Generative Adversarial Networks (GANs)

Hossein Azizpour

Most of the slides are courtesy of Dr. **Ian Goodfellow** (Research Scientist at OpenAI) and from his presentation at NIPS 2016 tutorial

Note. I am generally knowledgeable in deep learning but not particularly an expert for GANs

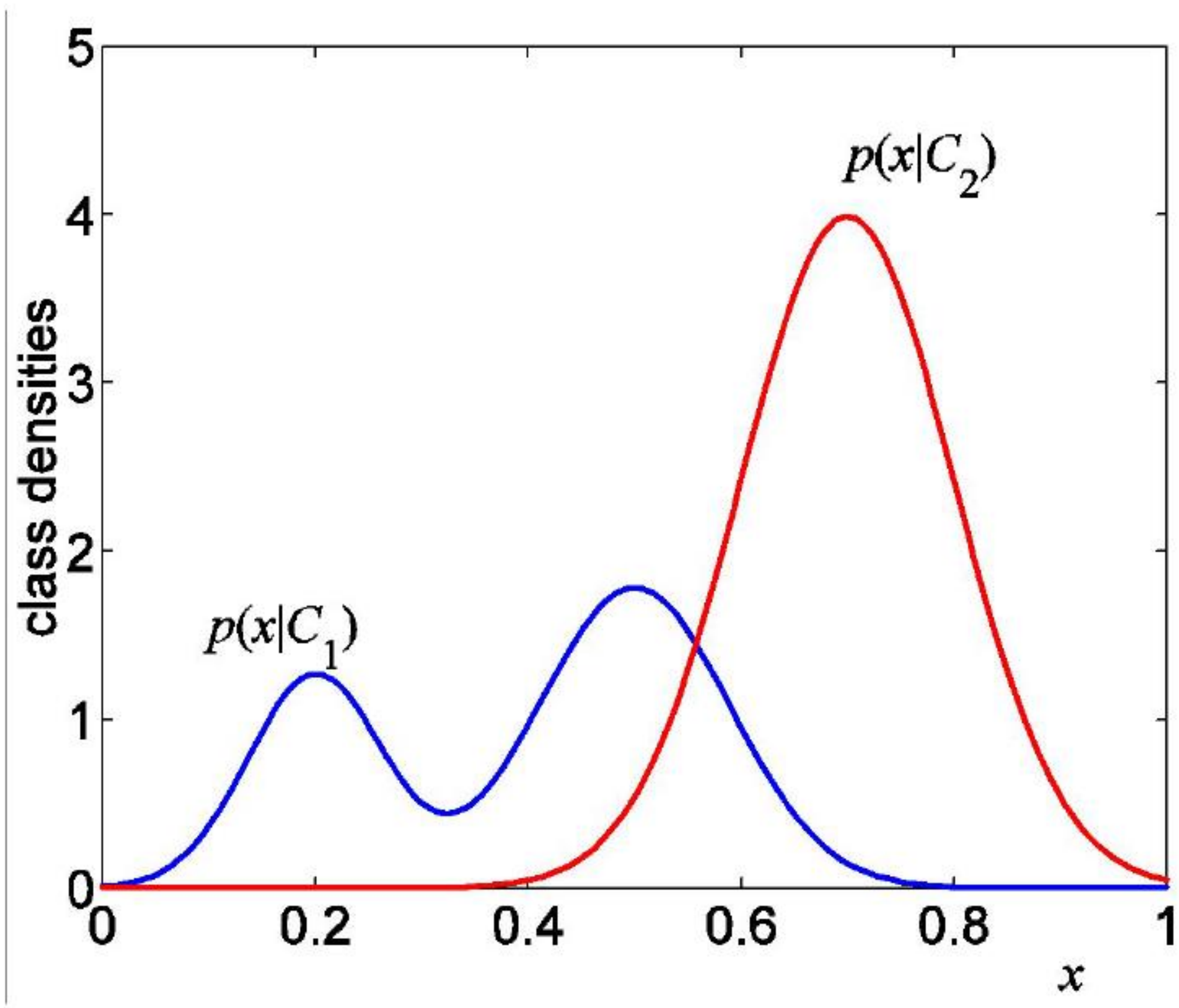
GANs

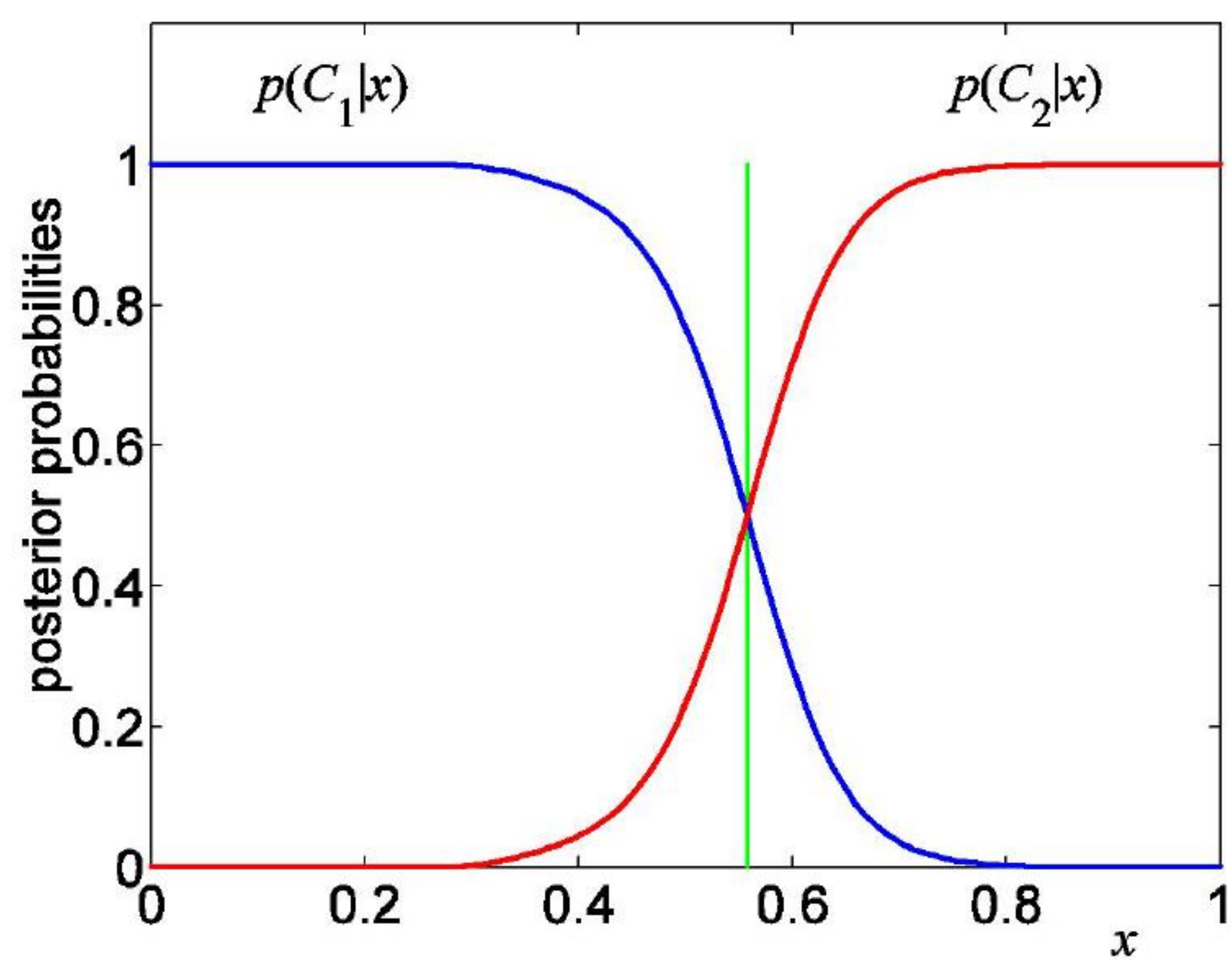
- Whatever you learned about deep learning in general applies to GANs to a higher degree
  - the theoretical ground is not comprehensive
  - we do not understand the inner workings very well
  - there is a lot to learn from practice
  - it is “h a r d” to train a GAN which looks successful to the human eye
  - it fails the vast majority of the time if you plainly use the original GAN formulation
  - the results are really cool!

(Made by Goodfellow
edited by Azizpour)

Modeling

- Discriminative Modeling $P_{Y|X}(\mathrm{y}|\mathrm{x})$ or $f(x, y)$
- Generative Modeling $P_{X|Y}(\mathrm{x}|\mathrm{y})$ and $P_X(x)$

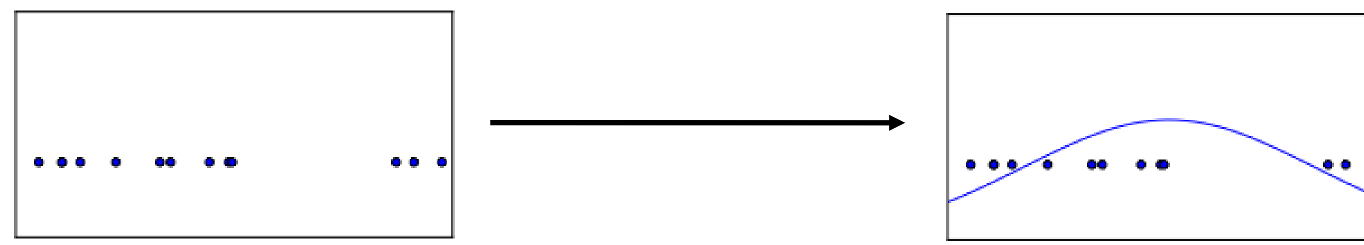




Generative Modeling

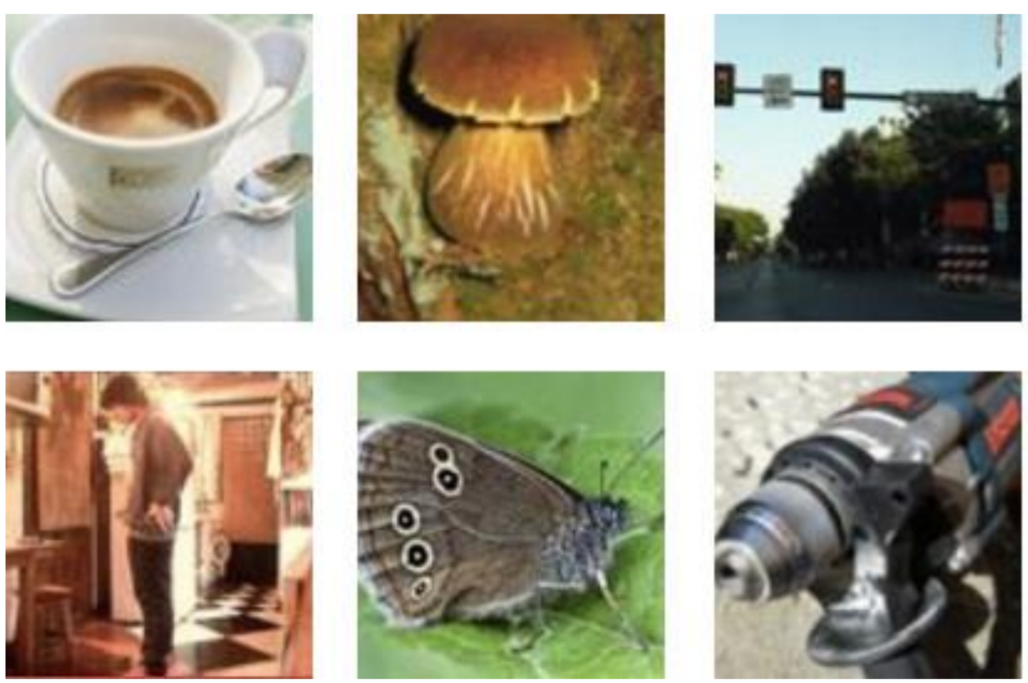
- Density estimation

$P_X(x)$, $P_{X|Y}(x|y)$, $\sum_z P_{X|Z}(x|z)P_{Z|Y}(z|y)$

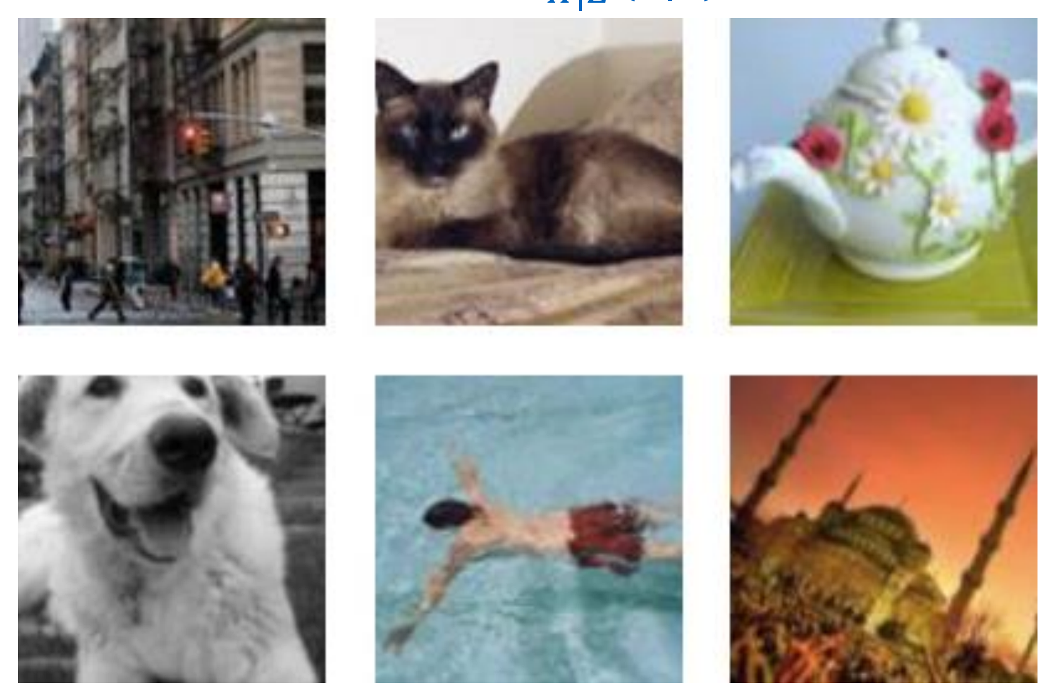


- Sample generation

$\hat{x} \sim P_{X|Z}(x|z)$



Training examples



Model samples

(Made by Goodfellow
edited by Azizpour)

Content

- Why study generative modeling?
- How do generative models work? How do GANs compare to others?
- How do GANs work?
- Tips and tricks
- Research frontiers

(Made by Goodfellow
edited by Azizpour)

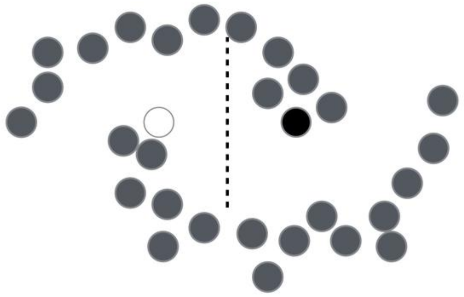
Why study generative models?

- Theoretical Reason:
  - Excellent test of our ability to use high-dimensional, complicated probability distributions

- Practical Reasons:
  - Simulate possible futures for planning or simulated RL

  - Missing data

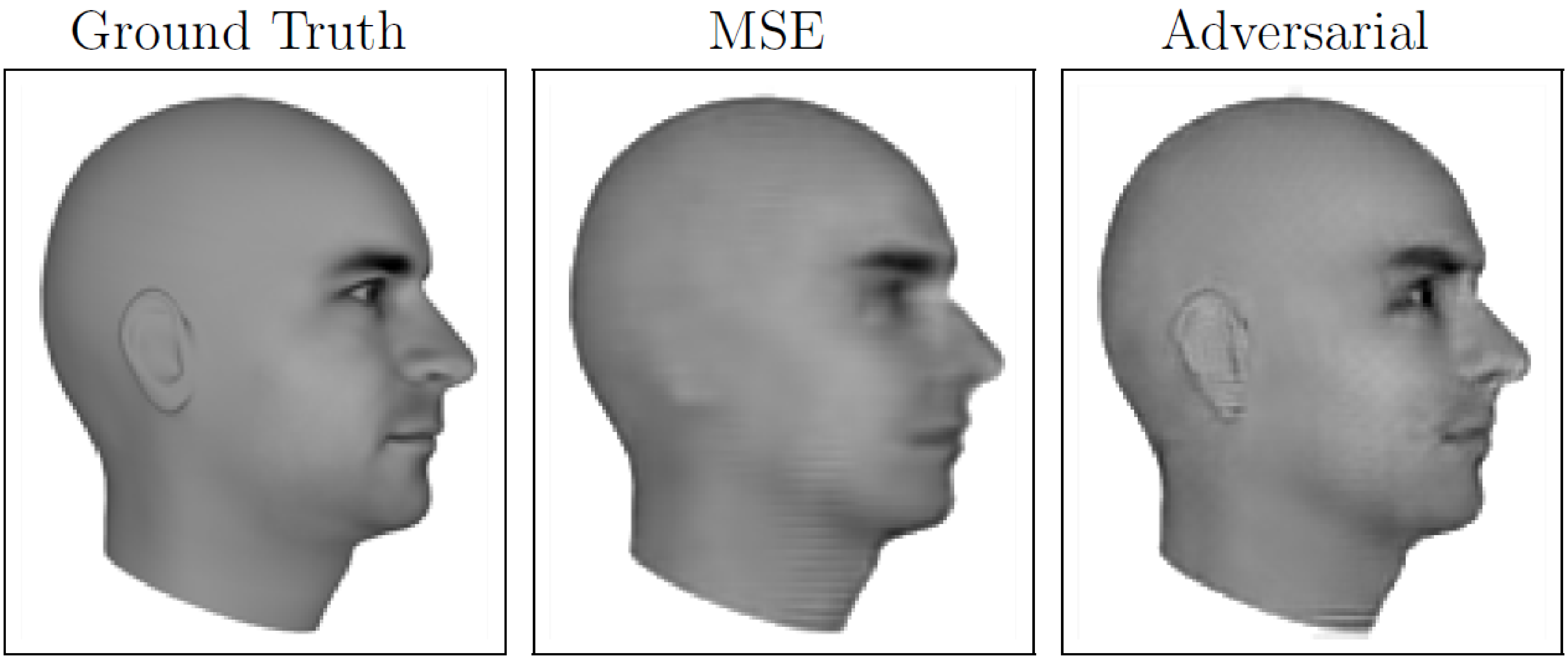
  - Semi-supervised learning



  - Multi-modal outputs
  - Realistic generation tasks

(Made by Goodfellow
edited by Azizpour)

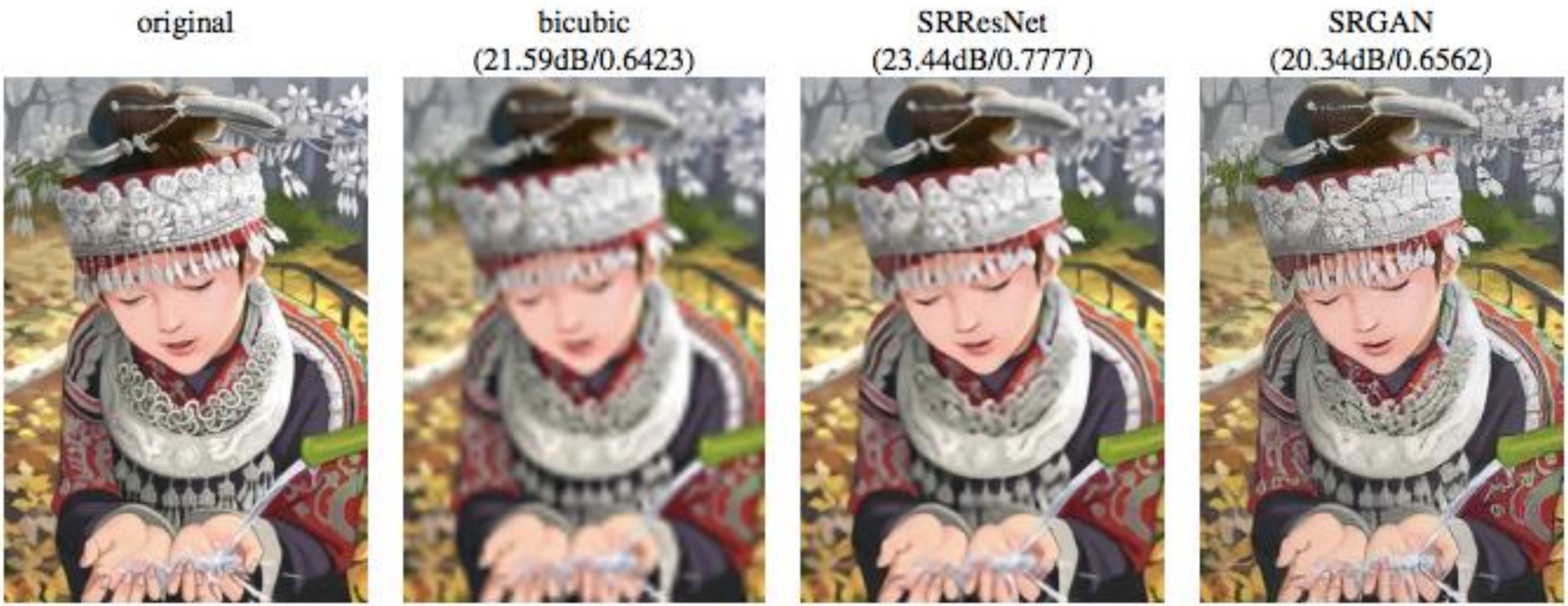
Next Video Frame Prediction



(Lotter et al 2016)

(Made by Goodfellow
edited by Azizpour)

Single Image Super-Resolution



(Ledig et al 2016)

(Made by Goodfellow
edited by Azizpour)

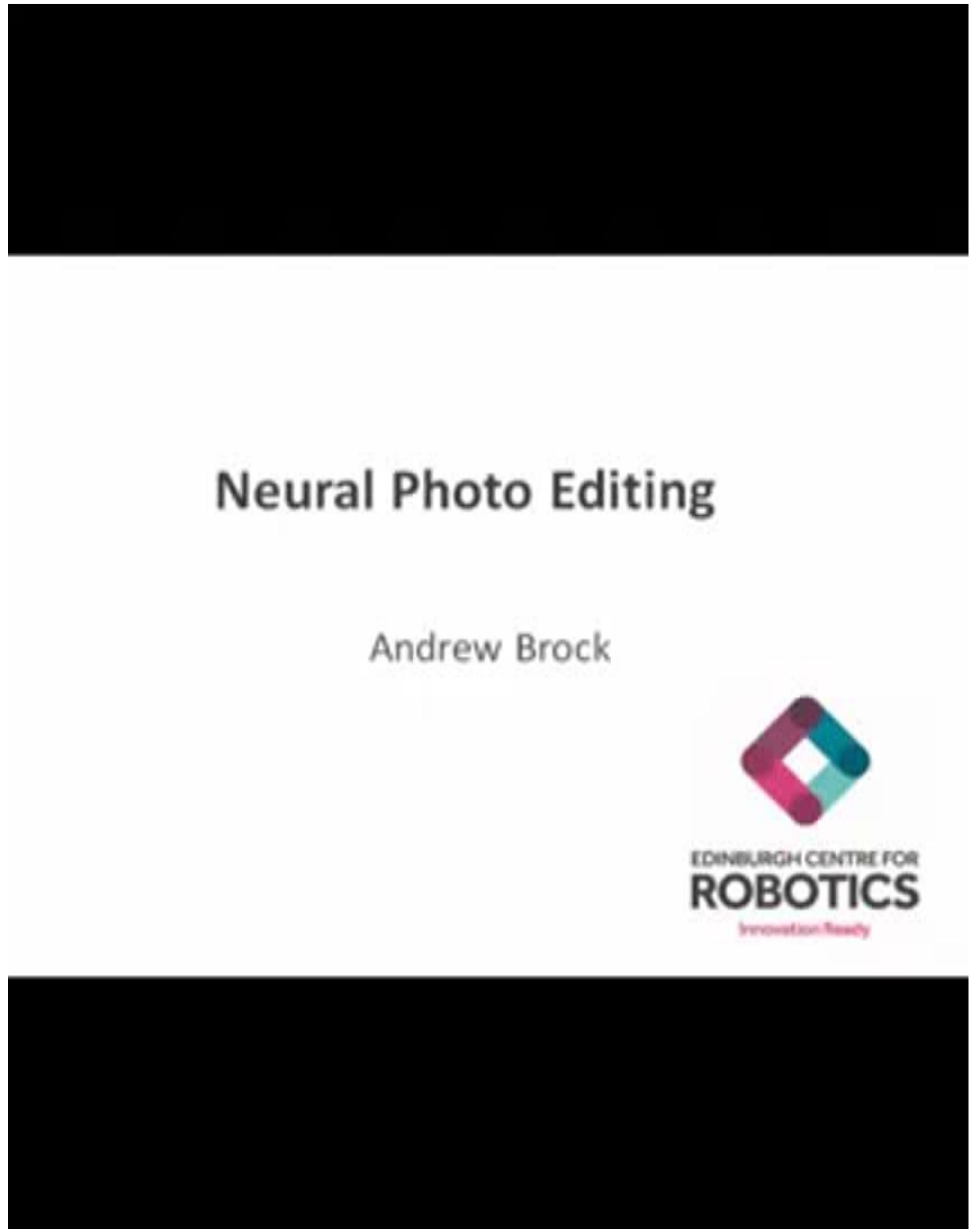
iGAN

youtube

Zhu et al **iGAN:** Generative Visual Manipulation on the Natural Image Manifold (2016) collaboration between Adobe and Berkeley

(Made by Goodfellow
edited by Azizpour)

Introspective Adversarial



(Brock et al 2016) youtube

(Made by Goodfellow
edited by Azizpour)

Image to Image Translation

(Isola et alImage-to-Image Translation with Conditional Adversarial Networks 2016)

(Made by Goodfellow
edited by Azizpour)

Content

- Why study generative modeling?
- How do generative models work? How do GANs compare to others?
- How do GANs work?
- Tips and tricks
- Research frontiers
- Combining GANs with other methods

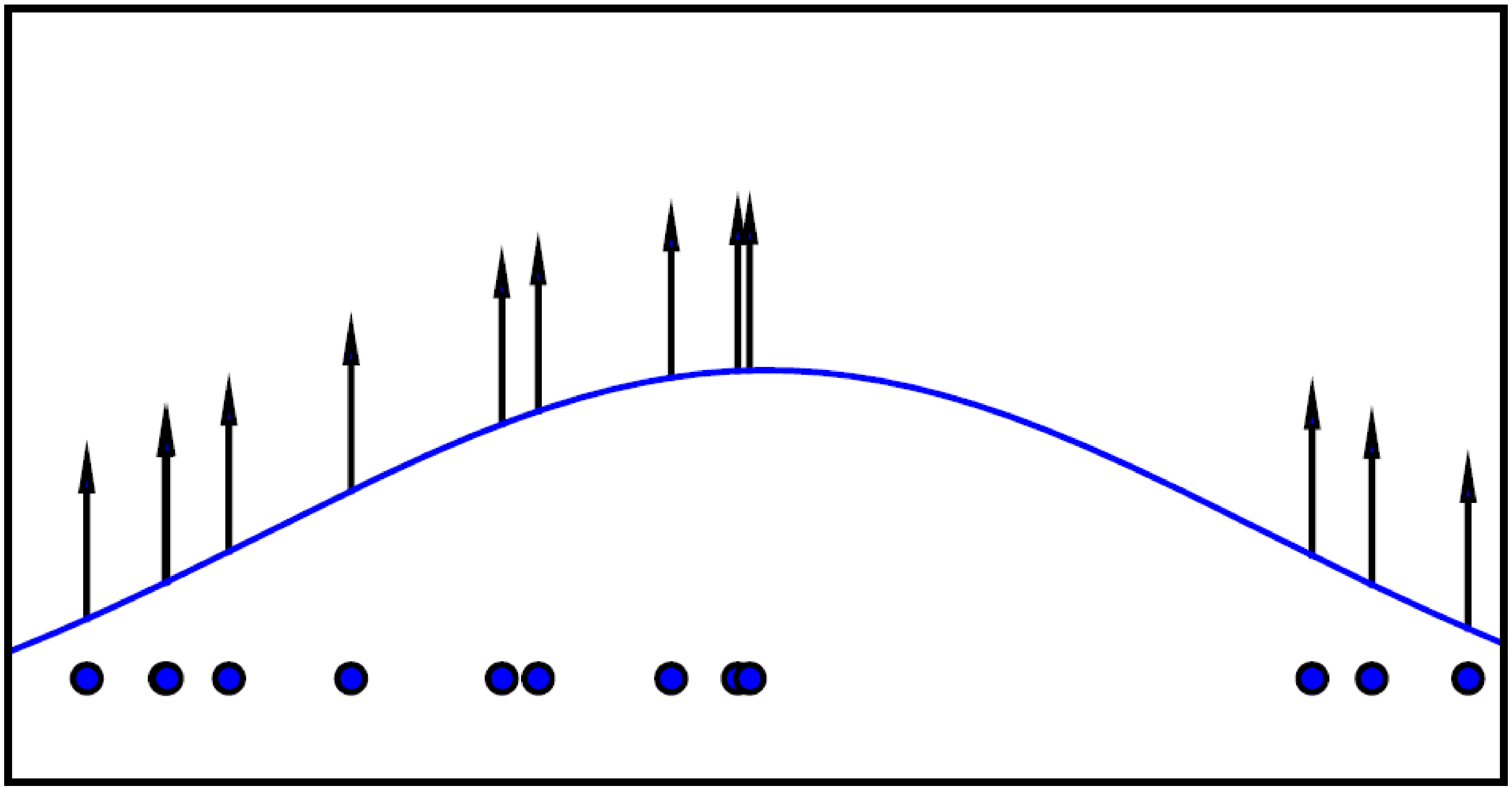
(Made by Goodfellow
edited by Azizpour)

Loss functions

- Two approaches:
  - Increase log-likelihood of data (ML)
  - Have your network learn it’s loss function! (Adversarial Learning)

(Made by Goodfellow
edited by Azizpour)

Maximum Likelihood

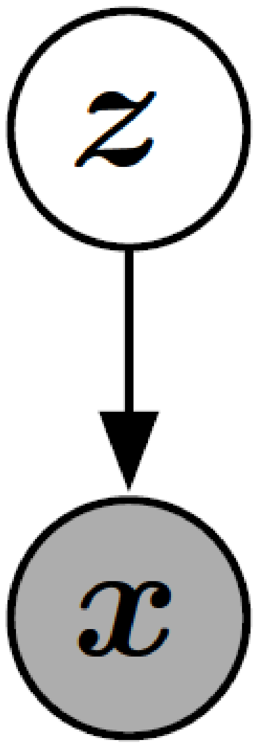


$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, E_{x\sim P_{data}} \log P_{model}(x;\theta)$$

(Made by Goodfellow
edited by Azizpour)

Variational Autoencoder

(Kingma and Welling 2013, Rezende et al 2014)



$$\log P(x) \geq \log P(x) - D_{KL}(Q(z)||P(z|x)E_{x\sim Q}$$
$$= P(x,z) + H(Q)$$

CIFAR-10 samples
(Kingma et al 2016)

Disadvantages:
-Not asymptotically consistent unless *q* is perfect
-Samples tend to have lower quality

(Made by Goodfellow
edited by Azizpour)

Adversarial Learning

- The output distribution of your desired input-output mapping function is non-trivial e.g. (class-conditional) image manifold
- You have access to samples of the output distribution
- Then you can learn your loss function to say how much it is possible to tell apart the samples of the output distribution from the samples of your mapping function

(Made by Goodfellow
edited by Azizpour)

GANs

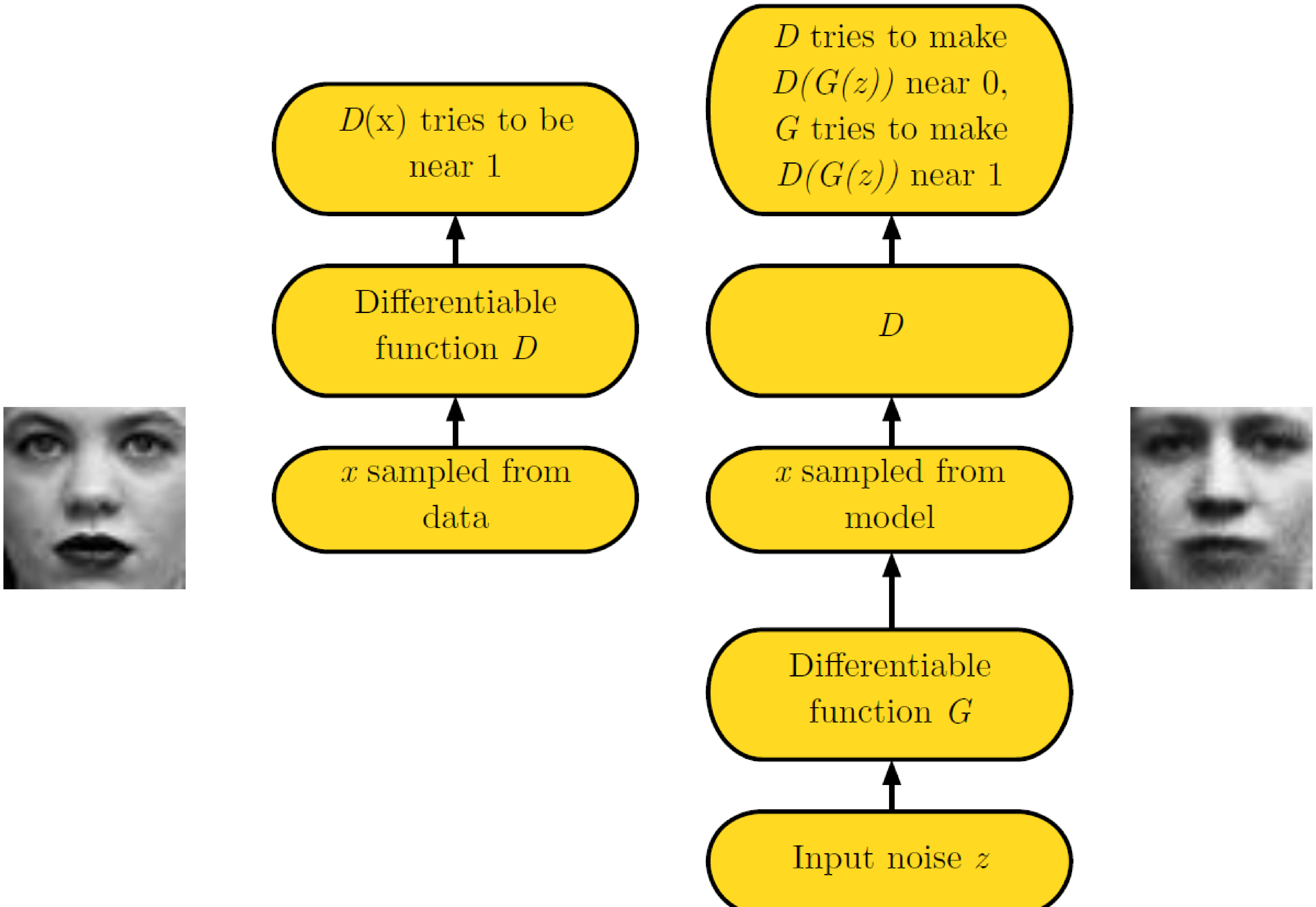
- Use a latent code
- Are unstable to train
- Often regarded as producing the best samples
  - No good way to quantify this

(Made by Goodfellow
edited by Azizpour)

Content

- Why study generative modeling?
- How do generative models work? How do GANs compare to others?
- How do GANs work?
- Tips and tricks
- Research frontiers

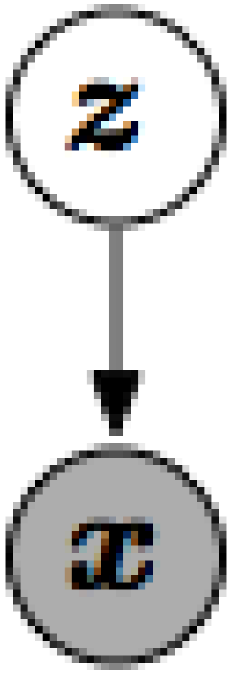
Adversarial Nets Framework



(Made by Goodfellow
edited by Azizpour)

Generator Network

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



- Must be differentiable
- No invertibility requirement
- Trainable for any size of ***z***
- **Can make *x* conditionally Gaussian given** *z* **but need not do so**

(Made by Goodfellow
edited by Azizpour)

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.

(Made by Goodfellow
edited by Azizpour)

Minimax Game

$$J^{(D)} = -\frac{1}{2} E_{x \sim P_{data}} \log D(x) - \frac{1}{2} E_x \log(1 - D\big(G(z)\big))$$

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

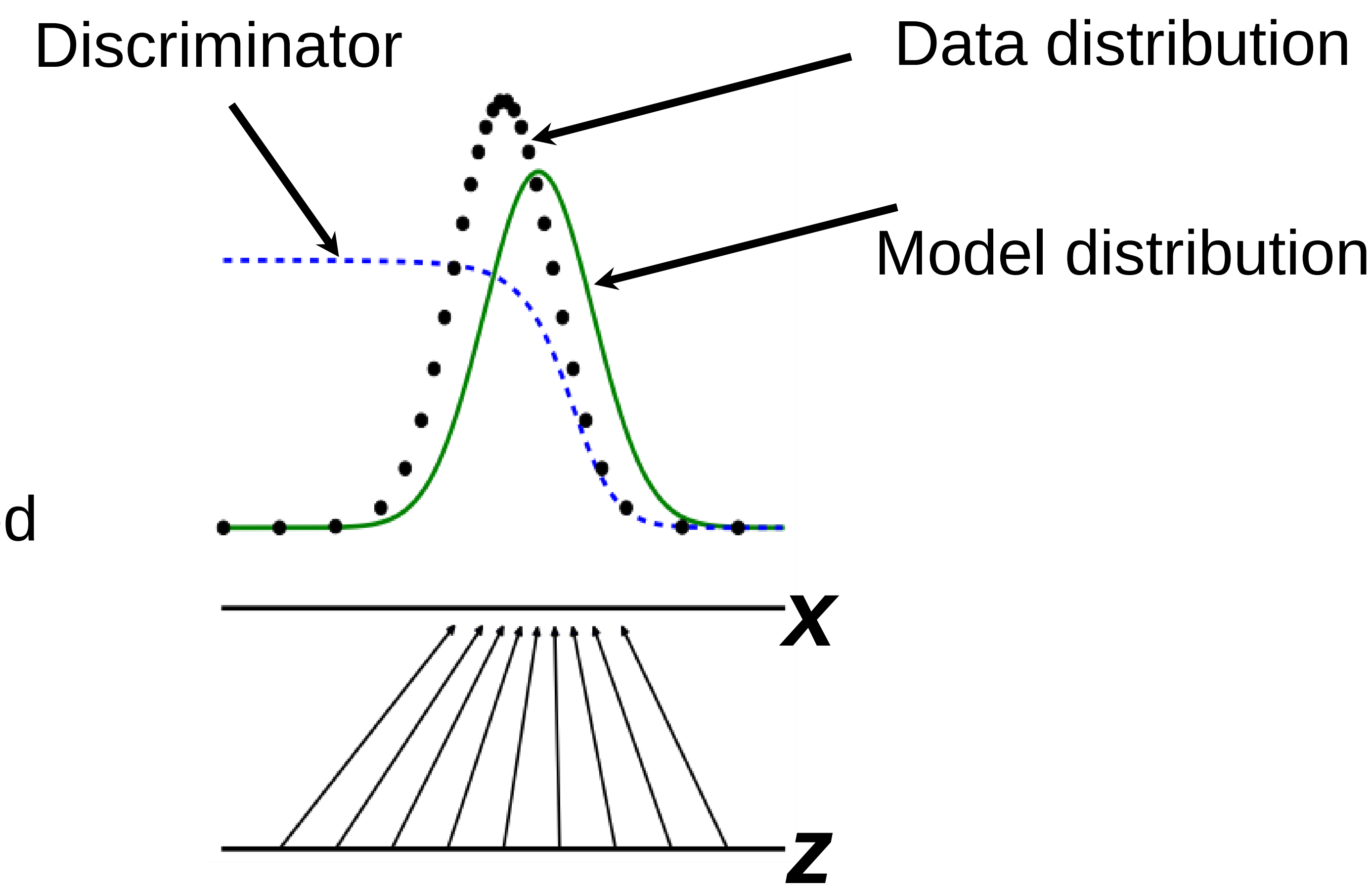
- Equilibrium is a saddle point of the discriminator loss
- Generator minimizes the log-probability of the discriminator being correct

(Made by Goodfellow
edited by Azizpour)

Discriminator Strategy

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

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



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

(Made by Goodfellow
edited by Azizpour)

Non-Saturating Game

$$J^{(D)} = -\frac{1}{2}E_{x\sim P_{data}} \log D(x) - \frac{1}{2}E_x \log(1 - D(G(z)))$$

$$J^{(G)} = -\frac{1}{2}E_z \log D(G(z))$$

- 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

(Made by Goodfellow
edited by Azizpour)

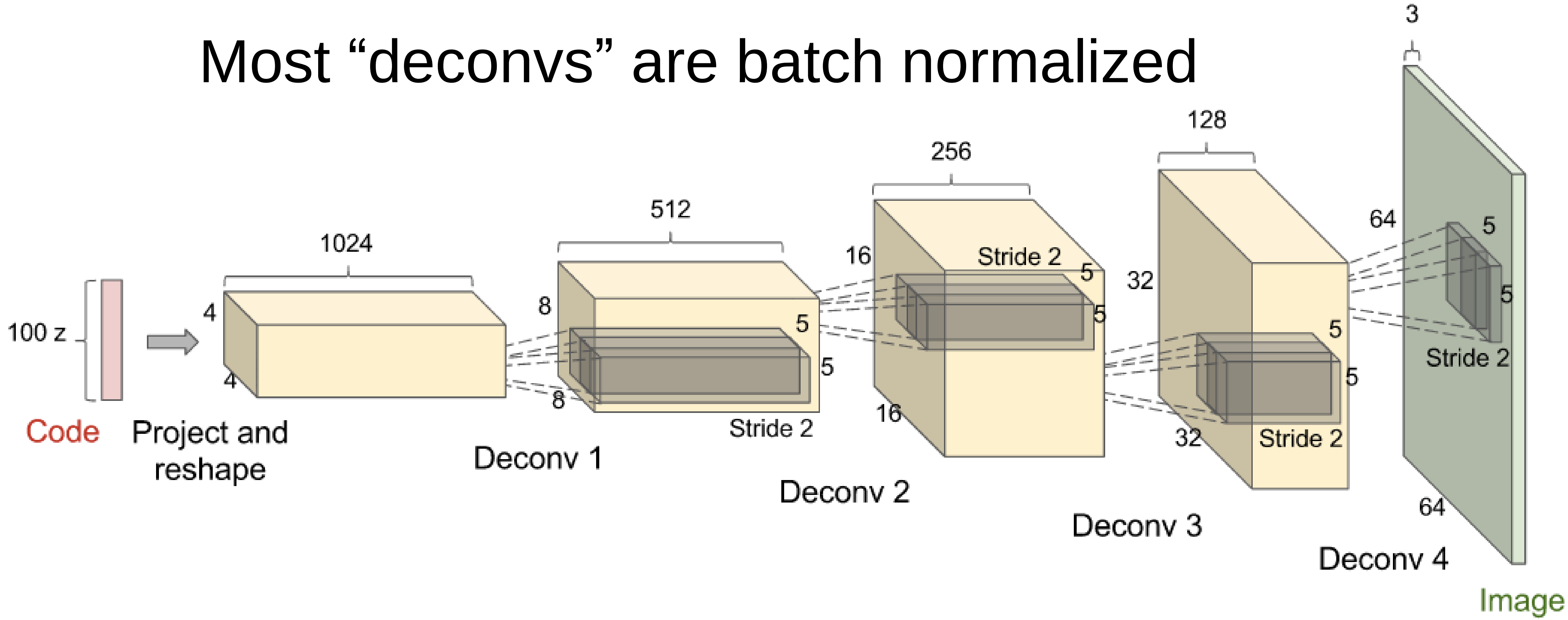
Non-Saturating Game

- How does it work in practice?

(Made by Goodfellow
edited by Azizpour)

DCGAN Architecture

Most "deconvs" are batch normalized



(Radford et al 2015)

(Made by Goodfellow
edited by Azizpour)

DCGANs for LSUN Bedrooms

(Radford et al 2015)

(Made by Goodfellow
edited by Azizpour)

Vector Space Arithmetic

-

+

=

Man
with glasses

Man

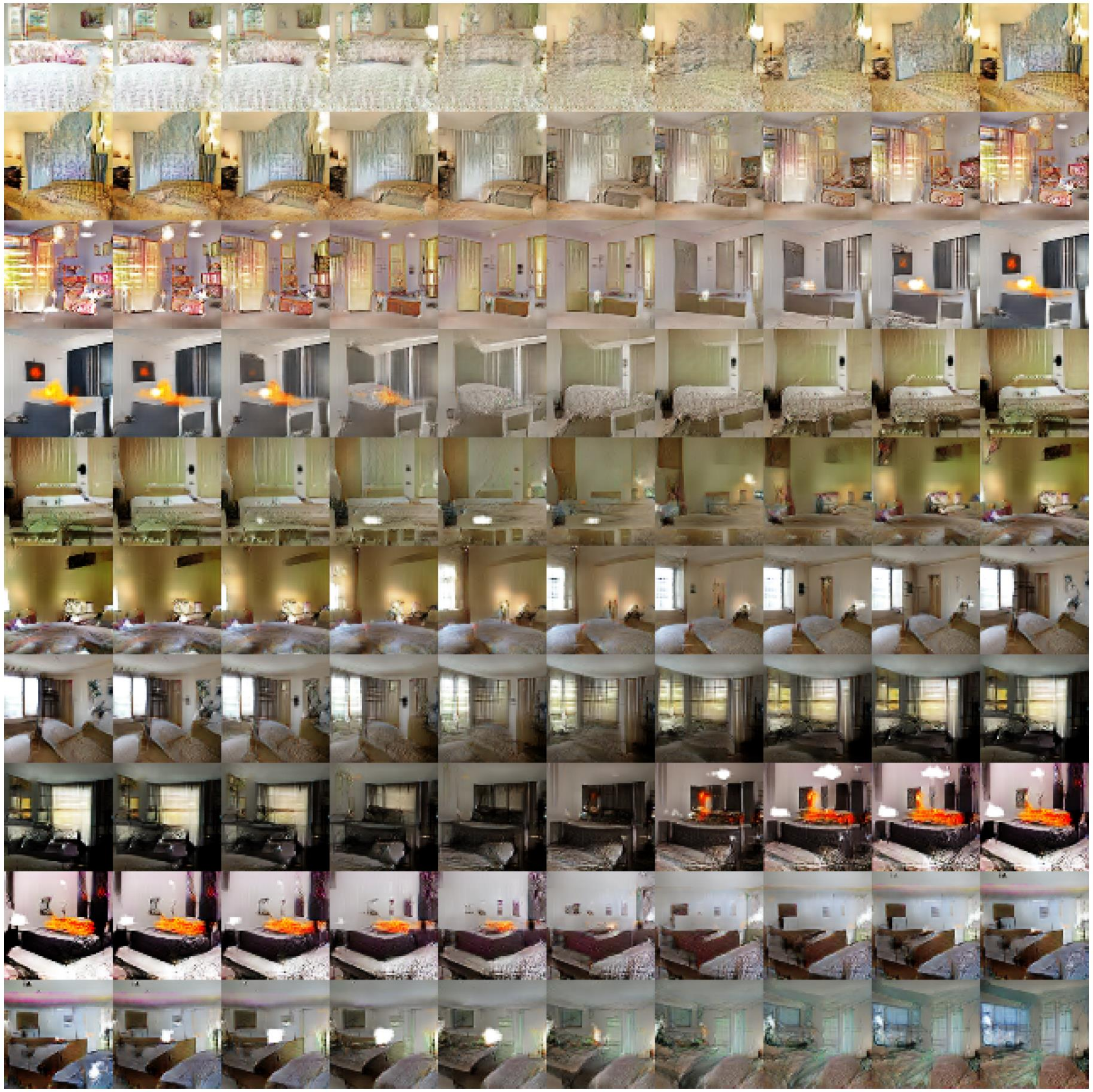
Woman

Woman with Glasses

(Radford et al, 2015)

(Made by Goodfellow
edited by Azizpour)

Latent Space Interpolation



(Made by Goodfellow
edited by Azizpour)

Content

- Why study generative modeling?
- How do generative models work? How do GANs compare to others?
- How do GANs work?
- Tips and tricks
- Research frontiers

(Made by Goodfellow
edited by Azizpour)

Labels improve subjective sample quality

- Learning a conditional model $p(x|y)$ often gives much better samples from all classes than learning $p(x)$ does (Denton et al 2015)
- Even just learning $p(x,y)$ makes samples from $p(x)$ look much better to a human observer (Salimans et al 2016)
- Note: this defines three categories of models (no labels, trained with labels, generating condition on labels) that should not be compared directly to each other

(Made by Goodfellow
edited by Azizpour)

One-sided label smoothing

- Default discriminator cost:

```
cross_entropy(1., discriminator(data))
+ cross_entropy(0., discriminator(samples))
```

- One-sided label smoothed cost (Salimans et al 2016):

```
cross_entropy(.9, discriminator(data))
+ cross_entropy(0., discriminator(samples))
```

Do not smooth negative labels

cross_entropy(1.-alpha, discriminator(data))
+ cross_entropy(beta, discriminator(samples))

Reinforces current generator behavior

$$D(\boldsymbol{x}) = \frac{(1-\alpha)p_{\text{data}}(\boldsymbol{x}) + \beta p_{\text{model}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_{\text{model}}(\boldsymbol{x})}$$

(Made by Goodfellow
edited by Azizpour)

Benefits of label smoothing

- Good regularizer (Szegedy et al 2015)
- Does not reduce classification accuracy, only confidence
- Benefits specific to GANs:
    - Prevents discriminator from giving very large gradient signal to generator
    - Prevents extrapolating to encourage extreme samples

(Made by Goodfellow
edited by Azizpour)

Batch Norm

- Given inputs $\boldsymbol{X}=\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, .., \boldsymbol{x}^{(m)}\}$
- **Compute mean and standard deviation of features of *X***
- **Normalize features (subtract mean, divide by standard deviation)**
- **Normalization operation is part of the graph**
  - **Backpropagation computes the gradient through the normalization**
  - **This avoids wasting time repeatedly learning to undo the normalization**

Batch norm in *G* can cause strong intra-batch correlation

(Made by Goodfellow
edited by Azizpour)

Reference Batch Norm

- Fix a *reference batch* $R=\{\boldsymbol{r}^{(1)}, \boldsymbol{r}^{(2)}, .., \boldsymbol{r}^{(m)}\}$
- Given new inputs $\boldsymbol{X}=\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, .., \boldsymbol{x}^{(m)}\}$
- **Compute mean and standard deviation of features of *R***
  - Note that though *R* does not change, the feature values change when the parameters change
- Normalize the features of *X* using the mean and standard deviation from *R*
- **Every $\boldsymbol{x}^{(i)}$ is always treated the same, regardless of which other examples appear in the minibatch**

(Made by Goodfellow
edited by Azizpour)

Balancing *G* and *D*

- Usually the discriminator “wins”
- This is a good thing—the theoretical justifications are based on assuming *D* is perfect
- Usually *D* is bigger and deeper than *G*
- Sometimes run *D* more often than *G*. Mixed results.
- Do not try to limit *D* to avoid making it “too smart”
  - Use non-saturating cost
  - Use label smoothing

(Made by Goodfellow
edited by Azizpour)

Tips and Tricks

- There are many heuristics listed on torch GAN page, collected by Soumith Chintala

Content

- Why study generative modeling?
- How do generative models work? How do GANs compare to others?
- How do GANs work?
- Tips and tricks
- Research frontiers
- Combining GANs with other methods

(Made by Goodfellow
edited by Azizpour)

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

(Made by Goodfellow
edited by Azizpour)

Non-convergence in GANs

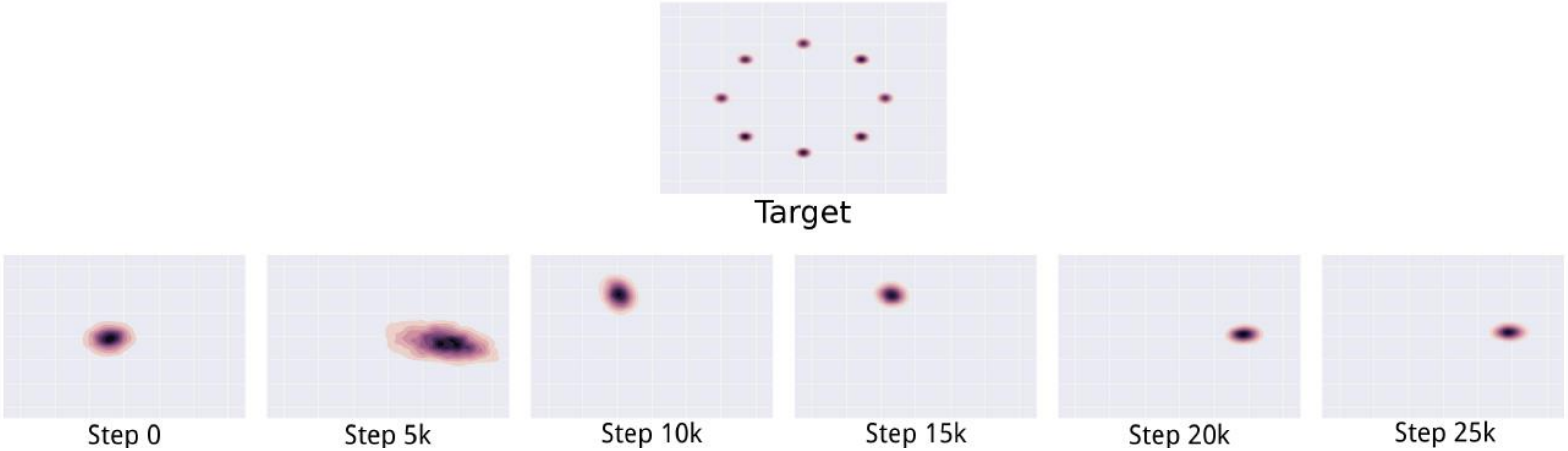
- Exploiting convexity in function space, GAN training is theoretically guaranteed to converge if we can modify the density functions directly, but:
  - Instead, we modify *G* (sample generation function) and *D* (density ratio), not densities
  - We represent *G* and *D* as highly non-convex parametric 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

(Made by Goodfellow
edited by Azizpour)

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



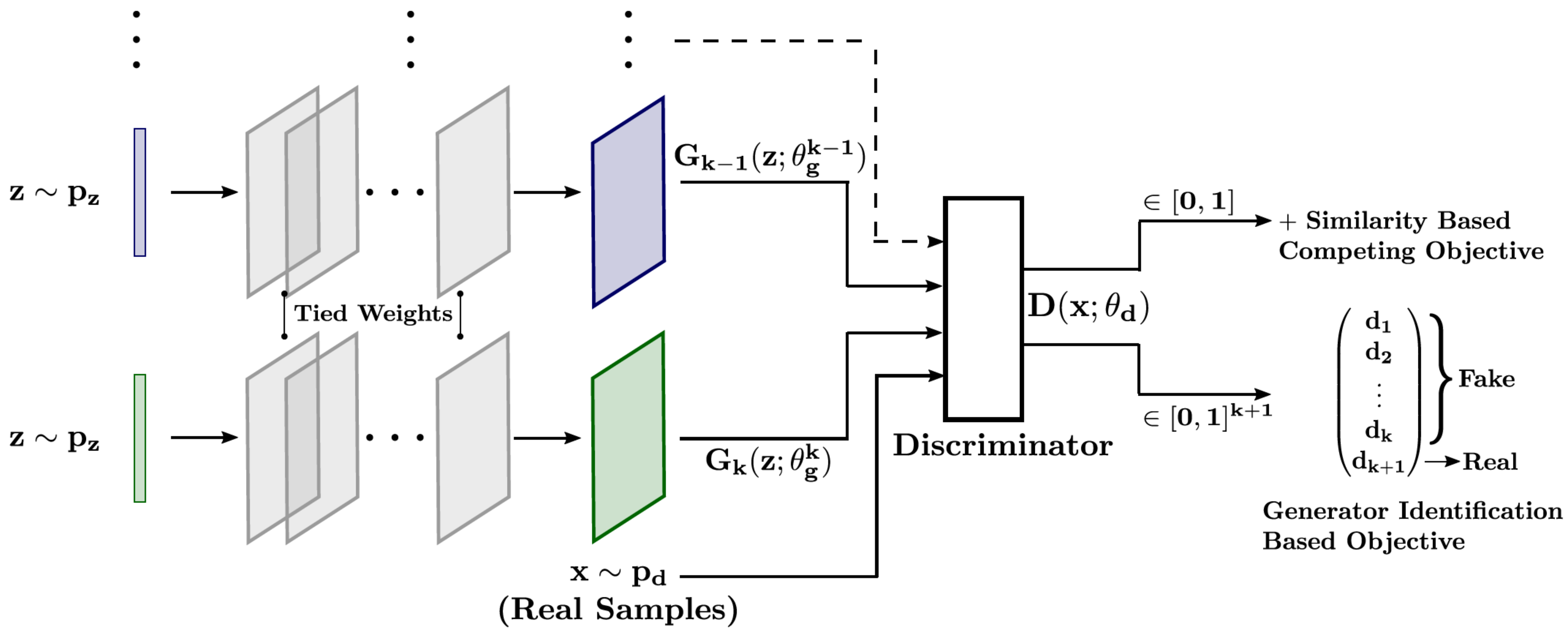
(Metz et al 2016)

(Made by Goodfellow, edited by Azizpour)

Mode Collapse

- GANs often seem to collapse to far fewer modes than the model can represent

Mode collapse causes low output diversity



- multiple parallel generators
- share parameters up to layer l
- applies a diversity loss on different generators
- Alternatively, have D predict which generator the fake sample came from!

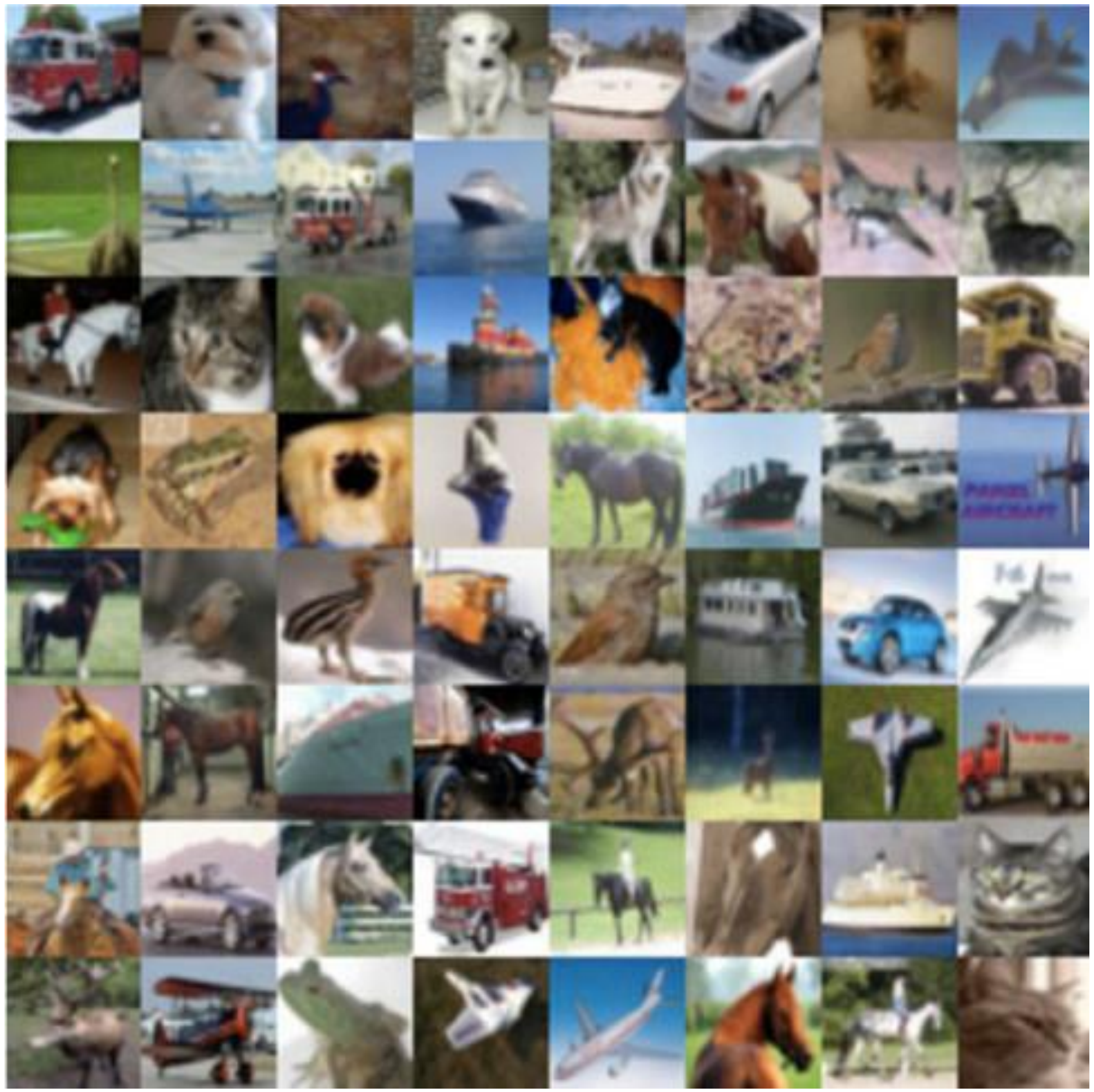
Ghosh et al. Multi-Agent Diverse Generative Adversarial Networks (2017)

(Made by Goodfellow
edited by Azizpour)

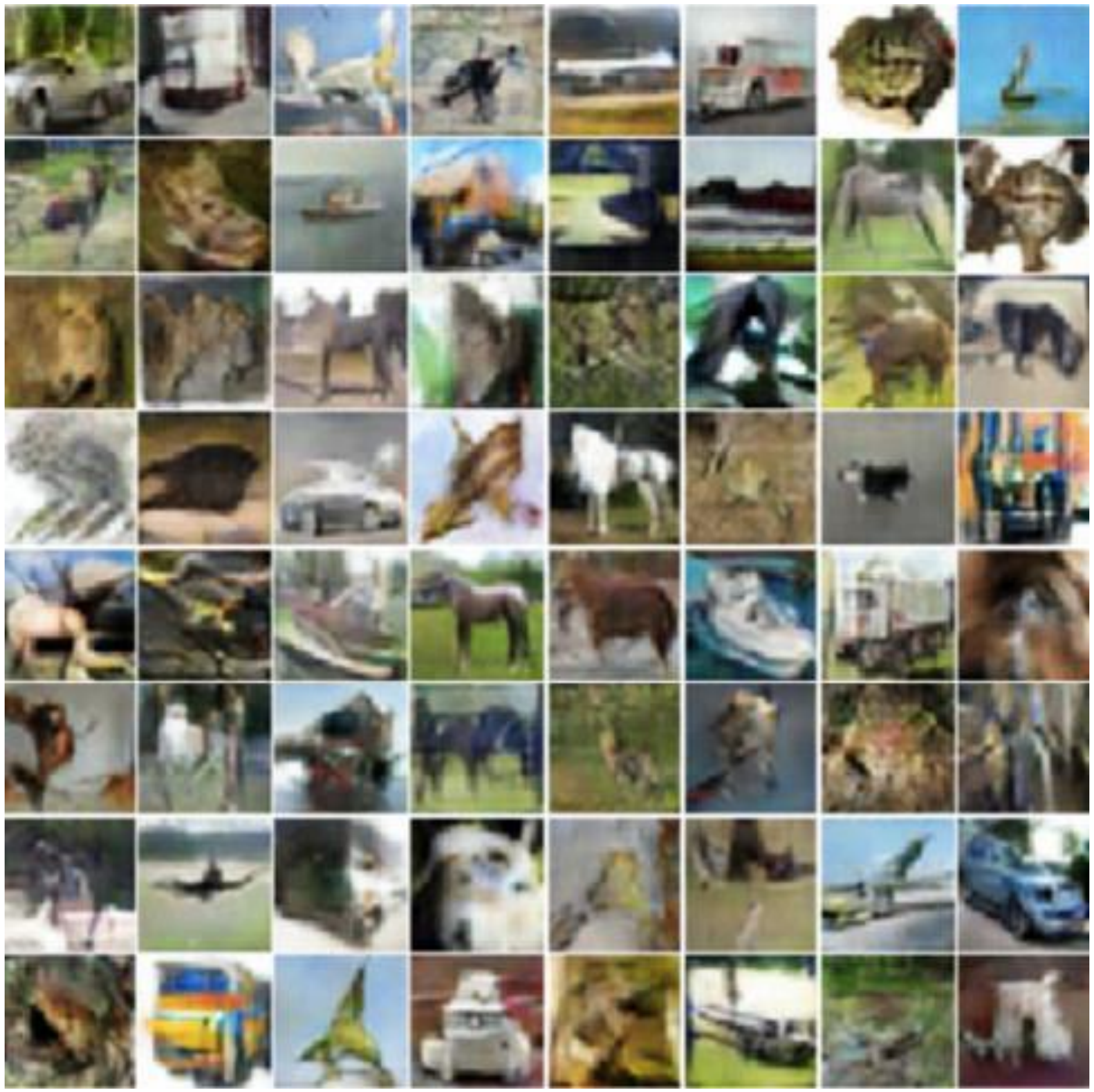
Minibatch Features

- Discriminator classifies a minibatch of either fake or true examples as opposed to an isolated one
- Add minibatch features that classify each example by comparing it to other members of the minibatch (Salimans et al 2016)
- Nearest-neighbor style features detect if a minibatch contains samples that are too similar to each other (coming from a single mode)

Minibatch GAN on CIFAR



Training Data

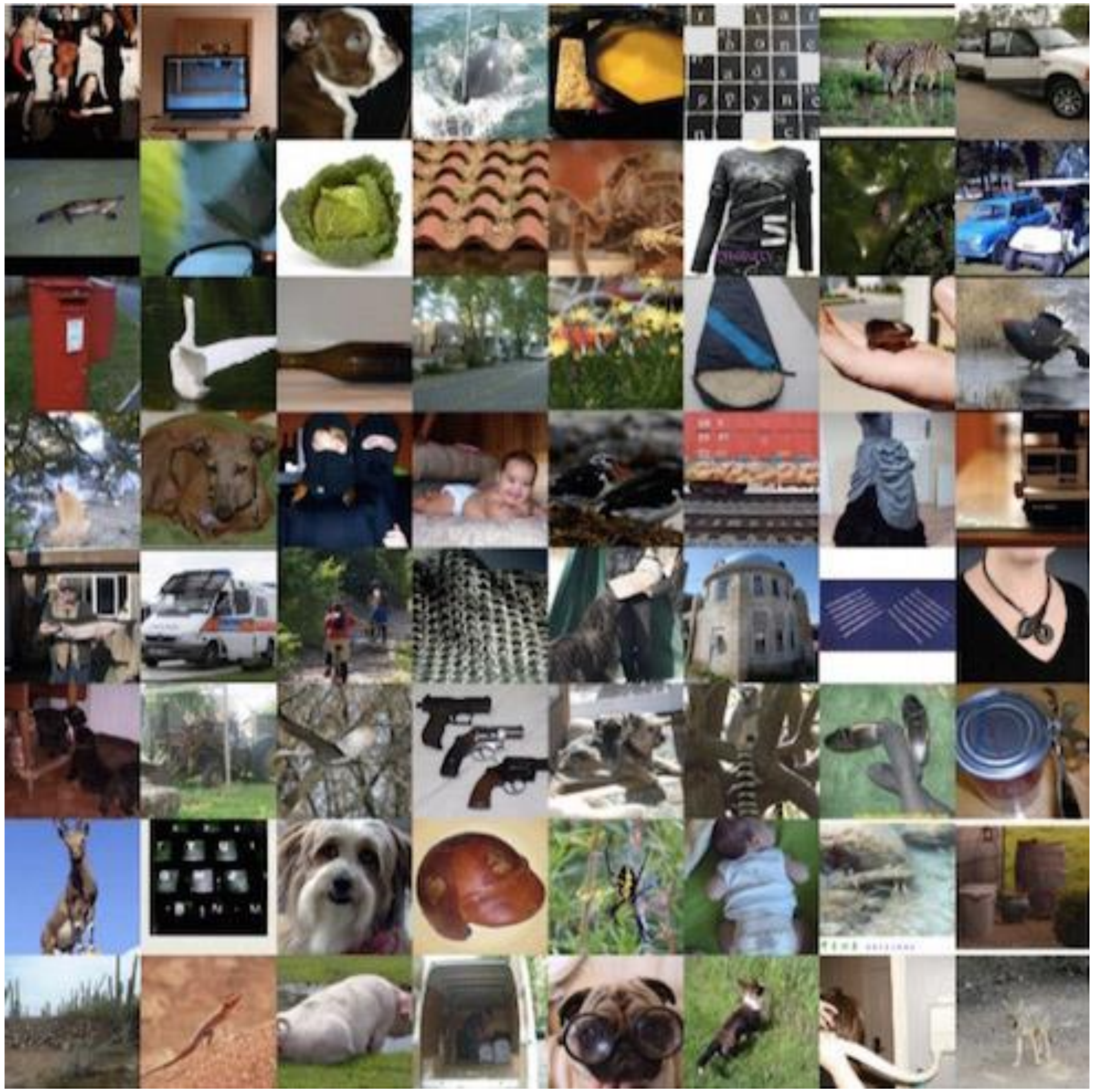


Samples

(Salimans et al 2016)

(Made by Goodfellow
edited by Azizpour)

Minibatch GAN on ImageNet



(Salimans et al 2016)

(Made by Goodfellow,
edited by Azizpour)

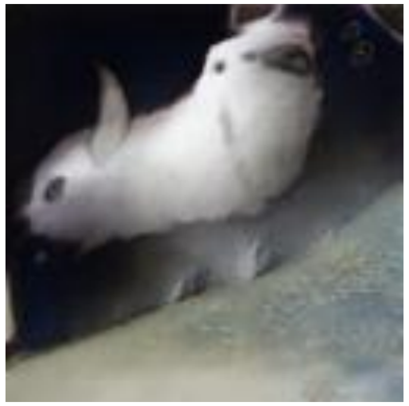
Cherry-Picked Results

(Made by Goodfellow
edited by Azizpour)

Problems with Counting

(Made by Goodfellow
edited by Azizpour)

Problems with Perspective




(Made by Goodfellow
edited by Azizpour)

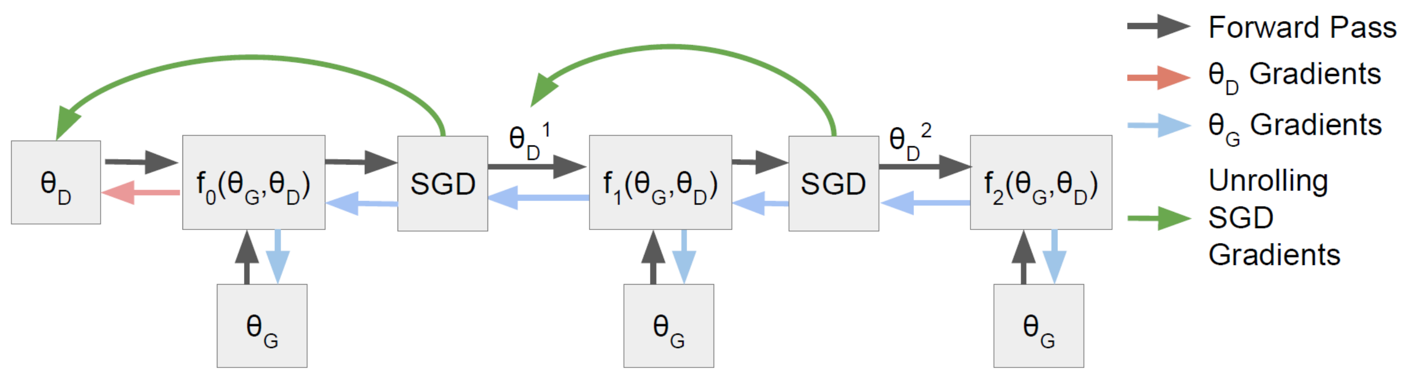
Problems with Global Structure

(Made by Goodfellow
edited by Azizpour)

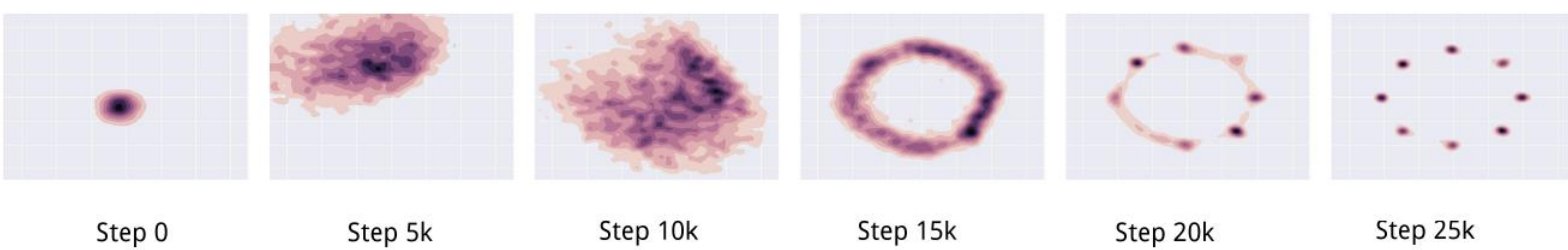
This one is real

(Made by Goodfellow
edited by Azizpour)

Unrolled GANs



- Backprop through *k* updates of the discriminator to prevent mode collapse:



Metz et al UNROLLED GENERATIVE ADVERSARIAL NETWORKS (2016)

(Made by Goodfellow
edited by Azizpour)

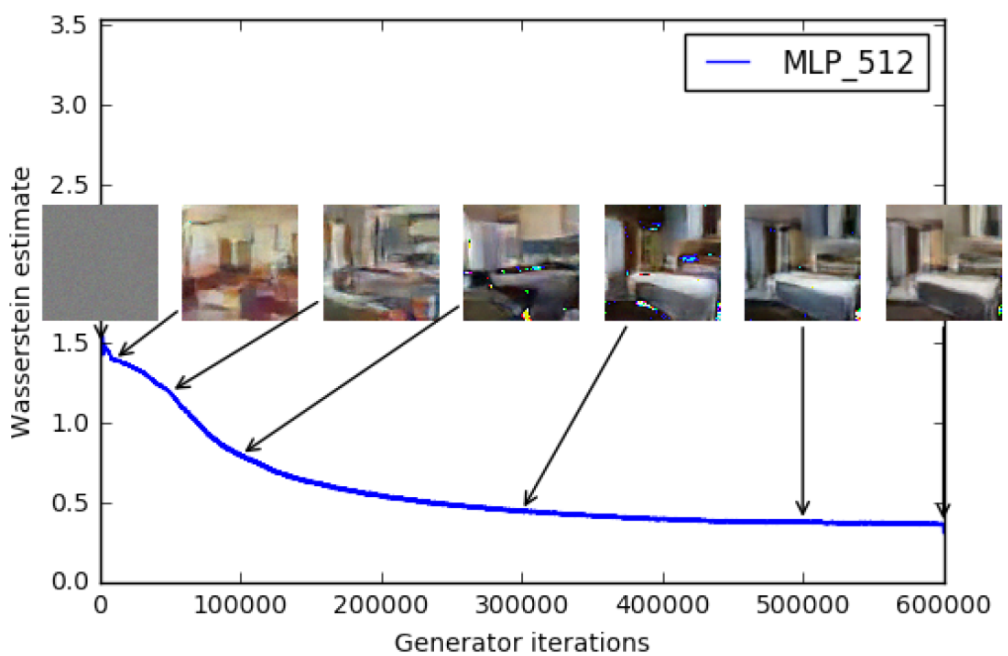
Wasserstein GAN

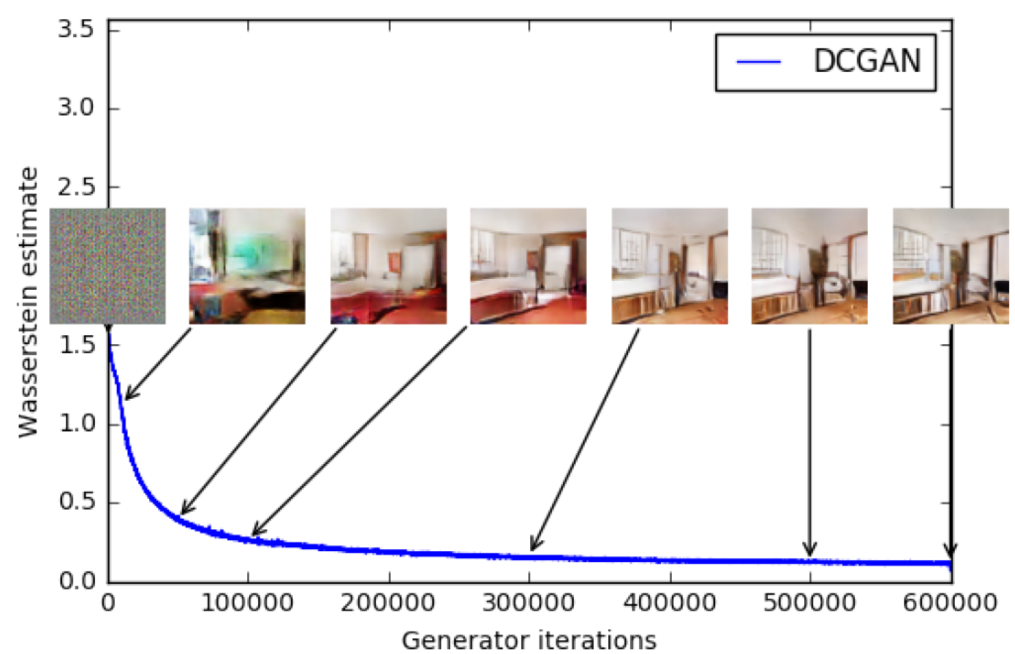
Attacks two problems:
- Stabilizing the training
- Convergence criteria

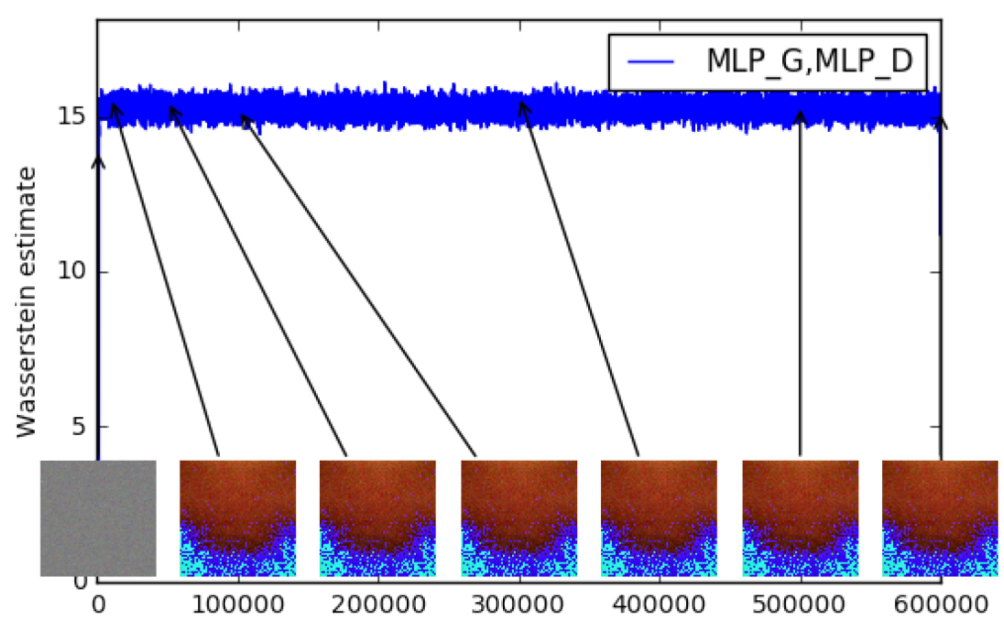
- No log in the loss. The output of DD is no longer a probability, hence we do not apply sigmoid at the output of DD
- Clip the weight of DD
- Train DD more than GG
- Use RMSProp instead of ADAM
- Lower learning rate, the paper uses α=0.00005

(Made by Goodfellow
edited by Azizpour)

Wasserstein GAN







(Made by Goodfellow
edited by Azizpour)

Evaluation

- There is not any single compelling way to evaluate a generative model
  - Models with good likelihood can produce bad samples
  - Models with good samples can have bad likelihood
  - There is not a good way to quantify how good samples are
- For GANs, it is also hard to even estimate the likelihood
- See “A note on the evaluation of generative models,” Theis et al 2015, for a good overview

(Made by Goodfellow
edited by Azizpour)

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

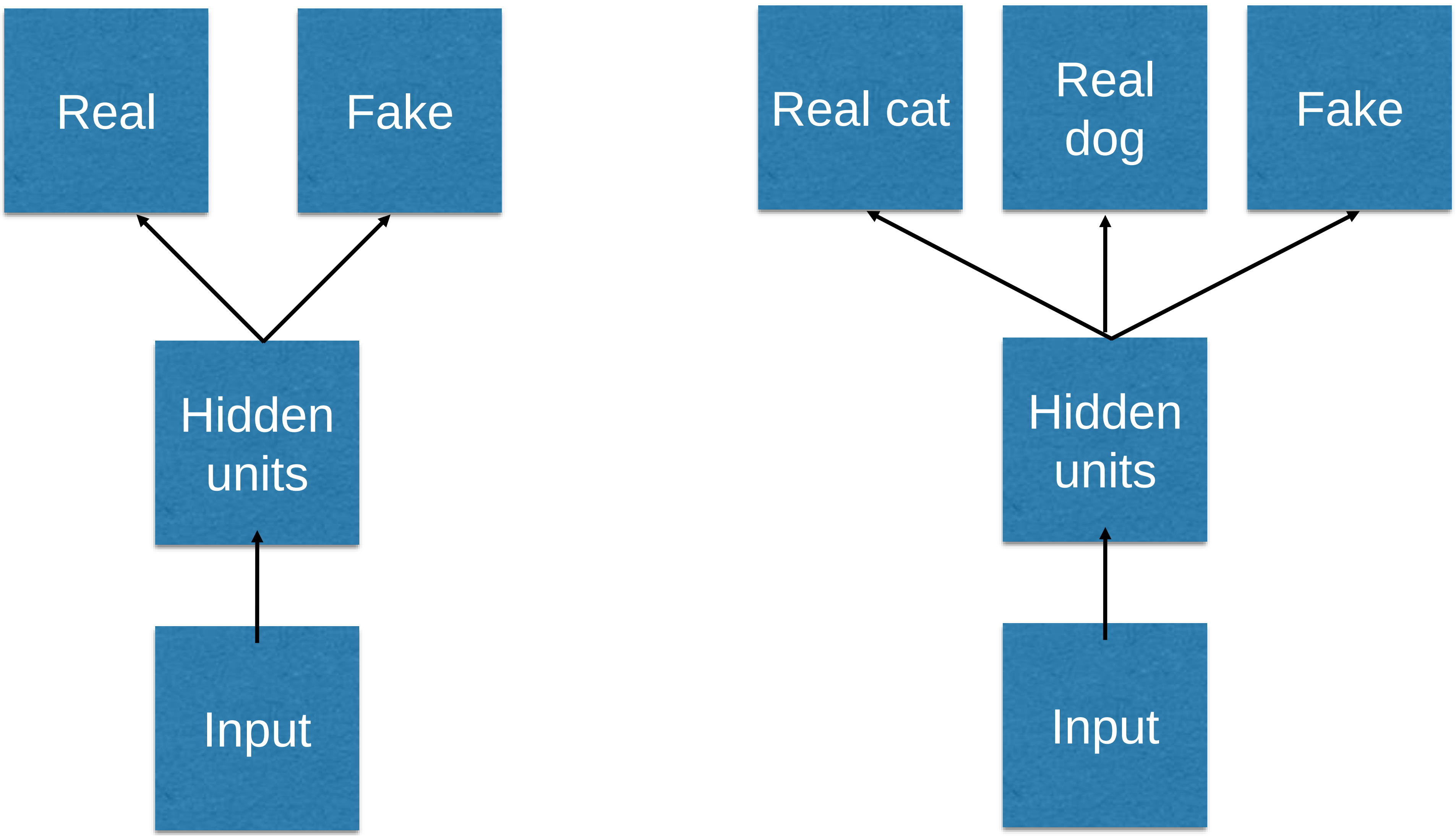
(Made by Goodfellow
edited by Azizpour)

Auxiliary Information

- One trick to stabilize the training is to include auxiliary information either in the input space (current frame for future frame prediction) or adding it to the output space (different classes of true examples)

(Made by Goodfellow
edited by Azizpour)

Supervised Discriminator



(Odena 2016, Salimans et al 2016)

(Made by Goodfellow
edited by Azizpour)

Learning interpretable latent codes / controlling the generation process

(a) Azimuth (pose) (b) Elevation

(c) Lighting (d) Wide or Narrow

InfoGAN (Chen et al 2016)

(Made by Goodfellow, edited by Azizpour)

The coolest GAN ever!

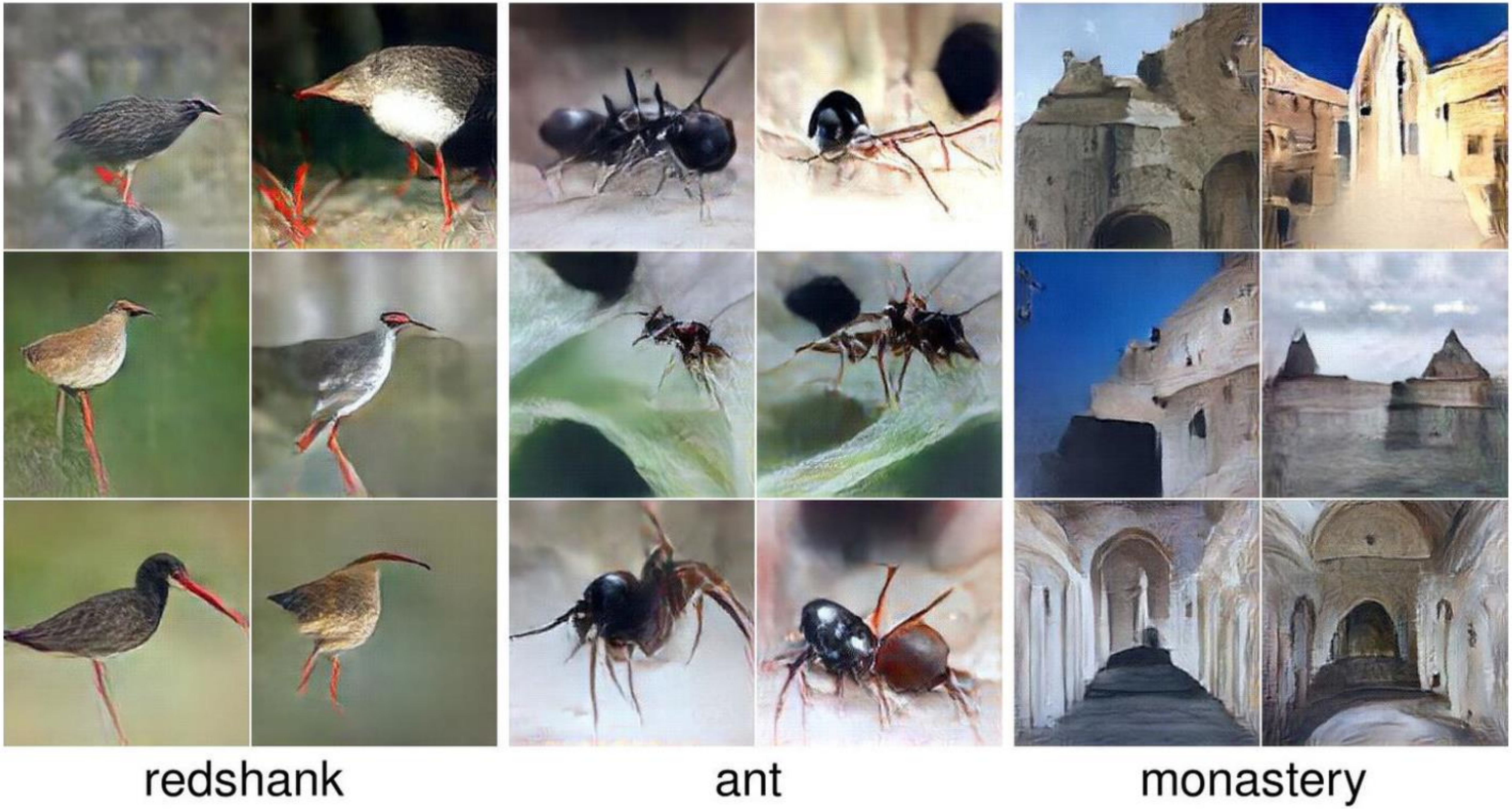
(Made by Goodfellow
edited by Azizpour)

Plug and Play Generative Models

- New state of the art generative model (Nguyen et al 2016)
- Generates 227x227 realistic images from all ImageNet classes
- Combines adversarial training, perceptual loss, autoencoders, and Langevin sampling

Nguyan et al. Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space (2016)

PPGN Samples



(Nguyen et al 2016)

(Made by Goodfellow
edited by Azizpour)

PPGN Samples

volcano

(Nguyen et al 2016)

(Made by Goodfellow
edited by Azizpour)

PPGN for caption to image

oranges on a table next to a liquor bottle

(Nguyen et al 2016)

(Made by Goodfellow
edited by Azizpour)

iGAN

$$z^* = \arg\min_{z \in \tilde{\mathbb{Z}}} \mathcal{L}(G(z),\, x^R)$$

Zhu et al **iGAN:** Generative Visual Manipulation on the Natural Image Manifold (2016) collaboration between Adobe and Berkeley

(Made by Goodfellow
edited by Azizpour)

Summary

- GANs are generative models that use supervised learning to approximate an intractable cost function
- GANs can simulate many cost functions, including the one used for maximum likelihood
- GANs are, at the moment, unstable to train and need many tricks to converge. Reaching Nash equilibrium is an important open research question.

(Made by Goodfellow
edited by Azizpour)

Questions

- Technical Report
- Ian Goodfellow’s technical report on GANs is a very easy and informative read:
- https://arxiv.org/pdf/1701.00160.pdf