Metropolis-Hastings Generative Adversarial Networks

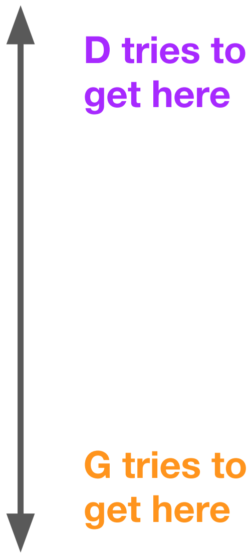
Ryan Turner, Jane Hung, Eric Frank, Yunus Saatci, Jason Yosinski

Poster #201

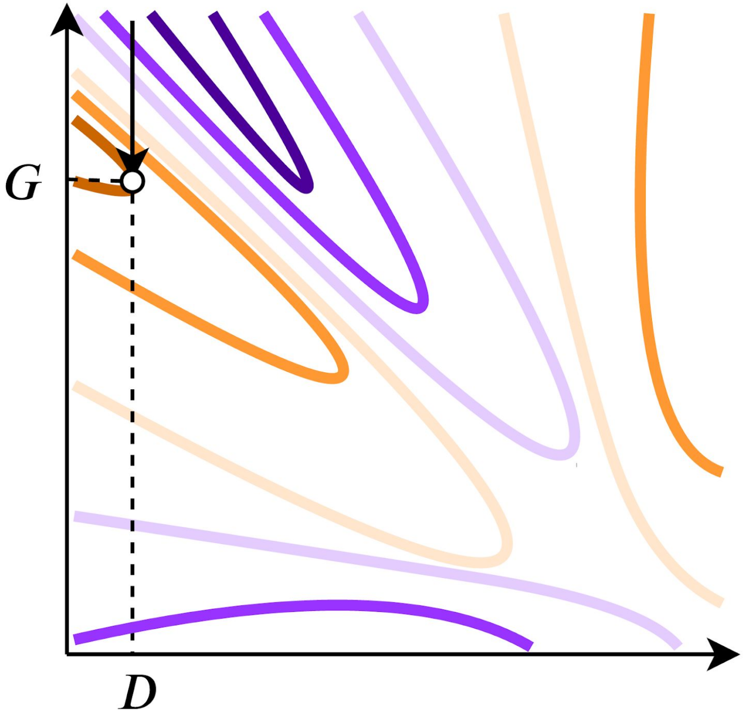
Uber AI

Typical GAN training

Typical GAN training



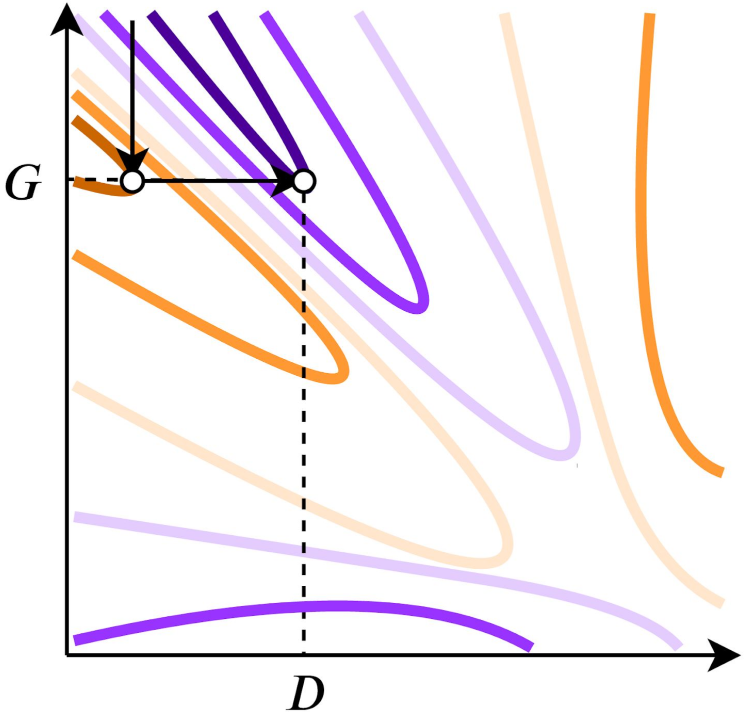
Typical GAN training

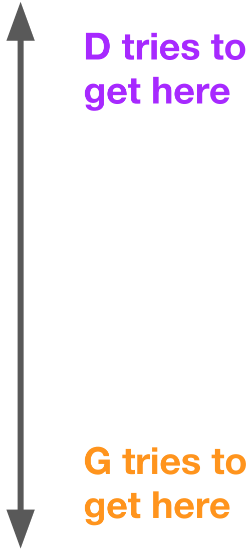


D tries to get here

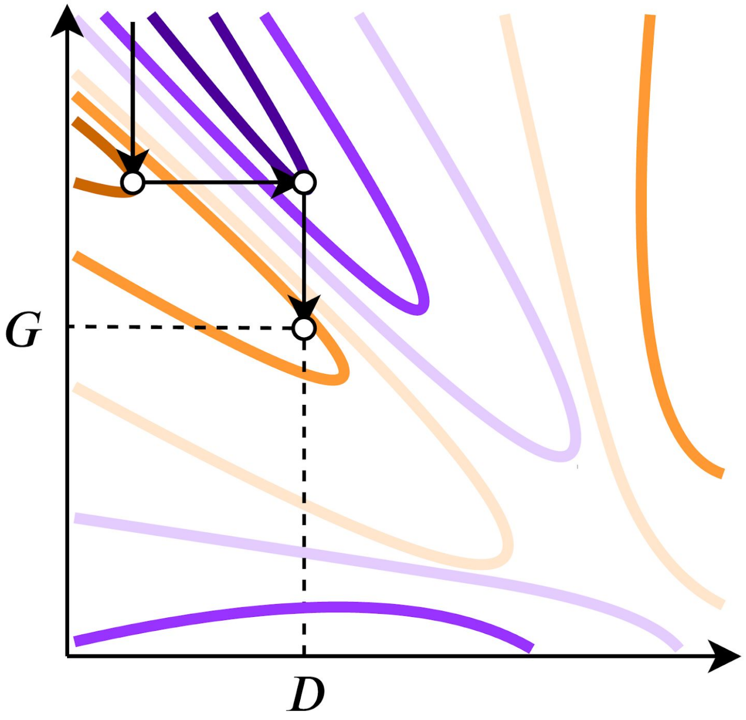
G tries to get here

Typical GAN training





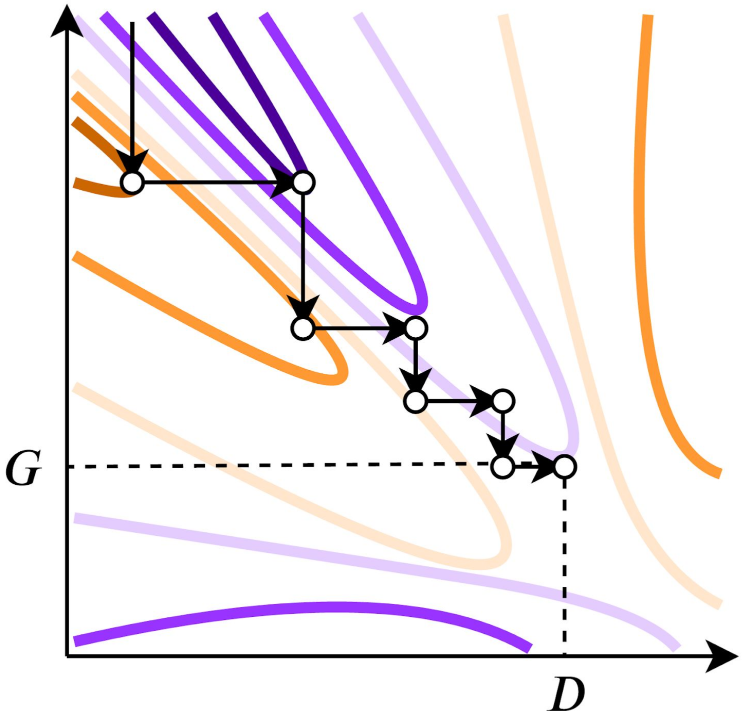
Typical GAN training

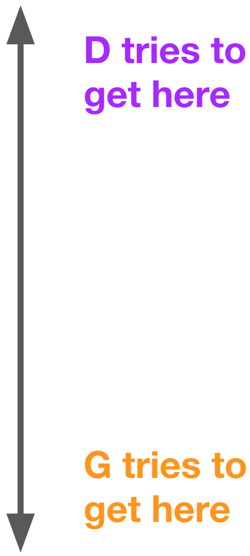


D tries to get here

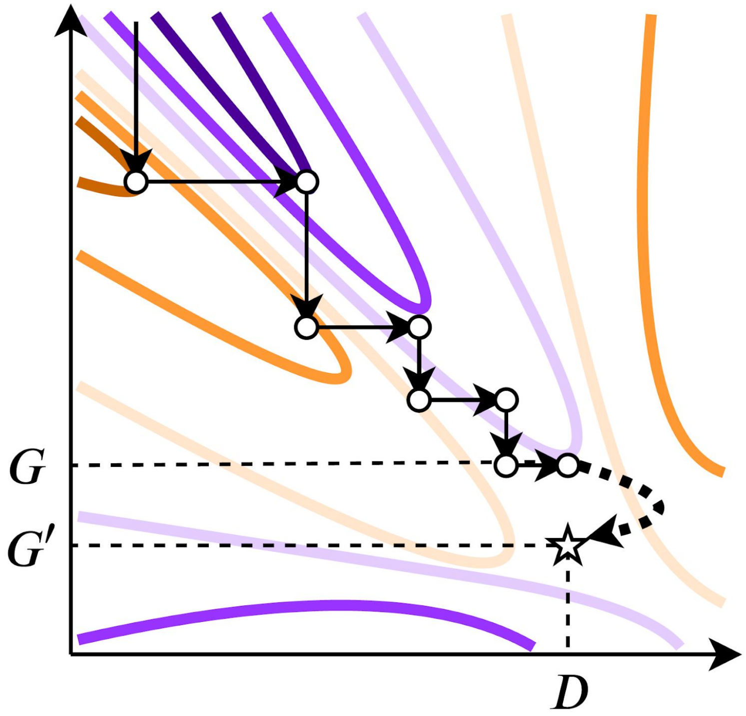
G tries to get here

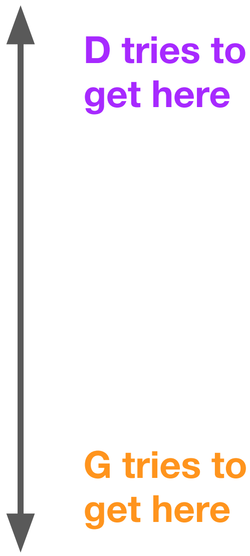
Typical GAN training … gets stuck





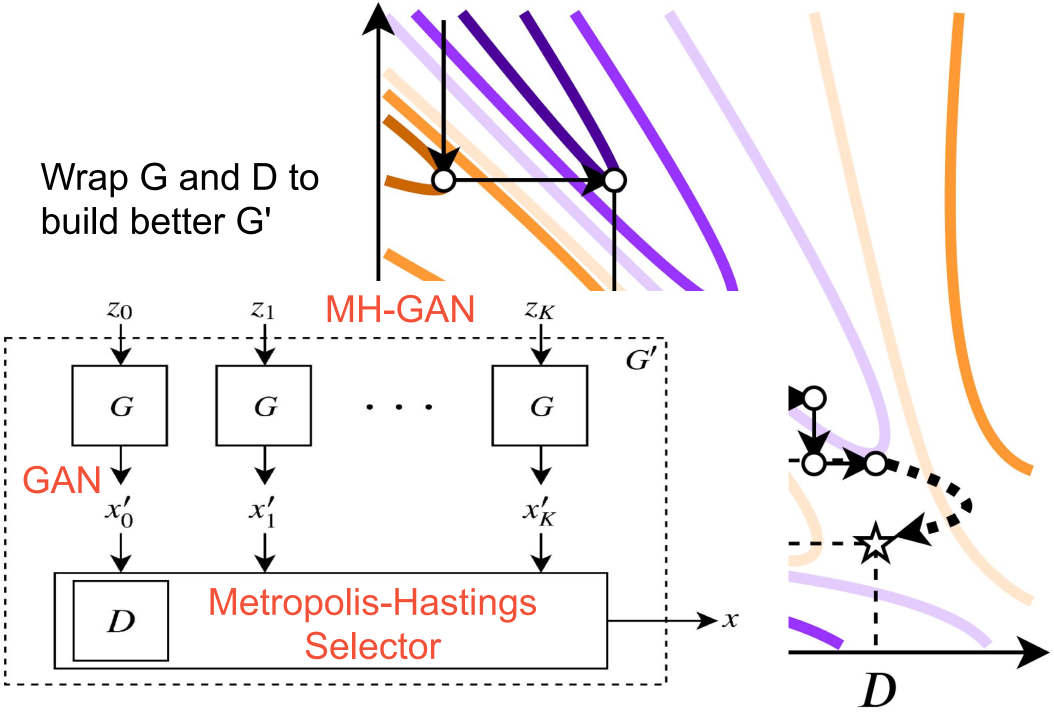
MH-GAN helps you reach the star

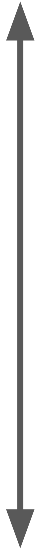




MH-GAN helps you reach the star

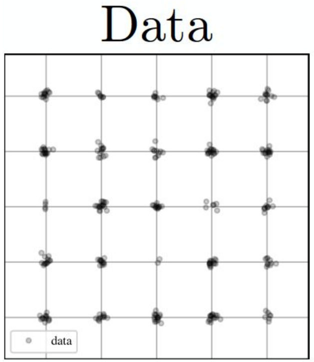
- Wrap G and D to build better G'





MH recovers the true data distribution

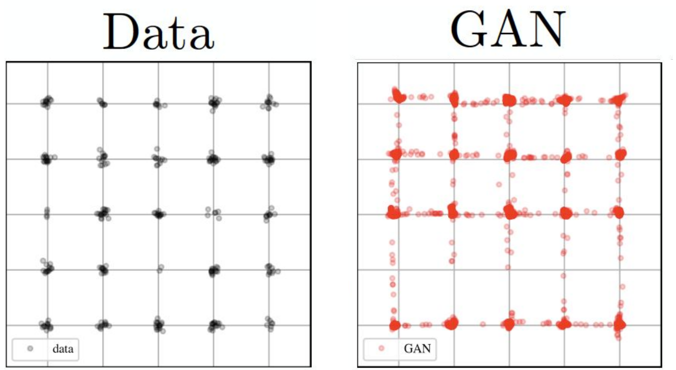
"Mixture of Gaussians" dataset [1]



[1] Azadi et al. ICLR 2019.

MH recovers the true data distribution

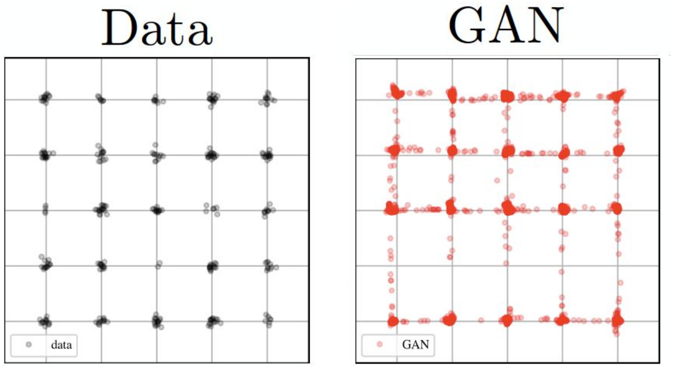
"Mixture of Gaussians" dataset [1]



[1] Azadi et al. ICLR 2019.

MH recovers the true data distribution

“Mixture of Gaussians” dataset [1]

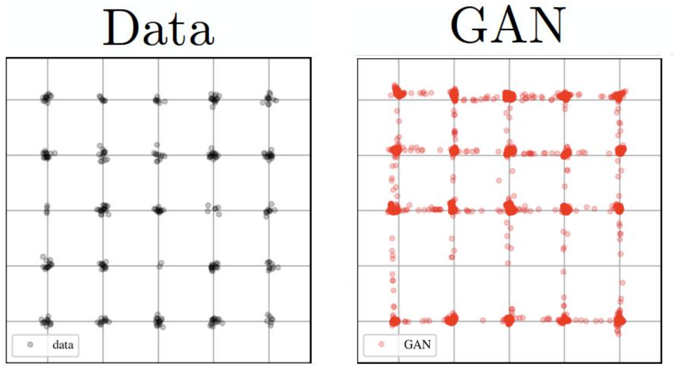


Dropped modes!

[1] Azadi et al. ICLR 2019.

MH recovers the true data distribution

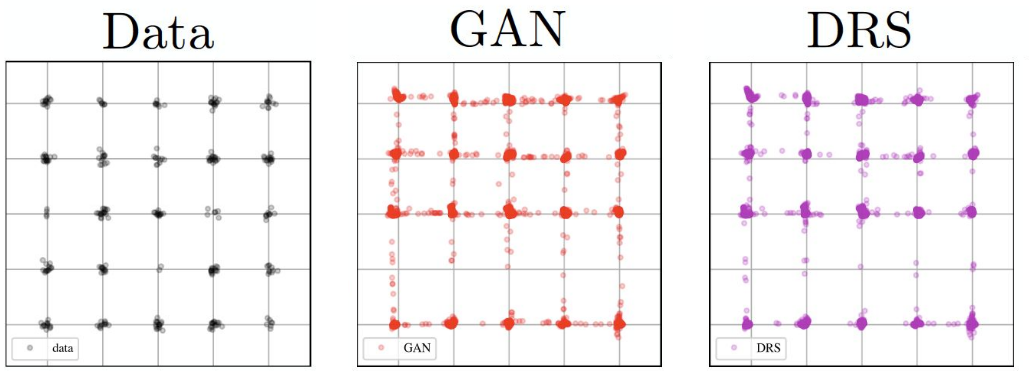
“Mixture of Gaussians” dataset [1]



[1] Azadi et al. ICLR 2019.

MH recovers the true data distribution

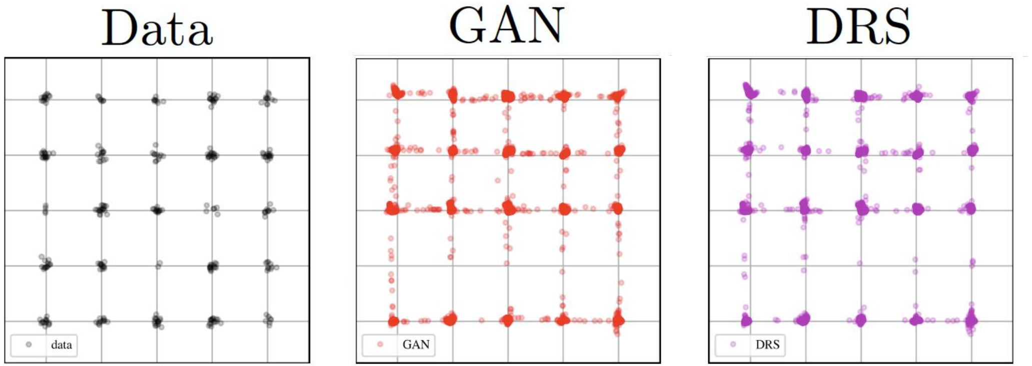
"Mixture of Gaussians" dataset [1]



[1] Azadi et al. ICLR 2019.

MH recovers the true data distribution

"Mixture of Gaussians" dataset [1]

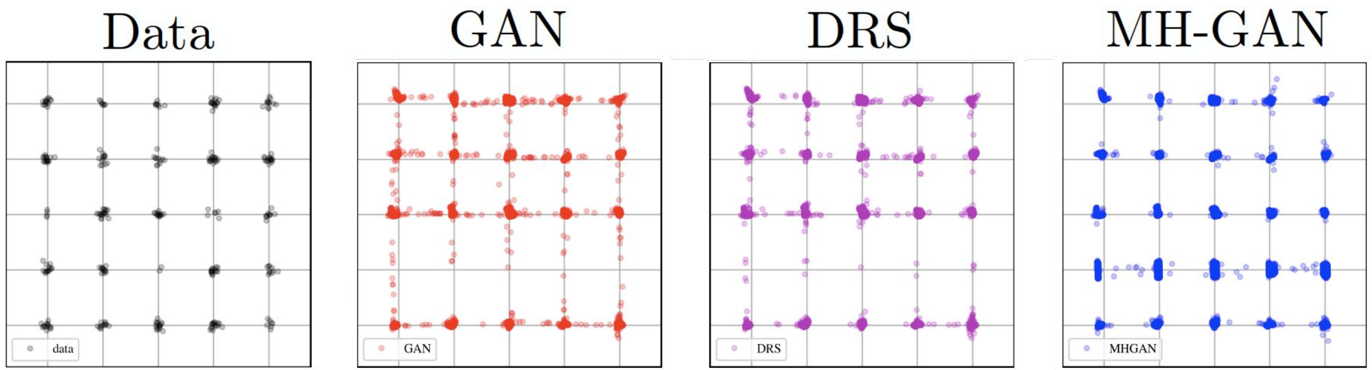


Dropped modes!

[1] Azadi et al. ICLR 2019.

MH recovers the true data distribution

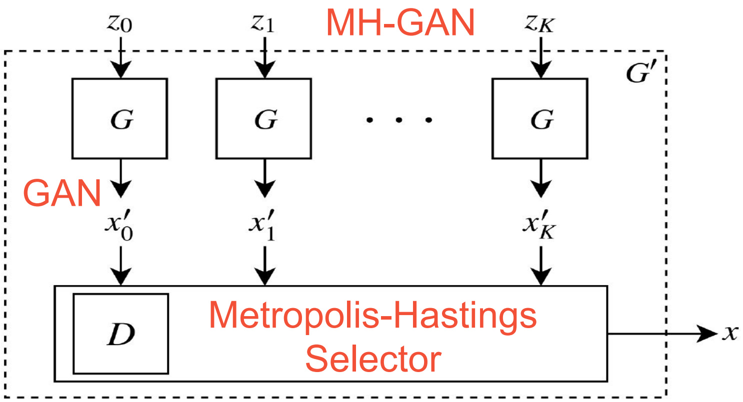
"Mixture of Gaussians" dataset [1]



[1] Azadi et al. ICLR 2019.

Motivation for Metropolis-Hastings

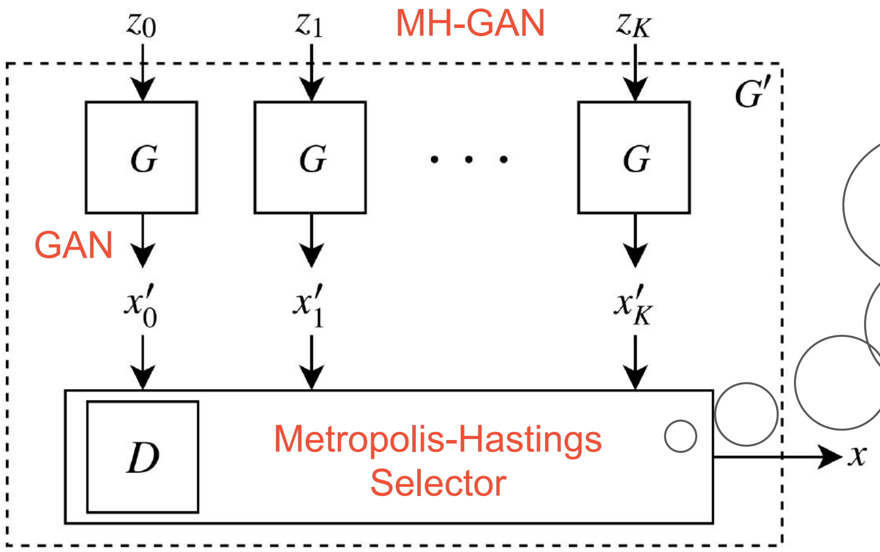
- Use MCMC *independence sampler*: sample p_D from G
- Given a perfect D and imperfect G, still obtain exact samples from true data distribution!
- Avoid densities in MCMC, just need *density ratios*:



$$D(\mathbf{x}) = \frac{p_D(\mathbf{x})}{p_D(\mathbf{x}) + p_G(\mathbf{x})}$$

$$\frac{p_D(\mathbf{x})}{p_G(\mathbf{x})} = \frac{D(\mathbf{x})}{1 - D(\mathbf{x})}$$

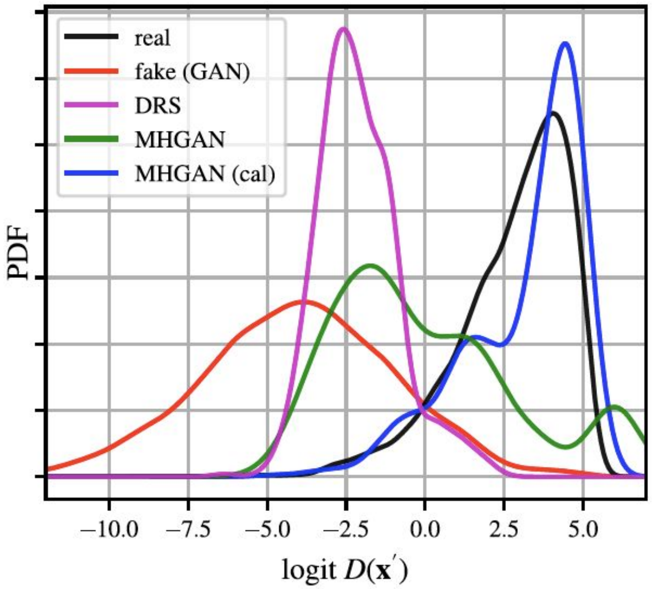
Metropolis-Hastings as a post-processing step for generators



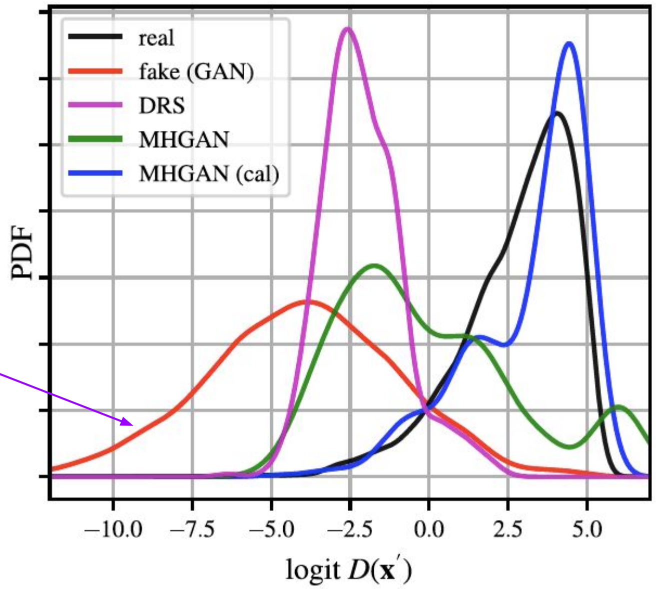
$$\frac{p_D}{p_G} = \frac{1}{D^{-1} - 1}$$

$$\implies \alpha(\mathbf{x}', \mathbf{x}_k) = \min\left(1, \frac{D(\mathbf{x}_k)^{-1} - 1}{D(\mathbf{x}')^{-1} - 1}\right)$$

MH recovers the correct score distribution

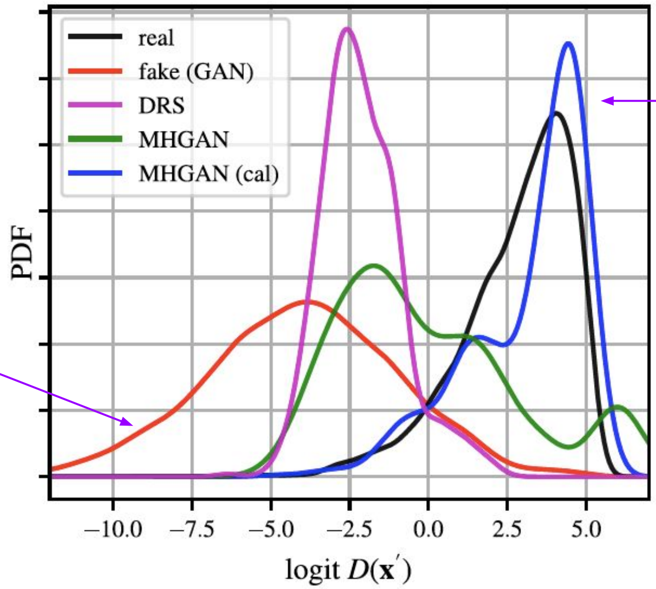


MH recovers the correct score distribution



Discriminator gives different scores to fakes

MH recovers the correct score distribution



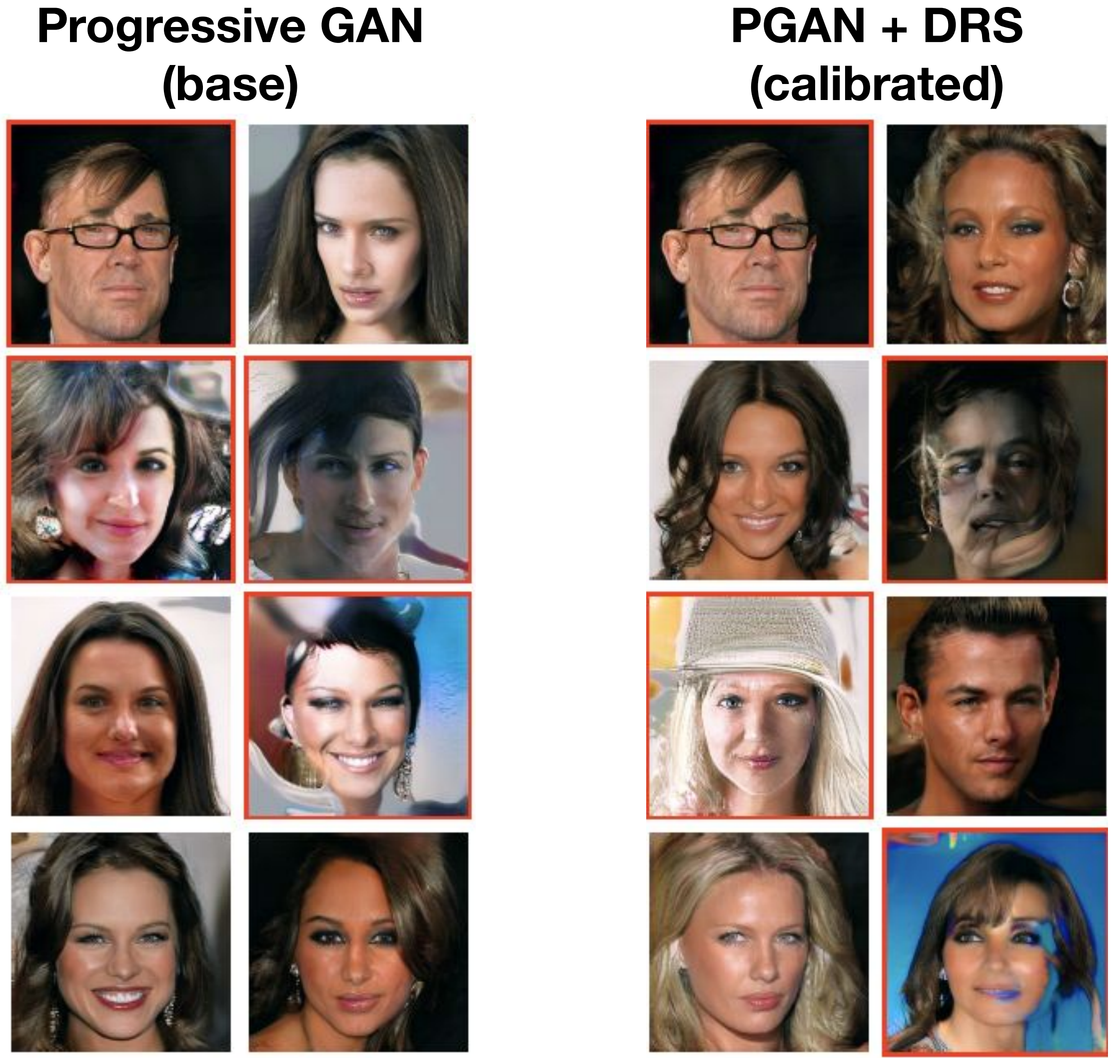
Discriminator gives different scores to fakes

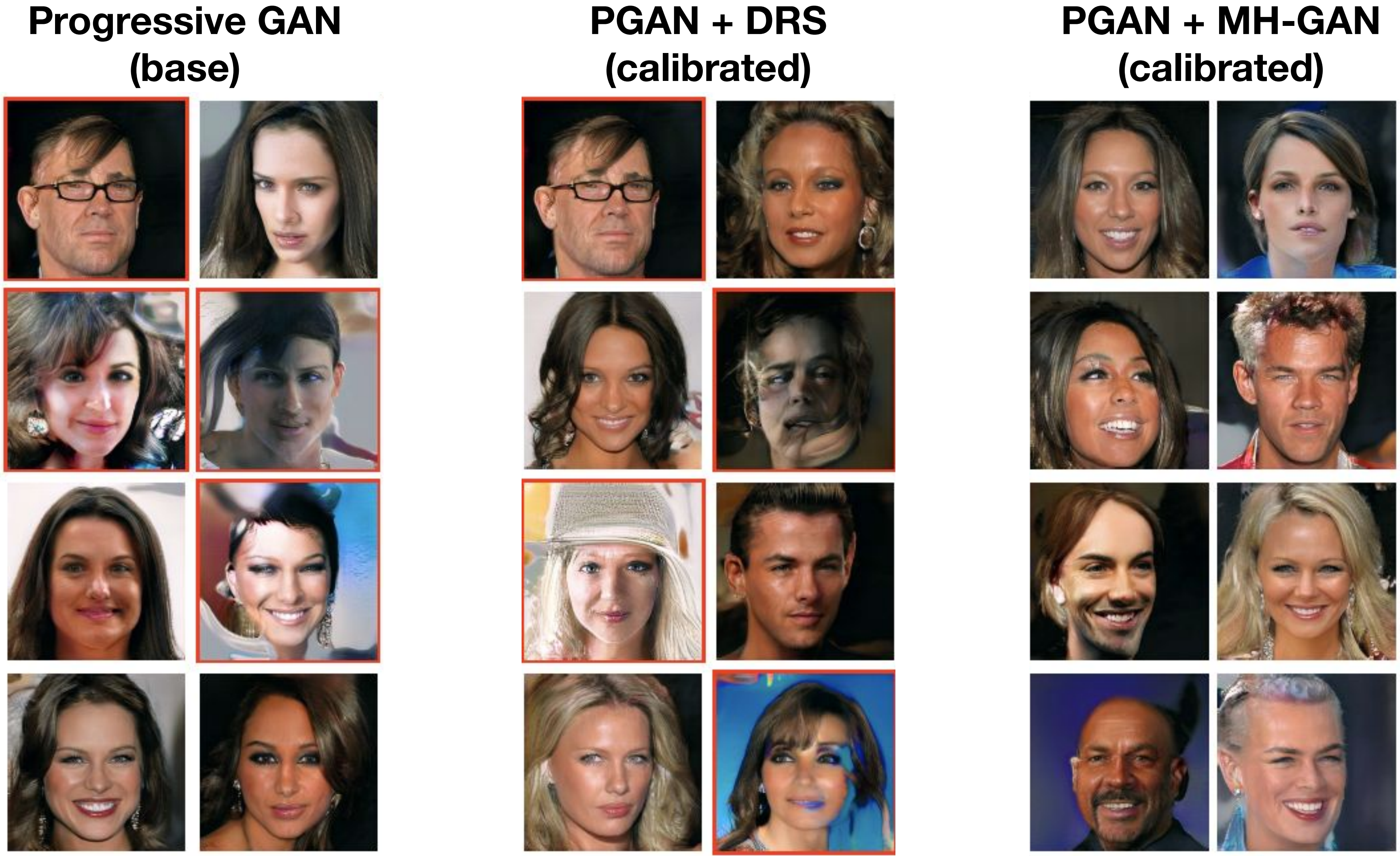
Score distribution now matches real data

Also… sample images

Progressive GAN (base)

Progressive GAN (base)


Progressive GAN
(base)
PGAN + DRS
(calibrated)


Progressive GAN
(base)
PGAN + DRS
(calibrated)
PGAN + MH-GAN
(calibrated)

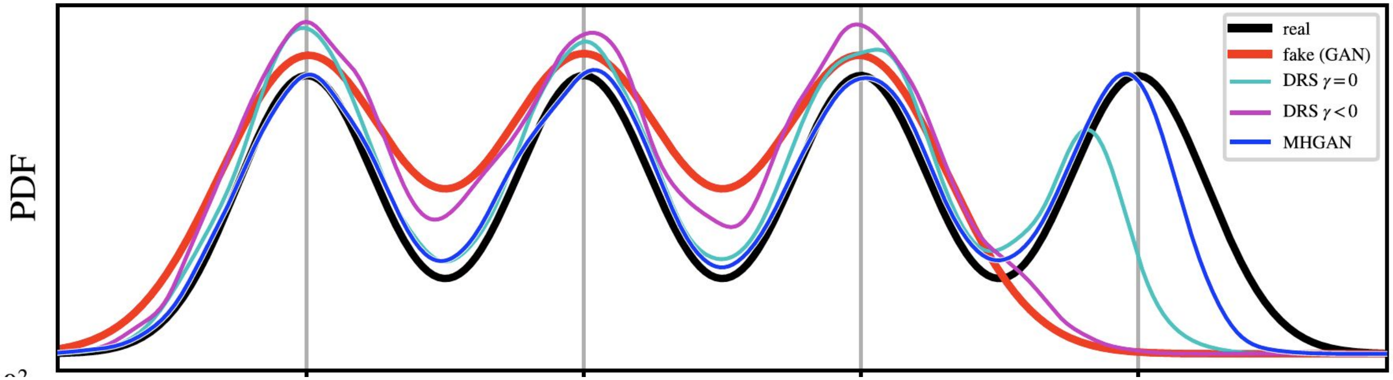
Metropolis-Hastings GAN

Ryan Turner, Jane Hung, Eric Frank, Yunus Saatci, Jason Yosinski

Poster #201

https://github.com/uber-research/metropolis-hastings-gans

MH recovers the true data distribution



1) 1D mixture of 4 Gaussians, missing one mixture