

Generative Ratio Matching Networks

Akash Srivastava^{*,1,2}, Kai Xu^{*,3}, Michael U. Gutmann³, Charles Sutton^{3,4,5}



* equal contributions

¹MIT-IBM Watson AI Lab ²IBM Research ³University of Edinburgh ⁴Google AI

⁵Alan Turing Institute

Introduction

Adversarial Generative Models(GANs, MMD-GANs)

-  can generate high-dimensional data such as natural images.
-  are very difficult to train due to the saddle-point optimization problem

GRaM is a *stable* learning algorithm for *implicit* deep generative models that does **not** involve a saddle-point optimization problem and therefore is easy to train 🎉

Overview

1. Learn a projection function (f_θ)
 - that projects the data (p_x) and the model (q_x) densities into a low-dimensional manifold which,
 - preserves the difference between this pair of densities.
 - We use the ratio ($r(x) = \frac{p_x}{q_x}$) of the two densities as the measure of this difference.
2. Train the model (G_γ) in the low-dimensional manifold
 - using the *Maximum Mean Discrepancy* criterion as it work very well in low dimensional data.

GRaM: the algorithm

1 Learn the manifold projection function $f_\theta(x)$ by minimising the squared difference between the pair of density ratios:

$$\begin{aligned} D(\theta) &= \int q_x(x) \left(\frac{p_x(x)}{q_x(x)} - \frac{\bar{p}(f_\theta(x))}{\bar{q}(f_\theta(x))} \right)^2 dx \\ &= C - \text{PD}(\bar{q}, \bar{p}) \end{aligned}$$

GRaM: the algorithm (continued)

2 Train the generator G_γ by minimizing the empirical estimator of MMD in the low-dimensional manifold,

$$\min_{\gamma} \left[\frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N k(f_{\theta}(x_i), f_{\theta}(x_{i'})) - \frac{2}{NM} \sum_{i=1}^N \sum_{j=1}^M k(f_{\theta}(x_i), f_{\theta}(G_{\gamma}(z_j))) \right. \\ \left. + \frac{1}{M^2} \sum_{j=1}^M \sum_{j'=1}^M k(f_{\theta}(G_{\gamma}(z_j)), f_{\theta}(G_{\gamma}(z_{j'}))) \right]$$

GRaM: the algorithm (continued)

Pearson divergence maximization and density ratio estimation

Monte Carlo approximation of PD,

$$\text{PD}(\bar{q}, \bar{p}) \approx \frac{1}{N} \sum_{i=1}^N \left(\frac{\bar{p}(f_{\theta}(x_i))}{\bar{q}(f_{\theta}(x_i))} \right)^2 - 1$$

where $x_i^q \sim q_x$.

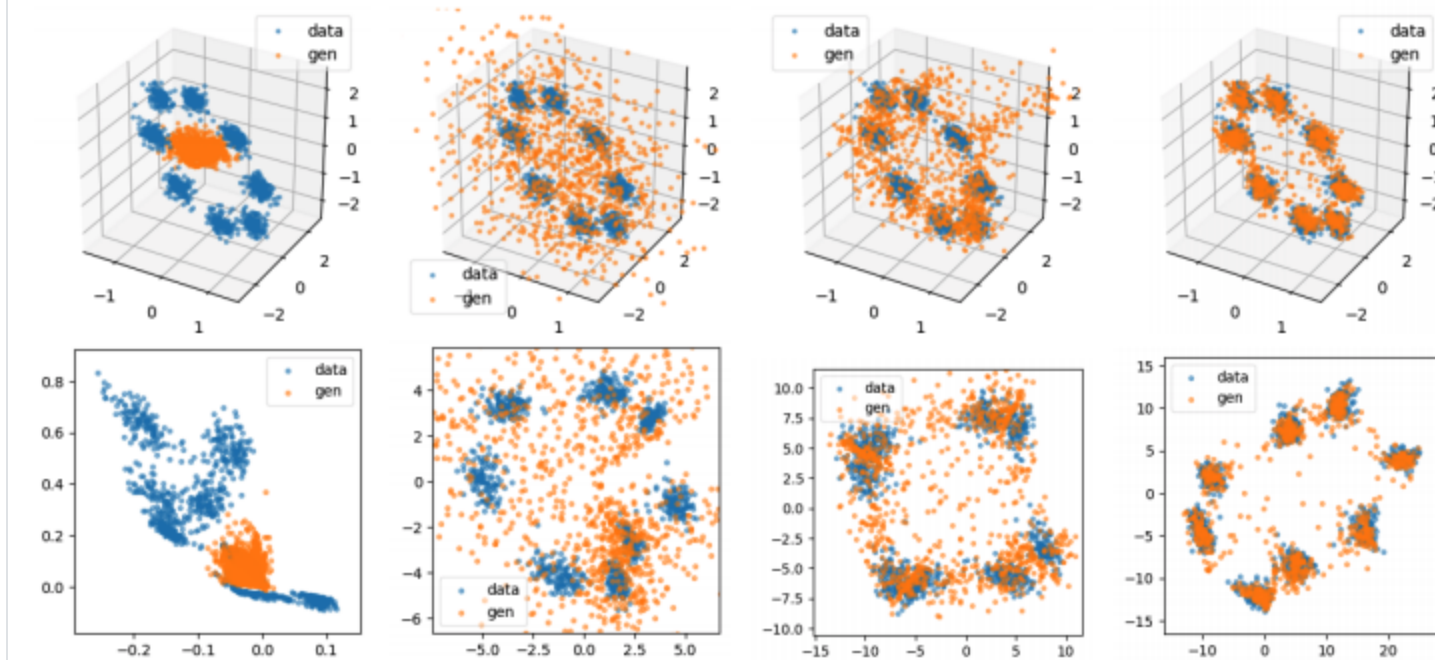
We use a MMD based density ratio estimator (Sugiyama et al., 2012) under the fixed-design setup: $\hat{r}_q = \mathbf{K}_{q,q}^{-1} \mathbf{K}_{q,p} \mathbf{1}$.

- $\mathbf{K}_{q,q}$ and $\mathbf{K}_{q,p}$ are Gram matrices defined by $[\mathbf{K}_{q,q}]_{i,j} = k(f_{\theta}(x_i^q), f_{\theta}(x_j^q))$ and $[\mathbf{K}_{q,p}]_{i,j} = k(f_{\theta}(x_i^q), f_{\theta}(x_j^p))$.

How do GRAM-nets compare to other methods

GAN	MMD-net	MMD-GAN	GRAM-net
<p>$z \sim p_z$ $\downarrow \gamma$ x^q $\downarrow \theta$ \mathcal{L}_γ</p> <p>$x^p \sim p_x$ $\downarrow \theta$ \mathcal{L}_θ</p> <p>θ (arrow from x^q to \mathcal{L}_θ)</p>	<p>$z \sim p_z$ $\downarrow \gamma$ x^q $\rightarrow \mathbf{K}$ \swarrow \mathcal{L}_γ</p> <p>$x^p \sim p_x$ \downarrow \mathbf{K}</p>	<p>$z \sim p_z$ $\downarrow \gamma$ x^q $\downarrow \theta$ $f_\theta(x^q)$ $\rightarrow \mathbf{K}$ \swarrow \mathcal{L}_γ</p> <p>$x^p \sim p_x$ $\downarrow \theta$ $f_\theta(x^p)$ \downarrow \mathbf{K} \downarrow \mathcal{L}_θ</p> <p>θ (arrow from x^q to $f_\theta(x^q)$)</p>	<p>$z \sim p_z$ $\downarrow \gamma$ x^q $\downarrow \theta$ $f_\theta(x^q)$ $\rightarrow \mathbf{K}$ \swarrow \mathcal{L}_γ</p> <p>$x^p \sim p_x$ $\downarrow \theta$ $f_\theta(x^p)$ \downarrow \mathbf{K} \downarrow \mathcal{L}_θ</p> <p>θ (arrow from x^q to $f_\theta(x^q)$)</p>

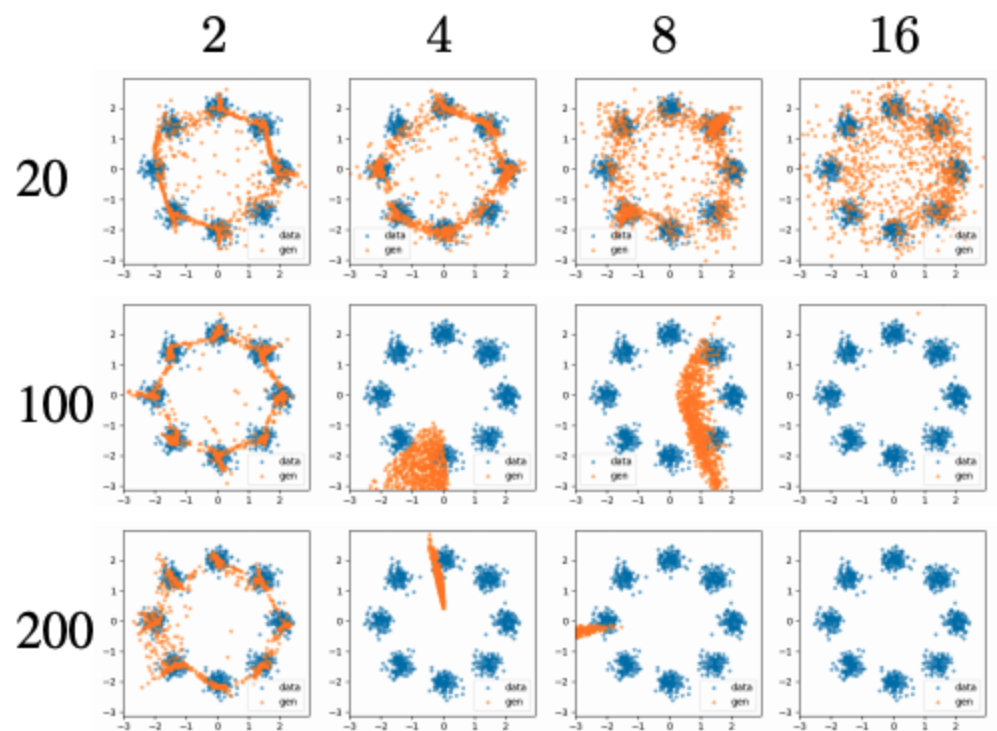
Illustration of GRAM training



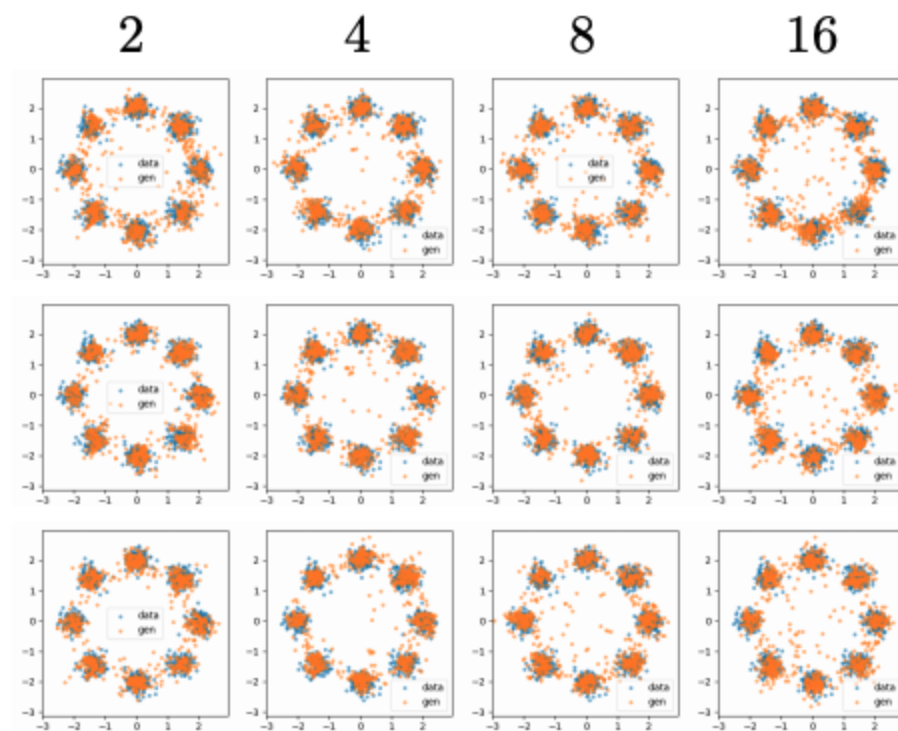
Blue: data
Orange: samples

Top: original
Bottom: projected

Evaluations: the stability of models



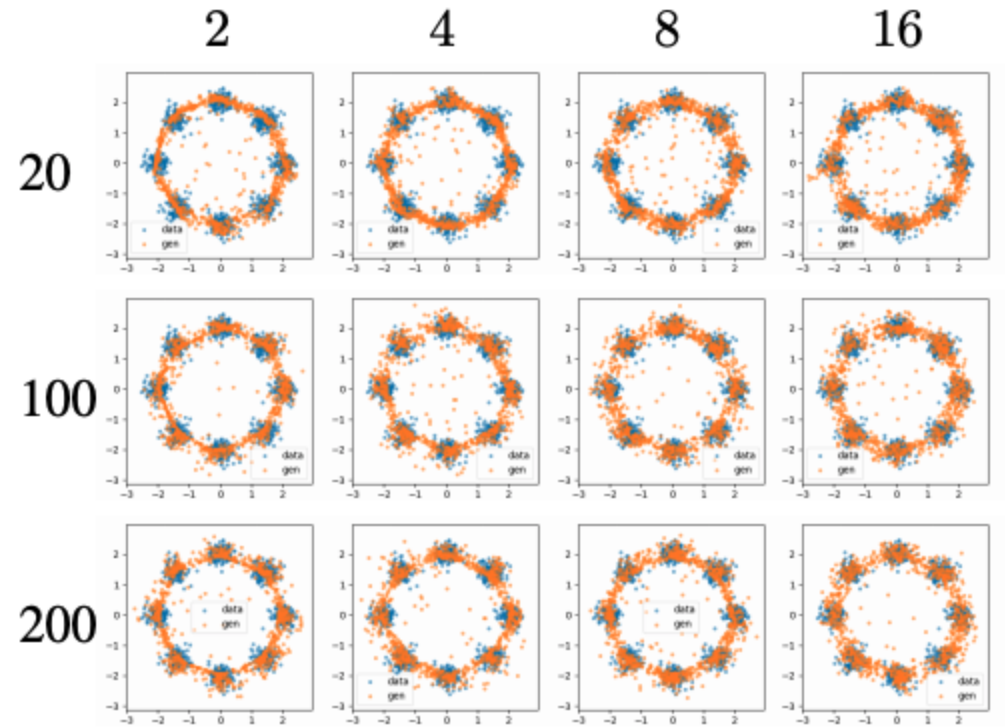
(a) GAN



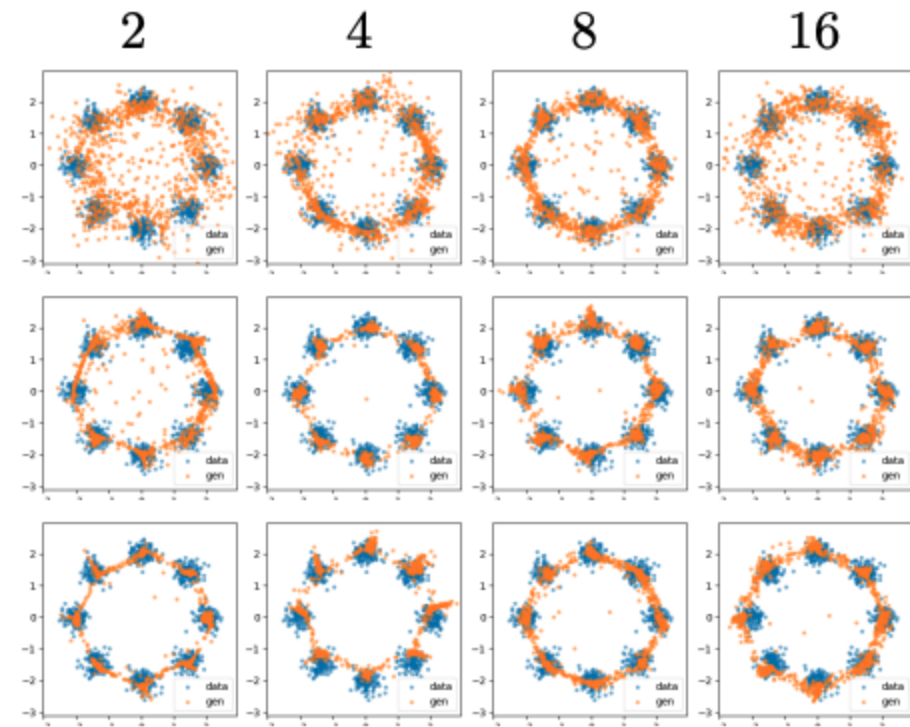
(b) GRAM-net

x-axis = noise dimension and y-axis = generator layer size

Evaluations: the stability of models (continued)



(a) MMD-nets



(b) MMD-GANs

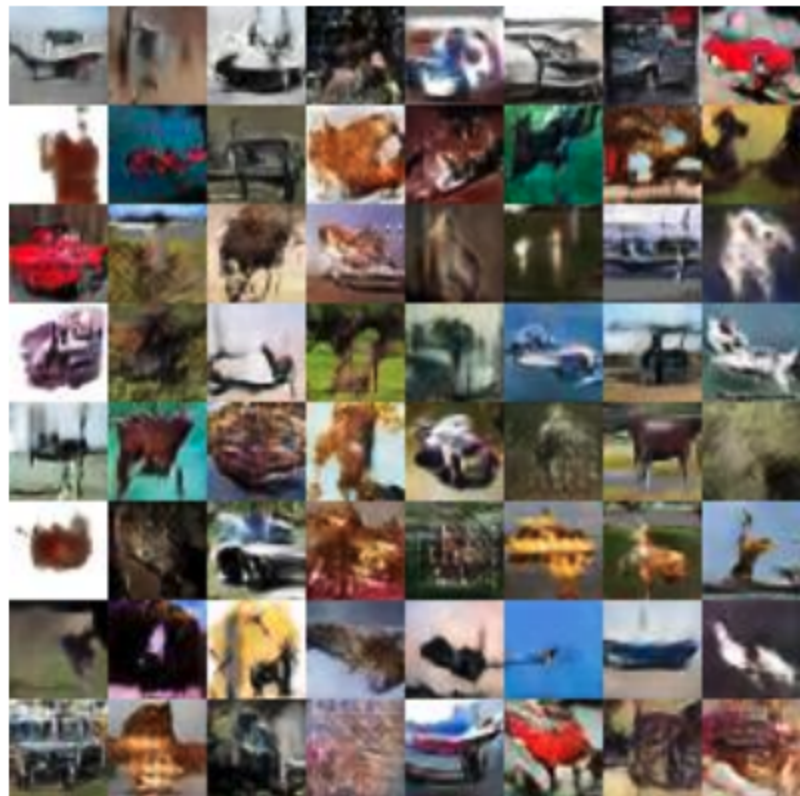
x-axis = noise dimension and y-axis = generator layer size

Quantitative results: sample quality

Table 1: Sample quality (measured by FID; lower is better) of GRAM-nets compared to GANs.

Arch.	Dataset	MMD-GAN	GAN	GRAM-net
DCGAN	Cifar10	40.00 ± 0.56	26.82 ± 0.49	24.85 ± 0.94
Weaker	Cifar10	210.85 ± 8.92	31.64 ± 2.10	24.82 ± 0.62
DCGAN	CelebA	41.105 ± 1.42	30.97 ± 5.32	27.04 ± 4.24

Qualitative results: random samples



(a) CIFAR10



(b) CelebA

The end

Extra slides to follow...

Density ratio estimation via (infinite) moment matching

Maximum mean discrepancy

$$\text{MMD}_{\mathcal{F}}(p, q) = \sup_{f \in \mathcal{F}} (\mathbb{E}_p[f(x)] - \mathbb{E}_q[f(x)])$$

Gretton et al. (2012) show that it is sufficient to choose \mathcal{F} to be a unit ball in an reproducing kernel Hilbert space \mathcal{R} with a characteristic kernel k .

Using this definition of MMD, the density ratio estimator $r(x)$ can be derived as the solution to

$$\min_{r \in \mathcal{R}} \left\| \int k(x; \cdot) p(x) dx - \int k(x; \cdot) r(x) q(x) dx \right\|_{\mathcal{R}}^2.$$

Generator training

The generator G_γ is trained by minimizing the empirical estimator of MMD

$$\min_{\gamma} \left[\frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N k(f_{\theta}(x_i), f_{\theta}(x_{i'})) - \frac{2}{NM} \sum_{i=1}^N \sum_{j=1}^M k(f_{\theta}(x_i), f_{\theta}(G_{\gamma}(z_j))) \right. \\ \left. + \frac{1}{M^2} \sum_{j=1}^M \sum_{j'=1}^M k(f_{\theta}(G_{\gamma}(z_j)), f_{\theta}(G_{\gamma}(z_{j'}))) \right]$$

with respect to its parameters γ .