Carnegie Mellon
School of Computer Science

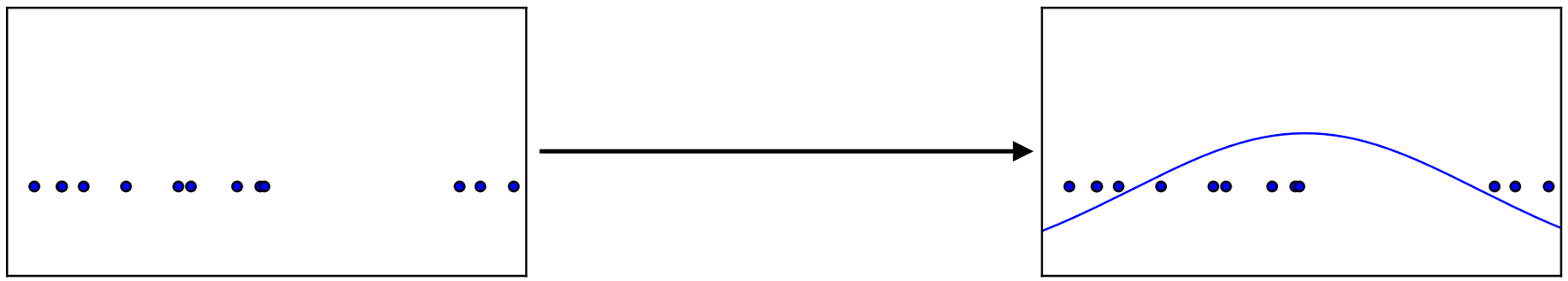
Deep Reinforcement Learning and Control

Adversarial imitation learning

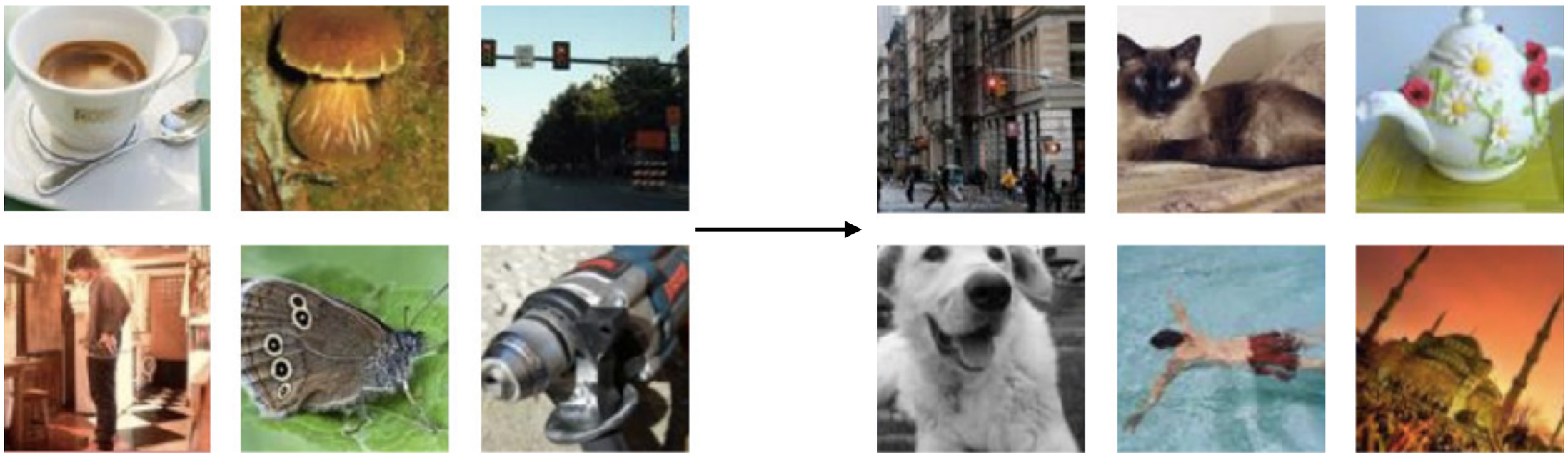
Katerina Fragkiadaki

Generative modeling

- Density estimation



- Sample generation

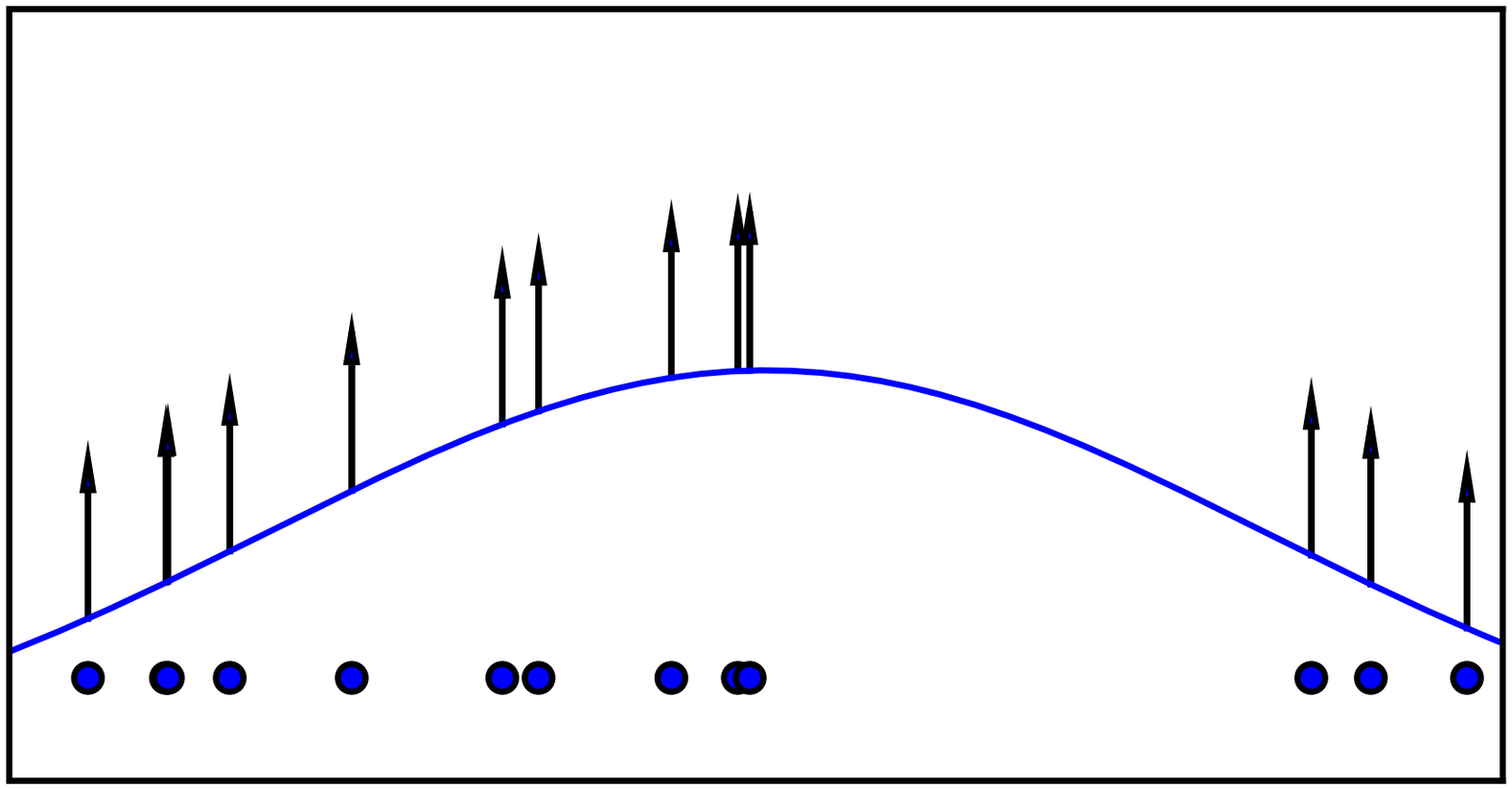


Training examples

Model samples

(Goodfellow 2016)

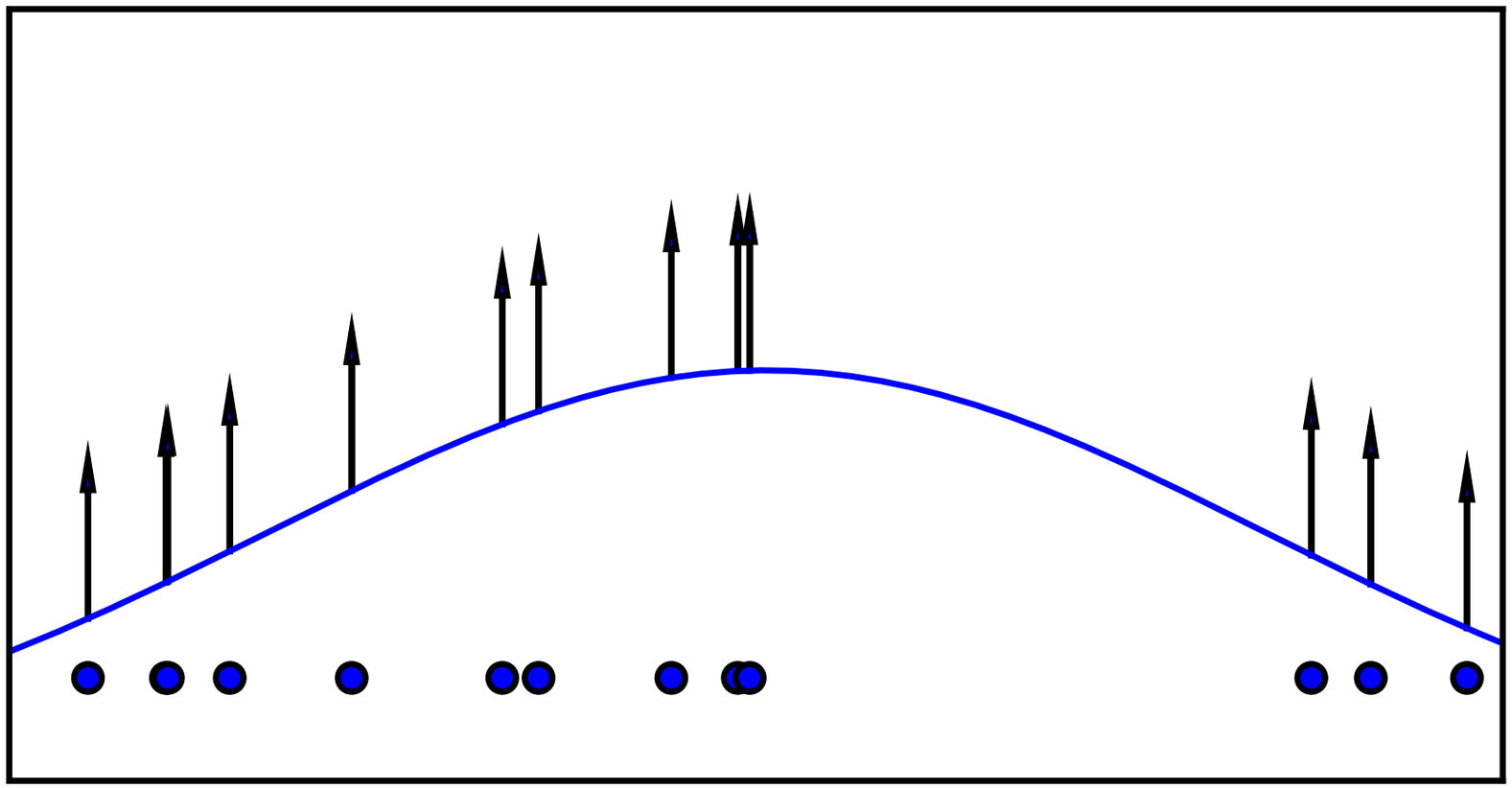
Maximum Likelihood



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

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

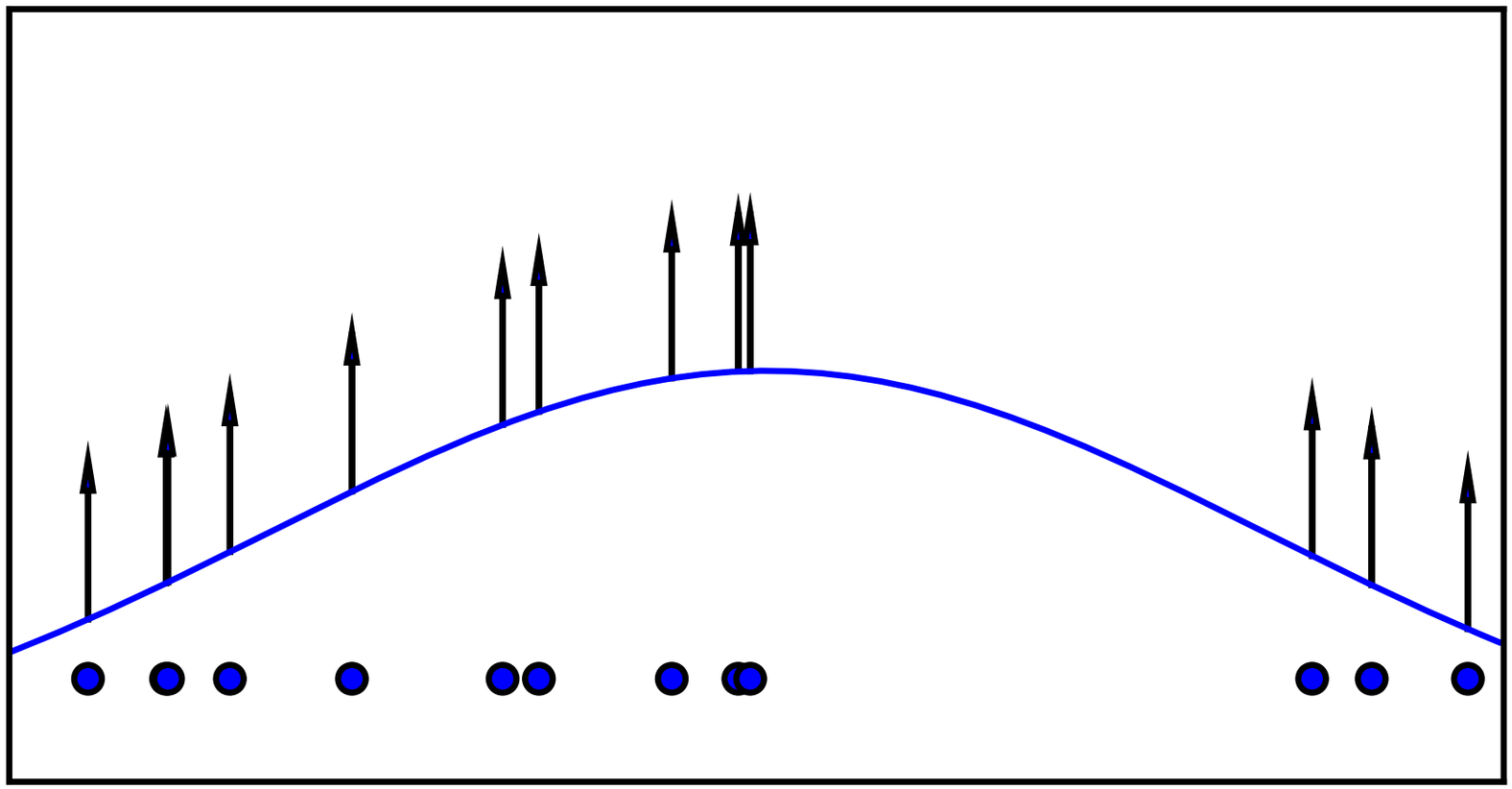
Maximum Likelihood



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

explicit density

Maximum Conditional Likelihood



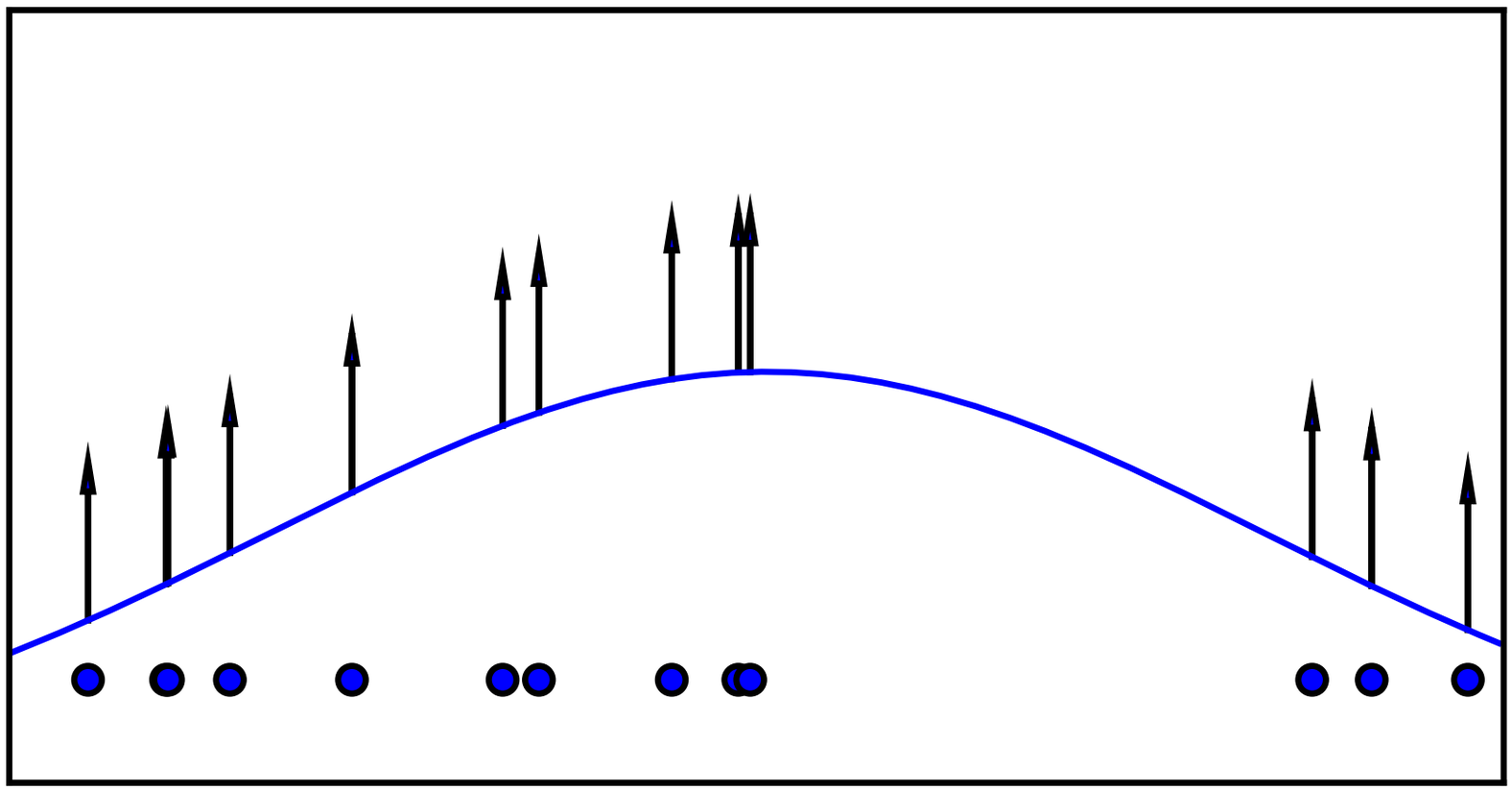
$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta, c)$$

explicit density

extra conditioning information

Maximum Conditional Likelihood

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

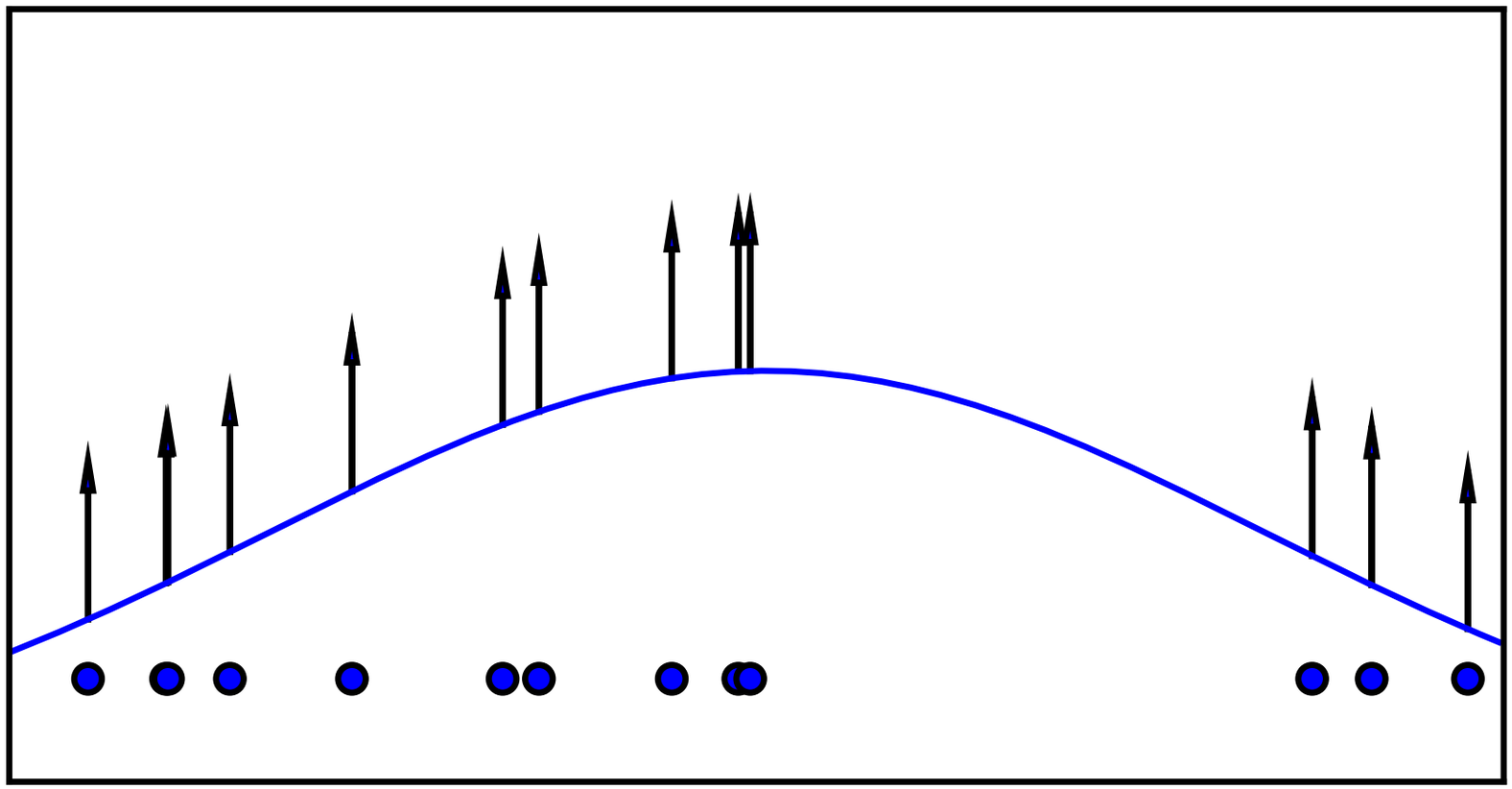


$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x}\,|\,\theta, c)$$

equiv. to

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

Maximum likelihood for imitation

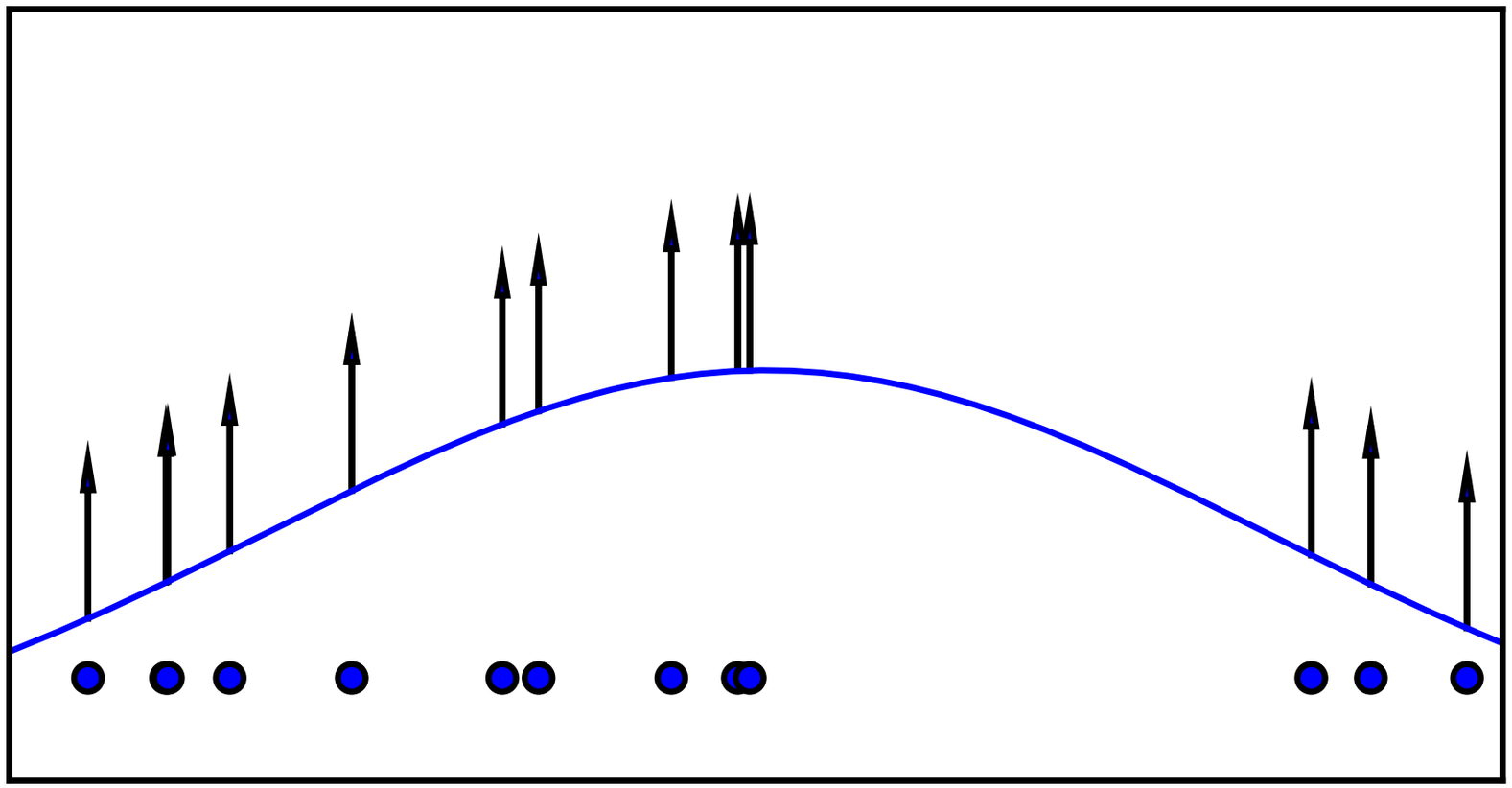


$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\pi}(\mathbf{a}_t \,|\, \theta, \mathbf{s}_t)$$

explicit density

extra conditioning information

Maximum Likelihood-Gaussian with fixed covariance

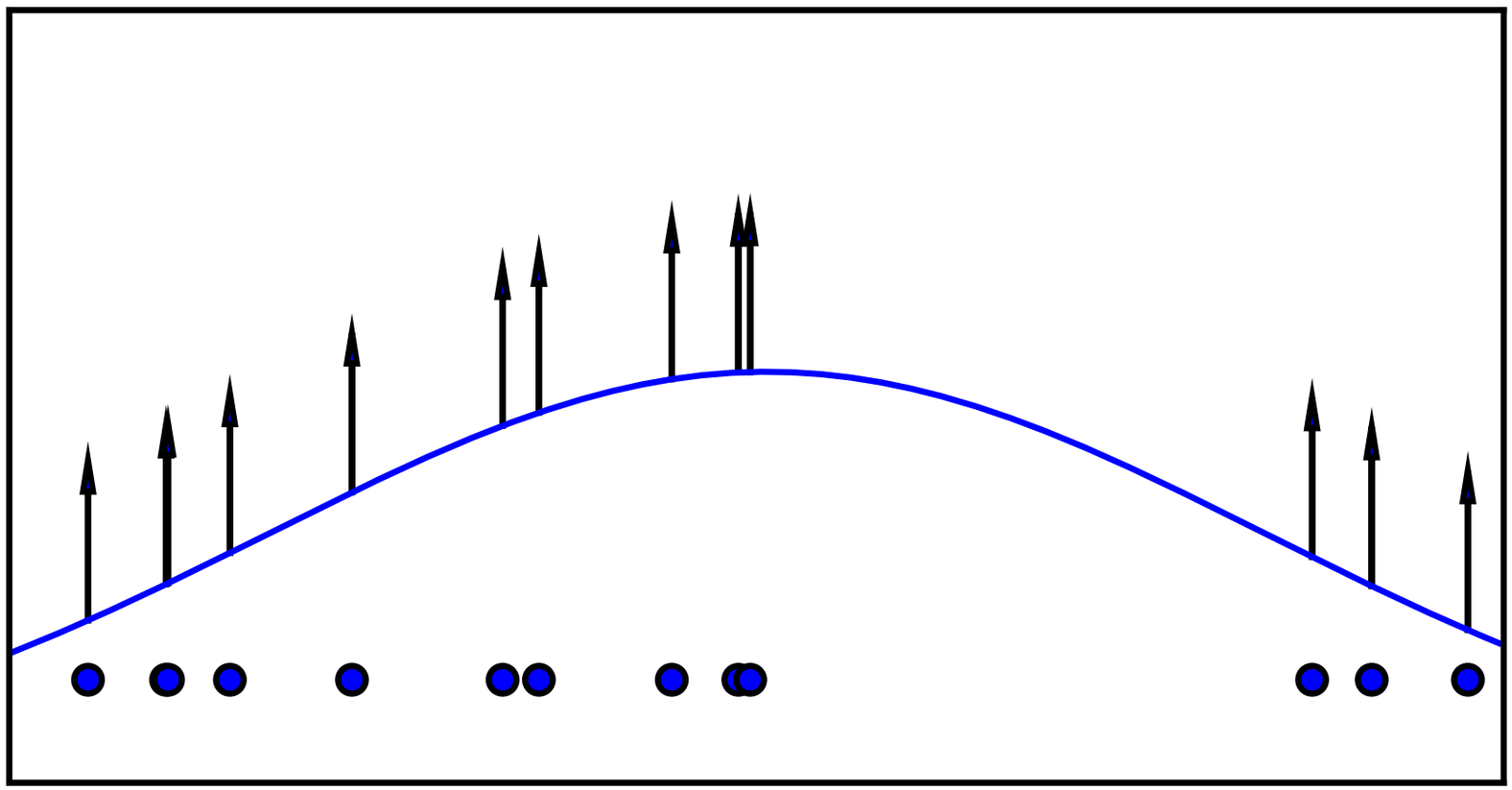


$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x} \mid \theta, c)$$

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

Maximum Likelihood-Gaussian with fixed covariance

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



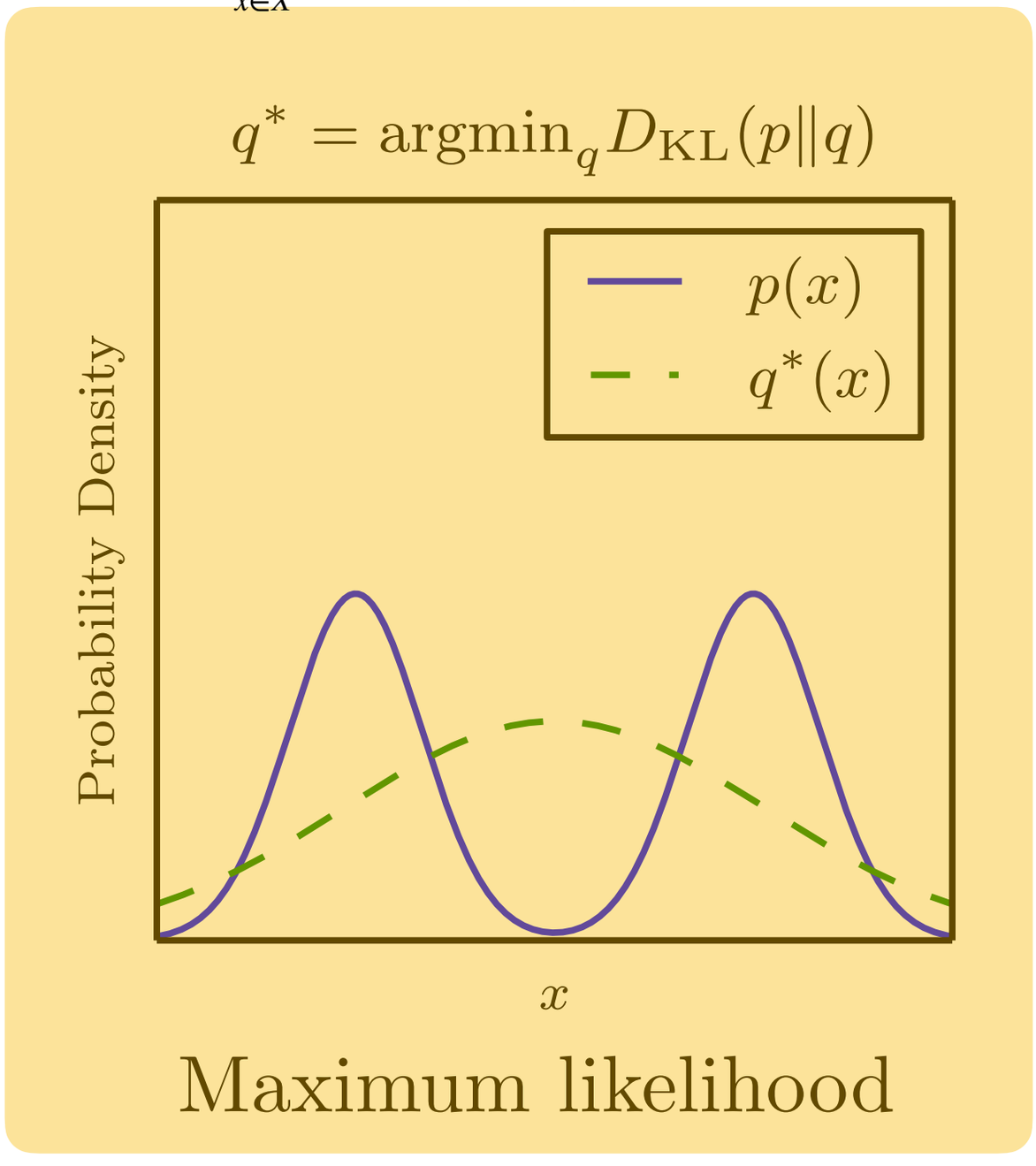
$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x}\,|\,\theta, c)$$

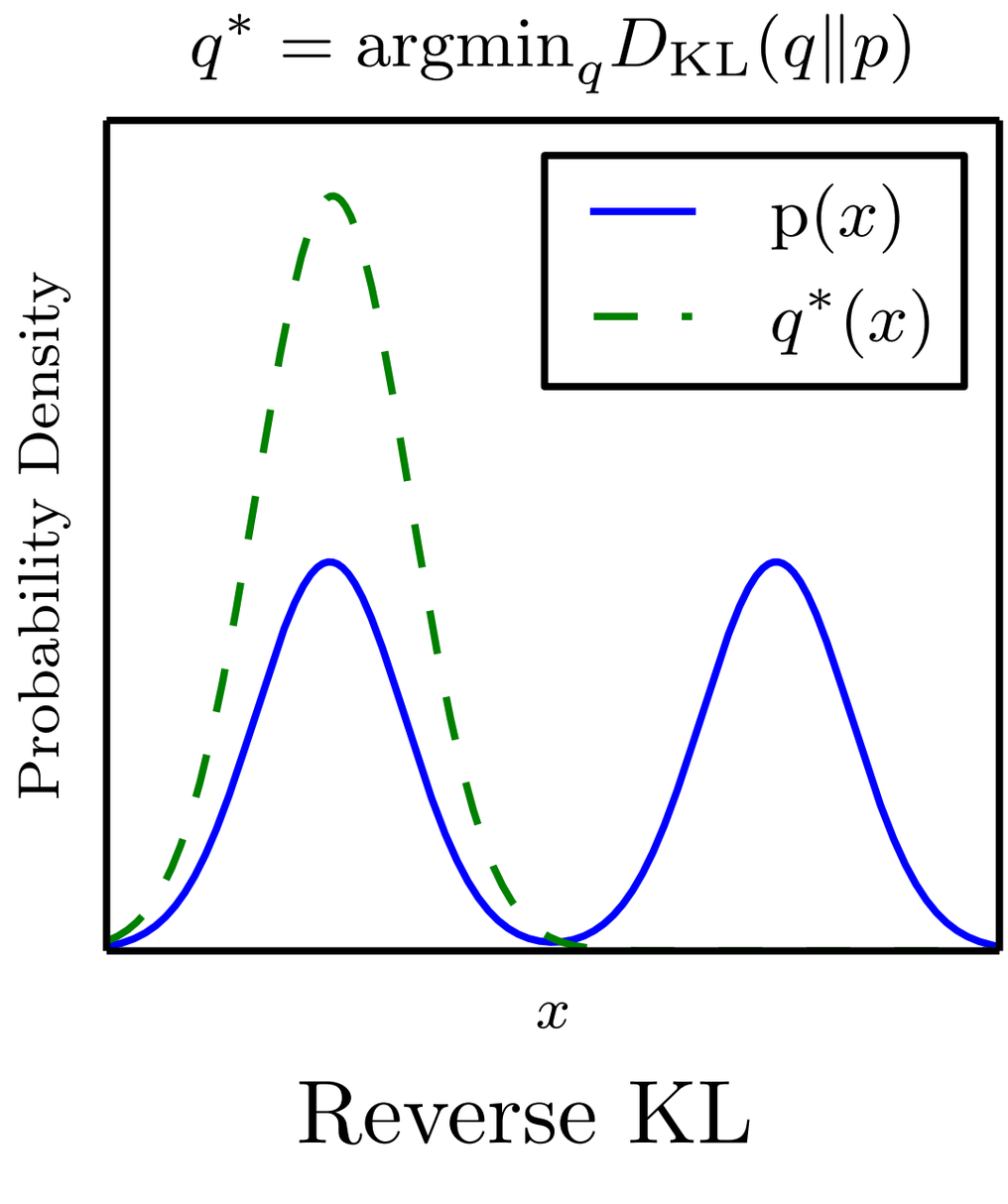
$\max_{\theta} \mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(\mathbf{x}\,|\,\theta, c)$ equiv. to $\min_{\theta} \mathbb{E}_{x\sim p_{\text{data}}} \|\mathbf{x}-\mu(\theta,\mathbf{c})\|_2^2$

e.g. behavior cloning with continuous actions

Behaviour cloning

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

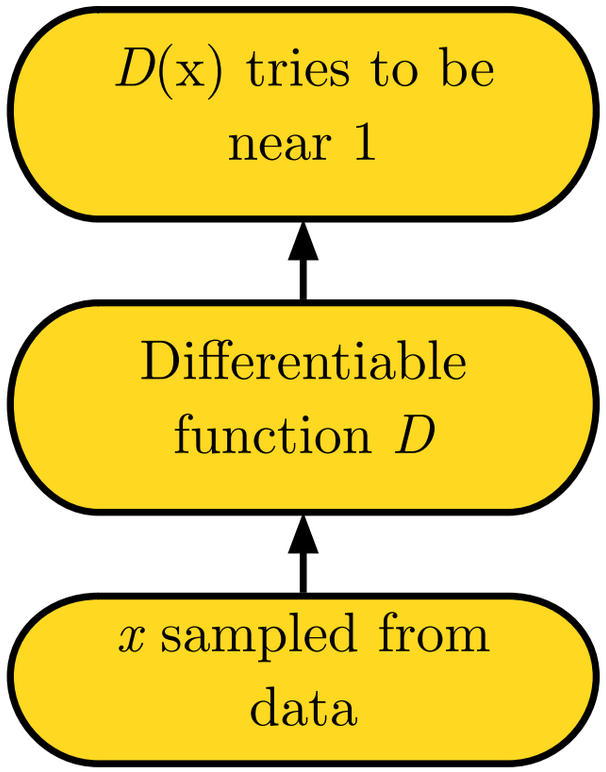


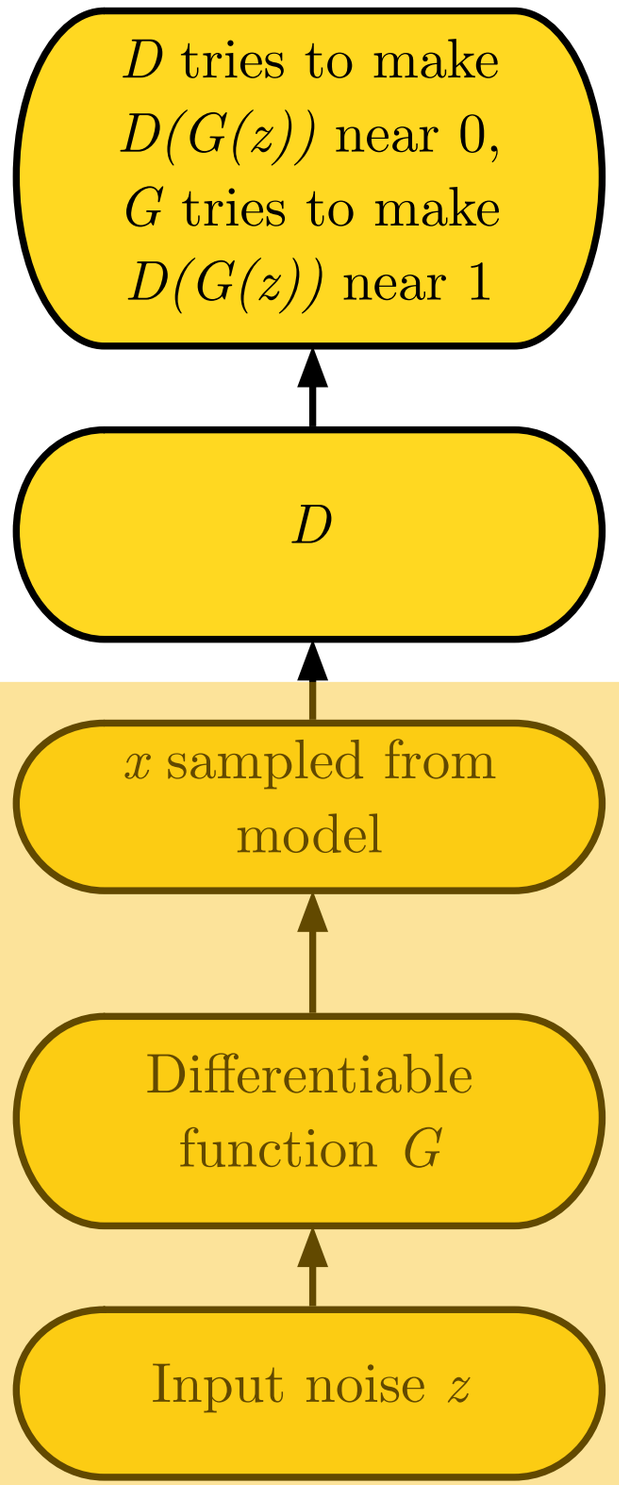


Non realistic action samples, due to non expressive policy (plain regressor)

GANs to the rescue

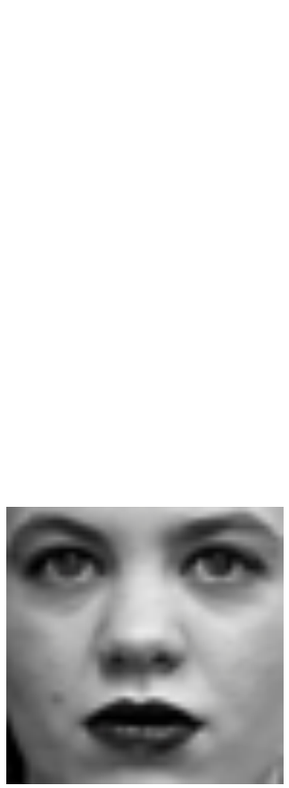
Adversarial Nets Framework

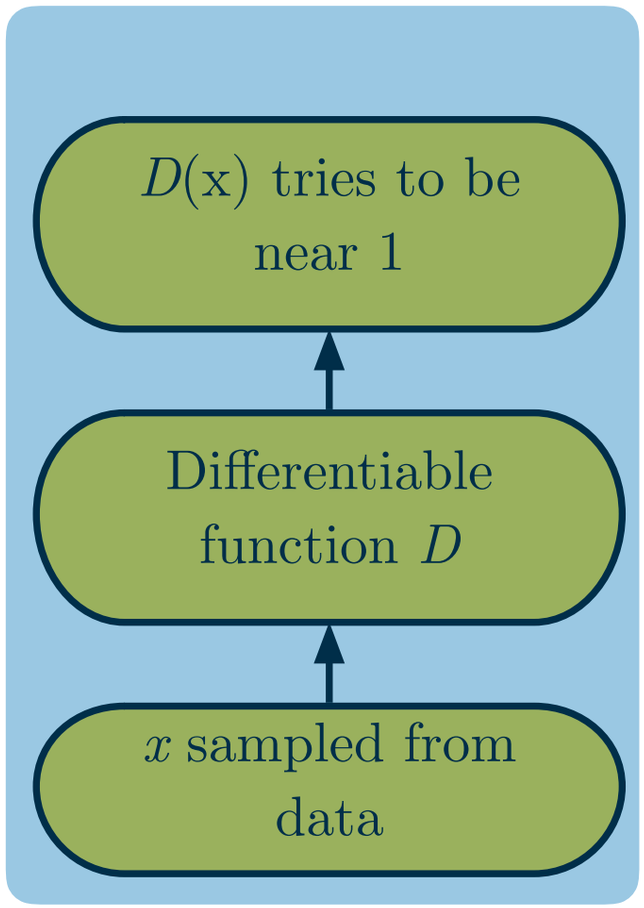


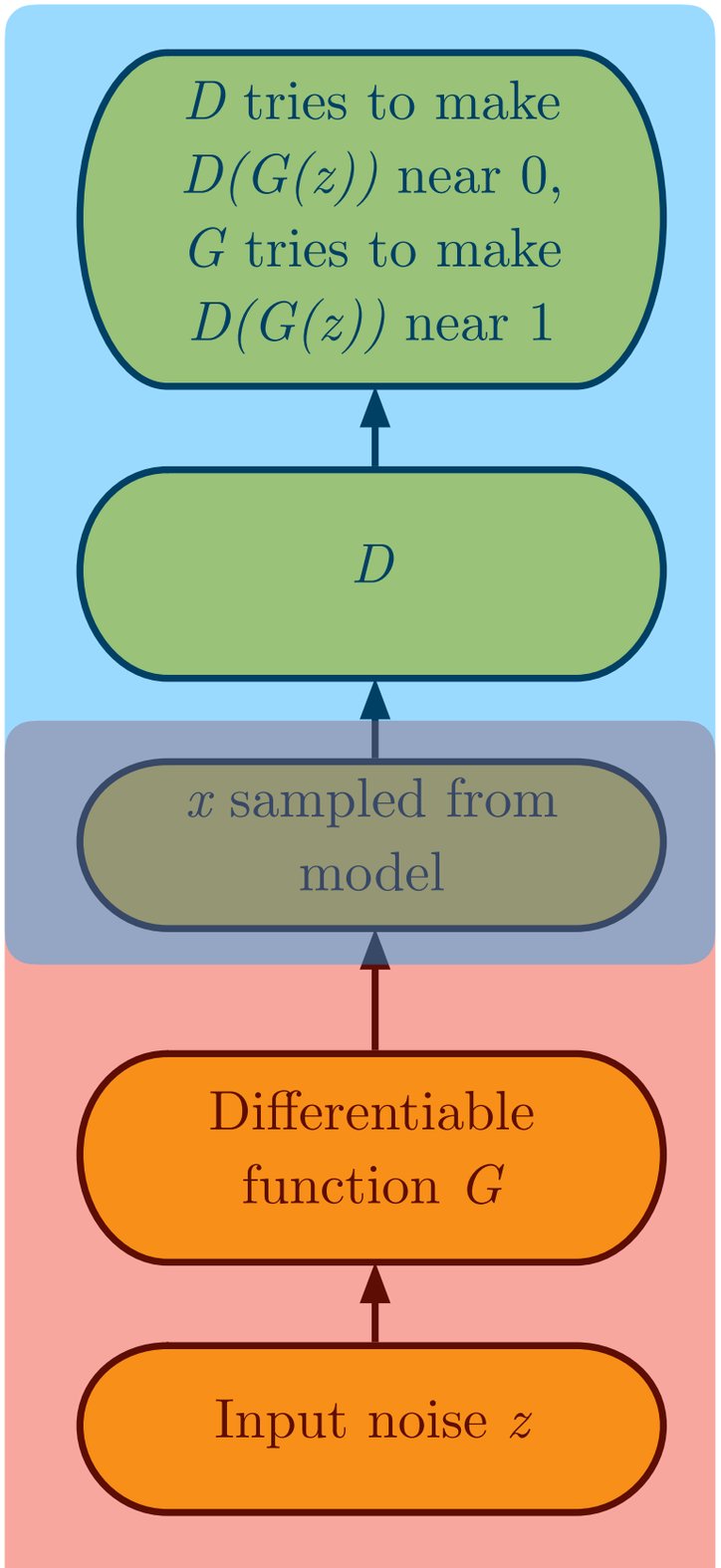


Generator G

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





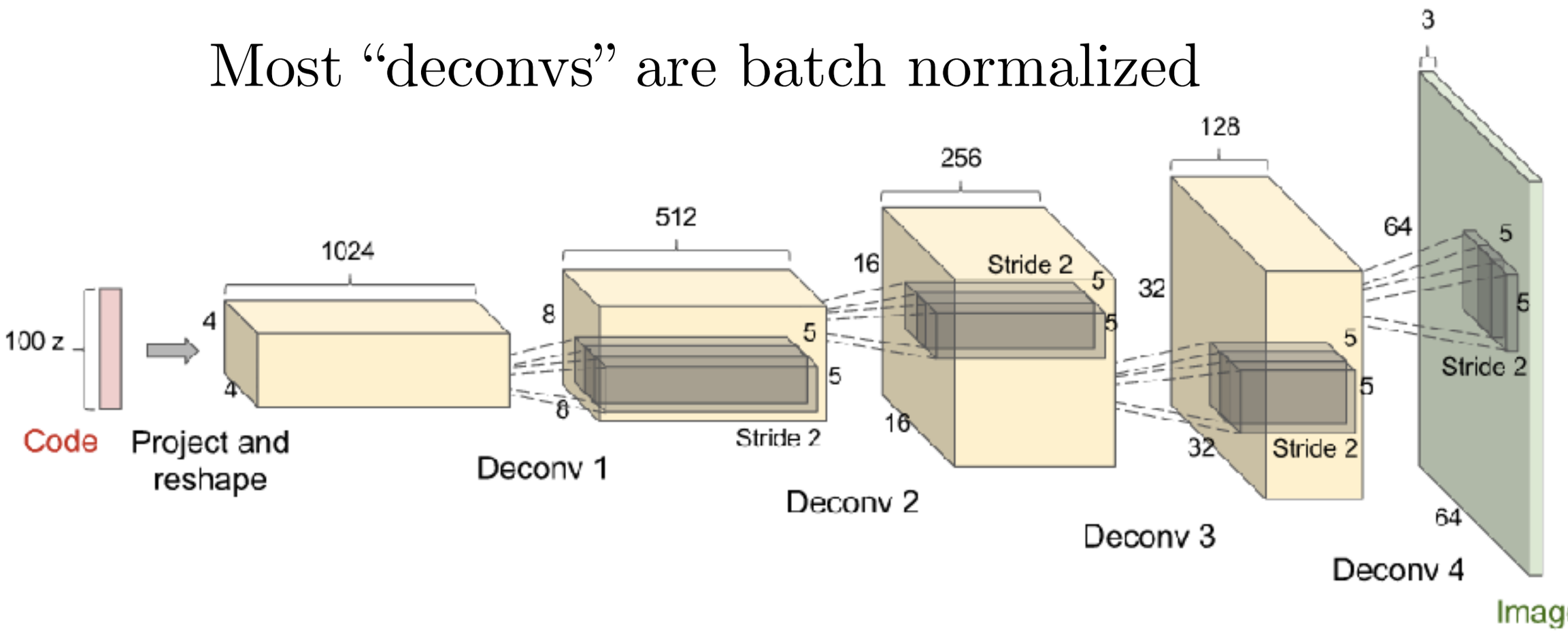


Discriminator D

Generator G

A Generator network (DCGAN)

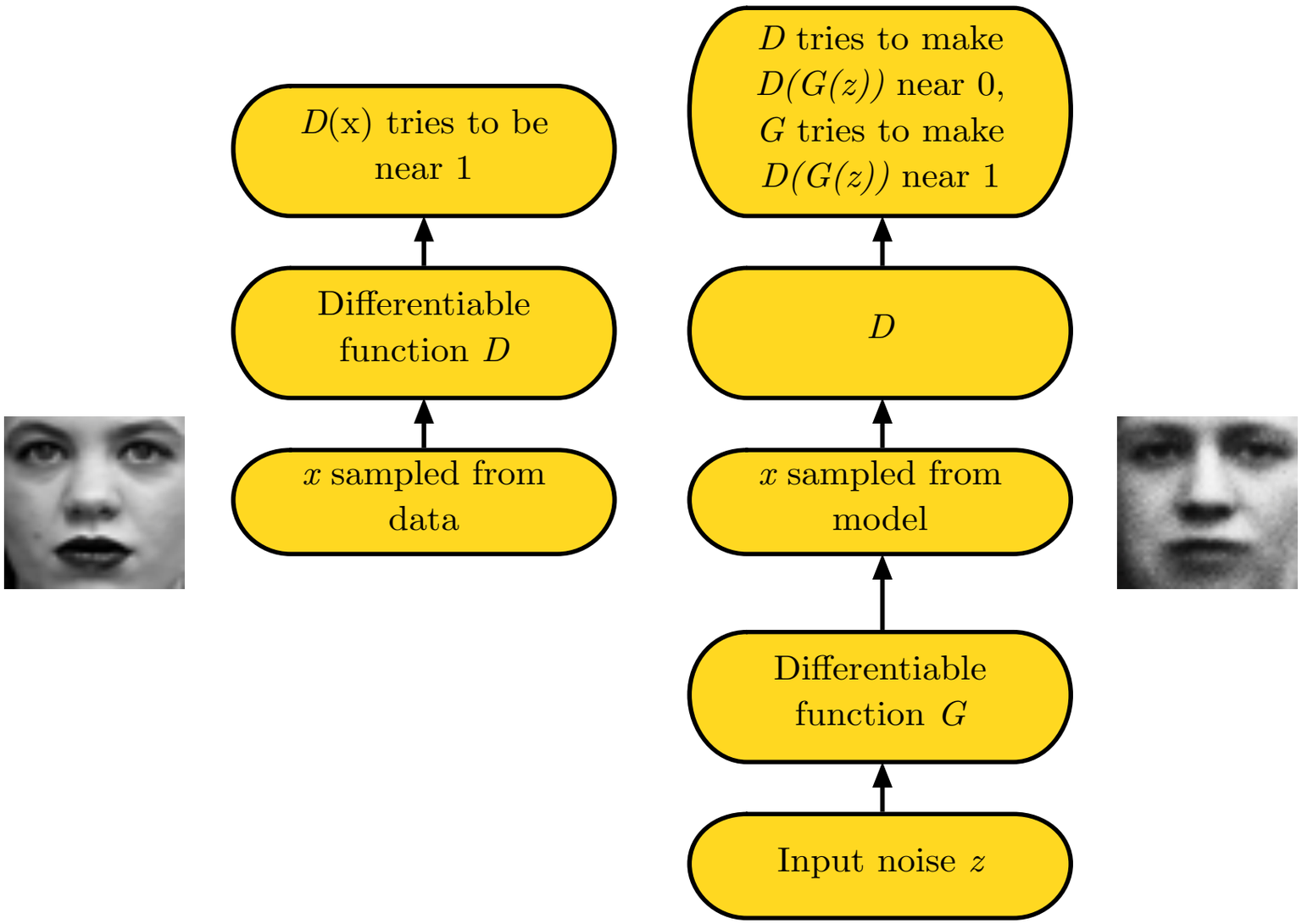
Most "deconvs" are batch normalized



(Radford et al 2015)

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)

Questions:

- What if the generator maps all noise vectors to a single super photorealistic image?
- What if we train the discriminator till convergence (it is just a supervised classifier…) and becomes perfect in distinguishing real from generated images?

A minimax game

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

Optimal discriminator strategy

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

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

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

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

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

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

Optimal discriminator strategy

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

The discriminator assigns values D(x) to each image x. Let's take the derivative to see where the optimum is attained.

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

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

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

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

Optimal discriminator strategy

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

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

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

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

Optimal discriminator strategy

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

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

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

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

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

Optimal discriminator strategy

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

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

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

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

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

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

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

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

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

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

Optimal generator strategy

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

Optimal generator strategy

$$
\begin{aligned}
C(G) &= \max_{D} V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}(x)}[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}] + \mathbb{E}_{x \sim p_G(x)}[\log(\frac{p_{\text{G}}(x)}{p_{\text{data}}(x) + p_G(x)})] \\
&= 2\mathrm{D}_{\mathrm{JSD}}\left(p_{data}(x) \,||\, p_G(x)\right) - \log 4
\end{aligned}
$$

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

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

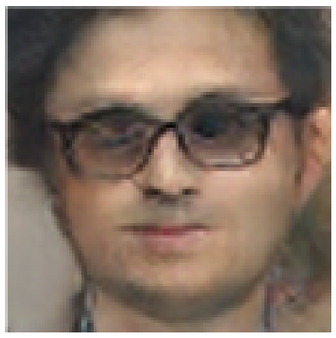
$$
C(G) = \mathbb{E}_{x \sim \mathrm{p}_{\text{data}}(x)} \log \frac{1}{2} + \mathbb{E}_{x \sim p_G(x)} \log \frac{1}{2} = -\log 4
$$

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

DCGANs for LSUN Bedrooms

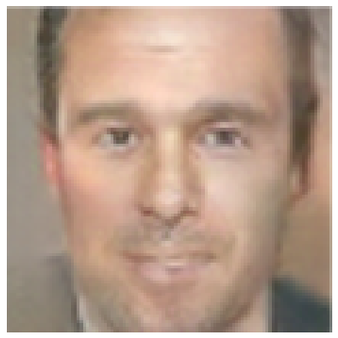
(Radford et al 2015)

Vector Space Arithmetic



Man with glasses

-



Man

+

Woman

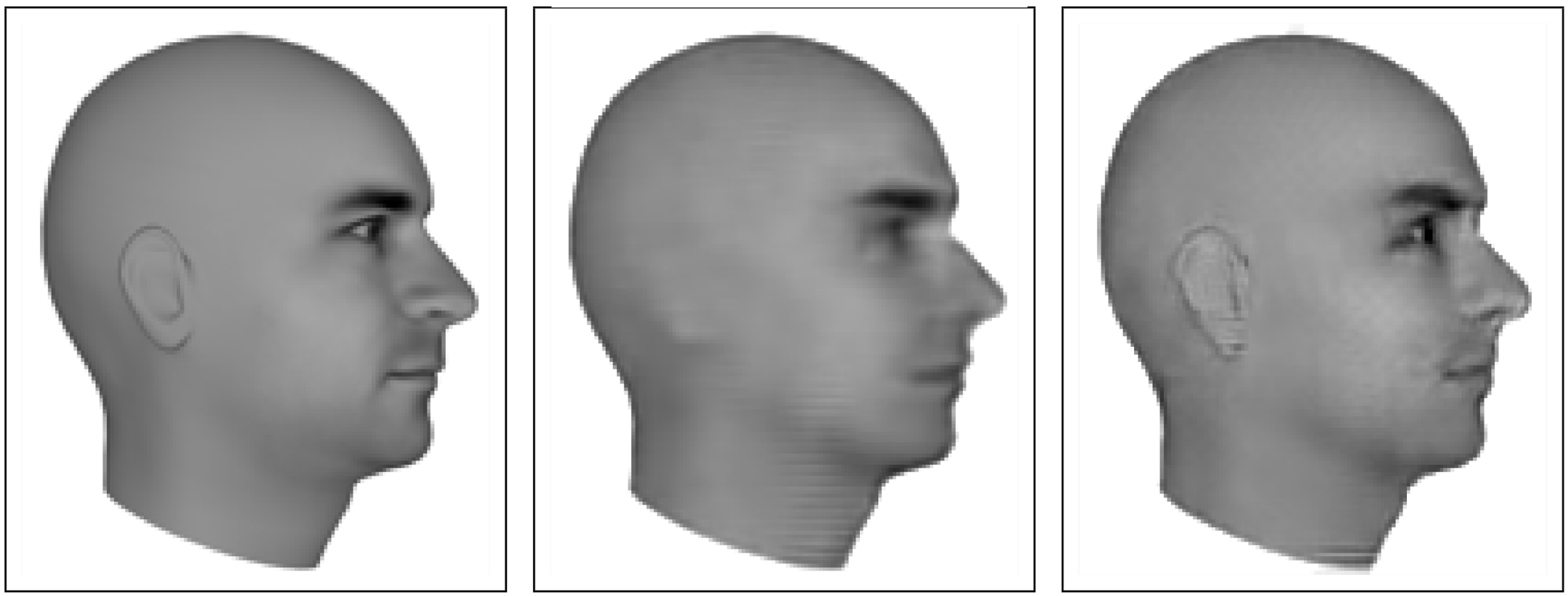
=

Woman with Glasses

(Radford et al, 2015)

Next Video Frame Prediction

Groundtruth | Max. Likelihood | Adversarial



(Lotter et al 2016)

Conditional GANs

There is extra conditioning information as input to the generator

image-to-image translation

image-to-image translation

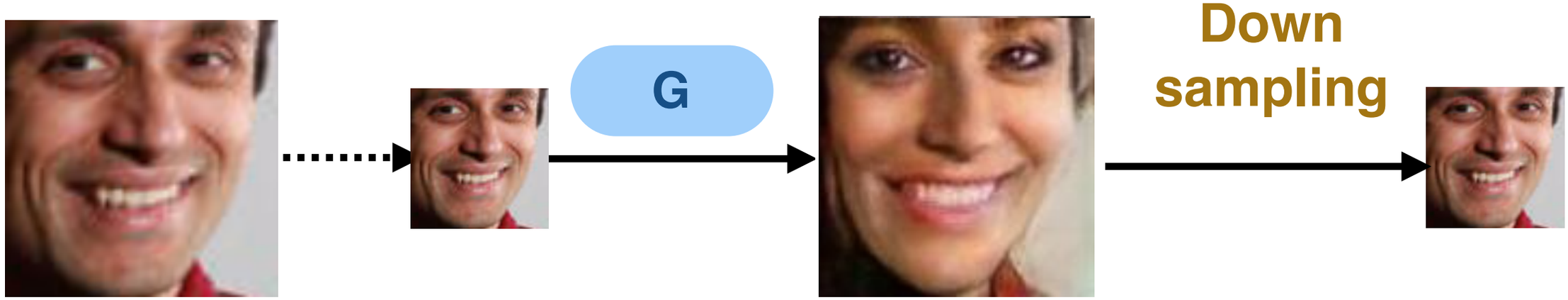


image-to-image translation

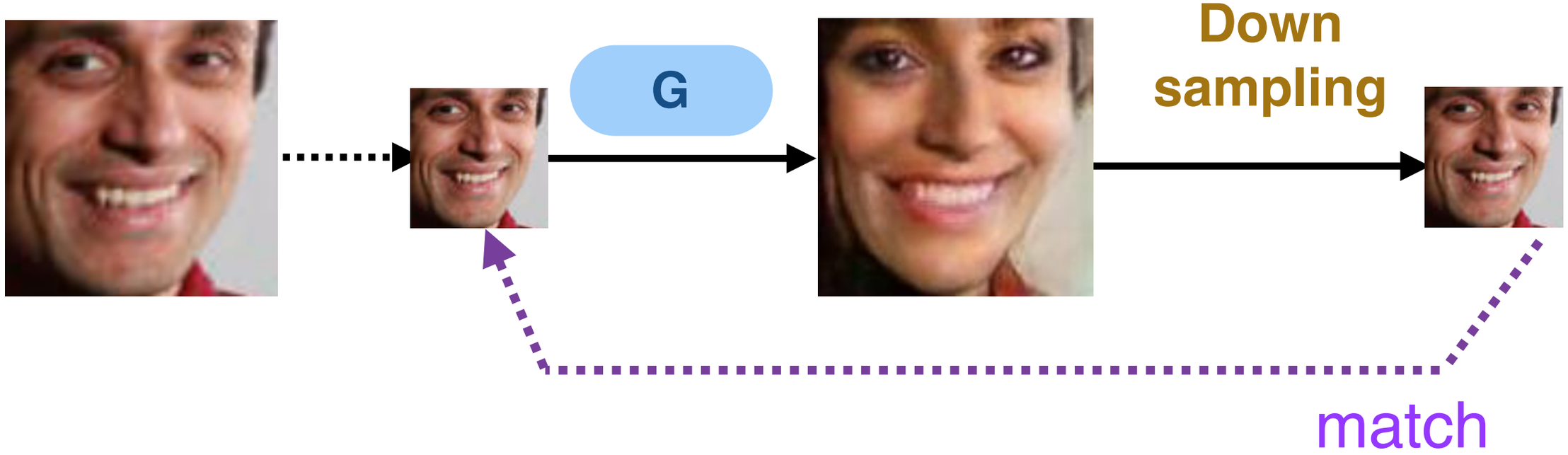
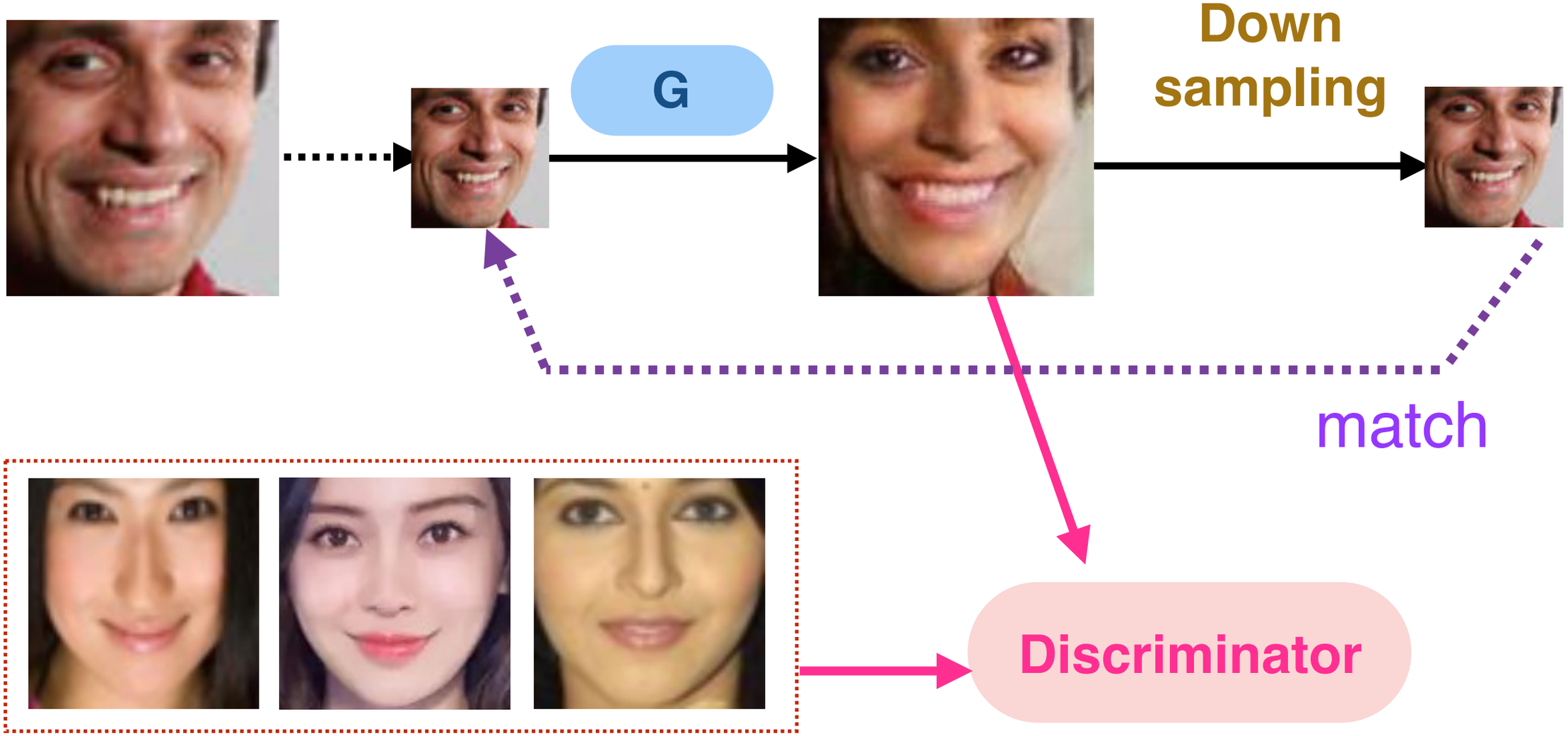
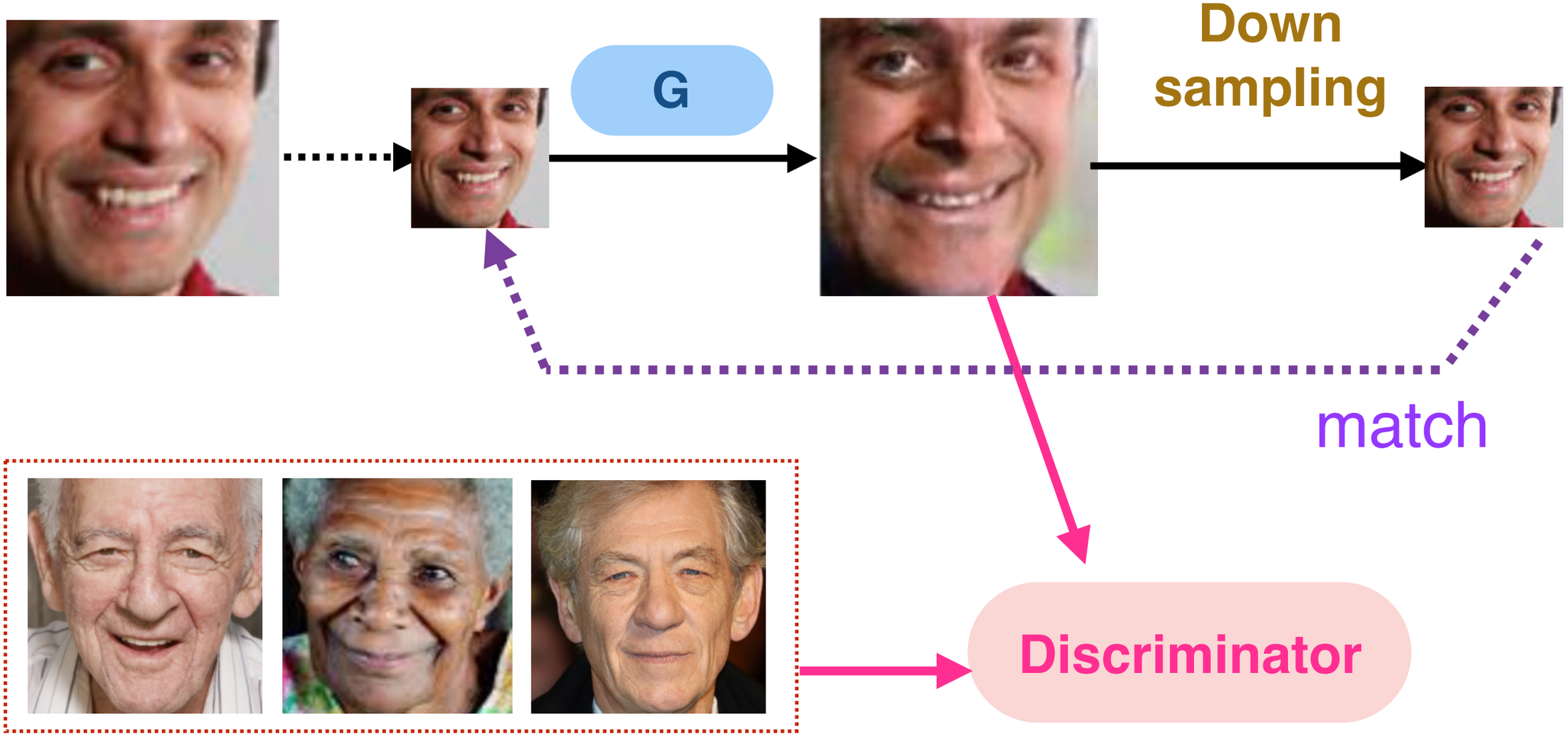


image-to-image translation



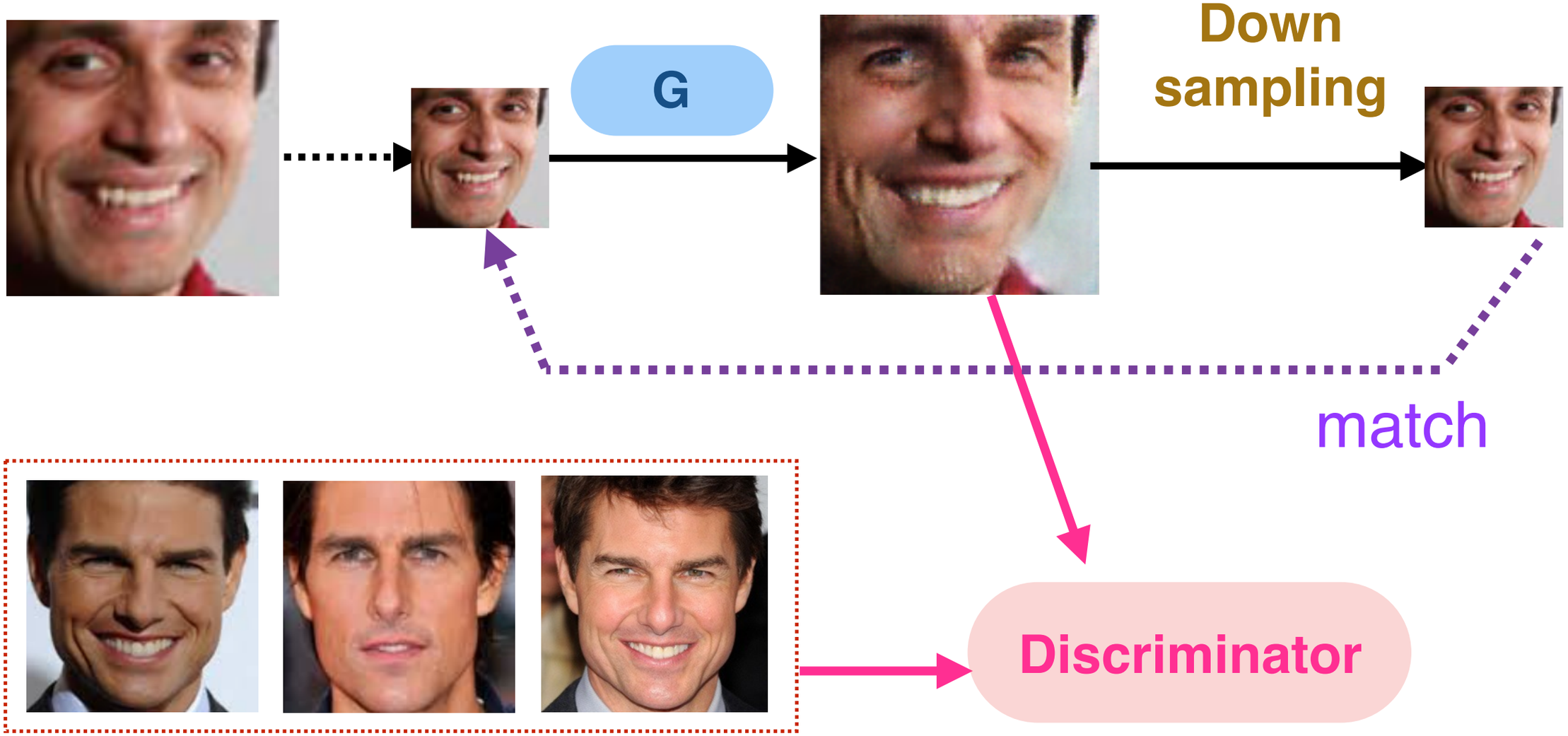
By putting different memory in the repositories, we get different results

image-to-image translation



By putting different images in the memory repo, 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-actions from the expert demonstrations and state-actions visited by the learnt policy π_θ

$$\min_{\pi_\theta} \max_{D} V(D,G) = \mathbb{E}_{(s,a)\sim \text{Demo}}[\log D(s,a)] + \mathbb{E}_{(s,a)\sim \pi_\theta}[\log(1-D(s,a))]$$

The reward for the policy optimization is how well I matched the demonstrator's trajectory distribution, else, how well I confused the discriminator.

$$r(s,a) = \log D(s,a), (s,a) \sim \pi_\theta$$

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

Generative Adversarial Imitation learning

Find a policy π_θ that makes it impossible for a discriminator network to distinguish between state-actions from the expert demonstrations and state-actions visited by the learnt policy π_θ

$$\min_{\pi_\theta} \max_{D} V(D,G) = \mathbb{E}_{(s,a)\sim \text{Demo}}[\log D(s,a)] + \mathbb{E}_{(s,a)\sim \pi_\theta}[\log(1-D(s,a))]$$

The reward for the policy optimization is how well I matched the demonstrator's trajectory distribution, else, how well I confused the discriminator.

$$r(s,a) = \log D(s,a), (s,a) \sim \pi_\theta$$

Q: how would we change the above if the action space between expert and imitator were different?

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

Generative Adversarial Imitation learning-action spaces differ between teacher and learner

Find a policy π_θ that makes it impossible for a discriminator network to distinguish between states from the expert demonstrations and states visited by the learnt policy π_θ

$$\min_{\pi_\theta} \max_{D} V(D,G) = \mathbb{E}_{(s)\sim \text{Demo}}[\log D(s)] + \mathbb{E}_{(a,s)\sim\pi_\theta}[\log(1-D(s))]$$

D outputs 1 if states come from the expert demonstrations Demo.

Reward for the policy optimization is how well I matched the demo trajectory distribution, else, how well I confused the discriminator.

$$r(s) = \log D(s)$$

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

Generative Adversarial Imitation Learning

We do not know policy gradients yet. In HW1 we will use the adversarial reward with natural evolution as our policy search method

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

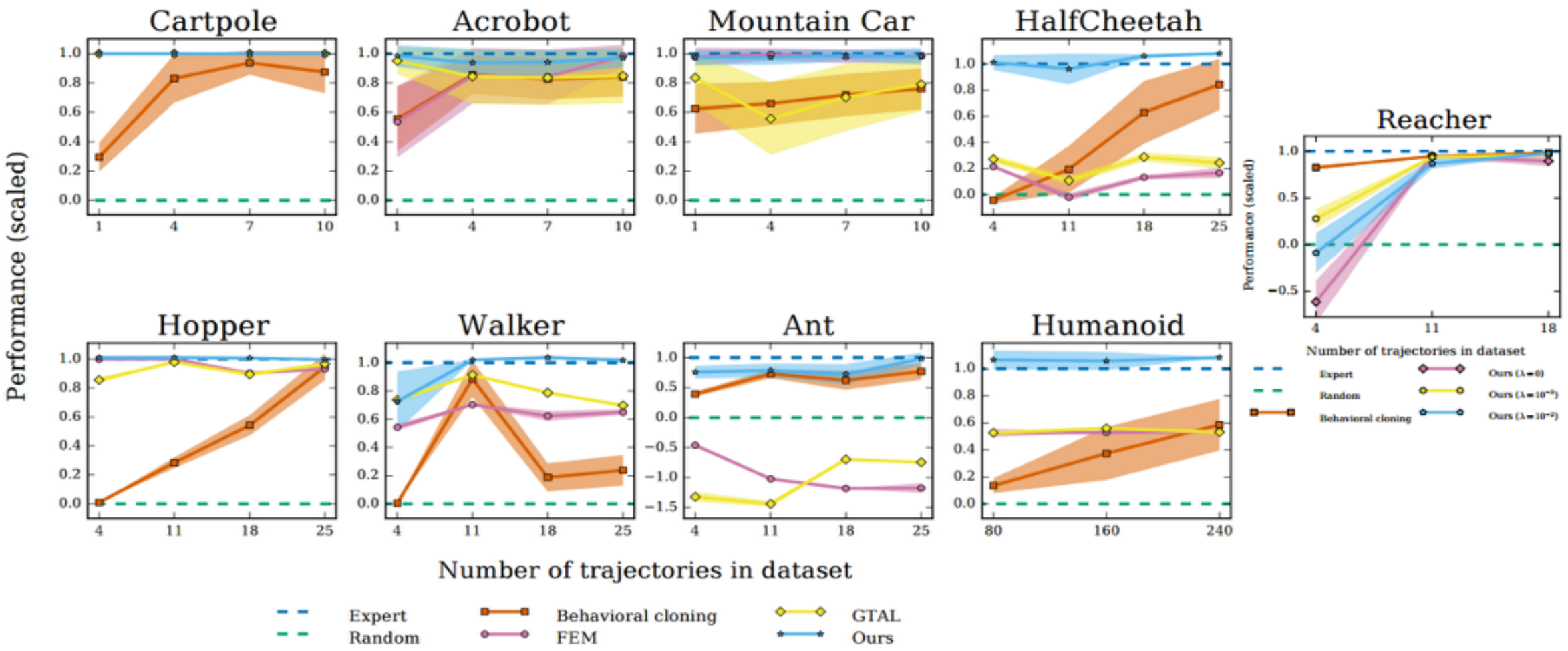
$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \tag{17}$$

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

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

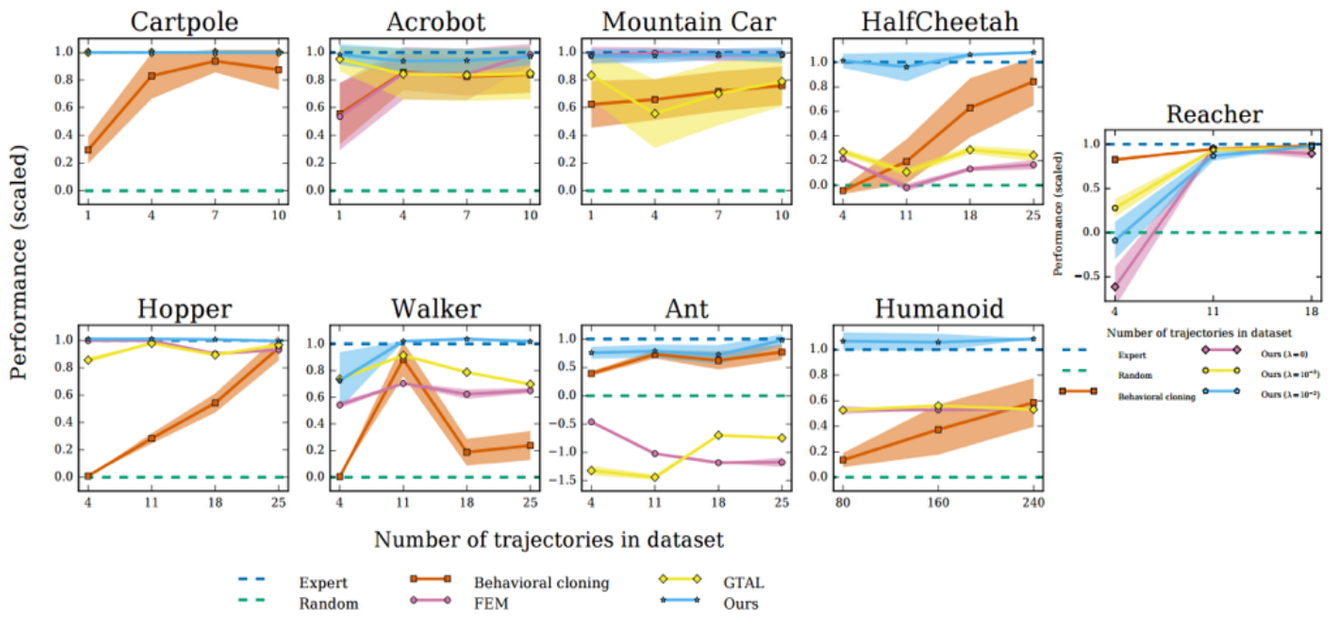
6: **end for**

Generative Adversarial Imitation learning



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

Generative Adversarial Imitation learning



- GAIL: a reinforcement learning method with a reward based on **trajectory distribution matching** between the agent and an expert. This does seem to be the right objective.
- BC: reduces imitation learning to supervised learning for individual time steps. Not the right cost function: the training objective is open loop action prediction while we test closed loop, that is why we suffer from distribution shift.
- GAIL performs better than behaviour cloning but it requires MORE interactions with the environment.