Introduction to GAN’s

John Thickstun

Transforming Random Variables

- Suppose I have access to samples from $\mathcal{N}(0,1)$.

- But I want samples from $\mathcal{N}(\mu,\sigma^2)$.

- Let $g(x) = \mu + \sigma x$. If $\varepsilon \sim \mathcal{N}(0,1)$ then $g(\varepsilon) \sim \mathcal{N}(\mu,\sigma^2)$.

Inverse Transform Sampling

- I have samples from $\mathrm{Uniform}([0,1])$. I want samples with CDF F.

- Define the inverse CDF by $F^{-1}(u) = \inf\{x : F(x) \geq u\}$.

- If $u \sim \mathrm{Uniform}([0,1])$, then $F^{-1}(u)$ is distributed according to F.

The Idea Behind GAN's

- I have access to samples from a simple distribution q on space $\mathcal{Z}$.

- I want samples from some complicated distribution p on space $\mathcal{X}$.

- Learn a function $g : \mathcal{Z} \to \mathcal{X}$ such that, if $z \sim \rho$, then $g(z) \sim p$.

The Potential of The GAN Idea

Large Scale GAN Training For High Fidelity Natural Image Synthesis
[Brock, Donahue, and Simonyan (2019)]

Pushforward Distributions

- Given a distribution ρ on $\mathcal{Z}$, $g : \mathcal{Z} \to \mathcal{X}$ induces a distribution on $\mathcal{X}$.

- For any set $A \subset \mathcal{X}$, $\Pr(A) \equiv \Pr(g^{-1}(A))$.

- $\Pr(g^{-1}(A)) = \int_{g^{-1}(A)} \rho(z)\, dz = \int_A \rho(g^{-1}(A)) |\nabla_x g^{-1}(x)|\, dx.$

Learning from Samples

- Given finite samples $x_1, \ldots, x_n \sim p$, unlimited samples $z \sim \rho$
- Learn a function $g_\theta : \mathcal{Z} \to \mathcal{X}$, which induces a distribution p_θ on $\mathcal{X}$.
- Learn the parameters so that $p_\theta \approx p$.

Maximize the Likelihood?

- Find a function that makes the observed data likely:

$$\sup_{\theta} \mathop{\mathbb{E}}_{x \sim p} \log p_\theta(x) \approx \sup_{\theta} \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i).$$

- How do we compute $p_\theta(x_i)$?

$$p_\theta(x_i) = q(g_\theta^{-1}(x_i)) |\nabla_x g_\theta^{-1}(x_i)|.$$

- That doesn't look fun!

What Are Our Options?

- Write down parameterized families with simple inverses and Jacobians
    - Dinh et al. 2017, Kingma and Dhariwal 2018
- Suck it up and compute the inverses and Jacobians
    - Hand and Voroninski 2019, Ma et al. 2018
- Give up and try something else (GAN)
    - Goodfellow et al. 2014, Brock et al. 2018

Towards the GAN

- Remember our broad goal: find a pushforward $g_\theta : \mathcal{Z} \to \mathcal{X}$ so that $p_\theta \approx p$.

- How do we define similarity/divergence between distributions?

- How do we compute/estimate the similarity?

Distributional f-Divergence

- Let $f : \mathbb{R} \to \mathbb{R}$ be convex, lower-semicontinuous, and $f(0) = 1$:

$$D_f(p||q) \equiv \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) dx.$$

- For example, if $f(x) = x \log x$ then $D_f(p||q)$ is KL-divergence.

- We can construct lower bounds on an f-divergence.

Lower Bounds on f-Divergences

- For *any* function $T : \mathcal{X} \to \mathbb{R}$, [Nguyen, Wainwright, and Jordan 2010]

$$D_f(p||q) \geq \mathop{\mathbb{E}}_{x \sim p} T(x) - \mathop{\mathbb{E}}_{x \sim q} f^*(T(x)).$$

- The function $f^* : \mathbb{R} \to \mathbb{R}$ is the convex conjugate of f:

$$f^*(t) \equiv \sup_x \{tx - f(x)\}.$$

- The lower bound only uses samples! No need to evaluate $p(x)$.

GAN's in Broad Strokes

- Solve a saddle-point problem

$$\theta_f = \arg\inf_{\theta} \sup_{\phi} \left[\mathop{\mathbb{E}}_{x\sim p} T_\phi(x) - \mathop{\mathbb{E}}_{z\sim\rho} f^*(T_\phi(g_\theta(z))) \right].$$

- Use an expressive parameterized family of functions $T_\phi : \mathcal{X} \to \mathbb{R}$.

- Adversarial: g_θ wants to minimize the objective, and T_ϕ wants to maximize.

Proof of the Lower Bound

- Fenchel duality: $f(x) = \sup_t \{tx - f^*(t)\}$.

$$
\begin{aligned}
D_f(p||q) &= \int_{\mathcal{X}} q(x) \sup_t \left[t\frac{p(x)}{q(x)} - f^*(t) \right] dx \\
&= \int_{\mathcal{X}} \sup_t \left[tp(x) - f^*(t)q(x) \right] dx \\
&= \sup_{T:\mathcal{X}\to\mathbb{R}} \int_{\mathcal{X}} \left(T(x)p(x) - f^*(T(x))q(x) \right) dx \\
&= \sup_{T:\mathcal{X}\to\mathbb{R}} \left[\mathbb{E}_{x\sim p} T(x) - \mathbb{E}_{x\sim q} f^*(T(x)) \right].
\end{aligned}
$$

GAN's in Broad Strokes

- Solve a saddle-point problem

$$\theta_f = \arg\inf_{\theta}\sup_{\phi}\left[\mathop{\mathbb{E}}_{x\sim p} T_\phi(x) - \mathop{\mathbb{E}}_{z\sim\rho} f^*(T_\phi(g_\theta(z)))\right].$$

- Use an expressive parameterized family of functions $T_\phi : \mathcal{X} \to \mathbb{R}$.

- Adversarial: g_θ wants to minimize the objective, and T_ϕ wants to maximize.

The Goodfellow GAN

- Pick a divergence, e.g. $f(x) = x \log x - (x+1)\log(x+1)$ results in

$$D_f(p||q) \equiv 2\mathrm{JSD}(p,q) - \log(4).$$

- Compute the convex conjugate (hint: calculus). In this case:

$$f^*(t) = -\log(1-e^t).$$

- Parameterizing $T_\phi(x) = \log(d_\phi(x))$ results in

$$\theta_f = \arg\inf_\theta \sup_\phi \left[\mathop{\mathbb{E}}_{x\sim p} \log d_\phi(x) + \mathop{\mathbb{E}}_{z\sim\rho} \log(1 - d_\phi(g_\theta(z))) \right].$$

The Discriminator Perspective

- The GAN objective looks a bit like a binary cross-entropy loss:

$$\mathop{\mathbb{E}}_{x\sim p} \log d_\phi(x) + \mathop{\mathbb{E}}_{z\sim\rho} \log(1 - d_\phi(g_\theta(z))).$$

- We can formalize this observation. Let $y \sim \text{Bernoulli}(.5)$ and define

$$r_\theta(x|y=0) = p_\theta(x)$$
$$r_\theta(x|y=1) = p(x).$$

(y labels whether x comes from p_θ or p)

- Let $p_\phi(y|x) = \text{Bernoulli}(d_\phi(x))$. The objective can be re-written as

$$\mathop{\mathbb{E}}_{\substack{y\sim\text{Bernoulli}(.5)\\ x\sim r_\theta}} \log p_\phi(y|x) = -H(r(y|x), p_\phi(y|x)).$$

The Bayes-Optimal Classifier

- Think of $p_\phi(y|x) = \mathrm{Bernoulli}(d_\phi(x))$ as a classifier that predicts y given x.

- The Bayes optimal classifier (for a given generator g_θ) is $r_\theta(y|x)$.

- Bayes' rule: $r(y=1|x) = \dfrac{r(x|y=1)r(y=1)}{r(x)} = \dfrac{p(x)}{p(x)+p_\theta(x)}$

Coming Full-Circle

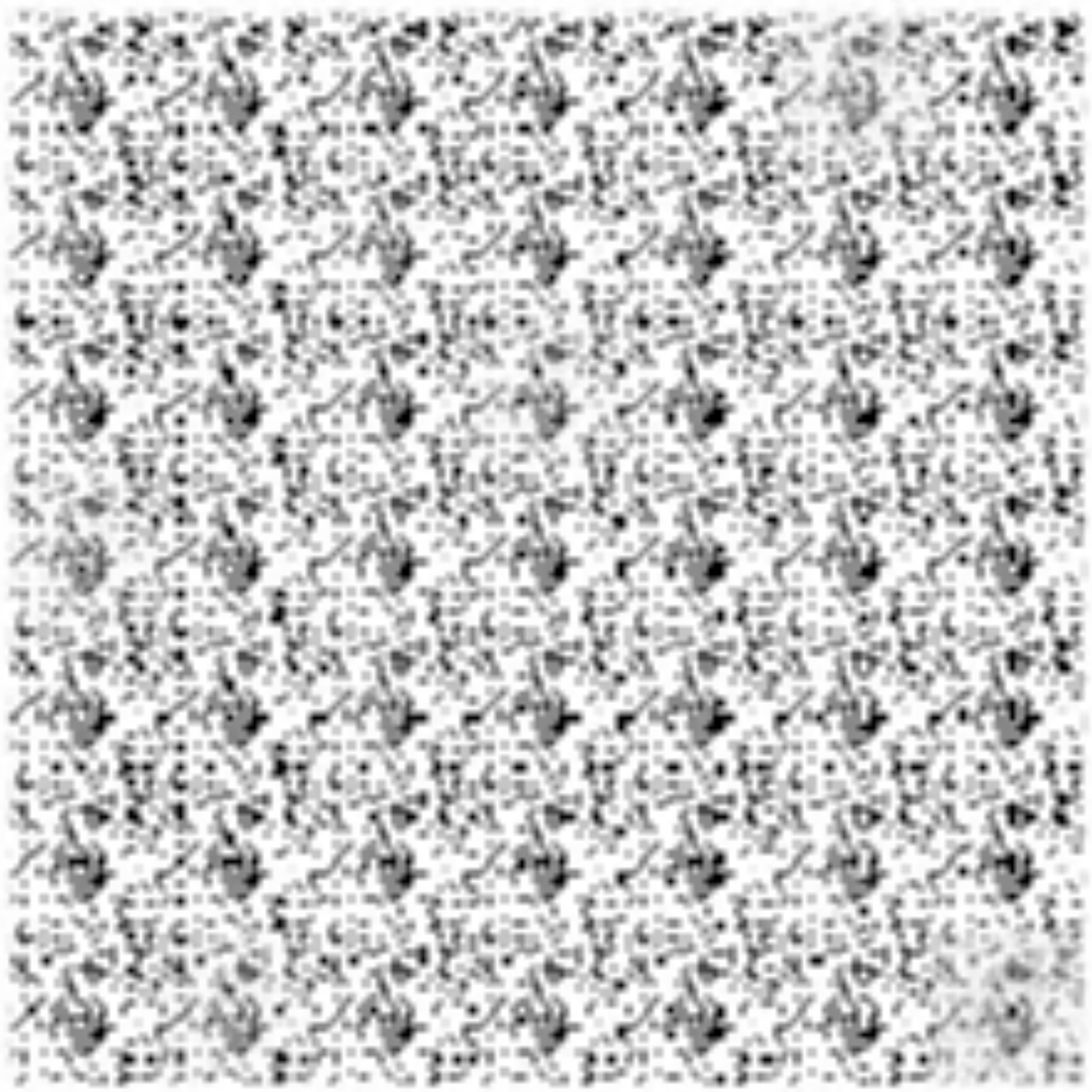
- What if we just plug the optimal classifier into the GAN objective?

$$\sup_{\phi} \left[\mathop{\mathbb{E}}_{x \sim p} \log d_{\phi}(x) + \mathop{\mathbb{E}}_{z \sim \rho} \log(1 - d_{\phi}(g_{\theta}(z))) \right]$$

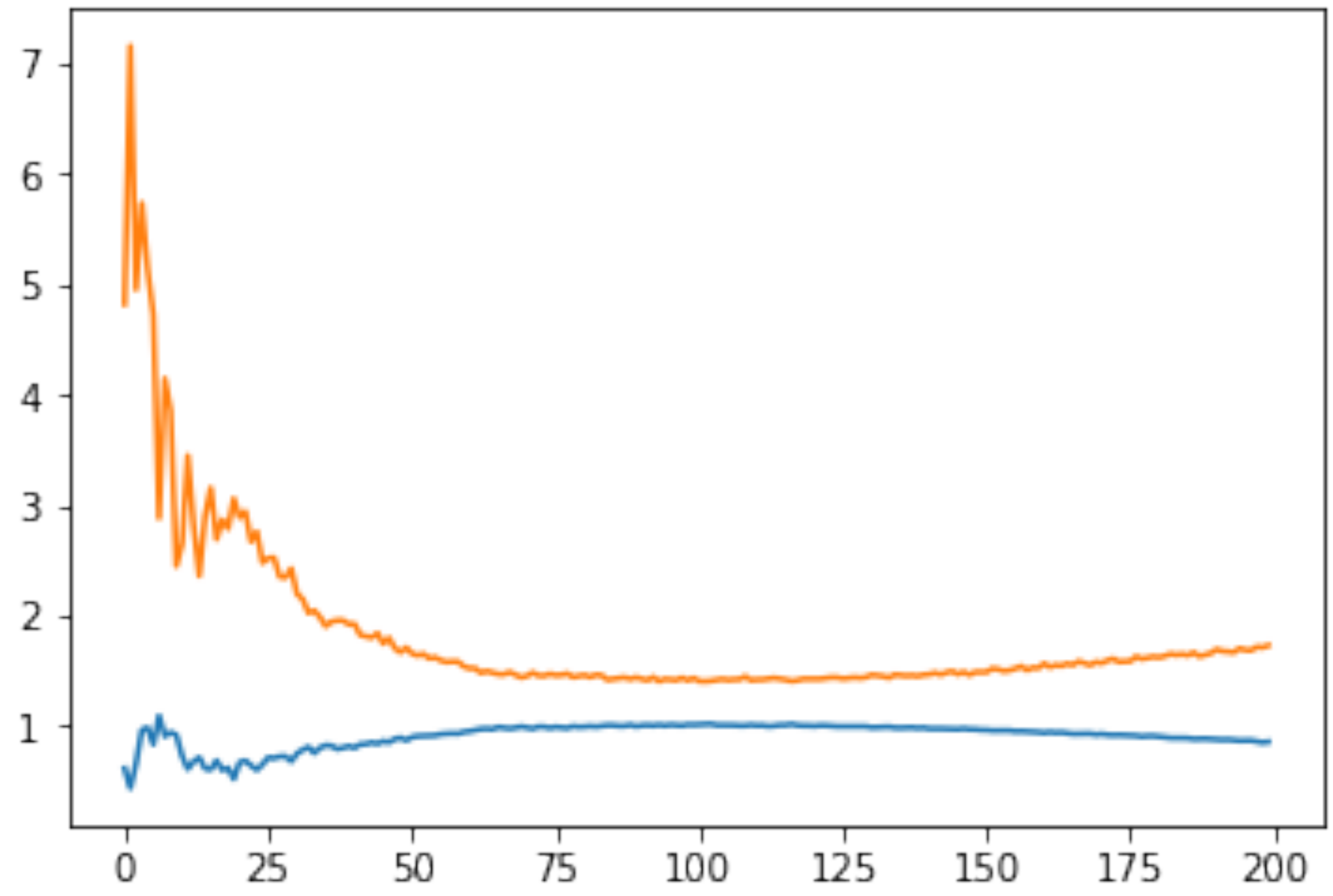
$$= \mathop{\mathbb{E}}_{x \sim p} \log \frac{p(x)}{p(x) + p_{\theta}(x)} + \mathop{\mathbb{E}}_{z \sim \rho} \log \left(1 - \frac{p(g_{\theta}(z))}{p(g_{\theta}(z)) + p_{\theta}(g_{\theta}(z))} \right)$$

- Don't need to solve a saddle point problem! But we can't evaluate $p(x)$…

Running a GAN on Data



Training Curves



Orange: Generator loss, Blue: Discriminator loss

Lingering Questions

- There are lots of saddle-points in this space! How do we find a good one?
- How do we evaluate our results? What makes a saddle-point good?
- Ethical concerns: how do we interact with media in the age of deepfakes?