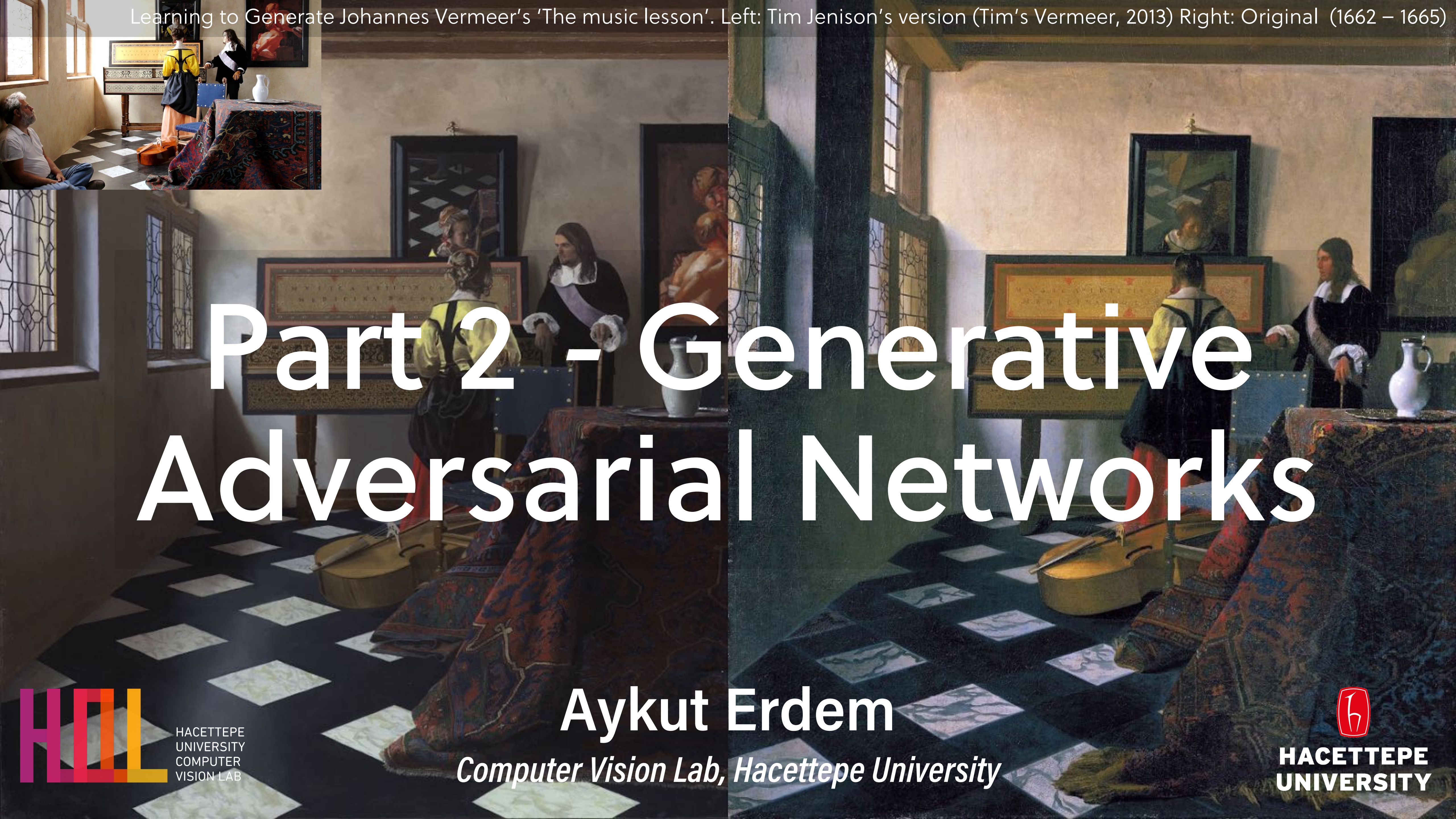
Learning to Generate Johannes Vermeer's 'The music lesson'. Left: Tim Jenison's version (Tim's Vermeer, 2013) Right: Original (1662 – 1665)



Part 2 - Generative Adversarial Networks

Aykut Erdem

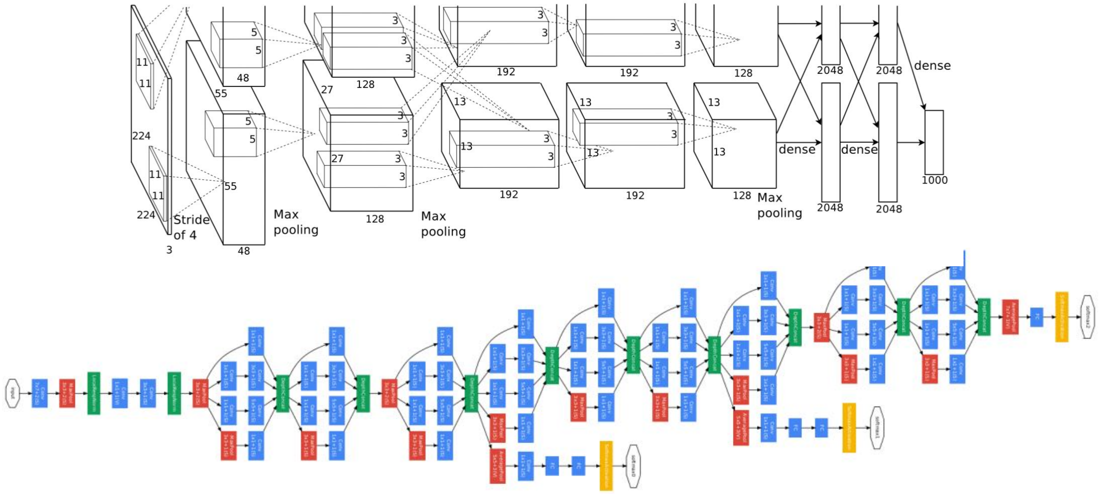
Computer Vision Lab, Hacettepe University

HACETTEPE UNIVERSITY COMPUTER VISION LAB

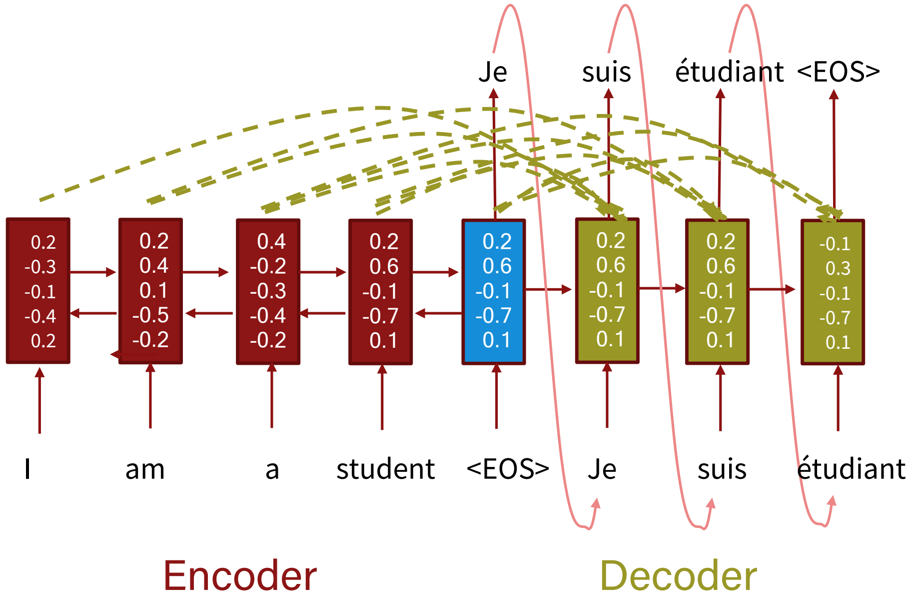
HACETTEPE UNIVERSITY

Deep Supervised Learning: A Success Story

- Obtain lots of input-output examples
- Train a deep neural network



Deep CNN



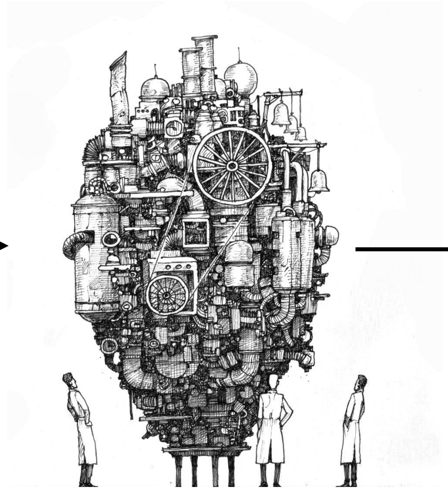
RNN with attention

- Achieve superior results

Discriminative vs. Generative Models

$$p(y|x)$$

$$p(x|y)$$



"Cat"

Discriminative models

Generative models

Why study deep generative models?

- Go beyond associating inputs to outputs
- Understand high-dimensional, complex probability distributions
- Discover the “true” structure of the data
    - Detect surprising events in the world (*anomaly detection*)
    - Missing Data (*semi-supervised learning*)
    - Generate models for planning (*model-based reinforcement learning*)

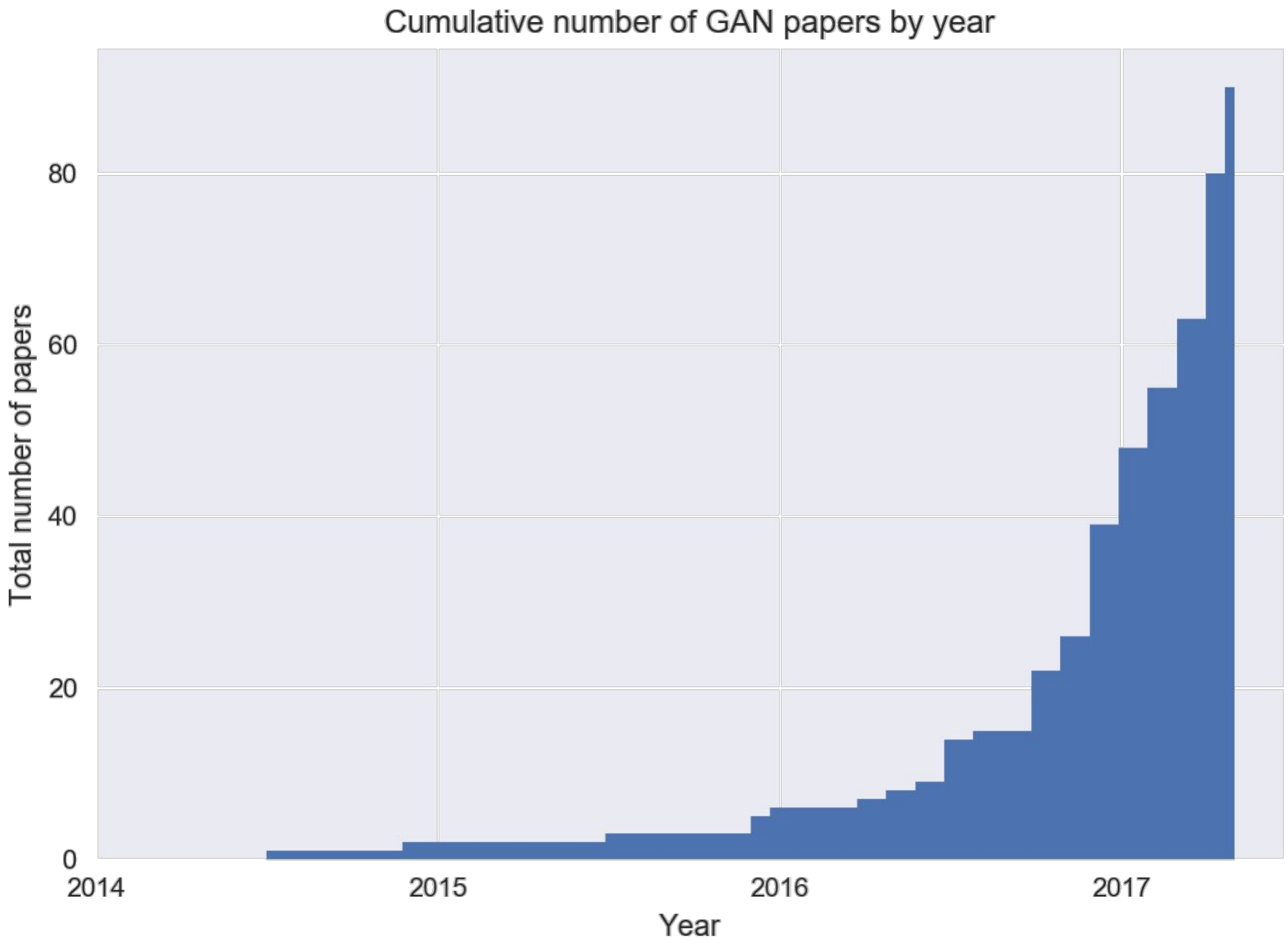
Why study Generative Adversarial Networks?

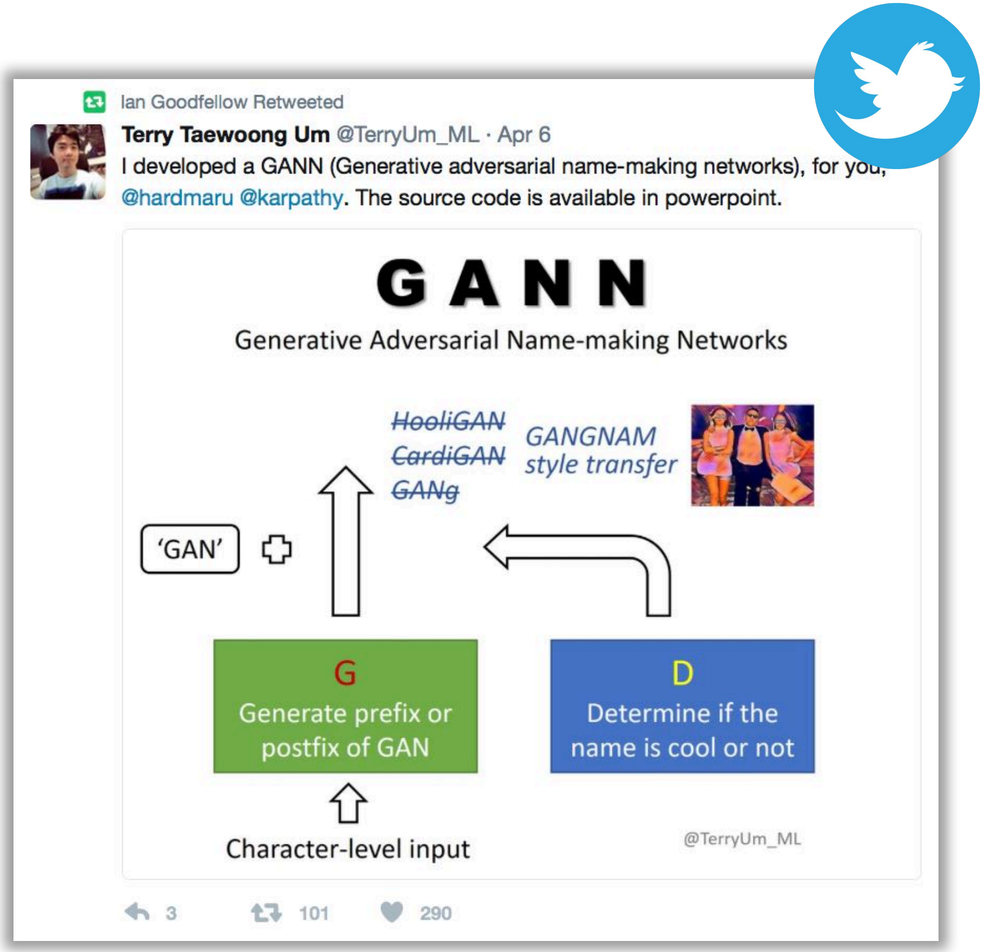
Q: What are some recent and potentially upcoming breakthroughs in deep learning?

A: The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks) ... This, and the variations that are now being proposed is **the most interesting idea in the last 10 years in ML**, in my opinion.

Quora Session with Yann LeCun
July 29, 2016

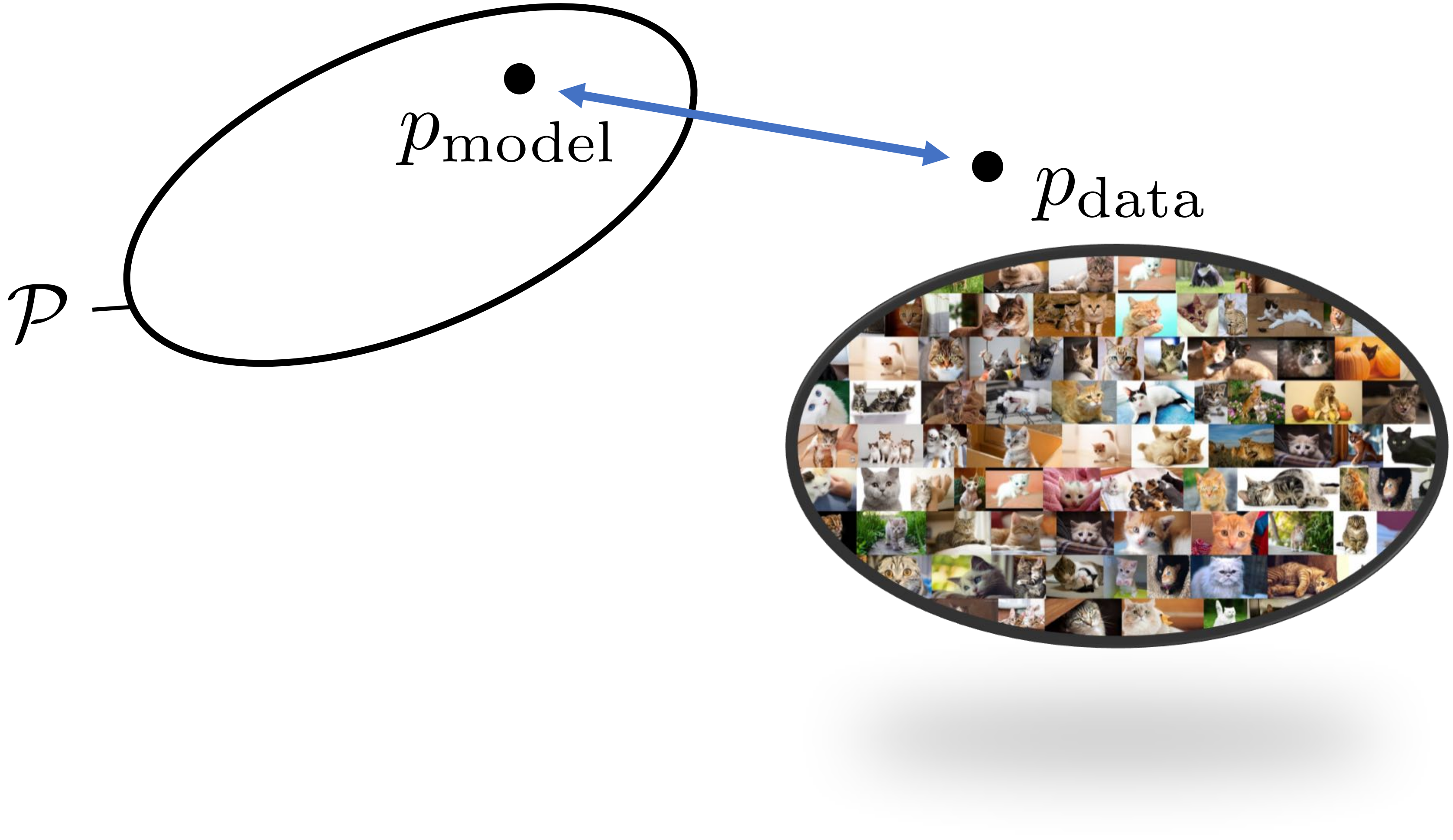
Progress in GANs



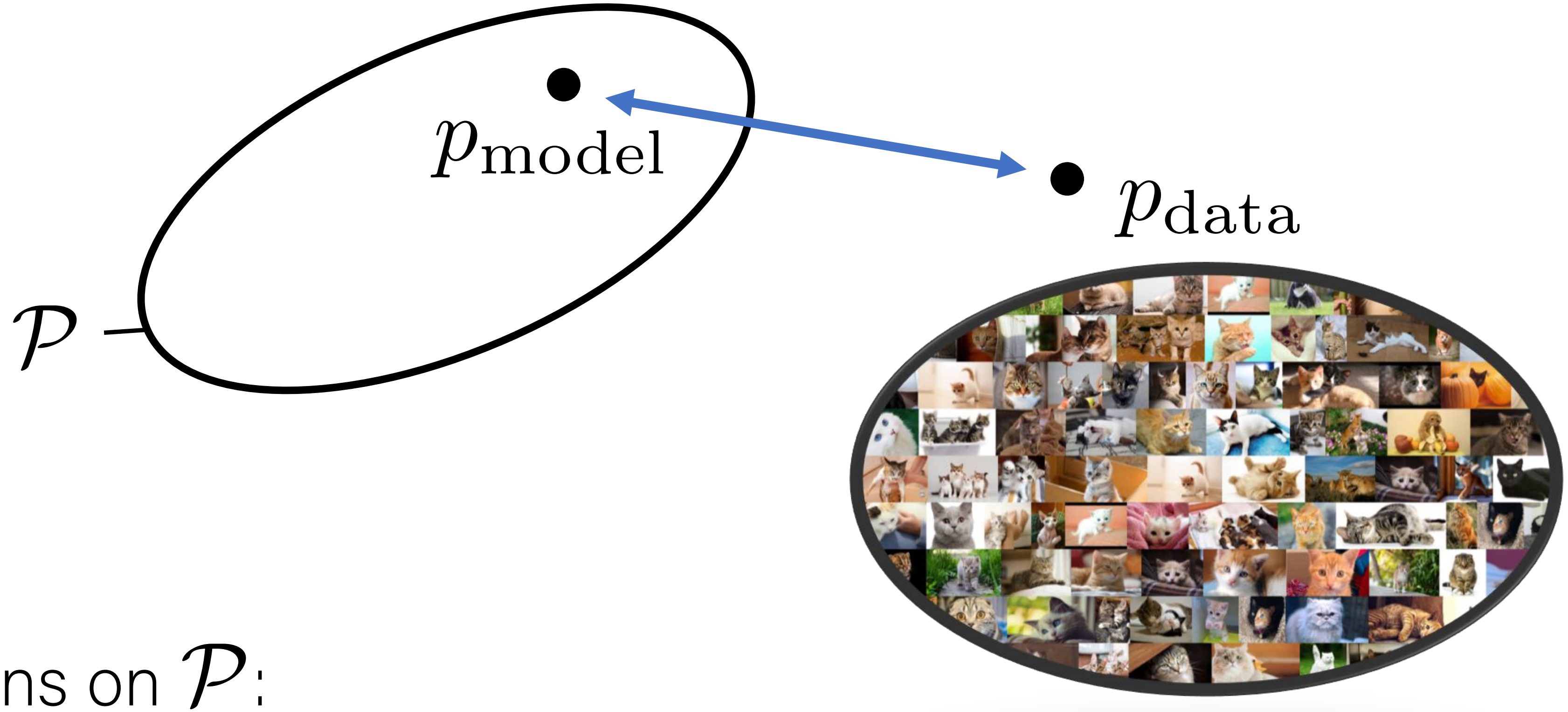


Source: https://deephunt.in/the-gan-zoo-79597dc8c347

Generative Modeling



Generative Modeling



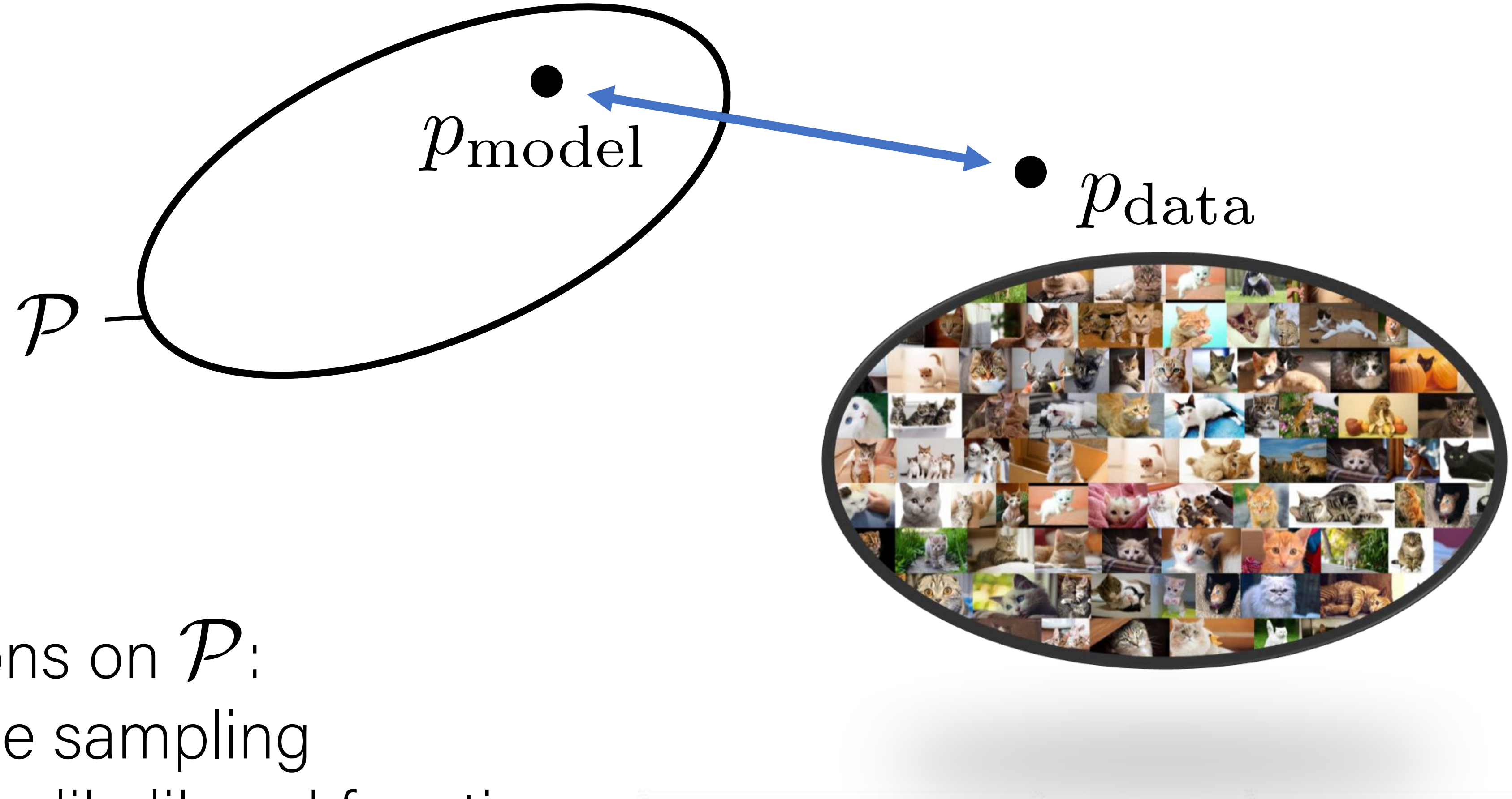
Assumptions on $\mathcal{P}$:

- tractable sampling

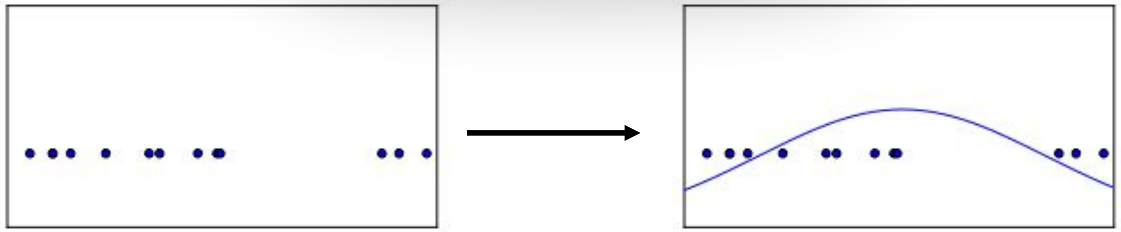
Training examples

Model samples

Generative Modeling



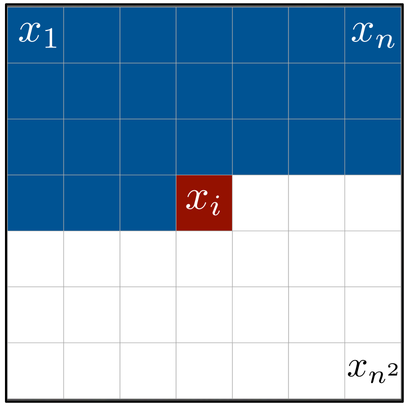
Assumptions on $\mathcal{P}$:
- tractable sampling
- tractable likelihood function



Three Broad Categories

- Autoregressive Models
- Variational Autoencoders
- Generative Adversarial Networks (GANs)

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:

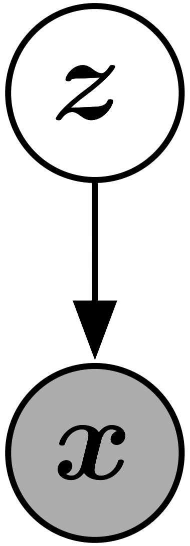
- Generation can be too costly
- Generation can not be controlled by a latent code

PixelCNN elephants
(van den Ord et al. 2016)

Variational Autoencoder

x → encoder → z → decoder → x̂

- Maximizes a variational lower bound on log-likelihood of $\boldsymbol{x}$



$$\log p(\boldsymbol{x}) \geq \log p(\boldsymbol{x}) - D_{\mathrm{KL}}\left(q(\boldsymbol{z}) \| p(\boldsymbol{z} \mid \boldsymbol{x})\right)$$
$$= \mathbb{E}_{\boldsymbol{z} \sim q} \log p(\boldsymbol{x}, \boldsymbol{z}) + H(q)$$

Face samples for
Labeled Faces in the Wild (LFW)
(Alec Radford)

Disadvantages:

- Not asymptotically consistent unless q is perfect
- Tends to produce blurry samples

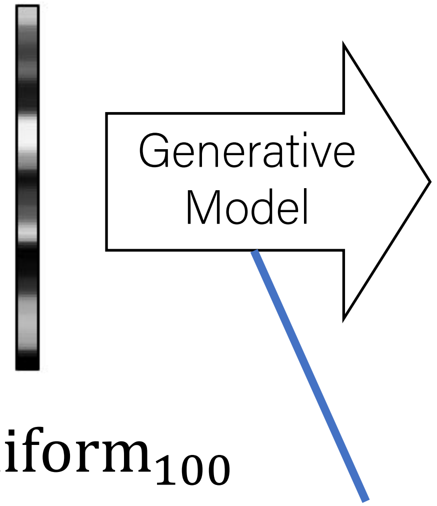
GANs

Generative
Adversarial
Networks

Genetive Adversarial Networks (GANs)

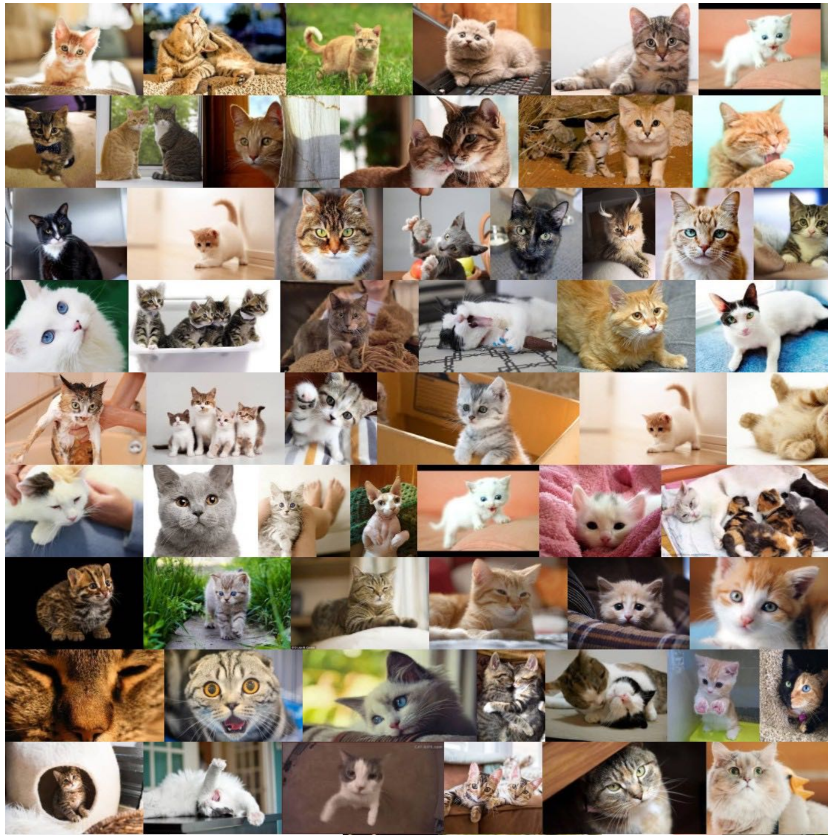
(Goodfellow et al., 2014)

Noise
(random input)



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

think of this as
a transformation



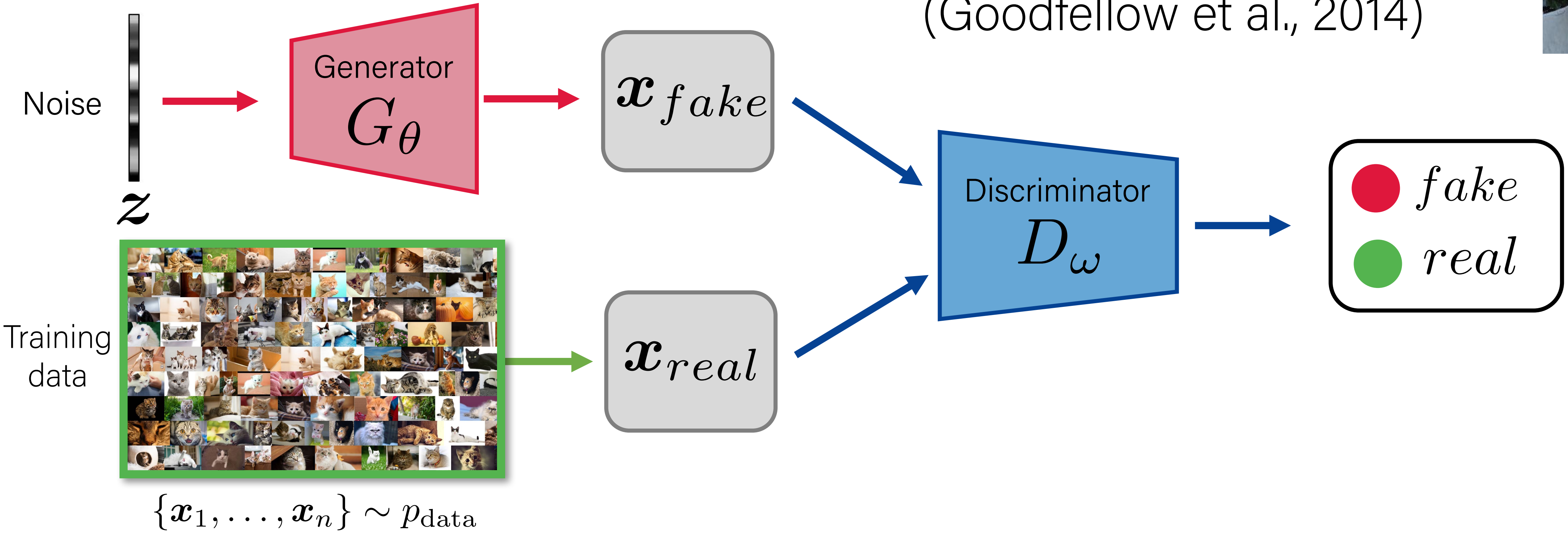
- A game-theoretic likelihood free model

Advantages:

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

Genetive Adversarial Networks (GANs)

(Goodfellow et al., 2014)

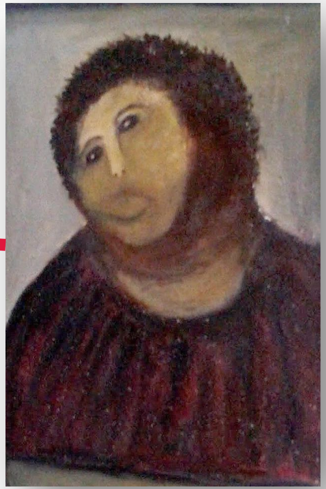


- A game between a generator $G_\theta(\boldsymbol{z})$ and a discriminator $D_\omega(\boldsymbol{x})$
  - Generator tries to fool discriminator (i.e. generate realistic samples)
  - Discriminator tries to distinguish fake from real samples

Intuition behind GANs

D_ω: Discriminator (*Art Critic*)

$\boldsymbol{x}_{real}$



$\boldsymbol{x}_{fake}$

G_θ: Generator (*Forger*)

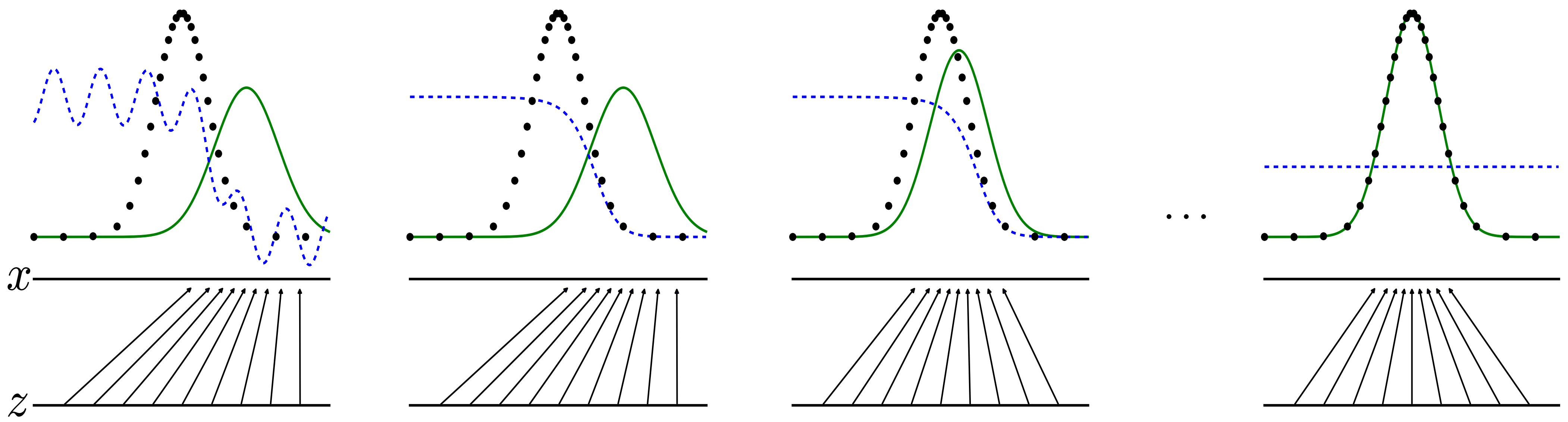
Intuition behind GAN Training

https://www.youtube.com/watch?v=No26JKQKZNE

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

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

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

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

ross-entropy
ss for binary
assification

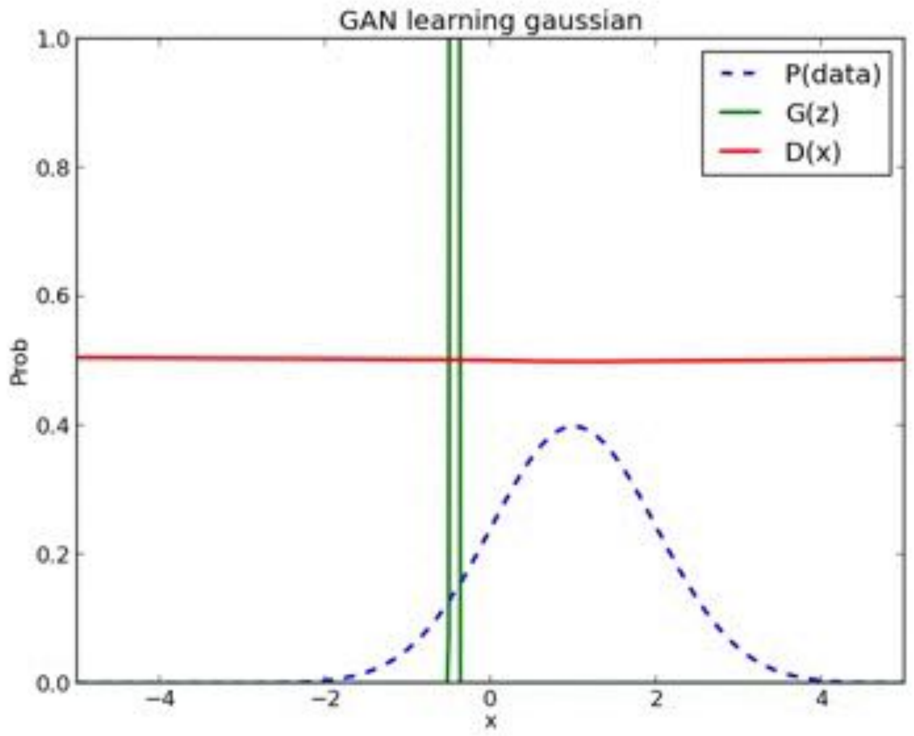
probability
of the discriminator being mistaken

Important question is "Does this converge??"

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

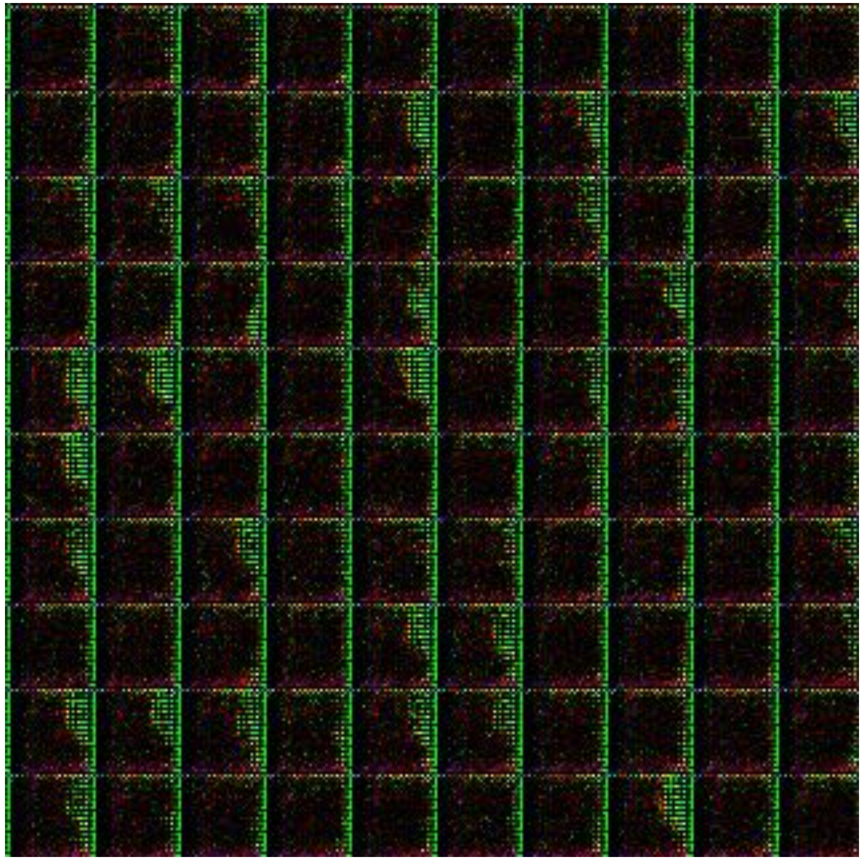
Training Procedure

(Goodfellow et al., 2014)



Source: Alec Radford

Generating 1D points



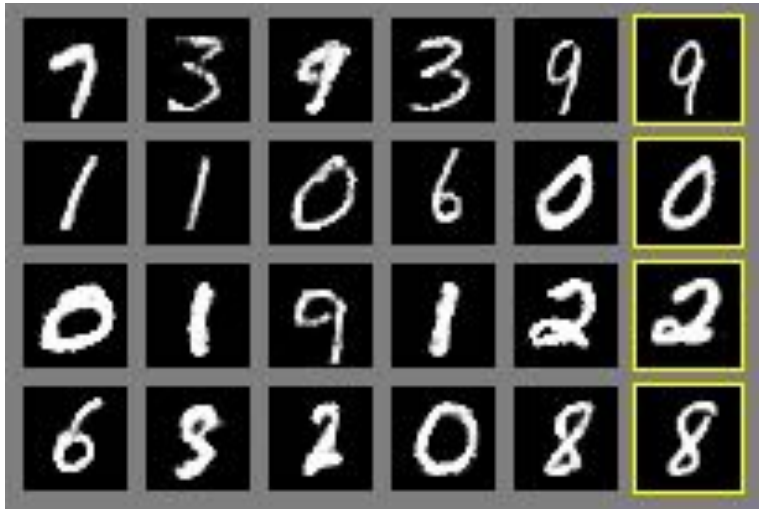
Source: OpenAI blog

Generating images

Results

(Goodfellow et al., 2014)

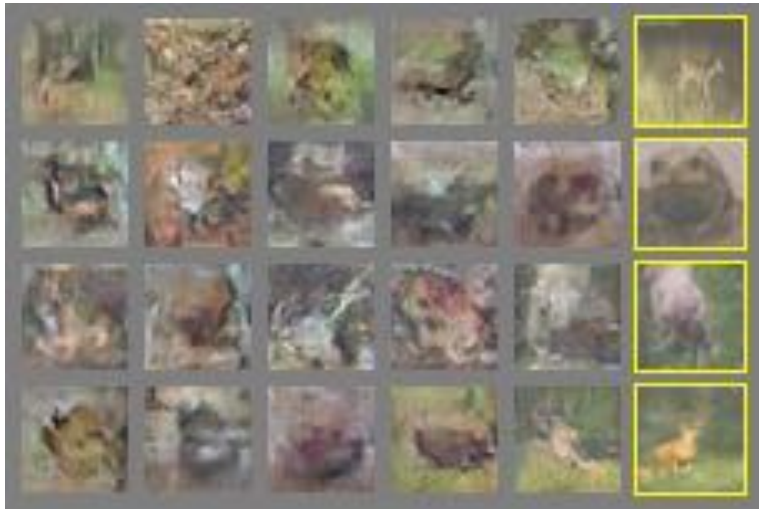
- The generator uses a mixture of rectifier linear activations and/or sigmoid activations
- The discriminator net used maxout activations.



MNIST samples

TFD samples

CIFAR10 samples
(*fully-connected model*)

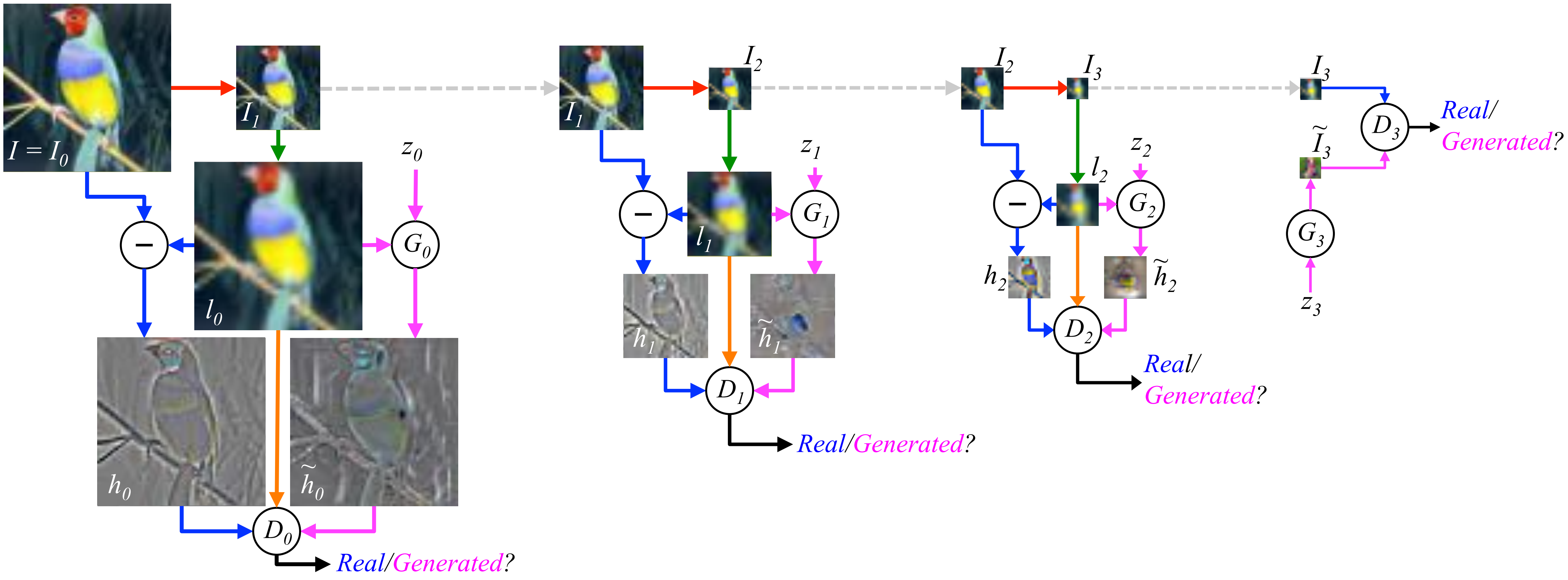


CIFAR10 samples
(*convolutional discriminator, deconvolutional generator*)

Laplacian GANs (LAPGAN)

(Denton et al., 2015)

- **Idea:** Combine GAN with a multi-scale image representation (Laplacian pyramid)



LAPGAN for LSUN Towers

64×64 pixels
~700K images

(Denton et al., 2015)

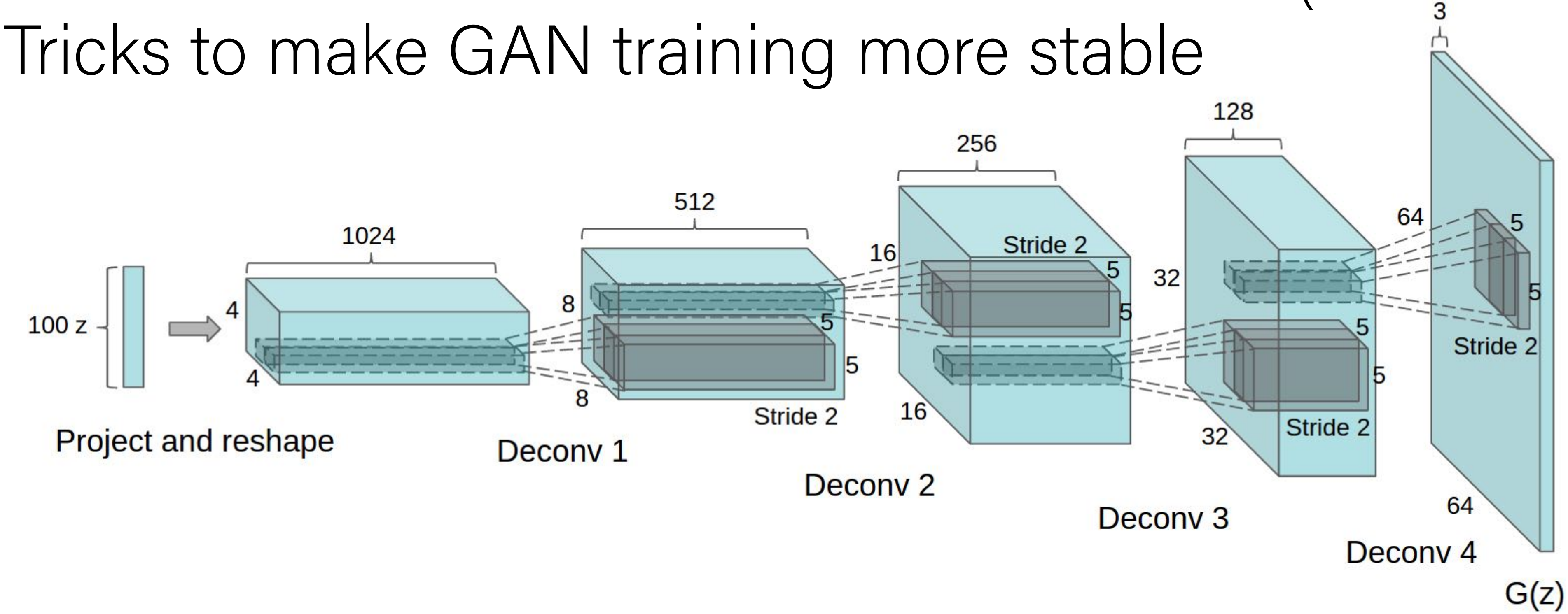
LAPGAN for LSUN Bedrooms

64×64 pixels
~3M images
(Denton et al., 2015)

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

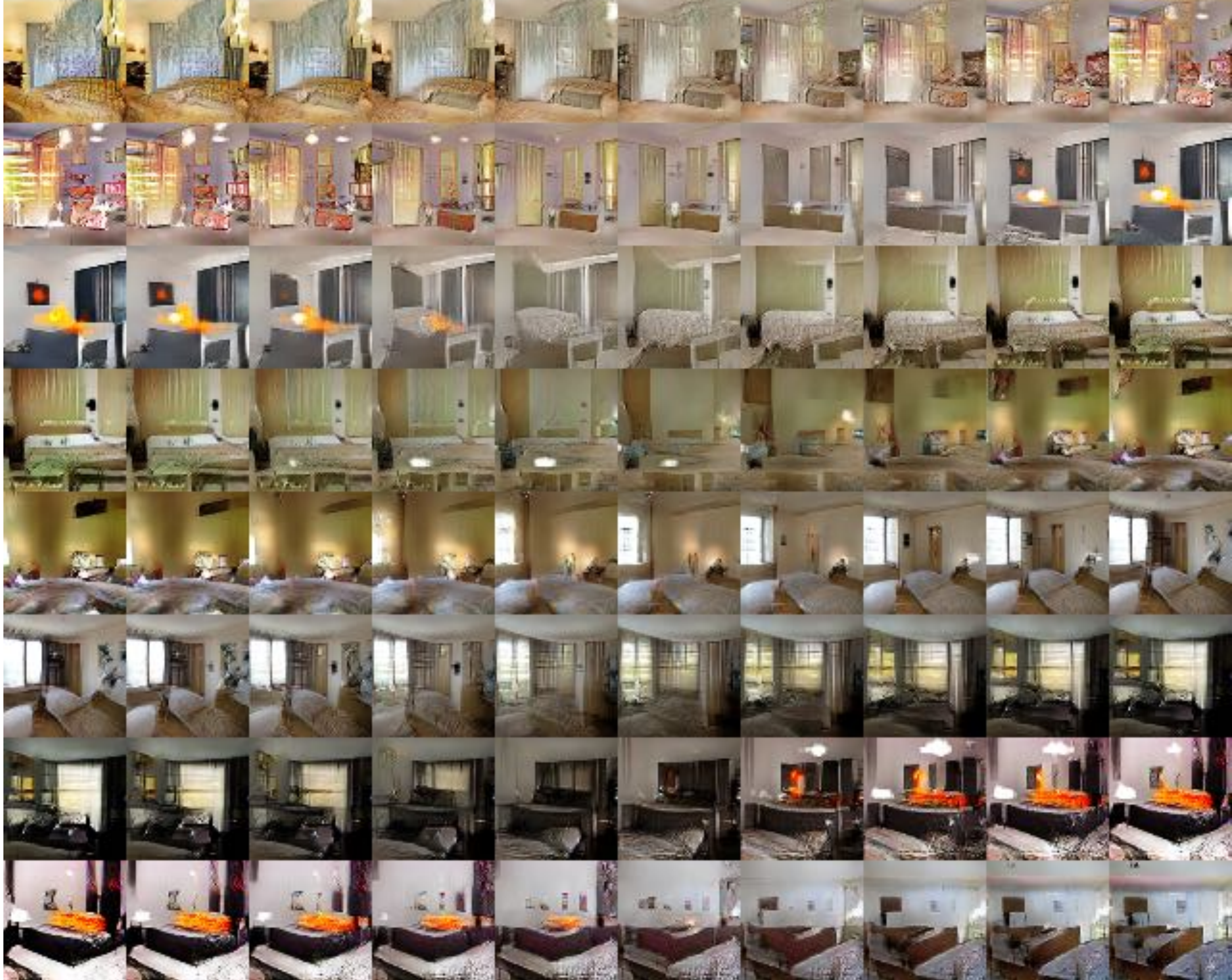
64×64 pixels
~3M images

(Radford et al., 2015)

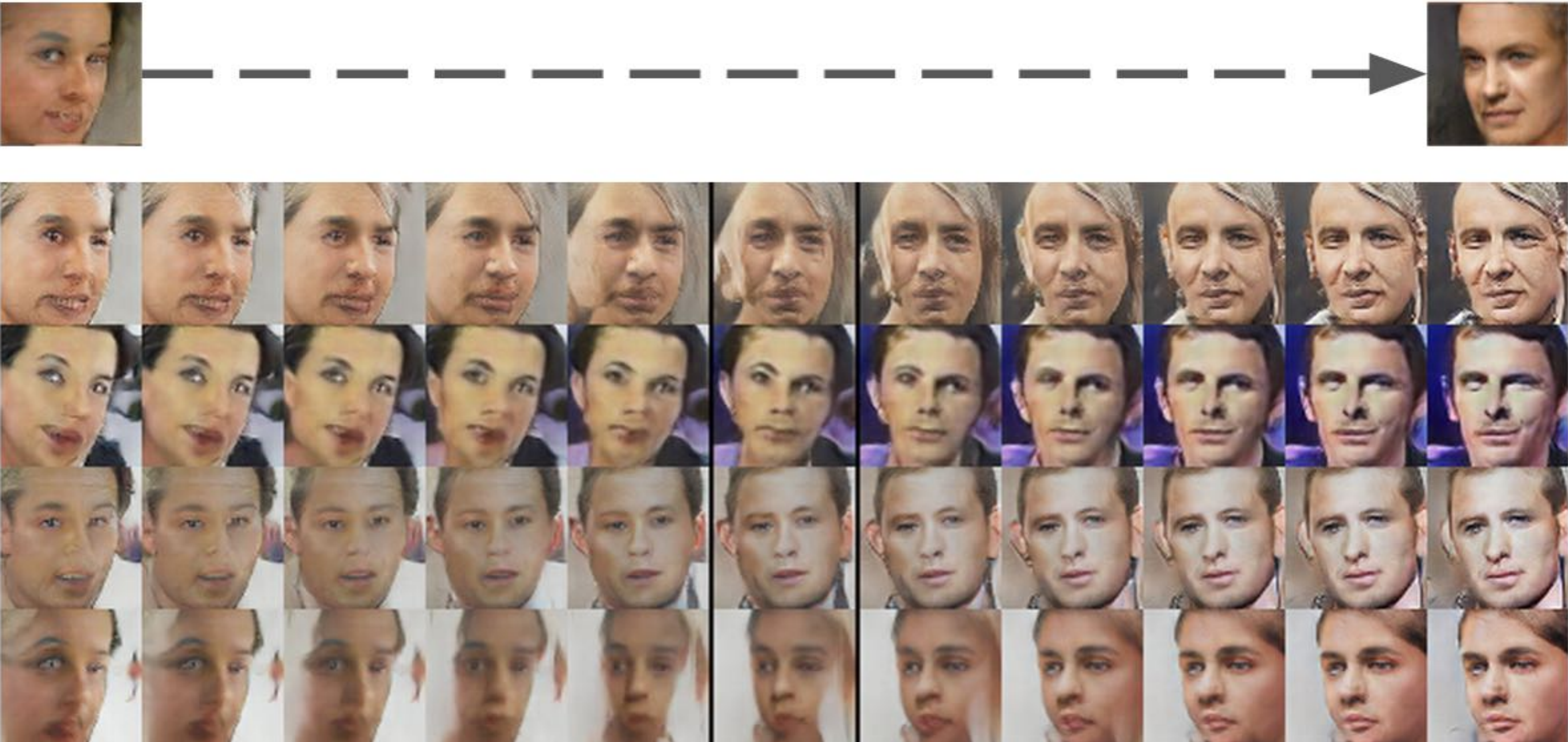
Walking over the latent space

(Radford et al., 2015)

- Interpolation suggests non-overfitting behavior

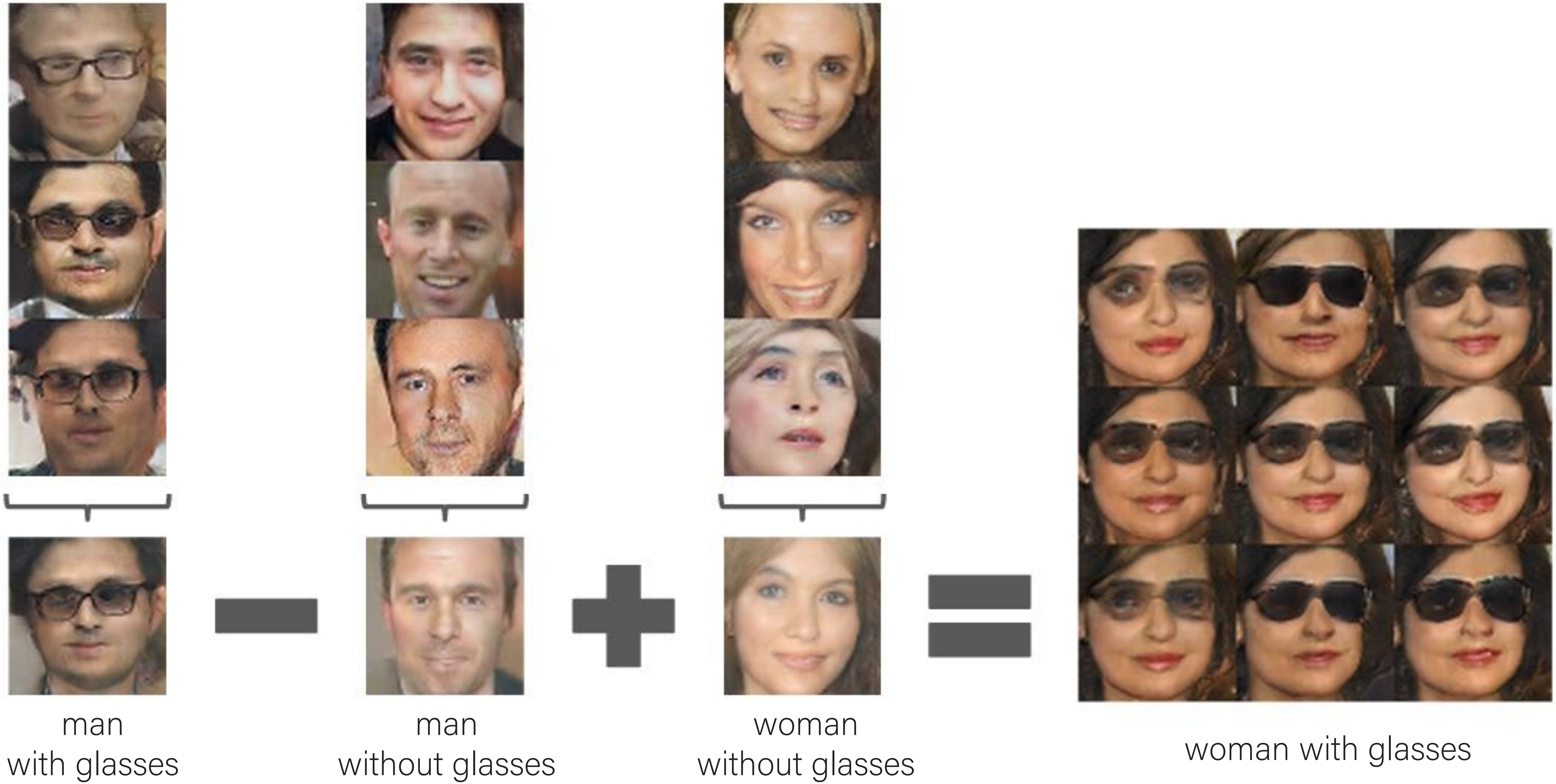


Walking over the latent space (Radford et al., 2015)



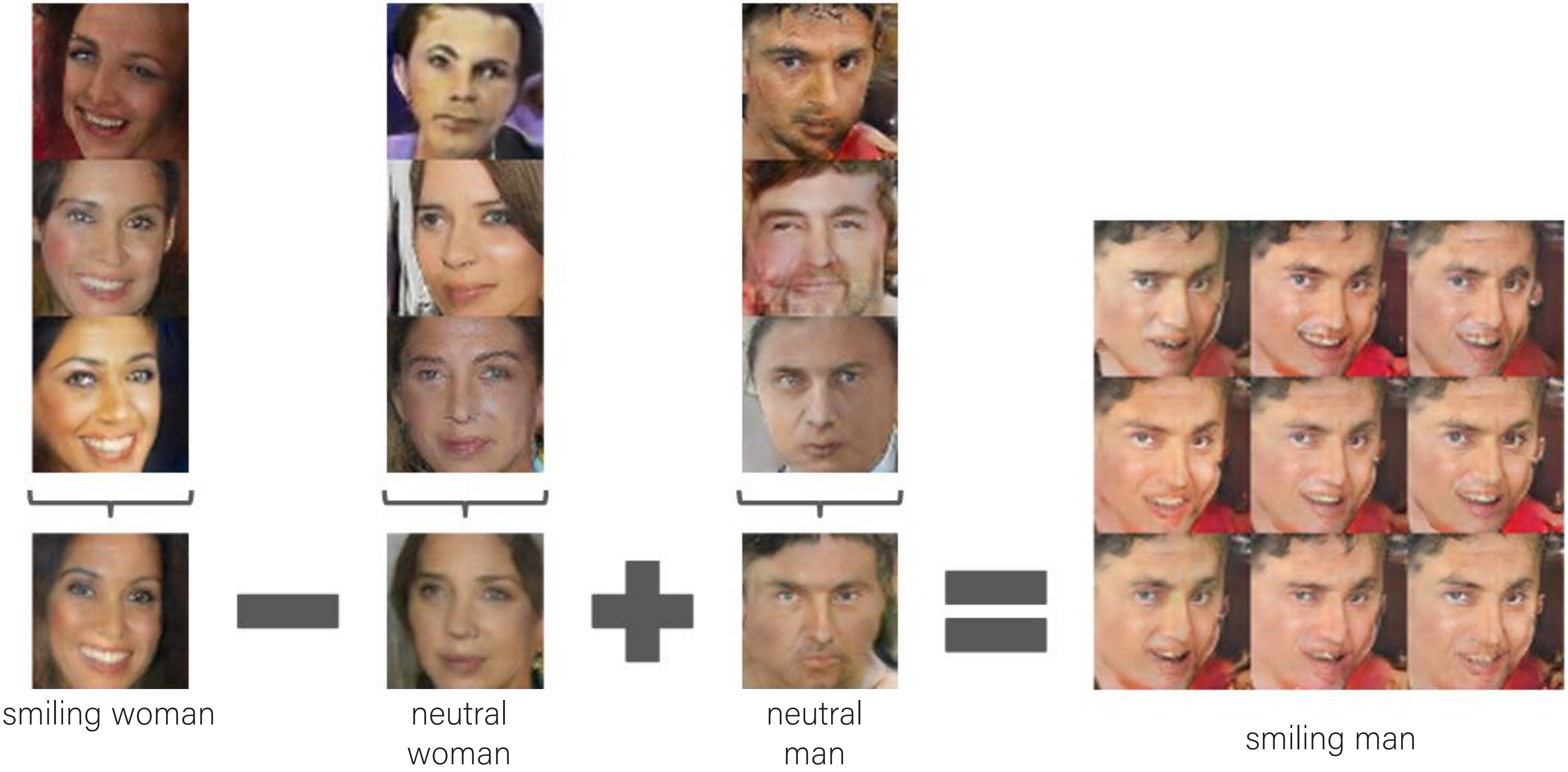
Vector Space Arithmetic

(Radford et al., 2015)



Vector Space Arithmetic

(Radford et al., 2015)



Subclasses of GANs

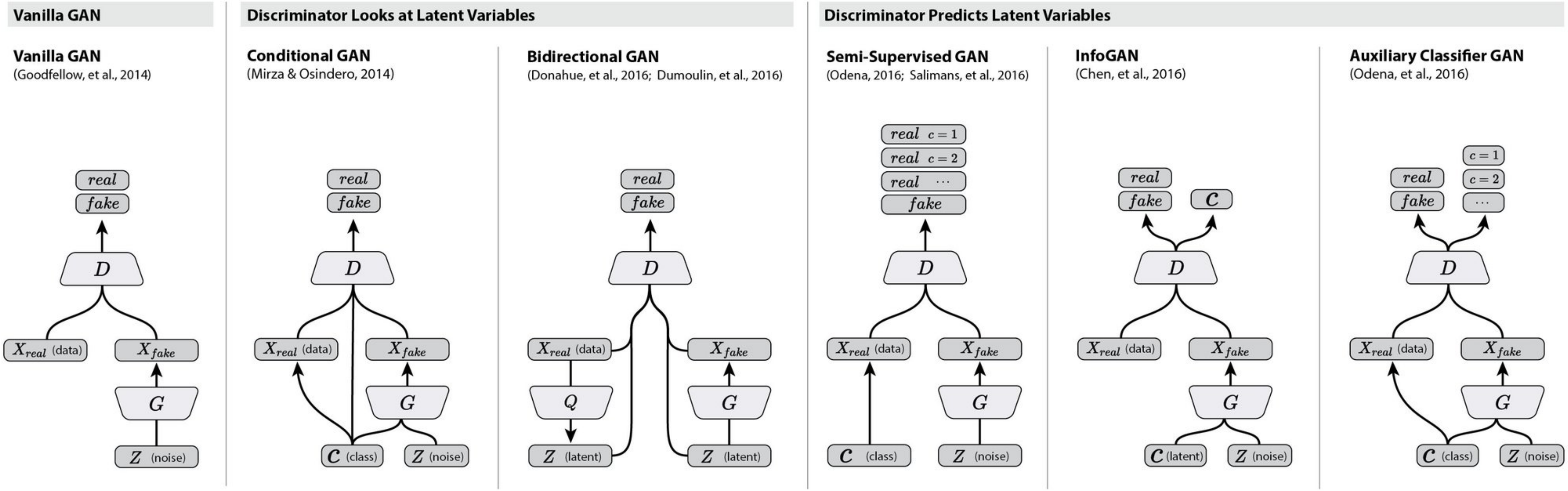
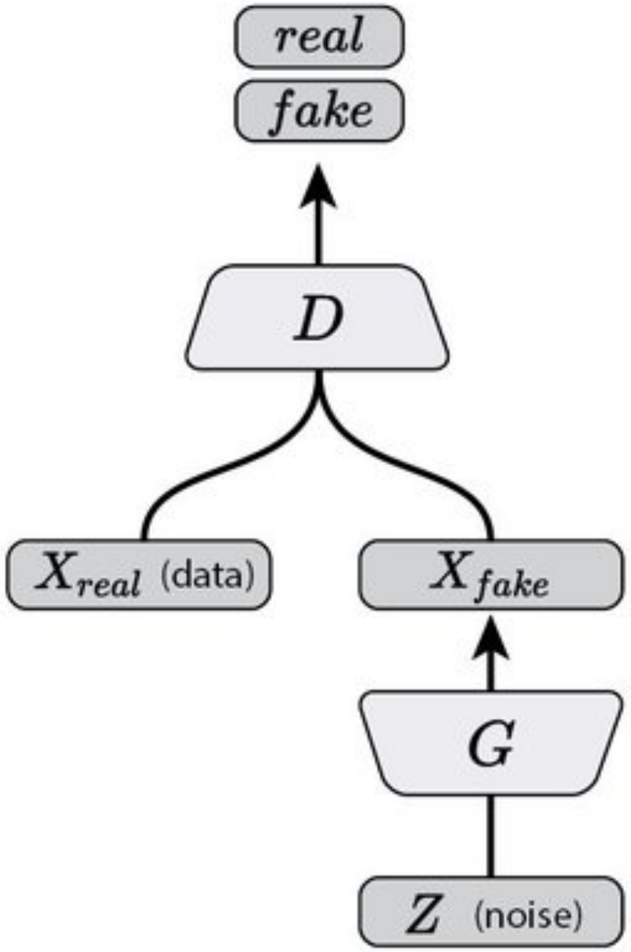
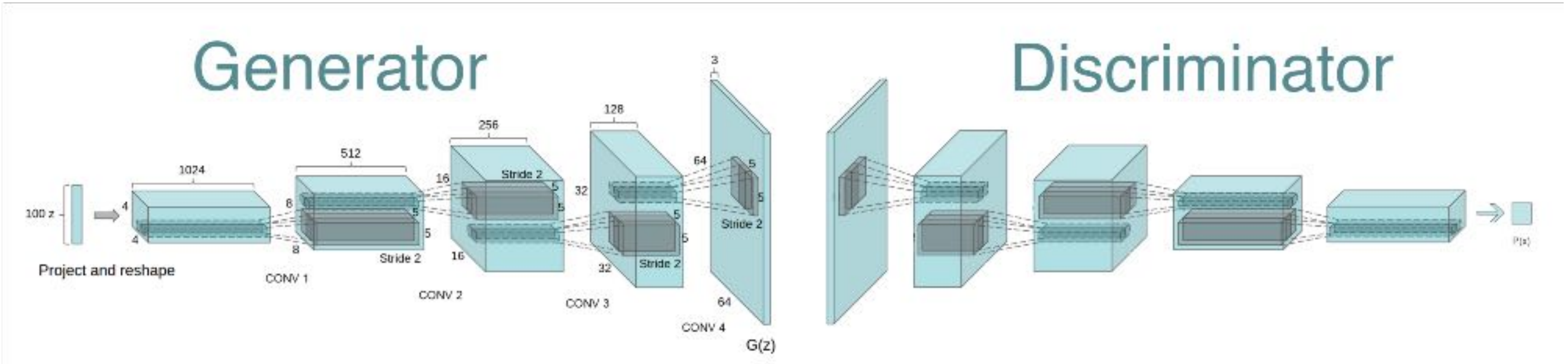


Image: Christopher Olah

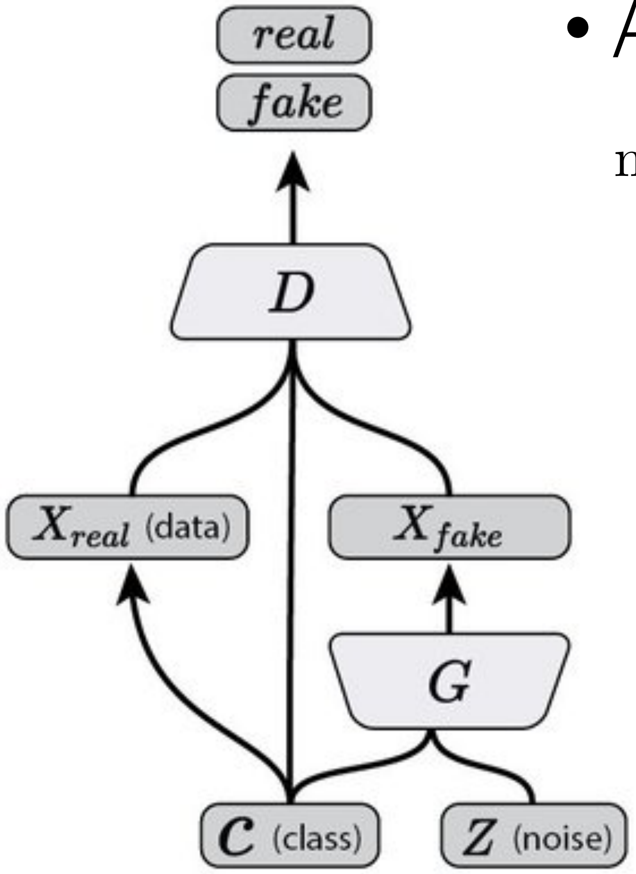
Vanilla GAN (Goodfellow et al., 2014)





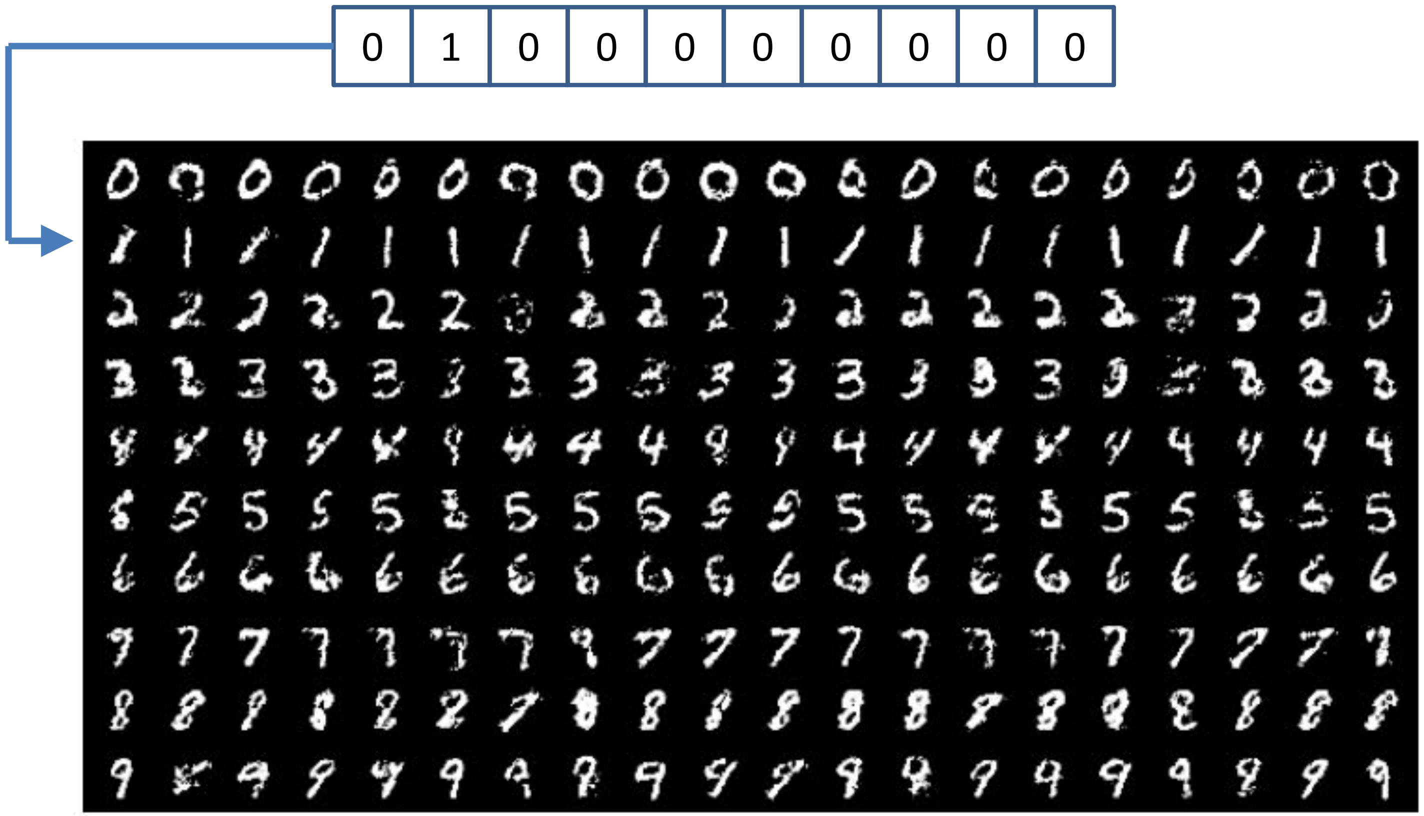
DCGAN (Radford et al., 2015)

Conditional GAN (Mirza and Osindero, 2014)

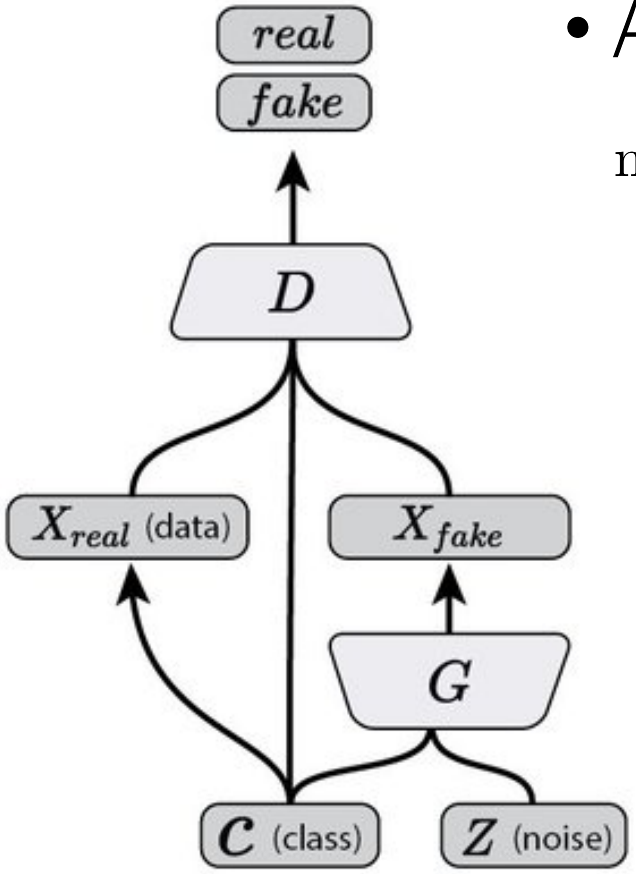


- Add conditional variables $\boldsymbol{y}$ into G and D

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



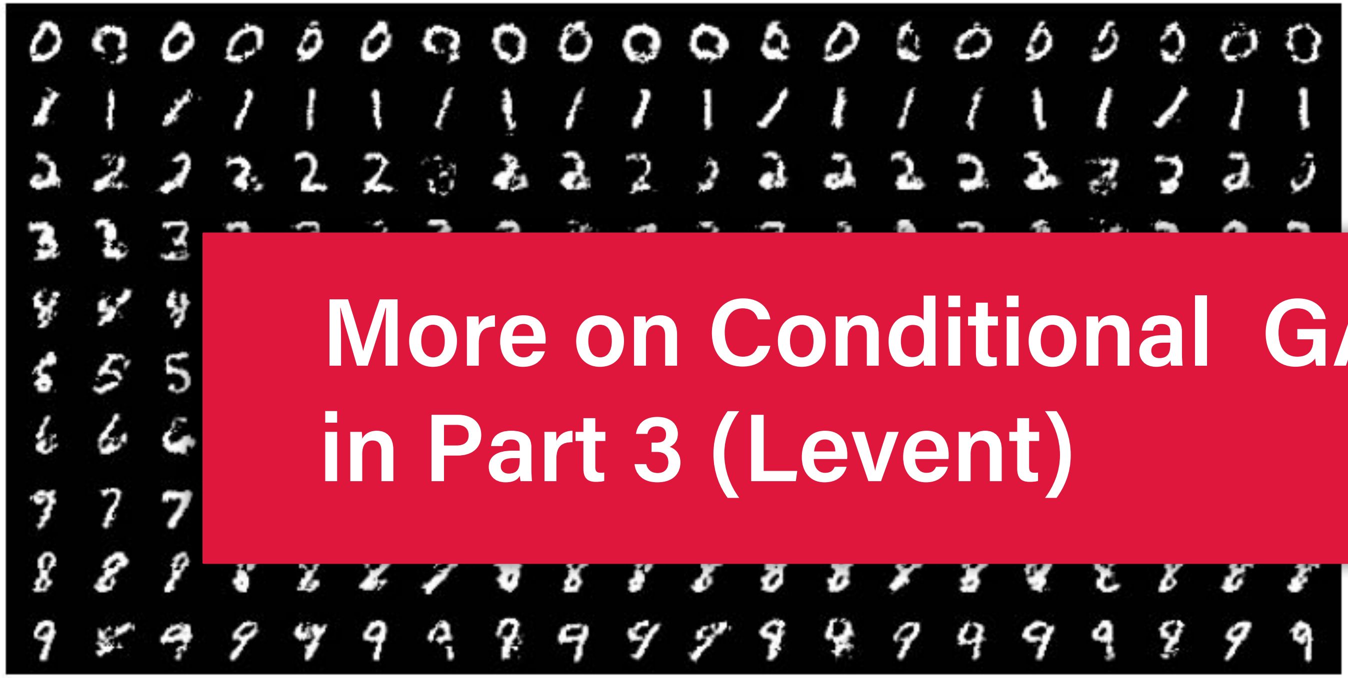
Conditional GAN (Mirza and Osindero, 2014)



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

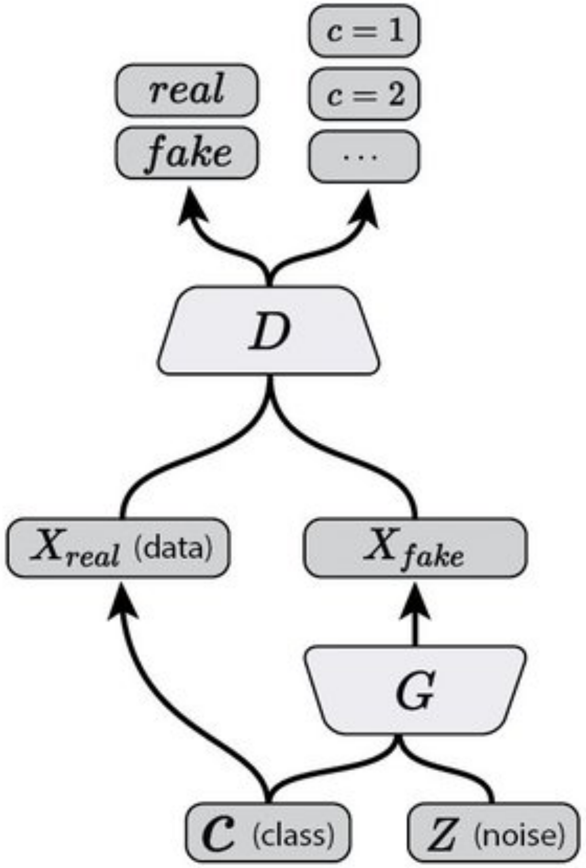
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$$

0	1	0	0	0	0	0	0	0	0



More on Conditional GANs in Part 3 (Levent)

Auxiliary Classifier GAN (Odena et al., 2016)



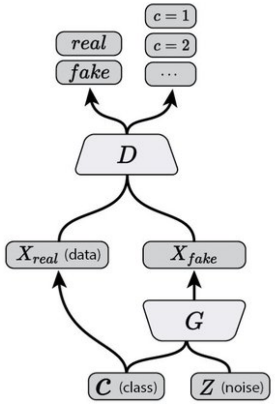
- Every generated sample has a corresponding class label

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

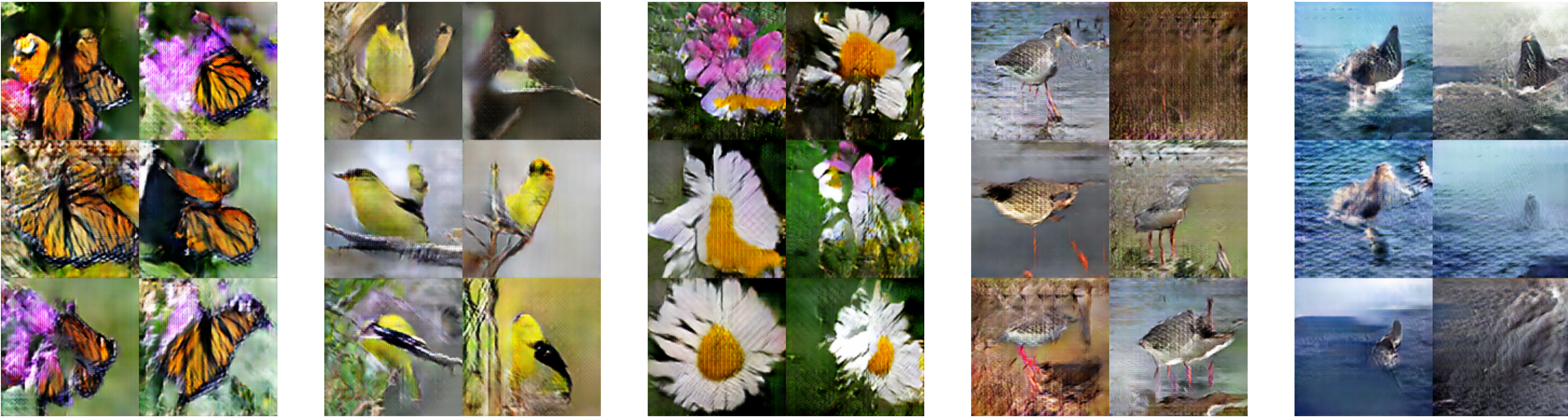
- D is trained to maximize $L_S + L_C$
- G is trained to maximize $L_C - L_S$

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

Auxiliary Classifier GAN (Odena et al., 2016)

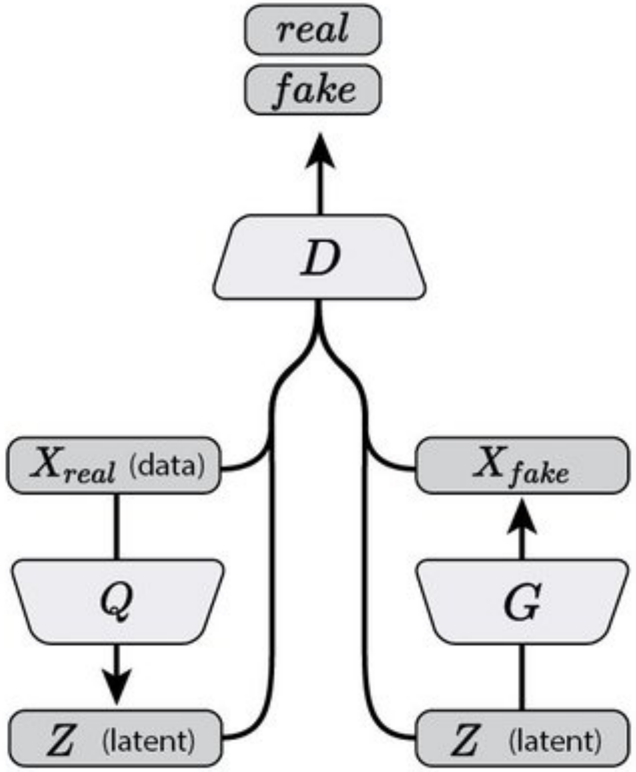


128×128 resolution samples from 5 classes taken from an AC-GAN trained on the ImageNet



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.

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

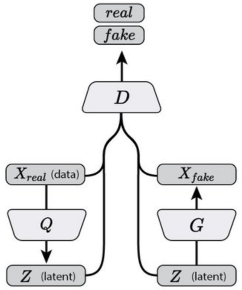
$$= \iint q(\boldsymbol{x})q(\boldsymbol{z} \mid \boldsymbol{x}) \log(D(\boldsymbol{x}, \boldsymbol{z})) d\boldsymbol{x} d\boldsymbol{z}$$

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

CelebA reconstructions

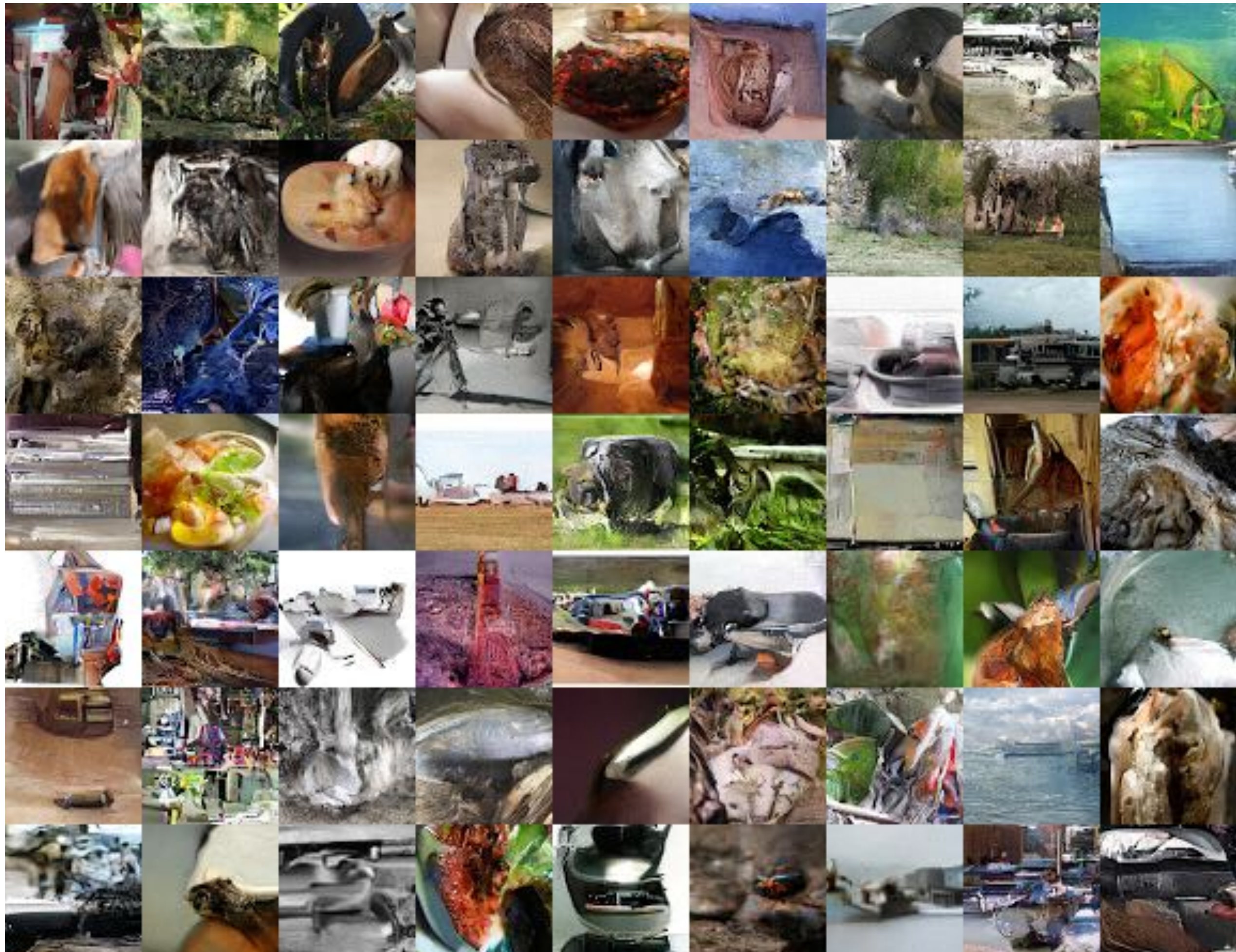
SVNH reconstructions

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



LSUN bedrooms

Tiny ImageNet



Applications of GANs

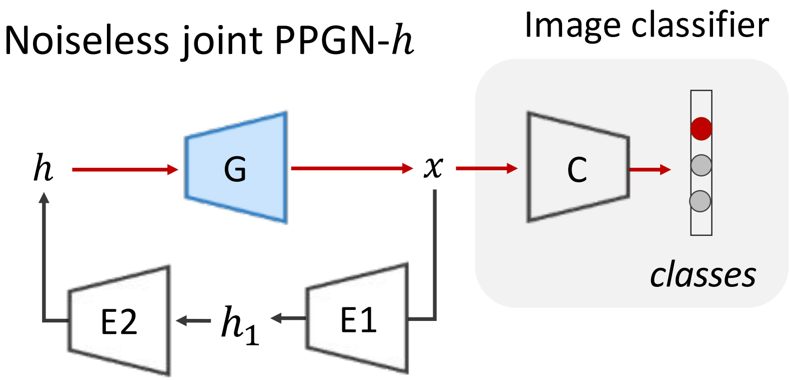
Semi-supervised Classification (Salimans et al., 2016; Dumoulin et al., 2016)

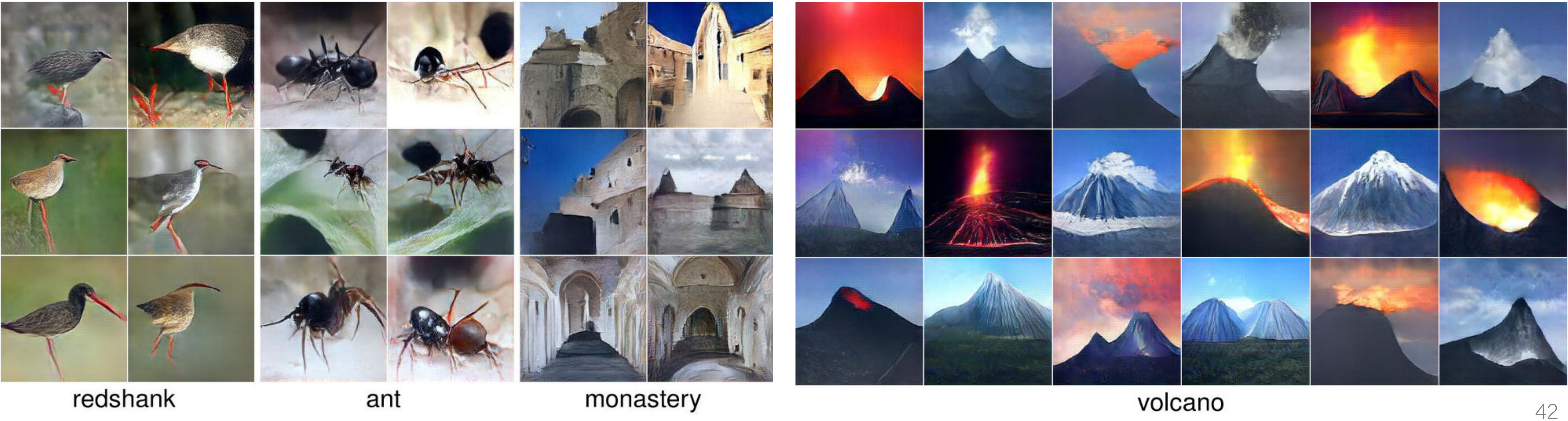
SVNH

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)	$\mathbf{7.42 \pm 0.65}$

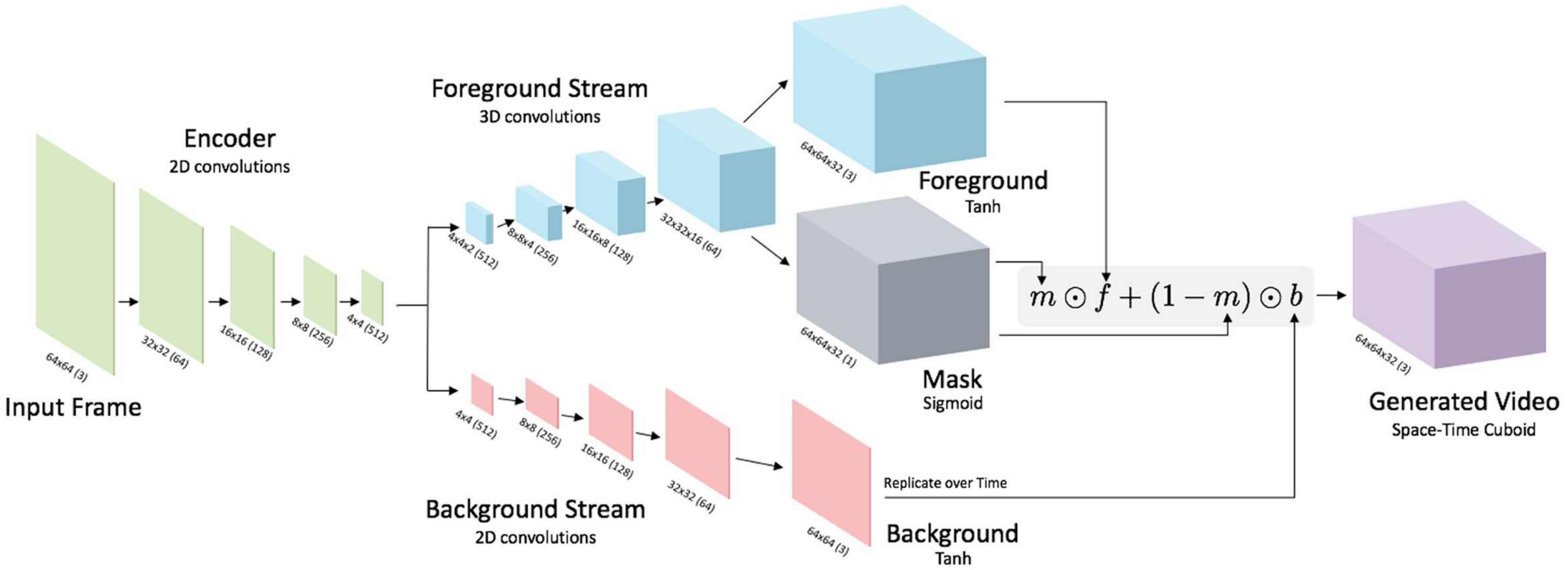
Class-specific Image Generation (Nguyen et al., 2016)

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





Video Generation (Vondrick et al., 2016)

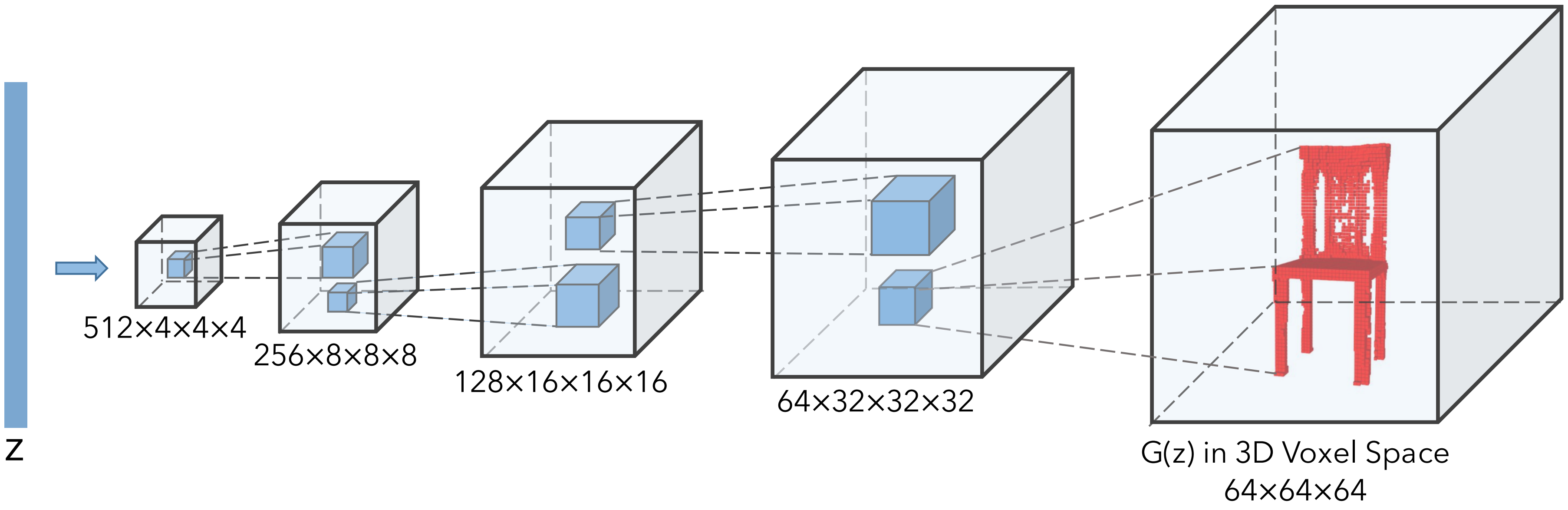


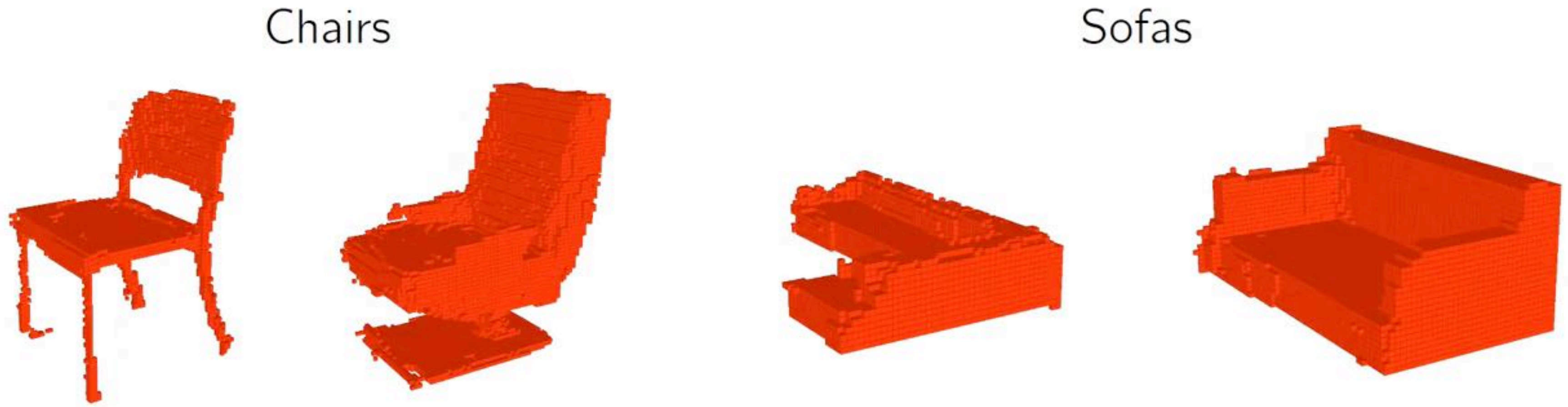
Beach

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

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

- Combine content loss with adversarial loss

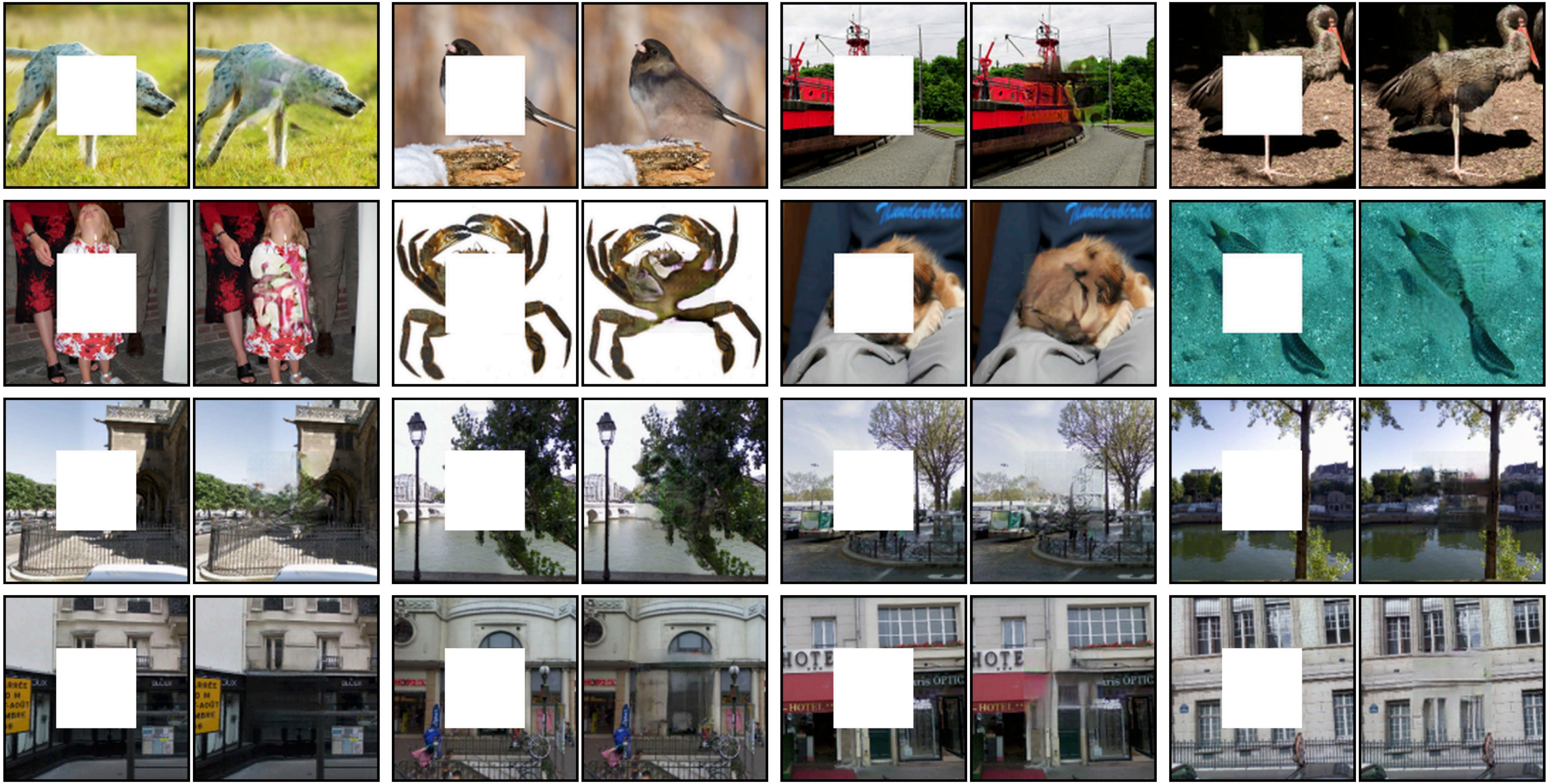
bicubic

SRResNet

SRGAN

original

Image Inpainting (Pathak et al., 2016)



Unsupervised Domain Adaptation (Bousmalis et al., 2016)

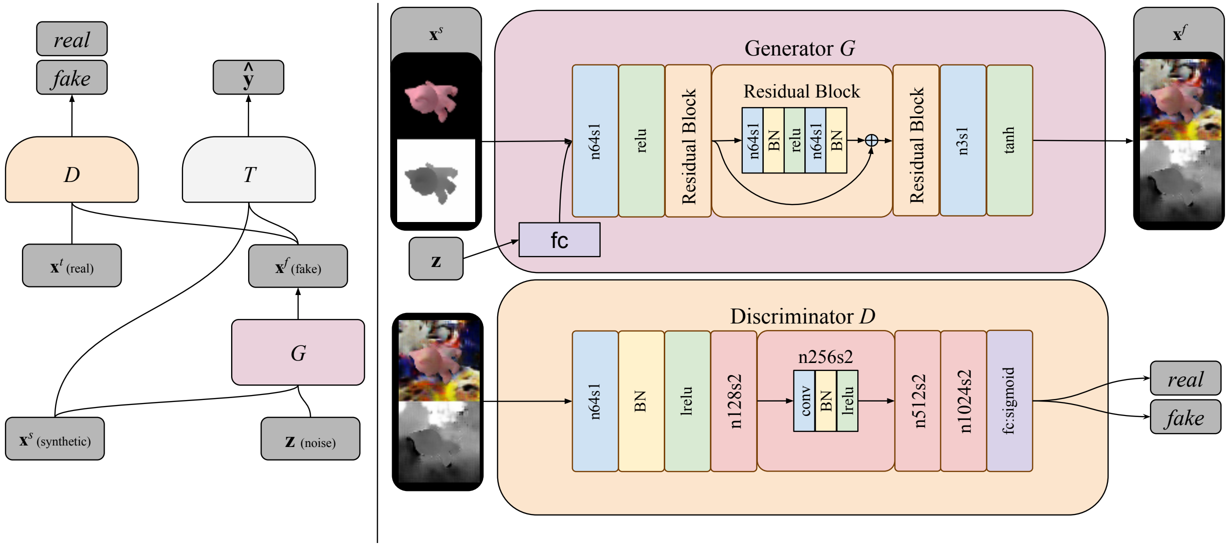


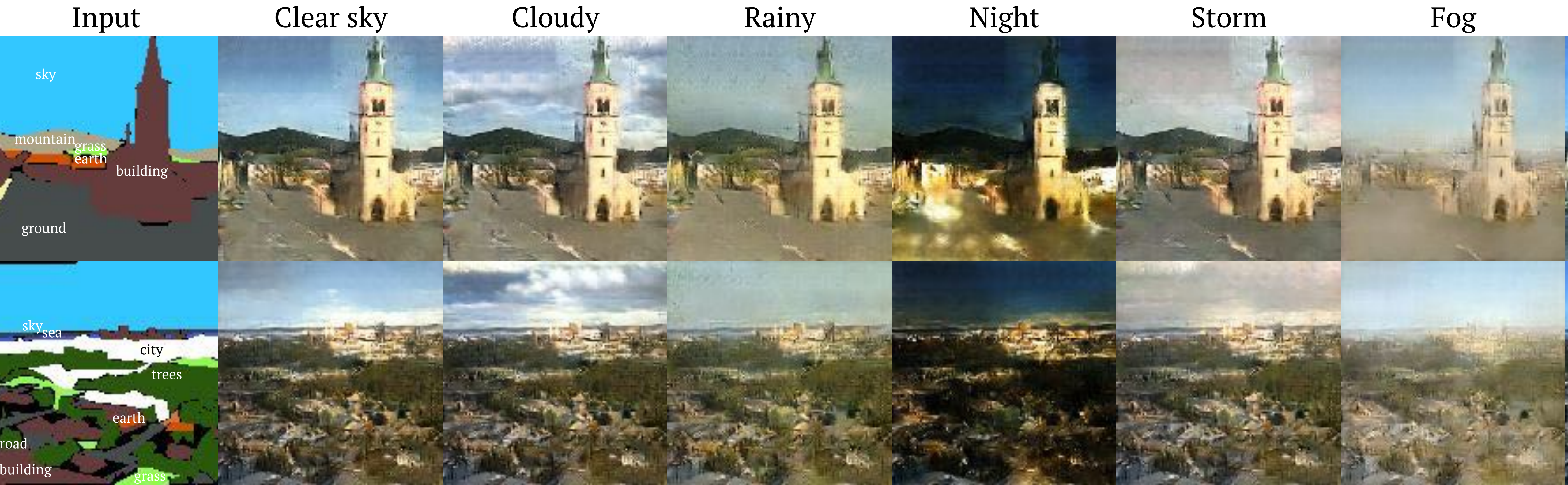
Image examples from the Linemod dataset

RGDB image samples
(*conditioned on a synthetic image*)

Image Editing

(Karacan et al., 2016)

"Maybe in our world lives a happy little tree over there."
— *Bob Ross*



How to Evaluate GANs?

Human Study

In this task, we present you computer generated pictures of outdoor scenes generated by different computer programs. Your task is to compare them and determine which is more realistic and natural looking. See the below table for some examples.

More realistic looking outdoor images	Less realistic looking outdoor images

Steps

1. Analyze both images and consider their features carefully
2. Determine which computer generated image (image A or image B) is more realistic than the other.

A

B

Which image seems more real?

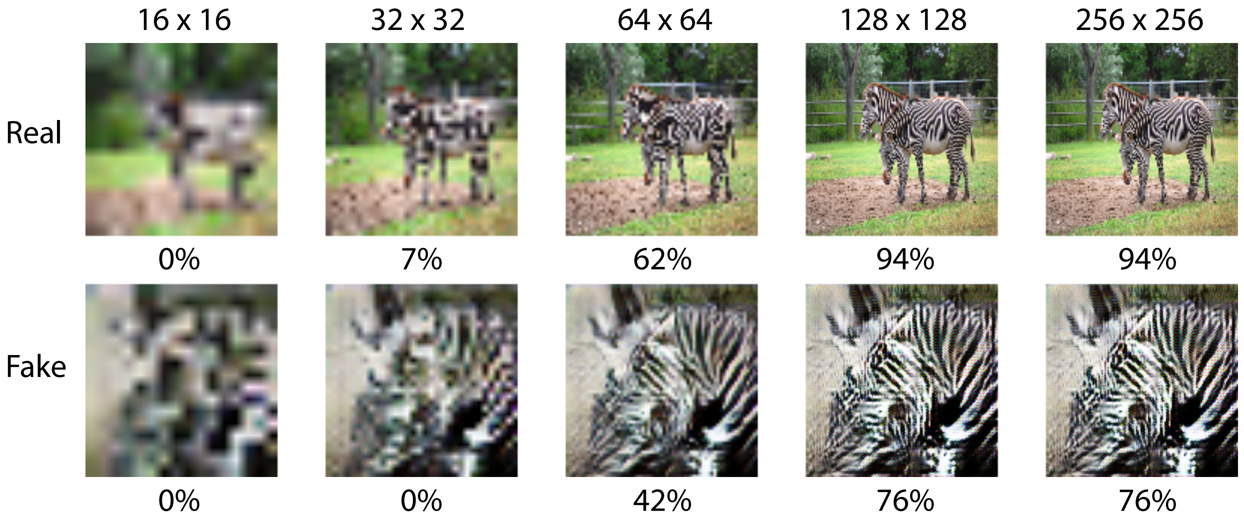
- A
- B

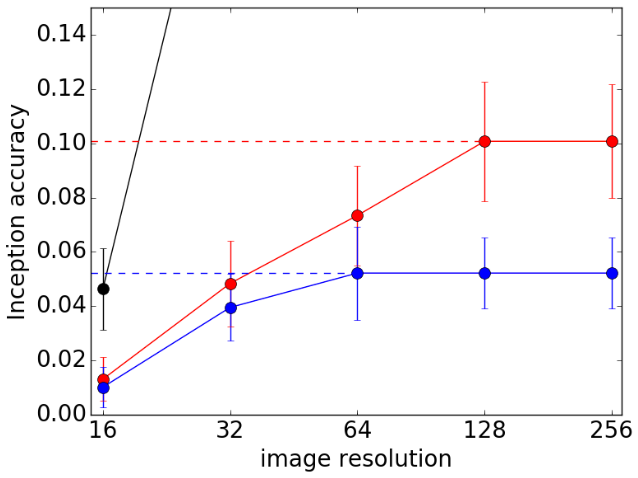
From (Karacan et al., 2016)

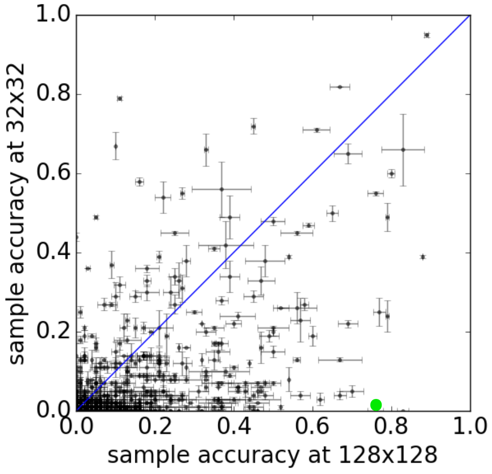
Evaluating Quality

- Hard to tell if progress is being made by looking at losses

- Inception Score (Salimans et al., 2016)
- Inception Accuracy (Odena et al., 2016)
  - Report the fraction of the samples for which the Inception network assigned the correct label

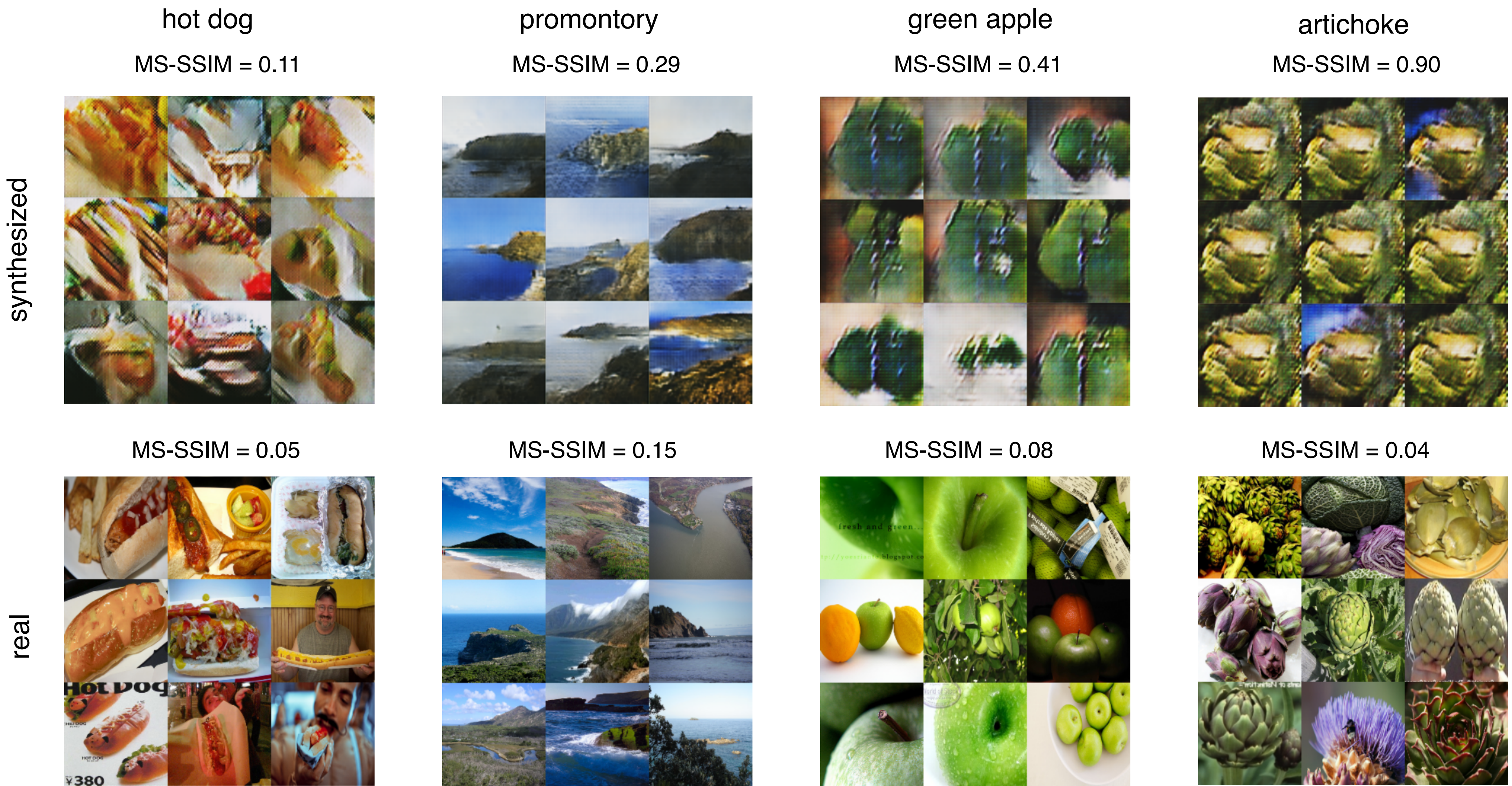






Measuring Diversity (Odena et al., 2016)

- MS-SSIM scores [between randomly chosen pairs of images within a given class]



Searching for Overfitting

- Nearest Neighbor Analysis (Odena et al., 2016)

Synthesized Samples

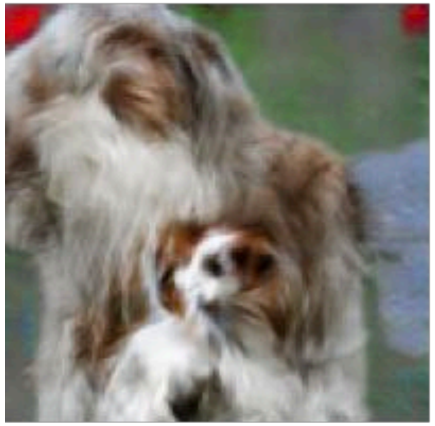
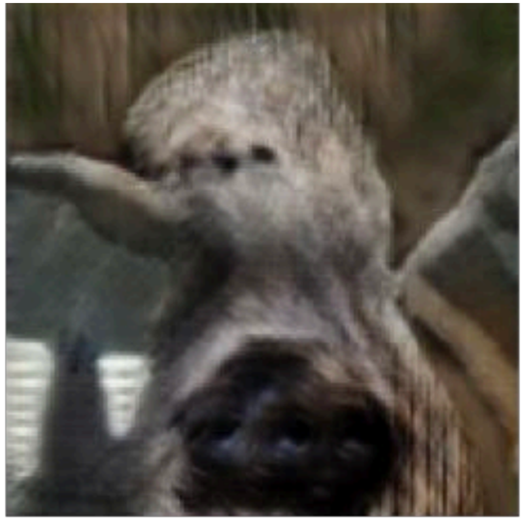
Corresponding Nearest Neighbors
In The Training Set

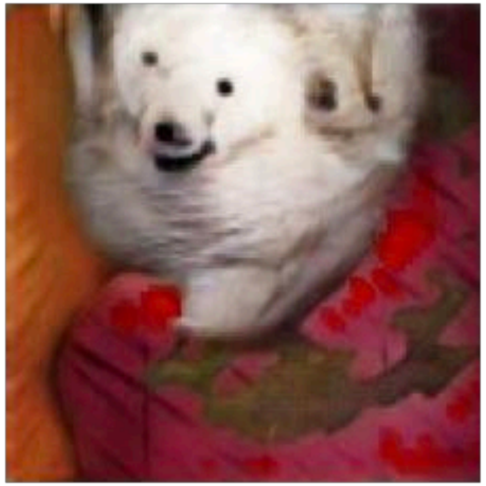
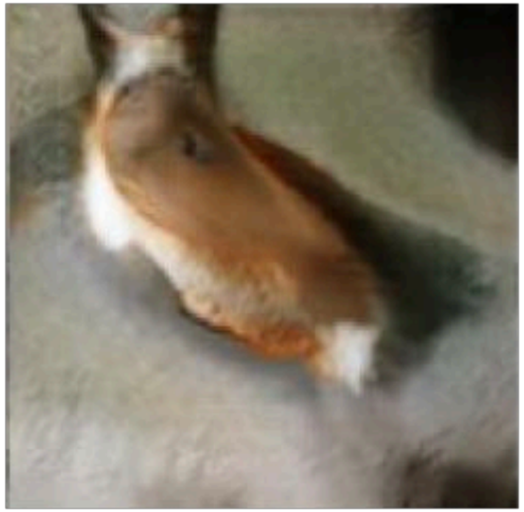
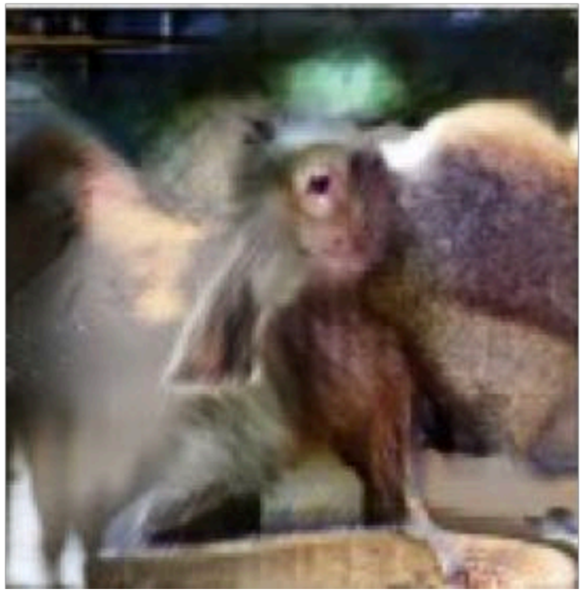
- Latent Space Interpolations

Image: (Dumoilin et al., 2016)

Limitations

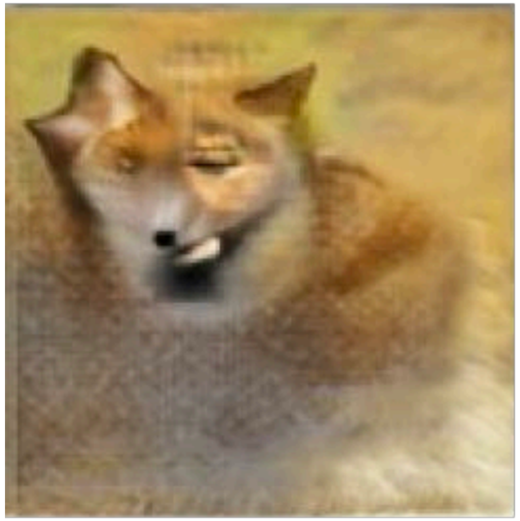
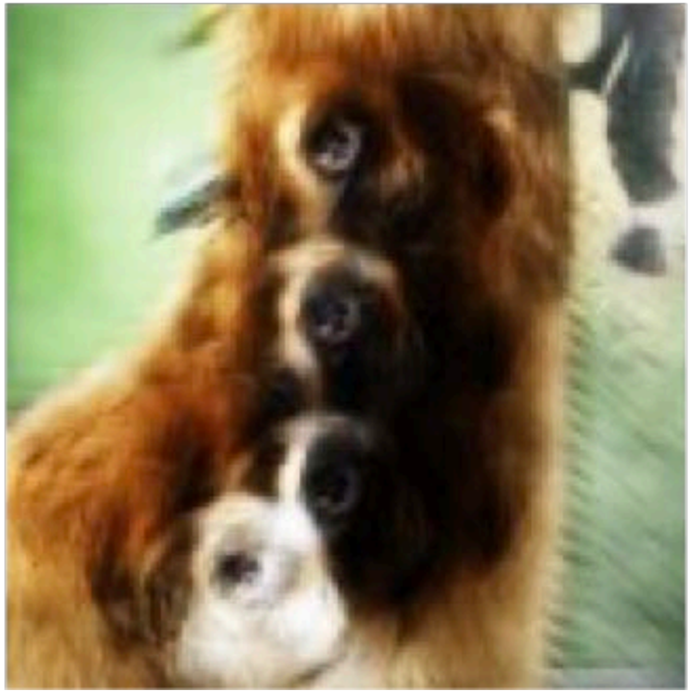
Cherry-Picked Results

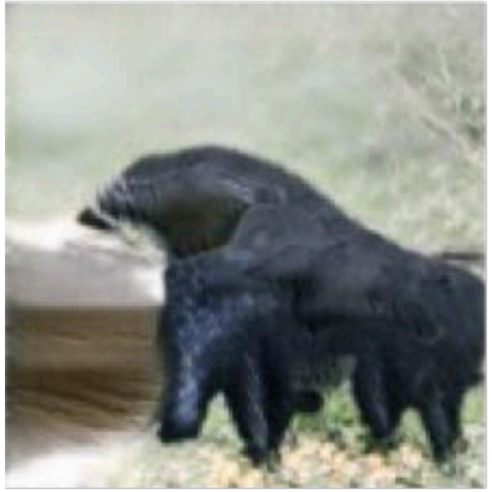




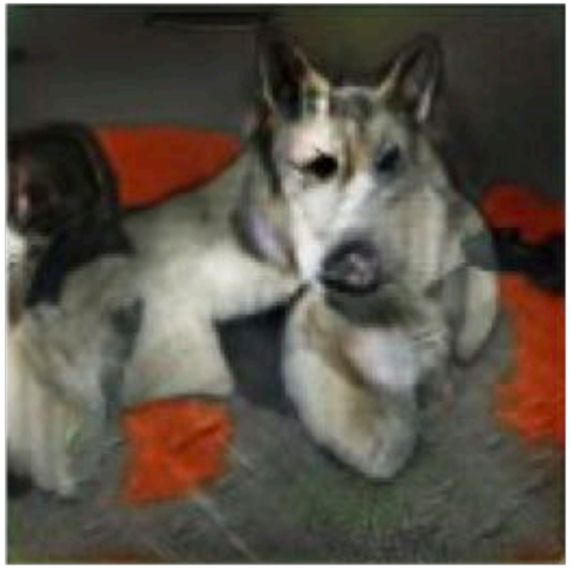
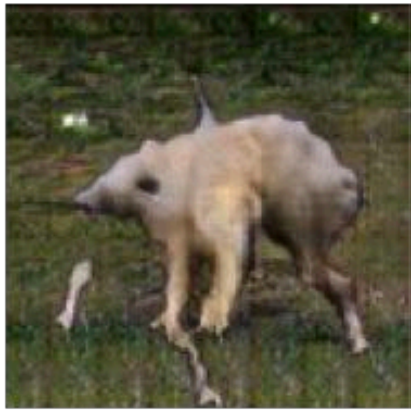
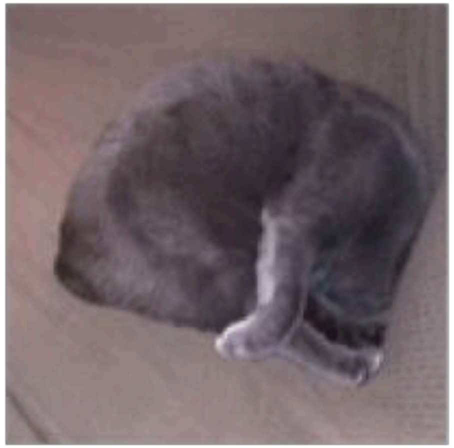
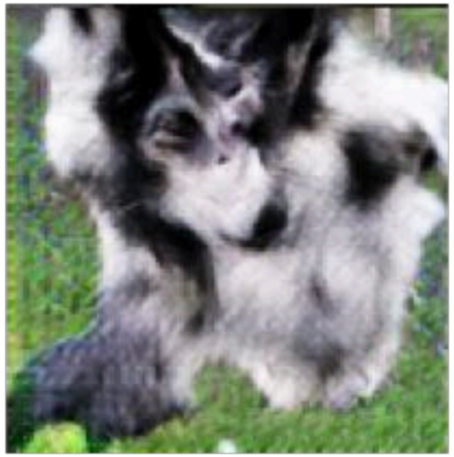
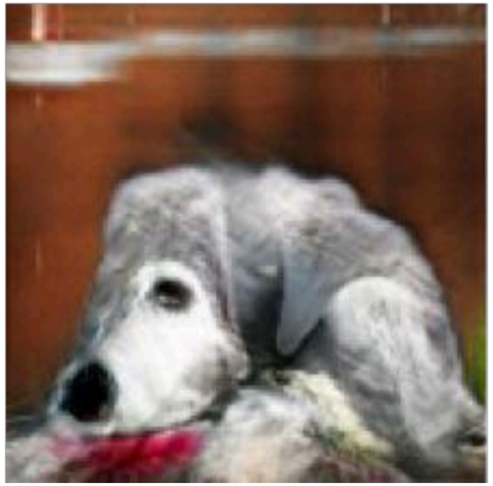

Problems with Counting



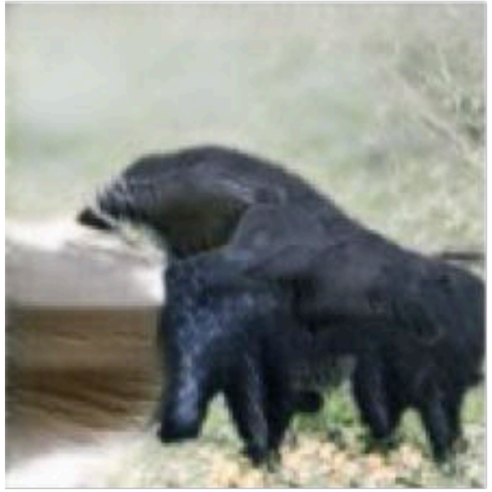




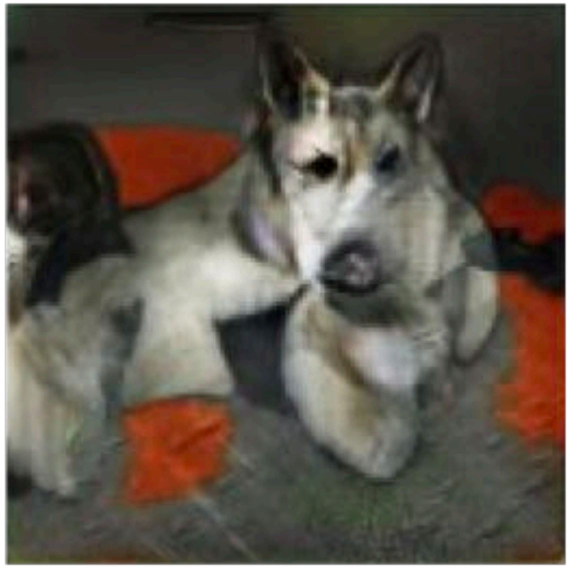

Problems with Perspective

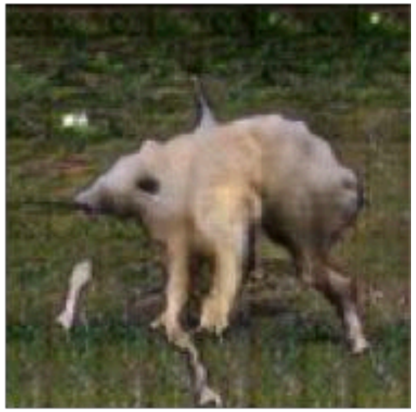


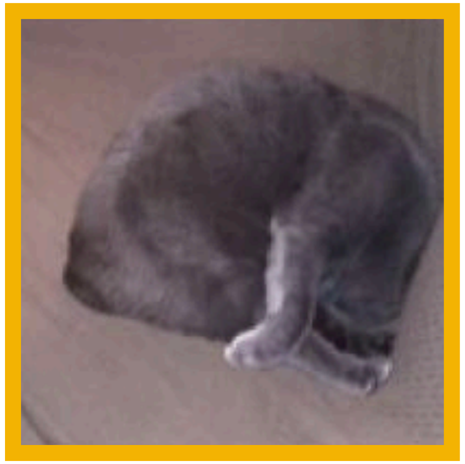
Problems with Perspective

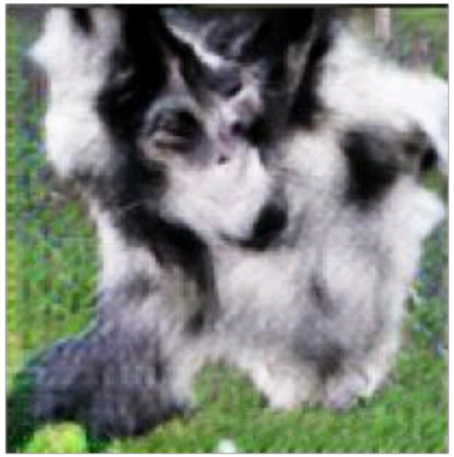


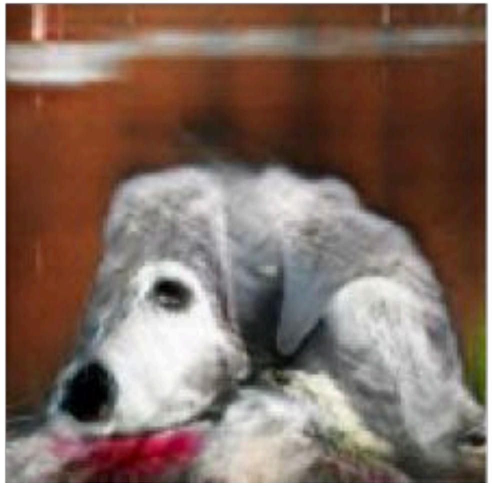




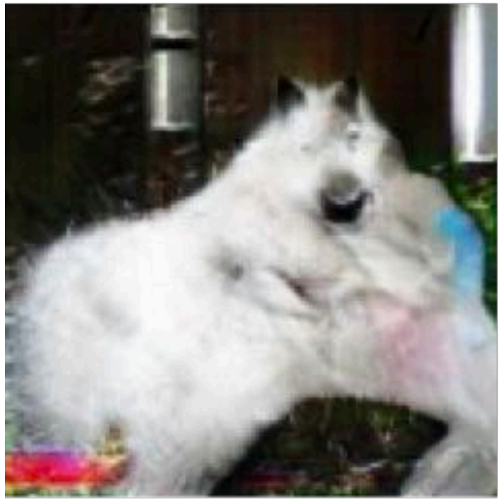
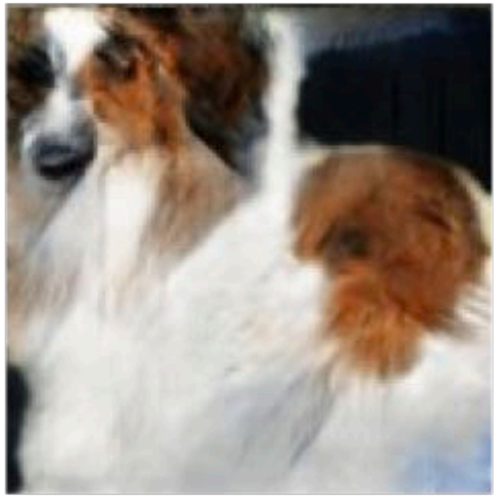
This one was real

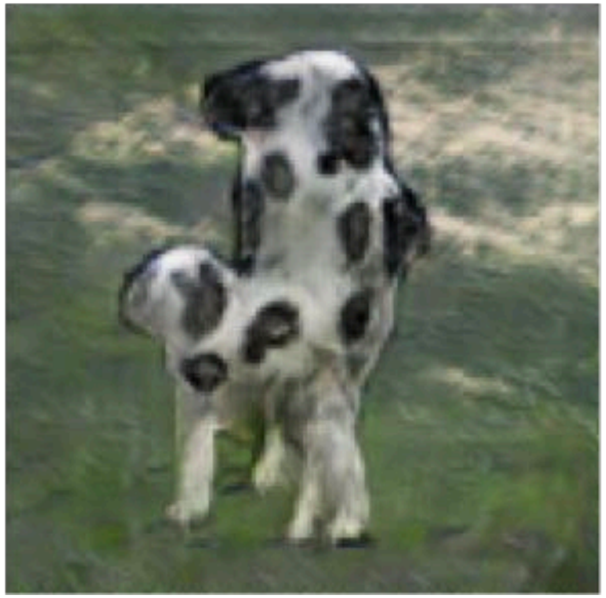
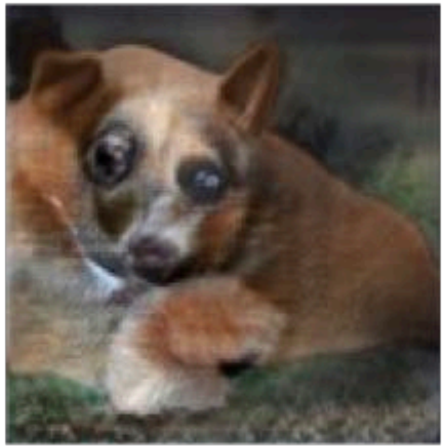
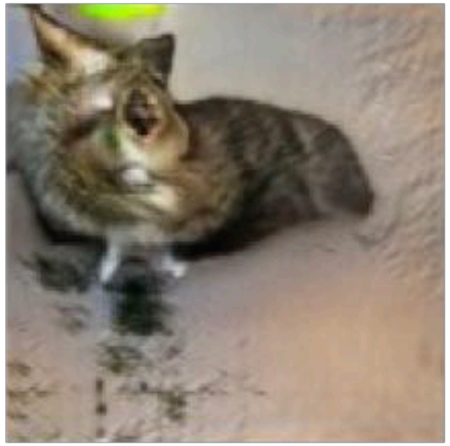






Problems with Global Structure

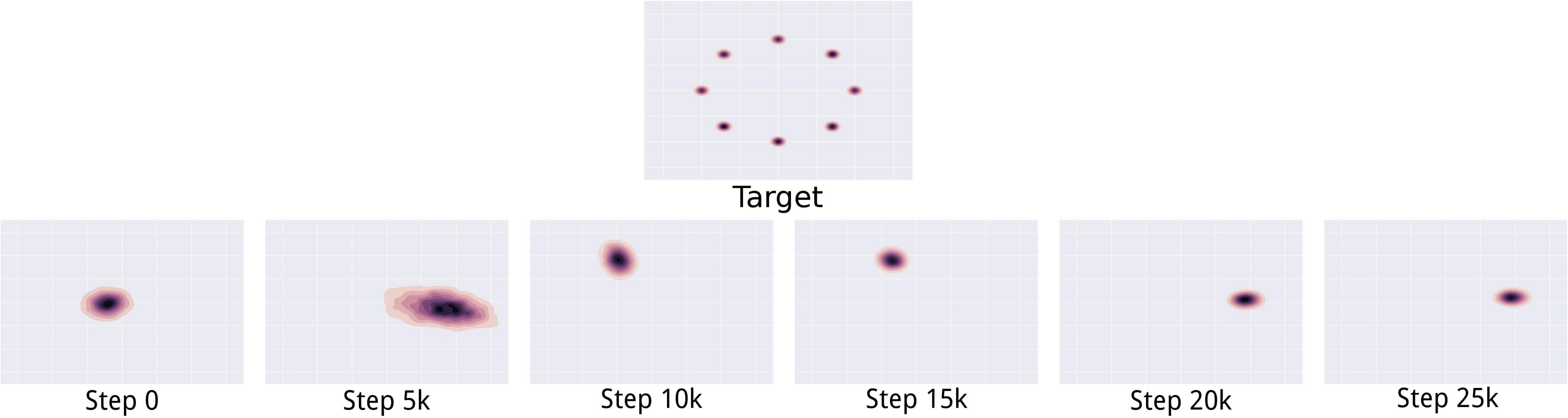




Slide credit: Ian Goodfellow

Mode Collapse (Metz et al., 2016)

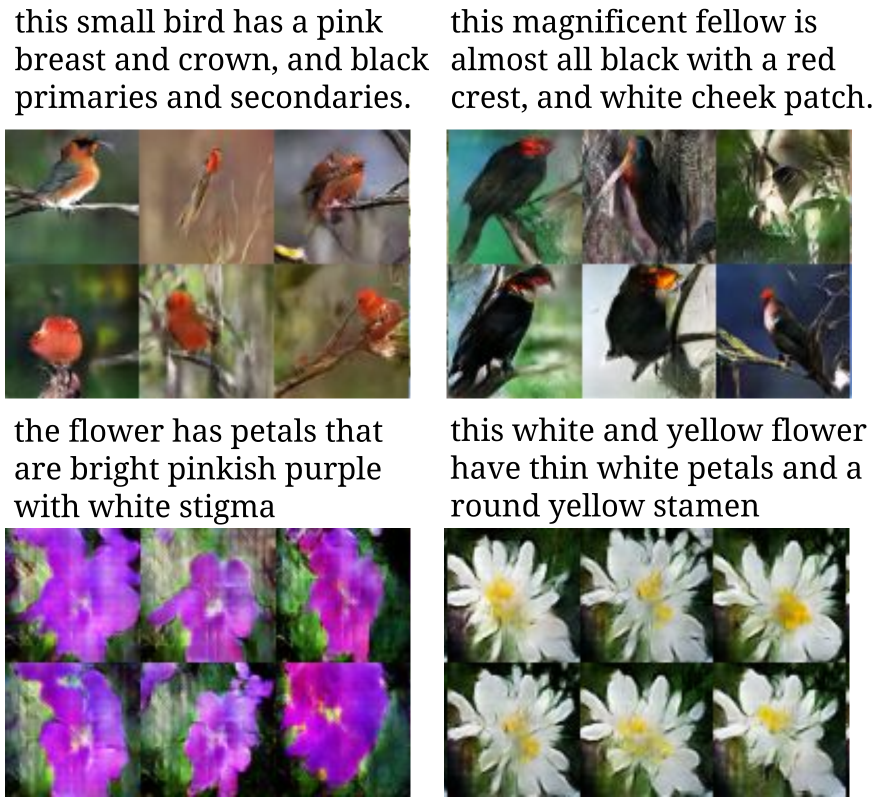
$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

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



Slide credit: Ian Goodfellow

Mode collapse causes low output diversity



(Reed et al. 2016)

(Reed et al., 2017)

Non-convergence

- Optimization algorithms often approach a saddle point or local minimum rather than a global minimum

- Game solving algorithms may not approach an equilibrium at all

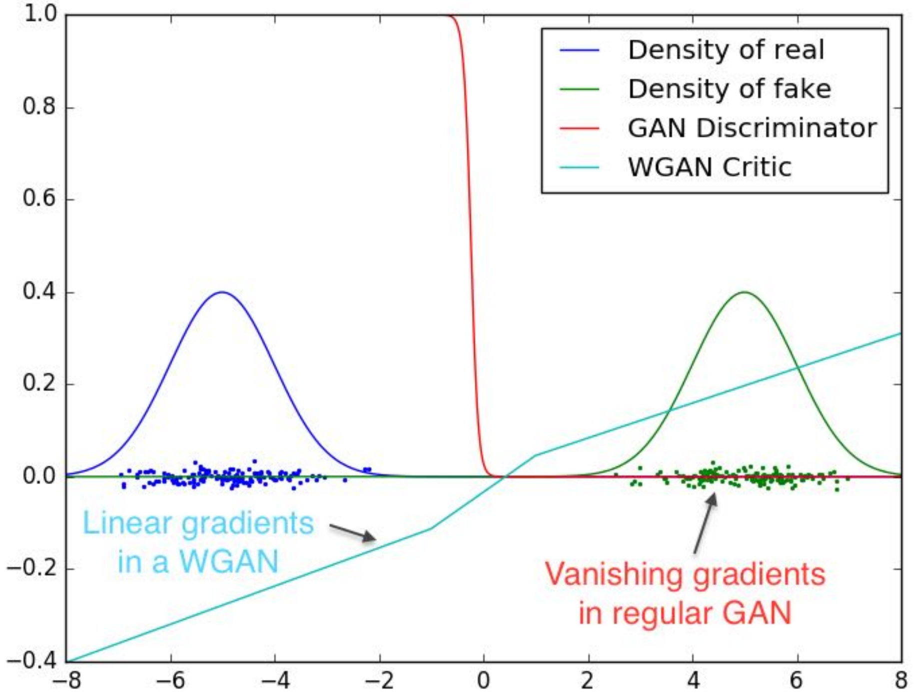
Frontiers

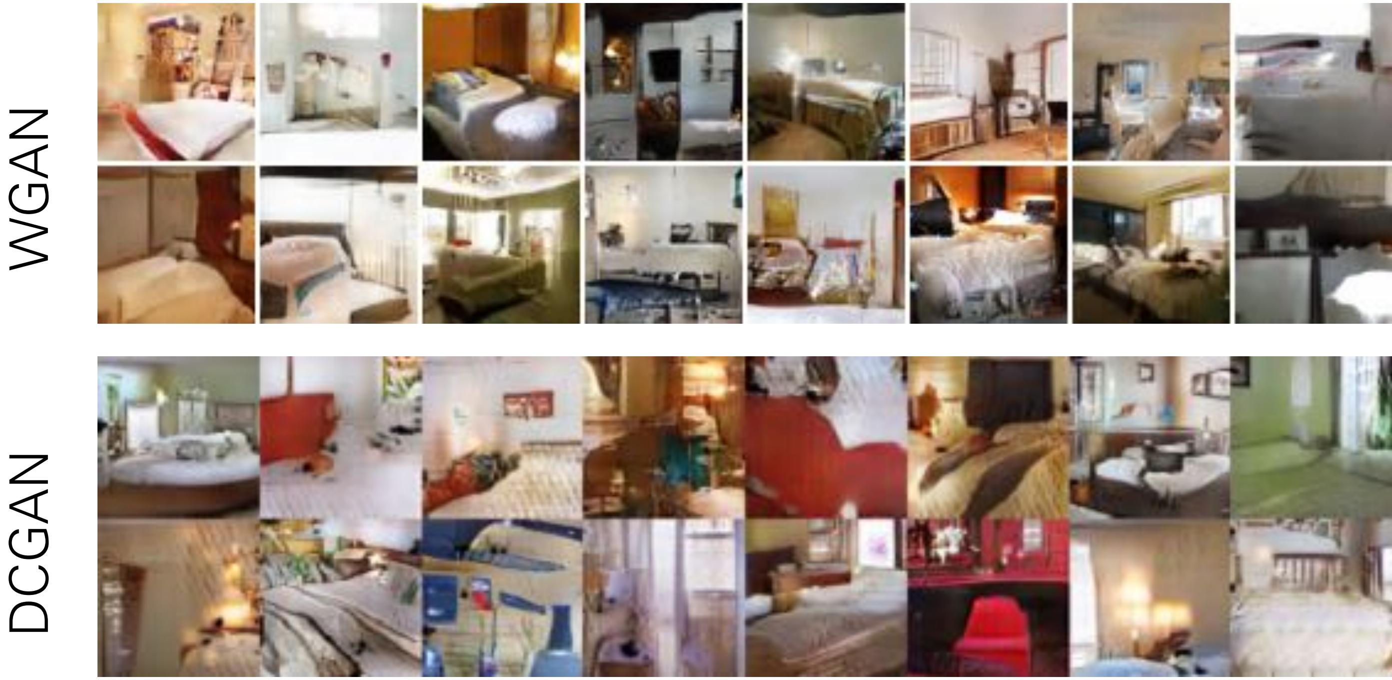
Wasserstein GAN (Arjovsky et al., 2016)

- Objective based on Earth-Mover or Wassertein distance:

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

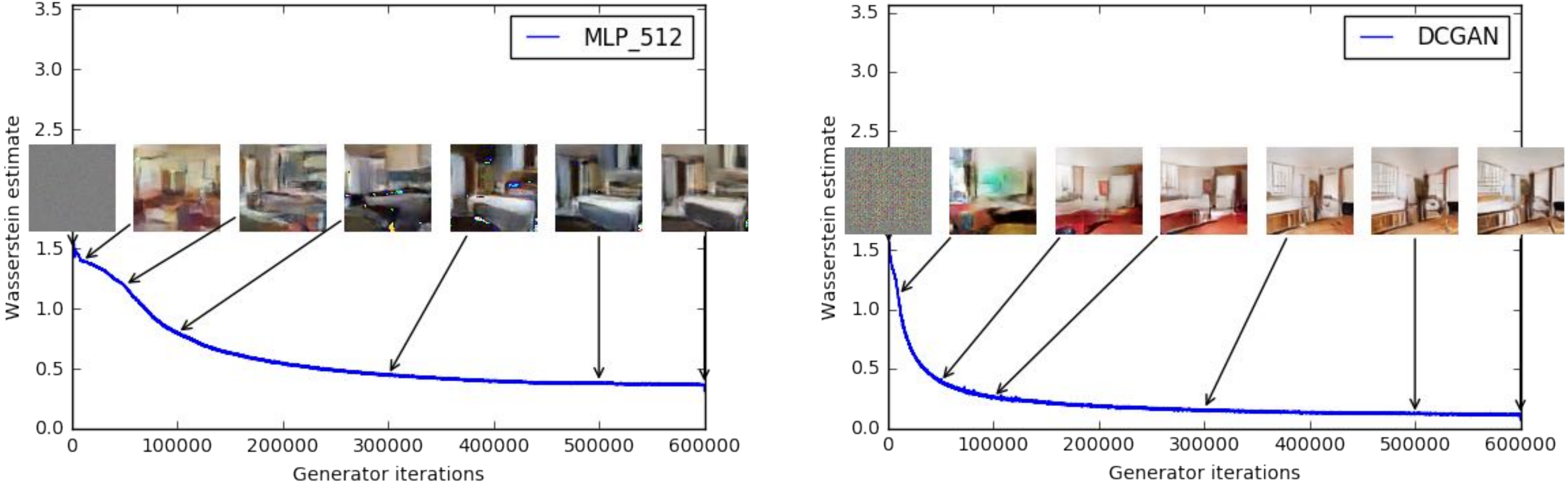
- Provides nice gradients over real and fake samples





Wasserstein GAN (Arjovsky et al., 2016)

- Wasserstein loss seems to correlate well with image quality.



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

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

- 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

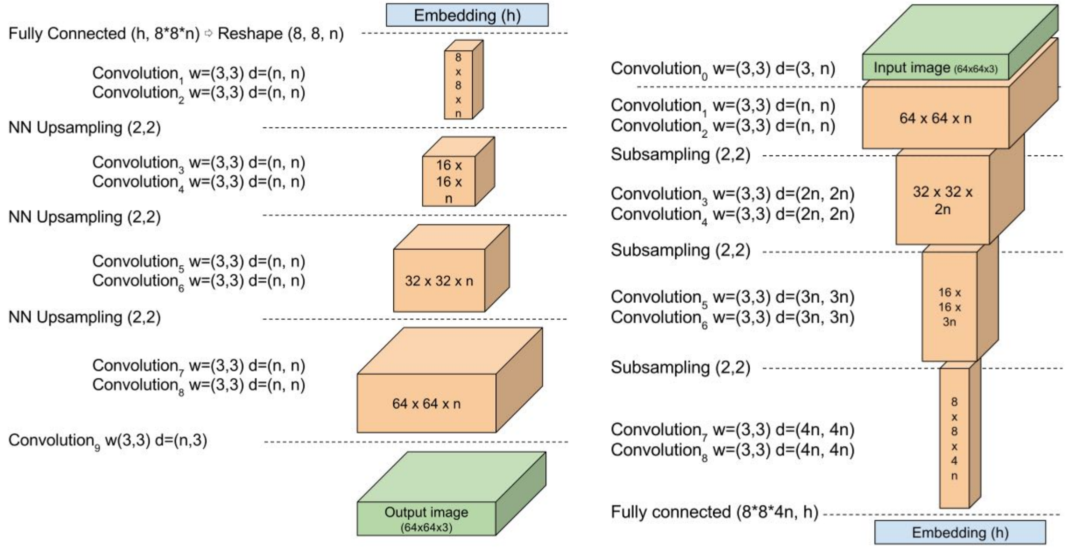
Boundary Equilibrium GAN (BEGAN)

(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



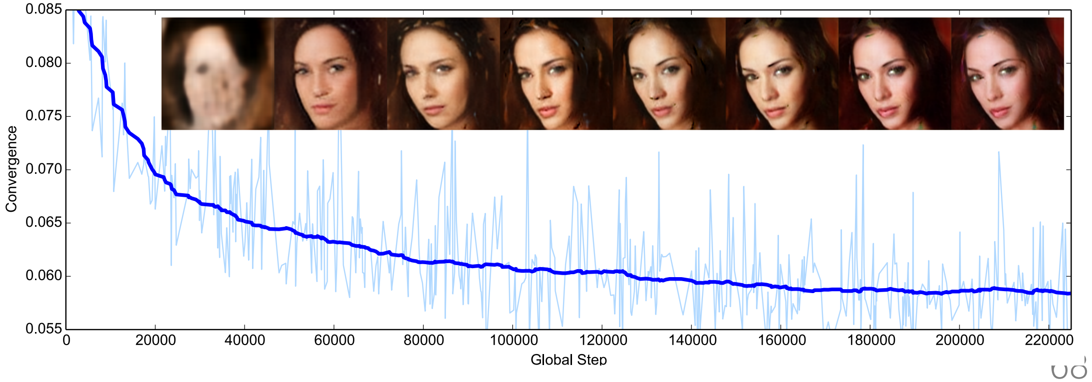
(a) Generator/Decoder (b) Encoder

- **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