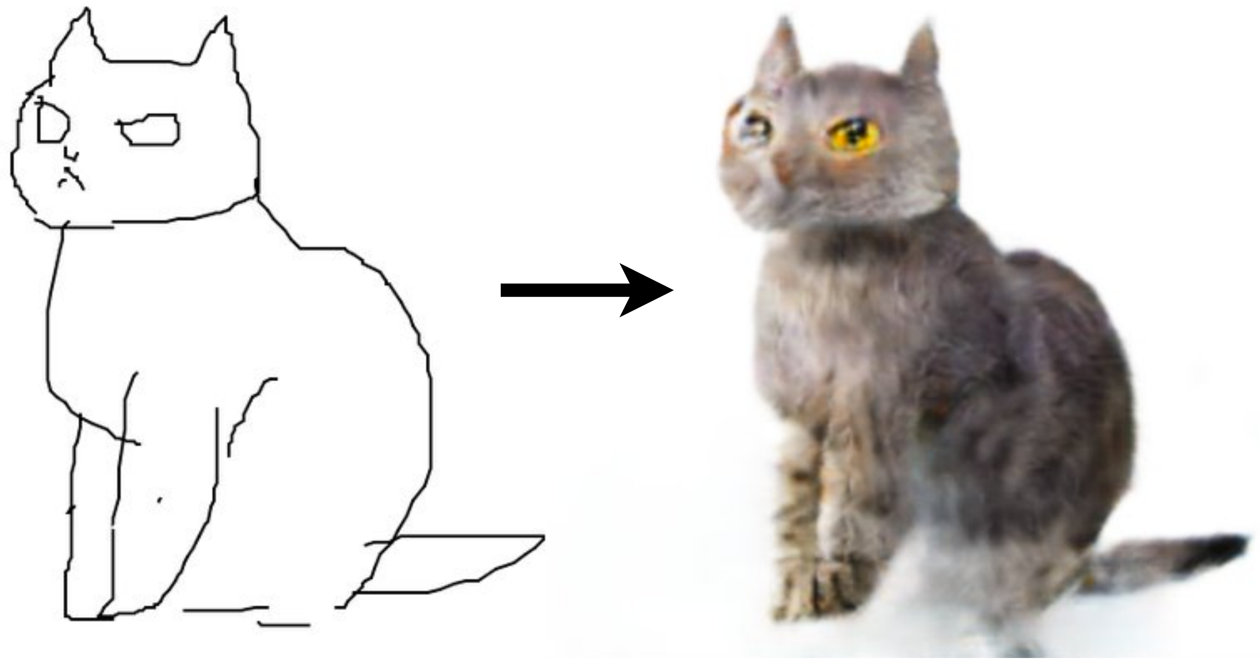
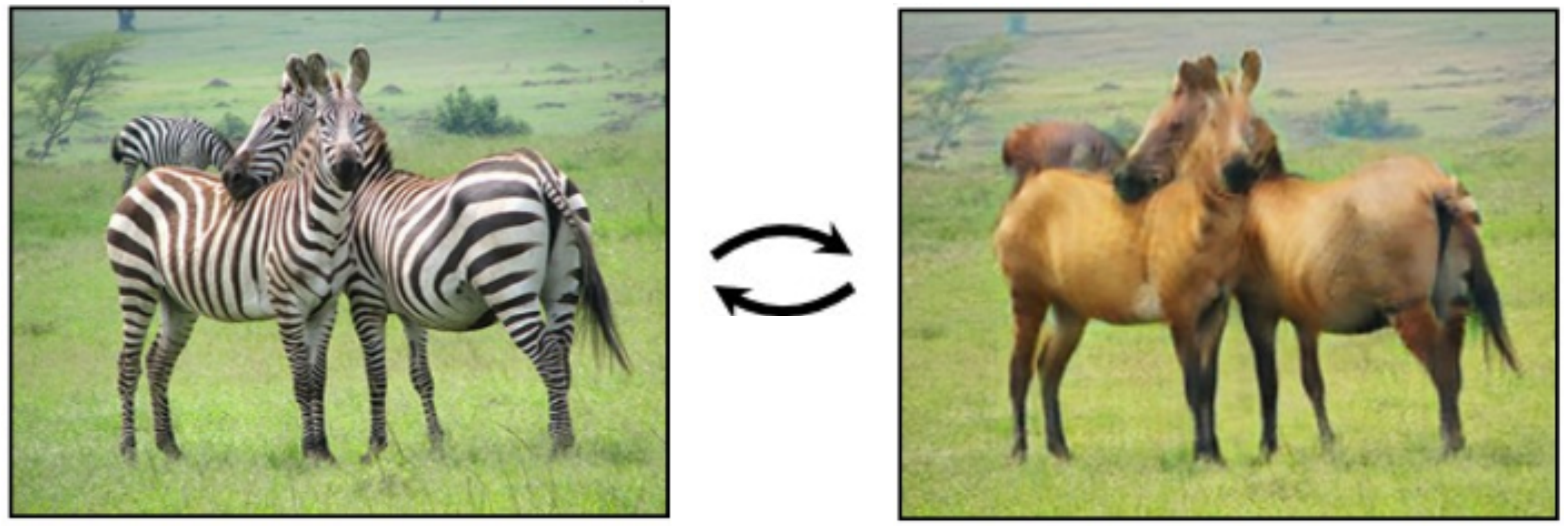
$z \sim \mathcal{N}(\vec{0}, 1) \longrightarrow$





Generative Adversarial Networks

Phillip Isola

9.520

10/17/18

Image classification

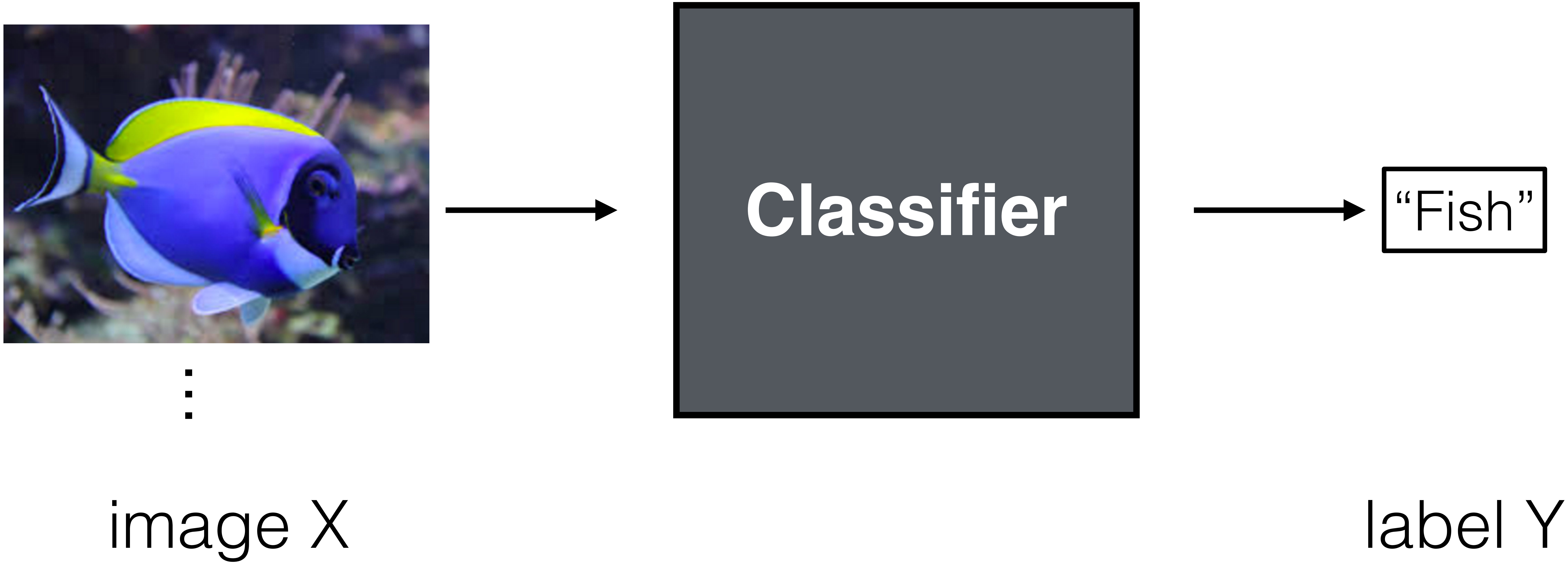


Image generation

"Fish" → **Generator** → 

label Y

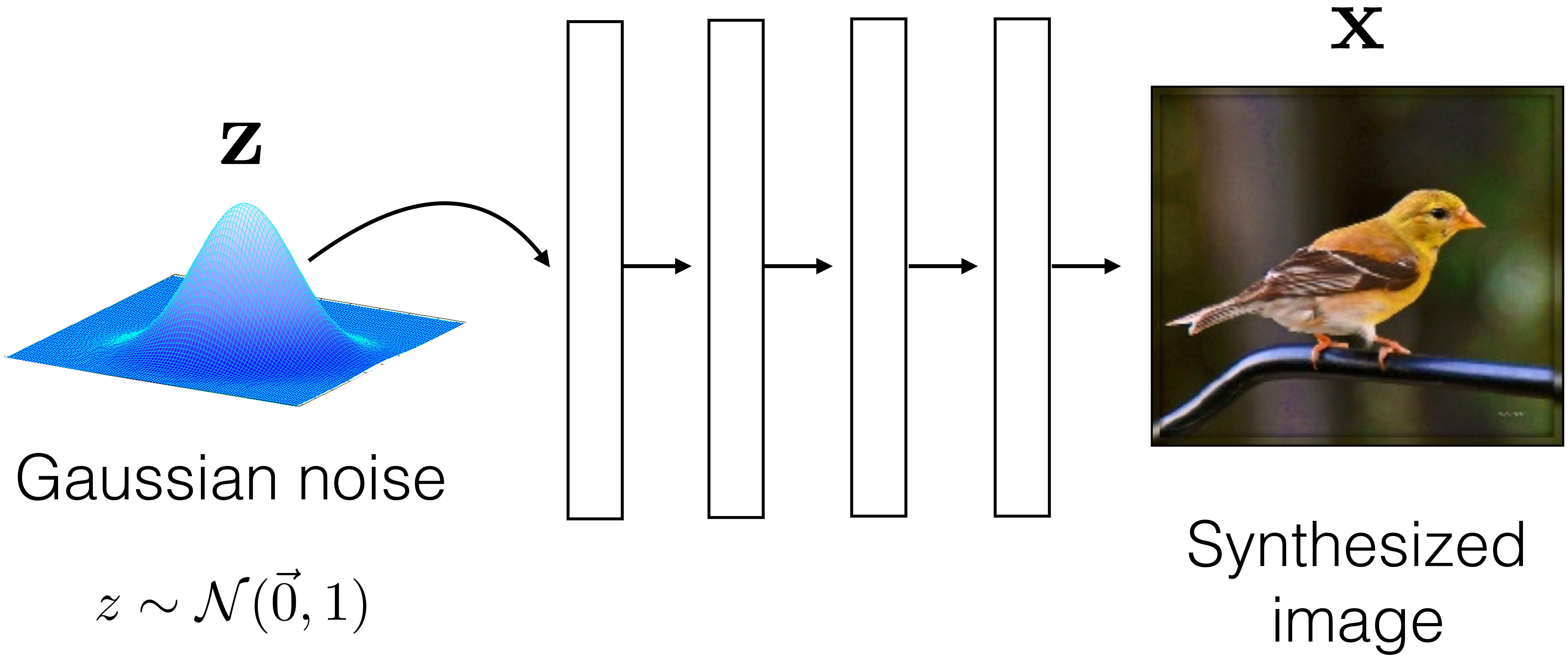
image X

What's a generative model?

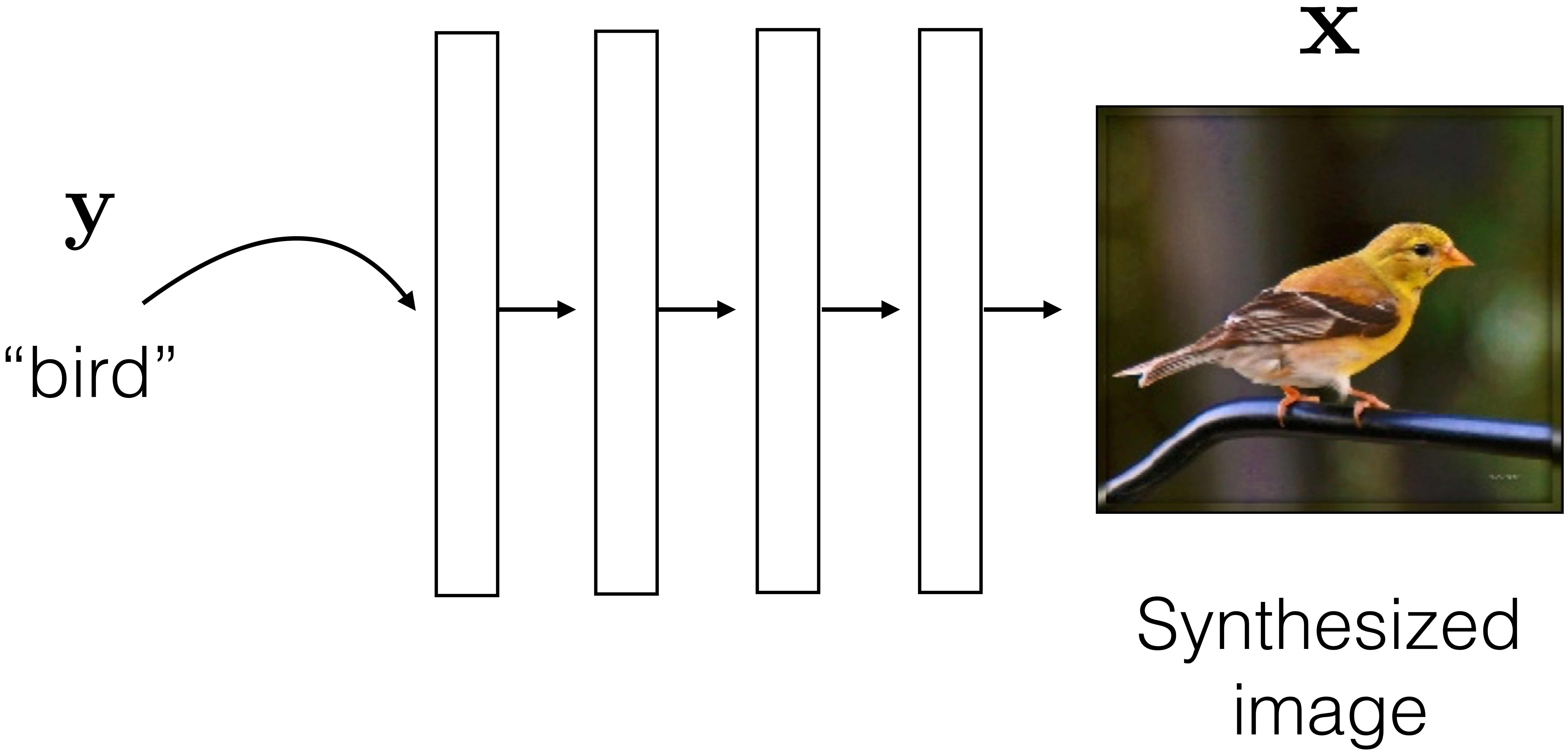
Model of high-dimensional unobserved variables $P(\mathbf{X}|\mathbf{Y}=\mathbf{y})$

Useful for lots of problems beyond sampling random images!

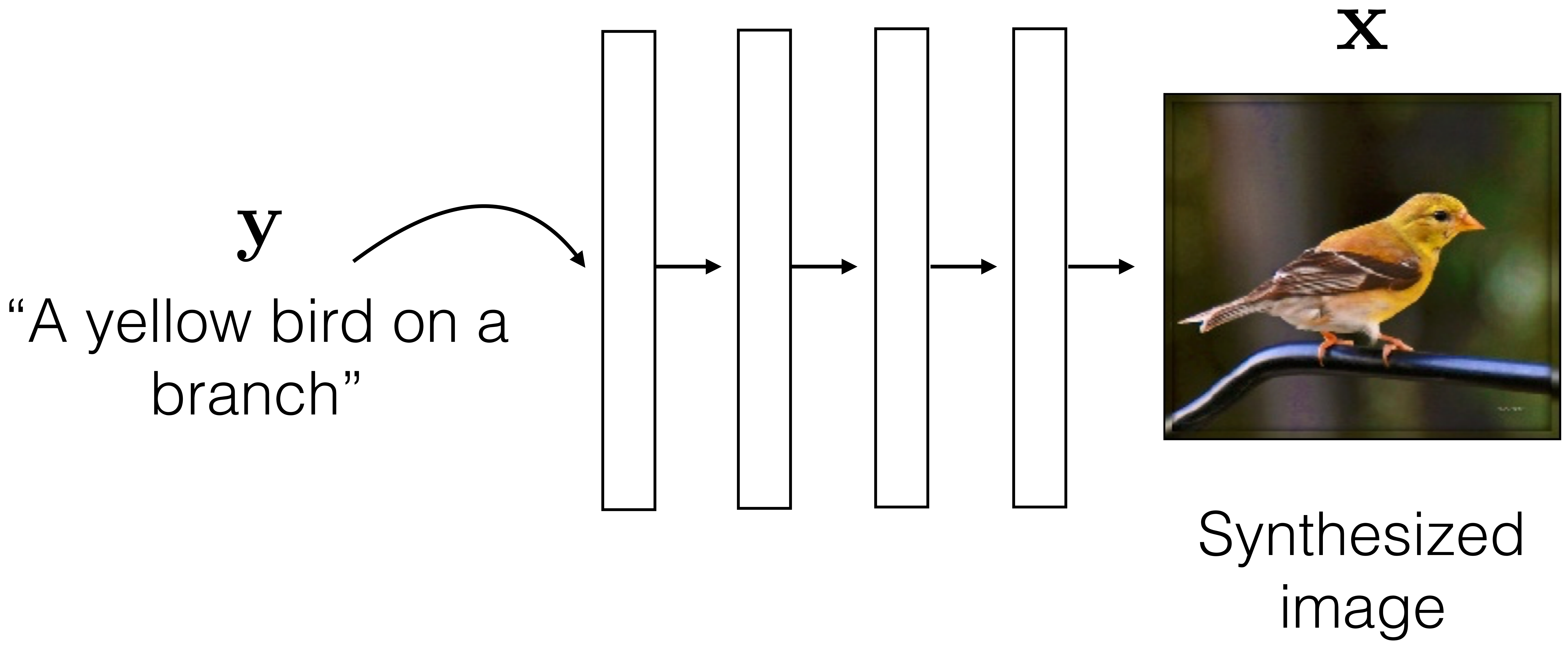
Generative Model



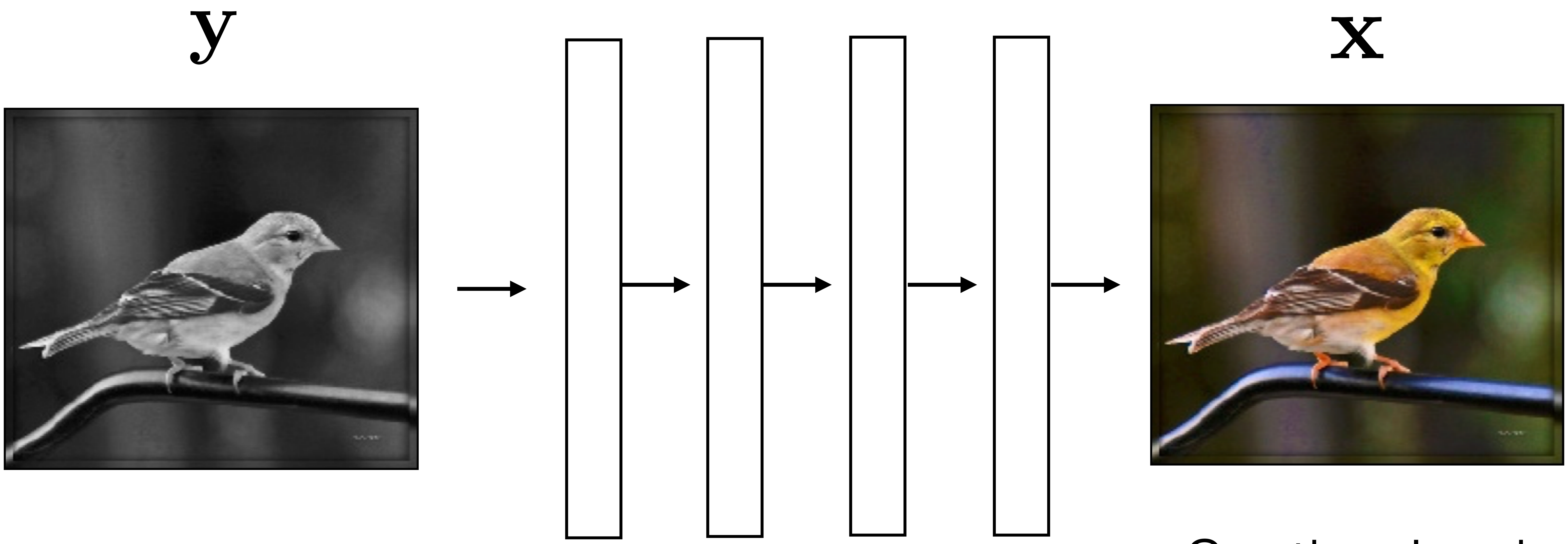
Conditional Generative Model



Conditional Generative Model



Conditional Generative Model



Synthesized image

Three perspectives on GANs

1. Structured loss

2. Generative model

3. Domain-level supervision / mapping

Three perspectives on GANs

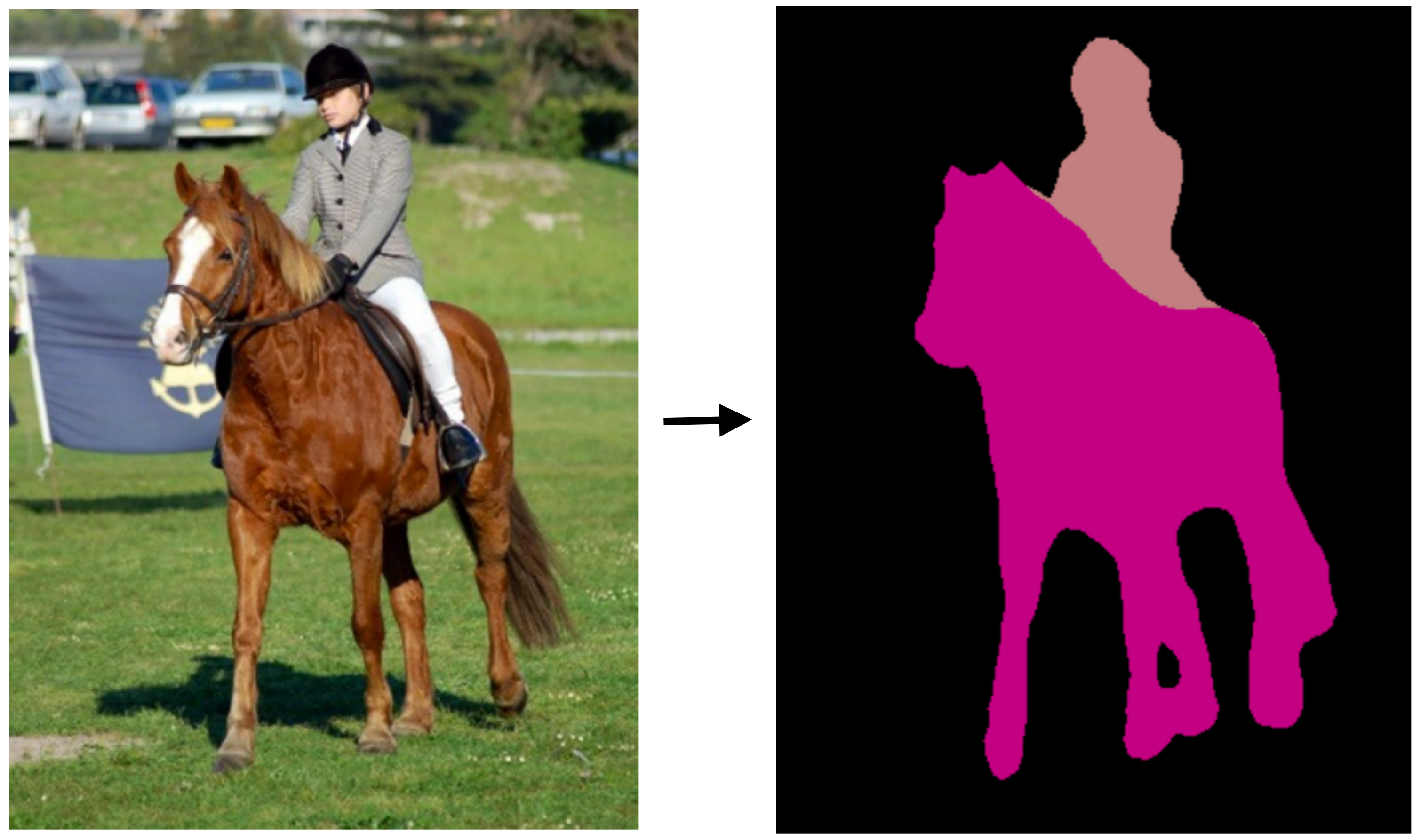
1. **Structured loss**

2. Generative model

3. Domain-level supervision / mapping

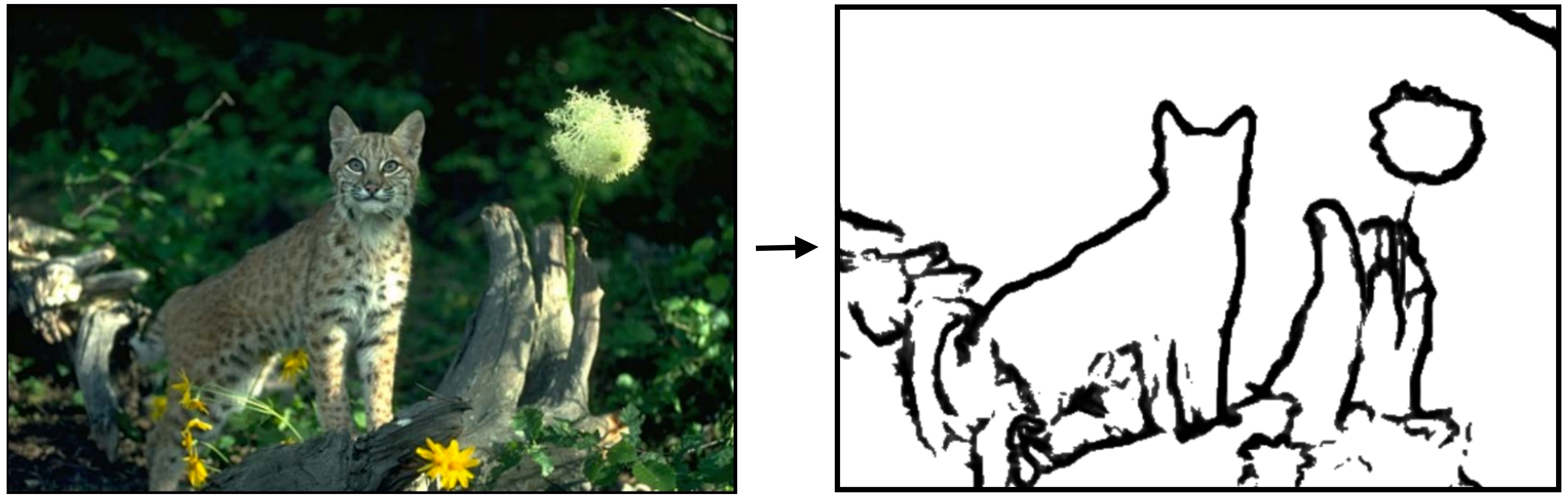
Data prediction problems ("structured prediction")

Object labeling



[Long et al. 2015, …]

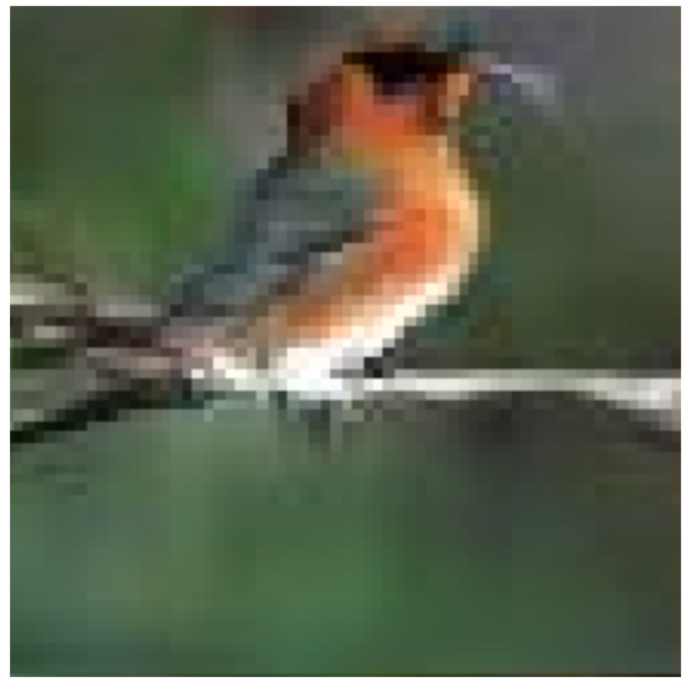
Edge Detection



[Xie et al. 2015, …]

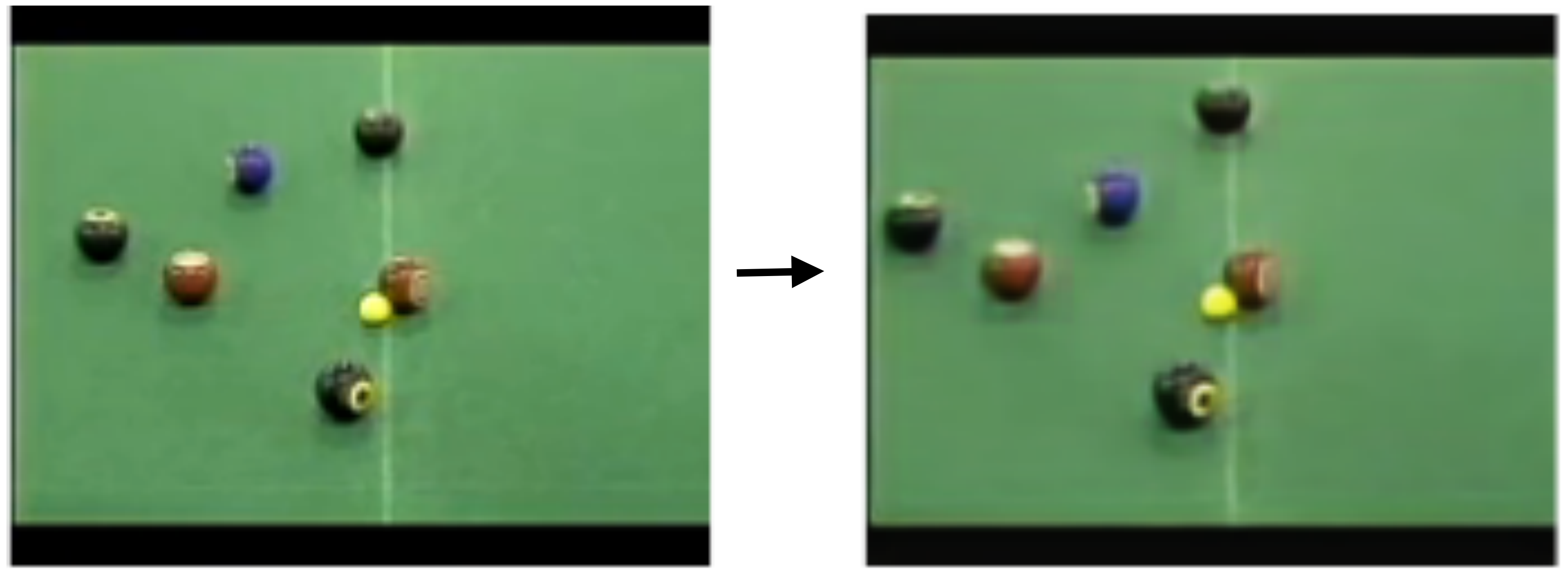
Text-to-photo

"this small bird has a pink breast and crown…" →



[Reed et al. 2014, …]

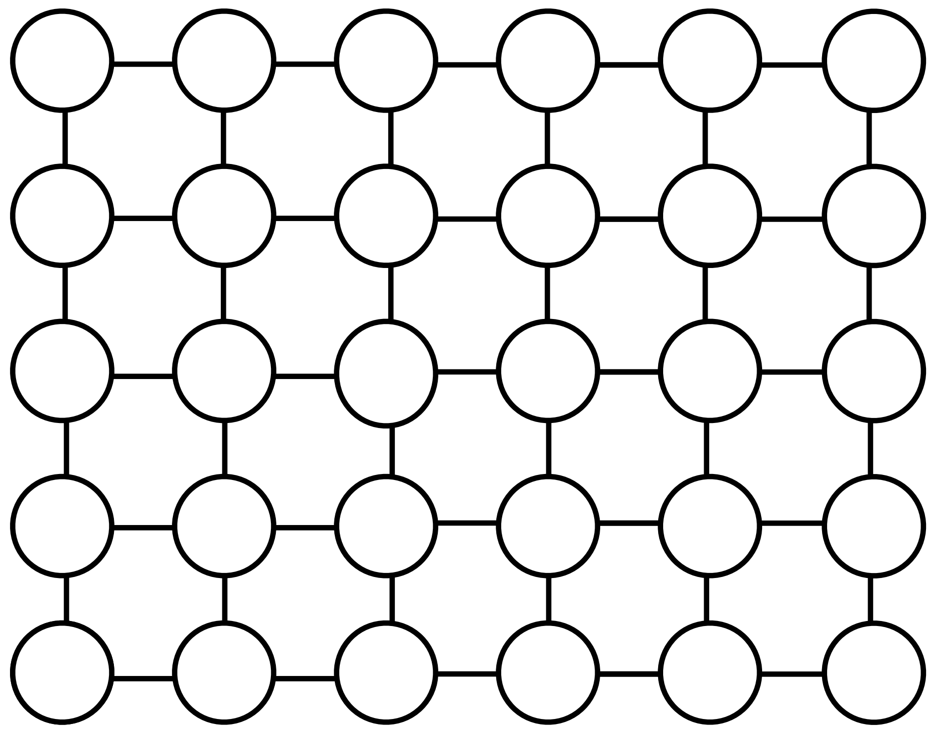
Future frame prediction



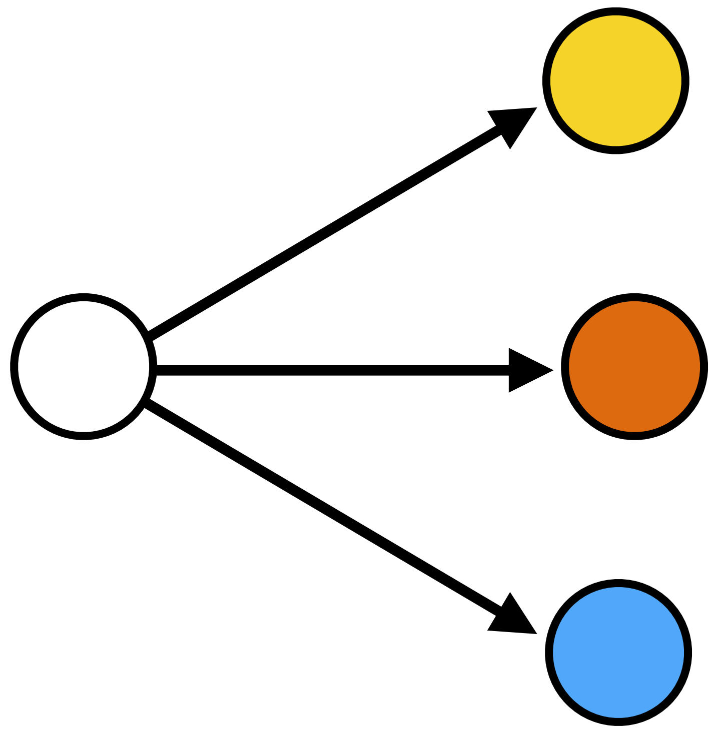
[Mathieu et al. 2016, …]

Challenges in data prediction

1. Output is a high-dimensional, structured object

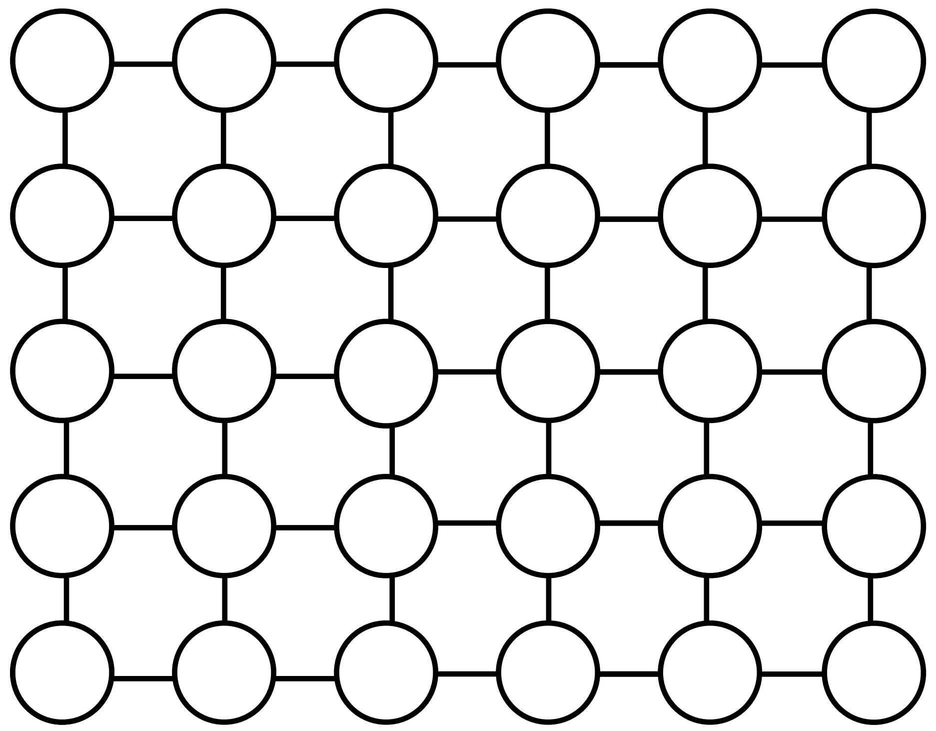


2. Uncertainty in the mapping, many plausible outputs



Properties of generative models

1. Model high-dimensional, structured output



2. Model uncertainty; a whole distribution of possible outputs

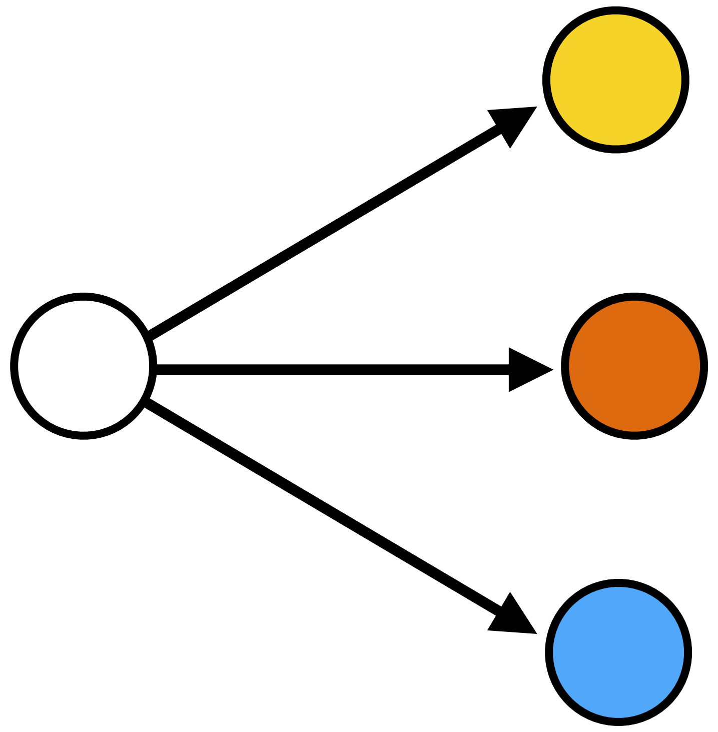
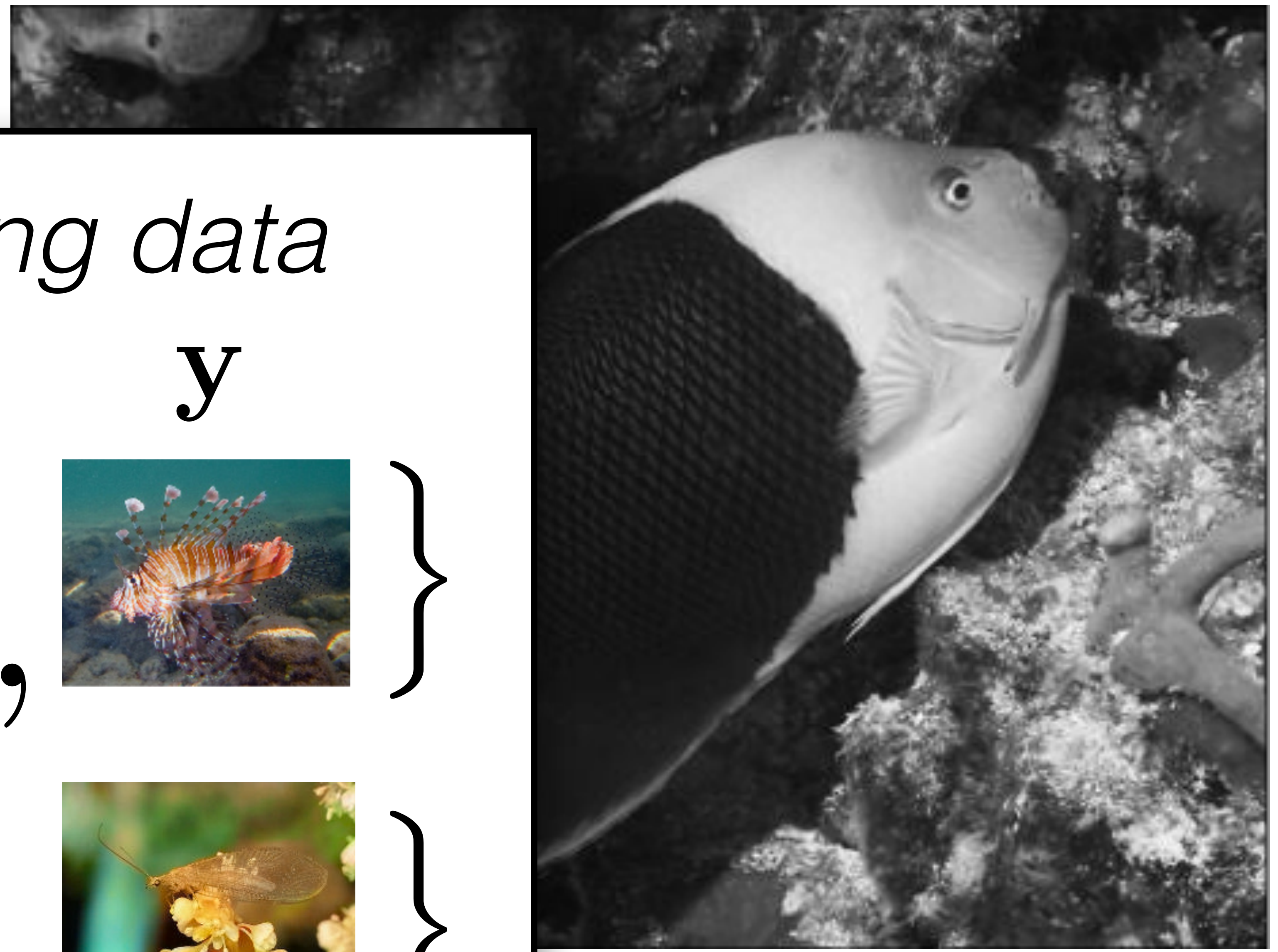


Image-to-Image Translation

Input $\mathbf{x}$



$\mathcal{F}$

Output $\mathbf{y}$

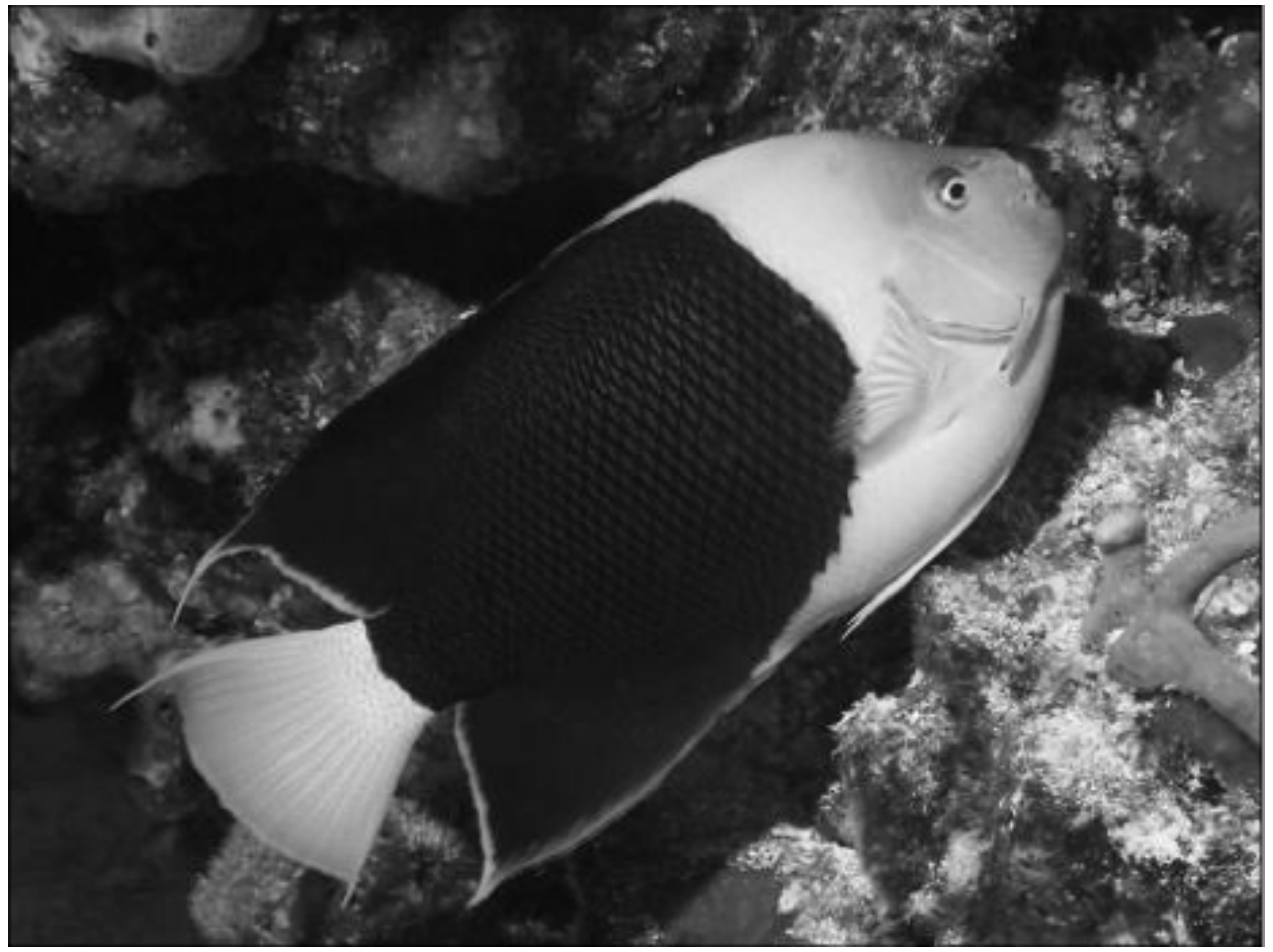
$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[L(\mathcal{F}(\mathbf{x}), \mathbf{y})]$$

Objective function (loss)

Neural Network

Image-to-Image Translation

Input $\mathbf{x}$



$\mathcal{F}$

Output $\mathbf{y}$

$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[L(\mathcal{F}(\mathbf{x}), \mathbf{y})]$$

"**What** should I do" "**How** should I do it?"

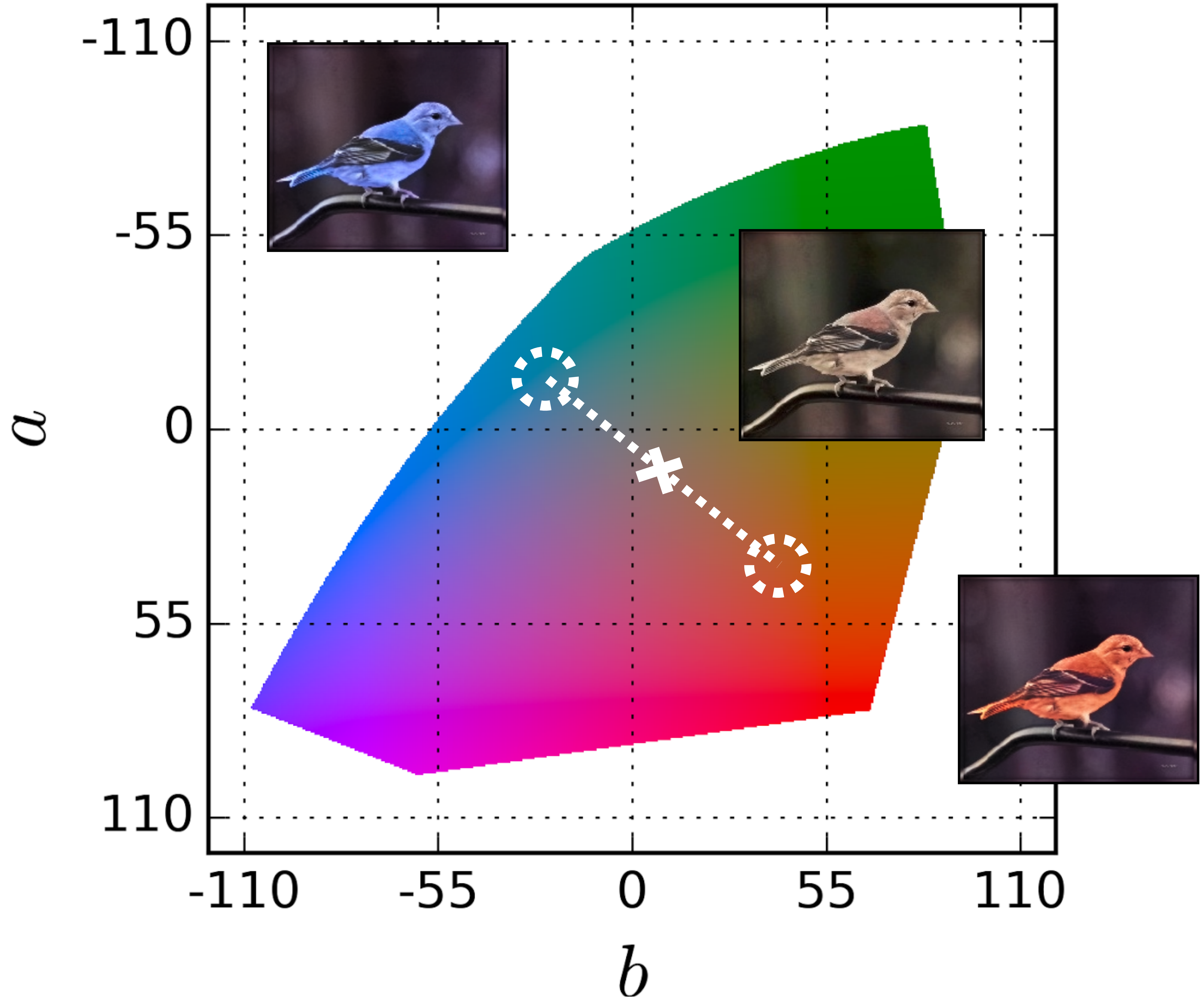
Designing loss functions

Input

Output

Ground truth

$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$



$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

Designing loss functions

Input

Zhang et al. 2016

Ground truth

Color distribution cross-entropy loss with colorfulness enhancing term.

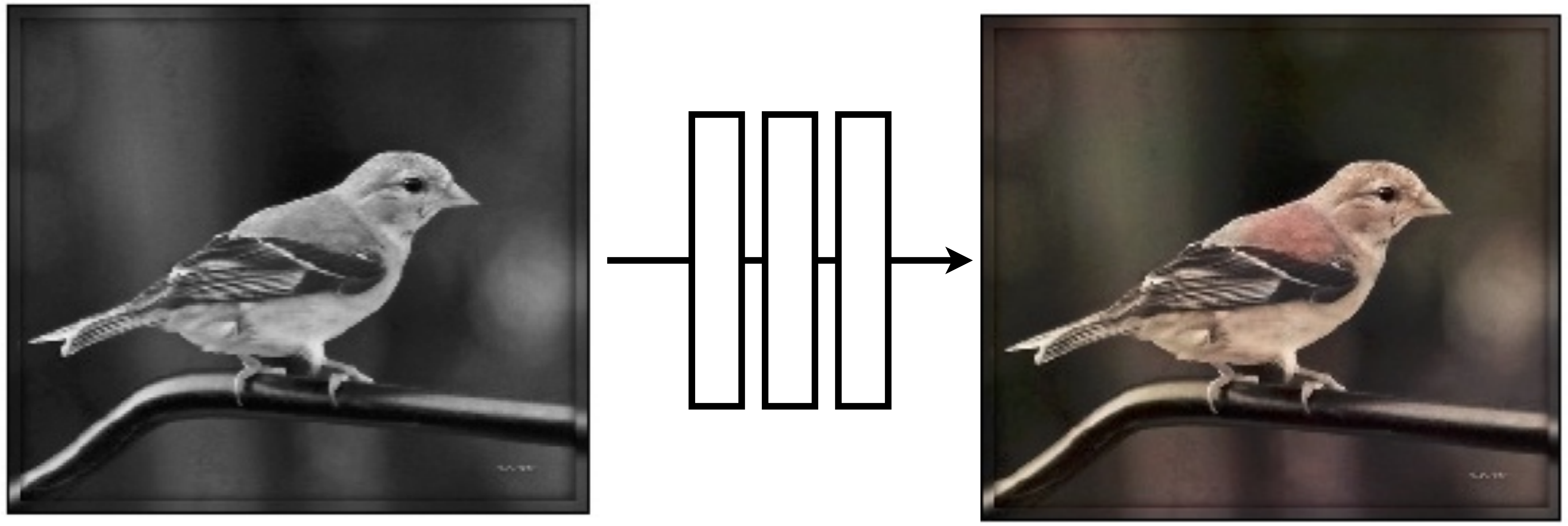

DADS
DADS

Designing loss functions

Be careful what you wish for!

Designing loss functions

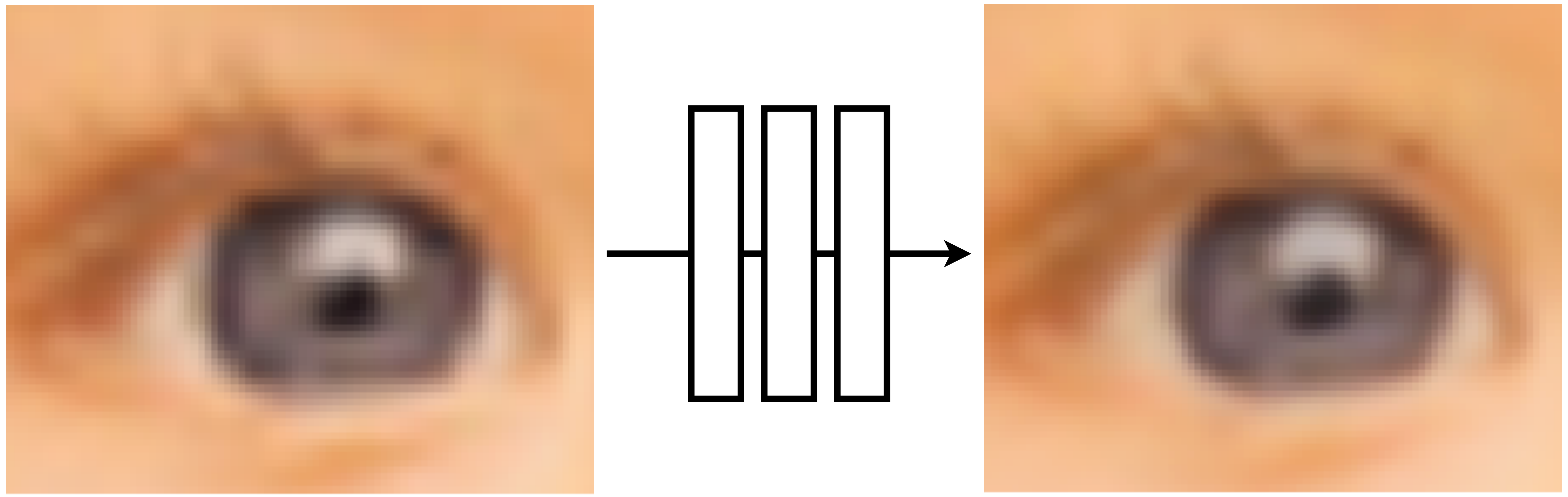
Image colorization



[Zhang, Isola, Efros, ECCV 2016]

L2 regression

Super-resolution

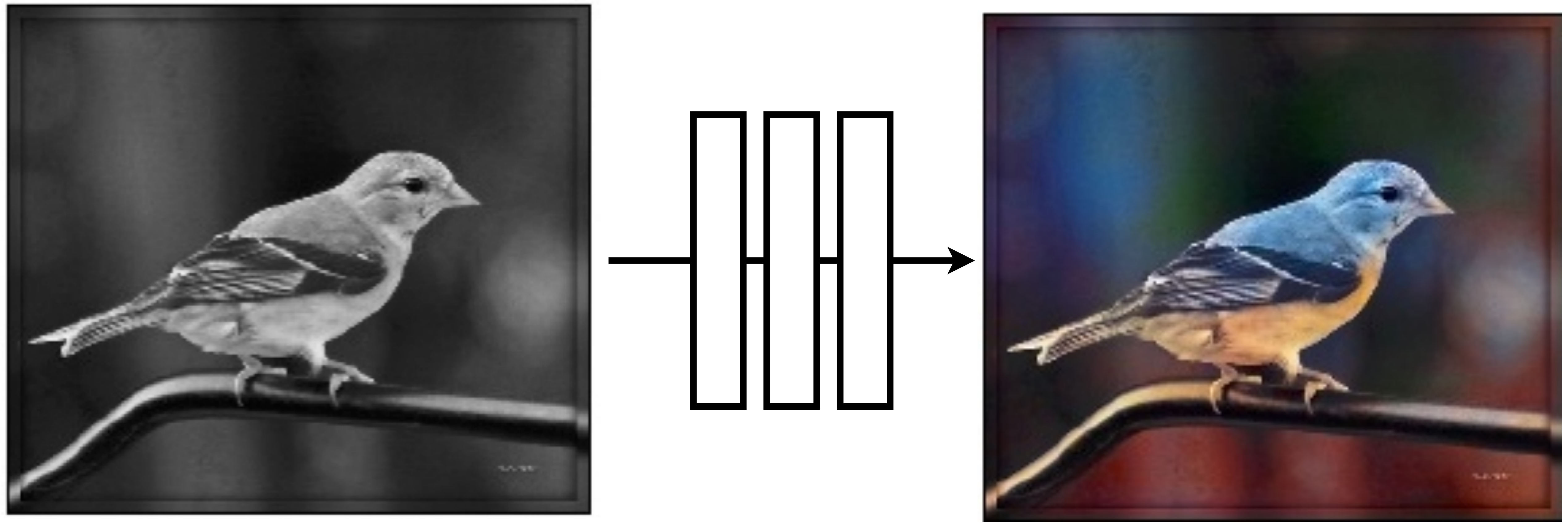


[Johnson, Alahi, Li, ECCV 2016]

L2 regression

Designing loss functions

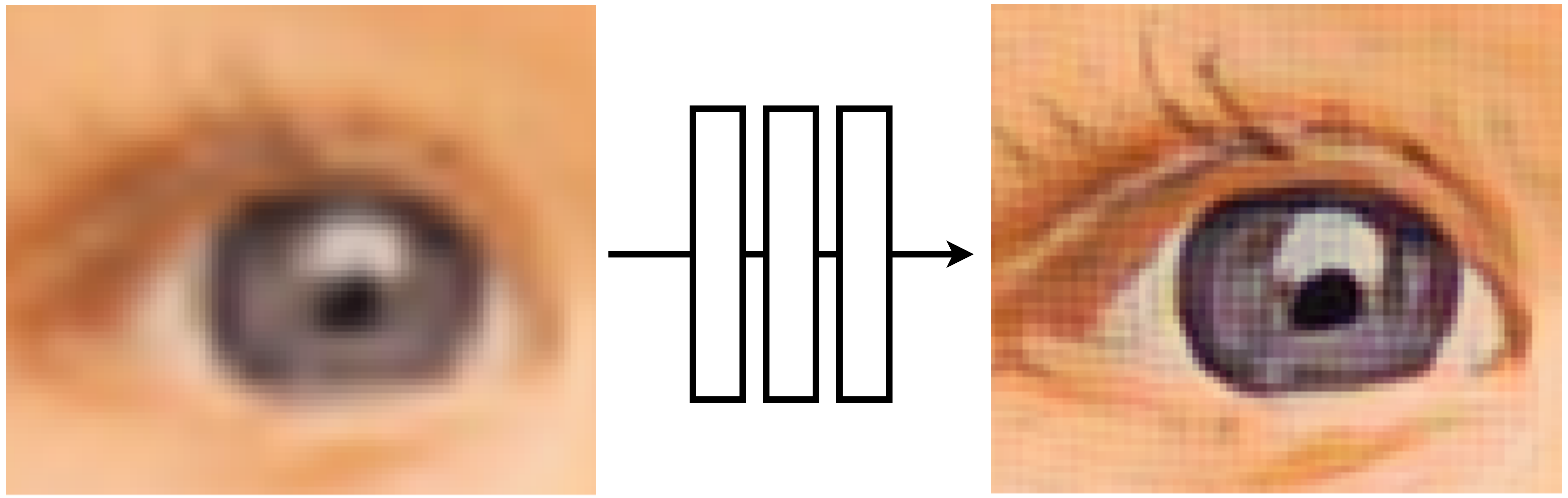
Image colorization



[Zhang, Isola, Efros, ECCV 2016]

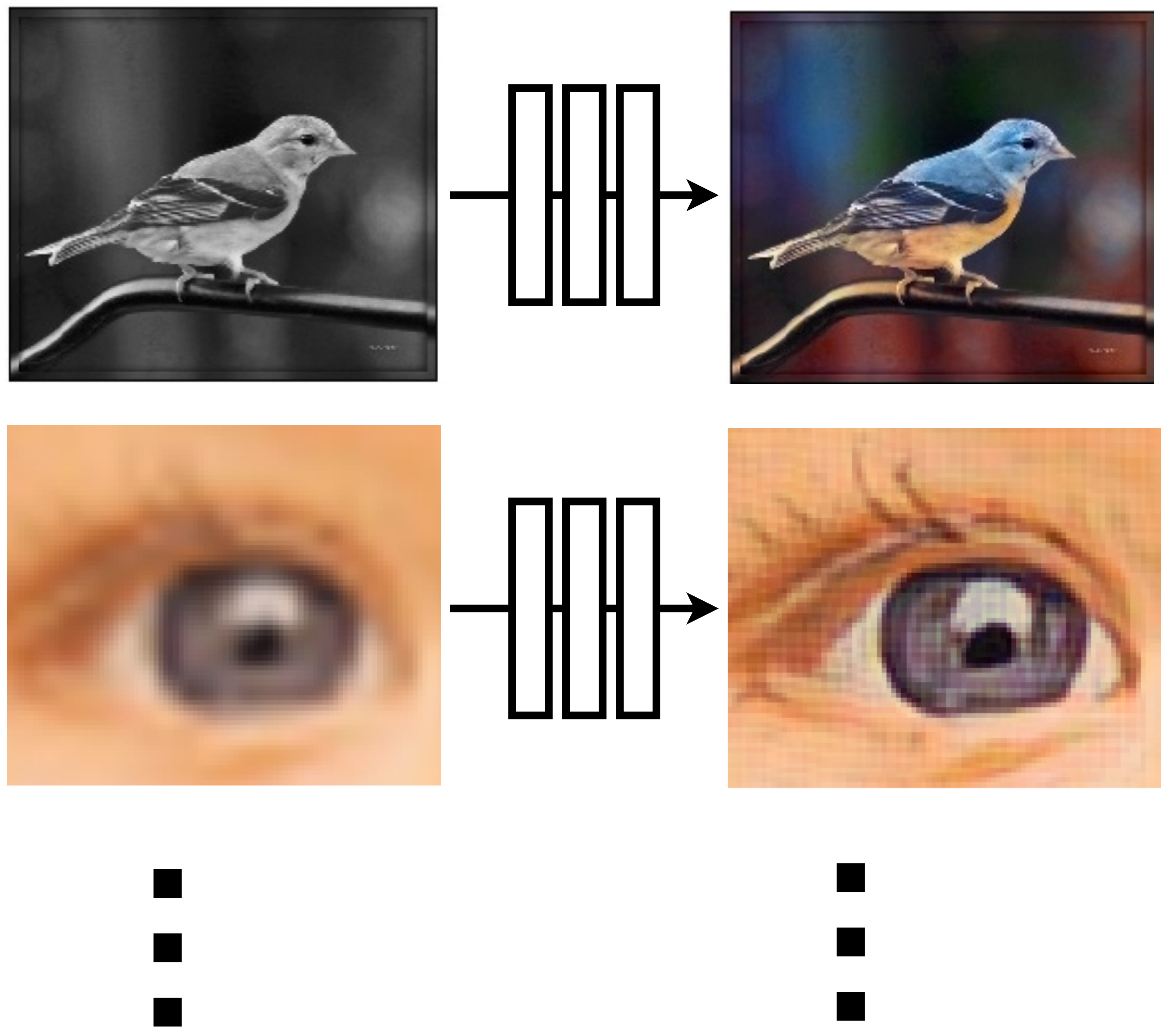
Cross entropy objective, with colorfulness term

Super-resolution



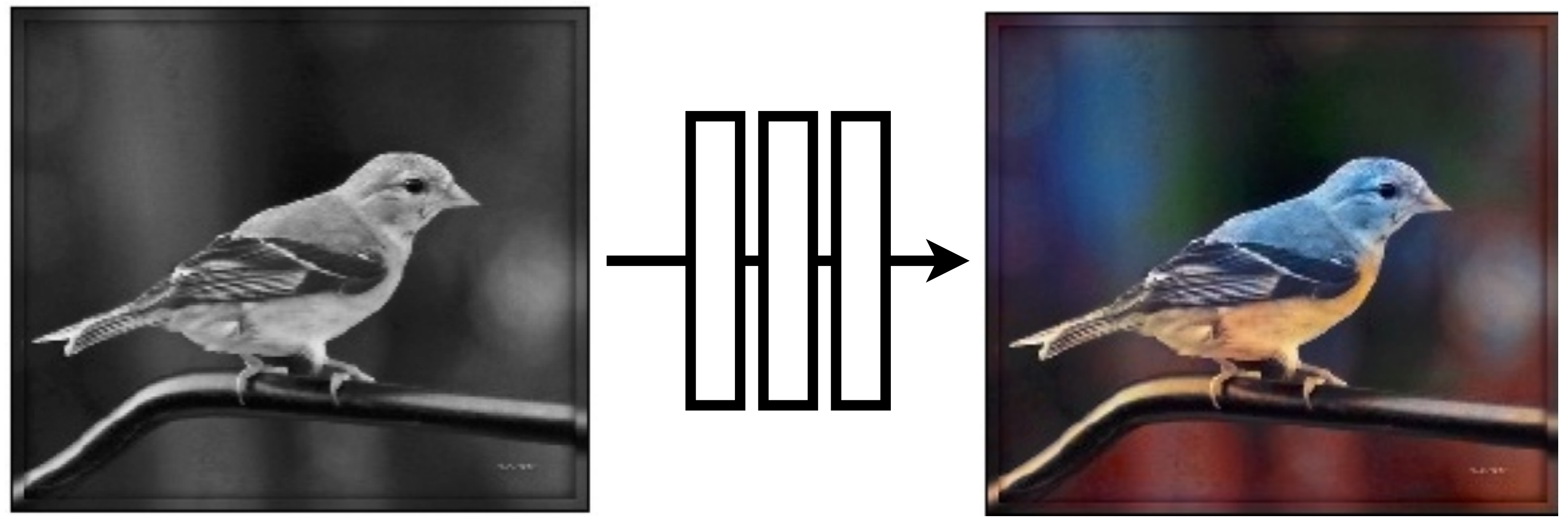
[Johnson, Alahi, Li, ECCV 2016]

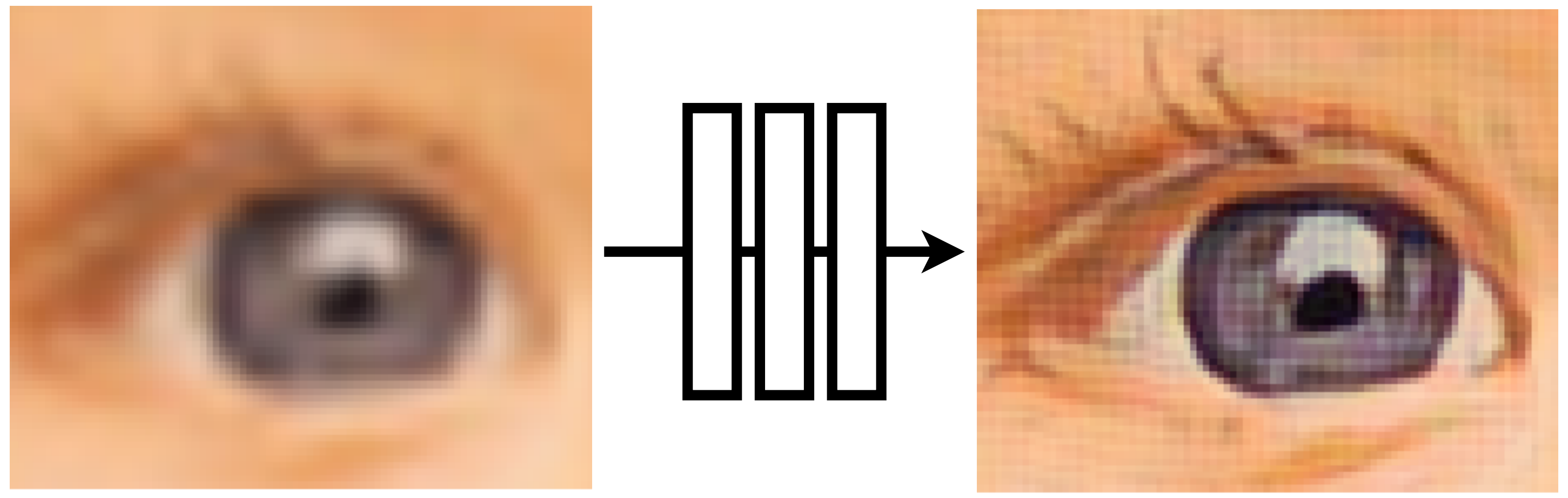
Deep feature covariance matching objective



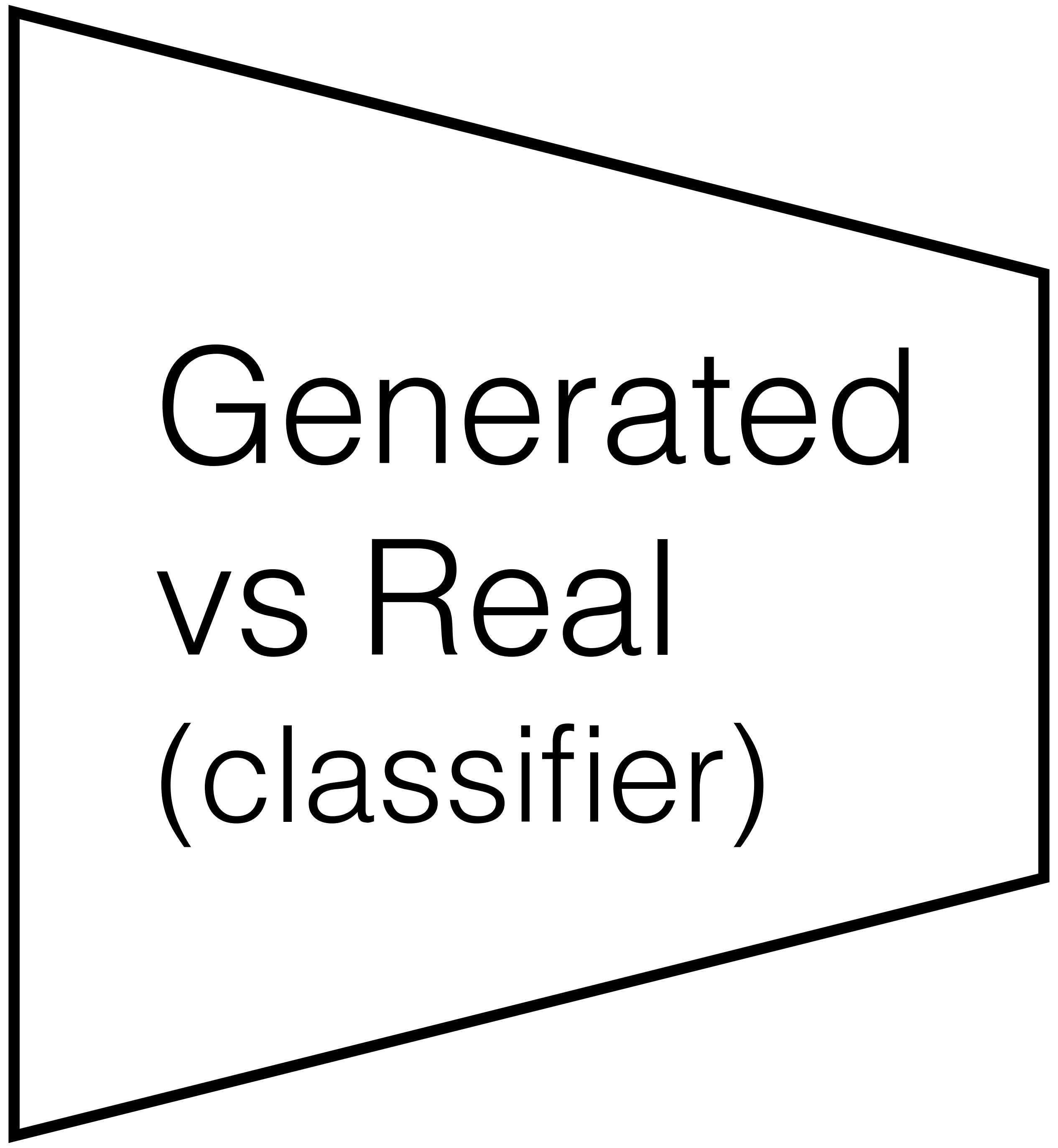
Universal loss?

Generated images





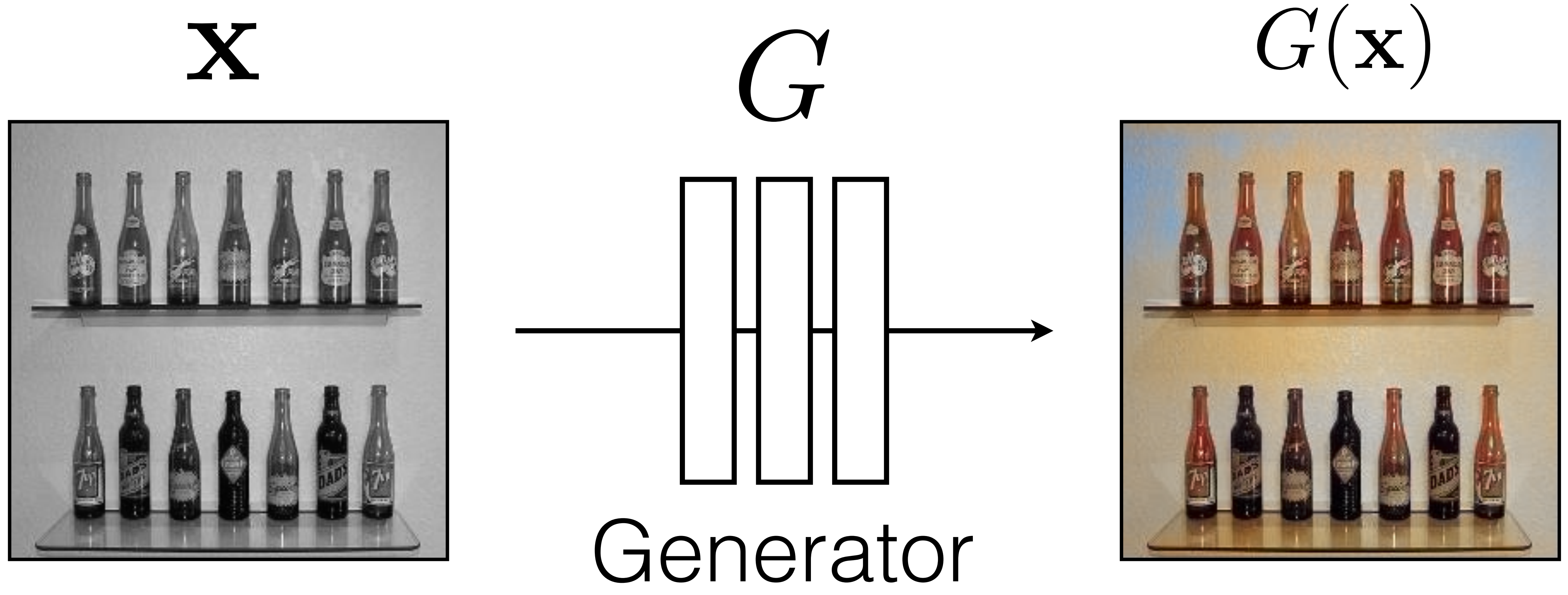
“Generative Adversarial Network” (GANs)



Real photos

...

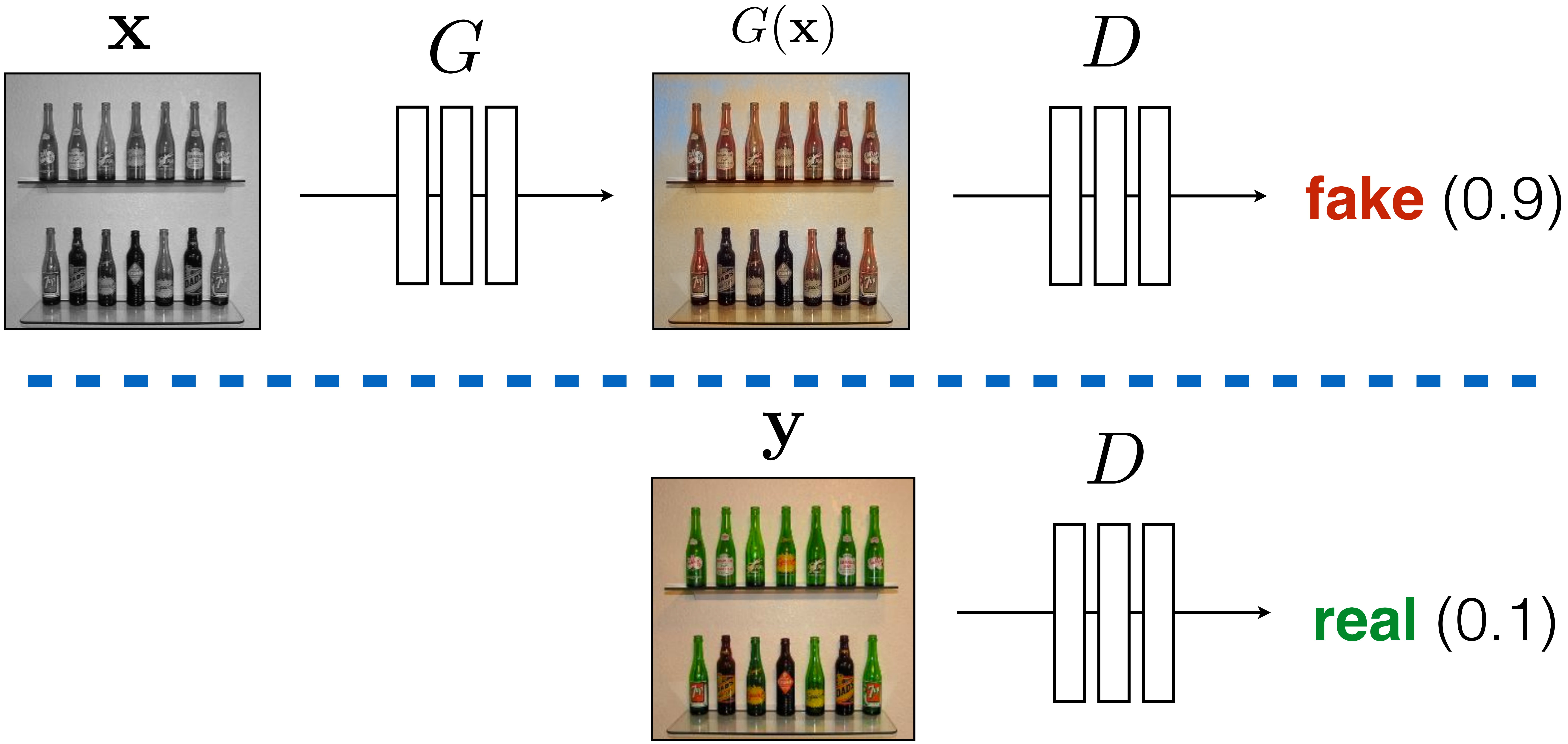
[Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, Bengio 2014]



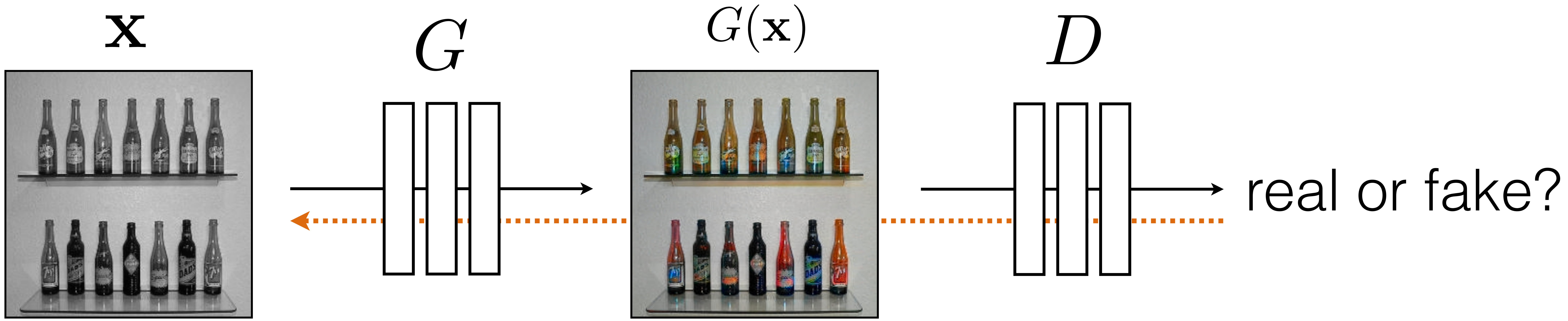
$\mathbf{x}$
G
Generator
$G(\mathbf{x})$

G tries to synthesize fake images that fool **D**

D tries to identify the fakes

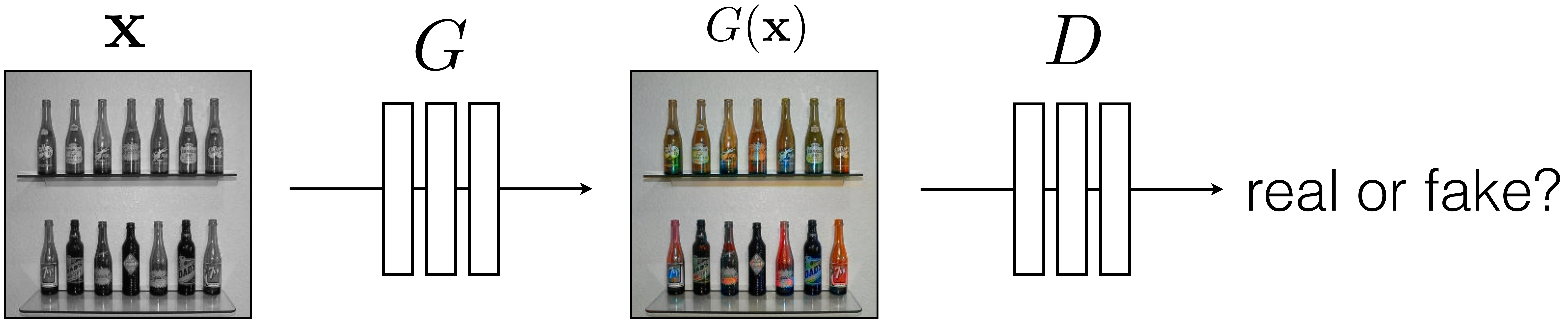


$$\arg\max_{D} \ \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\ \boxed{\log D(G(\mathbf{x}))} \ + \ \boxed{\log(1 - D(\mathbf{y}))}\ \right]$$



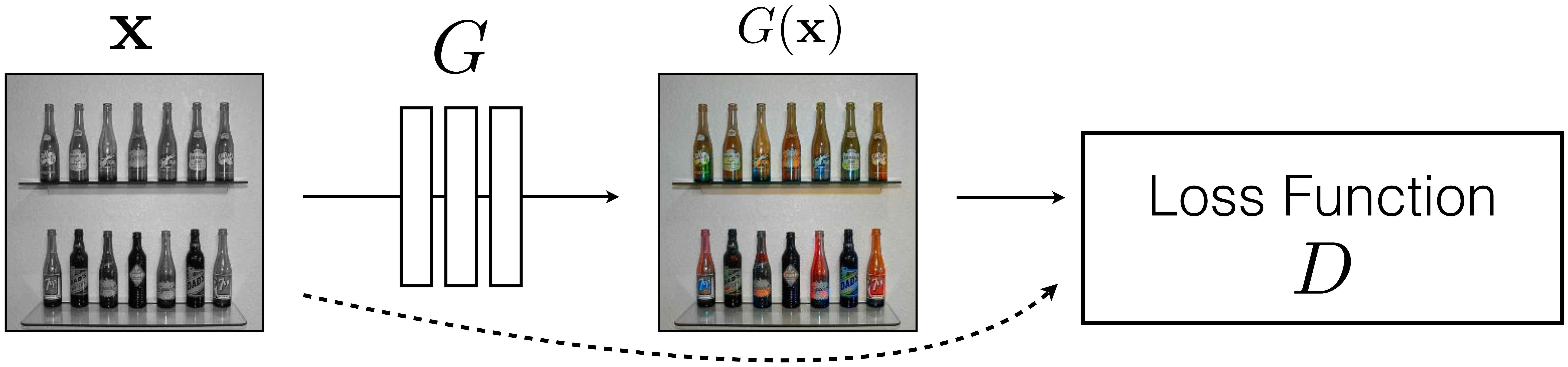
G tries to synthesize fake images that ***fool*** **D**:

$$\arg \boxed{\min_{G}} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y})) \quad]$$



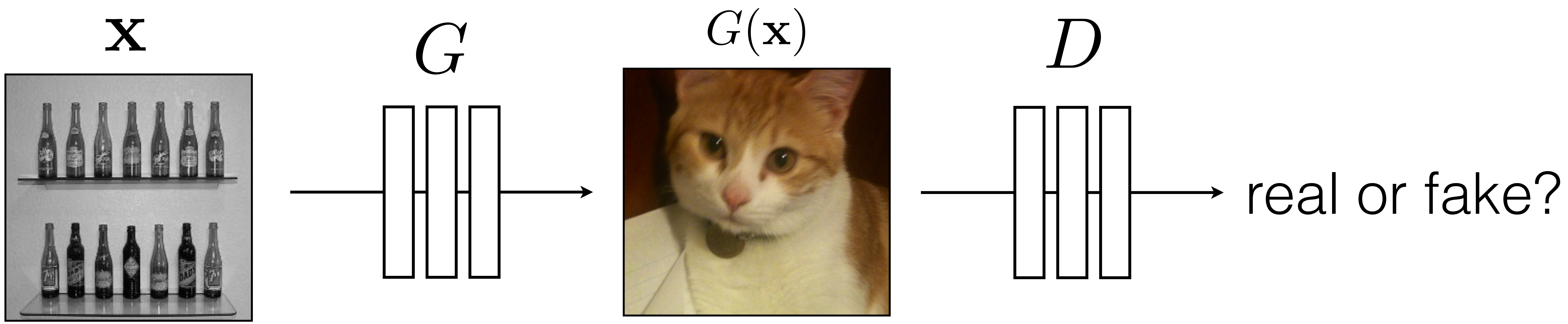
G tries to synthesize fake images that ***fool*** the ***best*** **D**:

$$\arg \boxed{\min_{G}} \boxed{\max_{D}} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y})) \;\;]$$

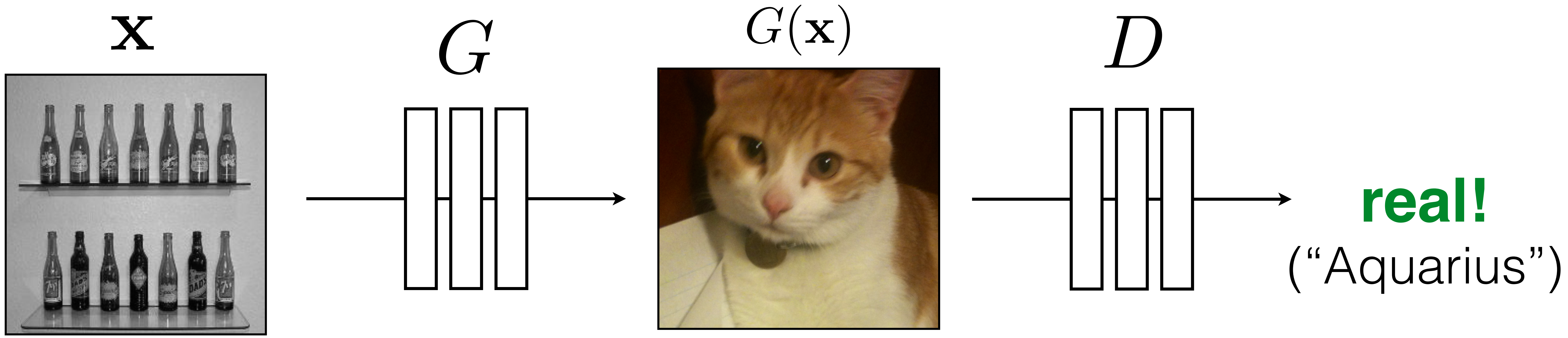


G’s perspective: **D** is a loss function.

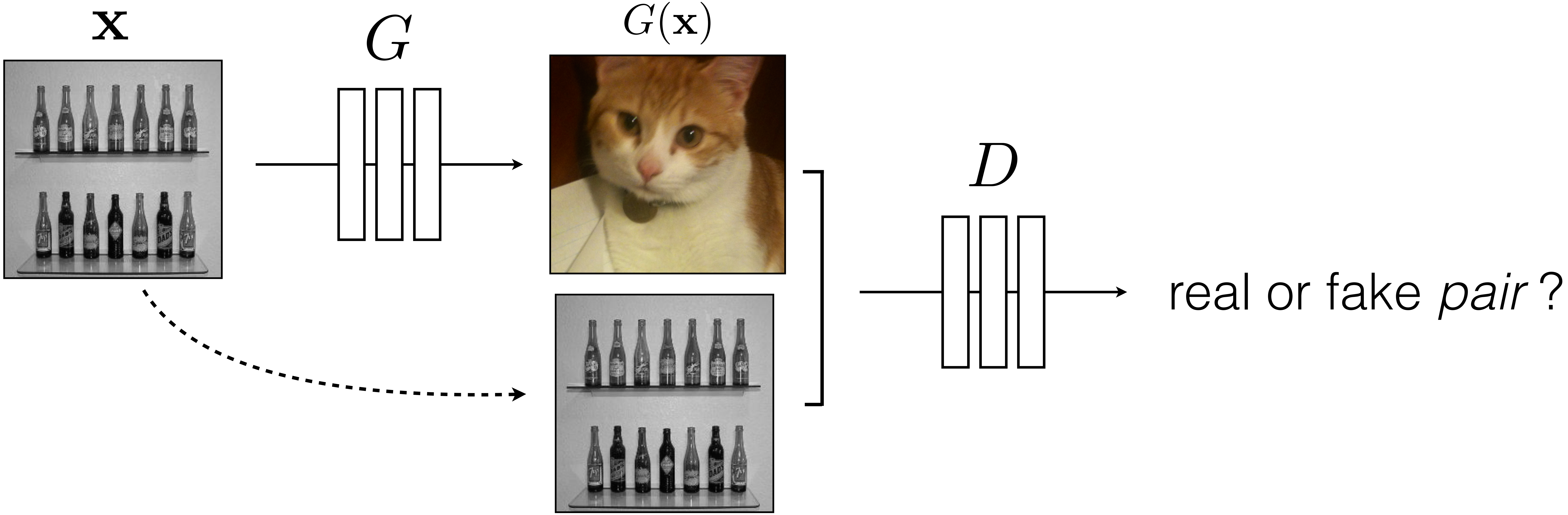
Rather than being hand-designed, it is *learned*.



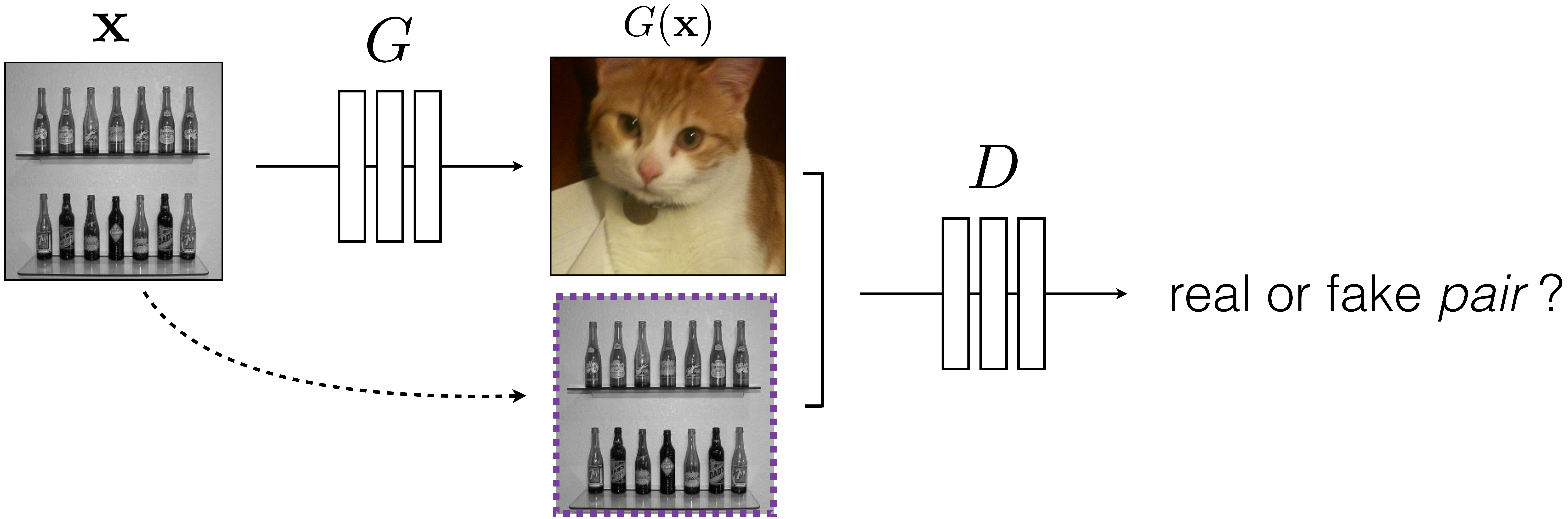
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y})) \;]$$



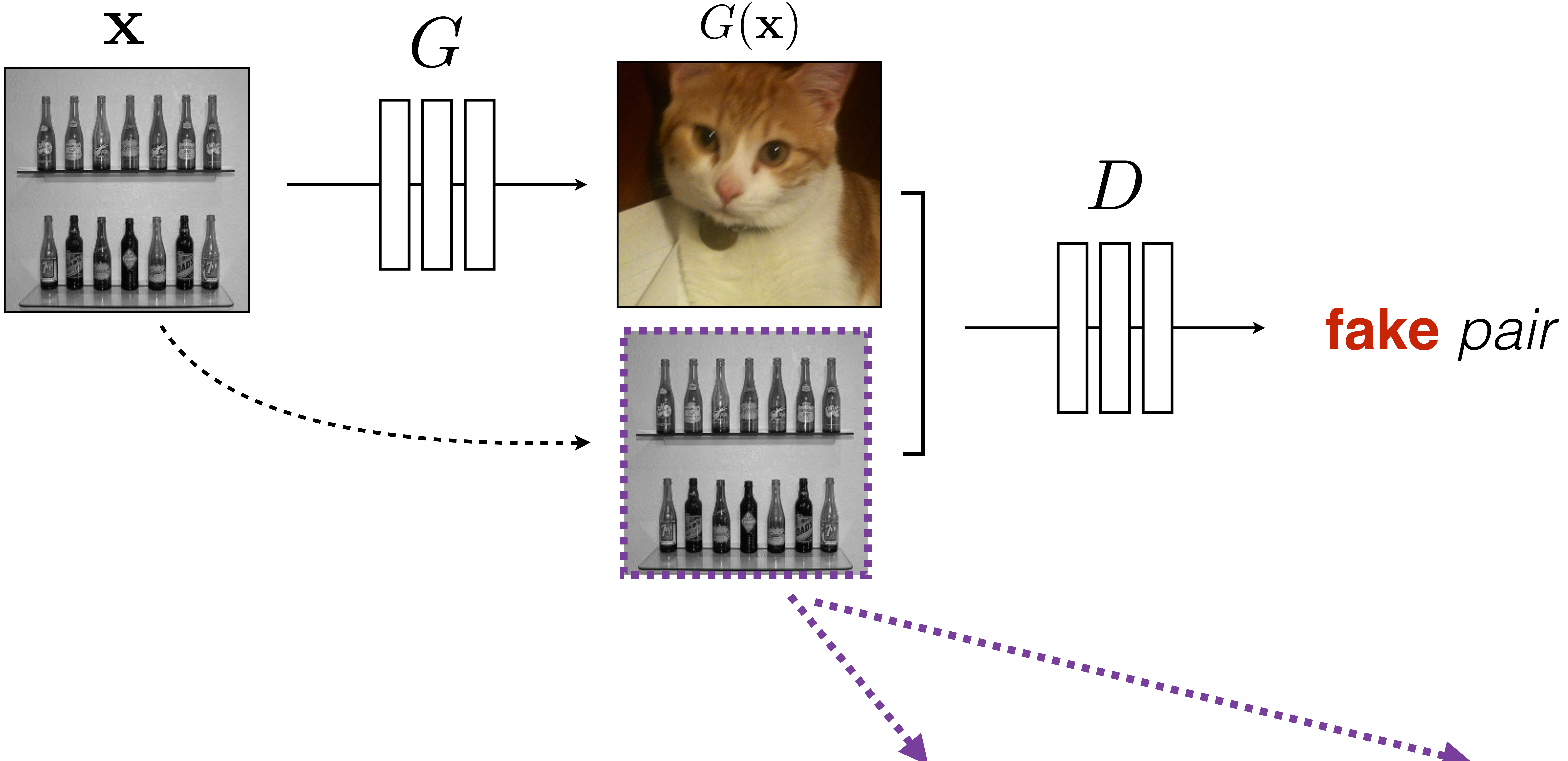
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ \ ]$$



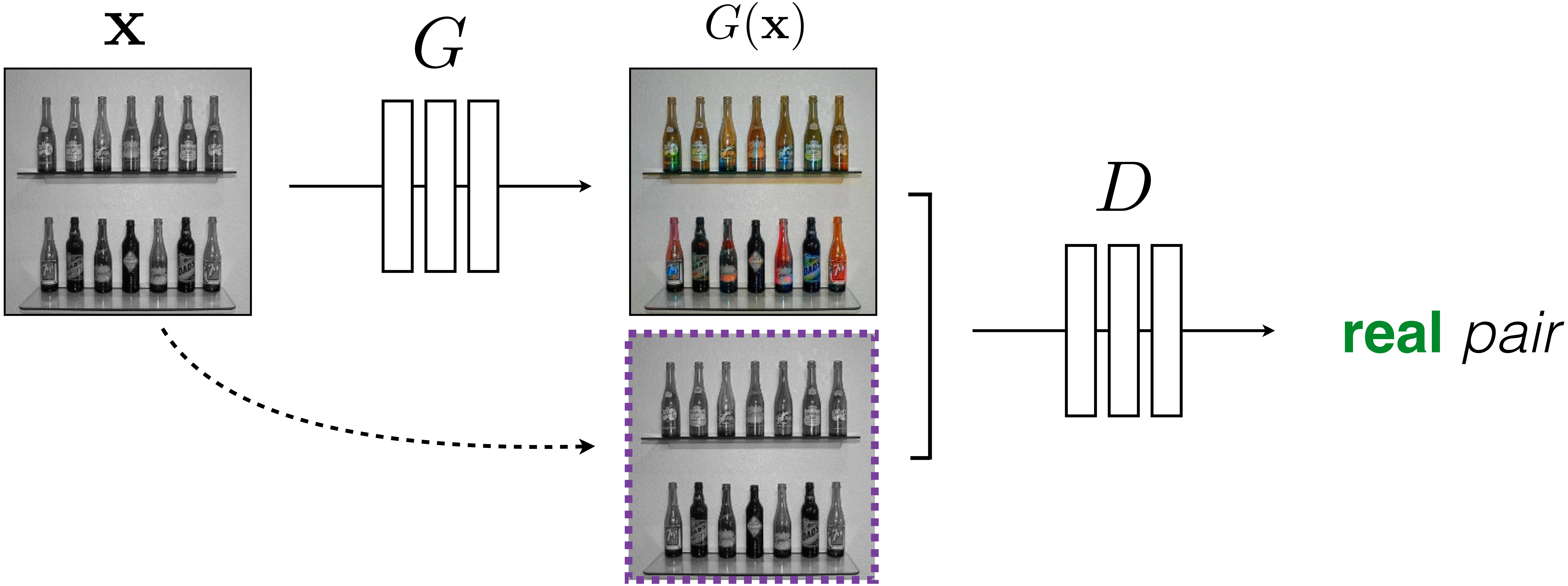
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1-D(\mathbf{y}))\ ]$$



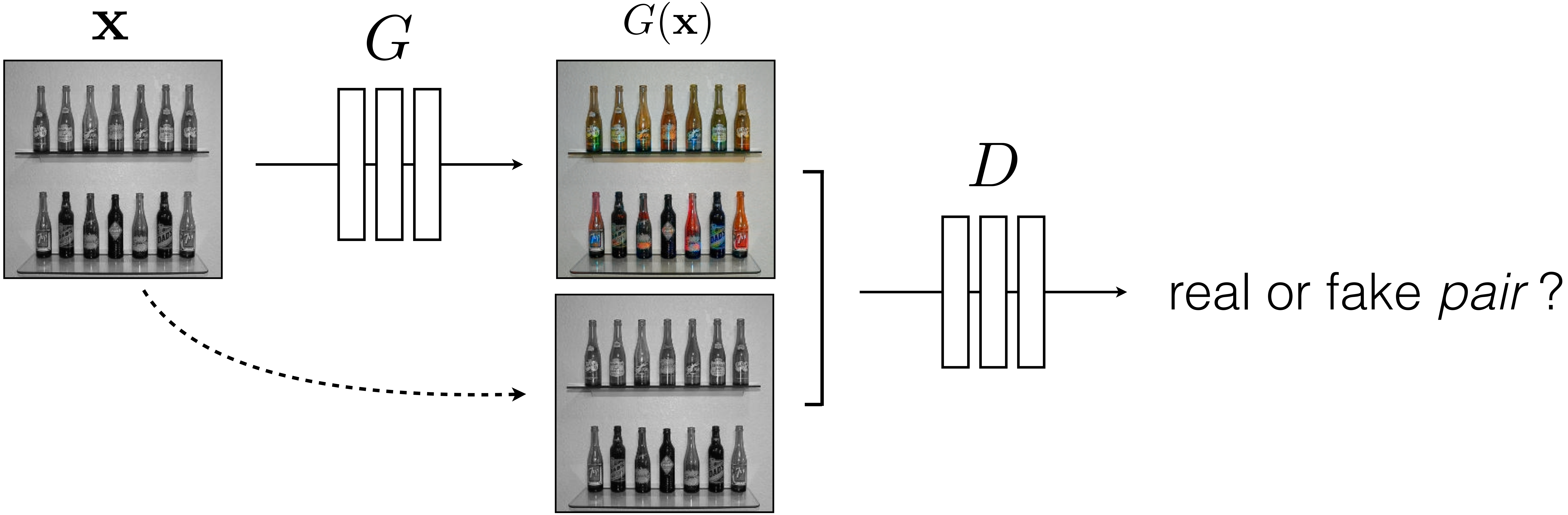
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ ]$$



$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ ]$$



$$\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ \right]$$



$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x},\mathbf{y}))\ ]$$

Training Details: Loss function

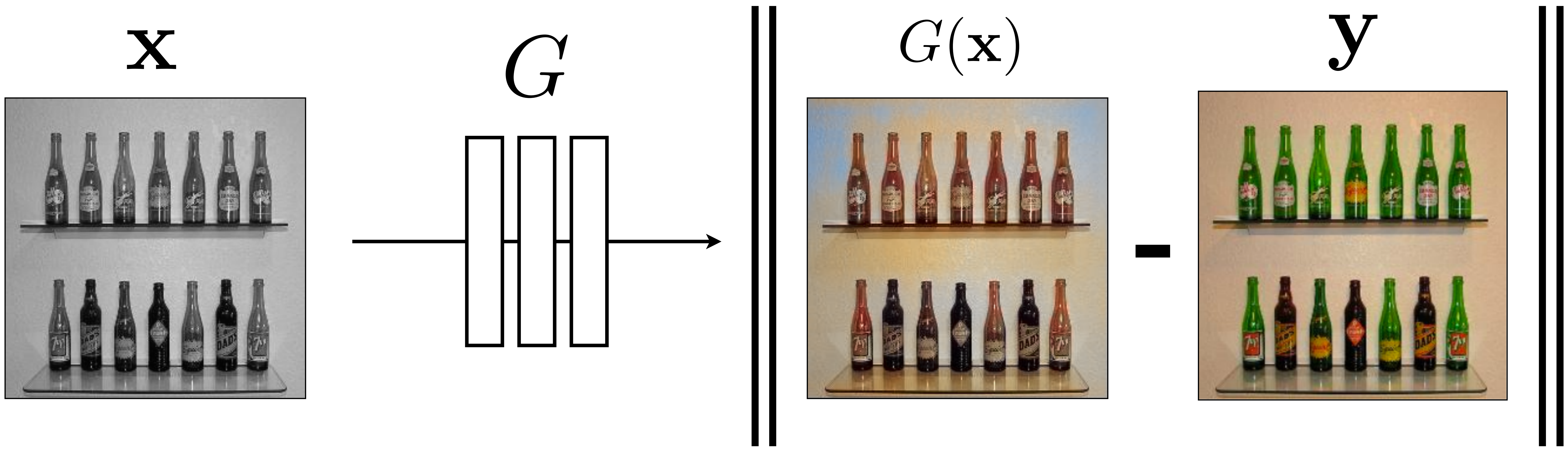
Conditional GAN

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G).$$

Training Details: Loss function

Conditional GAN

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$



Stable training + fast convergence

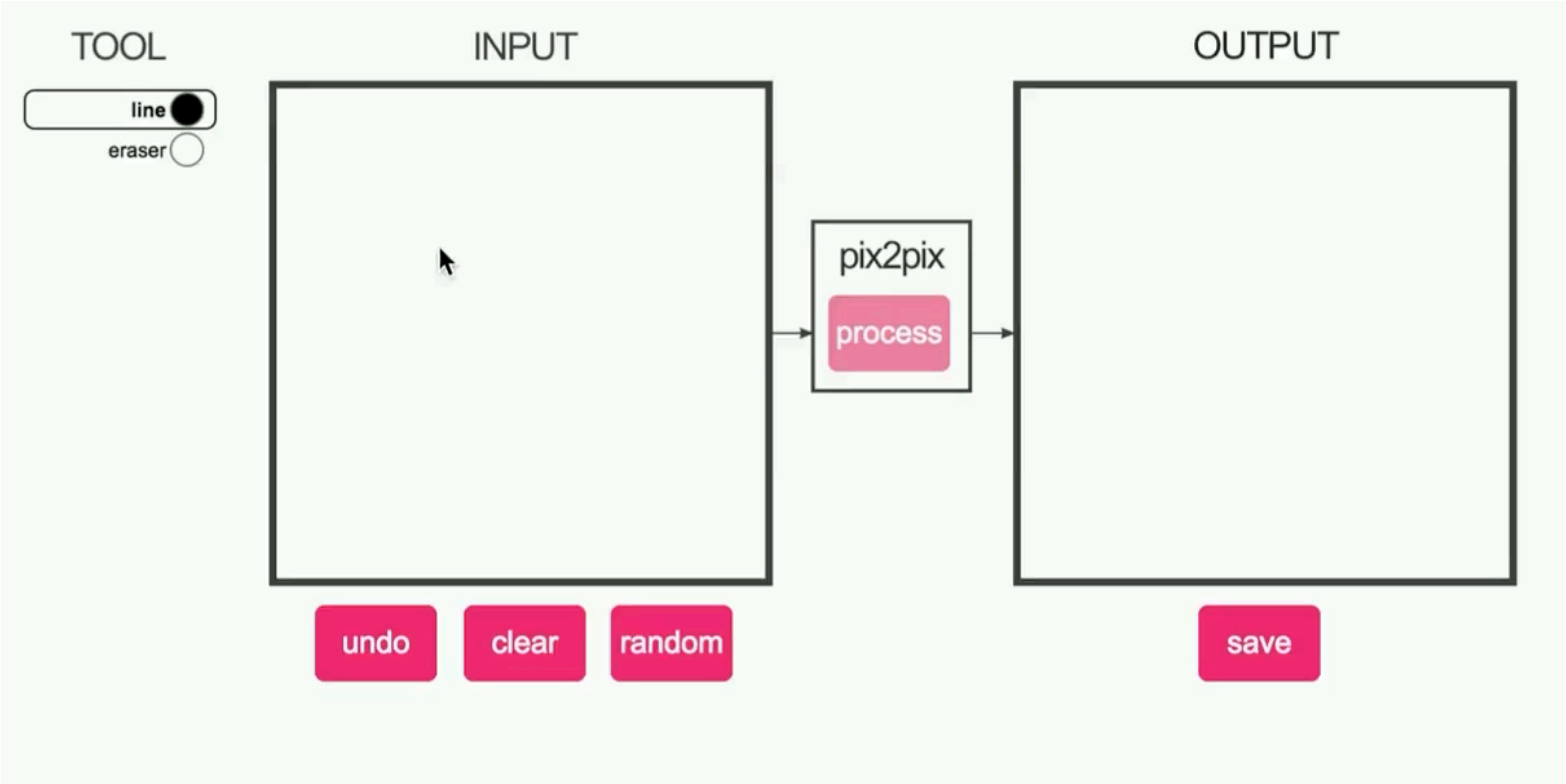
[c.f. Pathak et al. CVPR 2016]

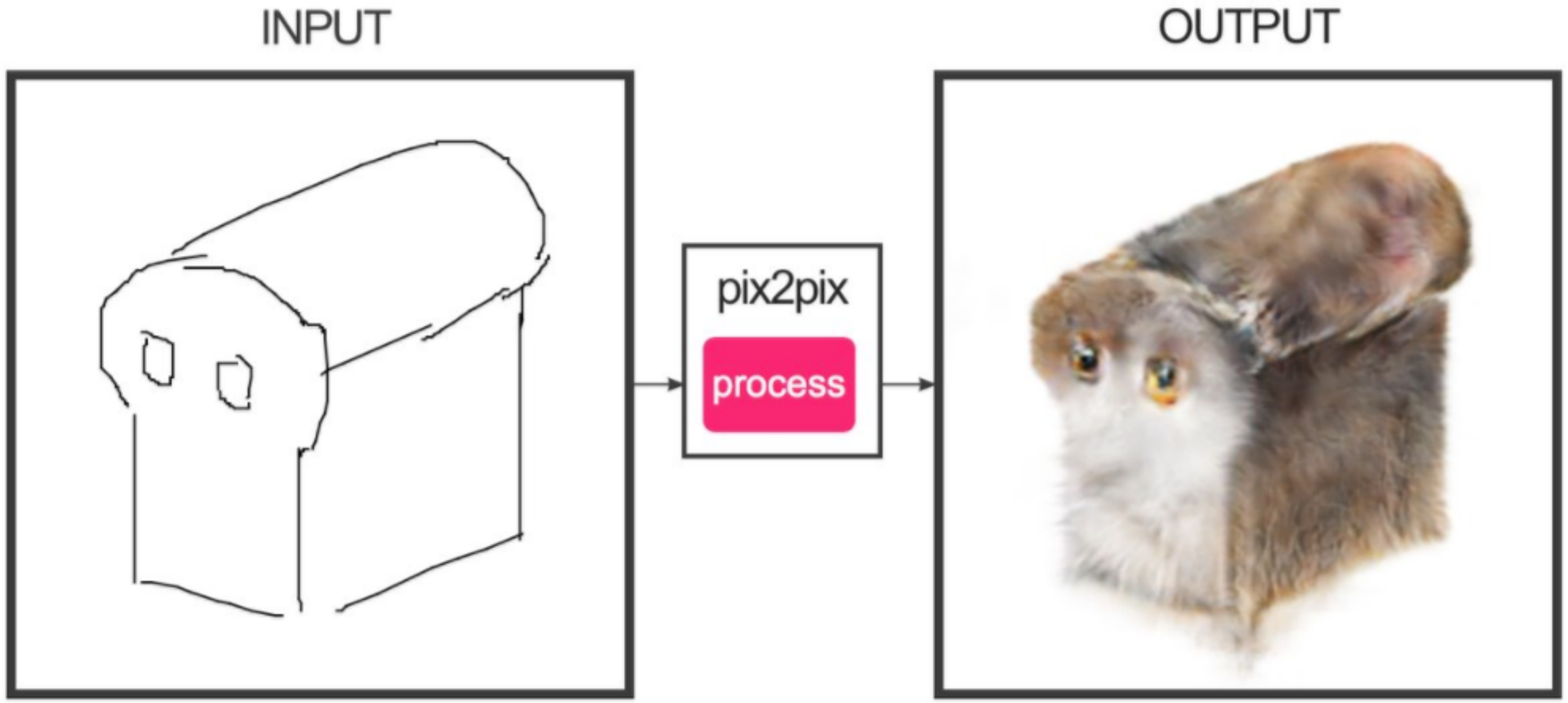
BW → Color

Input Output Input Output Input Output

Data from [Russakovsky et al. 2015]

#edges2cats [Chris Hesse]





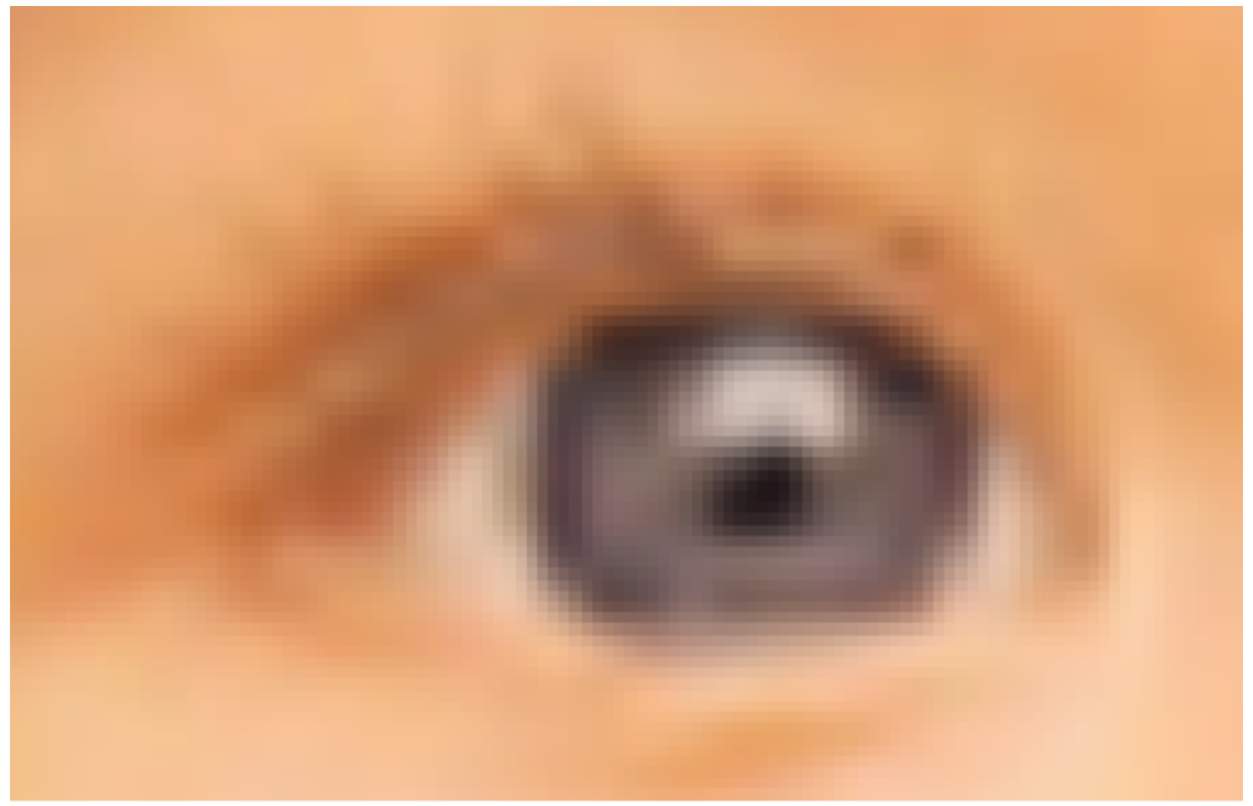
Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

Structured Prediction

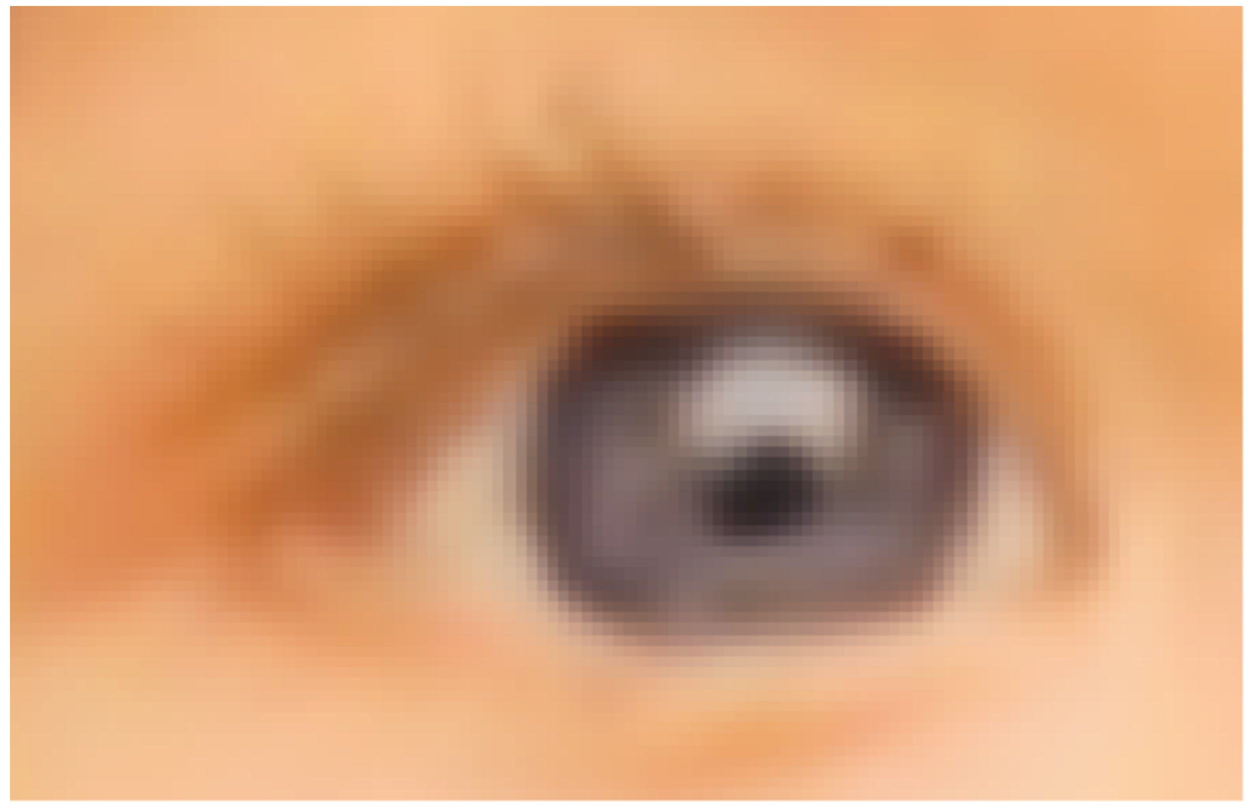
Input

$\mathbf{x}$



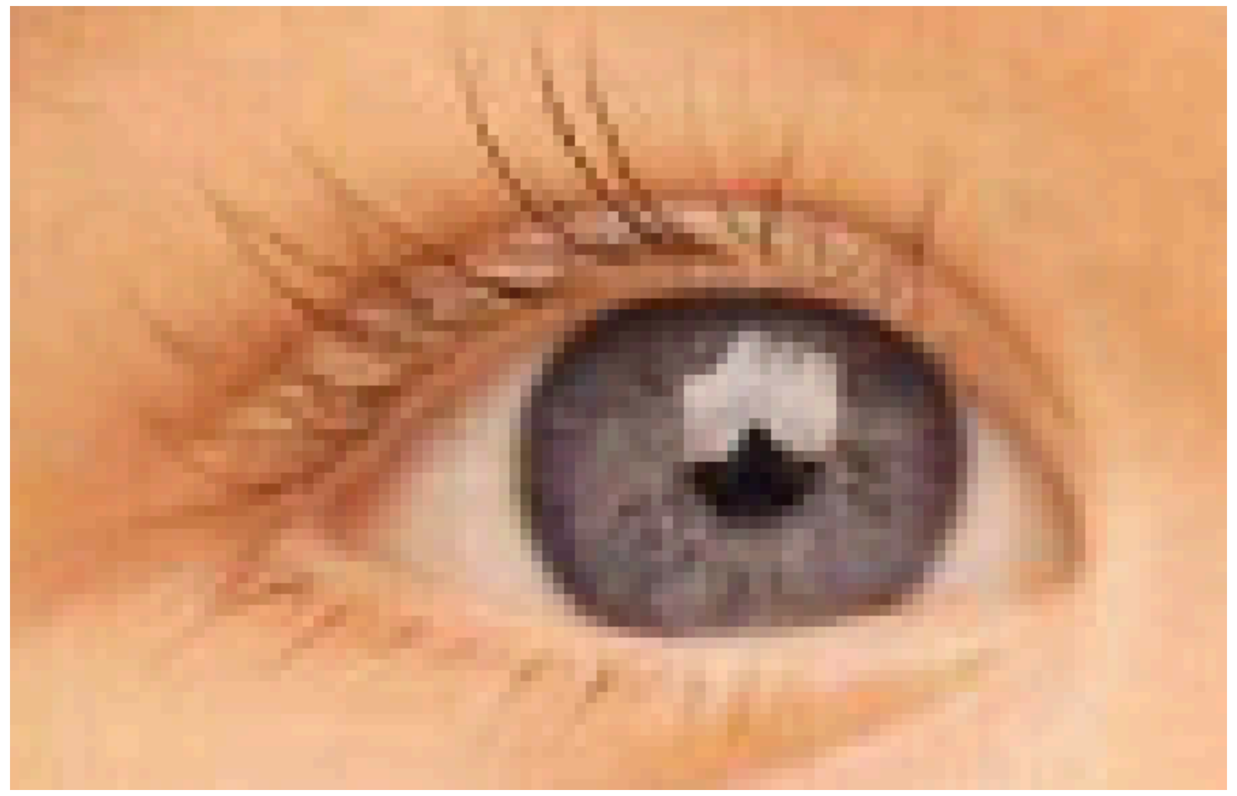
Output

$\hat{\mathbf{y}}$



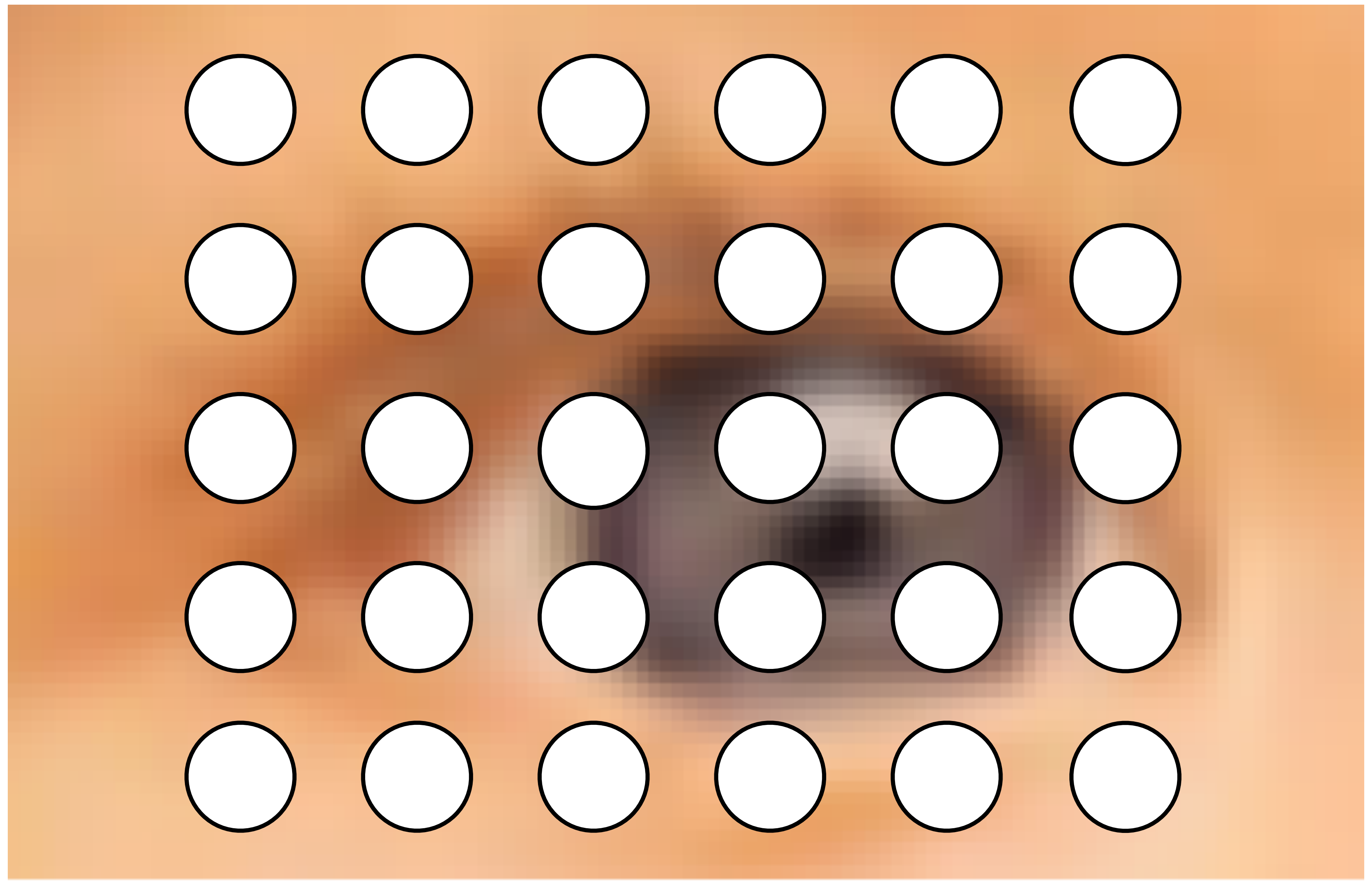
Target

$\mathbf{y}$



$$L(\hat{\mathbf{y}}, \mathbf{y}) = \|\hat{\mathbf{y}} - \mathbf{y}\|_2$$

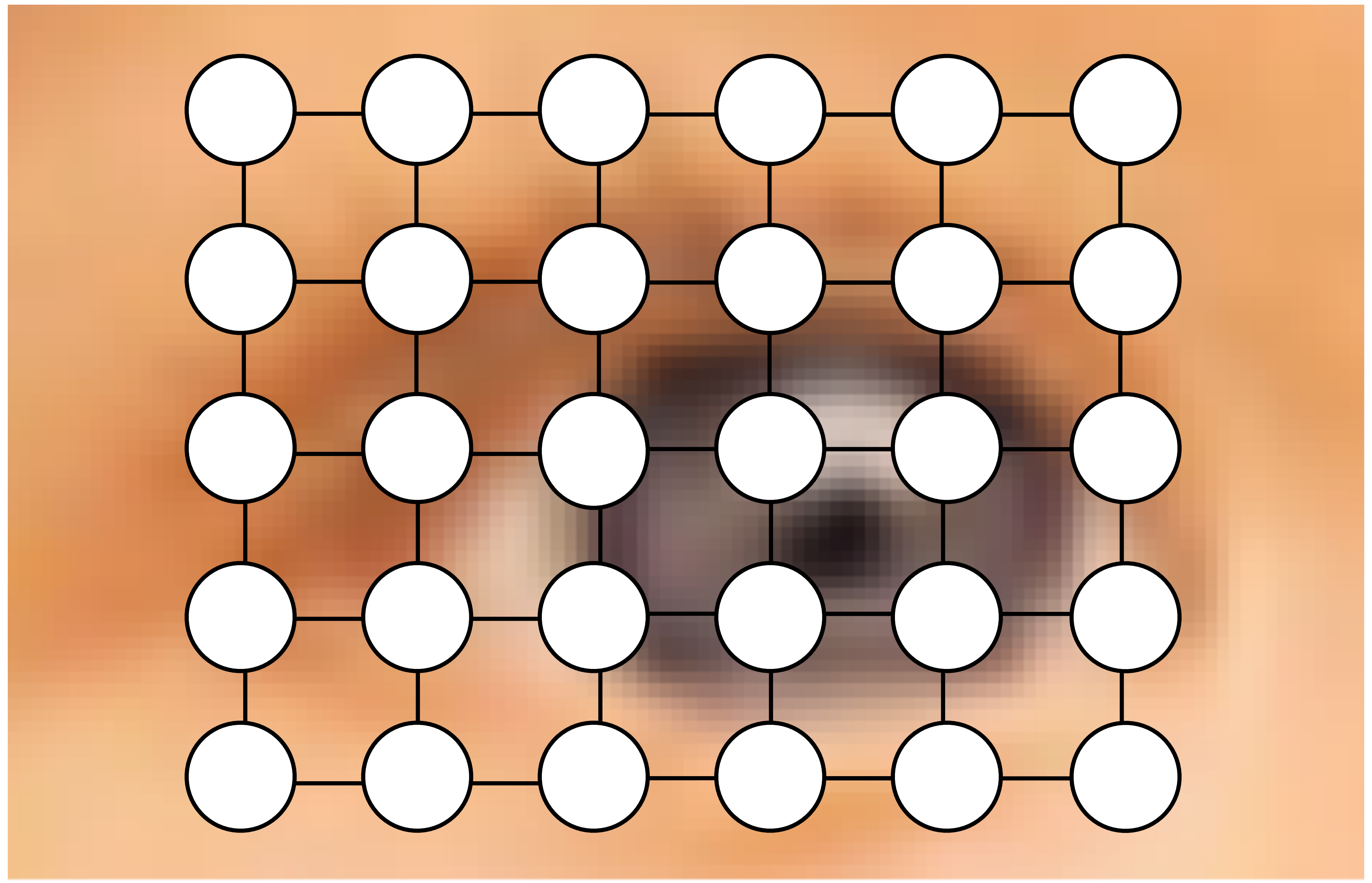
Structured Prediction



Each pixel treated as independent

$$\prod_i p(y_i|\mathbf{x})$$

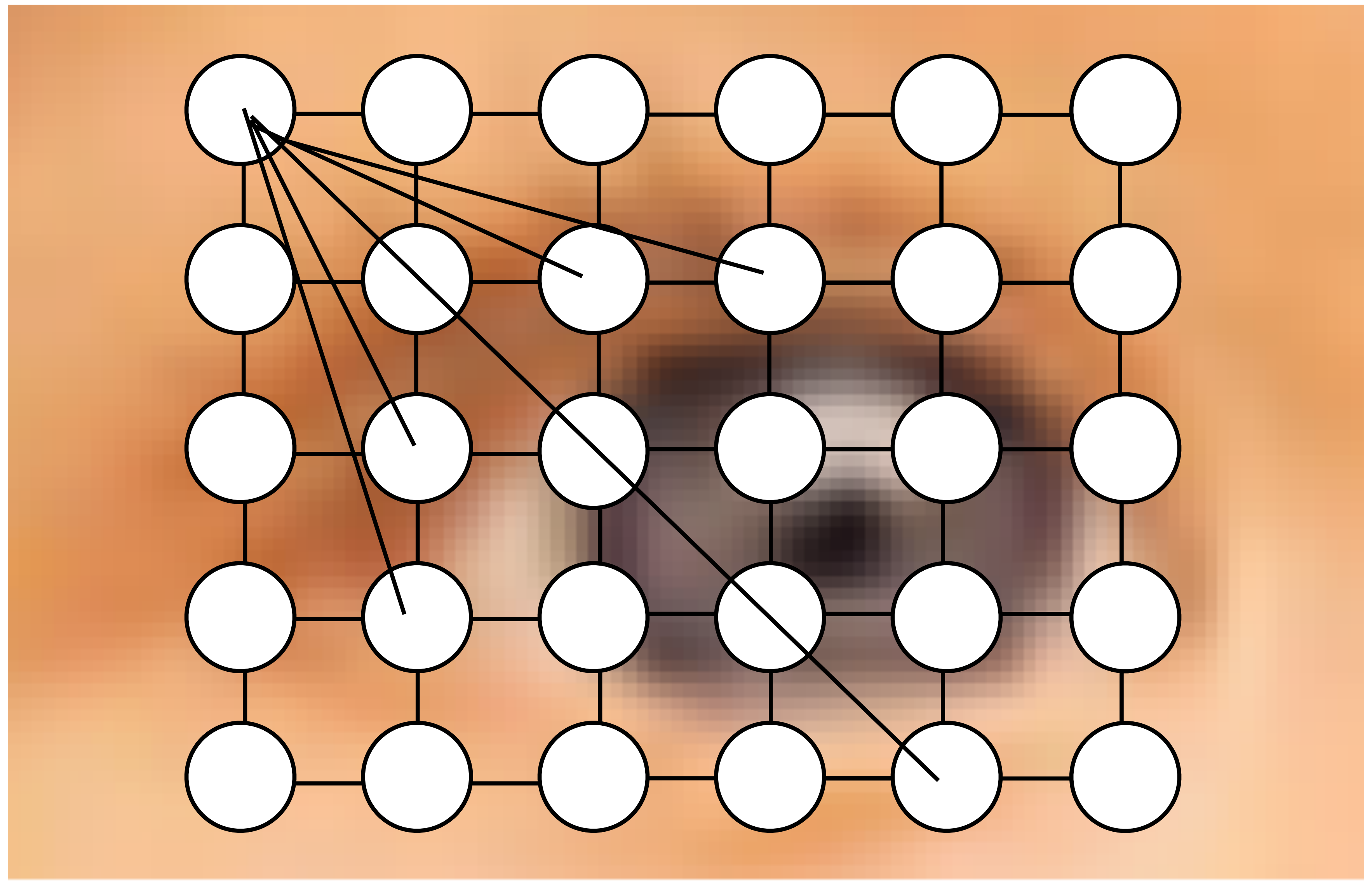
CRF



Models at pairwise configuration of pixels

$$\frac{1}{Z}\prod_{i,j} p(y_i, y_j|\mathbf{x})$$

Structured Prediction



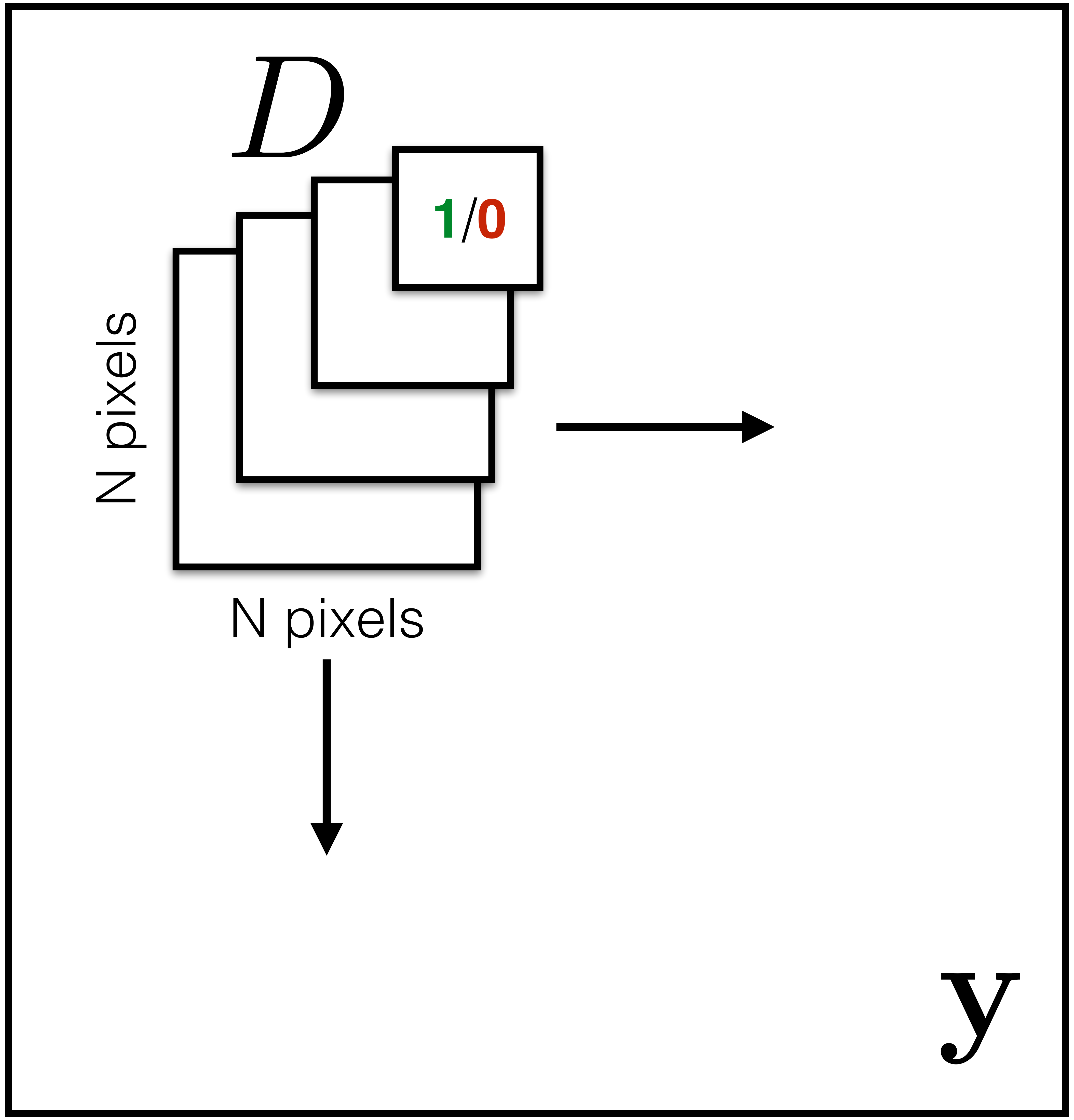
Model *joint* configuration of all pixels

$$p(\mathbf{y}|\mathbf{x})$$

A GAN, with sufficient capacity, samples from the full joint distribution when perfectly optimized.

Most generative models have this property! Give them **sufficient capacity** and **infinite data**, and they are the complete solution to prediction problems.

Shrinking the capacity: Patch Discriminator

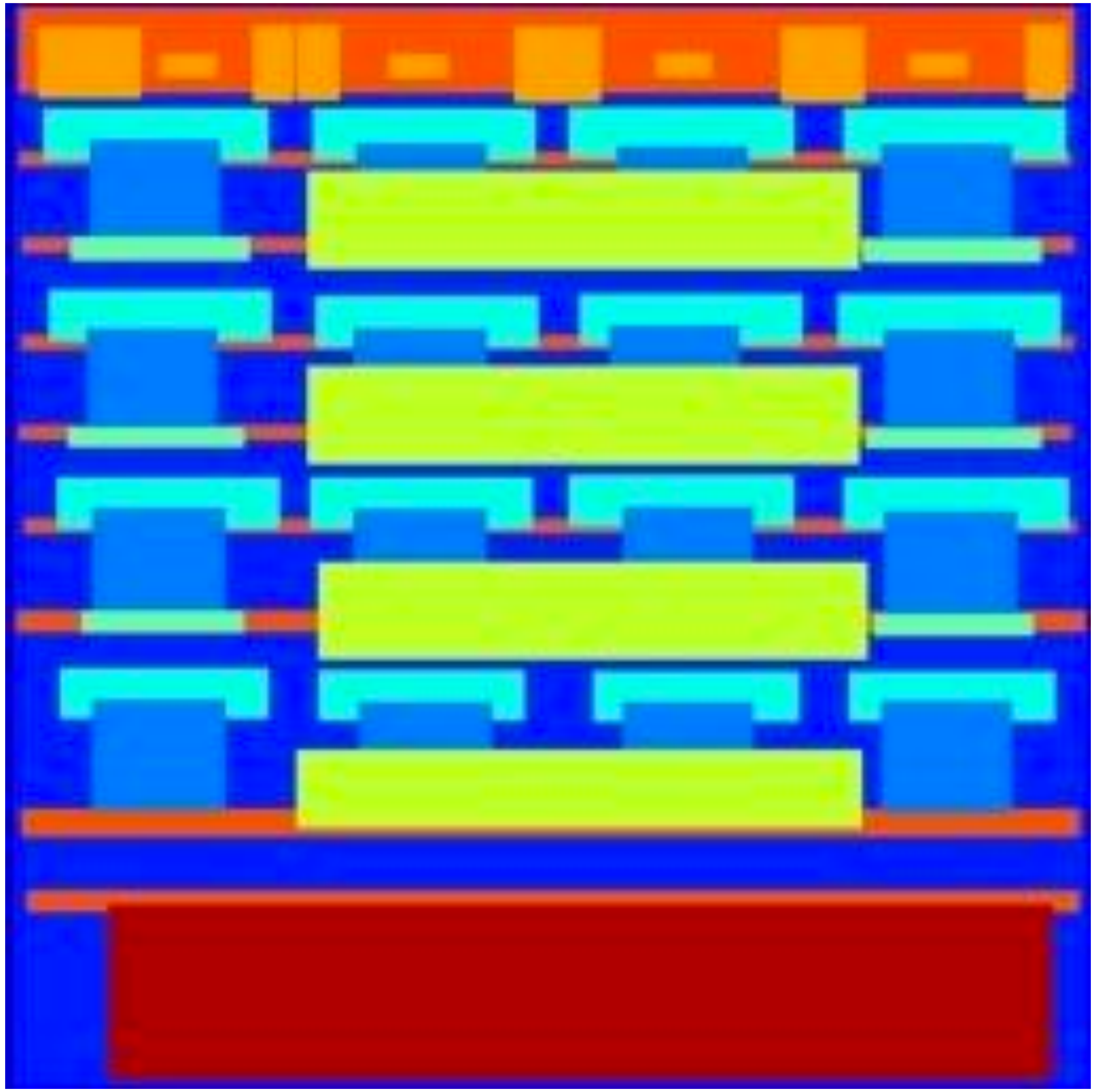


Rather than penalizing if output *image* looks fake, penalize if each overlapping *patch* in output looks fake

[Li & Wand 2016]
[Shrivastava et al. 2017]
[Isola et al. 2017]

Labels → Facades

Input

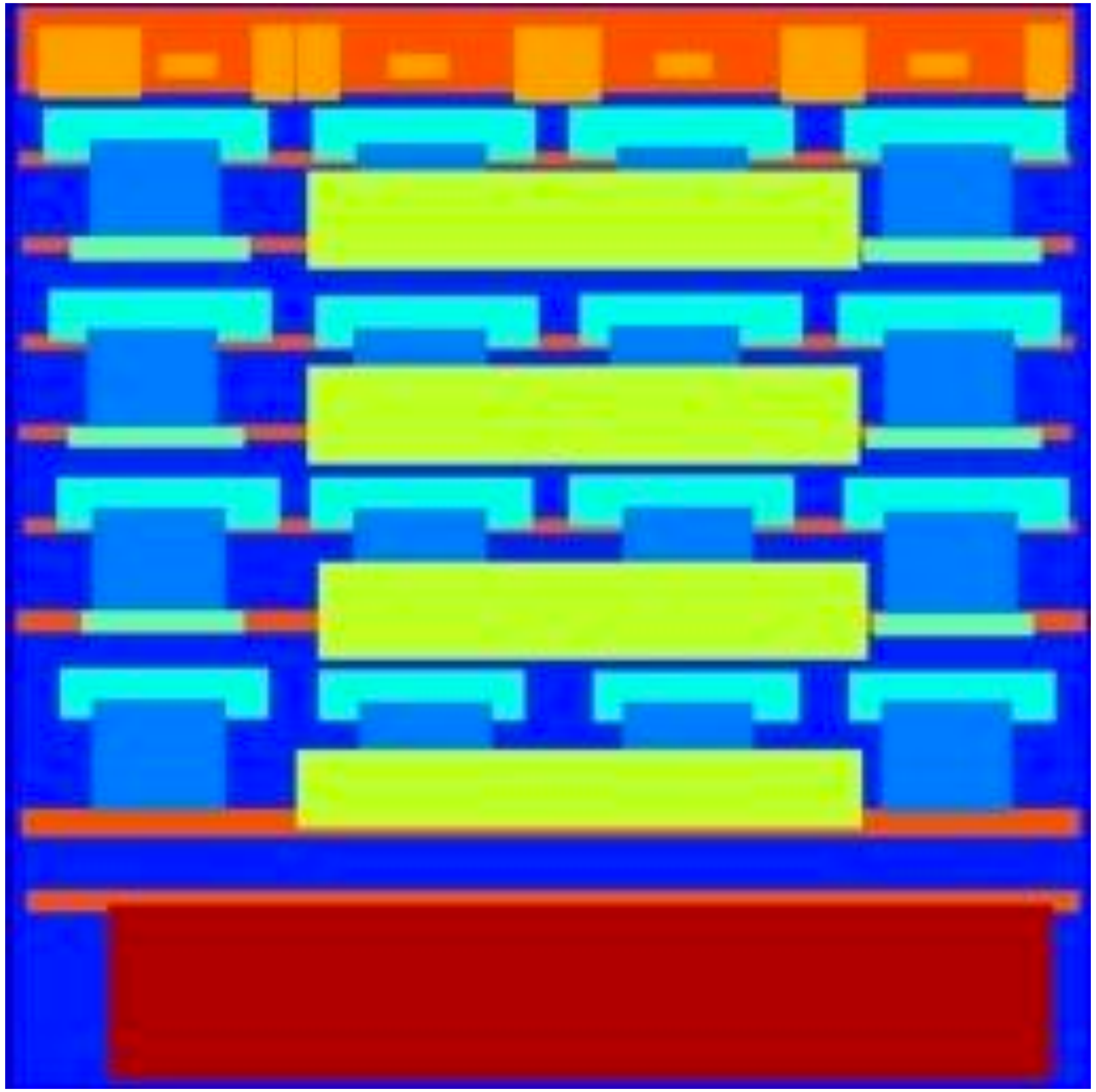


1x1 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

Input

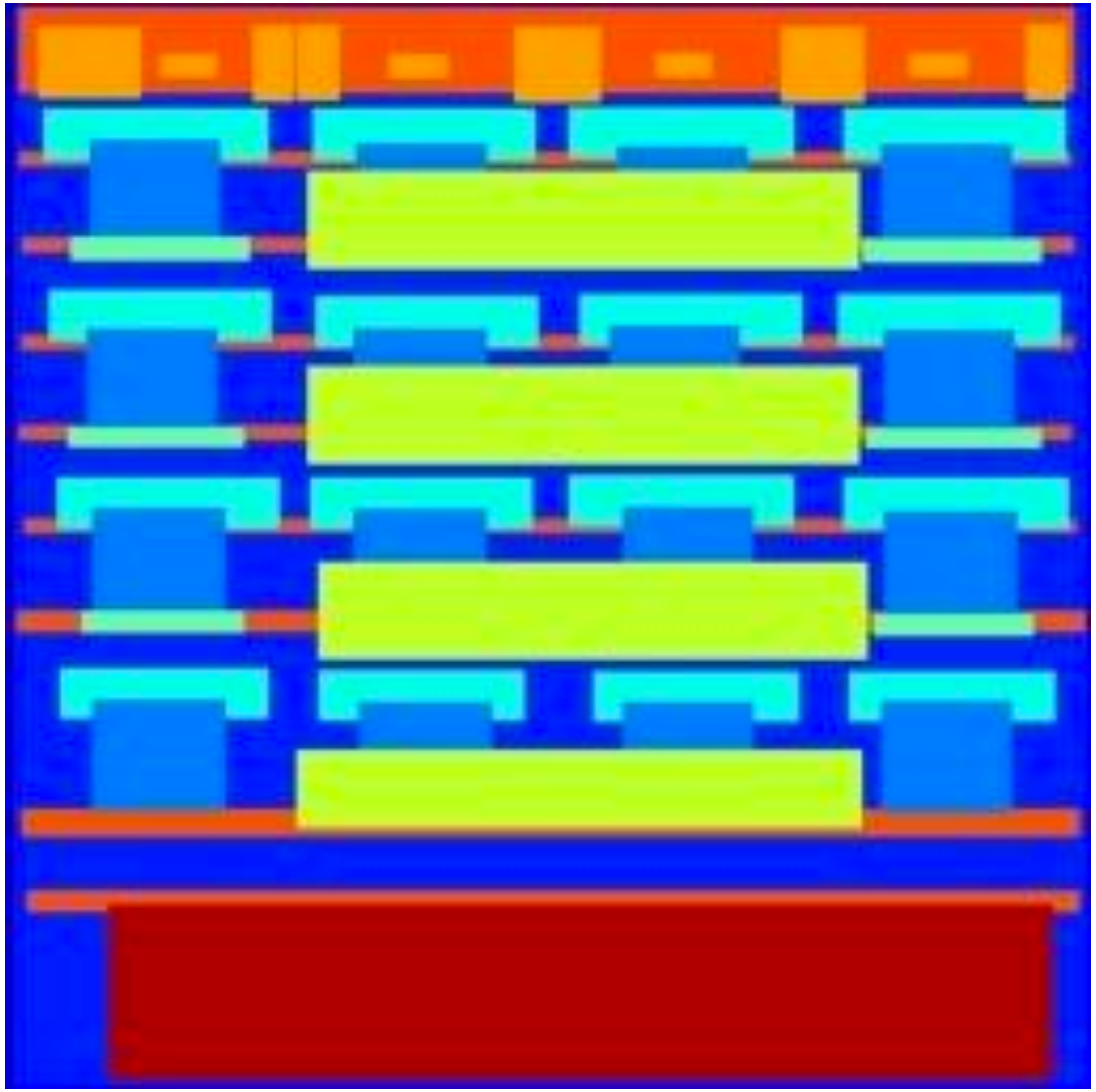


16x16 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

Input

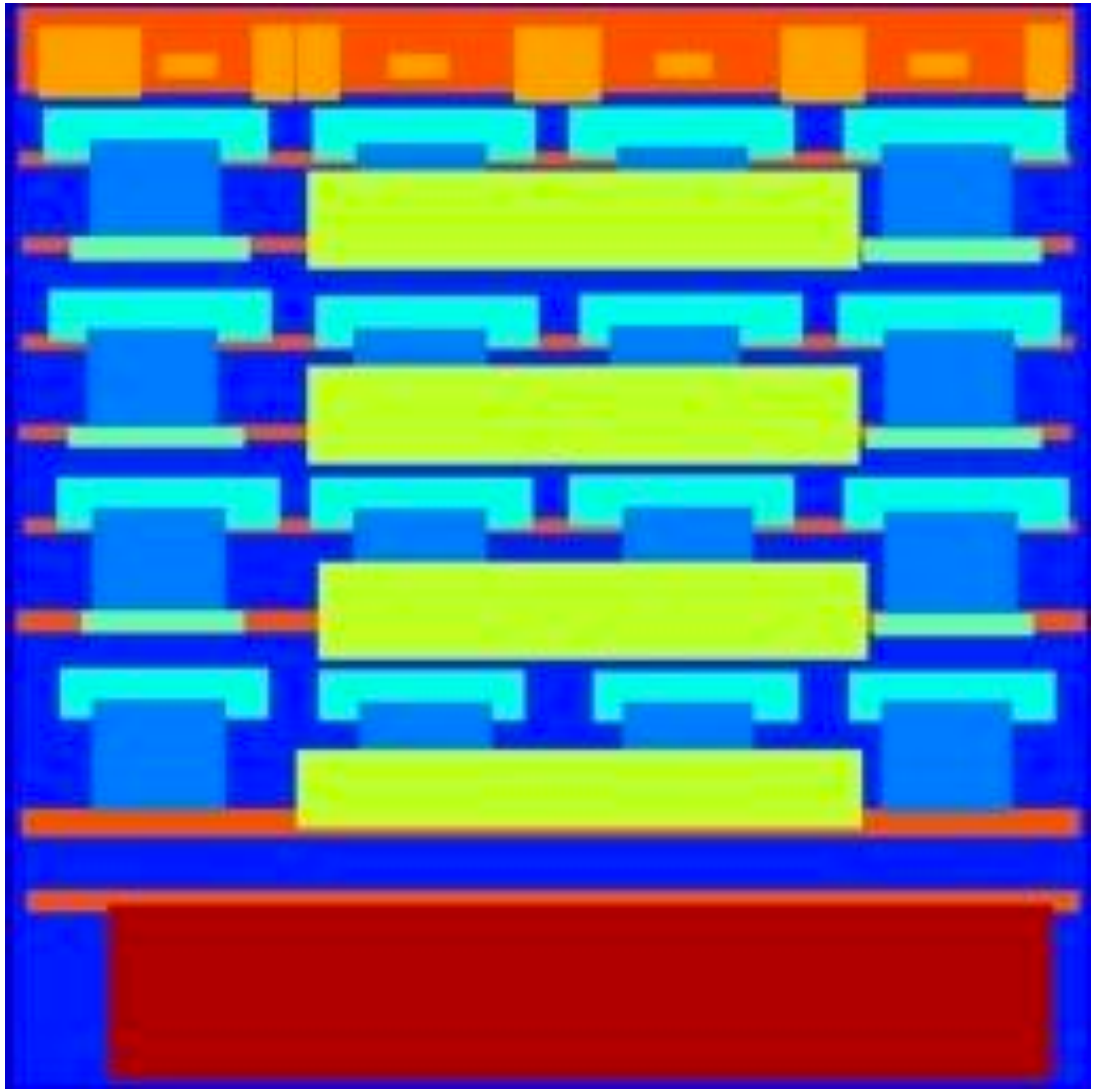


70x70 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

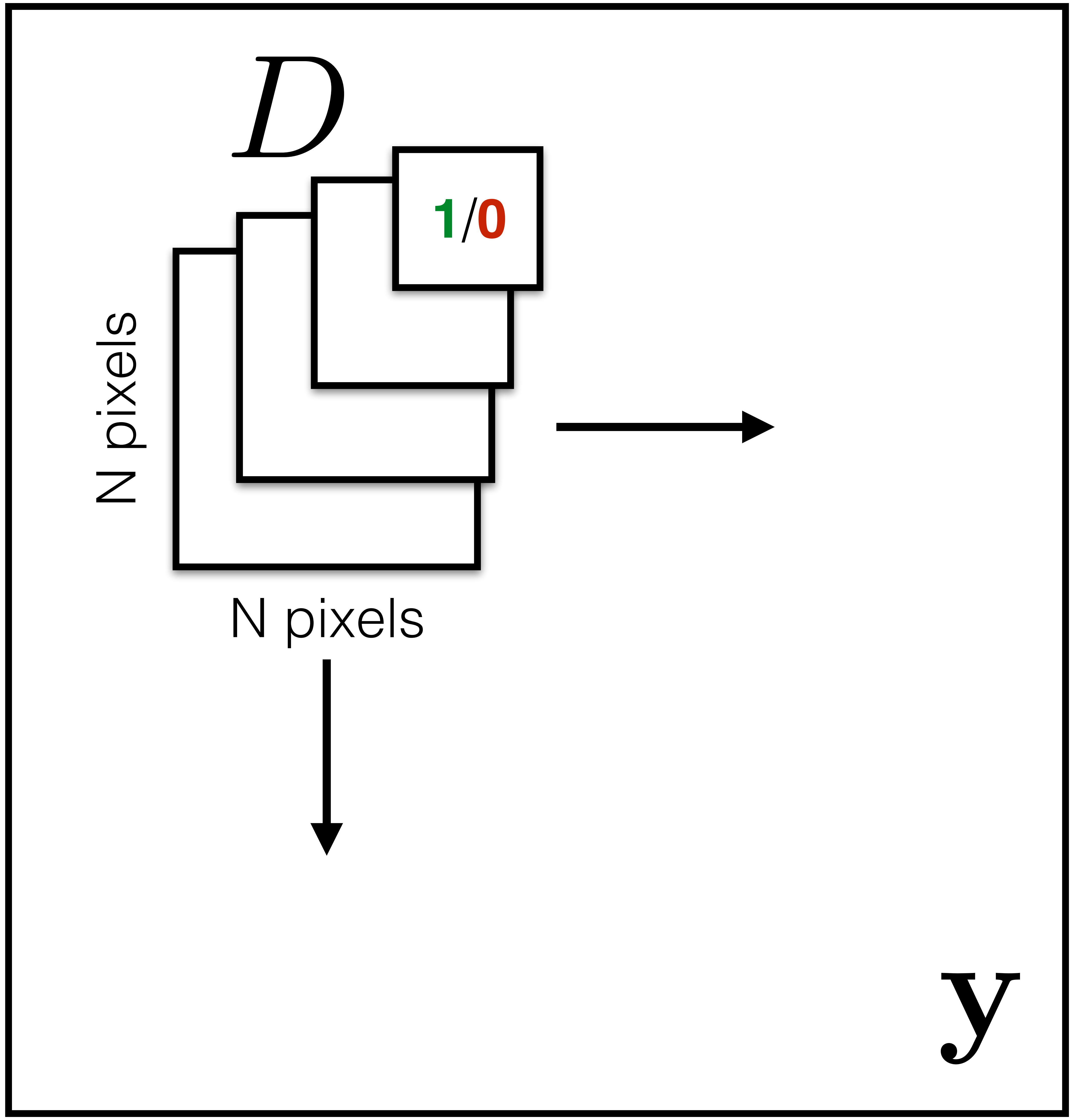
Input



Full image Discriminator

Data from [Tylecek, 2013]

Patch Discriminator



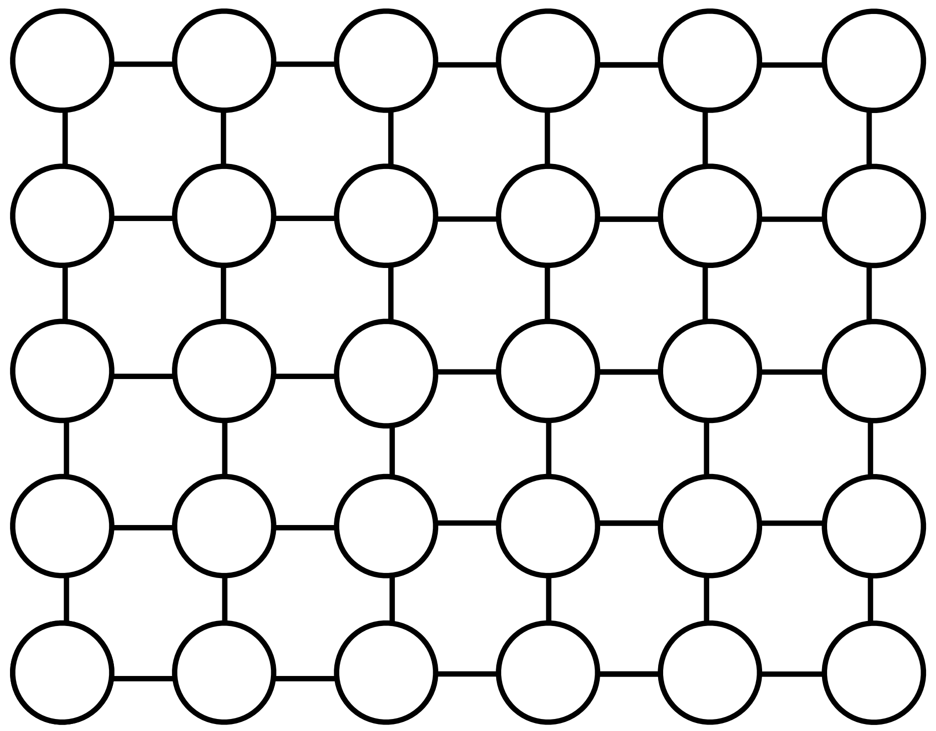
Rather than penalizing if output *image* looks fake, penalize if each overlapping *patch* in output looks fake

- Faster, fewer parameters
- More supervised observations
- Applies to arbitrarily large images

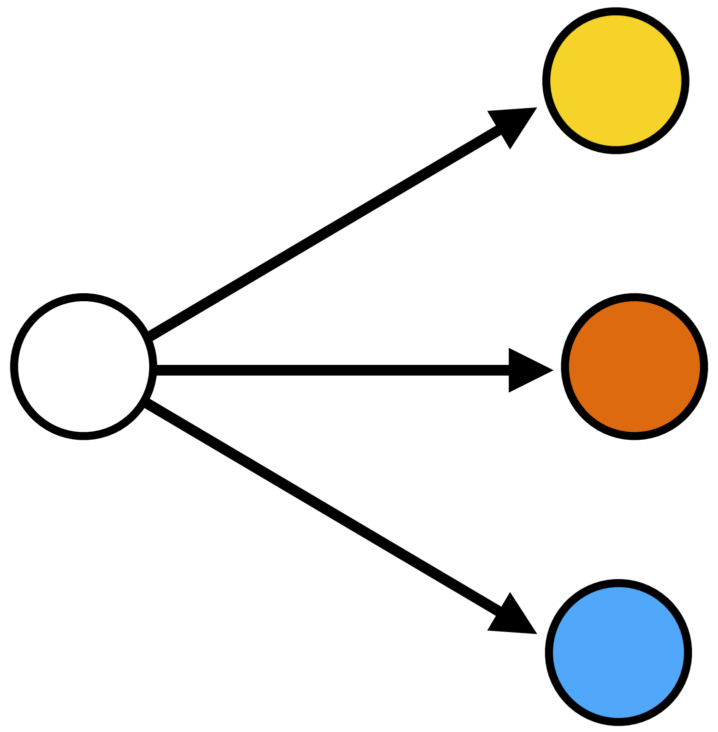
Properties of generative models

1. Model high-dimensional, structured output

    —> Use a deep net, D, to model output!



2. Model uncertainty; a whole distribution of possible outputs



Three perspectives on GANs

1. **Structured loss**
2. Generative model
3. Domain-level supervision / mapping

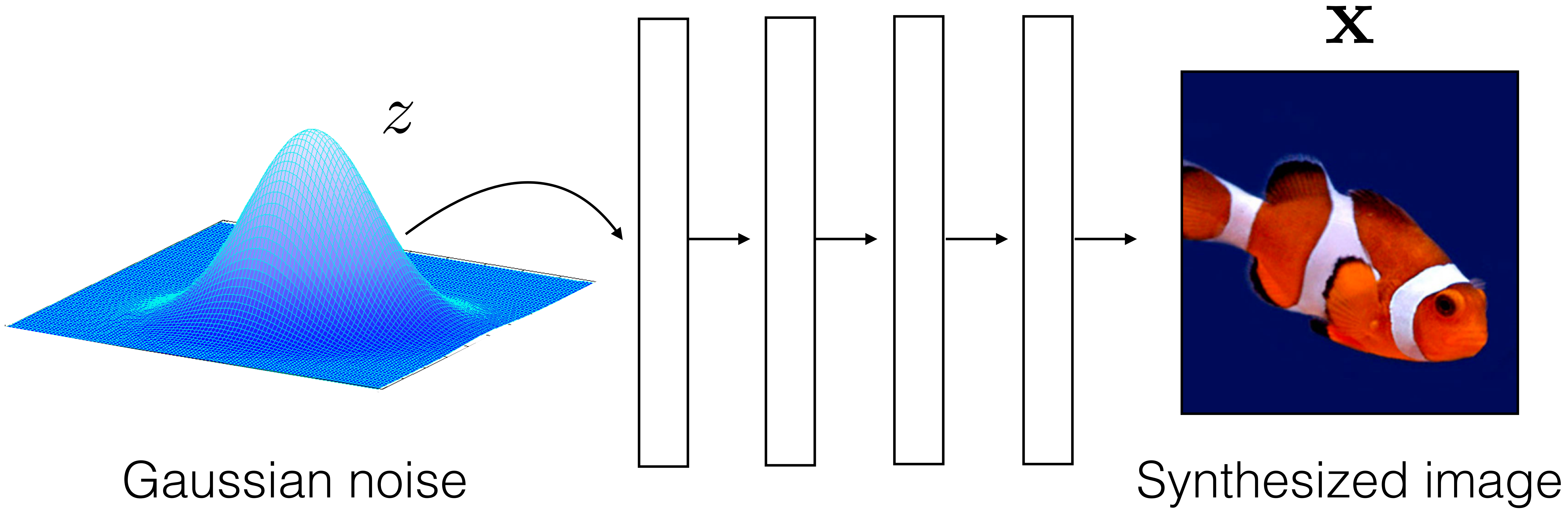
Three perspectives on GANs

1. Structured loss

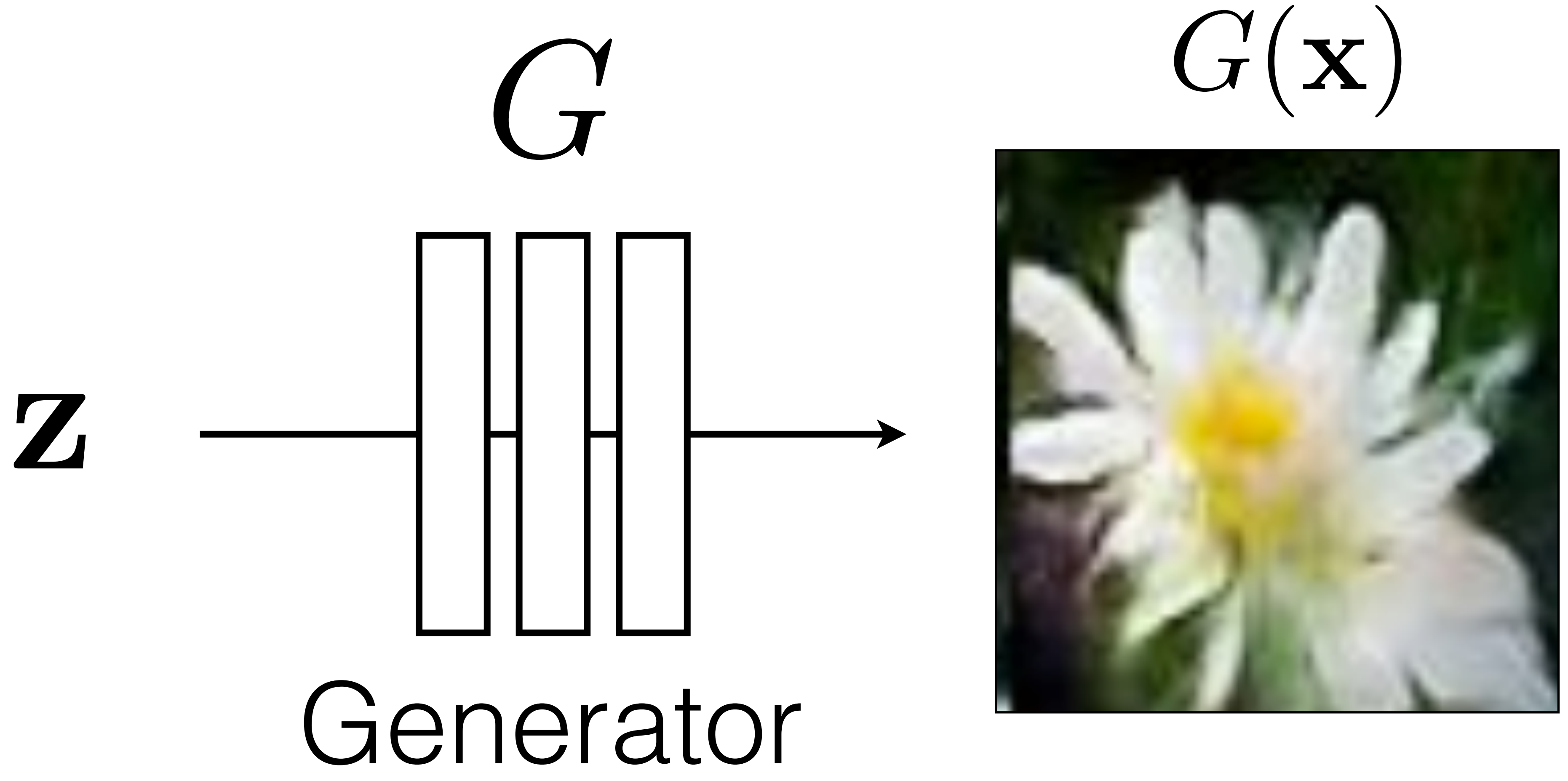
2. **Generative model**

3. Domain-level supervision / mapping

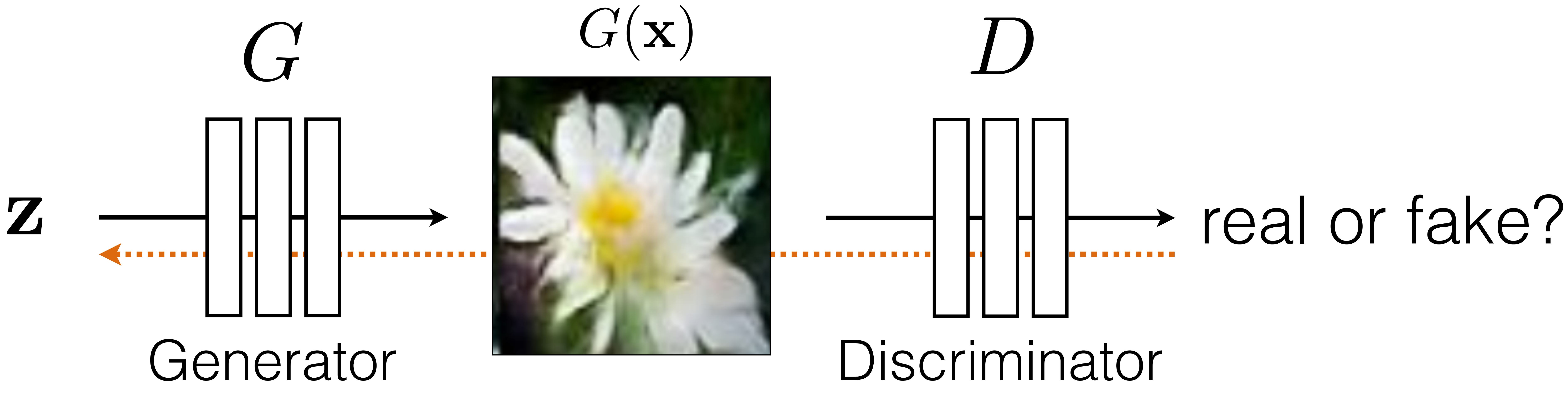
Can we generate images from scratch?



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



[Goodfellow et al., 2014]



G tries to synthesize fake images that fool **D**

D tries to identify the fakes

[Goodfellow et al., 2014]

GANs are implicit generative models

$p(\mathbf{x}) \longleftarrow$ "generative model" of the data $\mathbf{x}$

Noise distribution

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

$G(\mathbf{z})$

GAN

Data distribution

$\mathbf{x} \sim p(\mathbf{x})$

$G(\mathbf{z}) \sim p(\mathbf{x}) \longleftarrow$ Samples from a perfectly optimized GAN are samples from the data distribution

Progressive GAN [Karras et al., 2018]

Progressive GAN [Karras et al., 2018]

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

Proof

$$\begin{aligned} V(G, D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned}$$

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \rightarrow a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{\text{data}}) \cup Supp(p_g)$, concluding the proof. $\square$

$p_g = p_{data}$ is the unique global minimizer of the GAN objective.

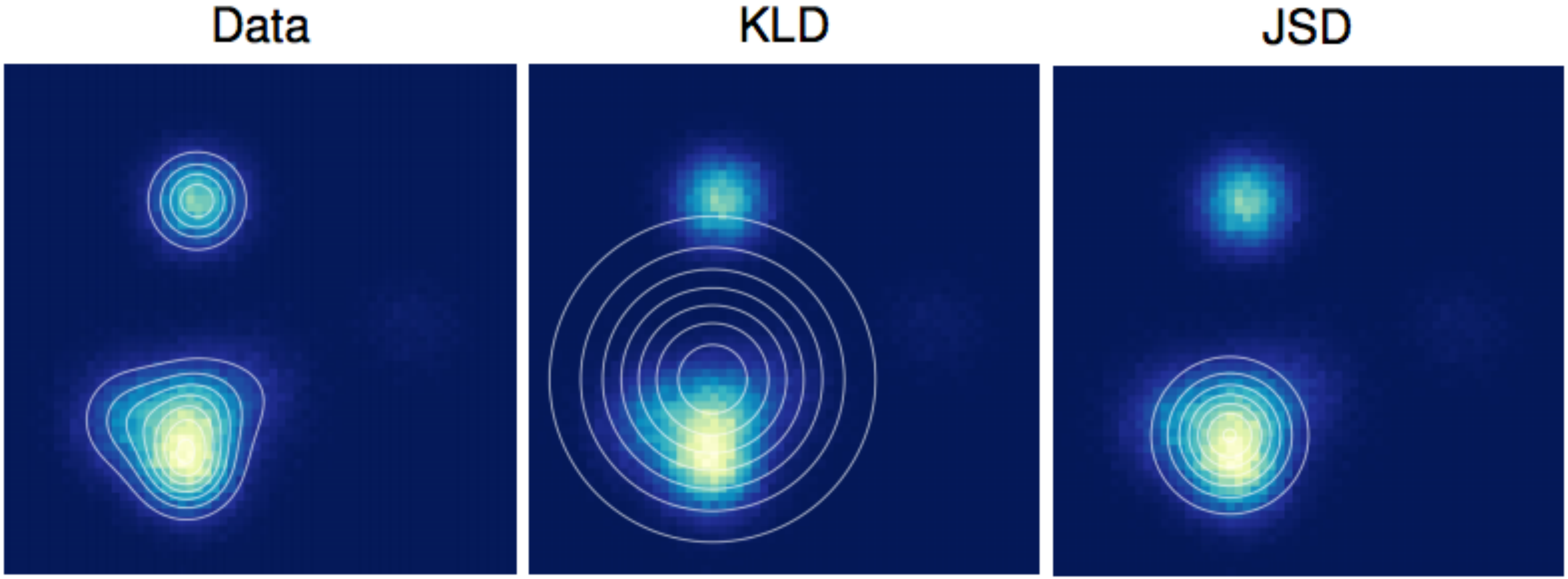
Proof

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D^*_G(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D^*_G(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D^*_G(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D^*_G(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned}
$$

$$
C(G) = -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right)
$$

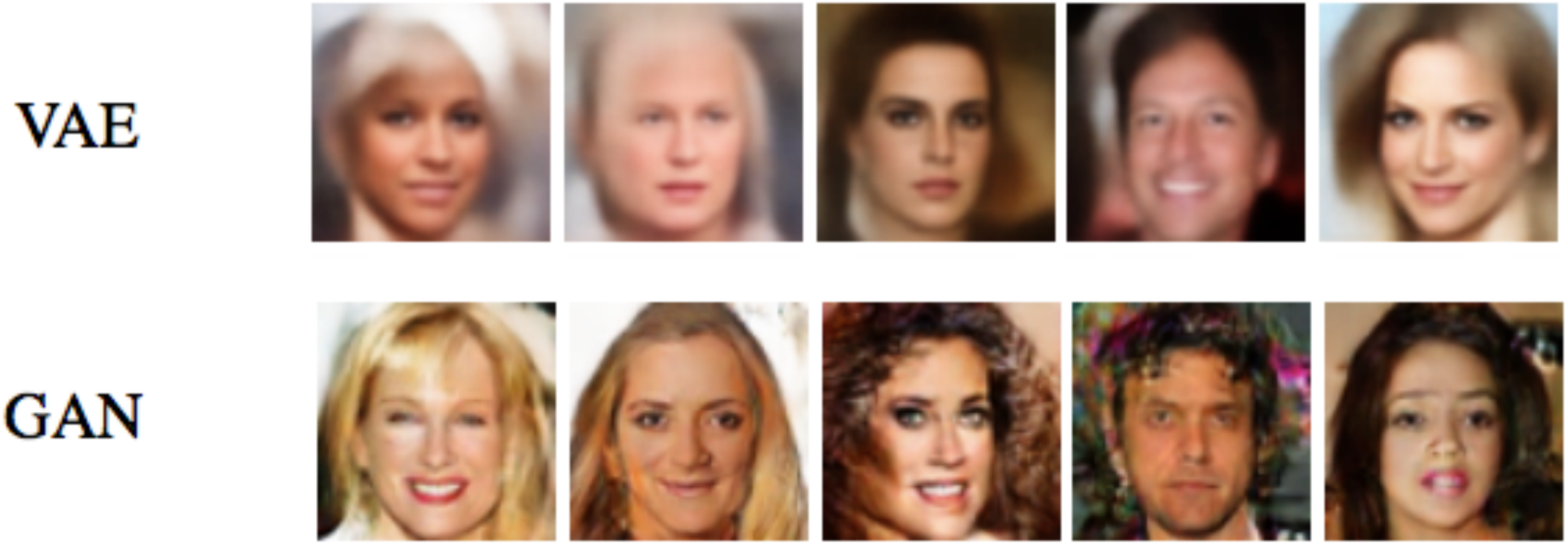
$$
C(G) = -\log(4) + 2 \cdot \underbrace{JSD\left(p_{\text{data}} \| p_g\right)}_{\geq 0, \quad 0 \iff p_g = p_{data}} \quad \square
$$

Behavior under model misspecification

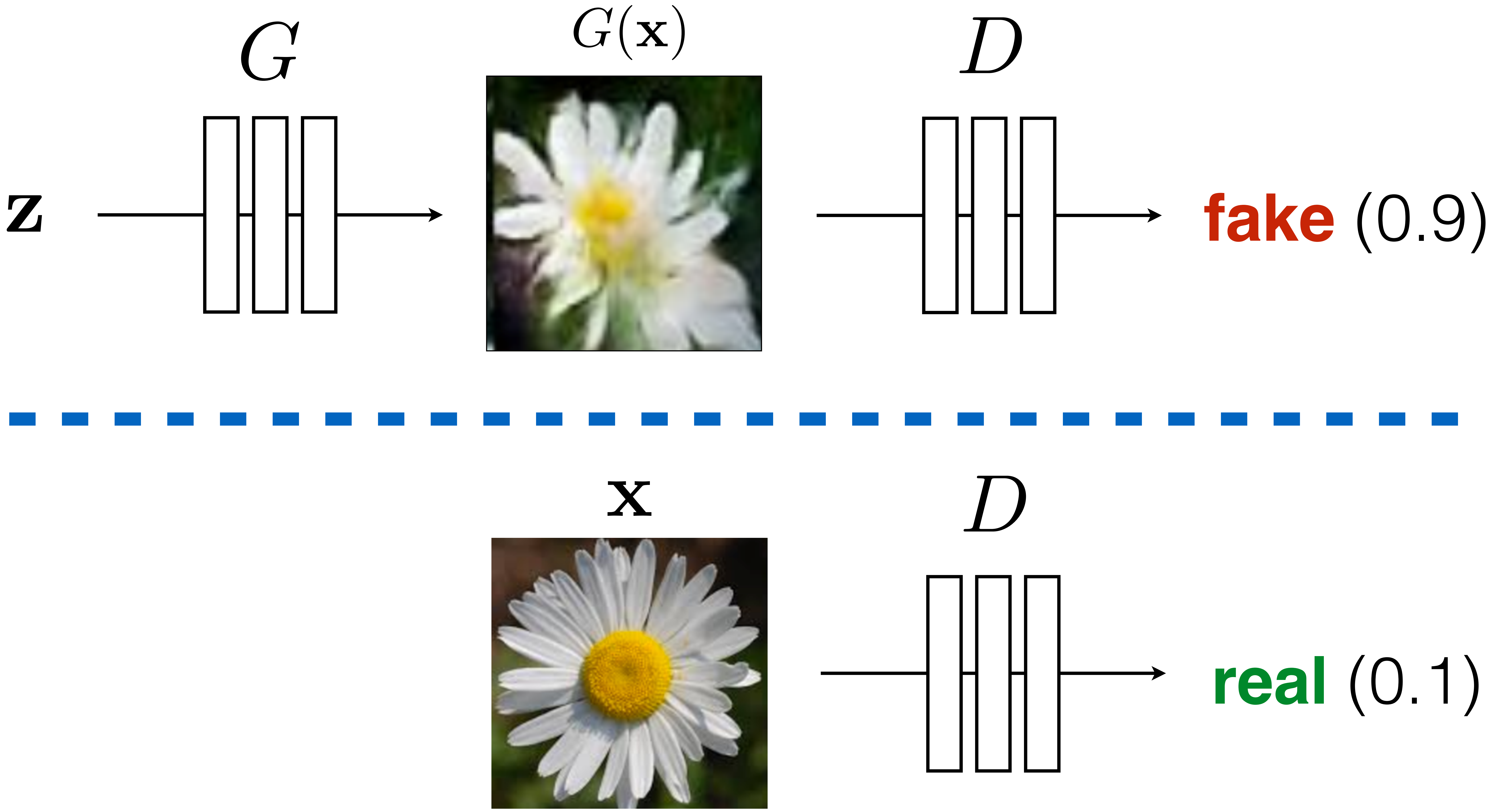


[Theis et al. 2016]

Mode covering versus mode seeking

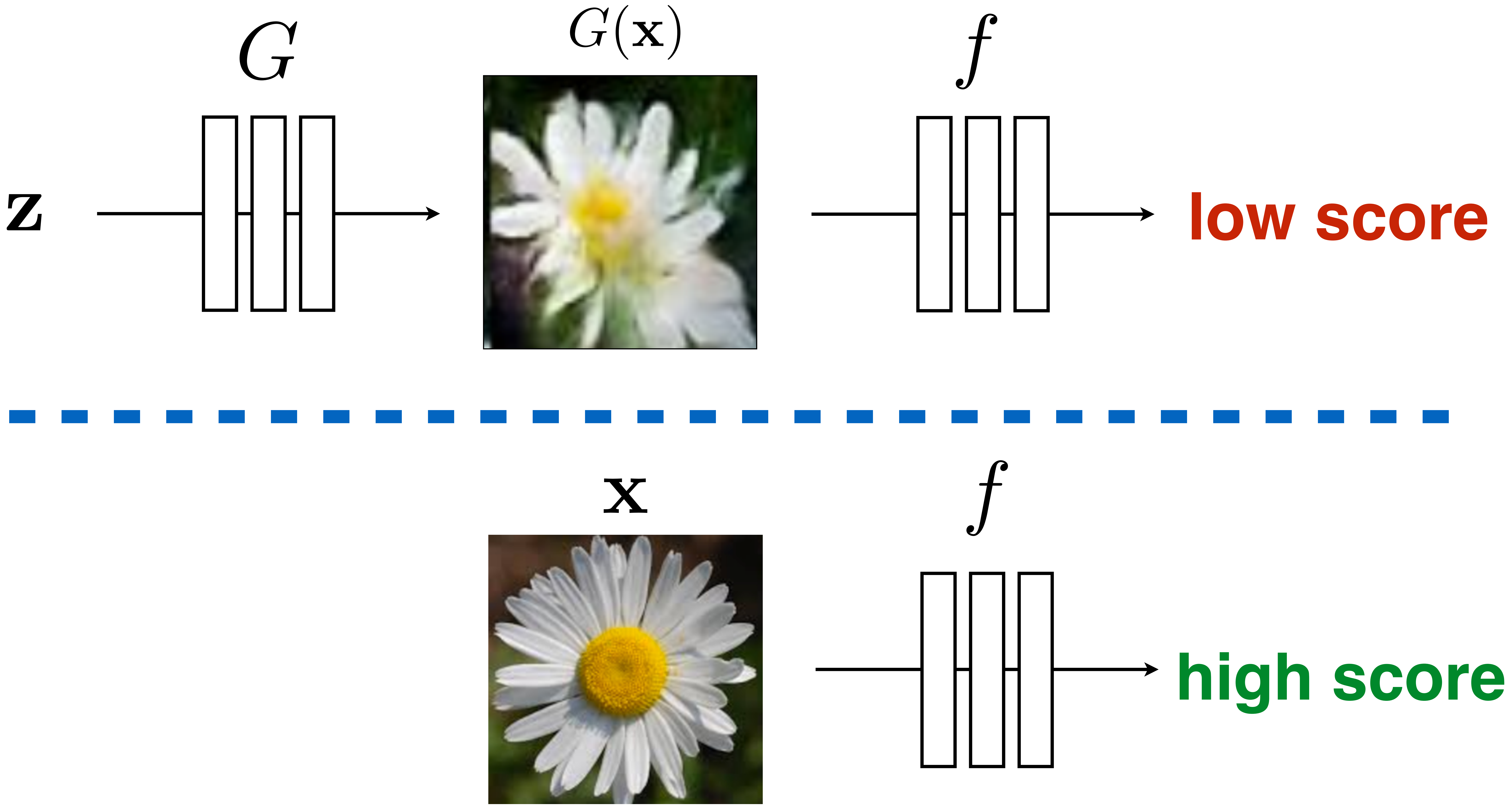


[Larsen et al. 2016]



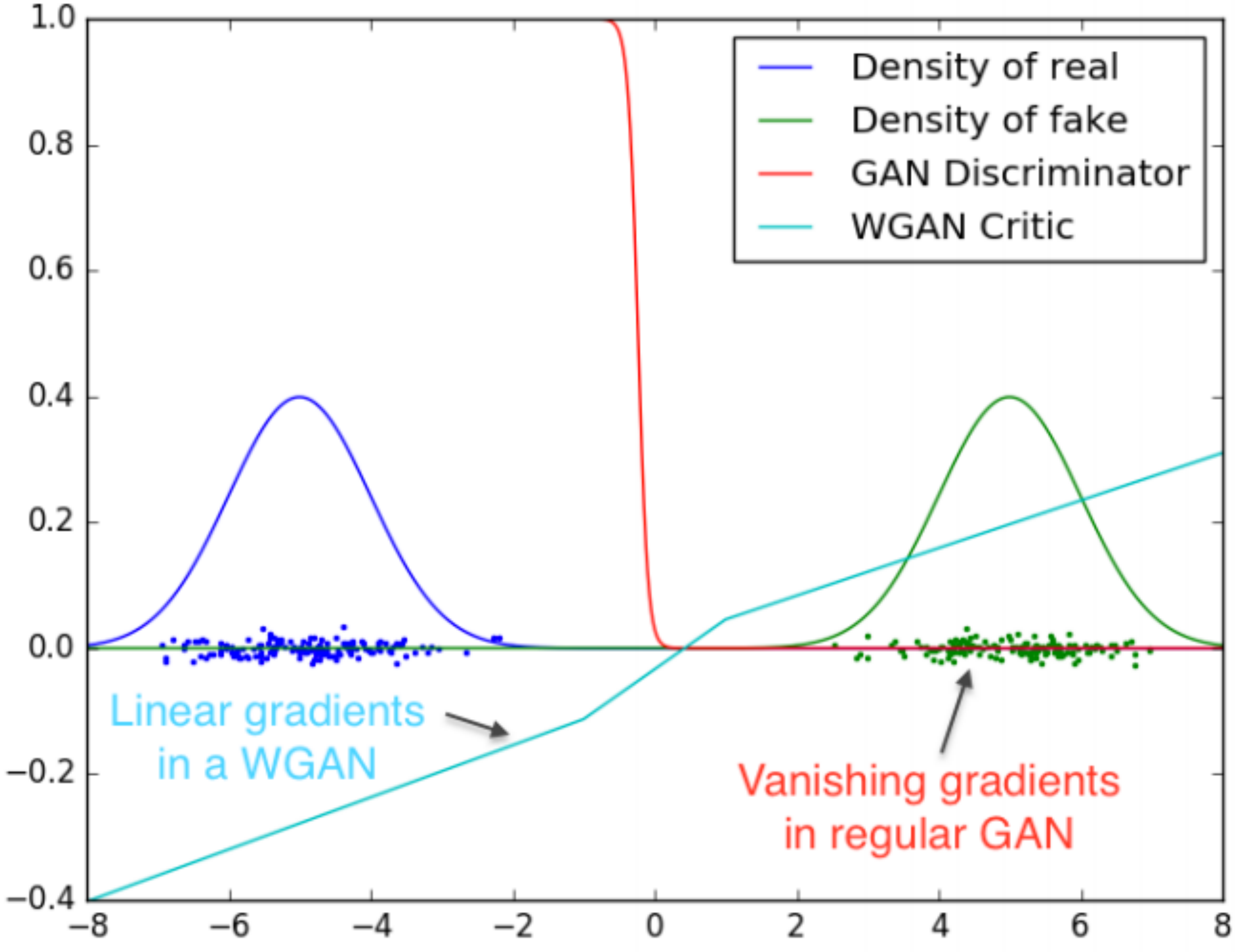
$$\arg\max_{D} \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\ \boxed{\log D(G(\mathbf{z}))} \ + \ \boxed{\log\left(1 - D(\mathbf{x})\right)}\ \right]$$

[Goodfellow et al., 2014]

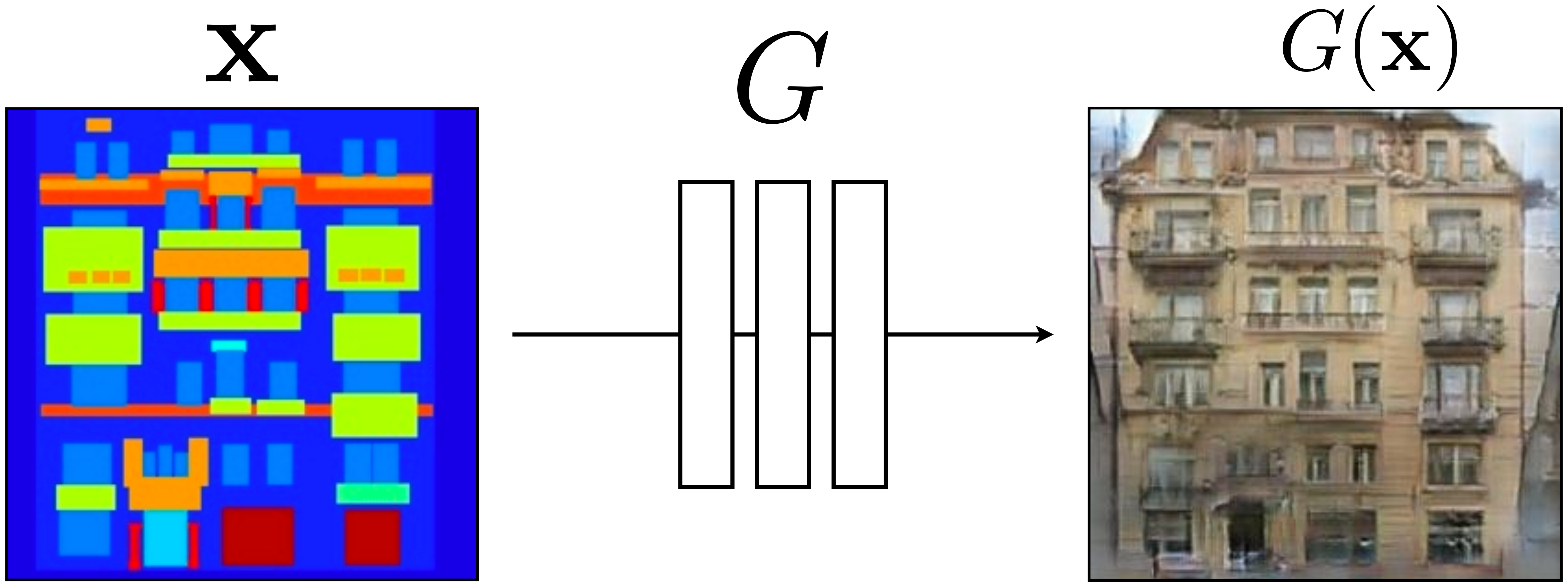


$$\arg\max_f \quad \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\ \boxed{-f(G(\mathbf{z}))} \ + \ \boxed{f(\mathbf{x})}\ \right]$$

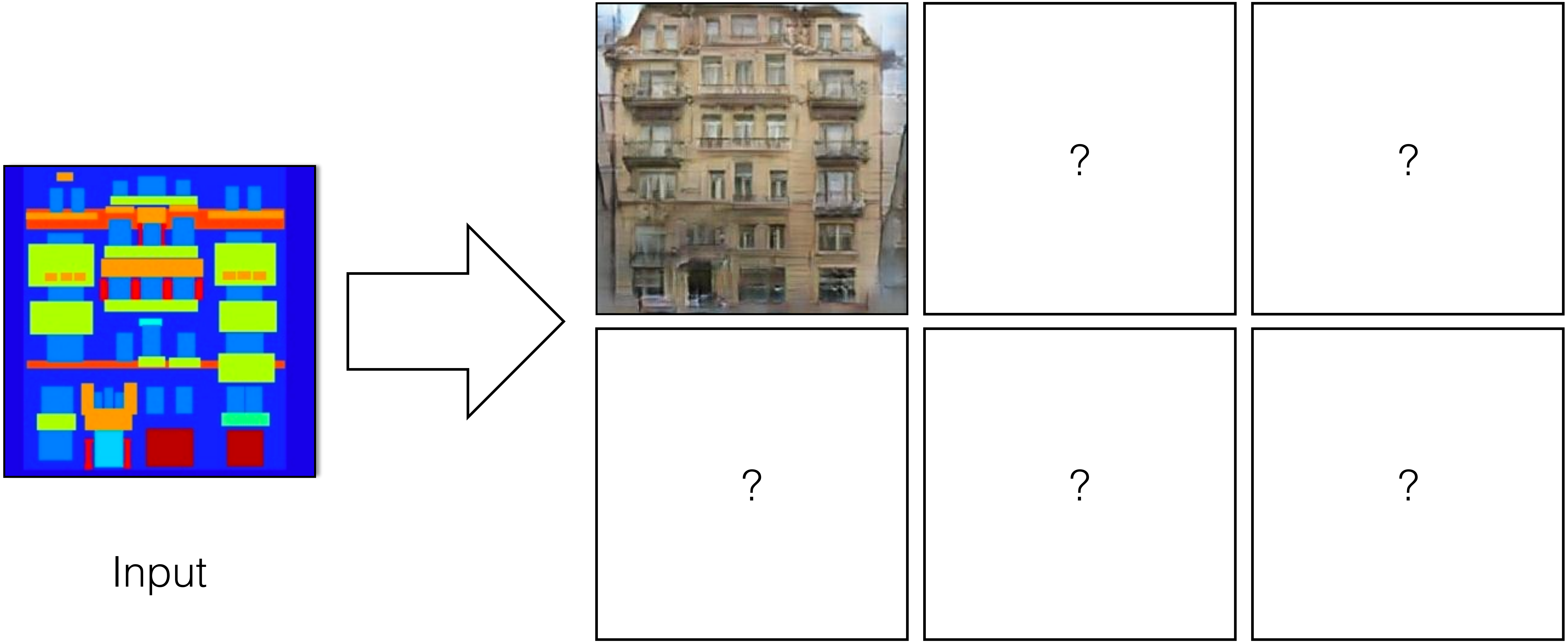
EBGAN, WGAN, LSGAN, etc


Density of real
Density of fake
GAN Discriminator
WGAN Critic
Linear gradients
in a WGAN
Vanishing gradients
in regular GAN
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−8
−6
−4
−2
0
2
4
6
8

Modeling multiple possible outputs



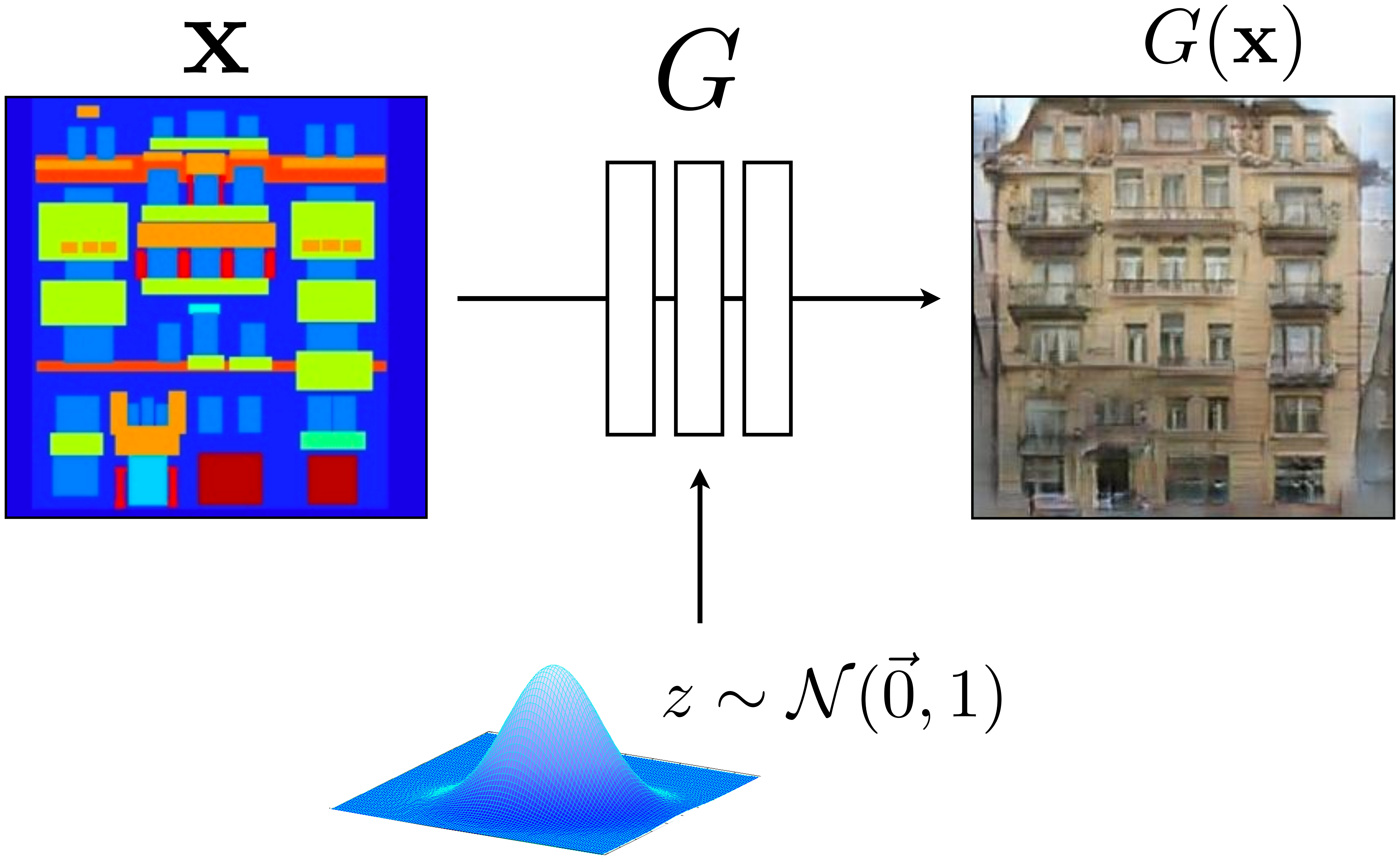
Modeling multiple possible outputs



Input

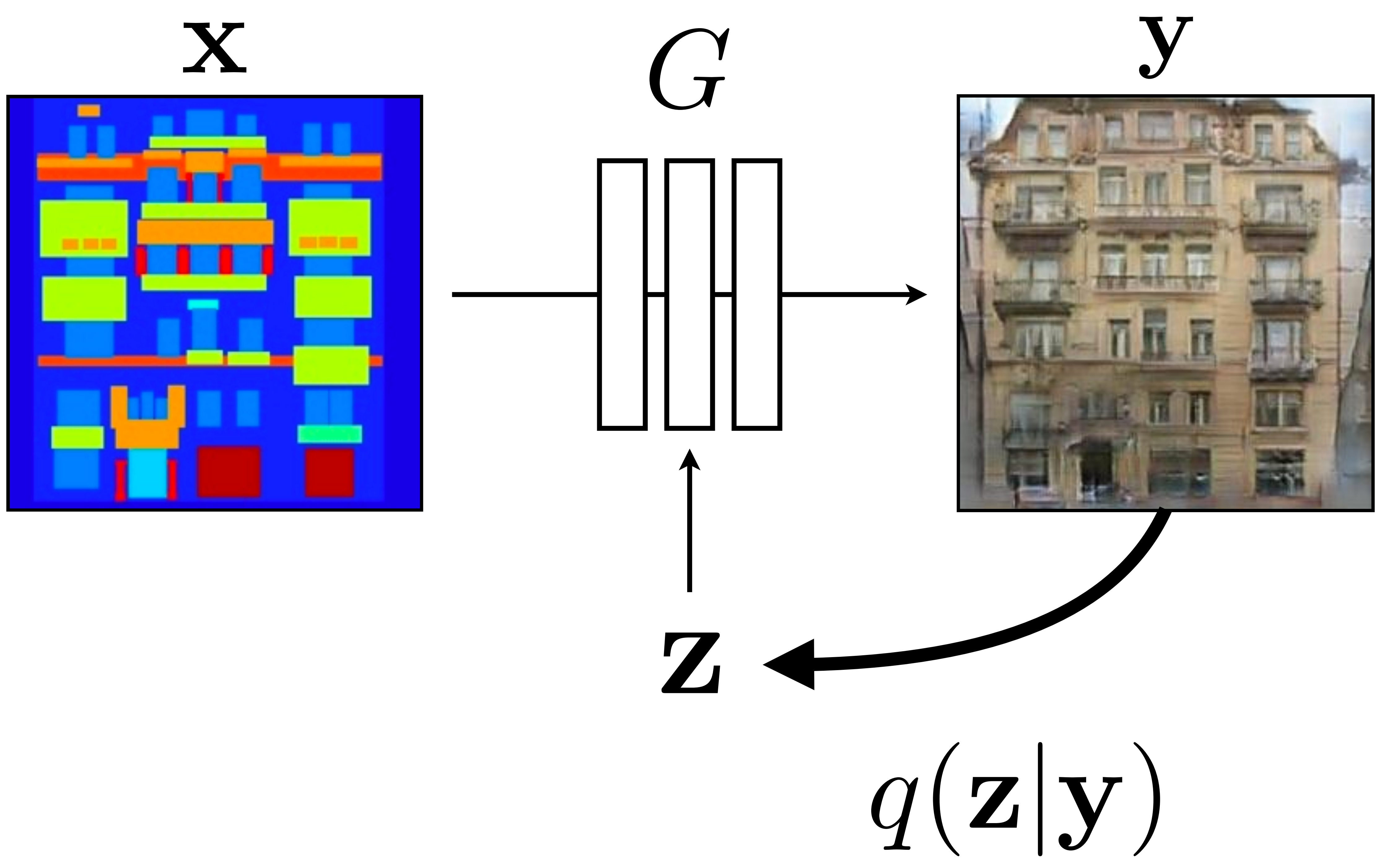
Possible outputs

Modeling multiple possible outputs

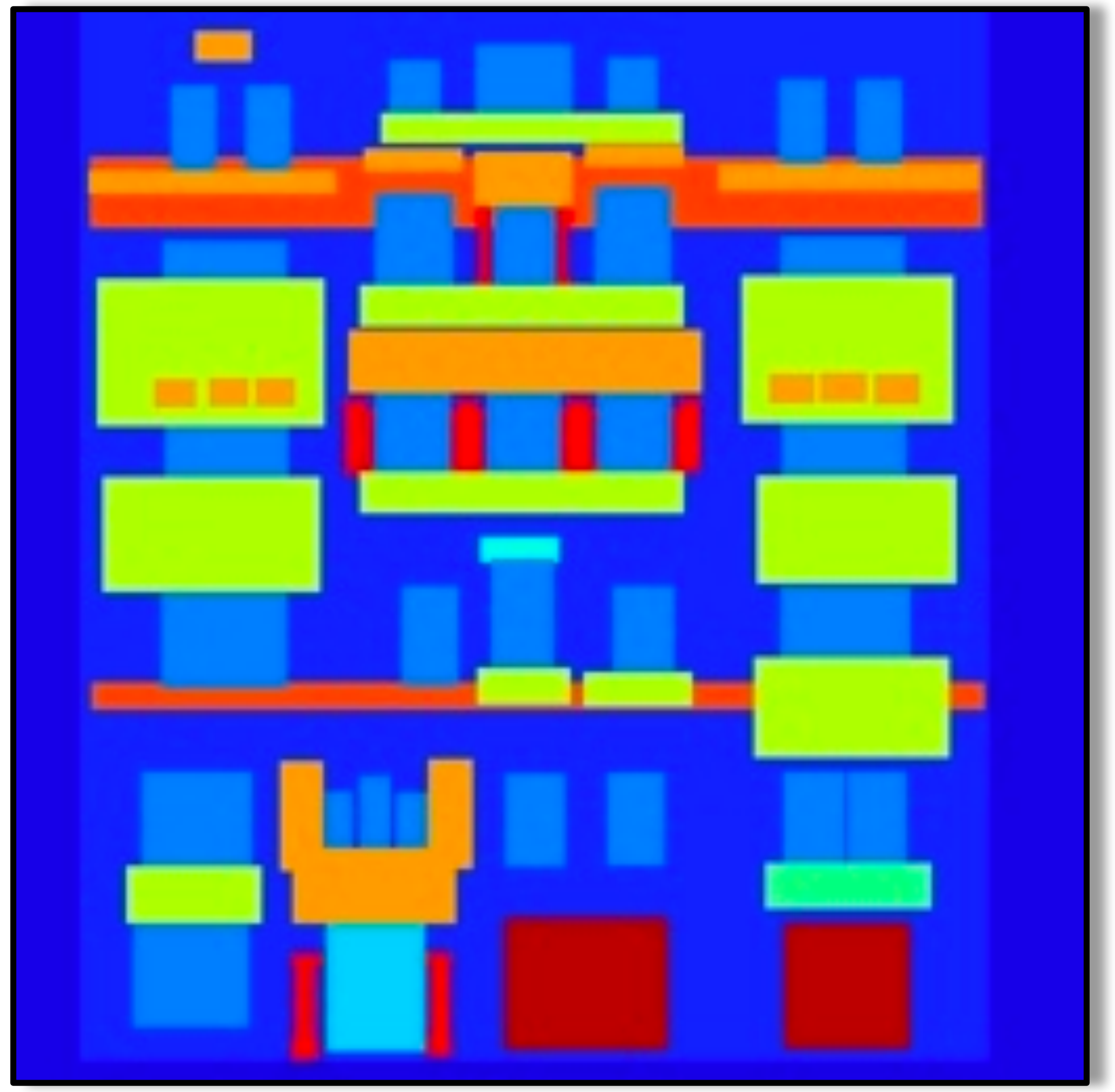


InfoGAN [Chen et al. 2016]

BiCycleGAN [Zhu et al., NIPS 2017]



Encourages z to relay information about the target.



Labels

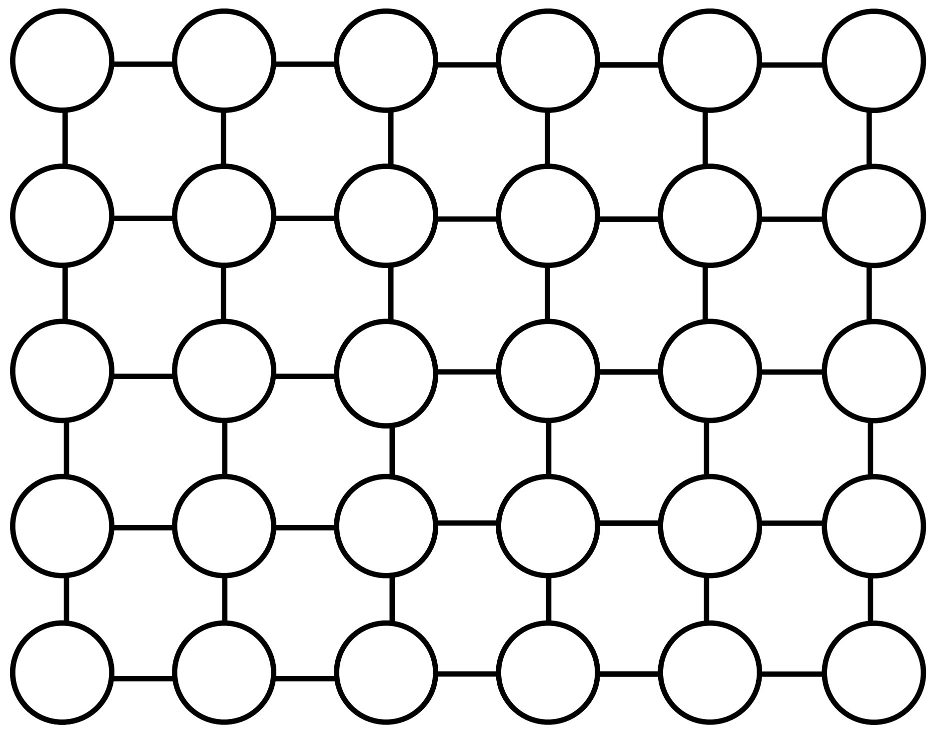
Randomly generated facades

[BiCycleGAN, Zhu et al., NIPS 2017]

Properties of generative models

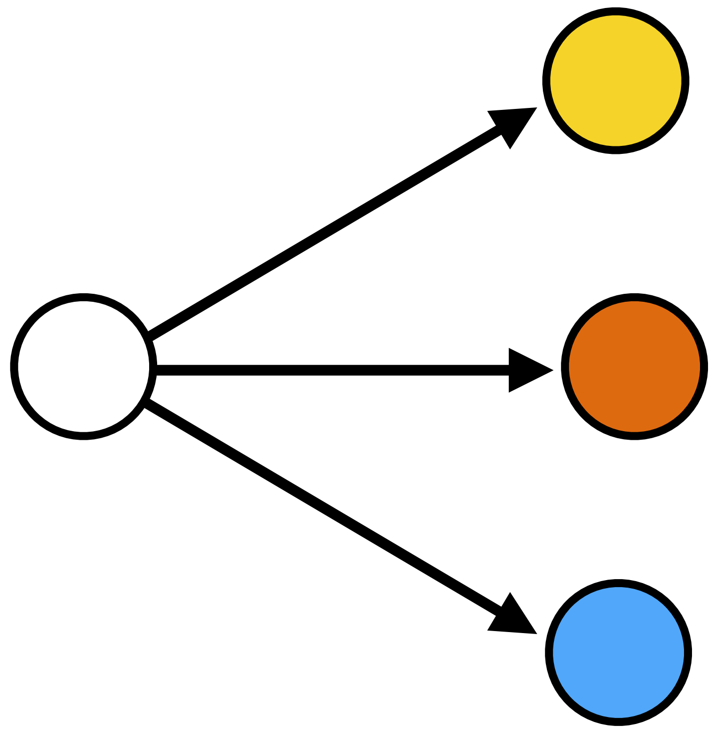
1. Model high-dimensional, structured output

   —> Use a deep net, D, to model output!



2. Model uncertainty; a whole distribution of possible outputs

   —> Generator is stochastic, learns to match data distribution



Three perspectives on GANs

1. Structured loss

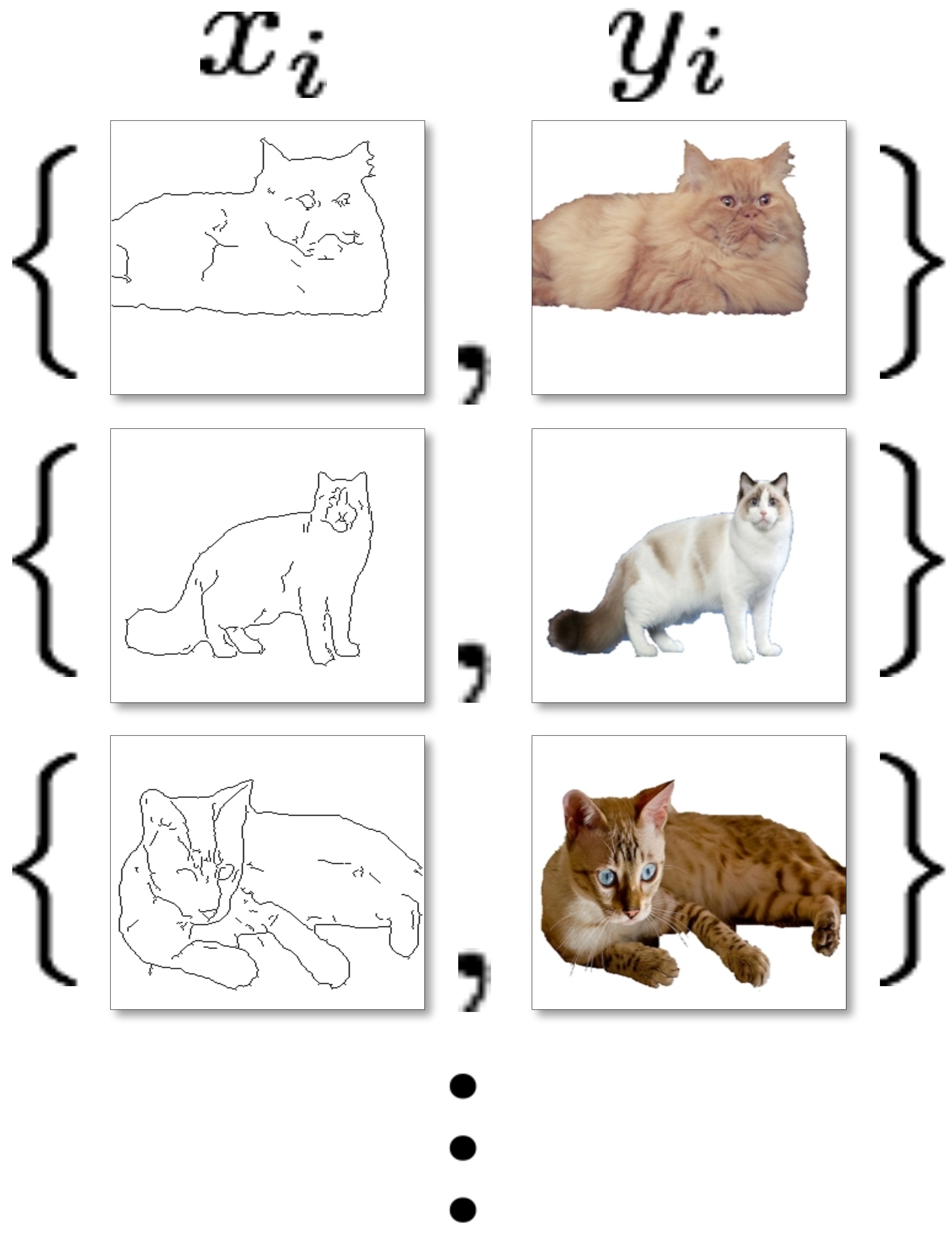
2. **Generative model**

3. Domain-level supervision / mapping

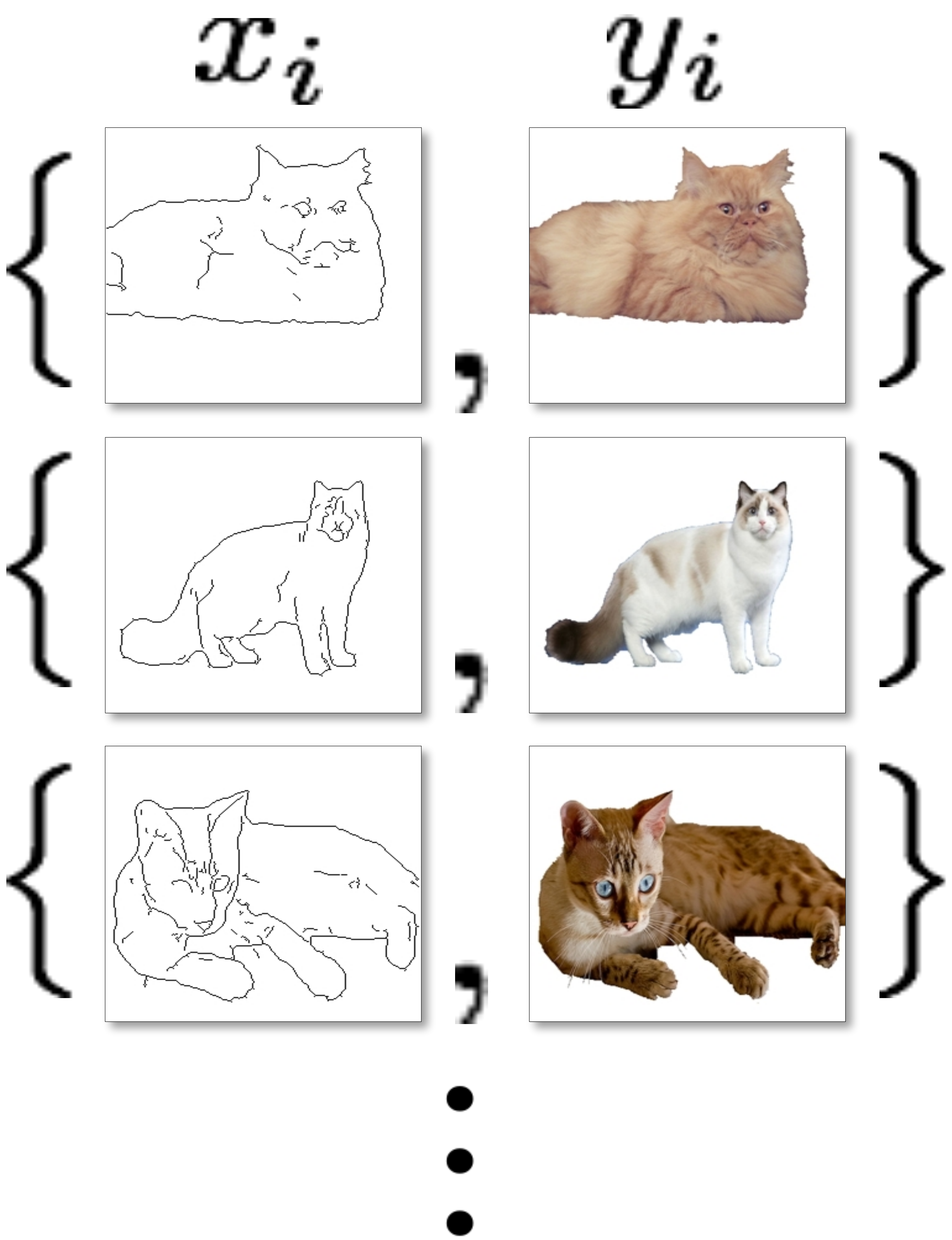
Three perspectives on GANs

1. Structured loss
2. Generative model
3. **Domain-level supervision / mapping**

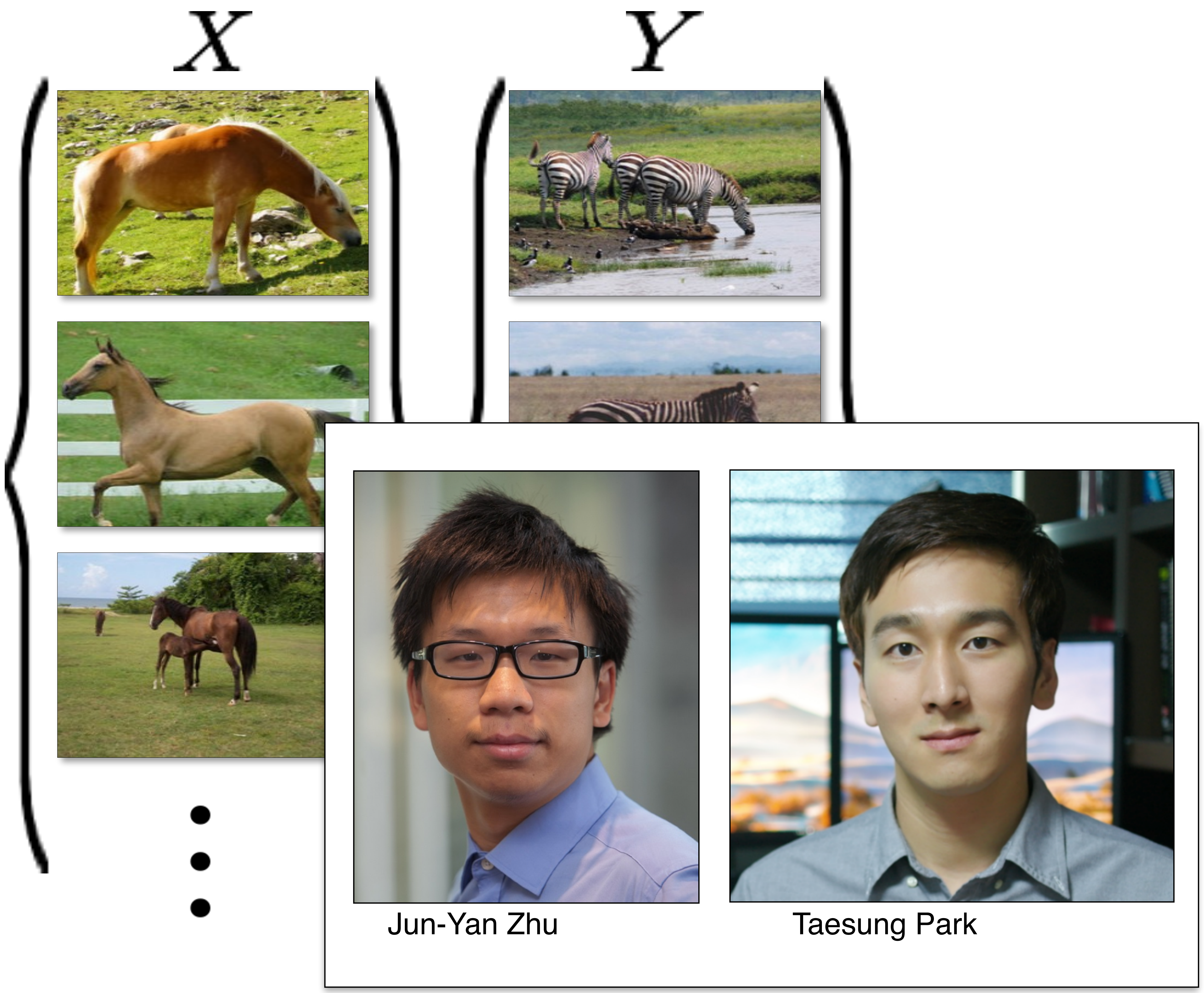
Paired data

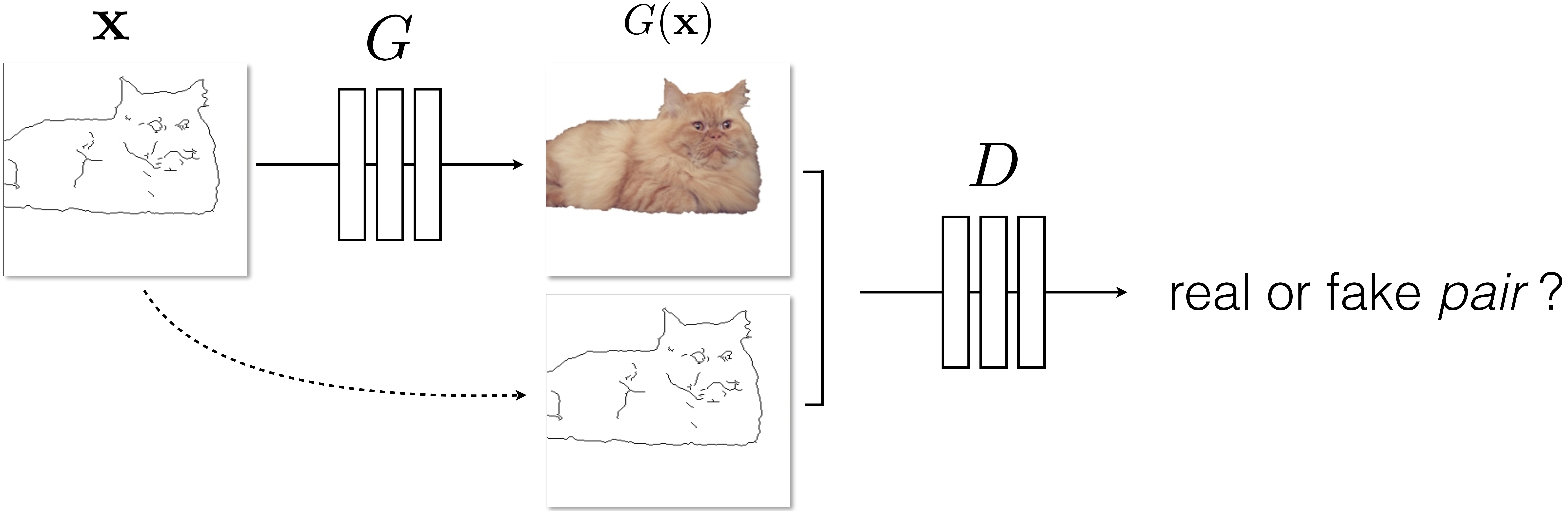


Paired data

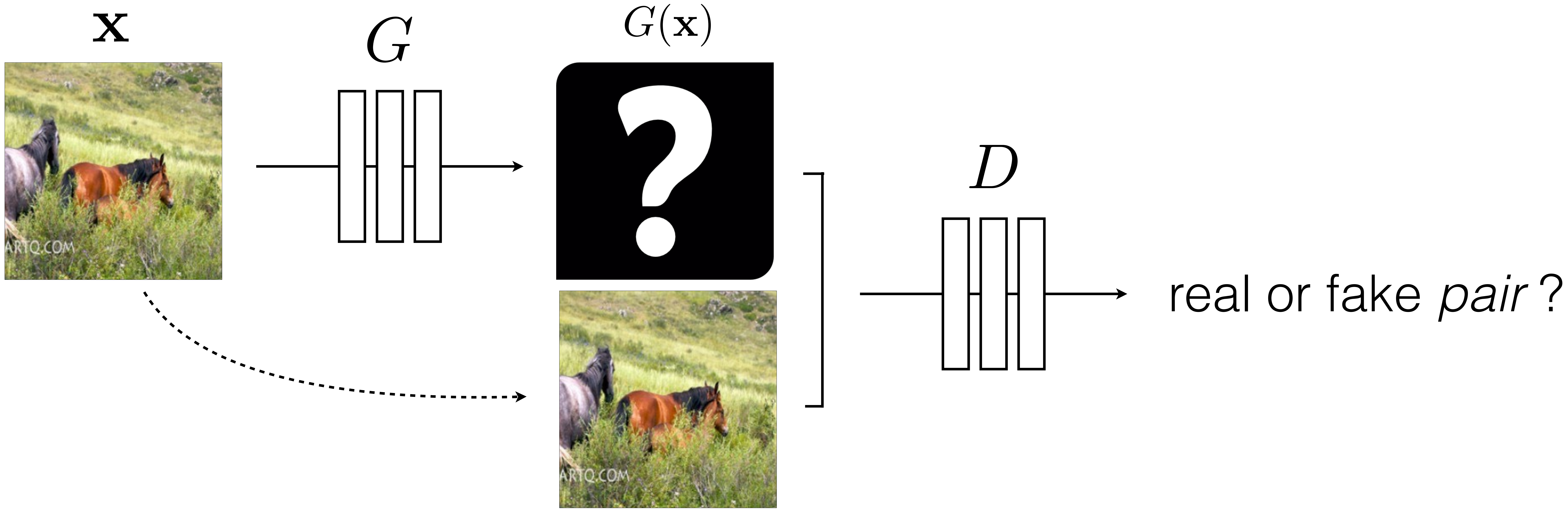


Unpaired data



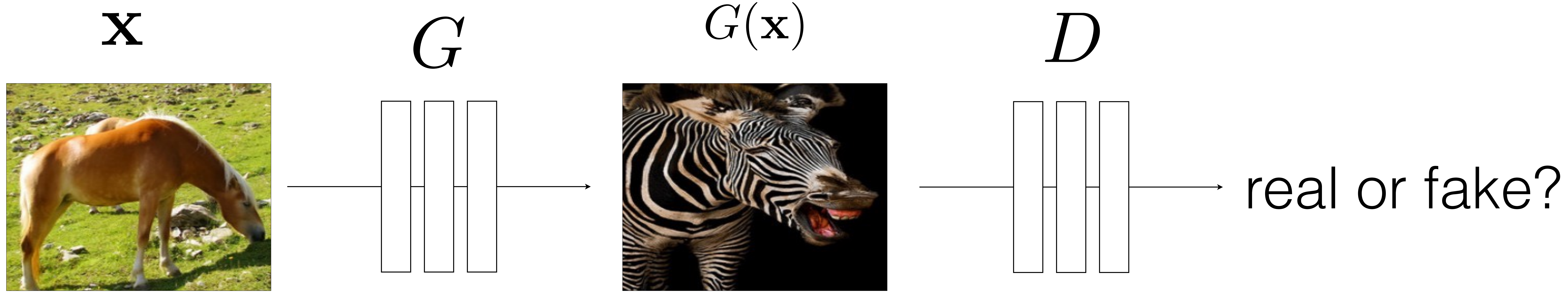


$$\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x},\mathbf{y}))\ ]$$



$$\arg\min_{G}\max_{D} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y})) \;]$$

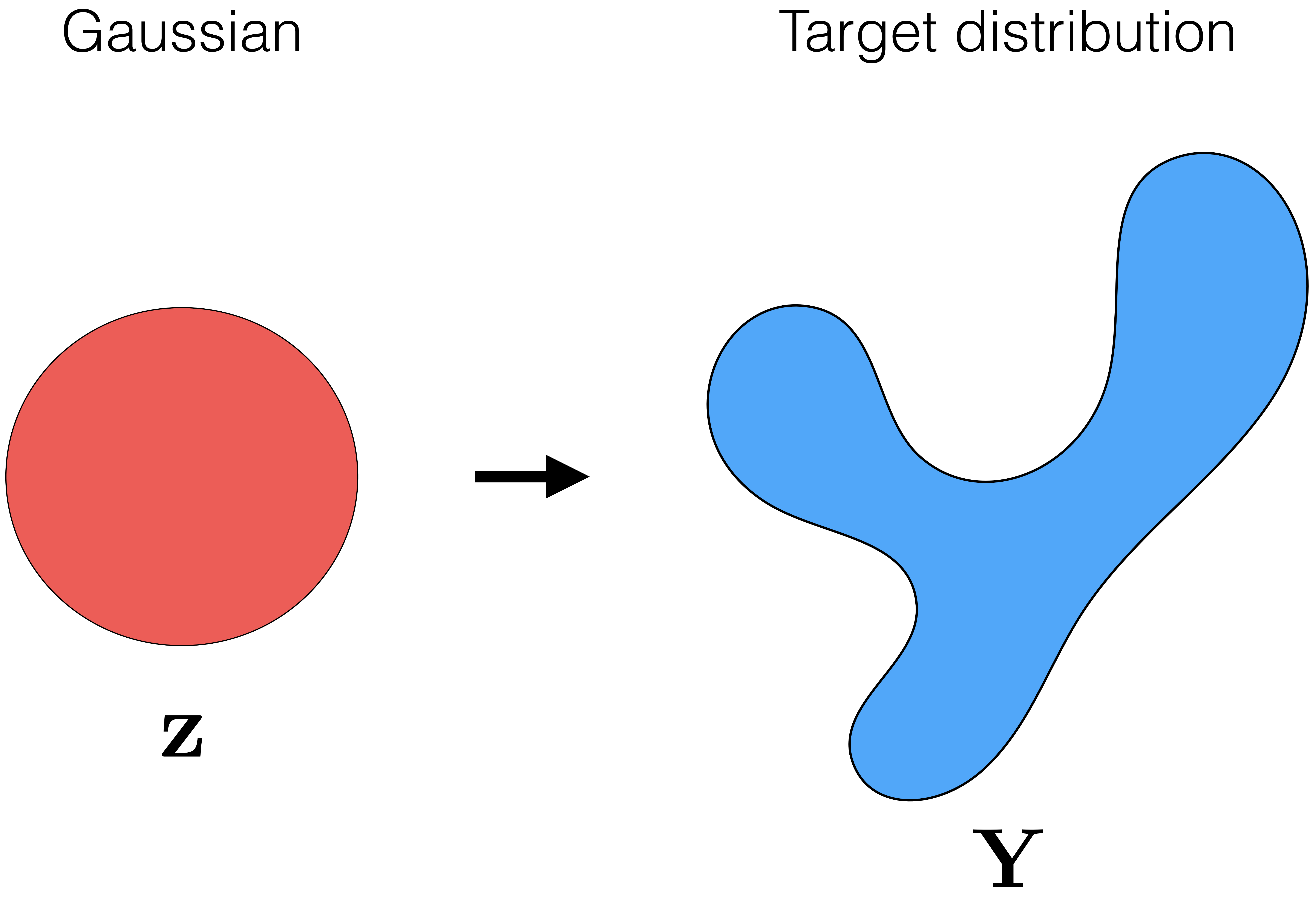
No input-output pairs!



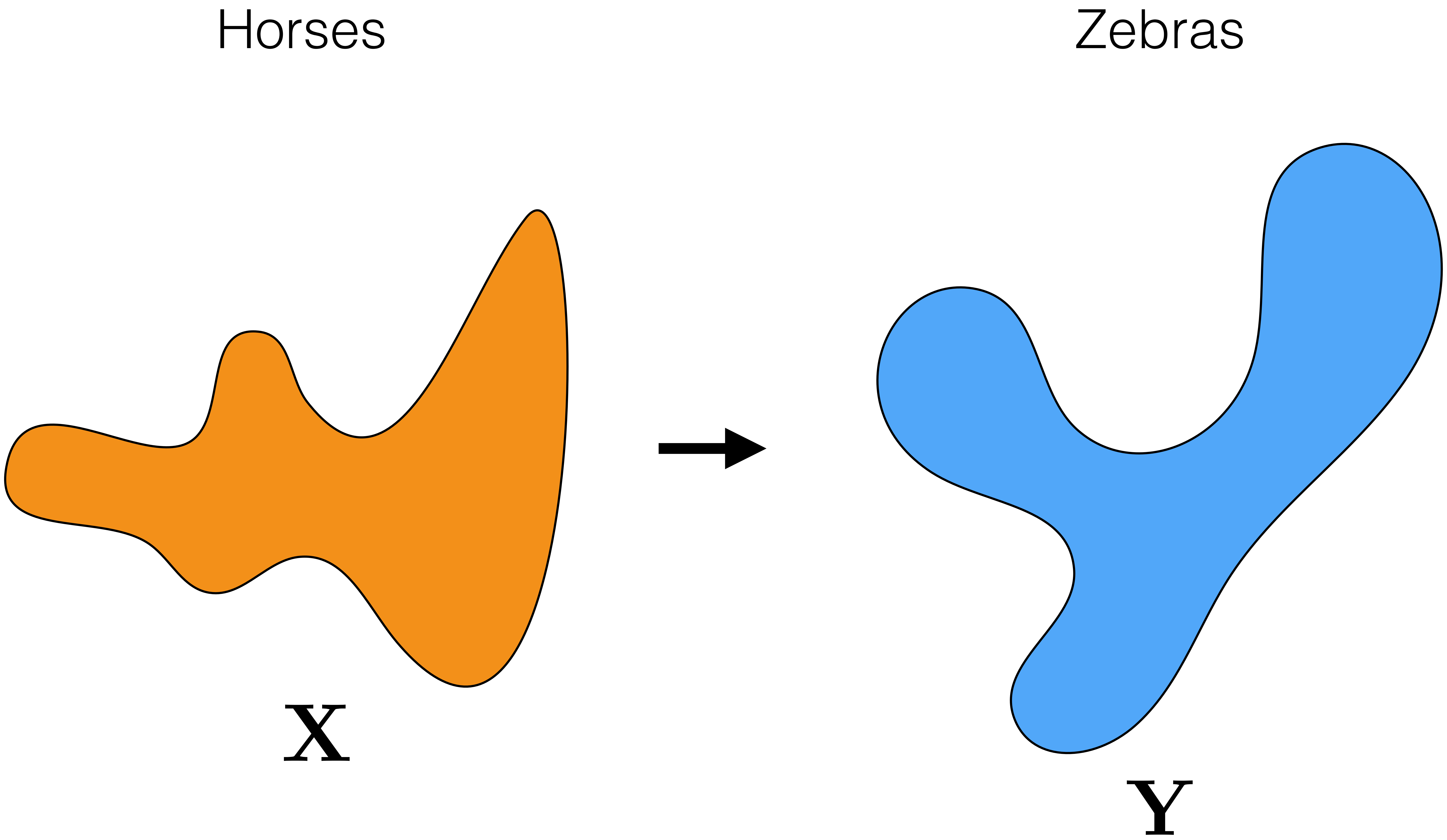
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x}))\ \ +\ \ \log(1-D(\mathbf{y}))\ ]$$

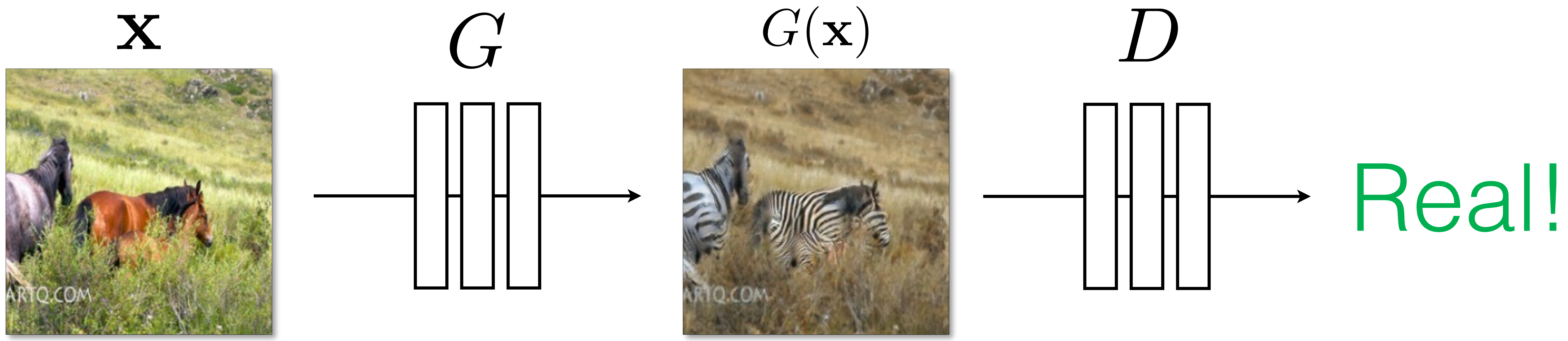
Usually loss functions check if output matches a target *instance*

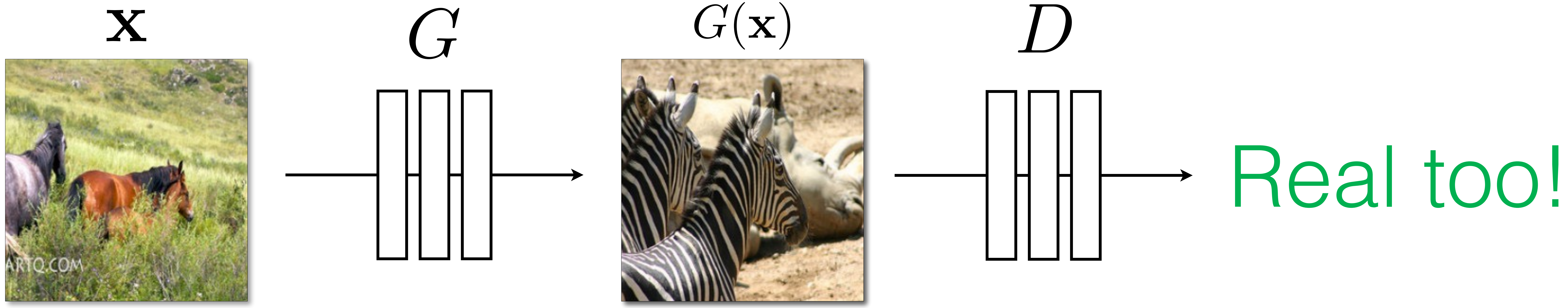
GAN loss checks if output is part of an admissible *set*



Gaussian
Target distribution
z
Y

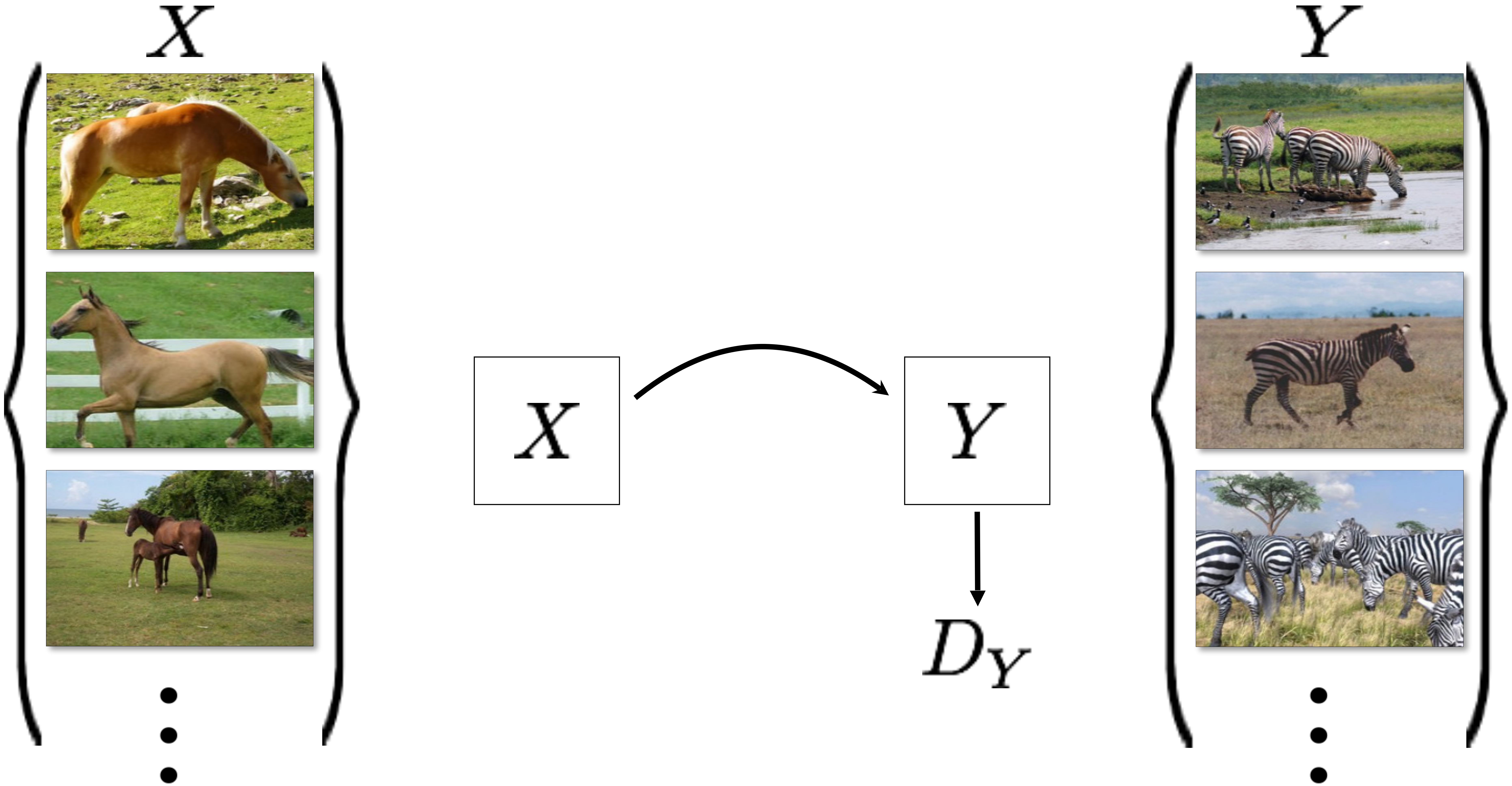

Horses
Zebras
X
Y


$\mathbf{x}$
G
$G(\mathbf{x})$
D
Real!



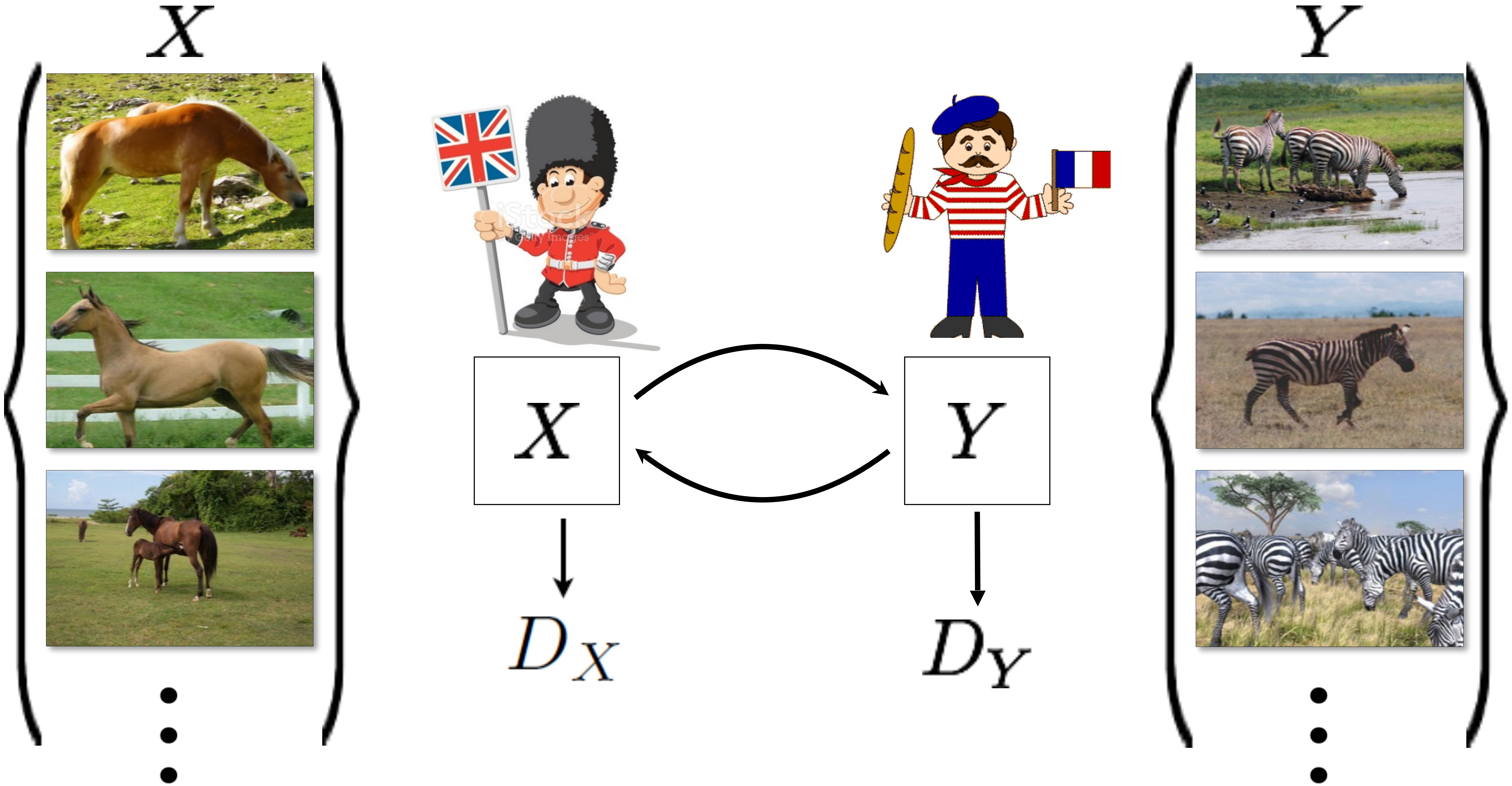
Nothing to force output to correspond to input

Cycle-Consistent Adversarial Networks

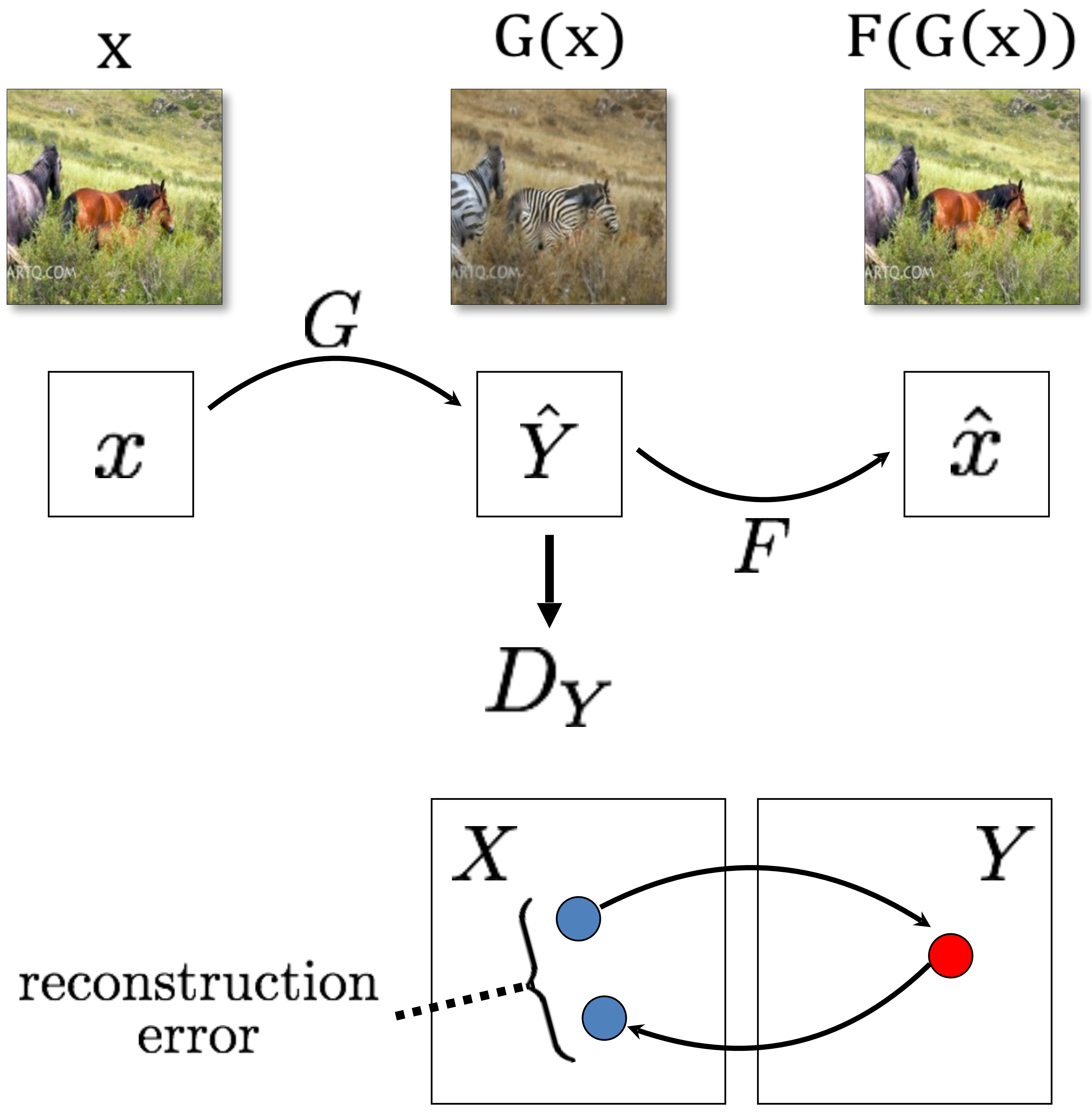


[Zhu et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

Cycle-Consistent Adversarial Networks

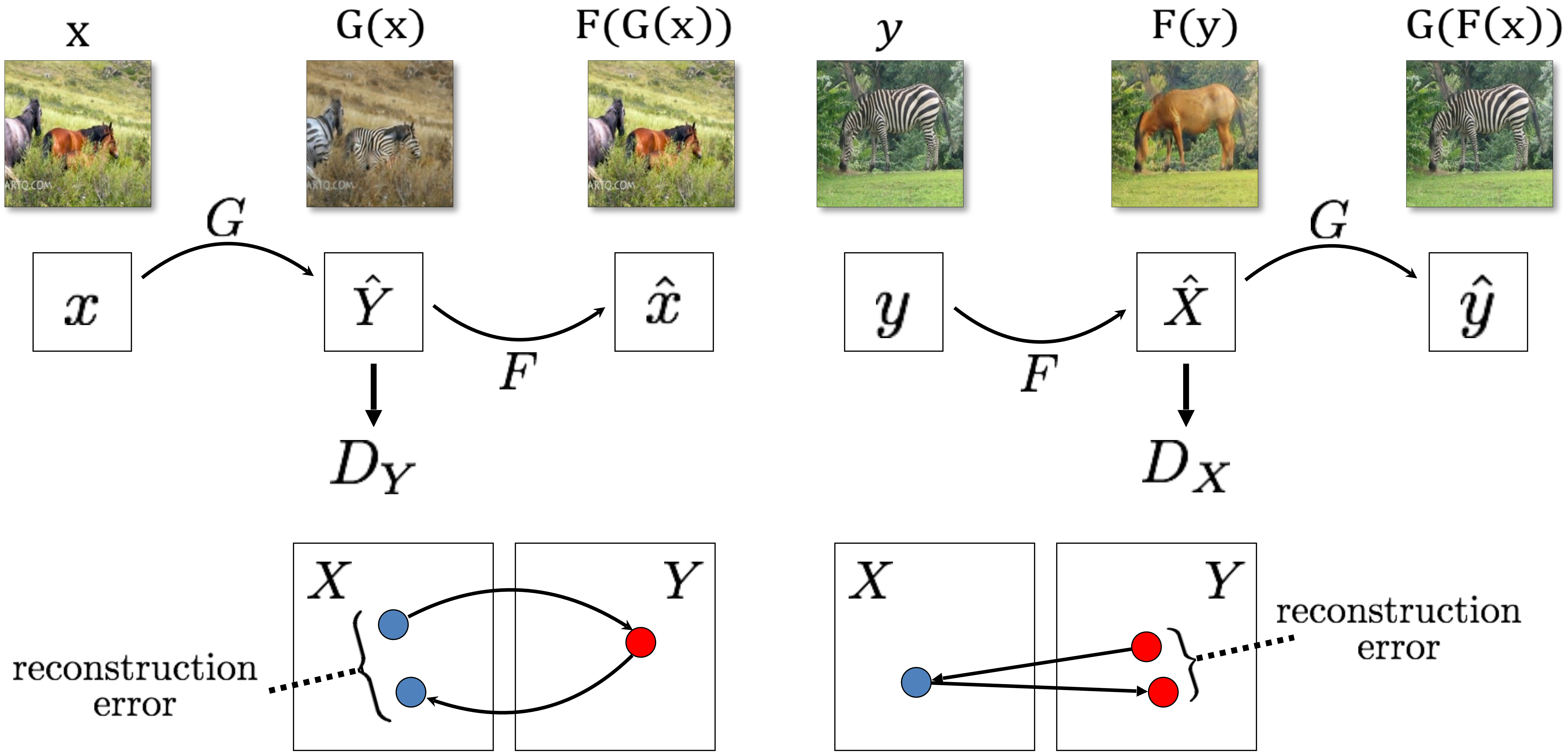


Cycle Consistency Loss

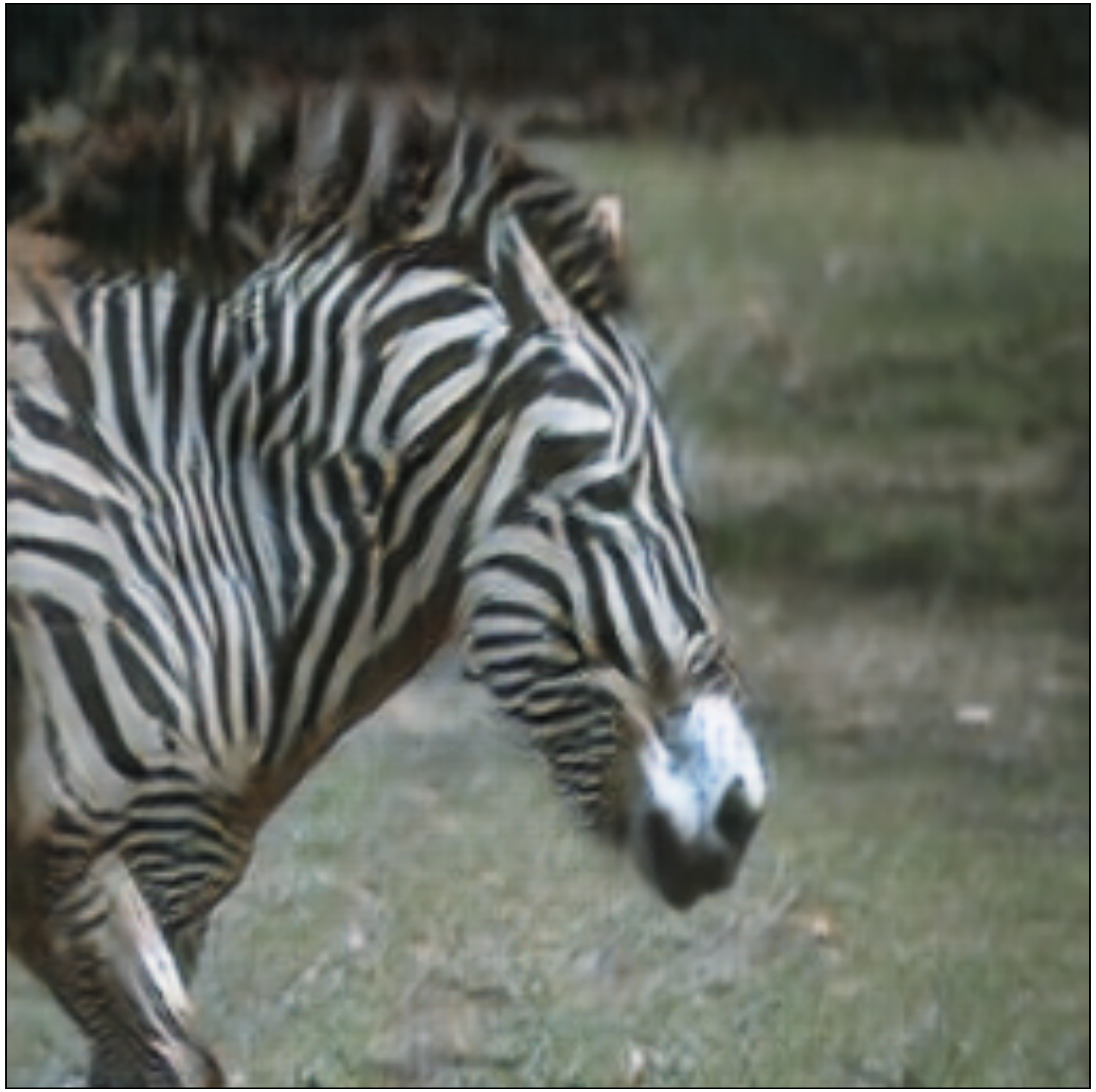


$$\left\| F(G(x)) - x \right\|_1$$

Cycle Consistency Loss



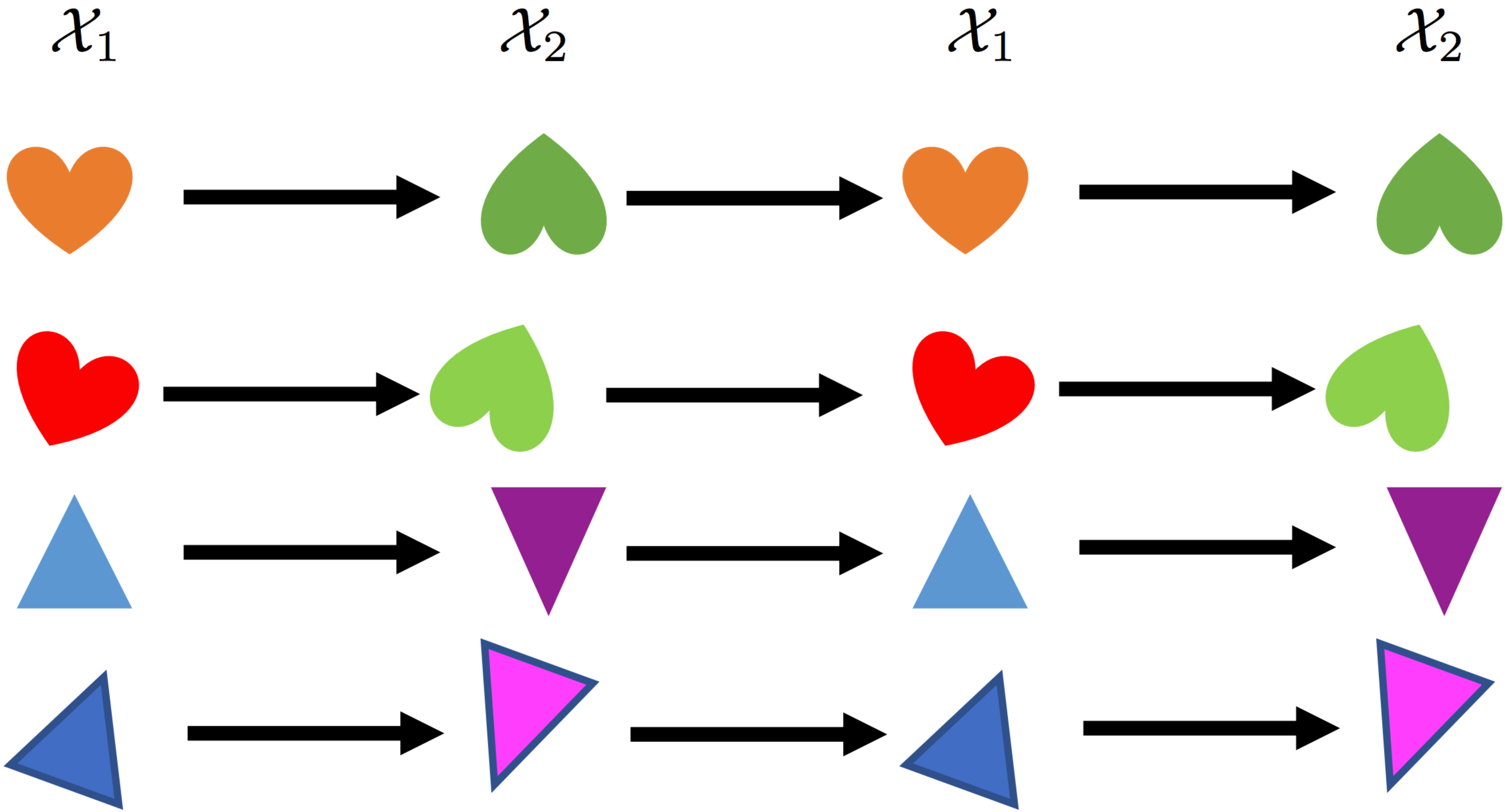
$$\|F(G(x)) - x\|_1 \qquad \|G(F(y)) - y\|_1$$



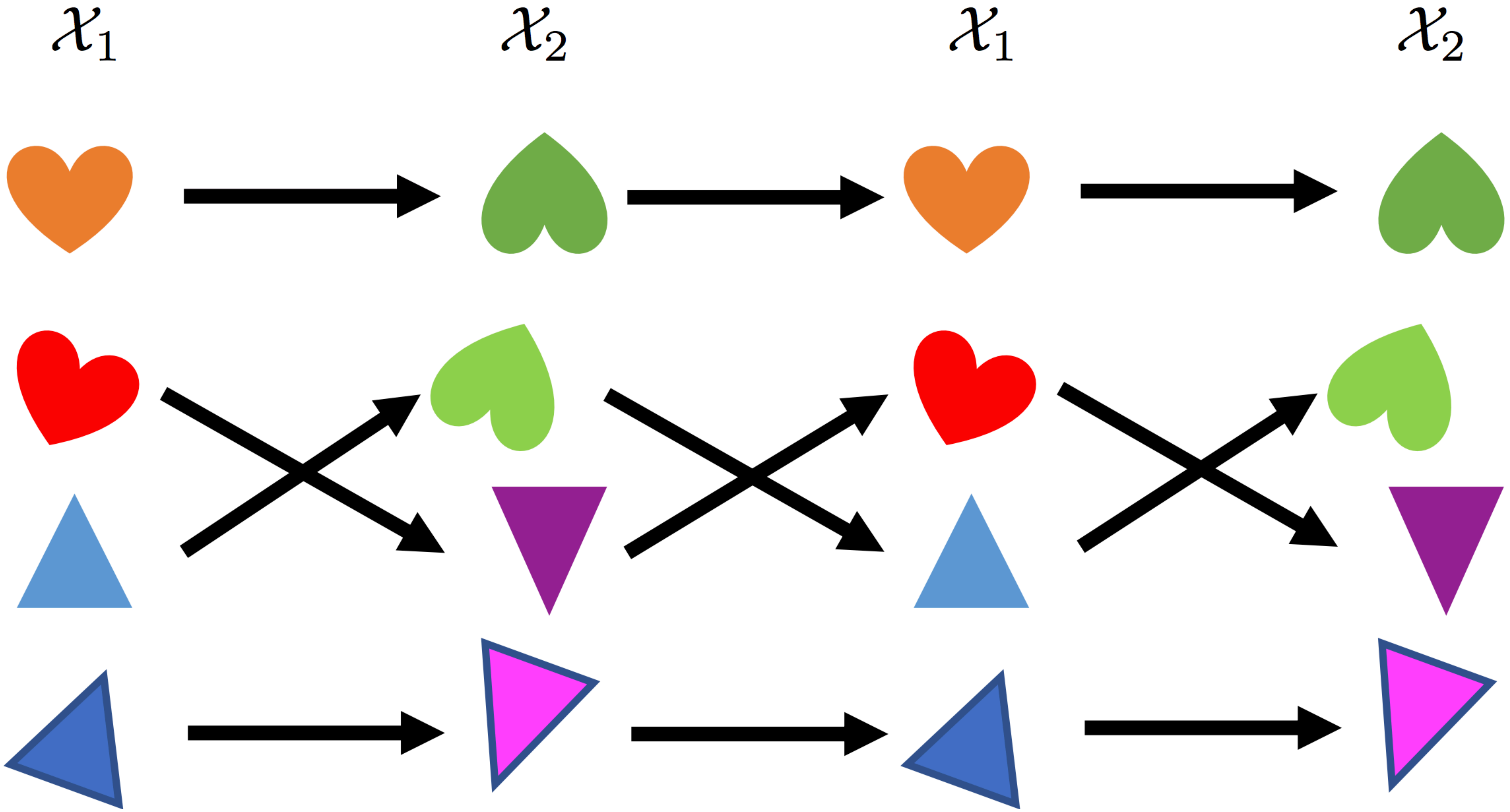

Failure case

Failure case

Why does CycleGAN work?



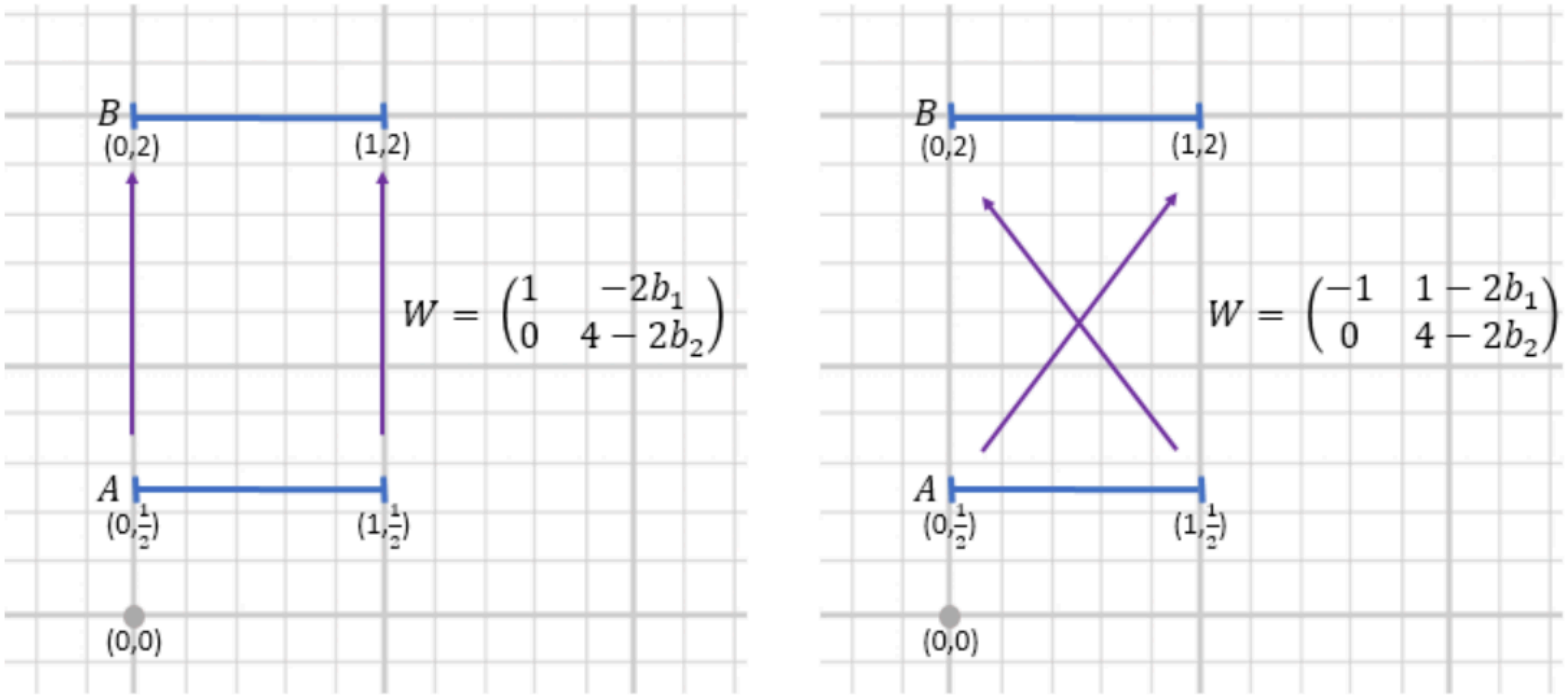
Slide credit: Ming-Yu Liu



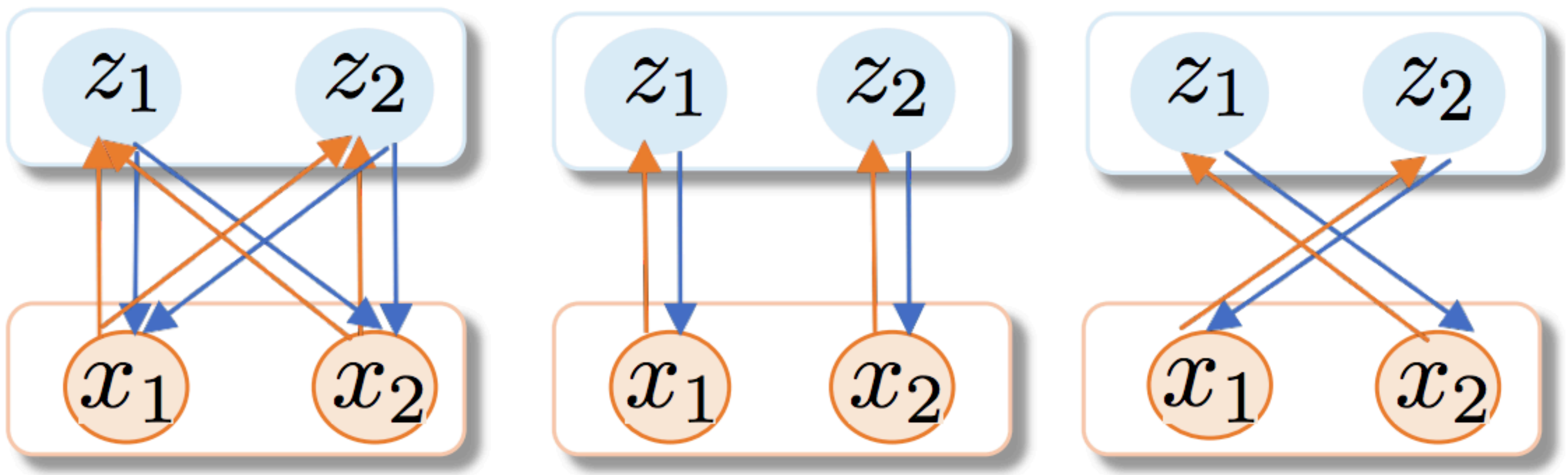
Slide credit: Ming-Yu Liu

Simplicity hypothesis

[Galanti, Wolf, Benaim, 2018]



Cycle Loss **upper bounds** Conditional Entropy



	z_1	z_2
x_1	$\delta/2$	$(1-\delta)/2$
x_2	$(1-\delta)/2$	$\delta/2$

	z_1	z_2
x_1	$1/2$	0
x_2	0	$1/2$

	z_1	z_2
x_1	0	$1/2$
x_2	$1/2$	0

High Conditional Entropy

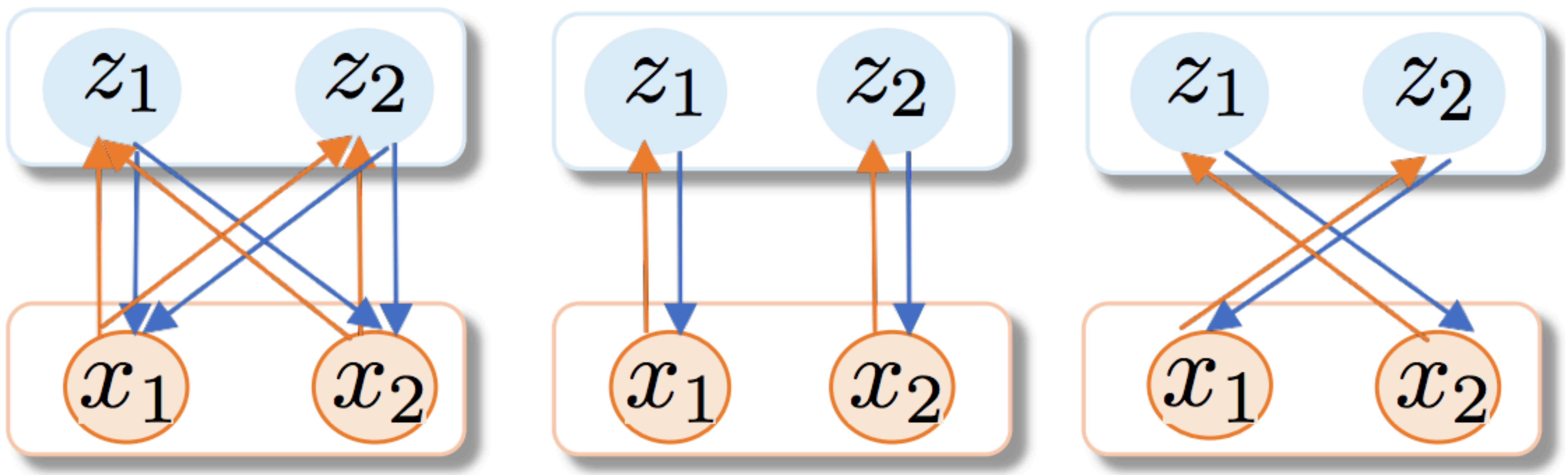
Low Conditional Entropy

Conditional Entropy

$$H^{\pi}(\boldsymbol{x}|\boldsymbol{z}) \triangleq -\mathbb{E}_{\pi(\boldsymbol{x},\boldsymbol{z})}[\log \pi(\boldsymbol{x}|\boldsymbol{z})]$$

"ALICE: Towards Understanding Adversarial Learning for Joint Distribution Matching" [Li et al. NIPS 2017]. Also see [Tiao et al. 2018] "CycleGAN as Approximate Bayesian Inference"

Cycle Loss **upper bounds** Conditional Entropy



	z_1	z_2
x_1	$\delta/2$	$(1-\delta)/2$
x_2	$(1-\delta)/2$	$\delta/2$

	z_1	z_2
x_1	1/2	0
x_2	0	1/2

	z_1	z_2
x_1	0	1/2
x_2	1/2	0

Conditional Entropy

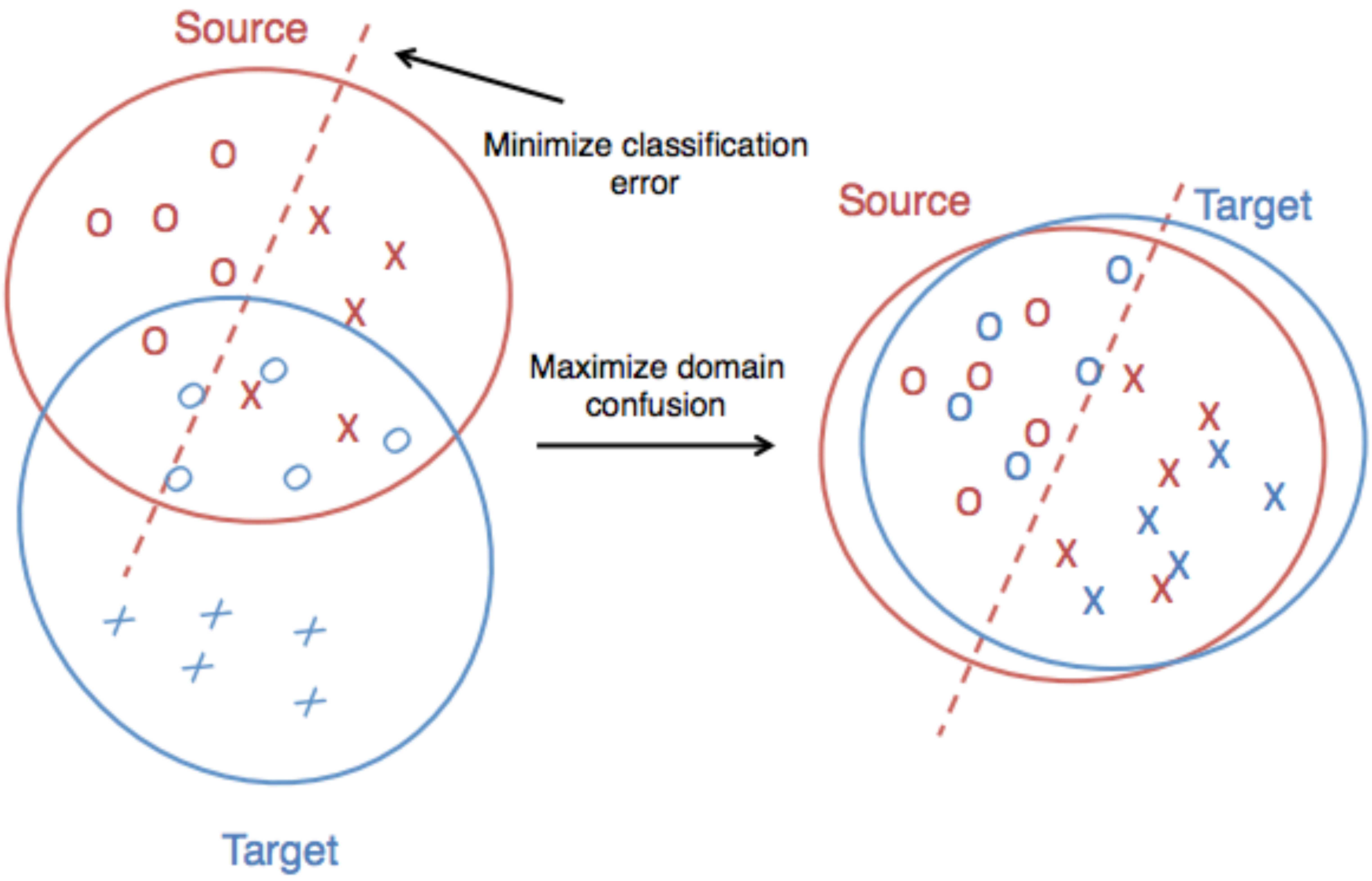
$$H^{\pi}(\boldsymbol{x}|\boldsymbol{z}) \triangleq -\mathbb{E}_{\pi(\boldsymbol{x},\boldsymbol{z})}[\log \pi(\boldsymbol{x}|\boldsymbol{z})]$$

Lemma 3 *For joint distributions $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ or $q_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})$, we have*

$$\boxed{H^{q_{\boldsymbol{\phi}}}(\boldsymbol{x}|\boldsymbol{z})} \triangleq -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}[\log q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})] = -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] - \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}) \| p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}))]$$
$$\boxed{\leq} -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] \triangleq \boxed{\mathcal{L}_{\text{Cycle}}(\boldsymbol{\theta}, \boldsymbol{\phi}).} \qquad (6)$$

“ALICE: Towards Understanding Adversarial Learning for Joint Distribution Matching” [Li et al. NIPS 2017]. Also see [Tiao et al. 2018] “CycleGAN as Approximate Bayesian Inference”

Domain Adaptation



[Tzeng et al. 2014]

Sim2real

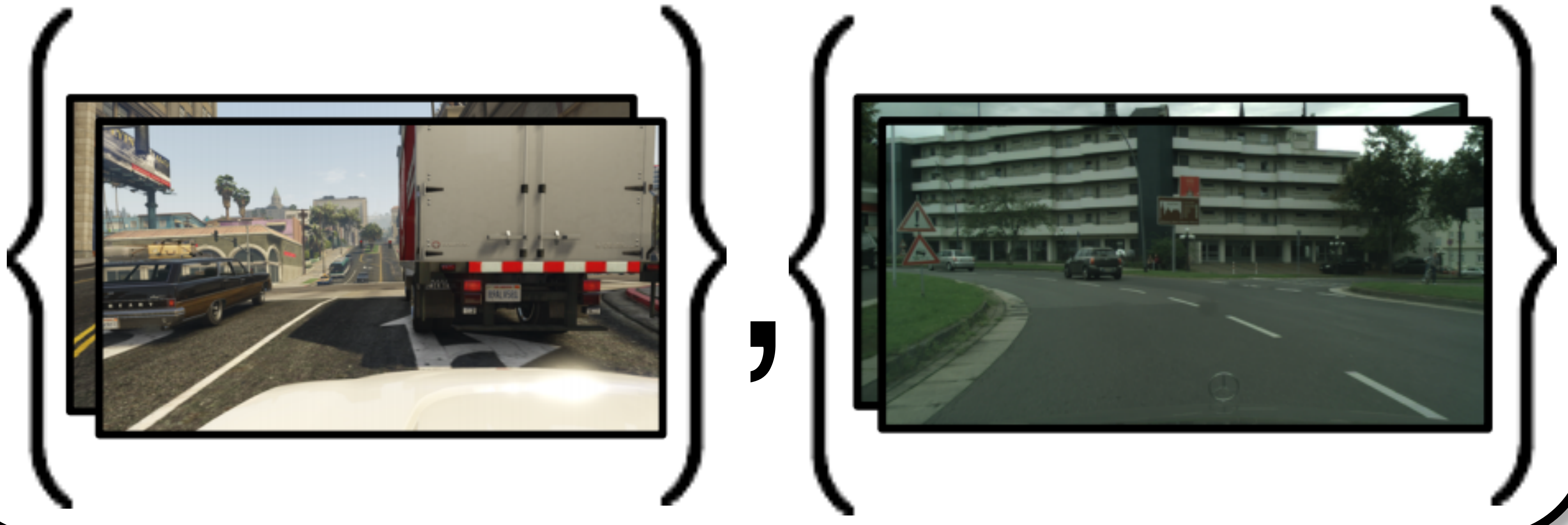
Simulated data

Real data

[Richter*, Vineet* et al. 2016] [Krähenbühl et al. 2018]

CycleGAN

Training data

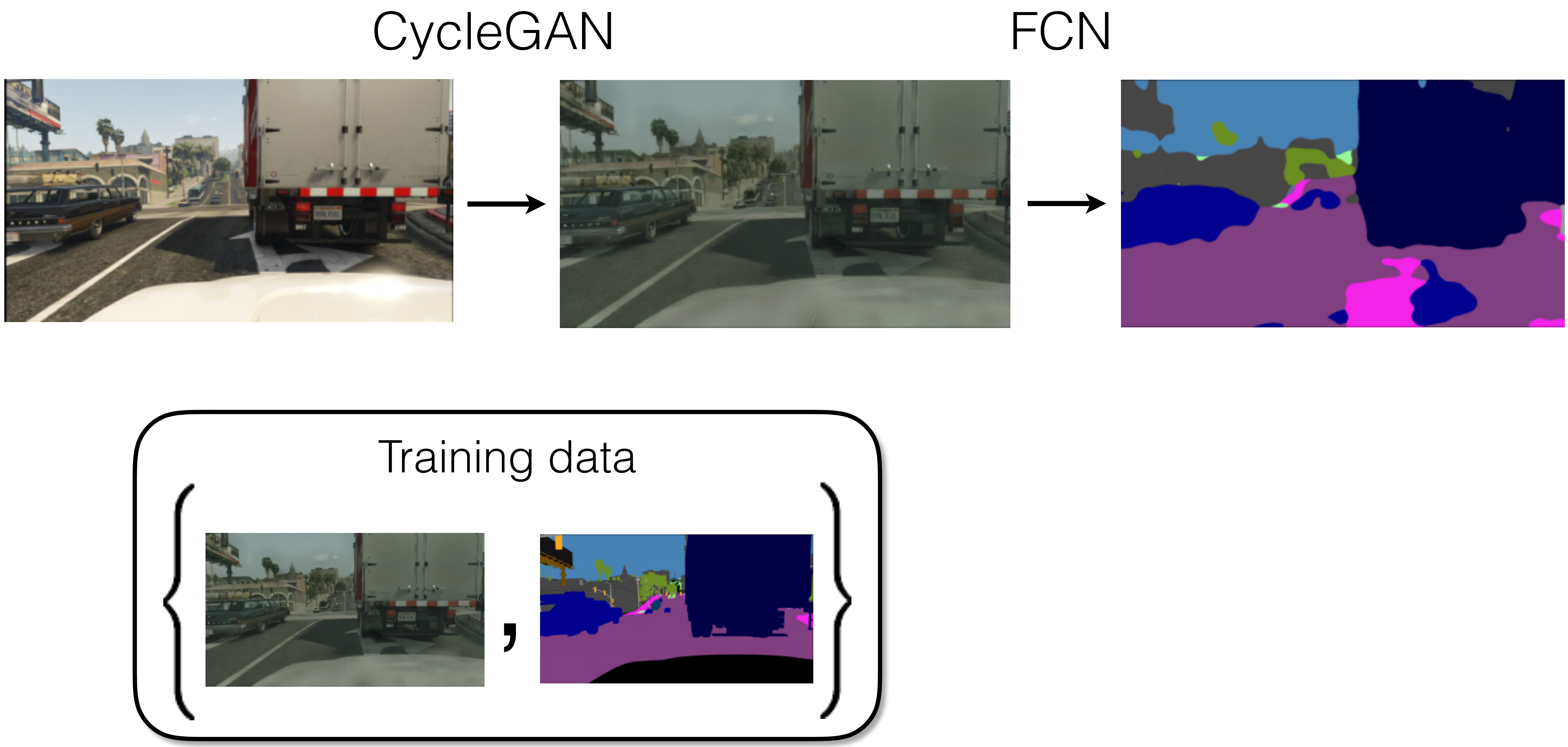


[Hoffman, Tzeng, Park, Zhu, Isola, Saenko, Darrell, Efros, 2018]

CycleGAN

Training data

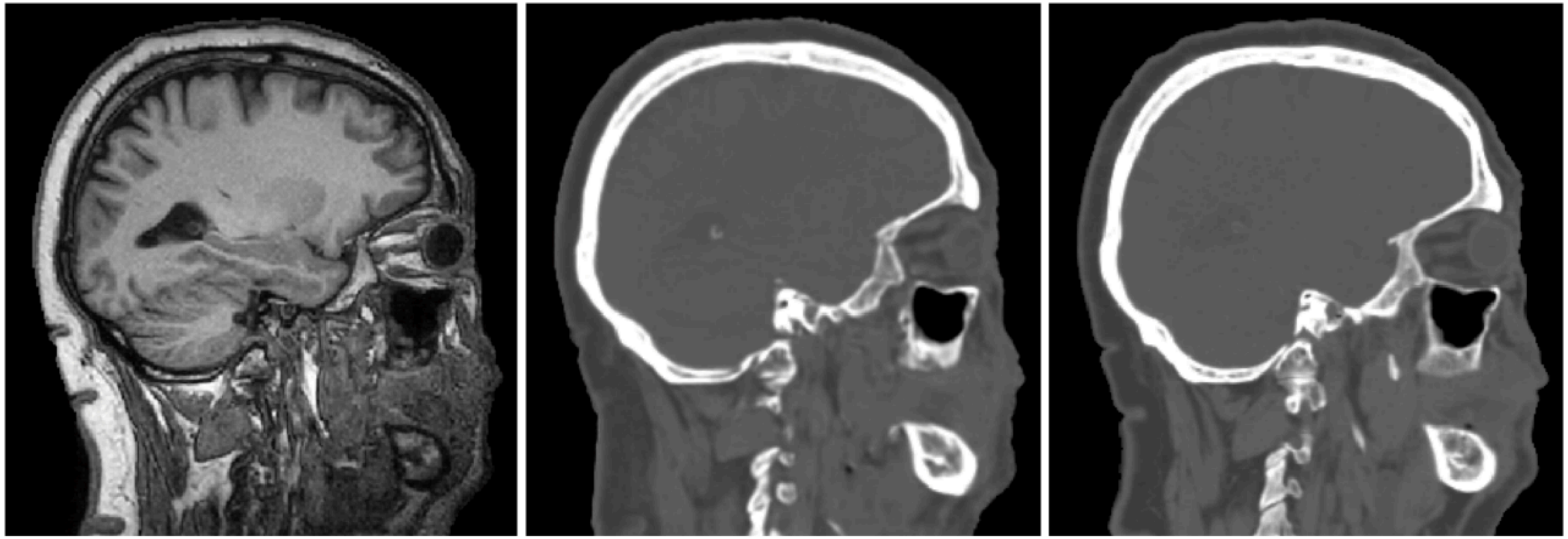
[Hoffman, Tzeng, Park, Zhu, Isola, Saenko, Darrell, Efros, 2018]



[Hoffman, Tzeng, Park, Zhu, Isola, Saenko, Darrell, Efros, 2018]

Medical domain adaptation

MR → CT [Wolterink et al] arxiv: 1708.01155



Input MR　　Generated CT　　Ground truth CT

- MRI reconstruction [Quan et al.] arxiv:1709.00753
- Cardiac MR images from CT [Chartsias et al. 2017]

Three perspectives on GANs

1. Structured loss

2. Generative model

3. Domain-level supervision / mapping

Thank you!