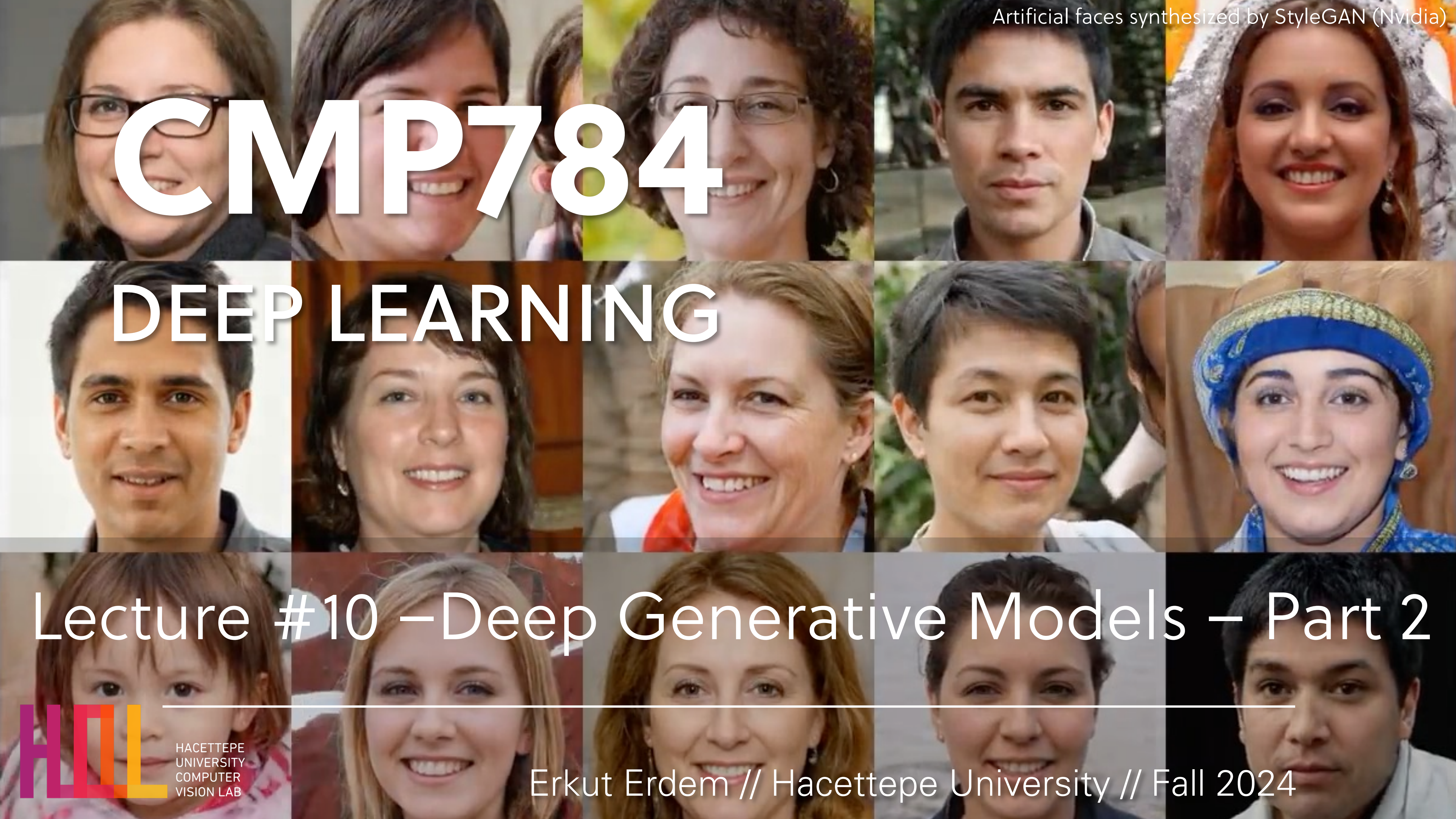


Artificial faces synthesized by StyleGAN (Nvidia)

CMP784

DEEP LEARNING

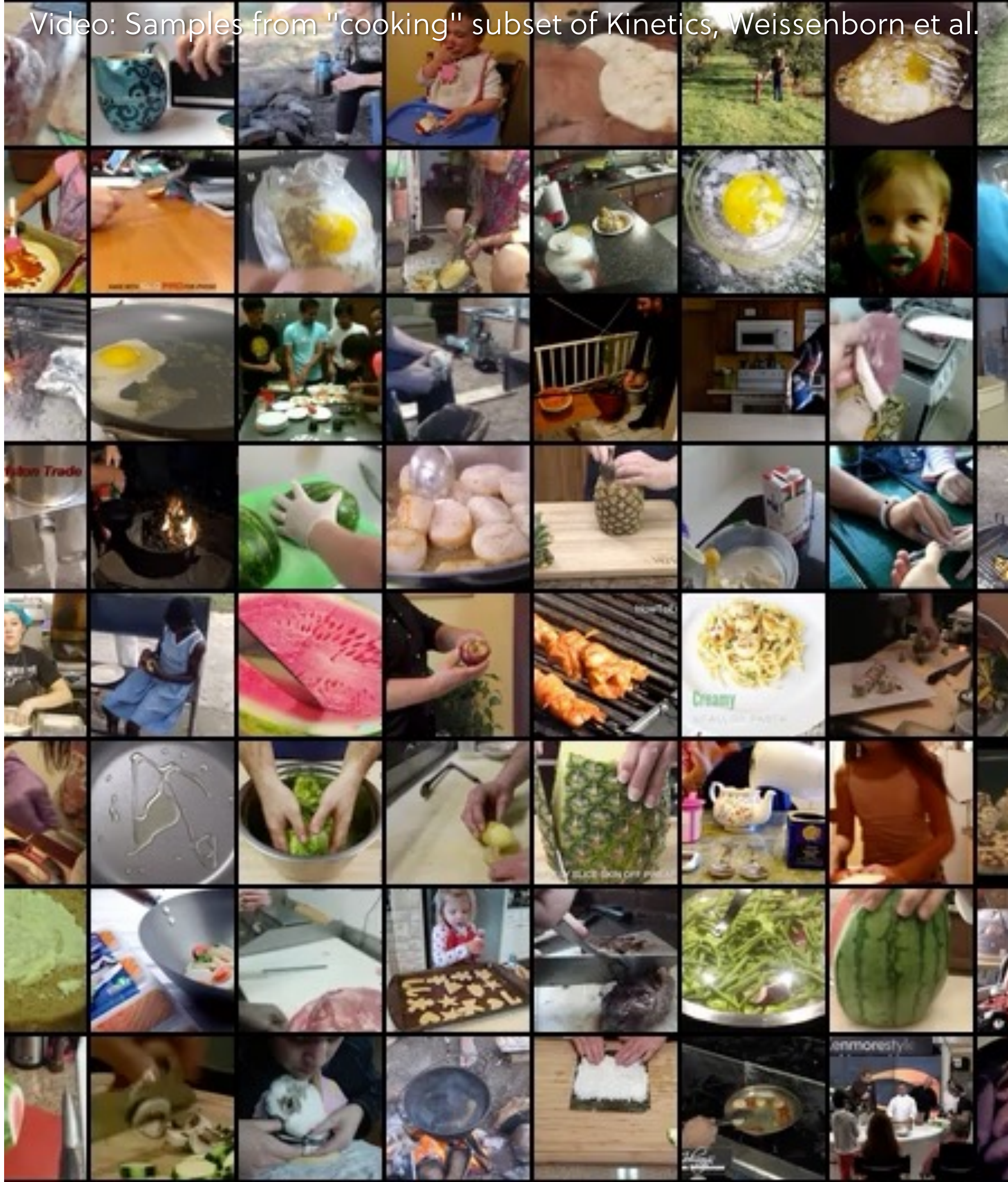
Lecture #10 –Deep Generative Models – Part 2

HACETTEPE UNIVERSITY COMPUTER VISION LAB

Erkut Erdem // Hacettepe University // Fall 2024

Previously on CMP784

- Supervised vs. Unsupervised Learning
- Generative Modeling
- Basic Foundations
  - Sparse Coding
  - Autoencoders
- Autoregressive Generative Models



Video: Samples from "cooking" subset of Kinetics, Weissenborn et al.

Lecture overview

- Generative Adversarial Networks (GANs)

Disclaimer: Some of the material and slides for this lecture were borrowed from

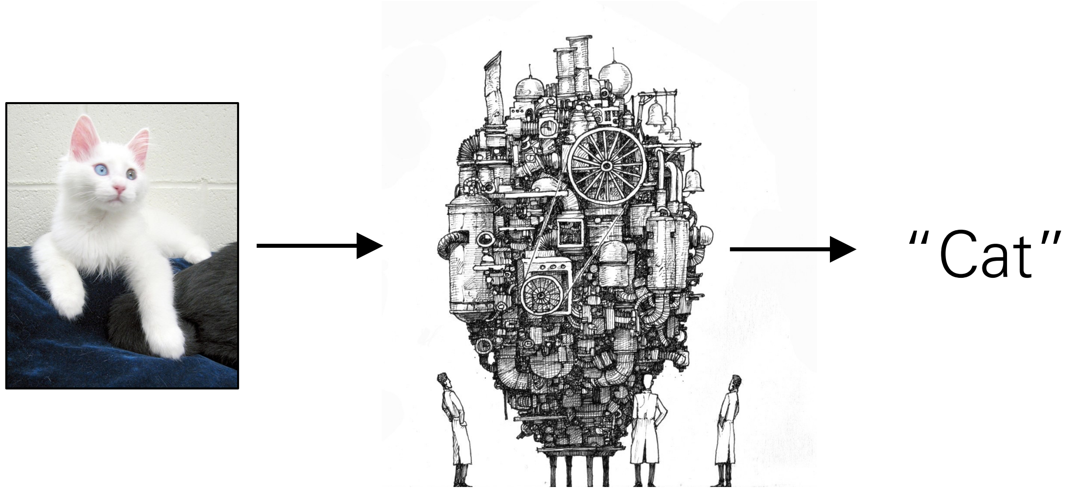
—Ian Goodfellow's tutorial on "Generative Adversarial Networks"

—Aaron Courville's IFT6135 class

—Bill Freeman, Antonio Torralba and Phillip Isola's MIT 6.869 class

Discriminative vs. Generative Models

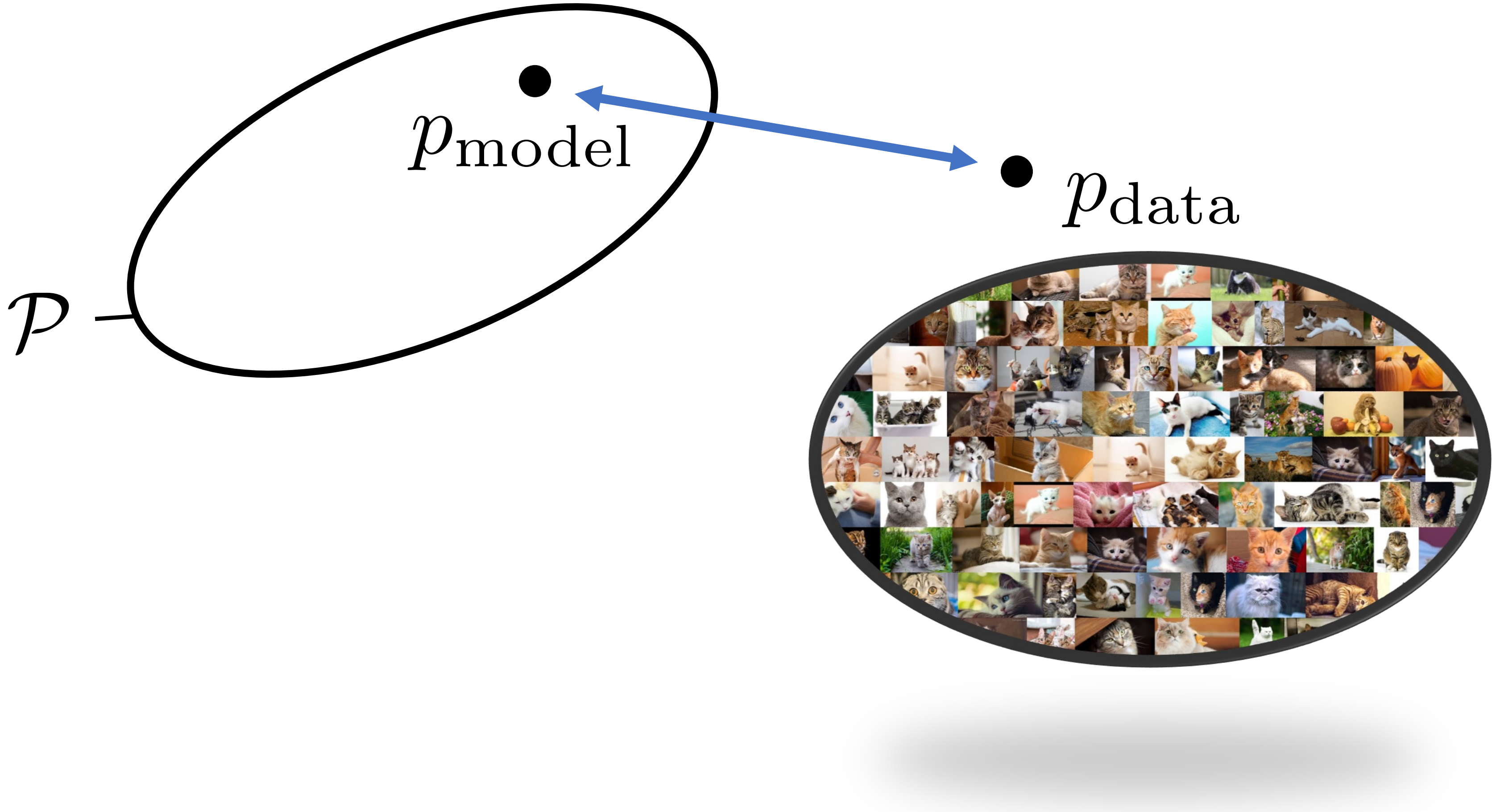
$$p(y|x) \qquad\qquad p(x|y)$$



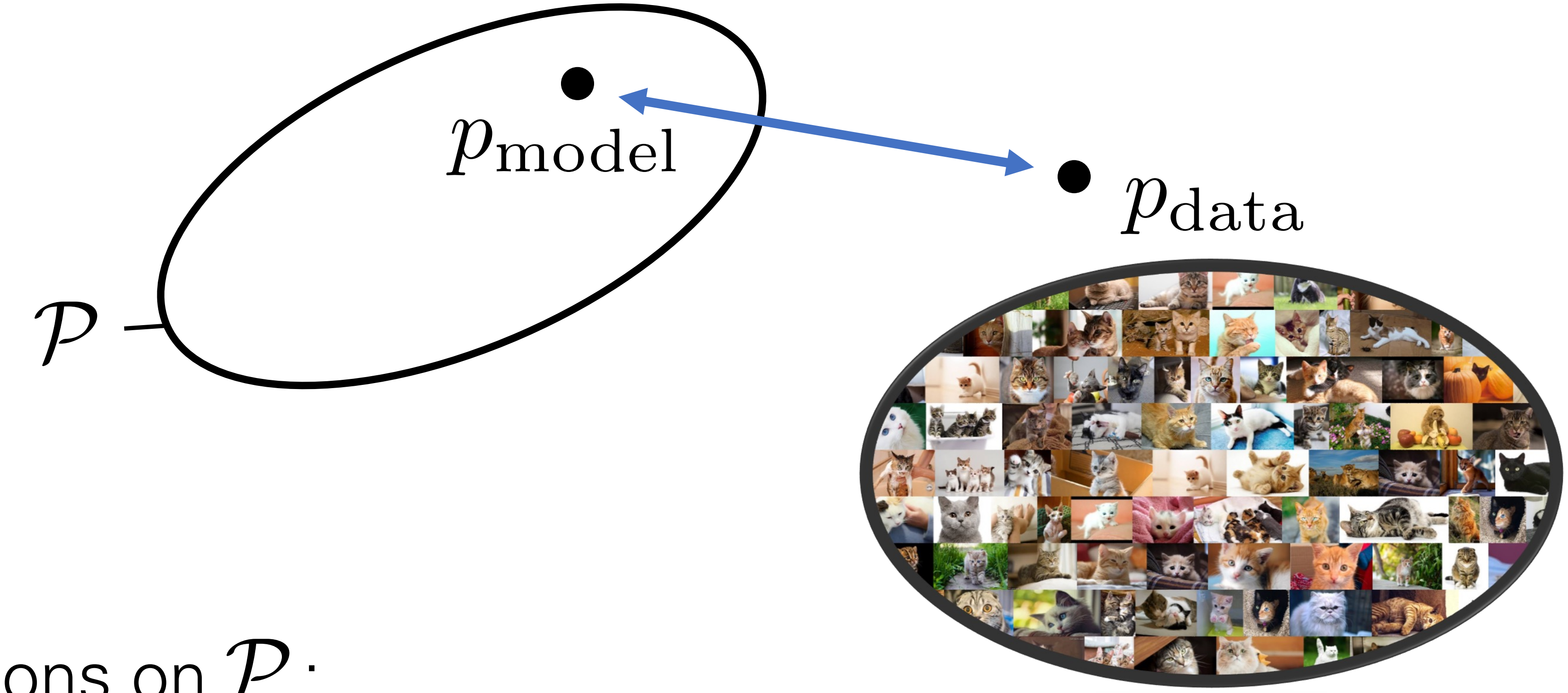
Discriminative models

Generative models

Generative Modeling



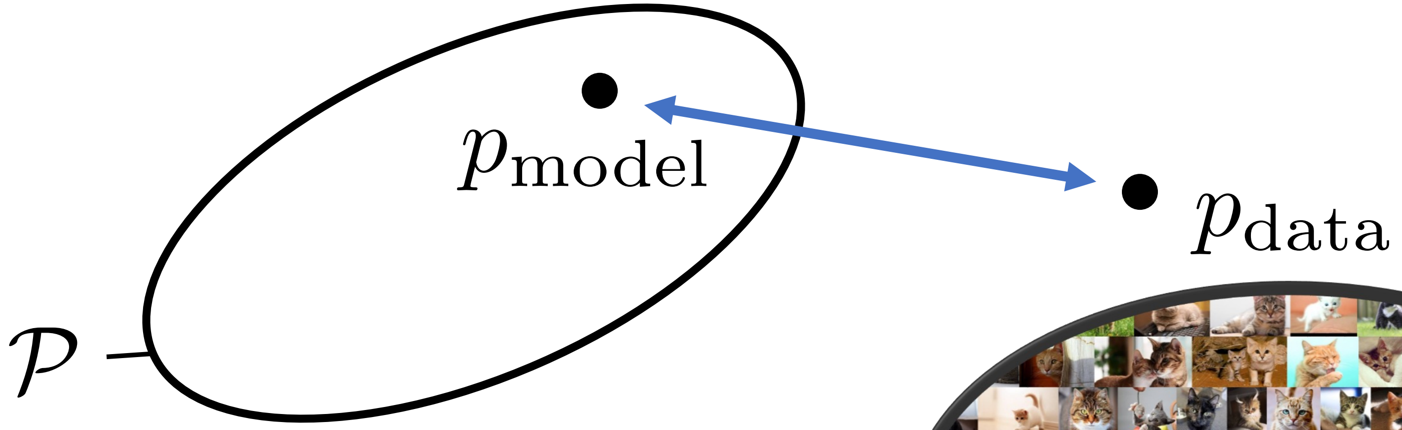
Generative Modeling



Assumptions on $\mathcal{P}$:
- tractable sampling

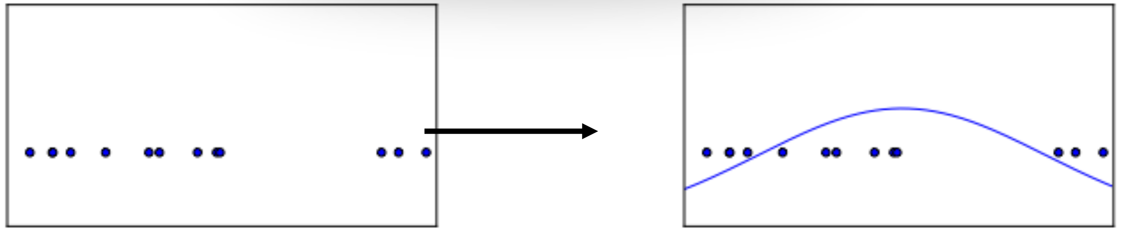
Training examples Model samples

Generative Modeling



Assumptions on $\mathcal{P}$:

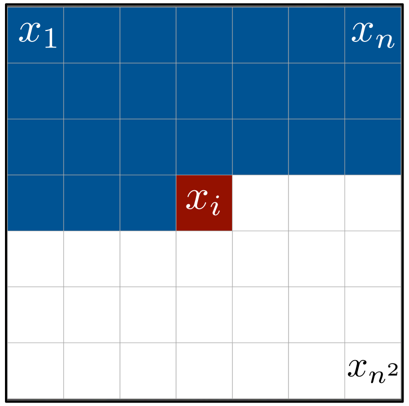
- tractable sampling
- tractable likelihood function



Broad Categories of Generative Models

- Autoregressive Models
- Generative Adversarial Networks (GANs)
- Flow-based Models
- Variational Autoencoders
- Energy-based Models

Autoregressive Models



- Explicitly model conditional probabilities:

$$p_{\text{model}}(\boldsymbol{x}) = p_{\text{model}}(x_1) \prod_{i=2}^{n} p_{\text{model}}(x_i \mid x_1, \ldots, x_{i-1})$$

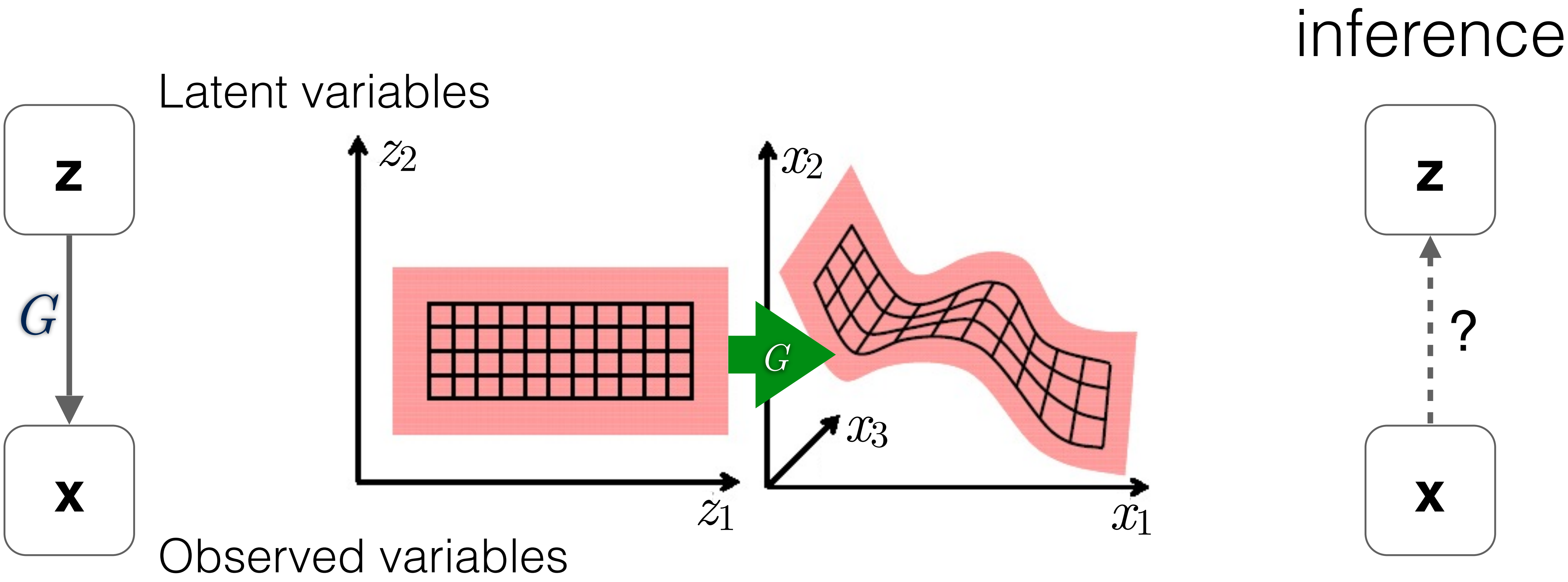
Each conditional can be a complicated neural net

Disadvantages:

- Generation can be too costly
- Generation can not be controlled by a latent code

PixelCNN elephants
(van den Ord et al. 2016)

Another way to train a latent variable model

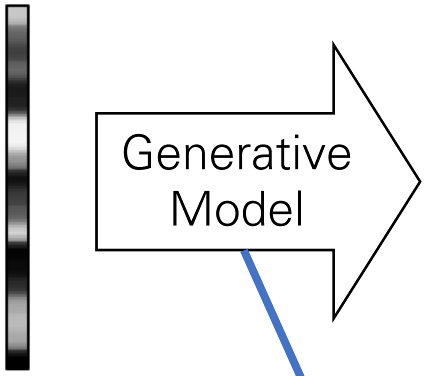


Generative Adversarial Networks

Generative Adversarial Networks (GANs)

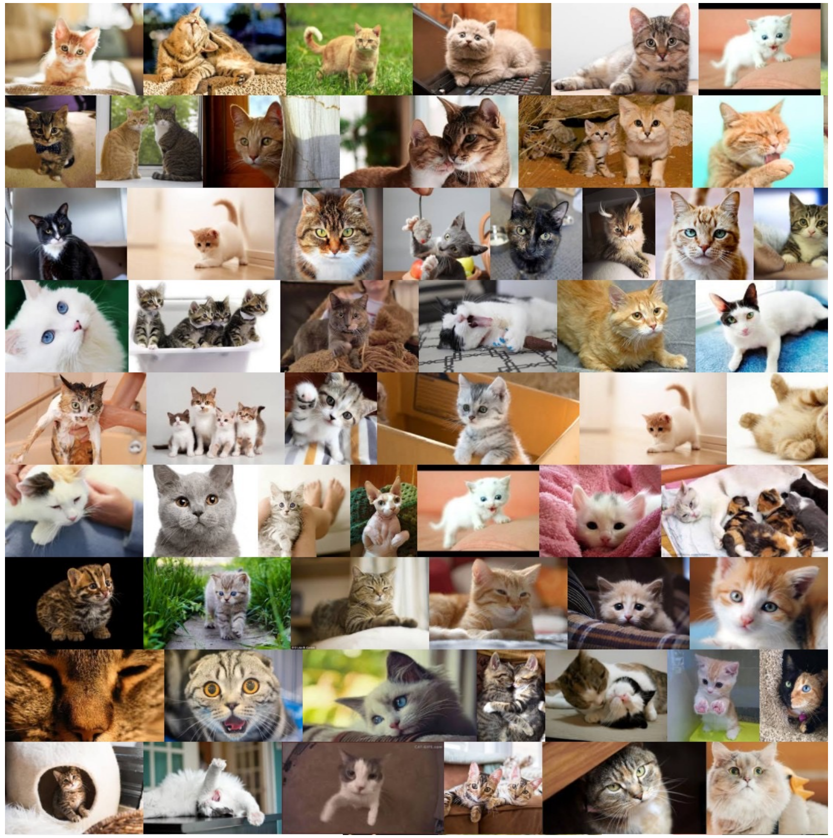
(Goodfellow et al., 2014)

Noise
(random input)



$z \sim \mathrm{Uniform}_{100}$

think of this as
a transformation



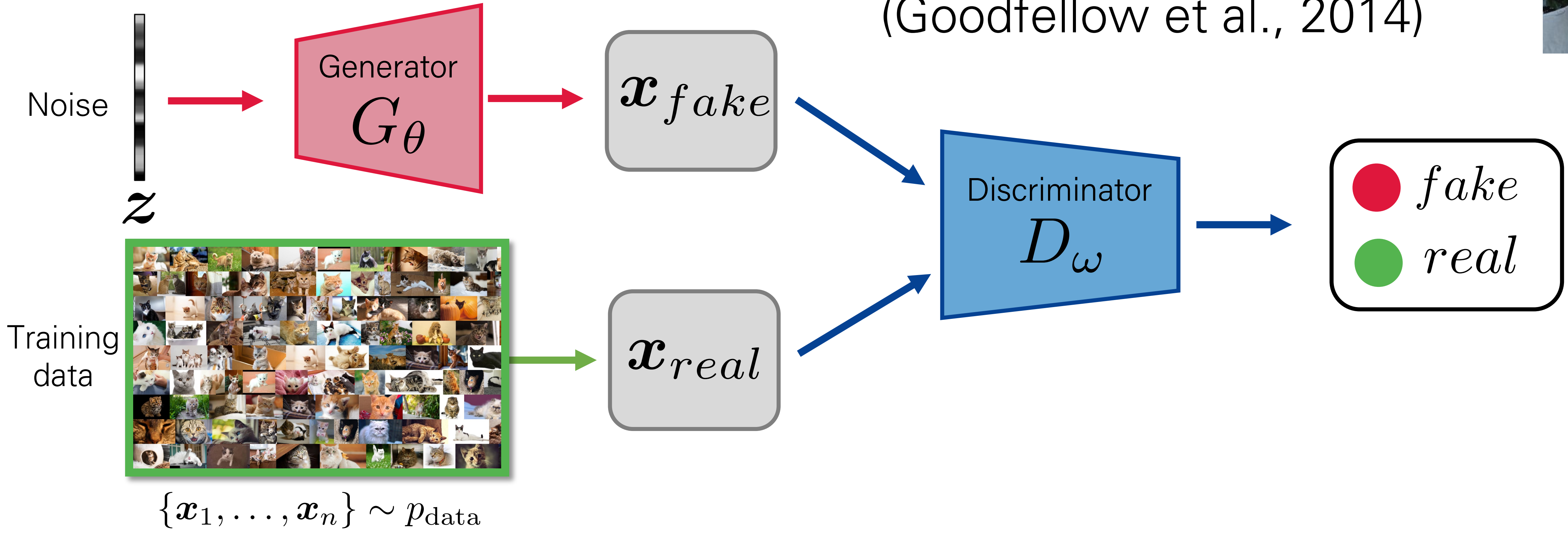
- A game-theoretic likelihood free model

Advantages:

- Uses a latent code
- No Markov chains needed
- Produces the best looking samples

Generative Adversarial Networks (GANs)

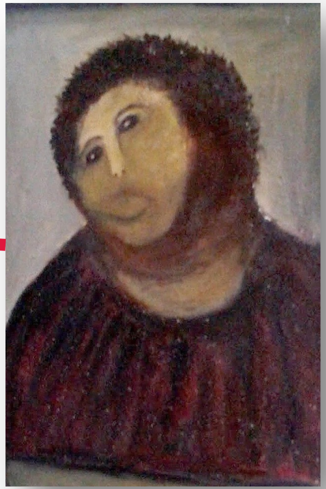
(Goodfellow et al., 2014)



- A game between a generator $G_\theta(\boldsymbol{z})$ and a discriminator $D_\omega(\boldsymbol{x})$
  - Generator tries to fool discriminator (i.e. generate realistic samples)
  - Discriminator tries to distinguish fake from real samples

Intuition behind GANs

D_ω: Discriminator (Art Critic)



$\boldsymbol{x}_{real}$

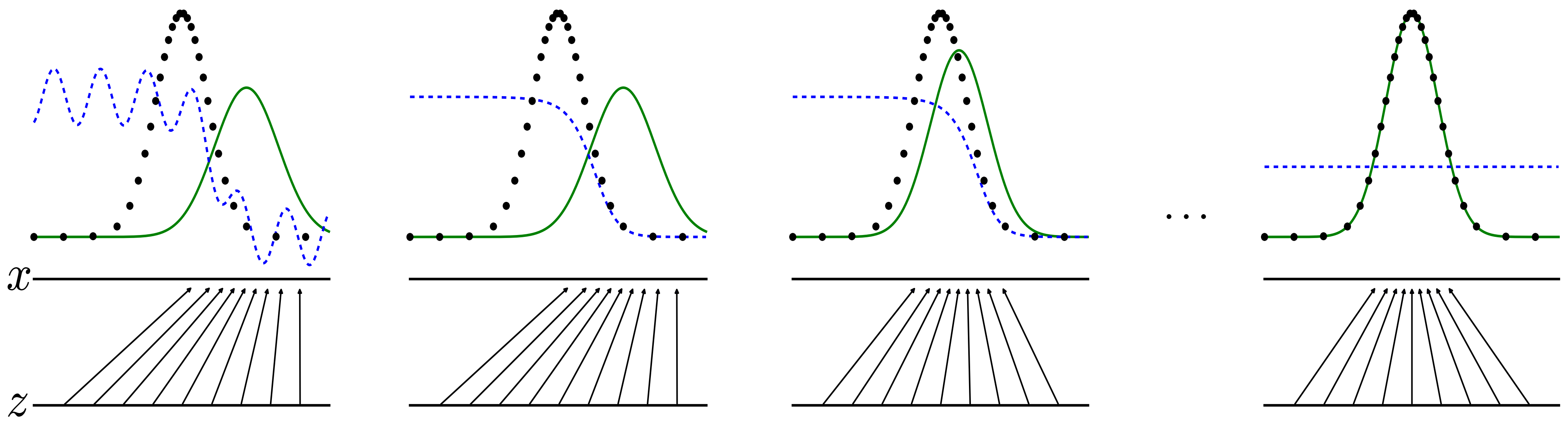
$\boldsymbol{x}_{fake}$

G_θ: Generator (Forger)

Training Procedure

(Goodfellow et al., 2014)

- Use SGD on two minibatches simultaneously:
  - A minibatch of training examples
  - A minibatch of generated samples



GAN Training: Minimax Game (Goodfellow et al., 2014)

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log D_\omega(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} \left[\log \left(1 - D_\omega(G_\theta(\boldsymbol{z}))\right)\right]$$

Real data

Noise vector used to generate data

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log \left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

Cross-entropy loss for binary classification

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log D\left(G(\boldsymbol{z})\right)$$

Generator maximizes the log-probability of the discriminator being mistaken

- Equilibrium of the game
- **Minimizes the Jensen-Shannon divergence between p_{data} and p_x**

GAN Training: Minimax Game (Goodfellow et al., 2014)

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log D_{\omega}(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} \left[\log \left(1 - D_{\omega}(G_{\theta}(\boldsymbol{z}))\right)\right]$$

Real data

Noise vector used

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim}$$

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}}\,\mathrm{l}$$

ross-entropy
ss for binary
assification

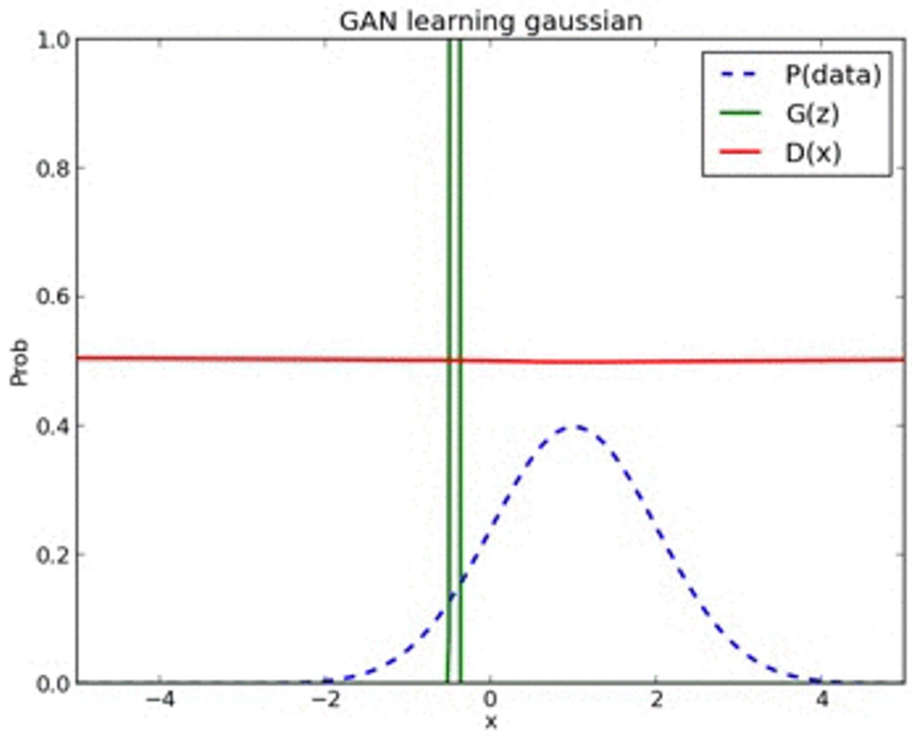
g-probability
of the discriminator being mistaken

Important question is
"Does this converge??"

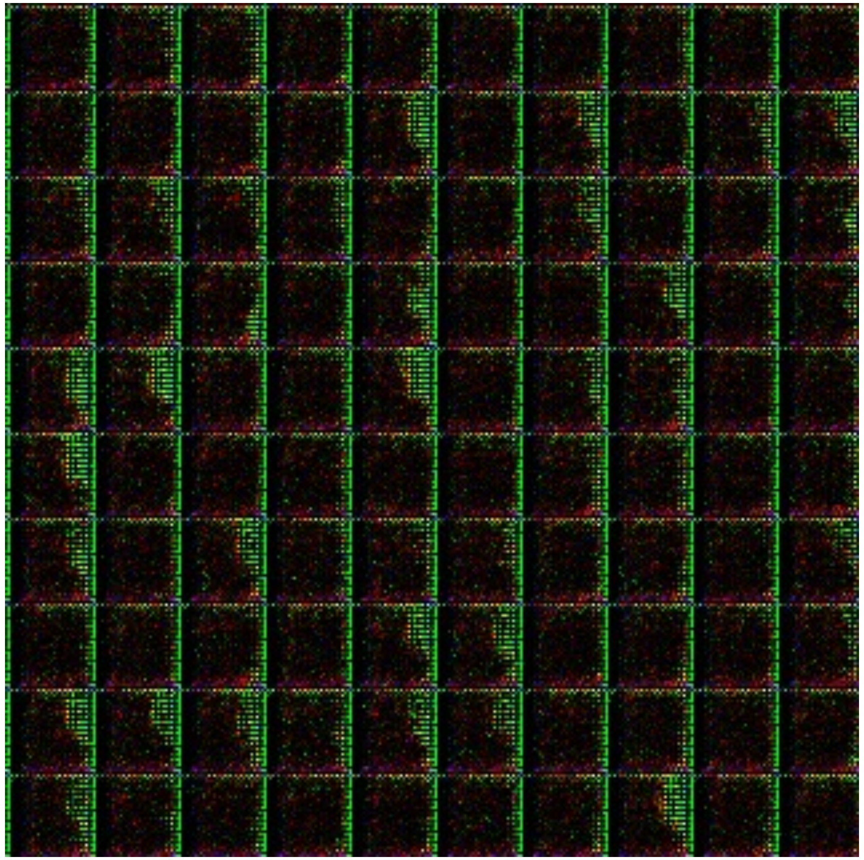
- Equilibrium of the game
- Minimizes the Jensen-Shannon divergence

Training Procedure

(Goodfellow et al., 2014)



Source: Alec Radford



Source: OpenAI blog

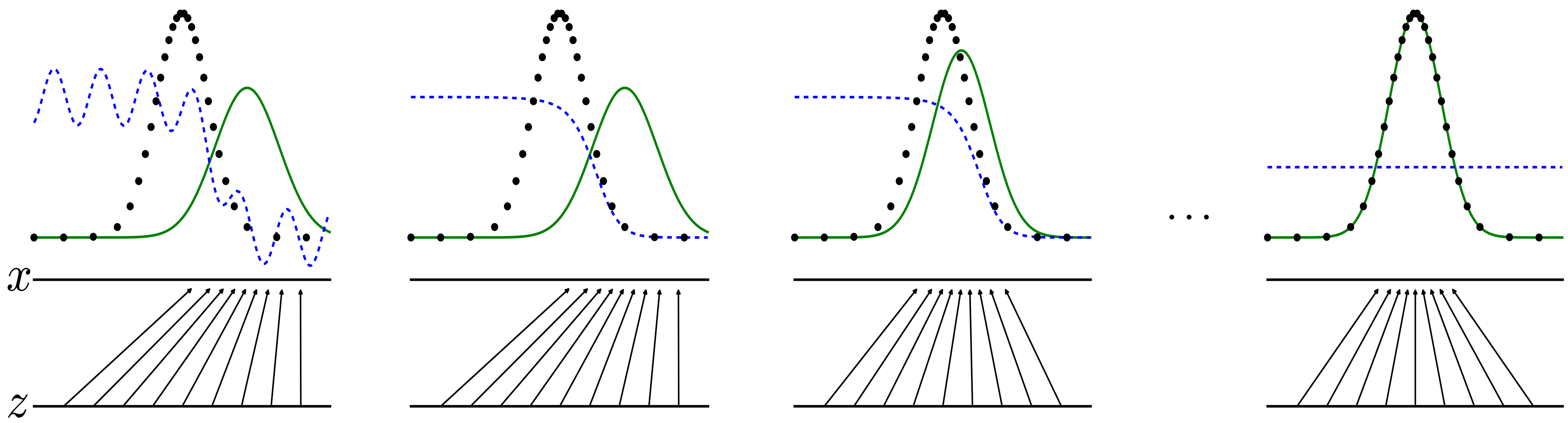
Generating 1D points

Generating images

Training Procedure

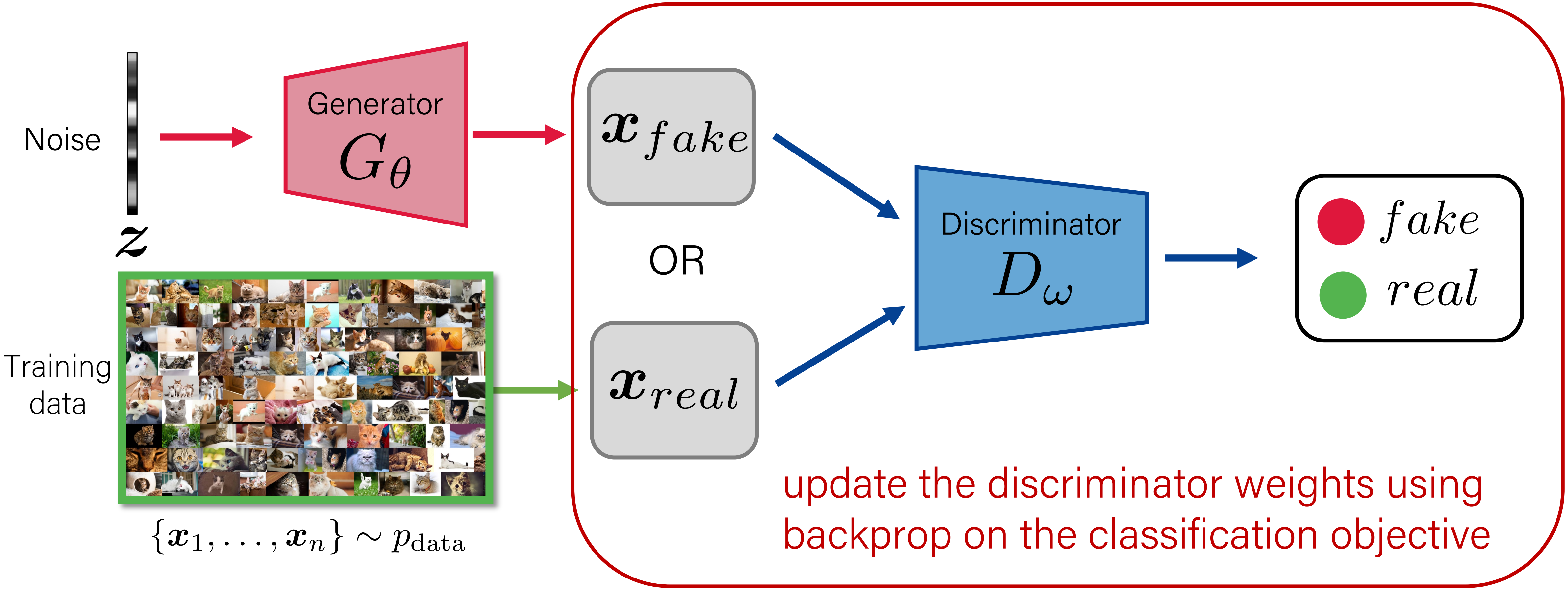
(Goodfellow et al., 2014)

- Use SGD on two minibatches simultaneously:
  - A minibatch of training examples
  - A minibatch of generated samples



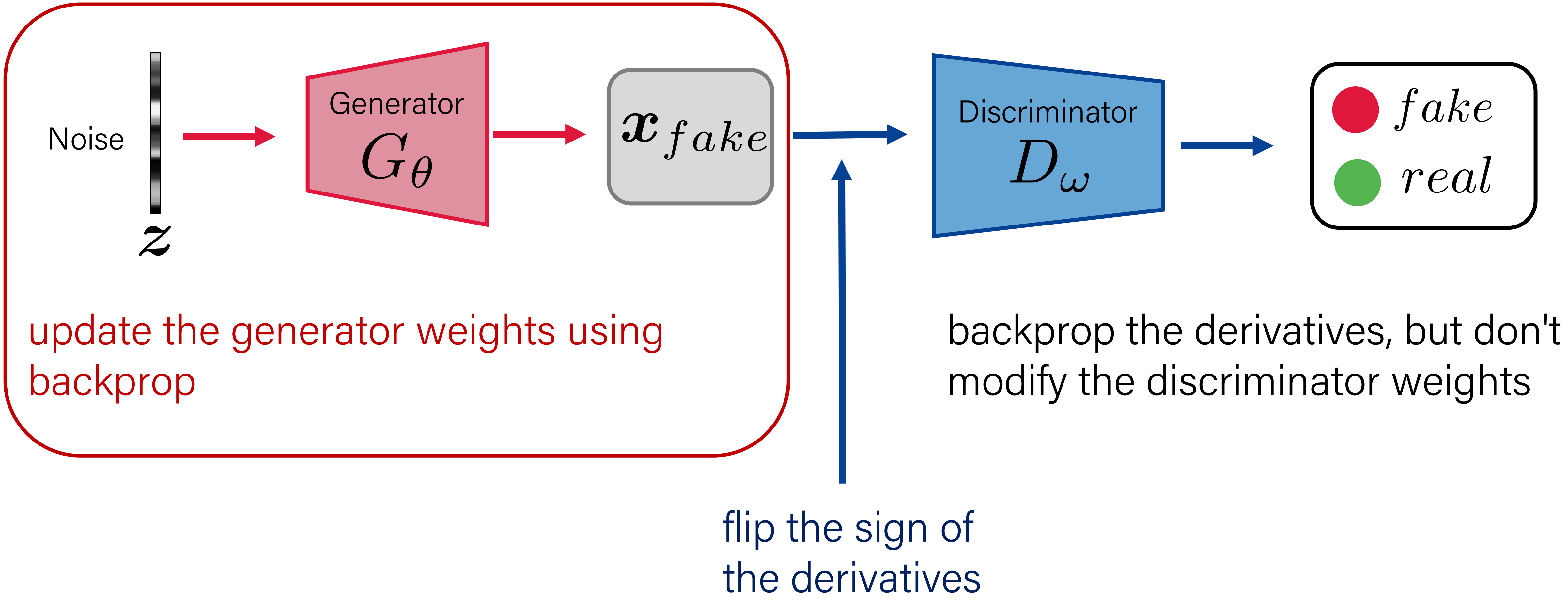
Training Procedure

- Updating the discriminator:



Training Procedure

- Updating the generator:

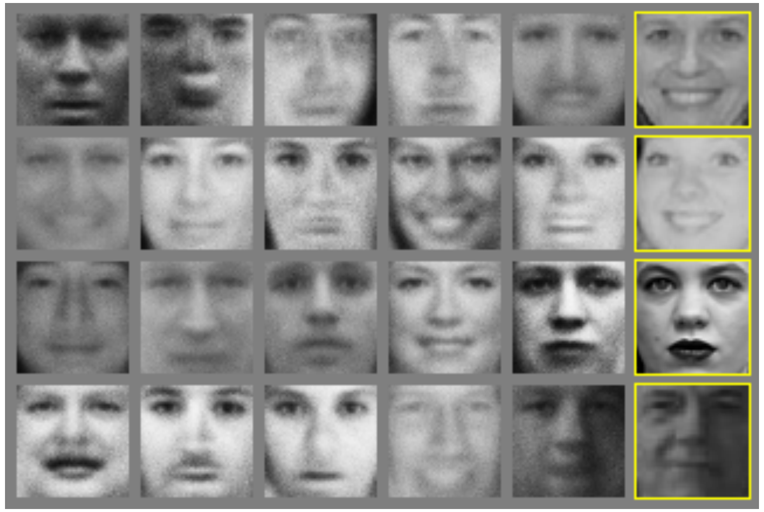


Results

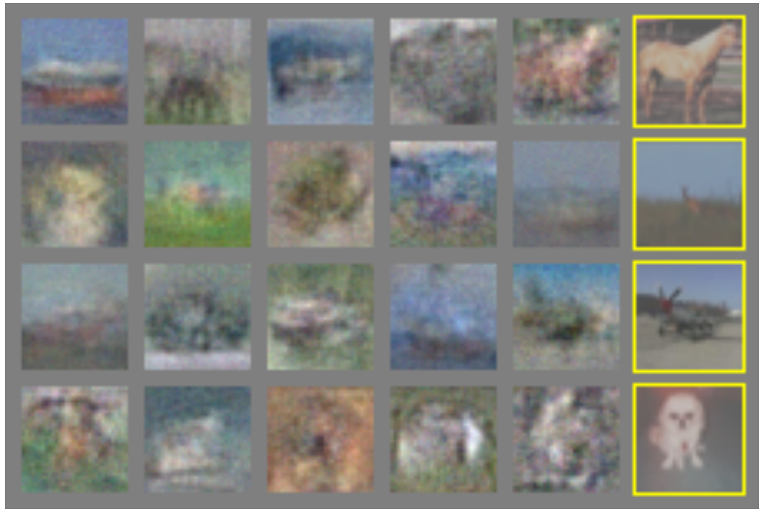
(Goodfellow et al., 2014)

- The generator uses a mixture of rectifier linear activations and/or sigmoid activations
- The discriminator net used maxout activations.

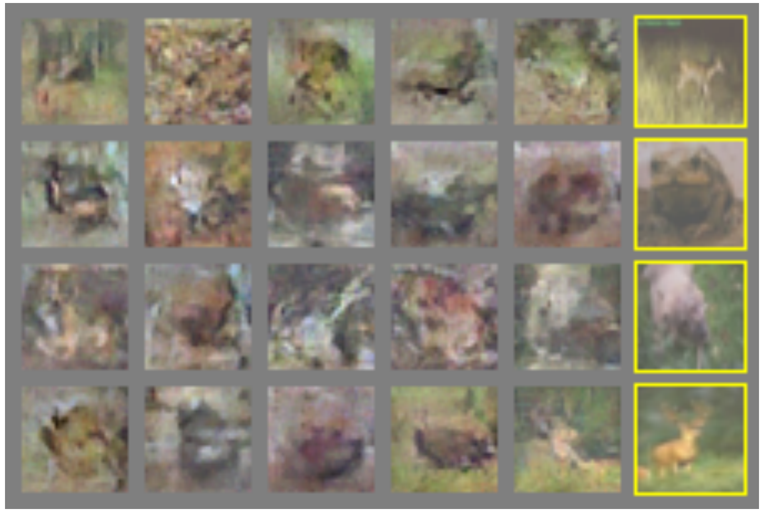
MNIST samples



TFD samples



CIFAR10 samples
(fully-connected model)

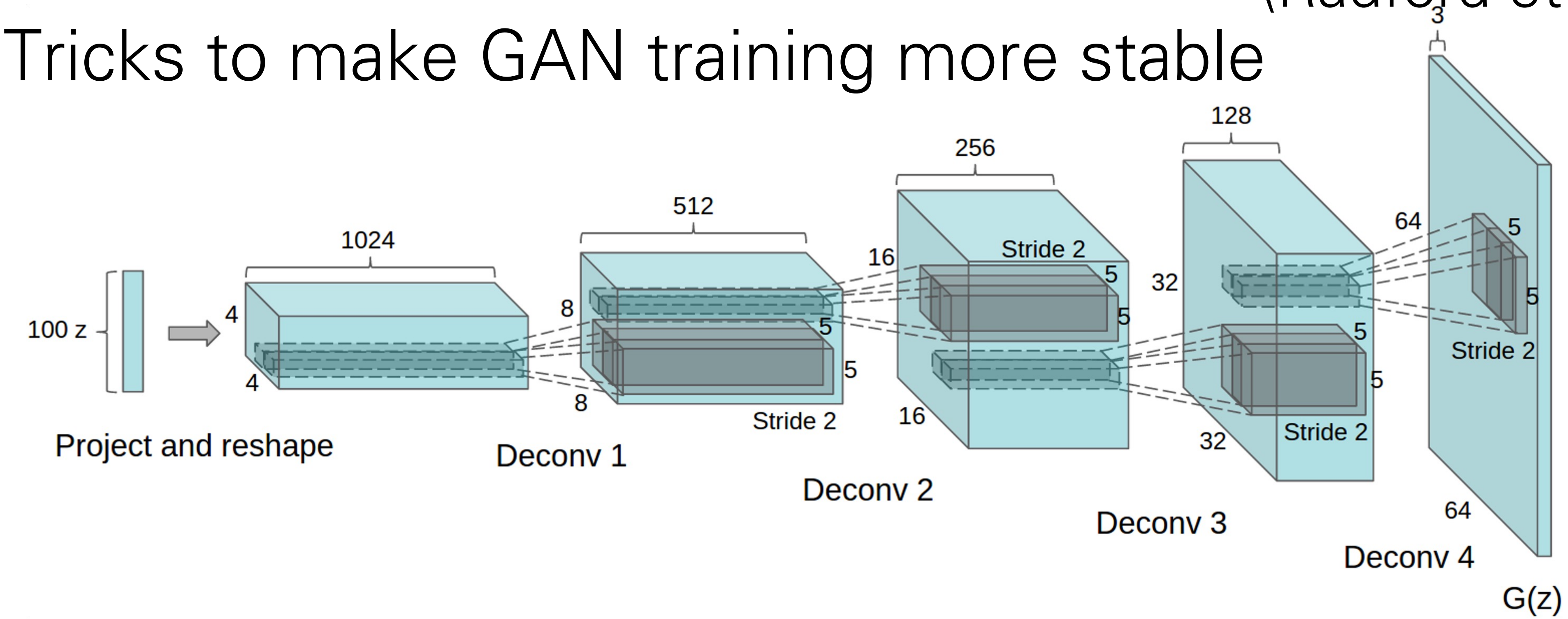


CIFAR10 samples
(convolutional discriminator, deconvolutional generator)

Deep Convolutional GANs (DCGAN)

(Radford et al., 2015)

- **Idea:** Tricks to make GAN training more stable



- No fully connected layers
- Batch Normalization (Ioffe and Szegedy, 2015)
- Leaky Rectifier in D
- Use Adam (Kingma and Ba, 2015)
- Tweak Adam hyperparameters a bit (lr=0.0002, b1=0.5)

DCGAN for LSUN Bedrooms

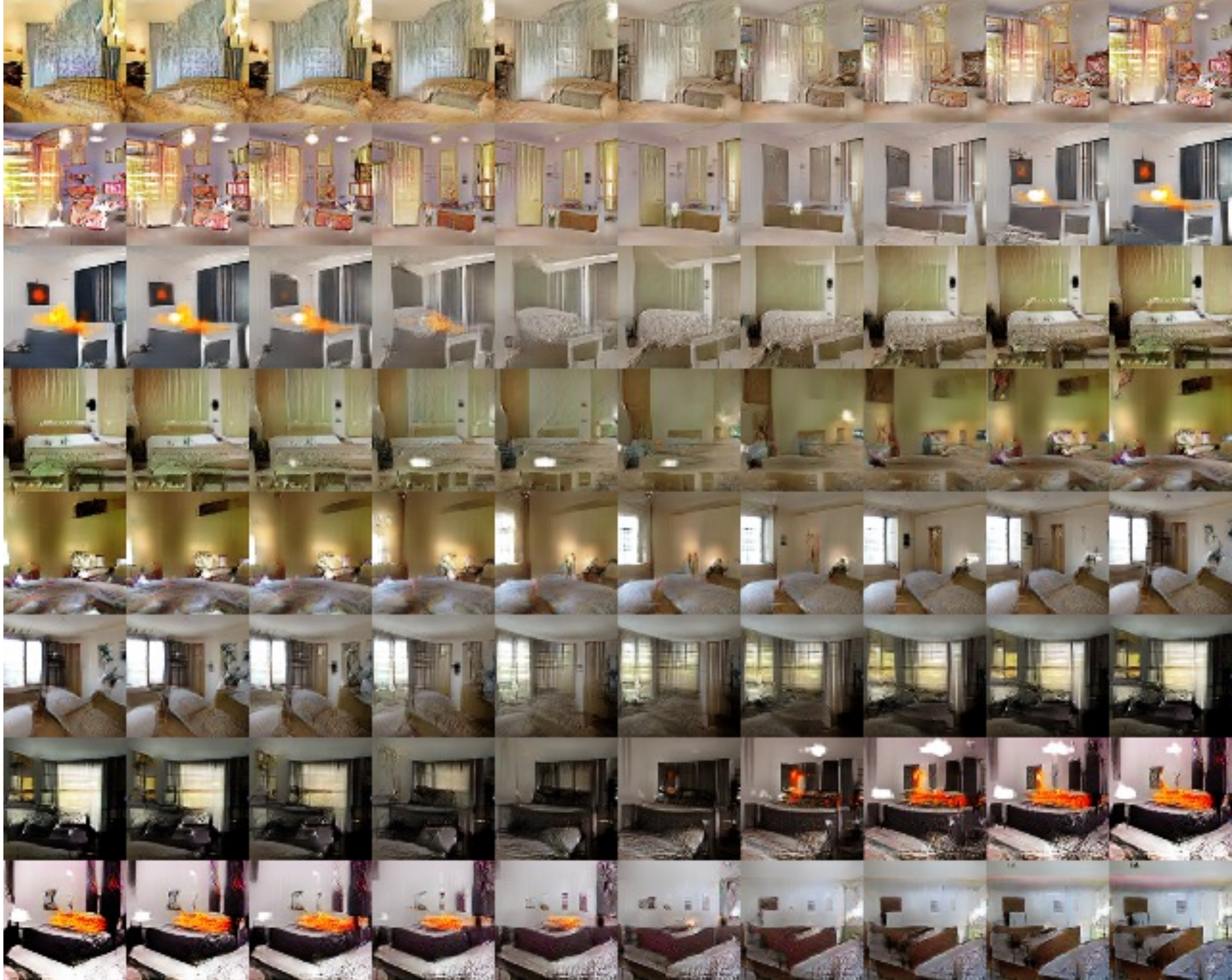
64×64 pixels
~3M images

(Radford et al., 2015)

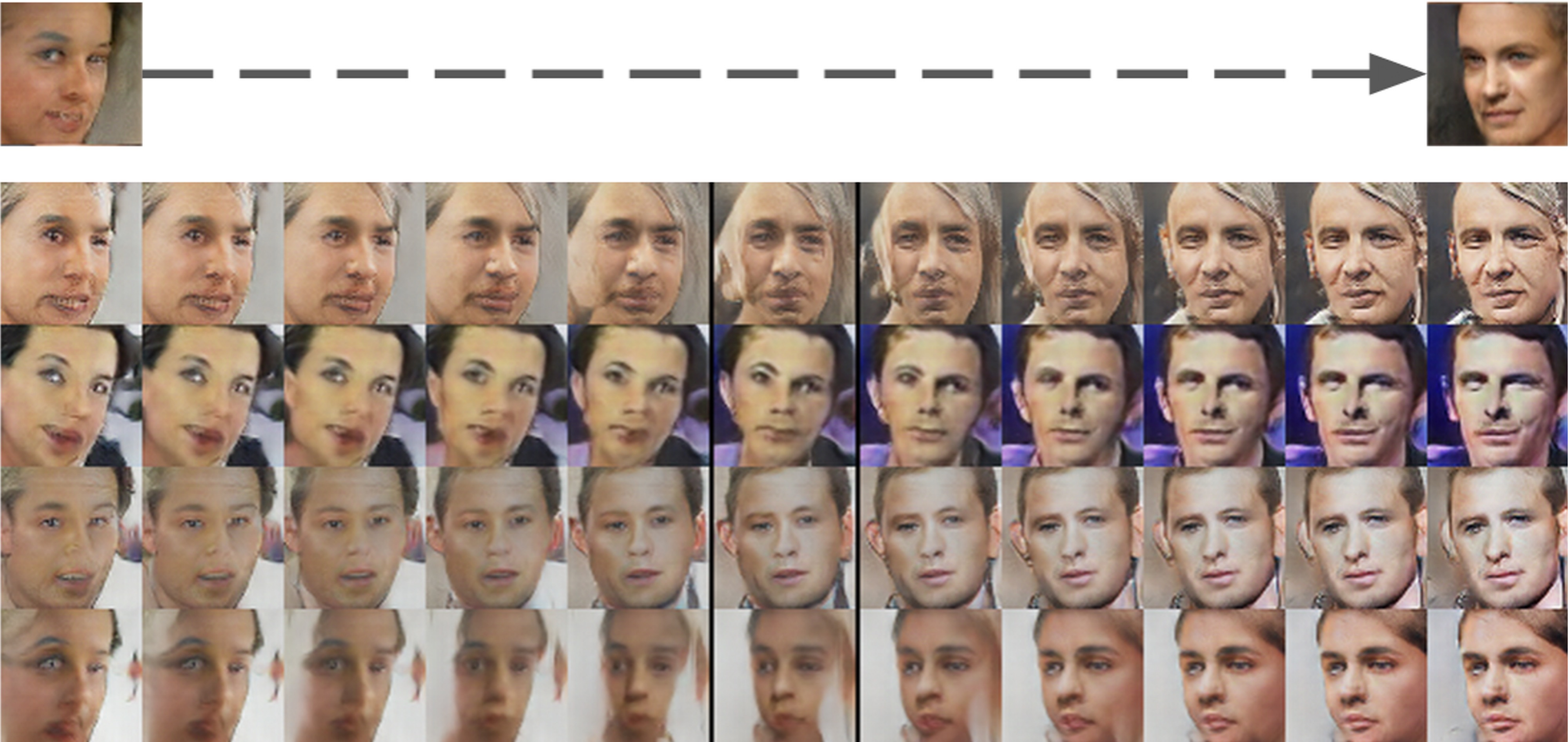
Walking over the latent space

(Radford et al., 2015)

- Interpolation suggests non-overfitting behavior

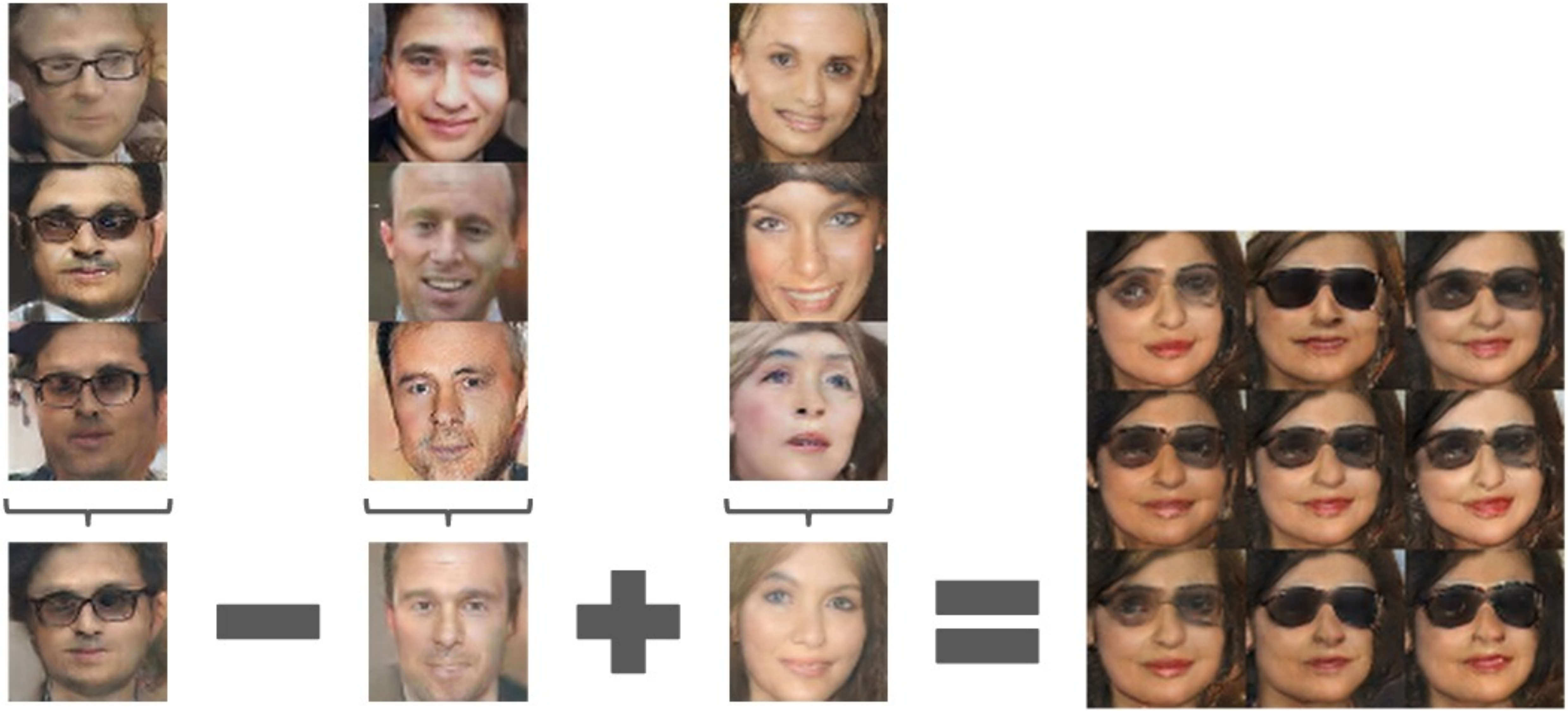


Walking over the latent space (Radford et al., 2015)



Vector Space Arithmetic

(Radford et al., 2015)



man with glasses − man without glasses + woman without glasses = woman with glasses

Vector Space Arithmetic

(Radford et al., 2015)

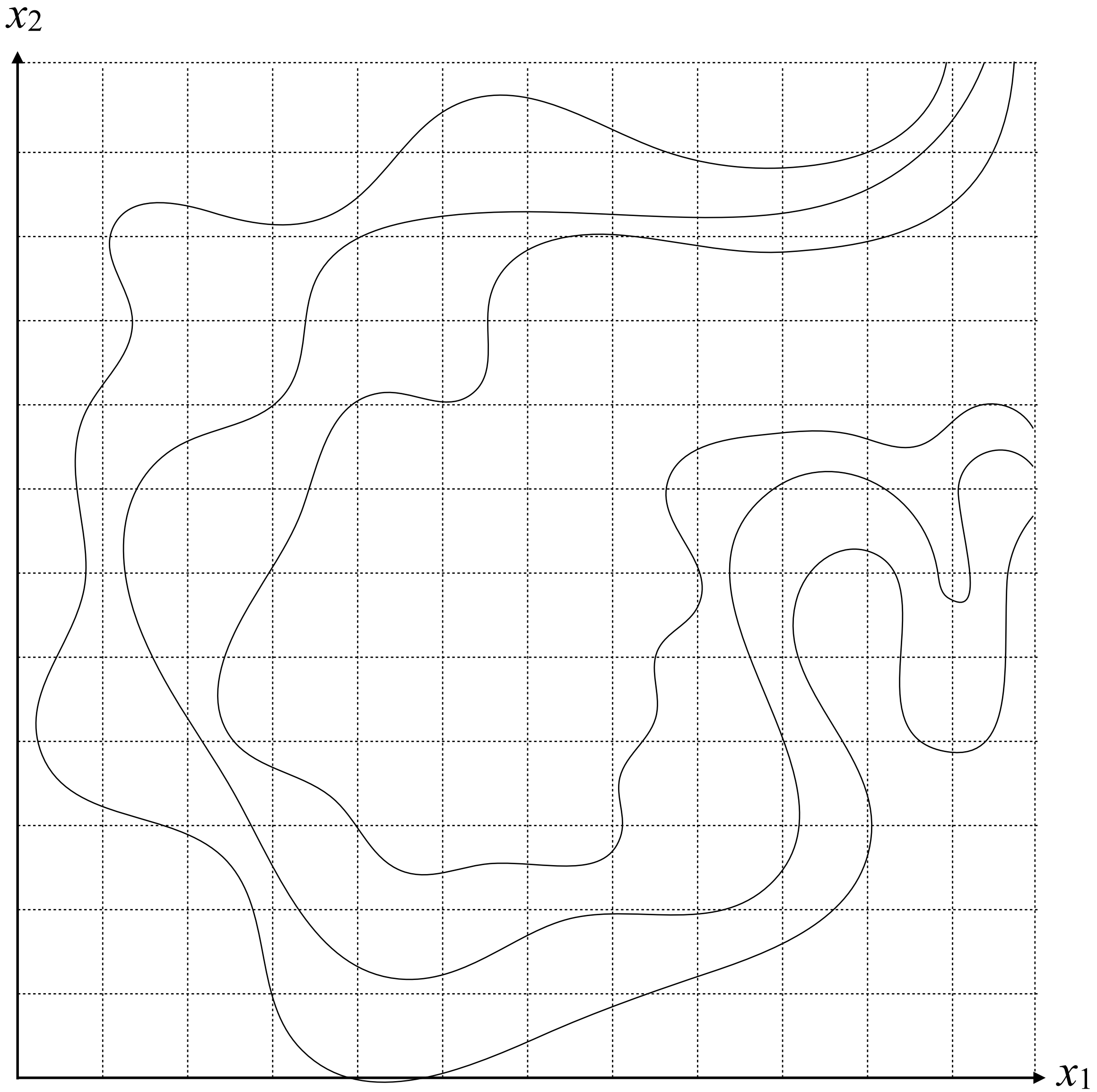
smiling woman

neutral woman

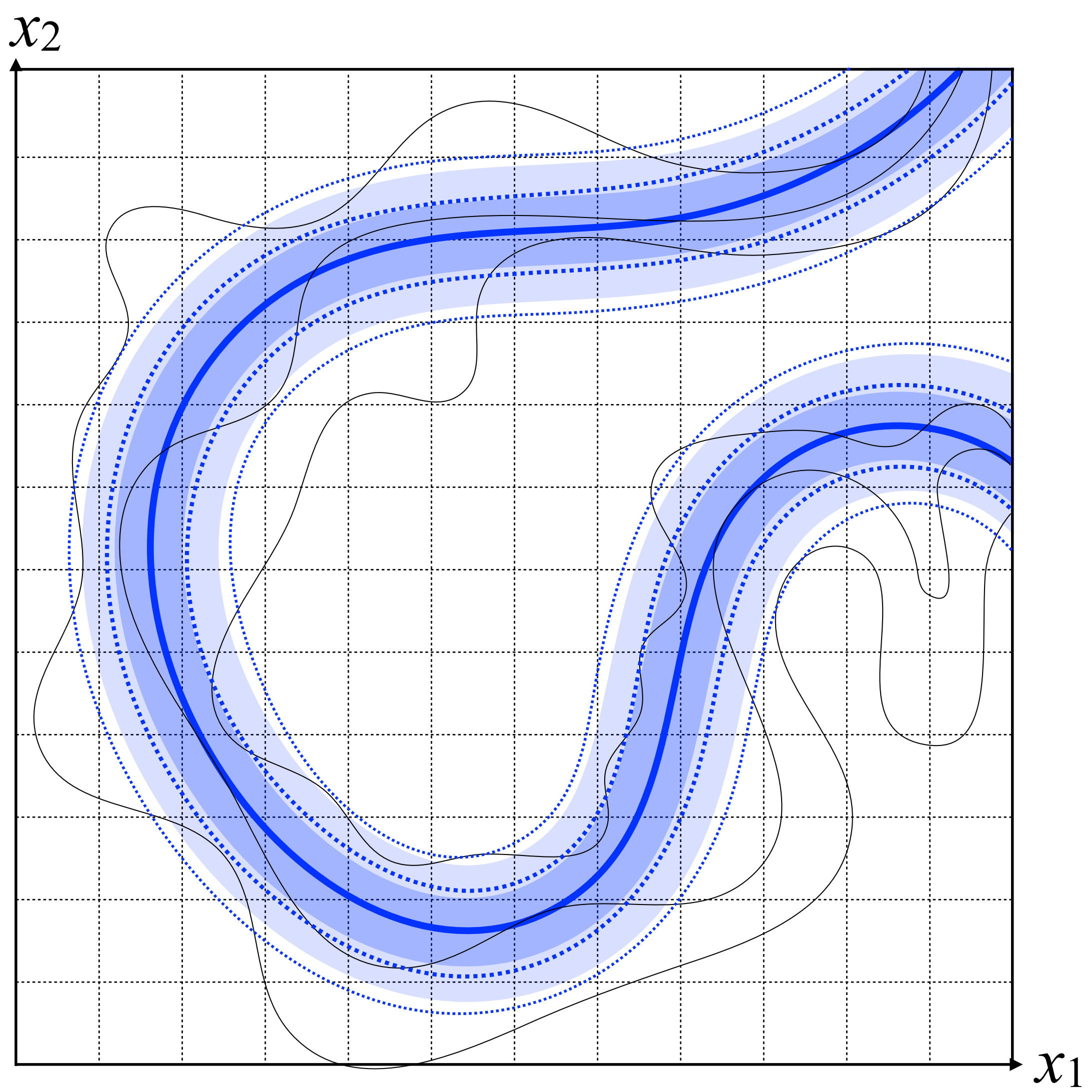
neutral man

smiling man

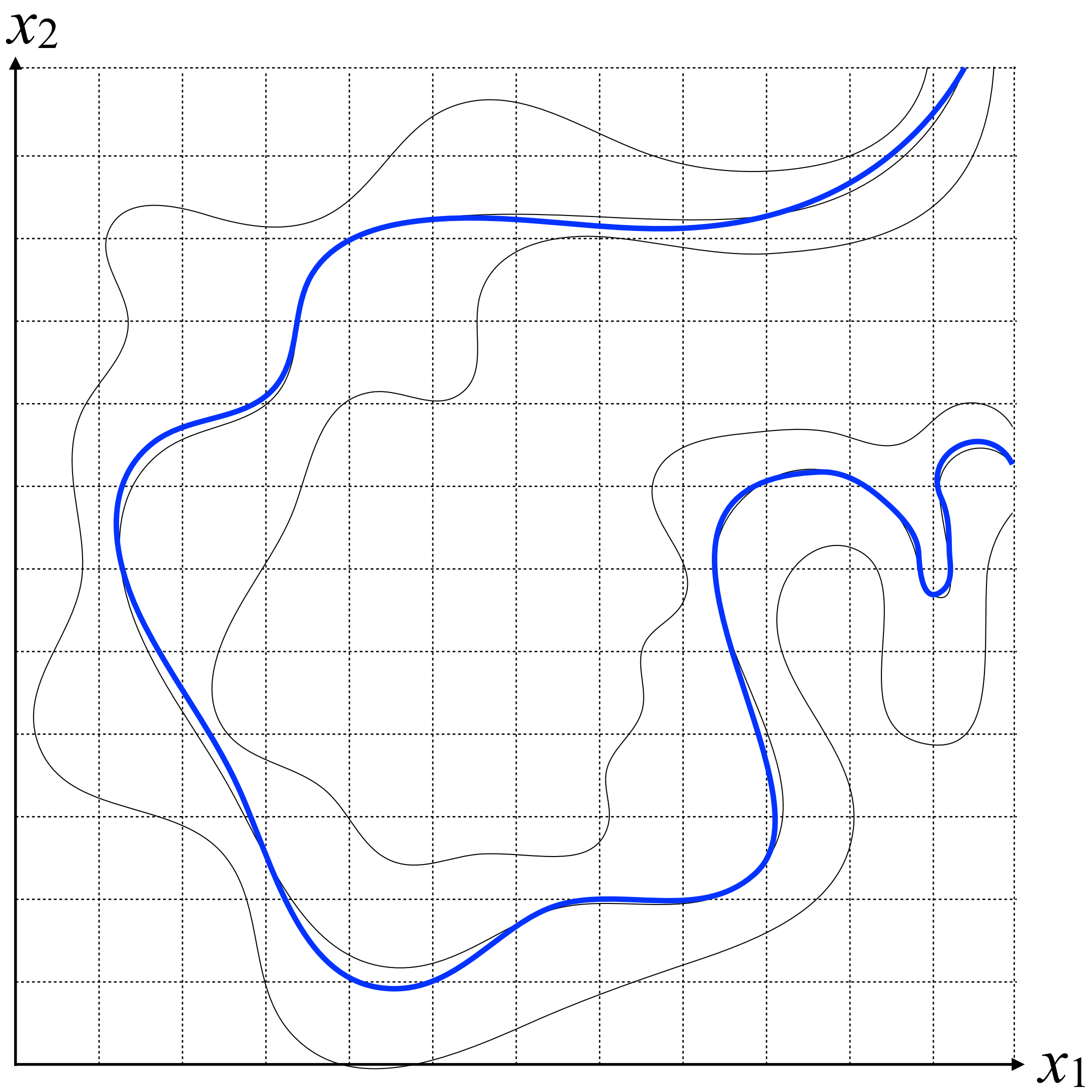
Cartoon of the Image manifold



What makes GANs special?



more traditional max-likelihood approach

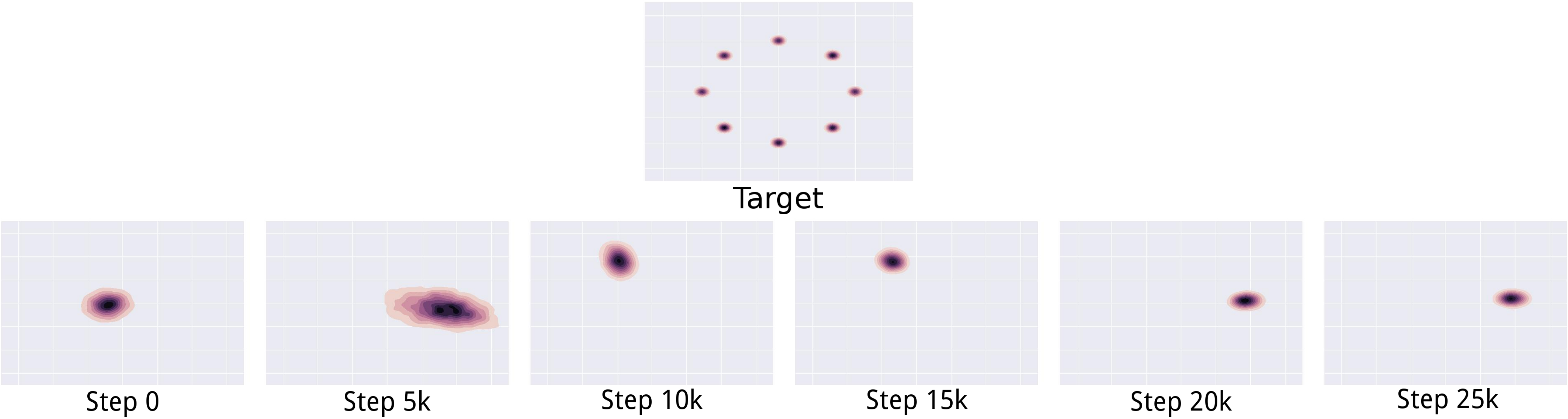


GAN

GAN Failures: Mode Collapse

$$\min_G \max_D V(G,D) \neq \max_D \min_G V(G,D)$$

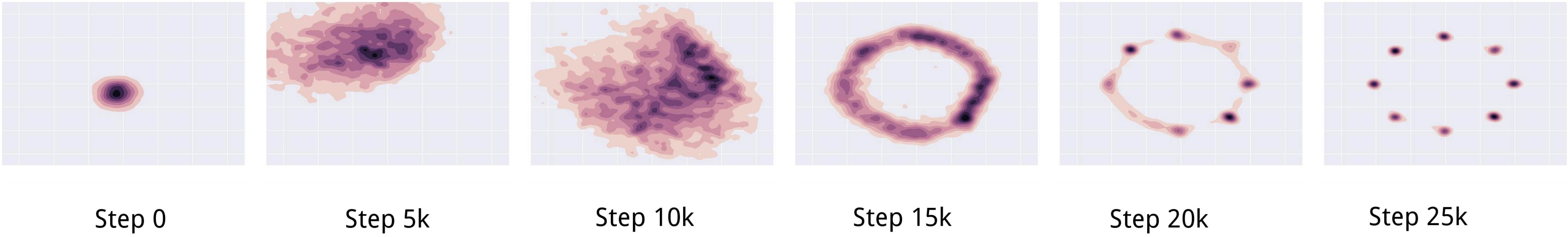
- D in inner loop: convergence to correct distribution
- G in inner loop: place all mass on most likely point



(Metz et al., 2016)

Mode Collapse: Solutions

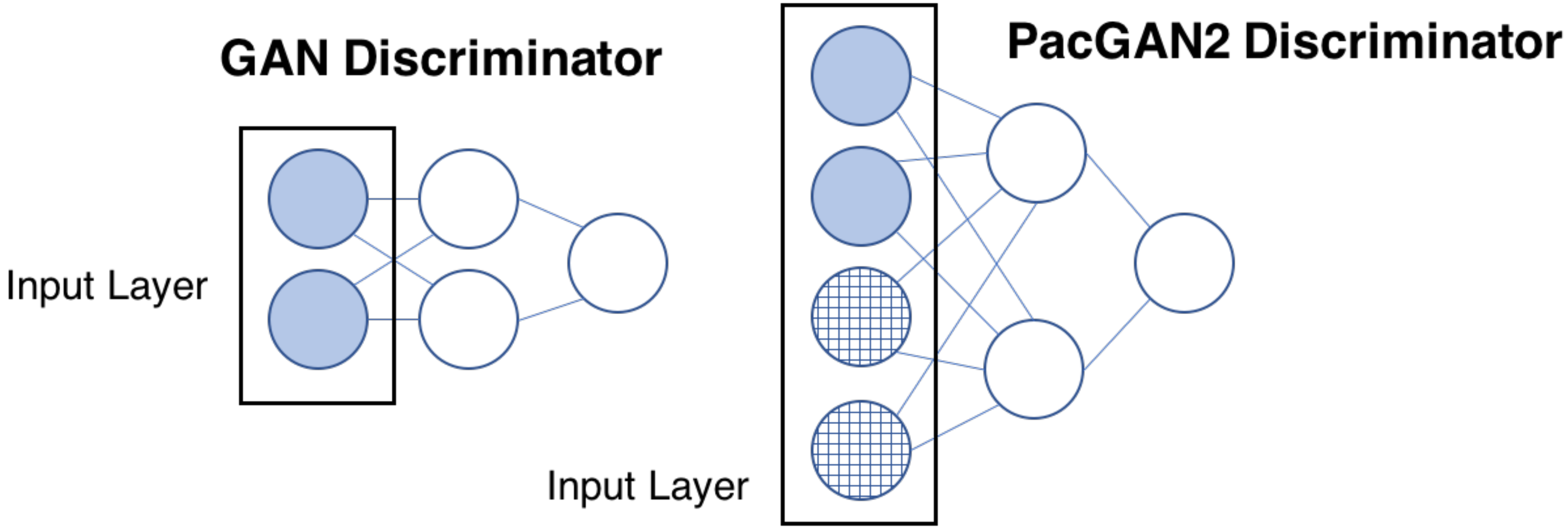
- **Unrolled GANs** (Metz et al 2016): Prevents mode collapse by backproping through a set of (k) updates of the discriminator to update generator parameters



- **VEEGAN** (Srivastava et al 2017): Introduce a reconstructor network which is learned both to map the true data distribution p(x) to a Gaussian and to approximately invert the generator network.

Mode Collapse: Solutions

- **Minibatch Discrimination** (Salimans et al 2016): Add minibatch features that classify each example by comparing it to other members of the minibatch (Salimans et al 2016)
- **PacGAN:** The power of two samples in generative adversarial networks (Lin et al 2017): Also uses multisample discrimination.



Mode Collapse: Solutions

- **PacGAN:** The power of two samples in generative adversarial networks (Lin et al 2017): Also uses multisample discrimination.

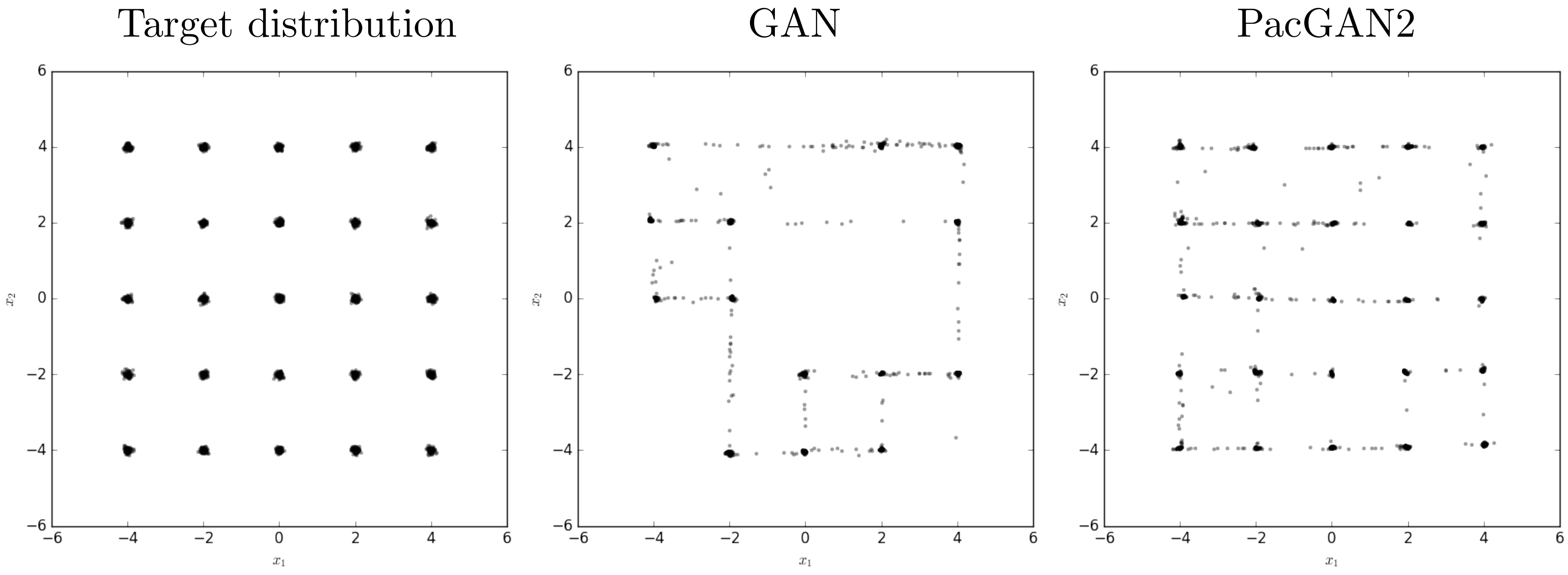


Figure 2: Scatter plot of the 2D samples from the true distribution (left) of 2D-grid and the learned generators using GAN (middle) and PacGAN2 (right). PacGAN2 captures all of the 25 modes.

GAN Evaluation

- Quantitatively evaluating GANs is not straightforward:
  - Max Likelihood is a poor indication of sample quality

- Some evaluation metrics
  - **Inception Score (IS)**:
    y = labels given gen. image. p(y|x) is from classifier - InceptionNet

$$\mathrm{IS}(\mathbb{P}_g) = e^{\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_g}[KL(p_{\mathcal{M}}(y|\mathbf{x})||p_{\mathcal{M}}(y))]}$$

  - **Fréchet inception distance (FID):** (Currently most popular)
    Estimate mean m and covariance C from classifier output - InceptionNet

$$d^2((\boldsymbol{m},\boldsymbol{C}),(\boldsymbol{m}_w,\boldsymbol{C}_w)) = \|\boldsymbol{m}-\boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\big(\boldsymbol{C}+\boldsymbol{C}_w - 2\big(\boldsymbol{C}\boldsymbol{C}_w\big)^{1/2}\big)$$

  - **Kernel MMD** (Maximum Mean Discrepancy):

$$\mathrm{MMD}(\mathbb{P}_r,\mathbb{P}_g) = \left(\mathbb{E}_{\substack{\mathbf{x}_r,\mathbf{x}_r'\sim\mathbb{P}_r,\\ \mathbf{x}_g,\mathbf{x}_g'\sim\mathbb{P}_g}}\left[k(\mathbf{x}_r,\mathbf{x}_r') - 2k(\mathbf{x}_r,\mathbf{x}_g) + k(\mathbf{x}_g,\mathbf{x}_g')\right]\right)^{\frac{1}{2}}$$

Subclasses of GANs

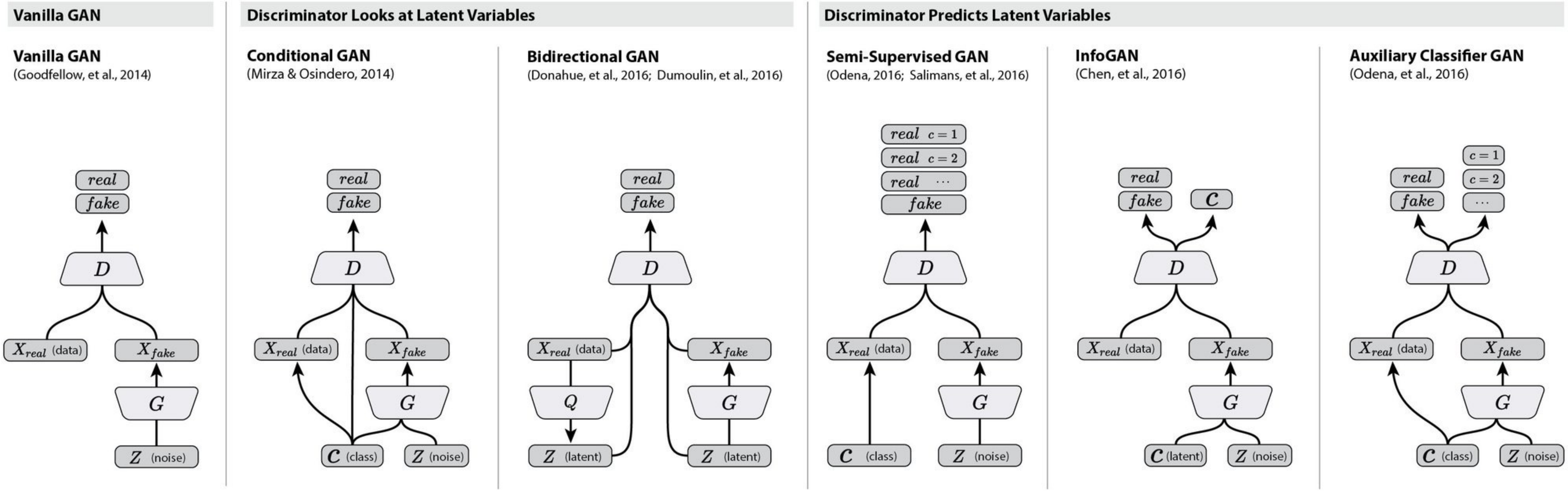
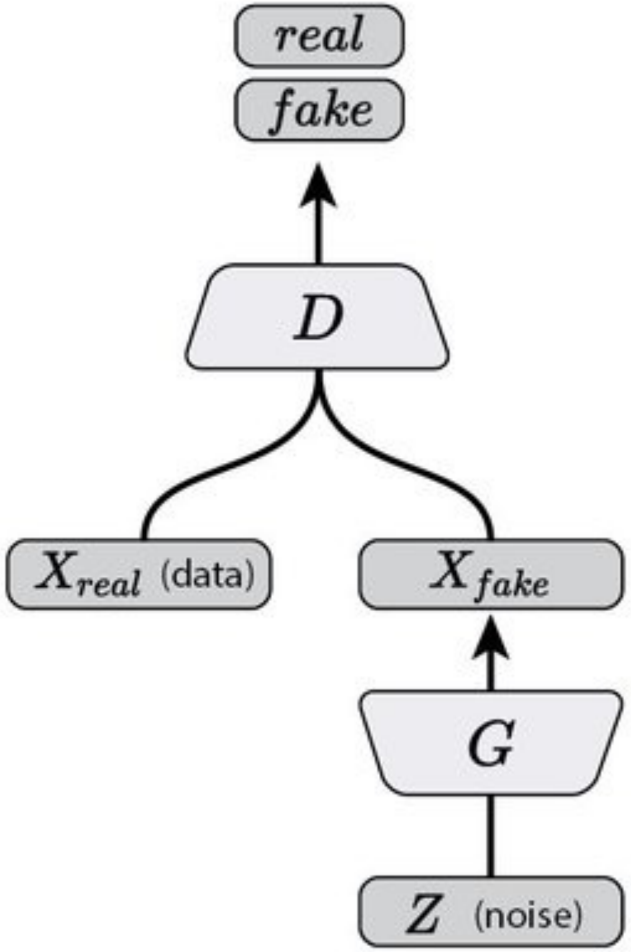
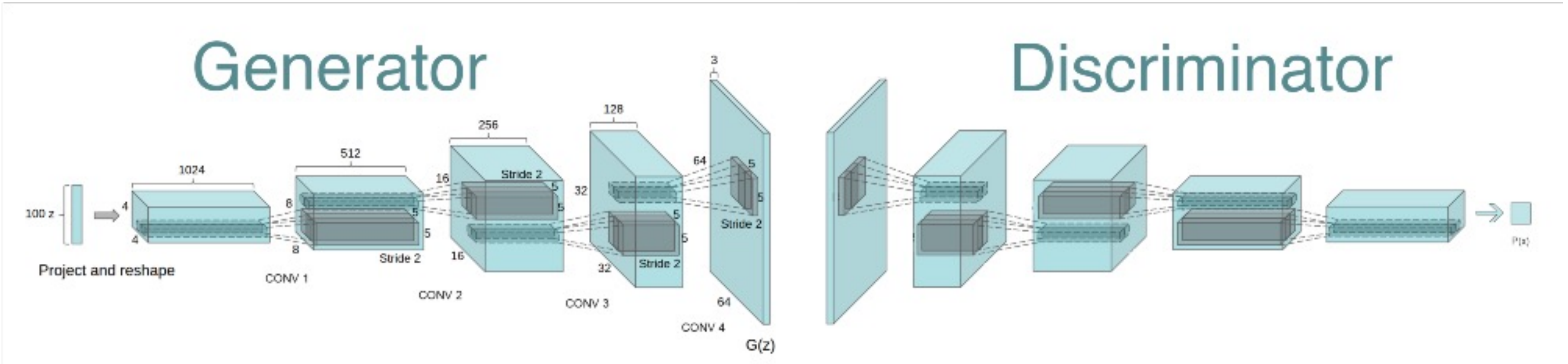


Image: Christopher Olah

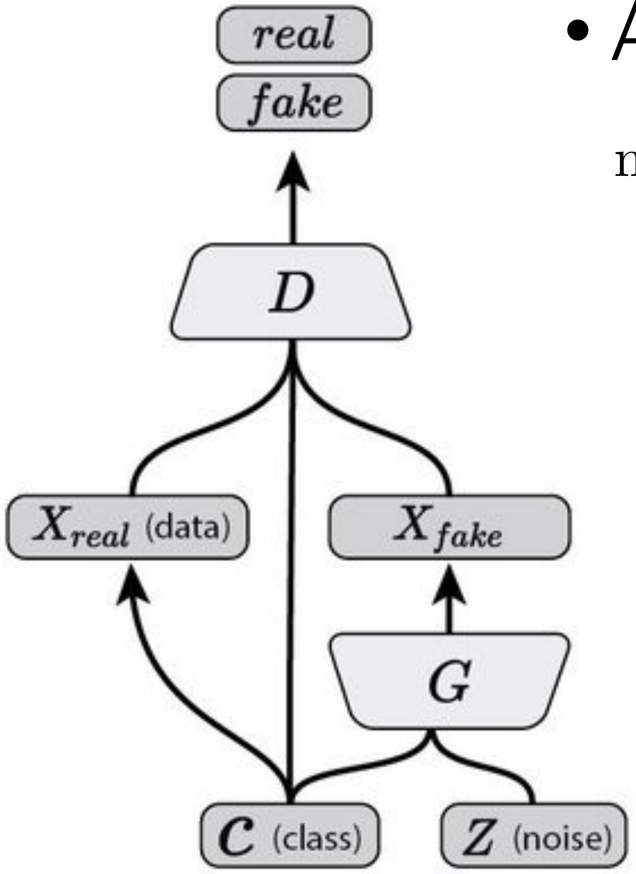
Vanilla GAN (Goodfellow et al., 2014)





DCGAN (Radford et al., 2015)

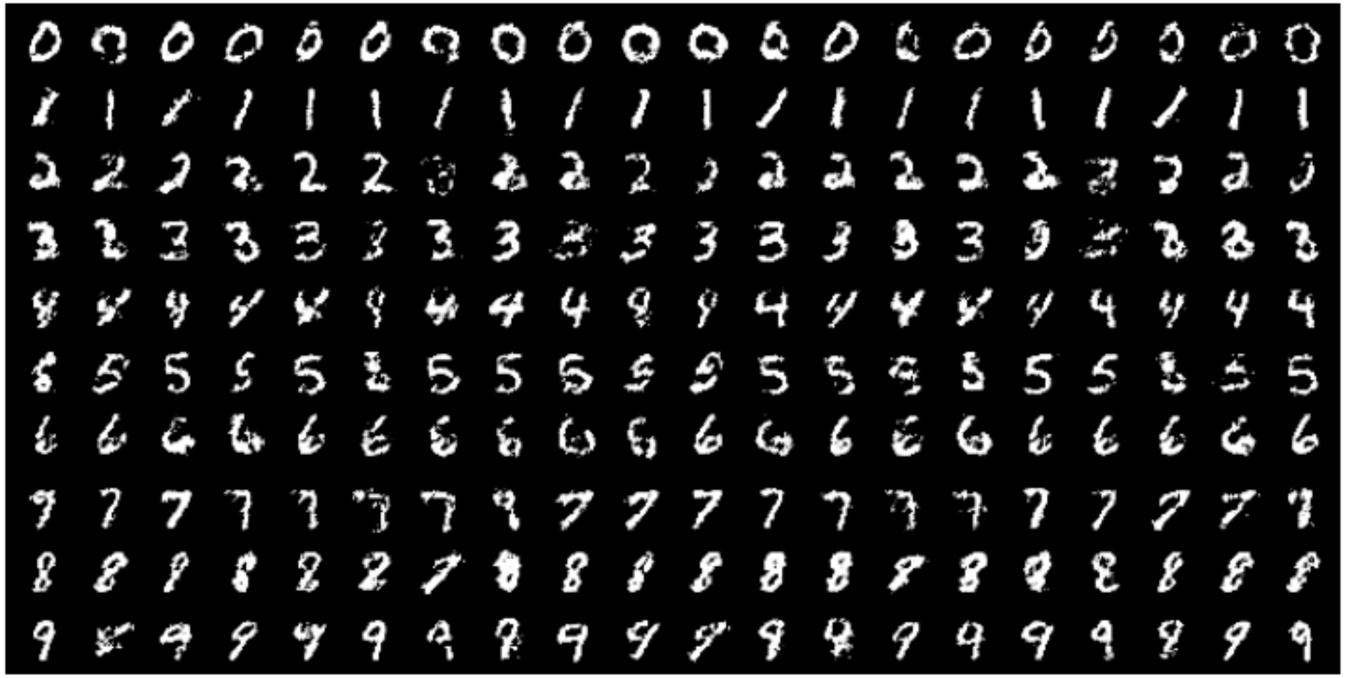
Conditional GAN (Mirza and Osindero, 2014)



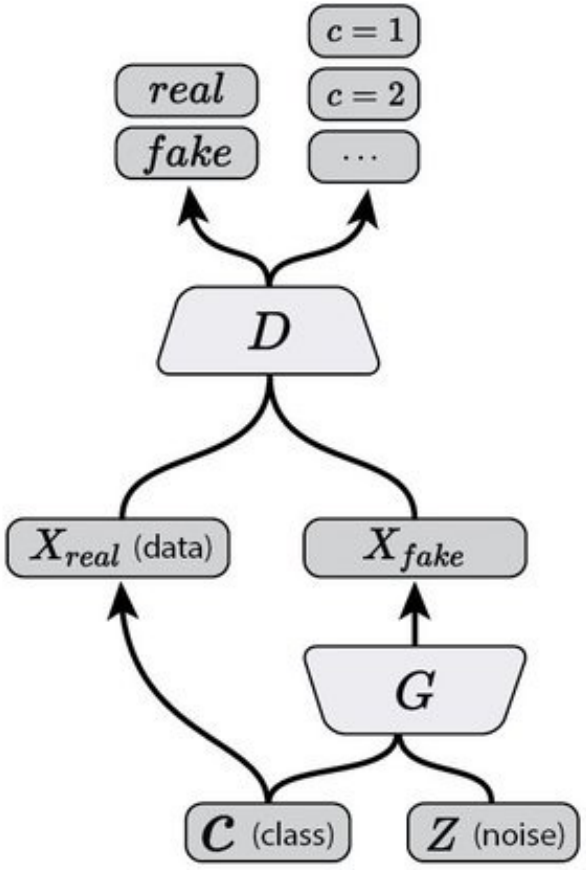
- Add conditional variables $\boldsymbol{y}$ into G and D

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$$

0	1	0	0	0	0	0	0	0	0



Auxiliary Classifier GAN (Odena et al., 2016)



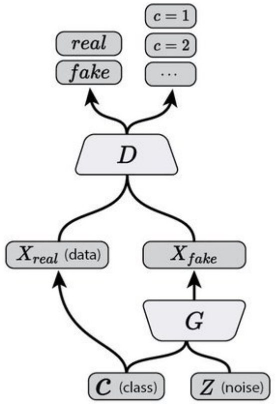
- Every generated sample has a corresponding class label

$$L_S = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$$
$$L_C = E[\log P(C = c \mid X_{real})] + E[\log P(C = c \mid X_{fake})]$$

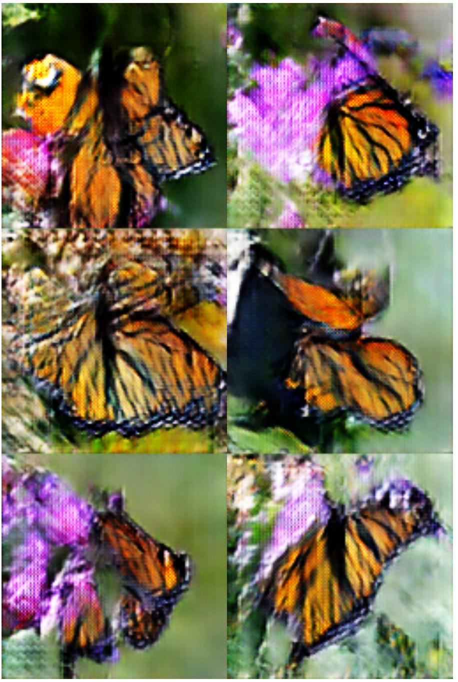
- D is trained to maximize $L_S + L_C$
- G is trained to maximize $L_C - L_S$

- Learns a representation for z that is independent of class label

Auxiliary Classifier GAN (Odena et al., 2016)



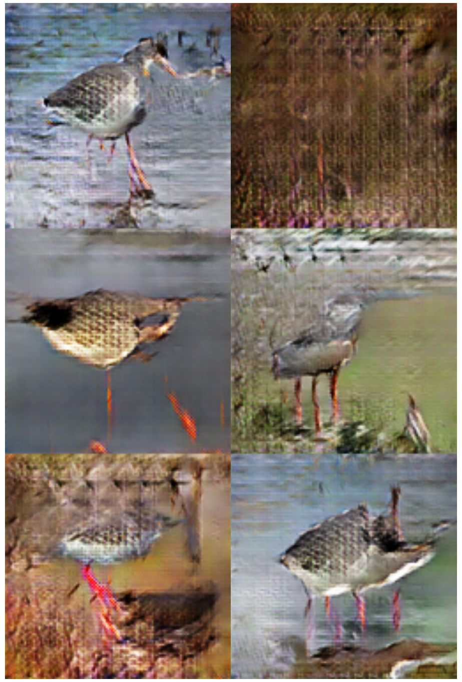
128×128 resolution samples from 5 classes taken from an AC-GAN trained on the ImageNet



monarch butterfly

goldfinch

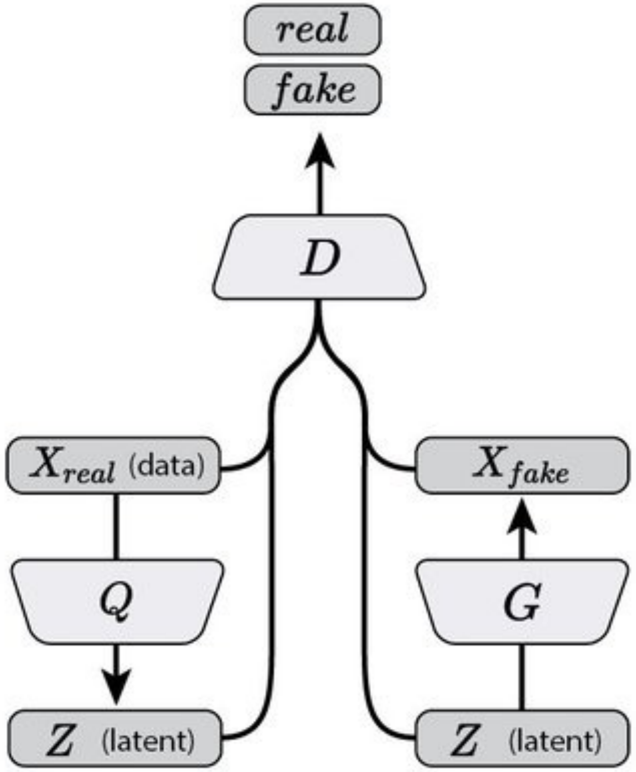
daisy



redshank

grey whale

Bidirectional GAN (Donahue et al., 2016; Dumoulin et al., 2016)



- Jointly learns a generator network and an inference network using an adversarial process.

$$\min_G \max_D V(D,G) = \mathbb{E}_{q(\boldsymbol{x})}[\log(D(\boldsymbol{x}, G_z(\boldsymbol{x})))] + \mathbb{E}_{p(\boldsymbol{z})}[\log(1 - D(G_x(\boldsymbol{z}), \boldsymbol{z}))]$$

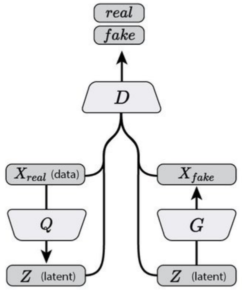
$$= \iint q(\boldsymbol{x}) q(\boldsymbol{z} \mid \boldsymbol{x}) \log(D(\boldsymbol{x}, \boldsymbol{z})) d\boldsymbol{x} d\boldsymbol{z}$$

$$+ \iint p(\boldsymbol{z}) p(\boldsymbol{x} \mid \boldsymbol{z}) \log(1 - D(\boldsymbol{x}, \boldsymbol{z})) d\boldsymbol{x} d\boldsymbol{z}.$$

CelebA reconstructions

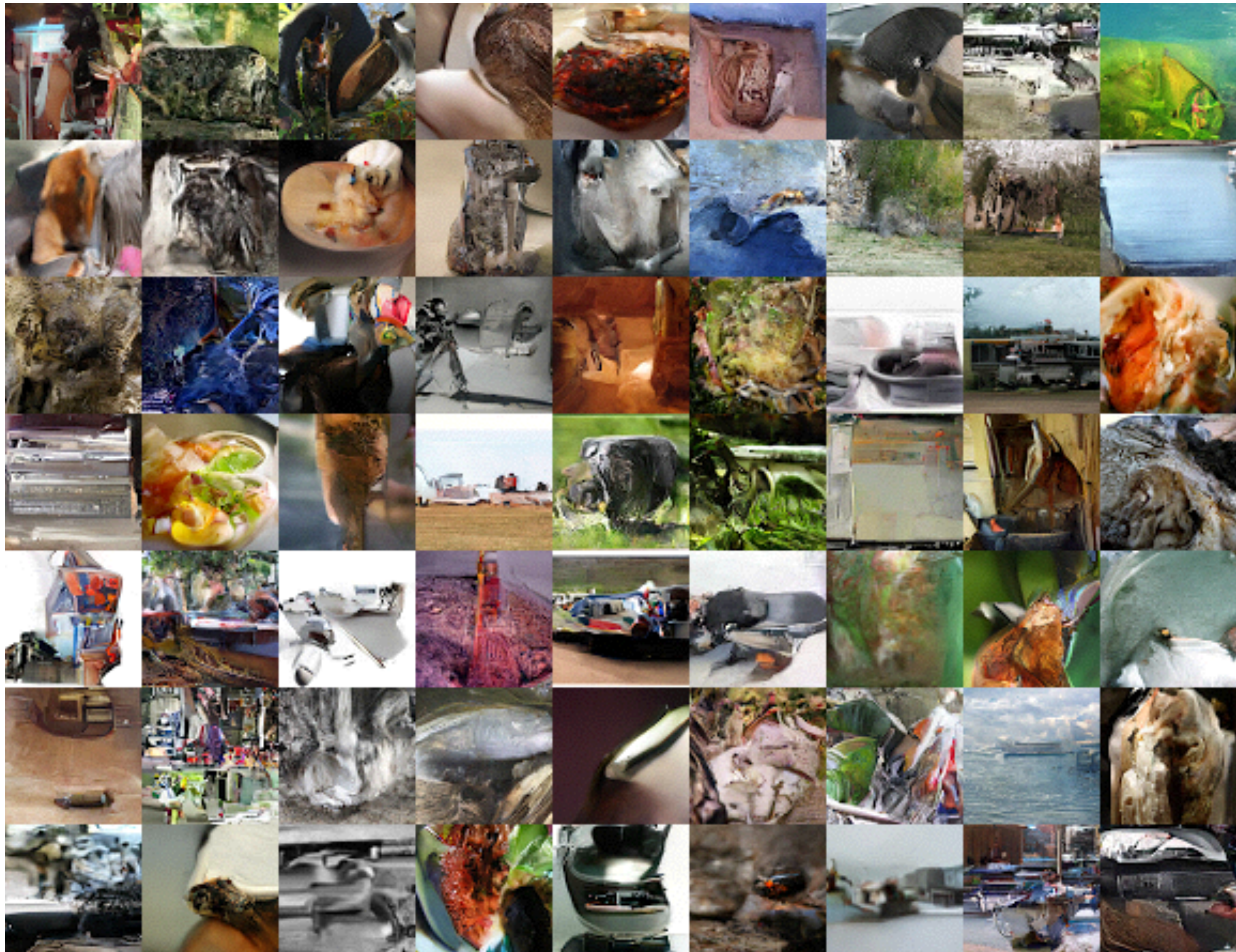
SVNH reconstructions

Bidirectional GAN (Donahue et al., 2016; Dumoulin et al., 2016)



LSUN bedrooms

Tiny ImageNet

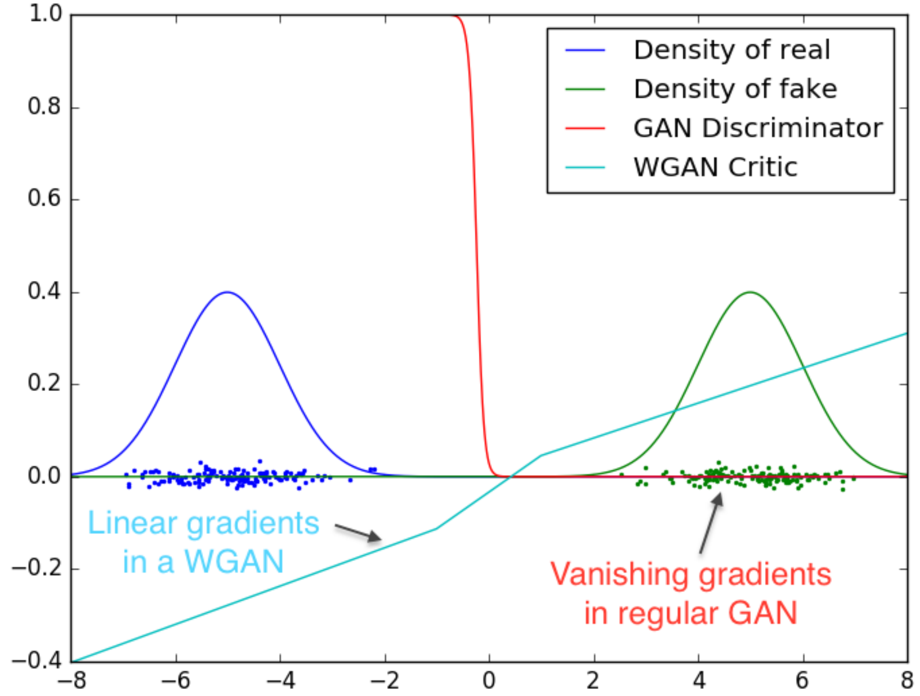


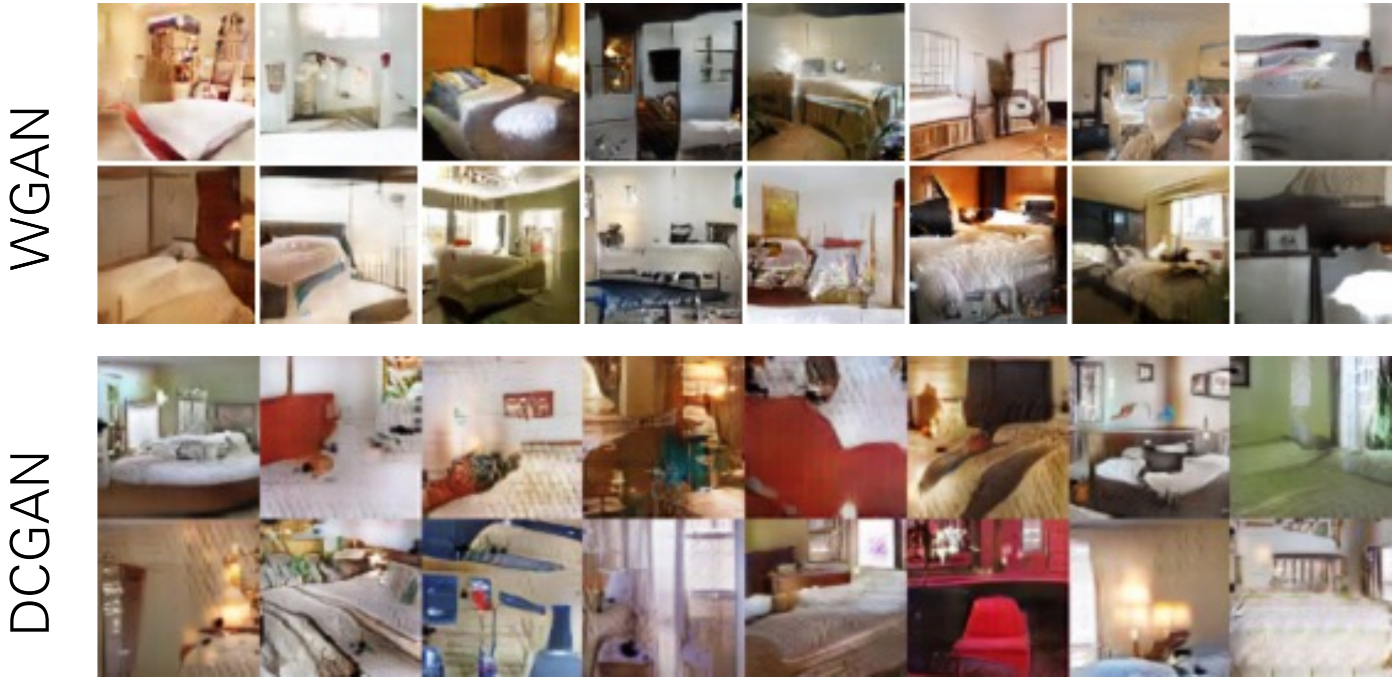
Wasserstein GAN (Arjovsky et al., 2016)

- Objective based on Earth-Mover or Wassertein distance:

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[D_\omega(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}\left[D_\omega(G_\theta(\boldsymbol{z}))\right]$$

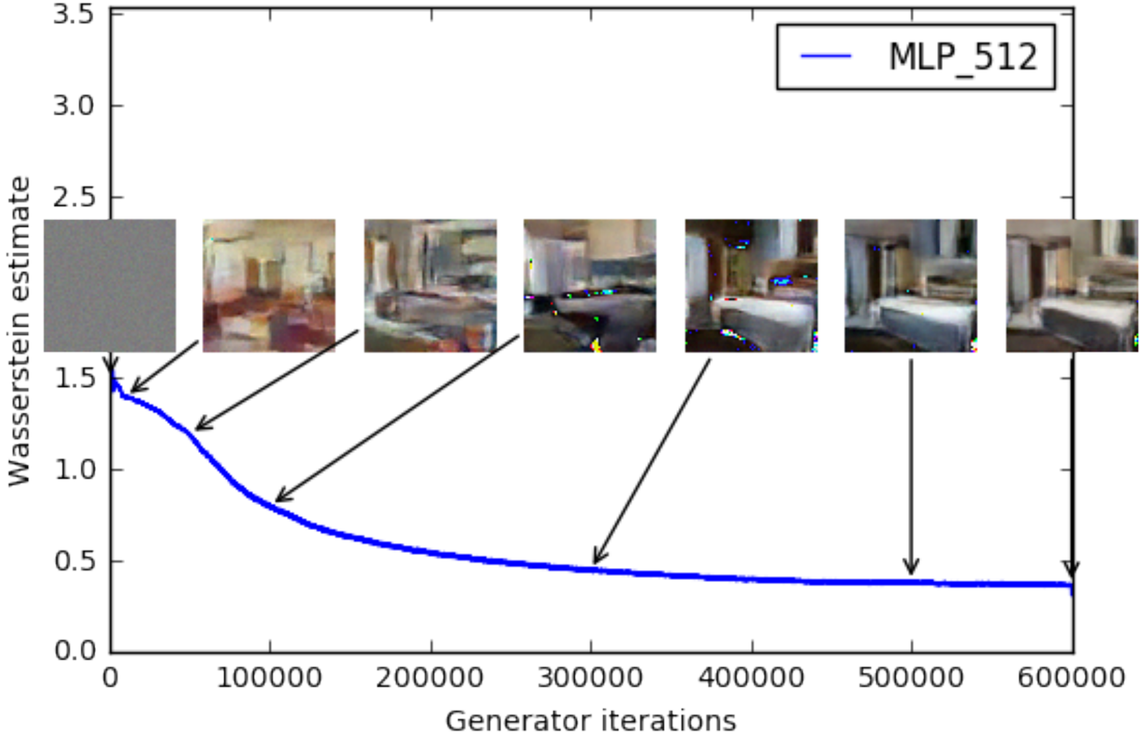
- Provides nice gradients over real and fake samples

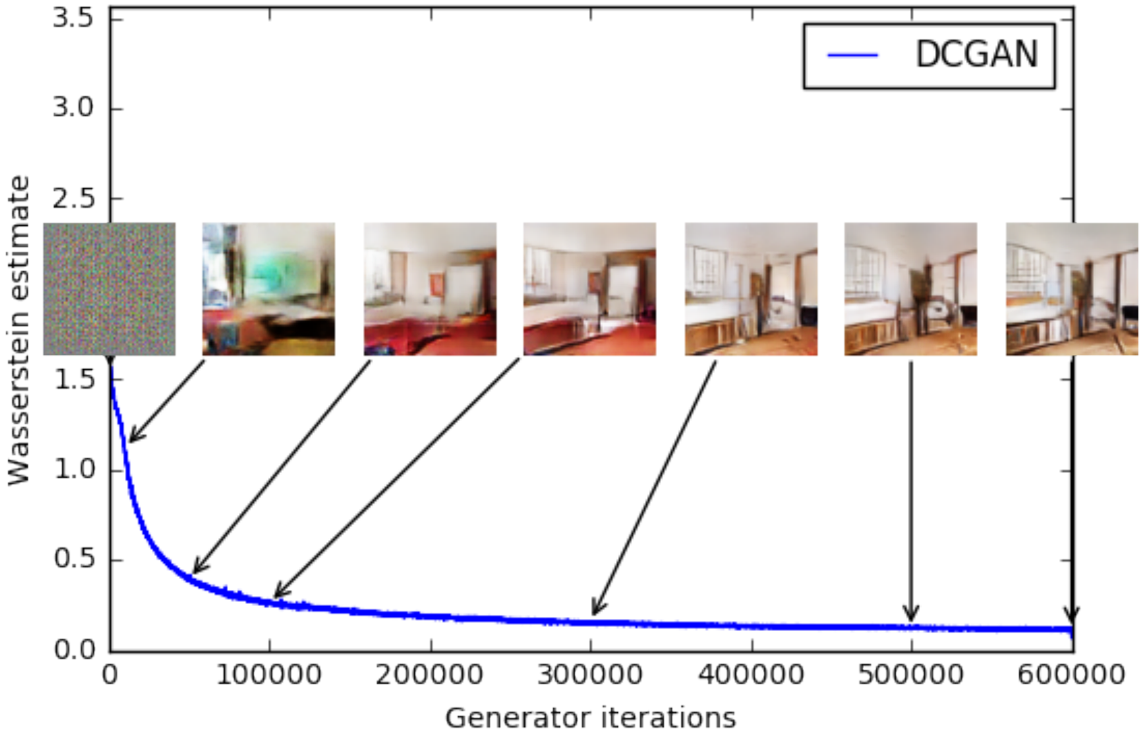




Wasserstein GAN (Arjovsky et al., 2016)

- Wasserstein loss seems to correlate well with image quality.





WGAN with gradient penalty (Gulraani et al., 2017)

$$L = \underbrace{\underset{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g}{\mathbb{E}} [D(\tilde{\boldsymbol{x}})] - \underset{\boldsymbol{x} \sim \mathbb{P}_r}{\mathbb{E}} [D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \underset{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}}{\mathbb{E}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2 \right]}_{\text{Our gradient penalty}}$$

- Faster convergence and higher-quality samples than WGAN with weight clipping
- Train a wide variety of GAN architectures with almost no hyperparameter tuning, including discrete models

WGAN with gradient penalty	
Busino game camperate spent odea	Solice Norkedin pring in since
In the bankaway of smarling the	ThiS record (31.) UBS) and Ch
SingersMay , who kill that imvic	It was not the annuas were plogr
Keray Pents of the same Reagun D	This will be us , the ect of DAN
Manging include a tudancs shat "	These leaded as most-worsd p2 a0
His Zuith Dudget , the Denmbern	The time I paid0a South Cubry i
In during the Uitational questio	Dour Fraps higs it was these del
Divos from The ' noth ronkies of	This year out howneed allowed lo
She like Monday , of macunsuer S	Kaulna Seto consficutes to repor
The investor used ty the present	A can teal , he was schoon news
A papees are cointry congress oo	In th 200. Pesish picriers rega
A few year inom the group that s	Konney Panice rimimber the teami
He said this syenn said they wan	The new centuct cut Denester of
As a world 1 88 ,for Autouries	The near , had been one injostie
Foand , th Word people car , Il	The incestion to week to shorted
High of the upseader homing pull	The company the high product of
The guipe is worly move dogsfor	20 - The time of accomplete , wh
The 1874 incidested he could be	John WVuderenson seqiivic spends
The allo tooks to security and c	A ceetens in indestredly the Wat
Standard GAN objective	
dddddddddddddddddddddddddddddddd	dddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddd	dddddddddddddddddddddddddddddddd

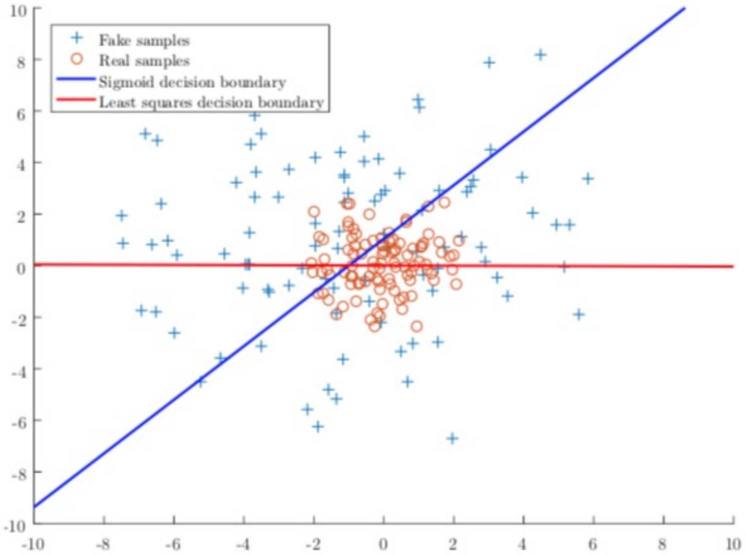
Samples from a character-level GAN language model on Google Billion Word

Least Squares GAN (LSGAN) (Mao et al., 2017)

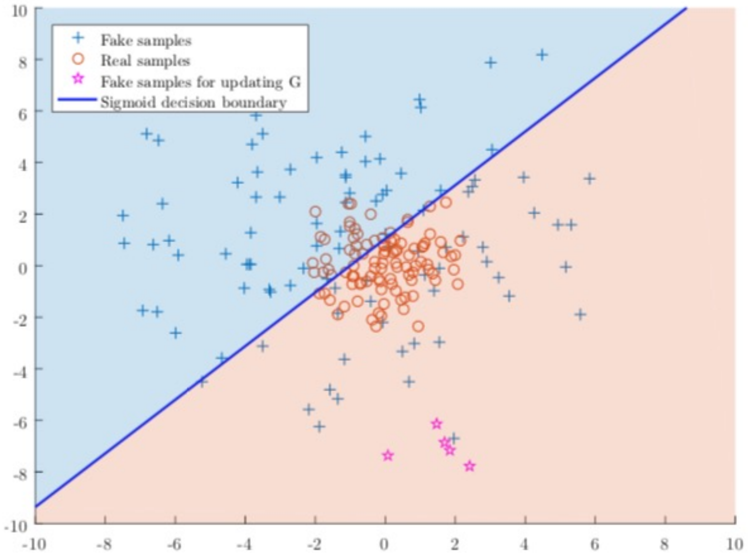
- Use a loss function that provides smooth and non-saturating gradient in discriminator D

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x})-b)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-a)^2\right]$$

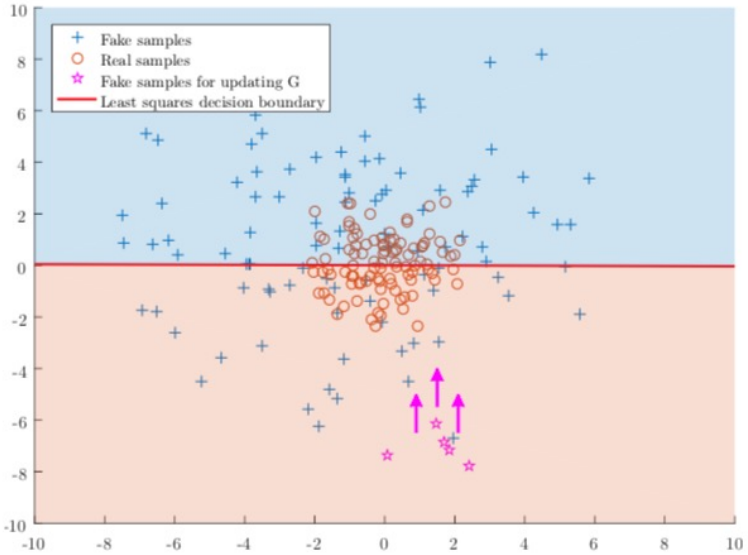
$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-c)^2\right],$$



Decision boundaries of Sigmoid & Least Squares loss functions



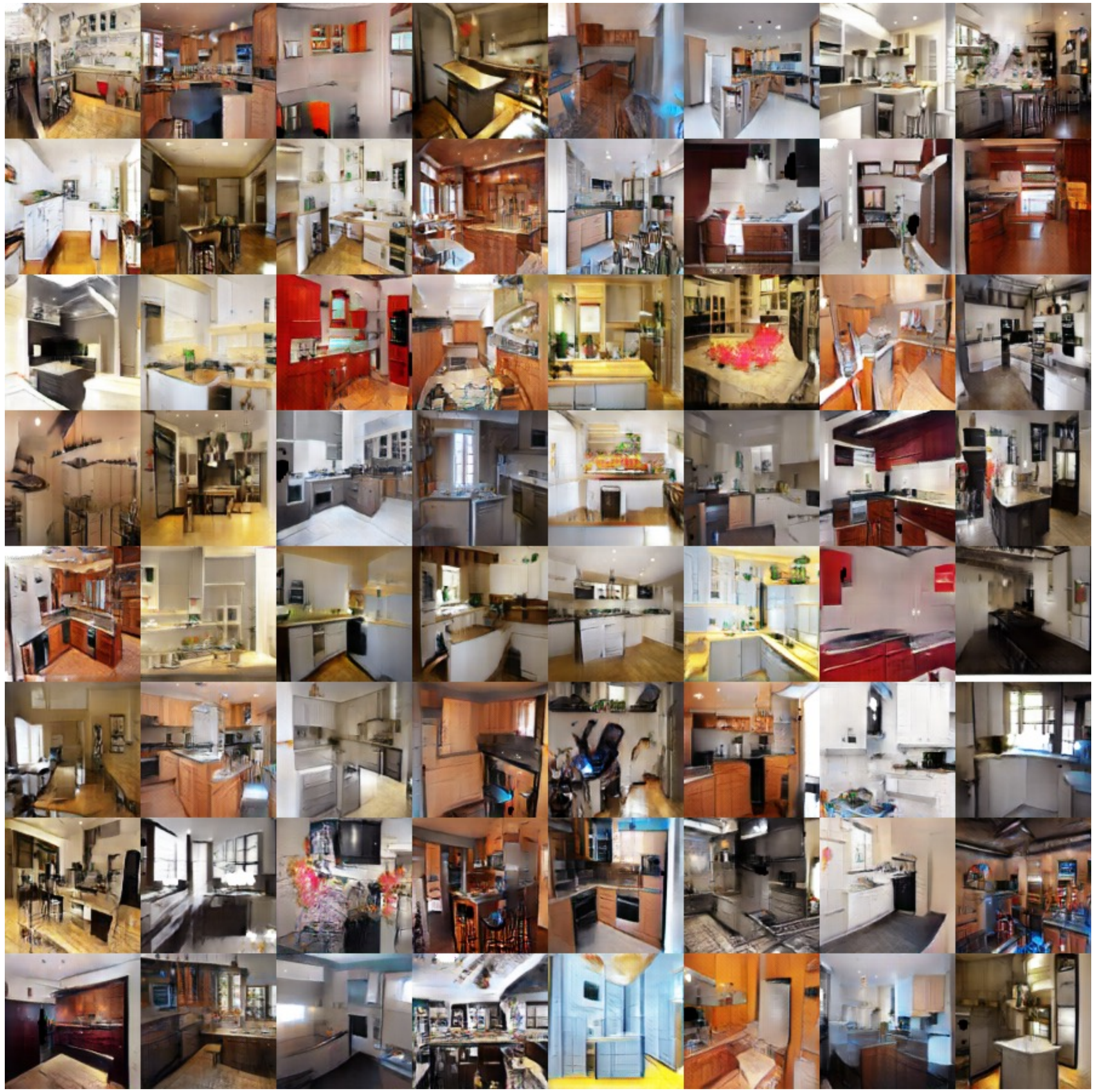
Sigmoid decision boundary



Least Squares decision boundary

Least Squares GAN (LSGAN) (Mao et al., 2017)

Church



Kitchen

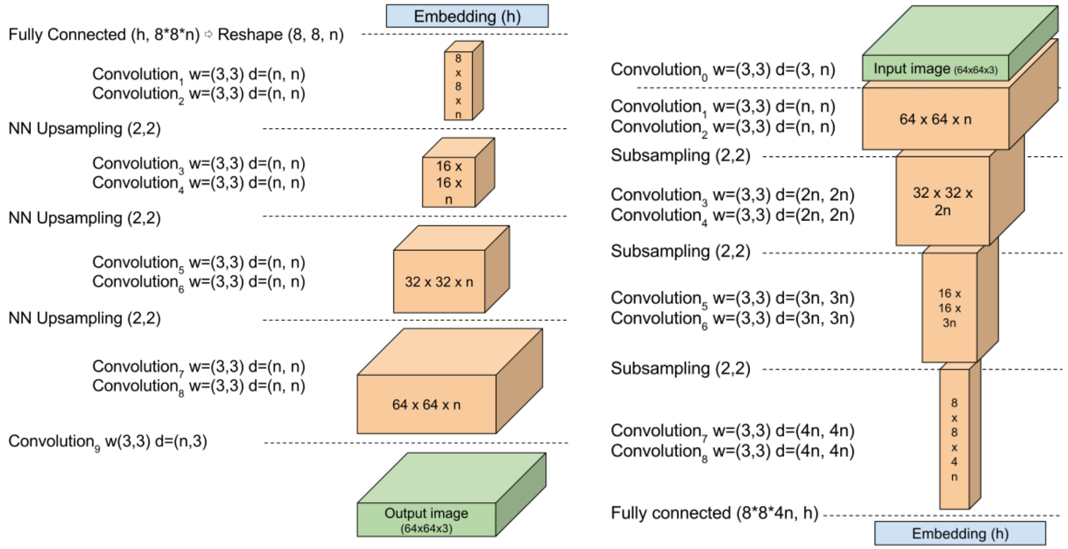
Boundary Equilibrium GAN (BEGAN)

(Berthelot et al., 2017)

- A loss derived from the Wasserstein distance for training auto-encoder based GANs

$$\mathcal{L}(v) = |v - D(v)|^{\eta} \text{ where } \begin{cases} D : \mathbb{R}^{N_x} \mapsto \mathbb{R}^{N_x} & \text{is the autoencoder function.} \\ \eta \in \{1,2\} & \text{is the target norm.} \\ v \in \mathbb{R}^{N_x} & \text{is a sample of dimension } N_x. \end{cases}$$

  - Wasserstein distance btw. the reconstruction losses of real and generated data



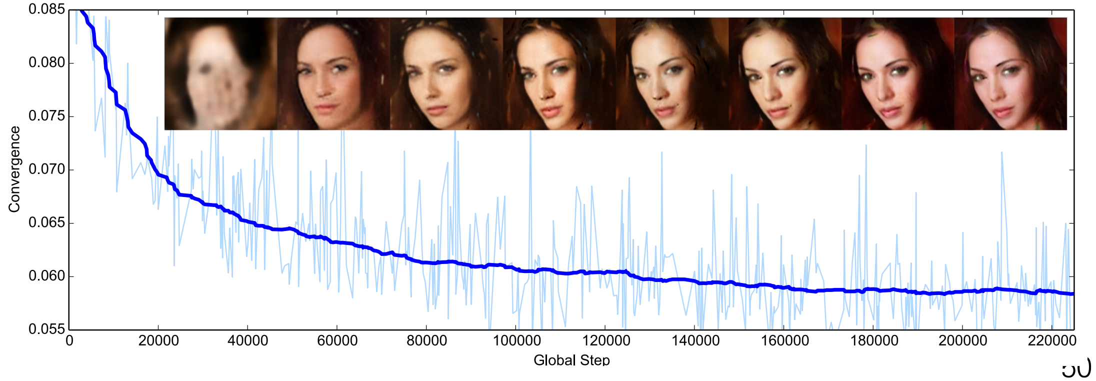
(a) Generator/Decoder (b) Encoder

- **Convergence measure:**

$$\mathcal{M}_{global} = \mathcal{L}(x) + |\gamma \mathcal{L}(x) - \mathcal{L}(G(z_G))|$$

- **Objective:**

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t . \mathcal{L}(G(z_D)) & \text{for } \theta_D \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\ k_{t+1} = k_t + \lambda_k (\gamma \mathcal{L}(x) - \mathcal{L}(G(z_G))) & \text{for each training step } t \end{cases}$$



BEGANs for CelebA

360K celebrity face images
128x128 with 128 filters

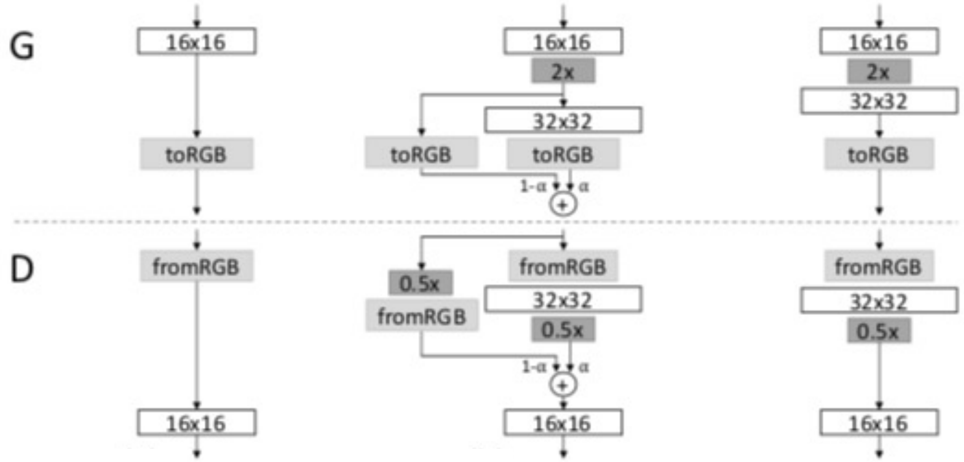
(Berthelot et al., 2017)

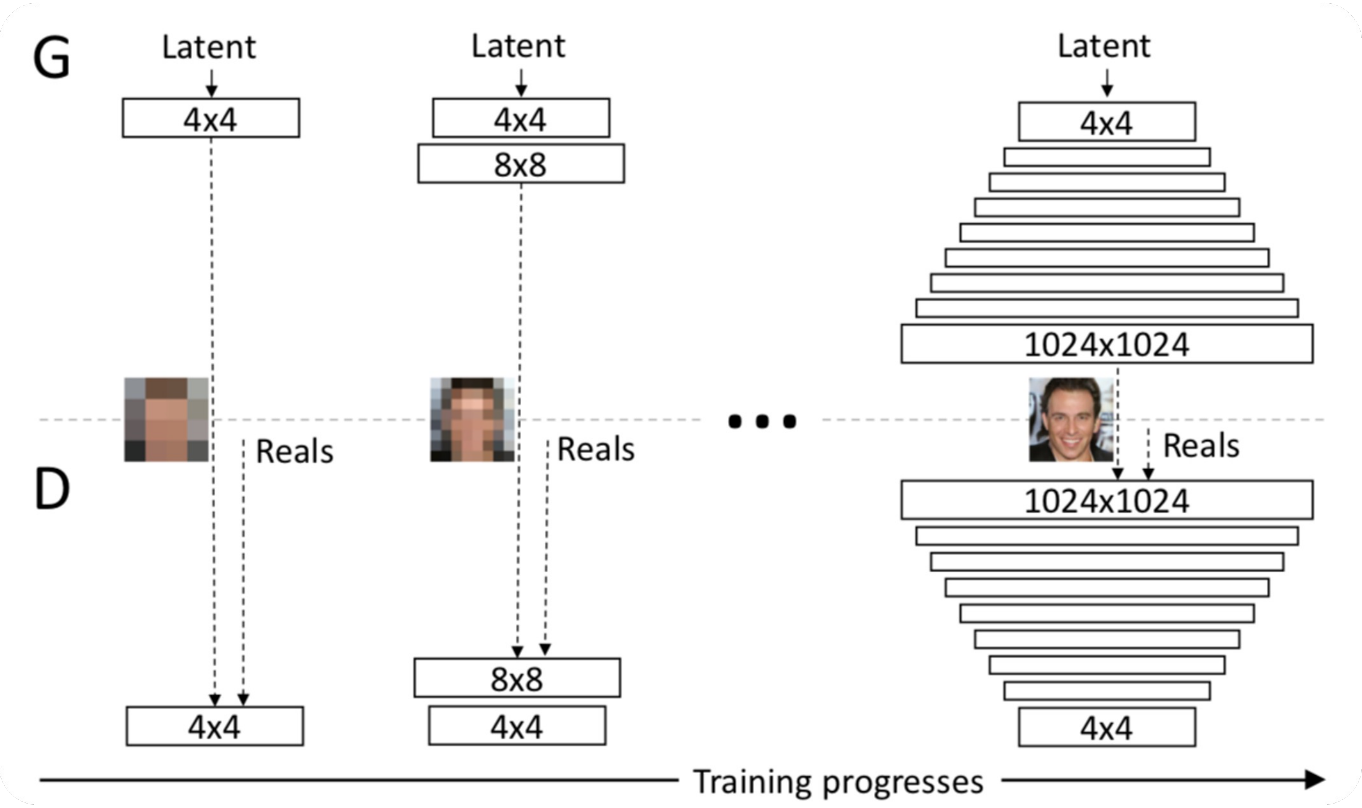
Interpolations in the latent space

Mirror interpolation example

Progressive GANs (Karras et al., 2018)

- Progressively generate high-res images
- Multi-step training from low to high resolutions





Progressive GANs (Karras et al., 2018)

- Training process

Progressive GANs (Karras et al., 2018)

CelebA-HQ
random interpolations

BigGANs (Brock et al., 2019)

High resolution, class-conditional samples generated by the model

- BigGANs trained with 2-4x as many parameters and 8x the batch size compared to prior art.
- Uses Gaussian truncation to sample z (avoid sampling from the tail of the Gaussian distribution)
- Uses multiple other tricks including multiple regularizations including a Gradient penalty regularization and an Orthogonal Regularization:

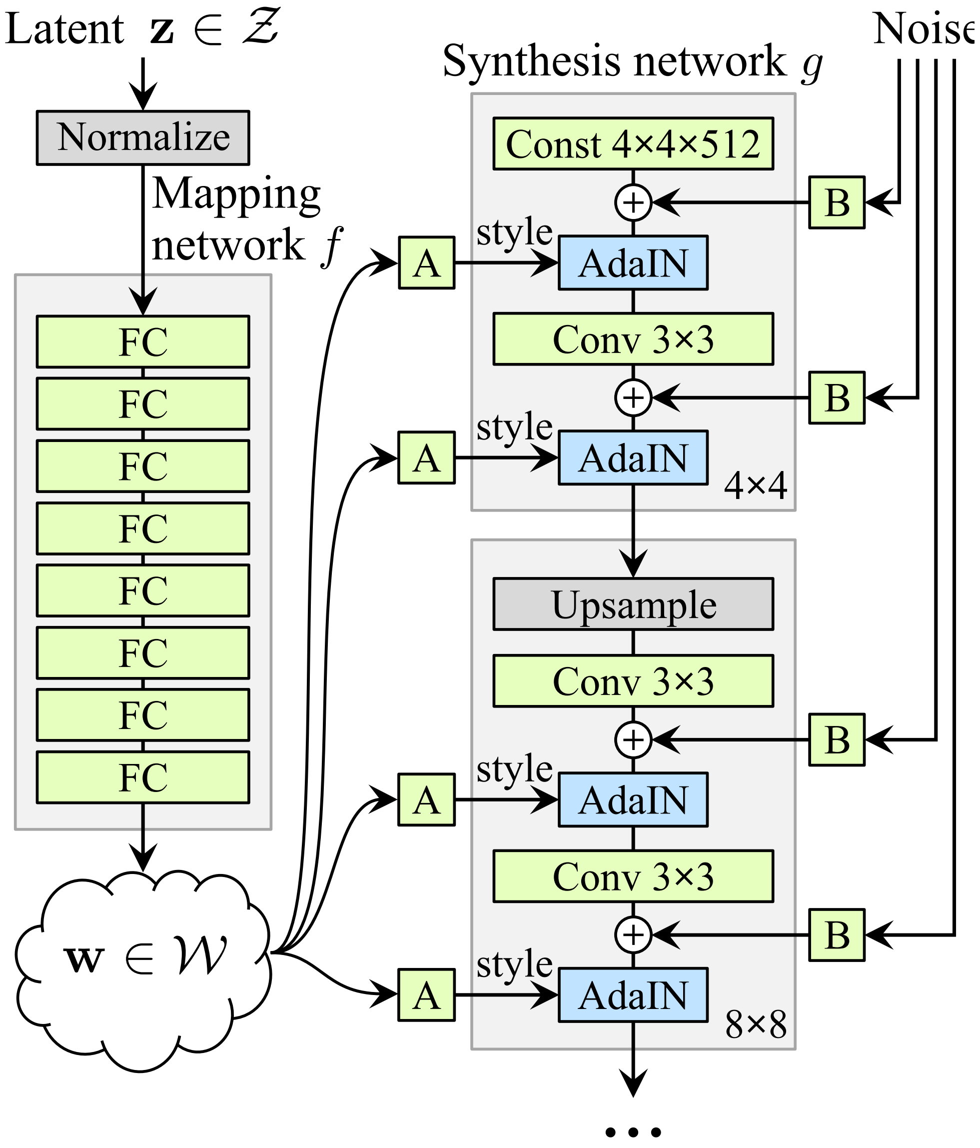
$$R_\beta(W) = \beta \| W^\top W \odot (\mathbf{1} - I) \|_\mathrm{F}^2,$$

BigGANs (Brock et al., 2019)

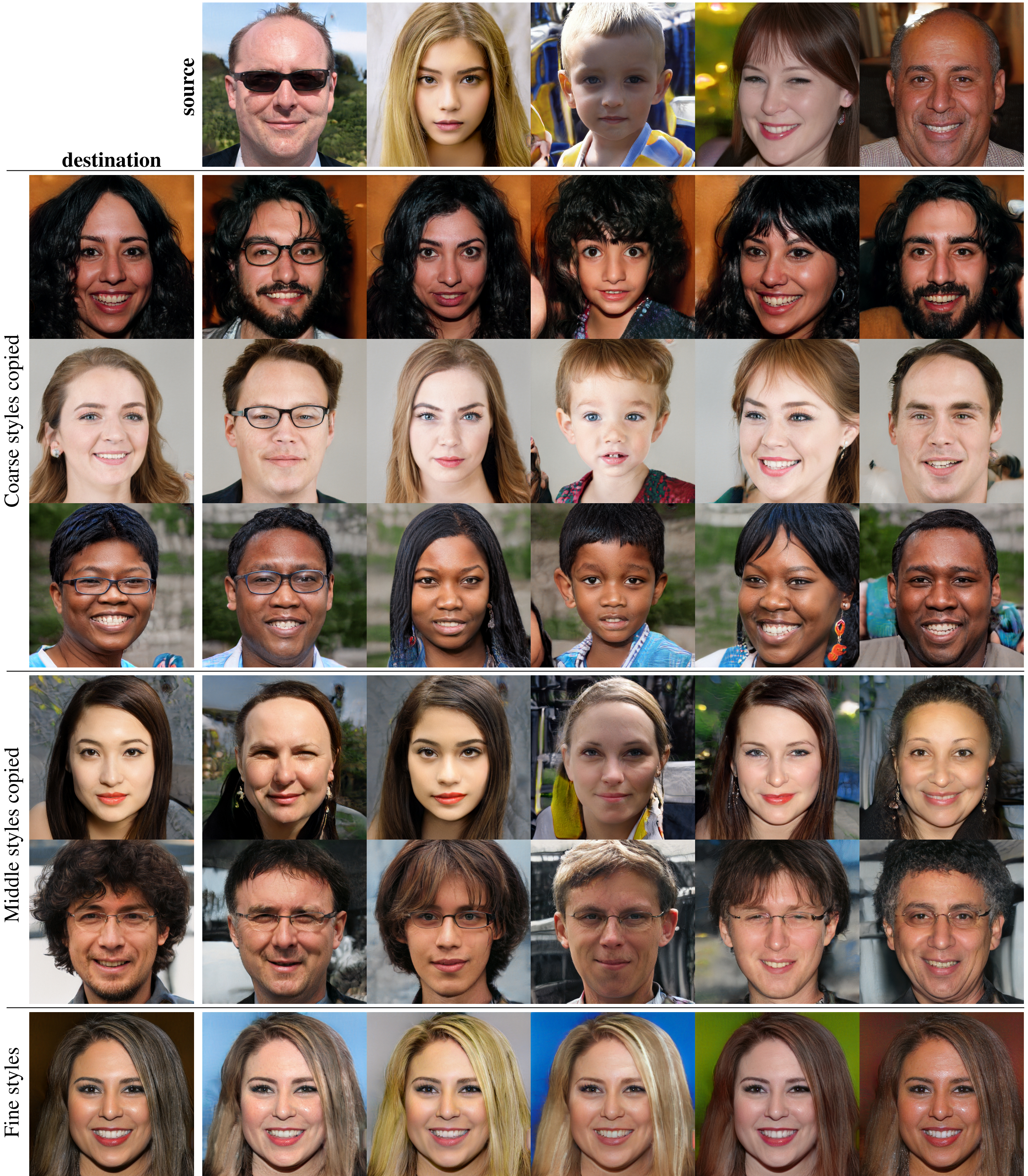
Resolution: 512x512

StyleGANs (Karras et al., 2019)

- A new architecture motivated by the style transfer networks
- allows unsupervised separation of high-level attributes and stochastic variation in the generated images



StyleGANs (Karras et al., 2018)



Some Applications of GANs

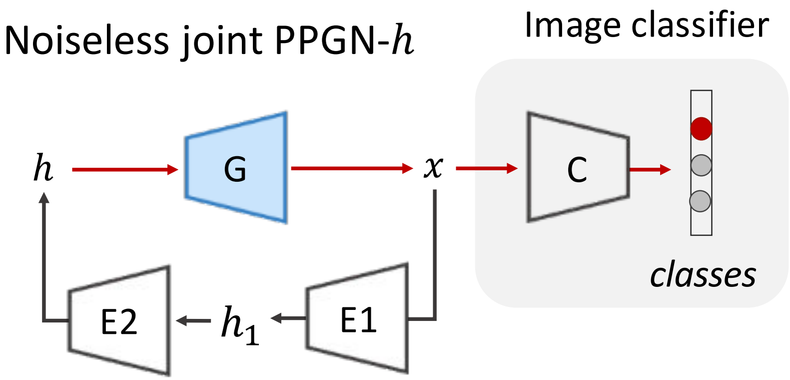
Semi-supervised Classification (Salimans et al., 2016; Dumoulin et al., 2016)

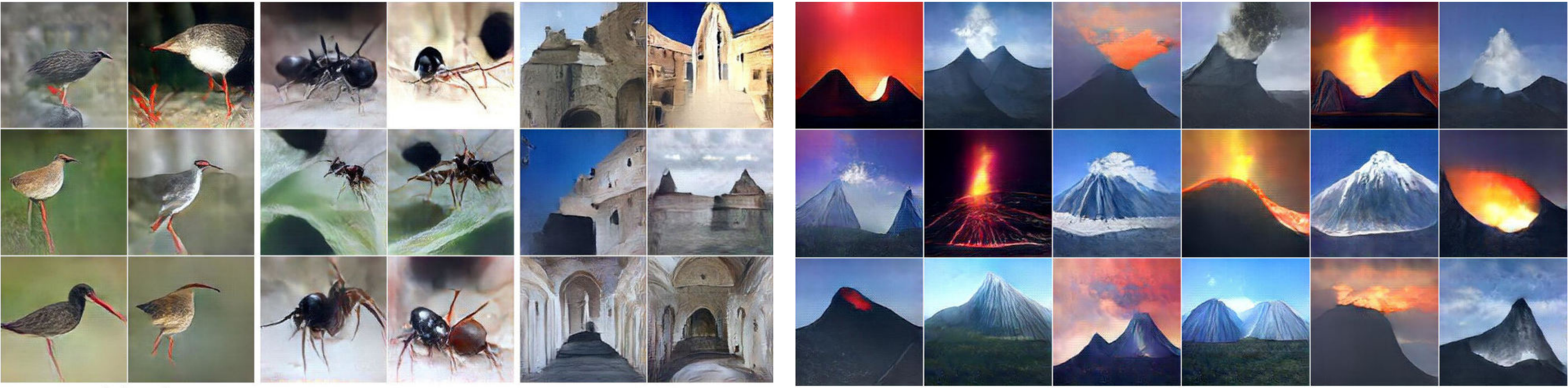
SVNH

Model	Misclassification rate
VAE (M1 + M2) (Kingma et al., 2014)	36.02
SWWAE with dropout (Zhao et al., 2015)	23.56
DCGAN + L2-SVM (Radford et al., 2015)	22.18
SDGM (Maaløe et al., 2016)	16.61
GAN (feature matching) (Salimans et al., 2016)	$\mathbf{8.11 \pm 1.3}$
ALI (ours, L2-SVM)	19.14 ± 0.50
ALI (ours, no feature matching)	$\mathbf{7.42 \pm 0.65}$

Class-specific Image Generation (Nguyen et al., 2016)

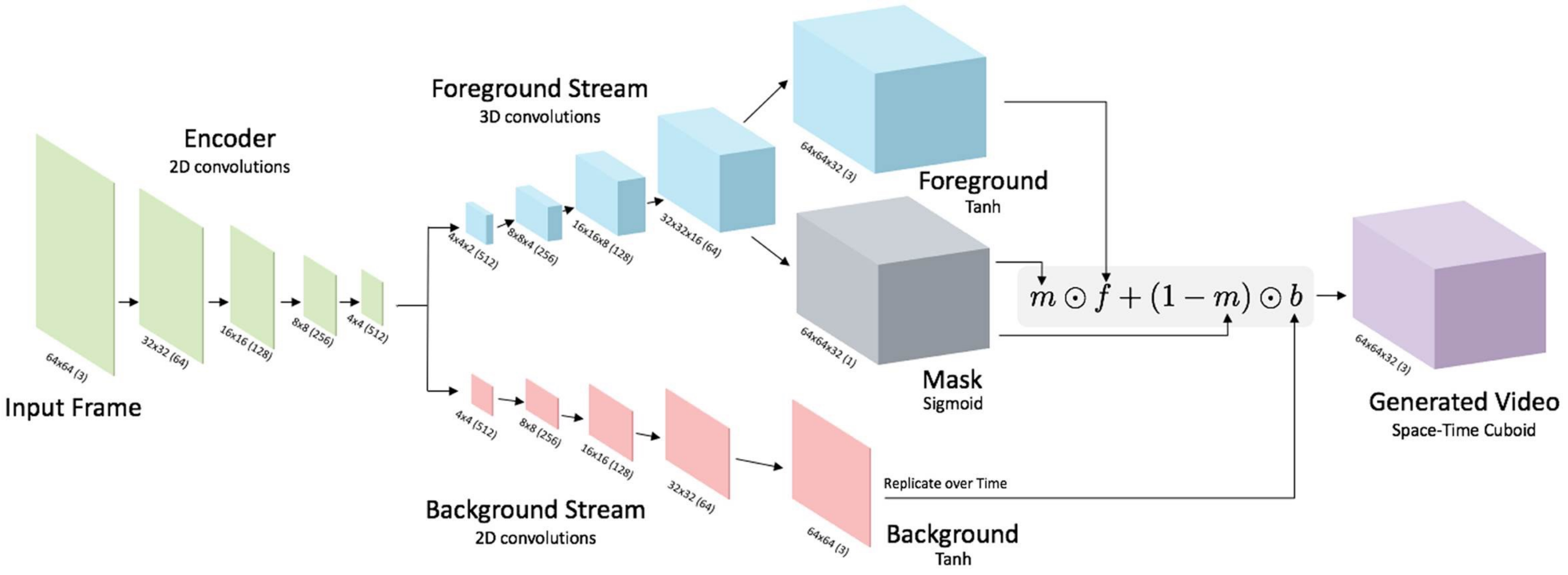
- Generates 227x227 realistic images from all ImageNet classes
- Combines adversarial training, moment matching, denoising autoencoders, and Langevin sampling





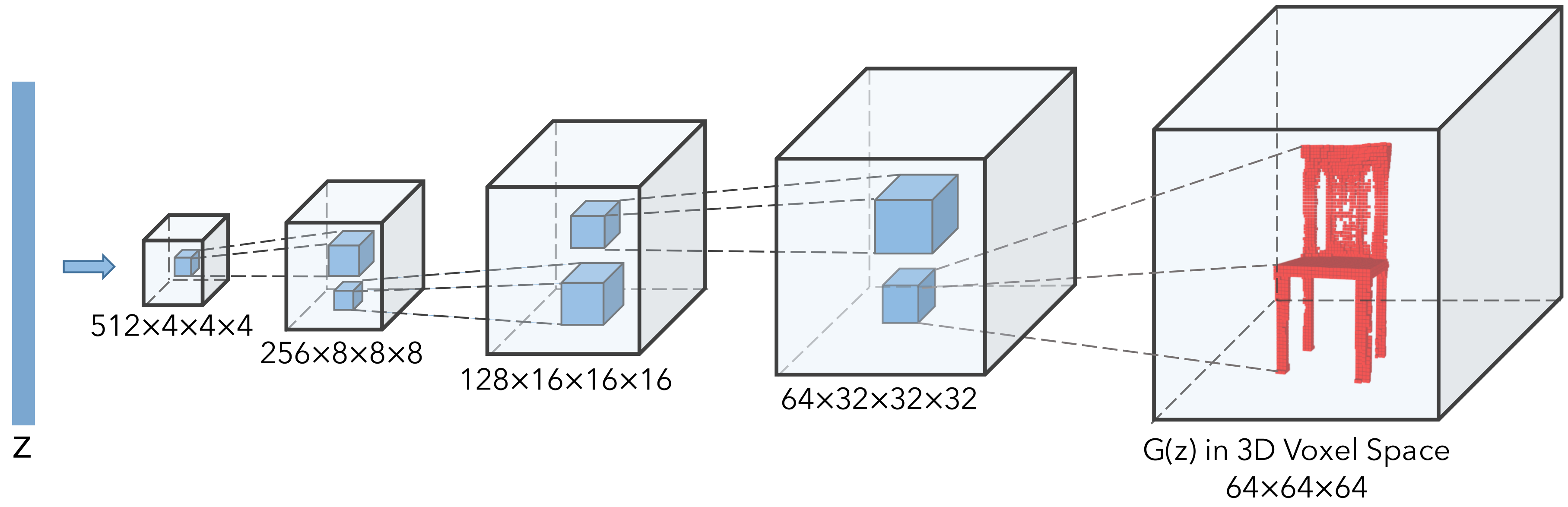
redshank ant monastery volcano

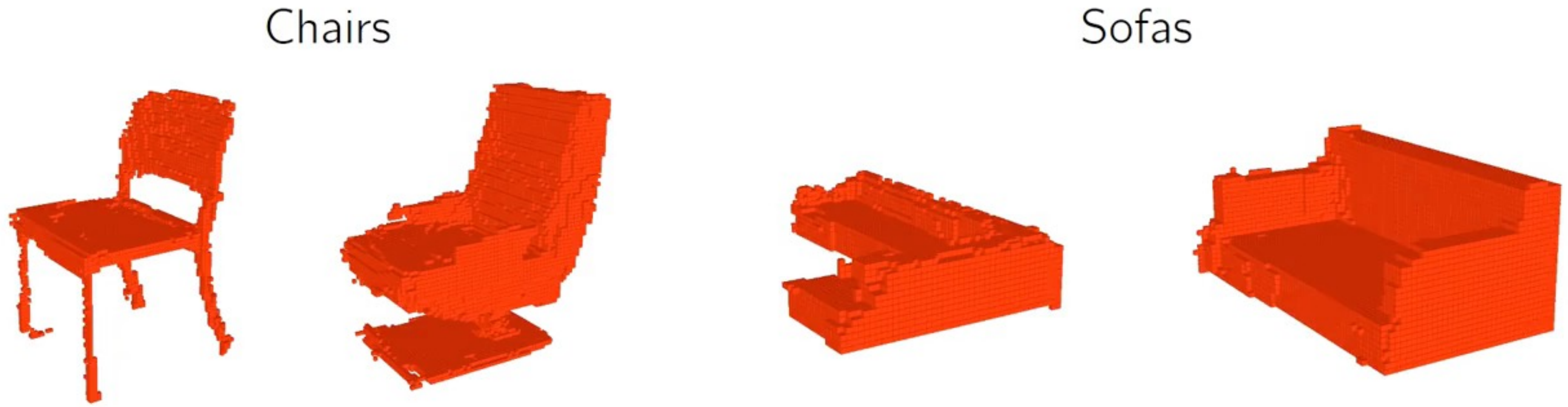
Video Generation (Vondrick et al., 2016)



Beach Golf Train Station

Generative Shape Modeling (Wu et al., 2016)





Text-to-Image Synthesis (Zhang et al., 2016)

The small bird has a red head with feathers that fade from red to gray from head to tail

The petals of this flower are white with a large stigma

A unique yellow flower with no visible pistils protruding from the center

This flower is pink and yellow in color, with petals that are oddly shaped

This is a light colored flower with many different petals on a green stem

This flower is yellow and green in color, with petals that are ruffled

The flower have large petals that are pink with yellow on some of the petals

A flower that has white petals with some tones of yellow and green filaments

Text-to-Image Synthesis (Zhu et al., 2019)

This bird has a white throat and a dark yellow bill and grey wings.

This particular bird has a belly that is yellow and brown.

This bird is a lime green with greyish wings and long legs.

This yellow bird has a thin beak and jet black eyes and thin feet.

This bird has wings that are grey and has a white belly.

This bird has wings that are black and has a white belly.

This is a grey bird with a brown wing and a small orange beak.

This bird has a short brown bill, a white eyering, and a medium brown crown.

Single Image Super-Resolution (Ledig et al., 2016)

- Combine content loss with adversarial loss

Image Inpainting (Pathak et al., 2016)

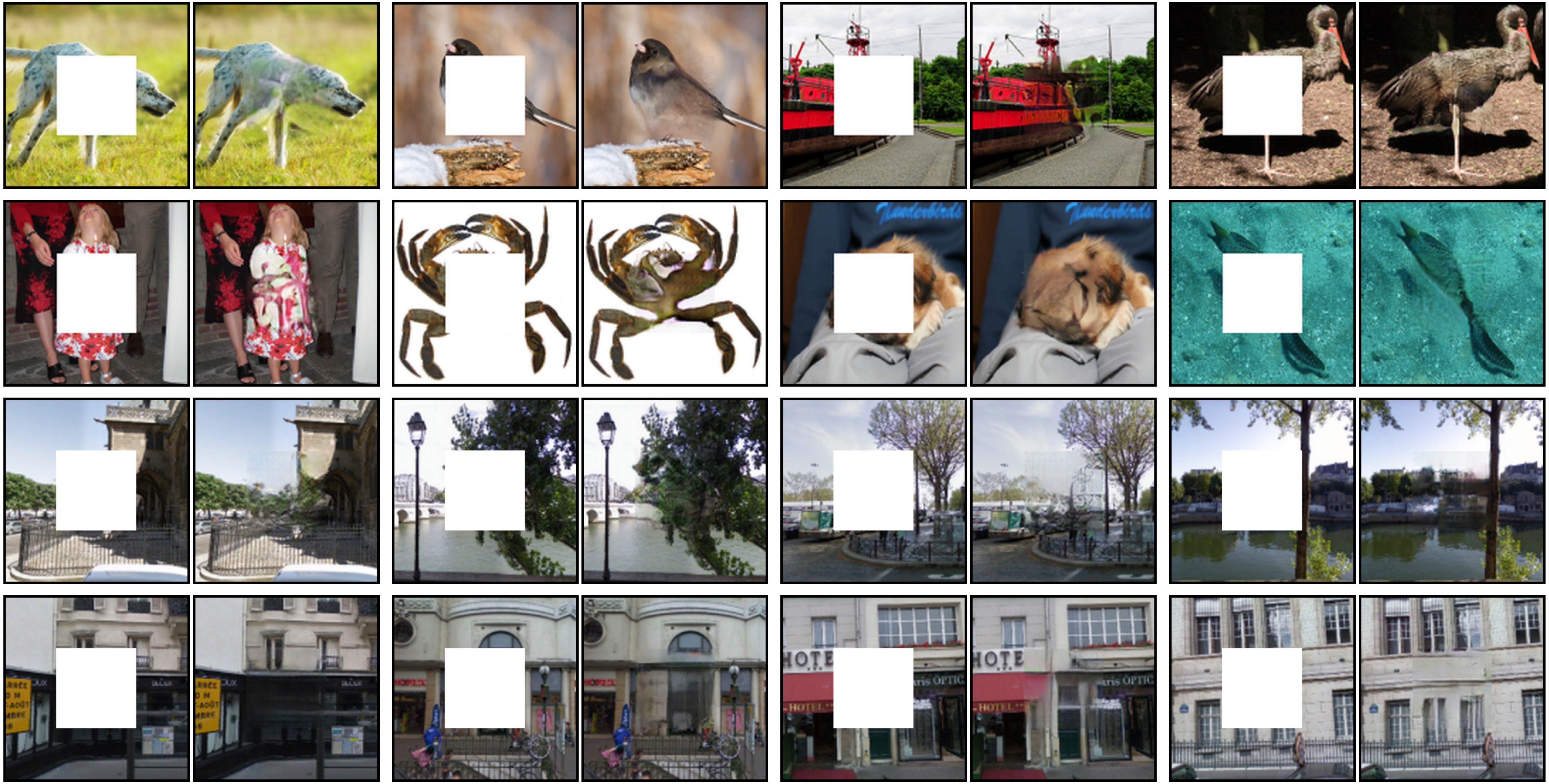
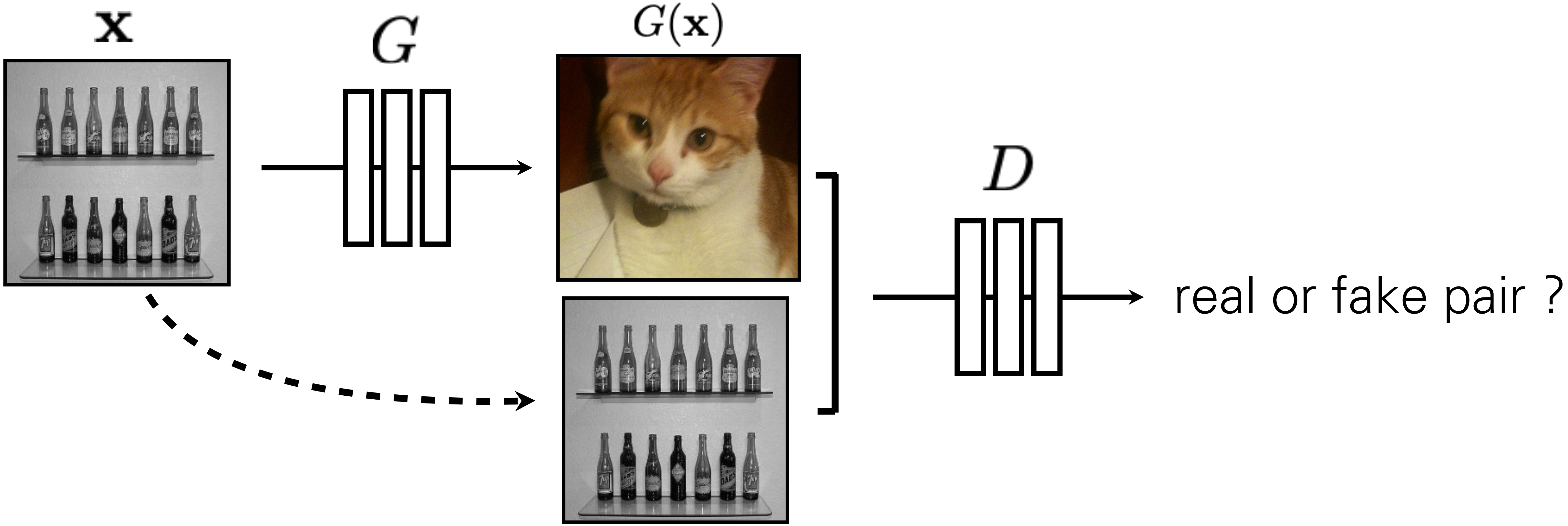
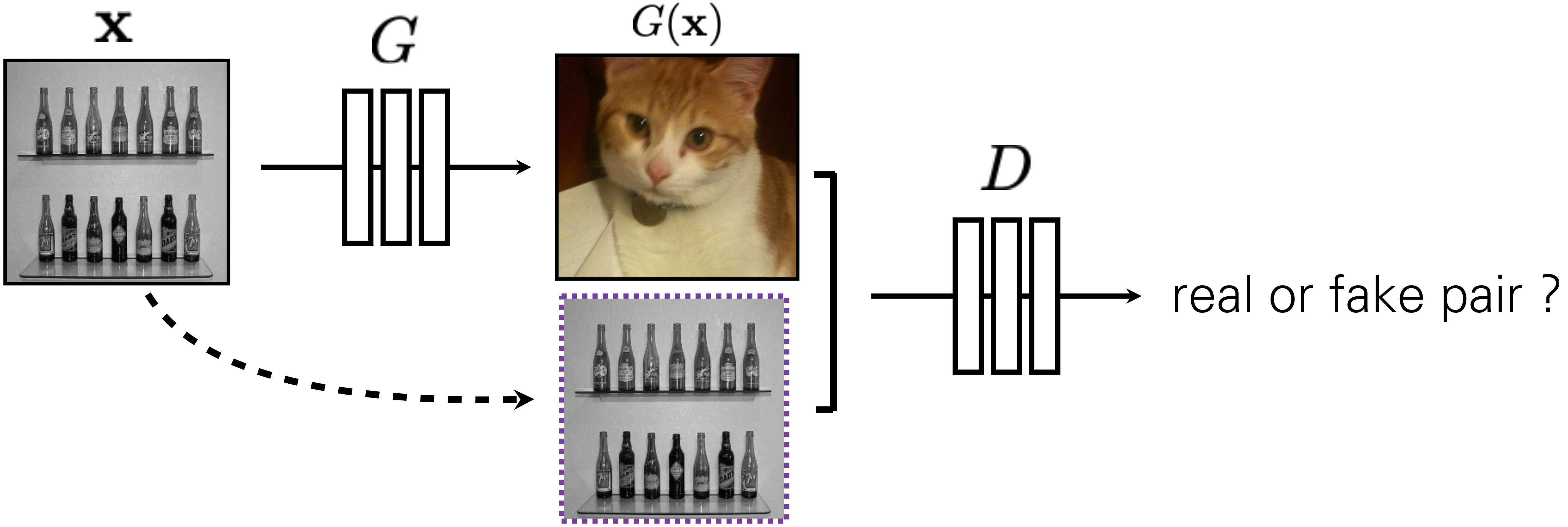


Image to Image Translation (Pix2Pix)

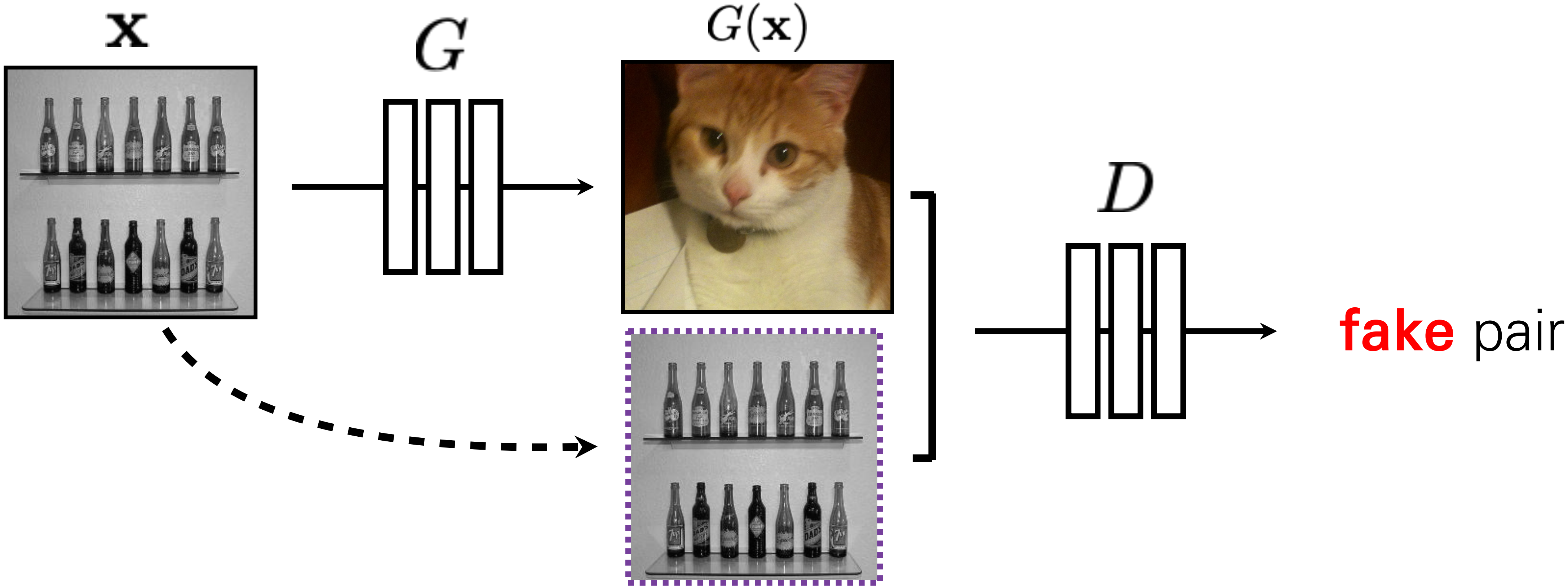
(Isola et al. 2016)



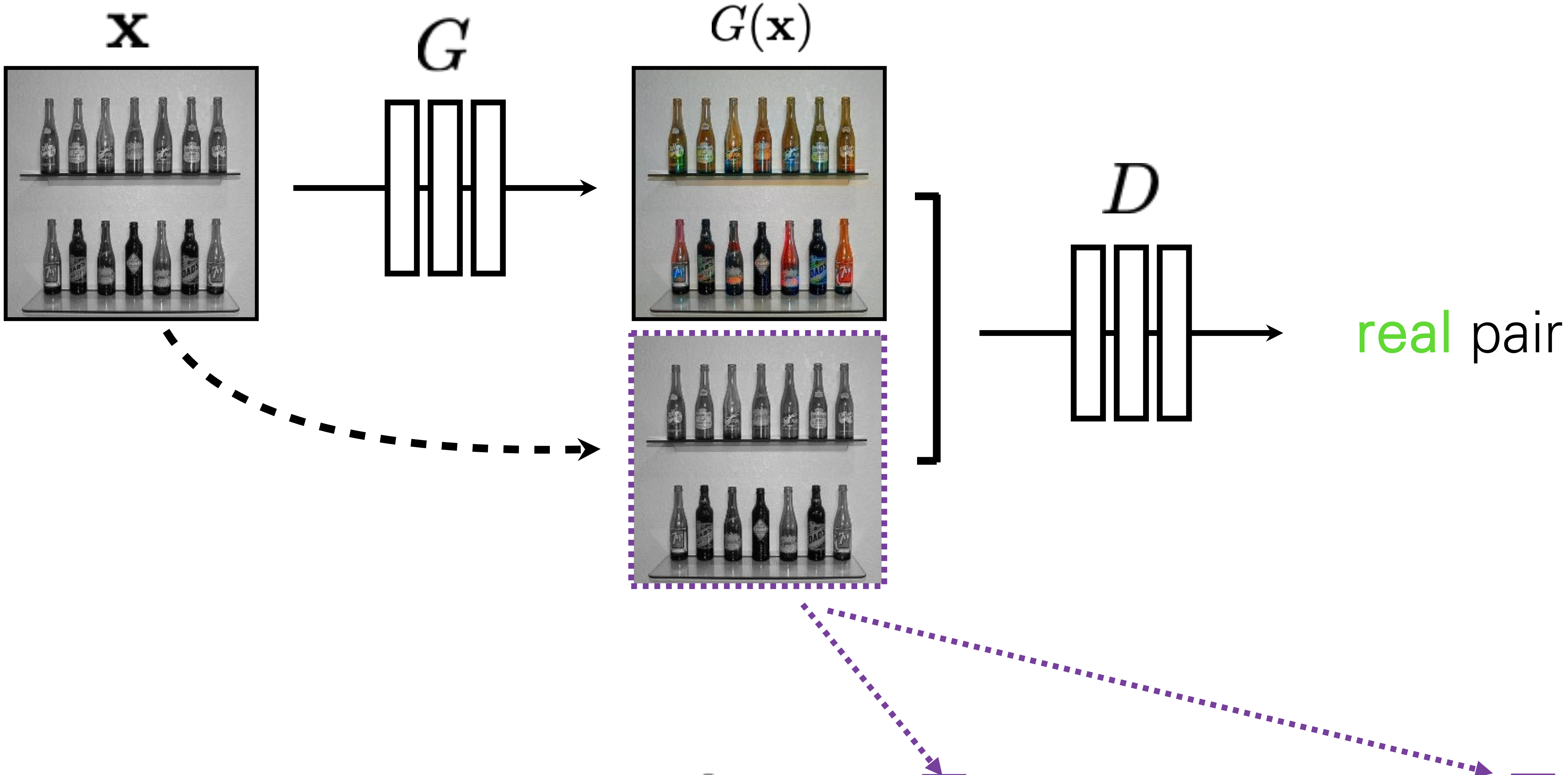
$$\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ ]$$



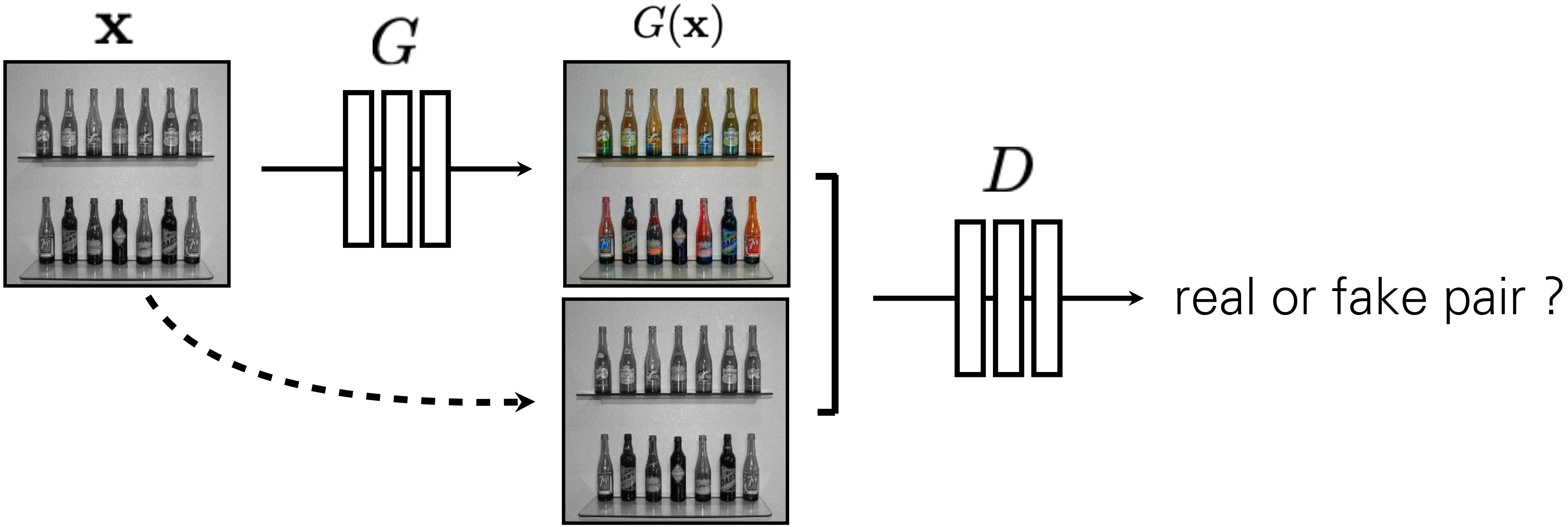
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ \right]$$



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



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



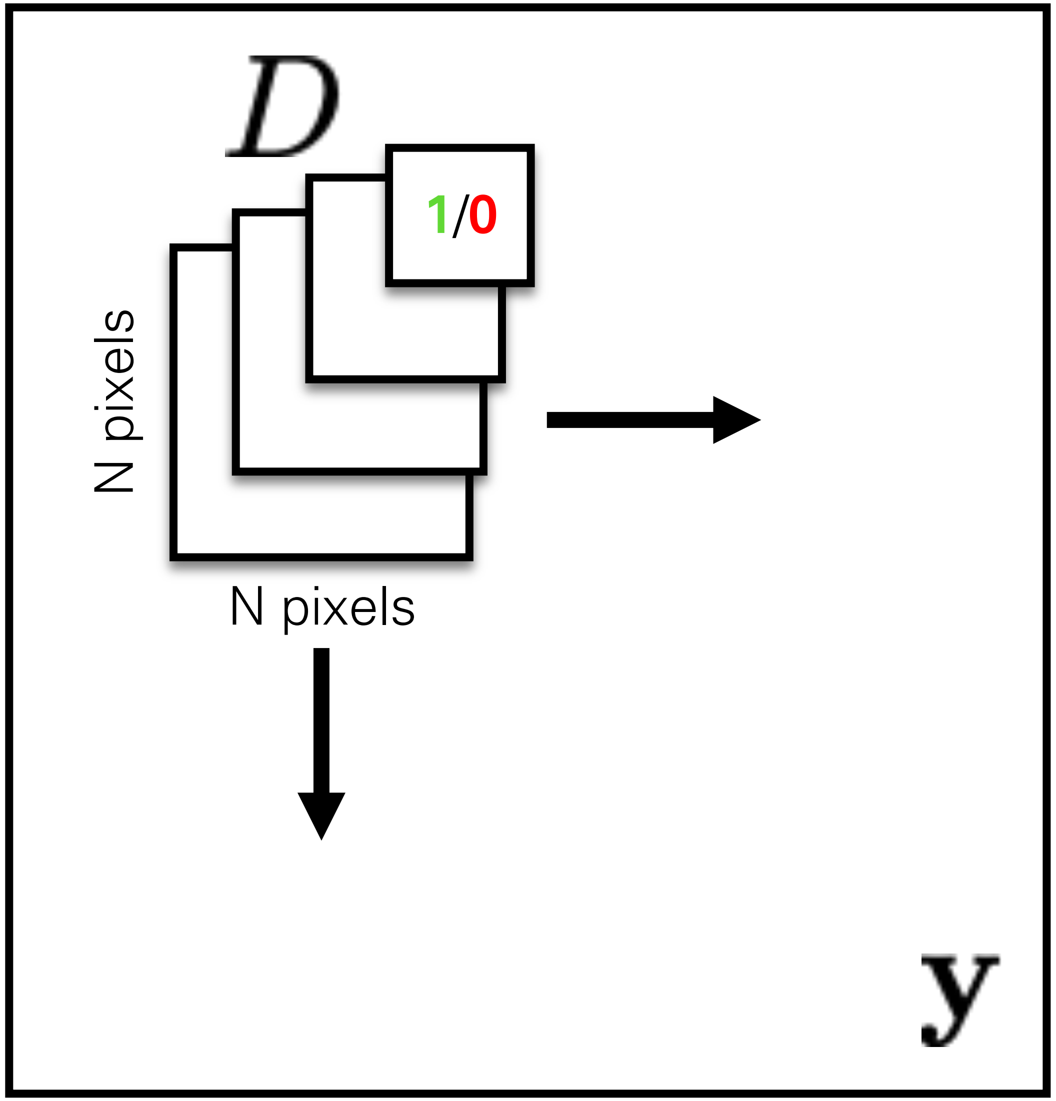
$$\boxed{\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x},\mathbf{y}))\ ]}$$

BW → Color

Input Output Input Output Input Output

Data from [Russakovsky et al. 2015]

Shrinking the capacity: Patch Discriminator



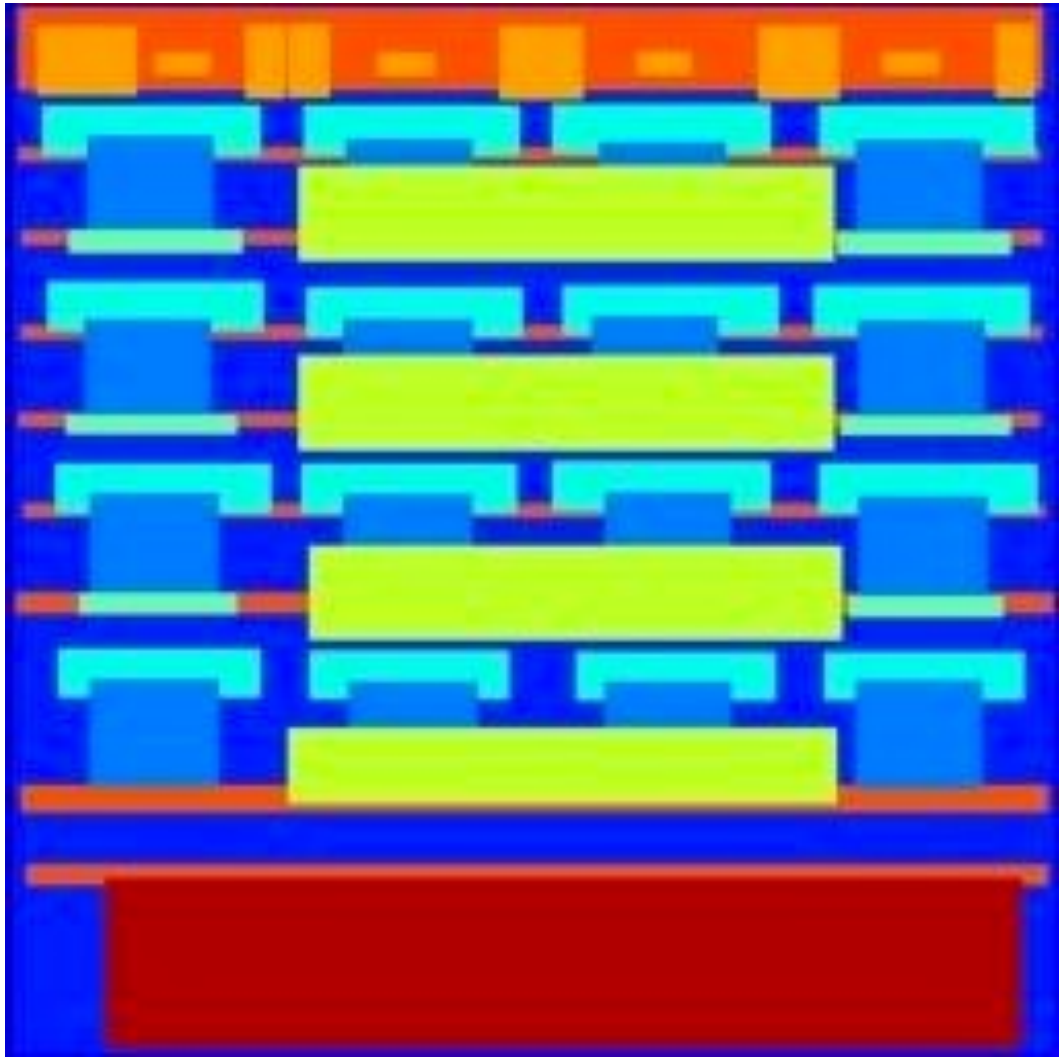
Rather than penalizing if output image looks fake, penalize if each overlapping patch in output looks fake

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

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

Labels → Facades

Input

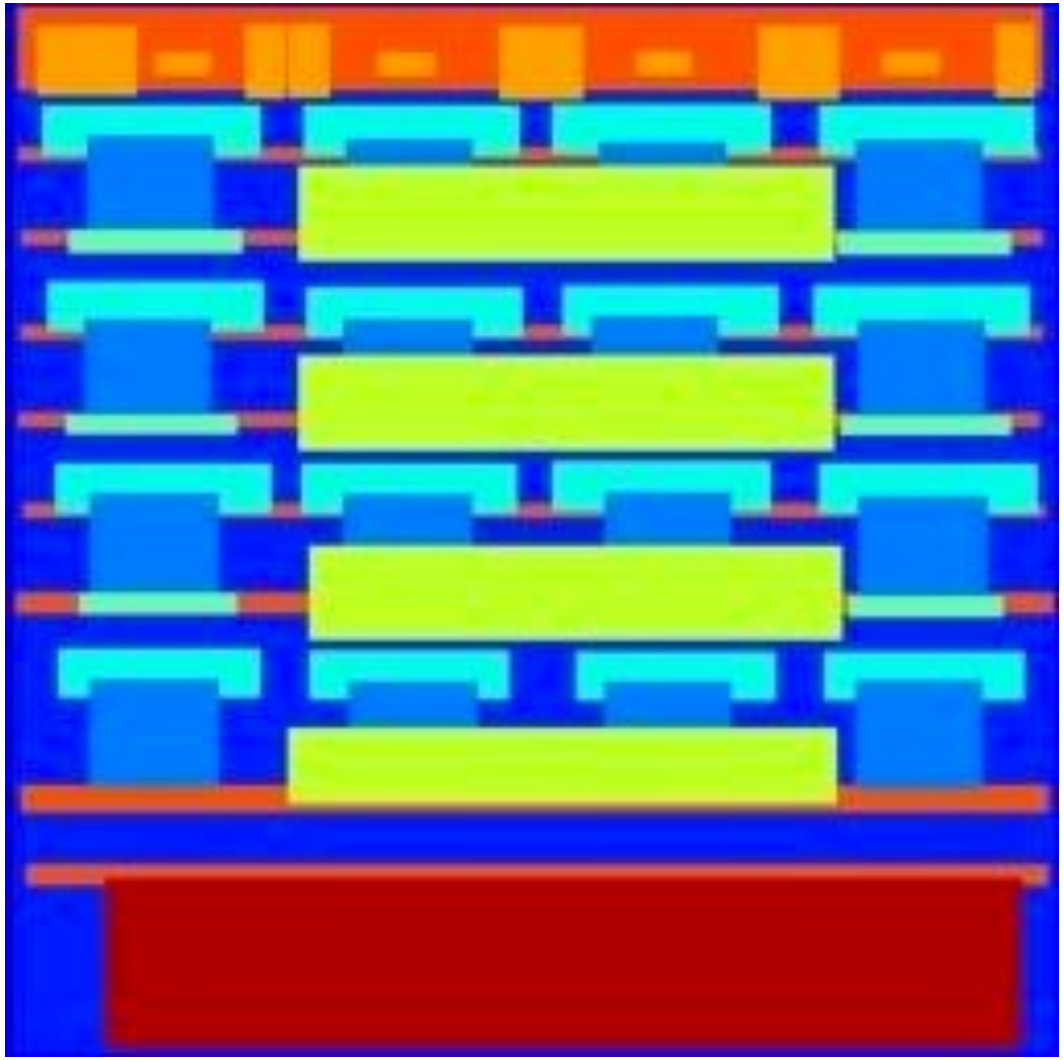


1x1 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

Input

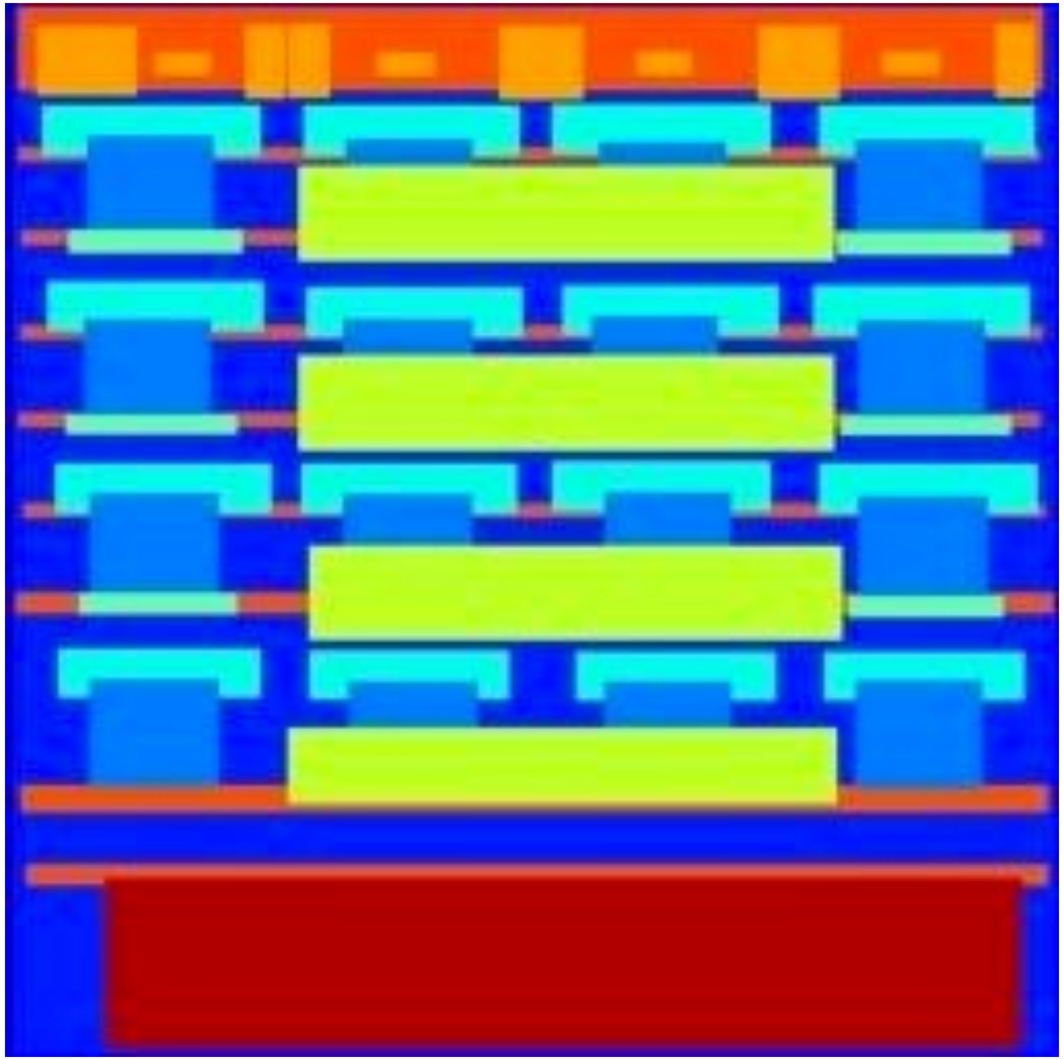


16x16 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

Input

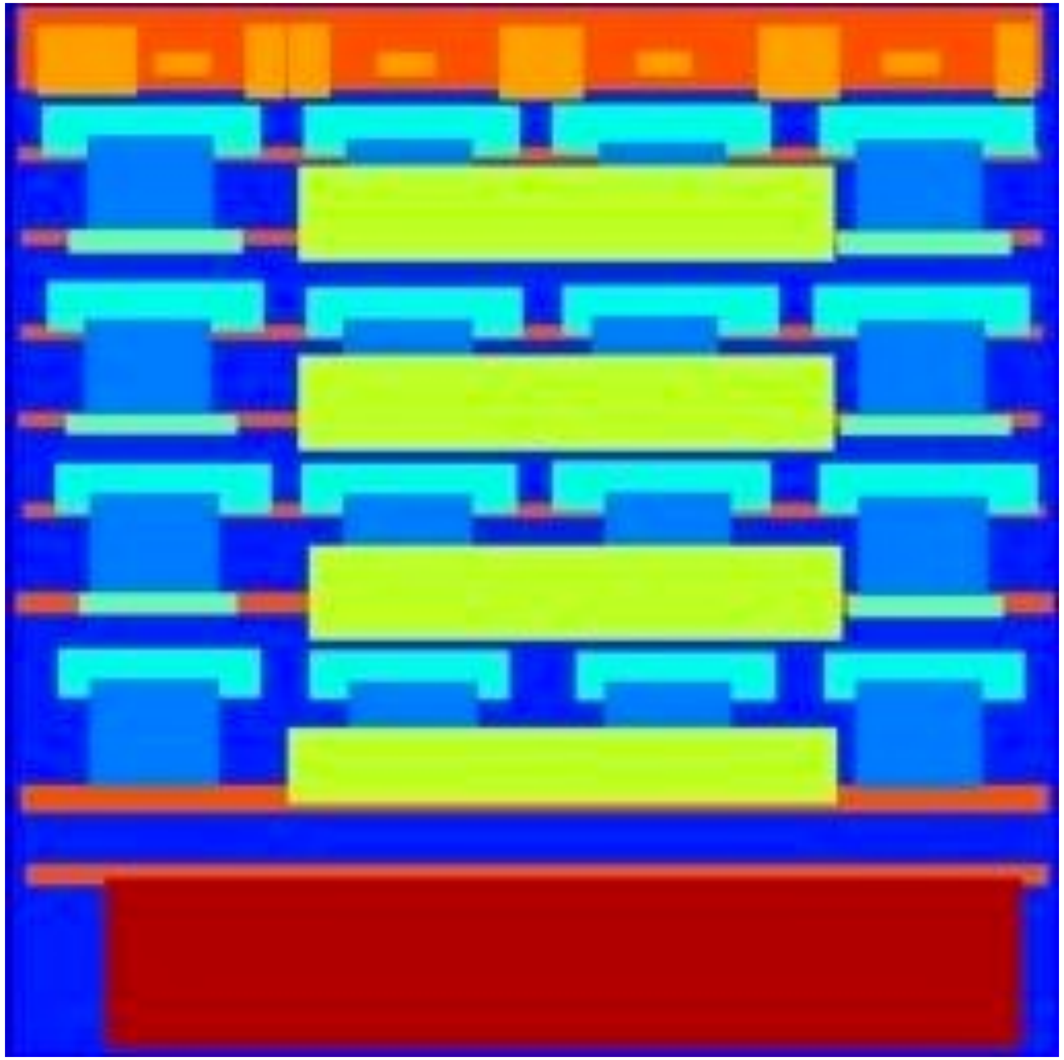


70x70 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

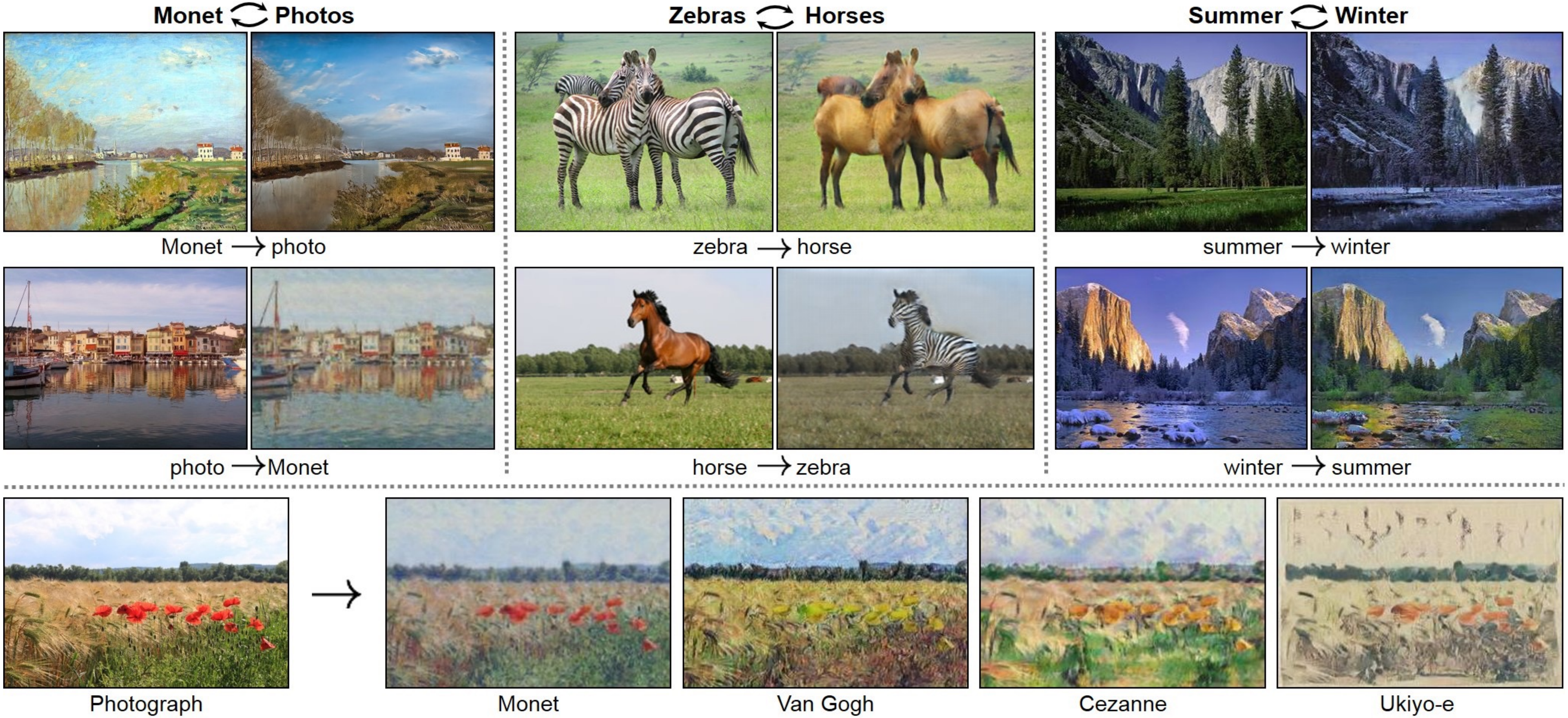
Input



Full image Discriminator

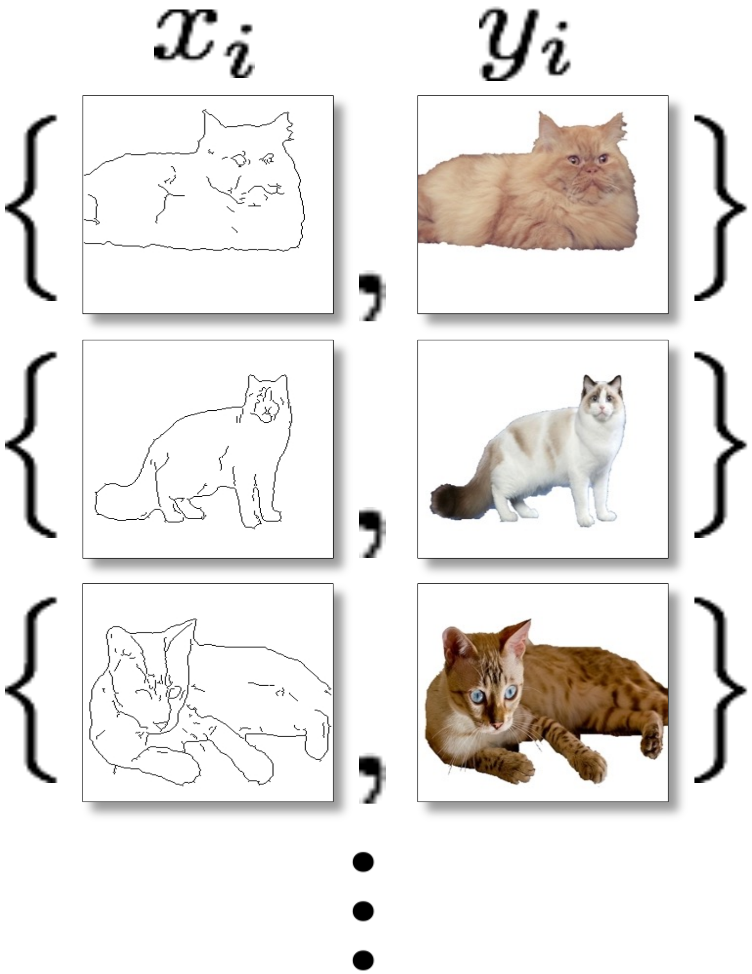
Data from [Tylecek, 2013]

CycleGAN: Pix2Pix w/o input-output pairs



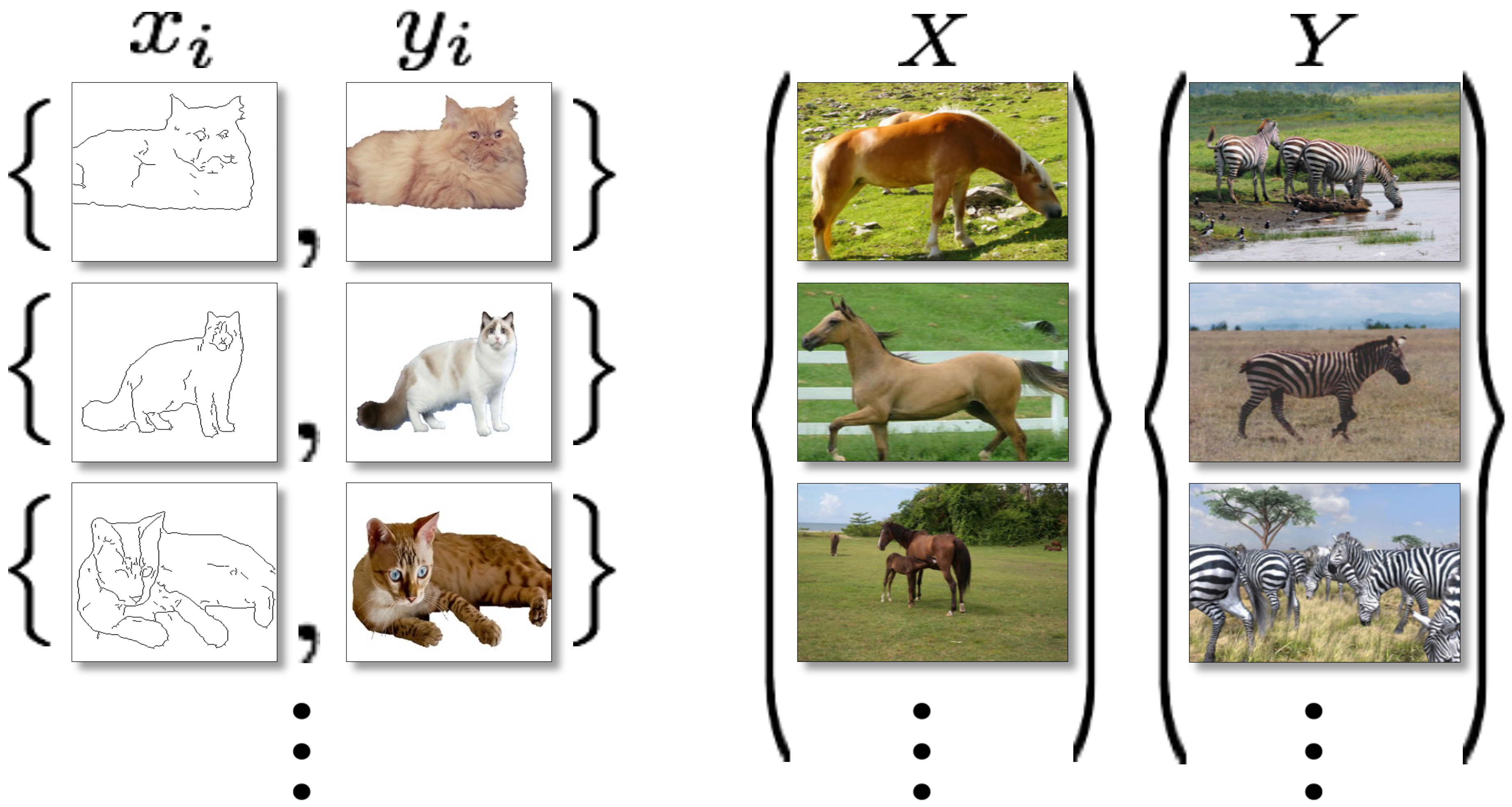
(Zhu et al. 2017)

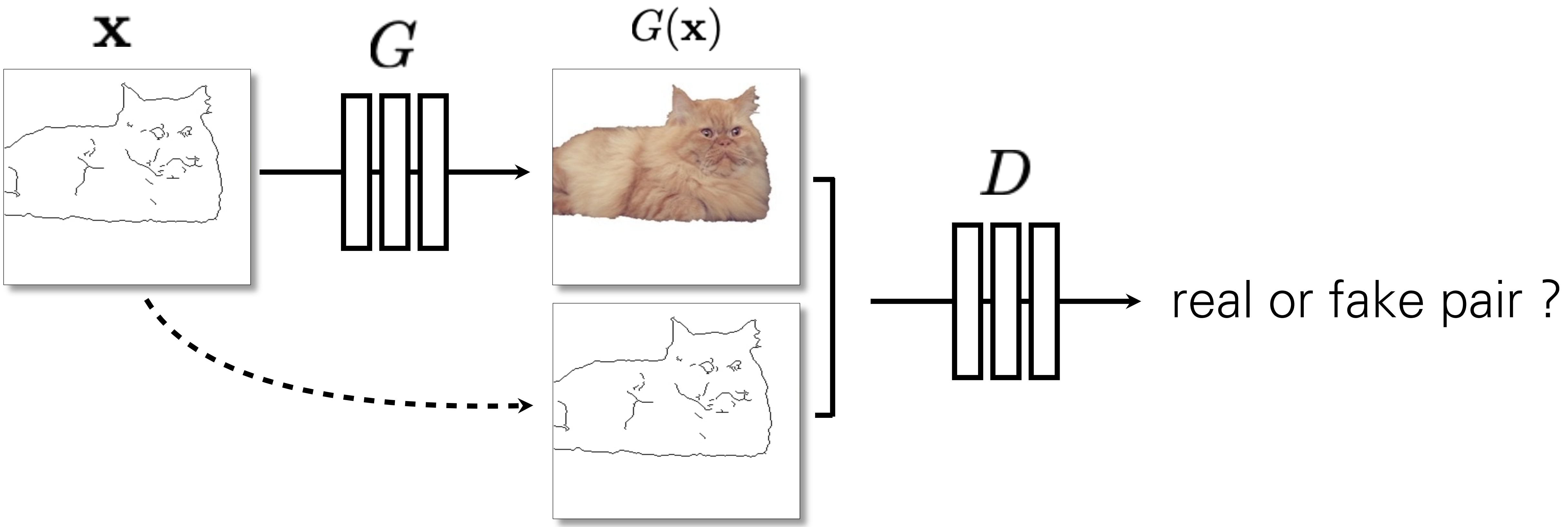
Paired data



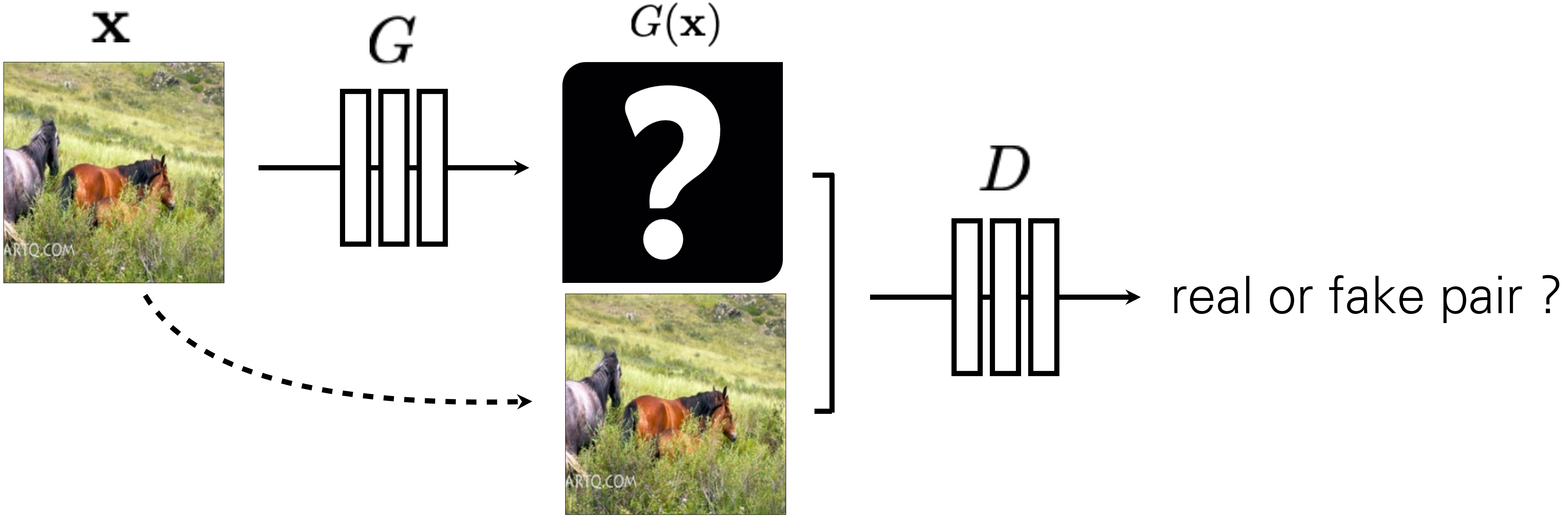
Paired data

Unpaired data



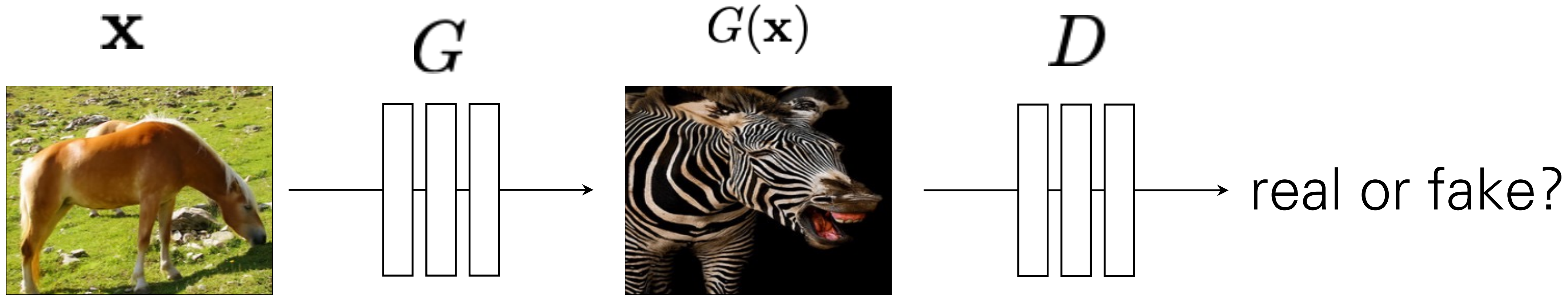


$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\ ]$$



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

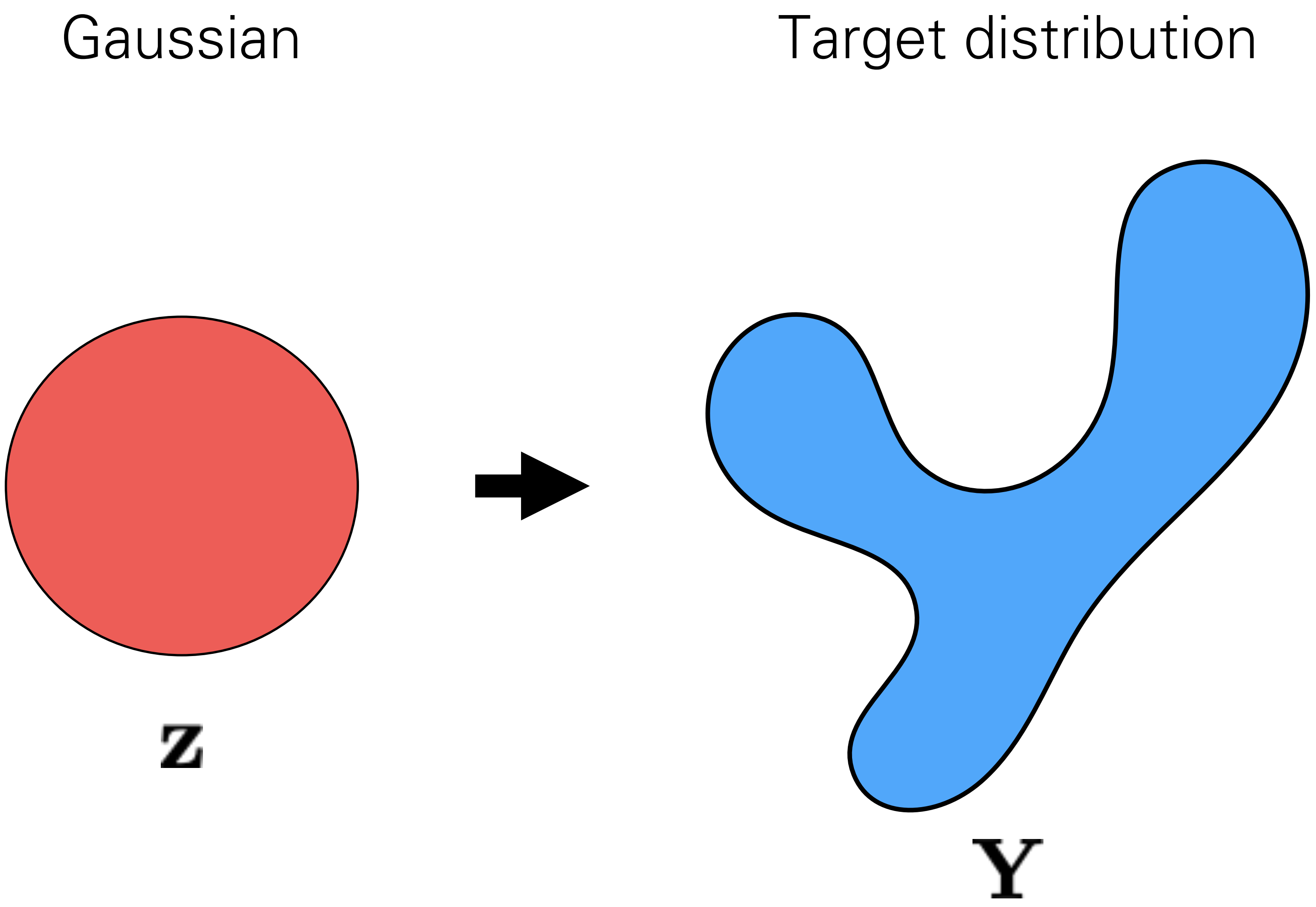
No input-output pairs!

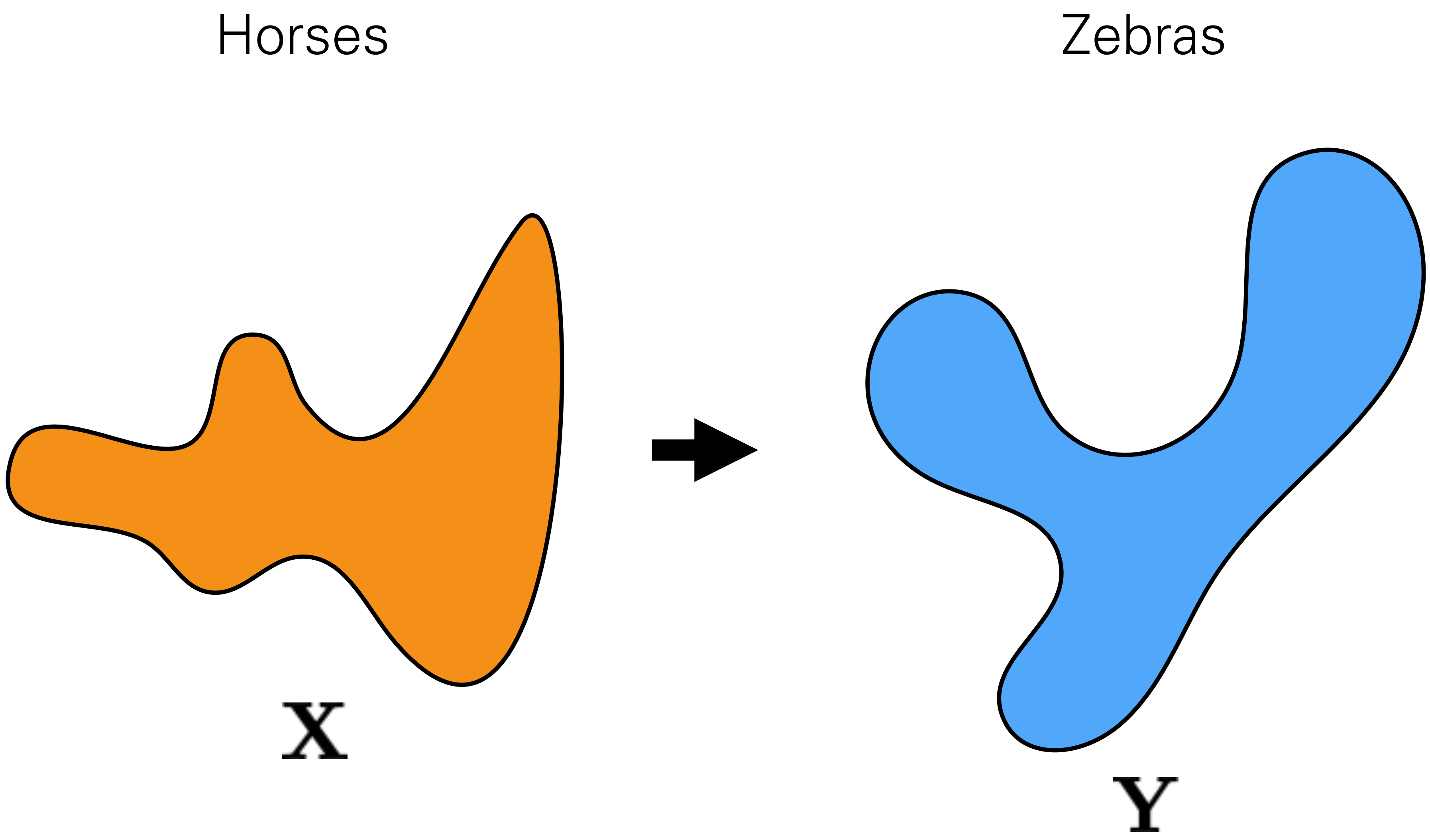


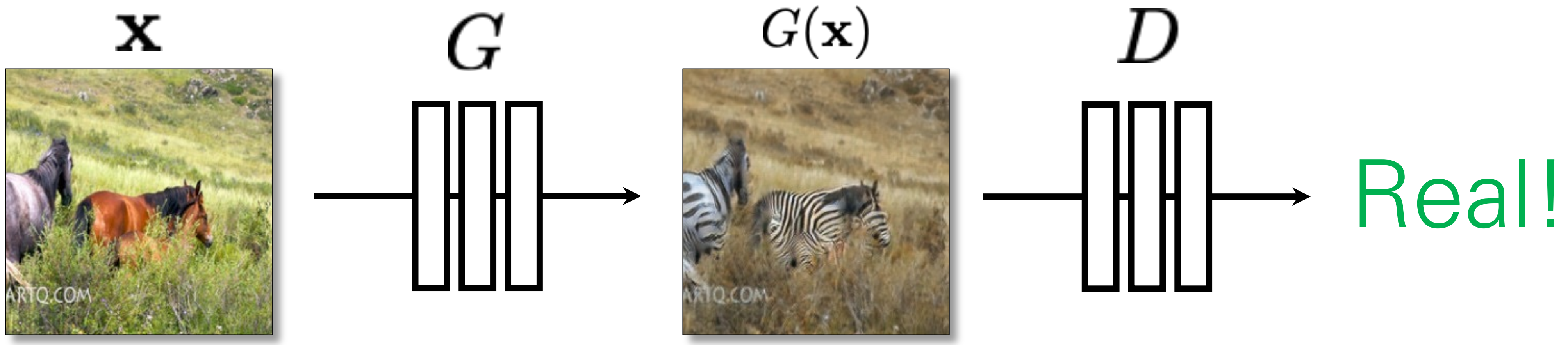
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ ]$$

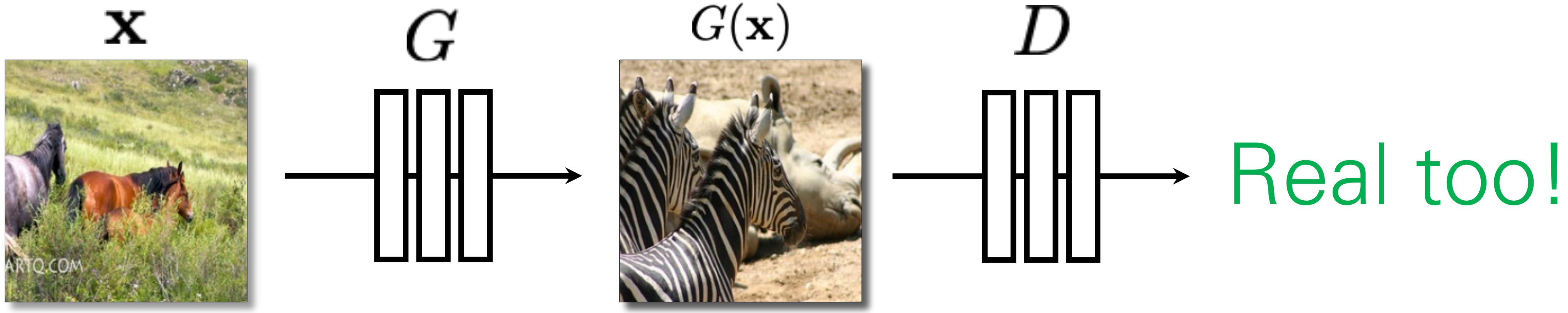
Usually loss functions check if output matches a target instance

GAN loss checks if output is part of an admissible set


Gaussian
Target distribution
z
Y

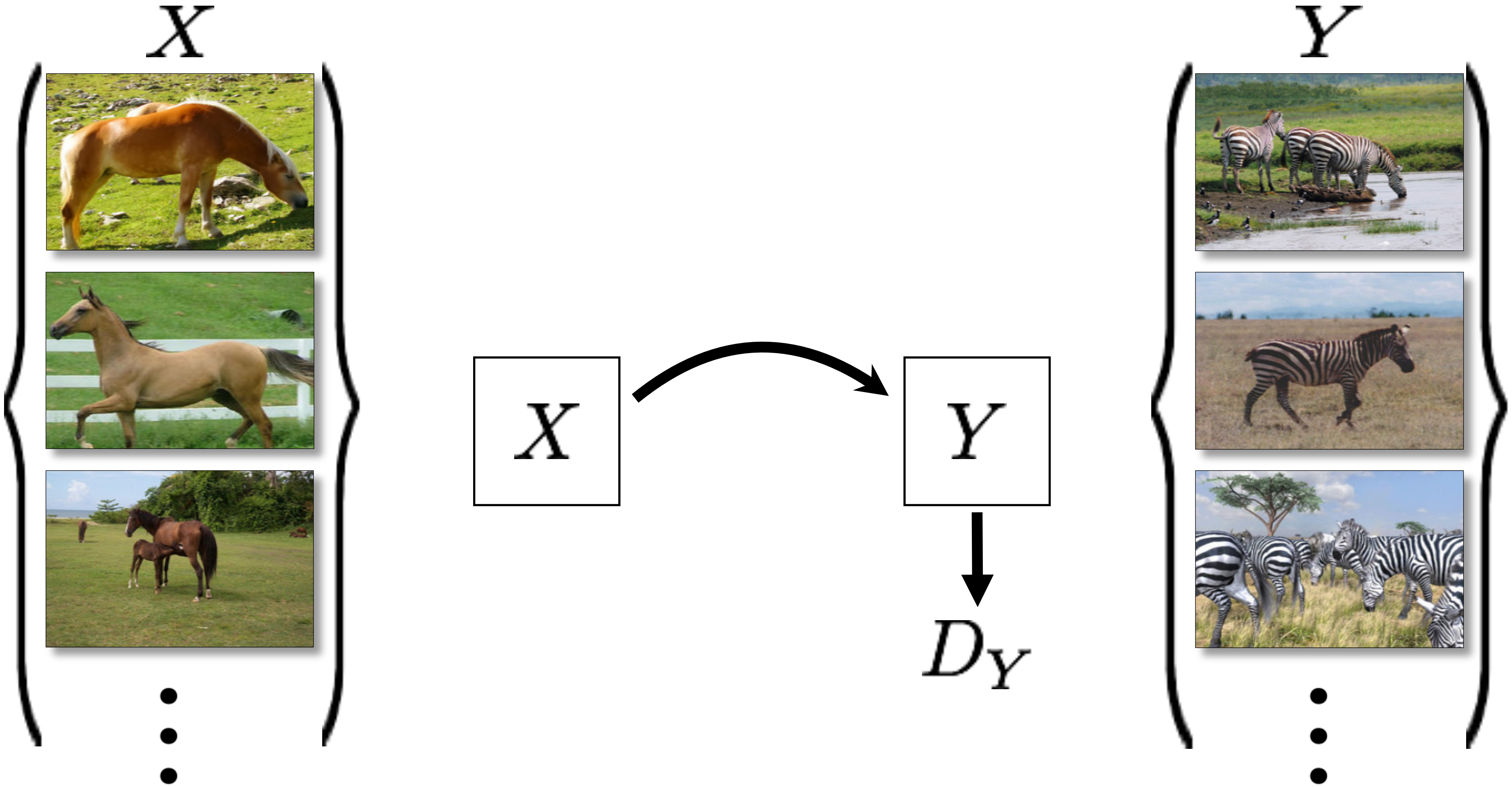

Horses
Zebras
X
Y


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



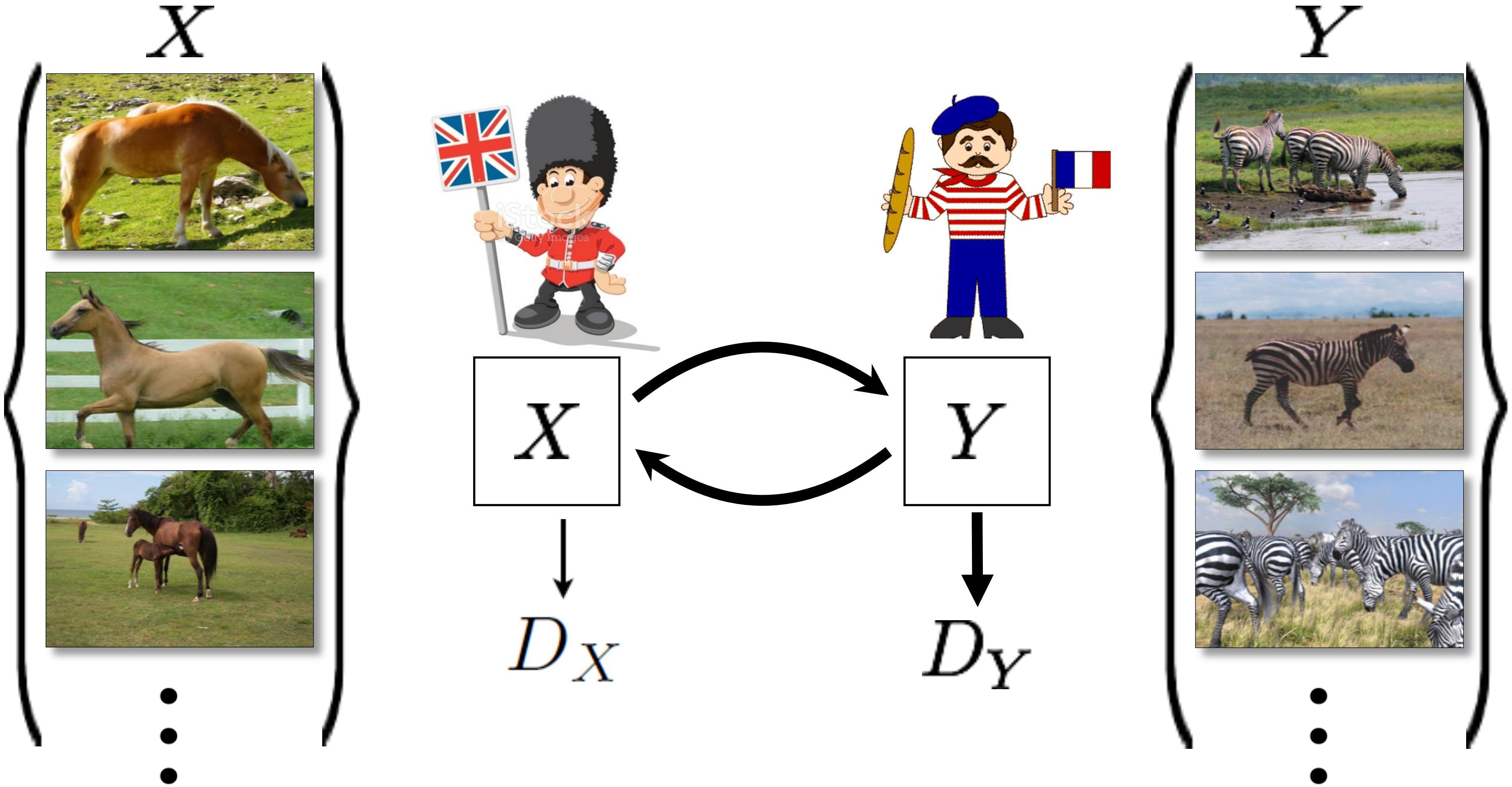
Nothing to force output to correspond to input

Cycle-Consistent Adversarial Networks

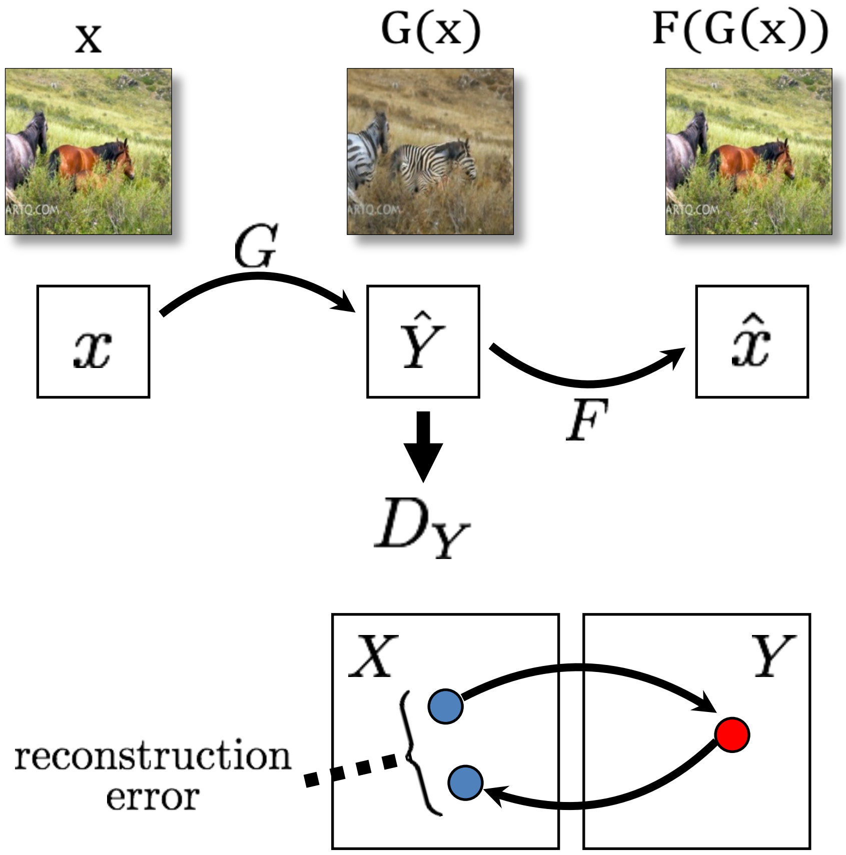


[Zhu et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

Cycle-Consistent Adversarial Networks

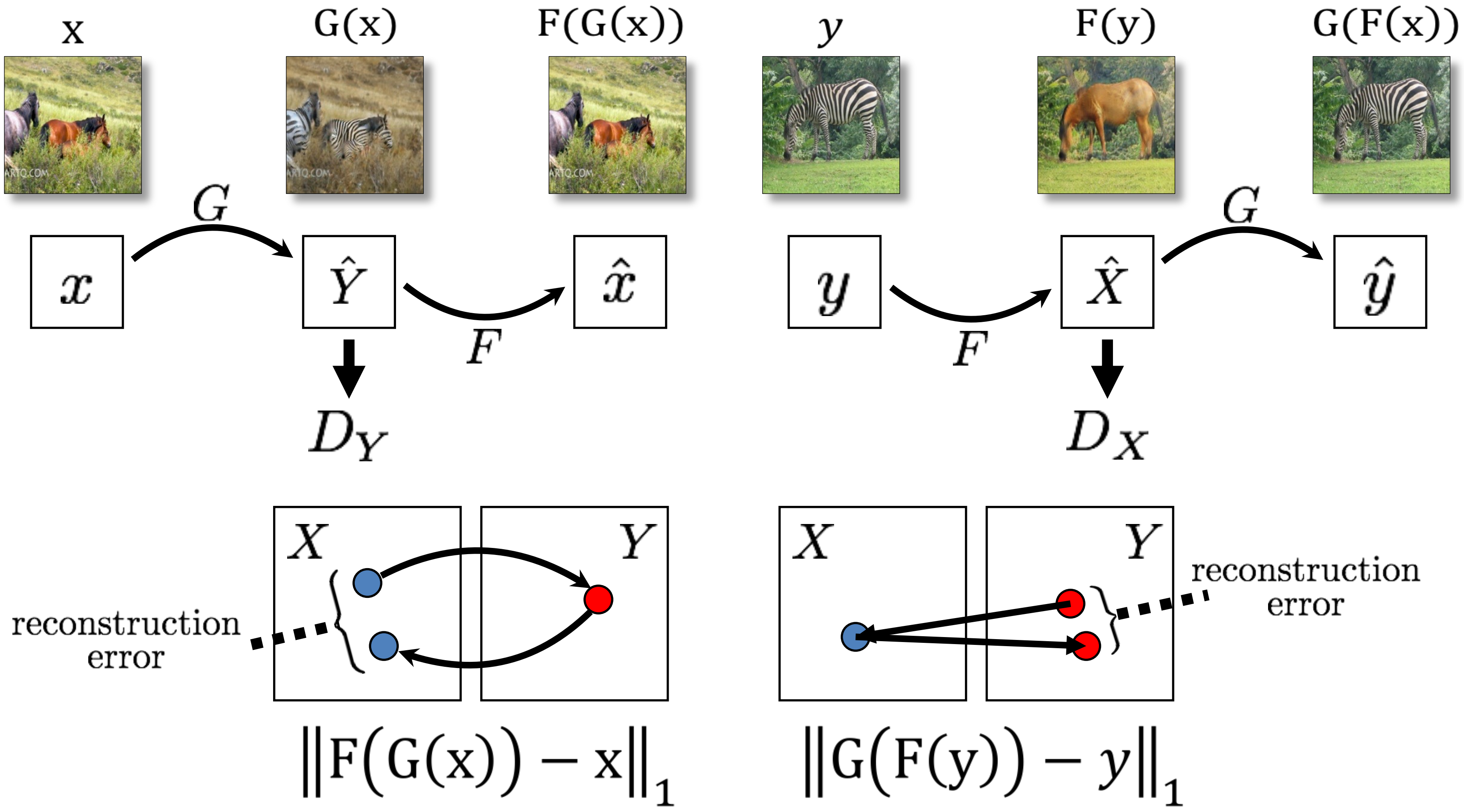


Cycle Consistency Loss



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

Cycle Consistency Loss




Collection Style Transfer

Photograph
@ Alexei Efros

Monet

Van Gogh

Cezanne

Ukiyo-e

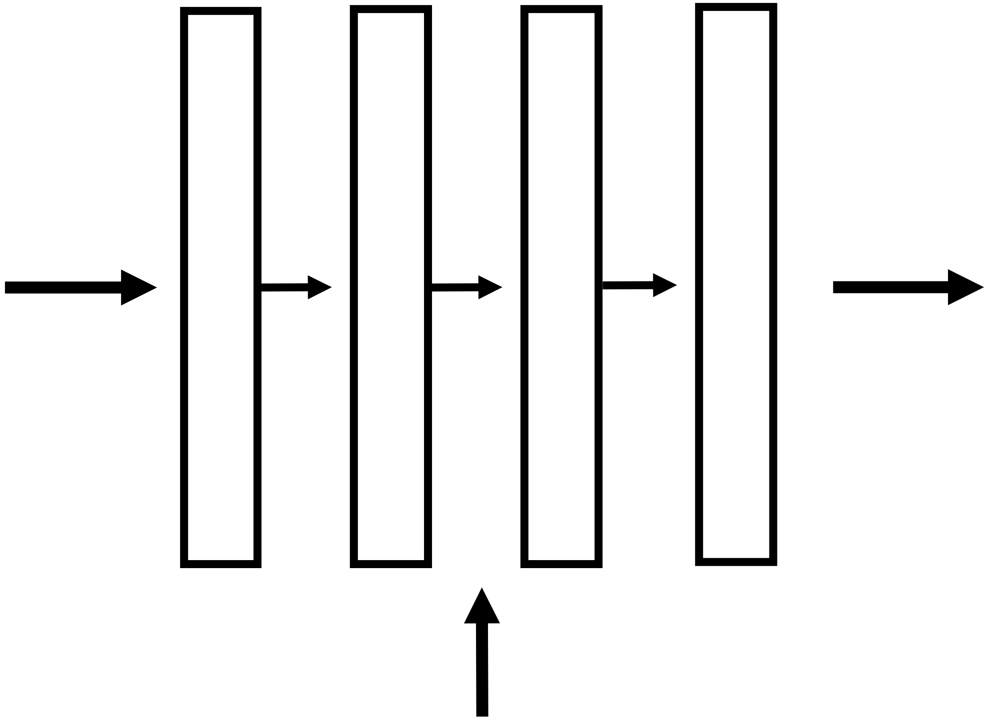

Input
Monet
Van Gogh
Cezanne
Ukiyo-e

Monet's paintings → photos

Monet's paintings → photos

Semantic Image Synthesis

semantic layout $\mathbf{x}$



synthesized image $\mathbf{y}$

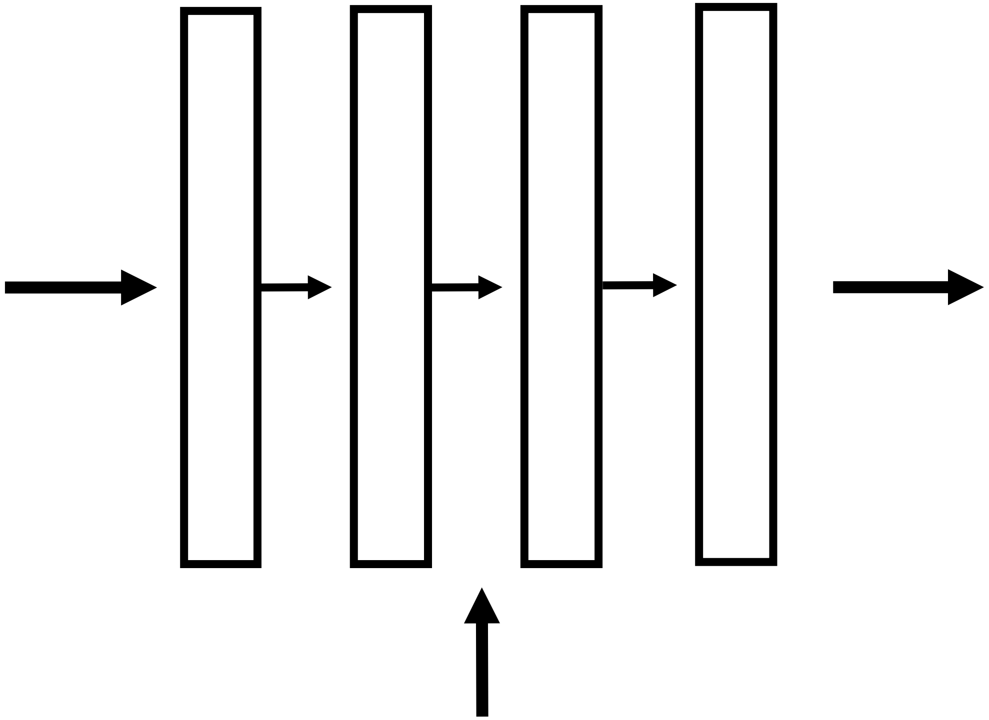
- **Input:** input layout + target style image
  Output: synthesized image

Target style image $\mathbf{t}$

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, Jun-Yan Zhu. **Semantic Image Synthesis with Spatially-Adaptive Normalization**. CVPR 2020.

Semantic Image Editing

image $\mathbf{x}$



"more flowers"
"more cloudy"

Target scene attribute(s) $\mathbf{t}$

manipulated image $\mathbf{y}$

- **Input:** input image
  (+ semantic layout)
  + target attribute

  Output: manipulated image

Levent Karacan, Zeynep Akata, Aykut Erdem, Erkut Erdem. **Manipulating Attributes of Natural Scenes via Hallucination.** ACM TOG 2020.

Semantic Image Synthesis (SPADE) (Park et al., 2019)

- Image generation conditioned on semantic layouts

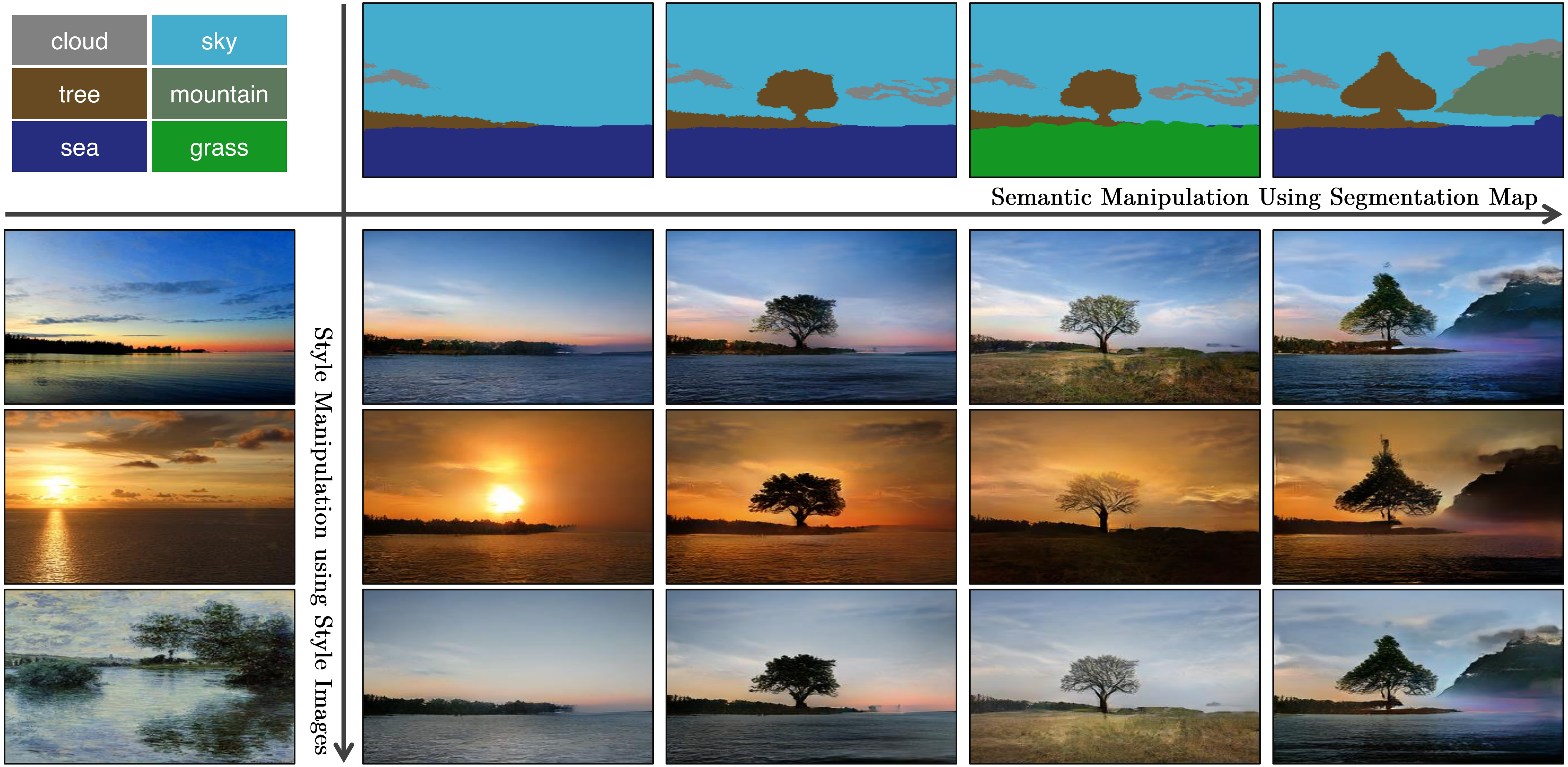
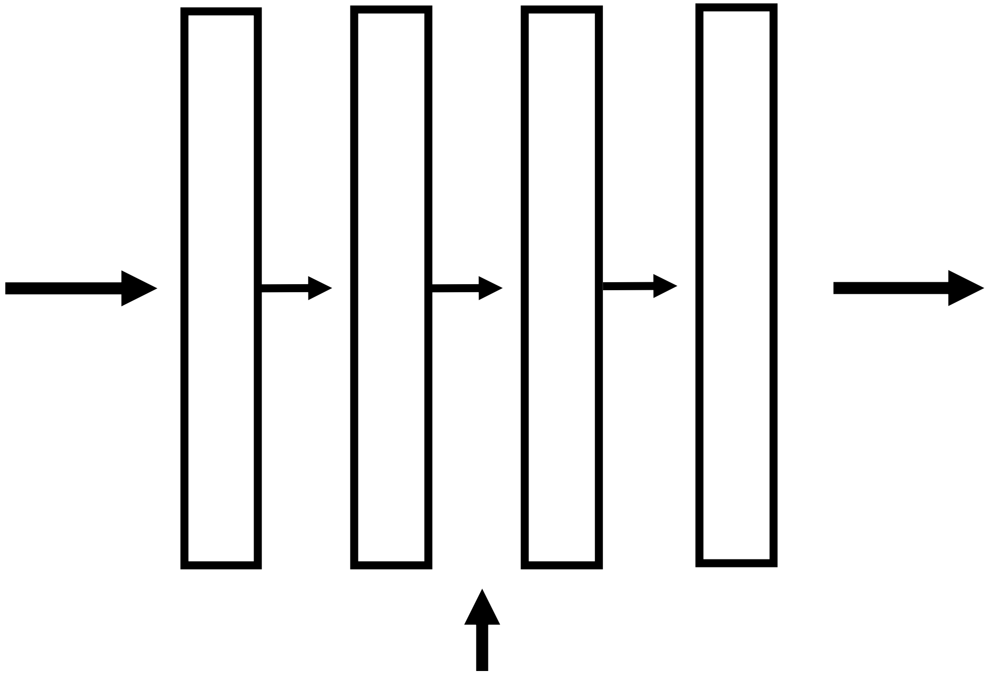


Photo: Aaron Sping // Chautauque Park, Colorado

Imagine this scene in a snowy winter day…


Photo: Aaron Sping // Chautauque Park, Colorado

image $\mathbf{x}$



"more flowers"
"more cloudy"

Target transient
scene attribute $\mathbf{t}$

manipulated
image $\mathbf{y}$

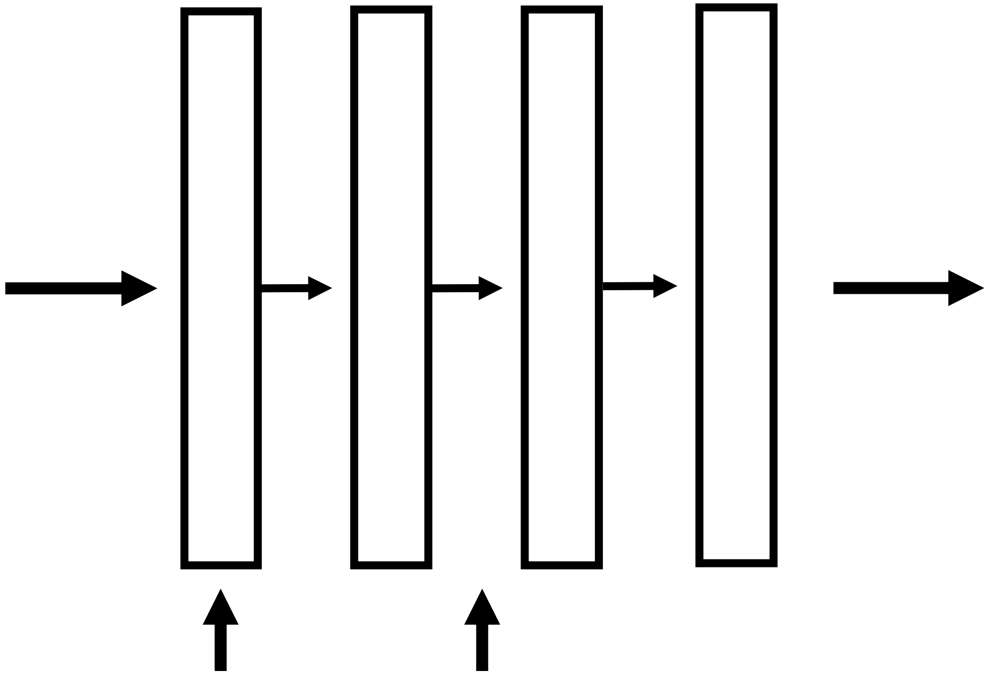
- **Input:** input image
  + target attribute

  Output: manipulated image

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

image $\mathbf{x}$

semantic layout $\mathbf{l}$



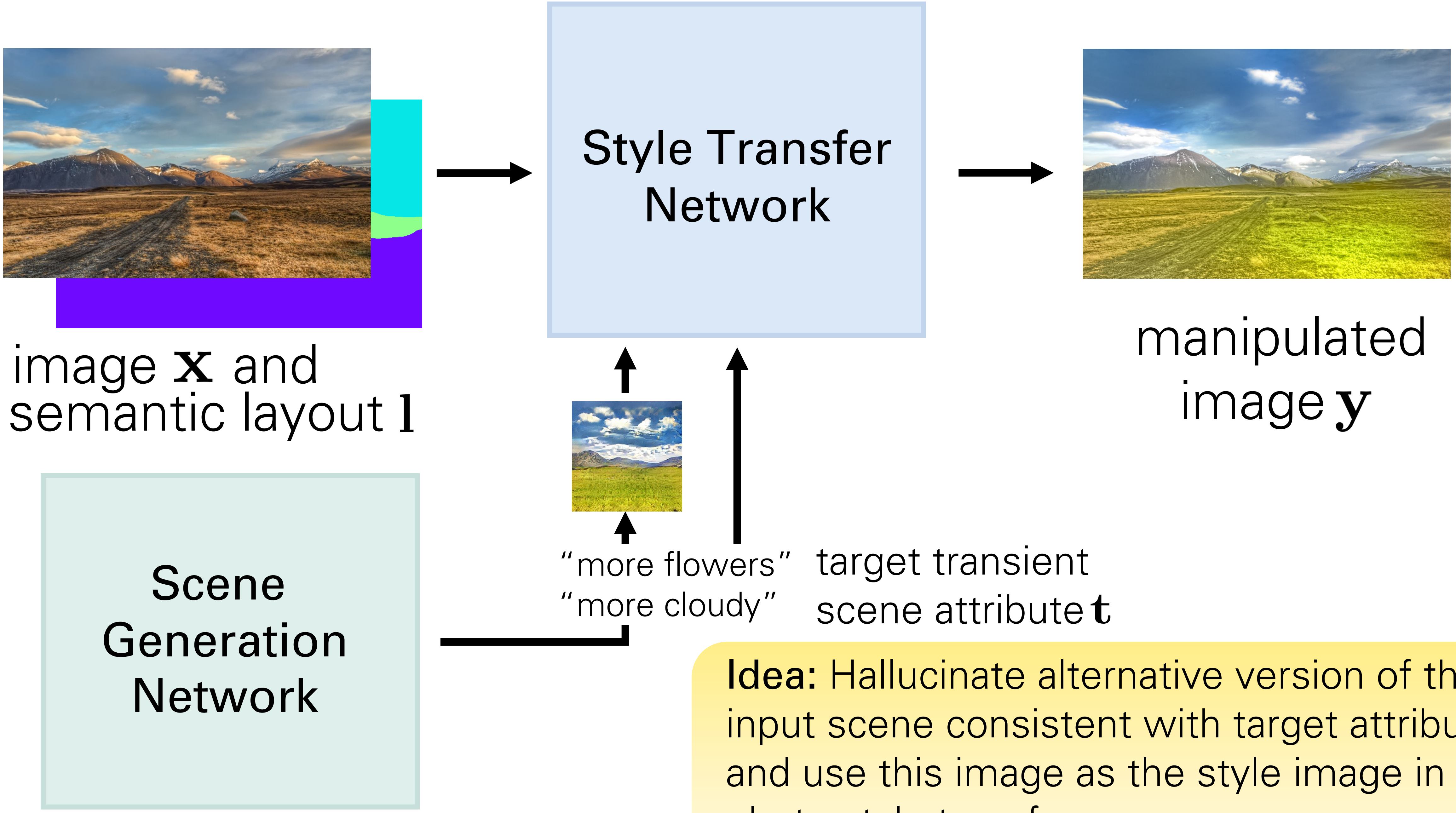
"more flowers"
"more cloudy"

Target transient scene attribute $\mathbf{t}$

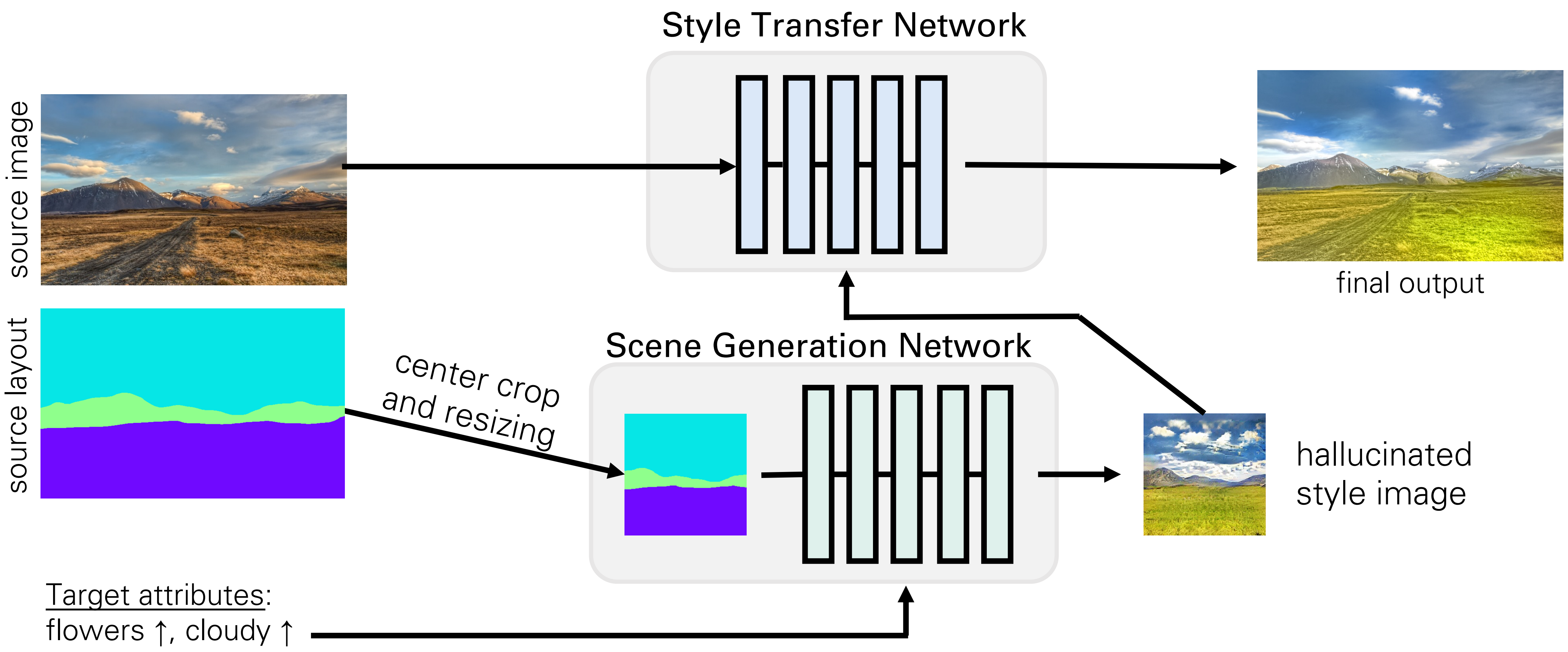
manipulated image $\mathbf{y}$

- **Input:** input image
  + semantic layout
  + target attribute

  Output: manipulated image



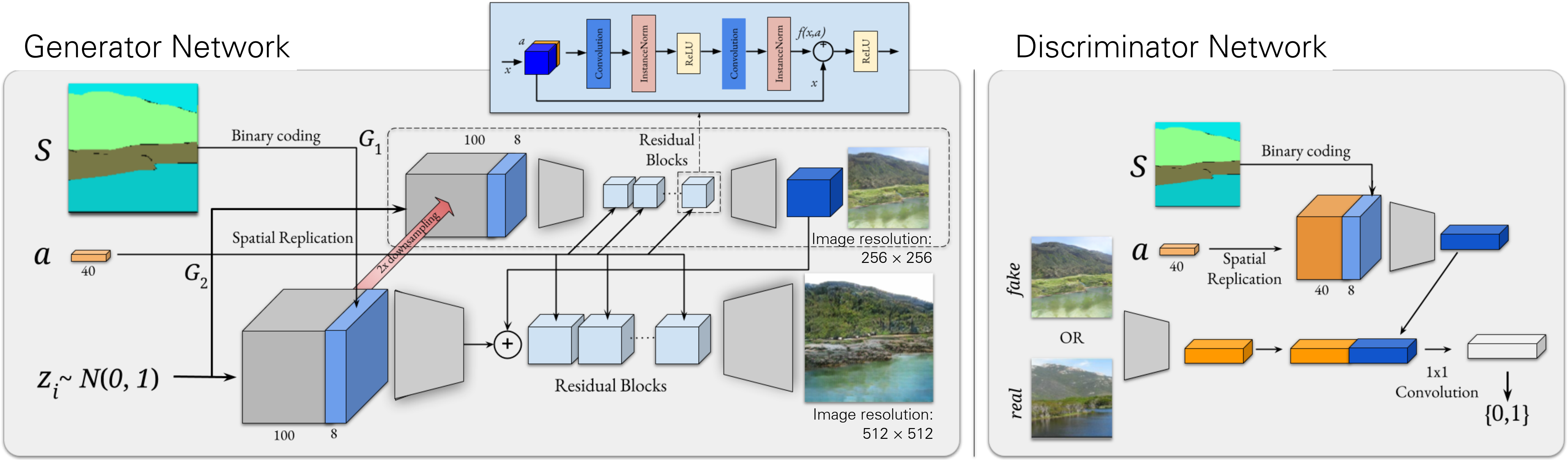
Idea: Hallucinate alternative version of the input scene consistent with target attributes and use this image as the style image in the photo style transfer.



- **Scene Generation Network**
  - A conditioned GAN model with two conditions:
    (1) semantic layout,
    (2) target attributes
- **Style Transfer Network**
  - A deep photo style transfer network that modifies the look of the source image based on the hallucinated style image

Scene Generation Network

- The semantic layout categories are encoded into 8-bit binary codes
- The transient attributes are represented by a 40-d vector.



- An architecture similar to Pix2pixHD model (Wang et al. 2018)
- **Generator network**: A coarse-to-fine model with 2 generator networks
- **Discriminator network**: A combination of three different discriminator networks operating at an image pyramid of 3 scales

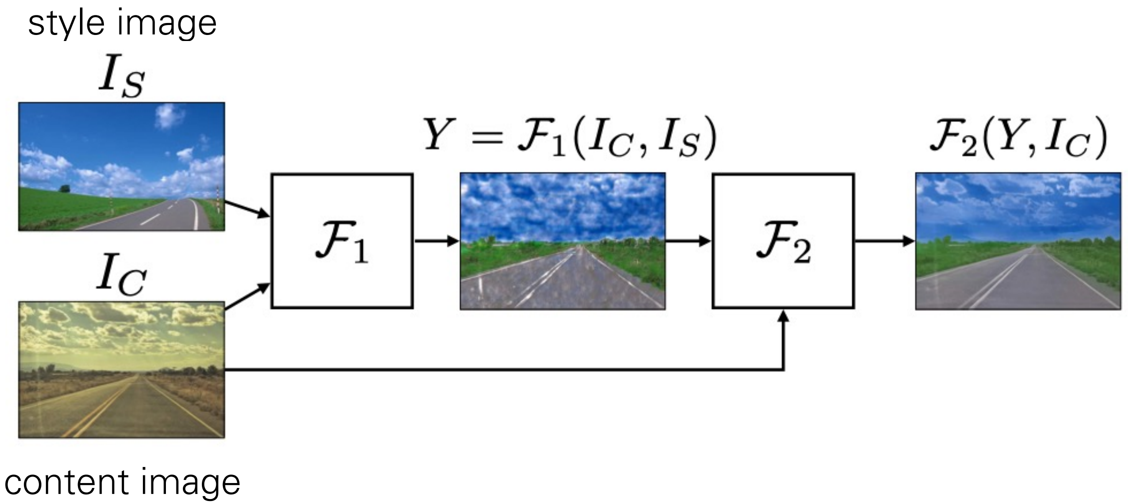
T.-C. Wang et al. **High-resolution image synthesis and semantic manipulation with conditional GANs.** CVPR 2018.

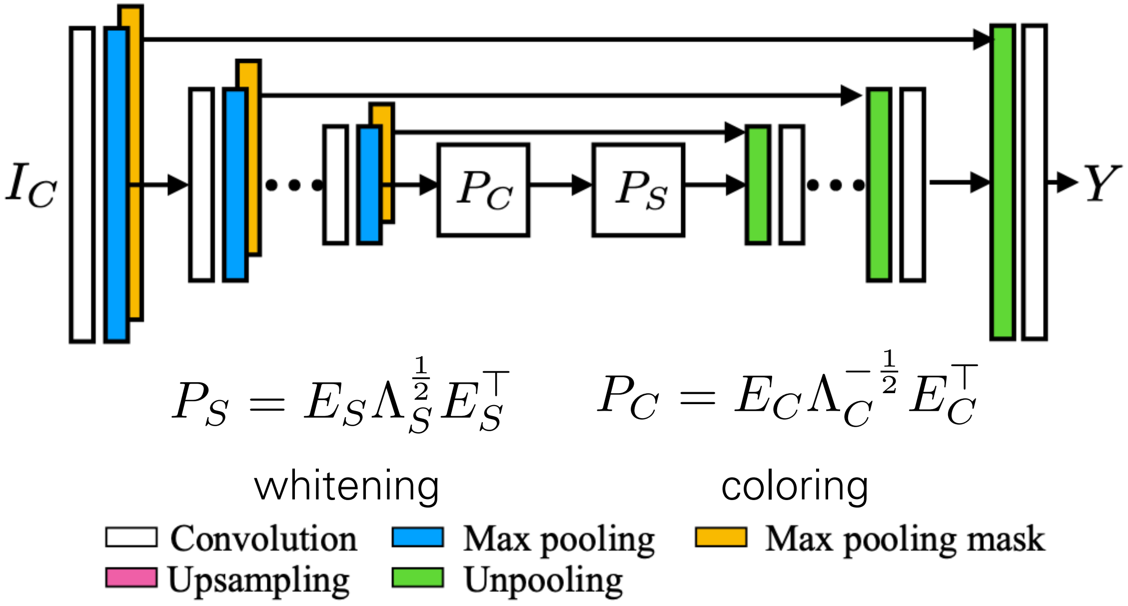
Style Transfer Network

- The FPST method of (Li et al., 2018), which is composed of two steps with close-form solutions:
  1. Stylization step $\mathcal{F}_1$
  2. Smoothing step $\mathcal{F}_2$

$$I_{out} = \mathcal{F}_2\left(\mathcal{F}_1(I_C, I_S), I_C\right)$$

- The **stylization step** is based on the whitening and coloring transform to stylize images via feature projections
  - Style information encoded by the covariance matrix of VGG features
- The **smoothing step** ensures spatially consistent stylizations via a manifold ranking operator.





Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

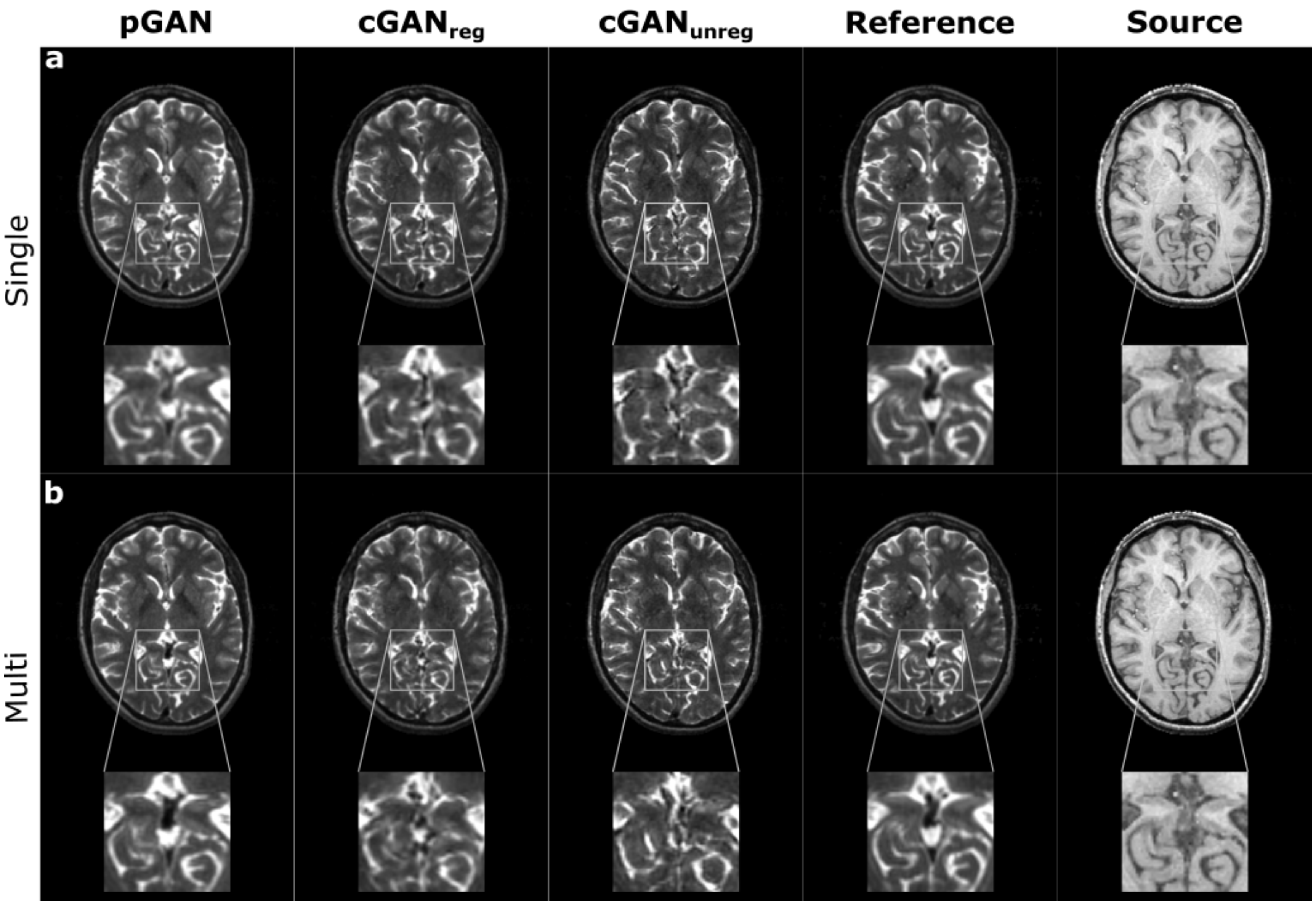
Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

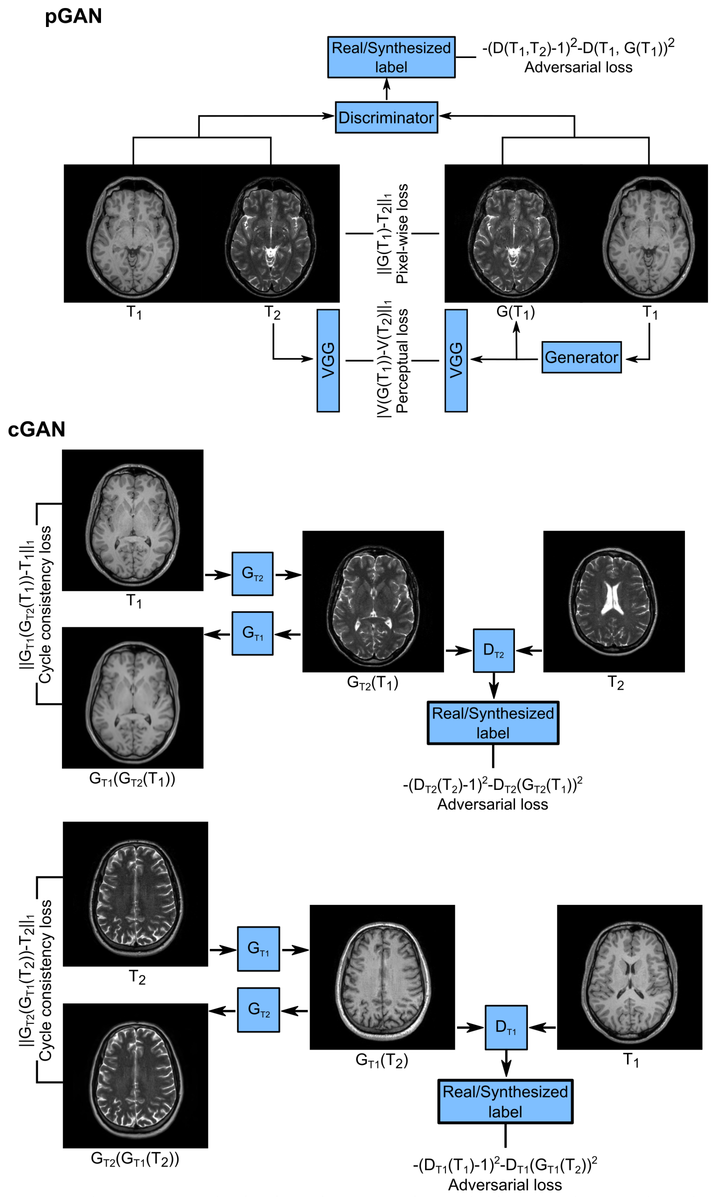
Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

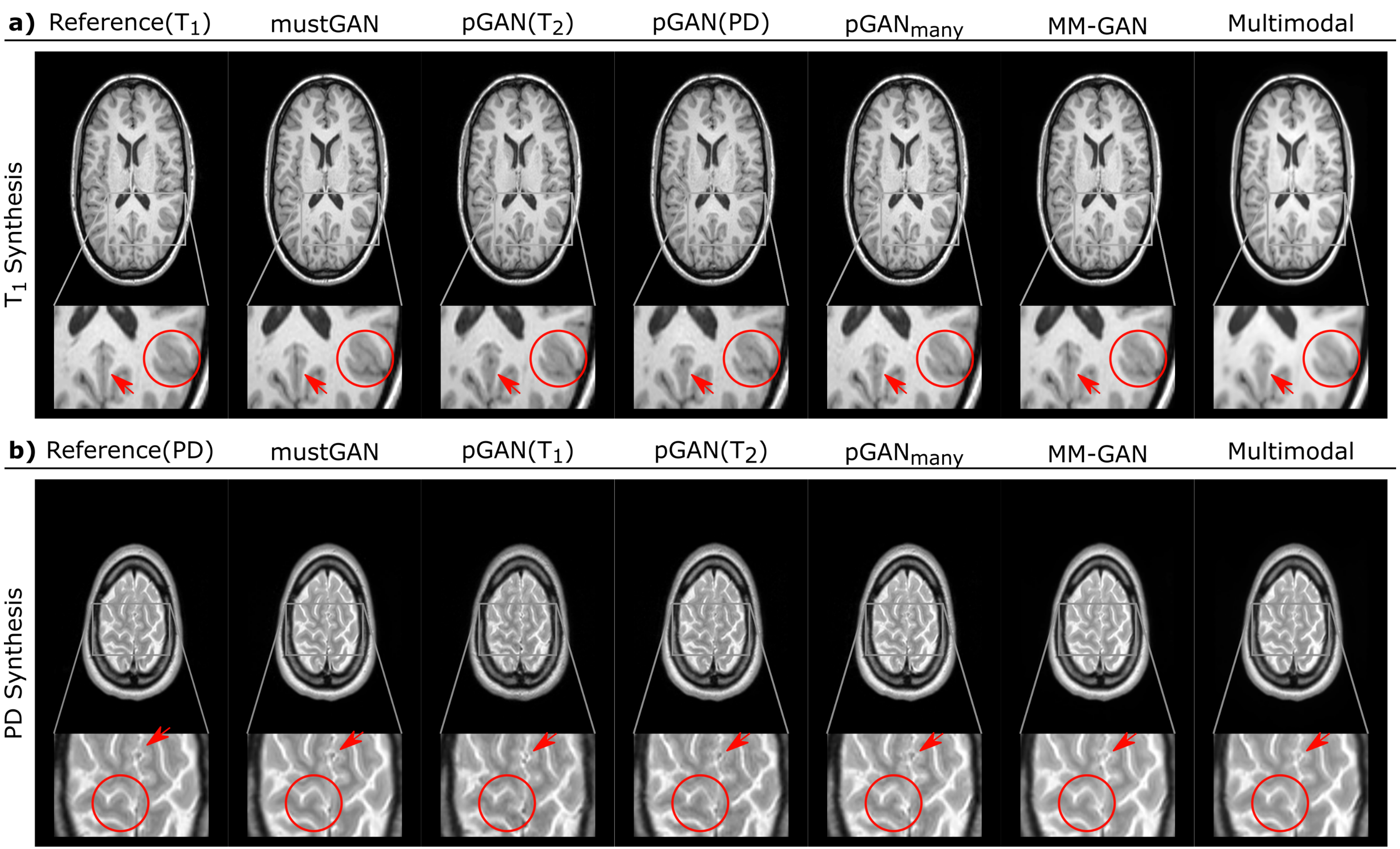
Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

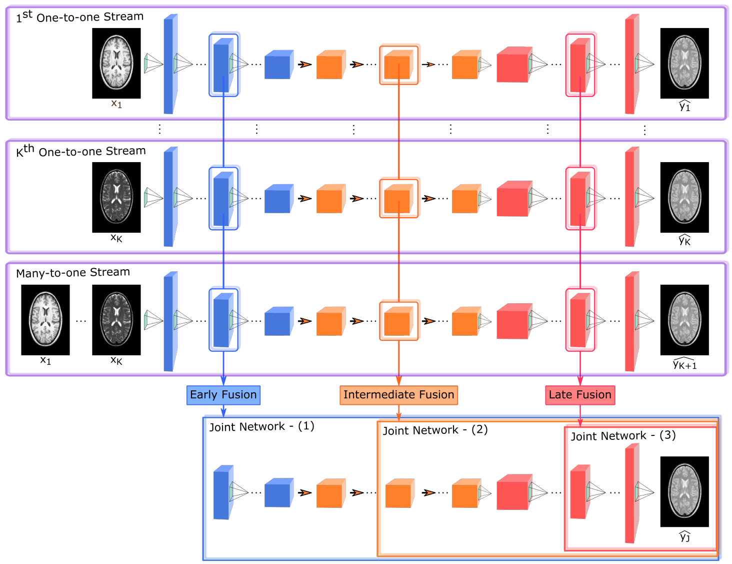
Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]





- Image Synthesis in Multi-Contrast MRI [Ul Hassan Dar et al. 2019]

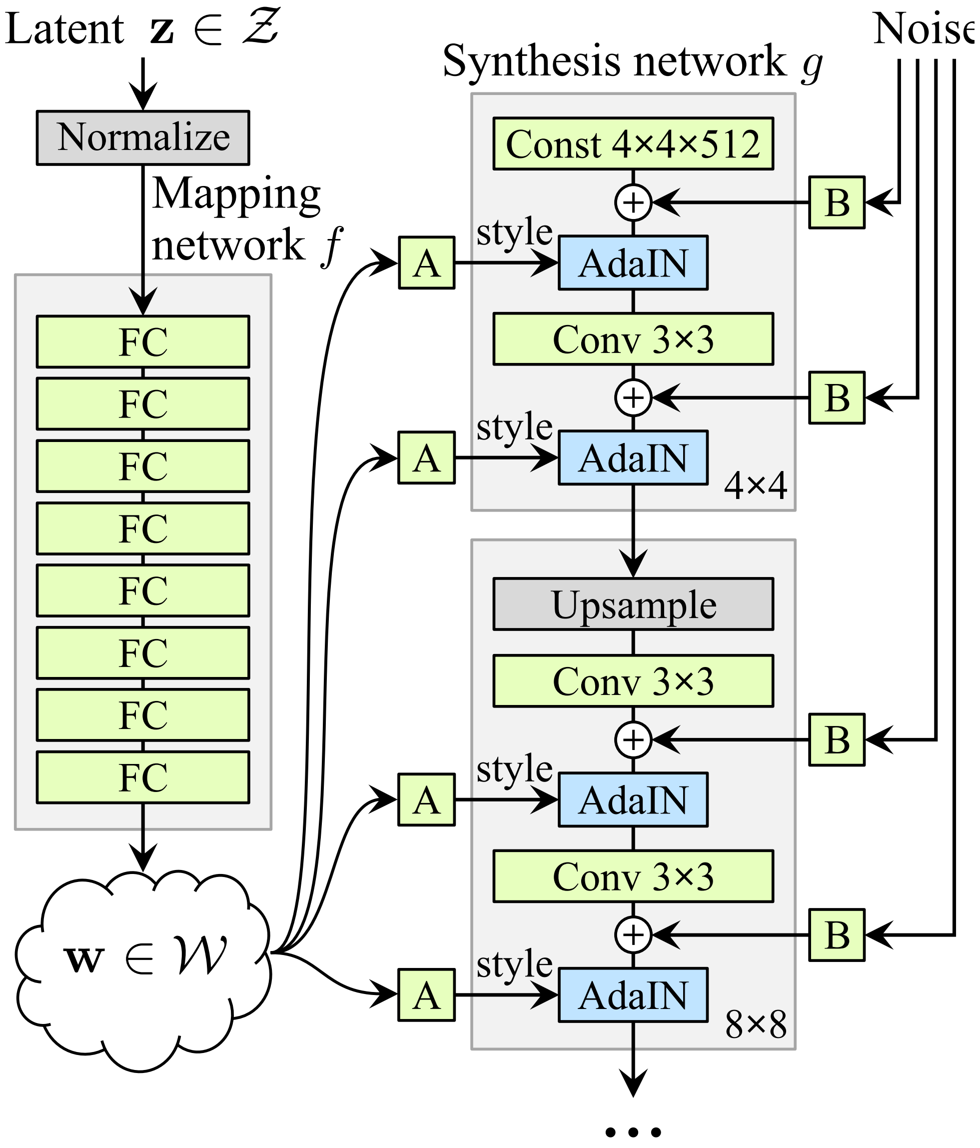




- Image Synthesis in Multi-Contrast MRI [Mahmut Yurt et al. 2021]

Recall StyleGANs (Karras et al., 2019)

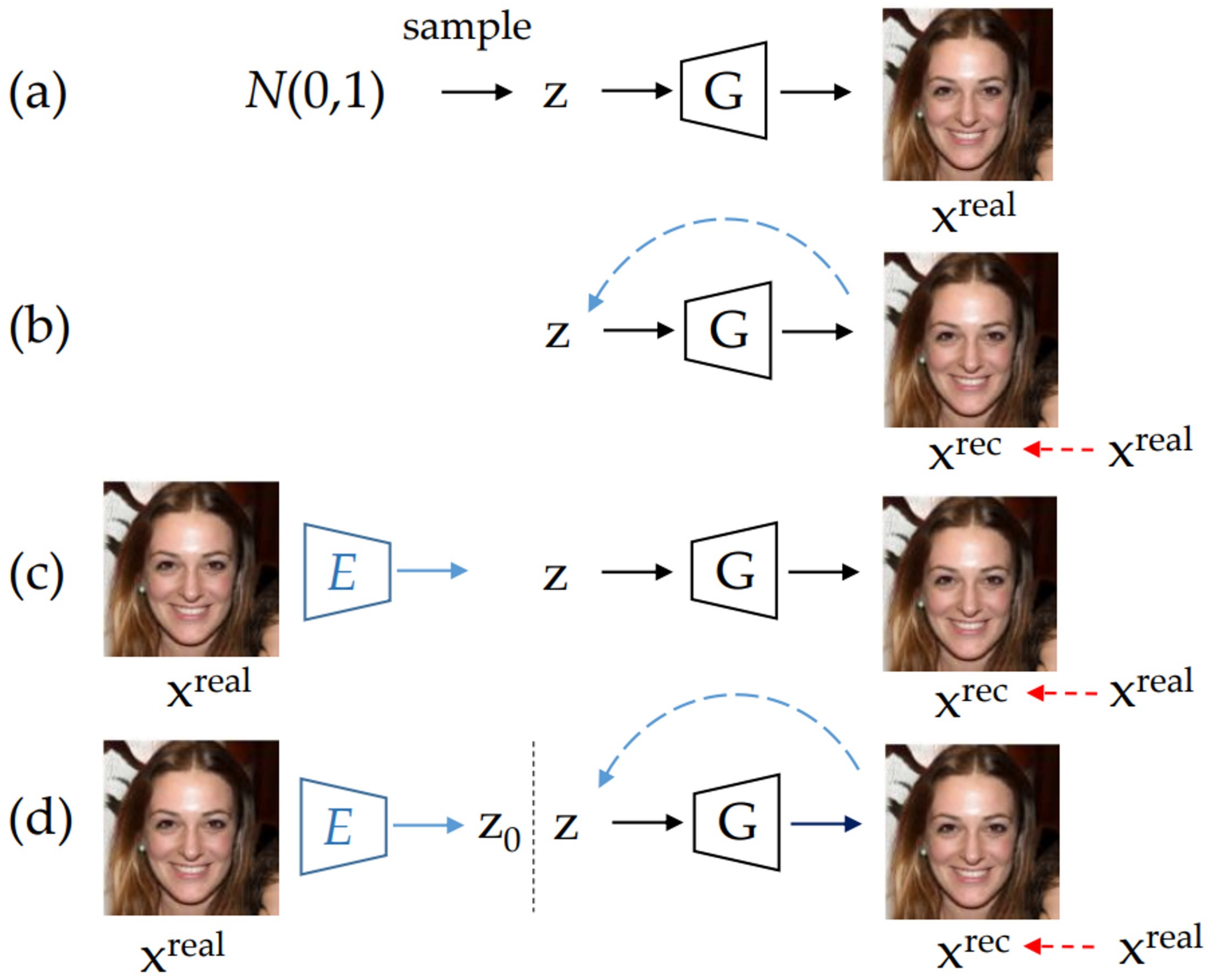
- A new architecture motivated by the style transfer networks
- allows unsupervised separation of high-level attributes and stochastic variation in the generated images



GAN Inversion

3 methods of inversion:

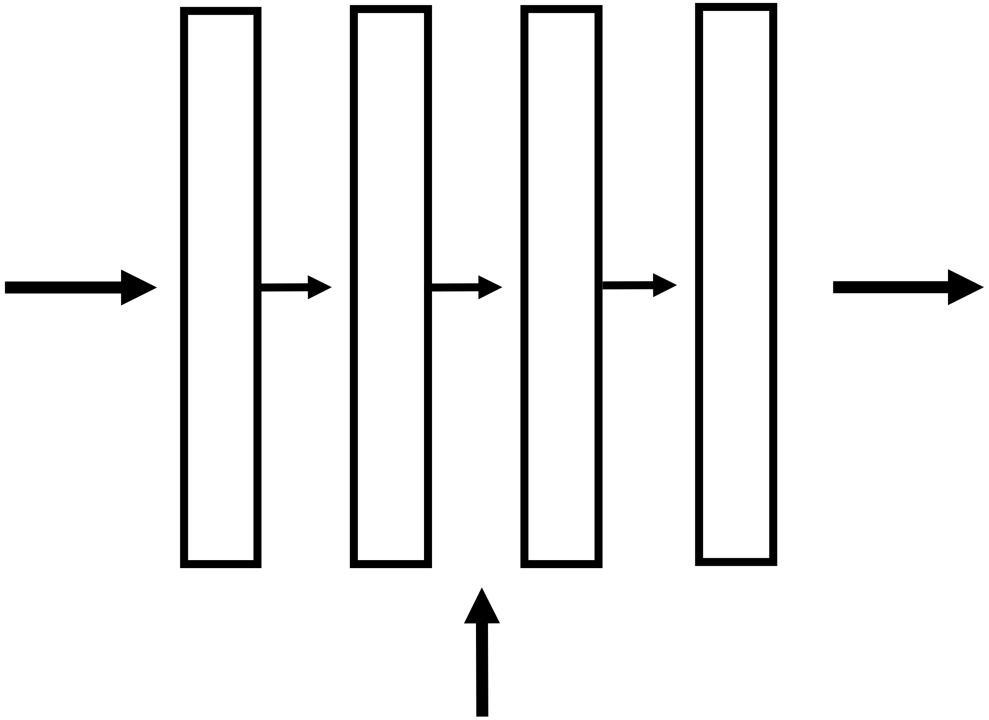
- Optimization-based (b)
- Learning-based (c)
- Hybrid (d)



[Xia et al., arXiv 2021]

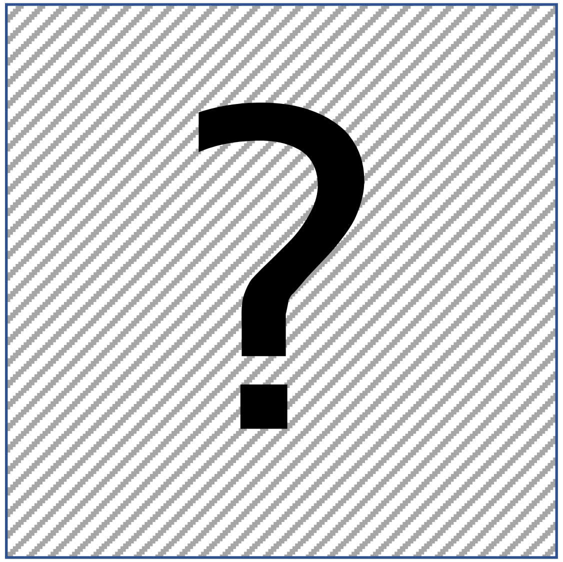
Semantic Image Editing

image $\mathbf{x}$



“he is tanned”

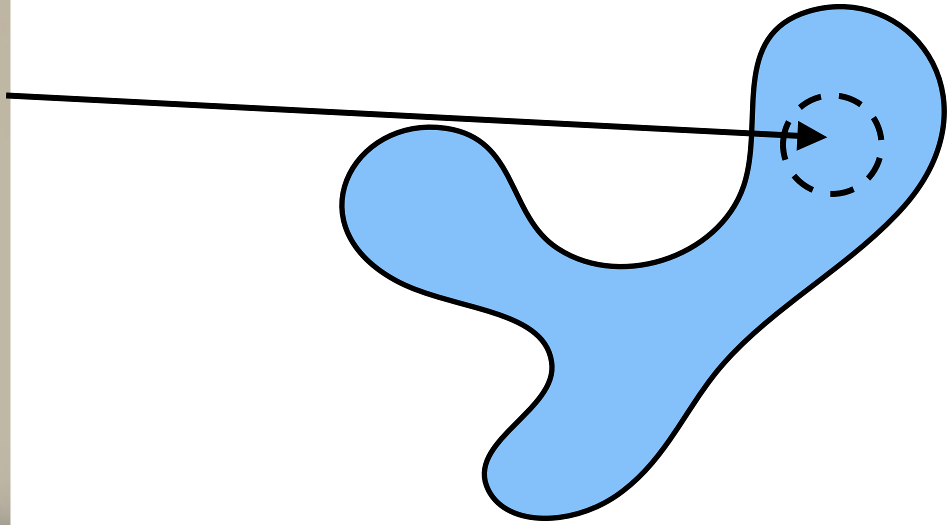
Target description $\mathbf{t}$



manipulated image $\mathbf{y}$

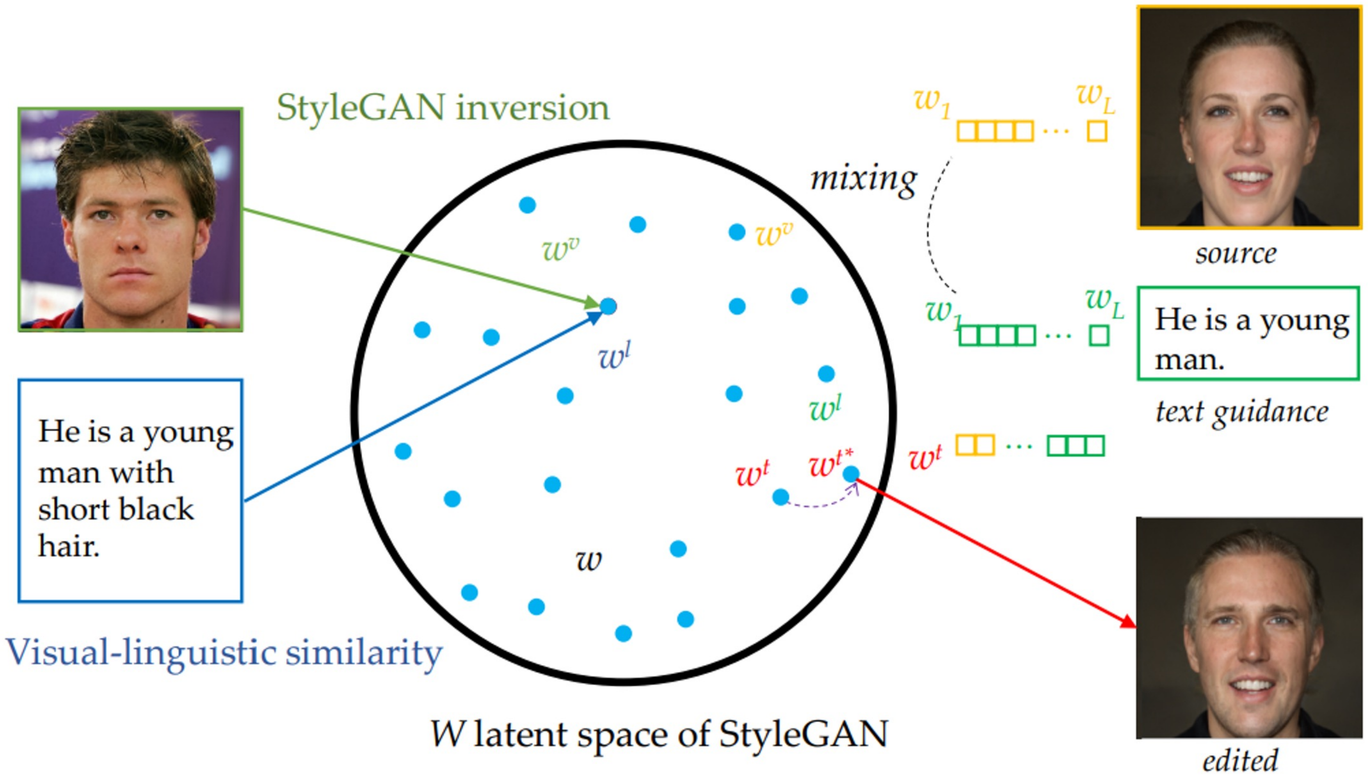
- **Input:** input image
  + target description
  Output: manipulated image

- Utilize pretrained StyleGAN model as a natural prior for face images
- Project input image to the latent space and perform edit in the vicinity of that point.



TediGAN

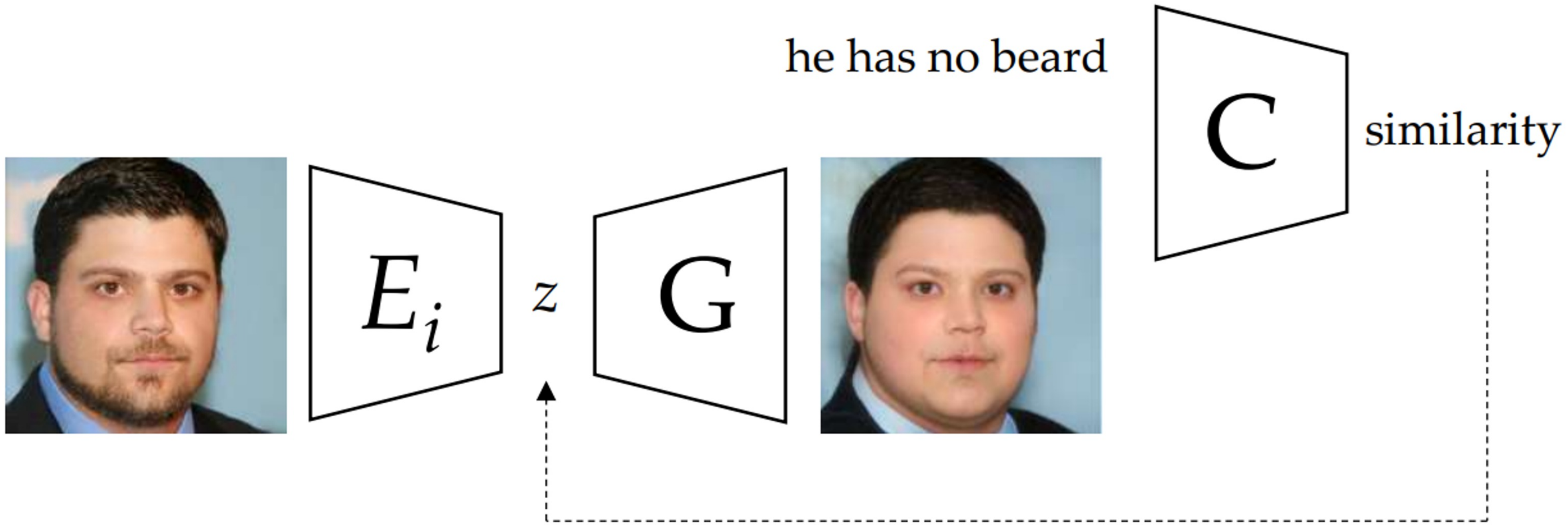
- A recent inversion based manipulation model.
- They proposed two different approaches.
- In their first approach, they learn a common visual-linguistic semantic space.
- They train a text encoder such that both modalities are embedded to the same space



(Xia et al., CVPR 2021)

TediGAN

- Their second method is optimization based.
- They use CLIP to provide a signal to directly optimize the latent code



(Xia et al., CVPR 2021)

StyleCLIP

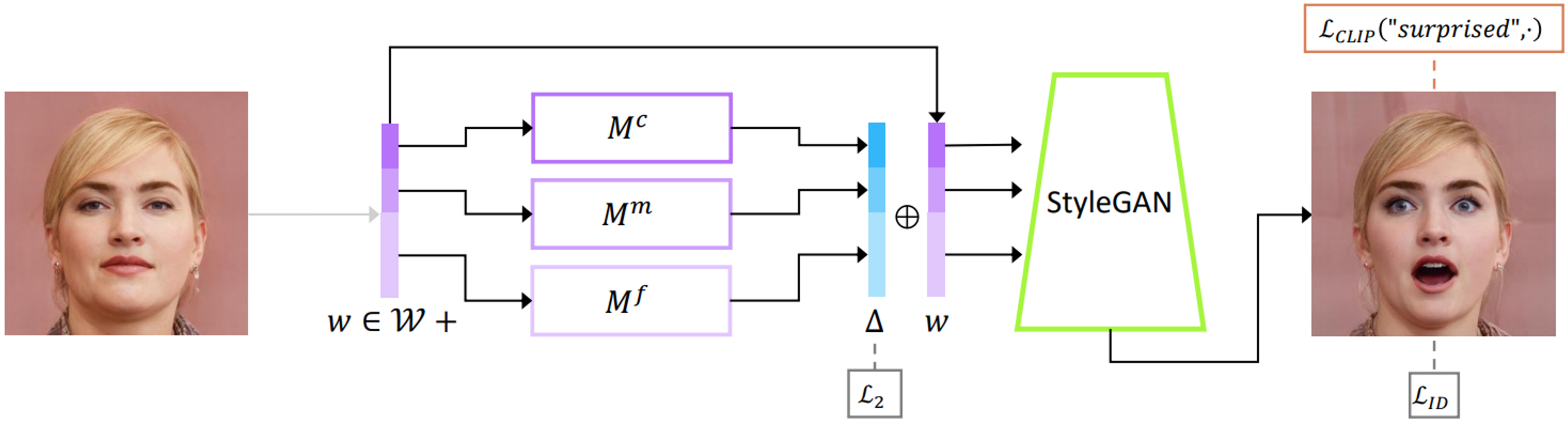
- StyleCLIP is another inversion based manipulation model
- Direct optimization
- Similar to TediGAN, they use CLIP to optimize the latent code directly
- They include a term to preserve identity

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

(Patashnik et al., ICCV 2021)

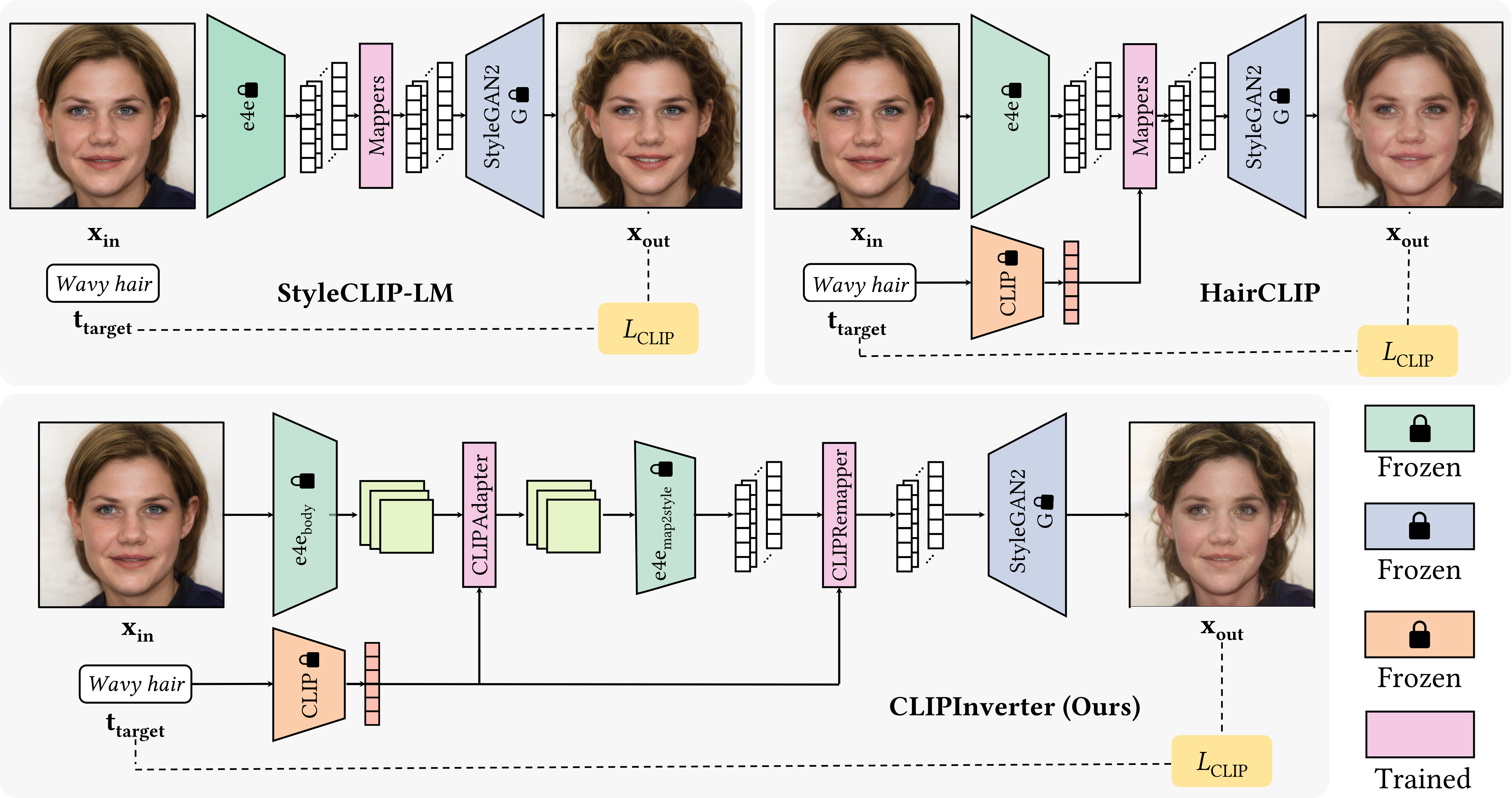
StyleCLIP

- Images are inverted to latent code and residuals (delta in the figure).
- Textual input is not used during inversion, it is only used in the loss
- Required to train different mappers for different text prompts.

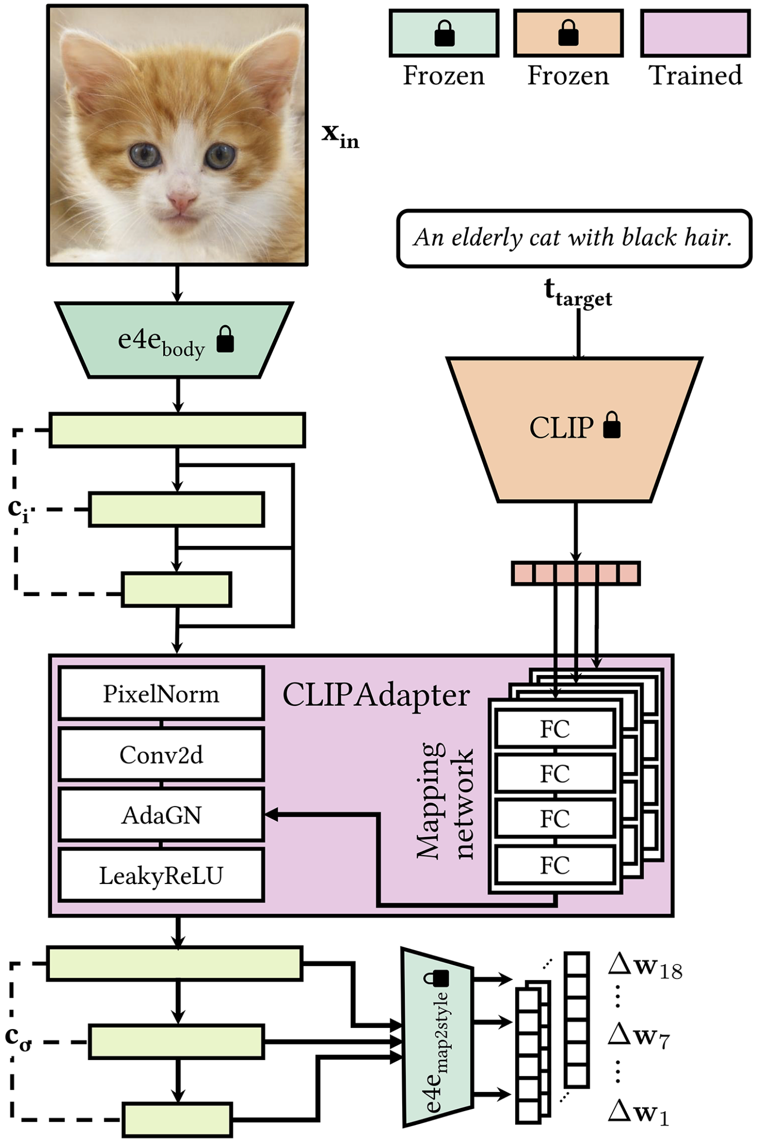


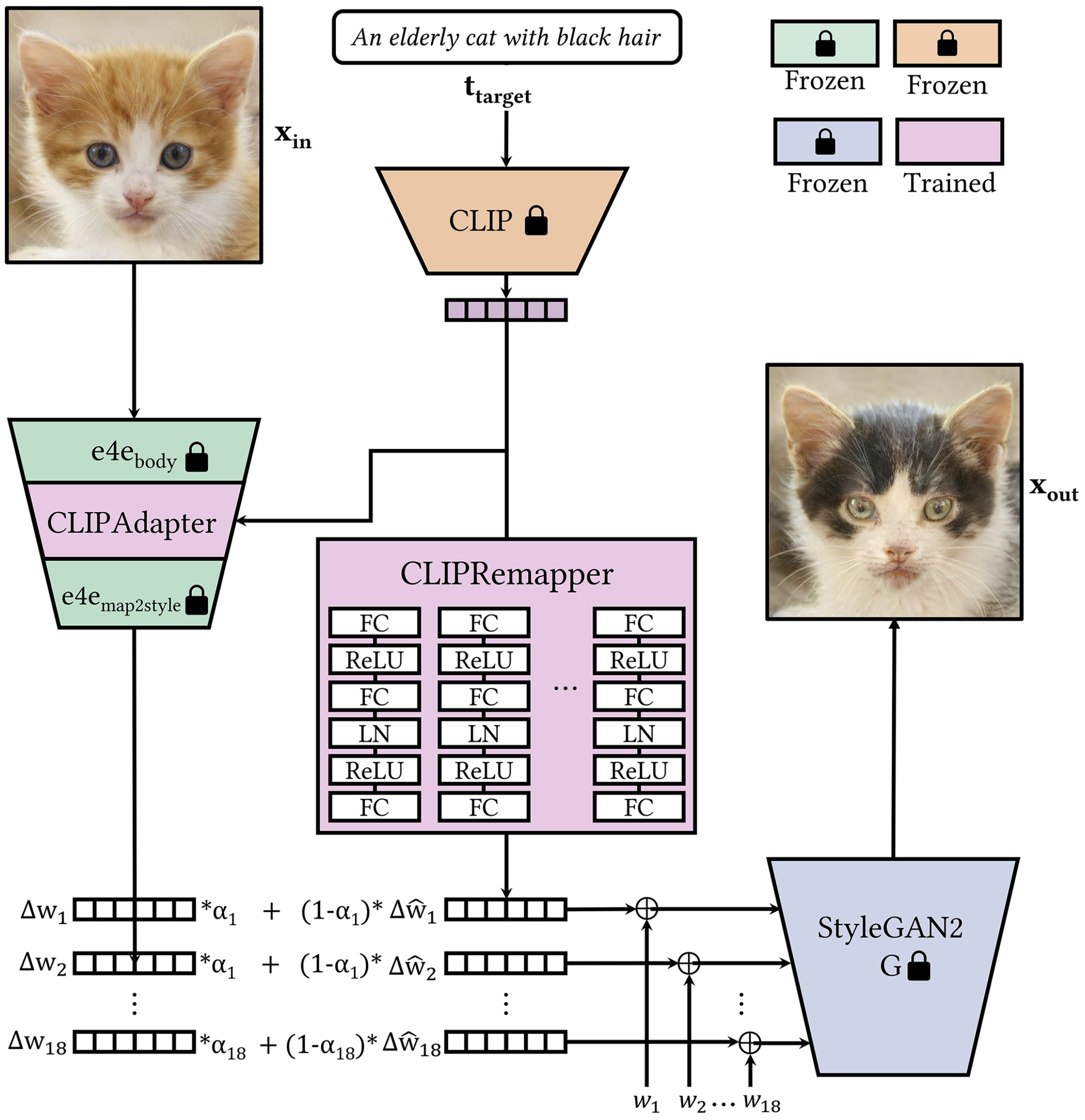
(Patashnik et al., ICCV 2021)

CLIPInverter



CLIPInverter





(Baykal et al., ACM TOG 2023)

CLIPInverter


He is smiling and has gray hair, high cheekbones, eyeglasses, double chin, and bags under eyes.


An old norwegian forest cat


This smaller bird is bright red and has black wings.


She has bangs. She is young and wears lipstick.


A grumpy elderly british shorthair cat


This is a small bird with green, yellow and blue on the breast, cheek patches, and crown.

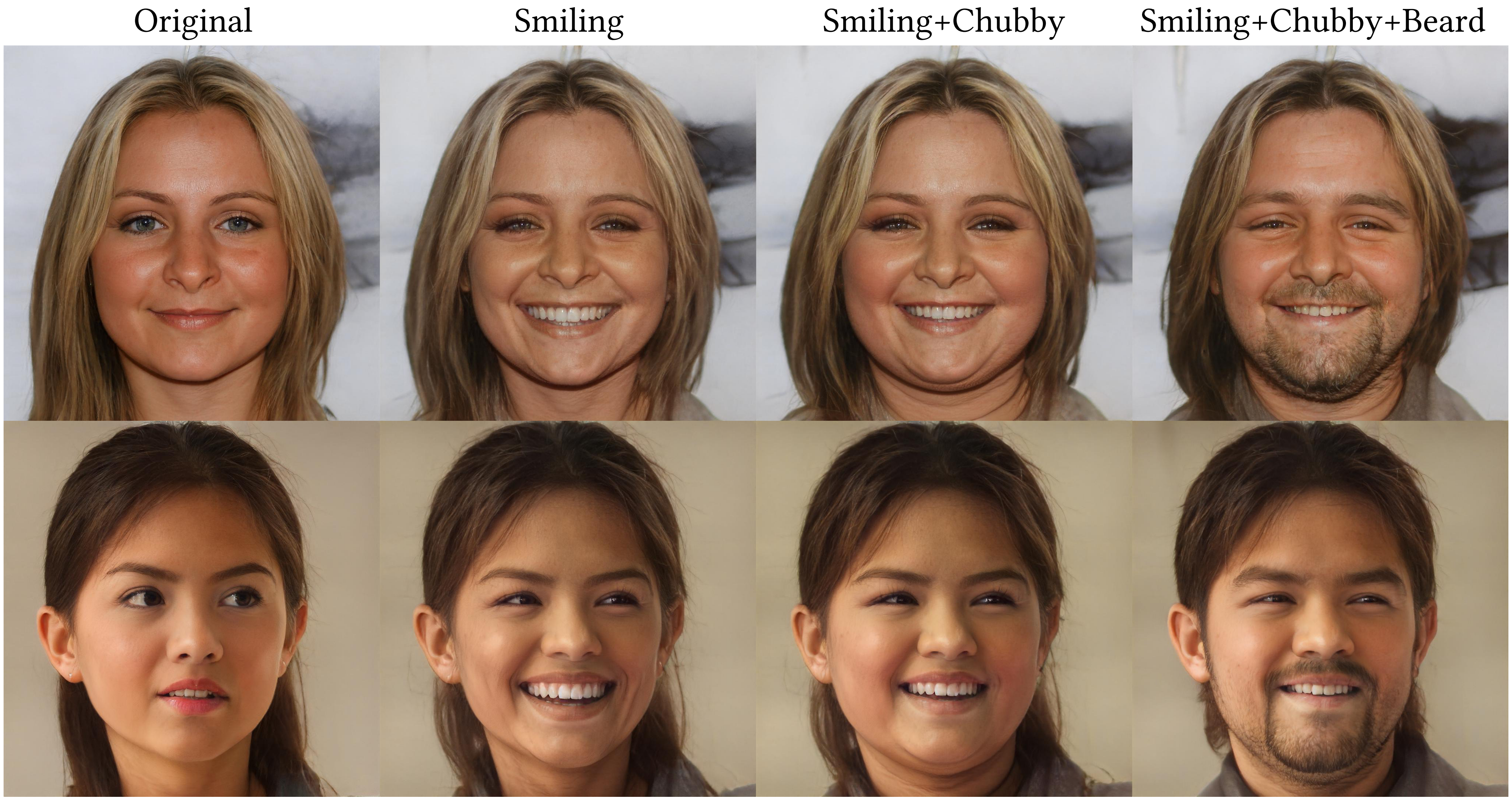

This woman is attractive and has arched eyebrows, straight hair, and blond hair.


A fearful cat with grey hair


This bird is brown with yellow and has a long, pointy beak.

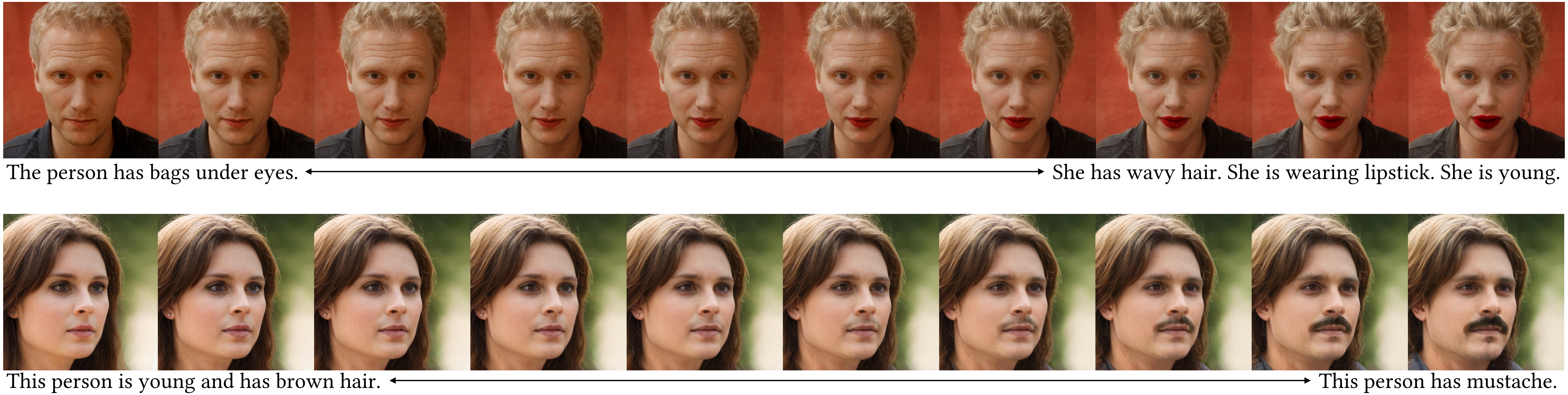
(Baykal et al., ACM TOG 2023)

CLIPInverter



(Baykal et al., ACM TOG 2023)

CLIPInverter



(Baykal et al., ACM TOG 2023)

CLIPInverter

(Baykal et al., ACM TOG 2023)

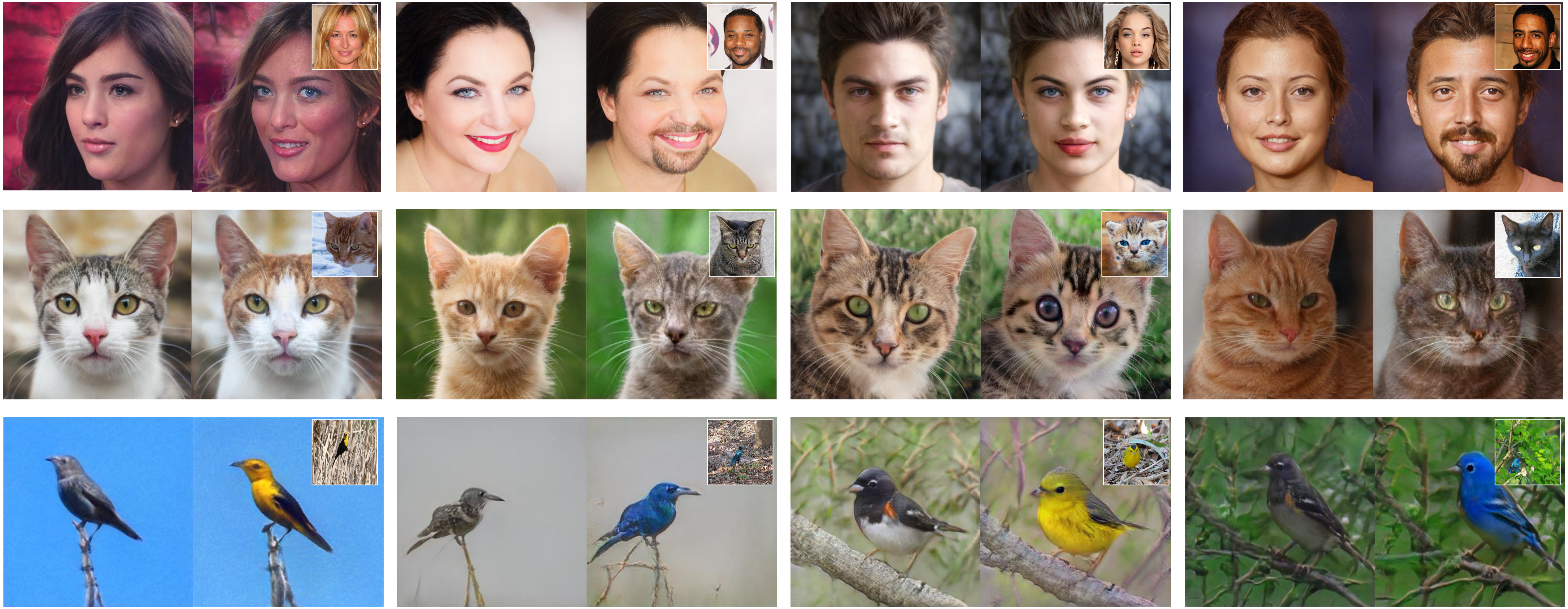
CLIPInverter

This small bird has a blue top and tail, and a small, straight beak that is pointed. ↔ This bird is gray, yellow, and orange in color, with a light colored beak.

This bird has a small head, a light grey belly and brown wings with long skinny black tarsus. ↔ This bird has wings that are blue and has a white belly.

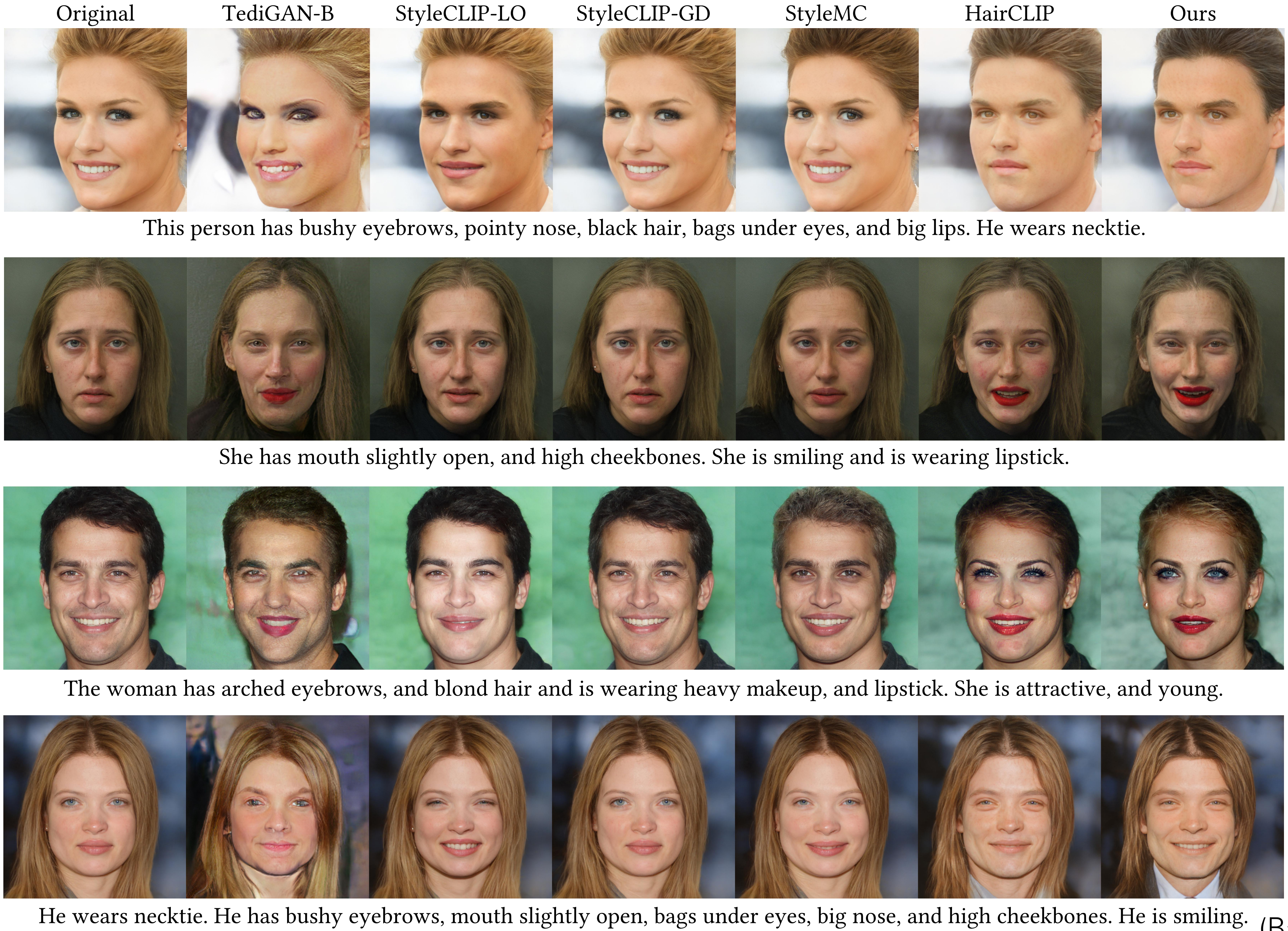
(Baykal et al., ACM TOG 2023)

CLIPInverter



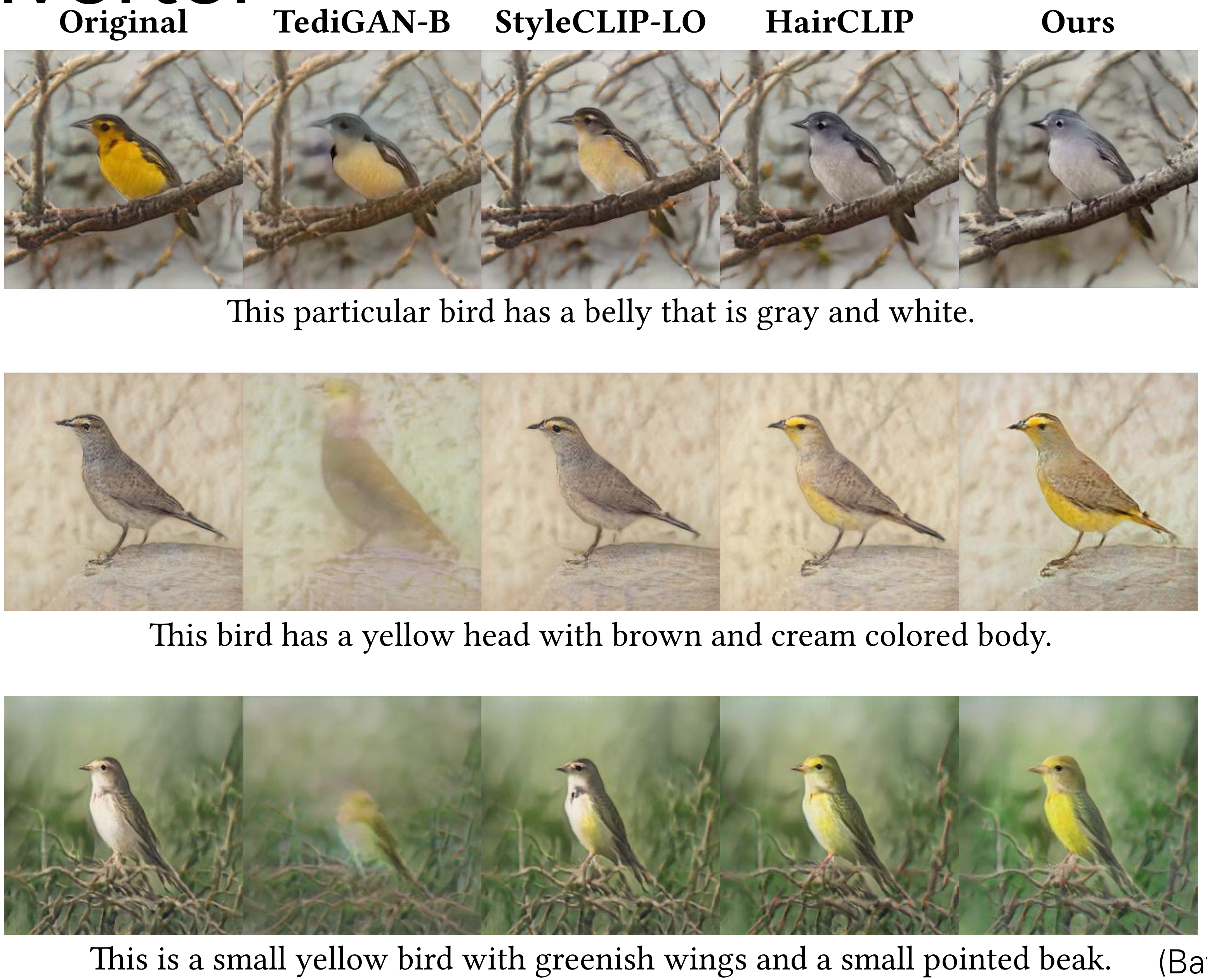
(Baykal et al., ACM TOG 2023)

CLIPInverter



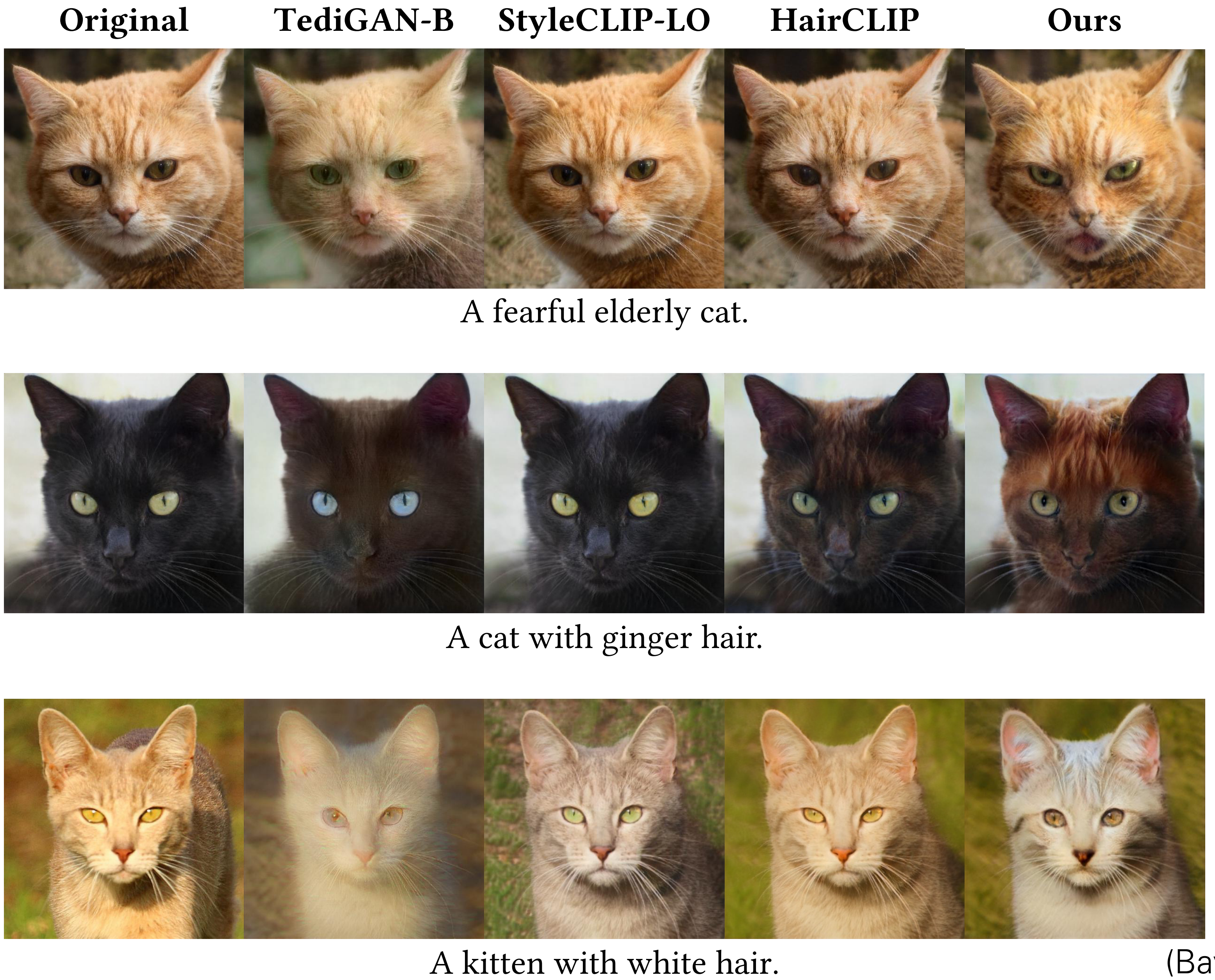
(Baykal et al., ACM TOG 2023)

CLIPInverter



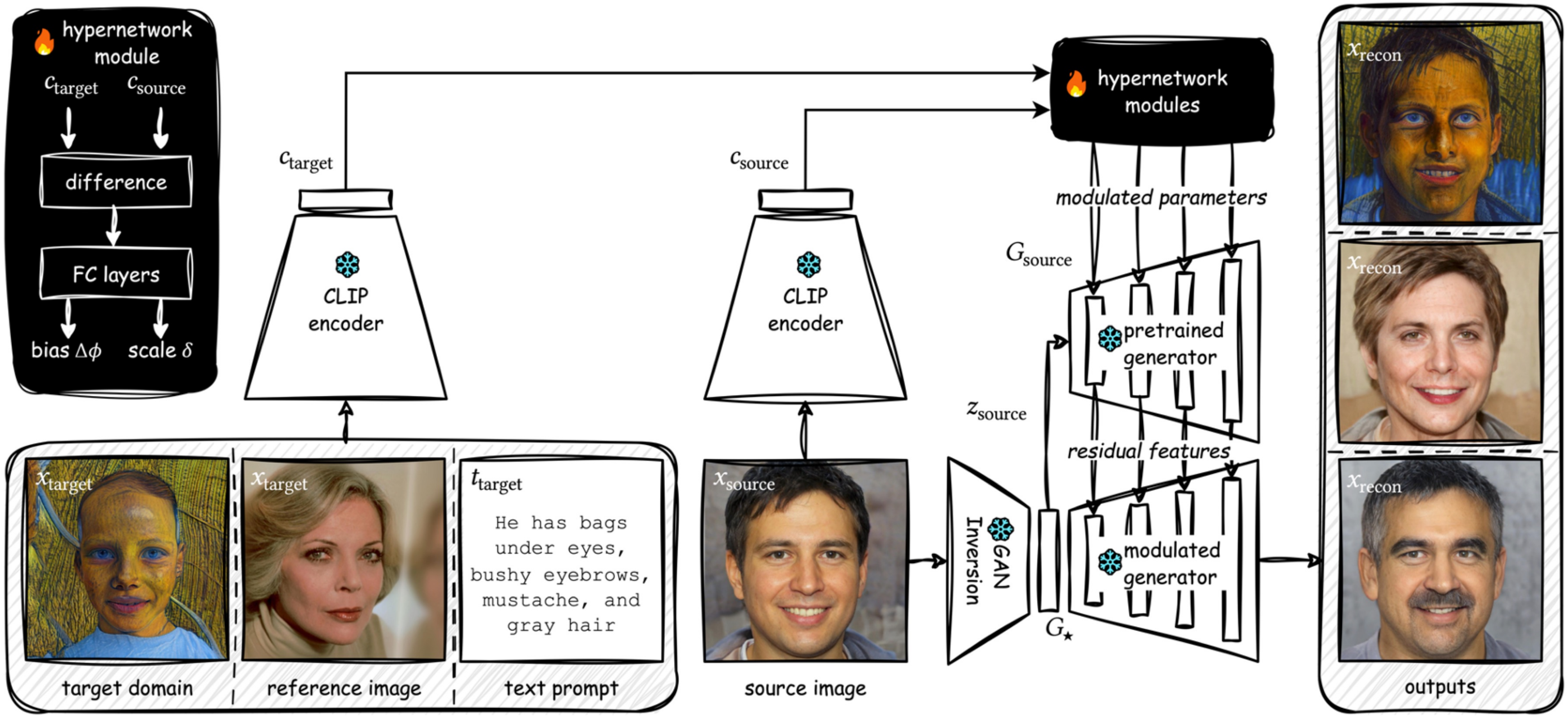
(Baykal et al., ACM TOG 2023)

CLIPInverter



(Baykal et al., ACM TOG 2023)

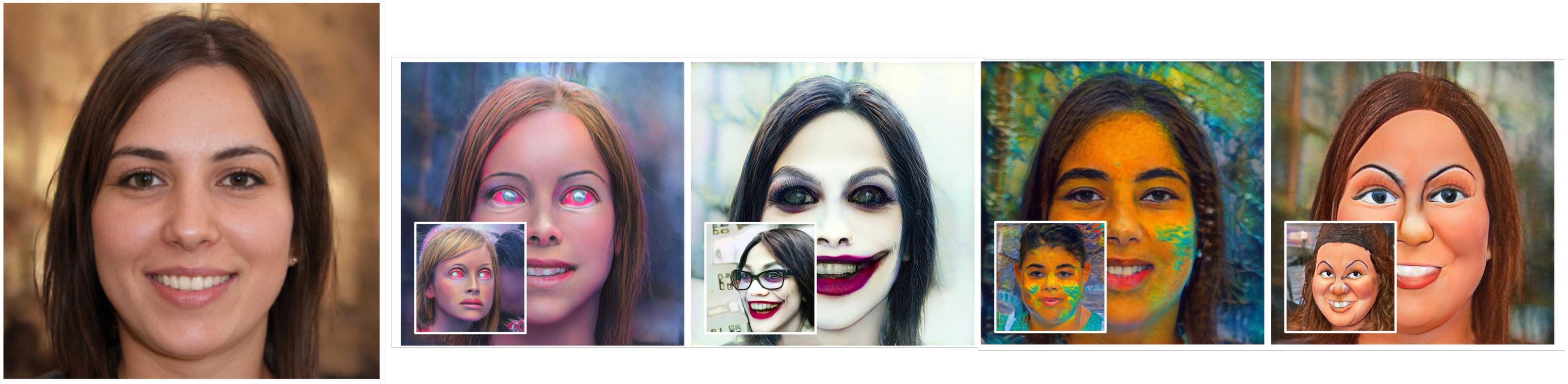
HyperGAN-CLIP



- a flexible framework that is capable of handling domain adaptation, reference-guided image synthesis and text-guided image manipulation.

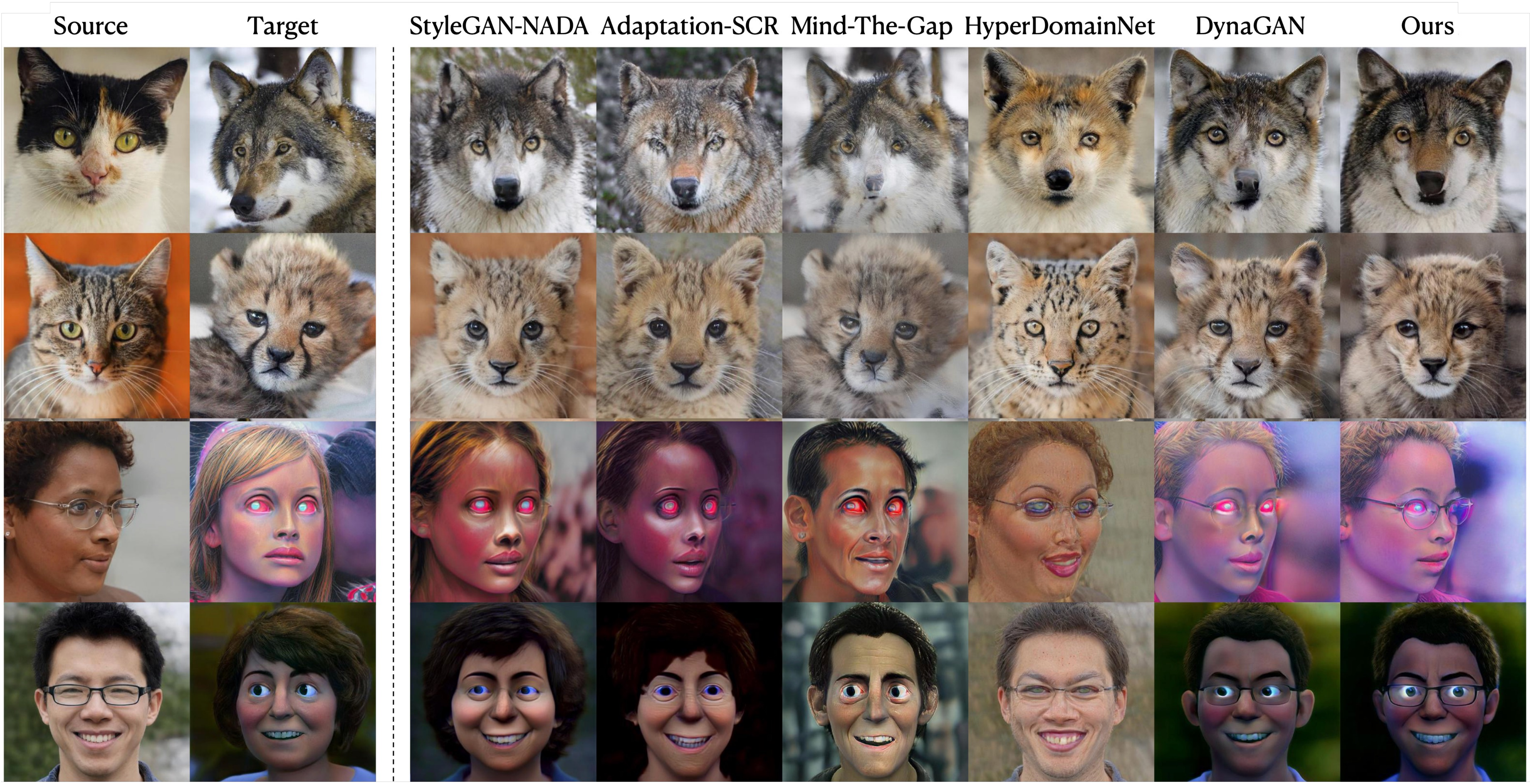
(Anees et al., SIGGRAPH Asia 2024)

HyperGAN-CLIP – Domain Adaptation



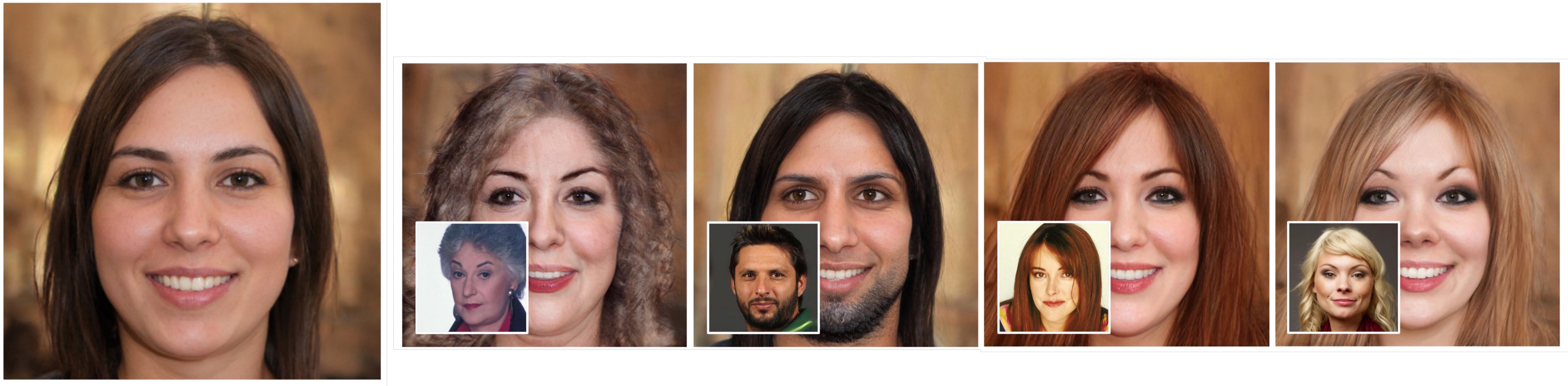
(Anees et al., SIGGRAPH Asia 2024)

HyperGAN-CLIP – Domain Adaptation



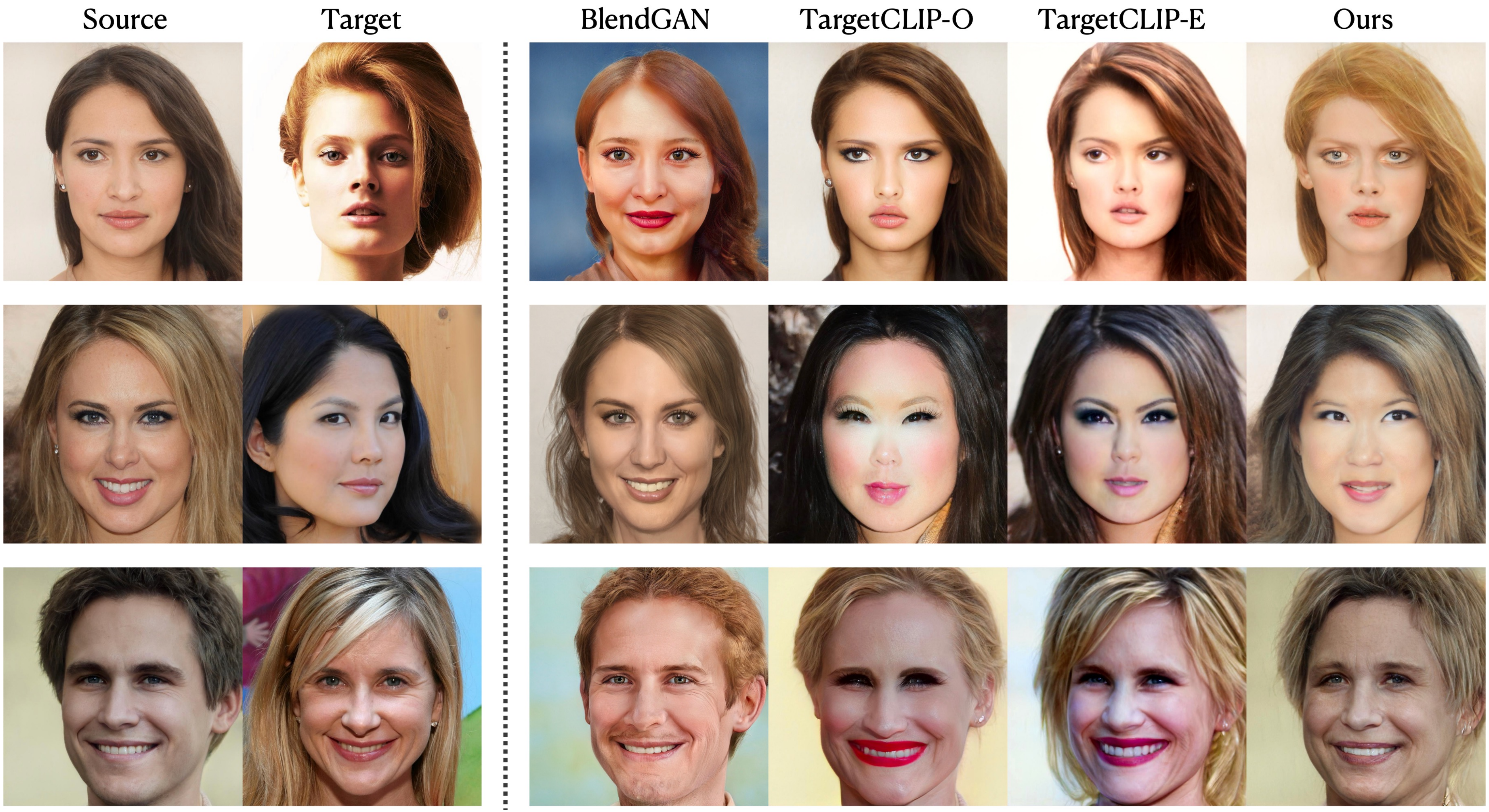
(Anees et al., SIGGRAPH Asia 2024)

HyperGAN-CLIP – Reference-Guided Image Synthesis



(Anees et al., SIGGRAPH Asia 2024)

HyperGAN-CLIP – Reference-Guided Image Synthesis

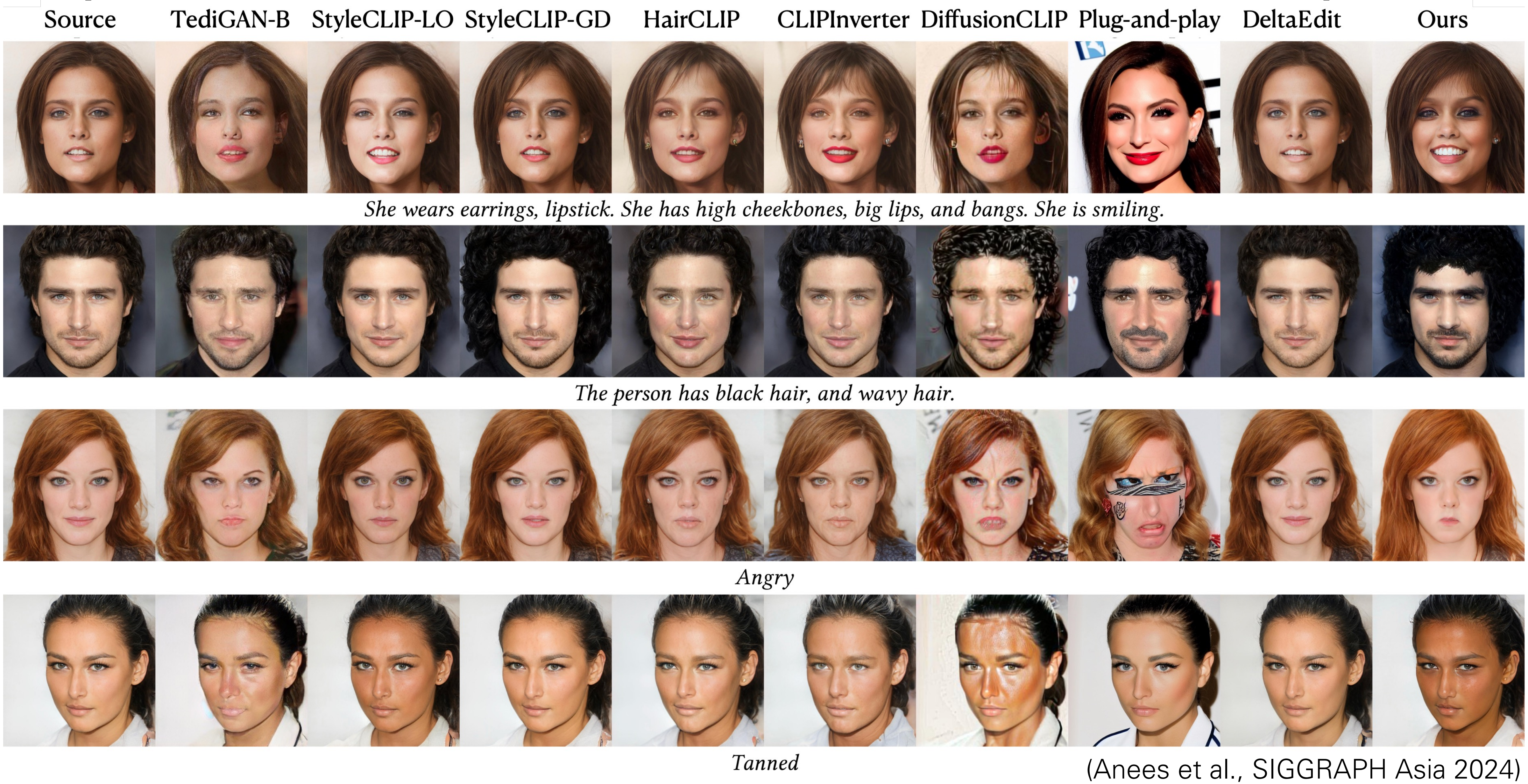


(Anees et al., SIGGRAPH Asia 2024)

HyperGAN-CLIP – Text-Guided Image Manipulation

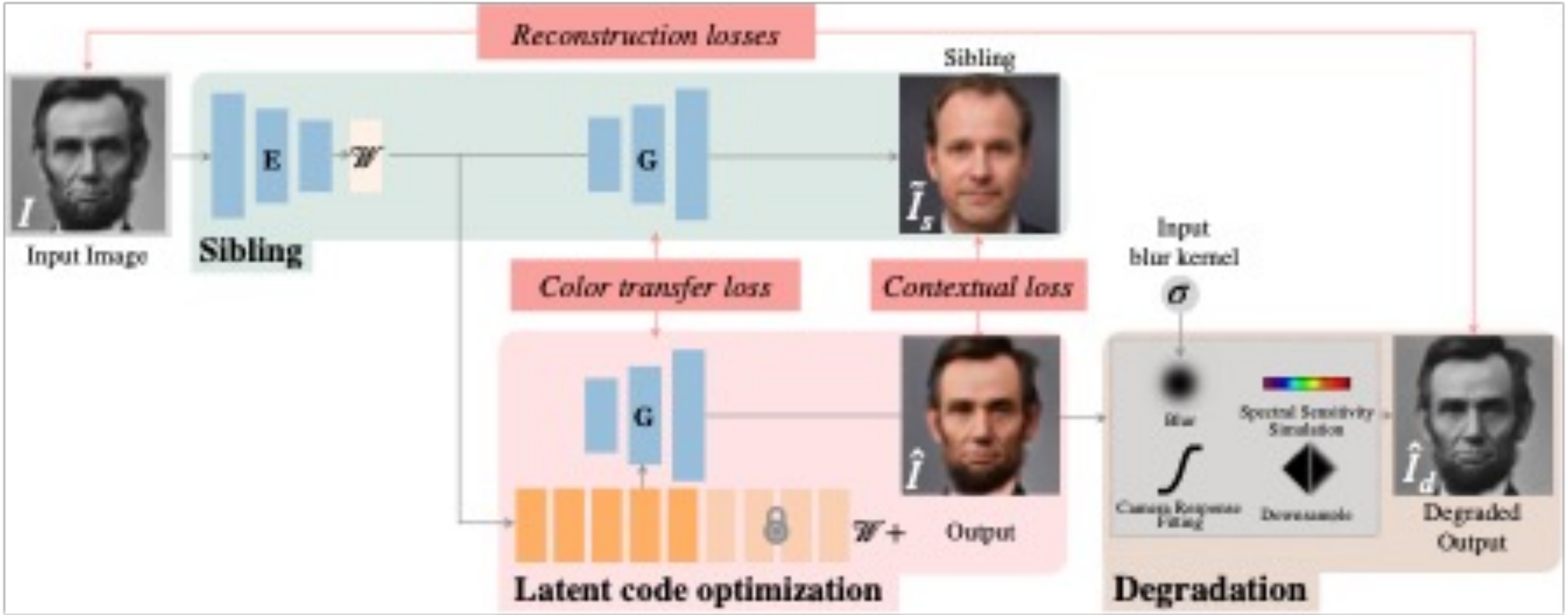
(Anees et al., SIGGRAPH Asia 2024)

HyperGAN-CLIP – Text-Guided Image Manipulation



(Anees et al., SIGGRAPH Asia 2024)

Time-travel Rephotography



- Key idea: Use the StyleGAN2 framework to project old photos into the space of modern high-resolution photos for enhancing their quality.

[Luo et al., SIGGRAPH Asia 2021]

Time-travel Rephotography

[Luo et al., SIGGRAPH Asia 2021]

Time-travel Rephotography

Input | Ours | DeOldify | InstColorization | Zhang | Zhang (FFHQ)

[Luo et al., SIGGRAPH Asia 2021]

Next Lecture:

Deep Generative Models
Part 3