

# Generative Ratio Matching Networks

Akash Srivastava<sup>\*,1,2</sup>, Kai Xu<sup>\*,3</sup>, Michael U. Gutmann<sup>3</sup>, Charles Sutton<sup>3,4,5</sup>



\* equal contributions

<sup>1</sup>MIT-IBM Watson AI Lab <sup>2</sup>IBM Research <sup>3</sup>University of Edinburgh <sup>4</sup>Google AI

<sup>5</sup>Alan 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

## Two steps in the training loop

1. Learn a projection function ( $f_\theta$ )
  - 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 ( $G_\gamma$ ) 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  $G_\gamma$  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  $[\mathbf{K}_{q,p}]_{i,j} = k(f_{\theta}(x_i^q), f_{\theta}(x_j^p))$ .

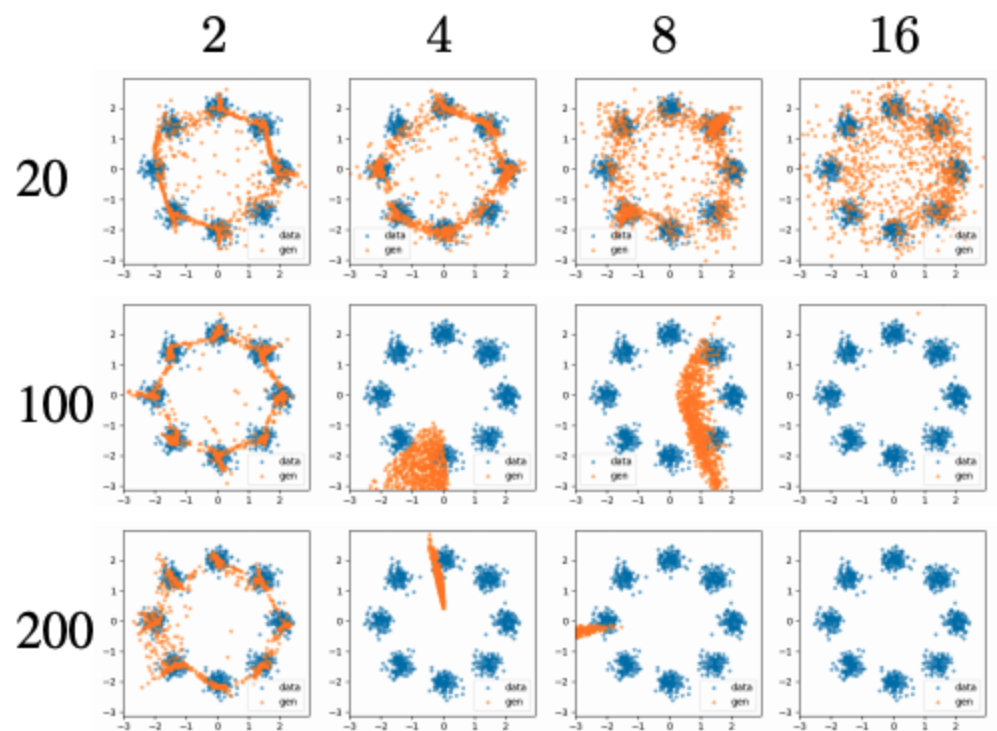
# How do GRAM-nets compare to other methods

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

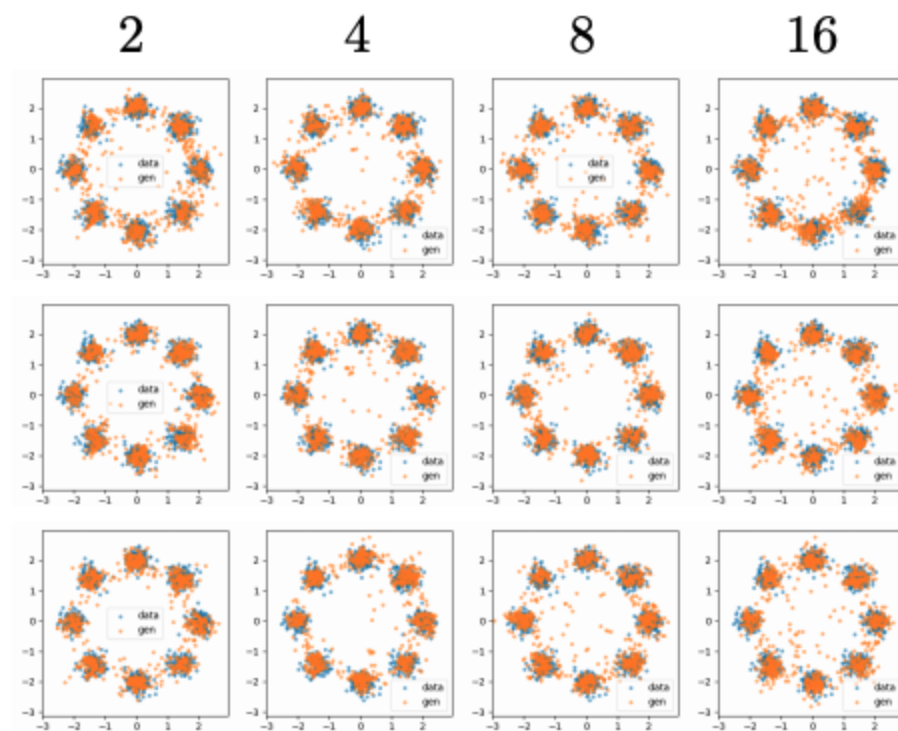
Top: original  
Bottom: projected



# Evaluations: the stability of models



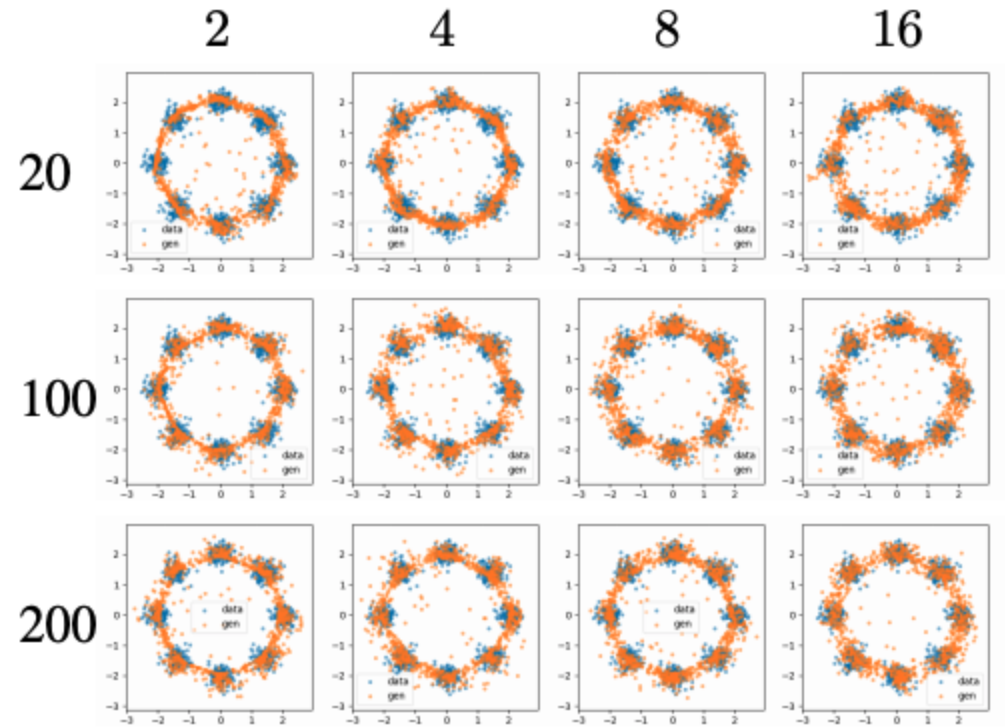
(a) GAN



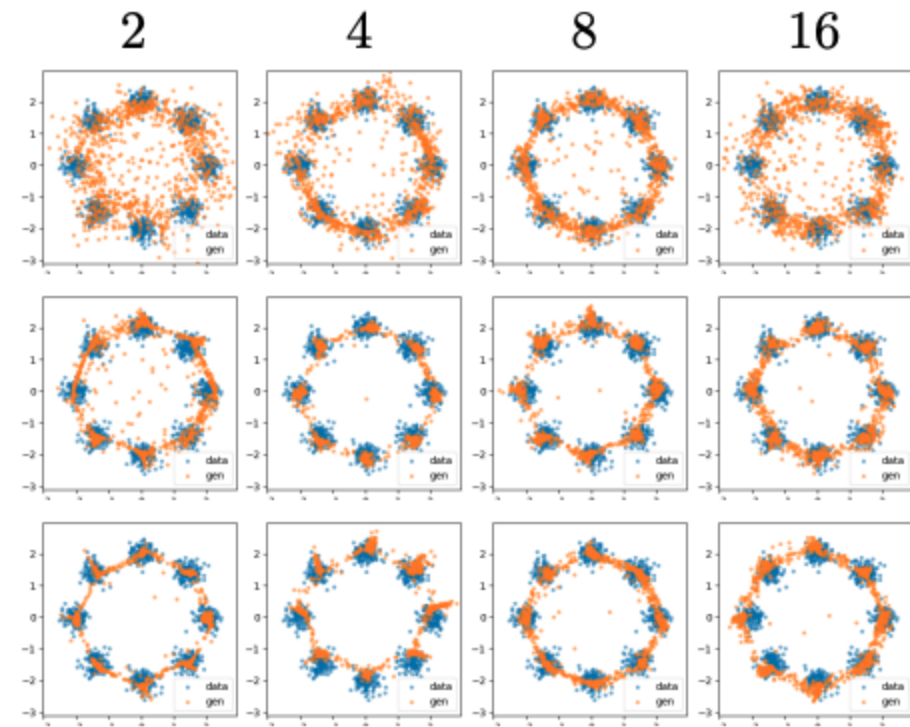
(b) GRAM-net

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

# Evaluations: the stability of models (continued)



(a) MMD-nets



(b) MMD-GANs

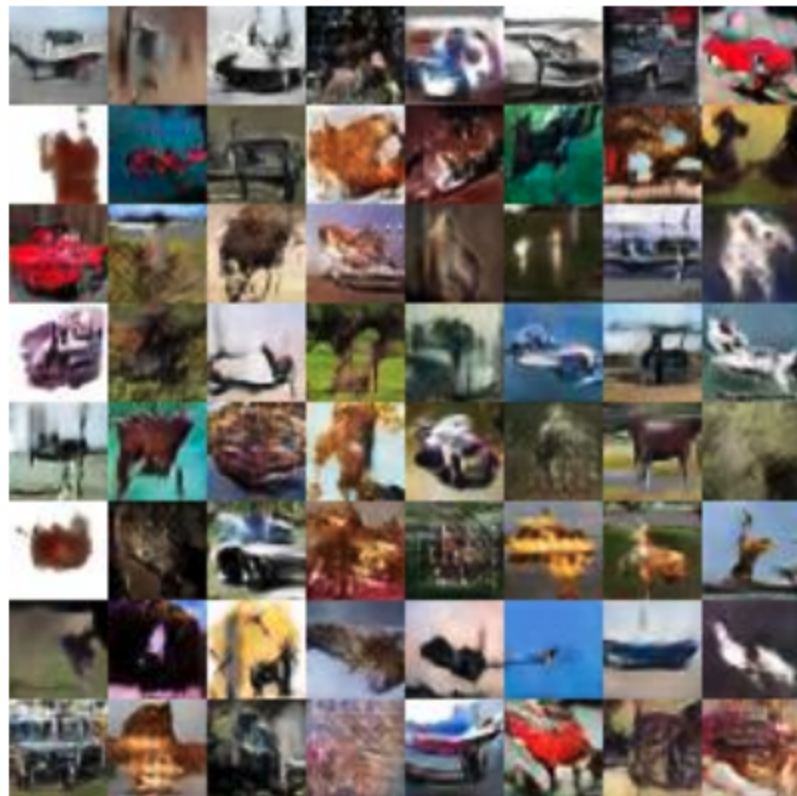
x-axis = noise dimension and y-axis = generator layer size

## 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	$40.00 \pm 0.56$	$26.82 \pm 0.49$	<b><math>24.85 \pm 0.94</math></b>
Weaker	Cifar10	$210.85 \pm 8.92$	$31.64 \pm 2.10$	<b><math>24.82 \pm 0.62</math></b>
DCGAN	CelebA	$41.105 \pm 1.42$	$30.97 \pm 5.32$	<b><math>27.04 \pm 4.24</math></b>

## 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  $G_\gamma$  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  $\gamma$ .