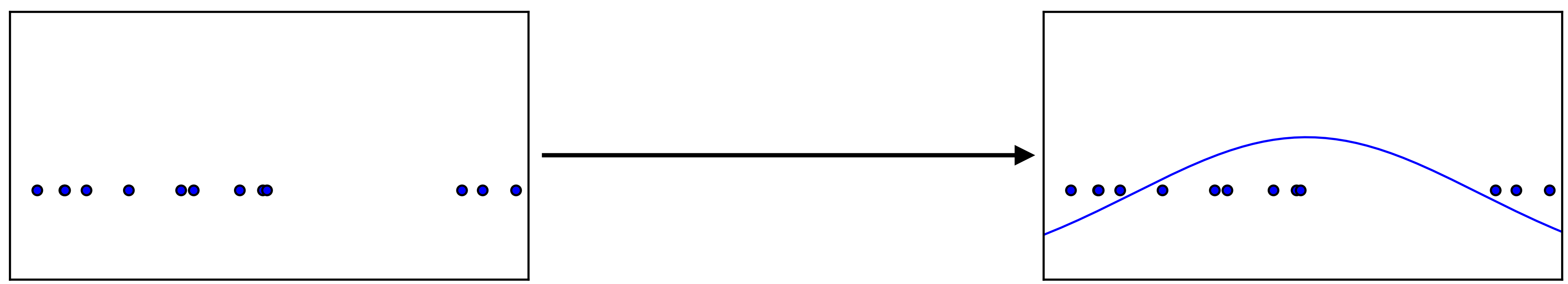
CSC412: Adversarial Training

David Duvenaud

Slides from Ian Goodfellow, Roger Grosse and Sebastian Nowozin

Generative Modeling

- Density estimation



- Sample generation

Training examples   Model samples

(Goodfellow 2016)

Fully Visible Belief Nets

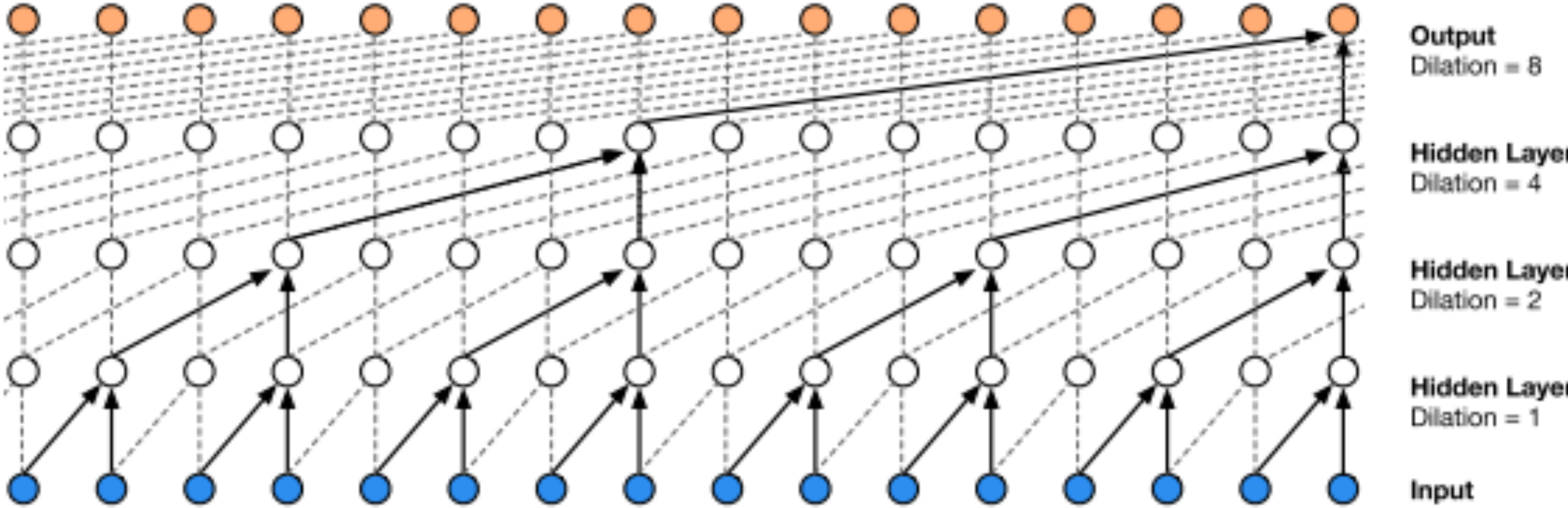
- Explicit formula based on chain rule: (Frey et al, 1996)

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

- Disadvantages:
  - O(n) sample generation cost
  - Generation not controlled by a latent code

PixelCNN elephants (van den Ord et al 2016)

WaveNet



Amazing quality
Sample generation slow

Two minutes to synthesize
one second of audio

(Goodfellow 2016)

Change of Variables

$$y = g(x) \Rightarrow p_x(\boldsymbol{x}) = p_y(g(\boldsymbol{x})) \left| \det \left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right|$$

e.g. Nonlinear ICA (Hyvärinen 1999)

64x64 ImageNet Samples

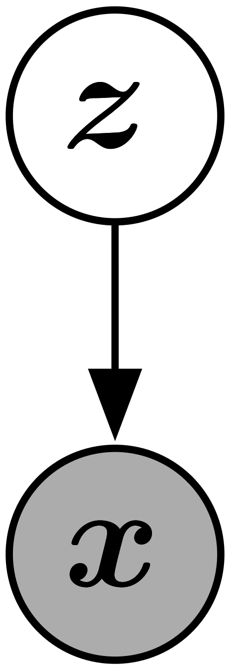
Real NVP (Dinh et al 2016)

Disadvantages:

- Transformation must be invertible
- Latent dimension must match visible dimension

Variational Autoencoder

(Kingma and Welling 2013, Rezende et al 2014)



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

CIFAR-10 samples
(Kingma et al 2016)

Disadvantages:

-Not asymptotically consistent unless q is perfect

-Samples tend to have lower quality

Boltzmann Machines

$$p(\boldsymbol{x}) = \frac{1}{Z} \exp\left(-E(\boldsymbol{x}, \boldsymbol{z})\right)$$

$$Z = \sum_{\boldsymbol{x}} \sum_{\boldsymbol{z}} \exp\left(-E(\boldsymbol{x}, \boldsymbol{z})\right)$$

- Partition function is intractable
- May be estimated with Markov chain methods
- Generating samples requires Markov chains too

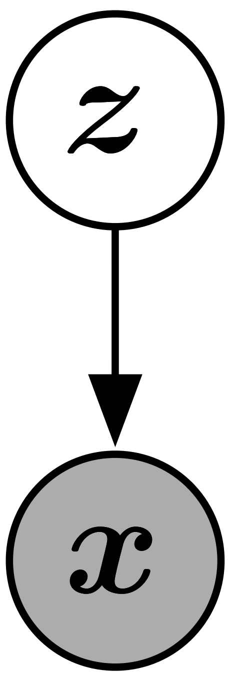
GANs

- Use a latent code
- Asymptotically consistent (unlike variational methods)
- No Markov chains needed
- Often regarded as producing the best samples
  - No good way to quantify this

(Goodfellow 2016)

Generator Network

$$\boldsymbol{x} = G(\boldsymbol{z}; \boldsymbol{\theta}^{(G)})$$



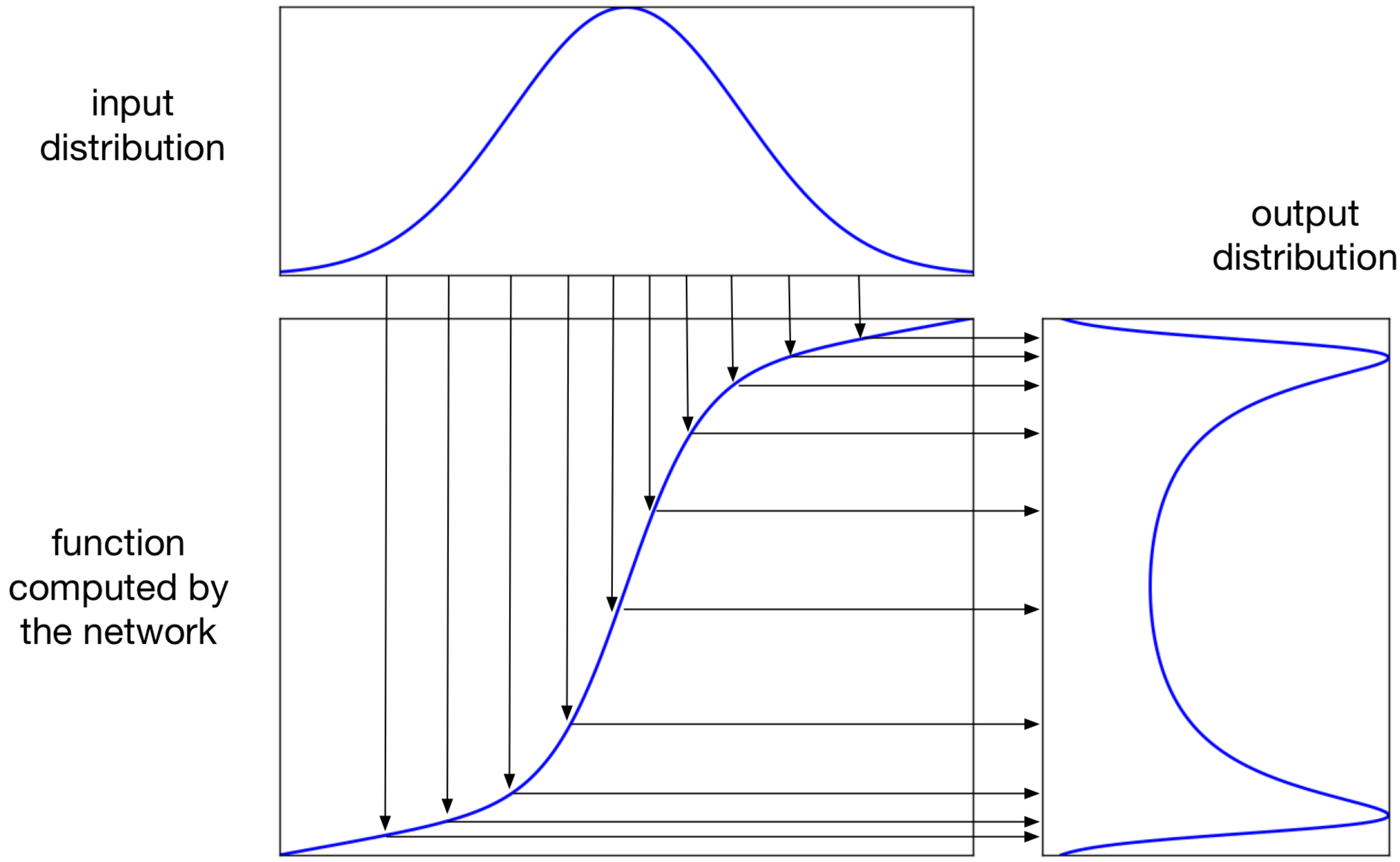
-Must be differentiable

- No invertibility requirement
- Trainable for any size of z
- Some guarantees require z to have higher dimension than x
- Can make x conditionally Gaussian given z but need not do so

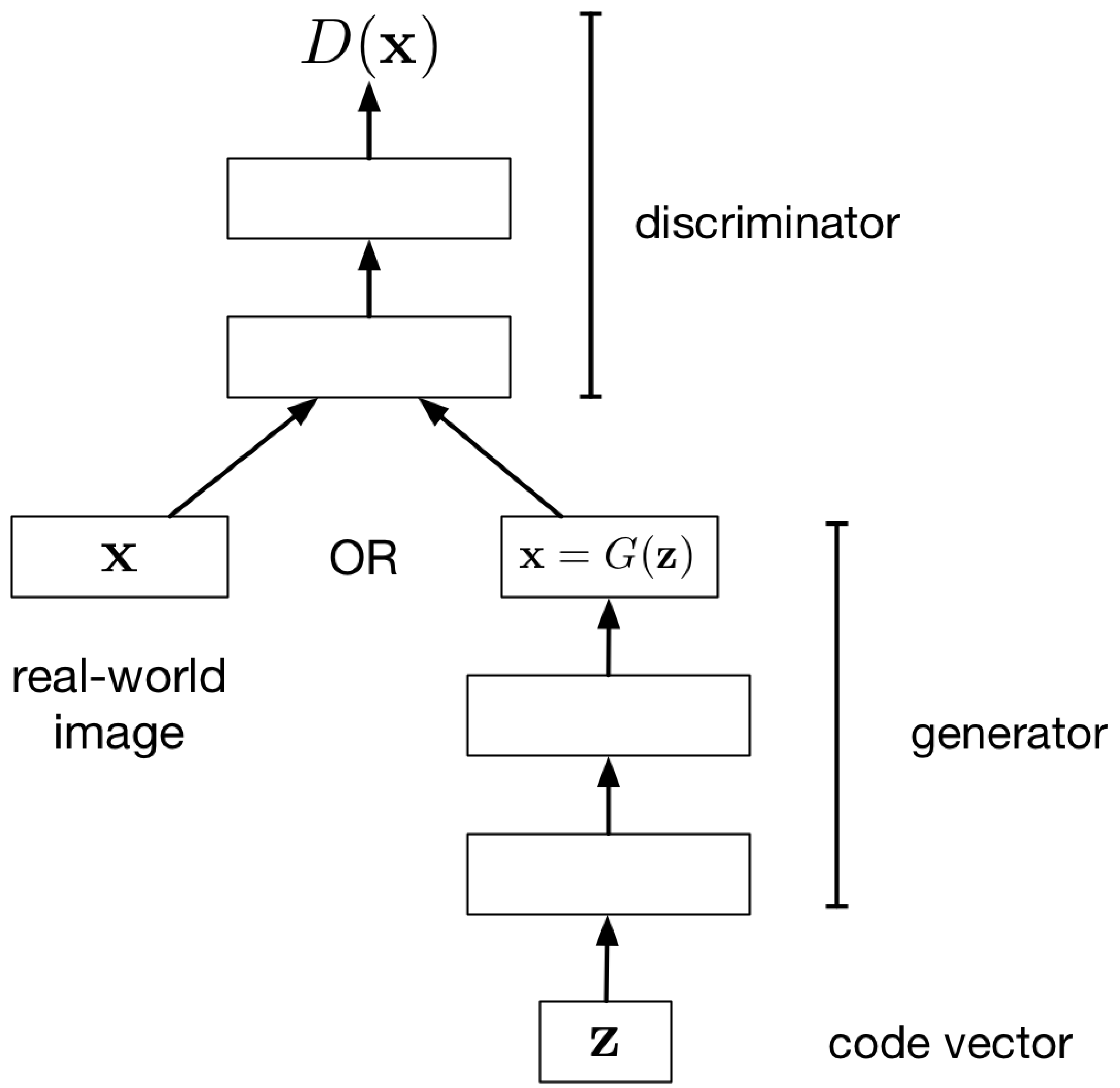
(Goodfellow 2016)

Generative Adversarial Networks

A 1-dimensional example:

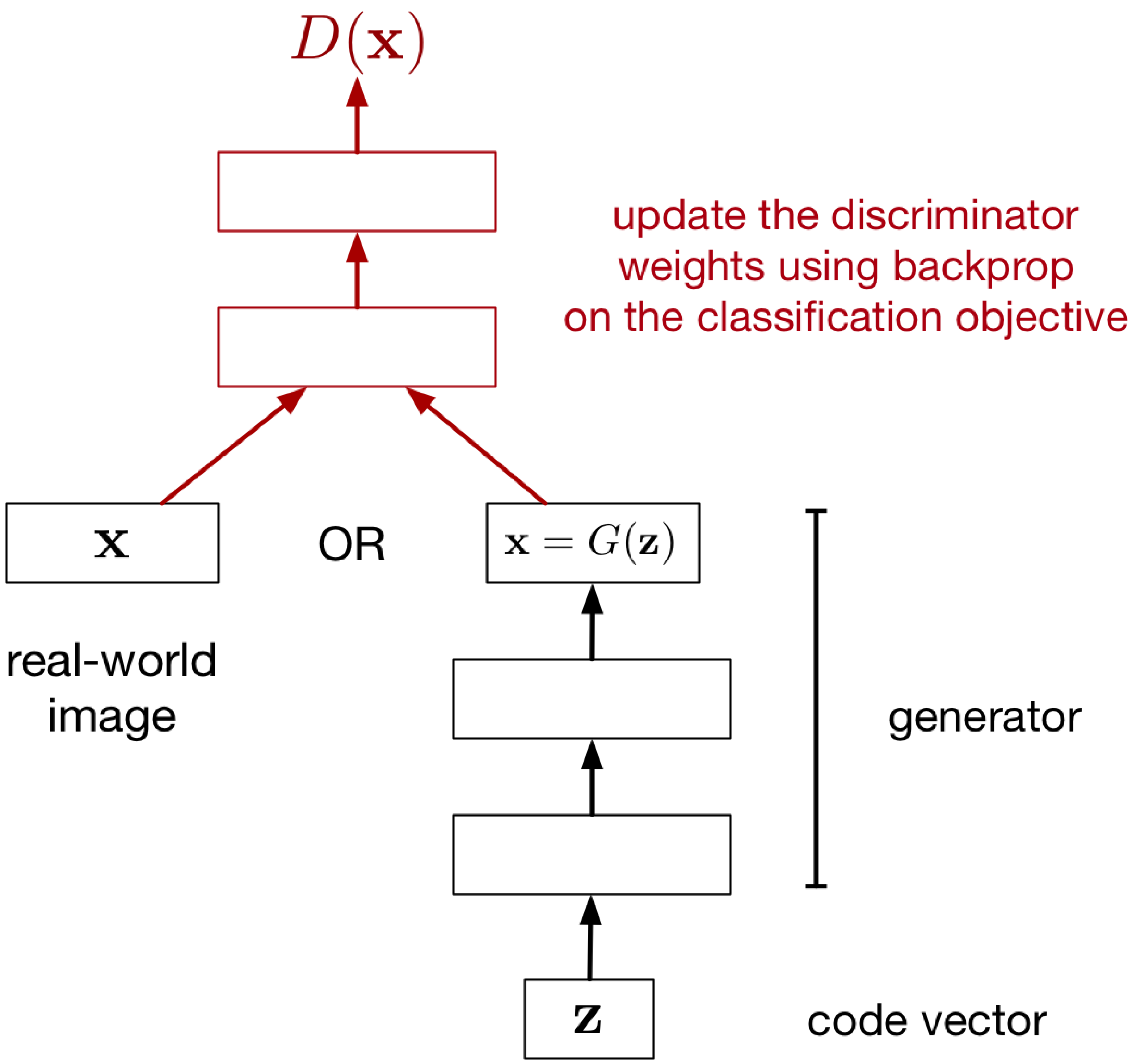


Generative Adversarial Networks



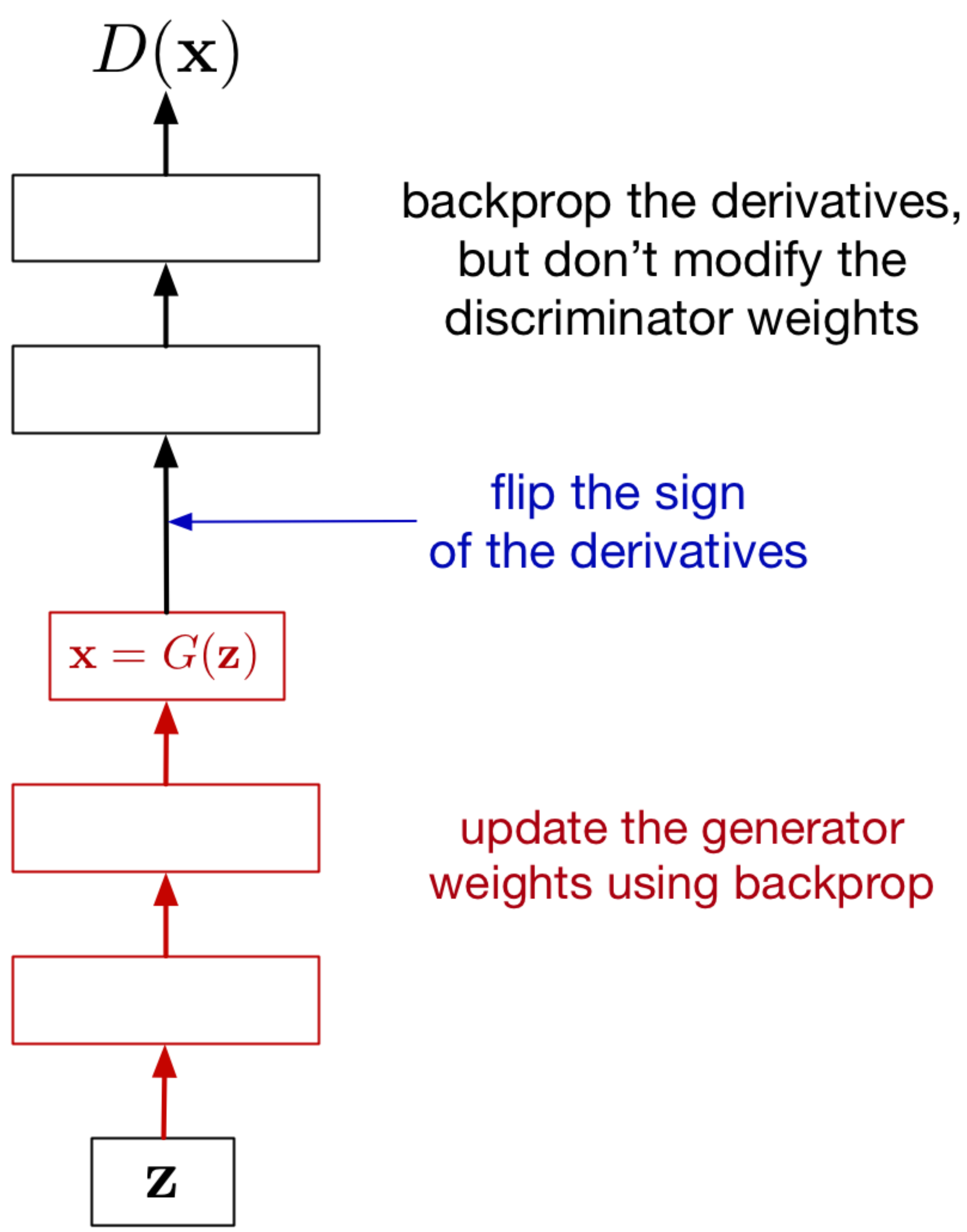
Generative Adversarial Networks

Updating the discriminator:



Generative Adversarial Networks

Updating the generator:



Training Procedure

- Use SGD-like algorithm of choice (Adam) on two minibatches simultaneously:
  - A minibatch of training examples
  - A minibatch of generated samples
- Optional: run k steps of one player for every step of the other player.

(Goodfellow 2016)

Minimax Game

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

$$J^{(G)} = -J^{(D)}$$

-Equilibrium is a saddle point of the discriminator loss

-Resembles Jensen-Shannon divergence

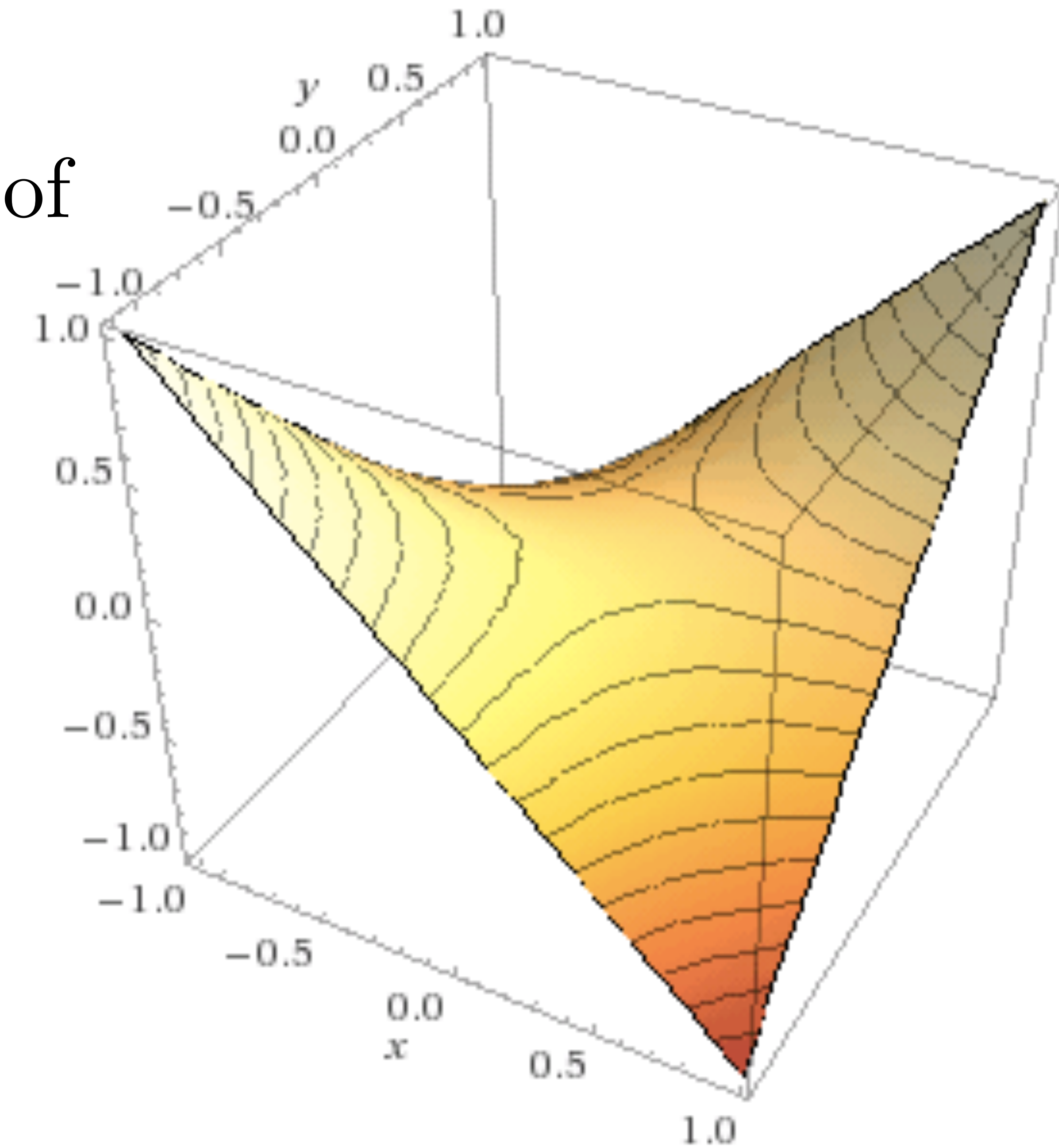
-Generator minimizes the log-probability of the discriminator being correct

(Goodfellow 2016)

Solution

This is the canonical example of a saddle point.

There is an equilibrium, at $x = 0$, $y = 0$.

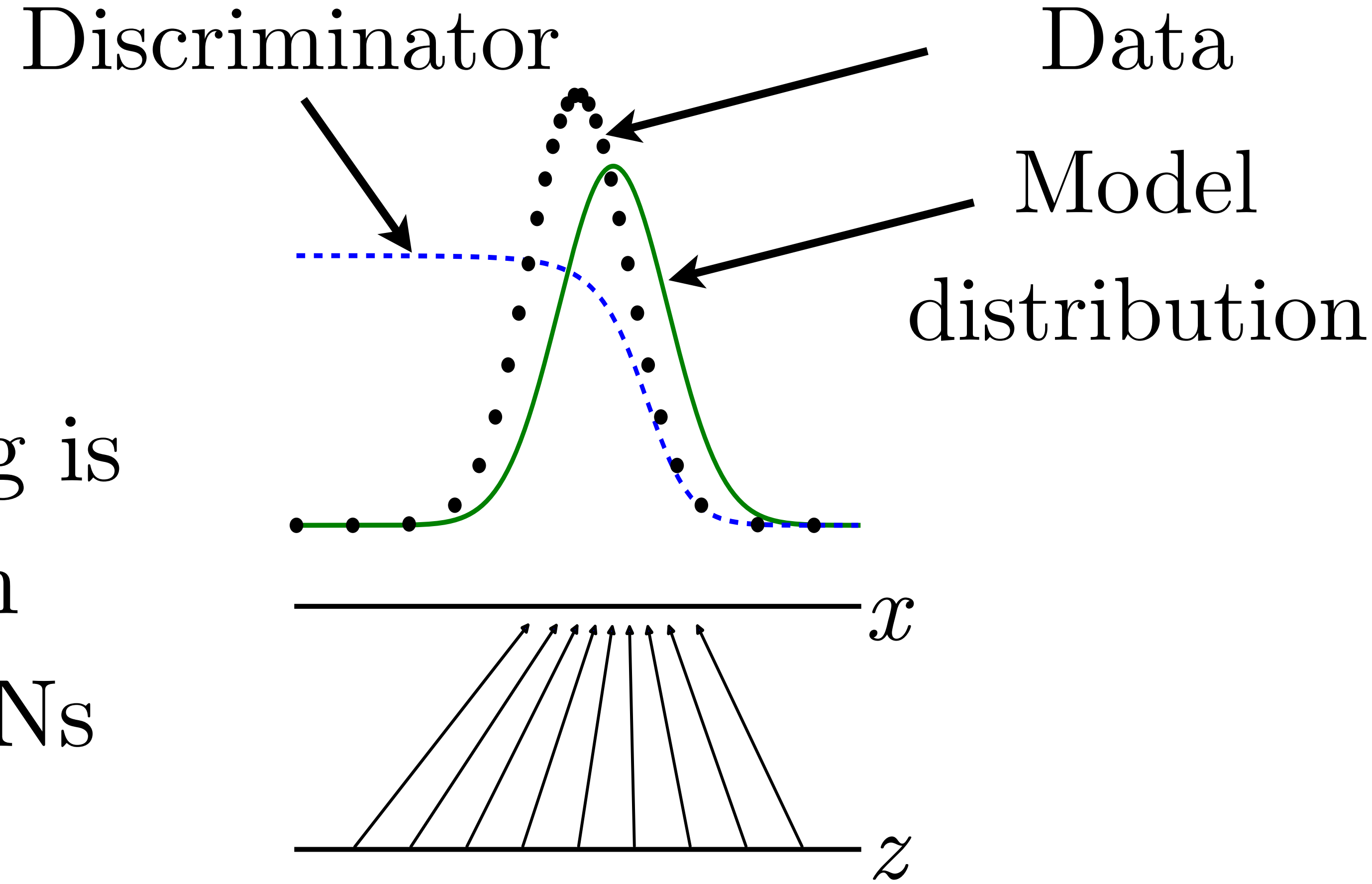


(Goodfellow 2016)

Discriminator Strategy

Optimal $D(\boldsymbol{x})$ for any $p_{\text{data}}(\boldsymbol{x})$ and $p_{\text{model}}(\boldsymbol{x})$ is always

$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$$



Estimating this ratio using supervised learning is the key approximation mechanism used by GANs

(Goodfellow 2016)

Non-Saturating Game

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

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

-Equilibrium no longer describable with a single loss

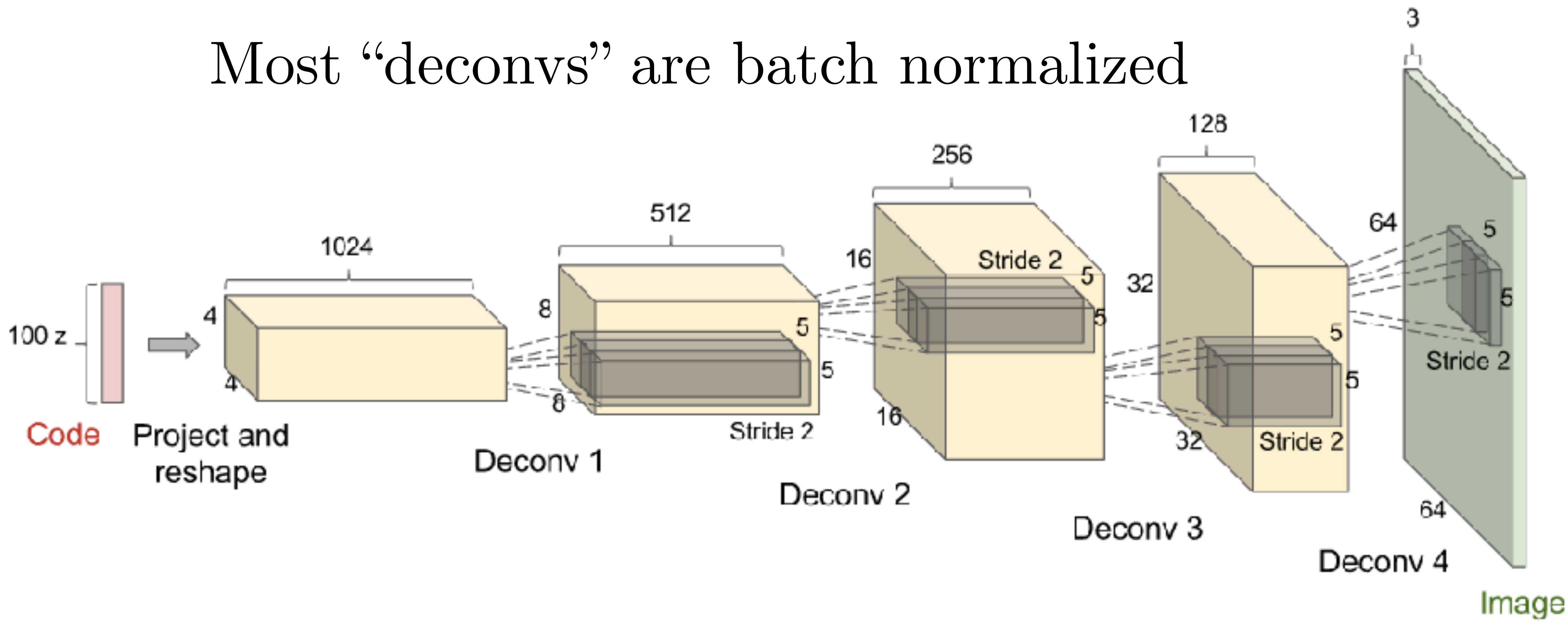
-Generator maximizes the log-probability of the discriminator being mistaken

-Heuristically motivated; generator can still learn even when discriminator successfully rejects all generator samples

(Goodfellow 2016)

DCGAN Architecture

Most “deconvs” are batch normalized

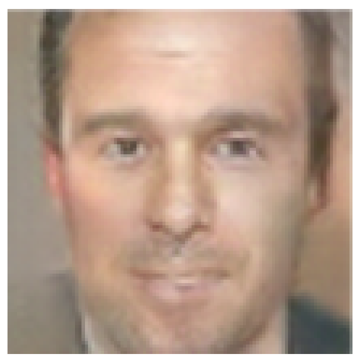
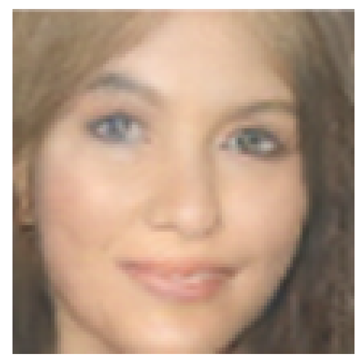
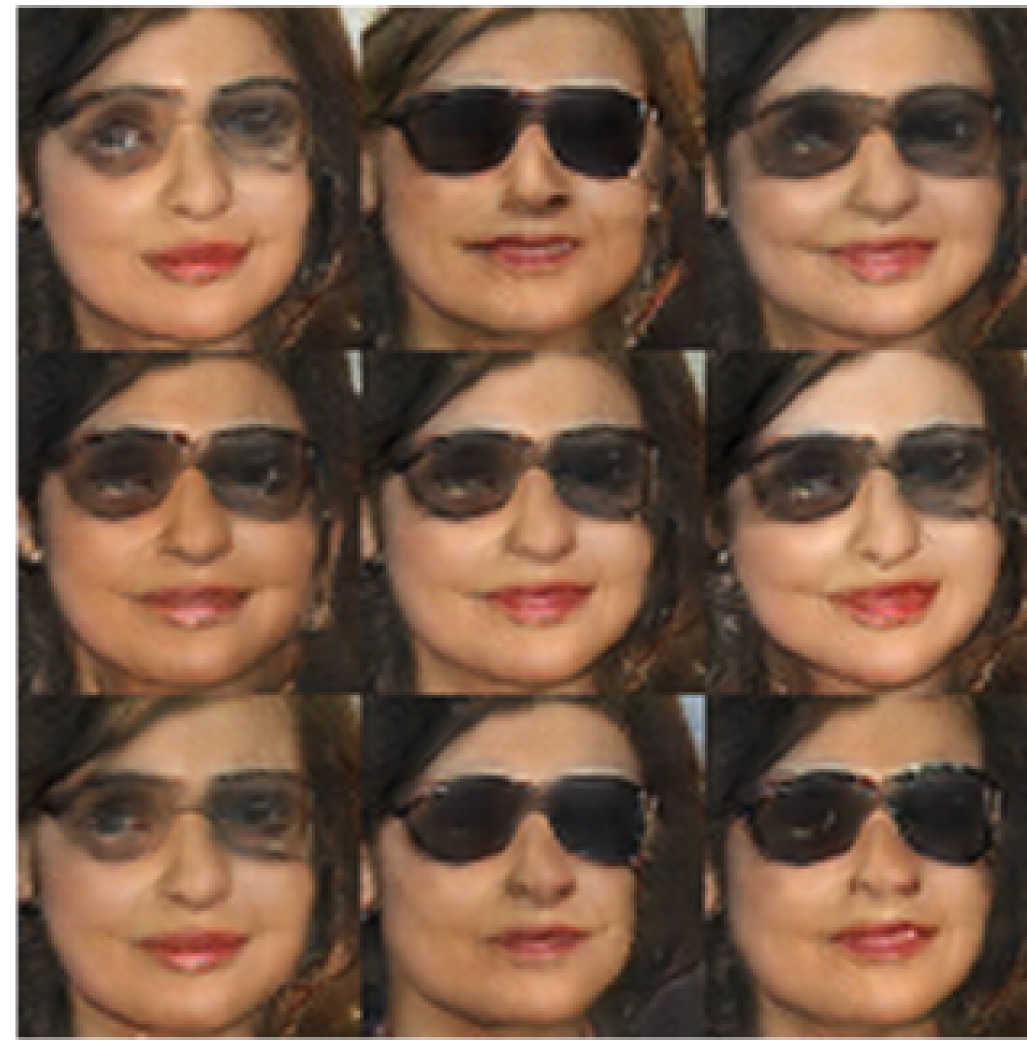


(Radford et al 2015)

DCGANs for LSUN Bedrooms

(Radford et al 2015)

Vector Space Arithmetic

 -  +  = 

Man with glasses

Man

Woman

Woman with Glasses

(Radford et al, 2015)

(Goodfellow 2016)

Batch norm in G can cause strong intra-batch correlation

(Goodfellow 2016)

Non-convergence in GANs

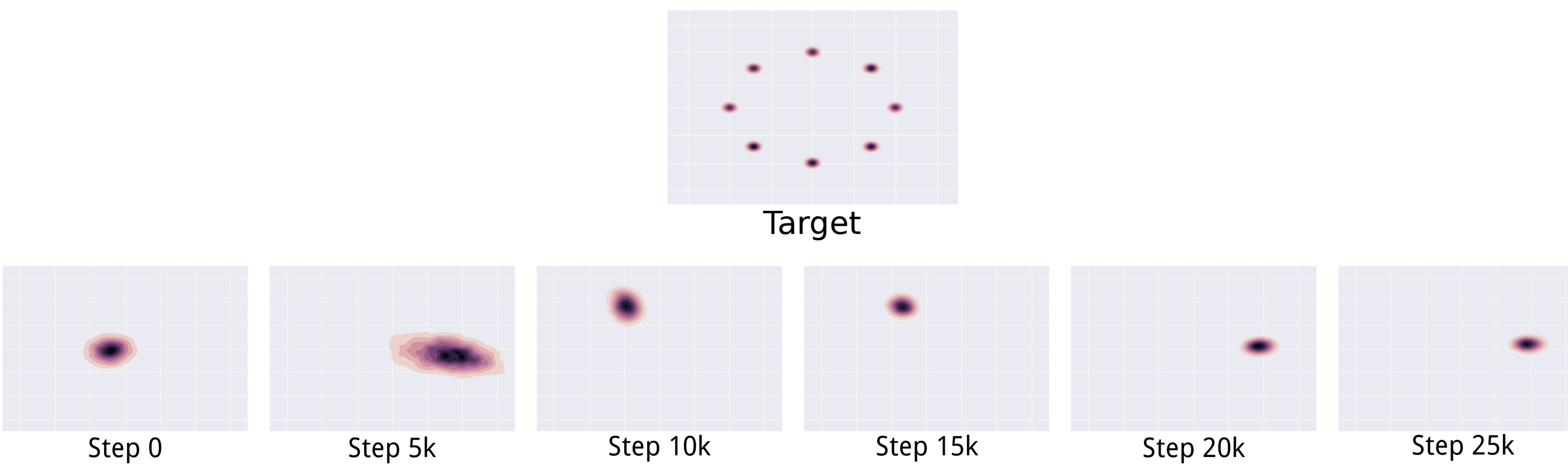
- Exploiting convexity in function space, GAN training is theoretically guaranteed to converge if we can modify the density functions directly, but:
  - Instead, we modify G (sample generation function) and D (density ratio), not densities
  - We represent G and D as highly non-convex parametric functions
- "Oscillation": can train for a very long time, generating very many different categories of samples, without clearly generating better samples
- Mode collapse: most severe form of non-convergence

(Goodfellow 2016)

Mode Collapse

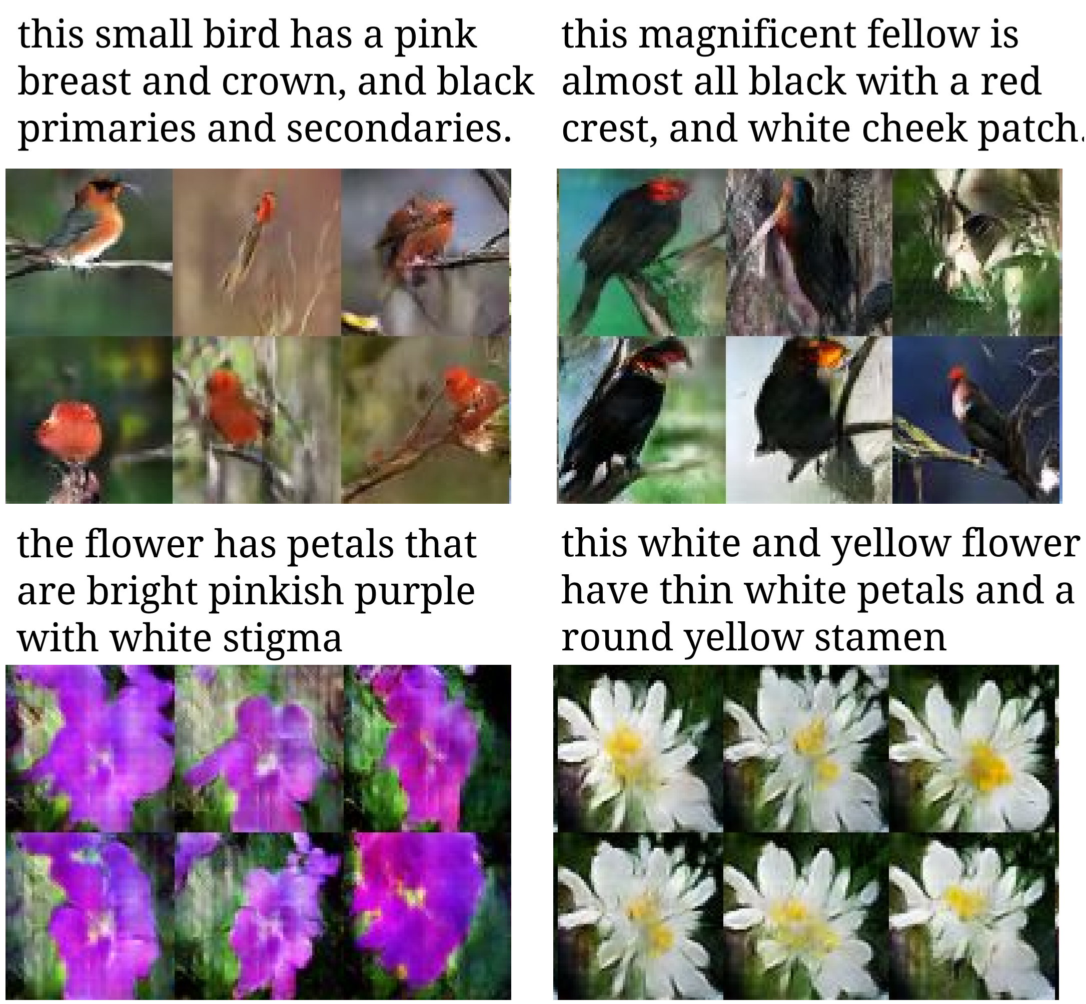
$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

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



(Metz et al 2016)

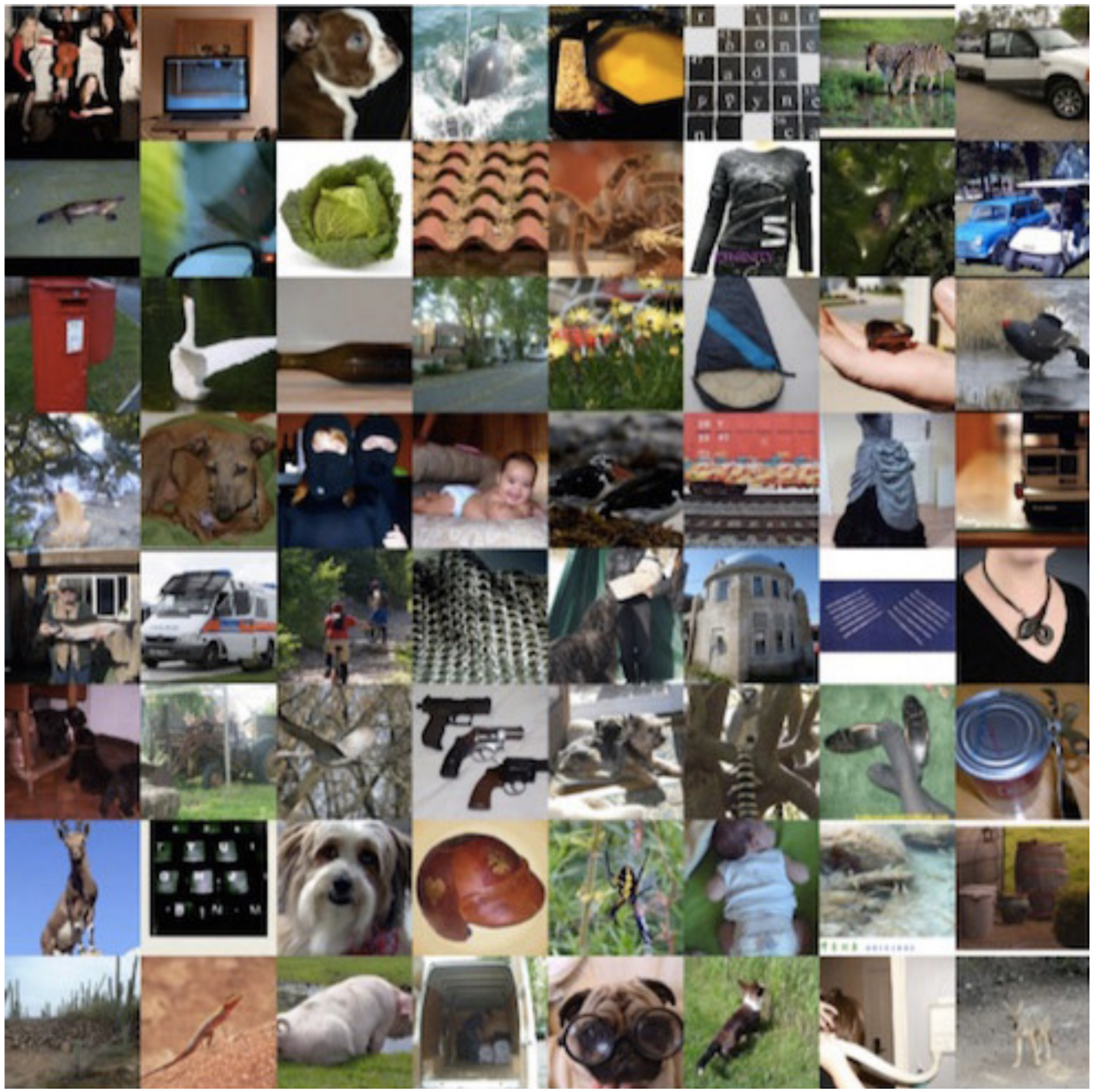
Mode collapse causes low output diversity

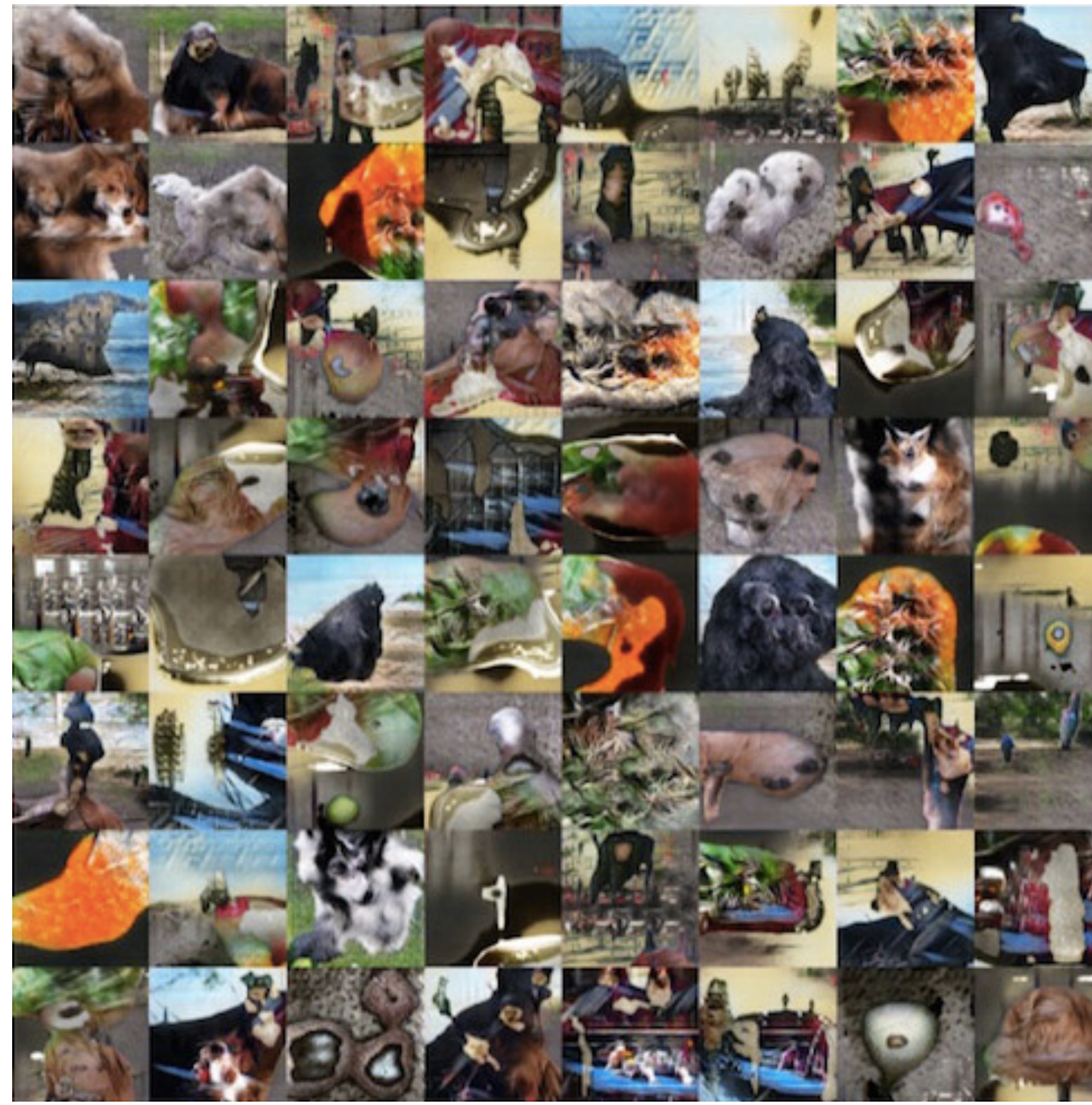


(Reed et al 2016)

(Reed et al, submitted to ICLR 2017)

Minibatch GAN on ImageNet





(Salimans et al 2016)

Problems with Counting

(Goodfellow 2016)

Problems with Global Structure



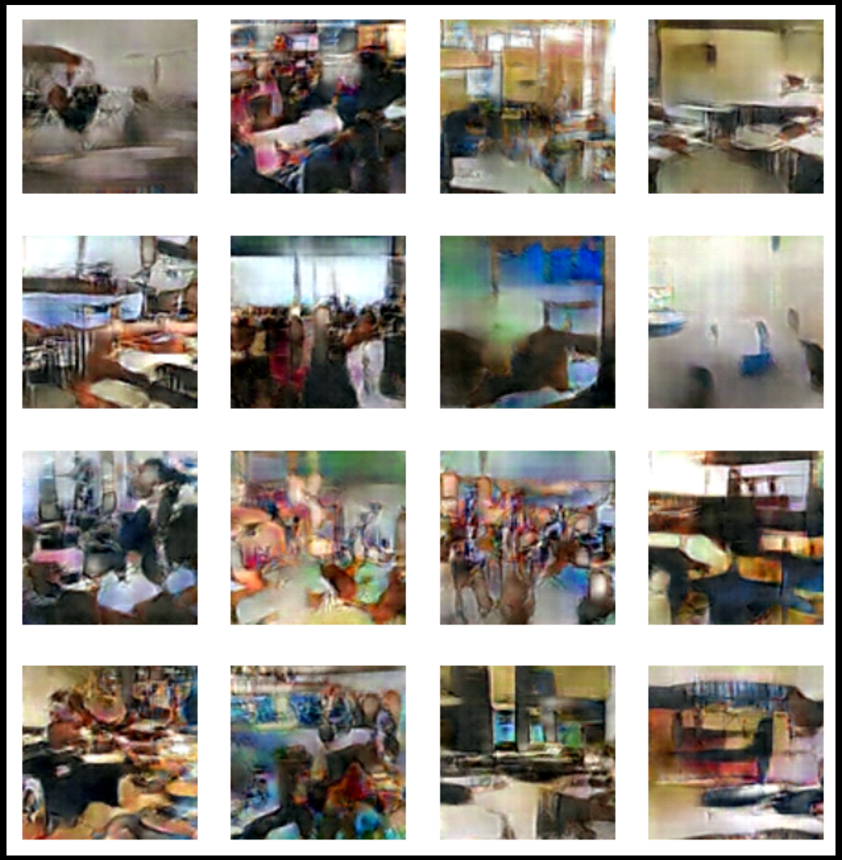



(Goodfellow 2016)

Discrete outputs

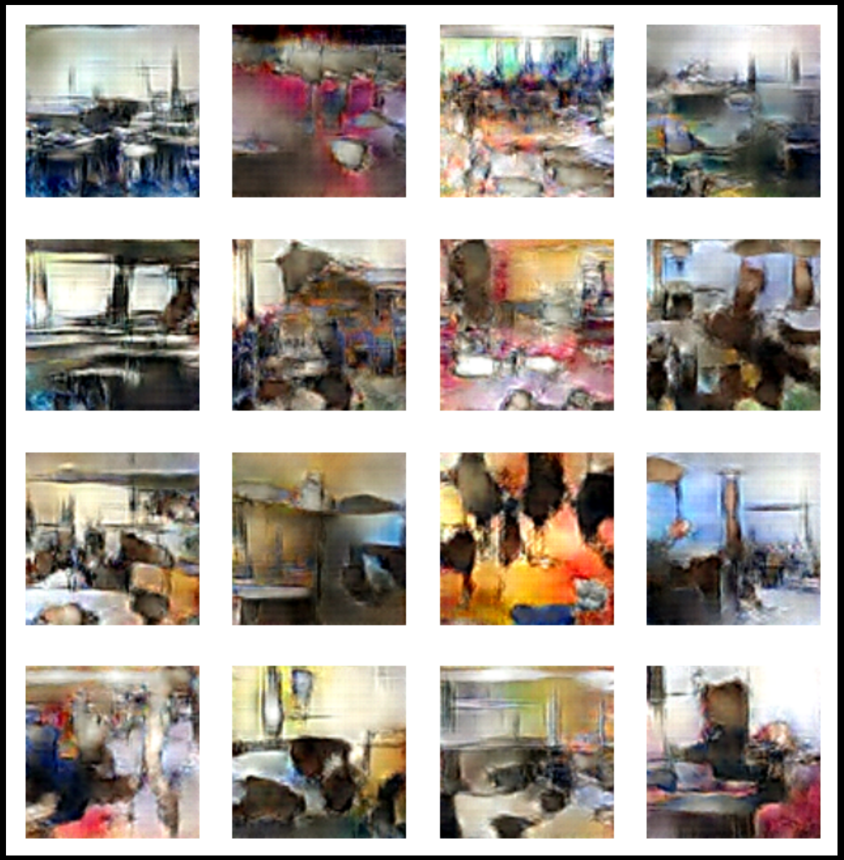
- G must be differentiable
- Cannot be differentiable if output is discrete
- Possible workarounds:
  - REINFORCE (Williams 1992)
  - Concrete distribution (Maddison et al 2016) or Gumbel-softmax (Jang et al 2016)
  - Learn distribution over continuous embeddings, decode to discrete

Can train GANs with any divergence



GAN (Jensen-Shannon)

Hellinger



Kullback-Leibler

Slide from Sebastian Nowozin

f-GAN [Nowozin et al, 2016]

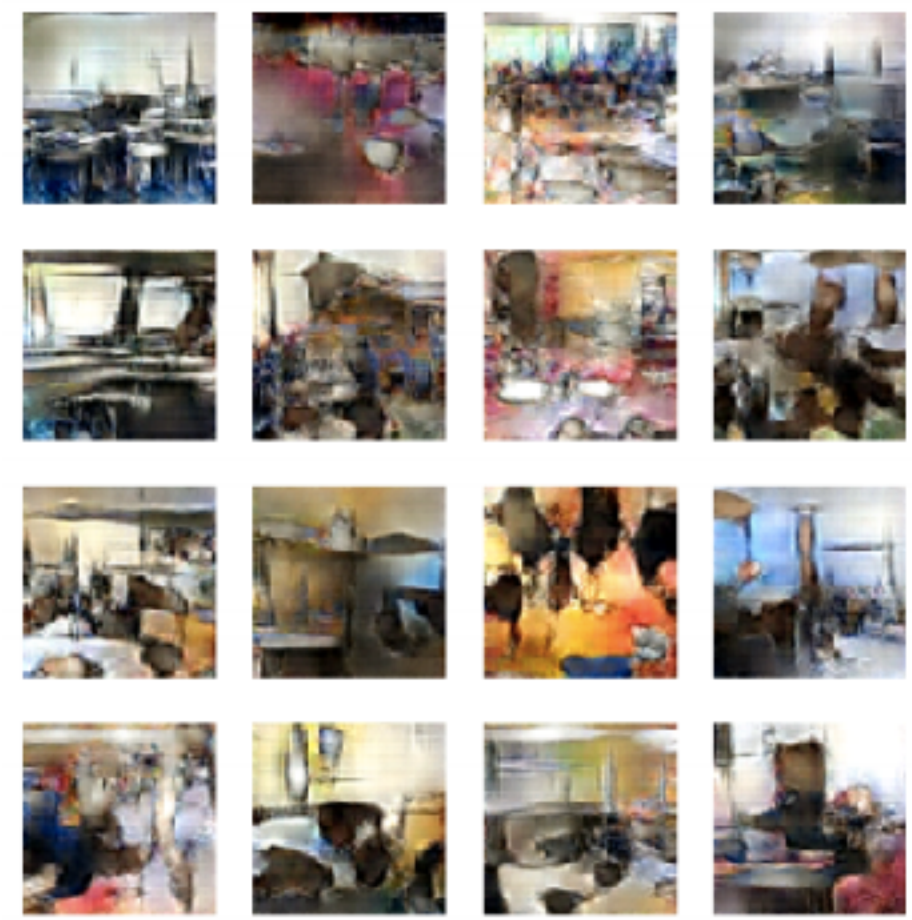
Name	Output activation g_f	dom_{f^*}	Conjugate $f^*(t)$	$f'(1)$
Total variation	$\frac{1}{2}\tanh(v)$	$-\frac{1}{2} \leq t \leq \frac{1}{2}$	t	0
Kullback-Leibler (KL)	v	$\mathbb{R}$	$\exp(t-1)$	1
Reverse KL	$-\exp(v)$	$\mathbb{R}_-$	$-1-\log(-t)$	-1
Pearson χ^2	v	$\mathbb{R}$	$\frac{1}{4}t^2+t$	0
Neyman χ^2	$1-\exp(v)$	$t<1$	$2-2\sqrt{1-t}$	0
Squared Hellinger	$1-\exp(v)$	$t<1$	$\frac{t}{1-t}$	0
Jeffrey	v	$\mathbb{R}$	$W(e^{1-t})+\frac{1}{W(e^{1-t})}+t-2$	0
Jensen-Shannon	$\log(2)-\log(1+\exp(-v))$	$t<\log(2)$	$-\log(2-\exp(t))$	0
Jensen-Shannon-weighted	$-\pi\log\pi-\log(1+\exp(-v))$	$t<-\pi\log\pi$	$(1-\pi)\log\frac{1-\pi}{1-\pi e^{t/\pi}}$	0
GAN	$-\log(1+\exp(-v))$	$\mathbb{R}_-$	$-\log(1-\exp(t))$	$-\log(2)$
α-div. ($\alpha<1$, $\alpha\neq 0$)	$\frac{1}{1-\alpha}-\log(1+\exp(-v))$	$t<\frac{1}{1-\alpha}$	$\frac{1}{\alpha}(t(\alpha-1)+1)^{\frac{\alpha}{\alpha-1}}-\frac{1}{\alpha}$	0
α-div. ($\alpha>1$)	v	$\mathbb{R}$	$\frac{1}{\alpha}(t(\alpha-1)+1)^{\frac{\alpha}{\alpha-1}}-\frac{1}{\alpha}$	0

Loss does not seem to explain why GAN samples are sharp

KL

Reverse KL

(Nowozin et al 2016)



KL samples from LSUN

Takeaway: the approximation strategy matters more than the loss

Relation to VAEs

- Same graphical model: z -> x
- VAEs have an explicit likelihood: $p(x|z)$
- GANs have no explicit likelihood
  - aka implicit models, likelihood-free models
- Can use same trick for implicit $q(z|x)$

Generalizing these ideas

- Adversarial Variational Bayes. Lars Mescheder, Sebastian Nowozin, Andreas Geiger, 2017
- Learning in Implicit Generative Models. Shakir Mohamed, Balaji Lakshminarayanan, 2016
- Variational Inference using Implicit Distributions. Ferenc Huszar, 2017
- Deep and Hierarchical Implicit Models. Dustin Tran, Rajesh Ranganath, David Blei, 2017

Takeaways

- Can train a latent-variable model without specifying a likelihood function at the last layer
- This is nice because most likelihoods (e.g. spherical Gaussians on pixels) are nonsense that we only added to smooth out the objective
- Similar to move from Exact inference to MCMC to var. inf: Don't restrict model to allow easy inference - just let a neural network clean up after.

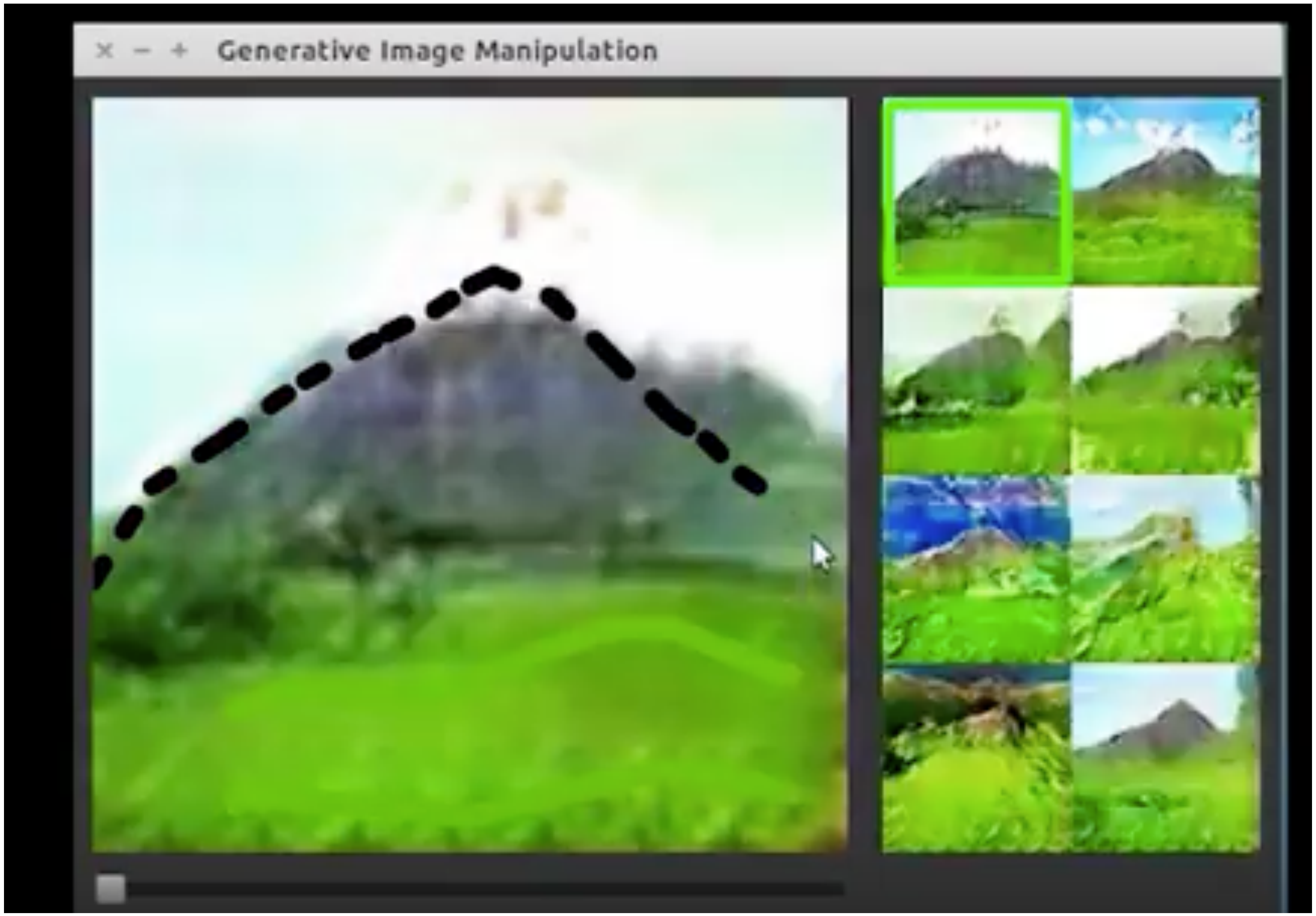
Other uses

- Same as any other generative latent-variable model

Image to Image Translation

(Isola et al 2016)

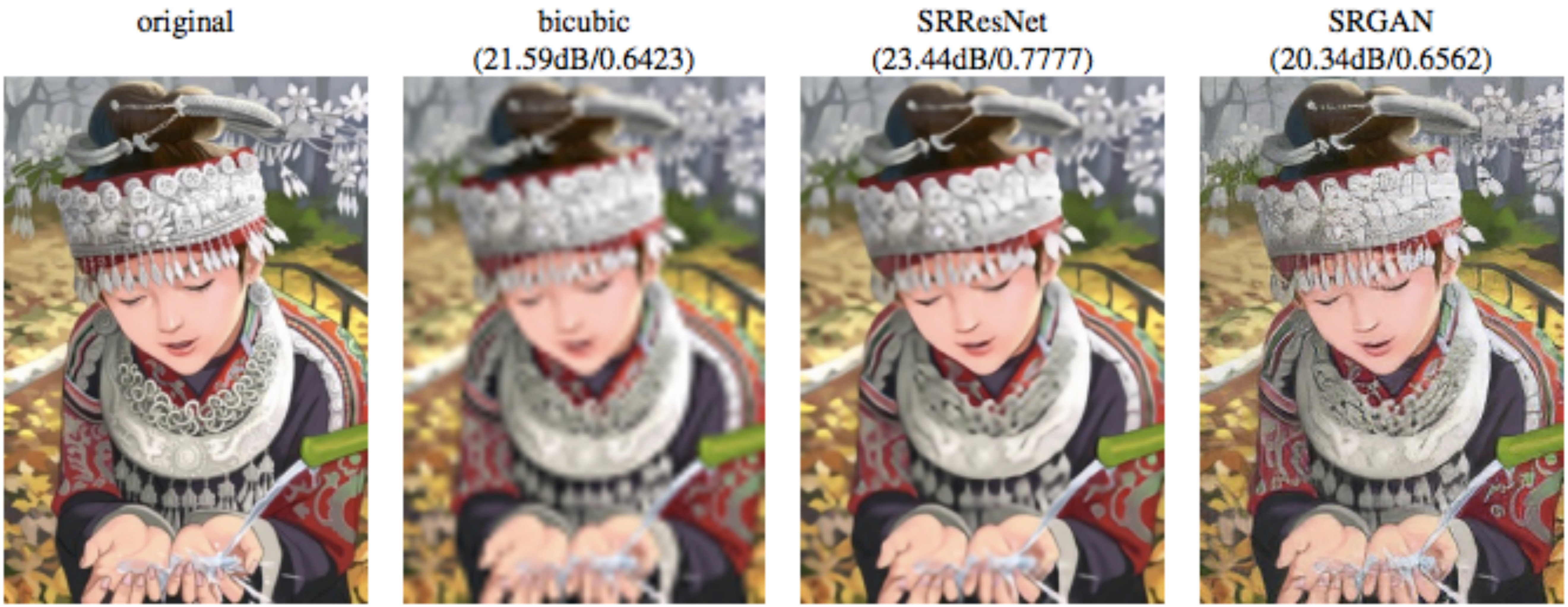
iGAN



youtube

(Zhu et al 2016)

Single Image Super-Resolution



(Ledig et al 2016)

Semi-Supervised Classification

CIFAR-10

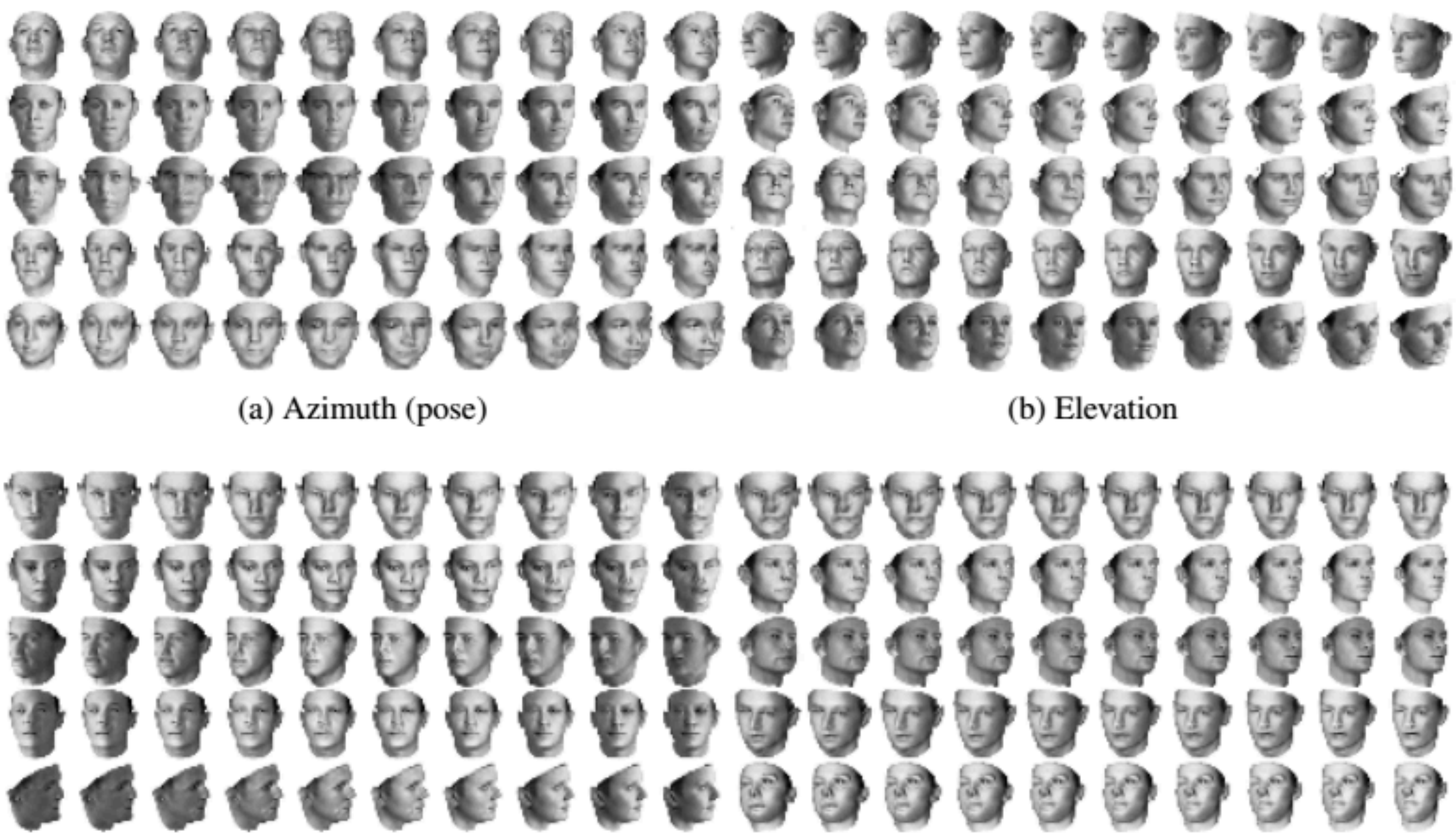
Model	Test error rate for a given number of labeled samples			
	1000	2000	4000	8000
Ladder network [24]			20.40 ± 0.47	
CatGAN [14]			19.58 ± 0.46	
Our model	21.83 ± 2.01	19.61 ± 2.09	18.63 ± 2.32	17.72 ± 1.82
Ensemble of 10 of our models	19.22 ± 0.54	17.25 ± 0.66	15.59 ± 0.47	14.87 ± 0.89

SVHN

Model	Percentage of incorrectly predicted test examples for a given number of labeled samples		
	500	1000	2000
DGN [21]		36.02 ± 0.10	
Virtual Adversarial [22]		24.63	
Auxiliary Deep Generative Model [23]		22.86	
Skip Deep Generative Model [23]		16.61 ± 0.24	
Our model	18.44 ± 4.8	8.11 ± 1.3	6.16 ± 0.58
Ensemble of 10 of our models		5.88 ± 1.0	

(Salimans et al 2016)

Learning interpretable latent codes / controlling the generation process

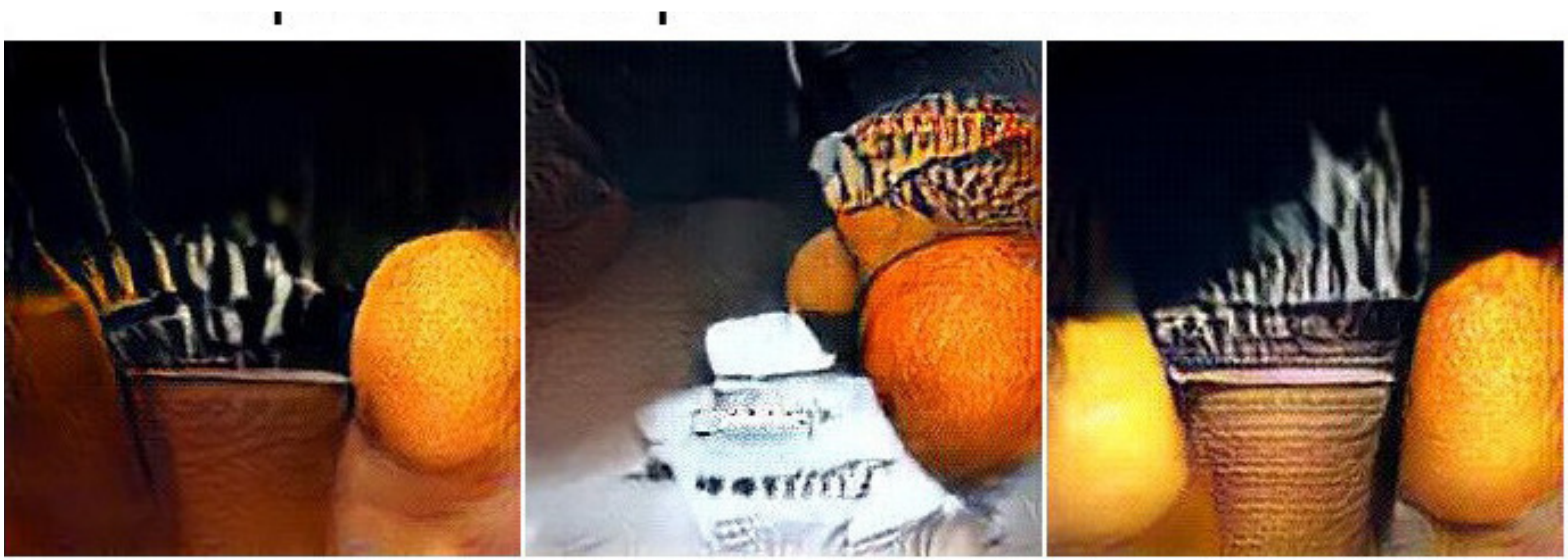


(a) Azimuth (pose) (b) Elevation

(c) Lighting (d) Wide or Narrow

InfoGAN (Chen et al 2016)

PPGN for caption to image



oranges on a table next to a liquor bottle

(Nguyen et al 2016)

Class wrap-up

ML as a bag of tricks

Fast special cases:

- K-means
- Kernel Density Estimation
- SVMs
- Boosting
- Random Forests
- K-Nearest Neighbors

Extensible family:

- Mixture of Gaussians
- Latent variable models
- Gaussian processes
- Deep neural nets
- Bayesian neural nets
- ??

Regularization as a bag of tricks

Fast special cases:

- Early stopping
- Ensembling
- L2 Regularization
- Gradient noise
- Dropout
- Expectation-Maximization

Extensible family:

- Stochastic variational inference

A language of models

- Hidden Markov Models, Mixture of Gaussians, Logistic Regression
- These are simply “sentences” - examples from a language of models.
- We will try to show larger family, and point out common special cases.

AI as a bag of tricks

Russel and Norvig's parts of AI:

- Machine learning
- Natural language processing
- Knowledge representation
- Automated reasoning
- Computer vision
- Robotics

Extensible family:

- Deep probabilistic latent-variable models + decision theory

Where are we now?

- Open research areas:
  - Optimization (especially minimax)
  - Generalizing style transfer
  - Bayesian GANs, VAEs
  - Model-based RL
  - Bayesian neural networks
  - Learning discrete latent structure
  - Learning discrete model structure

Thanks a lot!