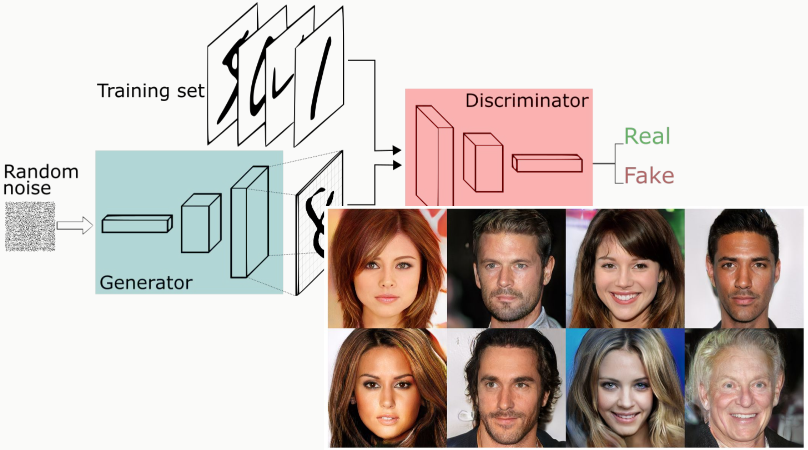
Probability Functional Descent:

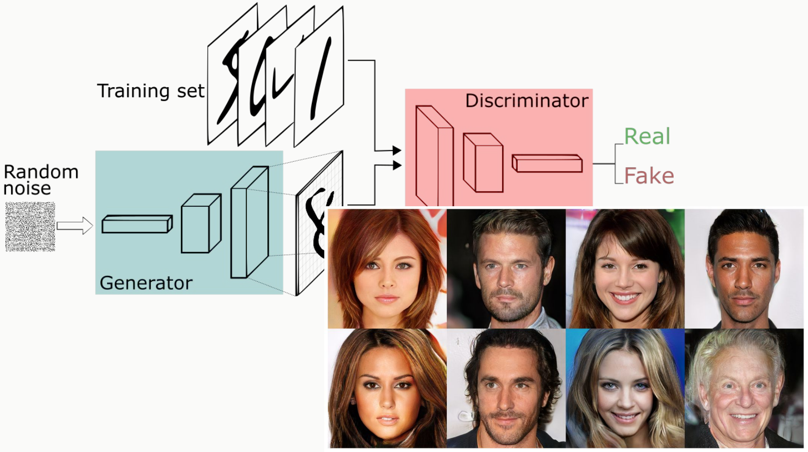
A Unifying Perspective on GANs, VI, and RL

Casey Chu *<caseychu@stanford.edu>* Jose Blanchet Peter Glynn

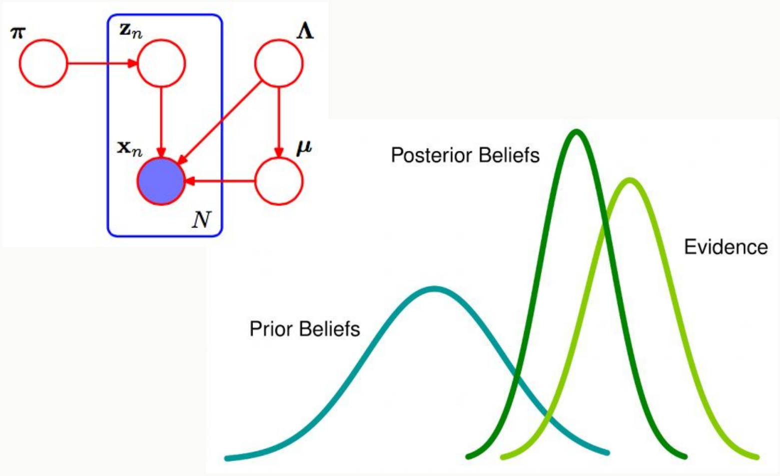
Deep generative models



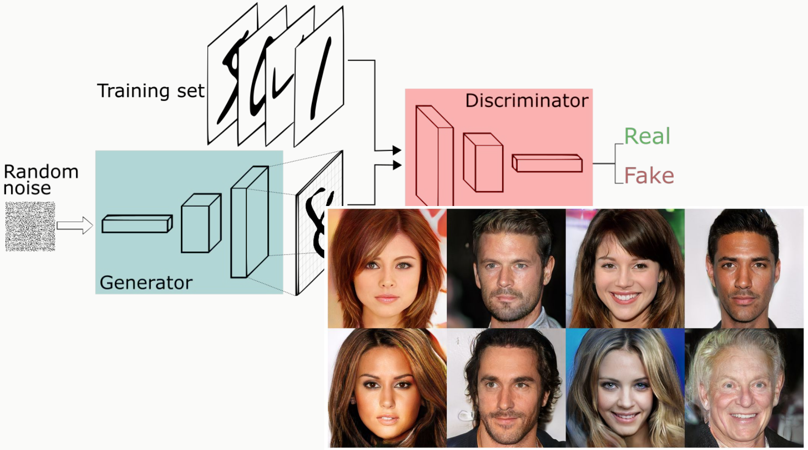
Deep generative models



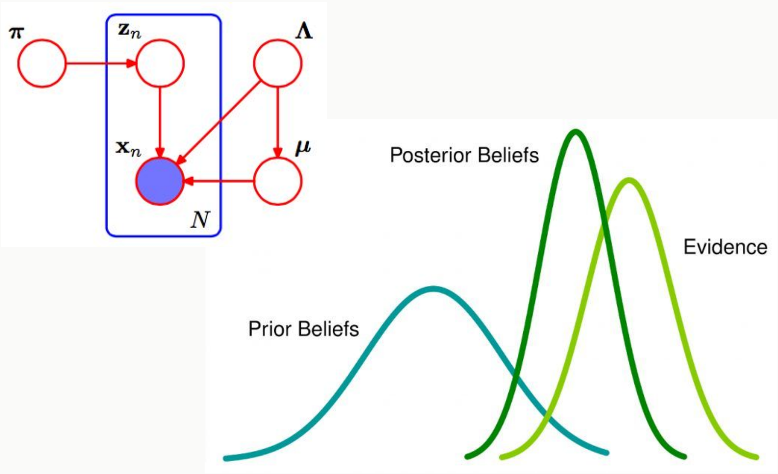
Variational inference



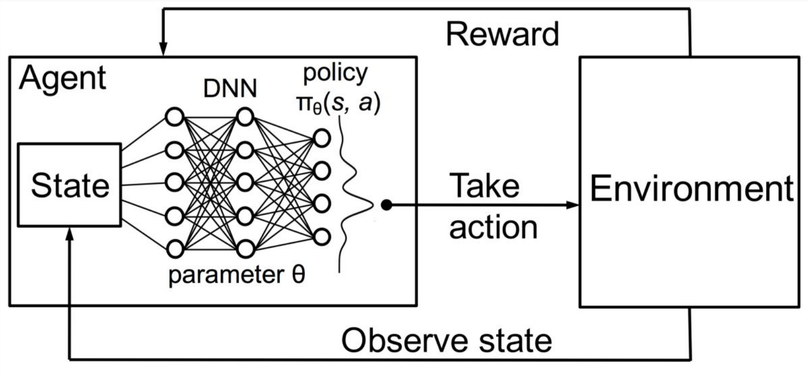
Deep generative models



Variational inference



Deep reinforcement learning



Probability functional

$$J : \mathbf{P}(X) \to \mathbb{R}$$

Probability functional

$$J : \mathbf{P}(X) \to \mathbb{R}$$

$\downarrow$ “gradient” ∇J

Probability functional

$$J : \mathbf{P}(X) \to \mathbb{R}$$

↓ “gradient” ∇J

von Mises influence function

$$\Psi : X \to \mathbb{R}$$

Gradient descent on $f : \mathbb{R}^n \to \mathbb{R}$

0. Initialize $x \in \mathbb{R}^n$ arbitrarily
1. Compute the gradient $g = \nabla f(x)$
2. Choose x' such that $x' \cdot g < x \cdot g$ (usually, we set $x' = x - \alpha g$)

Gradient descent on $f : \mathbb{R}^n \to \mathbb{R}$

0. Initialize $x \in \mathbb{R}^n$ arbitrarily
1. Compute the gradient $g = \nabla f(x)$
2. Choose x' such that $x' \cdot g < x \cdot g$ (usually, we set $x' = x - \alpha g$)

Probability functional descent on $J : \mathbf{P}(X) \to \mathbb{R}$

0. Initialize a distribution $\mu \in \mathbf{P}(X)$ arbitrarily
1. Compute the **influence function** Ψ of J at μ
2. Choose μ' such that $\mathbb{E}_{x \sim \mu'}[\Psi(x)] < \mathbb{E}_{x \sim \mu}[\Psi(x)]$

Generative modeling

$$J_{\mathrm{G}}(\mu) = \mathrm{D}(\mu \,\|\, \nu_0)$$

where D is e.g. Jensen–Shannon, Wasserstein

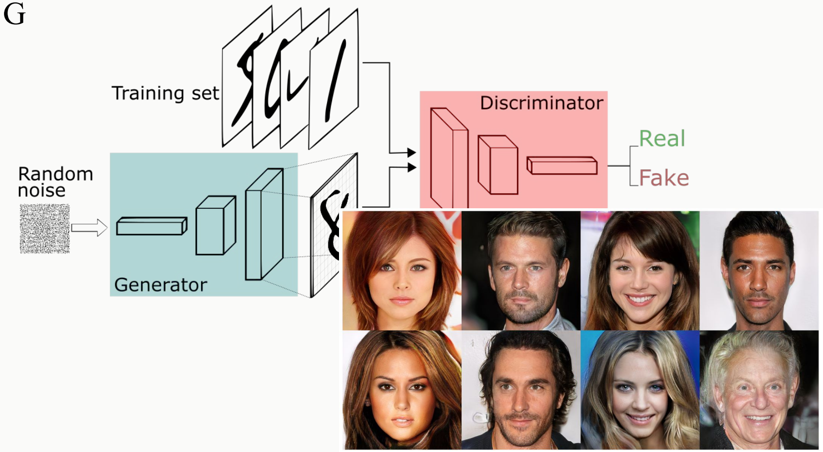
1. Optimize the **discriminator**, which approximates the influence function of J_{G}
2. Update the **generator** μ

PFD recovers:

- Minimax GAN
- Non-saturating GAN
- Wasserstein GAN

Probability functional descent

1. Compute the **influence function** Ψ of J at μ
2. Choose μ' such that $\mathbb{E}_{x \sim \mu'}[\Psi(x)] < \mathbb{E}_{x \sim \mu}[\Psi(x)]$



Variational inference

$$J_{\mathrm{VI}}(q) = \mathrm{KL}(q(\theta) \,||\, p(\theta|x))$$

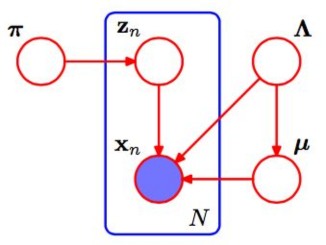
1. Compute the **ELBO**, $\log(q(\theta)/p(x,\theta))$, the influence function for J_{VI}
2. Update the **approximate posterior** q

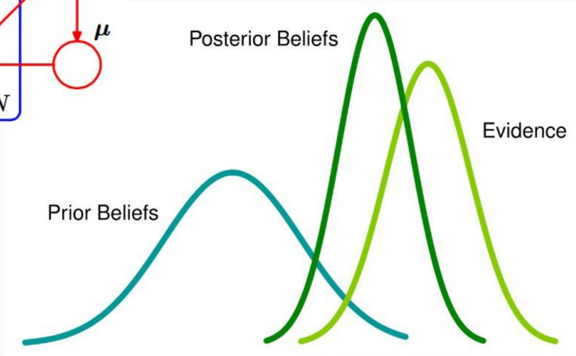
PFD recovers:

- Black-box variational inference
- Adversarial variational Bayes
- Approximate posterior distillation

Probability functional descent

1. Compute the **influence function** Ψ of J at μ
2. Choose μ' such that $\mathbb{E}_{x\sim\mu'}[\Psi(x)] < \mathbb{E}_{x\sim\mu}[\Psi(x)]$





Reinforcement learning

$$J_{\mathrm{RL}}(\pi) = \mathbb{E}_{\pi}[\textstyle\sum_t \gamma^t R_t]$$

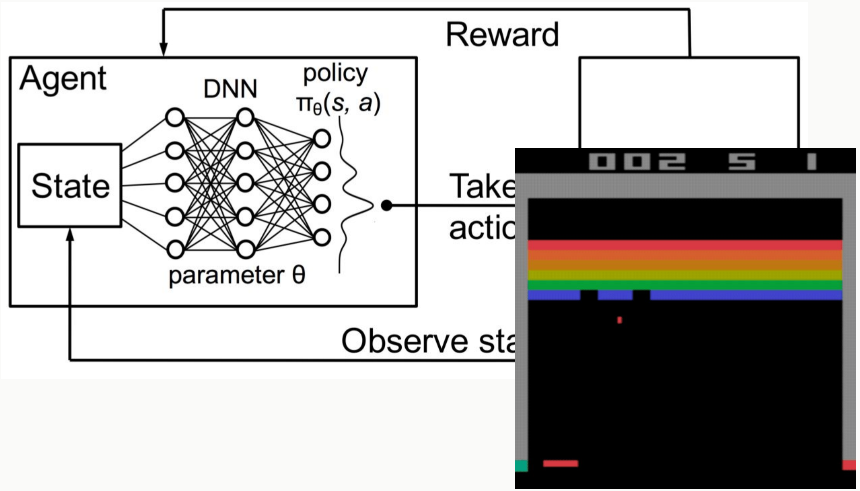
1. Approximate the **advantage** $Q^{\pi}(s,a) - V^{\pi}(s)$, the influence function for J_{RL}
2. Update the **policy** π

PFD recovers:

- Policy gradient
- Actor-critic
- Dual actor critic

Probability functional descent

1. Compute the **influence function** Ψ of J at μ
2. Choose μ' such that $\mathbb{E}_{x \sim \mu'}[\Psi(x)] < \mathbb{E}_{x \sim \mu}[\Psi(x)]$



Probability functional descent is a unifying perspective that enables the easy development of new algorithms.

Probability functional descent is a unifying perspective that enables the easy development of new algorithms.

https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/
https://arxiv.org/abs/1710.10196
https://www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english/
https://stats.stackexchange.com/questions/246117/applying-stochastic-variational-inference-to-bayesian-mixture-of-gaussian
http://people.csail.mit.edu/hongzi/content/publications/DeepRM-HotNets16.pdf
https://towardsdatascience.com/atari-reinforcement-learning-in-depth-part-1-ddqn-ceaa762a546f

Probability Functional Descent:

A Unifying Perspective on GANs, VI, and RL

Casey Chu *<caseychu@stanford.edu>* Jose Blanchet Peter Glynn