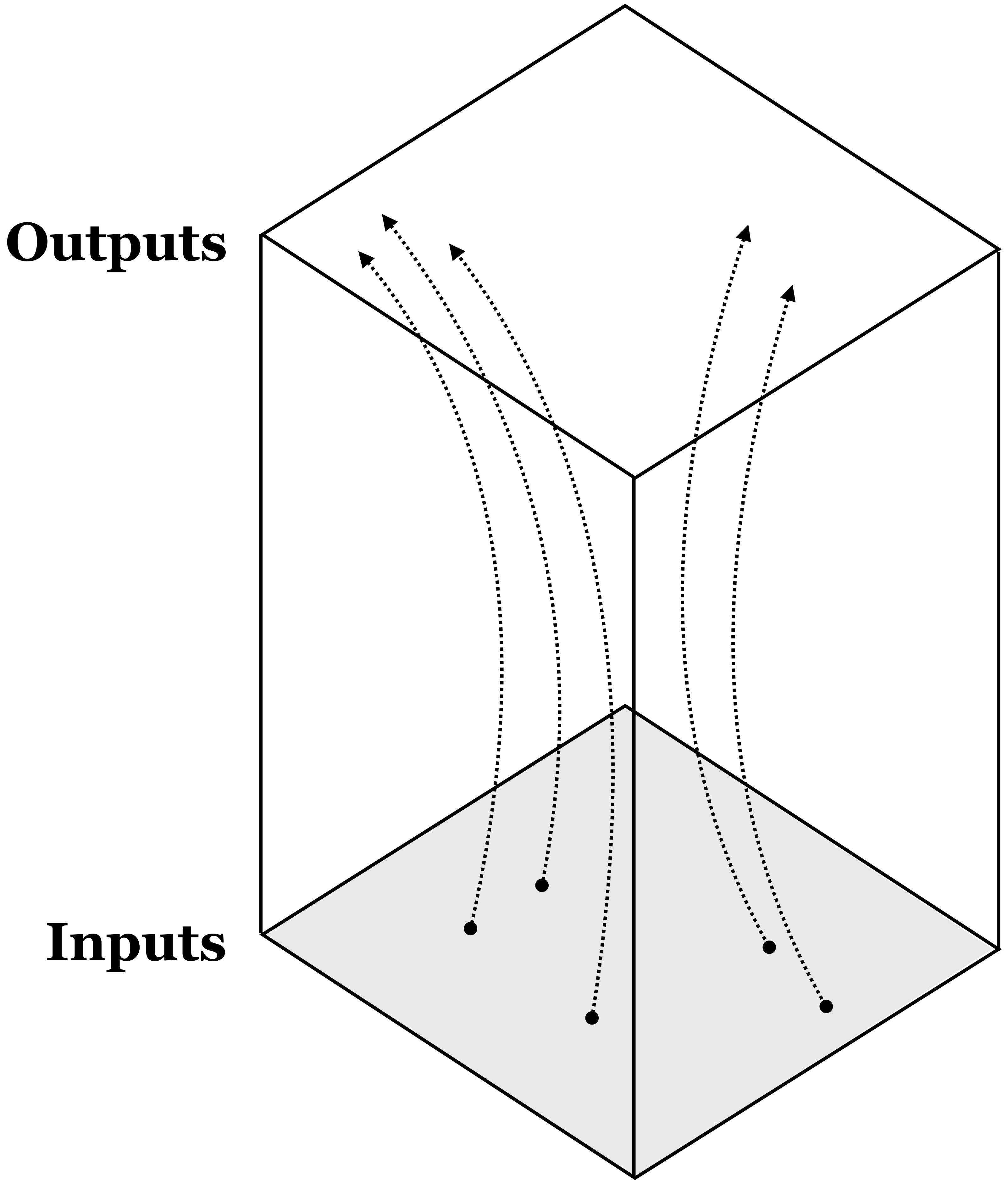
Lecture 19: Transfer Learning II

Speaker: Sara Beery

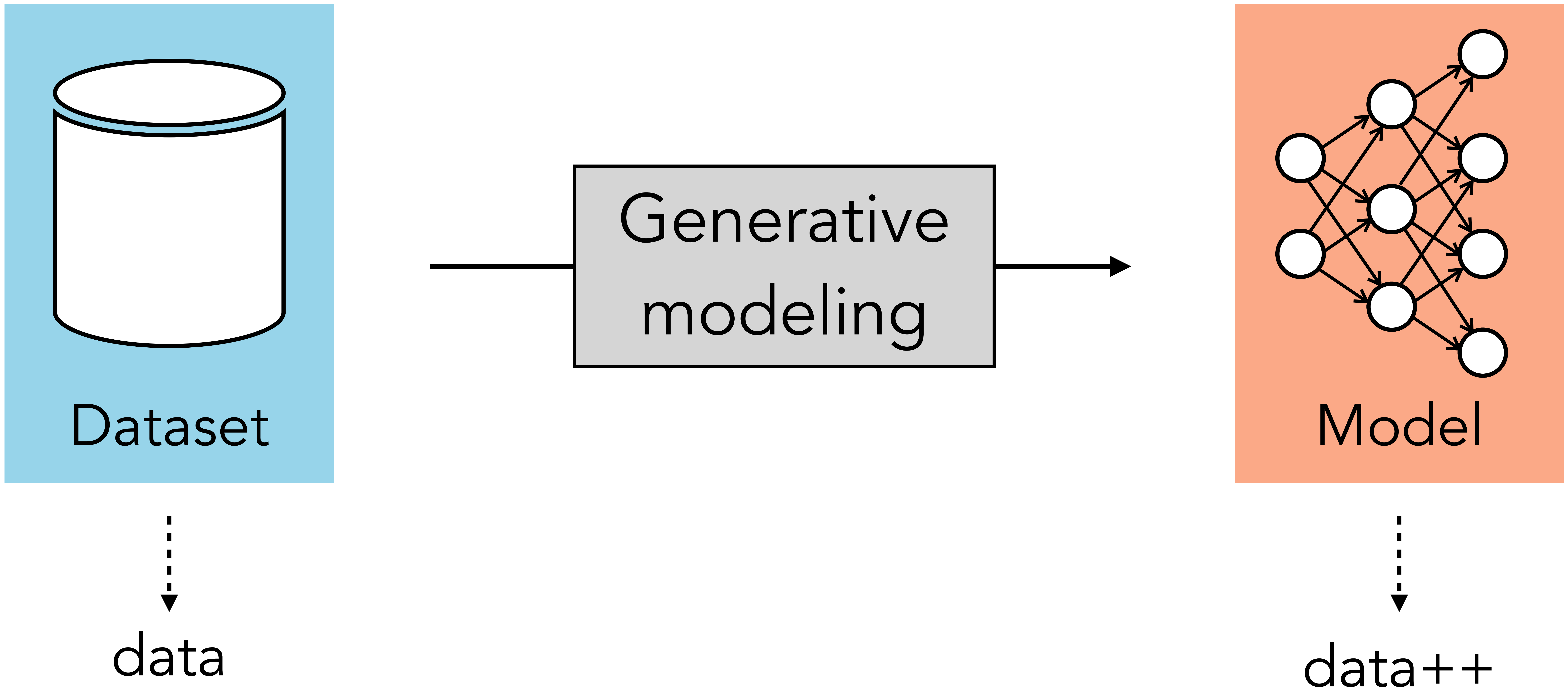
Transfer Learning II

- Transferring knowledge about the inputs
  - Generative models as data++
- Meta-learning

1. Knowledge about the mapping
2. Knowledge about the outputs
3. **Knowledge about the inputs**



Transfer Learning



"This release is the culmination of many hours of collective effort to create a single file that compresses the visual information of humanity into a few gigabytes."
Emad Mostaque [https://stability.ai/blog/stable-diffusion-public-release]

Data++: making data a first class object

$$\mathbb{X} = \{x, z, G, G^{-1}\}$$

Interpolation: $\alpha\mathbb{X}_1 + (1-\alpha)\mathbb{X}_2 \rightarrow \mathbb{X}_3$

[Goetschalckx*, Andonian*, Oliva, Isola, ICCV 2019]

Manipulation: $\mathbb{X}_1 + w \rightarrow \mathbb{X}_2$

[Jahanian*, Chai*, Isola, ICLR 2020]

Composition: $\mathbb{X}_1[m] + \mathbb{X}_2[1-m] \rightarrow \mathbb{X}_3$

[Chai, Wulff, Isola, ICLR 2021]

Optimization: $\arg\min_{\mathbb{X}} f(\mathbb{X})$

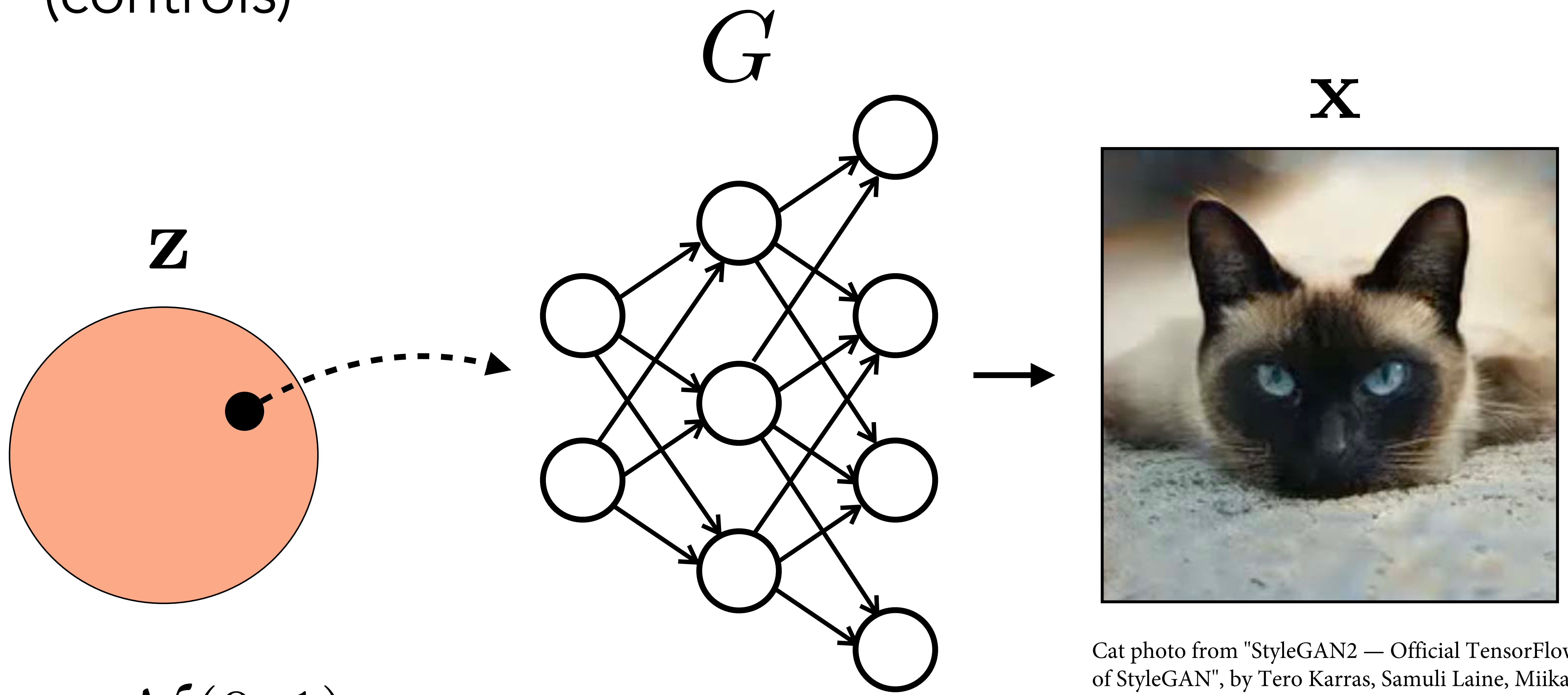
[Lin, Florence, Barron, et al. , IROS 2021]

$\rightarrow$ Graphics, visualization, data aug, counterfactual reasoning, …

Generative Models

Latent variables
(controls)

Synthesized Image



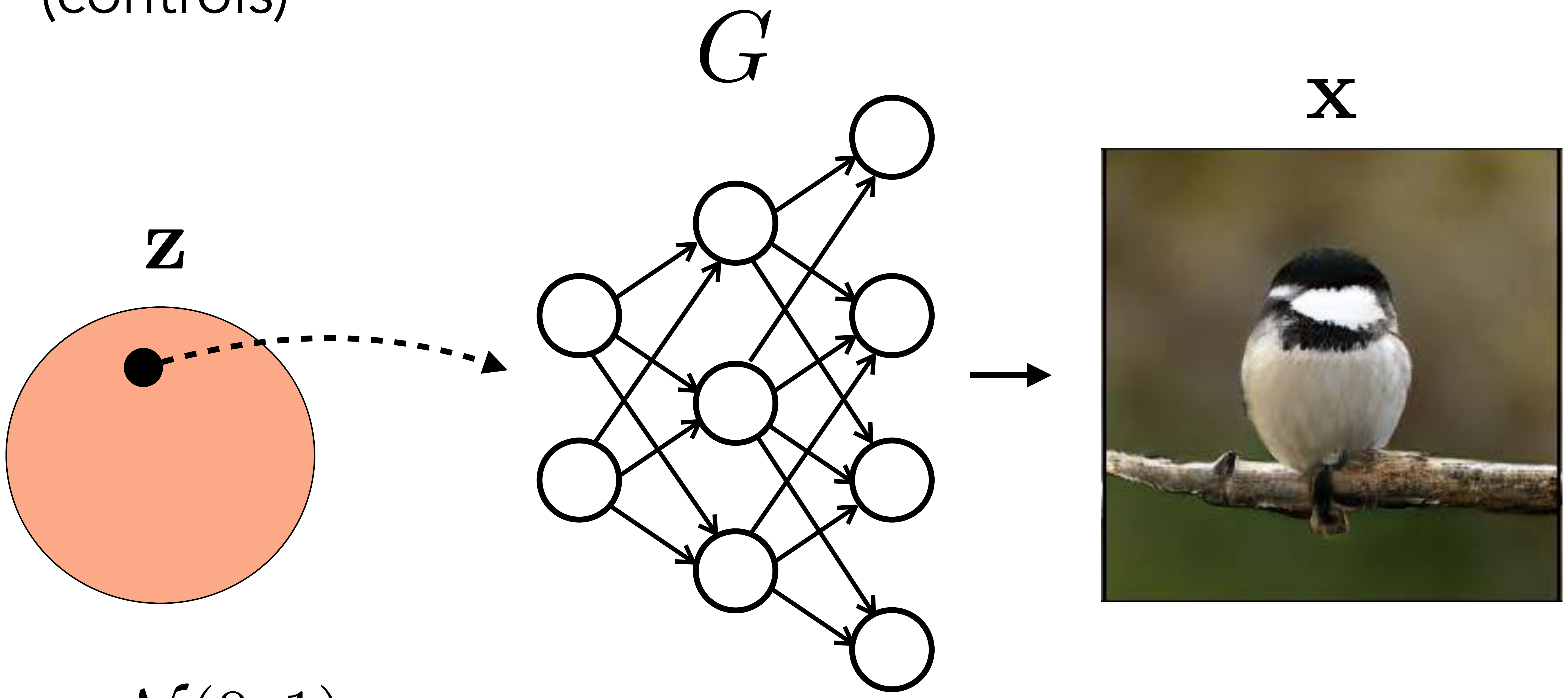
$$\mathbf{z} \sim \mathcal{N}(0, 1)$$

Cat photo from "StyleGAN2 — Official TensorFlow Implementation - Analyzing and Improving the Image Quality of StyleGAN", by Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila (https://github.com/NVlabs/stylegan2). Made available under the Nvidia Source Code License-NC (https://nvlabs.github.io/stylegan2/license.html).

Generative Models

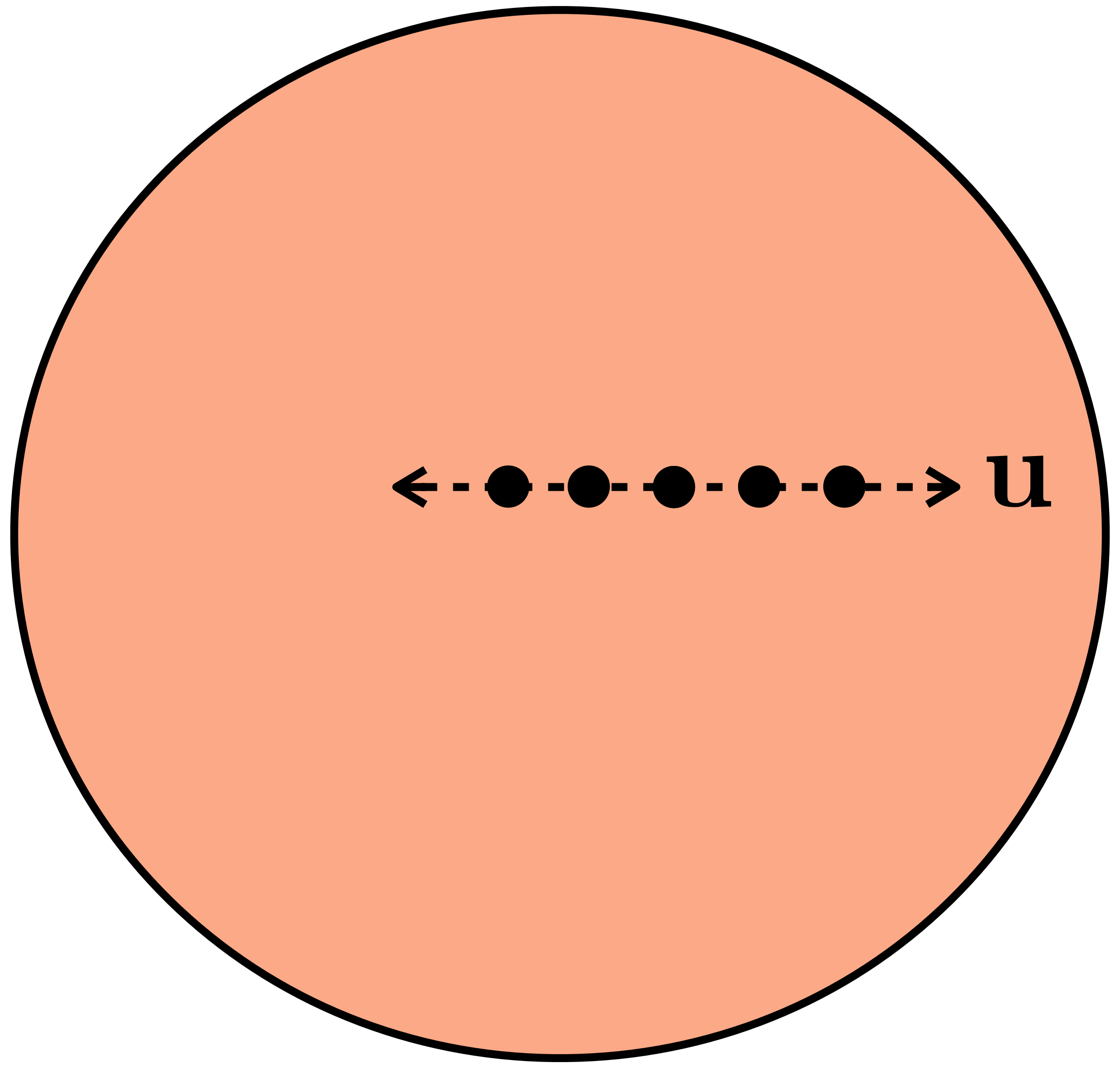
Latent variables
(controls)

Synthesized Image

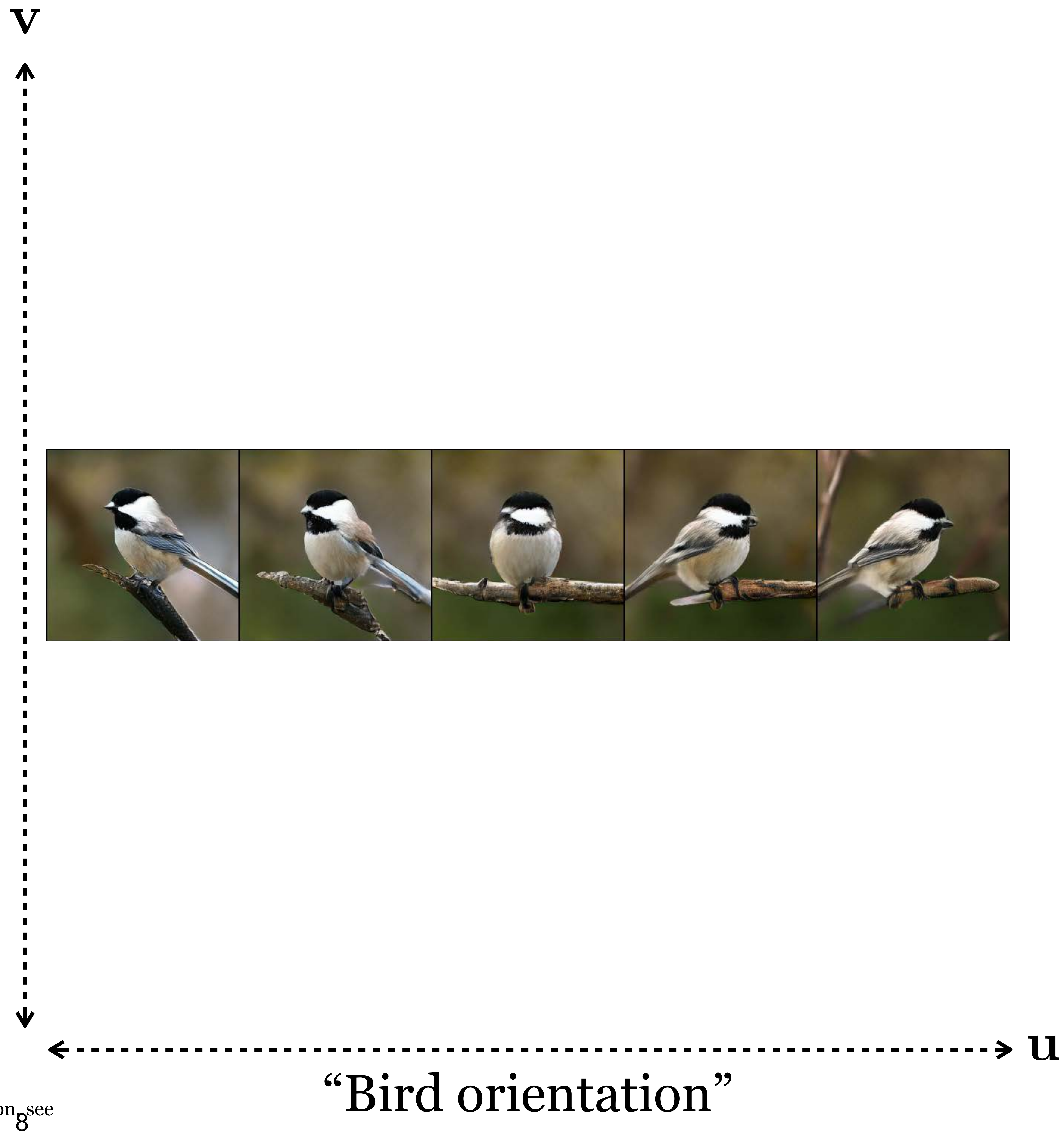


$\mathbf{z} \sim \mathcal{N}(0, 1)$

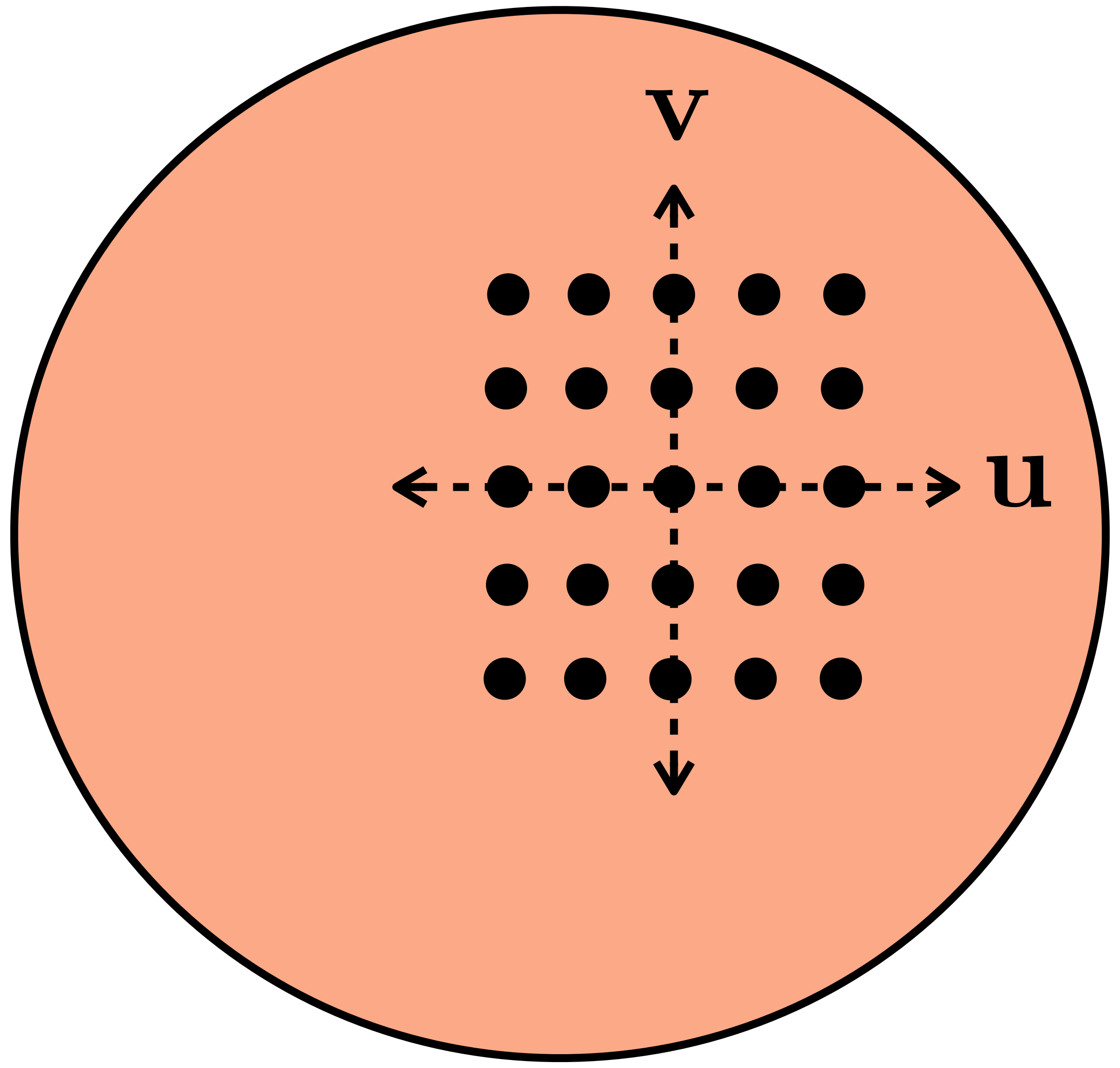
Latent variables
(controls)



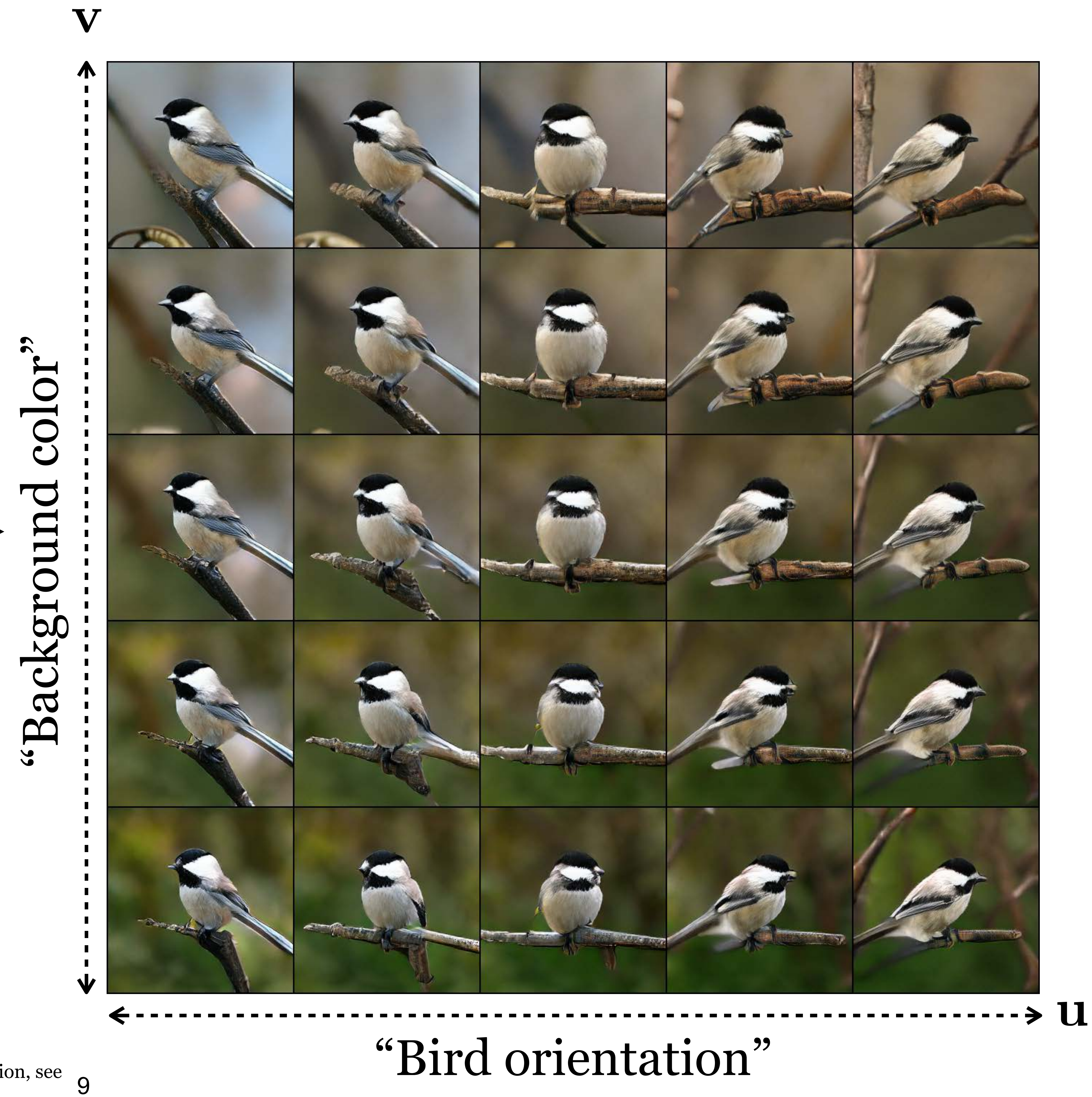
Synthesized Images



Latent variables
(controls)



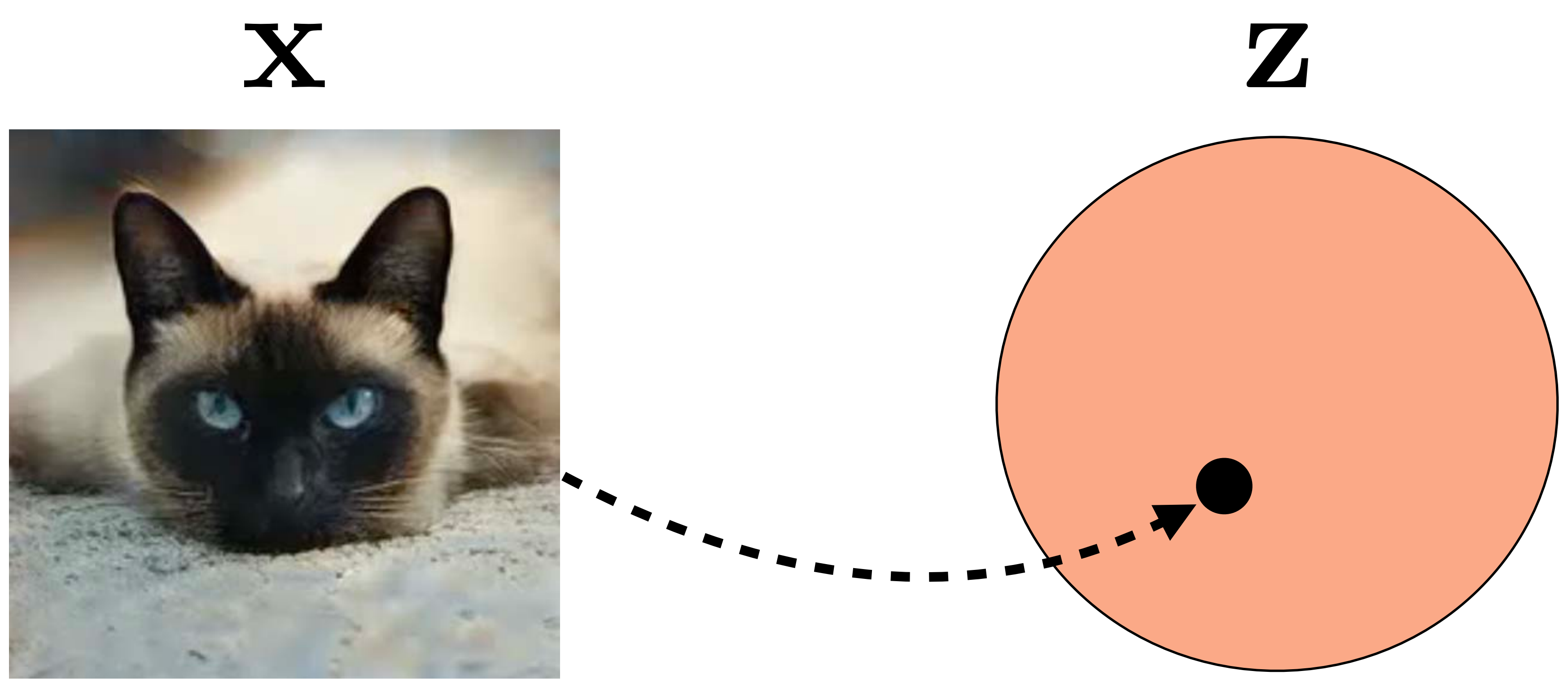
Synthesized Images



Data++ supports counterfactual reasoning

"What would it look like if…"

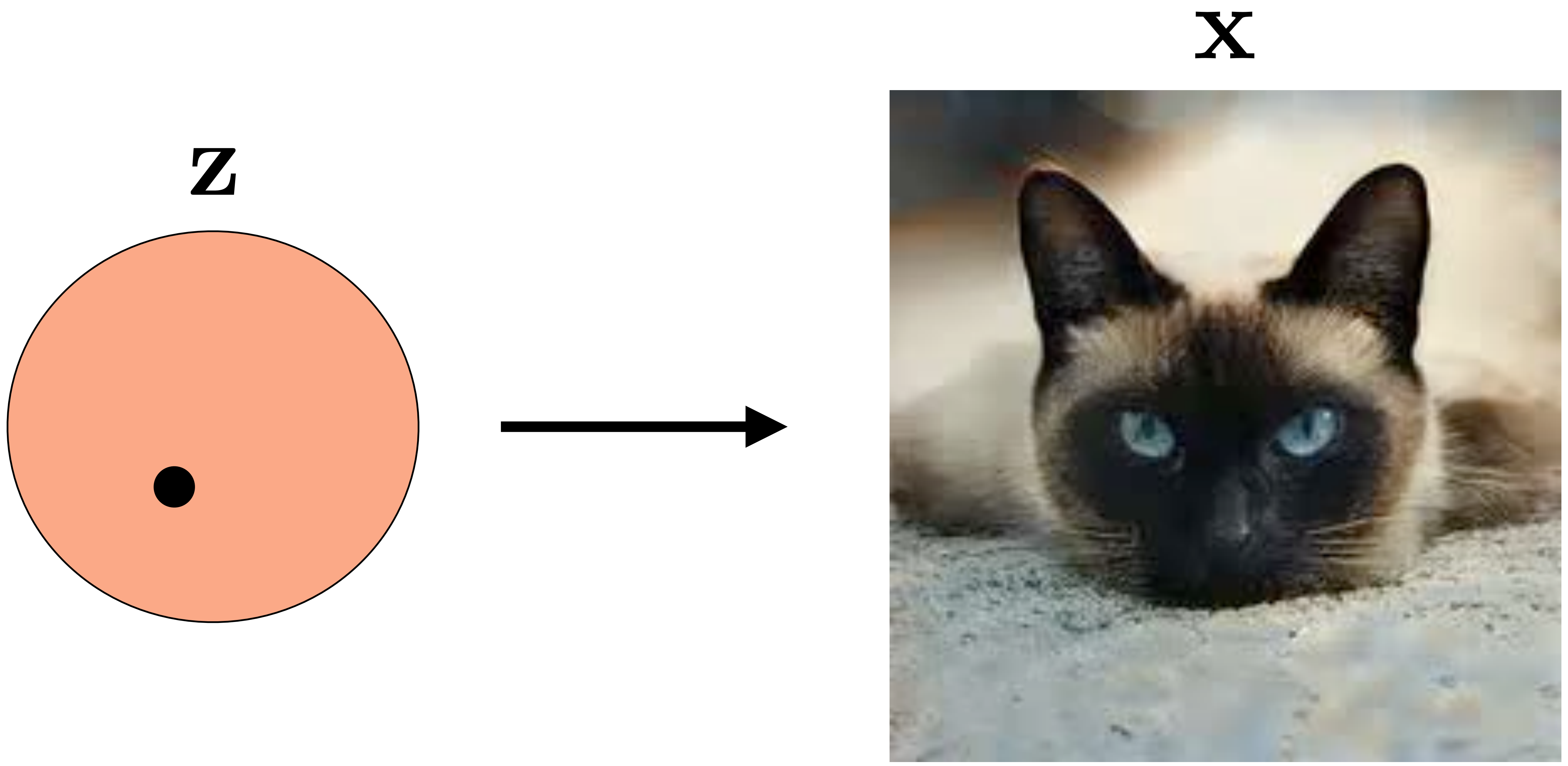
Data2Data++



Cat photo from "StyleGAN2 — Official TensorFlow Implementation - Analyzing and Improving the Image Quality of StyleGAN", by Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila (https://github.com/NVlabs/stylegan2). Made available under the Nvidia Source Code License-NC (https://nvlabs.github.io/stylegan2/license.html).

Data++ supports counterfactual reasoning

"What would it look like if…"

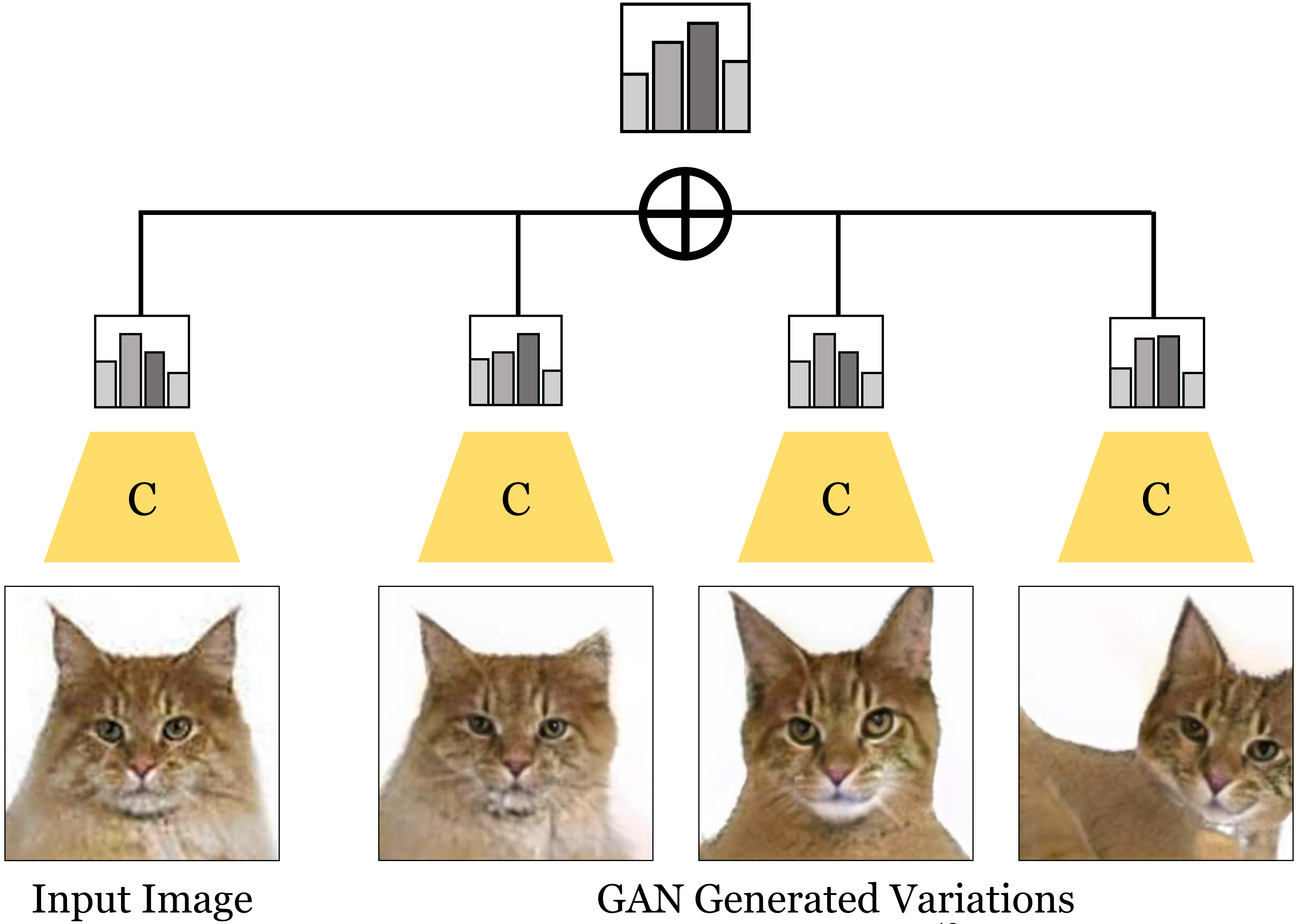


Cat photo from "StyleGAN2 — Official TensorFlow Implementation - Analyzing and Improving the Image Quality of StyleGAN", by Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila (https://github.com/NVlabs/stylegan2). Made available under the Nvidia Source Code License-NC (https://nvlabs.github.io/stylegan2/license.html).

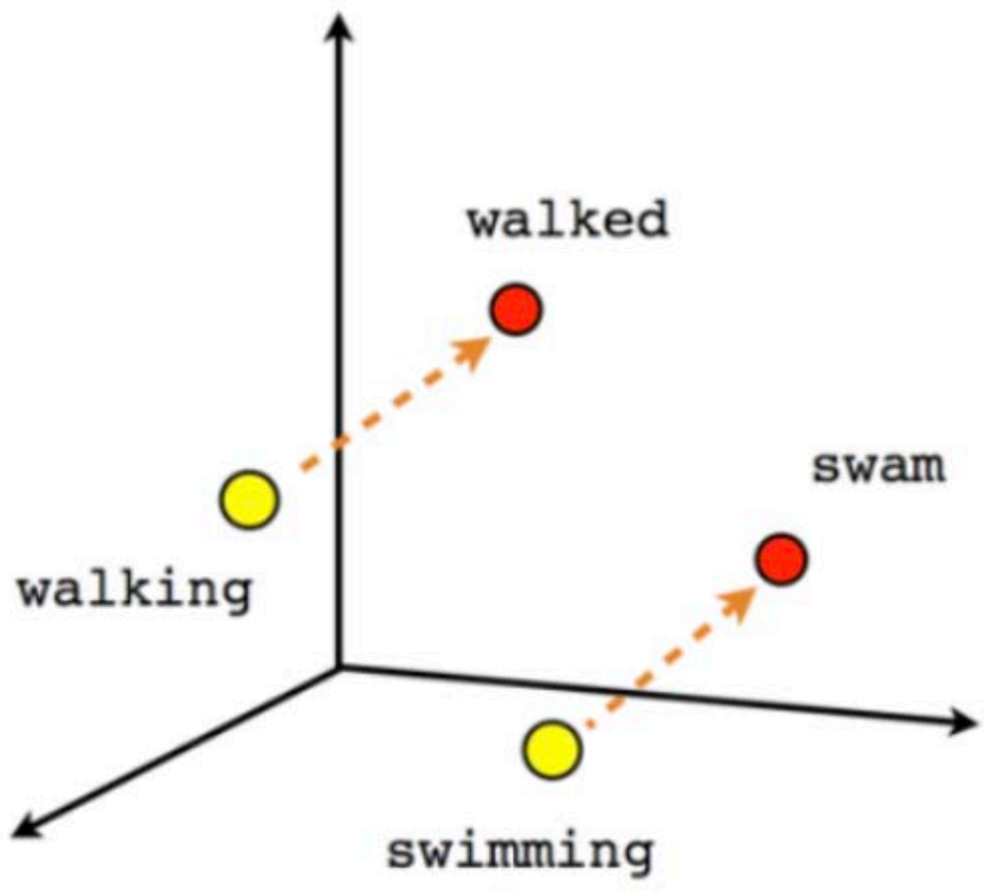
Ensembling with Deep Generative Views

Chai, Zhu, Shechtman, Isola, Zhang

CVPR 2021

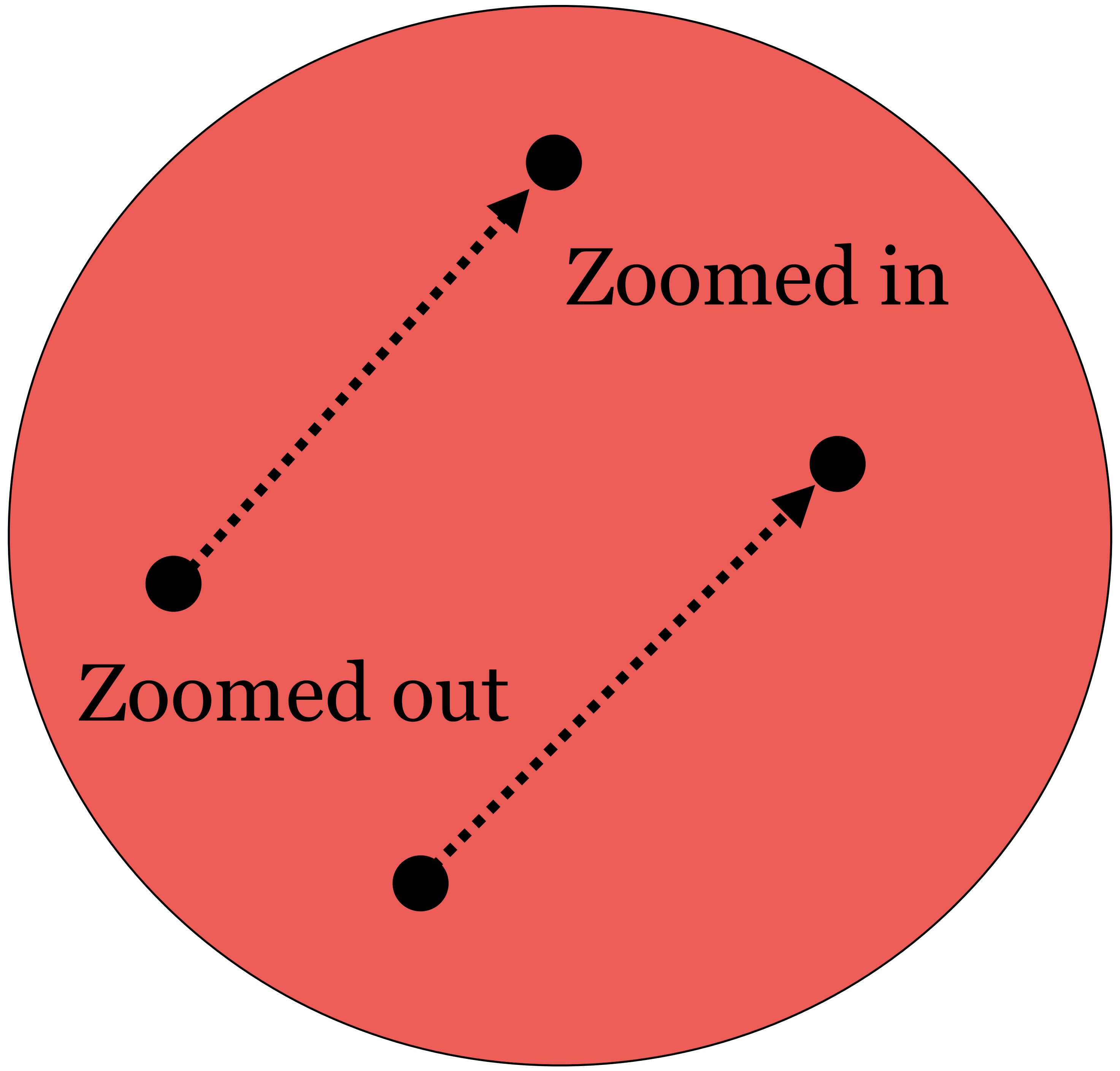


Improved accuracy and robustness



[word2vec, Mikolov et al., 2013]

[DCGAN, Radford, Metz, Chintala, 2015]



Zoom Shift

Brighten Darken

Color transformation vectors

Winter to spring

input output

Turning on the lights

input output

Day to night

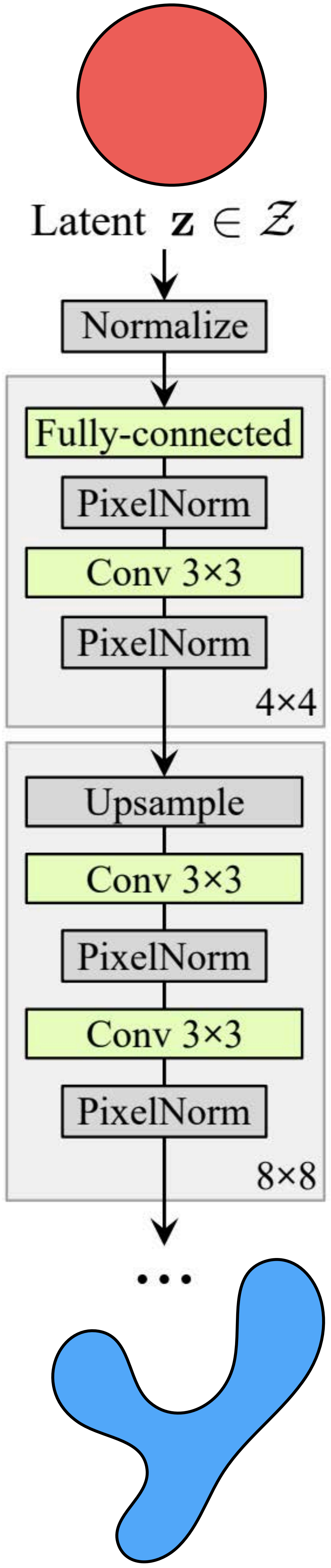
input output

Volcano!

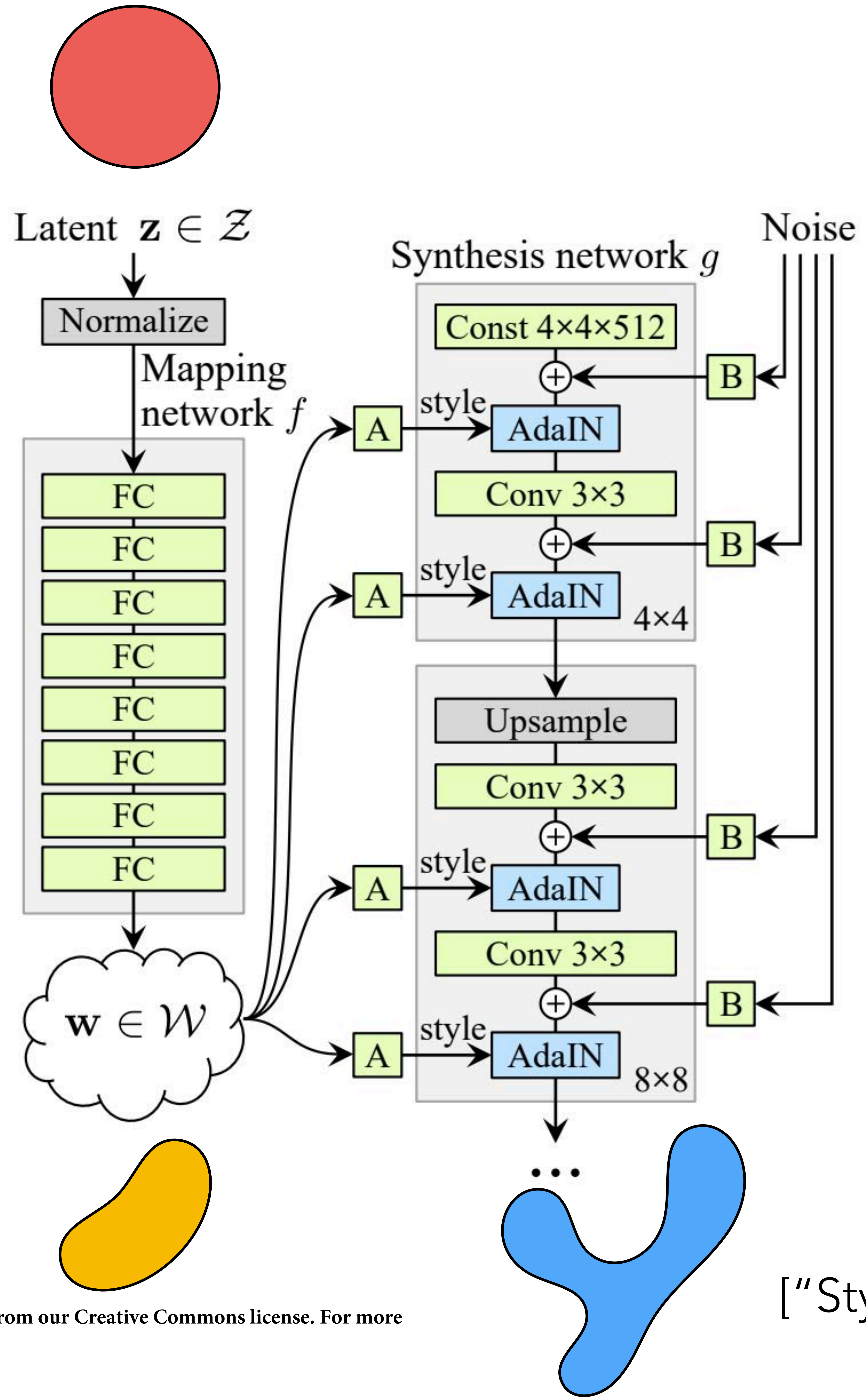
input output

There is a latent space “vector” for each of these transformations — the “spring vector”, the “volcano exploding vector”

Traditional GAN



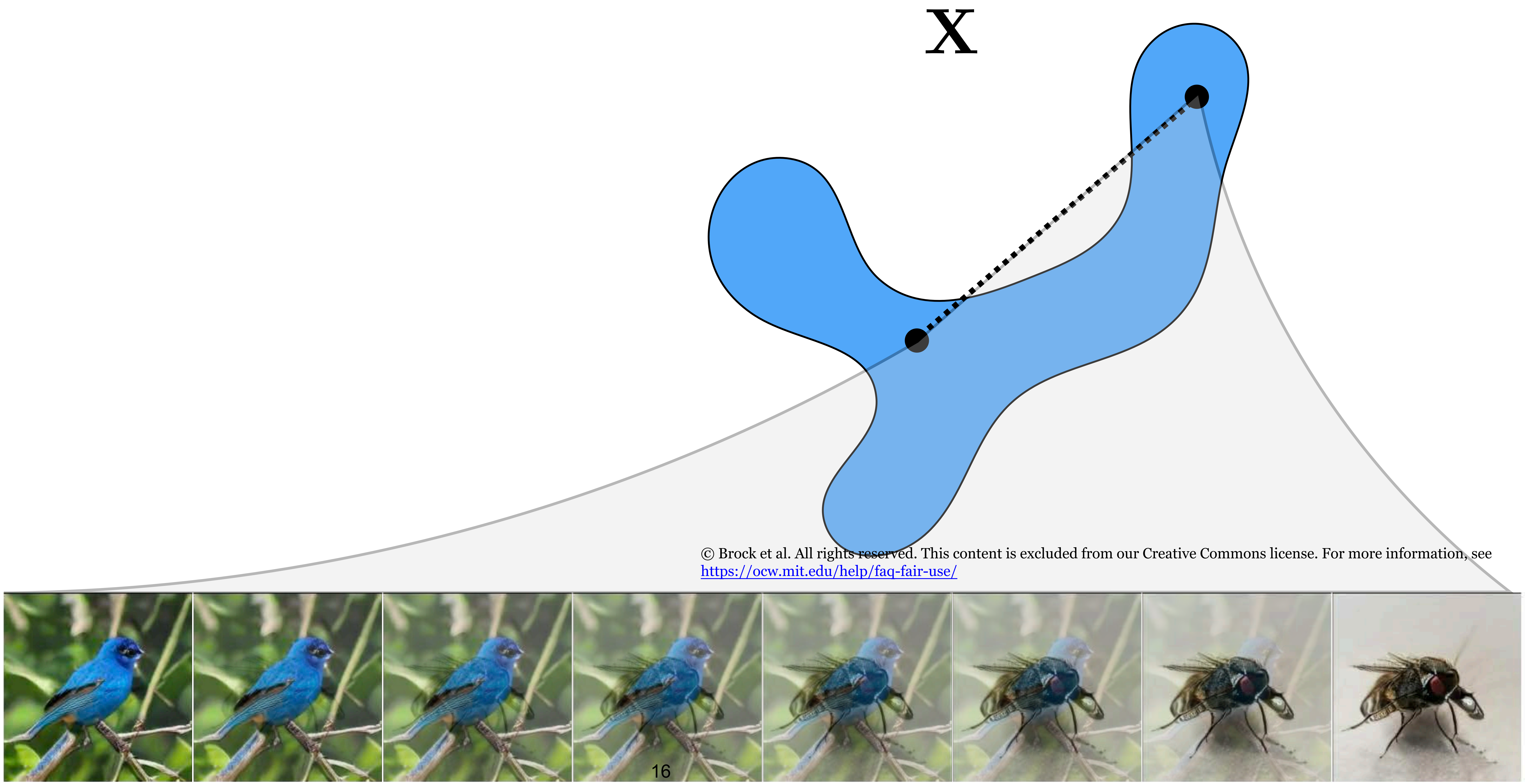
StyleGAN



["StyleGAN", Karras, Laine, Aila, 2018]

Interpolation in data space

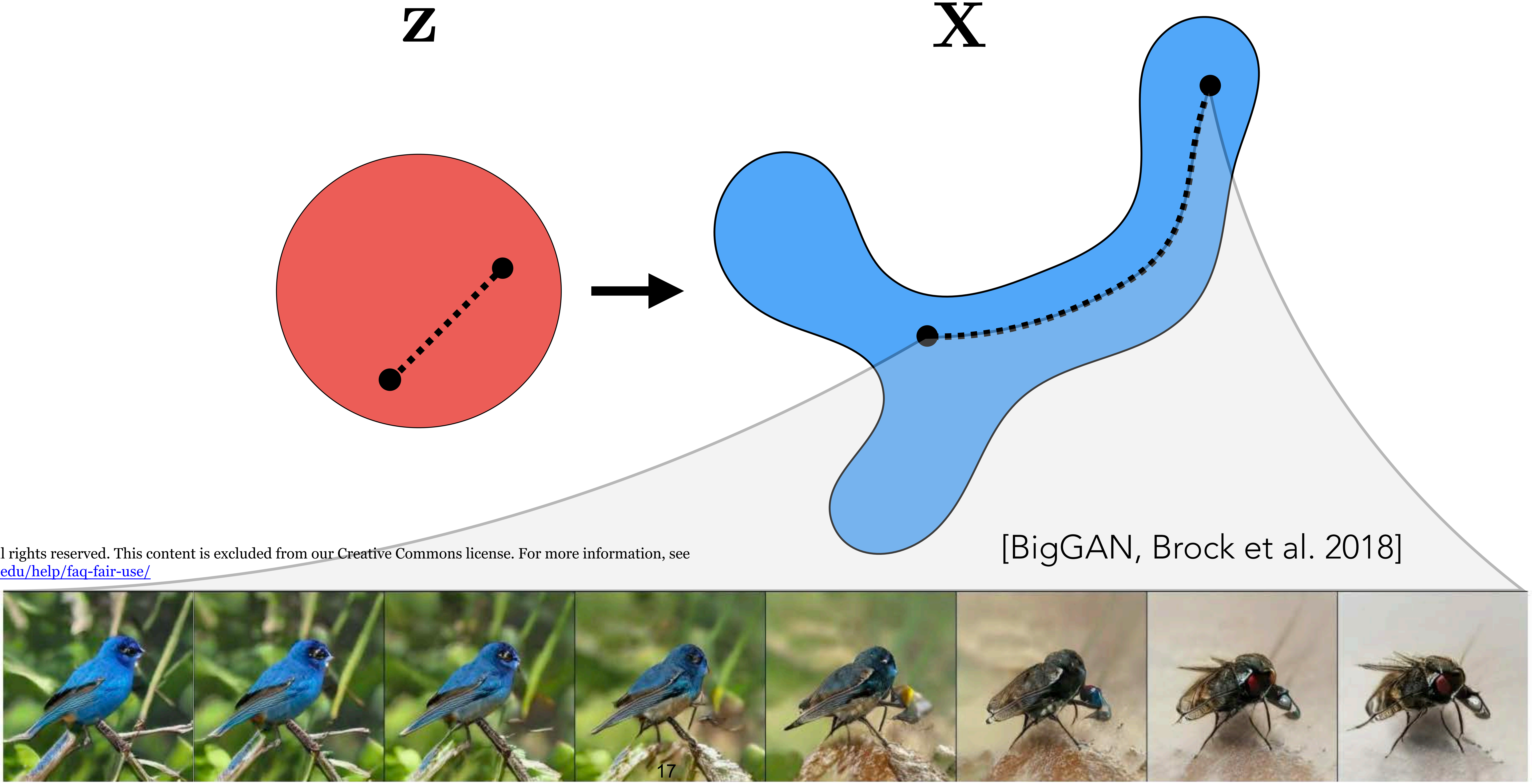
Data space
(Natural image manifold)



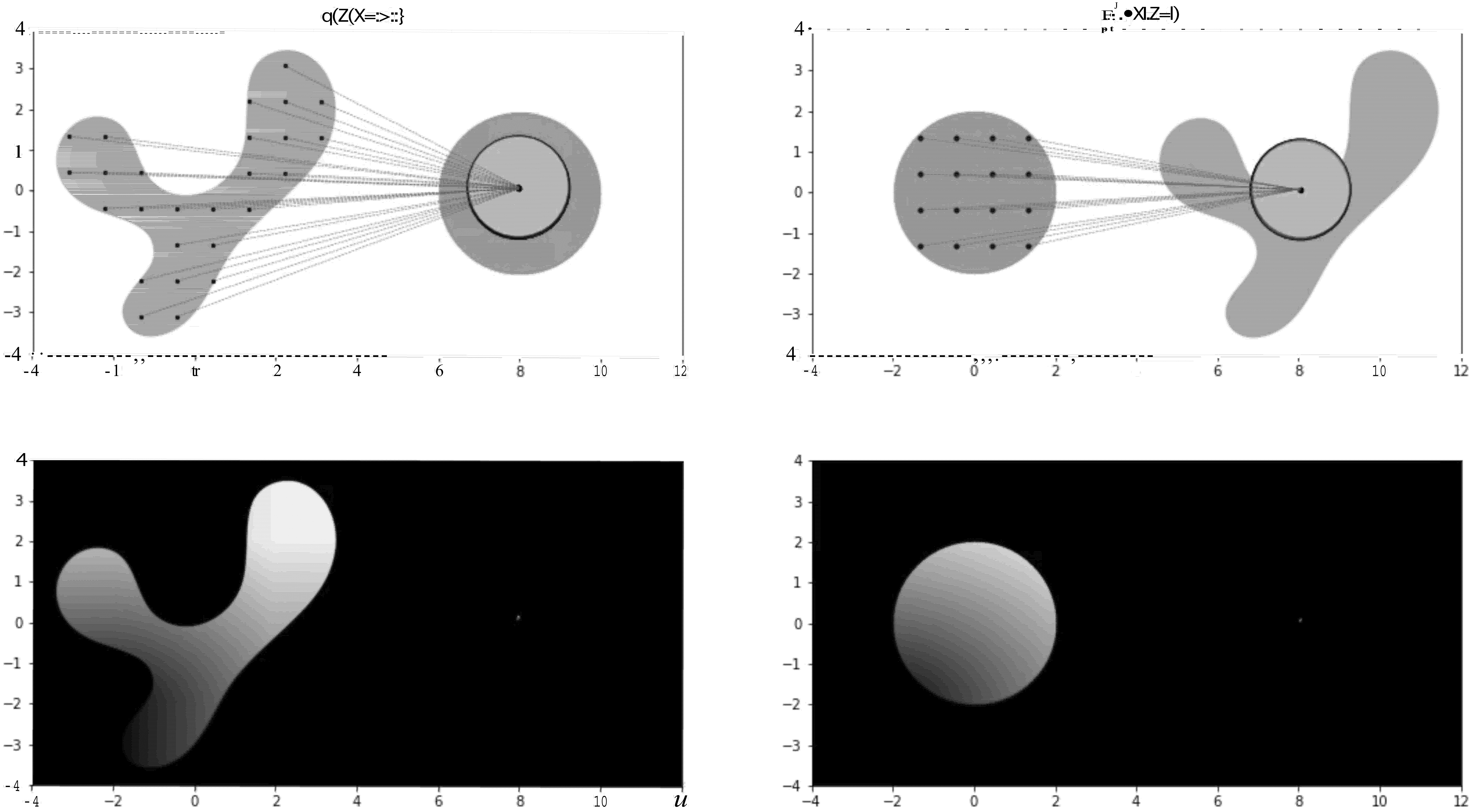
Interpolation in latent space

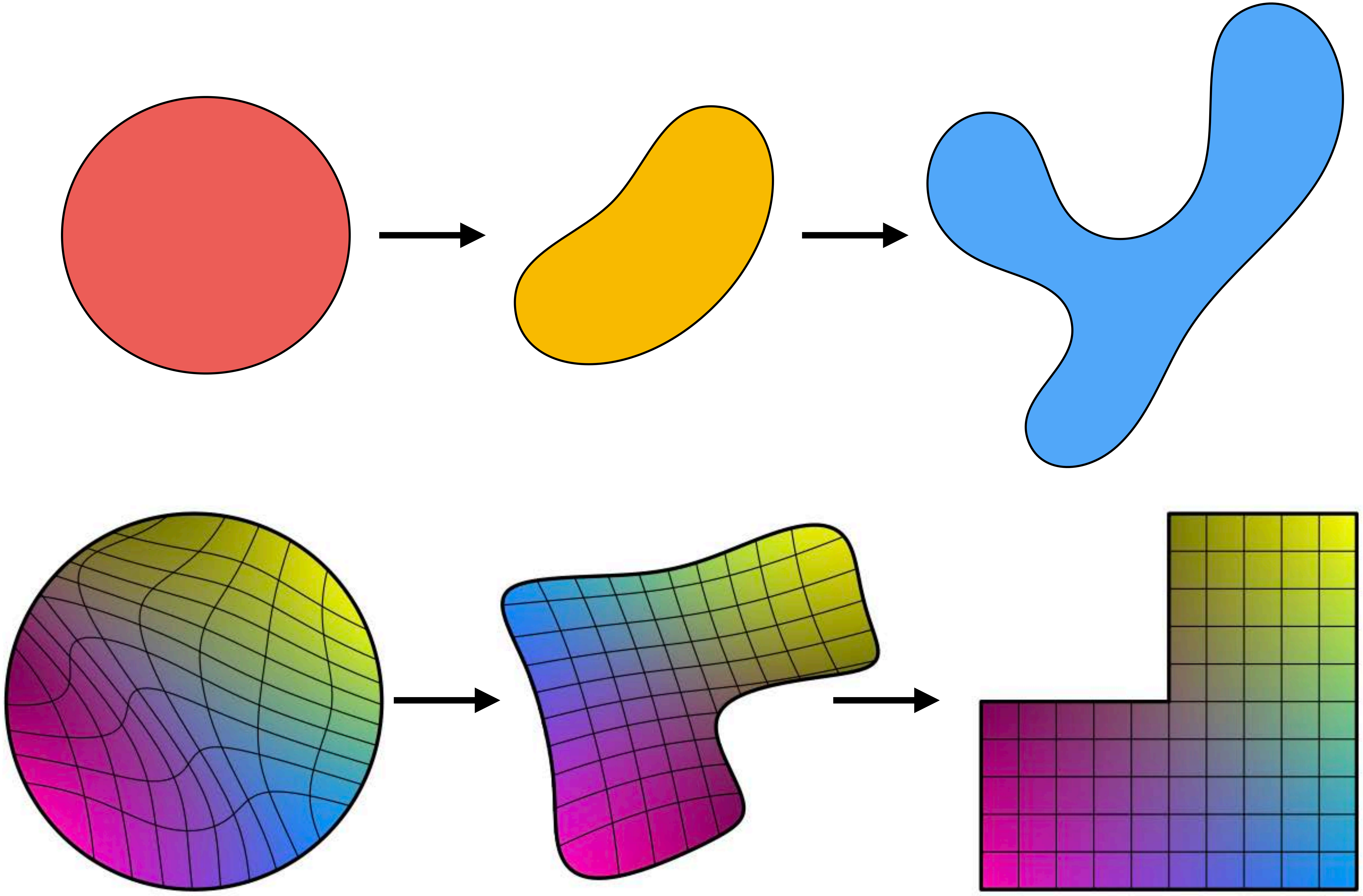
Latent space (Gaussian)

Data space (Natural image manifold)



[BigGAN, Brock et al. 2018]





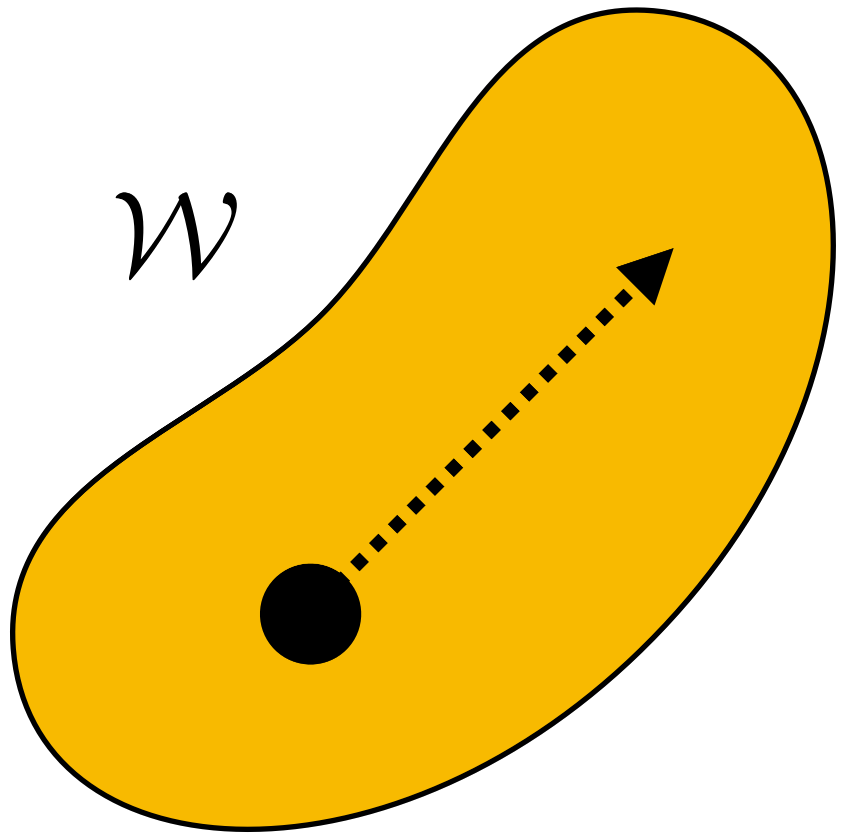
(b) Mapping from $\mathcal{Z}$ to features

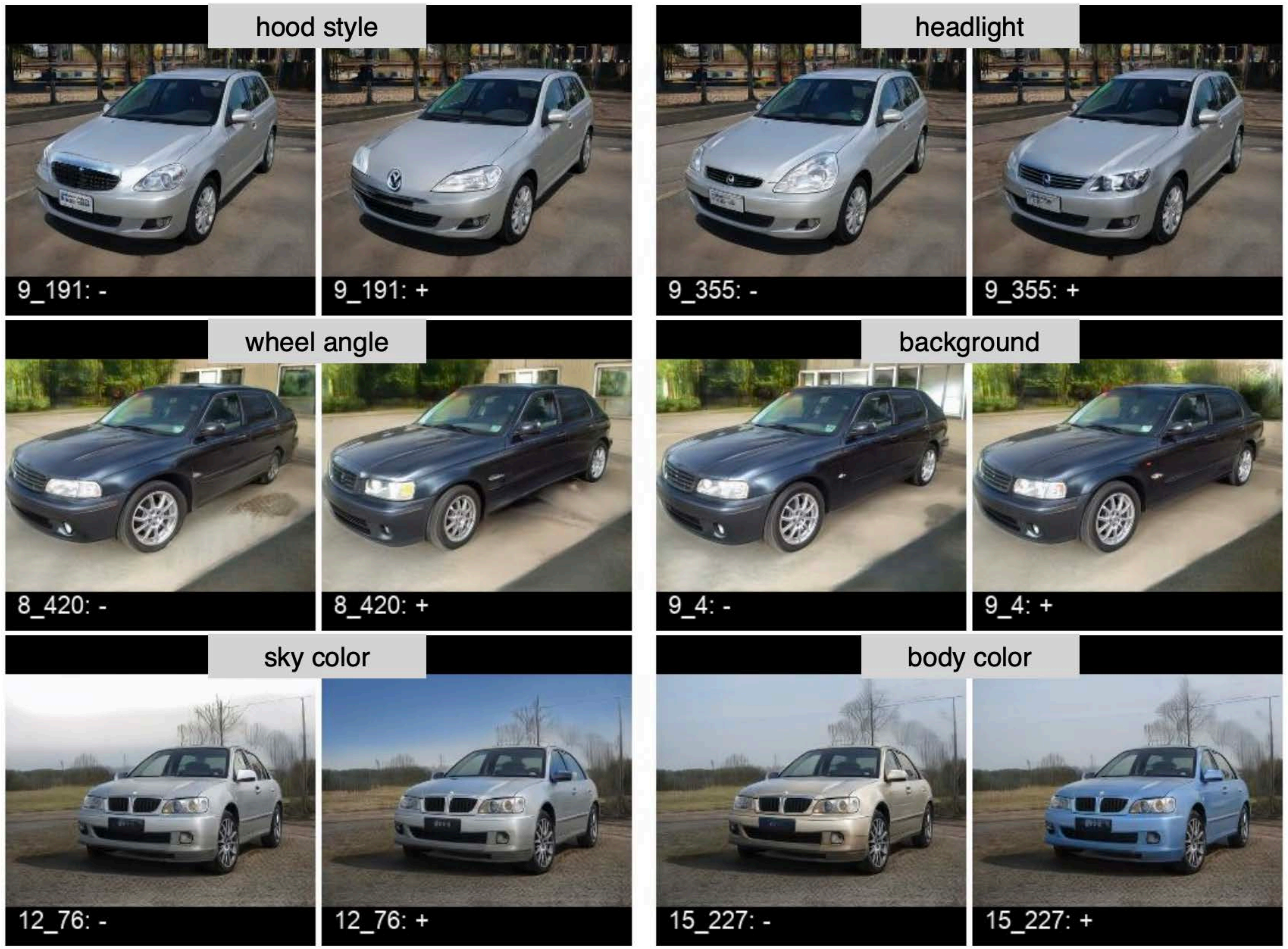
(c) Mapping from $\mathcal{W}$ to features

(a) Distribution of features in training set

["StyleGAN", Karras, Laine, Aila, 2018]

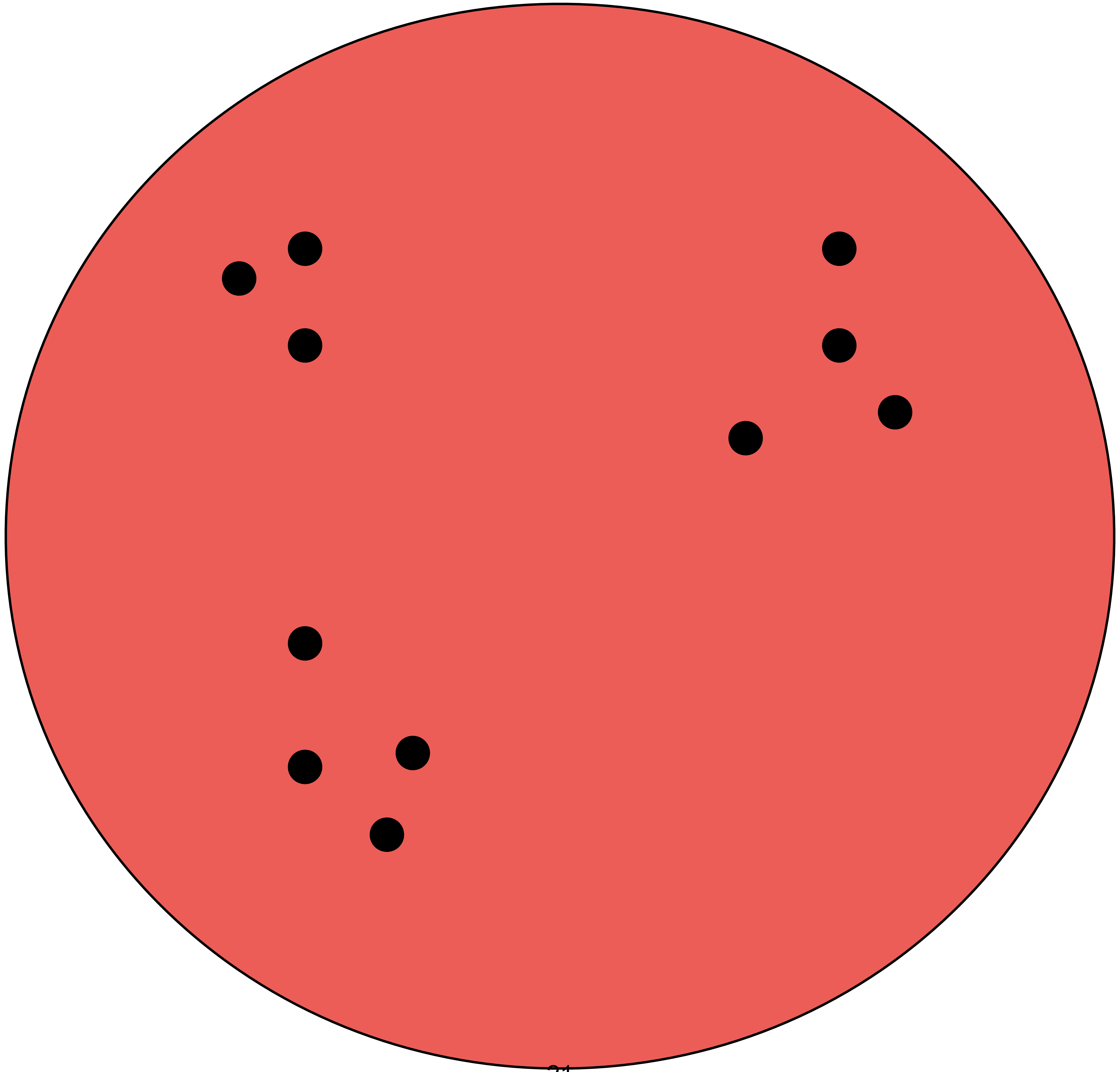
Different ways to navigate latent space



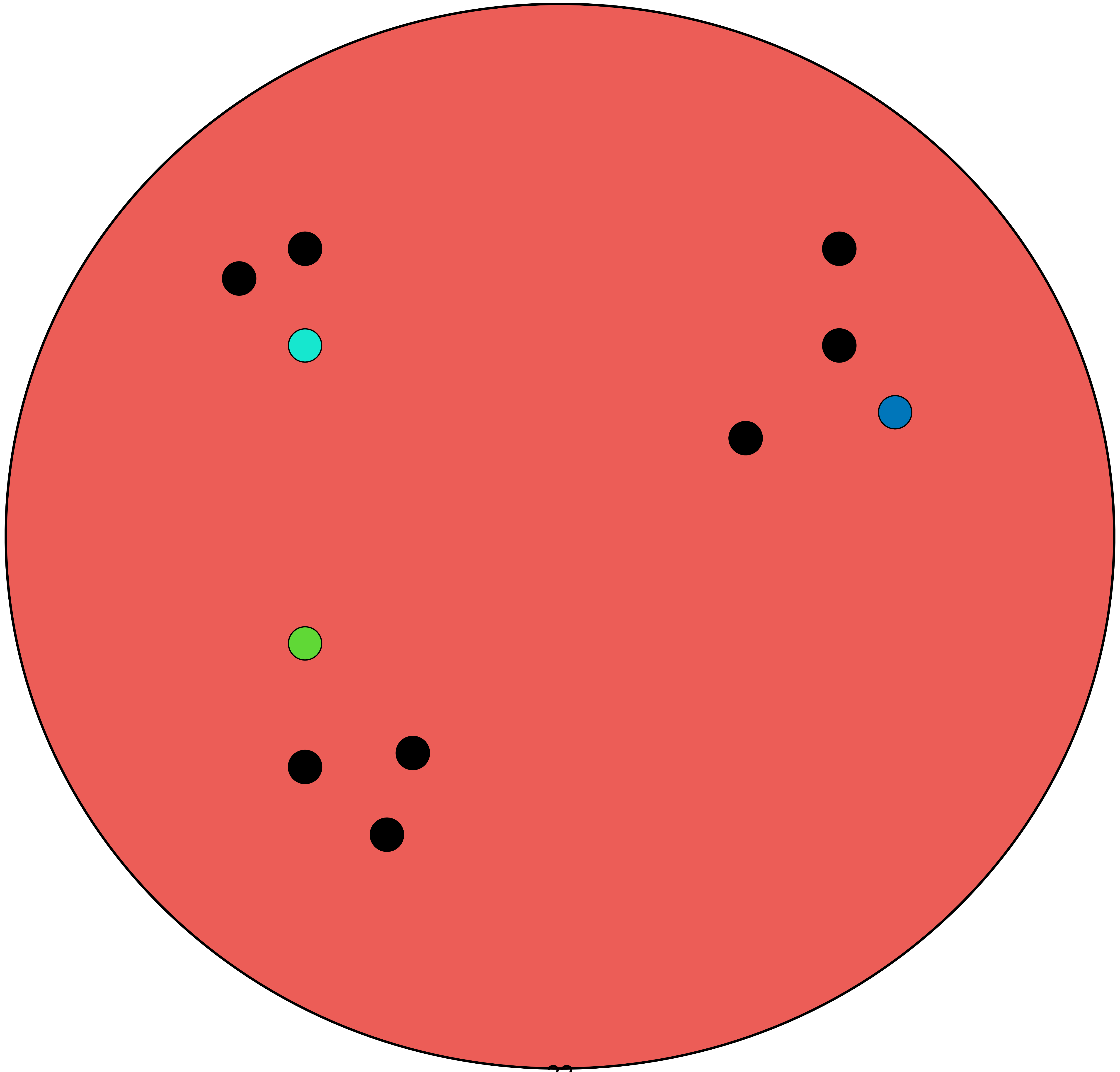


["StyleSpace", Wu, Lischinski, Shechtman, 2020]

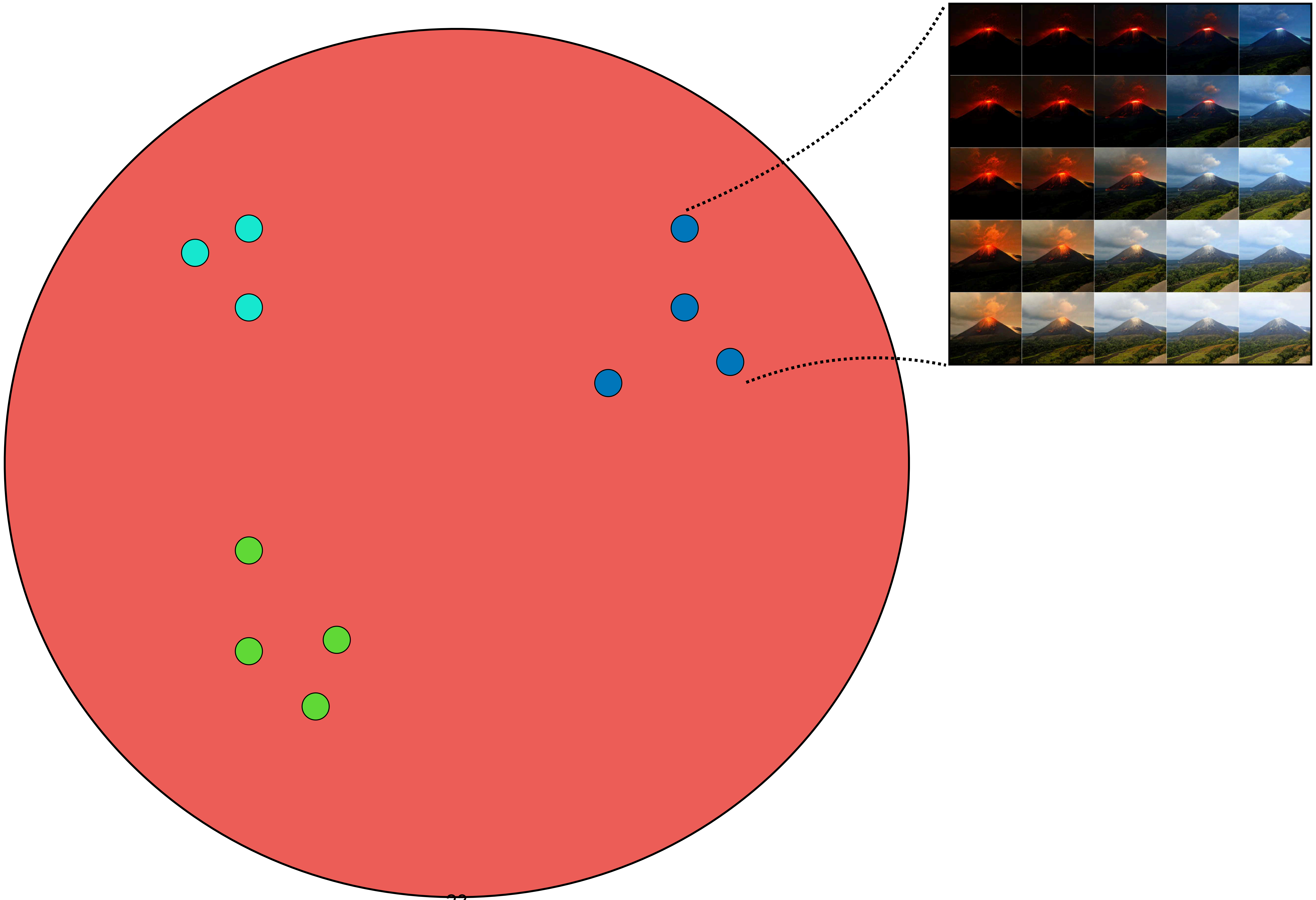
Labeling Data++



Labeling Data++

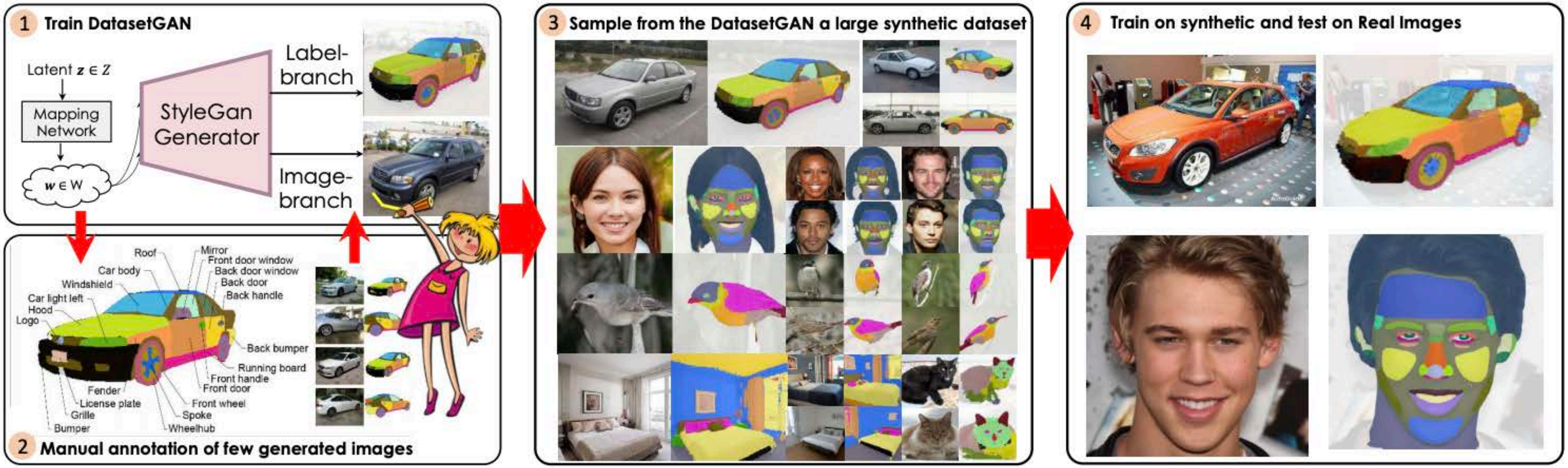


Labeling Data++



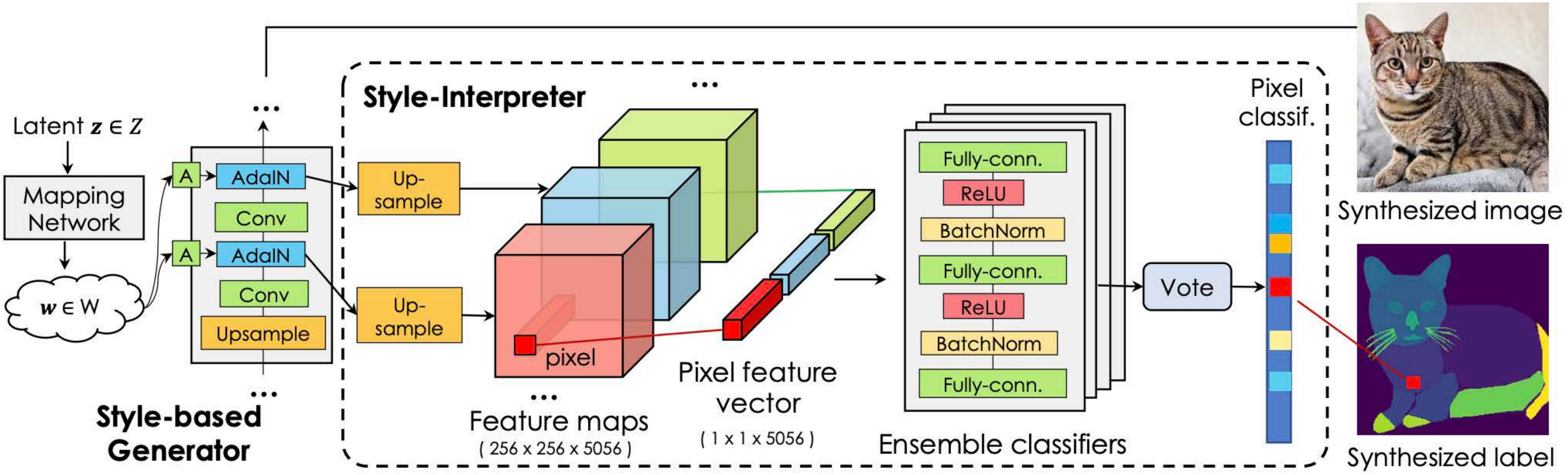
DatasetGAN

[Zhang*, Ling*, Gao, Yin, Lafleche, Barriuso, Torralba, Fidler 2021]

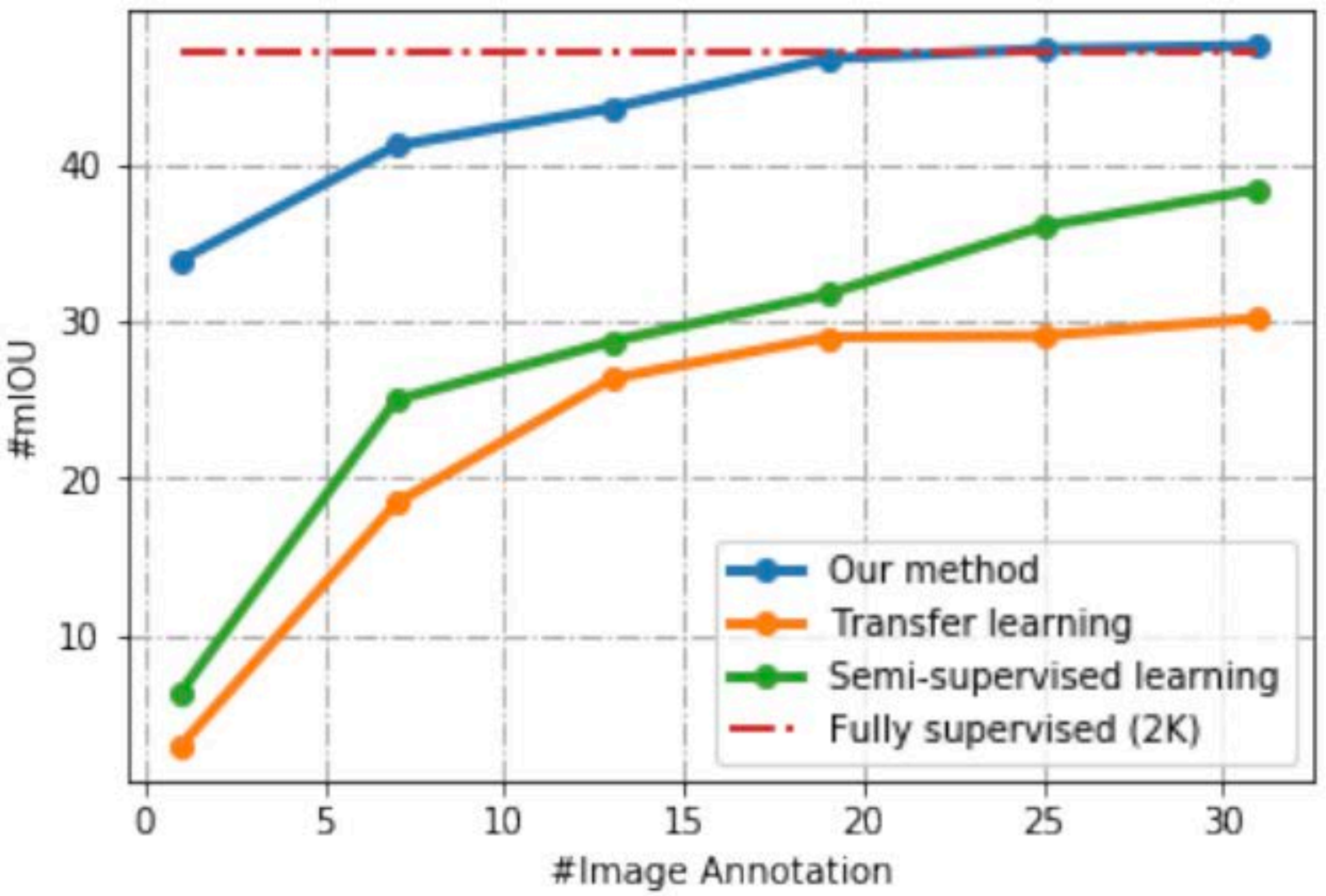


see also [Tritrong*, Rewatbowornwong*, Suwajanakorn, CVPR 2021]

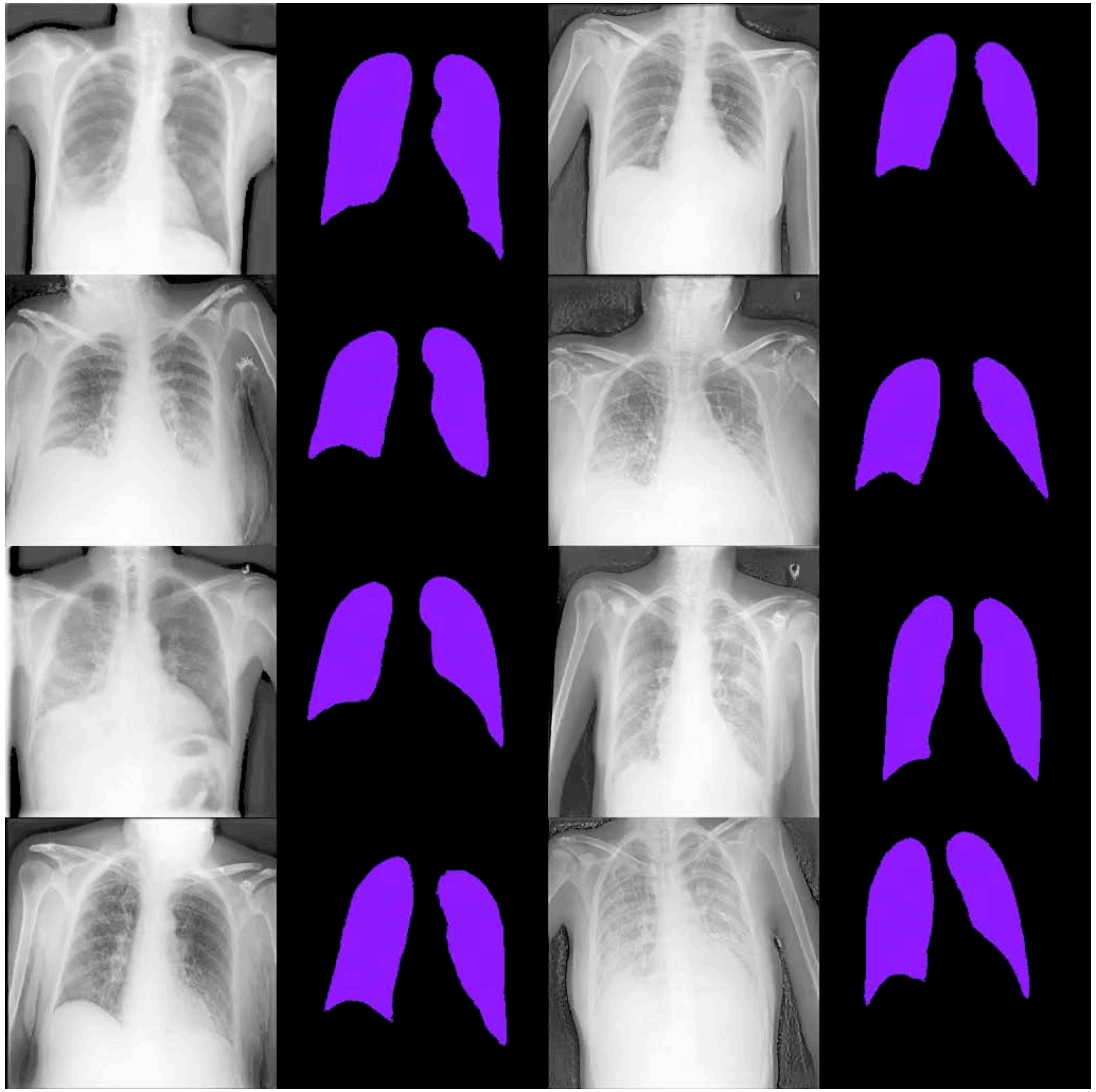
[Li, Yang, Kries, Torralba, Fidler, CVPR 2021]



[Zhang*, Ling*, Gao, Yin, Lafleche, Barriuso, Torralba, Fidler, CVPR 2021]



1 labeled GAN image is worth ~100 labeled regular images!



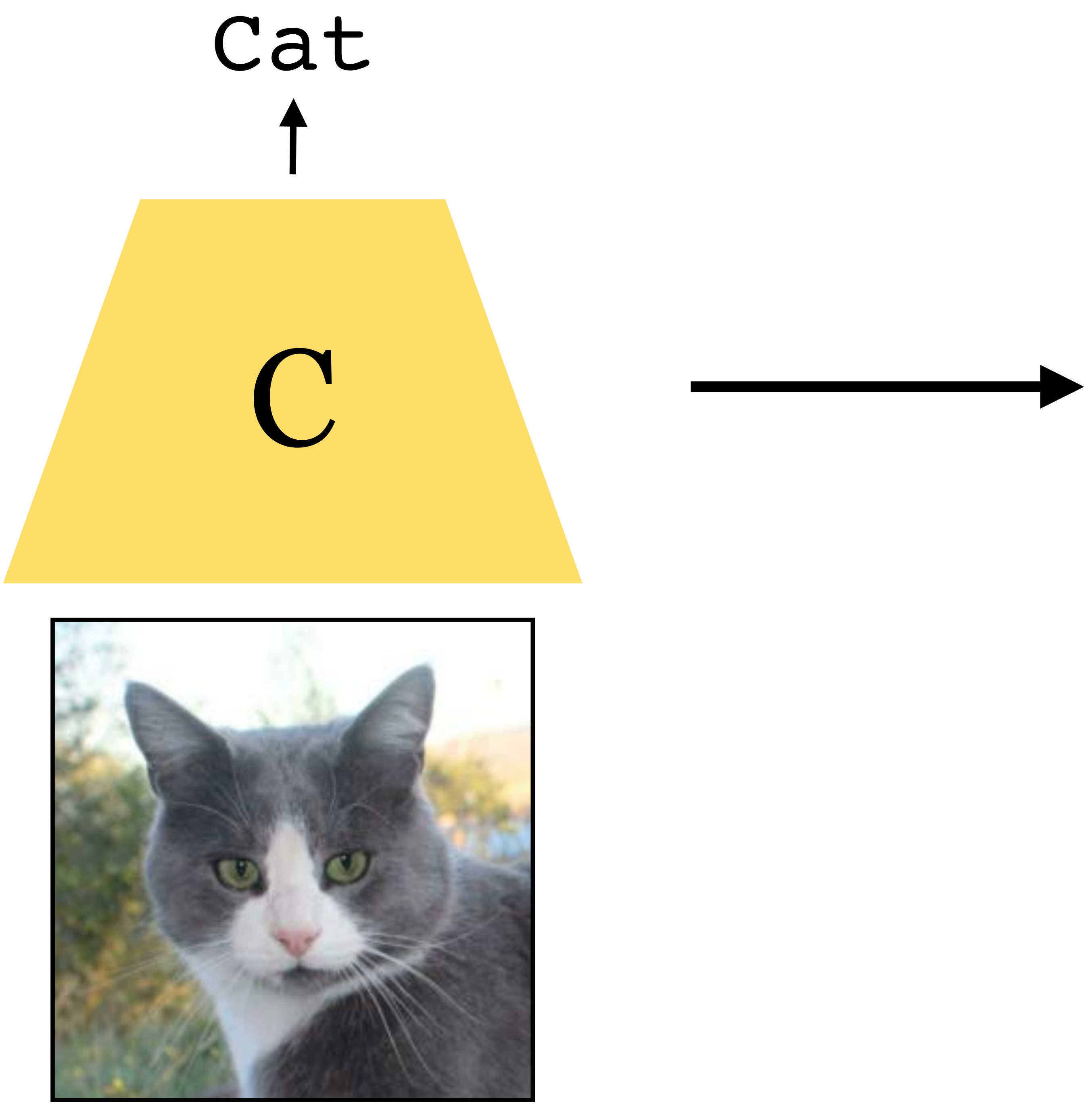
[Li, Yang, Kries, Torralba, Fidler, CVPR 2021]

[Tritrong*, Rewatbowornwong*, Suwajanakorn, CVPR 2021]

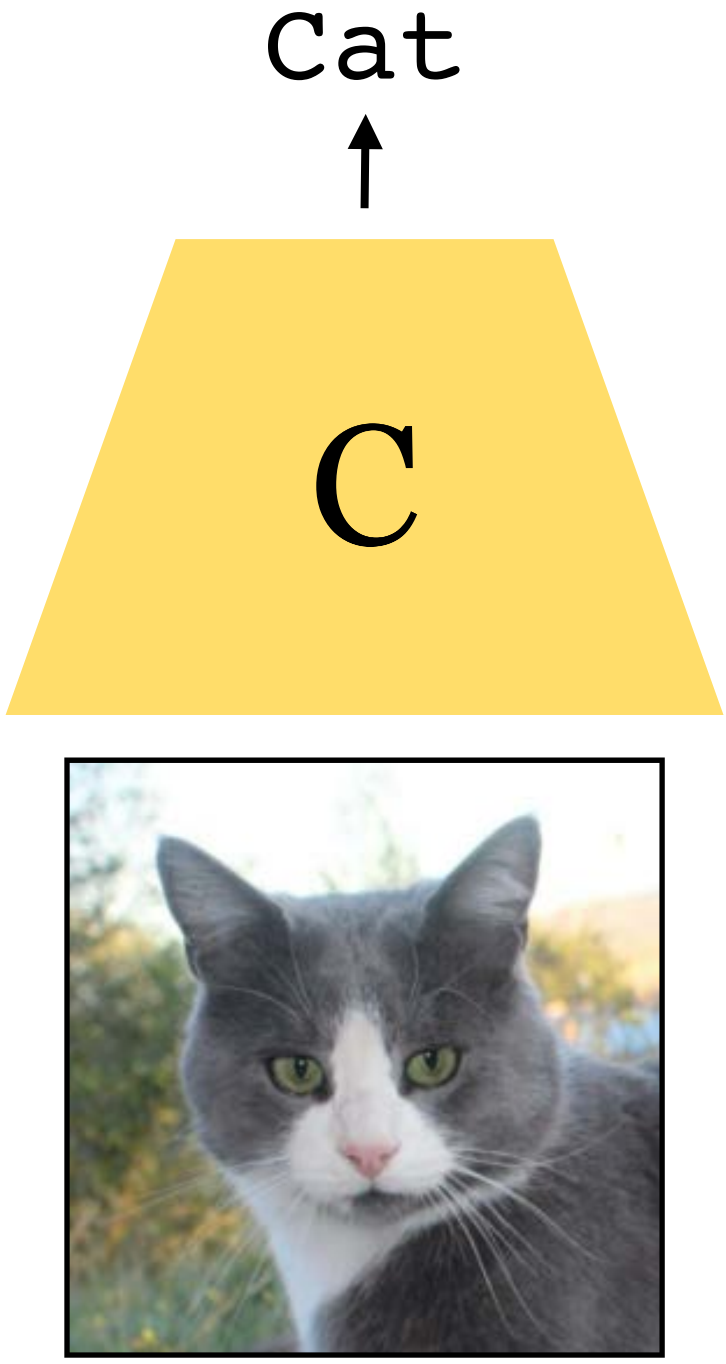
Explaining in Style: Training a GAN to Explain a Classifier

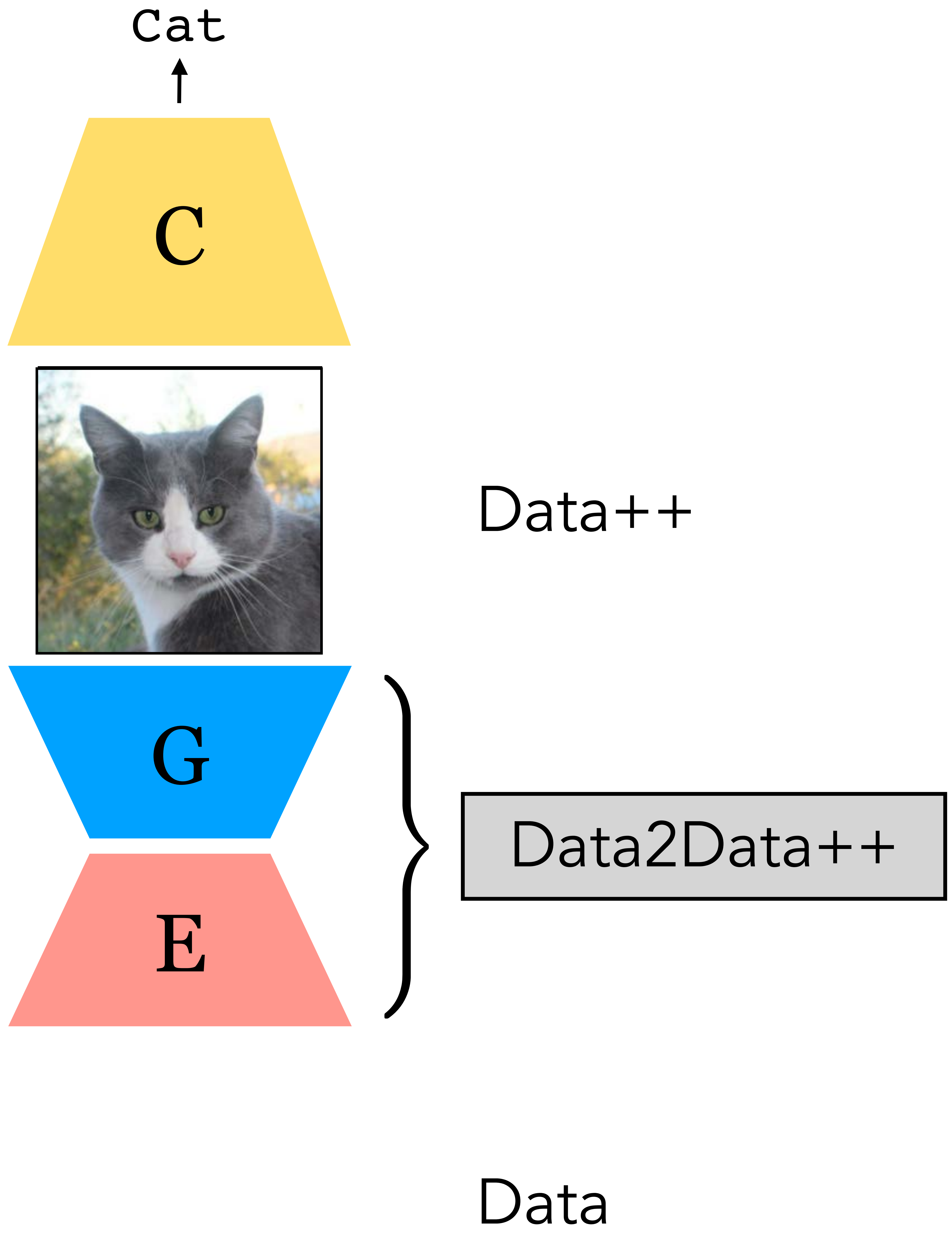
Oran Lang*, Yossi Gandelsman*, Michal Yarom*, Yoav Wald*, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, Inbar Mosseri

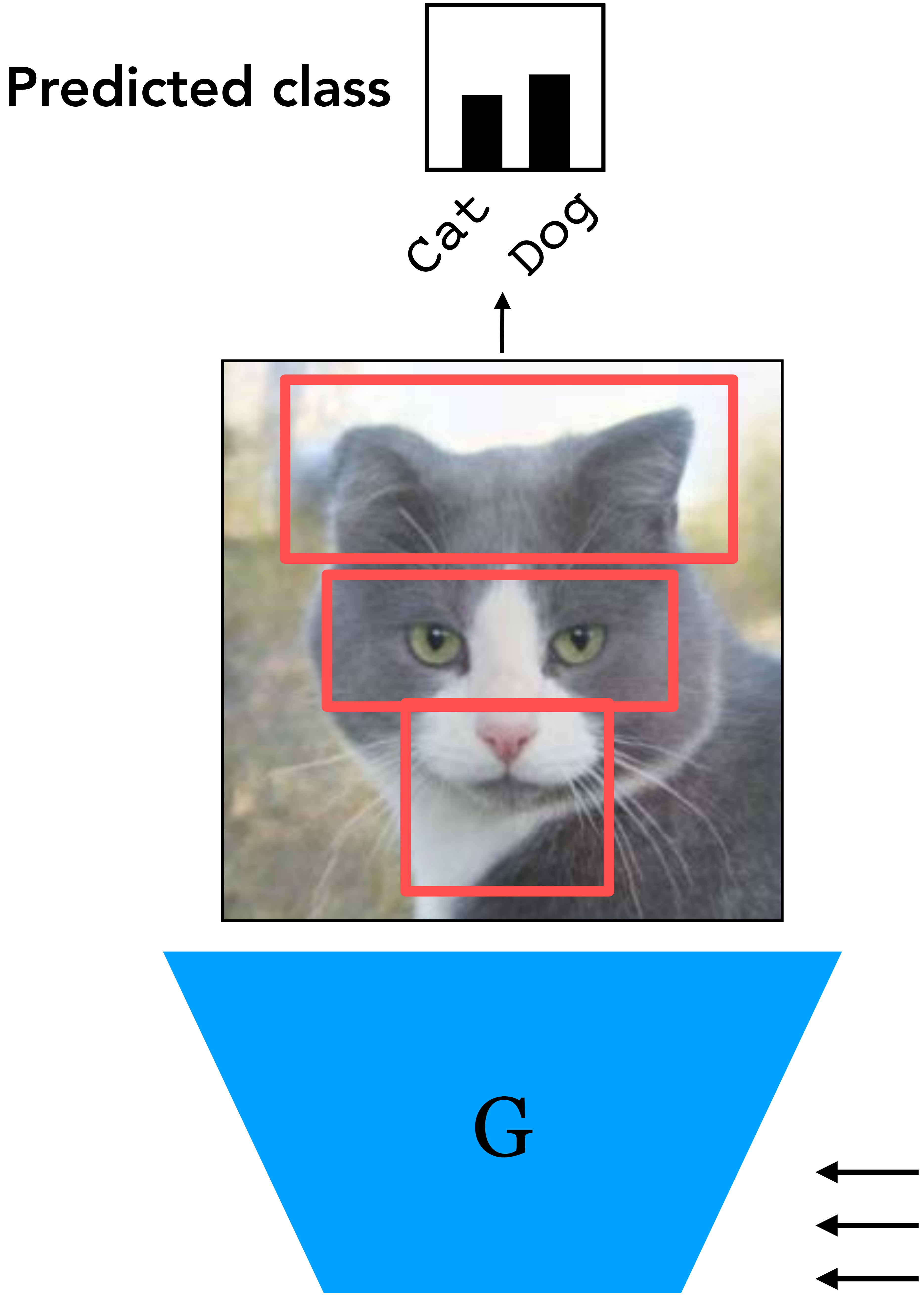
ICCV 2021



Why was this image classified as a cat?


Cat
C


Cat
C
G
E
Data++
Data2Data++
Data



StylEx

Find top-K StyleSpace directions that most affect predicted class

Manipulate latent variables

["StyleSpace": Wu, Lischinski, Shechtman 2020]

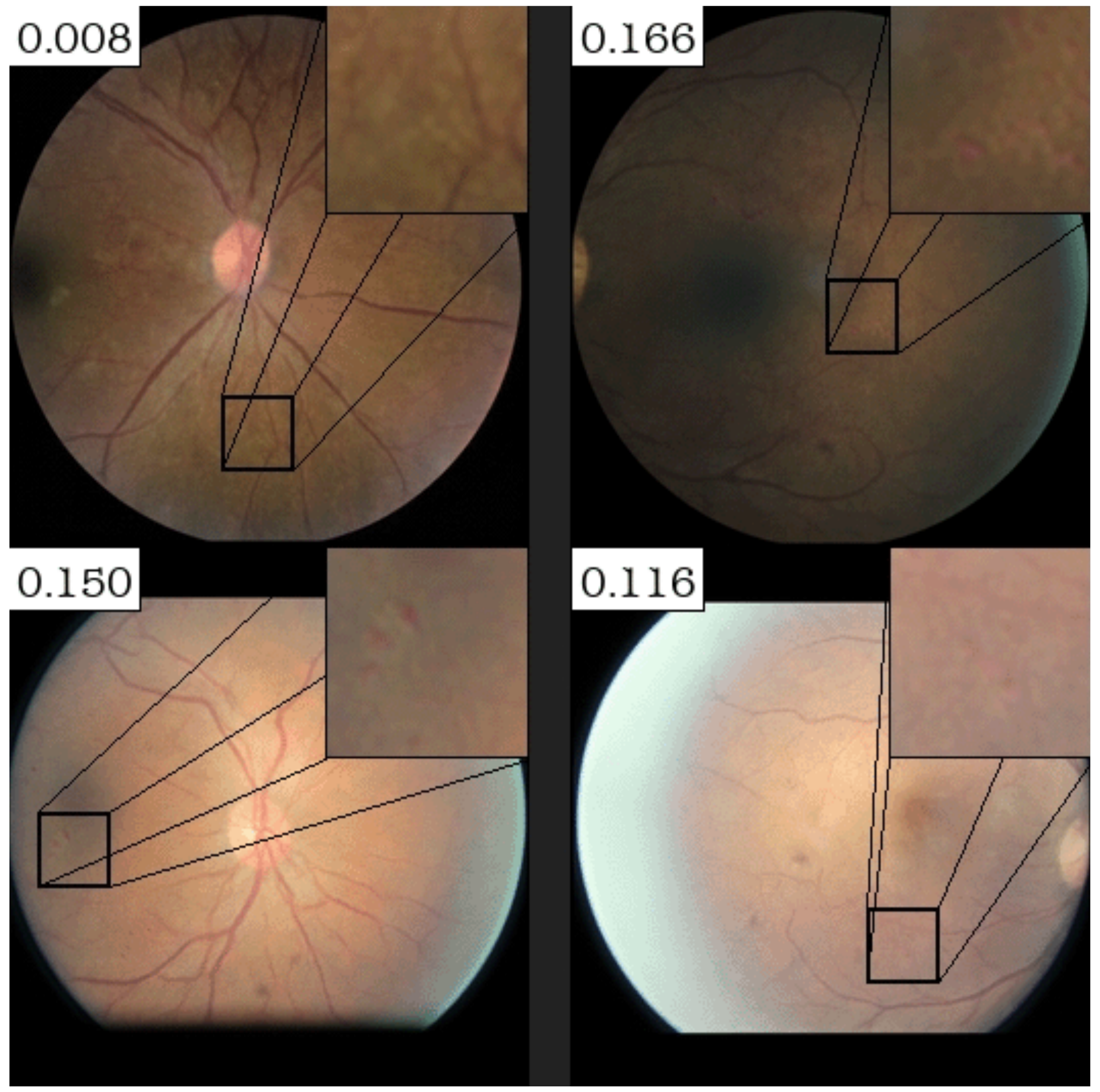
Class-specific explanation

Perceived Age Classifier:

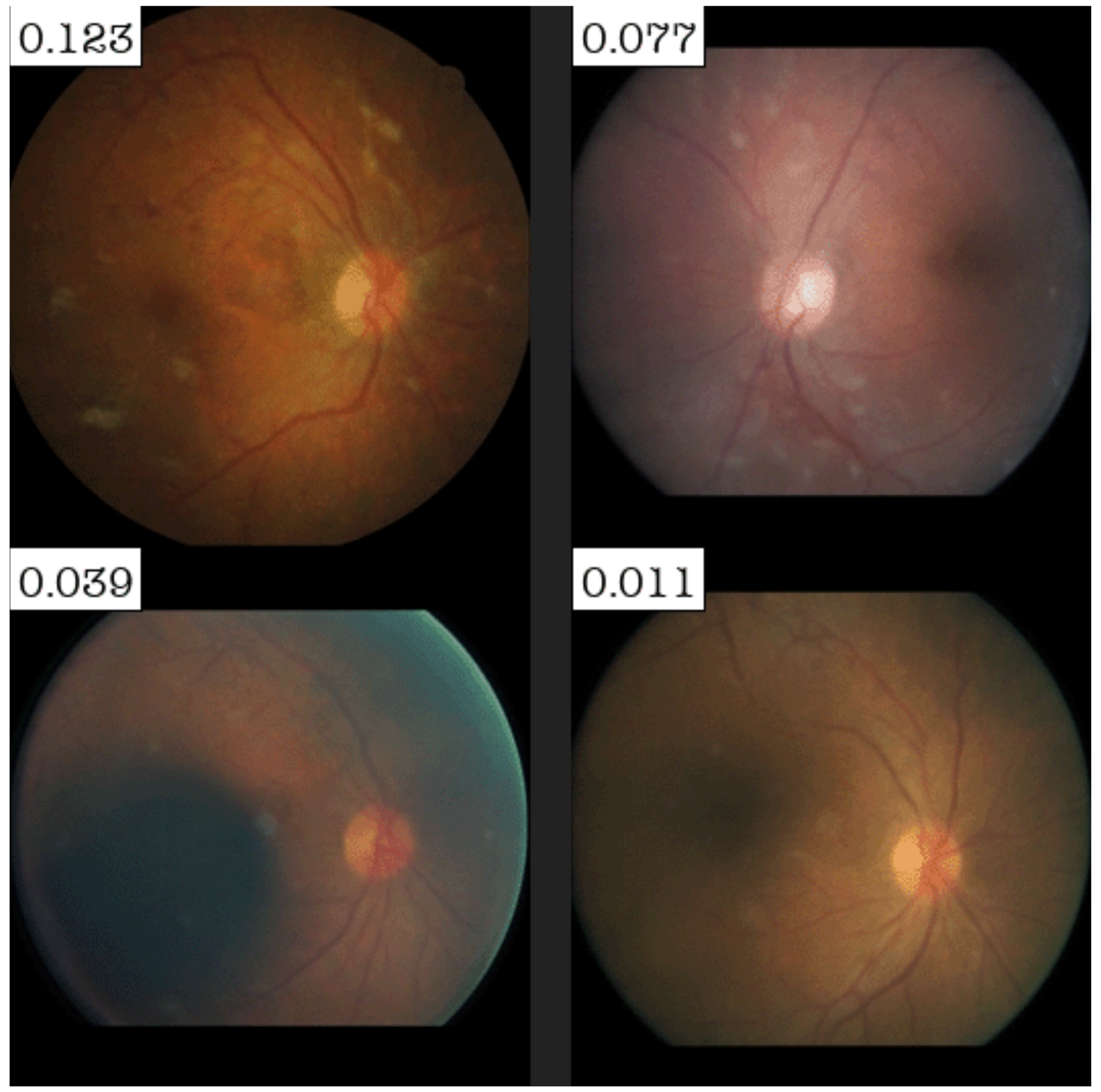
Image of people removed due to copyright restrictions.

Class-specific explanation

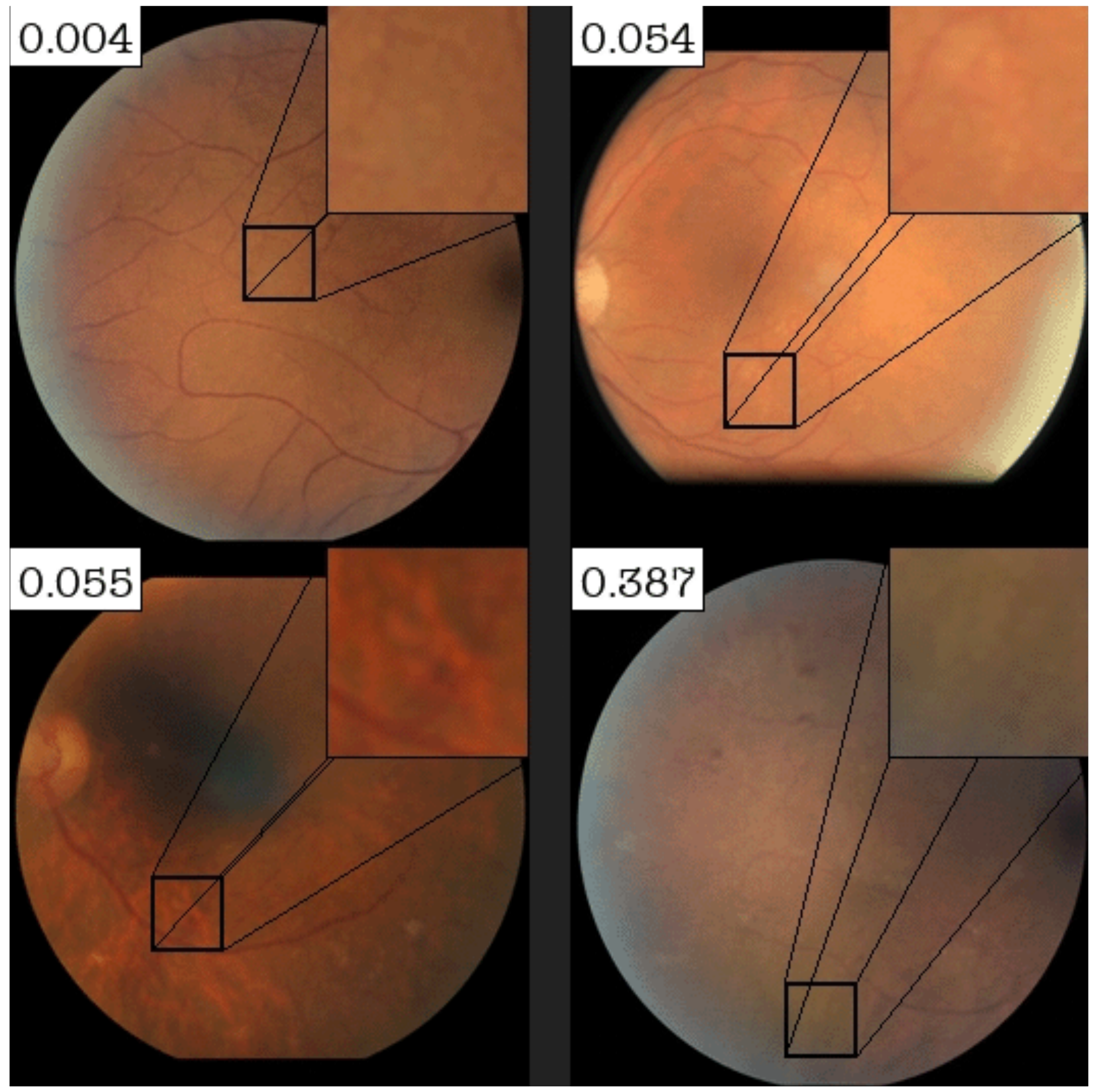
Retinal Fundus Classifier:



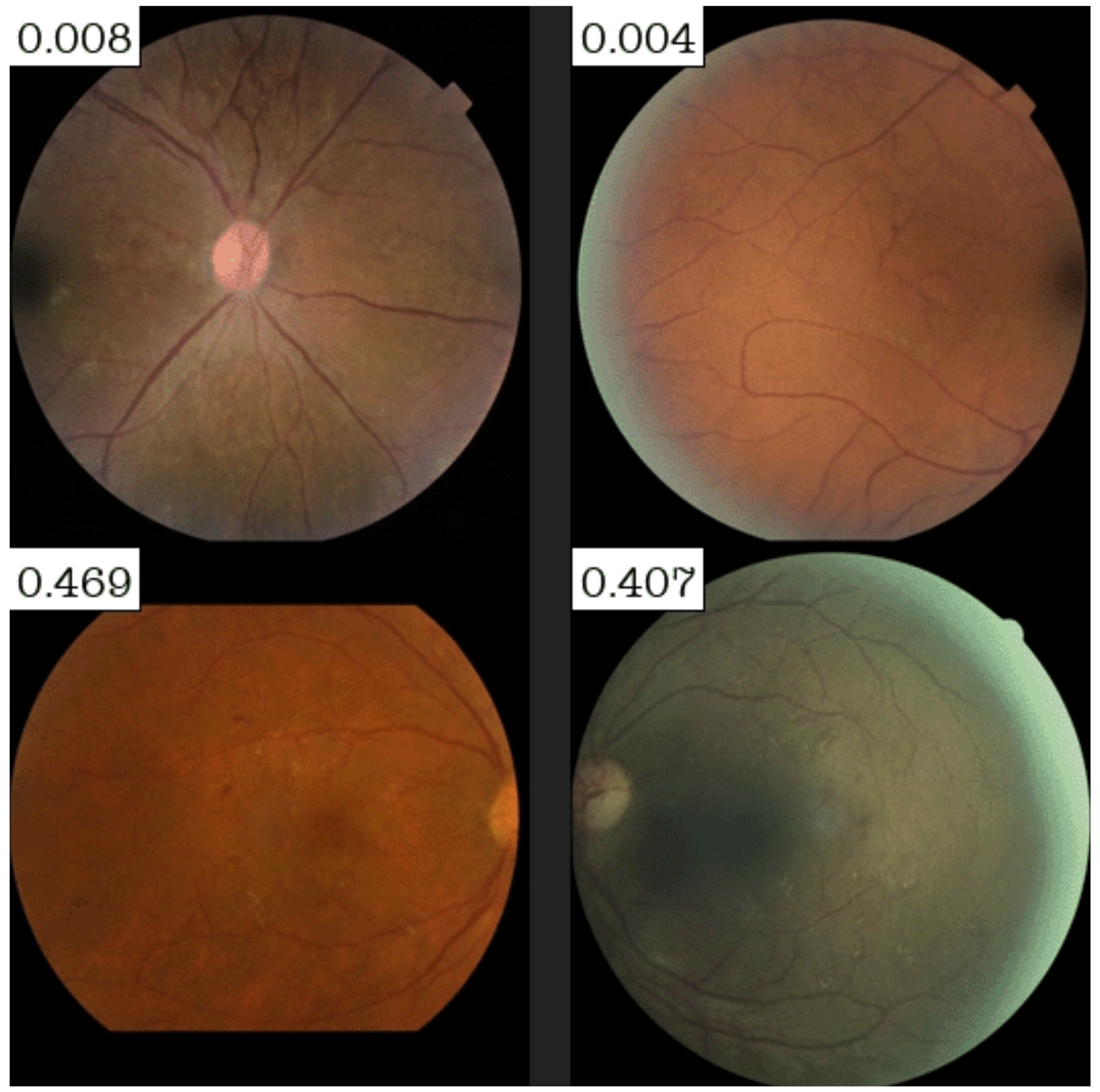
Attribute #1
“Exudates”



Attribute #2
“Cotton Wool”



Attribute #3
“Hemorrhages”



Attribute #4
“Clustered Exudates”

Original Plates

ImageNet

Acquiring images of plates with utensils

Bing

Stable Diffusion

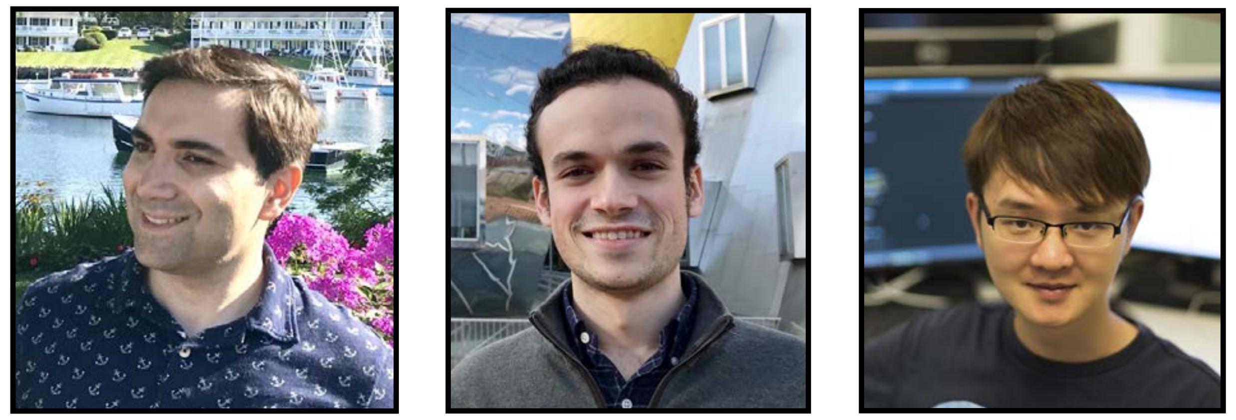
ImageNet*

tray | tray | tray | plate | plate | plate

tray | bucket | bucket | plate | plate | plate

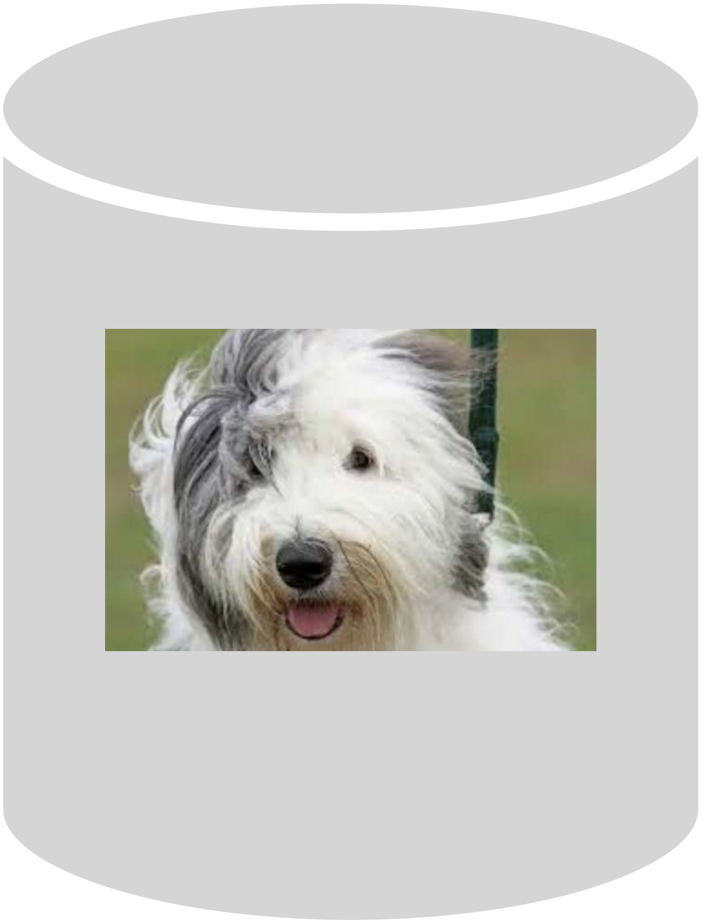
Dataset Interfaces: Diagnosing Model Failures Using Controllable Counterfactual Generation, Vendrow et al., 2023

Generative Models as a Data Source for Multiview Representation Learning



Jahanian, Puig, Tian, Isola
ICLR 2022

Dataset

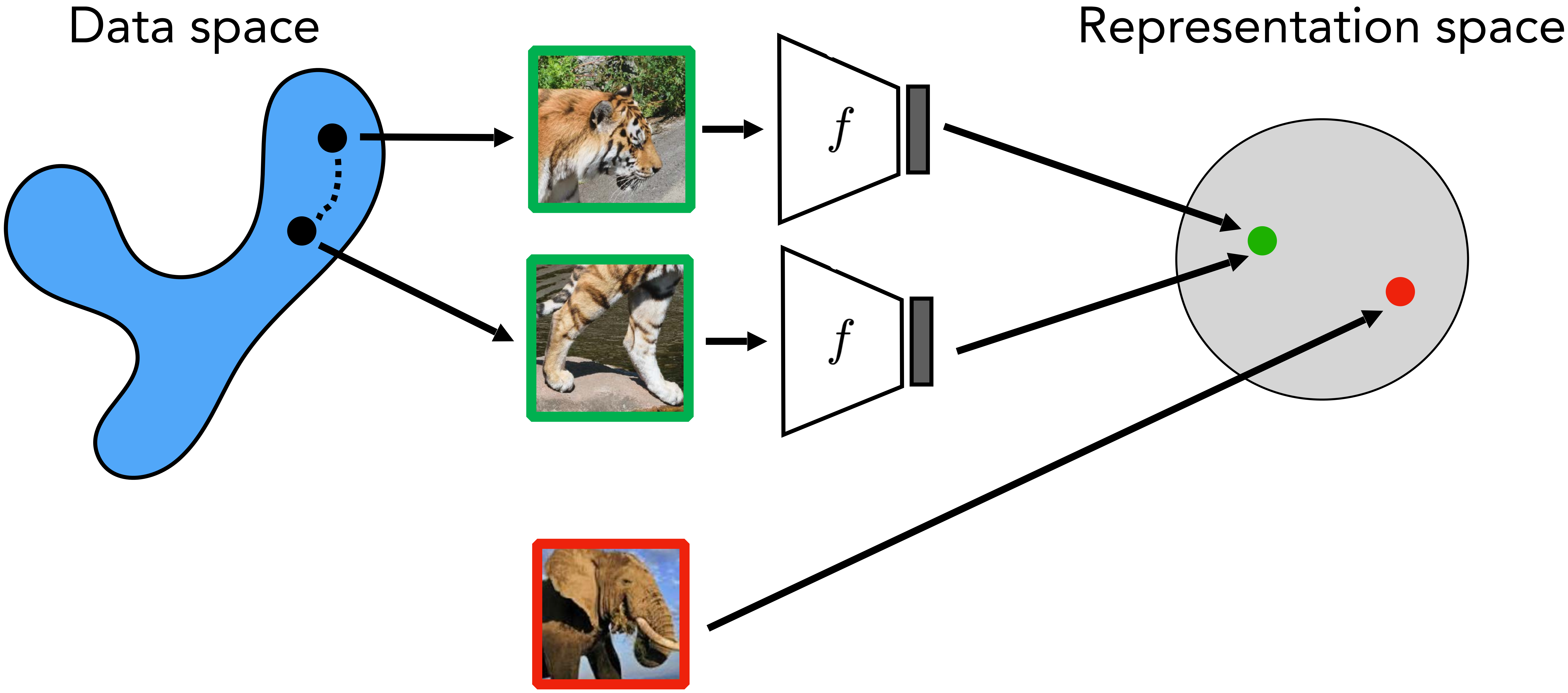


$$\mathbf{x} \sim \{\mathbf{x}_i\}_{i=1}^{N}$$

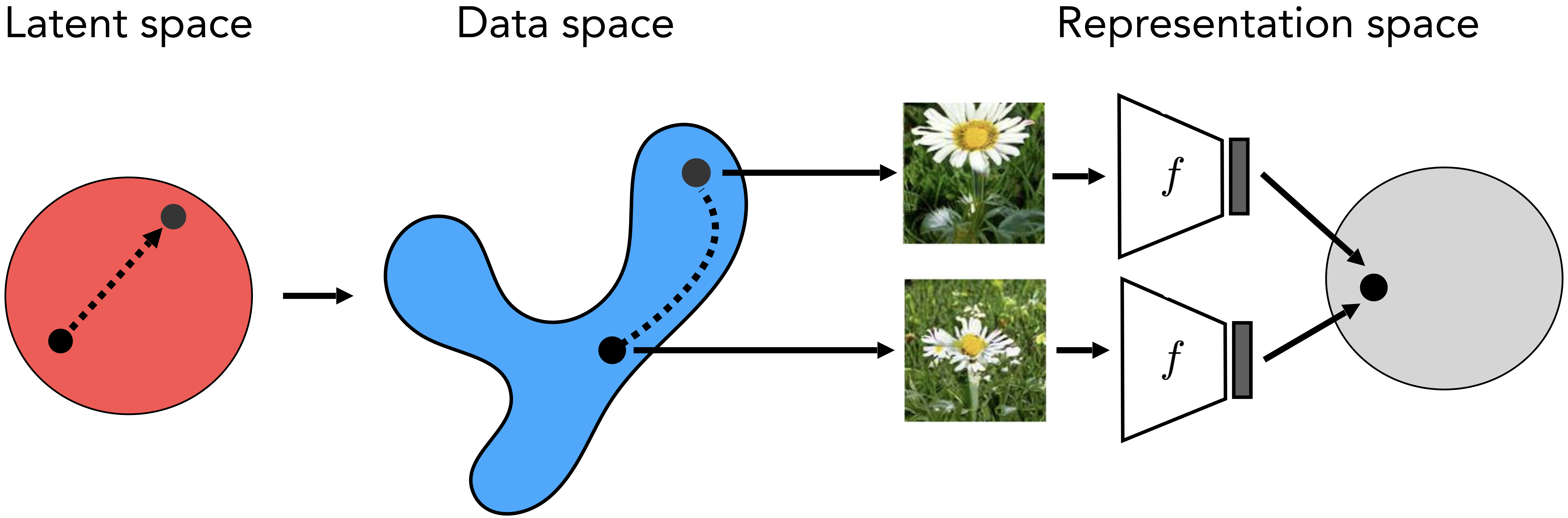
Generative Model

$$\mathbf{x} \sim G(T_{\mathbf{z}}(\mathbf{z}))$$

Contrastive Learning



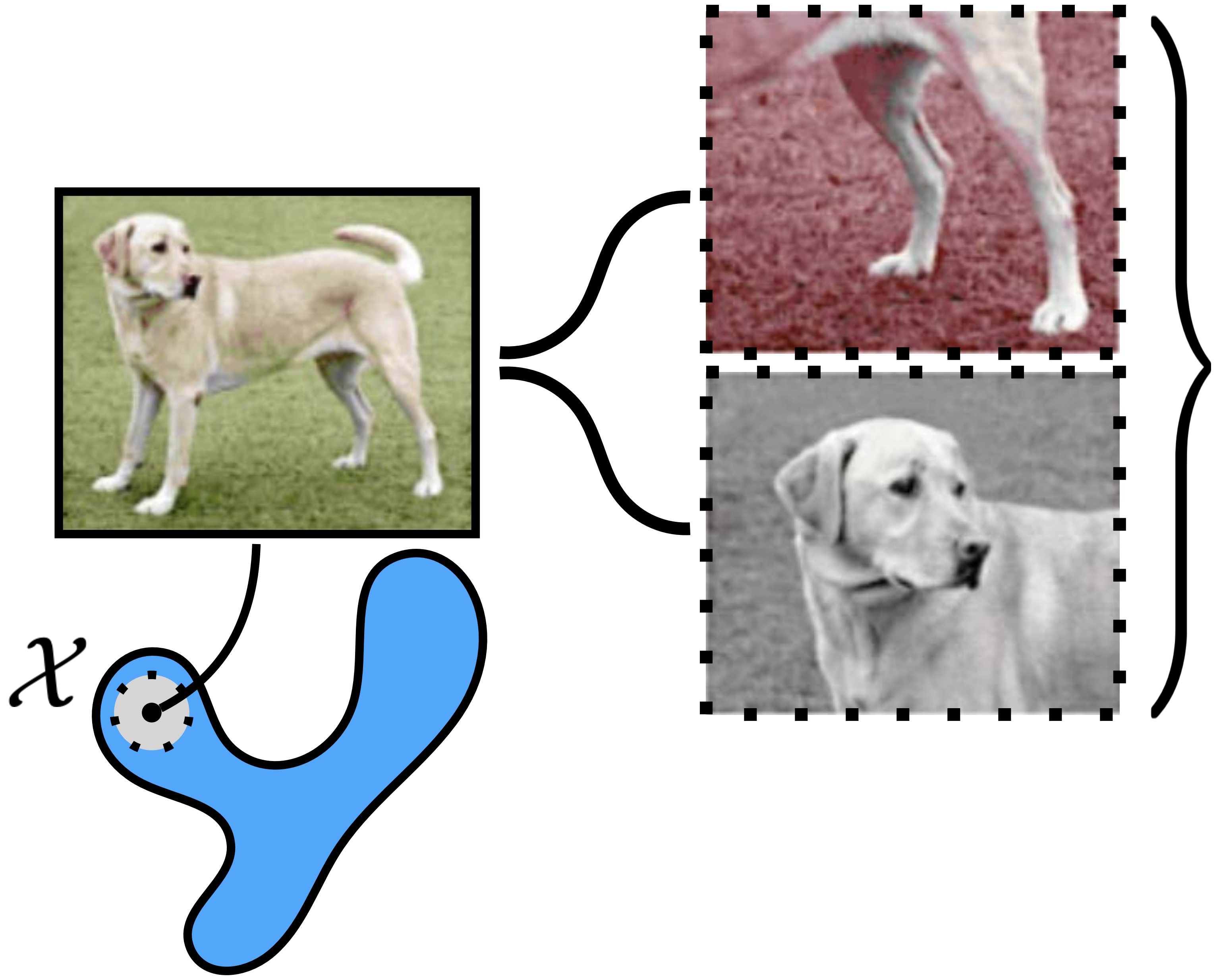
Contrastive learning + Generative modeling



Create positive pairs by transformations in latent space, rather than in data space

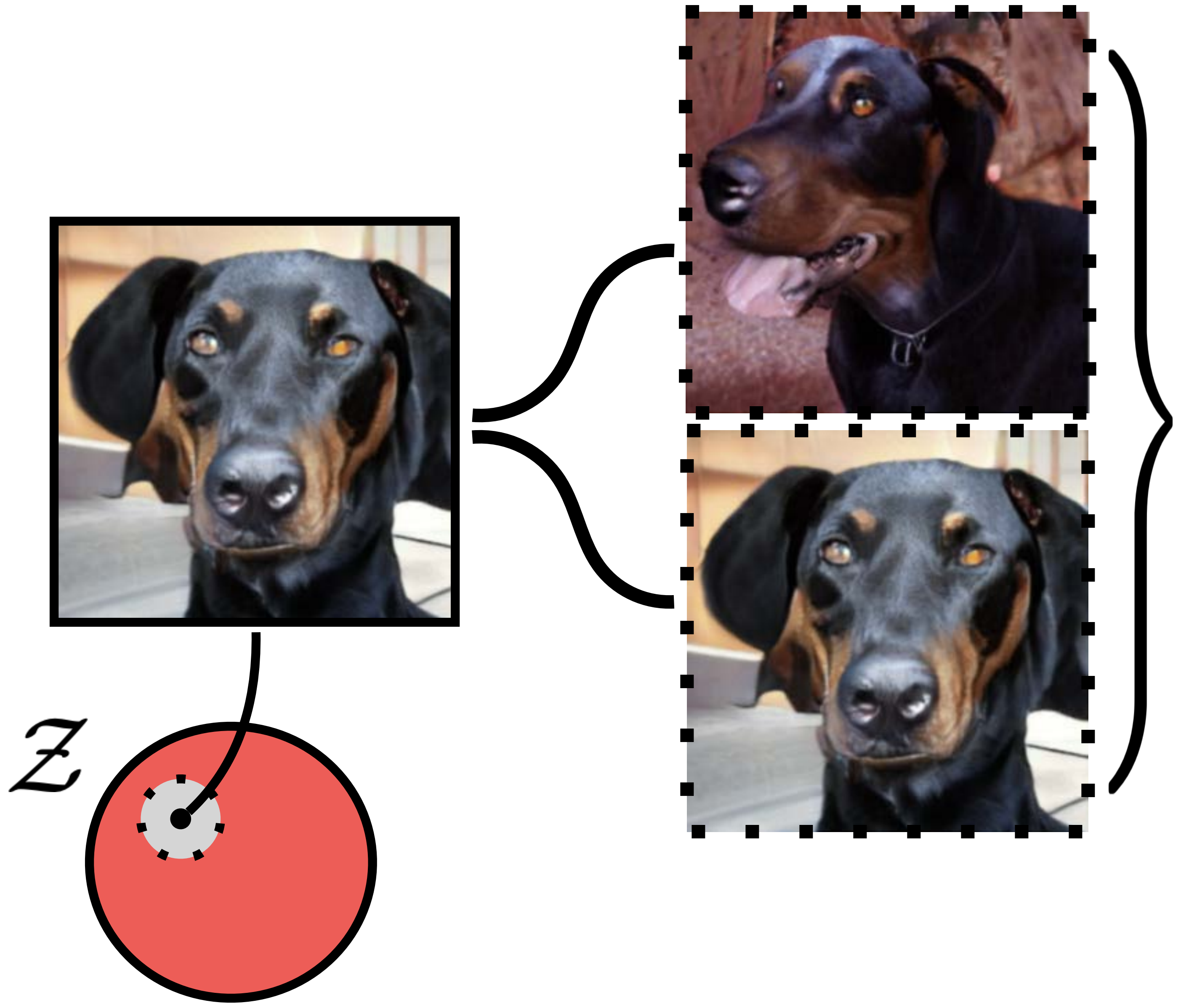
Contrastive learning + Generative modeling

SimCLR views



Objective: these are two views of the same thing

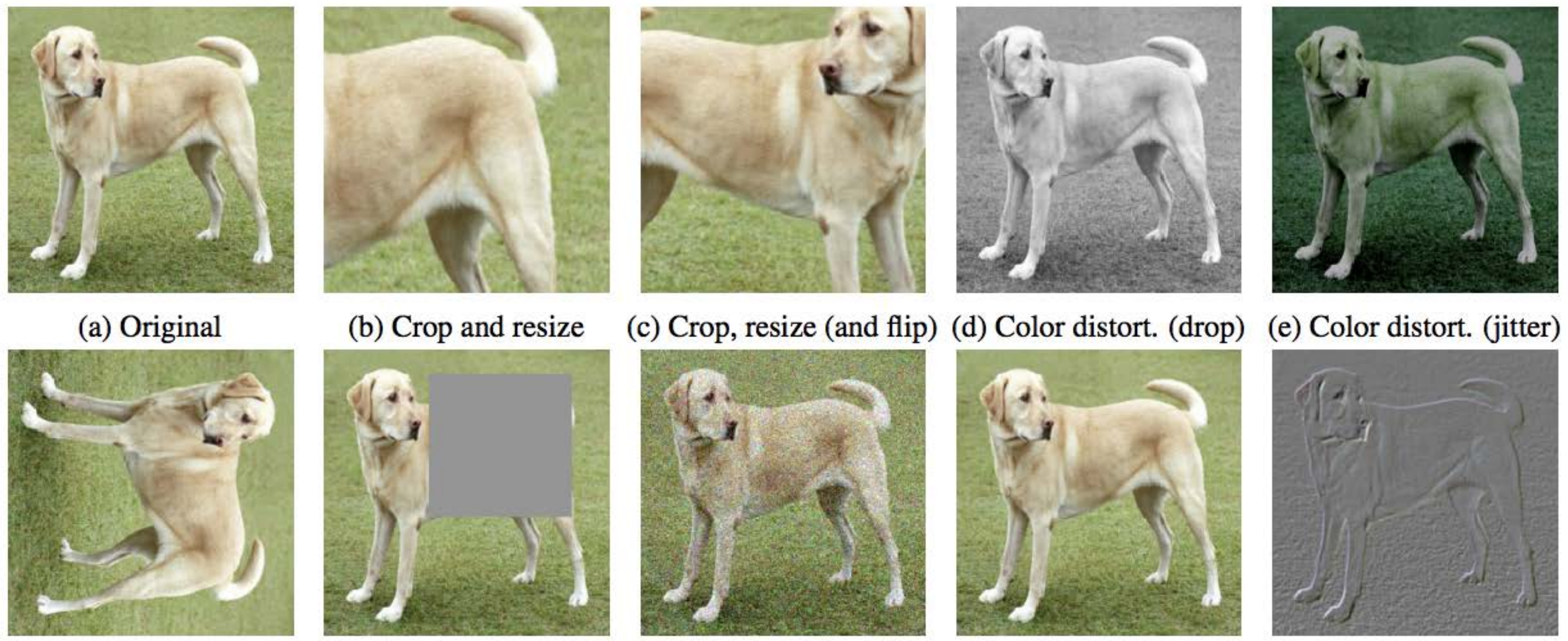
Latent views



Objective: these are two views of the same thing

Contrastive learning + Generative modeling

SimCLR views



These are all different views of the same thing

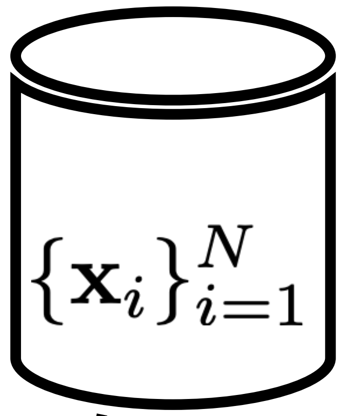
Latent views

These are all different views of the same thing

Contrastive learning + Generative modeling

Top-1 Accuracy on ImageNet1000 linear transfer

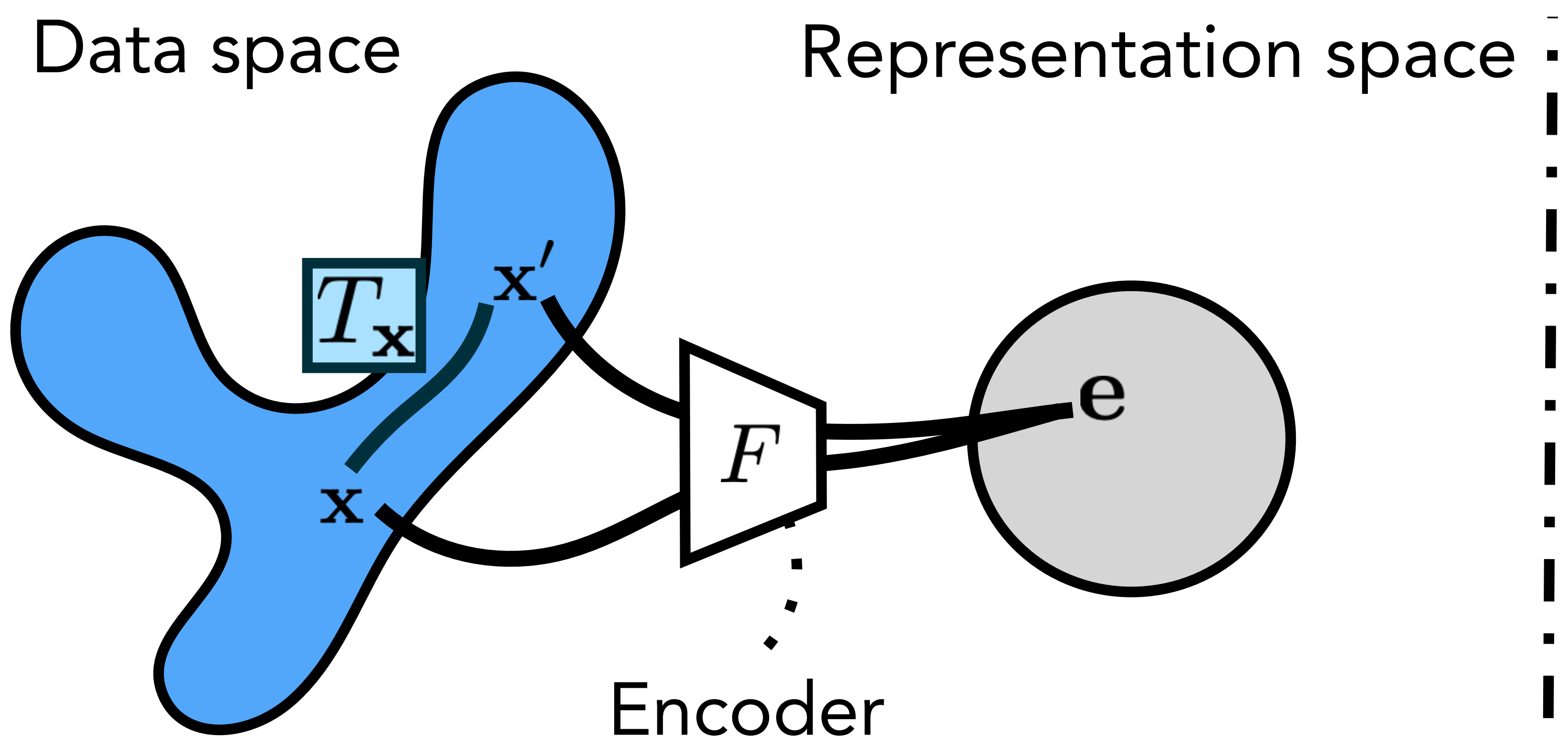
Contrastive learning from real data (SimCLR, etc)

Given: 

, $T_{\mathbf{x}}$

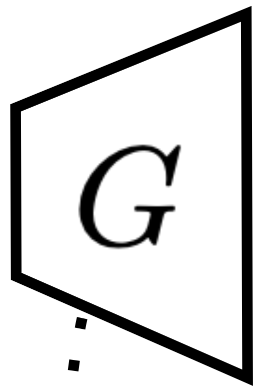
Dataset

Learn: F



43.9%

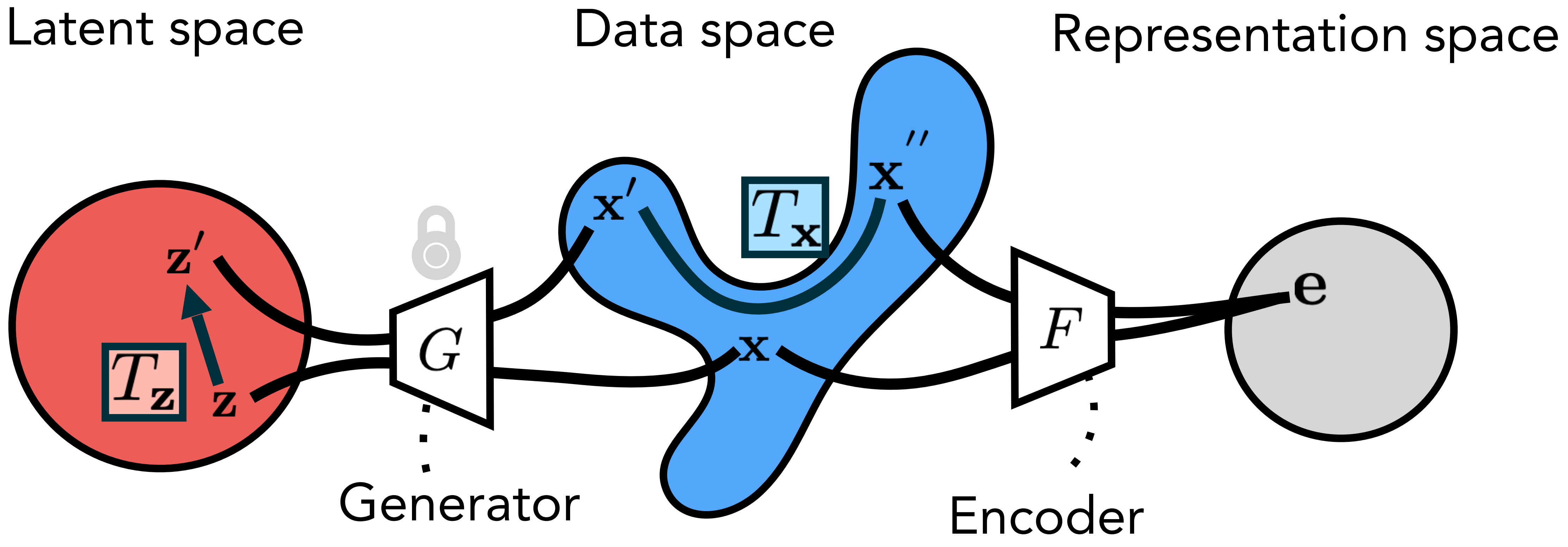
Contrastive learning from generated data

Given: 

, $T_{\mathbf{z}}$, $T_{\mathbf{x}}$

Generator

Learn: F



Only $T_{\mathbf{x}} \longrightarrow$ 35.7%

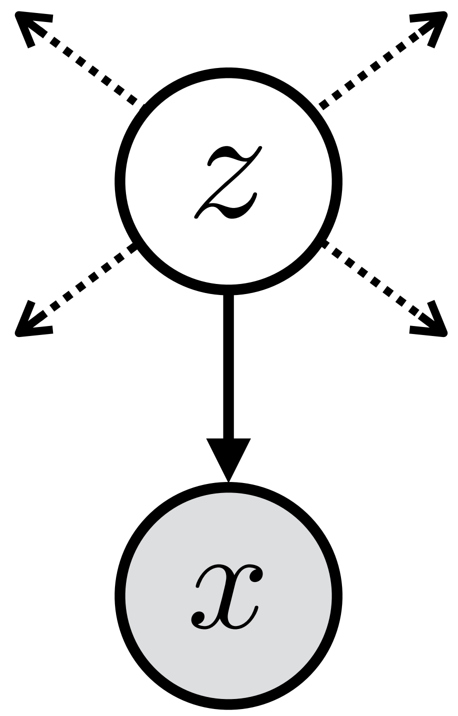
$T_{\mathbf{z}}$ + $T_{\mathbf{x}} \longrightarrow$ 42.6%

Contrastive learning + Generative modeling

Deep generative views improve contrastive learning beyond only using standard data augmentation.

When the generative model is high quality, can even outperform learning from real data.

Data++



Datapoint++

Samples from implicit generative models act like decorated data, *data++*, with extra functionality.

With special operators, you can *interpolate*, *extrapolate*, *manipulate*, *compose*, *optimize*, and *label* these datapoints.

Everything you can do with regular data, you can also try with data++. It might work better!

Meta-learning: learning to learn fast

MAML: Model-Agnostic Meta-Learning

[Finn, Abbeel, Levine 2017]

- So far we saw transfer learning as: given some prior model/representation, how can we quickly adapt it?
- Could we *learn* to do transfer learning?
- Consider finetuning: given an init, update it with SGD
- MAML: *learn* an init such just a few steps of SGD will update it very effectively

MAML: Model-Agnostic Meta-Learning

[Finn, Abbeel, Levine 2017]

Algorithm 1 Model-Agnostic Meta-Learning

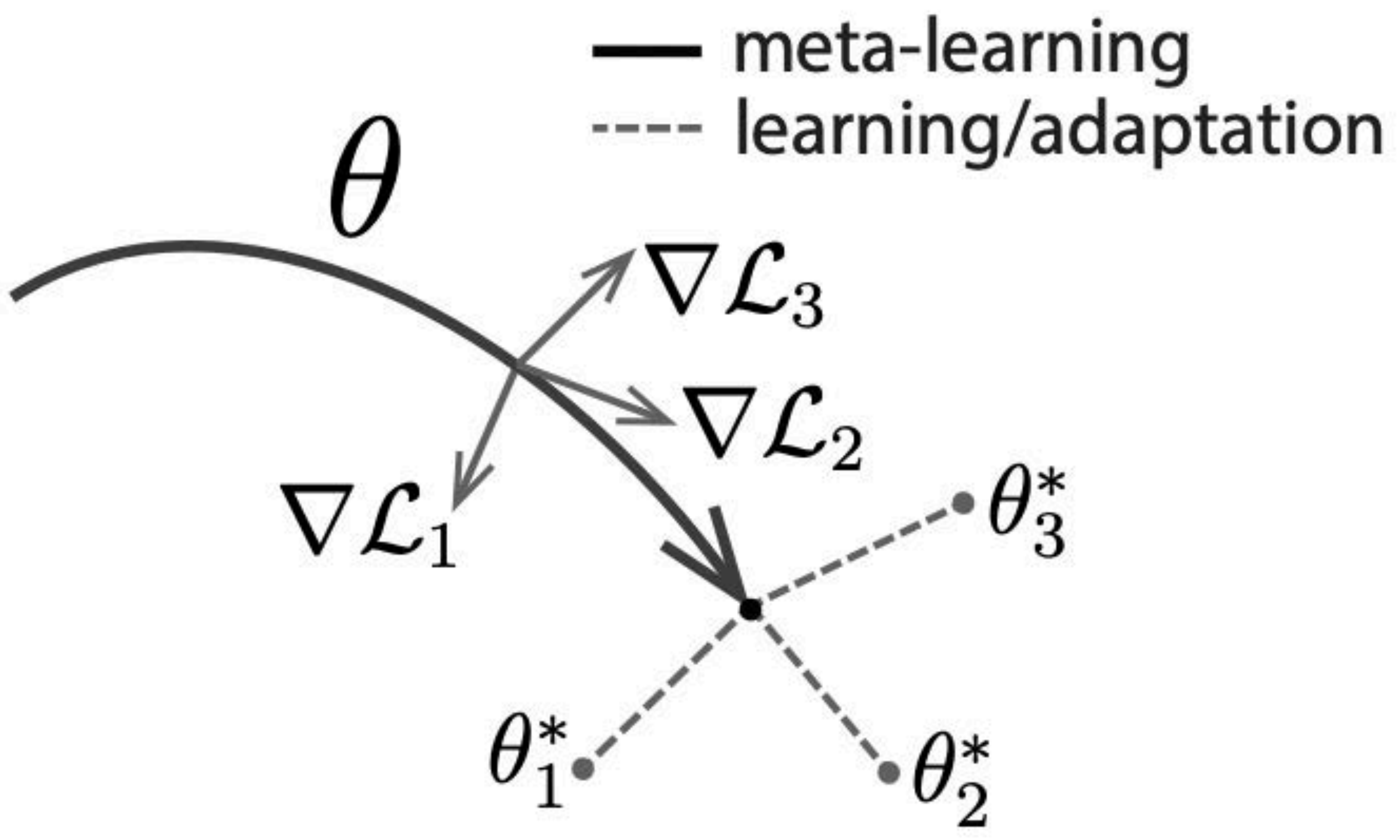
Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**

1. Starting with init, run a few steps of SGD on task i
2. Compute loss after these steps

  end for

init <— backprop through

end while



MAML: Model-Agnostic Meta-Learning

[Finn, Abbeel, Levine 2017]

init <— backprop through

How to backprop through ?

1 step SGD:

= `forward`(x, `backward`(`forward`(x, θ)) + θ)

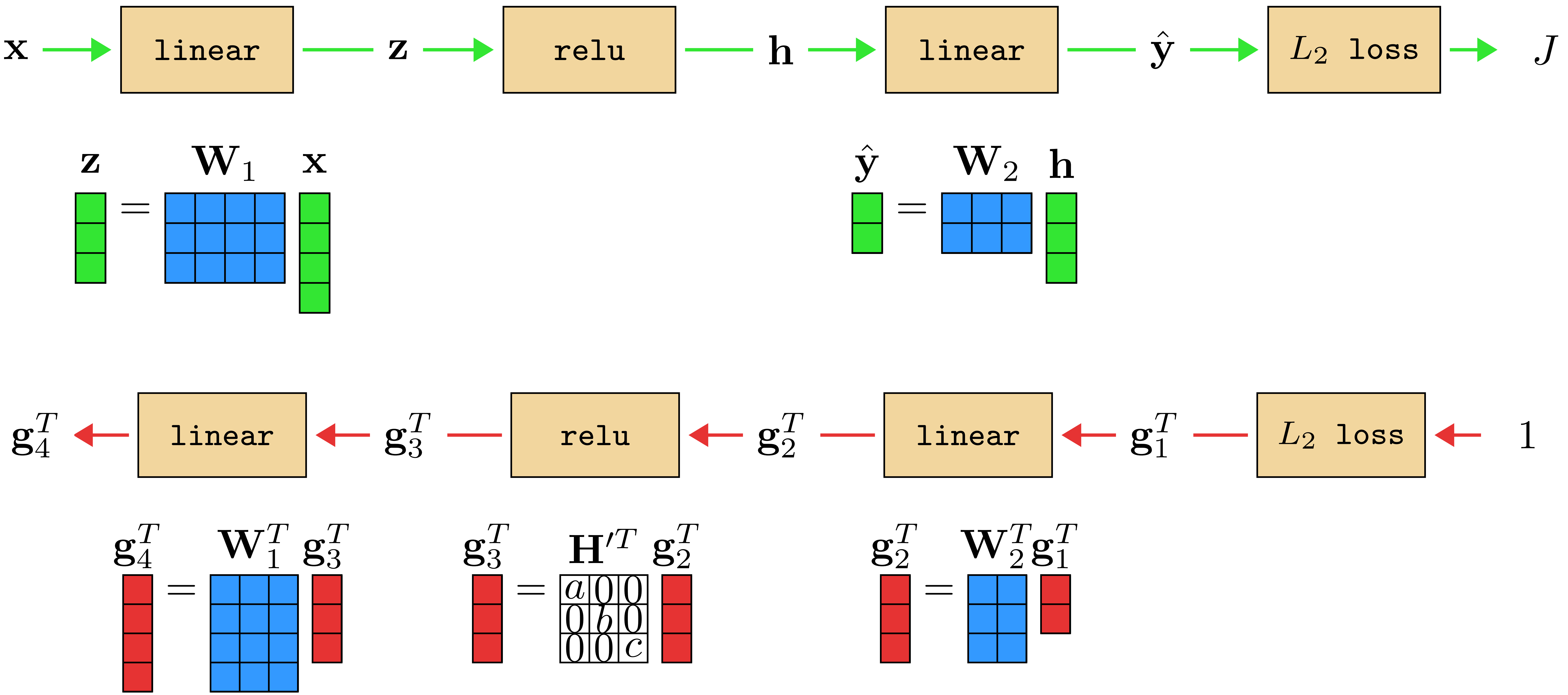
2 steps SGD:

= `forward`(x, `backward`(`forward`(x, `backward`(`forward`(x, θ)) + θ), θ) + θ)

You can backprop through backward!

forward(x, backward(forward(x, backward(forward(x, θ)) + θ), θ) + θ)

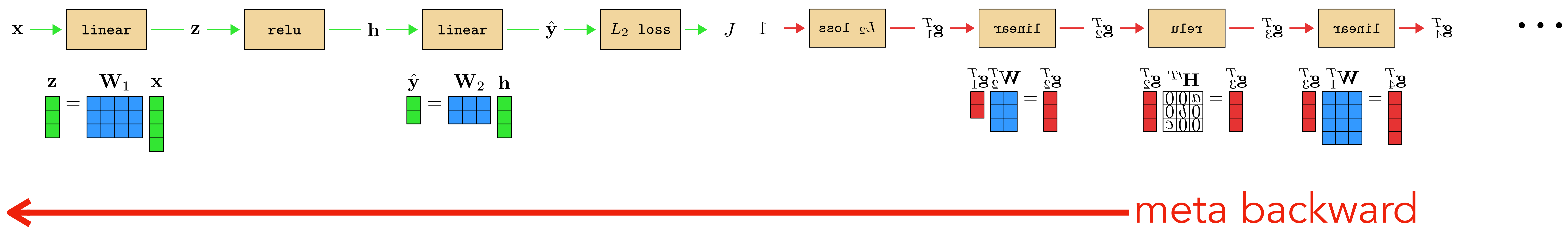
Example forward and backward through a simple net:



You can backprop through backward!

forward(x, backward(forward(x, backward(forward(x, θ)) + θ), θ) + θ)

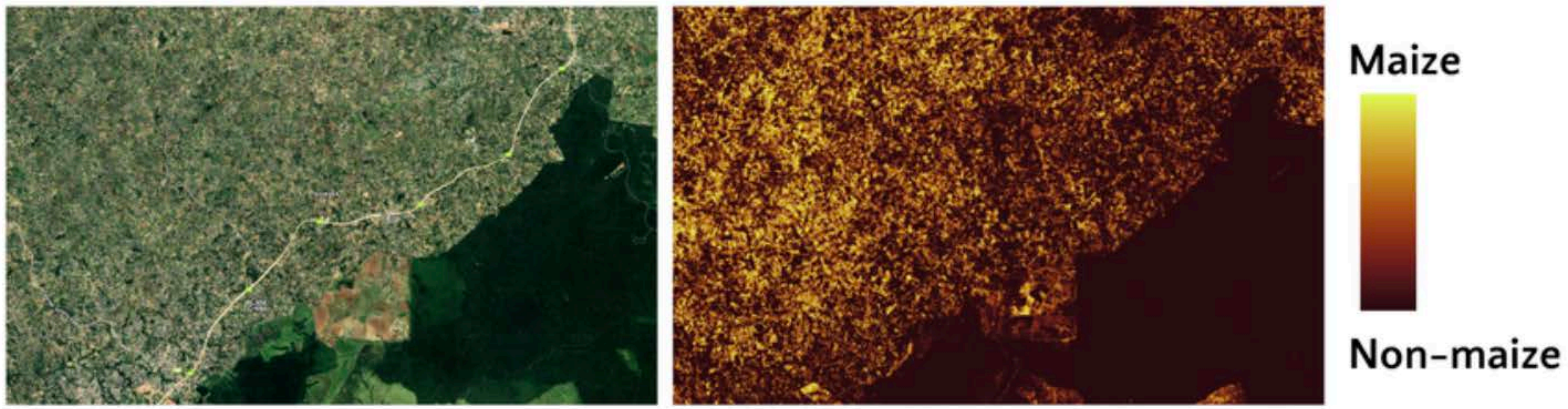
Example forward and backward through a simple net:

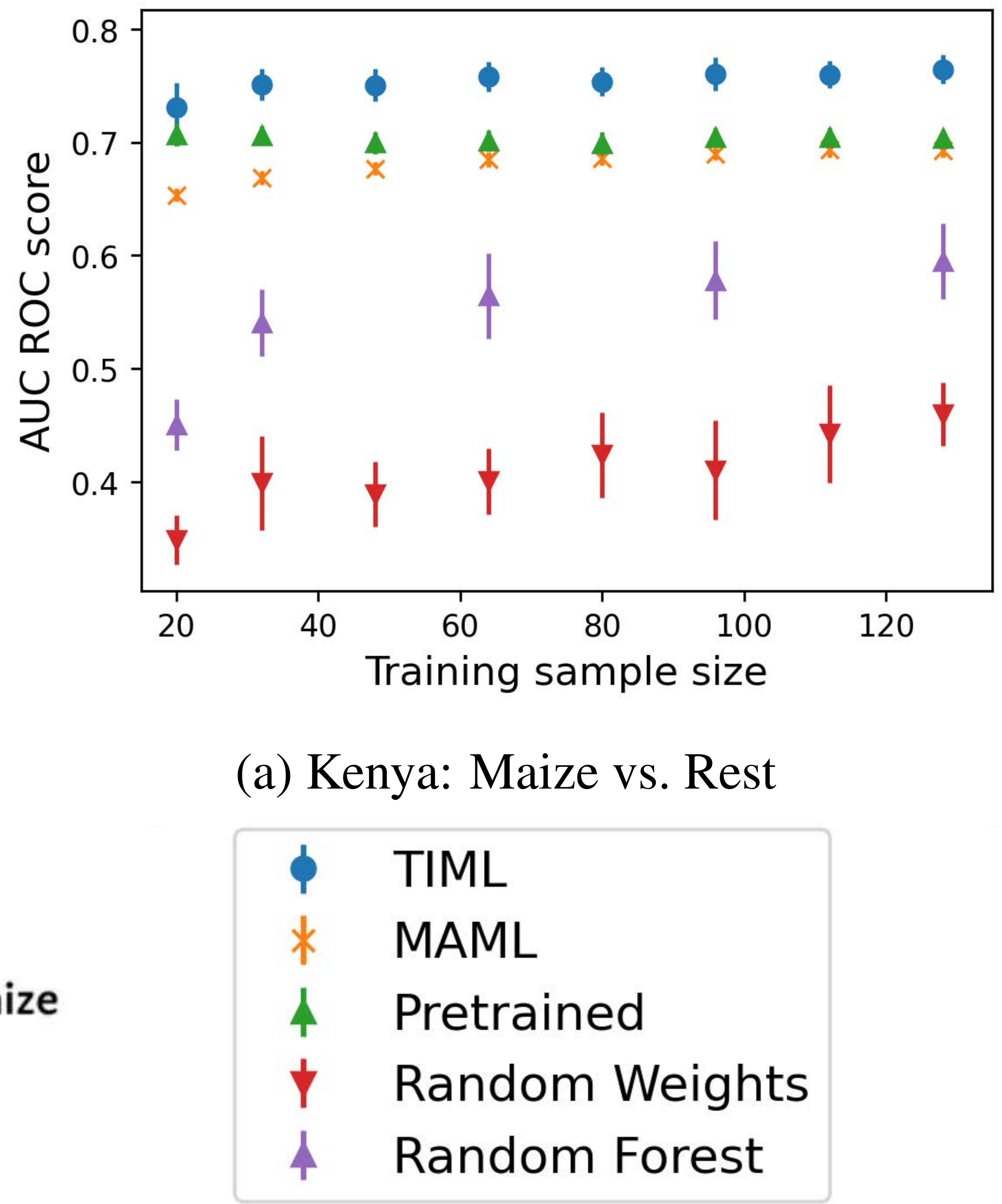


Task-Informed MAML for Agriculture

- In practice, the gains of this method are not often worth the additional computational cost at training time (worse than just pretraining?)

- Tseng et al. introduce a geospatial embedding that encodes task-specific context into meta-learning, and helps performance





(a) Kenya: Maize vs. Rest

TIML: Task-Informed Meta-Learning for Agriculture, Tseng et al., 2022

Meta-learning by sequence modeling

- Definition of learning:
  - input: a sequence x0, y0, x1, y1, x2, y2, …
  - output: a function f: x—> y
- x0, y0, x1, y1, x2, y2, x3, ?
- RNN state + params is f, apply that f to x3

Transfer Learning II

- Transferring knowledge about the inputs
  - Generative models as data++
- Meta-learning