Metropolis-Hastings GAN

Uber AI Labs

Ryan Turner, Jane Hung, Yunus Saatci, Jason Yosinski

Generative Adversarial Networks

- Have two neural networks, G and D, competing against each other.
- G attempts to fool D by generating datapoints which are, from D's perspective, indistinguishable from real datapoints.
- D in turn simply tries to classify samples coming from G as fake and actual datapoints as real.
- If this adversarial game converges, it can generate amazingly realistic samples.

Learning Objective:

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

High resolution samples generated by *Karras et. al, arXiv 1710.10196*

p_D could be closer to the real distribution, but we need a way to sample from it. We can draw samples from this distribution using sampling methods.

Sampling

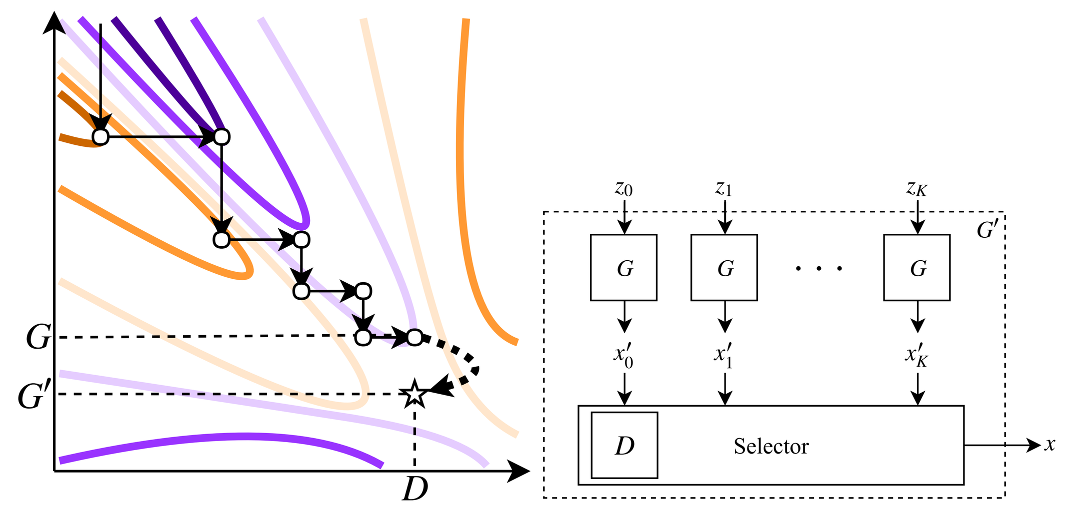
Two sampling methods are rejection sampling and Markov Chain Monte Carlo (MCMC). They can both be used as a post-processing step to improve the generator output. DRS uses rejection sampling, while our MH-GAN uses the Metropolis-Hastings MCMC approach.

Rejection sampling decides whether to accept or reject a sample from the generator. A sample is accepted with probability $p_D/(Mp_G)$, where M is the upper bound for the ratio p_D/p_G over all possible samples.

Finding M is a challenge, and choosing it empirically as DRS does carries the risk of using a very inaccurate acceptance probability. Even given a good M, rejection sampling suffers from a large number of sample rejections before first accept because of the high-dimensionality of the sampling space. To get around the sample waste problem, DRS has an additional γ heuristic to shift the discriminator scores, making the model sample from a distribution different from the real data even when D is perfect.

We use MCMC instead, which was invented precisely as a replacement for rejection sampling in higher dimensions. The MCMC class of methods sample from a possibly complicated probability distribution by taking multiple samples from a proposal distribution. The Metropolis-Hastings (MH) algorithm is a particular MCMC method that involves taking K samples from the proposed distribution (i.e., the generator) and choosing one sample from the K by sequentially deciding whether to accept the current sample or keep the previously chosen sample based on an acceptance rule

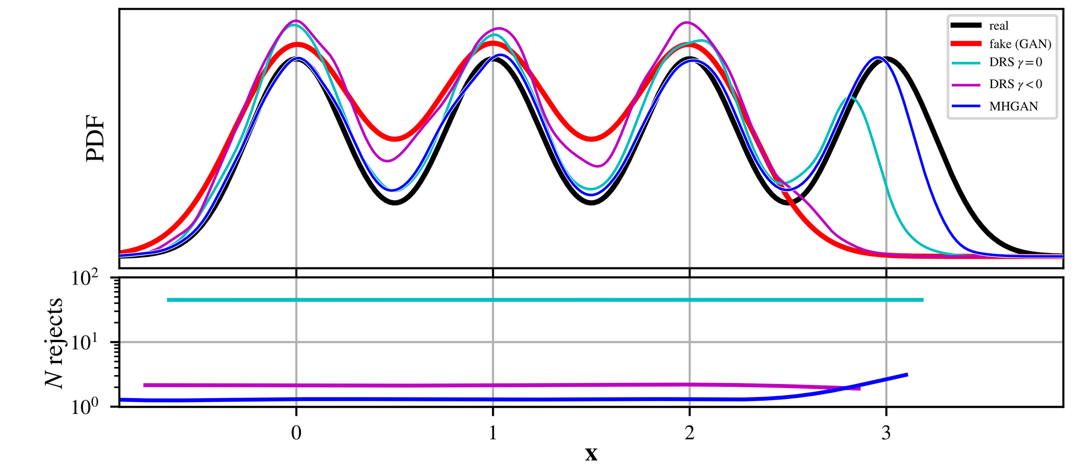
MH-GANs



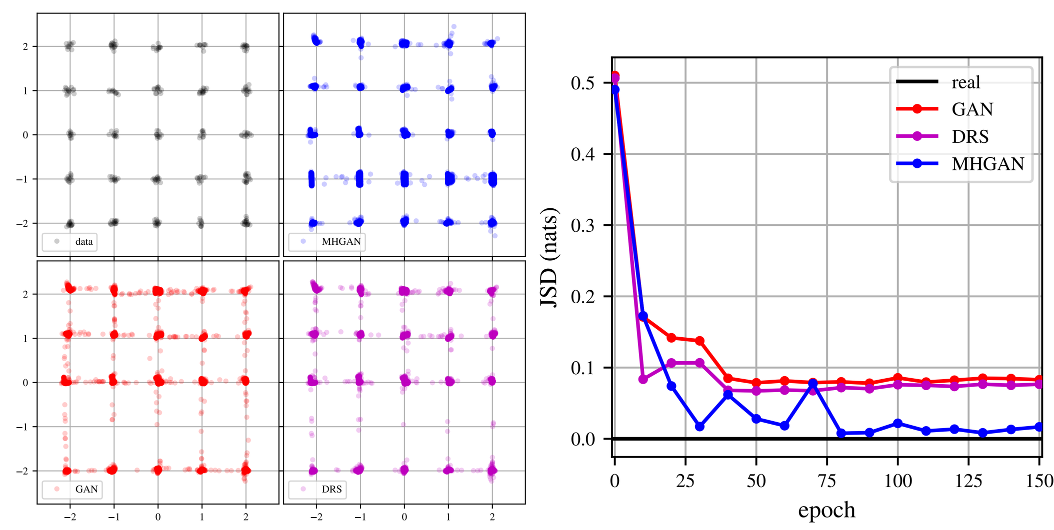
- Normally in GAN training, the generator moves towards the minimum of the value function (orange) while the discriminator moves towards the maximum (purple). Training stops at a point (D, G) with perfect D and imperfect G.
- With an imperfect G plus information from the discriminator, we can obtain a perfect generator G'; then we jump from (D, G) to (D, G').
- G' works as follows: Noise samples are drawn independently K times and used to generate the chain that the MH selector is applied to. Independent chains are used to obtain multiple samples from G'.
- MH selector uses discriminator score D to calculate acceptance probability

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

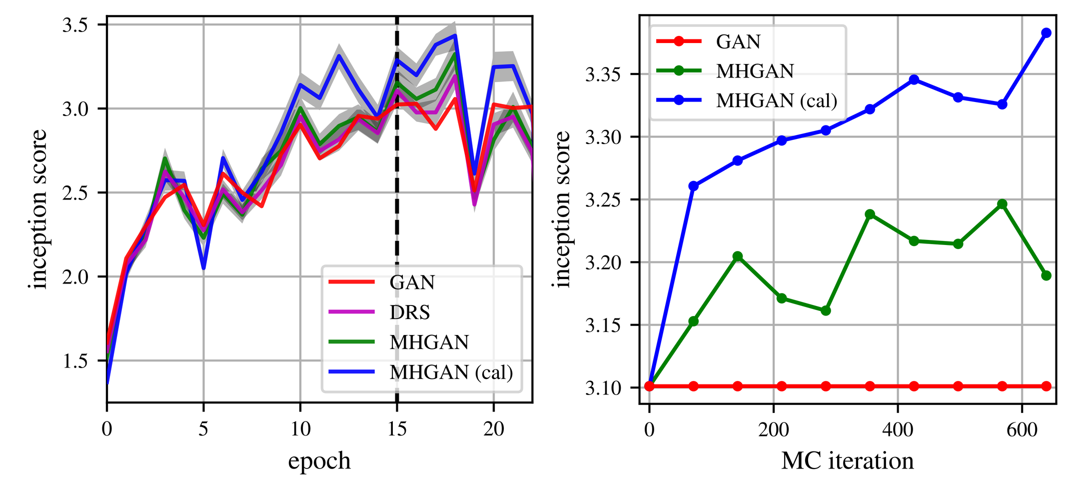
Results on Synthetic Data



The real data is a univariate mixture of 4 Gaussians, and the density of the generator p_G shows the common GAN pathology of missing one of the modes. DRS without γ shift and MH-GAN are able to recover the missing mode, but DRS with γ shift (the default used in their paper) cannot. However, DRS without γ shift increases the number of samples needed before a single accept by an order of magnitude.



Results on Benchmark Datasets



Code

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