The relativistic discriminator
a key element missing from standard GAN

Alexia Jolicoeur-Martineau

alexia.jolicoeur-martineau@mail.mcgill.ca

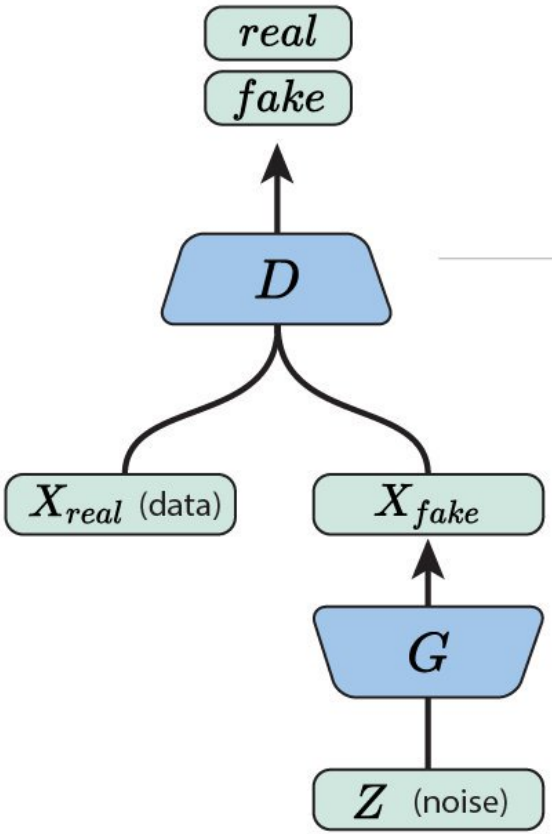
April 25, 2019

DREAM BIG
Developmental Research in Environmental Adversity, Mental health, Biological susceptibility and Gender

Jewish General Hospital
Lady Davis Institute for Medical Research

Standard GAN

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.

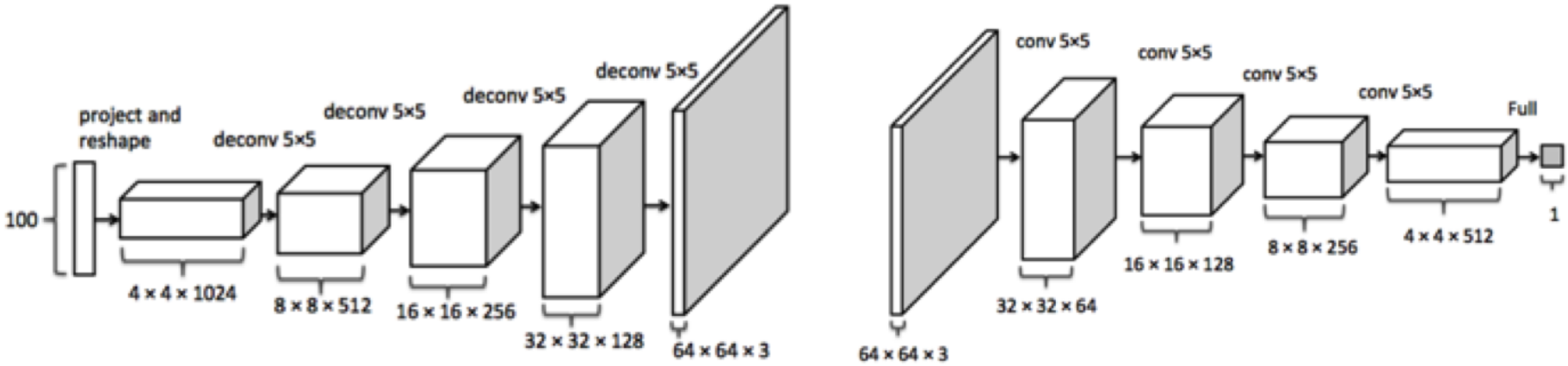


The **discriminator** tries to distinguish genuine data from forgeries created by the generator.

The **generator** turns random noise into immitations of the data, in an attempt to fool the discriminator.

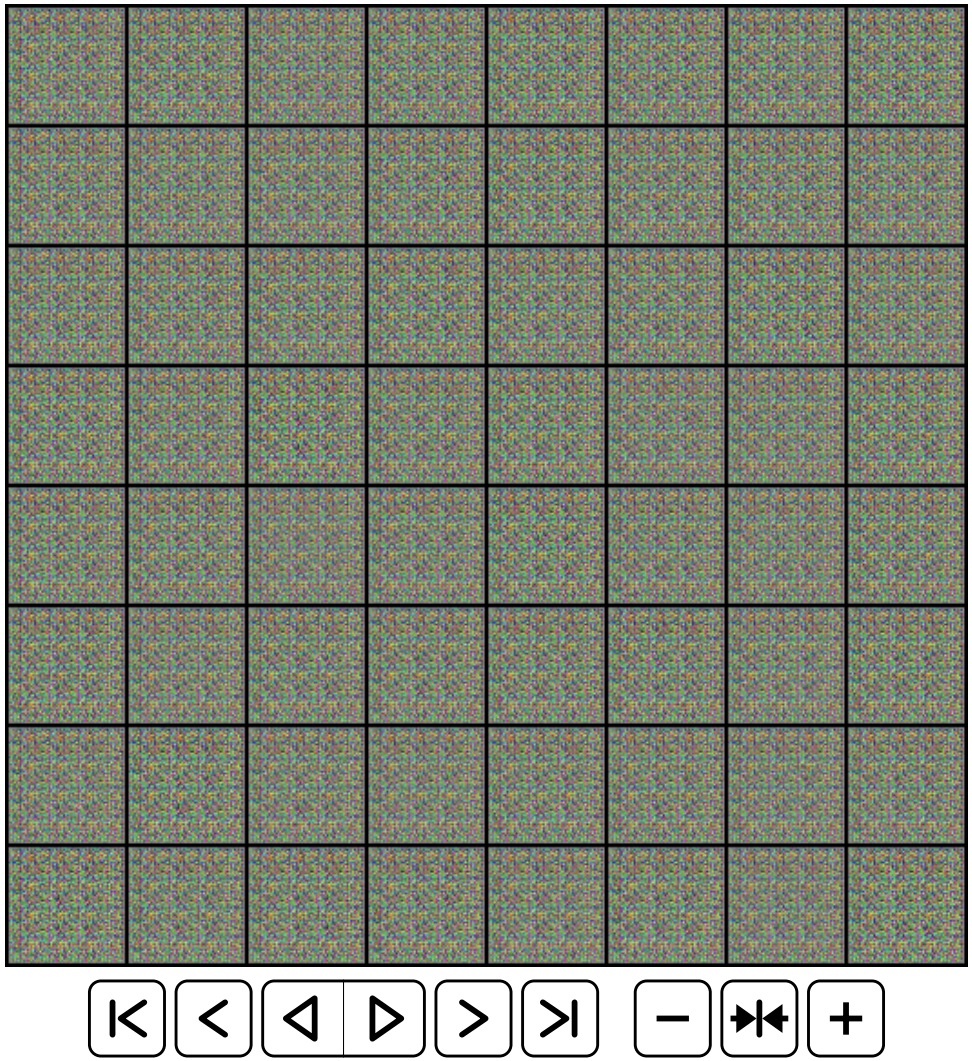
Image from Chris Olah (https://twitter.com/ch402/status/793911806494261248)

G and D with images



- We start with 100 random $z_i \sim Normal(0, 1)$ variables (pure noise)
  - The generator G transform noise into an image with deconvolution layers (64x64x3 = an image of 64x64 pixels with 3 colors)
  - The discriminator D takes a single image and transform it into a single number (the probability that the image is real) using convolution layers

Standard GAN



Other uses than images

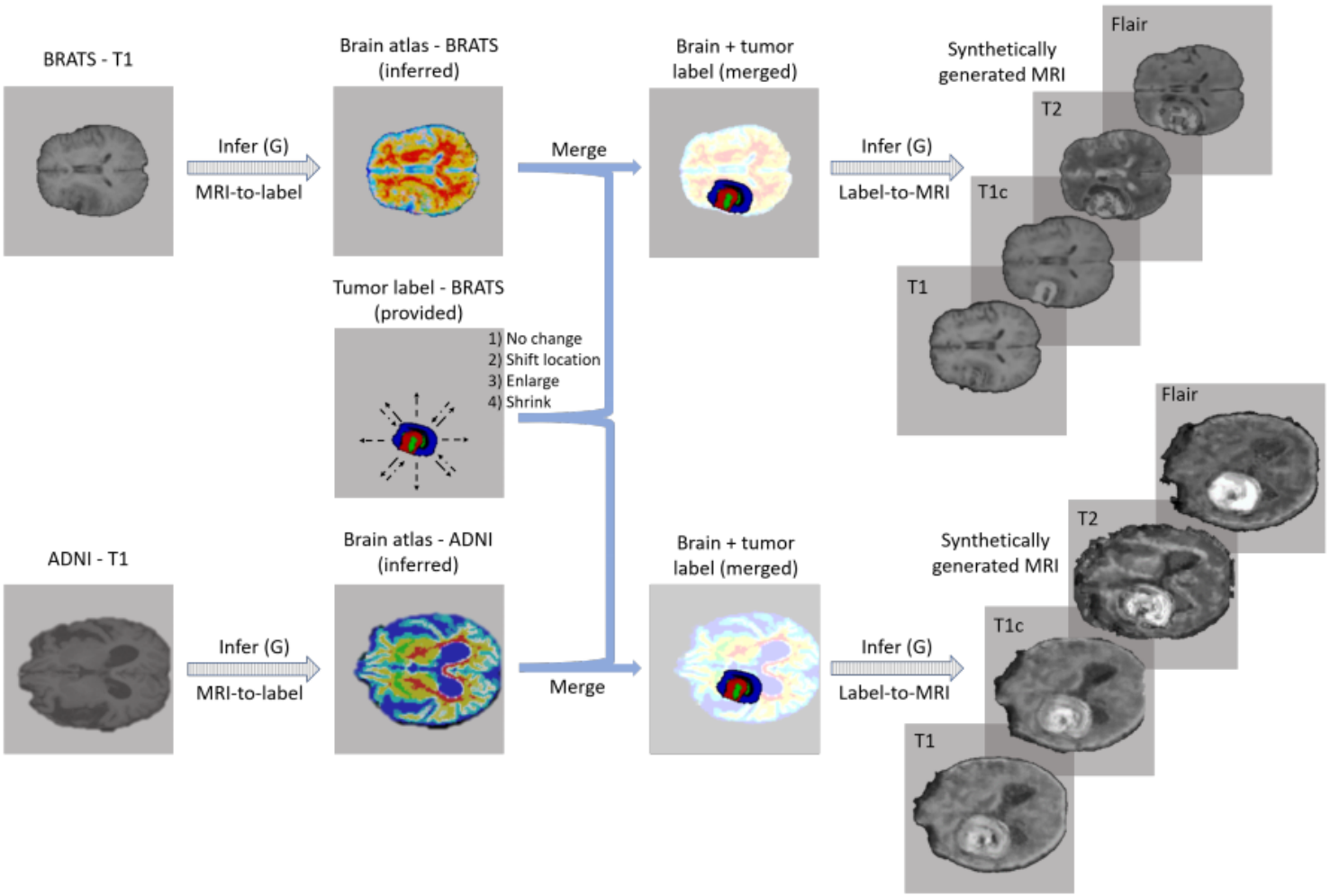
I focus mostly on images, but you can generate any kind of data. Furthermore, we can do other things:

- Generate data conditionally on a label (e.g. generate an image of cat vs dog, generate a healthy MRI vs one with a tumor)

Examples of use: Conditional GAN

BigGAN, Arxiv: 1809.11096.

Examples of use: Conditional GAN



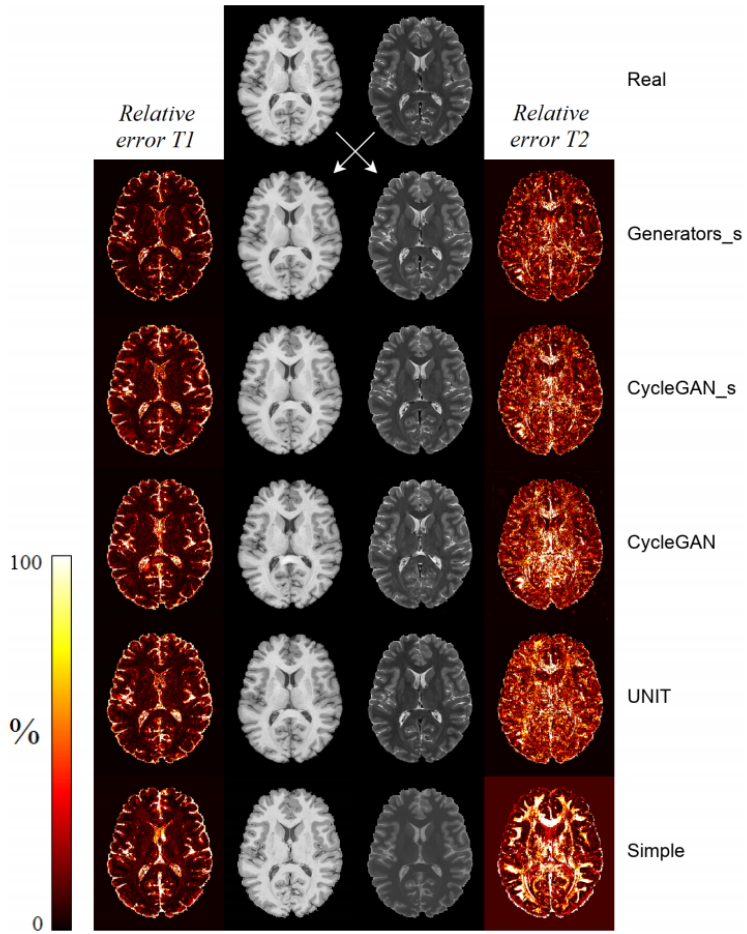
https://arxiv.org/pdf/1807.10225.pdf

Other uses than images

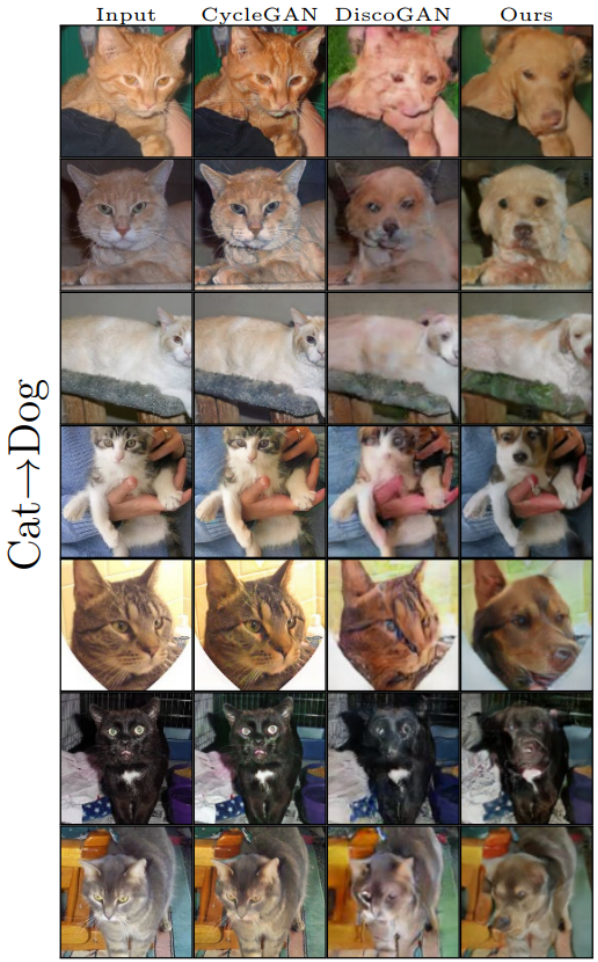
I focus mostly on images, but you can generate any kind of data. Furthermore, we can do other things:

- Generate data conditionally on a label (e.g. generate an image of cat vs dog, generate a healthy MRI vs one with a tumor)
- Domain translation (data from domain A transformed into its equivalent in domain B)

Examples of use: Domain translation



(a) MRI T1 to T2



(b) Cat to Dog

Figure: Arxiv: 1806.07777 and 1808.04325

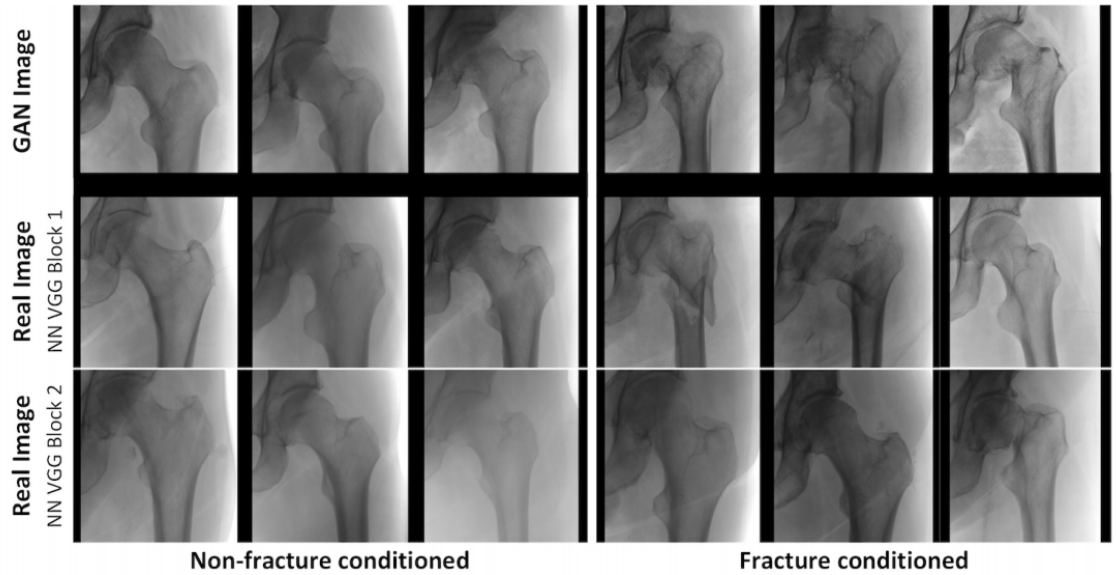
Other uses than images

I focus mostly on images, but you can generate any kind of data. Furthermore, we can do other things:

- Generate data conditionally on a label (e.g. generate an image of cat vs dog, generate a healthy MRI vs one with a tumor)
- Domain translation (data from domain A transformed into its equivalent in domain B)
- Generate data and then use it to augment your small dataset so that your sample size is more reasonable for your analyses (data augmentation)

Examples of use: Data augmentation

Example case: You have many images of fractures, but very few of certain types. You train a generator and have it produce images of the very rare types of fracture.
1) You augment your dataset with these images to reduce the class imbalances.
2) You use these images to help train doctors better since now you can give them more examples of these rare fractures.



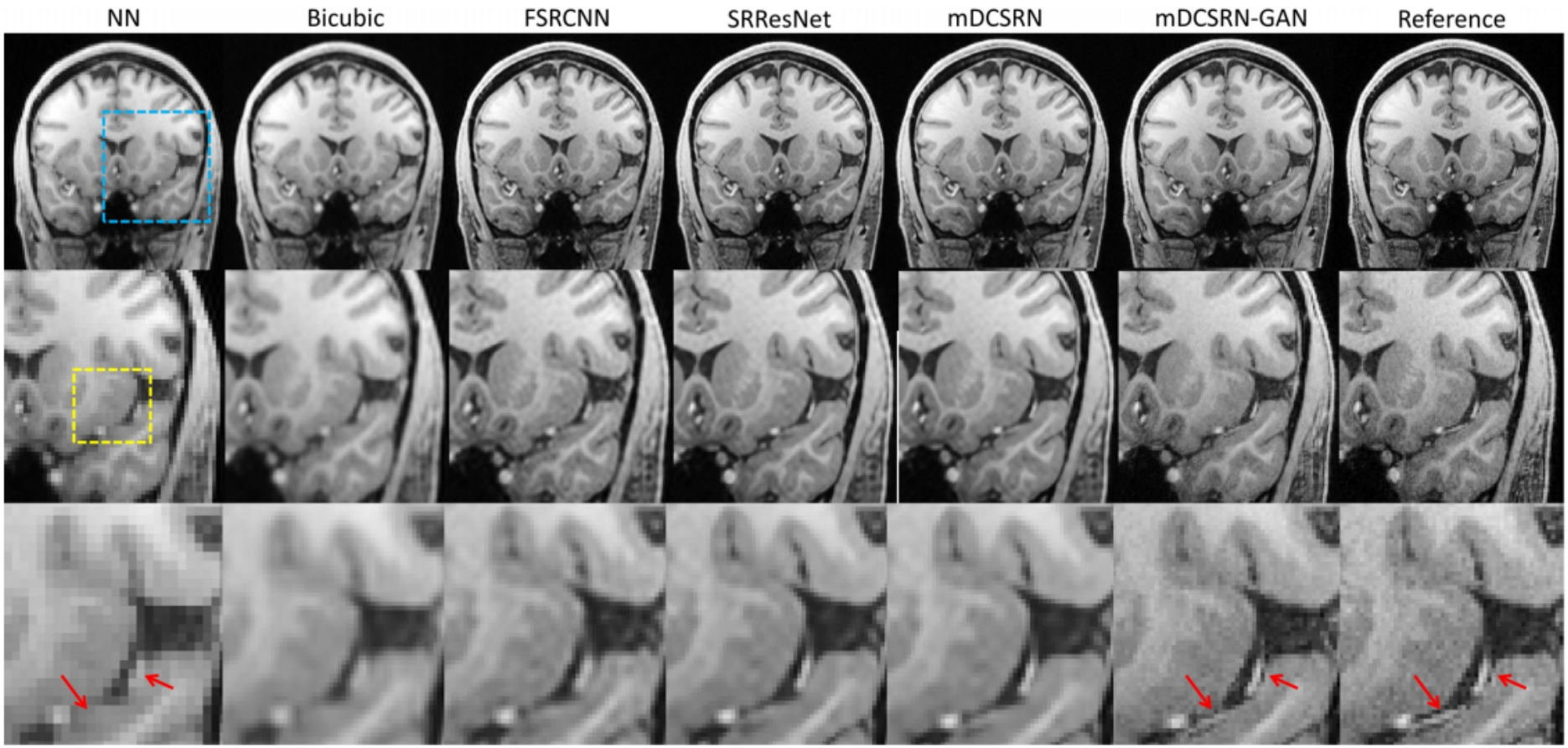
Arxiv: 1812.01547

Other uses than images

I focus mostly on images, but you can generate any kind of data. Furthermore, we can do other things:

- Generate data conditionally on a label (e.g. generate an image of cat vs dog, generate a healthy MRI vs one with a tumor)
- Domain translation (data from domain A transformed into its equivalent in domain B)
- Generate data and then use it to augment your small dataset so that your sample size is more reasonable for your analyses (data augmentation)
- Super-resolution (Enhance)

Examples of use: Enhance!



https://arxiv.org/ftp/arxiv/papers/1803/1803.01417.pdf

Simple explanation

- x is the image (real or fake)
- D is trained so $D(x_{real}) \to 1$ and $D(x_{fake}) \to 0$
- Therefore, $D(x) \approx P(x \text{ is real}) \in [0, 1]$

- How does the generator learn to generate realistic images?
- By fooling the discriminator into thinking that fake images are actually real images.
  i.e., G learns by making $D(x_{fake}) \to 1$

- When D cannot distinguish real from fake images ($D(x_{real}) = D(x_{fake}) = 1/2$), the generator has won the game.

Two formulations

Saturating GAN

$$\min_G \max_D \mathbb{E}_{x_r \sim \mathbb{P}} \left[f_1(D(x_r)) \right] + \mathbb{E}_{z \sim \mathbb{P}_z} \left[f_2(D(G(z))) \right] \tag{1}$$

Neat mathematically but leads to convergence problems.

Non-saturating GAN

$$\max_D \mathbb{E}_{x_r \sim \mathbb{P}} \left[f_1(D(x_r)) \right] + \mathbb{E}_{z \sim \mathbb{P}_z} \left[f_2(D(G(z))) \right] \tag{2}$$

$$\max_G \mathbb{E}_{z \sim \mathbb{P}_z} \left[f_1(D(G(z))) \right] \tag{3}$$

Can be interpreted as fooling D into thinking that fake images are real.

- f_1, f_2 are scalar-to-scalar functions
- $\mathbb{P}$ is the distribution of real data
- $\mathbb{P}_z \sim \mathcal{N}(0, 1)$ is the distribution of the noise
- $\mathbb{Q}$ is the distribution of fake data (formed by $G(z)$, $z \sim \mathbb{P}_z$)
- $D(x) = a(C(x))$, a is the activation function, $C(x) \in \mathbb{R}$ is the critic.

Divergences

Finding the optimal D is equivalent to estimating a divergence:

$$\max_D \mathbb{E}_{x_r \sim \mathbb{P}} \left[f_1(D(x_r)) \right] + \mathbb{E}_{z \sim \mathbb{P}_z} \left[f_2(D(G(z))) \right] \approx Div(\mathbb{P}||\mathbb{Q})$$

Standard GAN

$$D(x) = \text{sigmoid}(C(x))$$
$$f_1(y) = \log(y), f_2(y) = \log(1-y)$$
$$Div(\mathbb{P}||\mathbb{Q}) = JSD(\mathbb{P}||\mathbb{Q})$$

- Divergences are distances between probability distributions
  e.g., Div(domestic cats || wild cats) $<$ Div(domestic cats || fish).

Divergences

Finding the optimal D is equivalent to estimating a divergence:

$$\min_G \max_D \mathbb{E}_{x_r \sim \mathbb{P}} \left[f_1(D(x_r)) \right] + \mathbb{E}_{z \sim \mathbb{P}_z} \left[f_2(D(G(z))) \right] \approx \min_G Div(\mathbb{P} || \mathbb{Q})$$

Standard GAN

$$D(x) = \text{sigmoid}(C(x))$$
$$f_1(y) = \log(y), f_2(y) = \log(1 - y)$$
$$Div(\mathbb{P} || \mathbb{Q}) = JSD(\mathbb{P} || \mathbb{Q})$$

- Divergences are distances between probability distributions
  e.g., Div(domestic cats || wild cats) $<$ Div(domestic cats || fish).
- min/max GAN is like estimating and minimizing the divergence
  (kind of, but not really; see *GANs beyond divergence minimization*).

Divergences

Finding the optimal D is equivalent to estimating a divergence:

$$\min_G \max_D \mathbb{E}_{x_r \sim \mathbb{P}} [f_1(D(x_r))] + \mathbb{E}_{z \sim \mathbb{P}_z} [f_2(D(G(z)))] \approx \min_G Div(\mathbb{P}||\mathbb{Q})$$

Standard GAN

$$D(x) = \text{sigmoid}(C(x))$$
$$f_1(y) = \log(y), f_2(y) = \log(1-y)$$
$$Div(\mathbb{P}||\mathbb{Q}) = JSD(\mathbb{P}||\mathbb{Q})$$

- Divergences are distances between probability distributions
  e.g., Div(domestic cats || wild cats) $<$ Div(domestic cats || fish).
- min/max GAN is like estimating and minimizing the divergence
  (kind of, but not really; see *GANs beyond divergence minimization*).
- Saturating and non-saturating GANs converge to the same optimum

Problems with GANs

1. mode collapse: only generate certain modes of the data.
   - only generating grey cats
   - only generating from 10 of the 20 groups/clusters of data

Problems with GANs

1. mode collapse: only generate certain modes of the data.
   - only generating grey cats
   - only generating from 10 of the 20 groups/clusters of data
2. severe instability, especially in high dimension (high res images)

Problems with GANs

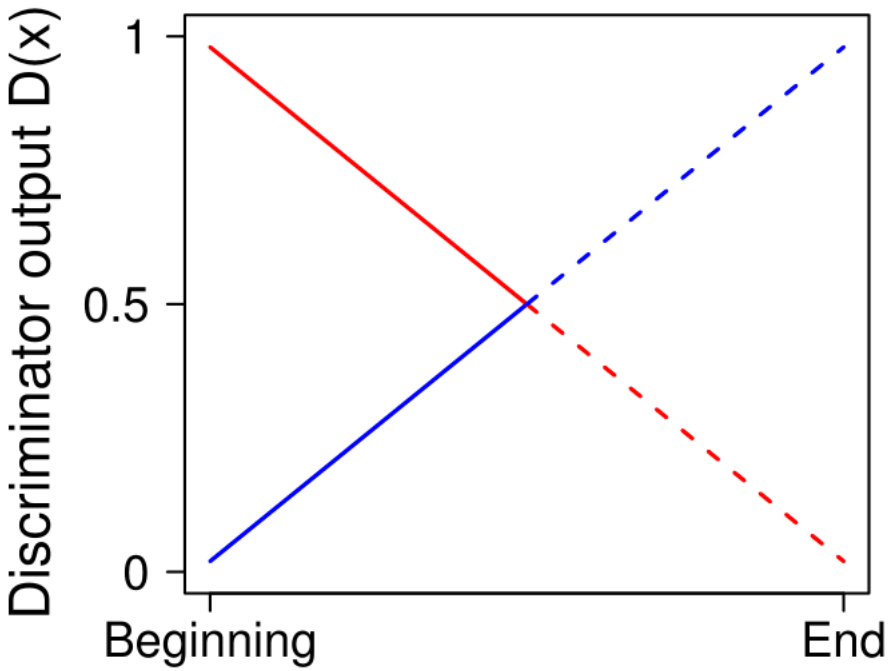
1. mode collapse: only generate certain modes of the data.
   - only generating grey cats
   - only generating from 10 of the 20 groups/clusters of data
2. severe instability, especially in high dimension (high res images)
3. D can easily become too good at distinguishing real from fake early in training. In SGAN this leads to vanishing gradients (with high-variance), which means that G stop learning.

Relativistic GANs improve 2 and 3 😸, but not 1 😿.

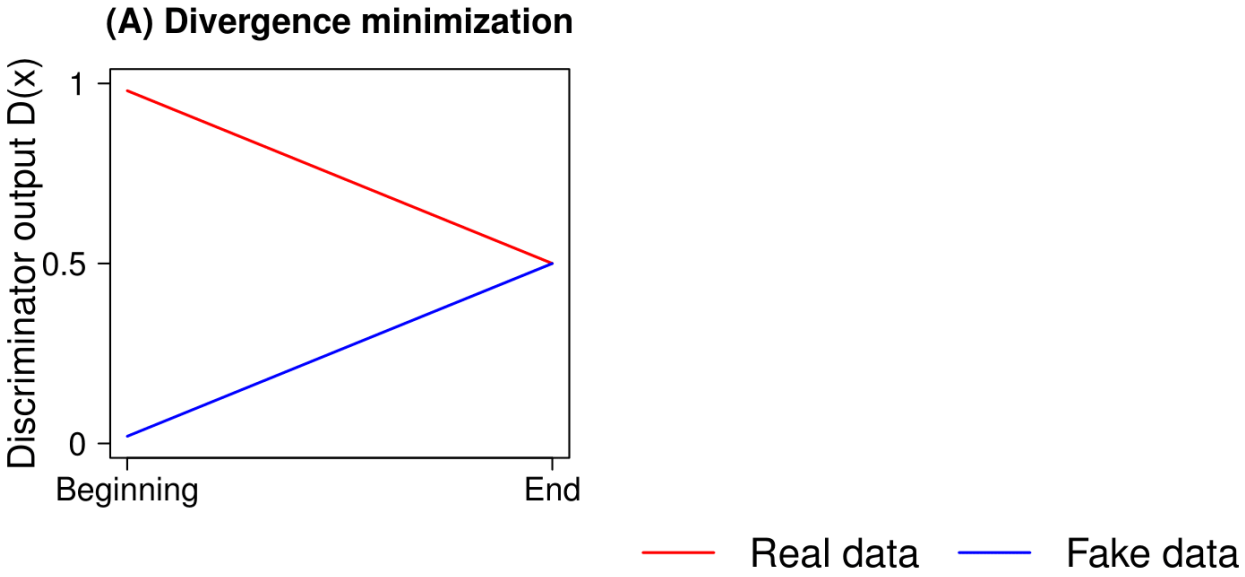
Key missing property

Key property missing from GANs:
Increasing $D(x_{fake})$ *should simultaneously* **decrease** $D(x_{real})$.

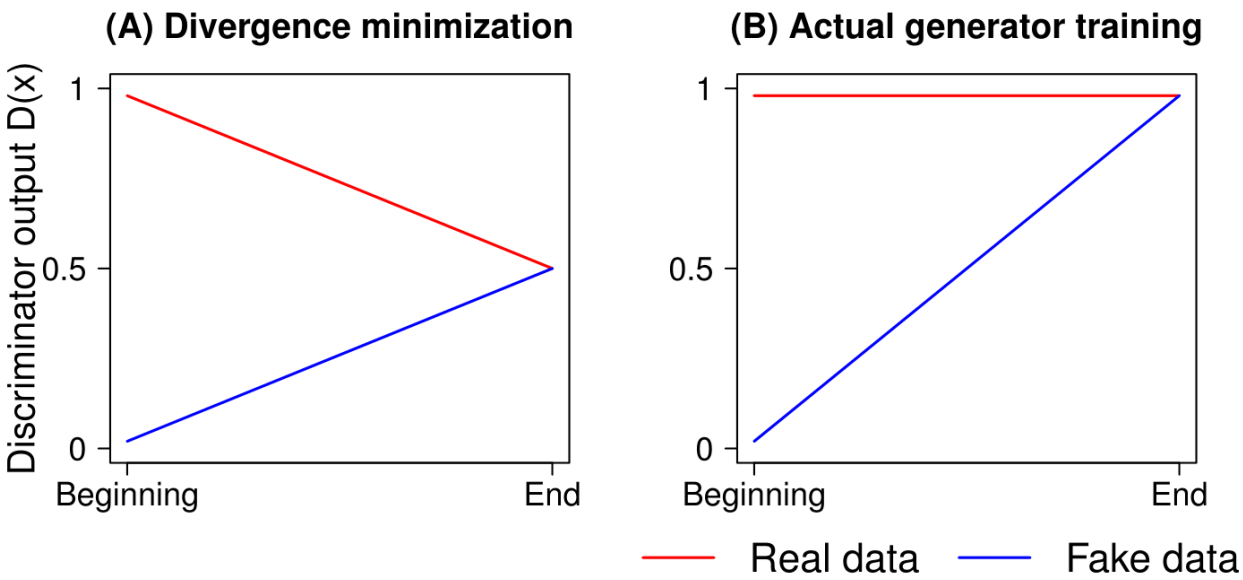
Less formally (for SGAN):
Increasing $P(x_{fake}$ *is real*$)$ *should simultaneously* **decrease** $P(x_{real}$ *is real*$)$.



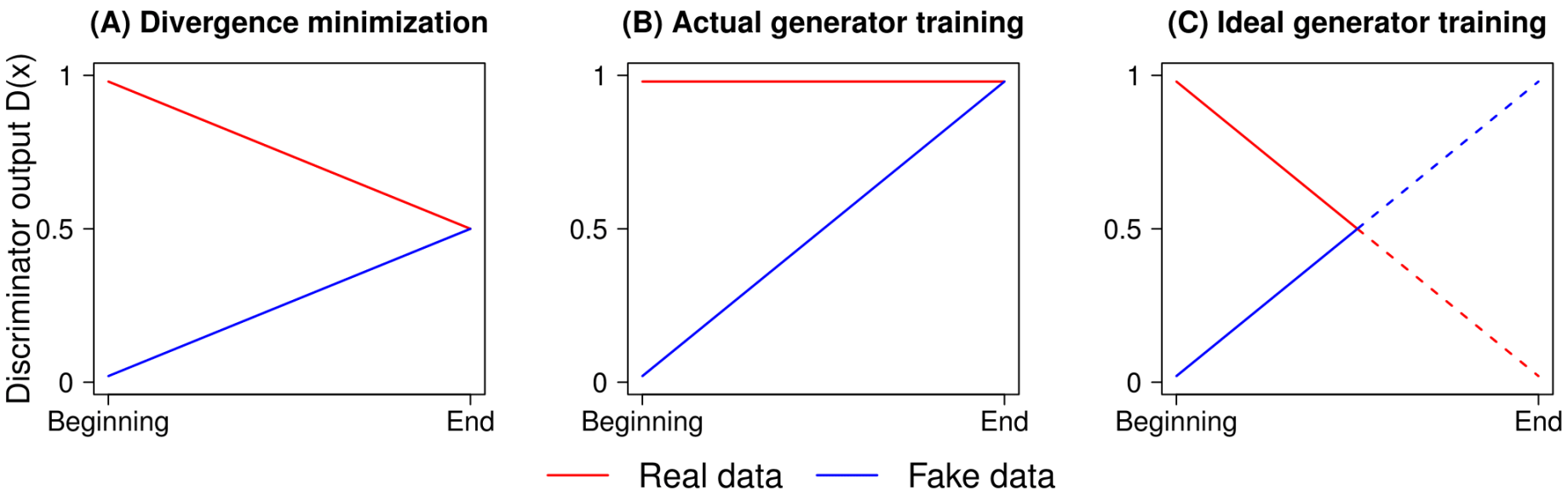
1) Divergence minimization



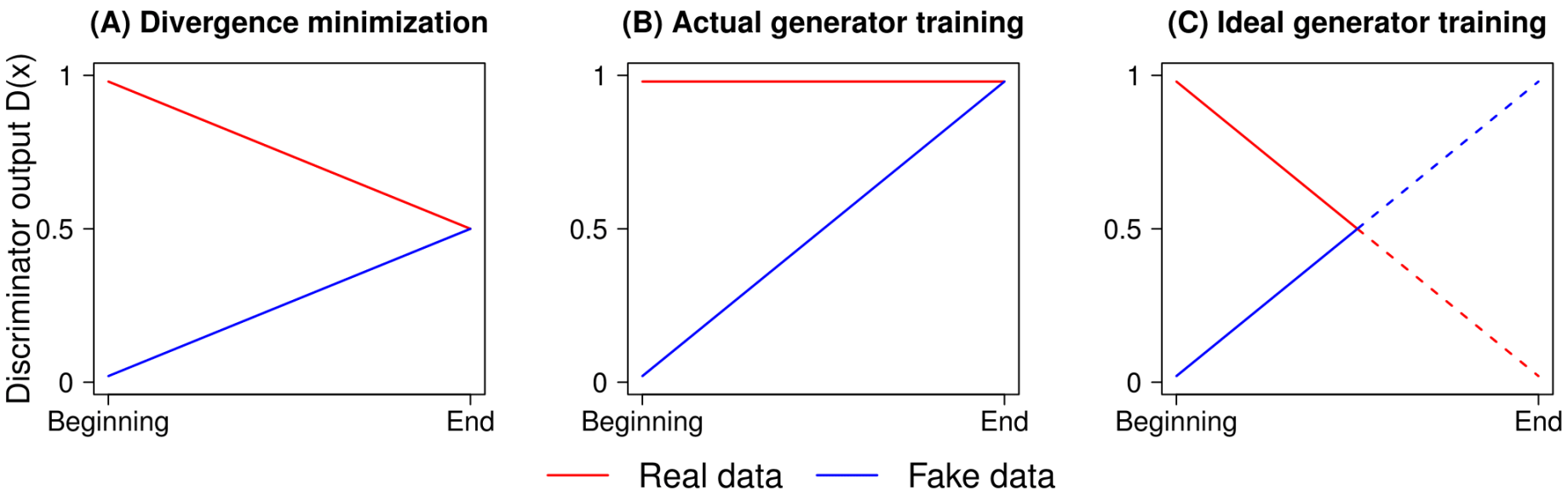
1) Divergence minimization



1) Divergence minimization



1) Divergence minimization



SGAN is not like divergence minimization 😿.

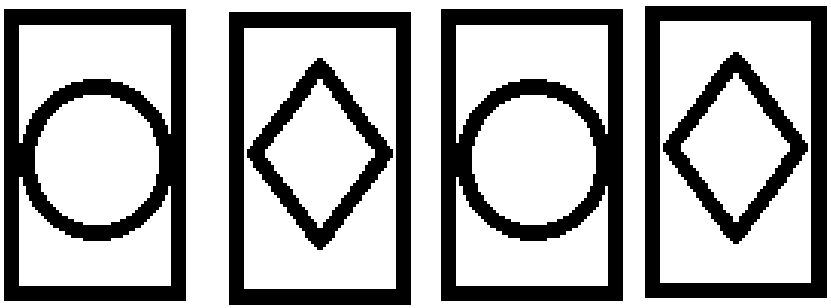
With the key missing property, it is like divergence minimization 😸.

2) Ignoring prior information

Imagine SGAN is a card game in real-life.

- Diamond cards are real samples
- mini-batch = 4 cards (2 fake, 2 real)
- You are D, you estimate $D(x)$=$P(x$ is real)

Let say that the current mini-batch is:



Human prediction

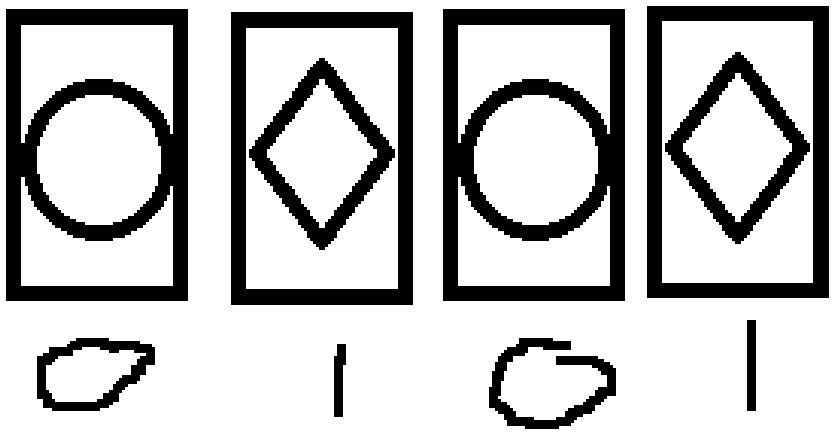
SGAN prediction

2) Ignoring prior information

Imagine SGAN is a card game in real-life.

- Diamond cards are real samples
- mini-batch = 4 cards (2 fake, 2 real)
- You are D, you estimate $D(x)$=$P(x$ is real)

Let say that the current mini-batch is:



Human prediction

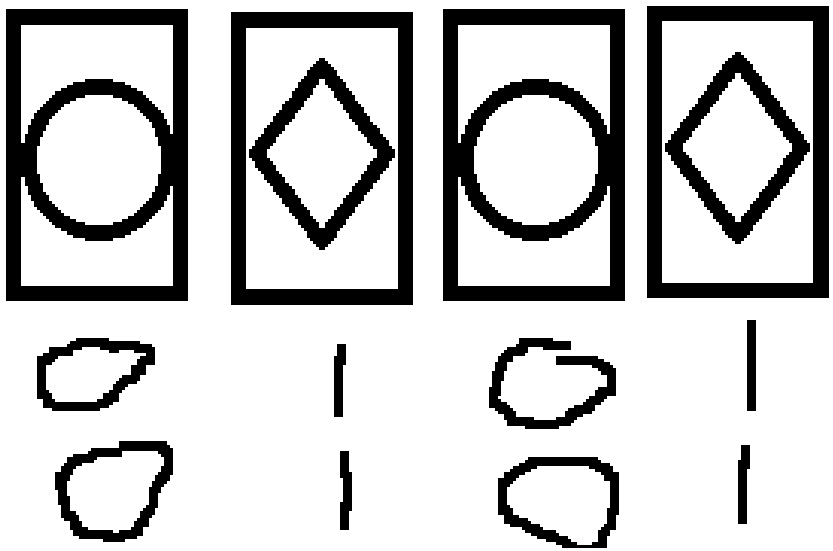
SGAN prediction

2) Ignoring prior information

Imagine SGAN is a card game in real-life.

- Diamond cards are real samples
- mini-batch = 4 cards (2 fake, 2 real)
- You are D, you estimate $D(x)$=$P(x$ is real)

Let say that the current mini-batch is:



Human prediction

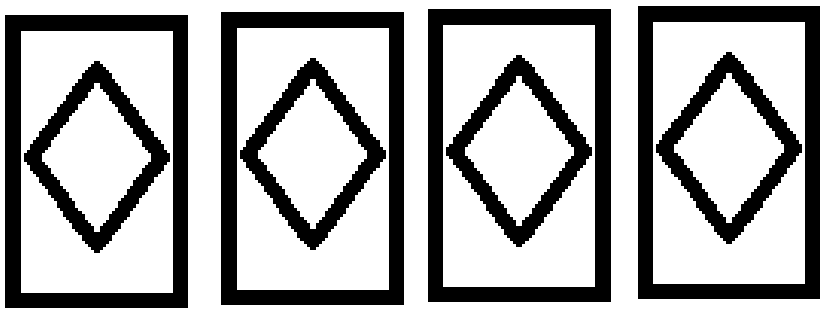
SGAN prediction

2) Ignoring prior information

Imagine SGAN is a card game in real-life.

- Diamond cards are real samples
- mini-batch = 4 cards (2 fake, 2 real)
- You are D, you estimate $D(x)$=$P(x$ is real)

Let say that G trained well so the next mini-batch is:



Human prediction

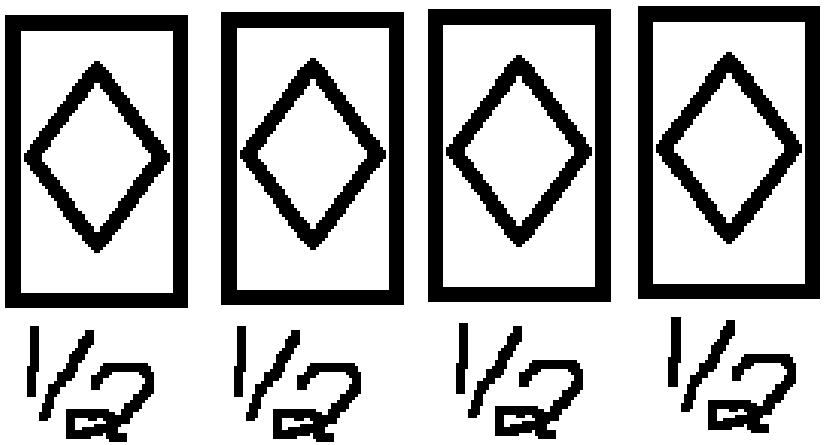
SGAN prediction

2) Ignoring prior information

Imagine SGAN is a card game in real-life.

- Diamond cards are real samples
- mini-batch = 4 cards (2 fake, 2 real)
- You are D, you estimate $D(x)$=$P(x$ is real)

Let say that G trained well so the next mini-batch is:



Human prediction

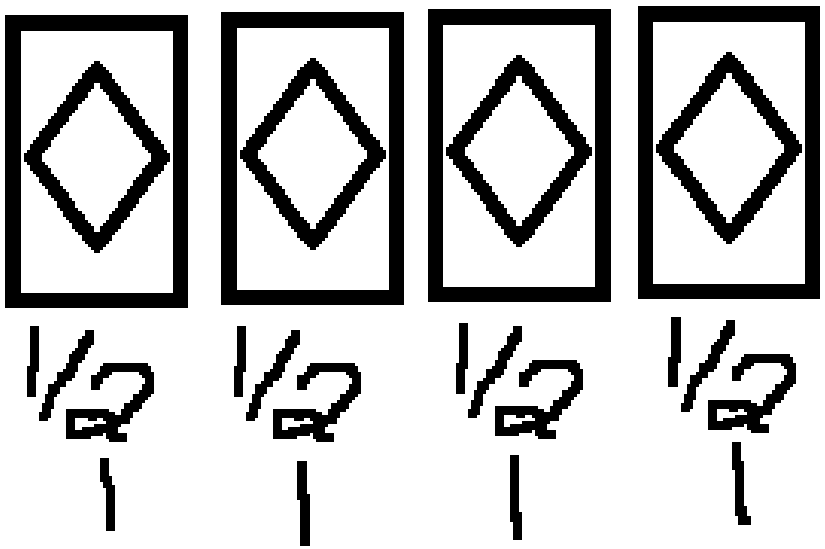
SGAN prediction

2) Ignoring prior information

Imagine SGAN is a card game in real-life.

- Diamond cards are real samples
- mini-batch = 4 cards (2 fake, 2 real)
- You are D, you estimate $D(x)$=$P(x$ is real)

Let say that G trained well so the next mini-batch is:



SGAN ignores the fact that half the cards are real, SGAN doesn't see the broad picture because it only looks at one image at a time.

Relativistic Discriminator

Question: How to impose the key missing property? 🤔
Solution: Relativistic Discriminator!

Relativistic Discriminator

Question: How to impose the key missing property? 🤔
Solution: Relativistic Discriminator!

Non-relativistic:
$D(x) = \mathrm{a}(C(x))$, a is the activation function, $C(x) \in \mathbb{R}$.

Relativistic Discriminator

Question: How to impose the key missing property? 🤔
Solution: Relativistic Discriminator!

Non-relativistic:
$D(x) = \mathrm{a}(C(x))$, a is the activation function, $C(x) \in \mathbb{R}$.

Relativistic:
Sample from real/fake data pairs $\tilde{x} = (x_r, x_f)$, $D(\tilde{x}) = \mathrm{a}(C(x_r) - C(x_f))$.
Relativistic SGAN: $D(\tilde{x}) \approx P(x_r \text{ is more realistic than } x_f)$
IPM GANs: Special case where $\mathrm{a}(y) = y$ and we use an IPM constraint.

Relativistic Discriminator

Question: How to impose the key missing property? 🤔
Solution: Relativistic Discriminator!

Non-relativistic:
$D(x) = \mathrm{a}(C(x))$, a is the activation function, $C(x) \in \mathbb{R}$.

Relativistic:
Sample from real/fake data pairs $\tilde{x} = (x_r, x_f)$, $D(\tilde{x}) = \mathrm{a}(C(x_r) - C(x_f))$.
Relativistic SGAN: $D(\tilde{x}) \approx P(x_r \text{ is more realistic than } x_f)$
IPM GANs: Special case where $\mathrm{a}(y) = y$ and we use an IPM constraint.

Relativistic average:
$\tilde{D}(x_r) = \mathrm{a}\left(C(x_r) - \mathbb{E}_{x_f \sim \mathbb{Q}} C(x_f)\right)$
$\tilde{D}(x_f) = \mathrm{a}\left(C(x_f) - \mathbb{E}_{x_r \sim \mathbb{P}} C(x_r)\right)$

Relativistic Discriminator

Question: How to impose the key missing property? 🤔
Solution: Relativistic Discriminator!

Non-relativistic:
$D(x) = \mathrm{a}(C(x))$, a is the activation function, $C(x) \in \mathbb{R}$.

Relativistic:
Sample from real/fake data pairs $\tilde{x} = (x_r, x_f)$, $D(\tilde{x}) = \mathrm{a}(C(x_r) - C(x_f))$.
Relativistic SGAN: $D(\tilde{x}) \approx P(x_r \text{ is more realistic than } x_f)$
IPM GANs: Special case where $\mathrm{a}(y) = y$ and we use an IPM constraint.

Relativistic average:
$\tilde{D}(x_r) = \mathrm{a}\left(C(x_r) - \mathbb{E}_{x_f \sim \mathbb{Q}} C(x_f)\right)$
$\tilde{D}(x_f) = \mathrm{a}\left(C(x_f) - \mathbb{E}_{x_r \sim \mathbb{P}} C(x_r)\right)$

Making D relativistic gives us the key property missing and it works with any GAN! 😸 It also improves stability and generated data quality! 🙀

Relativistic Discriminator

Relativistic Standard GAN (RSGAN):

$$\max_C \mathbb{E}_{(x_r,x_f)\sim(\mathbb{P},\mathbb{Q})} \left[\log(\text{sigmoid}(C(x_r) - C(x_f)))\right]. \tag{4}$$

$$\max_G \mathbb{E}_{(x_r,x_f)\sim(\mathbb{P},\mathbb{Q})} \left[\log(\text{sigmoid}(C(x_f) - C(x_r)))\right]. \tag{5}$$

Note that $\log(D(x_r, x_f)) = \log(1 - D(x_f, x_r))$ so we do not need two terms.

Relativistic average Standard GAN (RaSGAN):

$$\max_D \mathbb{E}_{x_r\sim\mathbb{P}} \left[\log\left(\tilde{D}(x_r)\right)\right] + \mathbb{E}_{x_f\sim\mathbb{Q}} \left[\log\left(1 - \tilde{D}(x_f)\right)\right] \tag{6}$$

$$\max_G \mathbb{E}_{x_f\sim\mathbb{Q}} \left[\log\left(\tilde{D}(x_f)\right)\right] + \mathbb{E}_{x_r\sim\mathbb{P}} \left[\log\left(1 - \tilde{D}(x_r)\right)\right] \tag{7}$$

$\tilde{D}(x_r) = \text{sigmoid}\left(C(x_r) - \mathbb{E}_{x_f\sim\mathbb{Q}} C(x_f)\right)$
$\tilde{D}(G(z)) = \text{sigmoid}\left(C(x_f) - \mathbb{E}_{x_r\sim\mathbb{P}} C(x_r)\right)$

Relativistic Divergence

Relativistic paired GANs (RpGANs):

$$\max_C \mathbb{E}_{(x_r,x_f)\sim(\mathbb{P},\mathbb{Q})} \left[f\left(C(x_r) - C(x_f) \right) \right]. \tag{8}$$

$$\max_G \mathbb{E}_{(x_r,x_f)\sim(\mathbb{P},\mathbb{Q})} \left[f\left(C(x_f) - C(x_r) \right) \right]. \tag{9}$$

Relativistic average Standard GAN (RaGANs):

$$\max_C \mathbb{E}_{x_r\sim\mathbb{P}} \left[\log\left(\tilde{D}(x_r) \right) \right] + \mathbb{E}_{x_f\sim\mathbb{Q}} \left[\log\left(1 - \tilde{D}(x_f) \right) \right] \tag{10}$$

$$\max_G \mathbb{E}_{x_f\sim\mathbb{Q}} \left[\log\left(\tilde{D}(x_f) \right) \right] + \mathbb{E}_{x_r\sim\mathbb{P}} \left[\log\left(1 - \tilde{D}(x_r) \right) \right] \tag{11}$$

$\tilde{D}(x_r) = \text{sigmoid}\left(C(x_r) - \mathbb{E}_{x_f\sim\mathbb{Q}} C(x_f) \right)$
$\tilde{D}(G(z)) = \text{sigmoid}\left(C(x_f) - \mathbb{E}_{x_r\sim\mathbb{P}} C(x_r) \right)$

Results - CIFAR10 (50k 32x32 images from 10 classes)

Table: Frechet Inception Distance (FID) at $100k$ generator iterations, using spectral normalization in D and batch norm in G.

Loss	$lr = .0002$ $\beta = (.50, .999)$ $n_D = 1$	$lr = .0001$ $\beta = (.50, .9)$ $n_D = 5$
SGAN	40.64	41.32
RSGAN	36.61	55.29
RaSGAN	31.98	37.92
LSGAN	29.53	187.01
RaLSGAN	30.92	219.39
HingeGAN	49.53	80.85
RaHingeGAN	39.12	37.72
WGAN-GP	83.89	**27.81**
RSGAN-GP	**25.60**	28.13
RaSGAN-GP	331.86	

Results - CAT (low res images of cats)

Table: Frechet Inception Distance (FID) at 20k, 30k ..., 100k generator iterations, using $lr = .0002$, $\beta = (.50, .999)$, $n_D = 1$, and batch norm (BN) in D and G.

Loss	Min	Max	Mean	SD
64x64 images (N=9304)				
SGAN	16.56	310.56	52.54	96.81
RSGAN	19.03	42.05	32.16	7.01
RaSGAN	15.38	33.11	20.53	5.68
LSGAN	20.27	224.97	73.62	61.02
RaLSGAN	**11.97**	**19.29**	**15.61**	2.55
HingeGAN	17.60	50.94	32.23	14.44
RaHingeGAN	14.62	27.31	20.29	3.96
RSGAN-GP	16.41	22.34	18.20	1.82
RaSGAN-GP	17.32	22	19.58	**1.81**

Results - CAT (high res images of cats)

Loss	Min	Max	Mean	SD
128x128 images (N=6645)				
SGAN	-	-	-	-
RaSGAN	21.05	**39.65**	28.53	**6.52**
LSGAN	19.03	51.36	30.28	10.16
RaLSGAN	**15.85**	40.26	**22.36**	7.53
256x256 images (N=2011)				
SGAN	-	-	-	-
RaSGAN	**32.11**	102.76	**56.64**	21.03
SpectralSGAN	54.08	**90.43**	64.92	**12.00**
LSGAN	-	-	-	-
RaLSGAN	35.21	299.52	70.44	86.01
WGAN-GP	155.46	437.48	341.91	101.11

Results - CAT256

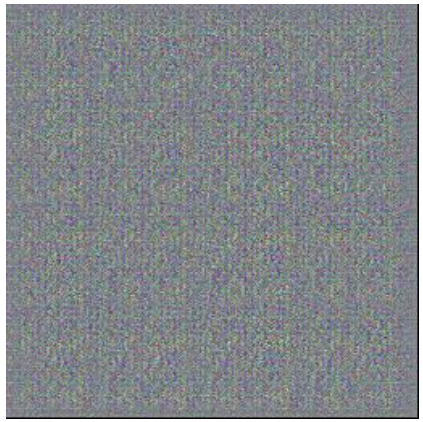


Figure: 256x256 cats with GAN (5k iterations)

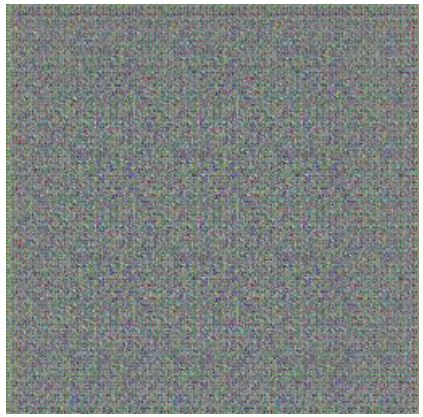


Figure: 256x256 cats with LSGAN (5k iterations)

Results - CAT256

Figure: 256x256 cats with RaSGAN (FID = 32.11)

Results - CAT256

Figure: 256x256 cats with RaLSGAN (FID = 35.21)

Results - CAT256

Figure: 256x256 cats with SpectralSGAN (FID = 54.73)

Results - CAT256

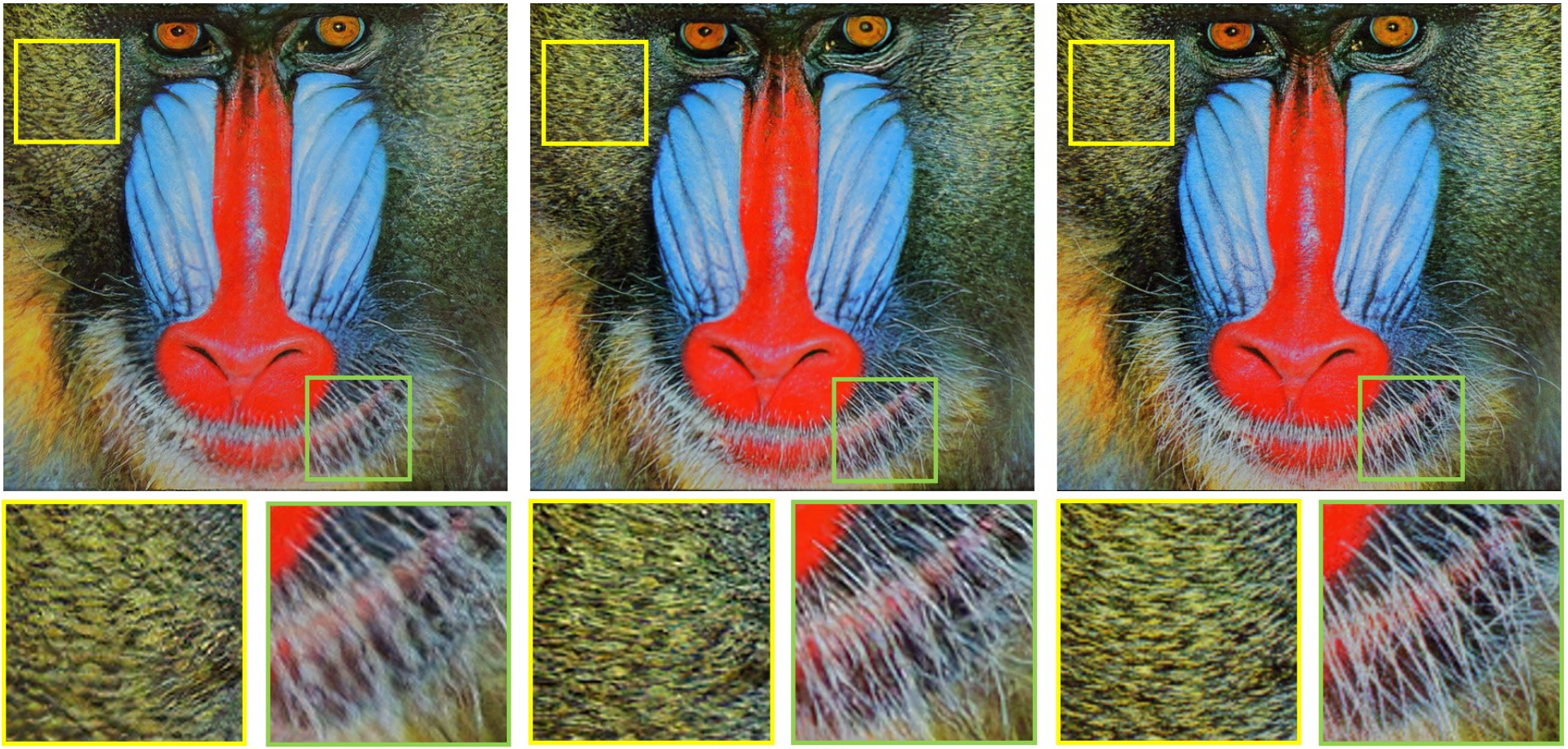
Figure: 256x256 cats with WGAN-GP (FID $>$ 100)

Paper conclusion

1. D should be relativistic
2. RGANs or RaGANs = Enhanced stability
3. RaGANs = Higher quality of generated data
4. RGANs and RaGANs can obtain state-of-the-art WGAN-GP results while speeding up computations by AT LEAST 3 times
5. RaGANs can generate high res images from scratch, while Standard GAN and Least Squares GAN cannot
6. high-dimensional small datasets, such as CAT, are challenging and thus great for testing GANs

Applications (ESRGAN)

- Xintao Wang et al. (2018) used a Standard RaGAN to win first place in the PIRM2018-SR competition (region 3) and obtain the best perceptual index.



Applications (ESRGAN)

- ESRGAN is now used in video game emulation to enhance the quality of textures in old video games!
- See: https://www.reddit.com/r/GameUpscale/ 😸

Relativistic-f divergences

In my last paper:

- I introduce more variants of Relativistic GANs
- I mathematically prove that Relativistic GANs (and variants) are actual divergences using any concave function f with certain basic properties.
- I show that RaGANs are biased, but the bias is small and negligible with a batch size ≥ 32. Also, interestingly, RaLSGAN bias has a simple closed form and it can be easily removed.

This paper provide a stronger mathematical foundation to my approach.
See: https://arxiv.org/abs/1901.02474

Final comments

- I did this work alone without supervision. Everything was run on my single GPU, a Geforce 1060. Every night, I booted my computer on Ubuntu and trained a few models.
- This is already my most cited paper at 31 citations! It beats my grant-funded multi-author papers from 2015.
- I only scratched the surface of Relativistic GANs, there is so much potentially interesting variants to study.
- **Make sure to use a relativistic discriminator for best results!**

The End