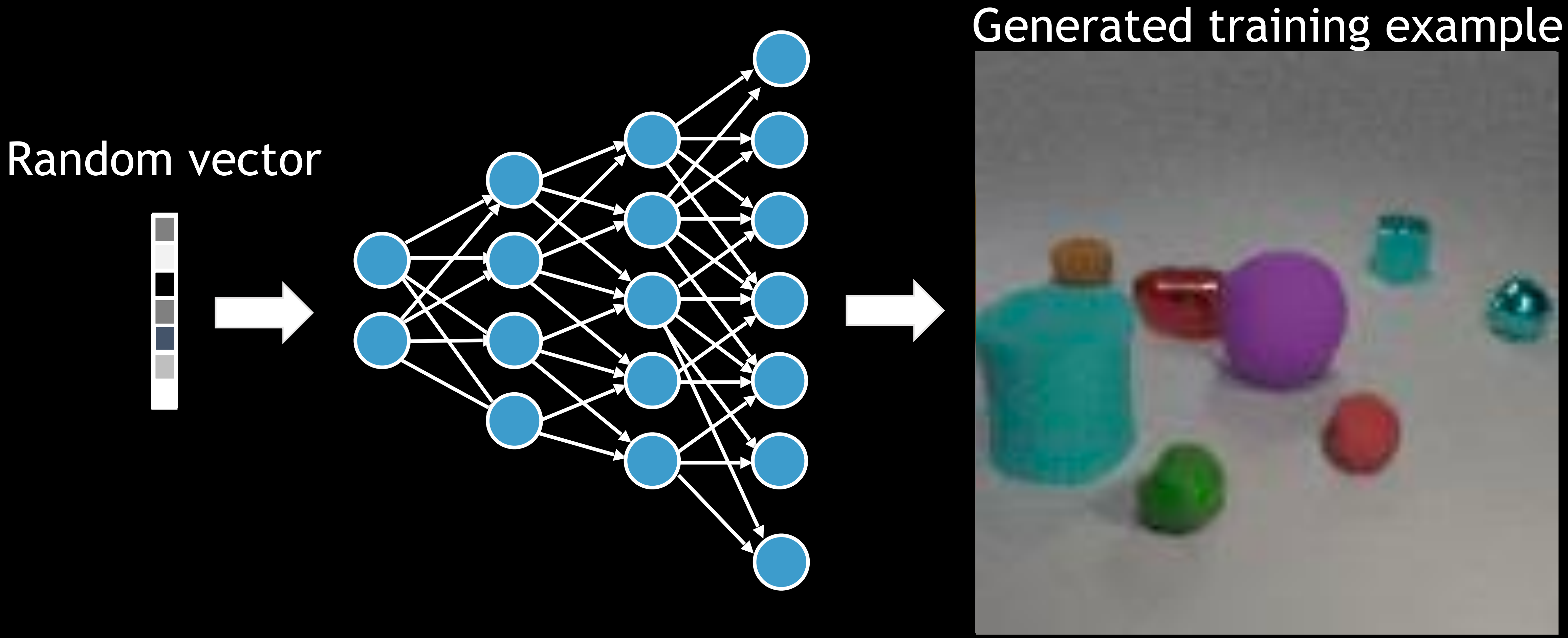


Generated training example

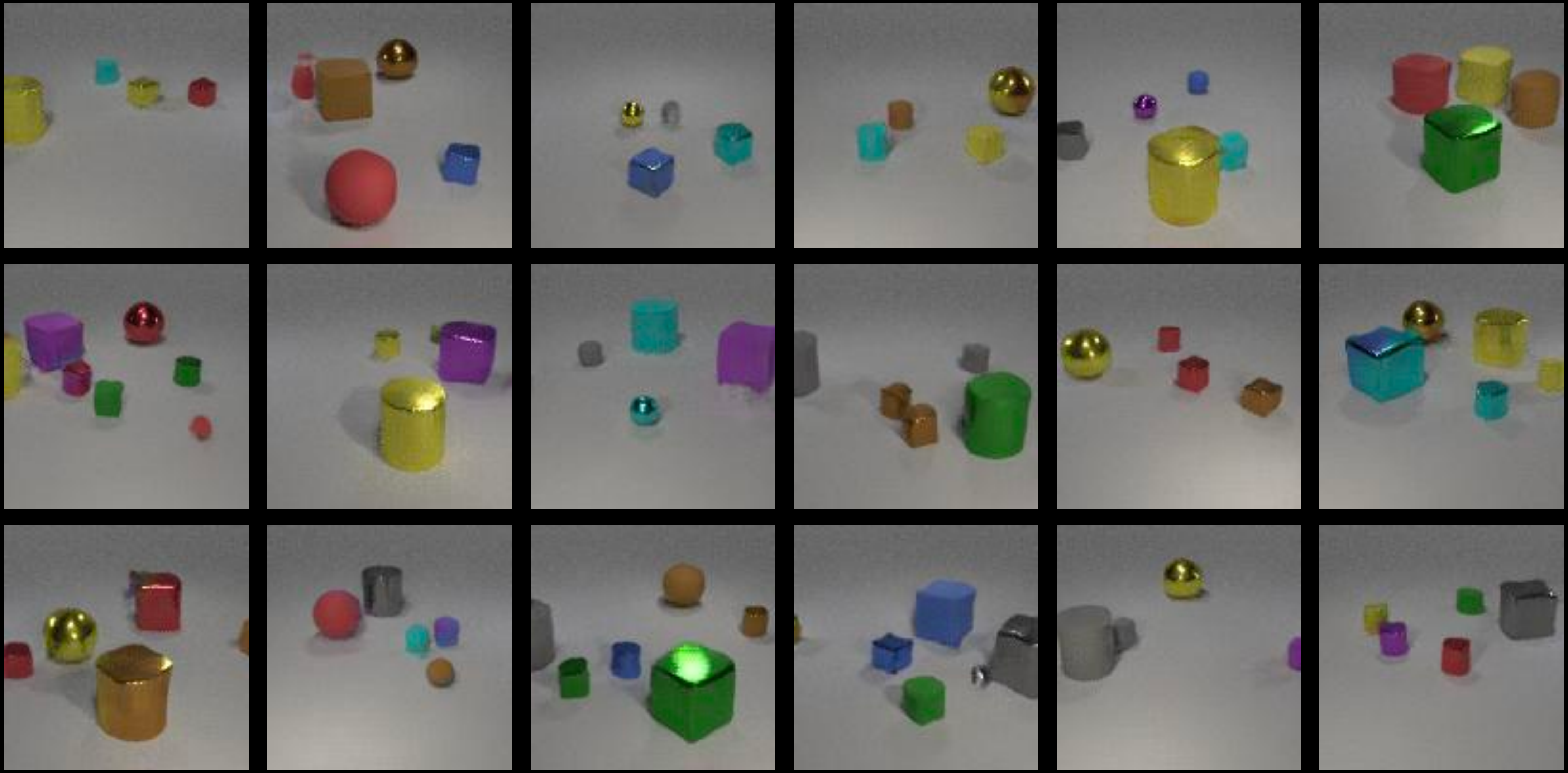
Generating our training data



Generating our training data

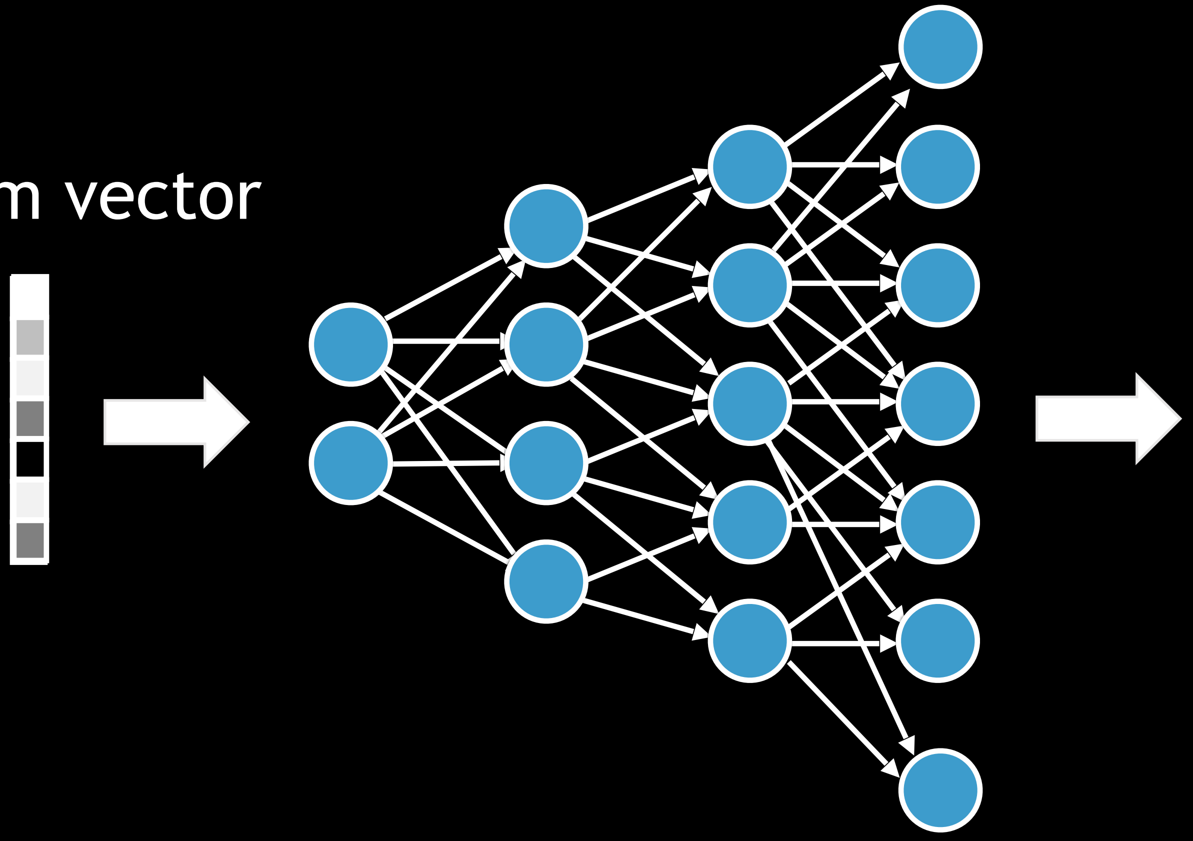


GAN-generated dataset

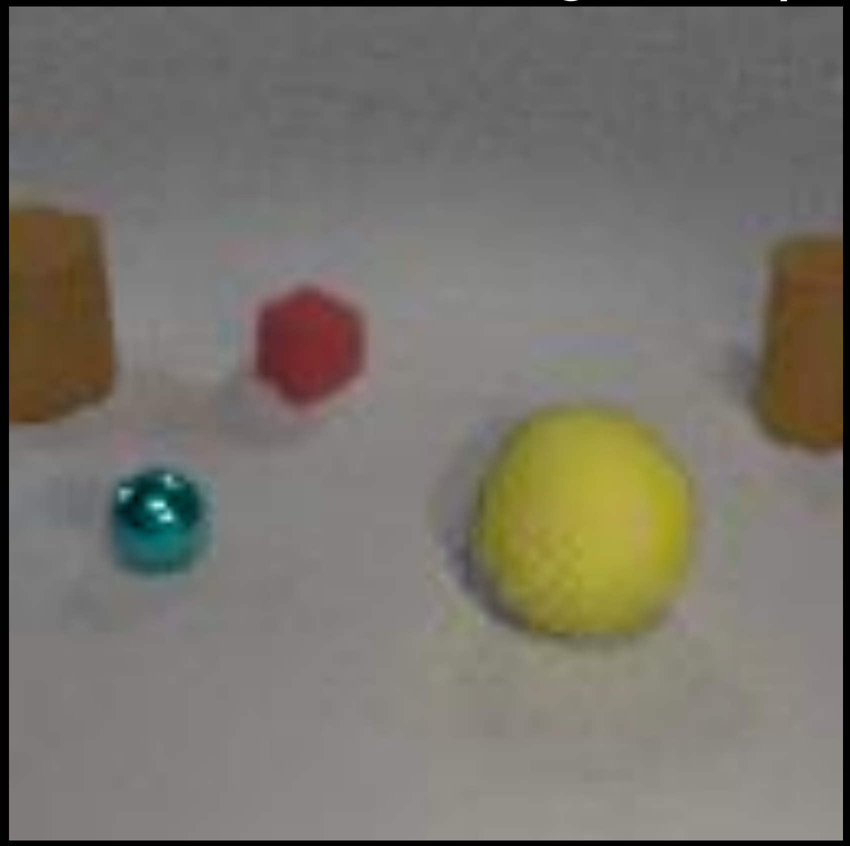


Editing training examples

Random vector

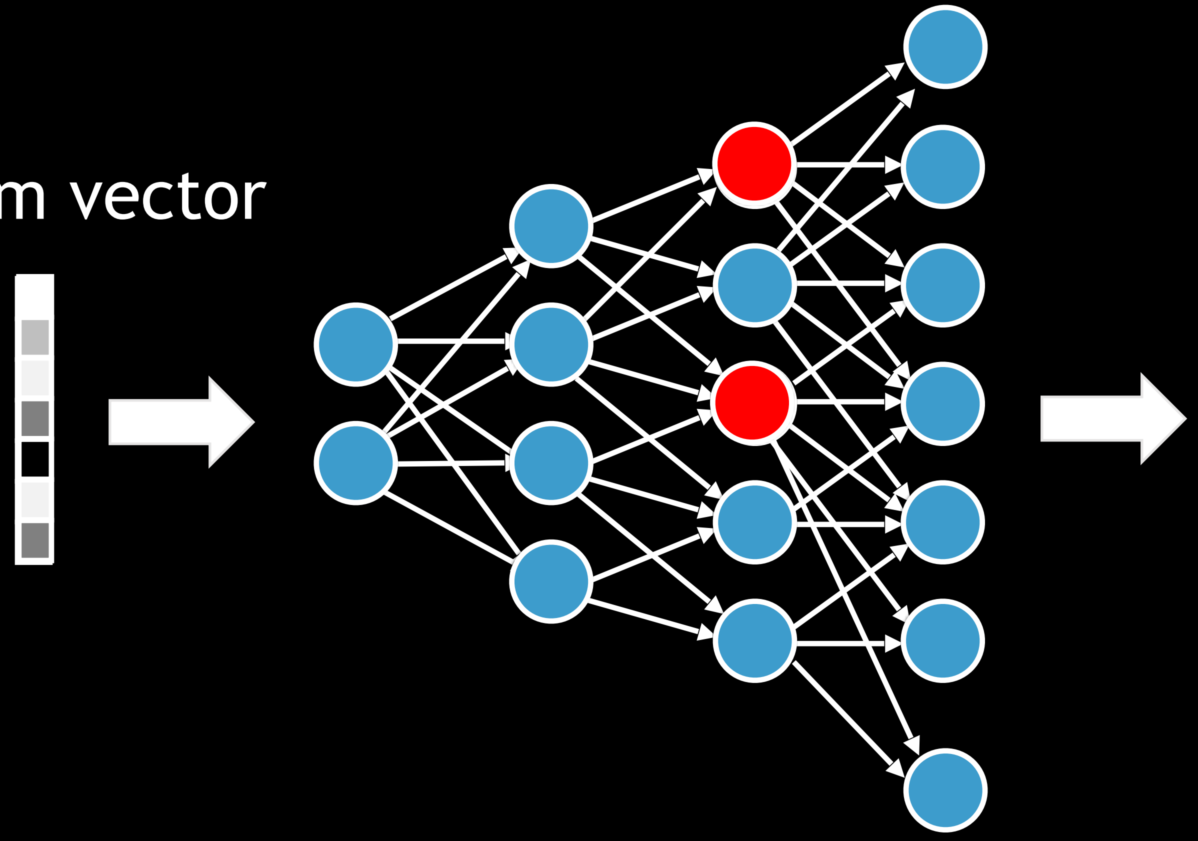


Generated training example



Editing training examples

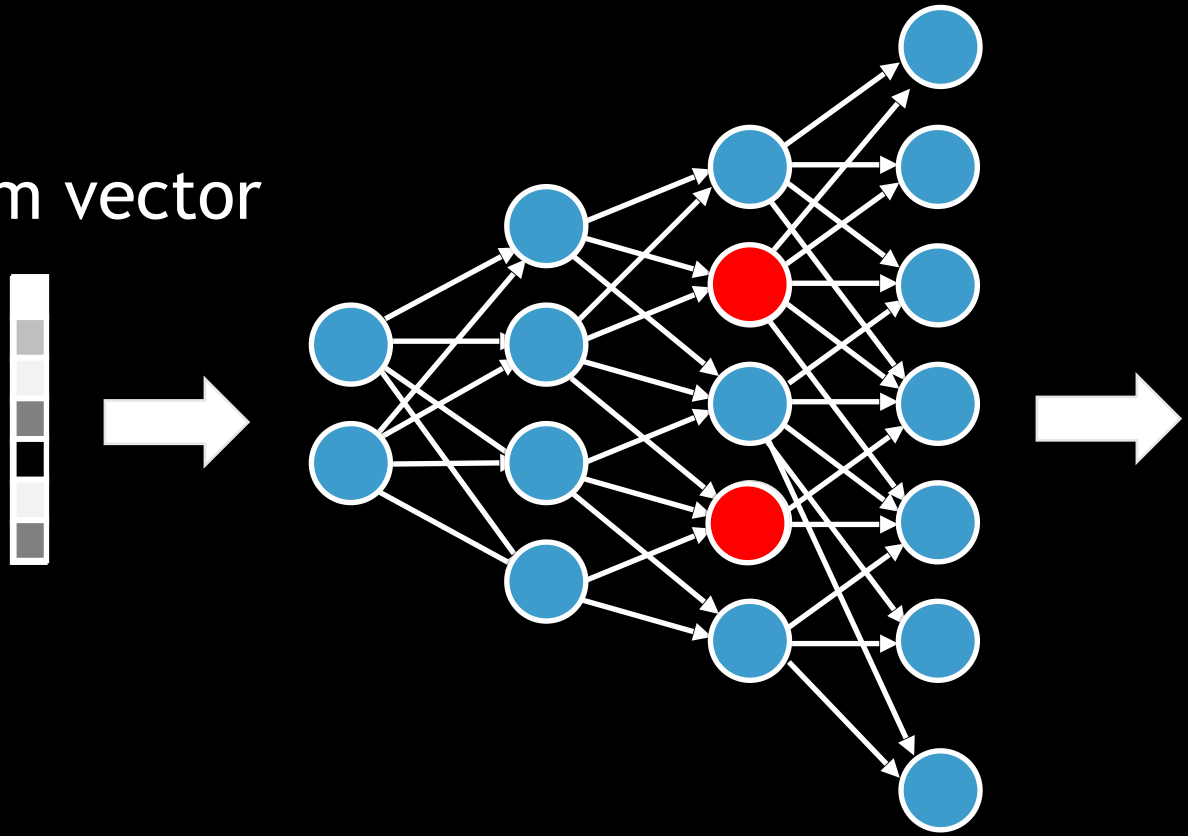
Random vector



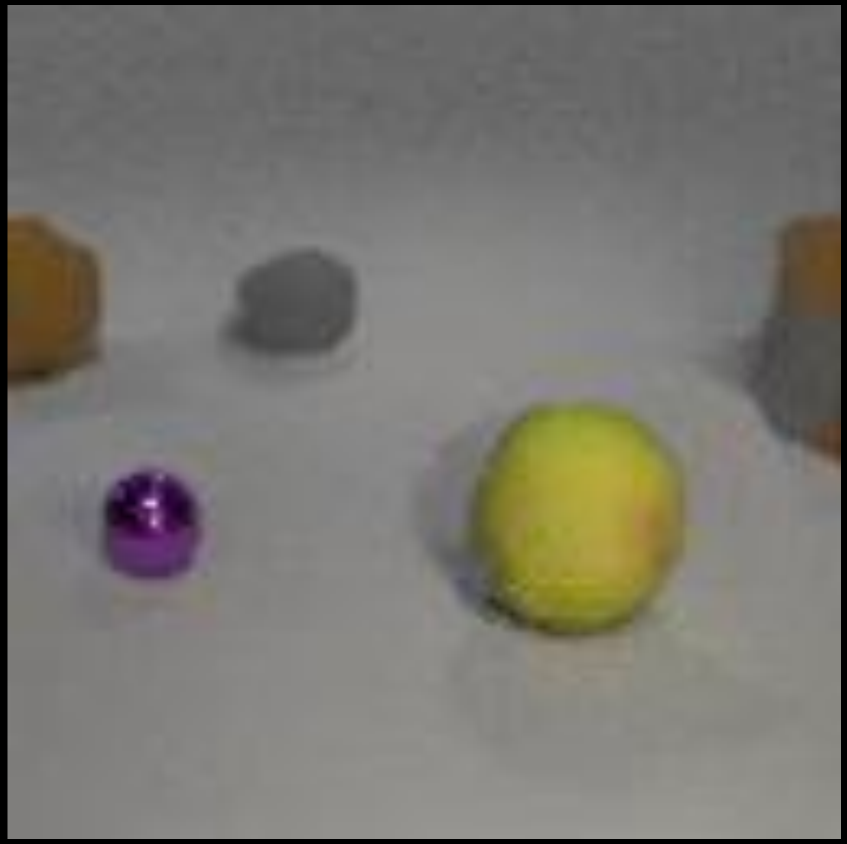
Generated training example

Editing training examples

Random vector

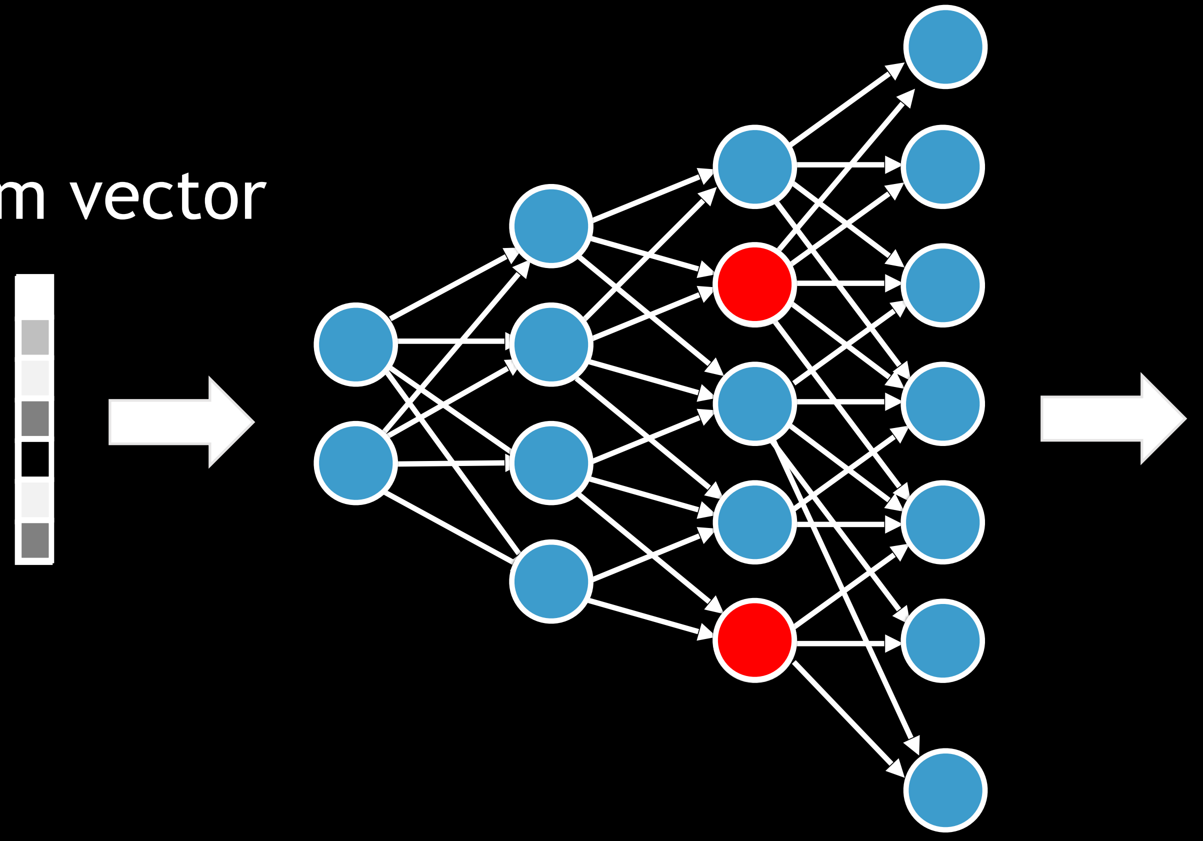


Generated training example

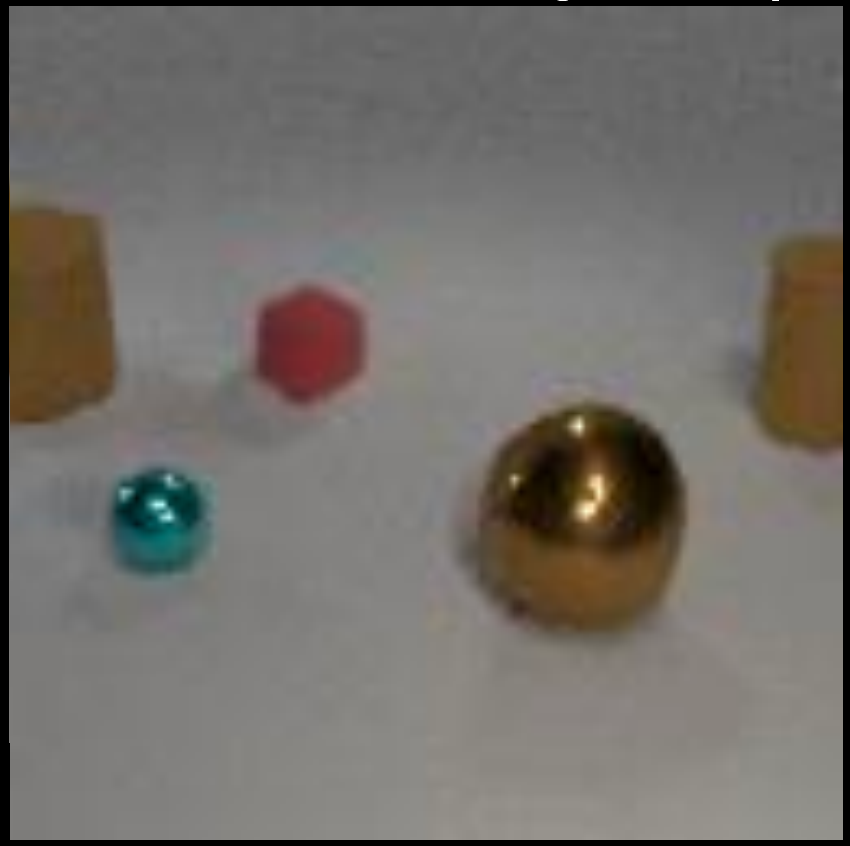


Editing training examples

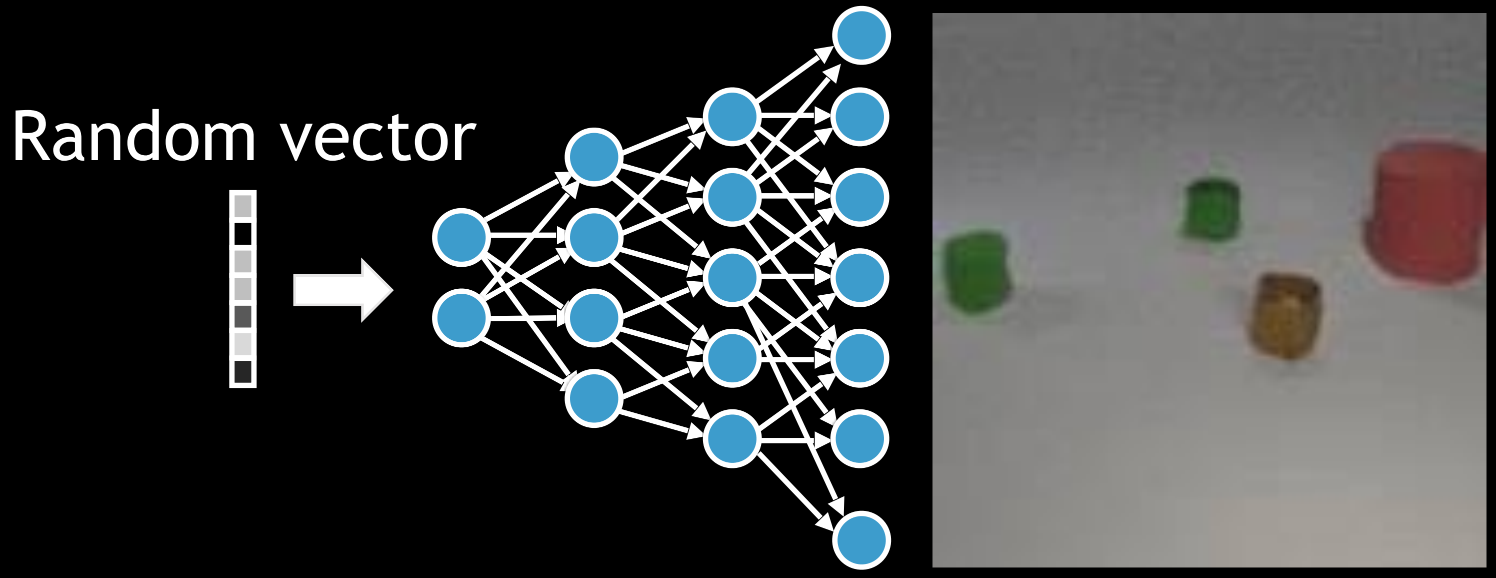
Random vector



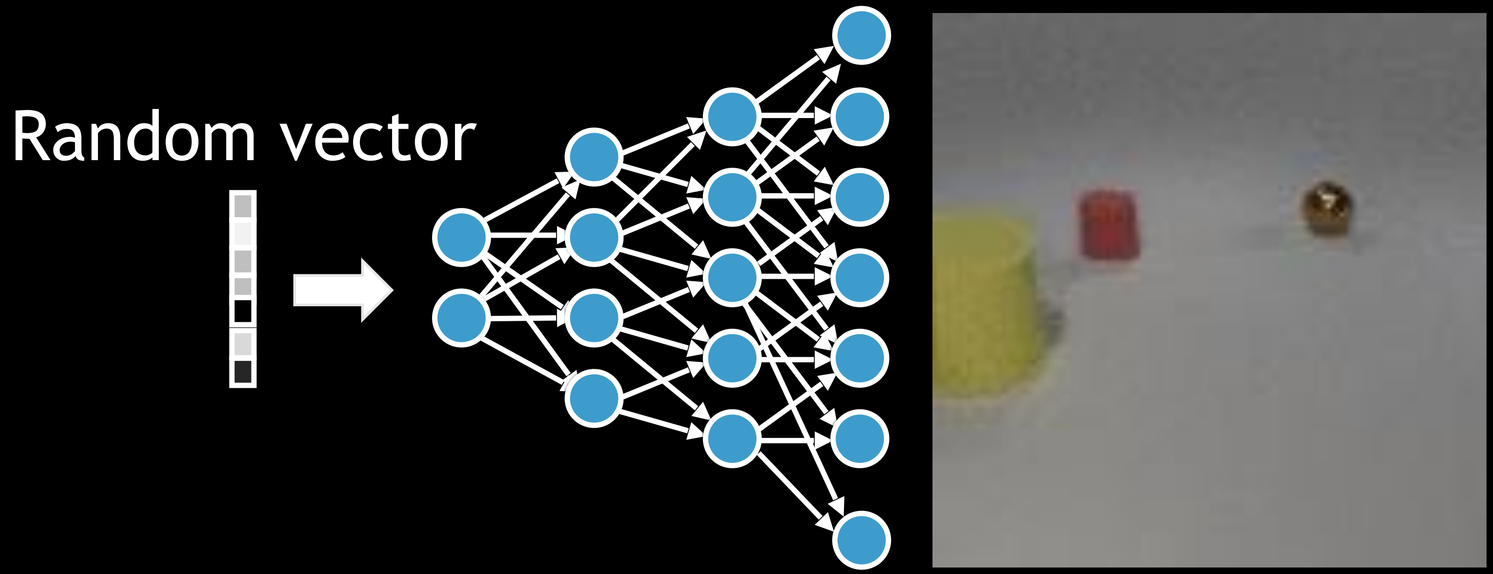
Generated training example



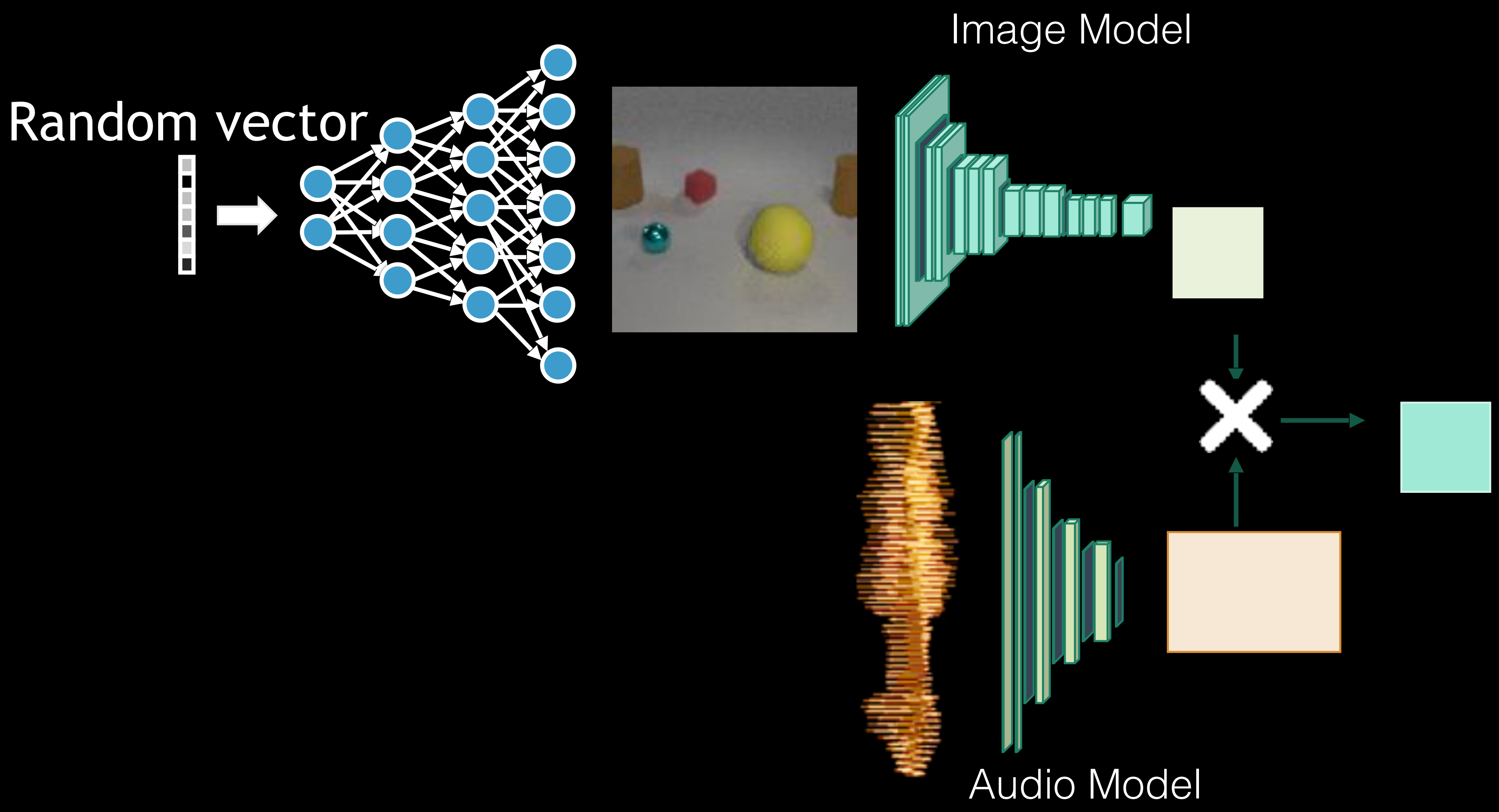
Learning language through GAN-generated images



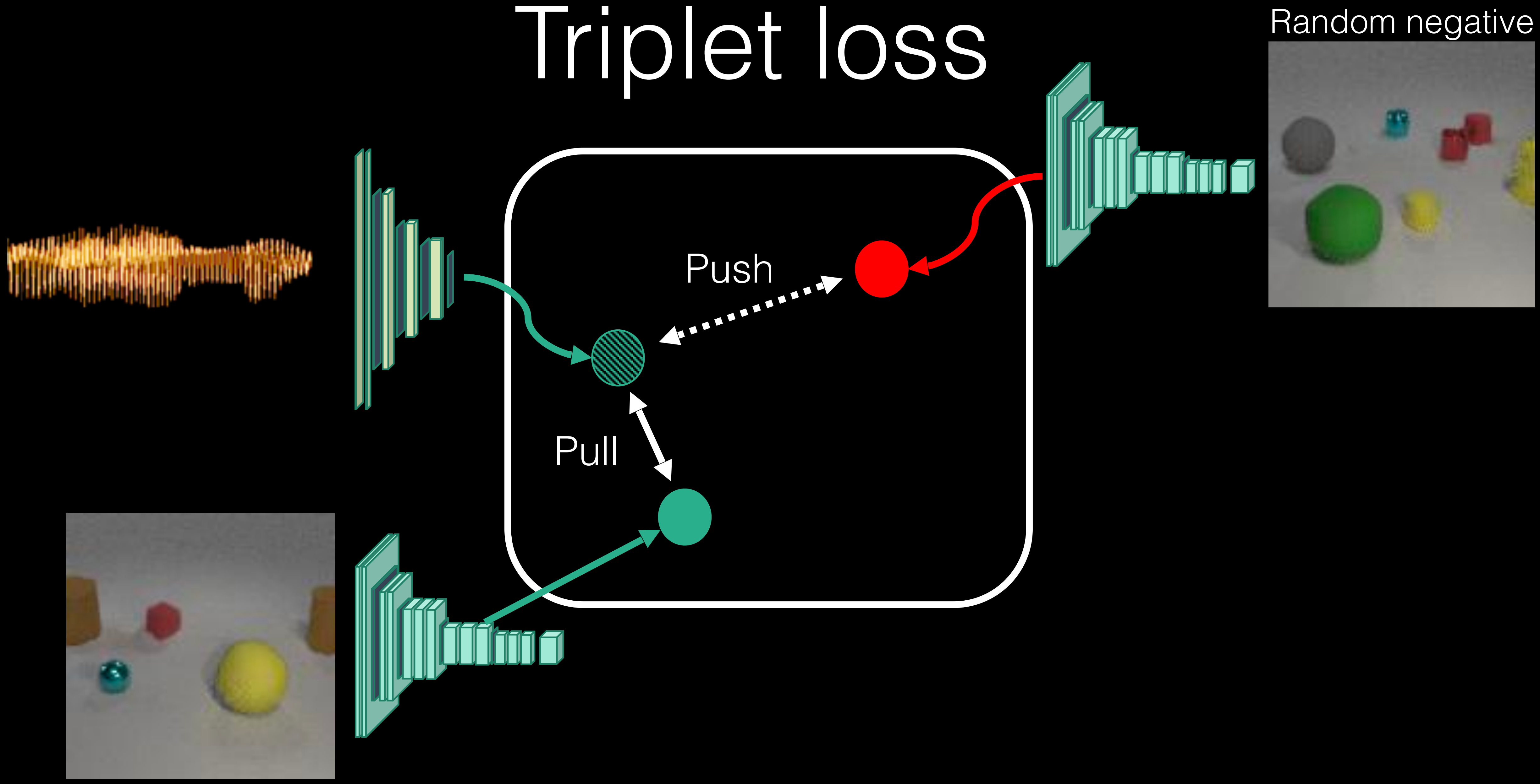
Learning language through GAN-generated images



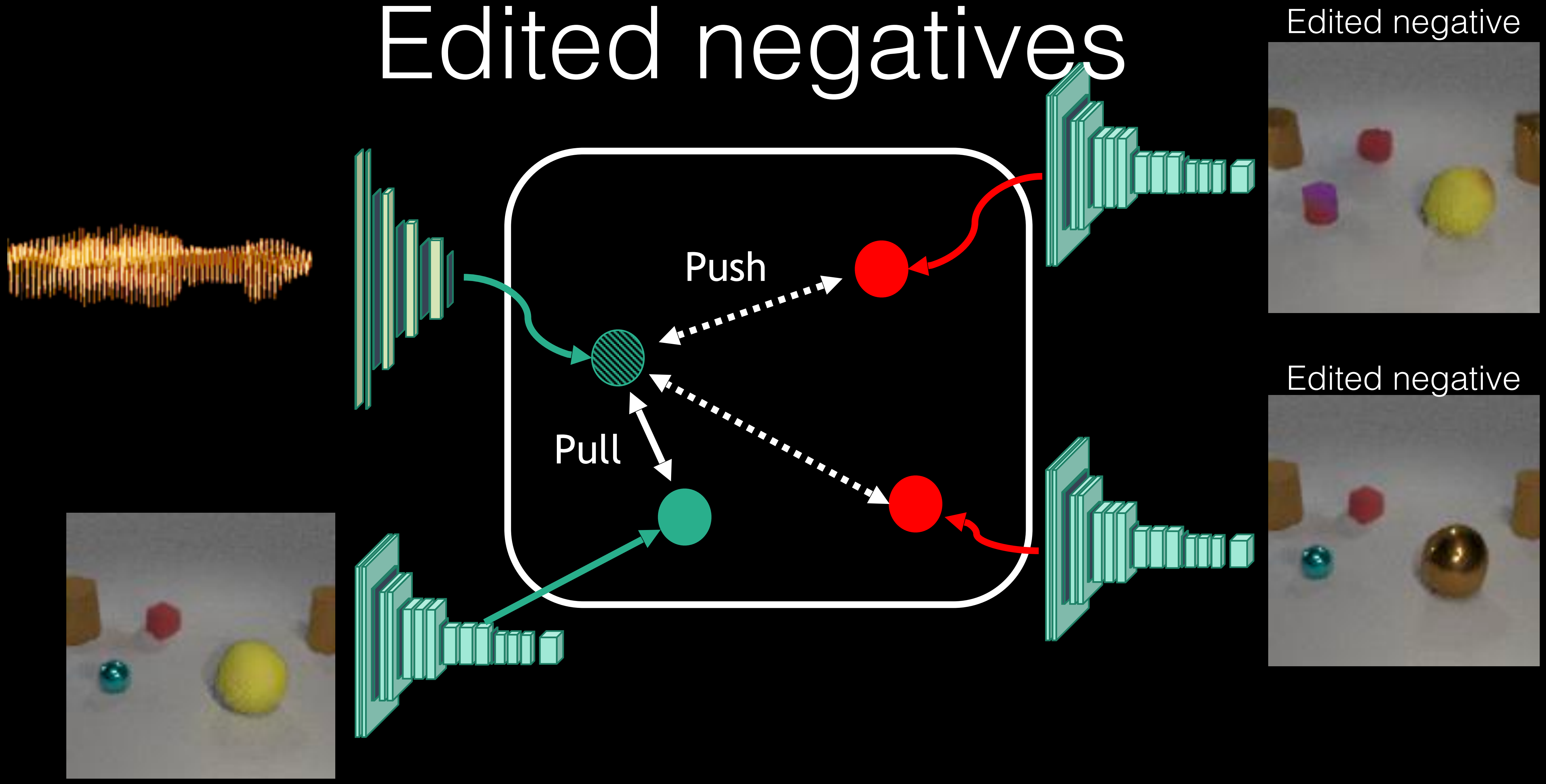
Audio-visual similarity model



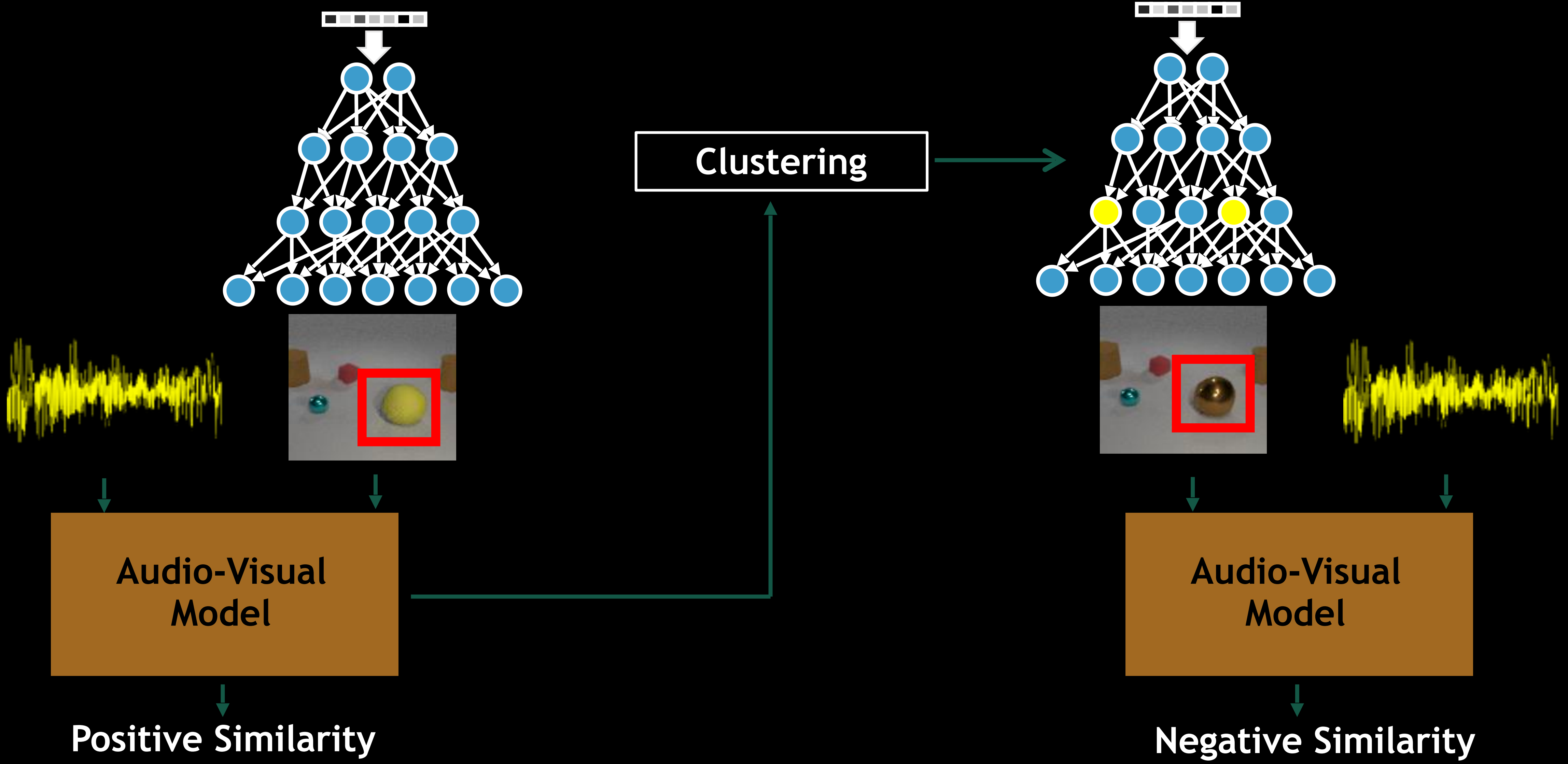
Triplet loss



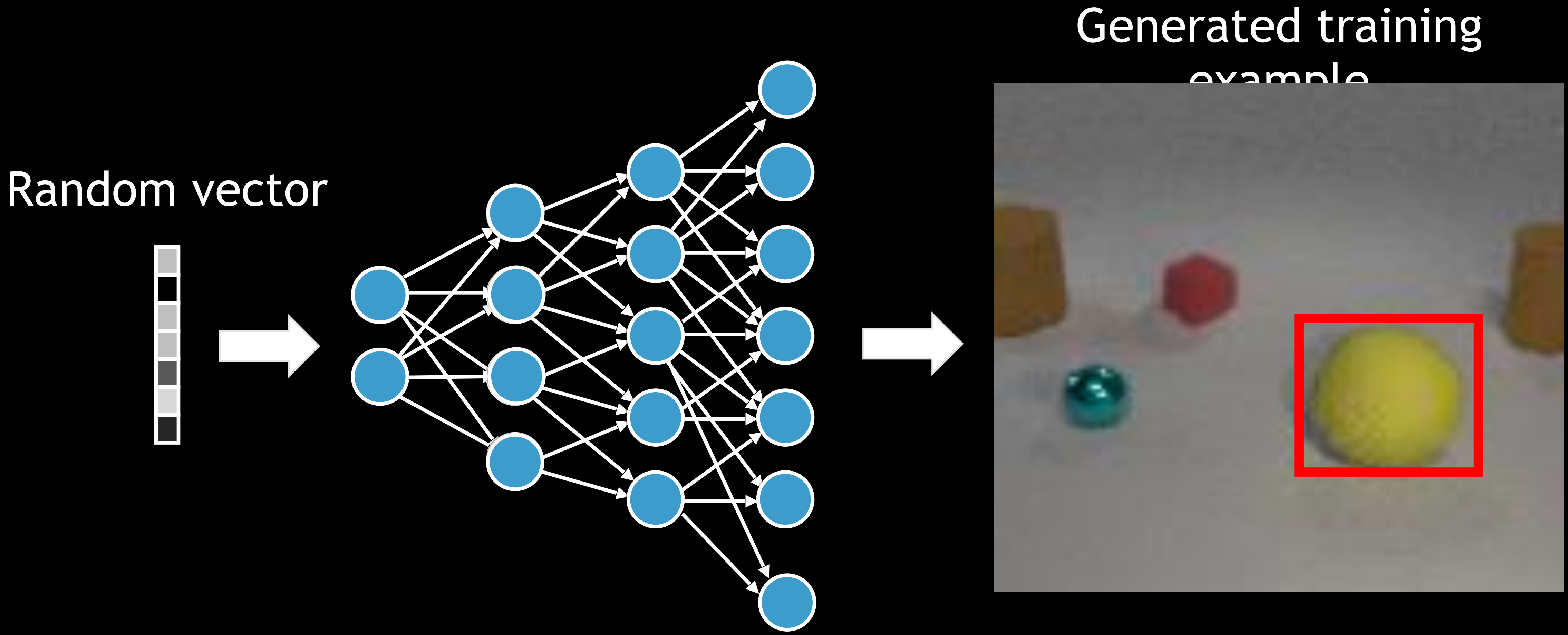
Edited negatives



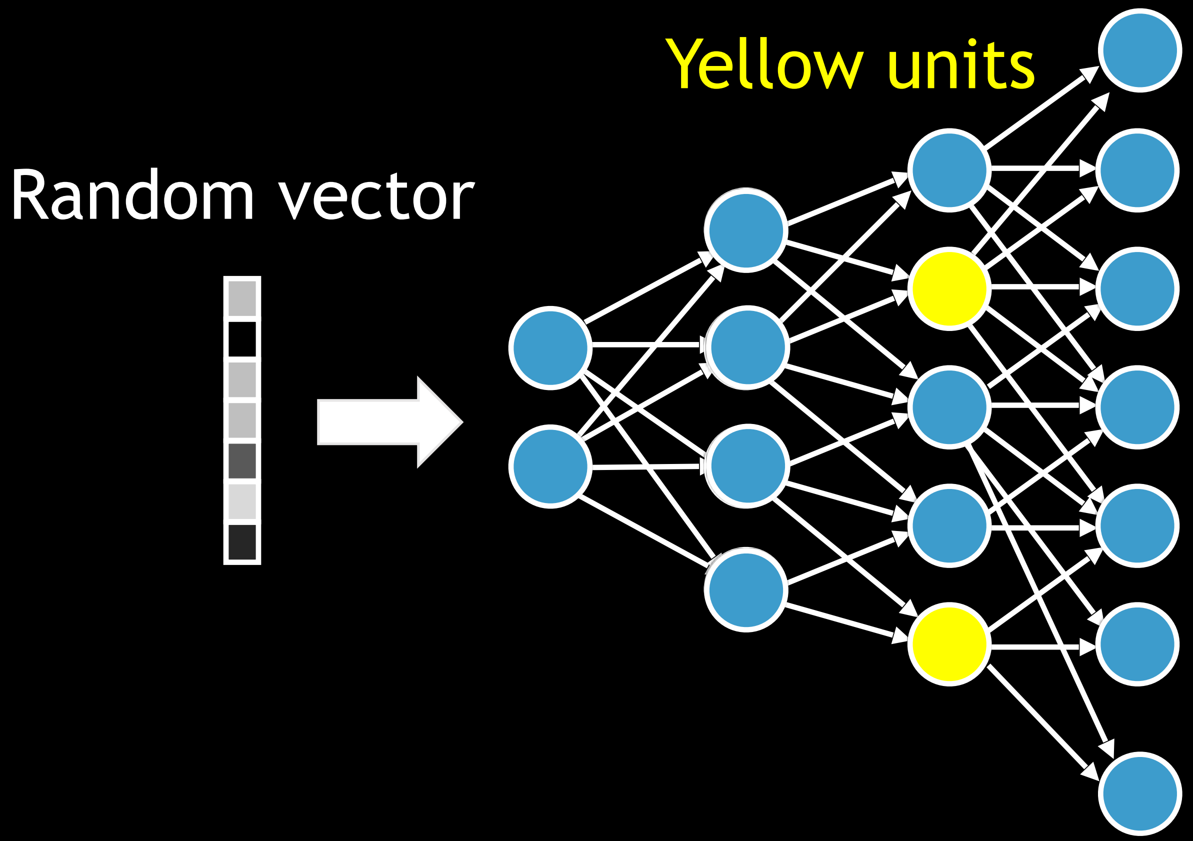
System overview

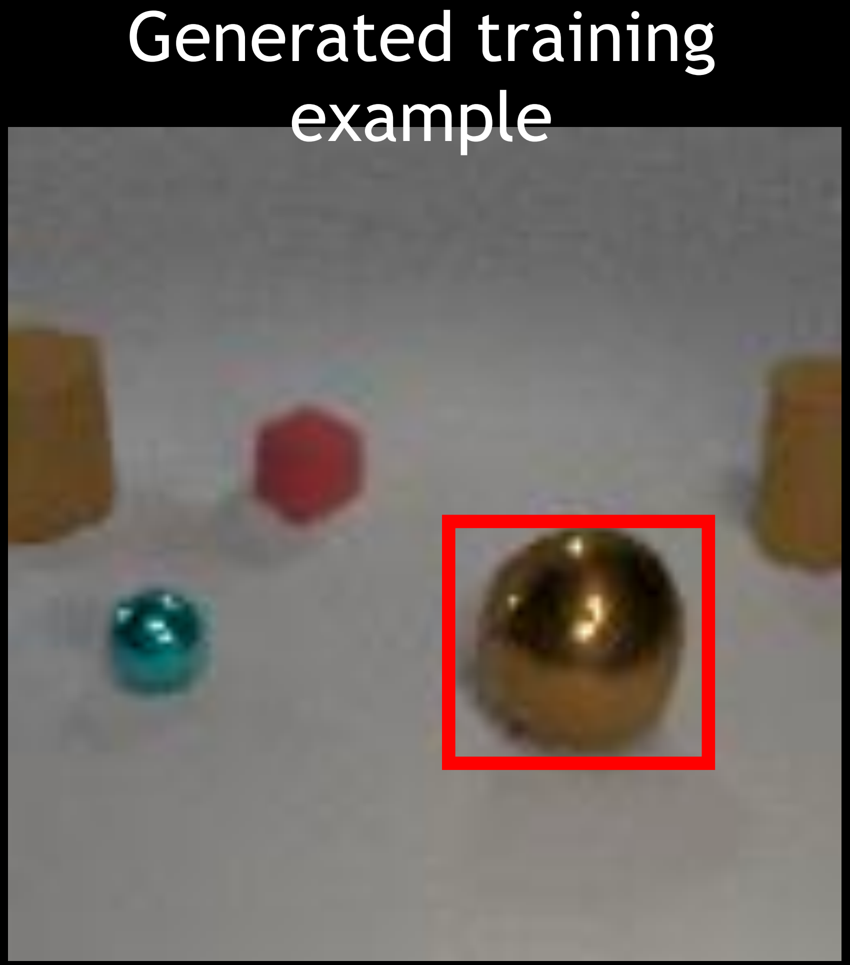


Removing concepts from images

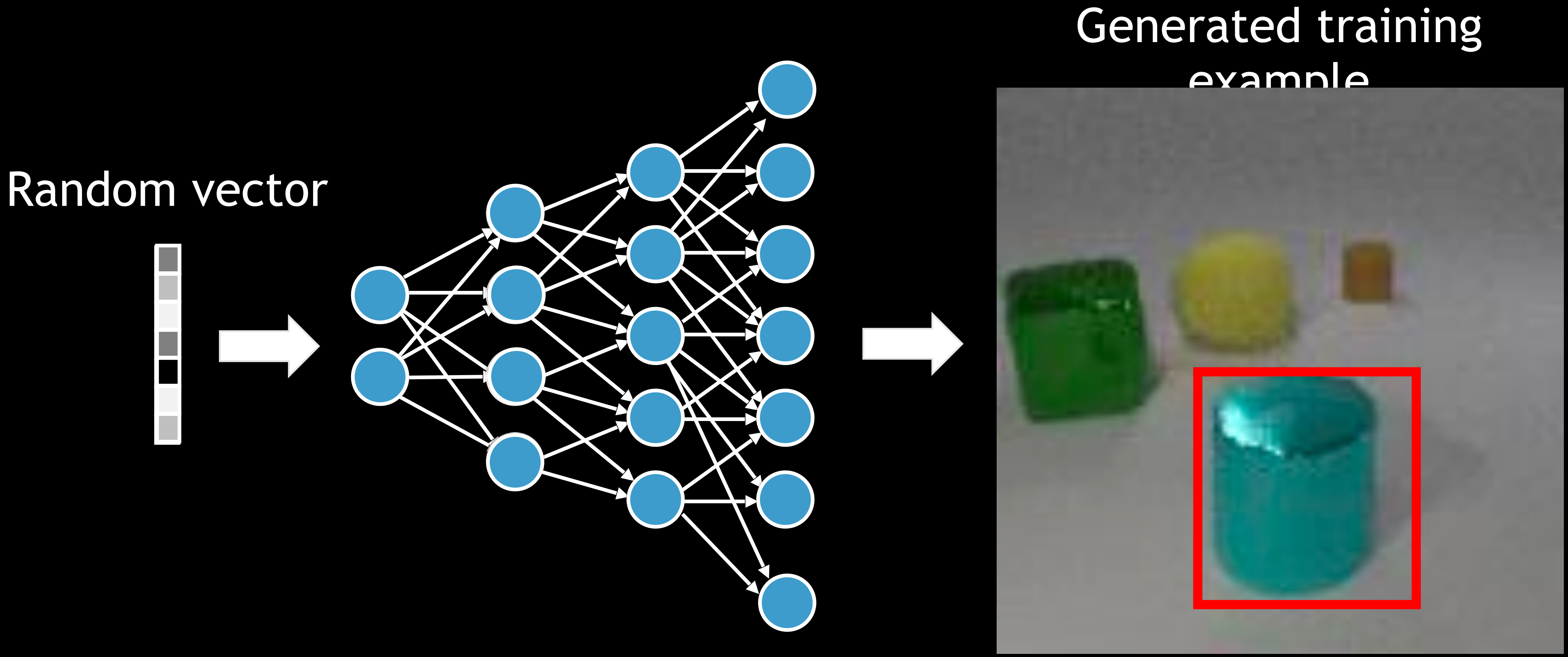


Removing concepts from images

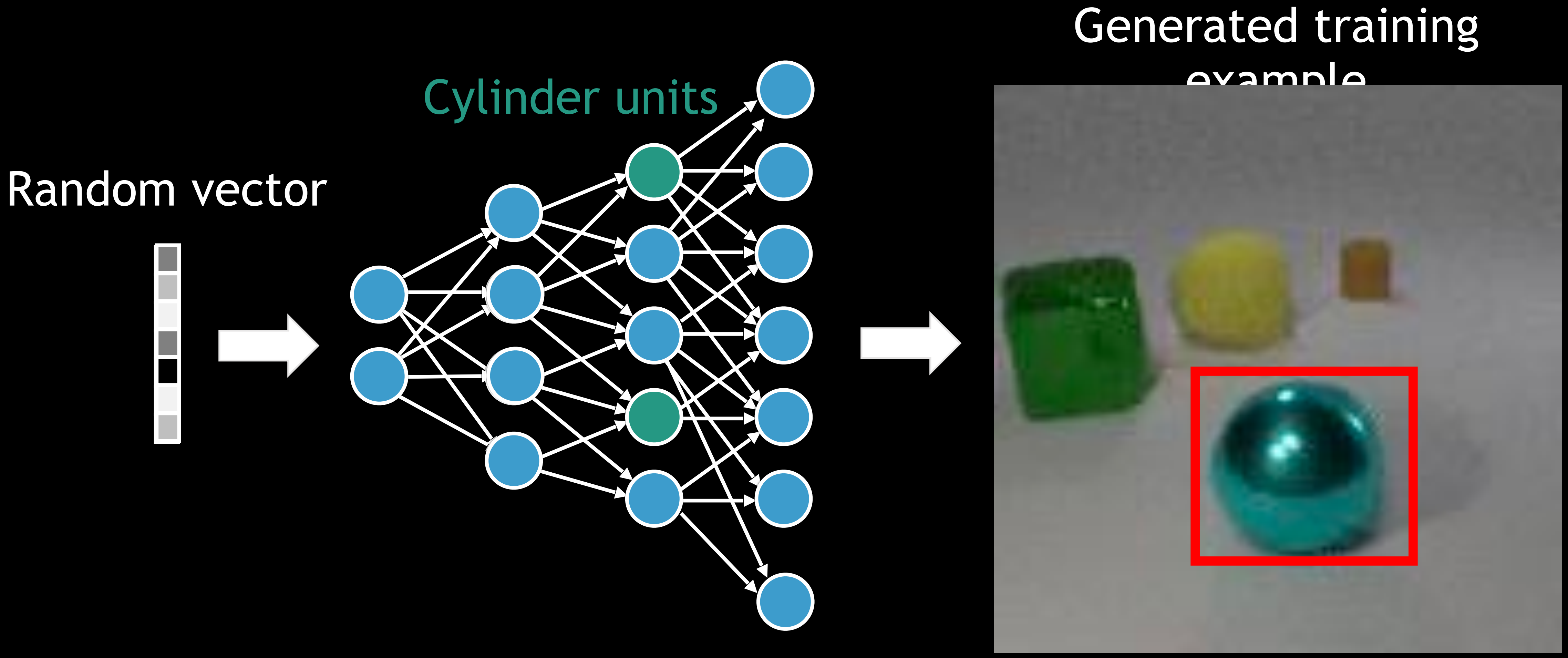




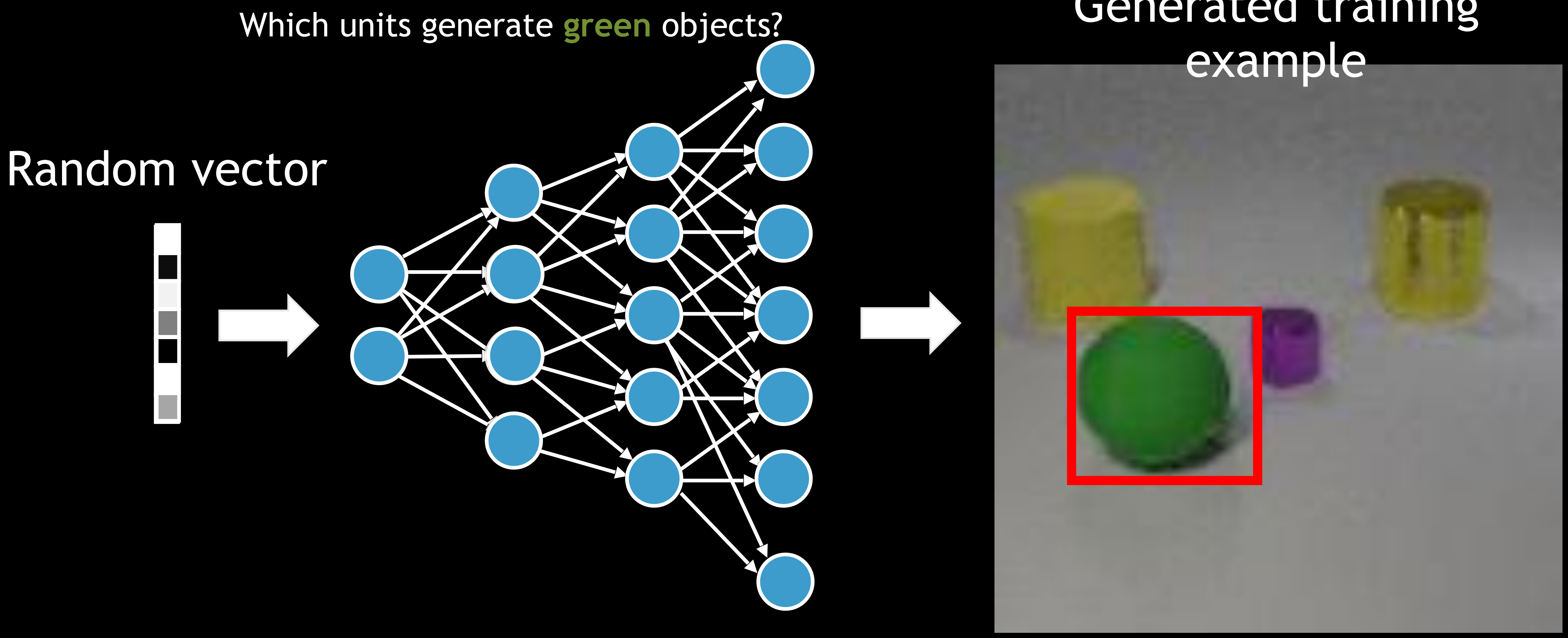
Removing concepts from images



Removing concepts from images

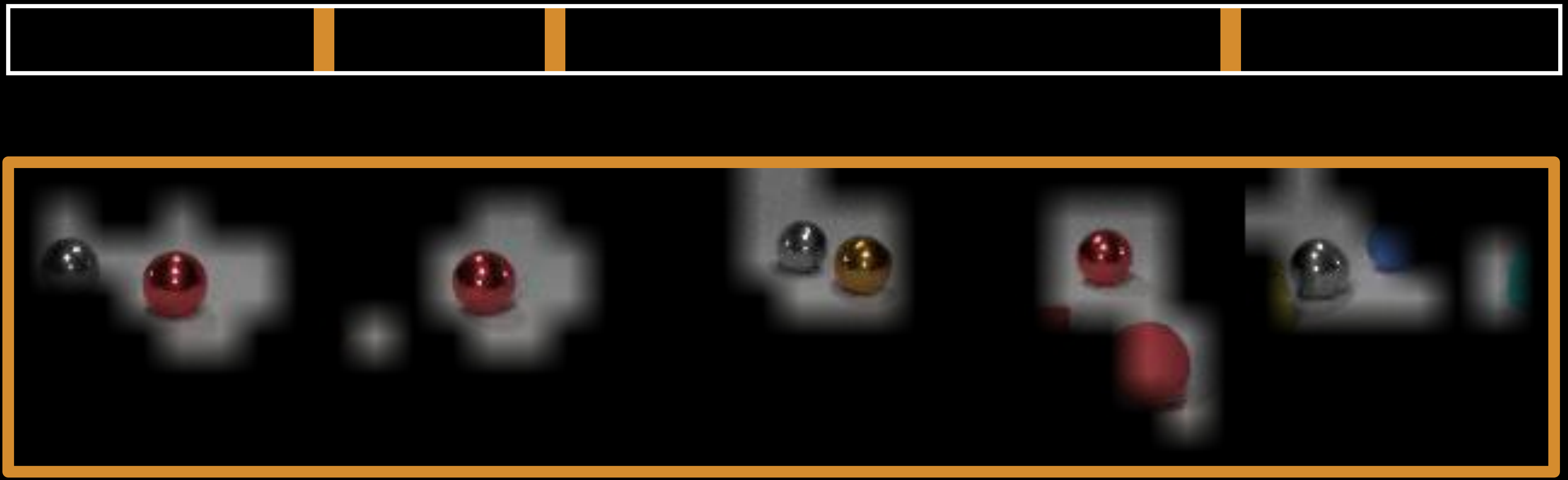


Learning which units to modify



Clustering embedding features

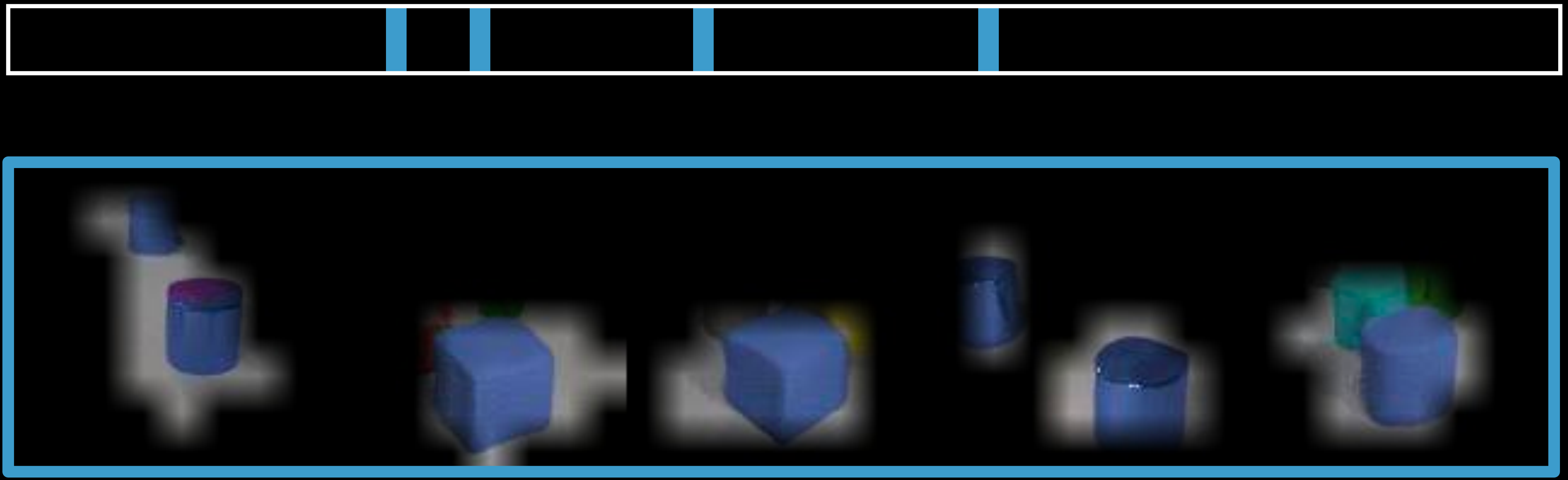
Compute co-occurrence of embedding dimensions and group them in clusters.



Ball

Clustering embedding features

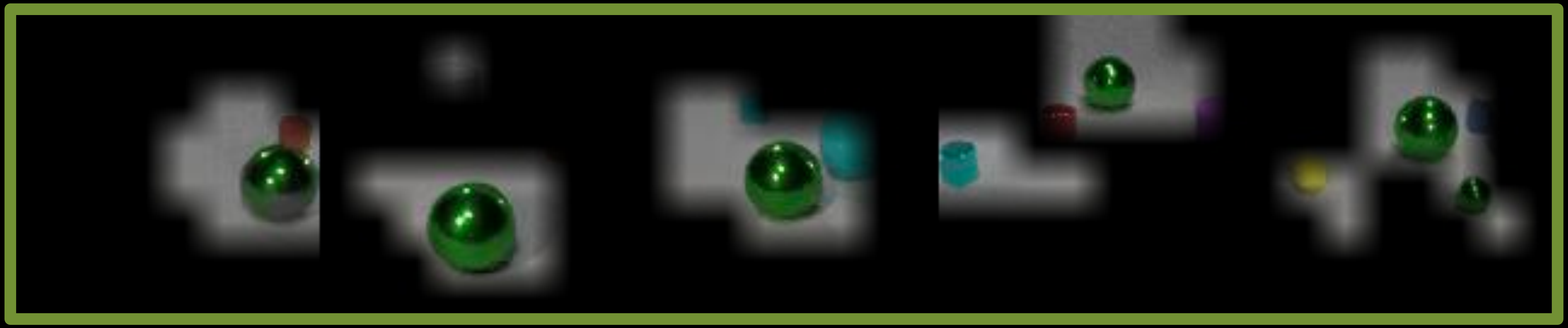
Compute co-occurrence of embedding dimensions and group them in clusters.



Blue

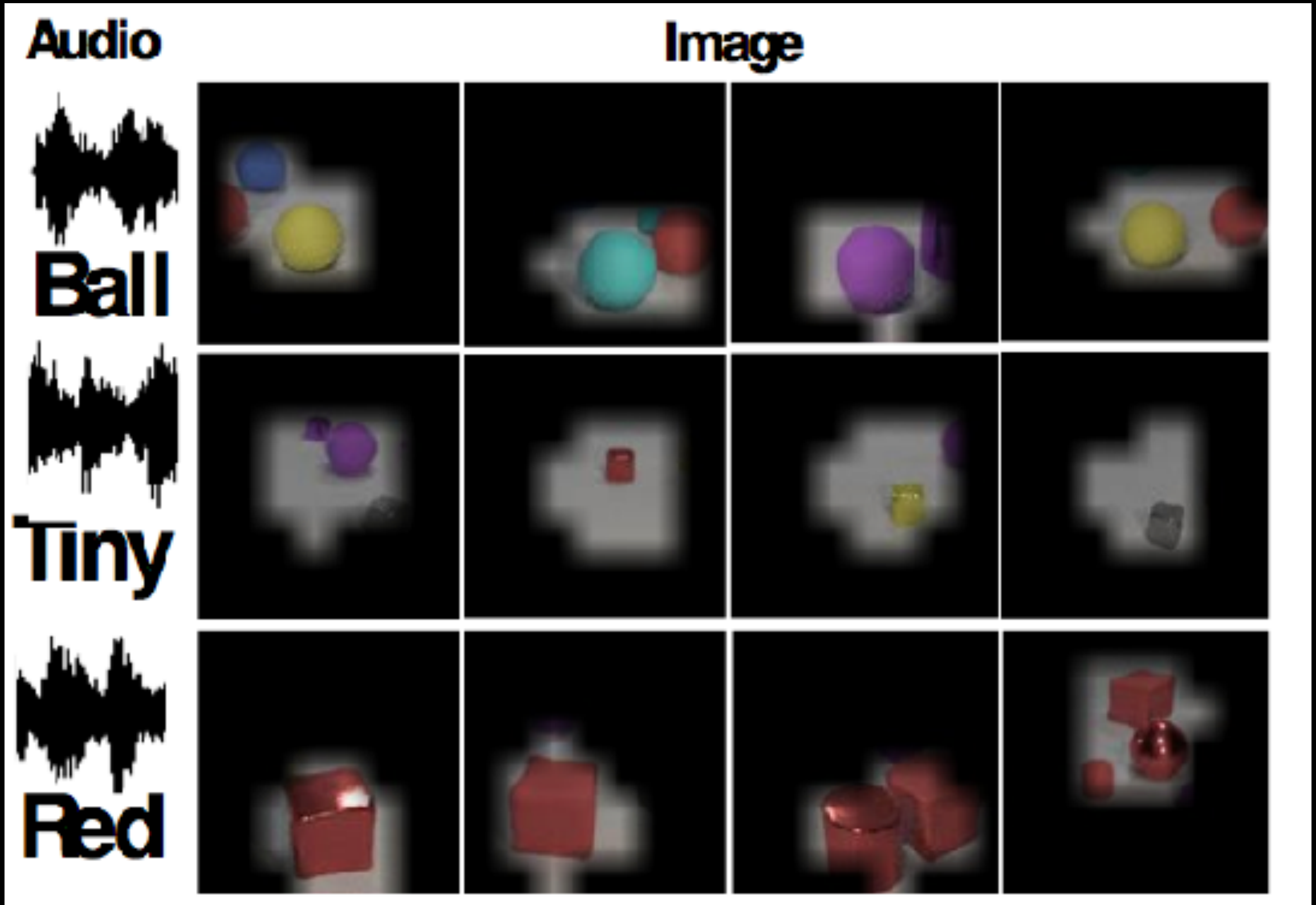
Clustering embedding features

Compute co-occurrence of embedding dimensions and group them in clusters.

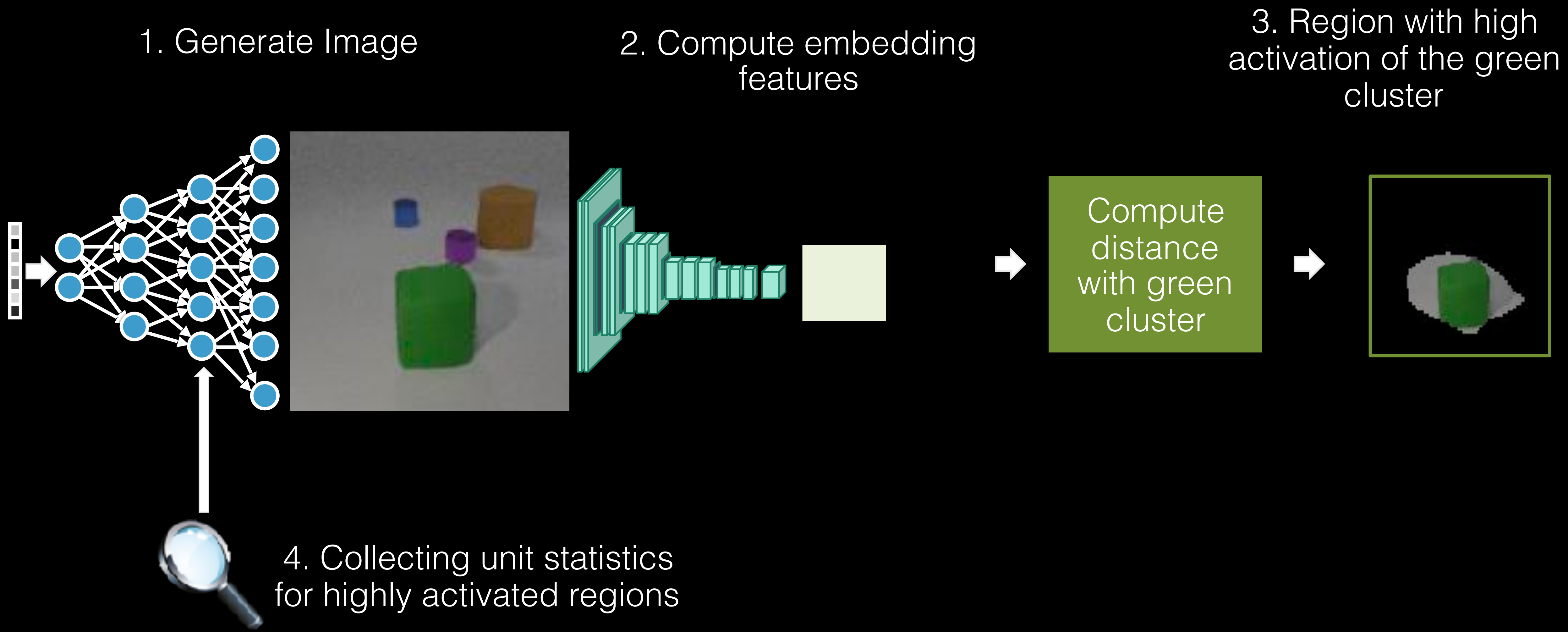


Green

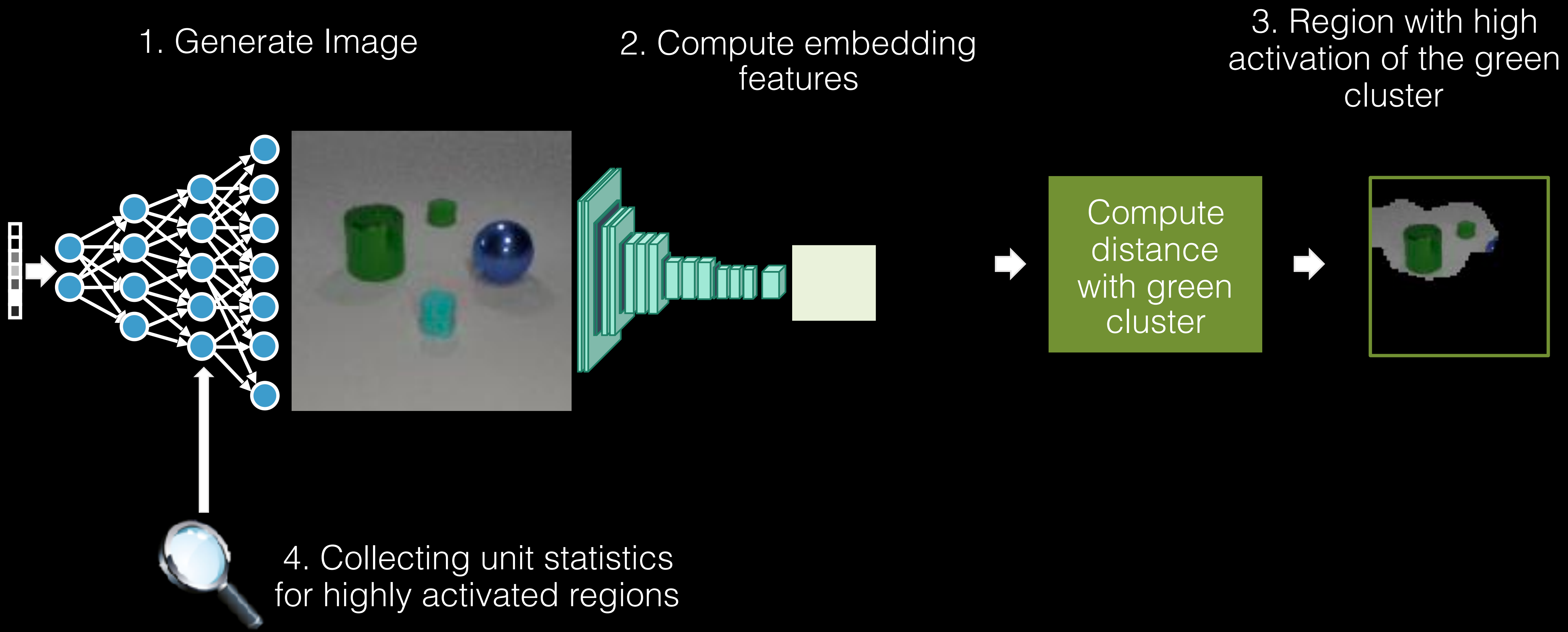
Examples of clustered concepts



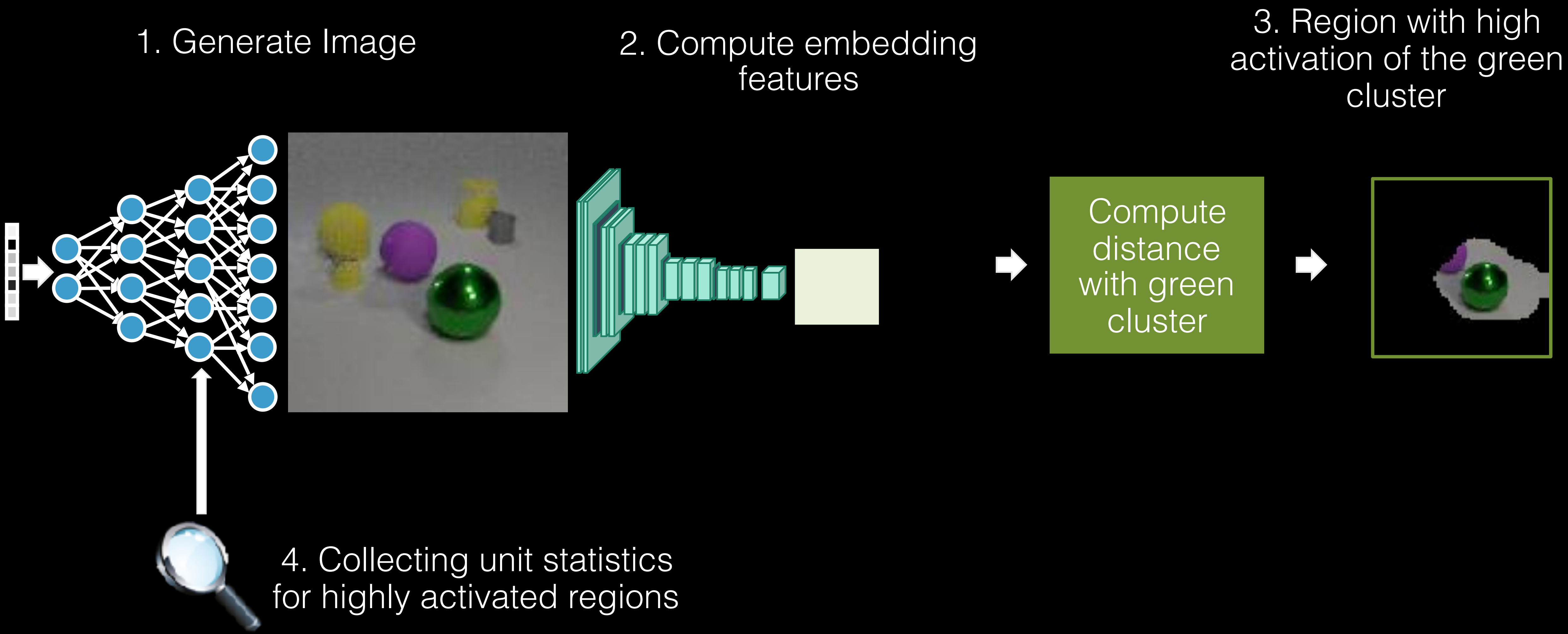
Learning which units to modify



Learning which units to modify

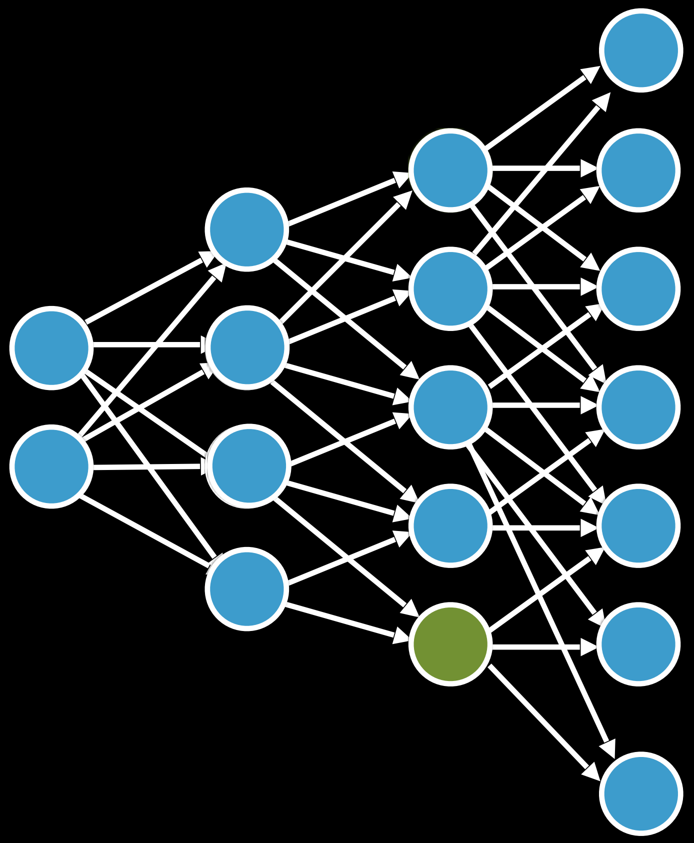


Learning which units to modify

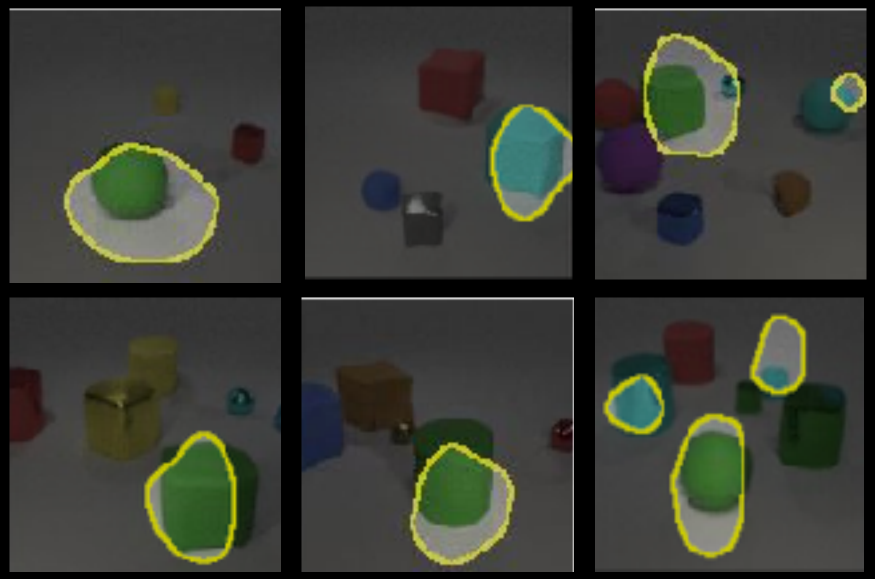


Learning which units to modify

Which units generate **green** objects?

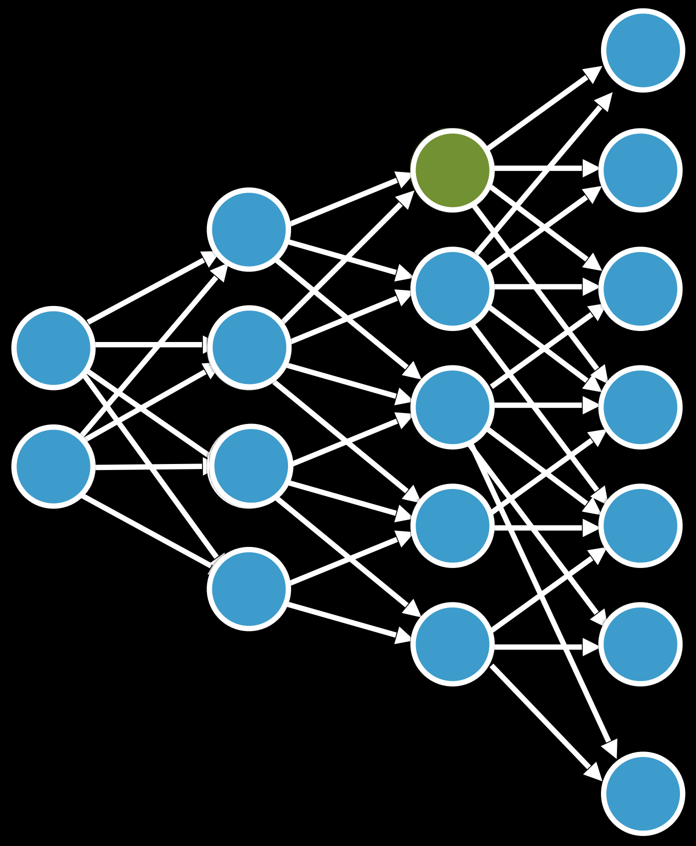


Top activated regions

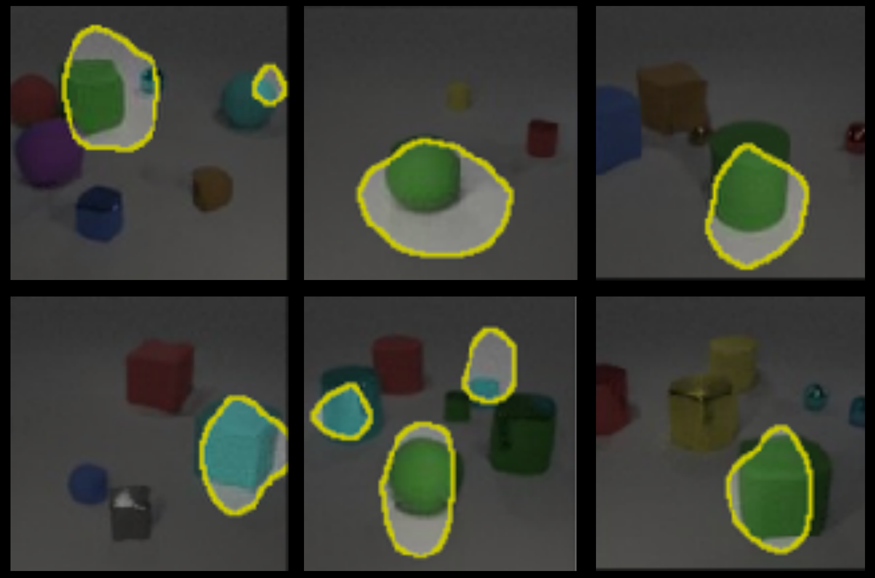


Learning which units to modify

Which units generate **green** objects?



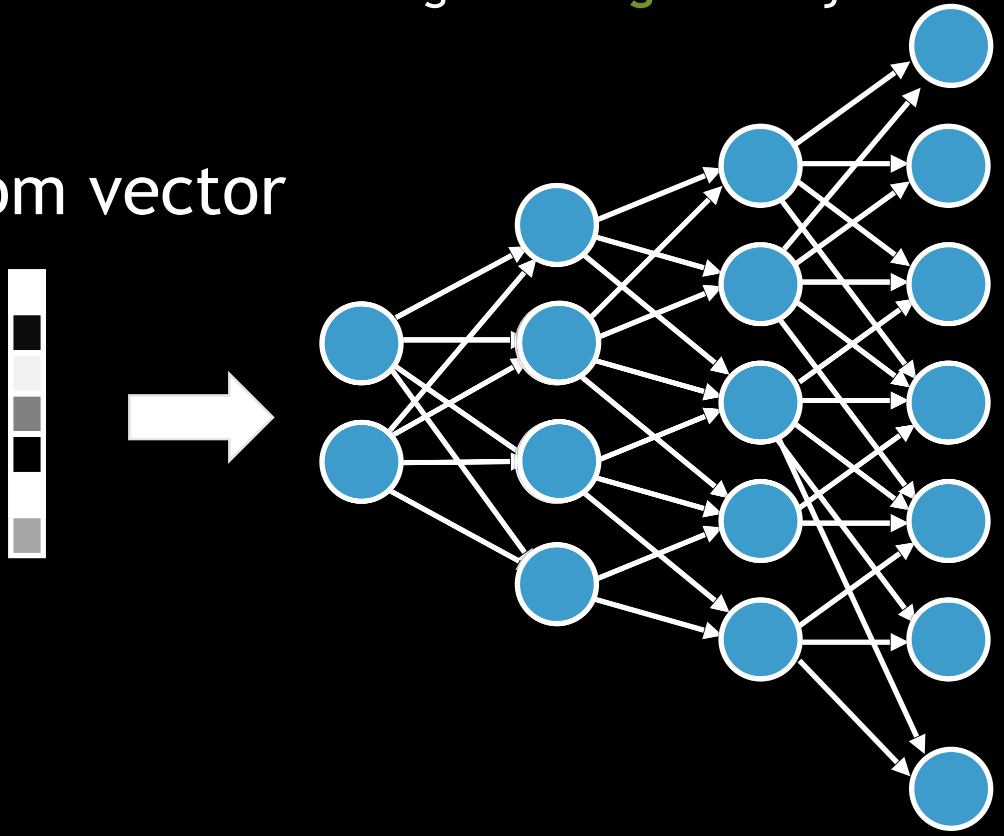
Top activated regions



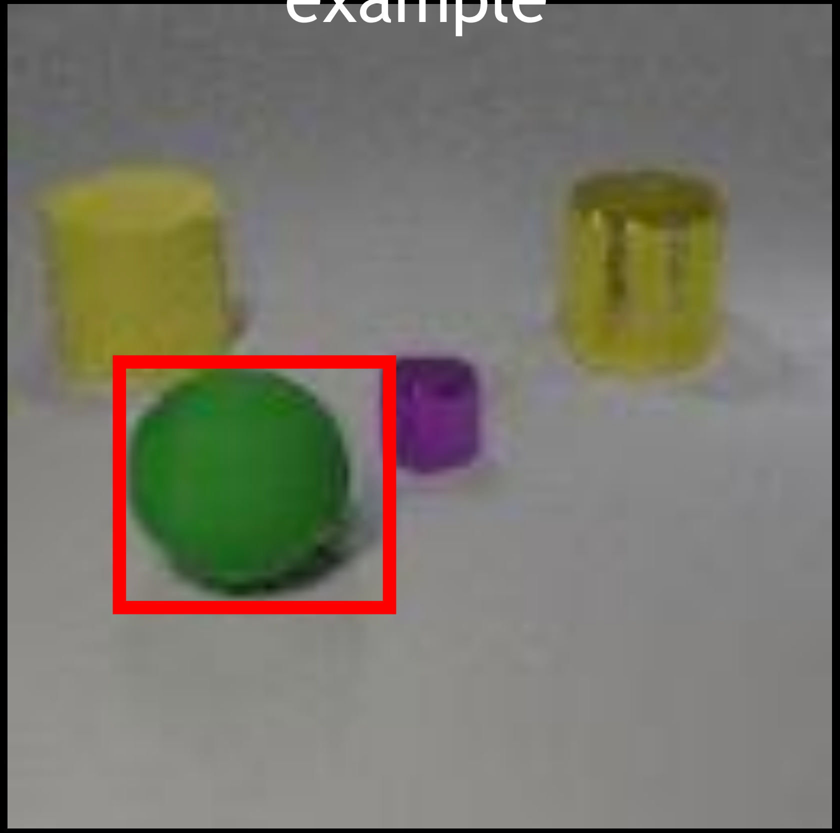
Learning which units to modify

Random vector

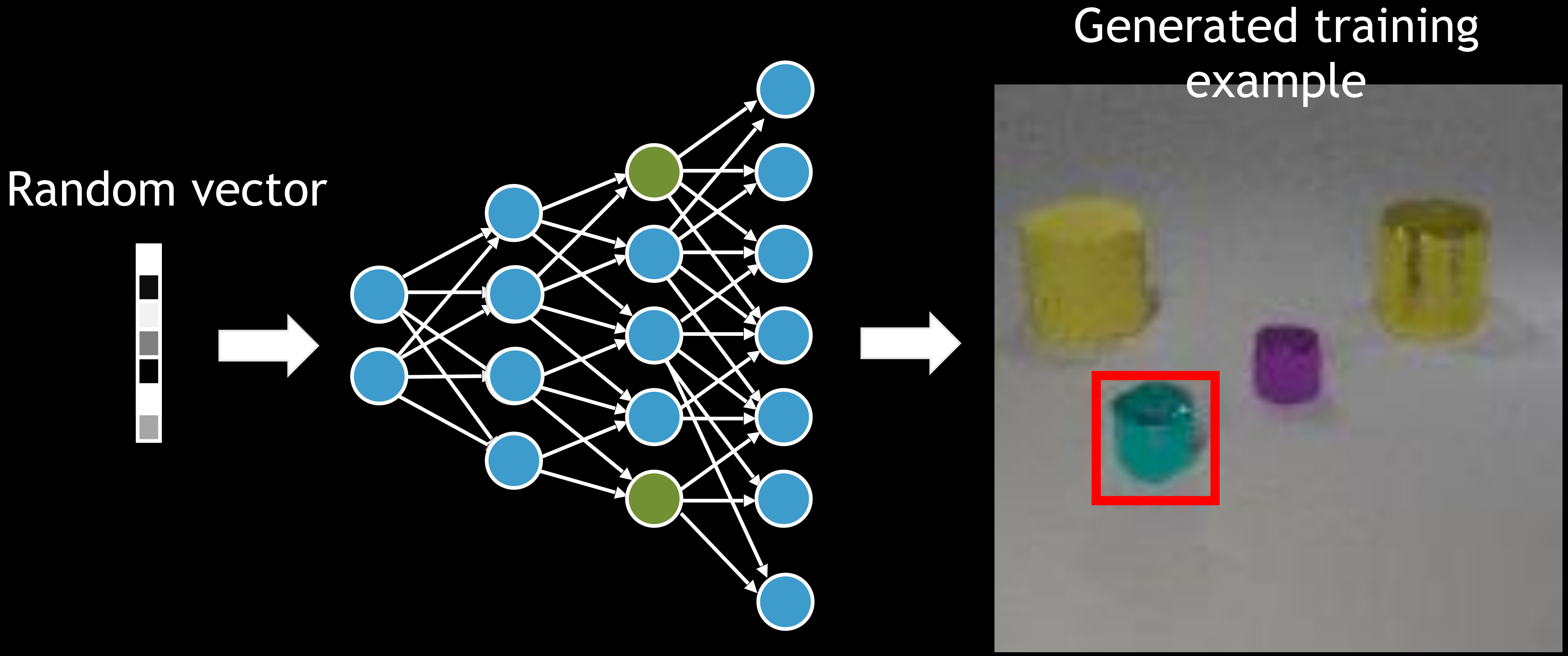
Which units generate **green** objects?



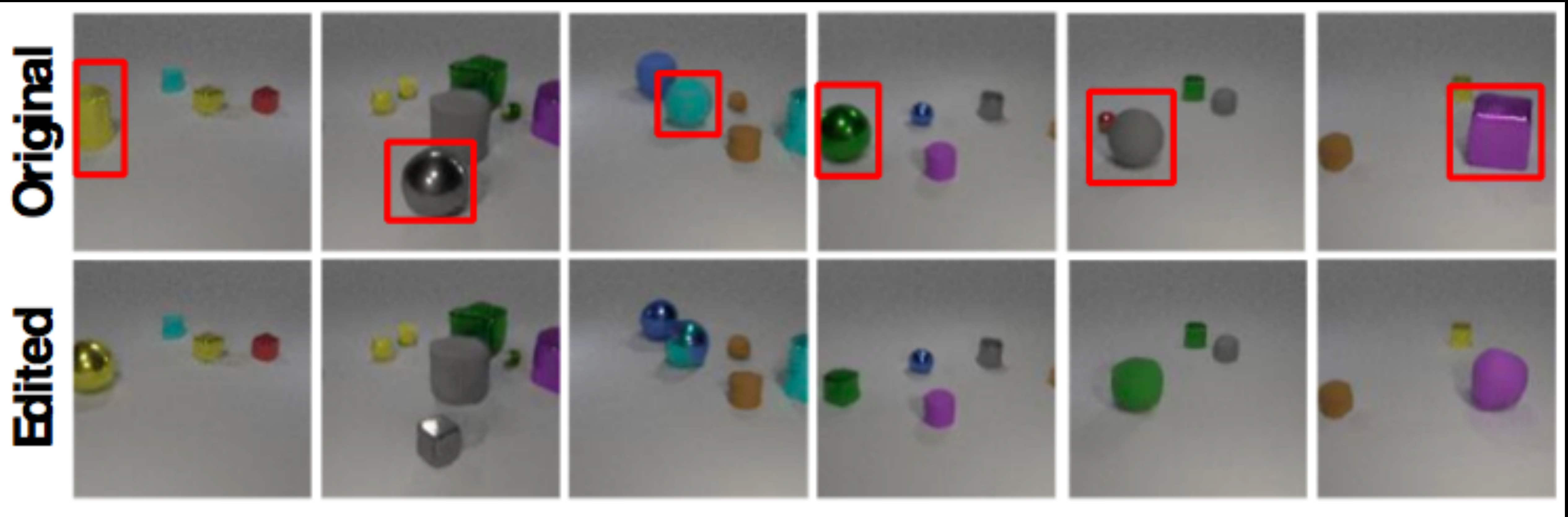
Generated training example



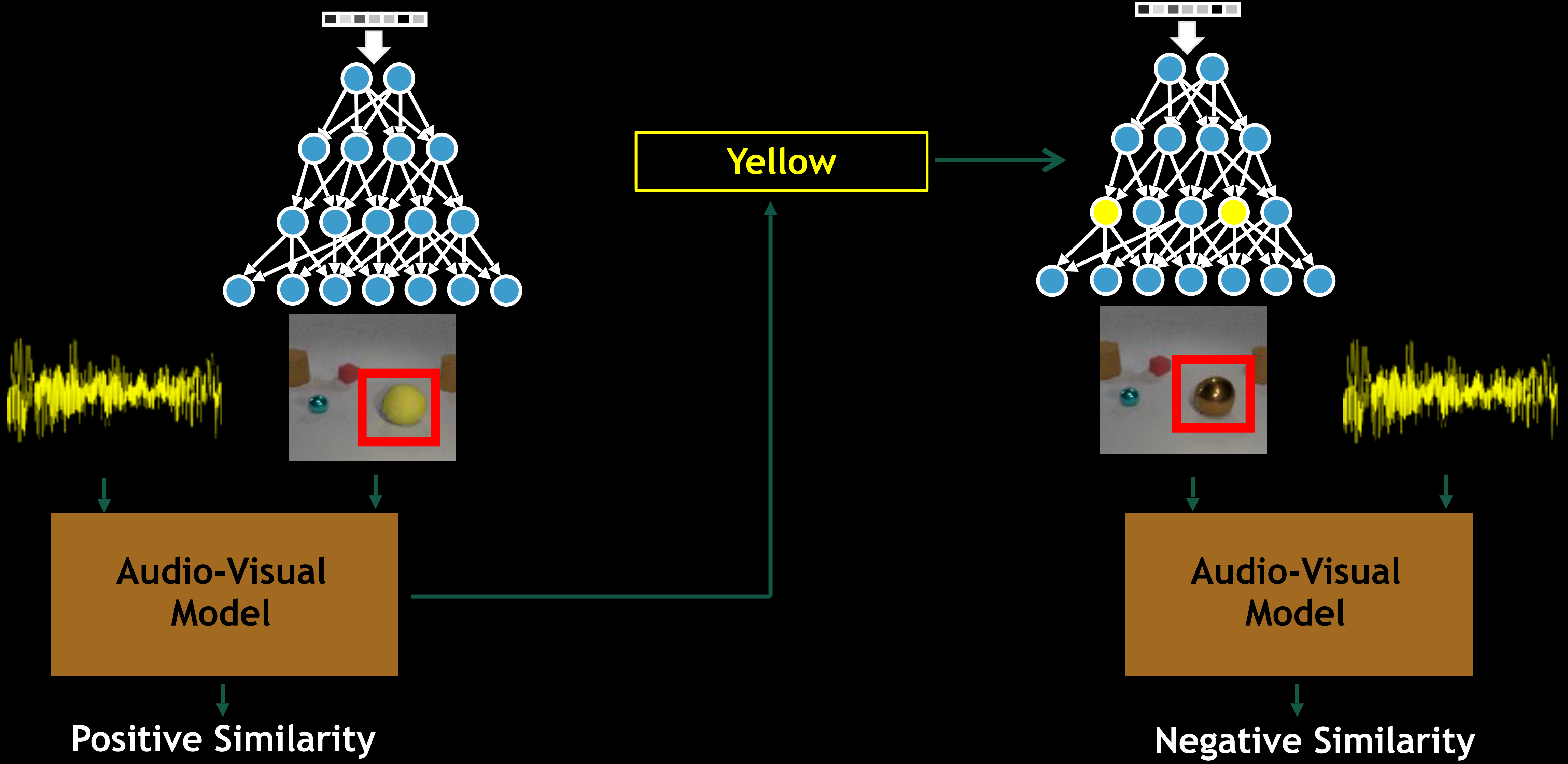
Learning which units to modify



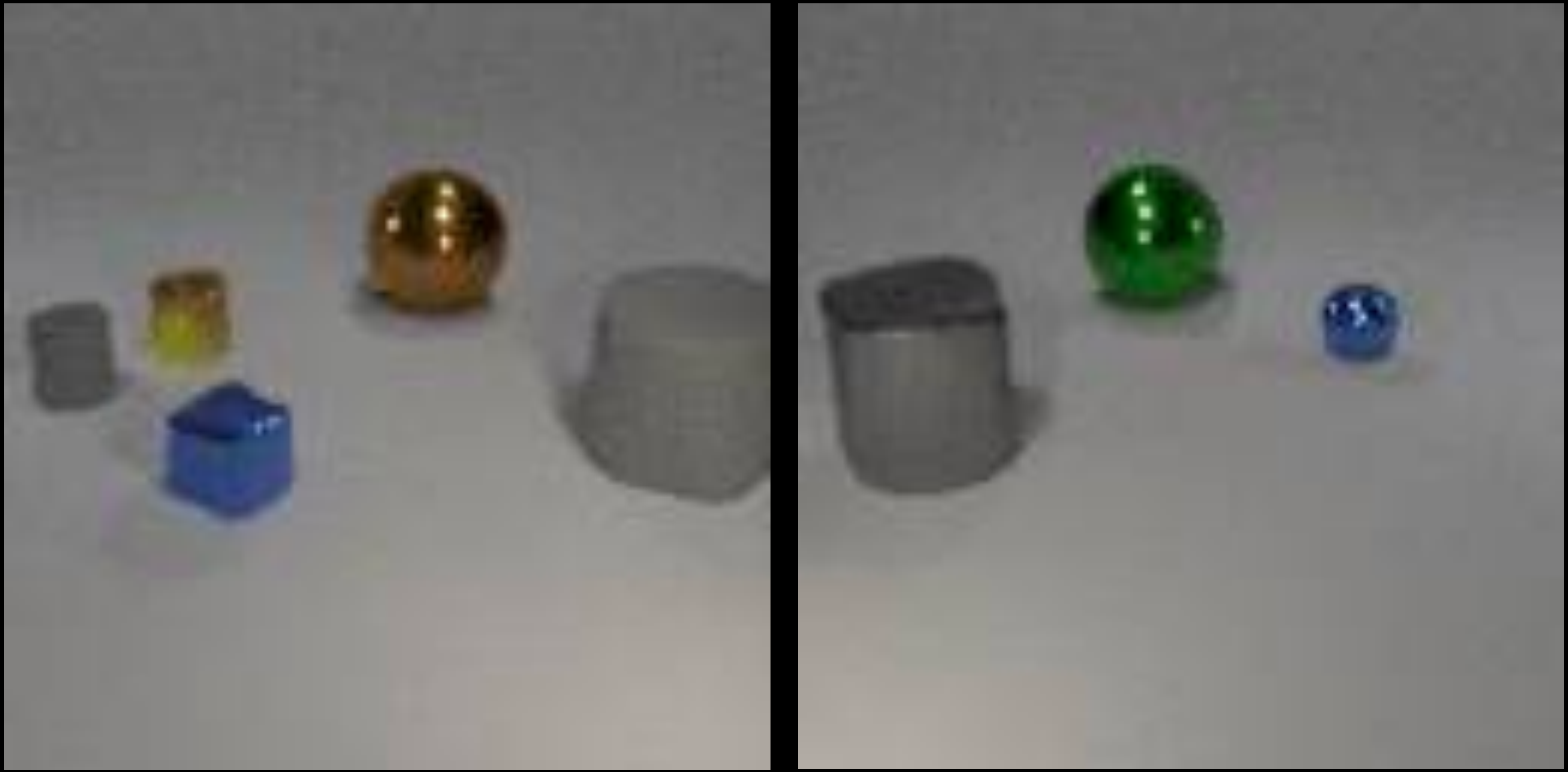
Editing generated images



System overview



Evaluating attributes



Results