Introduction to Deep Generative Modeling

Lecture #16

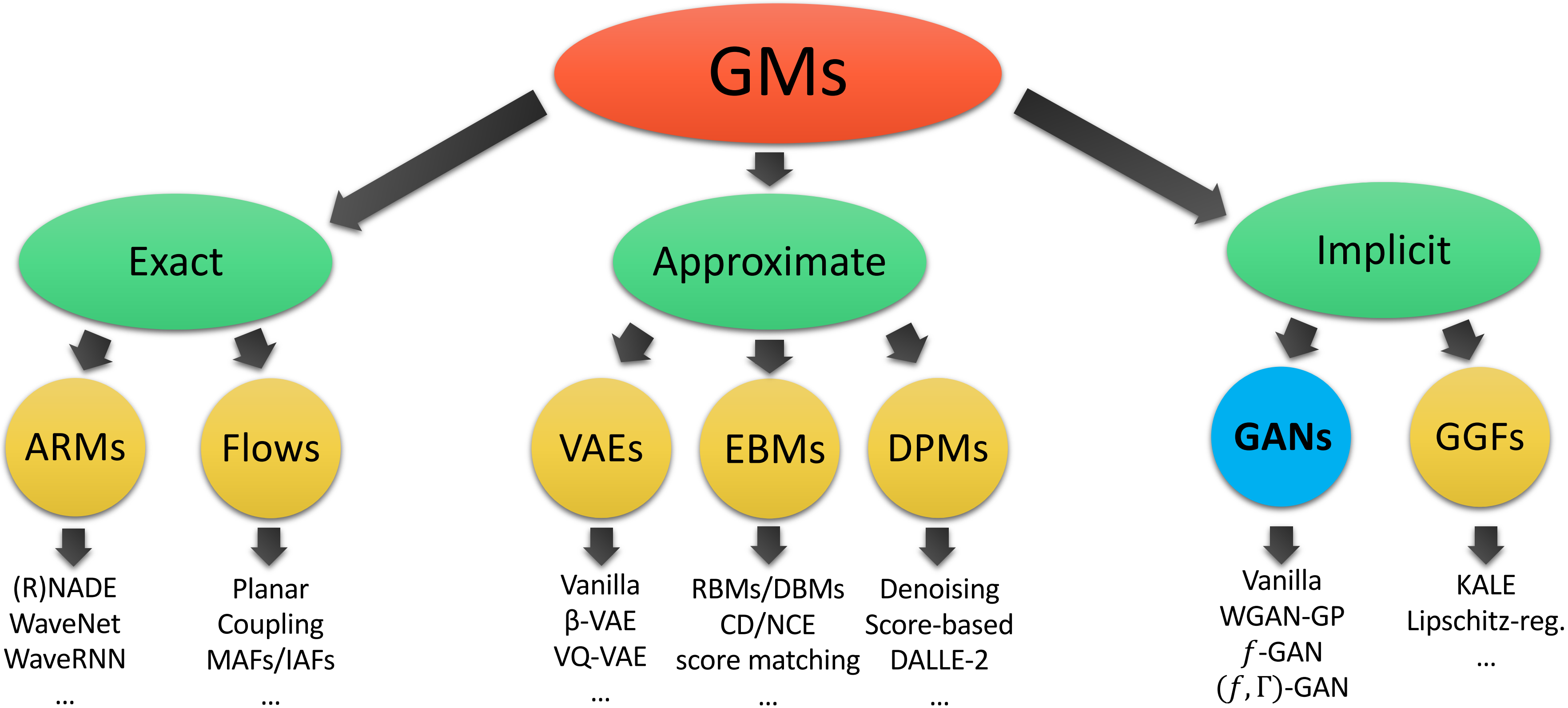
HY-673 – Computer Science Dep., University of Crete

Professors: Yannis Pantazis, Yannis Stylianou

Teaching Assistant: Michail Raptakis

Taxonomy of GMs

Lecture #16



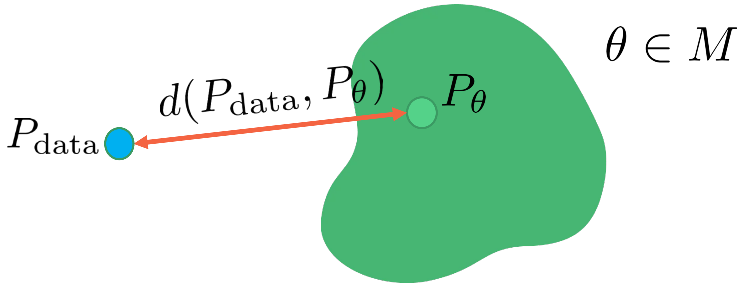
Generative Adversarial Networks

Lecture #16

Recap:

$x_i \sim P_{\text{data}}$

$i = 1, 2, \ldots, n$



Model Family

- Generative Model families
  - Autoregressive Models: $p_\theta(x) = \prod_{i=1}^{n} p_\theta(x_i|x_{<i})$.
  - Variational Autoencoders: $p_\theta(x) = \int p_\theta(x, z)dz$.
  - Normalizng Flow Models: $p_X(x;\theta) = p_Z(f_\theta^{-1}(x))\left|\det\left(\frac{\partial f_\theta^{-1}(x)}{\partial x}\right)\right|$.
  - Diffusion Probabilistic Models: $p_\theta(x) = p_\theta(x|x_1)\prod_{t=T}^{2} p_\theta(x_{t-1}|x_t)p(x_T)$.

Why Maximum Likelihood?

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log p_{\theta}(x_i), \quad x_1, x_2, \cdots, x_n \sim p_{\text{data}}(x).$$

- **Optimal statistical efficiency**
  - Assume sufficient model capacity, such that there exists a unique $\theta^* \in \mathcal{M}$ that satisfies $p_{\theta^*} = p_{\text{data}}$.
  - The convergence of $\hat{\theta}$ to θ^* when $n \to \infty$ is the "fastest" among all statistical methods when using maximum likelihood training.

- **Higher likelihood $\equiv$ better lossless compression**.

- Is the likelihood a good indicator of the quality of samples generated by the model?

Towards Likelihood-Free Learning

- **Case 1:** Optimal generative model will give best **sample quality** and highest test **log-likelihood**.
- For imperfect models, achieving high log-likelihoods might not always imply good sample quality, and vice-versa (Theis et al., 2016)
- **Case 2:** Great test log-likelihoods, poor samples. E.g., For a discrete noise mixture model $p_\theta(x) = 0.01 p_{\text{data}}(x) + 0.99 p_{\text{noise}}(x)$
  - 99% of the samples are just noise
  - Taking logs, we get a lower bound

$$\log p_\theta(x) = \log \left[0.01 p_{\text{data}}(x) + 0.99 p_{\text{noise}}(x) \right] \geq \log 0.01 p_{\text{data}}(x) = \log p_{\text{data}}(x) - \log 100$$

Towards Likelihood-Free Learning

Lecture #16

- For expected likelihoods, we know that
  - Lower bound
  - Upper bound (via non-negativity of KL)

$$\mathbb{E}_{p_{\text{data}}}\big[\log p_\theta(x)\big] \geq \mathbb{E}_{p_{\text{data}}}\big[\log p_{\text{data}}(x)\big] - \log 100$$

$$\mathbb{E}_{p_{\text{data}}}\big[\log p_{\text{data}}(x)\big] \geq \mathbb{E}_{p_{\text{data}}}\big[\log p_\theta(x)\big]$$

- As we increse the dimension of x, absolute value of $\log p_{\text{data}}(x)$ increases proportionally but $\log 100$ remains constant.
  Hence, $\mathbb{E}_{p_{\text{data}}}\big[\log p_\theta(x)\big] \approx \mathbb{E}_{p_{\text{data}}}\big[\log p_{\text{data}}(x)\big]$ in very high dimensions!

Towards Likelihood-Free Learning

- **Case 3:** Great samples, poor test log-likelihoods. E.g., Memorizing training set
  - Samples look exactly like the training set (cannot do better!)
  - Test set will have zero probability assigned (cannot do worse!)
- The above cases suggest that it might be useful to disentangle likelihoods and samples
- **Likelihood-free learning** consider objectives that do not depend directly on a likelihood function

Comparing Distributions via Samples

$S_1 = \{x \sim P\}$

vs.

$S_2 = \{x \sim Q\}$

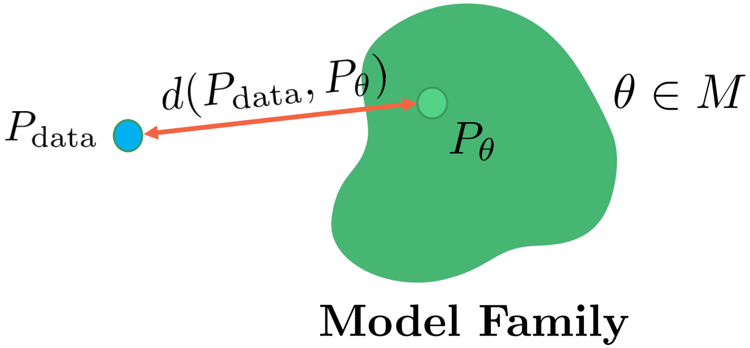
- Given a finite set of samples from two distributions $S_1 = \{x \sim P\}$ and $S_2 = \{x \sim Q\}$, how can we tell if these samples are from the same distribution? (i.e., P = Q?)

Two-Sample Tests

- Given $S_1 = \{x \sim P\}$ and $S_2 = \{x \sim Q\}$, a **two-sample test** considers the following hypotheses
  - Null hypothesis $H_0 : P = Q$
  - Alternative hypothesis $H_1 : P \neq Q$
- Test statistic T compares S_1 and S_2 e.g., difference in means, variances of the two sets of samples.
- If T is larger than a threshold α, then reject H_0 otherwise we say H_0 is consistent with observation.
- **Key observation:** Test statistic is **likelihood-free** since it does not involve the densities P or Q (only samples)

Generative Modeling and Two-Sample Tests

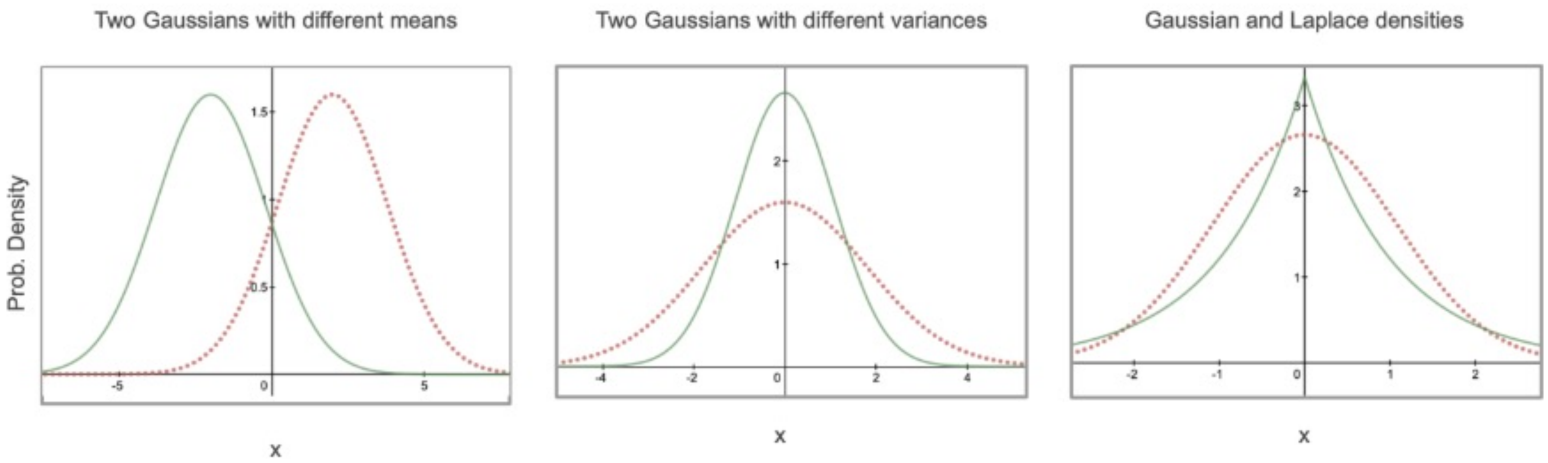
$$x_i \sim P_{\text{data}}$$
$$i = 1, 2, \dots, n$$



- A priori we assume direct access to $S_1 = \mathcal{D} = \{x \sim p_{\text{data}}\}$
- In addition, we have a model distribution p_θ
- Assume that the model distribution permits efficient sampling (e.g., directed models). Let $S_2 = \{x \sim p_\theta\}$
- **Altrernative notion of distance between distributions:** Train the generative model to minimize a two-sample test objective between S_1 and S_2

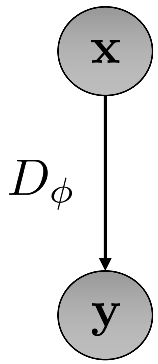
Two-Sample Test via a Discriminator

- Finding a two-sample test objective in high dimensions is hard



- In the generative model setup, we know that S_1 and S_2 come from different distributions p_{data} and p_θ respectively
- **Key idea: Learn** a statistic that **maximizes** a suitable notion of distance between the two sets of samples S_1 and S_2

Two-Sample Test via a Discriminator

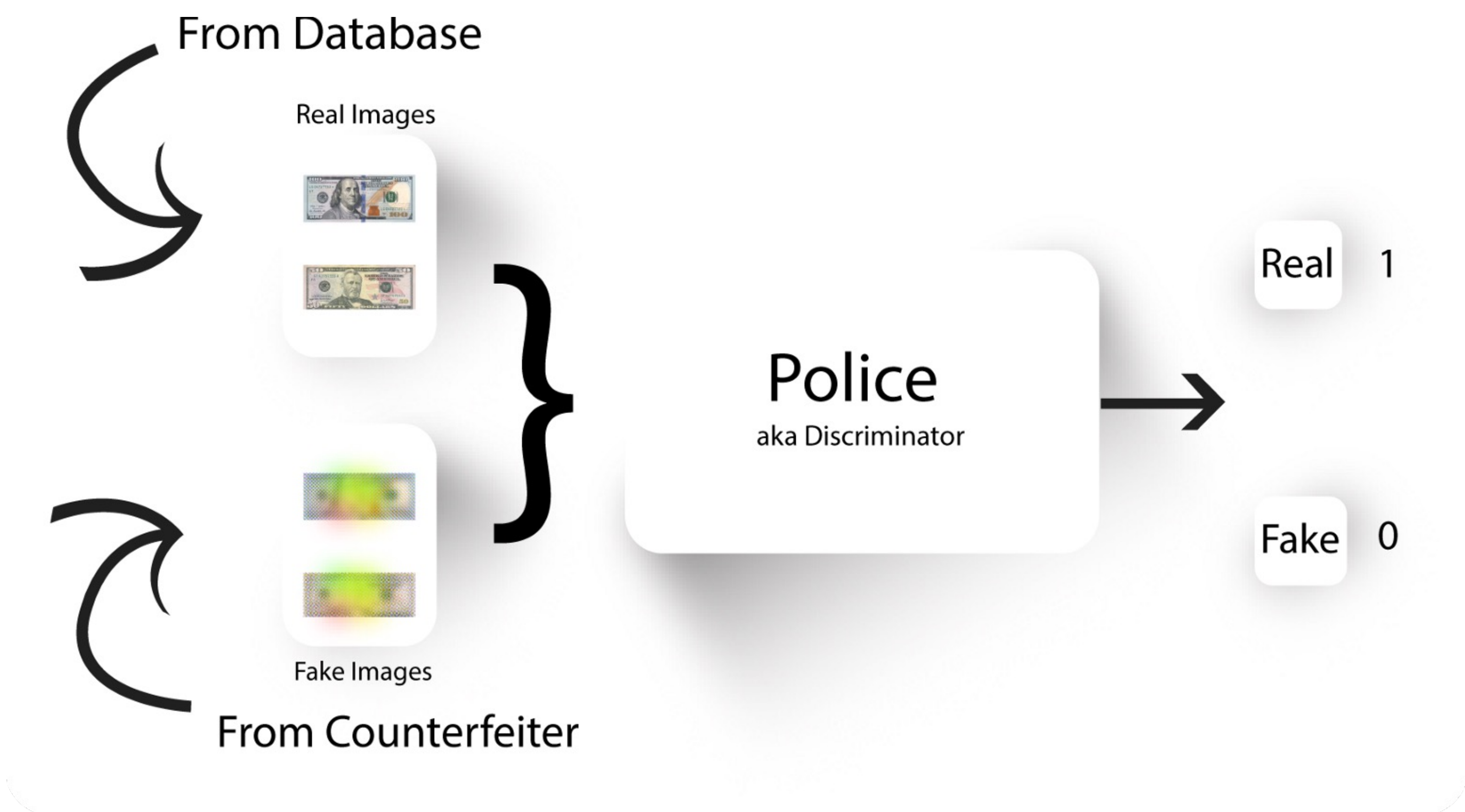


$$D_\phi(x) = 1 \text{ when } x \sim p_{\text{data}}$$

$$D_\phi(x) = 0 \text{ when } x \sim p_\theta$$

- **Two-Sample Test via a Discriminator**
  - Any function (e.g., neural network) which tries to distinguish "real" samples from the dataset and "fake" samples generated from the model
  - Maximizes the two-sample test objective (in support of the alternative hypothesis $p_{\text{data}} \neq p_\theta$)

Two-Sample Test via a Discriminator



Two-Sample Test via a Discriminator

- **Training objective for discriminator:**

$$\max_D V(D) = \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{x \sim p_G}[\log(1 - D(x))].$$

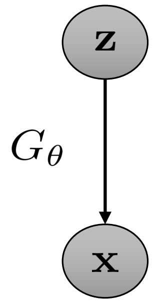
- For a fixed generator G, the discriminator is performing binary classification with the cross entropy objective:
  - Assign probability 1 to true data points $x \sim p_{\text{data}}$
  - Assign probability 0 to fake samples $x \sim p_G$

- Optimal Discriminator: $D^*_G(x) = \dfrac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}.$

Generative Adversarial Networks (GAN

Lecture #16

- A two player minimax game between a **generator** and a **discriminator**

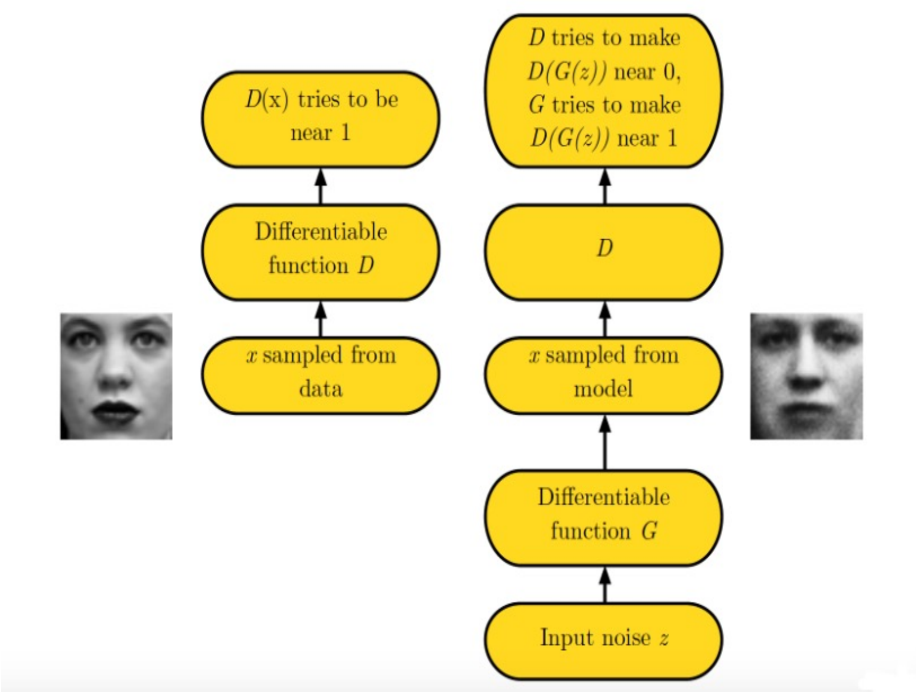


- **Generator**
  - Directed, latent variable model with a deterministic mapping between z and x given by G_θ
  - Minimizes a two-sample test objective (in support of the null hypothesis $p_{\text{data}} = p_\theta$)

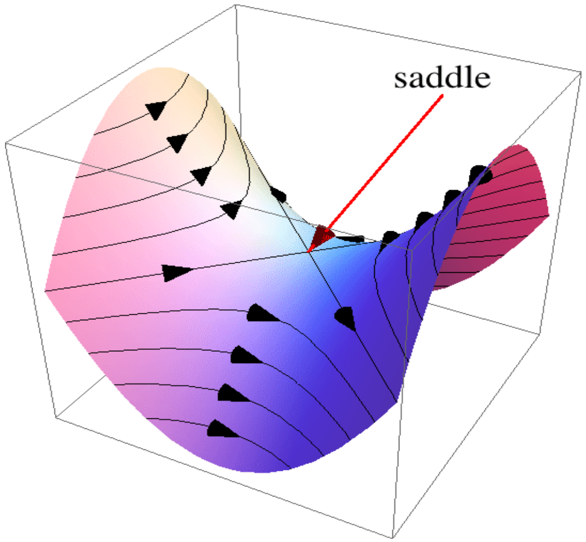
Example of GAN Objective

- Training objective for both generator and discriminator:

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



The joint optimum (G^*, D^*) is a saddle point.



Example of GAN Objective

- For the optimal discriminator $D_G^*(\cdot)$ and fixed generator $G(\cdot)$, we have

$$
\begin{aligned}
V(G, D_G^*(x)) &= \mathbb{E}_{x\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x)+p_G(x)}\right] + \mathbb{E}_{x\sim p_G}\left[\log \frac{p_G(x)}{p_{\text{data}}(x)+p_G(x)}\right] \\
&= \mathbb{E}_{x\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{\frac{1}{2}(p_{\text{data}}(x)+p_G(x))}\right] + \mathbb{E}_{x\sim p_G}\left[\log \frac{p_G(x)}{\frac{1}{2}(p_{\text{data}}(x)+p_G(x))}\right] - \log 4 \\
&= \underbrace{D_{\text{KL}}\left(p_{\text{data}}||\frac{p_{\text{data}}+p_G}{2}\right) + D_{\text{KL}}\left(p_G||\frac{p_{\text{data}}+p_G}{2}\right)}_{2\times \text{ Jensen-Shannon Divergence (JSD)}} - \log 4 \\
&= 2D_{\text{JS}}(p_{\text{data}}, p_G) - \log 4.
\end{aligned}
$$

Jenson-Shannon Divergence

- Also called as the symmetric KL divergence

$$D_{\mathrm{JS}}(p,q) = \frac{1}{2}\left(D_{\mathrm{KL}}\left(p||\frac{1}{2}(p+q)\right) + D_{\mathrm{KL}}\left(q||\frac{1}{2}(p+q)\right)\right).$$

- Properties
  - $D_{\mathrm{JS}}(p,q) \geq 0$
  - $D_{\mathrm{JS}}(p,q) = 0$ iff $p = q$
  - $D_{\mathrm{JS}}(p,q) = D_{\mathrm{JS}}(q,p)$
  - $\sqrt{D_{\mathrm{JS}}(p,q)}$ satisfies triangle inequality $\Rightarrow$ Jenson-Shannon Distance
- Optimal generator for the JSD/Negative Cross Entropy GAN $\quad p_{G^*} = p_{\mathrm{data}}$
- For the optimal discriminator $D^*_{G^*}$ and generator G^*, we have $\quad V(G^*, D^*_{G^*}) = -\log 4.$

The GAN Training Algorithm

- Sample minibatch of m training points $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ from $\mathcal{D}$

- Sample minibatch of m noise vectors $z^{(1)}, z^{(2)}, \ldots, z^{(m)}$ from p_Z

- Update the discriminator parameters ϕ by stochastic gradient **ascent**

$$\nabla_\phi V(G_\theta, D_\phi) = \frac{1}{m} \nabla_\phi \sum_{i=1}^{m} \Big[\log D_\phi(x^{(i)}) + \log(1 - D_\phi(G_\theta(z^{(i)}))) \Big].$$

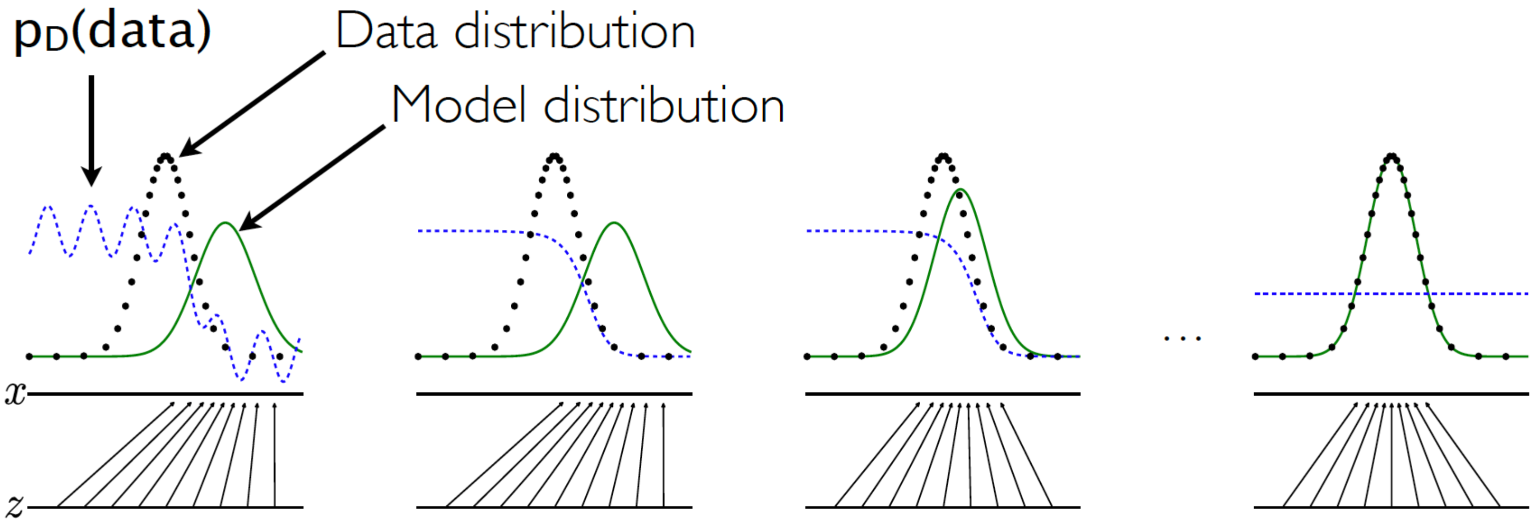
- Update the generator parameters θ by stochastic gradient **descent**

$$\nabla_\theta V(G_\theta, D_\phi) = \frac{1}{m} \nabla_\theta \sum_{i=1}^{m} \log(1 - D_\phi(G_\theta(z^{(i)}))).$$

- Repeat for fixed number of iterations

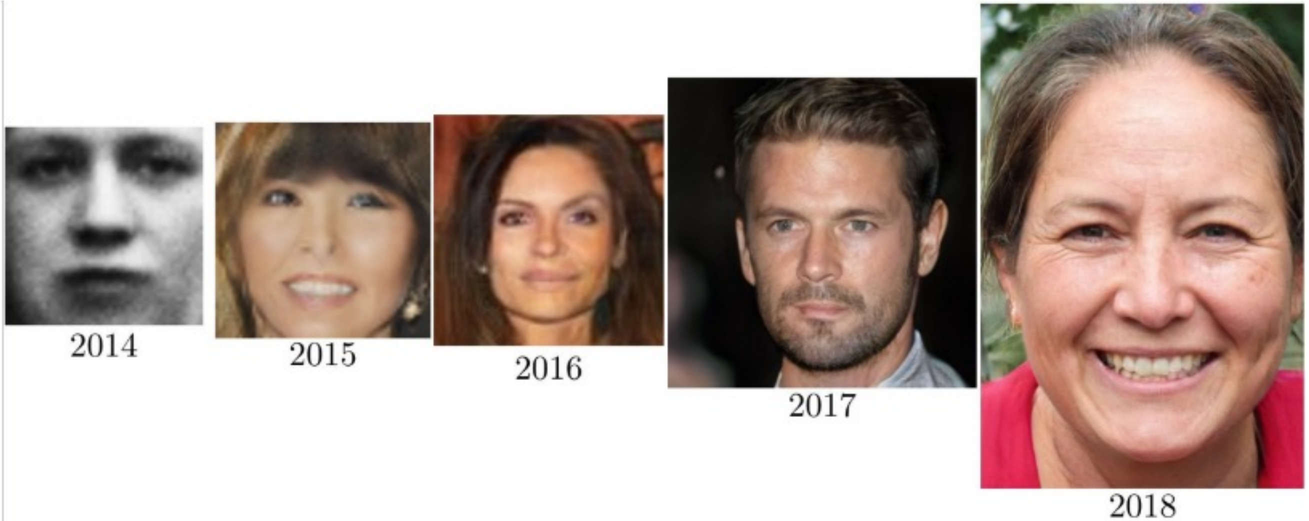
Alternating Optimization in GANs

$$\min_{\theta}\max_{\phi} V(G_\theta, D_\phi) = \mathbb{E}_{x\sim p_{\text{data}}}[\log D_\phi(x)] + \mathbb{E}_{z\sim p_Z}[\log(1 - D_\phi(G_\theta(z)))].$$



Poorly fit model | After updating D | After updating G | Mixed strategy equilibrium

Frontiers in GAN Research

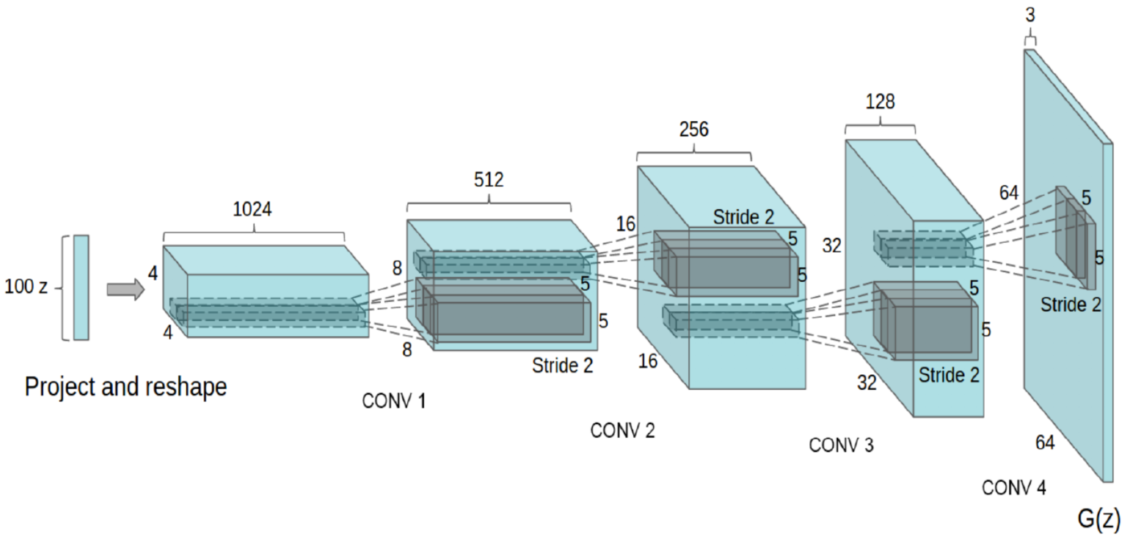


- GANs have been successfully applied to several domains and tasks
- However, working with GANs can be very challenging in practice
  - Unstable optimization
  - Mode collapse
  - Performance evaluation
- Many tricks have been proposed to successfully train GANs

Image Source: Ian Goodfellow. Samples from Goodfellow et al., 2014, Radford et al., 2015, Liu et al., 2016, Karras et al., 2017, Karras et al., 2018

Deep Convolutional GAN (DCGAN)

Generator Architecture



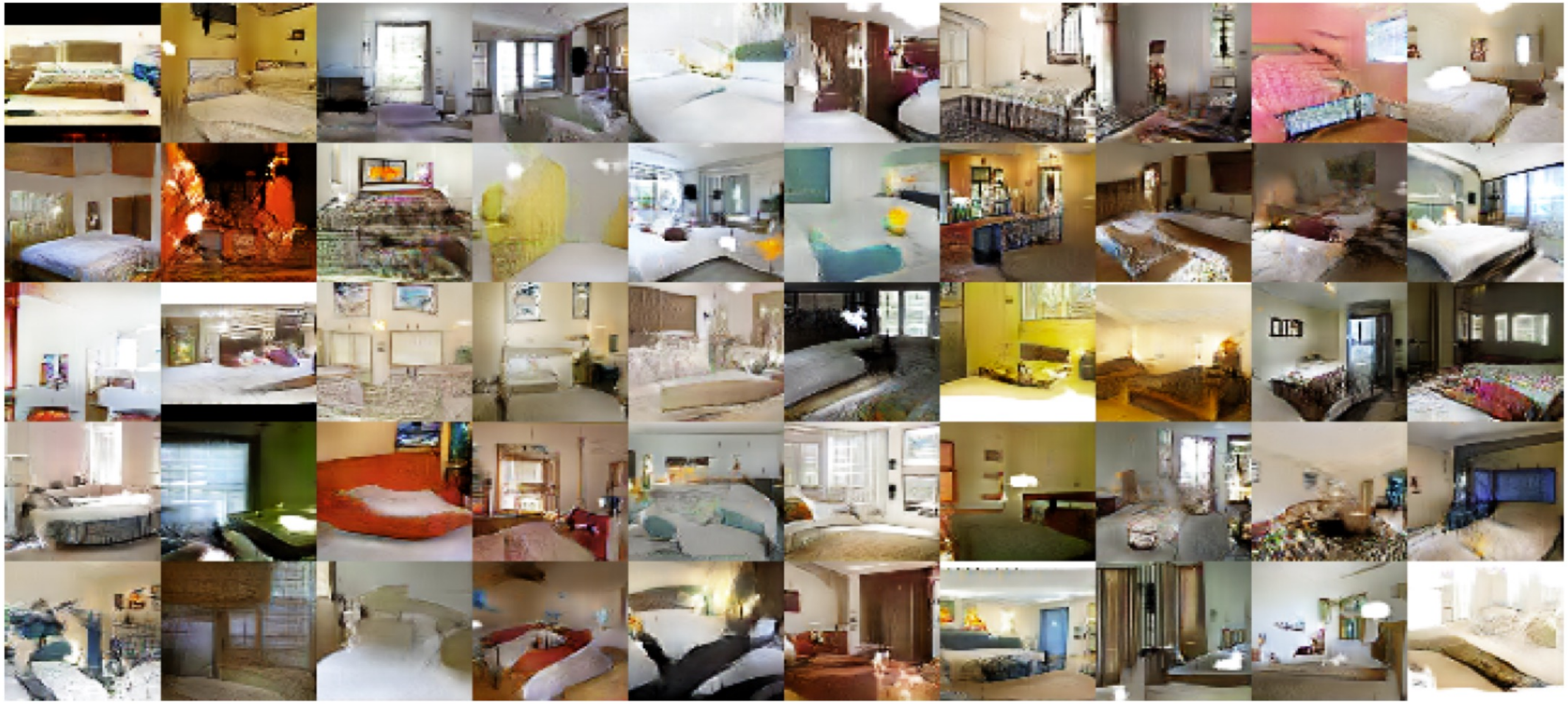
Key ideas:

- Replace FC hidden layers with Convolutions
  - **Generator:** Fractional-Strided convolutions
- Use Batch Normalization after each layer
- **Inside Generator**
  - Use ReLU for hidden layers
  - Use Tanh for the output layer

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

DCGAN Example – LSUN bedrooms

Lecture #16



Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

Conditional GAN

- GAN is too free. How to add some constraints?
- Add conditional variables ***y*** into *G* and *D*

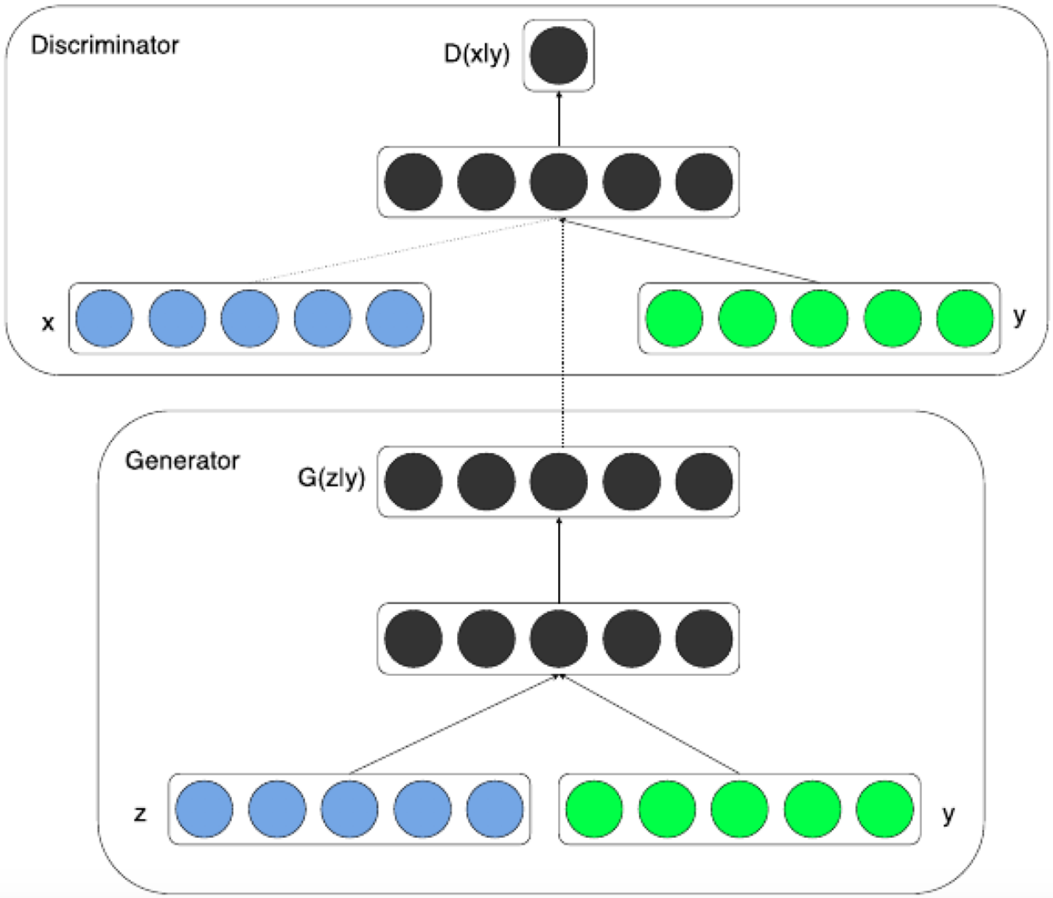
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

↓

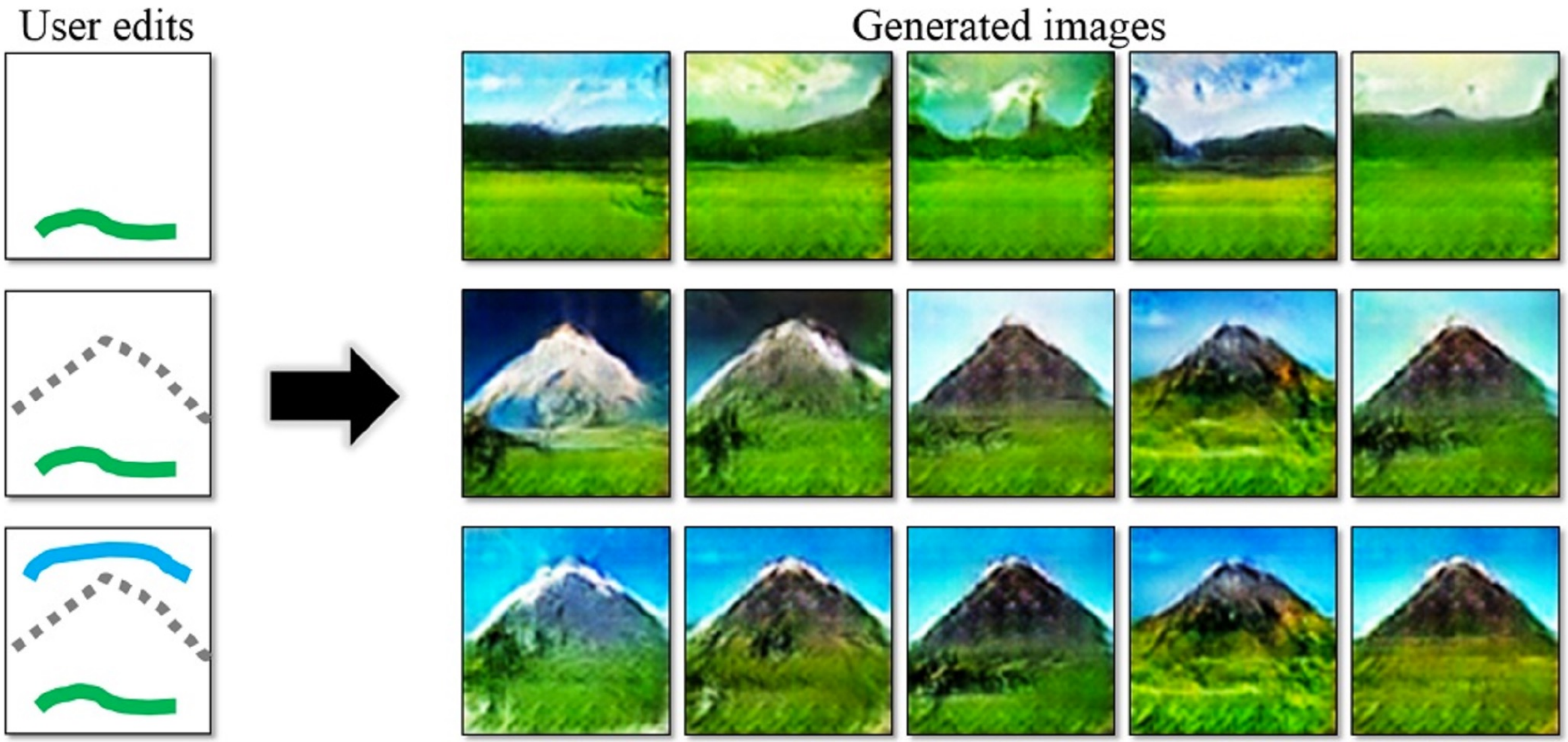
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))].$$

Mirza and Osindero 2016

Conditional GAN

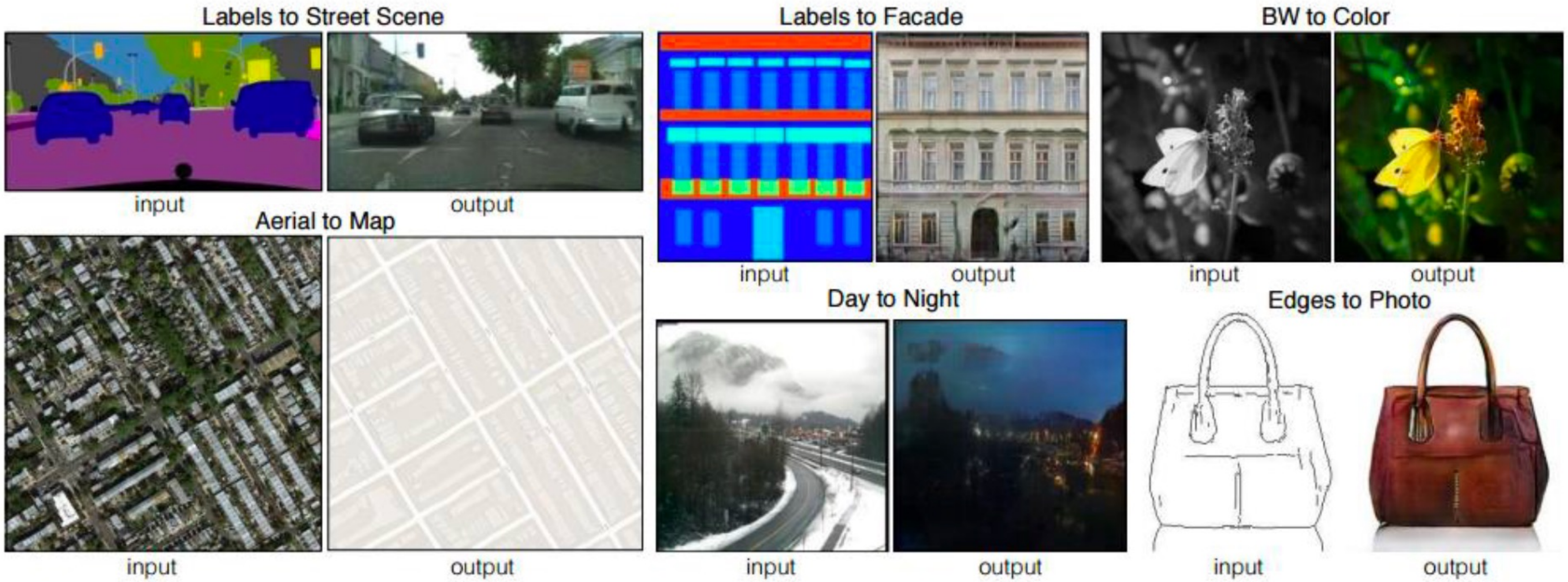


Conditional GAN - Examples



http://people.eecs.berkeley.edu/~junyanz/projects/gvm/

Conditional GAN - Examples

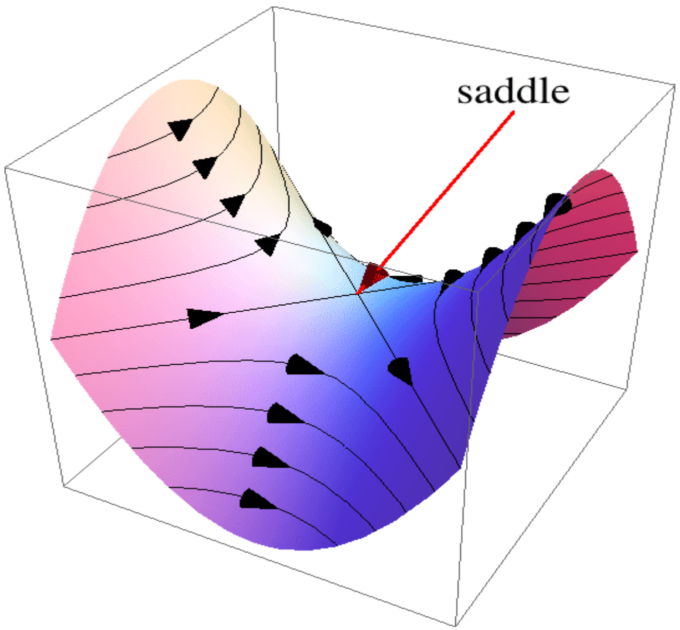


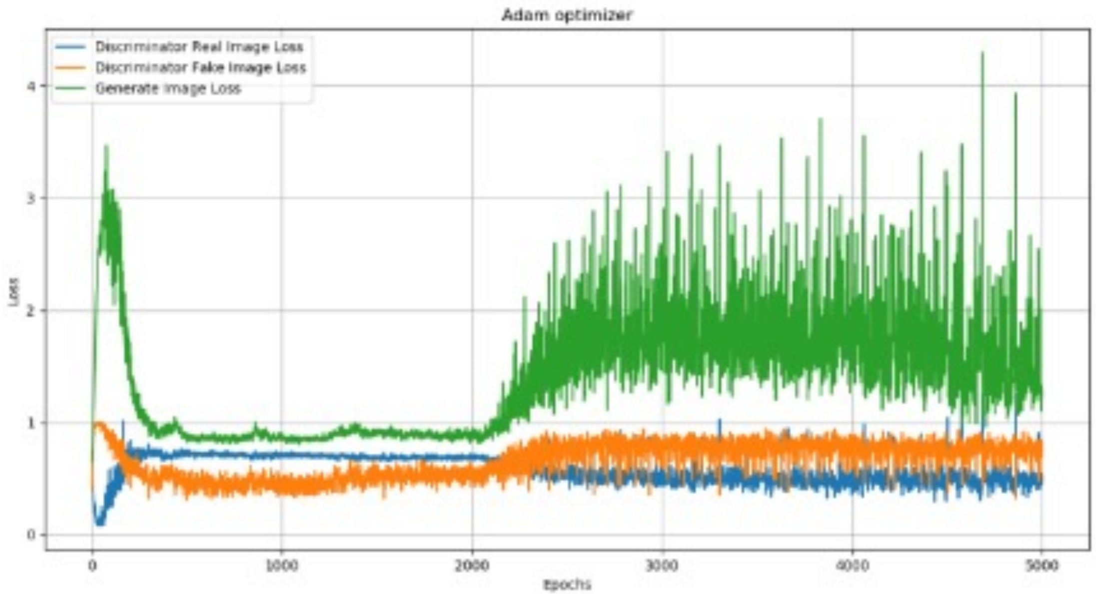
Isola et al. 2016

Optimization Challenges

- **Theorem (informal):** If the generator updates are made in function space and discriminator is optimal at every step, then the generator is guaranteed to converge to the data distribution
- **Unrealistic assumptions!**
- In practice, the generator and discriminator loss keeps oscillating during GAN training
- No robust stopping criteria in practice (unlike MLE)

Optimization Challenges

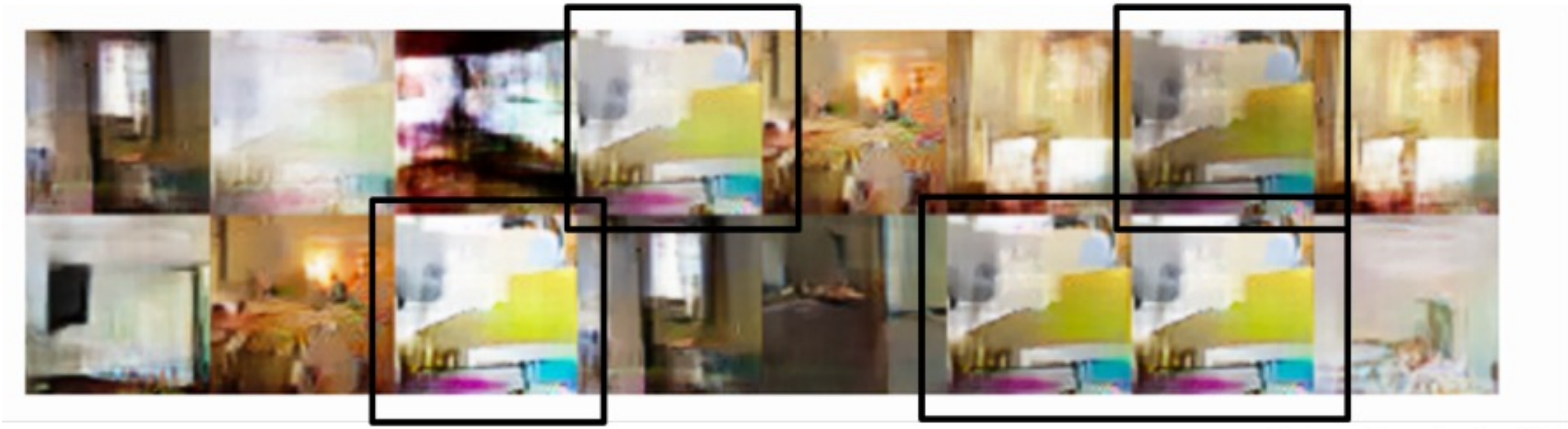




Source: Mirantha Jayathilaka

Mode Collapse

- GANs are notorious for suffering from **mode collapse**
- Intuitively, this refers to the phenomena where the generator of a GAN collapses to one of few samples (dubbed as "modes")



Arjovsky et al., 2017

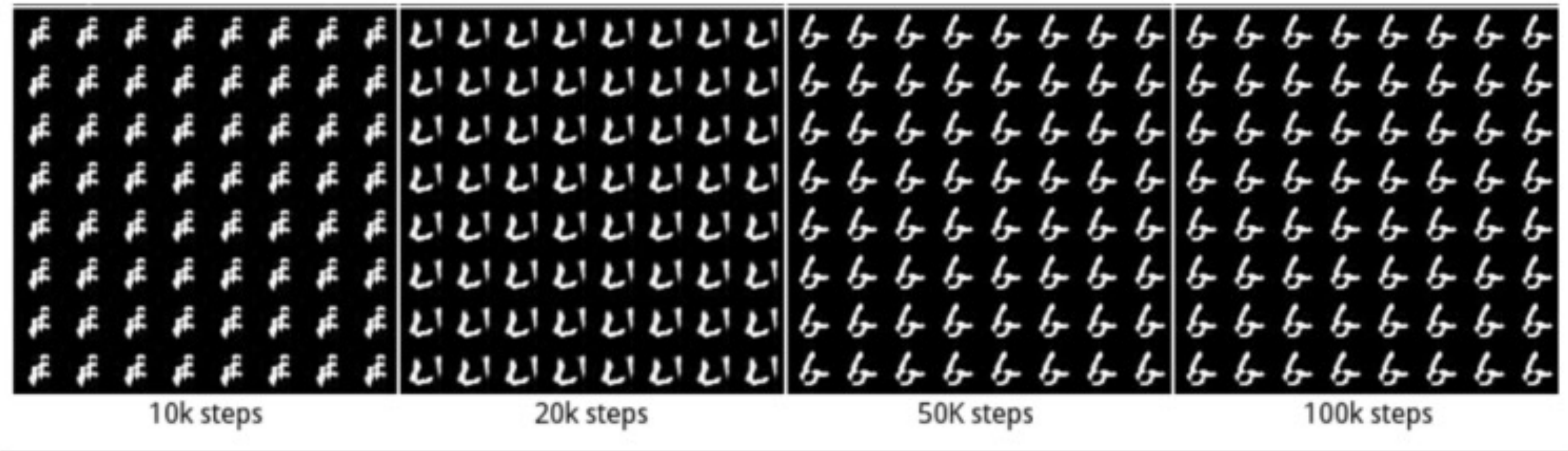
Mode Collapse

- True distribution is a mixture of Gaussians

- The generator distribution keeps oscillating between different modes

Source: Metz et al., 2017

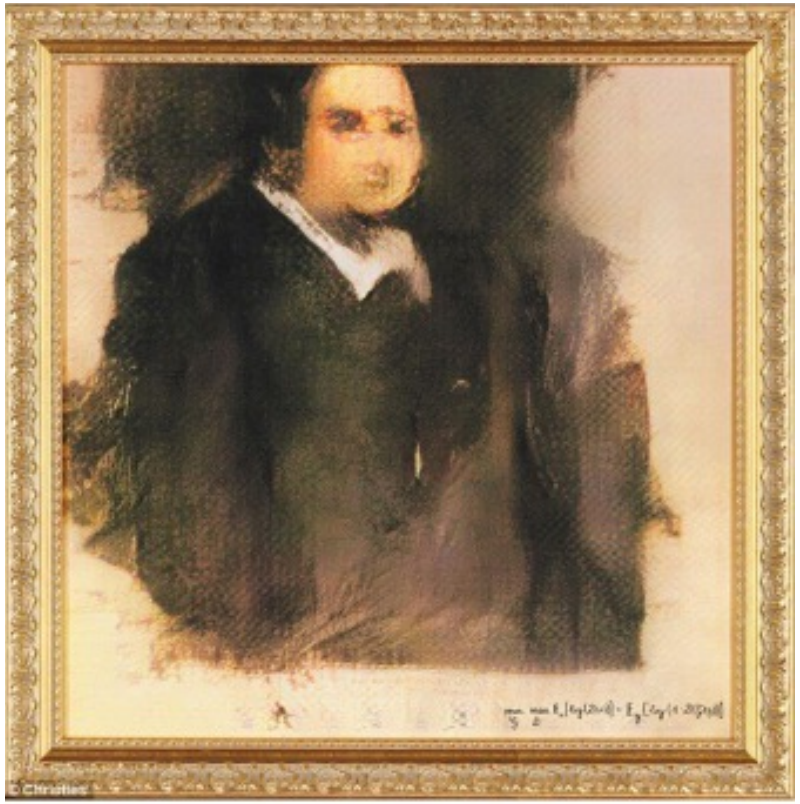
Mode Collapse



Source: Metz et al., 2017

- Fixes to mode collapse are mostly empirically driven: alternative architectures, alternative GAN loss, adding regularization terms, etc.
- How to Train a GAN? Tips and tricks to make GANs work by Soumith Chintala et al. https://github.com/soumith/ganhacks

Beauty Lies in the Eyes of the Discriminator



Source: Robbie Barrat, Obvious

GAN generated art auctioned at Christie's.

Expected Price: $\$7,000 – \$10,000$

True Price: $\$432,500$

Introduction to Deep Generative Modeling

Lecture #16

HY-673 – Computer Science Dep., University of Crete

Professors: Yannis Pantazis, Yannis Stylianou

Teaching Assistant: Michail Raptakis