Carnegie Mellon
School of Computer Science

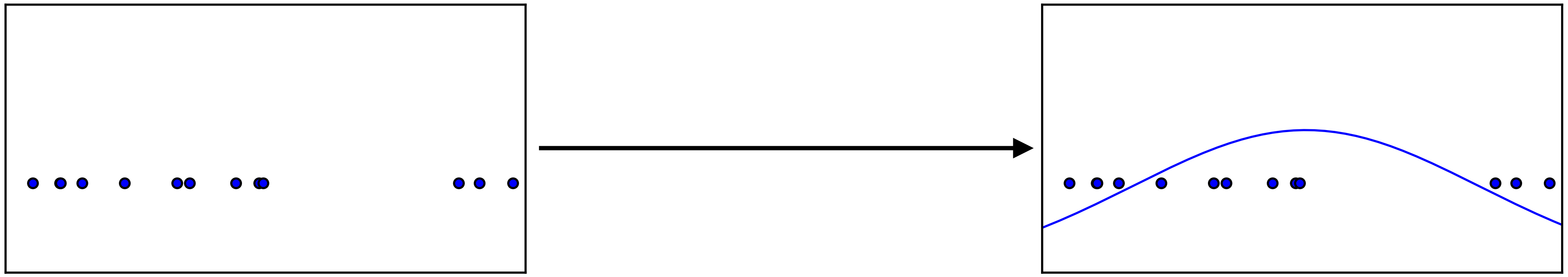
Deep Reinforcement Learning and Control

Generative Models, Adversarial imitation learning

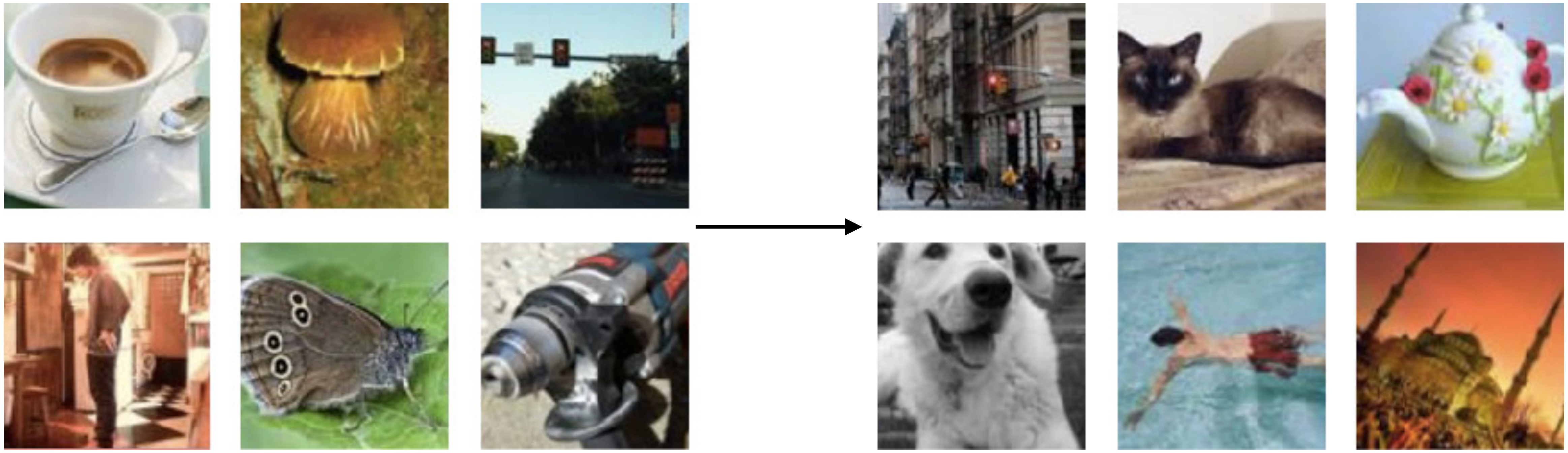
Katerina Fragkiadaki

Generative modeling

- Density estimation



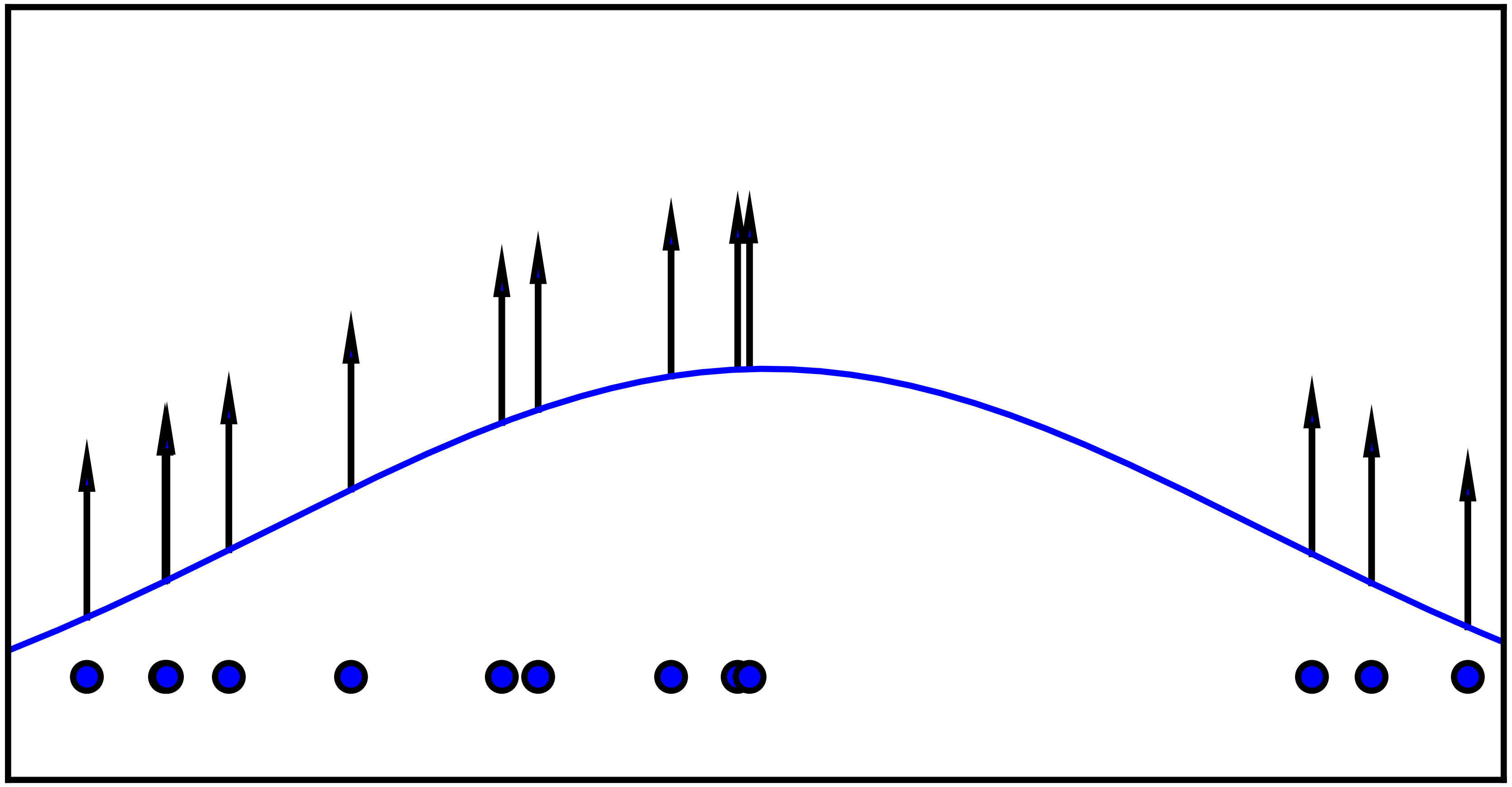
- Sample generation



Training examples Model samples

(Goodfellow 2016)

Maximum Likelihood

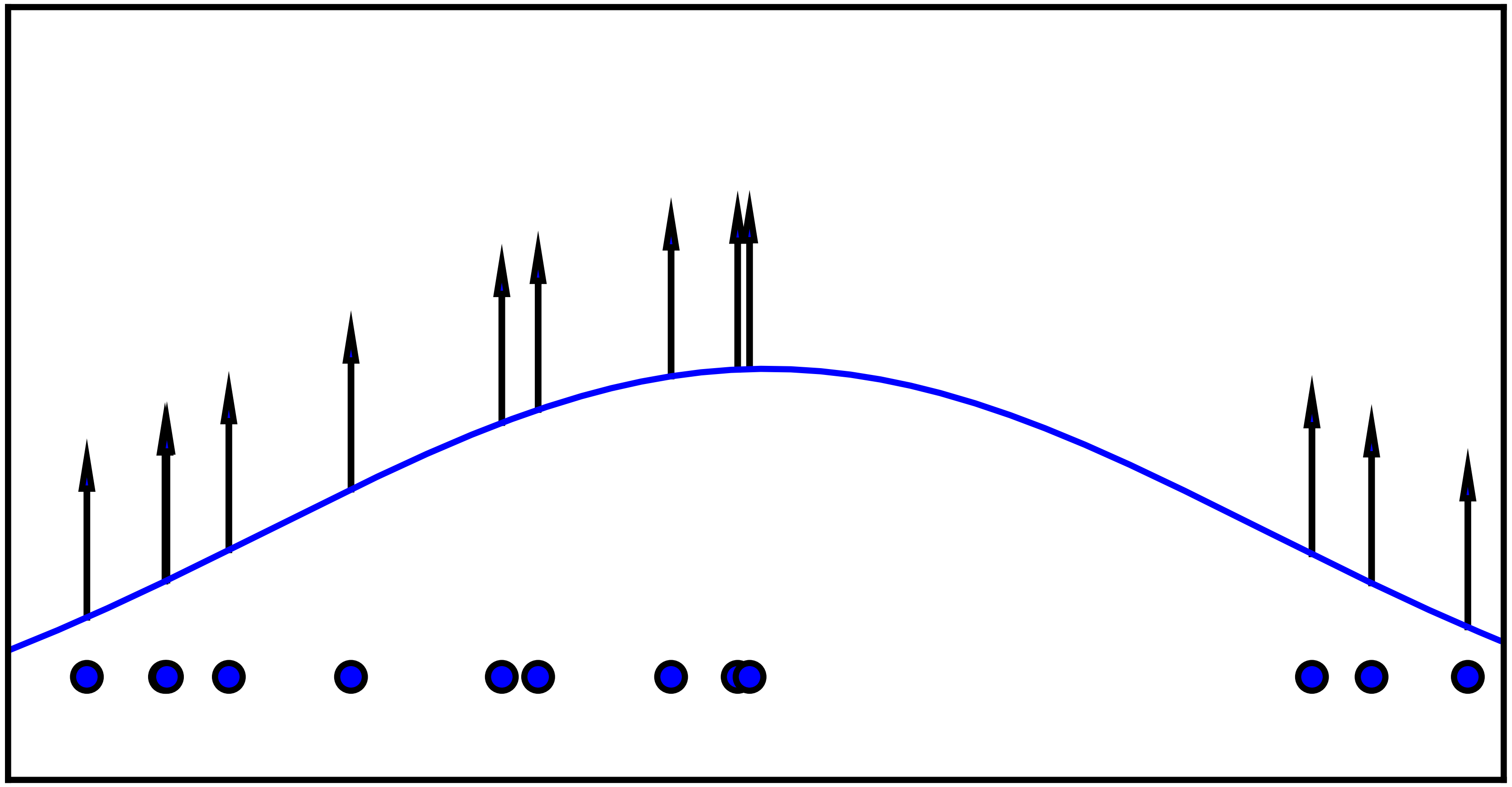


$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta)$$

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{N} \log p_{\text{model}}(\mathbf{x}_i \mid \theta)$$

(Goodfellow 2016)

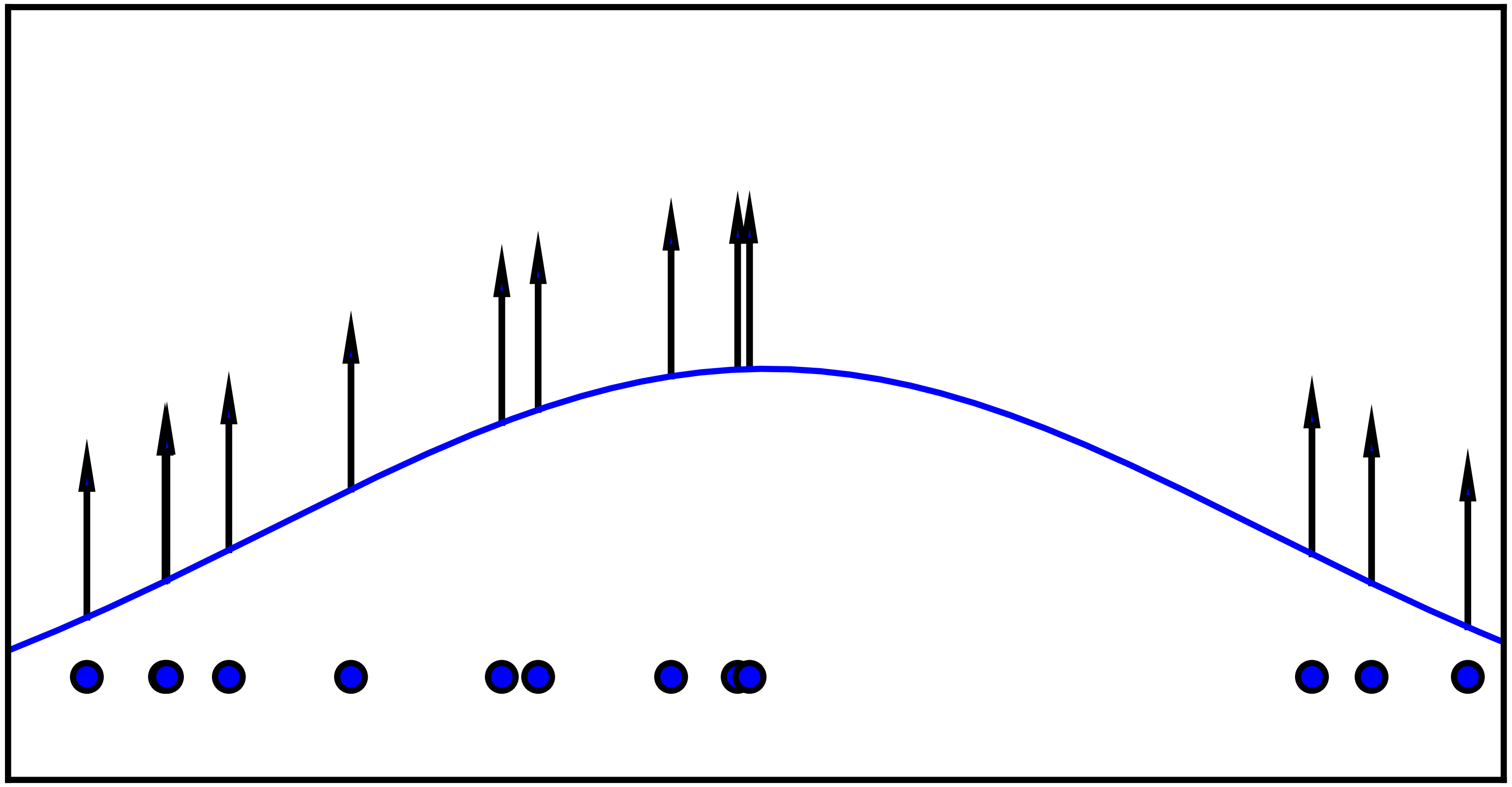
Maximum Likelihood



$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta)$$

explicit density

Maximum Conditional Likelihood



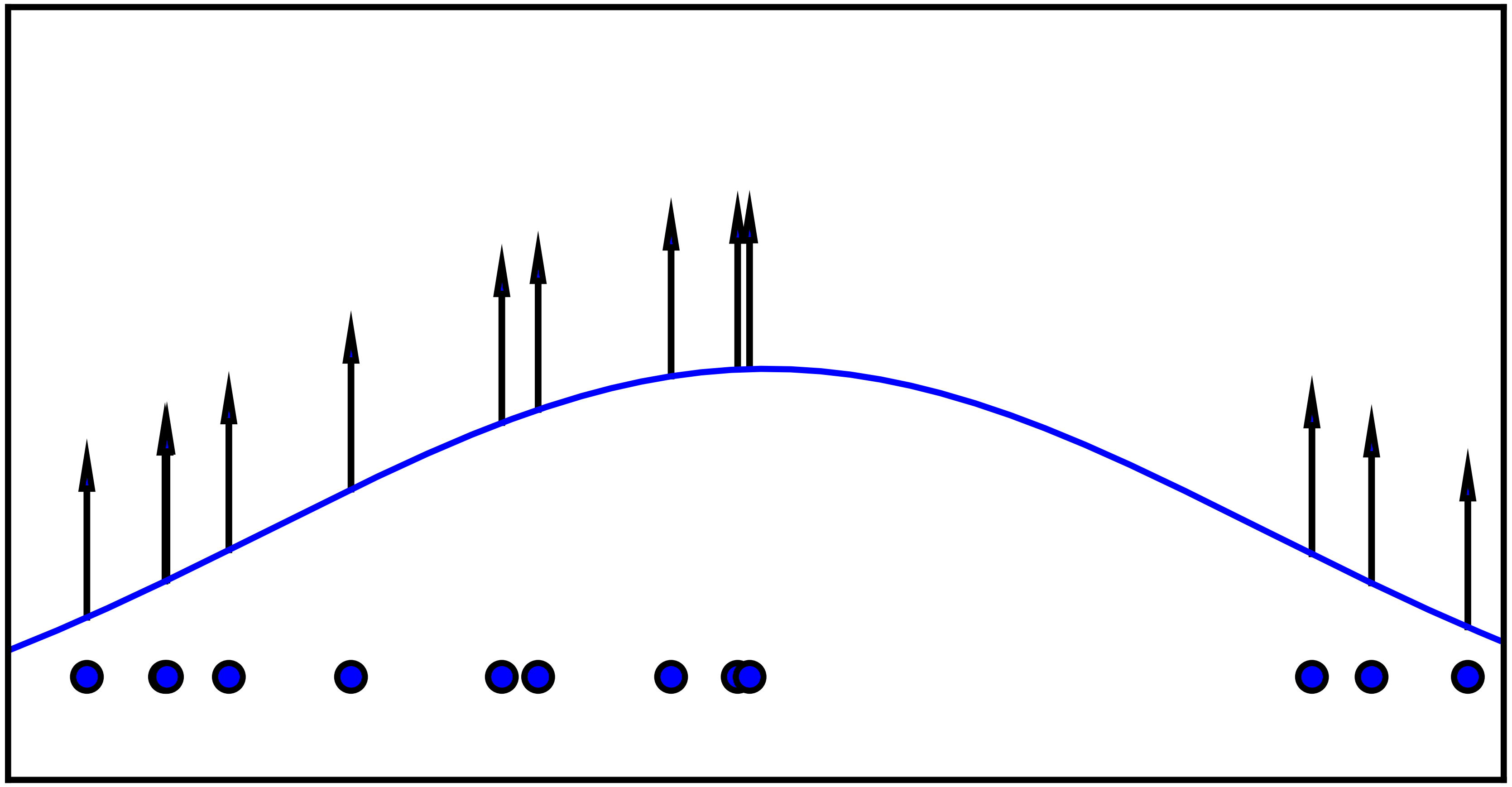
$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta, c)$$

explicit density

extra conditioning information

Maximum Conditional Likelihood

$$\mathrm{D_{KL}}(P \| Q) = -\sum_{x \in X} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$



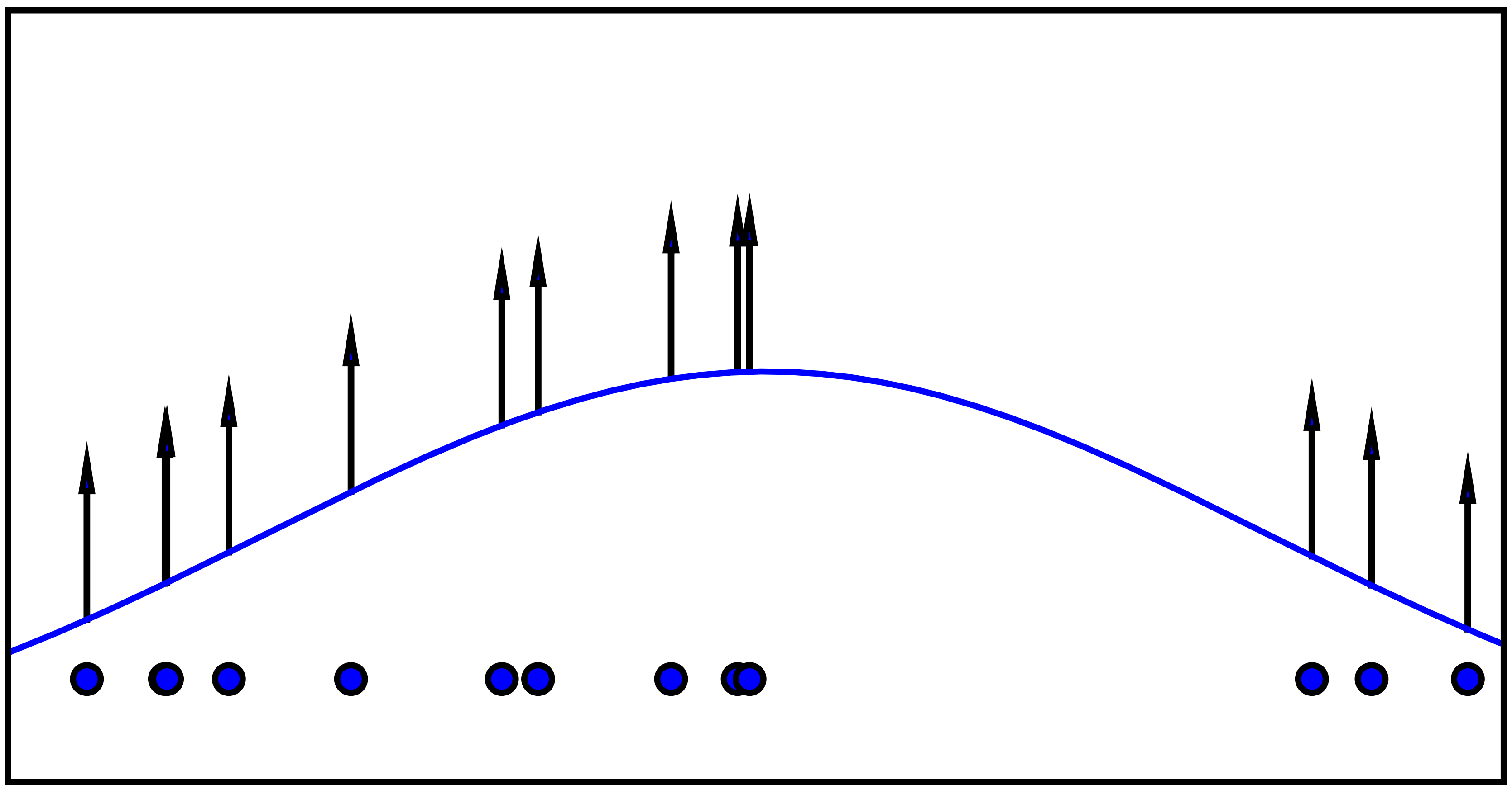
$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta, c)$$

equiv. to

$$\theta^* = \arg\min_{\theta} \mathrm{D_{KL}}\left(p_{\text{data}} \| p_{\text{model}}(\mathbf{x} \mid \theta, c)\right)$$

Maximum Conditional Likelihood

$$D_{KL}(P\|Q) = -\sum_{x \in X} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

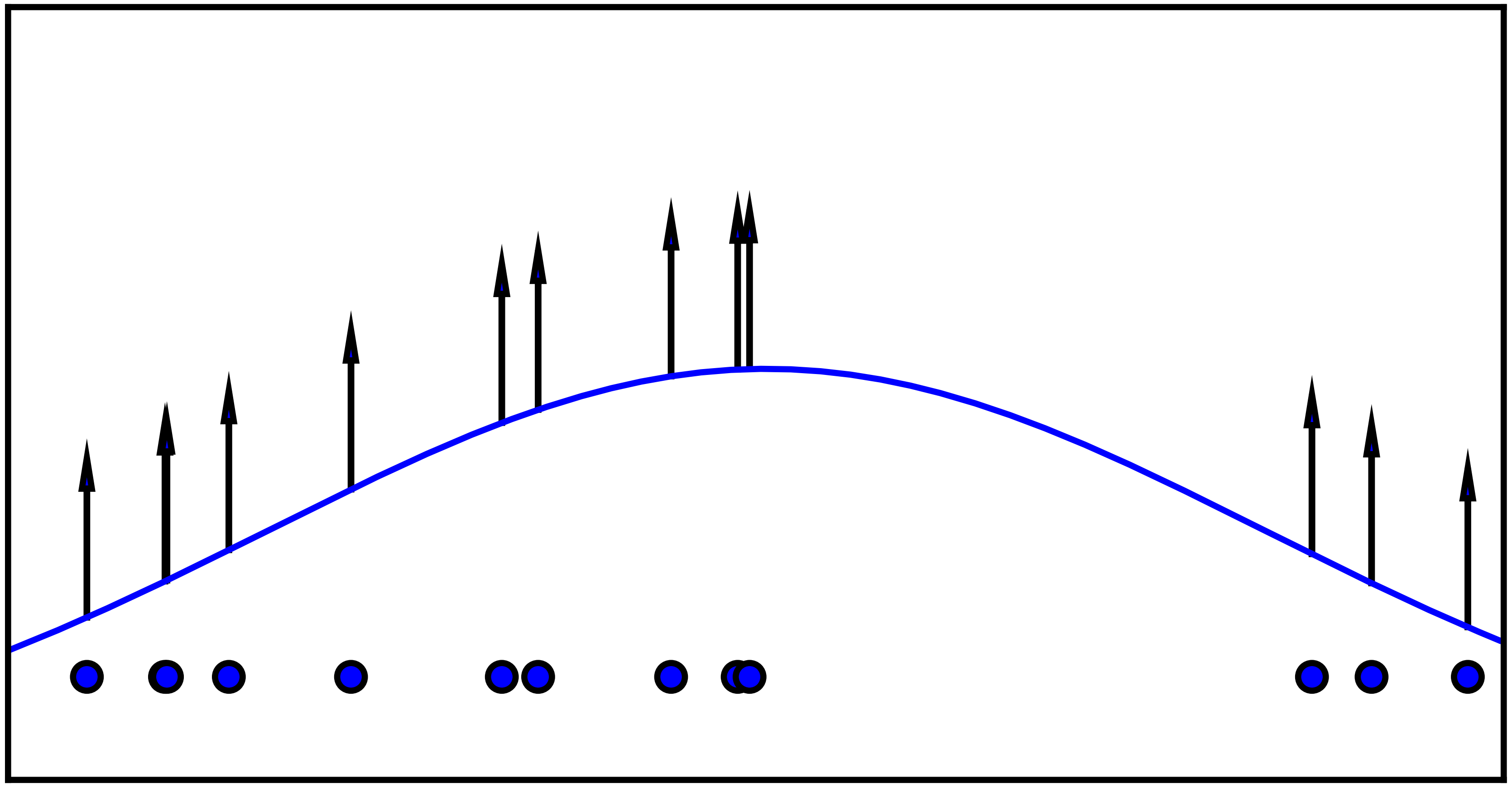


$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta)$$

equiv. to

$$\theta^* = \arg\min_{\theta} \mathrm{D}_{\mathrm{KL}}\left(p_{\text{data}} \| p_{\text{model}}(\mathbf{x} \mid \theta)\right)$$

Maximum likelihood for model learning

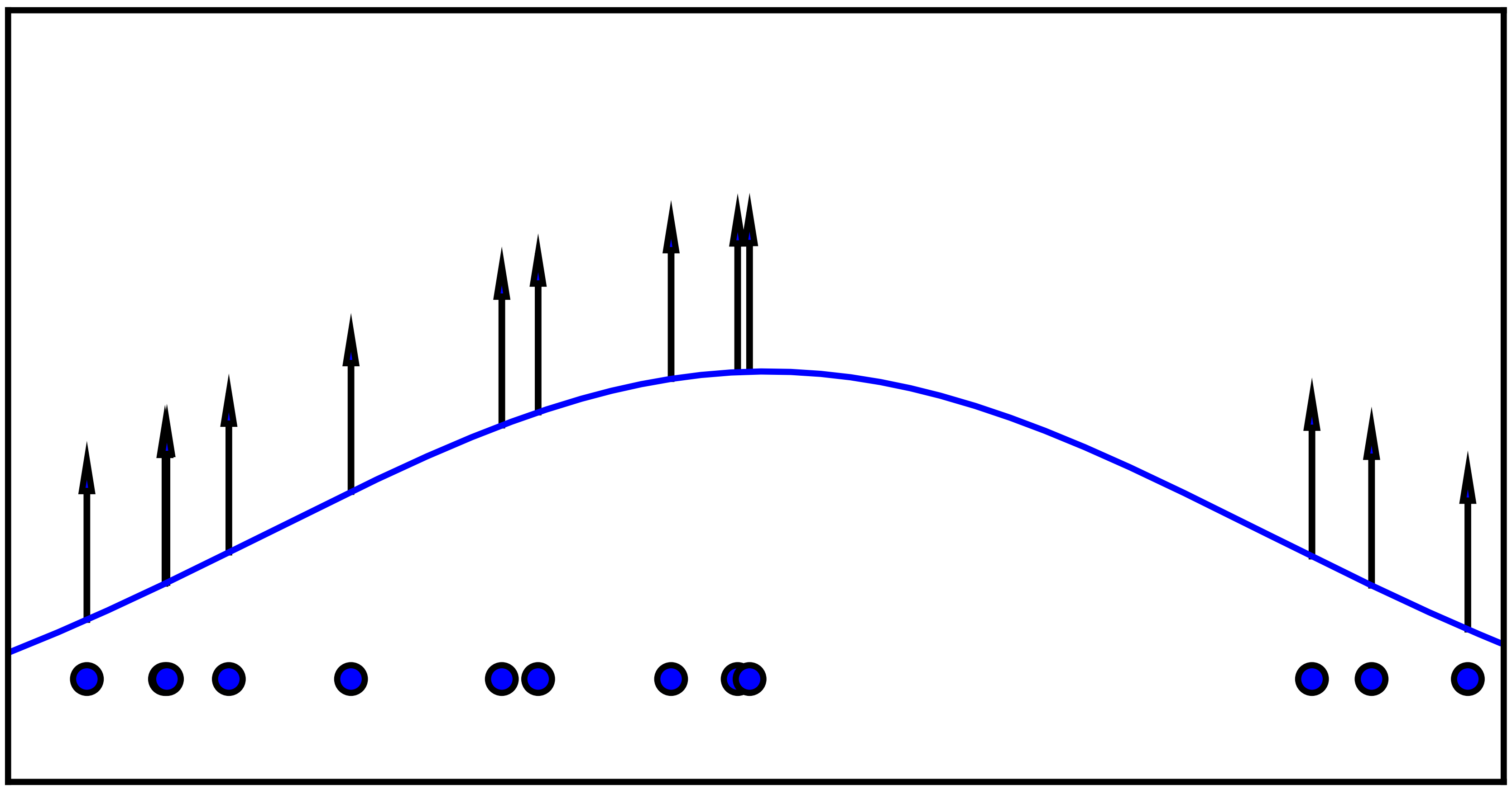


$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{s}_{t+1} \mid \theta, \mathbf{s}_t, a_t)$$

explicit density

extra conditioning information

Maximum Likelihood

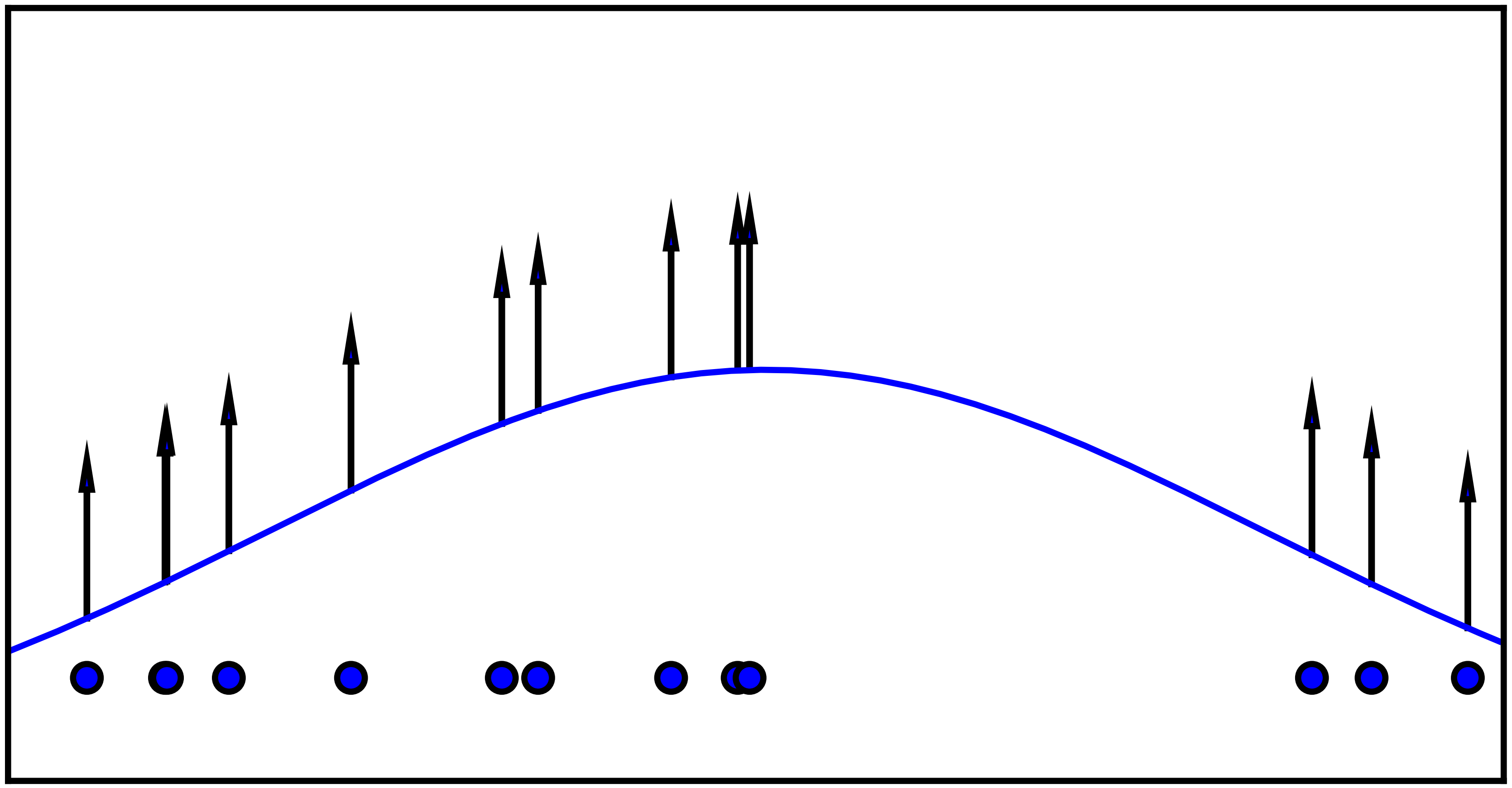


$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta, c)$$

$$p_{\text{model}}(\mathbf{x} \mid \theta, c) = \frac{1}{(2\pi)^{-\frac{k}{2}} \det(\Sigma)^{-\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu(\theta, \mathbf{c}))^\top \Sigma^{-1} (\mathbf{x} - \mu(\theta, \mathbf{c}))\right), \text{where } \Sigma = \mathbf{I}$$

Maximum Likelihood-Gaussian with fixed covariance

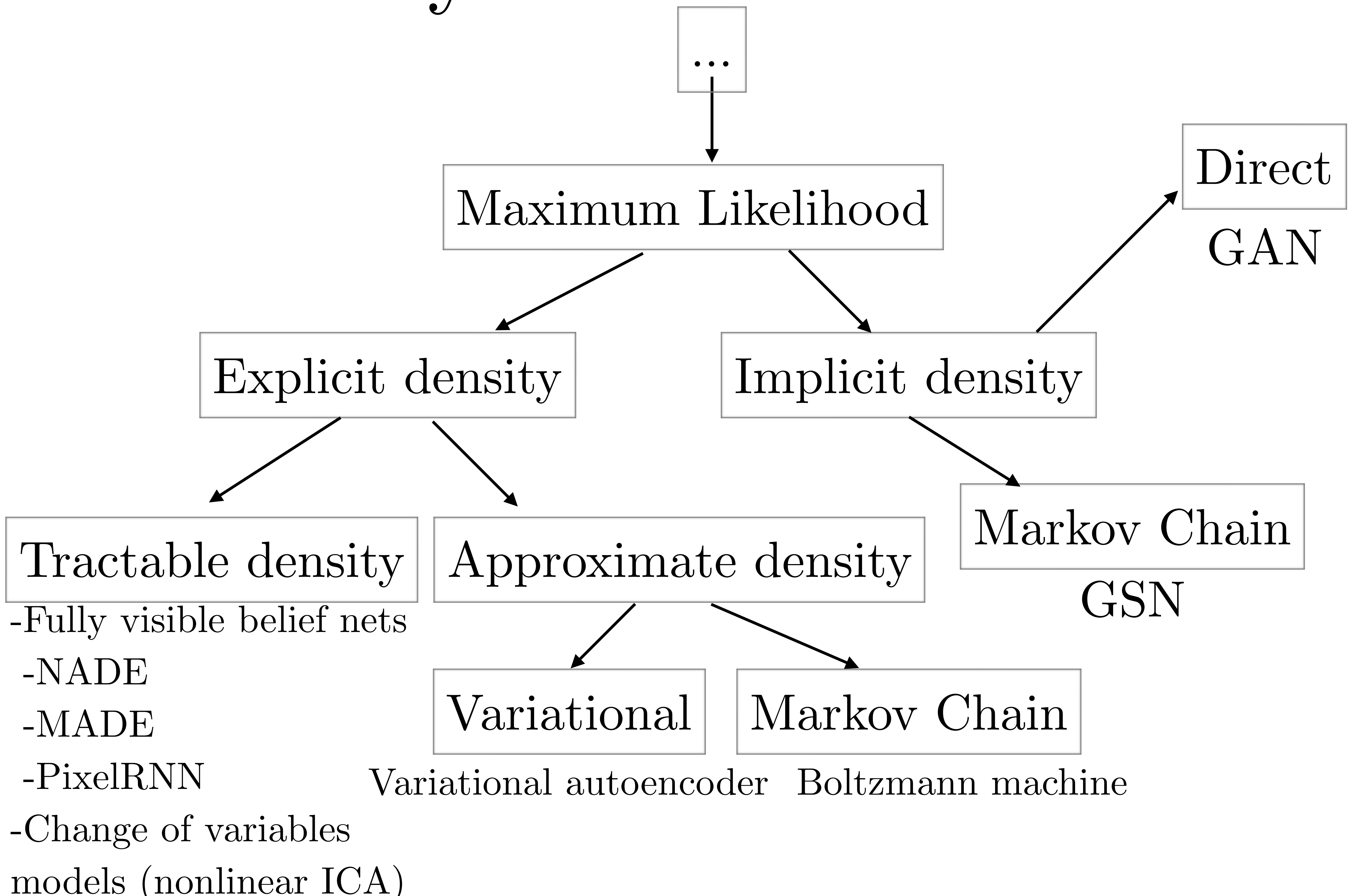
$$p_{\text{model}}(\mathbf{x} \mid \theta, c) = \frac{1}{(2\pi)^{-\frac{k}{2}} \det(\Sigma)^{-\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu(\theta, \mathbf{c}))^\top \Sigma^{-1} (\mathbf{x} - \mu(\theta, \mathbf{c}))\right), \text{ where } \Sigma = \mathbf{I}$$



$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta, c)$$

$$\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta, c) \quad \text{equiv. to} \quad \min_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \|\mathbf{x} - \mu(\theta, \mathbf{c})\|_2^2$$

Taxonomy of Generative Models



(Goodfellow 2016)

Fully Visible Belief Nets

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

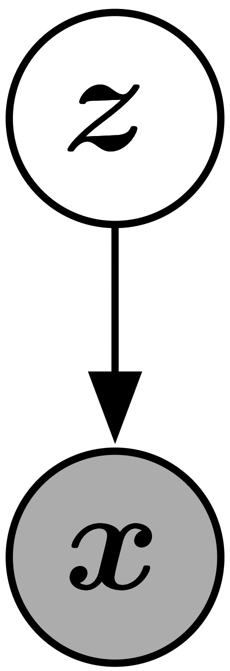
$$p_{\text{model}}(\boldsymbol{x}) = p_{\text{model}}(x_1) \prod_{i=2}^{n} p_{\text{model}}(x_i \mid x_1, \ldots, x_{i-1})$$

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

PixelCNN elephants (van den Ord et al 2016)

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

Energy based models

$$p_\theta(\mathbf{x}) = \frac{1}{Z} \exp\left(-E_\theta(\mathbf{x})\right)$$

Remember from previous class our energy based model over trajectories, where we parametrized the trajectory cost:

$$p(\tau \,|\, \theta) = \frac{e^{-c_\theta(\tau)}}{\sum_{\tau'} e^{-c_\theta(\tau')}}$$

where $\tau = \{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$ is a state/action trajectory

and the cost of a trajectory $c_\theta(\tau)$ is additive over states: $c_\theta(\tau) = \sum_t c_\theta(s_t, a_t)$.

Energy based models

$$p_\theta(\mathbf{x}) = \frac{1}{Z} \exp\left(-E_\theta(\mathbf{x})\right)$$

Maximizing likelihood requires sampling to estimate Z:

$$\frac{d}{d\theta_i} \log p_\theta(\mathbf{x}) = \frac{d}{d\theta_i} \left(-E_\theta(\mathbf{x}) - \log Z\right)$$

Maximum Likelihood

$$\max_{\theta} . \sum_{\tau_i \in \mathrm{D}_{\text{demo}}} \log p(\tau_i)$$

Maximum Likelihood

$$\max_{\theta} . \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} \log p(\tau_i)$$

$$\Longleftrightarrow \max_{\theta} . \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} \log \frac{e^{-c_\theta(\tau_i)}}{Z}$$

Maximum Likelihood

$$\max_{\theta} . \sum_{\tau_i \in \mathrm{D}_{\text{demo}}} \log p(\tau_i)$$

$$\iff \max_{\theta} . \sum_{\tau_i \in \mathrm{D}_{\text{demo}}} \log \frac{e^{-c_\theta(\tau_i)}}{Z}$$

$$\iff \max_{\theta} . \sum_{\tau_i \in \mathrm{D}_{\text{demo}}} - c_\theta(\tau_i) - \sum_{\tau_i \in \mathrm{D}_{\text{demo}}} \log Z$$

Maximum Likelihood

$$\max_{\theta}. \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} \log p(\tau_i)$$

$$\Longleftrightarrow \max_{\theta}. \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} \log \frac{e^{-c_\theta(\tau_i)}}{Z}$$

$$\Longleftrightarrow \max_{\theta}. \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} - c_\theta(\tau_i) - \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} \log Z$$

$$\Longleftrightarrow \max_{\theta}. \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} - c_\theta(\tau_i) - \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} \log\left(\sum_{\tau} e^{-c_\theta(\tau)}\right)$$

This is a huge sum, intractable to compute in large state spaces.

Sample approximation for Z

This is a huge integral, intractable to compute: $Z = \int e^{-c_\theta(\tau)} d\tau$

$$Z = \int e^{-c_\theta(\tau)} d\tau = \int q(\tau) \frac{e^{-c_\theta(\tau)}}{q(\tau)} d\tau \approx \frac{1}{|\mathrm{D}_{\mathrm{samp}}|} \sum_{\tau_j \in \mathrm{D}_{\mathrm{samp}}} \frac{e^{-c_\theta(\tau_j)}}{q(\tau_j)}$$

$$\mathscr{L}(\theta) = \frac{1}{|\mathrm{D}_{\mathrm{demo}}|} \sum_{\tau_i \in \mathrm{D}_{\mathrm{demo}}} c_\theta(\tau_i) + \log \left(\frac{1}{|\mathrm{D}_{\mathrm{samp}}|} \sum_{\tau_j \in \mathrm{D}_{\mathrm{samp}}} \frac{e^{-c_\theta(\tau_j)}}{q(\tau_j)} \right)$$

$$\mathscr{L}(\theta) = \mathbb{E}_{\tau \sim p_{\mathrm{demo}}} c_\theta(\tau) + \log \left(\mathbb{E}_{\tau \sim q} \frac{\exp(-c_\theta(\tau))}{q(\tau)} \right)$$

What q shall we use? Let’s adapt it over time!

MaxEntIRL with Adaptive Importance Sampling

1. Initialize q_0 either from a random policy or using behavior cloning on expert demonstations.
2. for iteration $k = 1 \ldots I$
    3. Generate samples D_{traj} from $q_k(\tau)$
    4. Append samples: $\mathrm{D}_{\text{samp}} \leftarrow \mathrm{D}_{\text{samp}} \cup \mathrm{D}_{\text{traj}}$.
    5. Use D_{samp} to update cost c_θ using gradient descent.
    6. Update $q_k(\tau)$ using any RL method

$$\nabla_\theta \mathscr{L}(\theta) = \frac{1}{|\mathrm{D}_{\text{demo}}|} \sum_{\tau_i \in \mathrm{D}_{\text{demo}}} \frac{dc_\theta}{d\theta}(\tau_i) - \log\left(\frac{1}{|\mathrm{D}_{\text{samp}}|} \sum_{\tau_j \in \mathrm{D}_{\text{samp}}} \frac{e^{-c_\theta(\tau_j)}}{q(\tau_j)} \frac{dc_\theta}{d\theta}(\tau_j) \right)$$

Guided cost learning, Finn et al. 2016

MaxEntIRL with Adaptive Importance Sampling

1. Initialize q_0 either from a random policy or using behavior cloning on expert demonstations.
2. for iteration $k = 1...I$
   3. Generate samples D_{traj} from $q_k(\tau)$
   4. Append samples: $\mathrm{D}_{\mathrm{samp}} \leftarrow \mathrm{D}_{\mathrm{samp}} \cup \mathrm{D}_{\mathrm{traj}}$.
   5. Use $\mathrm{D}_{\mathrm{samp}}$ to update cost c_θ using gradient descent.
   6. Update $q_k(\tau)$ using any RL method

maximize entropy of the policy

$$\mathscr{L}(q) = \boxed{\mathbb{E}_{\tau \sim q} c_\theta(\tau)} + \boxed{\mathbb{E}_{\tau \sim q}[\log q(\tau)]}$$

Minimize cost (equiv. to maximize reward)

Guided cost learning, Finn et al. 2016

IRL versus IL

In the first lecture, we had seen methods that imitate the experts directly, without trying to recover a reward.

Behaviour cloning: $$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \log \pi_\theta(a_i \mid s_i)$$

equiv. to $$\min_\theta \sum_{i=1}^{N} \|\pi_\theta(s_i) - a_i\|_2^2$$ for a gaussian policy with a unit covariance

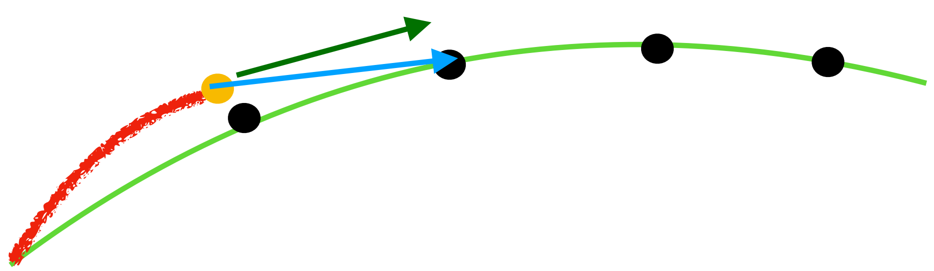
or, using an RNN: $$\min_\theta \sum_{i=1}^{N} \|\pi_\theta(s_i - T .. s_i) - a_i\|_2^2$$

One problem we had was distribution shift.

Scheduled sampling (sampling from the output of the model during training) could alleviate that.

IRL versus IL

$$\min_{\theta} \sum_{i=1}^{N} \|\pi_\theta(s_i - T..s_i) - a_i\|_2$$

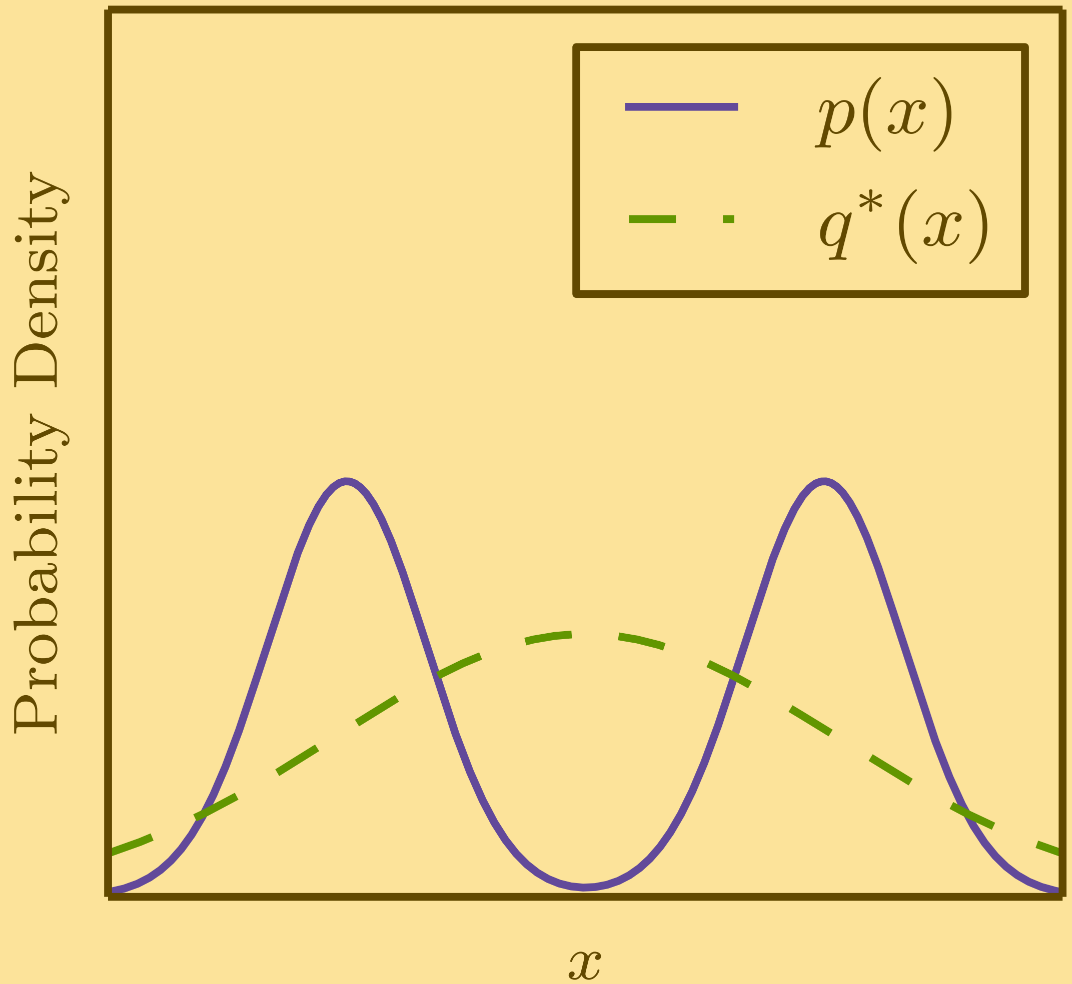


- From that point, either you query the expert on what to do
- Or you are asked to map back to the original trajectory

Other problems?

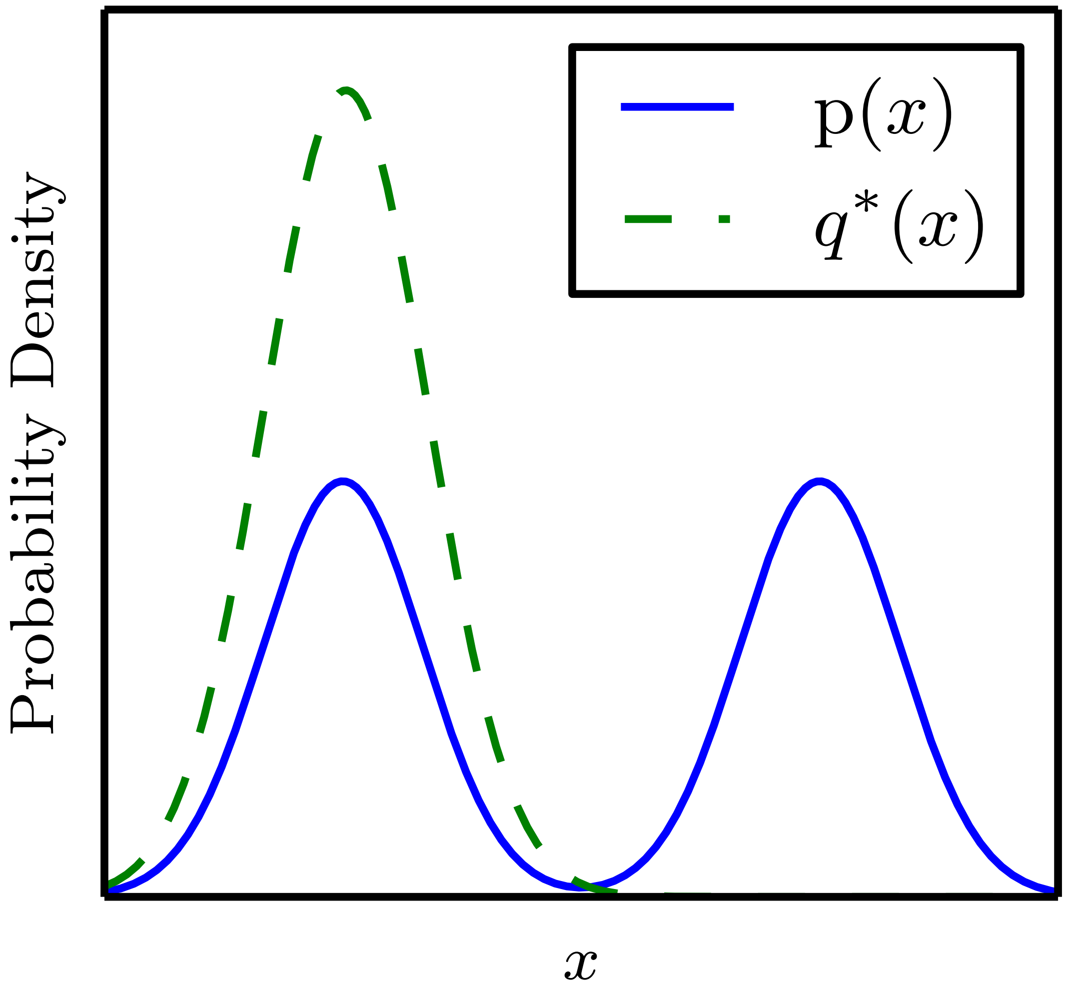
Behaviour cloning

$q^* = \text{argmin}_q D_{\text{KL}}(p\|q)$



Maximum likelihood

$q^* = \text{argmin}_q D_{\text{KL}}(q\|p)$

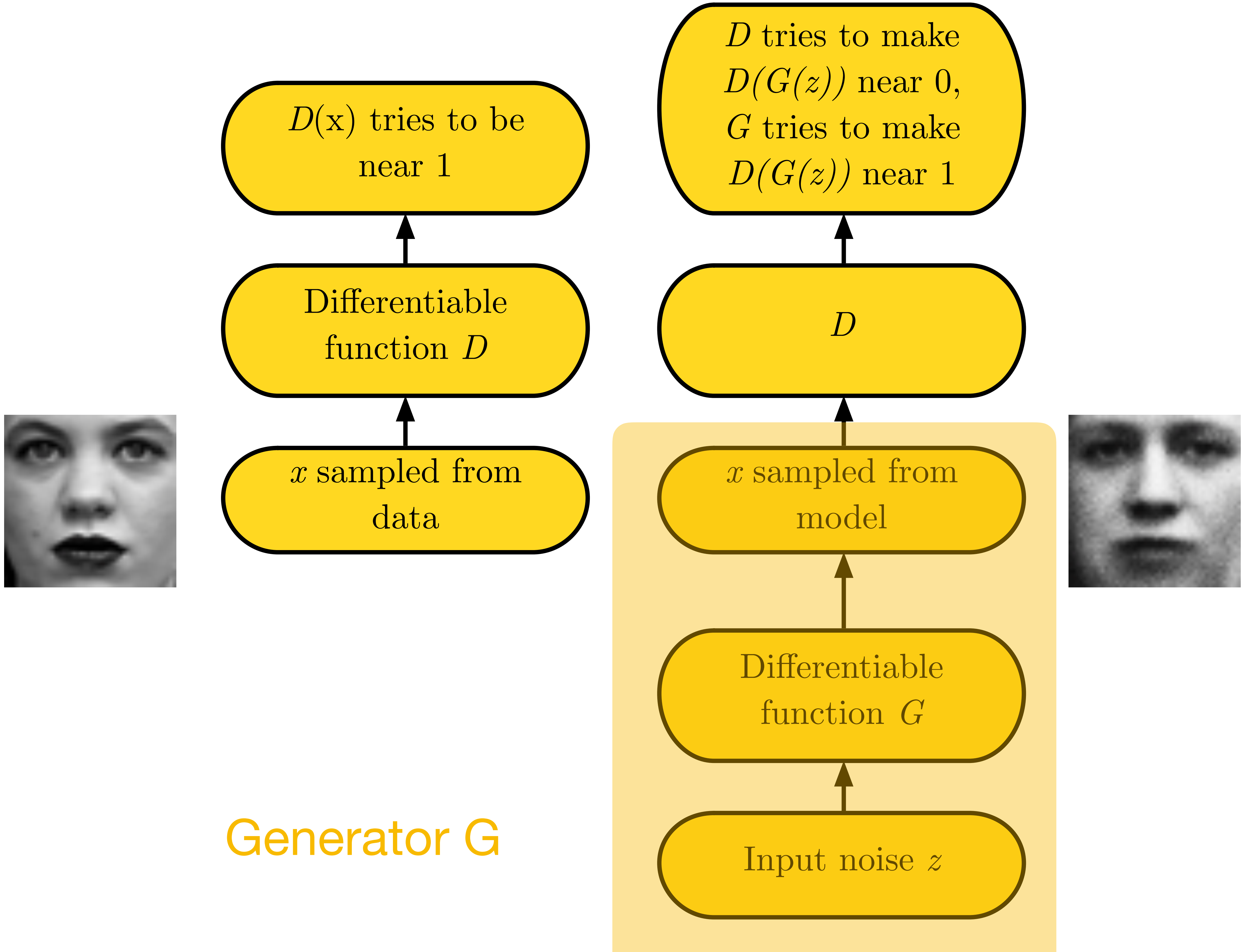


Reverse KL

Non realistic action samples, due to non expressive policy (plain regressor)

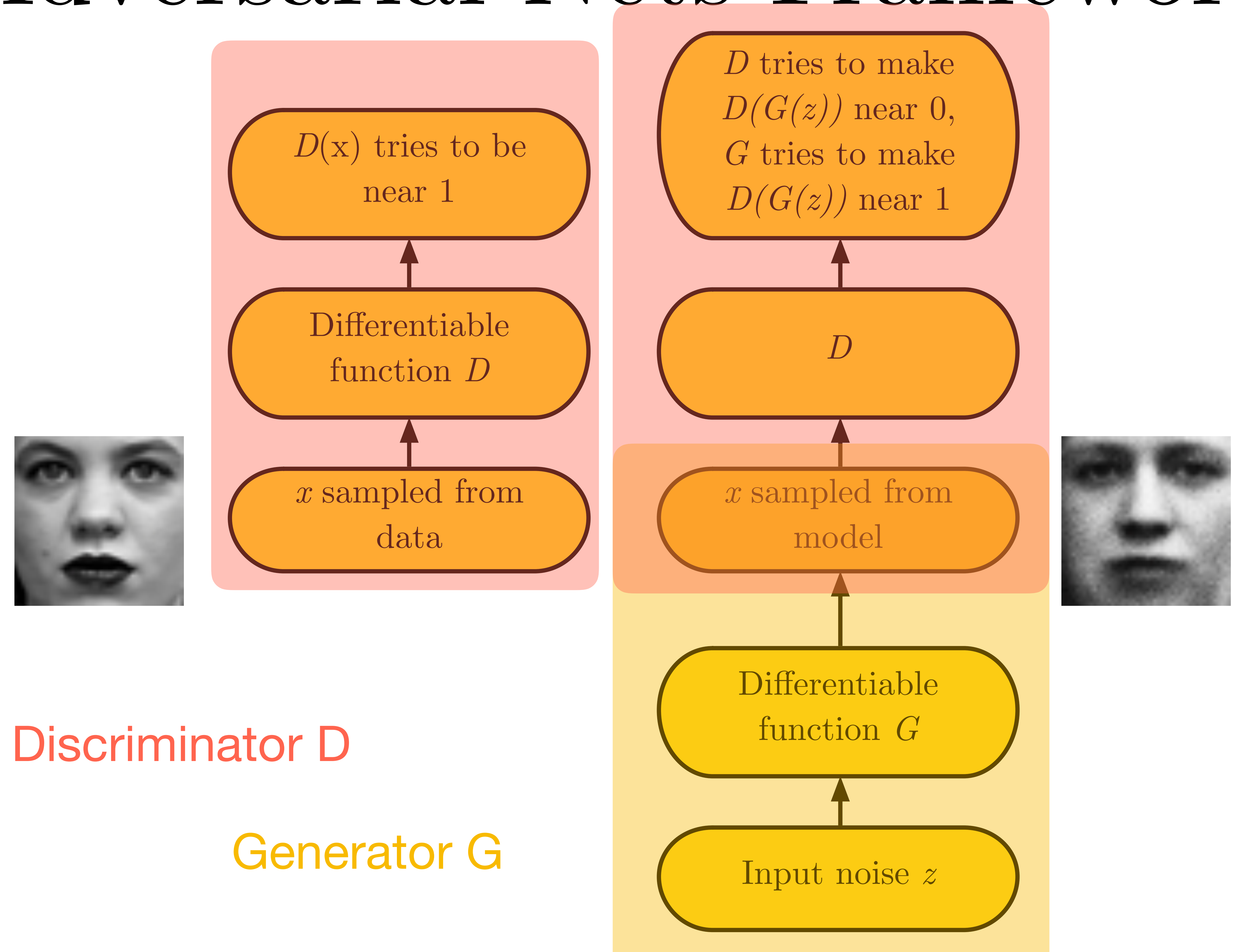
GANs to the rescue

Adversarial Nets Framework



(Goodfellow 2016)

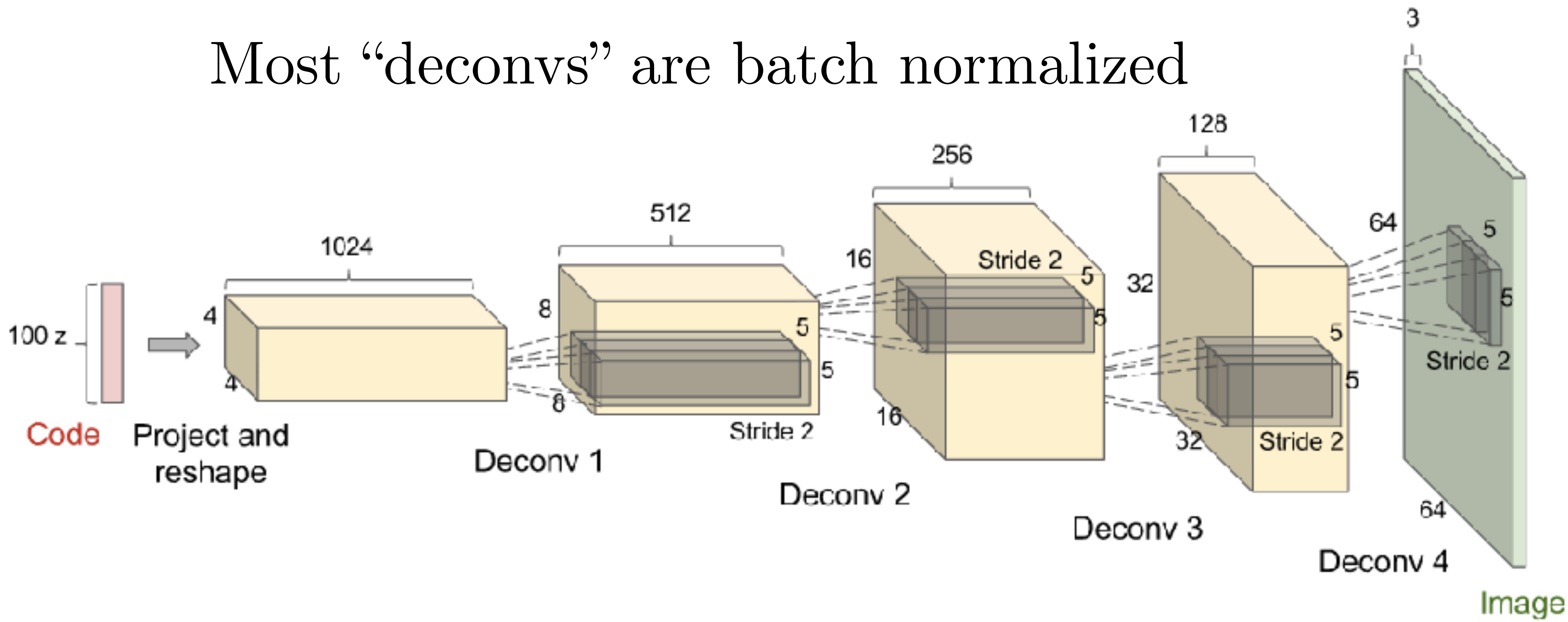
Adversarial Nets Framework



(Goodfellow 2016)

A Generator network (DCGAN)

Most “deconvs” are batch normalized



(Radford et al 2015)

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)

Optimal discriminator strategy

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

Optimal discriminator strategy

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

$$V(D,G) = \int_x p_{\text{data}}(x)\log D(x)dx + \int_z p_z(z)\log(1-D(G(z)))dz$$

Optimal discriminator strategy

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

$$V(D, G) = \int_x p_{\text{data}}(x) \log D(x) dx + \int_z p_z(z) \log(1 - D(G(z))) dz$$

$$\int_x p_{\text{data}}(x) \log D(x) dx + \int_x p_G(x) \log(1 - D(x)) dx$$

Optimal discriminator strategy

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

$$V(D,G) = \int_x p_{\mathrm{data}}(x)\log D(x)dx + \int_z p_z(z)\log(1-D(G(z)))dz$$

$$\int_x p_{\mathrm{data}}(x)\log D(x)dx + \int_x p_G(x)\log(1-D(x))dx$$

$$\int_x p_{\mathrm{data}}(x)\log D(x) + p_G(x)\log(1-D(x))dx$$

Optimal discriminator strategy

$$V(D,G) = \int_x p_{\text{data}}(x)\log D(x) + p_G(x)\log(1 - D(x))dx$$

Optimal discriminator strategy

$$V(D,G) = \int_x p_{\text{data}}(x) \log D(x) + p_G(x) \log(1 - D(x)) dx$$

$$\frac{d}{dD(x)} \left(p_{\text{data}}(x) \log D(x) + p_G(x) \log(1 - D(x)) \right) = 0$$

Optimal discriminator strategy

$$V(D,G) = \int_x p_{\text{data}}(x)\log D(x) + p_G(x)\log(1 - D(x))dx$$

$$\frac{d}{dD(x)}\left(p_{\text{data}}(x)\log D(x) + p_G(x)\log(1 - D(x)\right) = 0$$

$$p_{\text{data}}(x)\frac{1}{D(x)} - p_G(x)\frac{1}{1 - D(x)} = 0$$

Optimal discriminator strategy

$$V(D,G) = \int_x p_{\text{data}}(x)\log D(x) + p_G(x)\log(1 - D(x))dx$$

$$\frac{d}{dD(x)}\left(p_{\text{data}}(x)\log D(x) + p_G(x)\log(1 - D(x)\right) = 0$$

$$p_{\text{data}}(x)\frac{1}{D(x)} - p_G(x)\frac{1}{1 - D(x)} = 0$$

$$p_{\text{data}}(x)\frac{1}{D(x)} = p_G(x)\frac{1}{1 - D(x)}$$

Optimal discriminator strategy

$$V(D,G) = \int_x p_{\text{data}}(x)\log D(x) + p_G(x)\log(1 - D(x))dx$$

$$\frac{d}{dD(x)}\left(p_{\text{data}}(x)\log D(x) + p_G(x)\log(1 - D(x))\right) = 0$$

$$p_{\text{data}}(x)\frac{1}{D(x)} - p_G(x)\frac{1}{1 - D(x)} = 0$$

$$p_{\text{data}}(x)\frac{1}{D(x)} = p_G(x)\frac{1}{1 - D(x)}$$

$$p_{\text{data}}(x)(1 - D(x)) = p_G(x)D(x)$$

Optimal discriminator strategy

$$V(D,G) = \int_x p_{\text{data}}(x) \log D(x) + p_G(x) \log(1 - D(x))dx$$

$$\frac{d}{dD(x)}\left(p_{\text{data}}(x) \log D(x) + p_G(x) \log(1 - D(x))\right) = 0$$

$$p_{\text{data}}(x)\frac{1}{D(x)} - p_G(x)\frac{1}{1 - D(x)} = 0$$

$$p_{\text{data}}(x)\frac{1}{D(x)} = p_G(x)\frac{1}{1 - D(x)}$$

$$p_{\text{data}}(x)(1 - D(x)) = p_G(x)D(x)$$

$$D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}$$

Optimal generator strategy

$$C(G) = \max_D V(G, D)$$

Optimal generator strategy

$$\begin{aligned} C(G) &= \max_D V(G, D) \\ &= \mathbb{E}_{x\sim p_{data}(x)}[\log D^*_G(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D^*_G(G(z))] \end{aligned}$$

Optimal generator strategy

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_G^*(G(z))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - D_G^*(x)]
\end{aligned}
$$

Optimal generator strategy

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_G^*(G(z))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - D_G^*(x)] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)})]
\end{aligned}
$$

Optimal generator strategy

$$\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D^*_G(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D^*_G(G(z))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D^*_G(x)] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - D^*_G(x)] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\mathrm{data}}(x)}{p_{\mathrm{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - \frac{p_{\mathrm{data}}(x)}{p_{\mathrm{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\mathrm{data}}(x)}{p_{\mathrm{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\mathrm{G}}(x)}{p_{\mathrm{data}}(x) + p_G(x)})]
\end{aligned}$$

Optimal generator strategy

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_G^*(G(z))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - D_G^*(x)] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4 + \log 4
\end{aligned}
$$

Optimal generator strategy

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_G^*(G(z))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - D_G^*(x)] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4 + \log 4 \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{2p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{2p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4
\end{aligned}
$$

Optimal generator strategy

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_G^*(G(z))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - D_G^*(x)] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4 + \log 4 \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{2p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{2p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4 \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{\frac{p_{\text{data}}(x) + p_G(x)}{2}}] + \mathbb{E}_{x \sim p_G(x)}[\log \frac{p_{\text{G}}(x)}{\frac{p_{\text{data}}(x) + p_G(x)}{2}}] - \log 4
\end{aligned}
$$

Optimal generator strategy

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_G^*(G(z))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - D_G^*(x)] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4 + \log 4 \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{2p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{2p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4 \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{\frac{p_{\text{data}}(x) + p_G(x)}{2}}] + \mathbb{E}_{x \sim p_G(x)}[\log \frac{p_{\text{G}}(x)}{\frac{p_{\text{data}}(x) + p_G(x)}{2}}] - \log 4 \\
&= \mathrm{D}_{\text{KL}}\left(p_{data}(x) \,||\, \frac{p_{\text{data}}(x) + p_G(x)}{2}\right) + \mathrm{D}_{\text{KL}}\left(p_G(x) \| \frac{p_{\text{data}}(x) + p_G(x)}{2}\right) - \log 4
\end{aligned}
$$

Optimal generator strategy

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D_G^*(G(z)))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log D_G^*(x)] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - D_G^*(x))] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(1 - \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4 + \log 4 \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{2p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{2p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] - \log 4 \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{\frac{p_{\text{data}}(x) + p_G(x)}{2}}] + \mathbb{E}_{x \sim p_G(x)}[\log \frac{p_{\text{G}}(x)}{\frac{p_{\text{data}}(x) + p_G(x)}{2}}] - \log 4 \\
&= \mathrm{D_{KL}}\left(p_{data}(x) \,||\, \frac{p_{\text{data}}(x) + p_G(x)}{2}\right) + \mathrm{D_{KL}}\left(p_G(x) \| \frac{p_{\text{data}}(x) + p_G(x)}{2}\right) - \log 4 \\
&= 2\mathrm{D_{JSD}}\left(p_{data}(x) \,||\, p_G(x)\right) - \log 4
\end{aligned}
$$

Optimal generator strategy

$$C(G) = \max_D V(G, D)$$

$$= \mathbb{E}_{x\sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x\sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})]$$

$$= 2\mathrm{D}_{\text{JSD}}\left(p_{data}(x) \,||\, p_G(x)\right) - \log 4$$

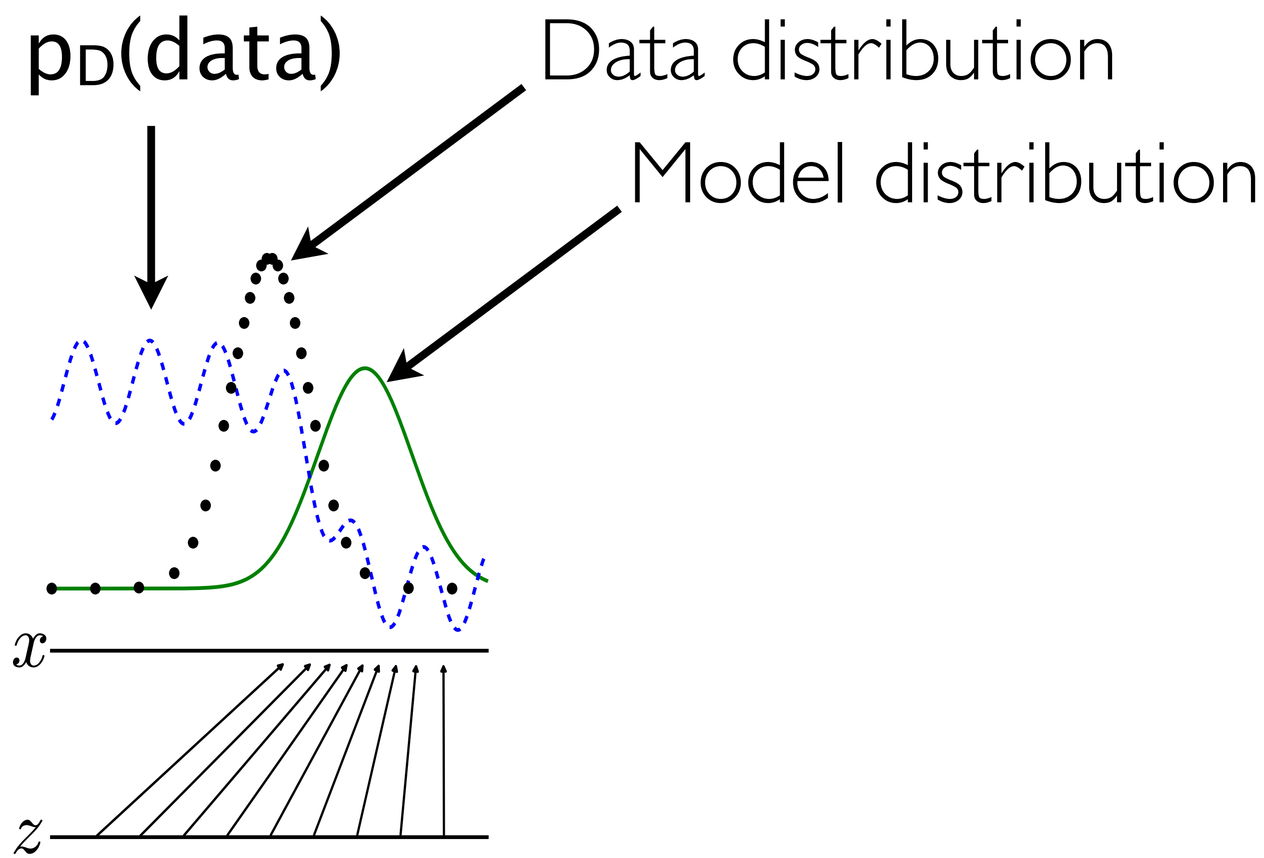
Since $\mathrm{D}_{\text{JSD}} \geq 0, \quad C(G) \geq -\log 4$

We setting $P_G(x) = \mathrm{p}_{\text{data}}(\mathrm{x})$ in the equation above, we get:

$$C(G) = \mathbb{E}_{x\sim \mathrm{p}_{\text{data}}(x)} \log \frac{1}{2} + \mathbb{E}_{x\sim p_G(x)} \log \frac{1}{2} = -\log 4$$

Thus generator achieves the optimum when $P_G(x) = \mathrm{p}_{\text{data}}(\mathrm{x})$.

Learning process



Poorly fit model

Diagram from Ian Goodfellow

Learning process

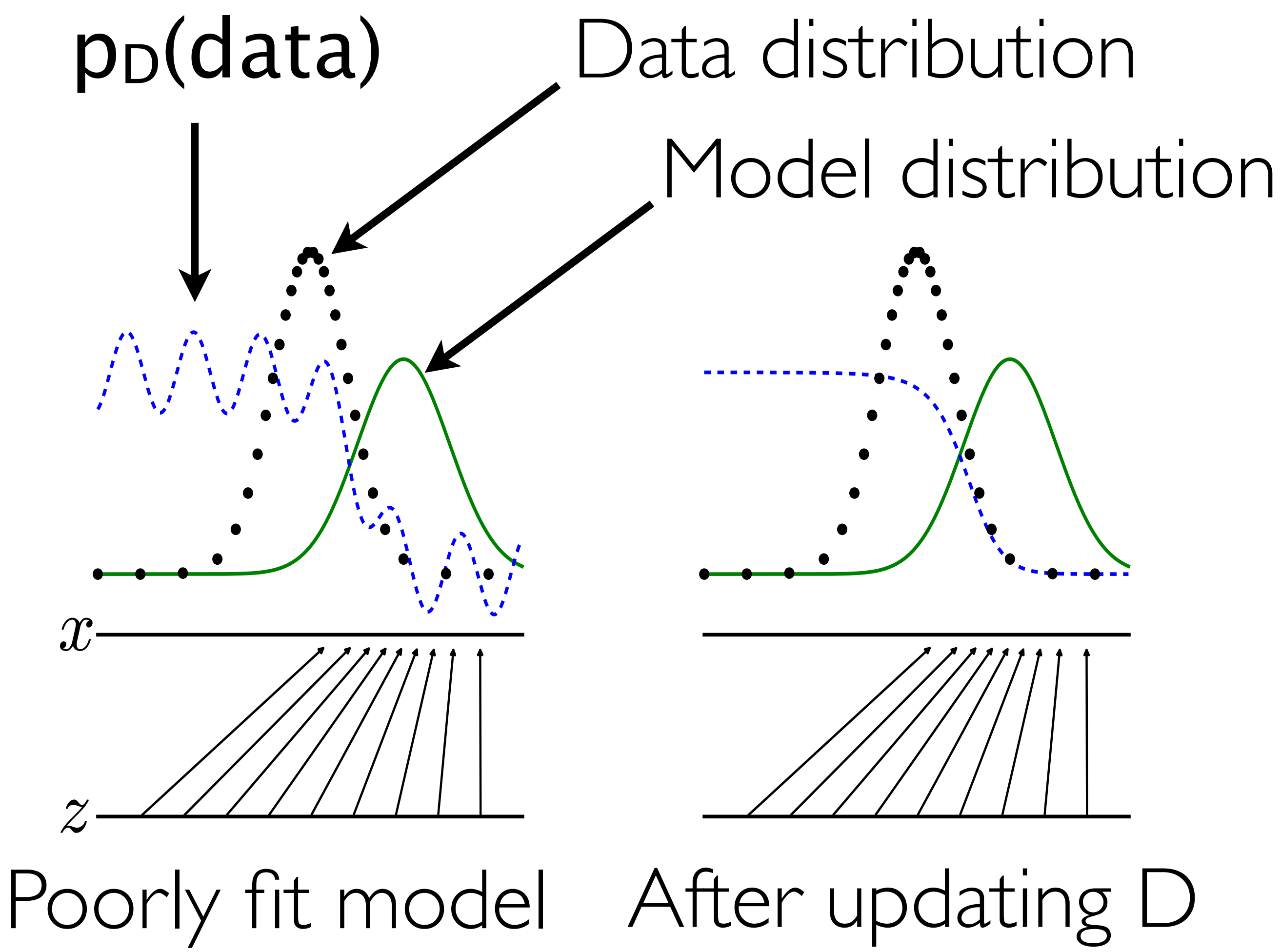
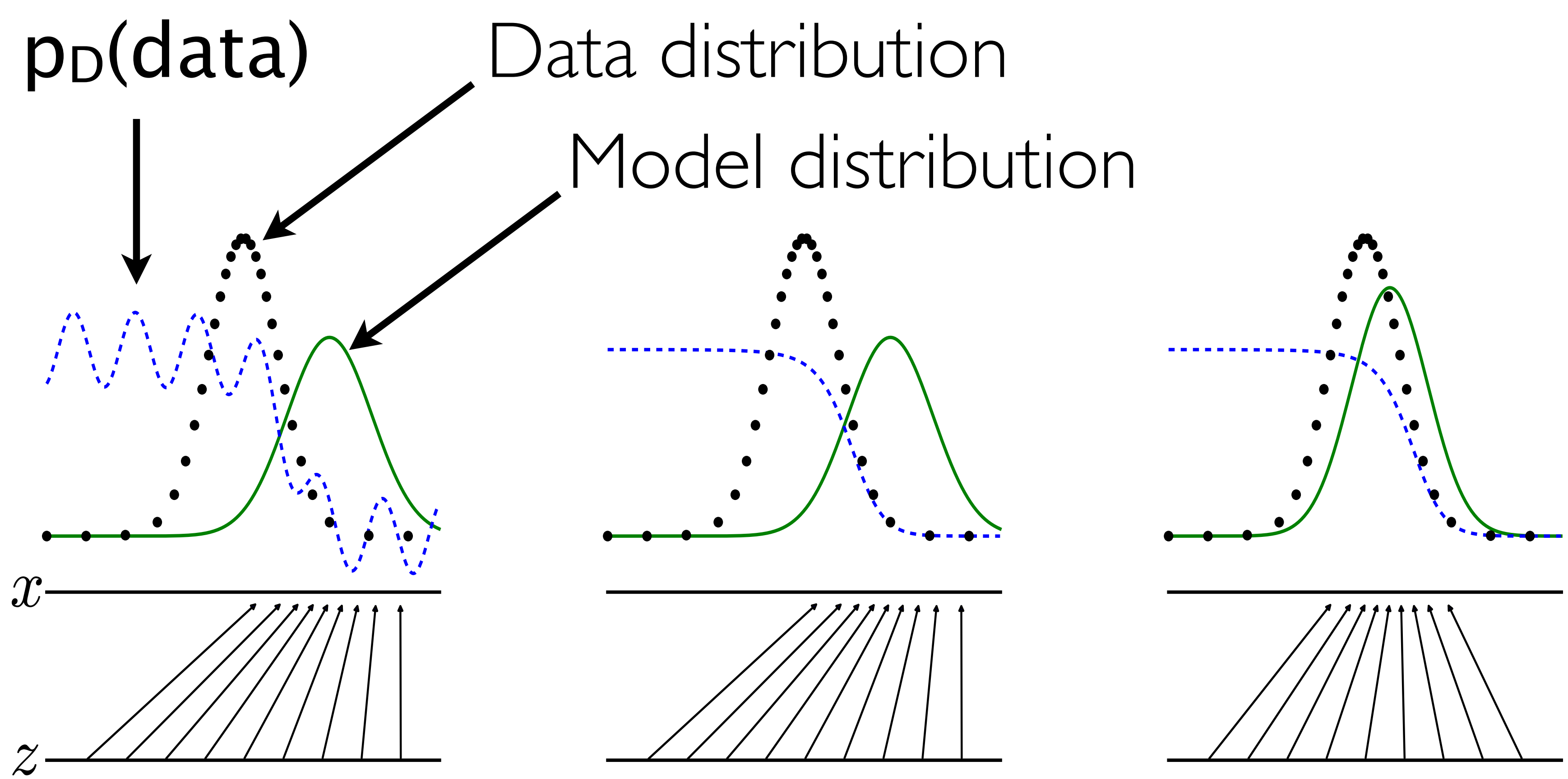


Diagram from Ian Goodfellow

Learning process



Poorly fit model After updating D After updating G

Diagram from Ian Goodfellow

Learning process

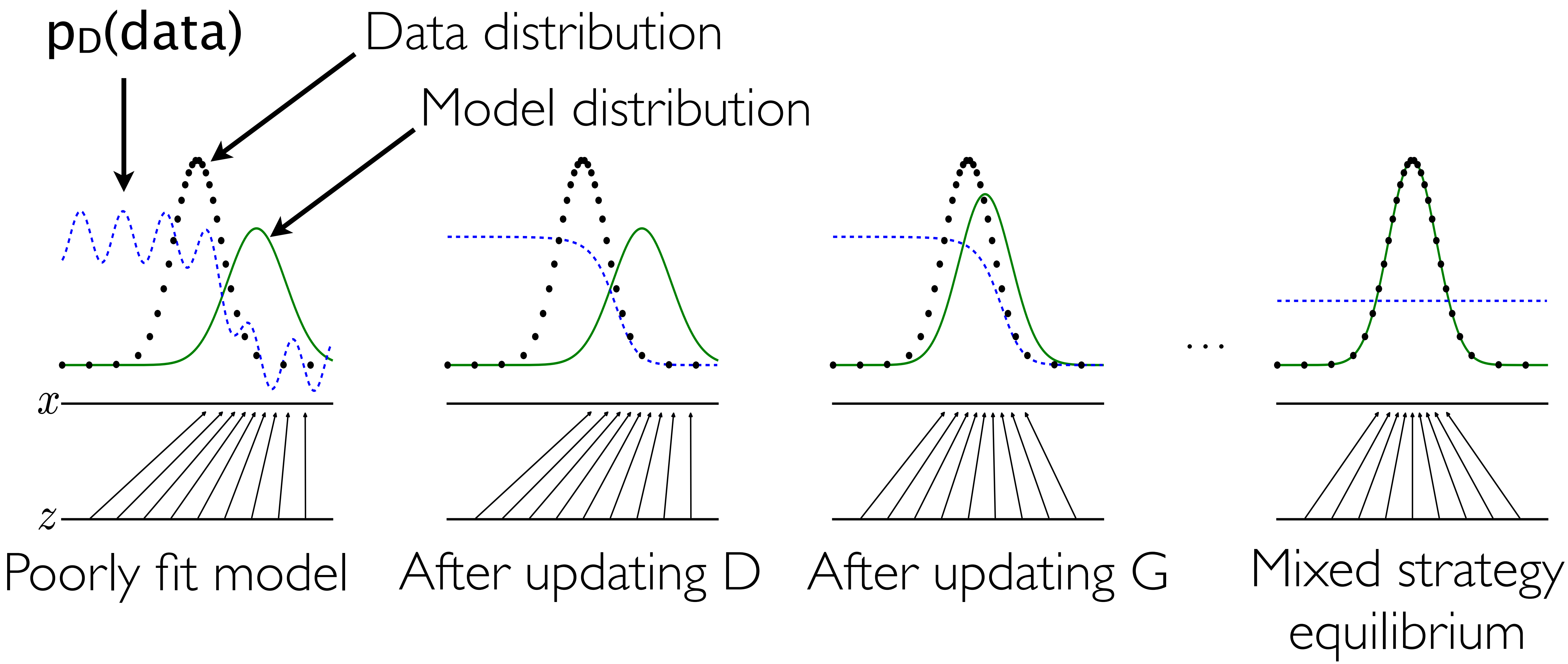


Diagram from Ian Goodfellow

Non-Saturating Game

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

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

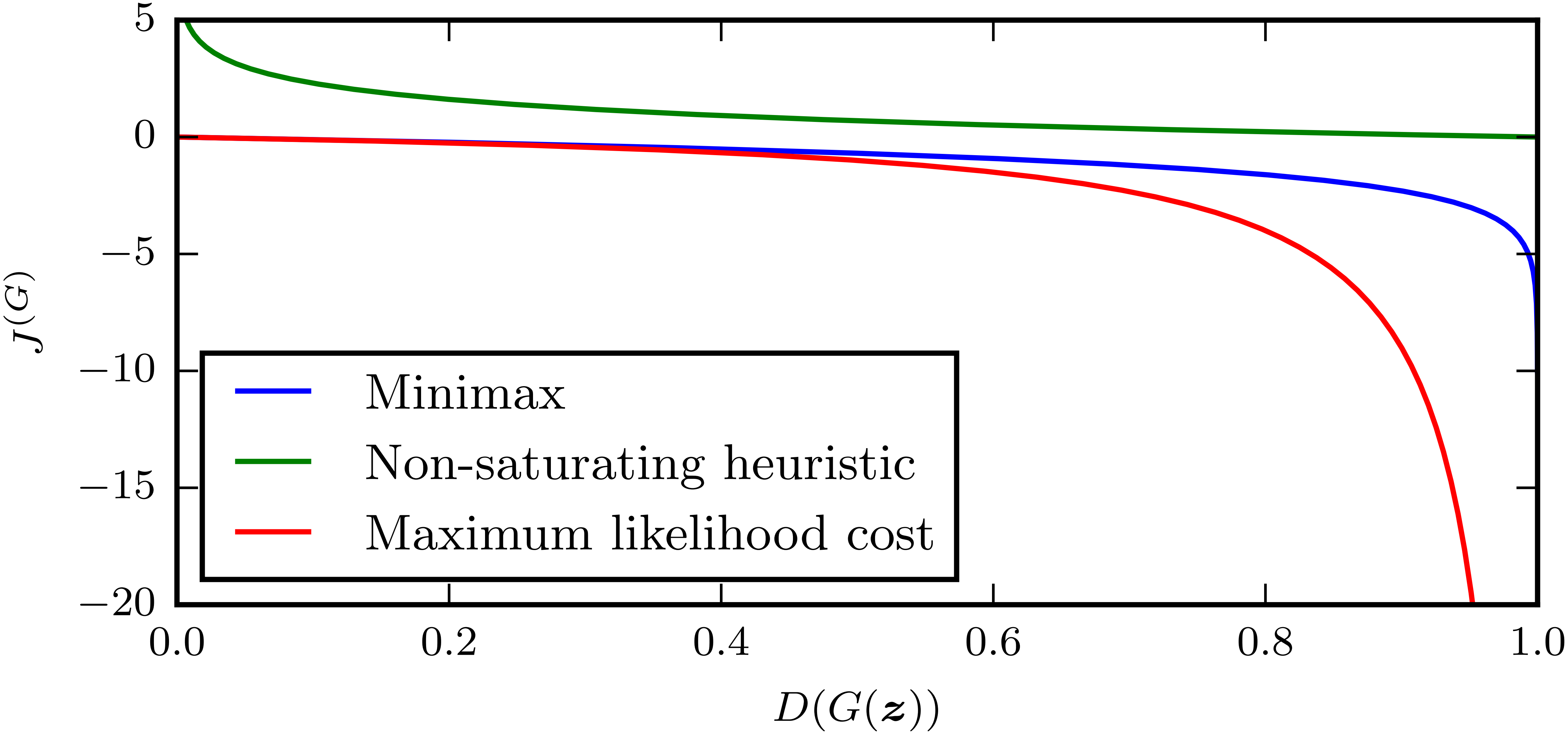
-Equilibrium no longer describable with a single loss

-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)

Comparison of Generator Losses

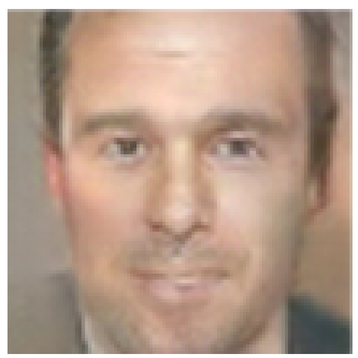
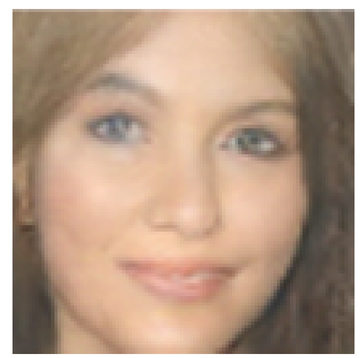
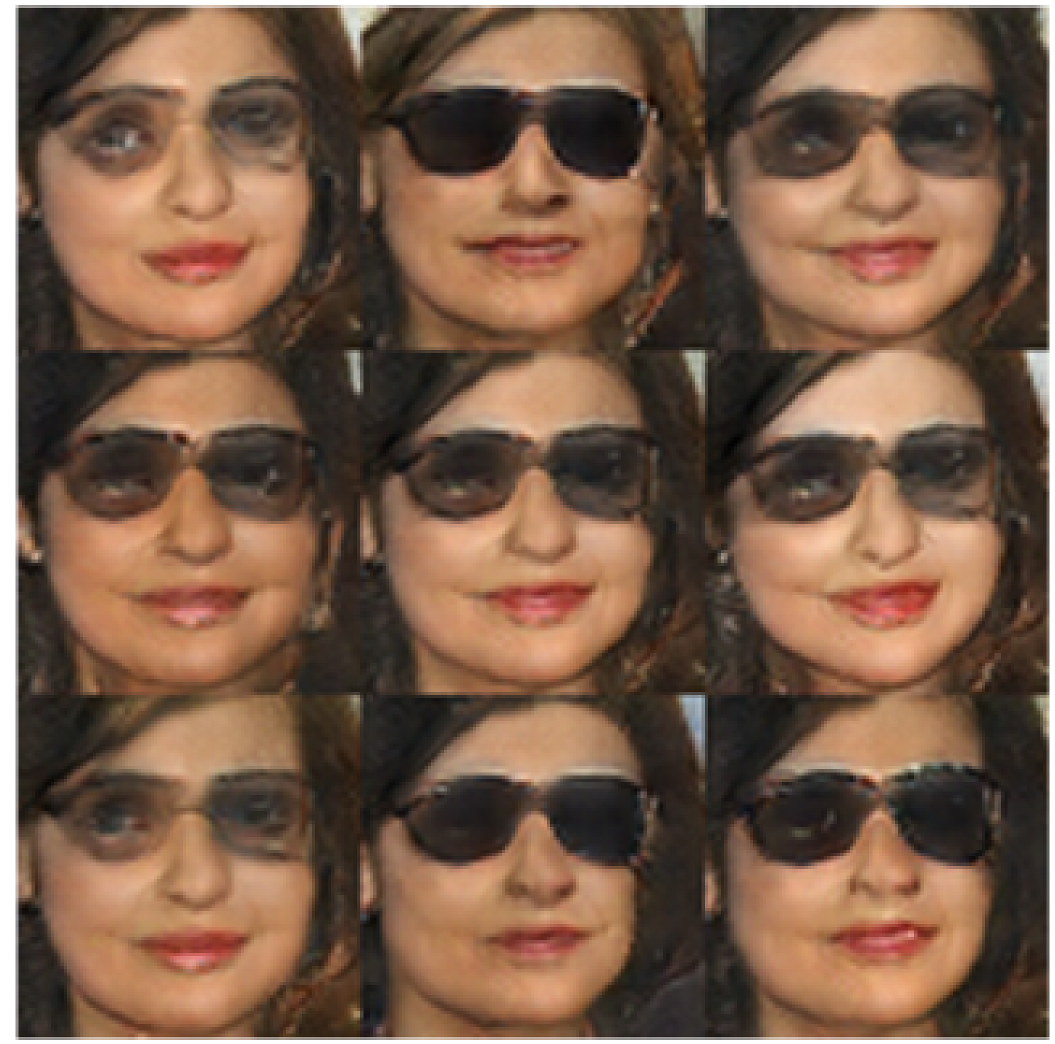


(Goodfellow 2014)

DCGANs for LSUN Bedrooms

(Radford et al 2015)

Vector Space Arithmetic

 -  +  = 

Man with glasses

Man

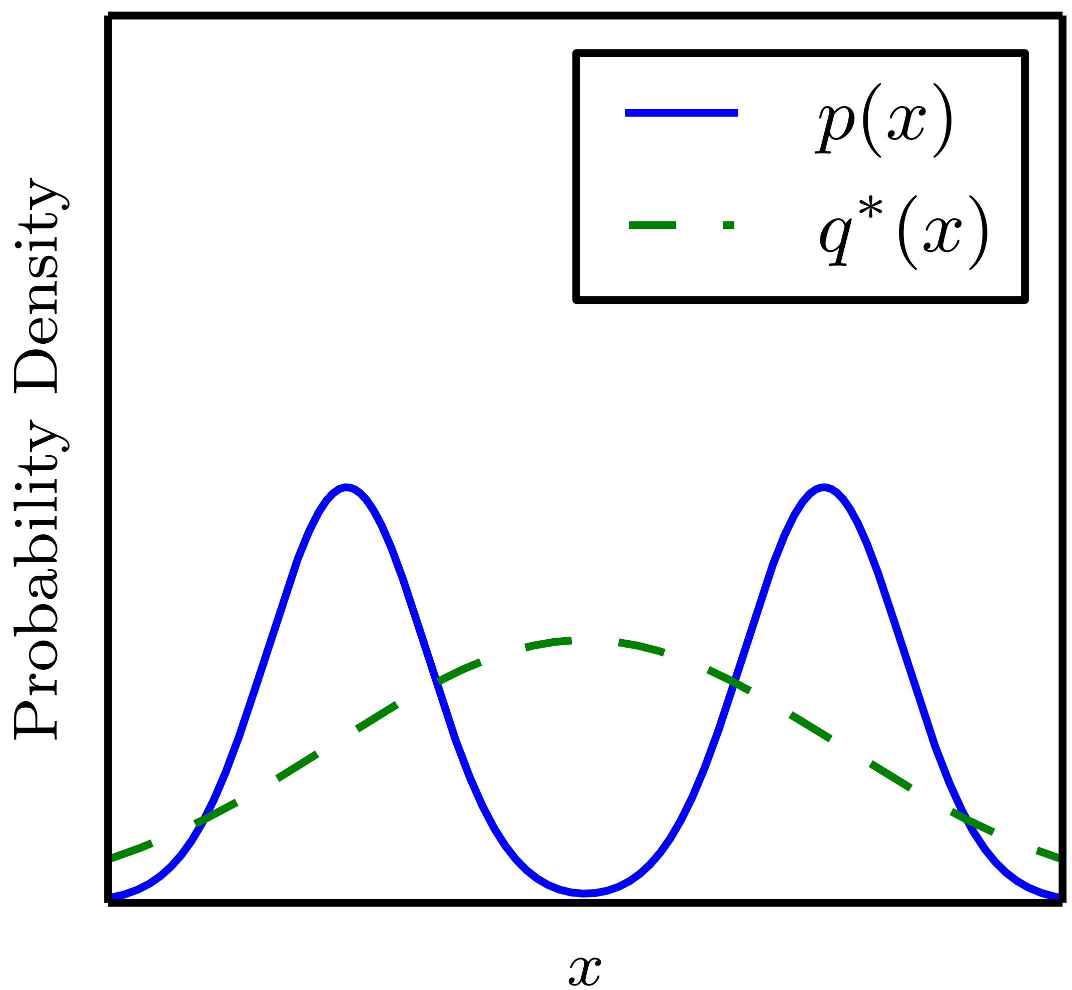
Woman

Woman with Glasses

(Radford et al, 2015)

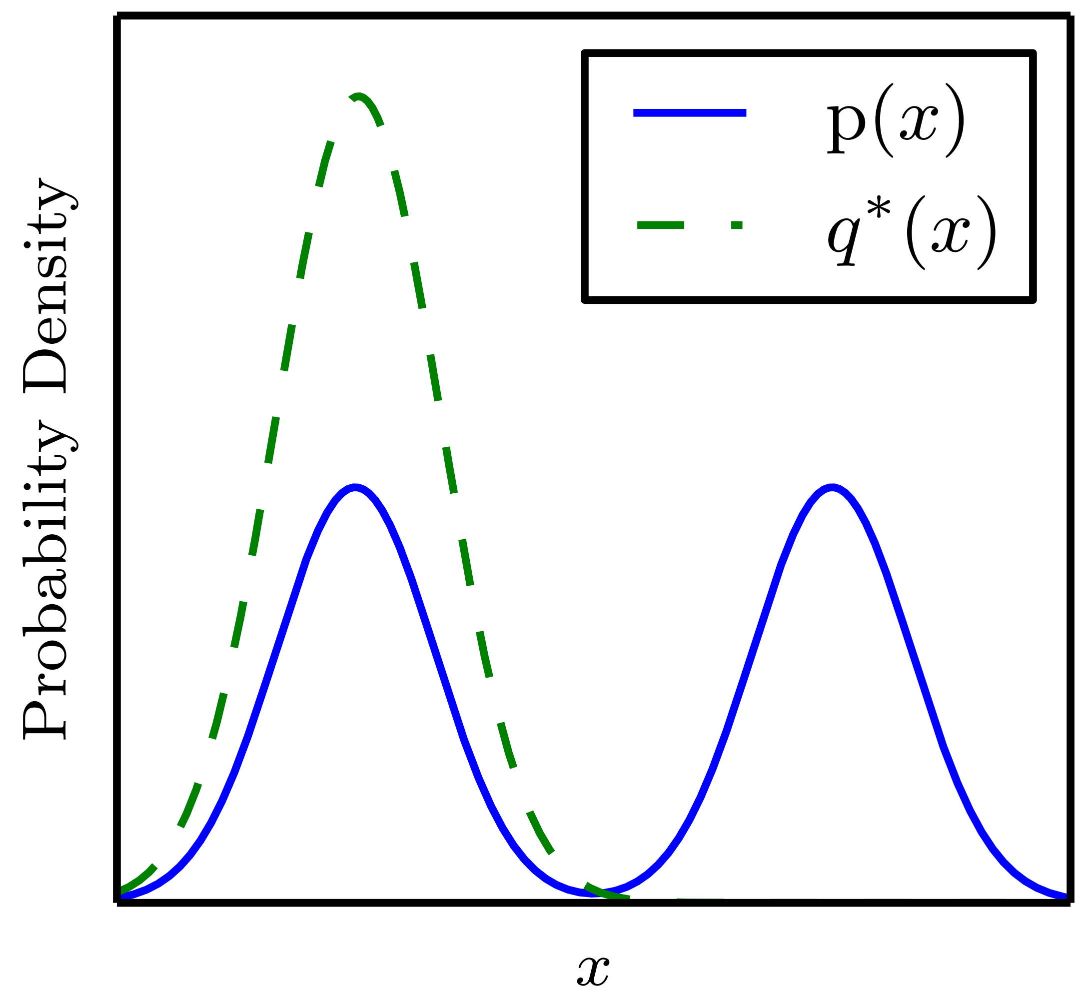
Is the divergence important?

$q^* = \text{argmin}_q D_{\text{KL}}(p \| q)$



Maximum likelihood

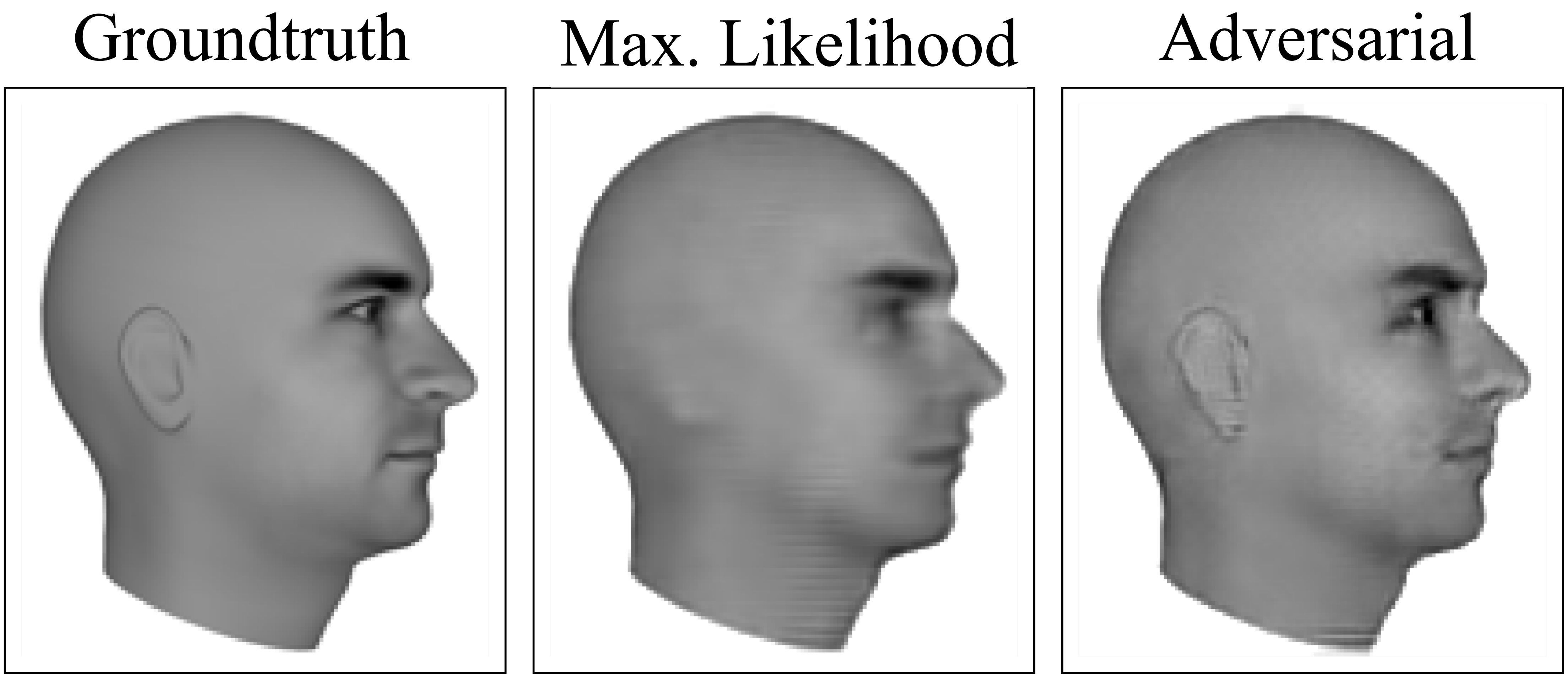
$q^* = \text{argmin}_q D_{\text{KL}}(q \| p)$



Reverse KL

(Goodfellow et al 2016)

Next Video Frame Prediction



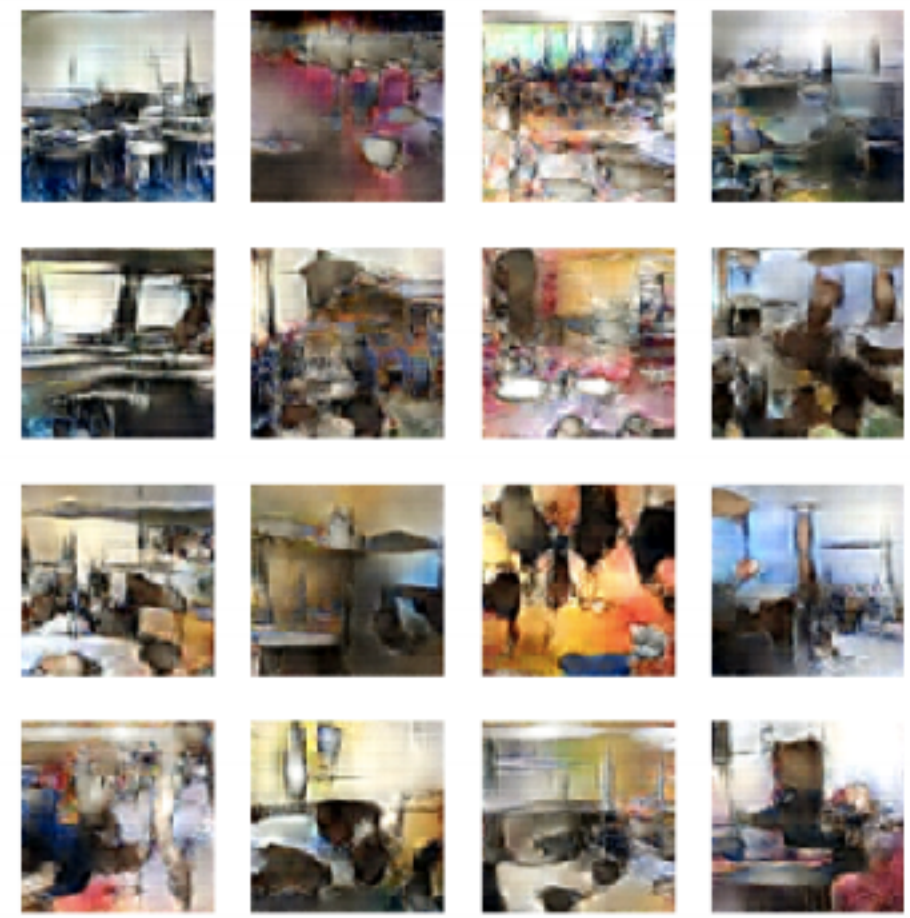
(Lotter et al 2016)

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

Conditional GANs

There is extra conditioning information as input to the generator

image-to-image translation

image-to-image translation

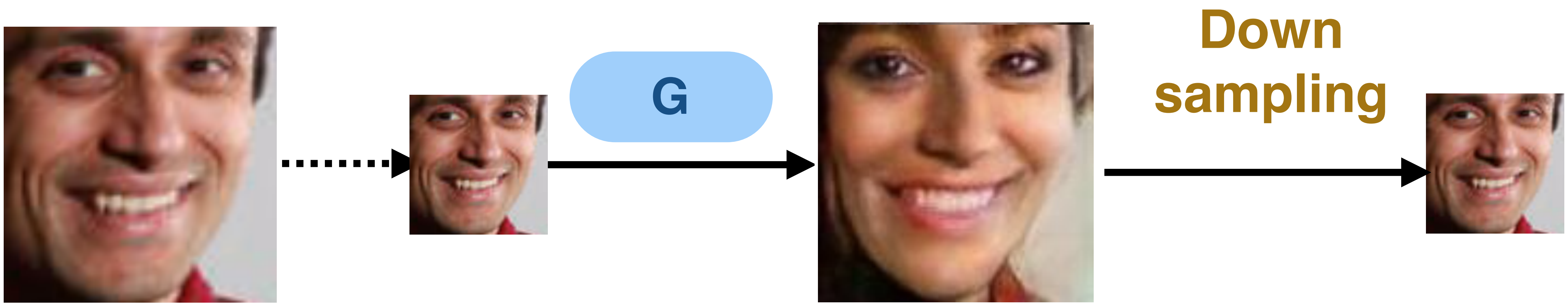


image-to-image translation

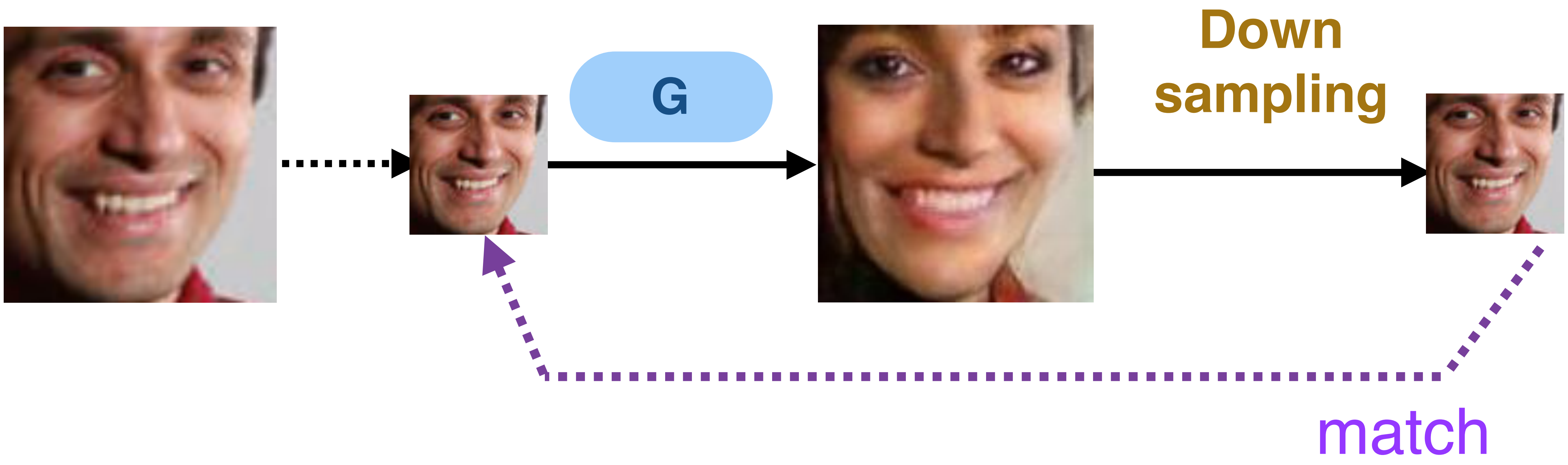
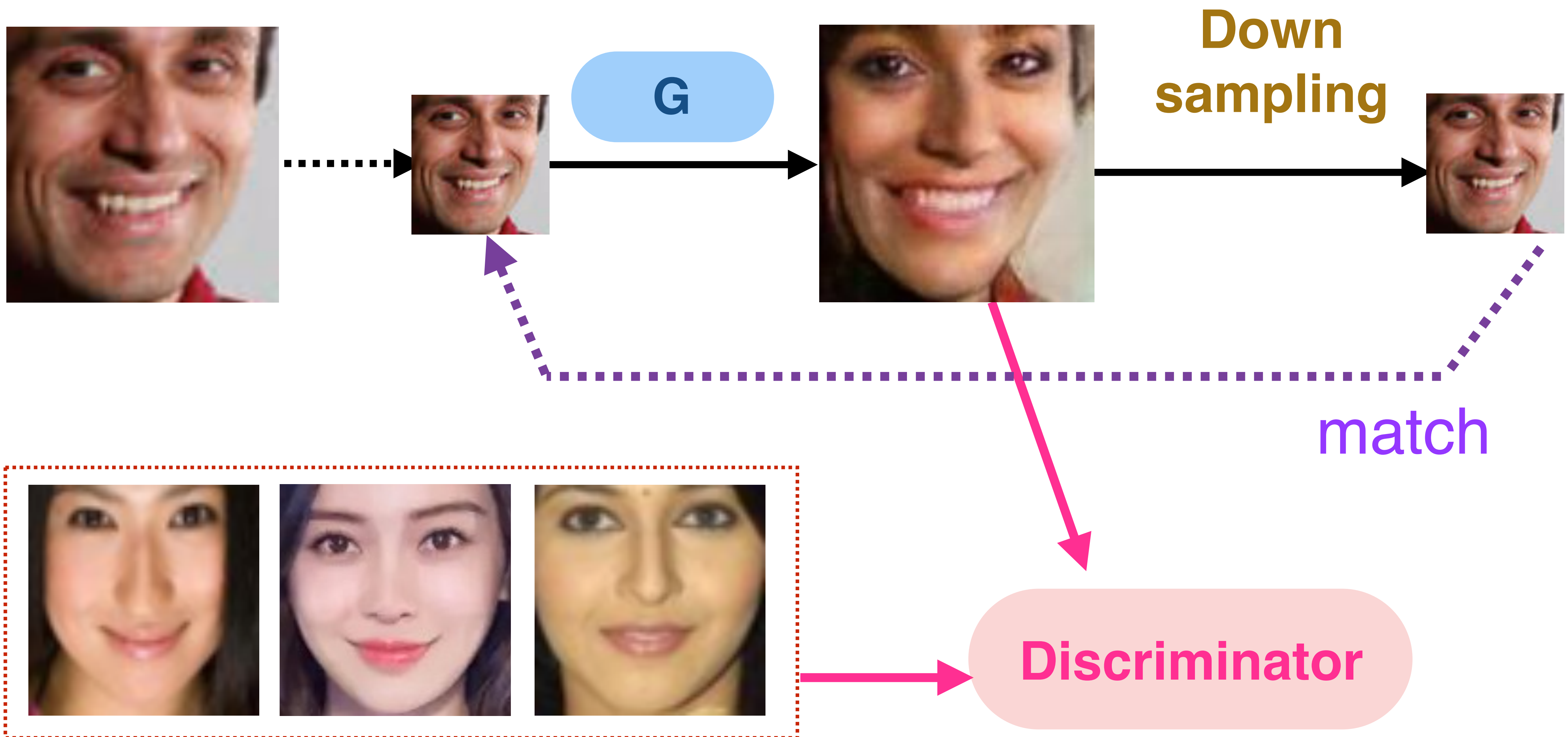
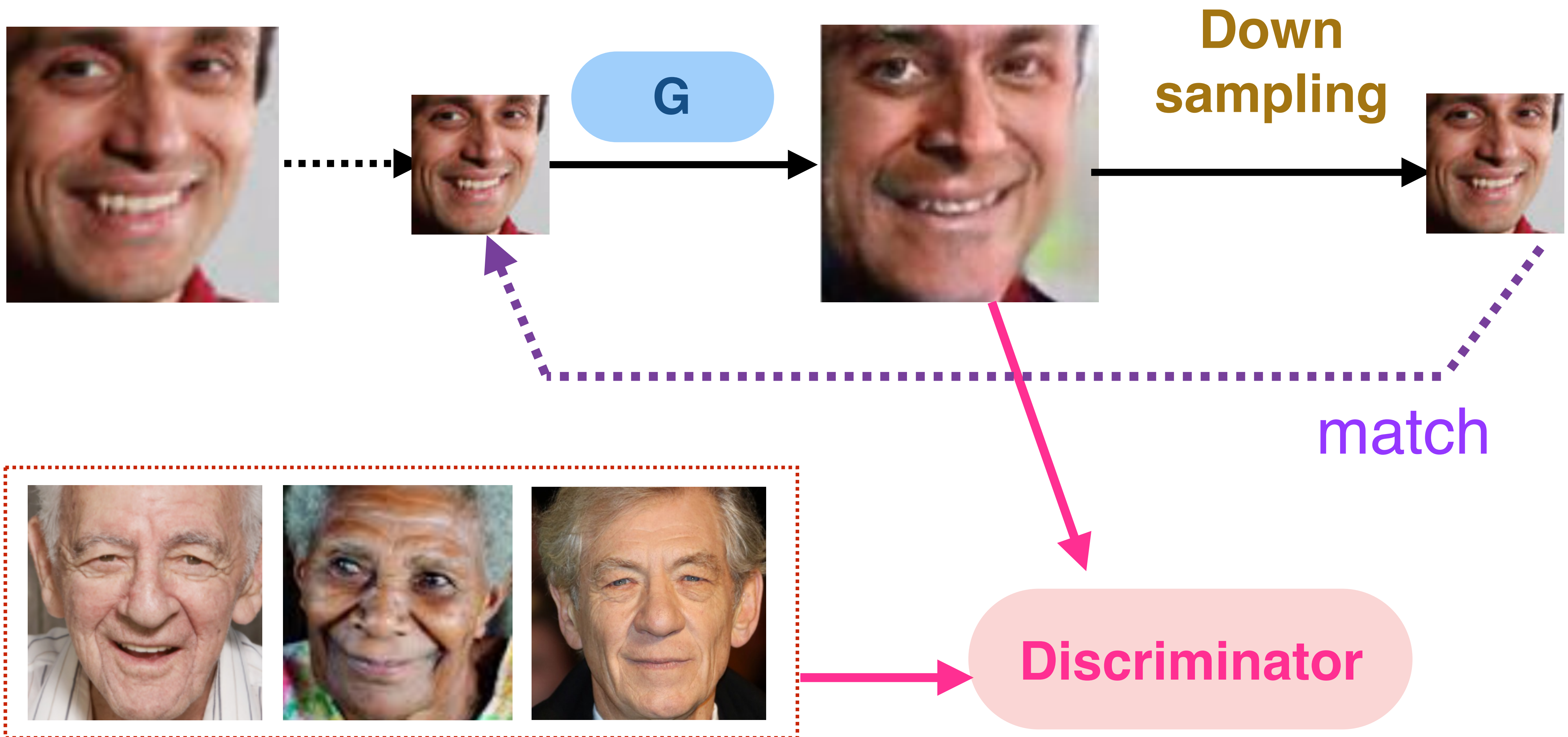


image-to-image translation



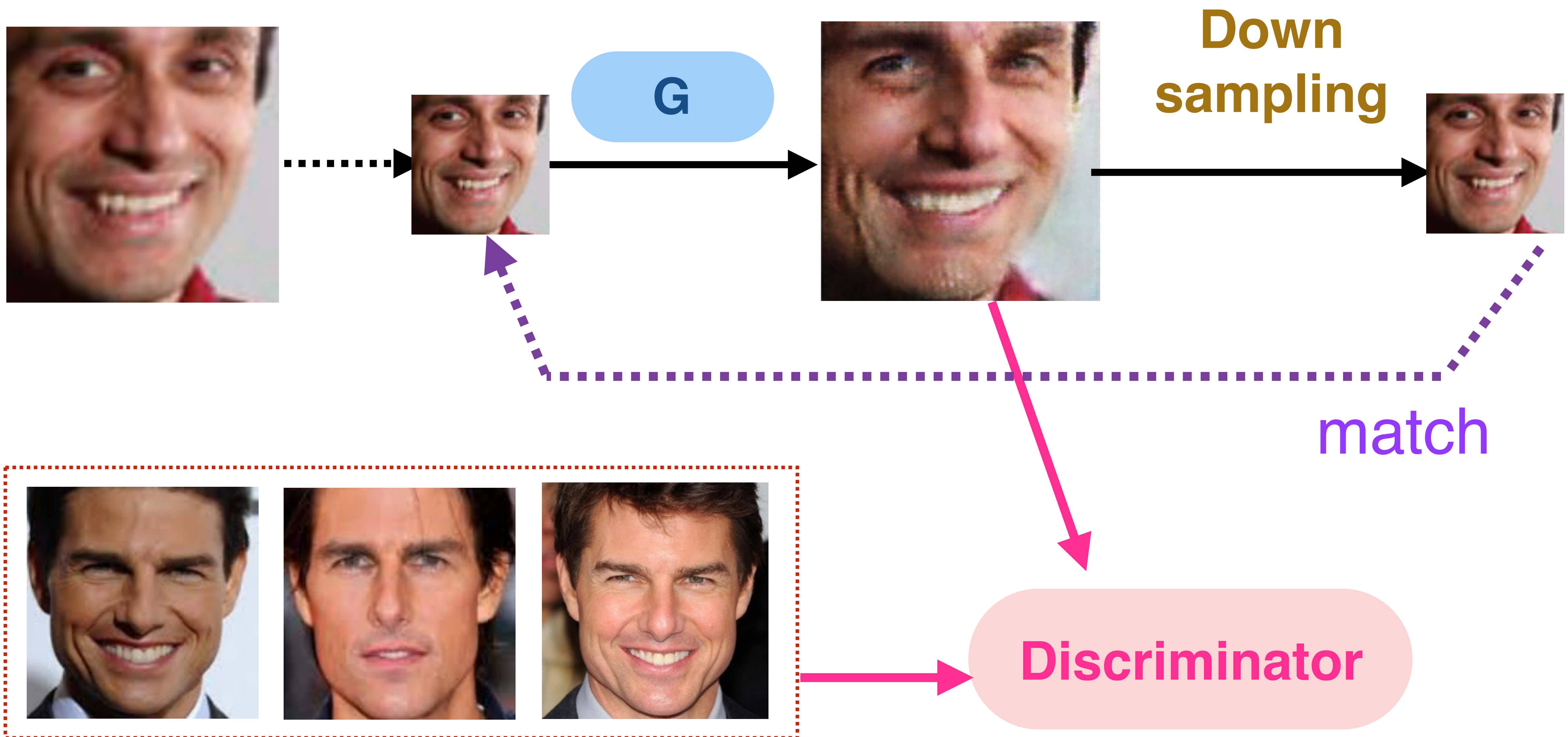
By putting different memory in the repositories, we get different results

image-to-image translation



By putting different memory in the repositories, we get different results

image-to-image translation



By putting different memory in the repositories, we get different results

Adversarial Inverse Graphics Networks

Adversarial Inverse Graphics Networks

Generative Adversarial Imitation learning

Find a policy π_θ that makes it impossible for a discriminator network to distinguish between state-action pairs from the expert demonstations and those produced by the learnt policy π_θ

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

D outputs 1 if states comes from the demo policy

$$\min_{\pi_\theta} \max_D \mathbb{E}^*_\pi[\log D(s)] + \mathbb{E}_{\pi_\theta}[\log(1 - D(s))]$$

Reward for the policy optimization is how well I matched the demo trajectory distribution, else, how well I confused the discriminator: logD(s)

Generative Adversarial Imitation Learning

Jonathan Ho
Stanford University
hoj@cs.stanford.edu

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

NIPS 2016

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

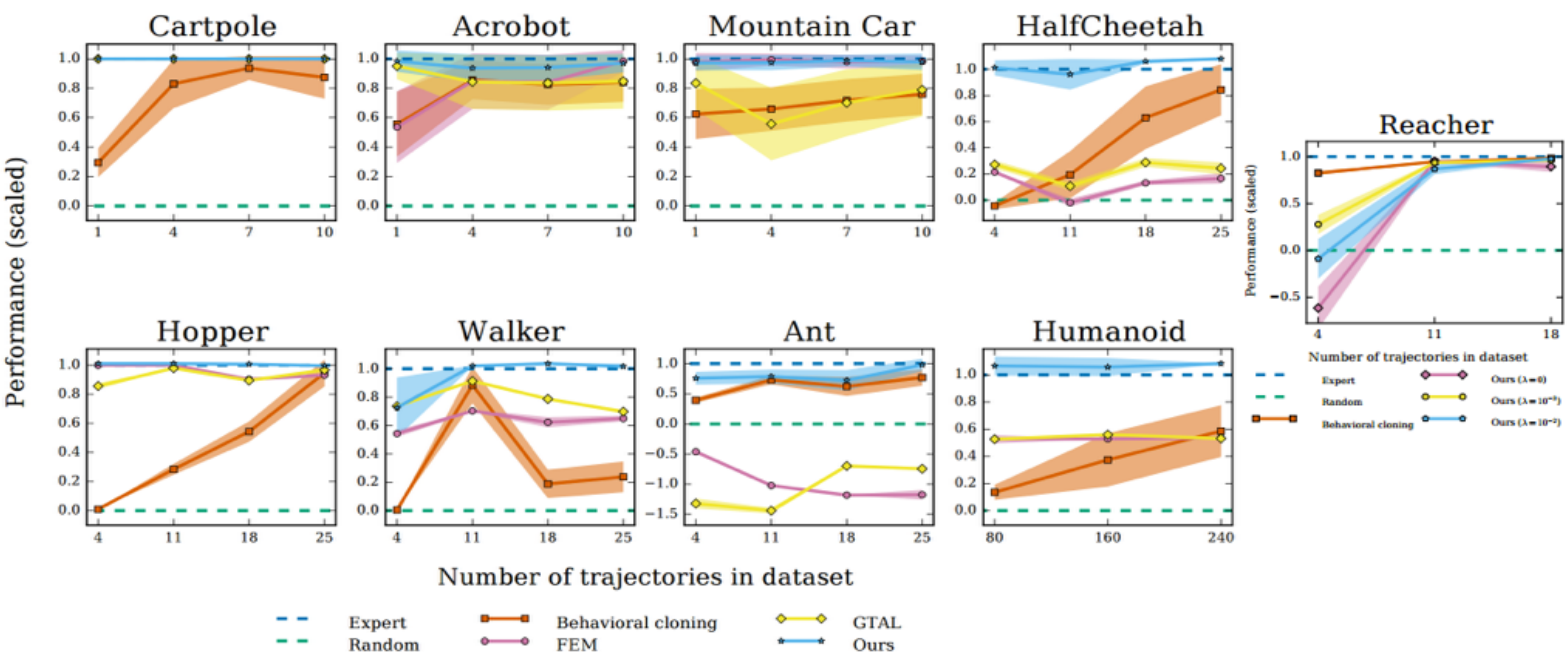
$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \tag{17}$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\begin{aligned} &\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)\right] - \lambda \nabla_\theta H(\pi_\theta), \\ &\text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \,|\, s_0 = \bar{s}, a_0 = \bar{a}] \end{aligned} \tag{18}$$

6: **end for**

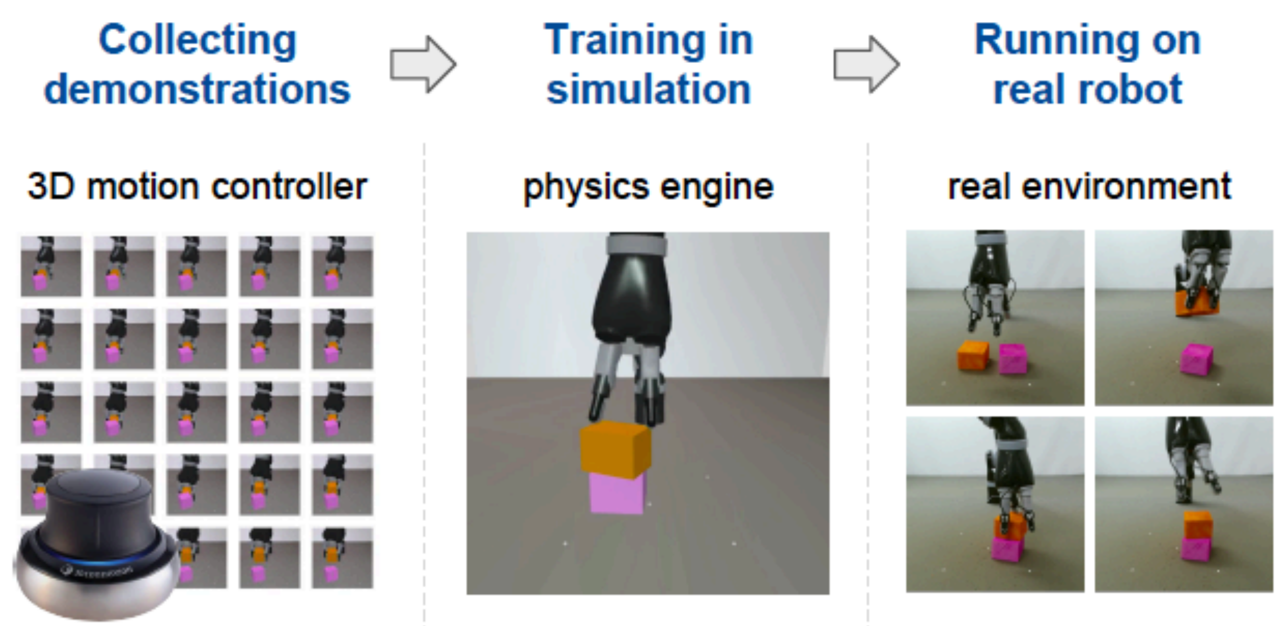
Generative Adversarial Imitation learning



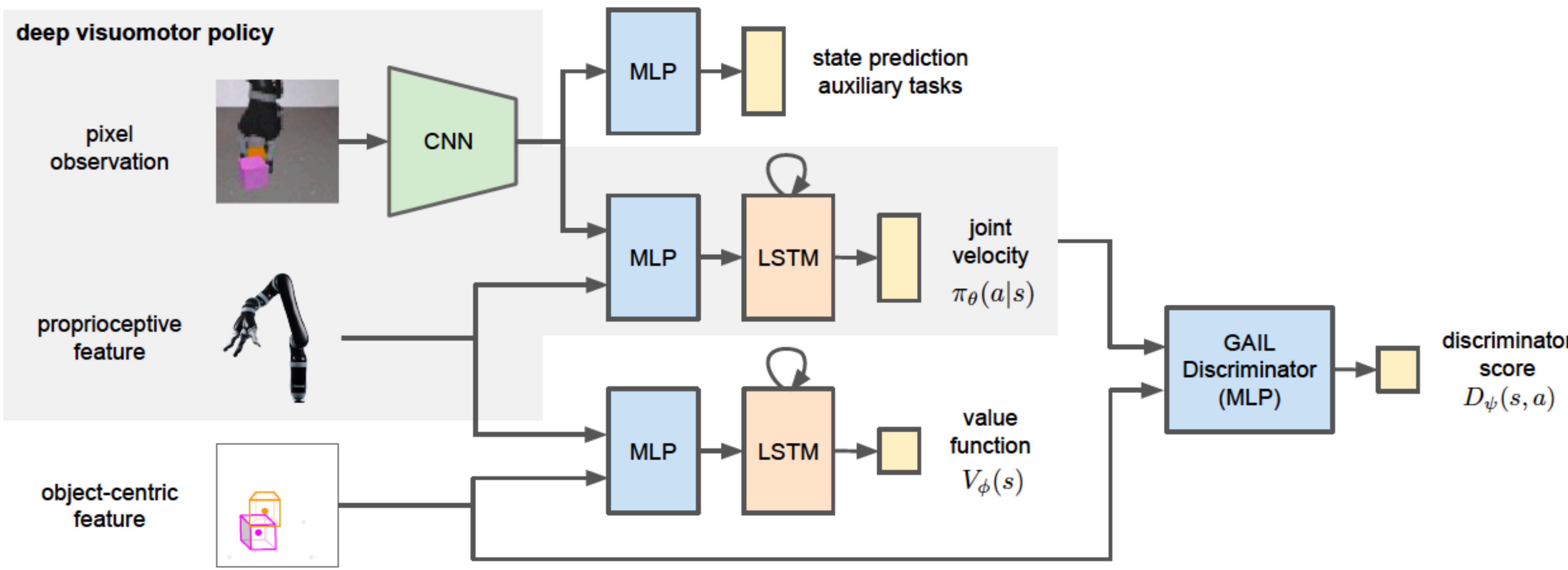
- GAIL performs better but it requires interactions with the environment,
- Behaviour cloning wo DAGGER simply fits expert demonstations
- DAGGER requires both interactive expert and interactions with the environment

Reinforcement and Imitation Learning for Diverse Visuomotor Skills

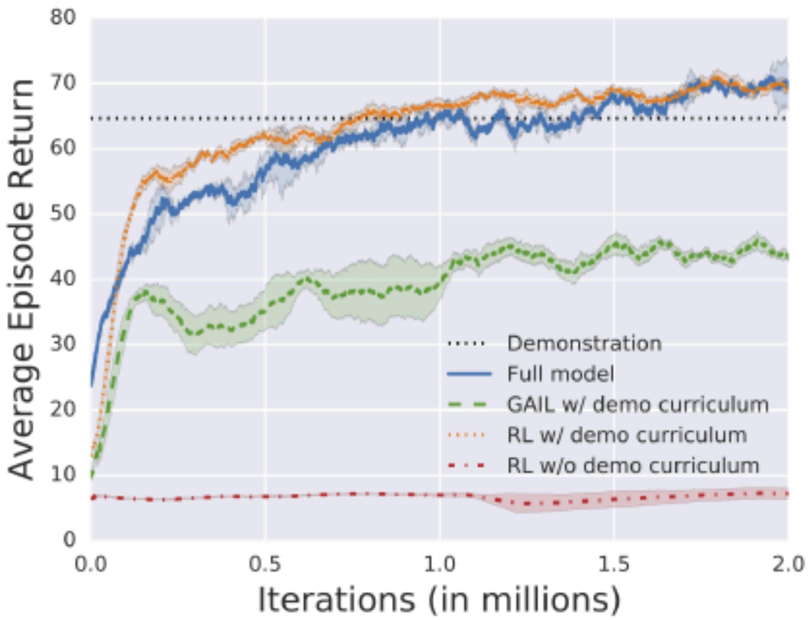
Yuke Zhu[†] Ziyu Wang[‡] Josh Merel[‡] Andrei Rusu[‡] Tom Erez[‡] Serkan Cabi[‡]
Saran Tunyasuvunakool[‡] János Kramár[‡] Raia Hadsell[‡] Nando de Freitas[‡] Nicolas Heess[‡]
[†]Computer Science Department, Stanford University, USA
[‡]DeepMind, London, UK



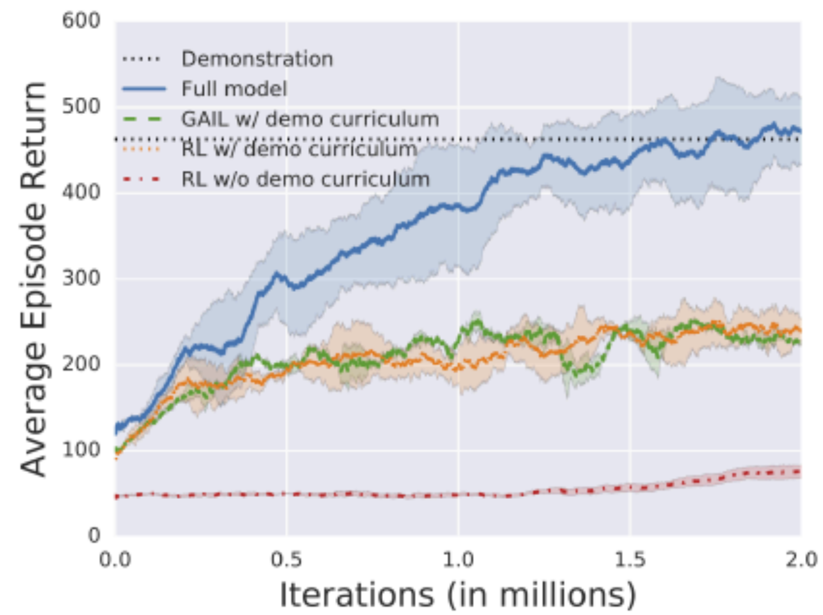
- Combining learning from demonstations with RL from sparse extrinsic rewards
- Used adersarial rewards, where state features were supplied to the discriminator
- Used state information for training the critic, while the actor(policy) was trained directly from pixels
- Varies appearance and dynamics to permit sim2real tranfer
- Auxiliary tasks to help train visual features for the policy net



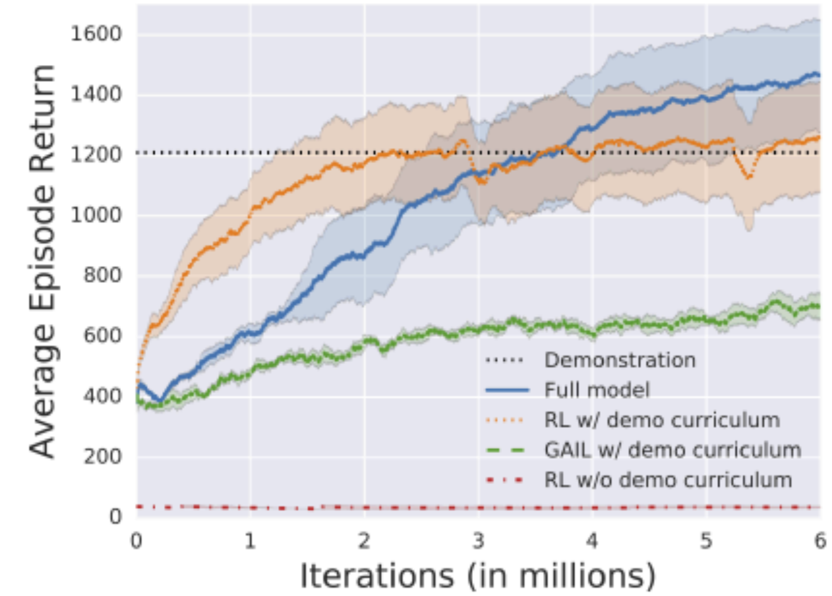
- GAIL on state object-centric features: including actions of the robot deteriorated policy learning!



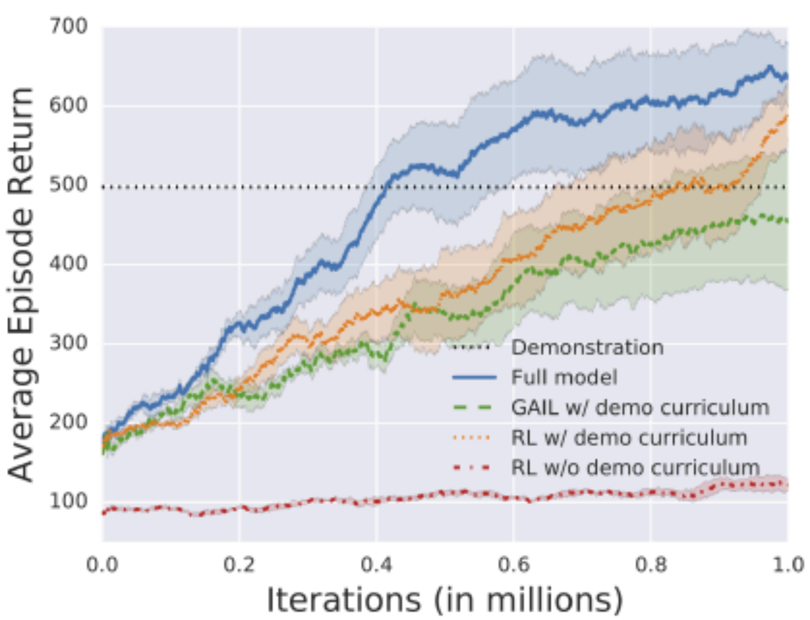
(a) Block lifting



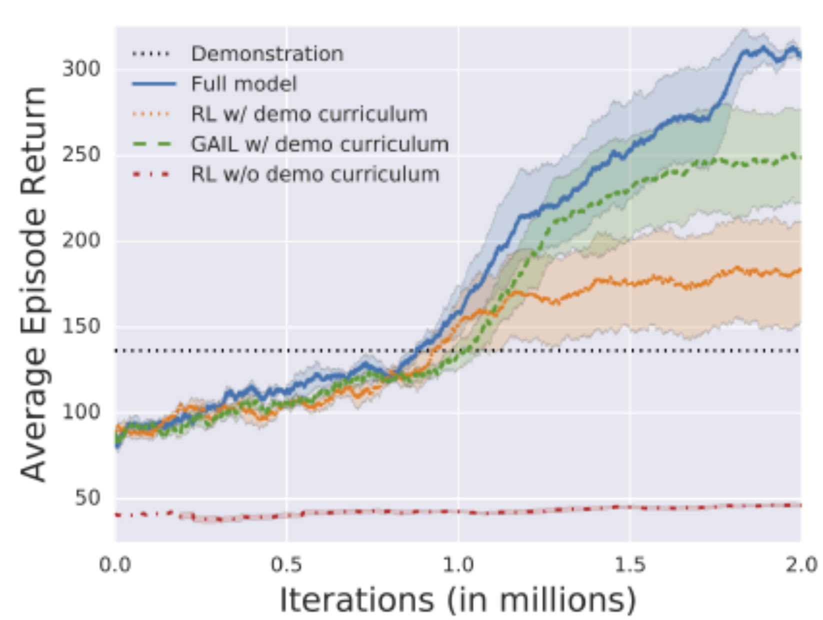
(b) Block stacking



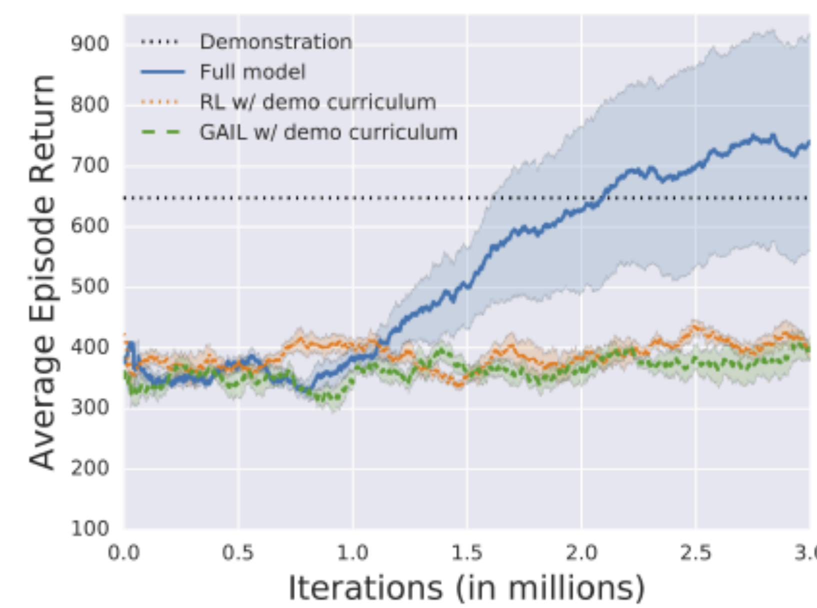
(c) Clearing table with blocks



(d) Clearing table with a box



(e) Pouring liquid



(f) Order fulfillment

Fig. 4: Learning efficiency of our reinforcement and imitation model against baselines. The plots are averaged over 5 runs with different random seeds. All the policies use the same network architecture and the same hyperparameters (except λ).