Artificial faces synthesized by StyleGAN (Nvidia)

COMP547

DEEP UNSUPERVISED LEARNING

Lecture #8 – Generative Adversarial Networks Part 1

KOÇ UNIVERSITY

Aykut Erdem // Koç University // Spring 2022

Good news, everyone!

- Assignment 2 will be out today! (due April 3)
- Let us know if you want to contribute to COMP547 lecture notes!

Previously on COMP547

- Motivation
- Training Latent Variable Models (including VAE and IWAE)
- Variations
- Related ideas

Image: Synthetic faces sampled from NVAE model by Vahdat and Kautz

Lecture overview

- Motivation & Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- GAN Progression
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

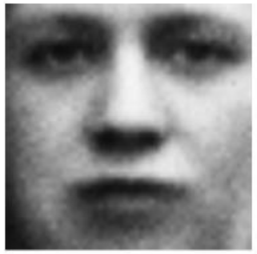
Disclaimer: Much of the material and slides for this lecture were borrowed from
—Pieter Abbeel, Peter Chen, Jonathan Ho, Aravind Srinivas' Berkeley CS294-158 class
—Aaron Courville's IFT6135 class
—Bill Freeman, Antonio Torralba and Phillip Isola's MIT 6.869 class

Lecture overview

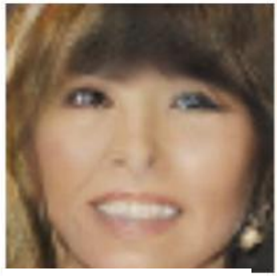
- **Motivation and Definition of Implicit Models**
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- GAN Progression
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

Motivation: Evolution of GANs

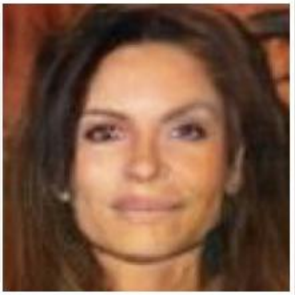
- 5 years of GAN progress



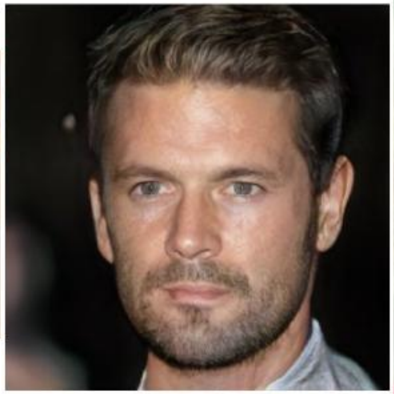
2014



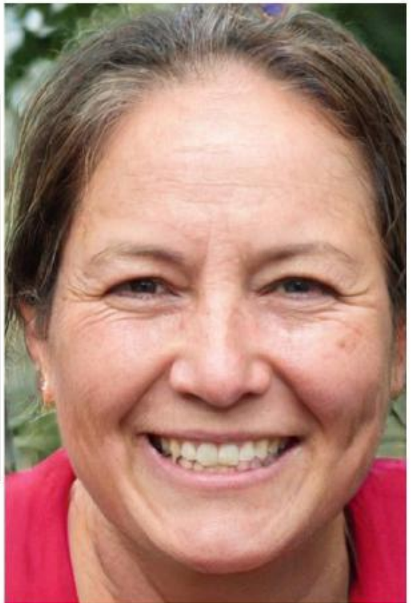
2015



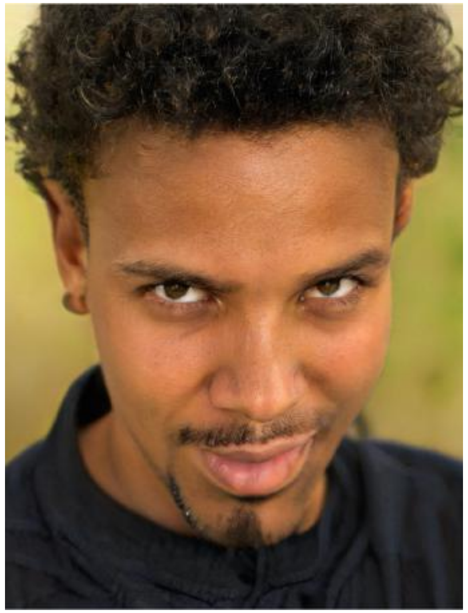
2016



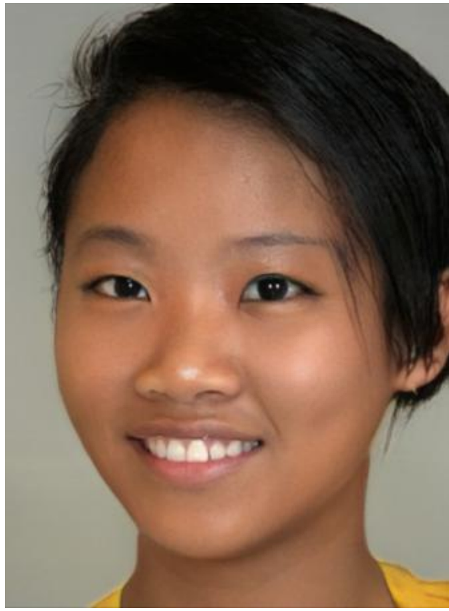
2018



2019



2020



2021

- GAN is most prominent of Implicit Models

I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. **Generative Adversarial Networks**. NIPS 2014.

A. Radford, L. Metz, S. Chintala. **Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks**. ICLR 2016.

M.-Y. Liu, O. Tuzel. Coupled Generative Adversarial Networks. NIPS 2016.

T. Karras, T. Aila, S. Laine, J. Lehtinen. **Progressive Growing of GANs for Improved Quality, Stability, and Variation**. ICLR 2018.

T. Karras, S. Laine, T. Aila. **A style-based generator architecture for generative adversarial networks**. In CVPR 2018.

T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila. **Analyzing and Improving the Image Quality of StyleGAN**. CVPR 2020.

T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, T. Aila. **Alias-Free Generative Adversarial Networks.** NeurIPS 2021.

Slide adapted from Ian Goodfellow

Motivation: BigGAN

Andrew Brock, Jeff Donahue, Karen Simonyan, Large Scale GAN Training for High Fidelity Natural Image Synthesis, ICLR 2019

So far…

- Autoregressive models
  - MADE, PixelRNN/CNN, Gated PixelCNN, PixelSNAIL
- Flow models
  - Autoregressive Flows, NICE, RealNVP, Glow, Flow++
- Latent Variable Models
  - VAE, IWAE, VQ-VAE, VLAE, PixelVAE

- Common aspect: Likelihood-based models
  - exact (autoregressive and flows)
  - approximate (VAE)

Generative Models

- Sample
- Evaluate likelihood
- Train
- Representation

→ What if all we care about is sampling?

Building a sampler

- How about this sampler?

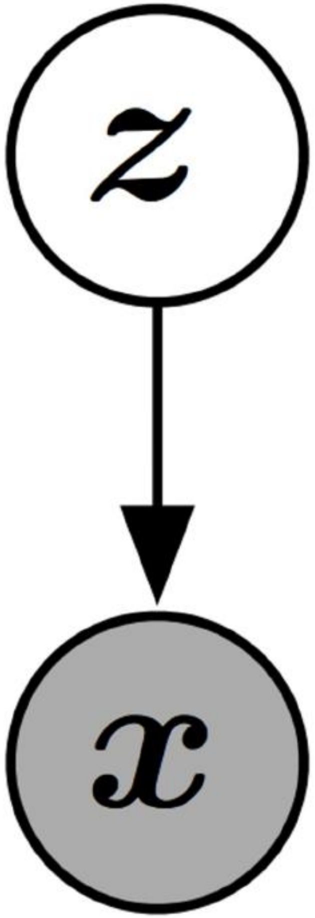
```
import glob, cv2, numpy as np
files = glob.glob(‘*.jpg’)
def _sample():
    idx = np.random.randint(len(files))
    return cv2.imread(files[idx])
def sample(*, n_samples):
    samples = np.array([_sample() for _ in range(n_samples)])
    return samples
```

Building a sampler

- You don’t just want to sample the exact data points you have.
- You want to build a generative model that can understand the underlying distribution of data points and
  - smoothly interpolate across the training samples
  - output samples similar but not the same as training data samples
  - output samples representative of the underlying factors of variation in the training distribution.
  - Example: digits with unseen strokes, faces with unseen poses, etc.

Implicit Models

- Sample z from a fixed noise source distribution (uniform or gaussian).
- Pass the noise through a deep neural network to obtain a sample x.
- Sounds familiar? Right:
  - Flow Models
  - VAE
- What's going to be different here?
  - Learning the deep neural network **<u>without</u>** explicit density estimation



Implicit Models

Given samples from data distribution $p_{data} : x_1, x_2, \ldots, x_n$

Given a sampler $q_\phi(z) = \mathrm{DNN}(z; \phi)$ where $z \sim p(z)$

$x = q_\phi(z)$ induces a density function p_{model}

- Do not have an explicit form for p_{data} or p_{model}; can only draw samples

- Make p_{model} as close to p_{data} as possible by learning an appropriate ϕ

Departure from maximum likelihood

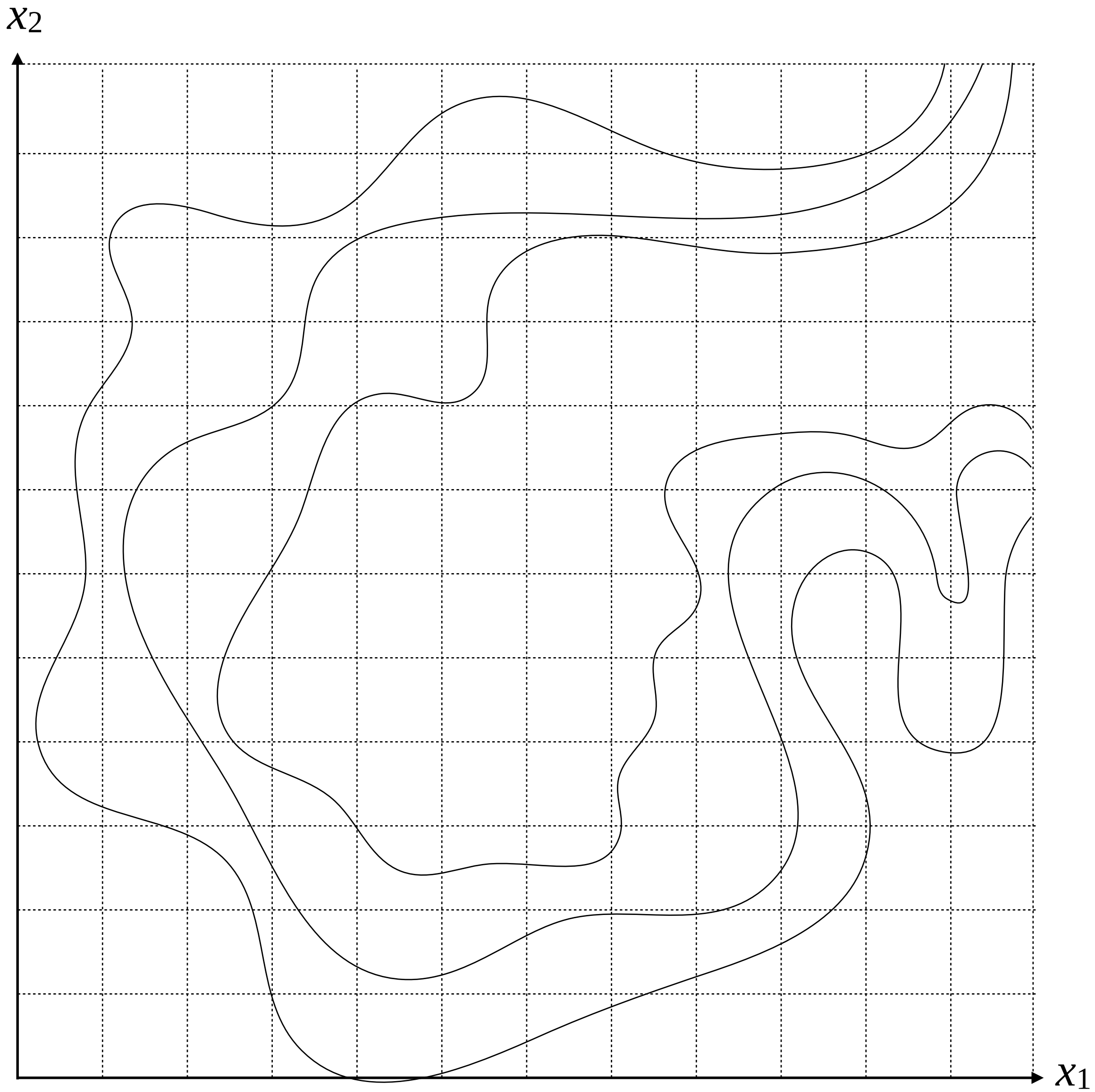
- We need some measure of how far apart p_{data} and induces p_{model} are

- With density models, we used $KL(p_{data} \| p_{model})$ which gave us the objective $\mathbb{E}_{x \sim p_{data}}[\log p_\theta(x)]$ (discarding the term independent of θ) where we explicitly modeled p_{model} as $p_\theta(x)$

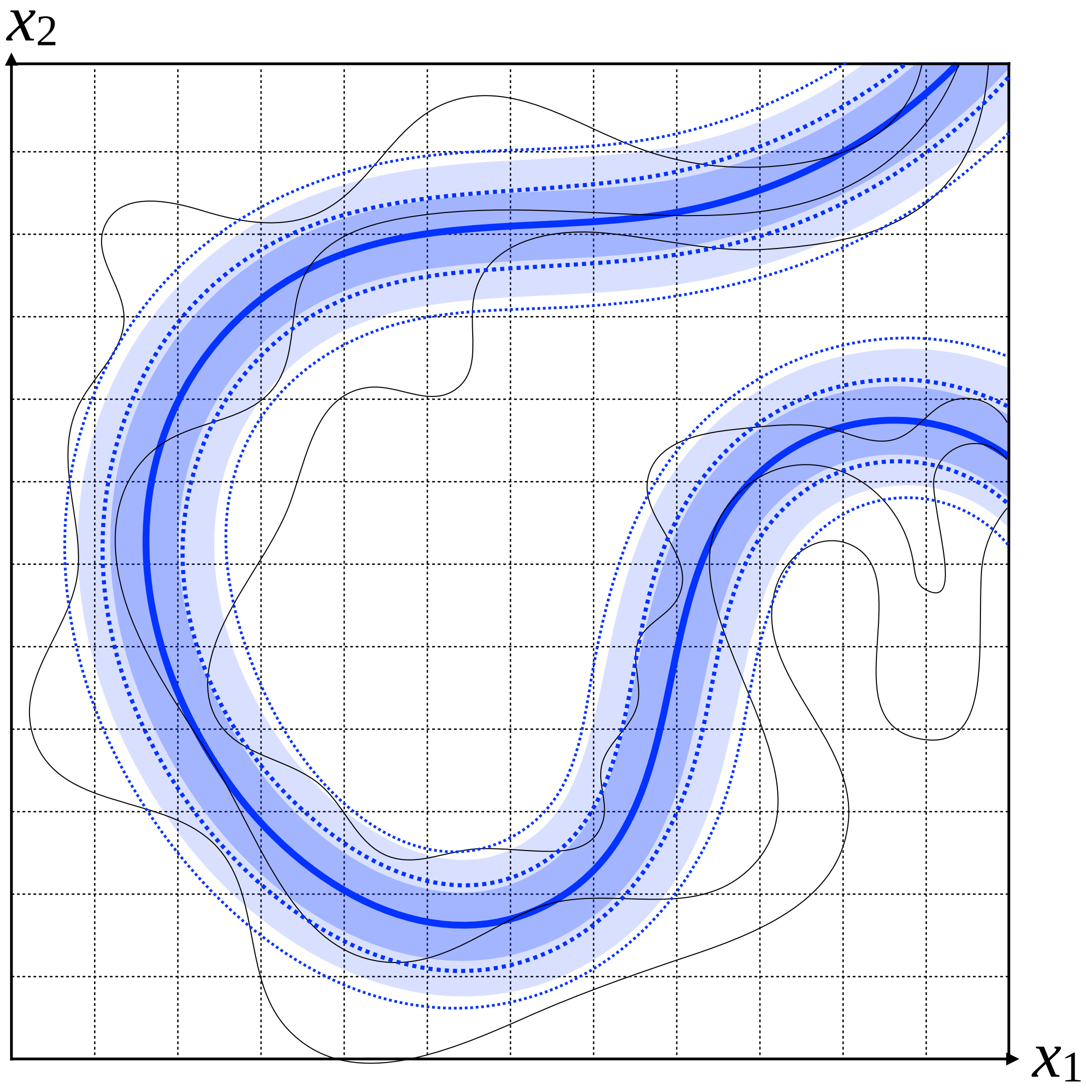
- Not having an explicit $p_\theta(x)$ requires us to come up distance measures that potentially behave differently from maximum likelihood.

- Example: Maximum Mean Discrepancy (MMD), Jensen Shannon Divergence (JSD), Earth Mover's Distance, etc.

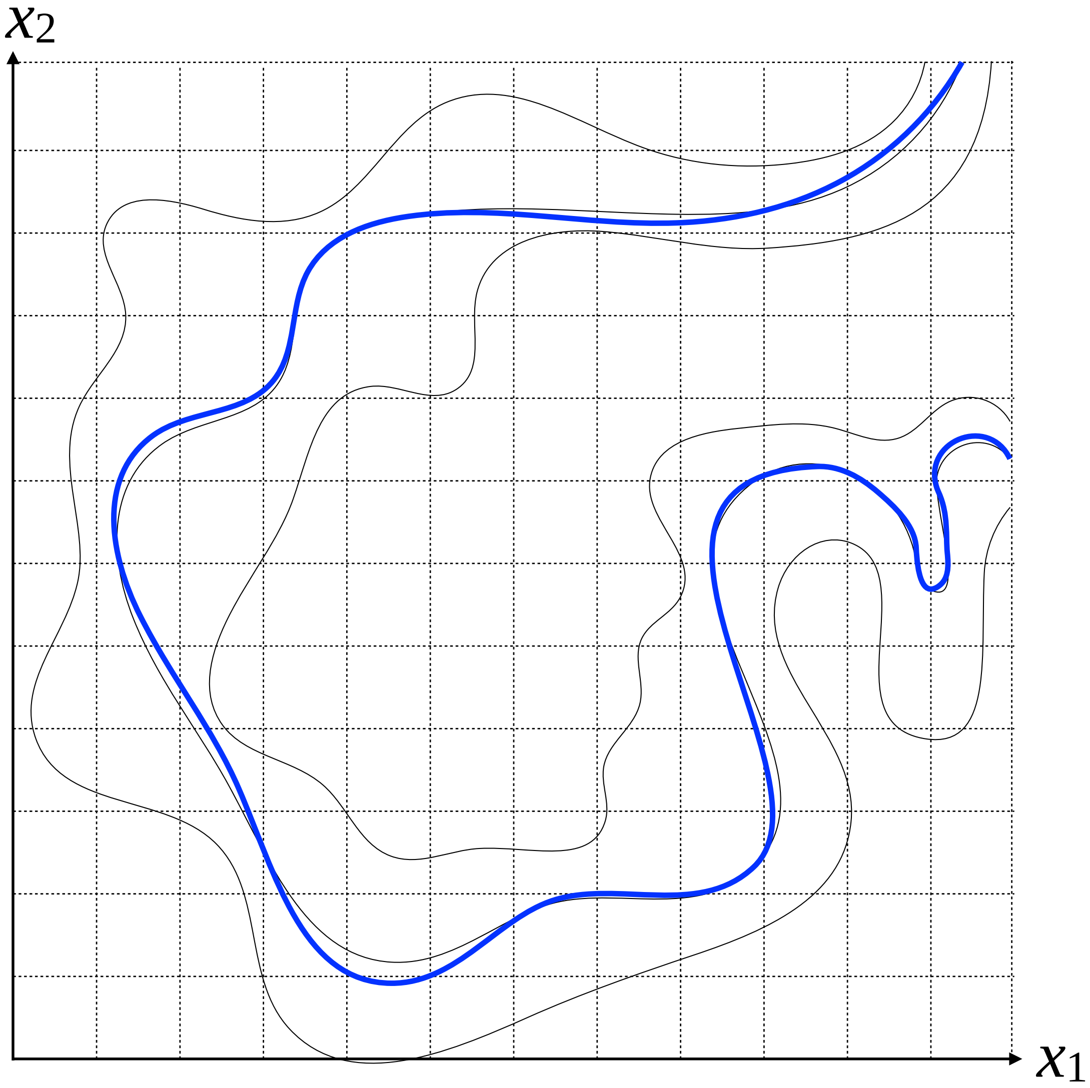
Cartoon of the Image manifold



What makes GANs special?



more traditional max-likelihood approach

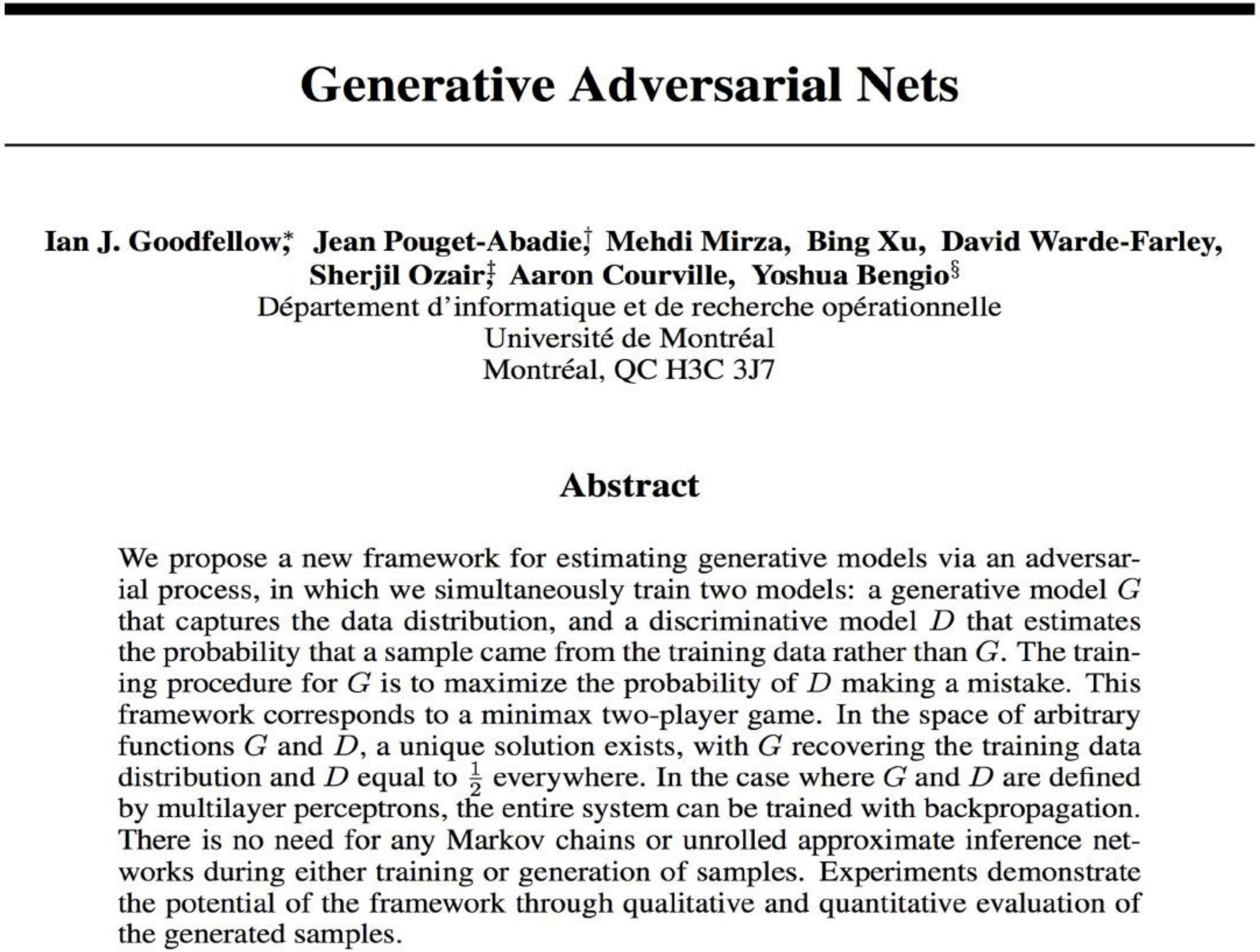


GAN

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- GAN Progression
- Conditional GANs, Cycle-Consistent Adversarial Networks
- Applications

Generative Adversarial Networks



Generative Adversarial Nets

Ian J. Goodfellow,[*] Jean Pouget-Abadie,[†] Mehdi Mirza, Bing Xu, David Warde-Farley,
Sherjil Ozair,[‡] Aaron Courville, Yoshua Bengio[§]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

[Goodfellow et al 2014]

Generative Adversarial Networks

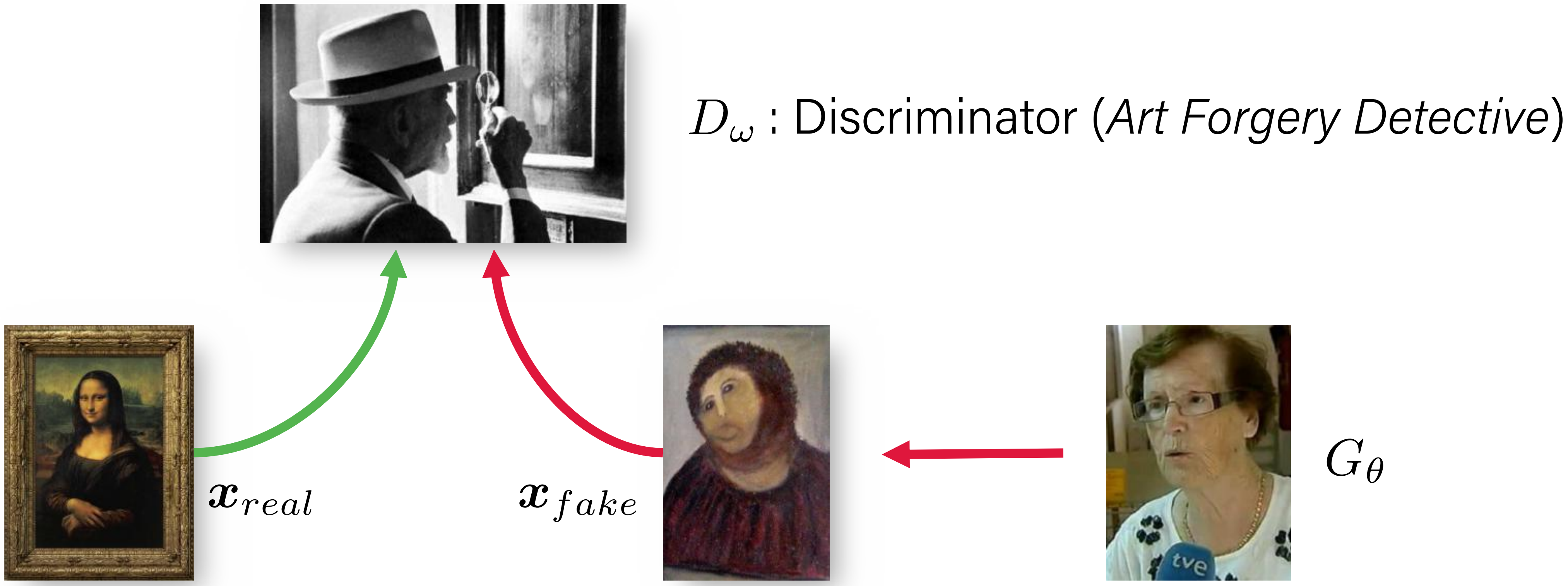
$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z\sim p(z)} [\log(1 - D(G(z)))]$$

- Two player minimax game between generator (G) and discriminator (D)
- (D) tries to maximize the log-likelihood for the binary classification problem

  - data: real (1)
  - generated: fake (0)
- (G) tries to minimize the log-probability of its samples being classified as “fake” by the discriminator (D)

Intuition behind GANs

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



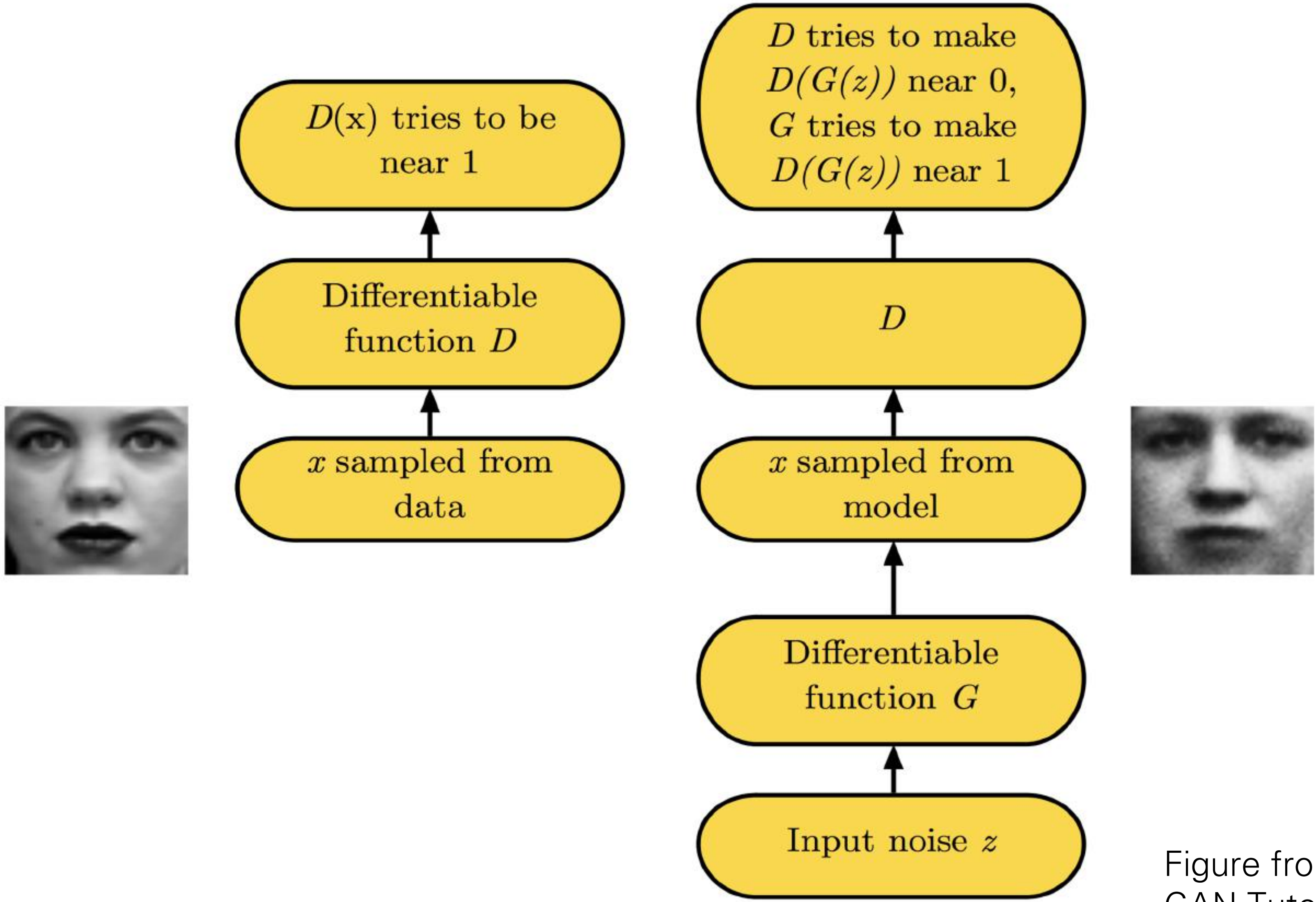
D_ω : Discriminator (*Art Forgery Detective*)

$\boldsymbol{x}_{real}$

$\boldsymbol{x}_{fake}$

G_θ

Generative Adversarial Networks



D(x) tries to be near 1

Differentiable function D

x sampled from data

D tries to make $D(G(z))$ near 0, G tries to make $D(G(z))$ near 1

D

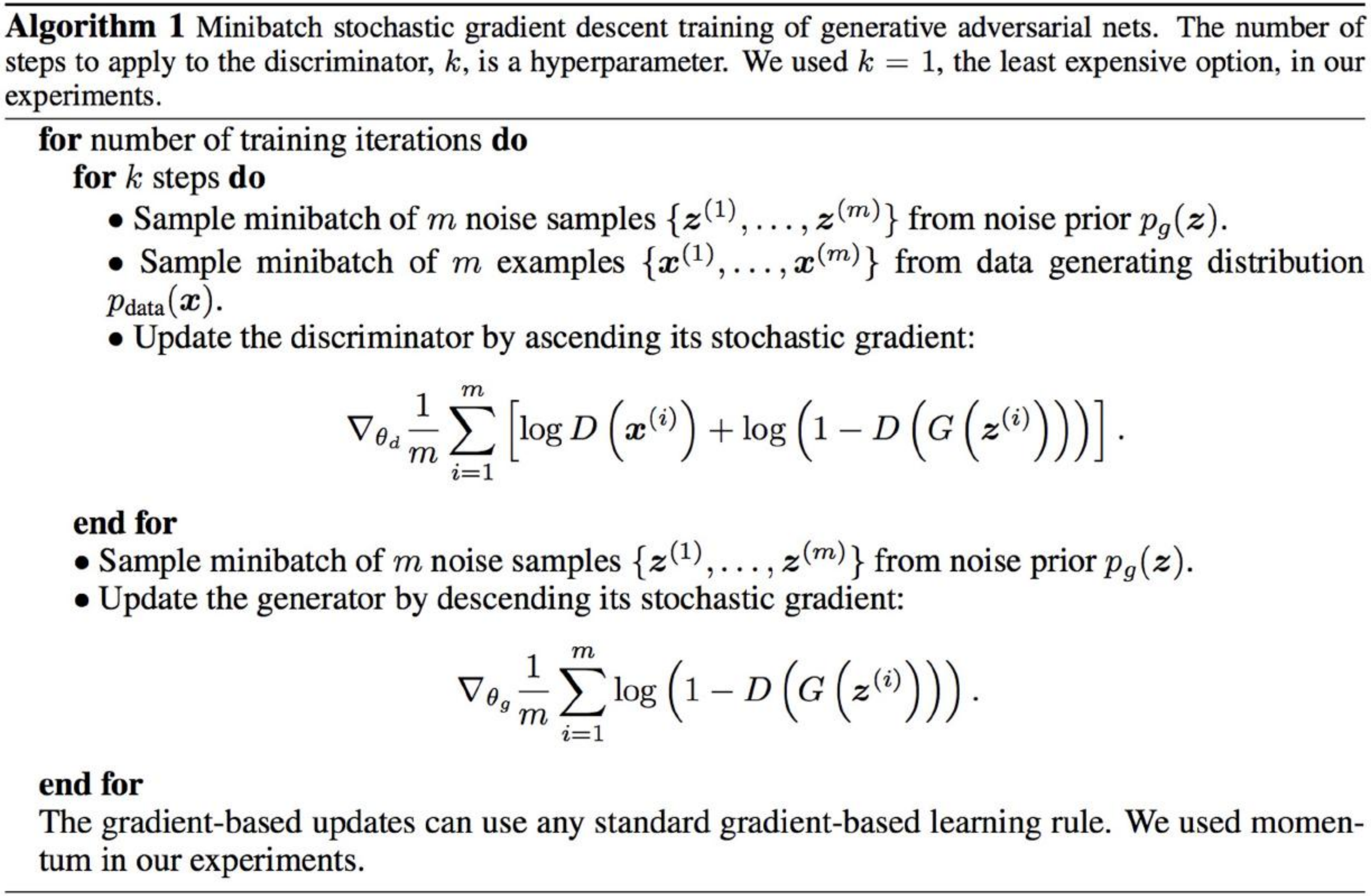
x sampled from model

Differentiable function G

Input noise z

Figure from NeurIPS 2016 GAN Tutorial (Goodfellow)

GANs - Pseudocode



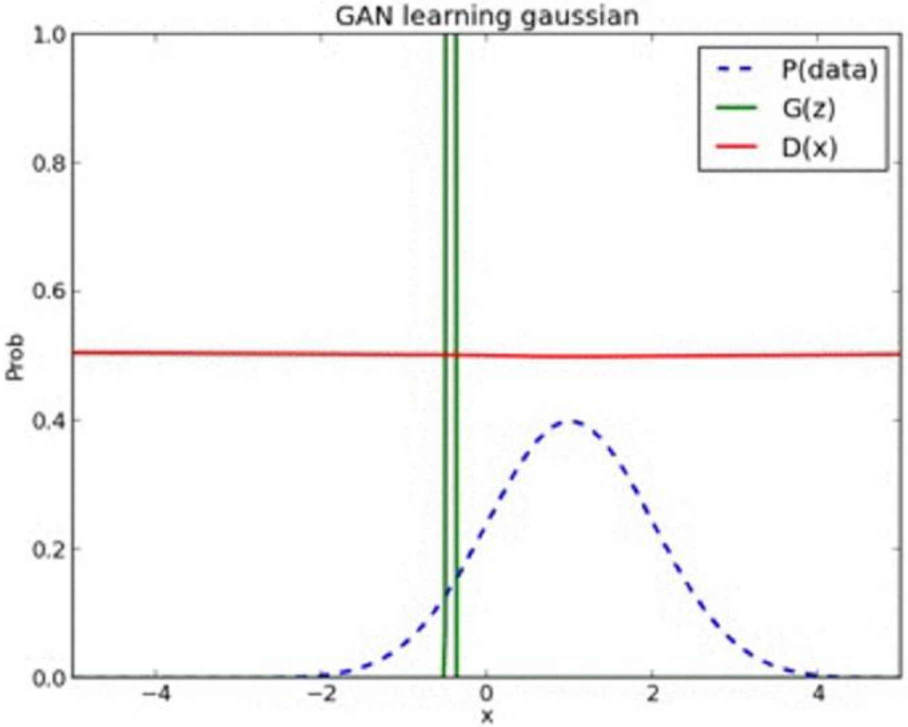
Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

```
for number of training iterations do
  for $k$ steps do
    • Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of $m$ examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:
        $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$
  end for
  • Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:
      $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$
end for
```

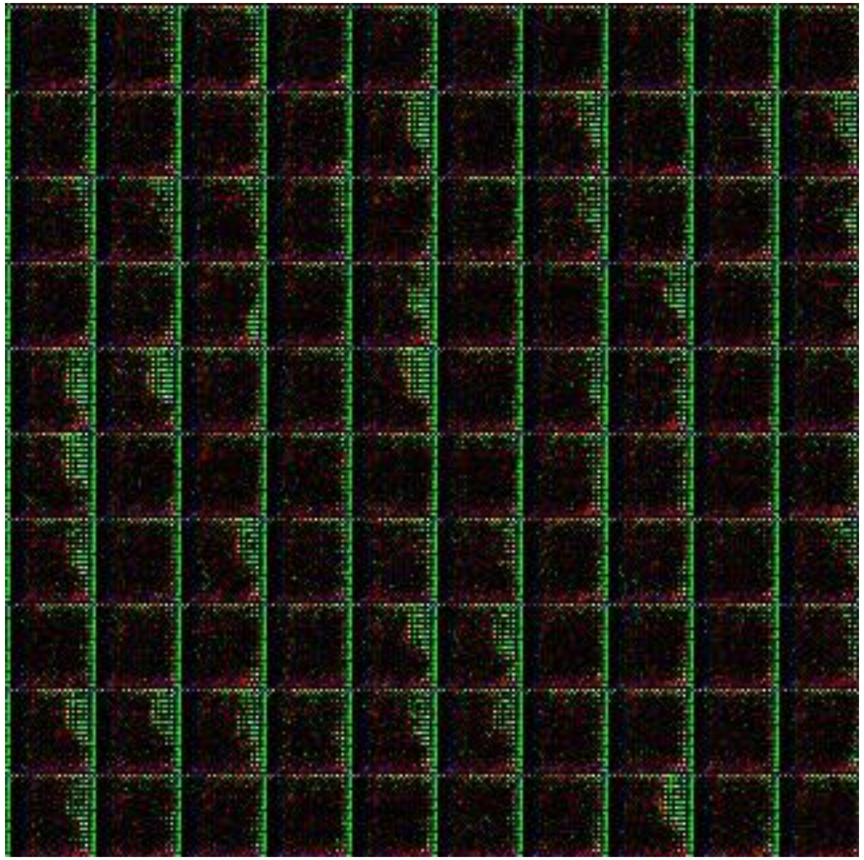
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

[Goodfellow et al 2014]

Training Procedure



Source: Alec Radford



Source: OpenAI blog

Generating 1D points

Generating images

GAN in Action

https://poloclub.github.io/ganlab/

GAN samples from 2014

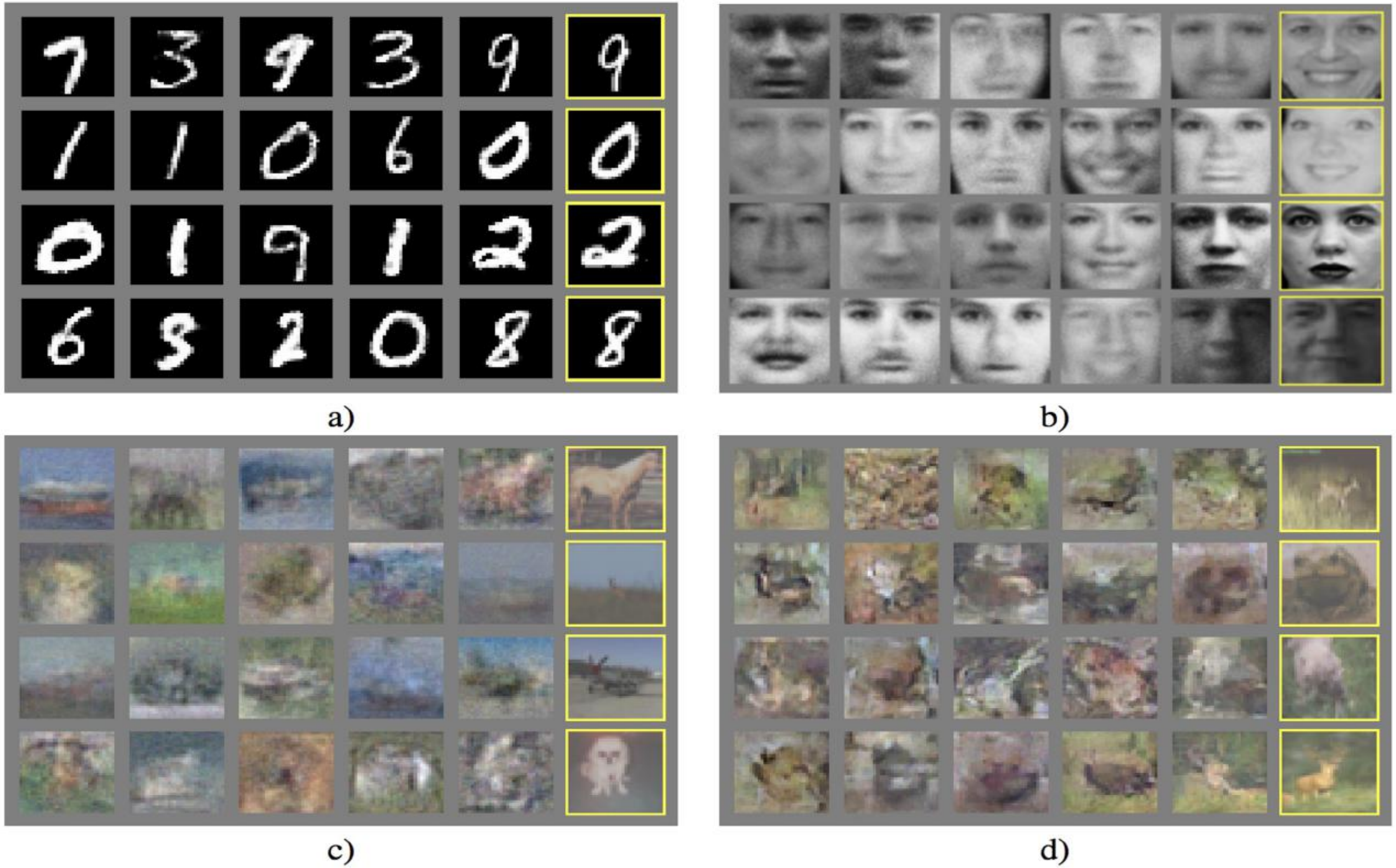


Figure from [Goodfellow et al. 2014]

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- **Evaluation: Parzen, Inception, Frechet**
- Theory of GANs
- GAN Progression
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

How to evaluate?

- Evaluation for GANs is still an open problem

- Unlike density models, you cannot report explicit likelihood estimates on test sets.

Parzen-Window density estimator

- Also known as Kernel Density Estimator (KDE)
- An estimator with kernel K and bandwidth h:

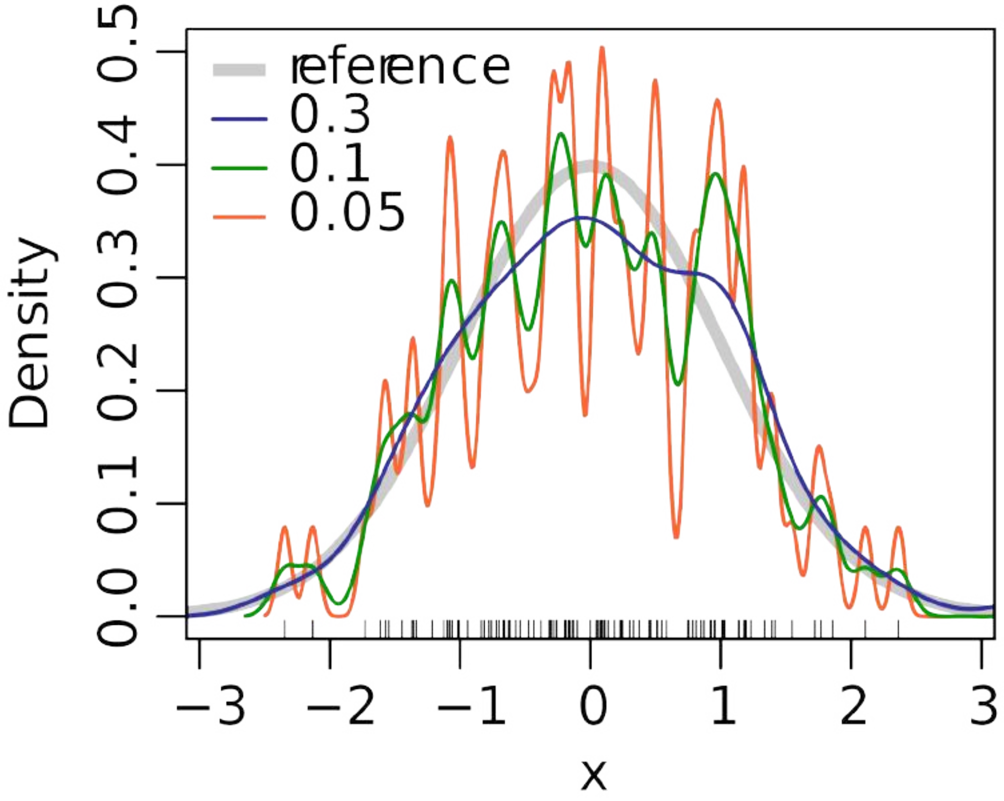
$$\hat{p_h}(x) = \frac{1}{nh}\sum_i K\left(\frac{x - x_i}{h}\right)$$

- In generative model evaluation, K is usually density function of standard Normal distribution

[Bishop 2006]

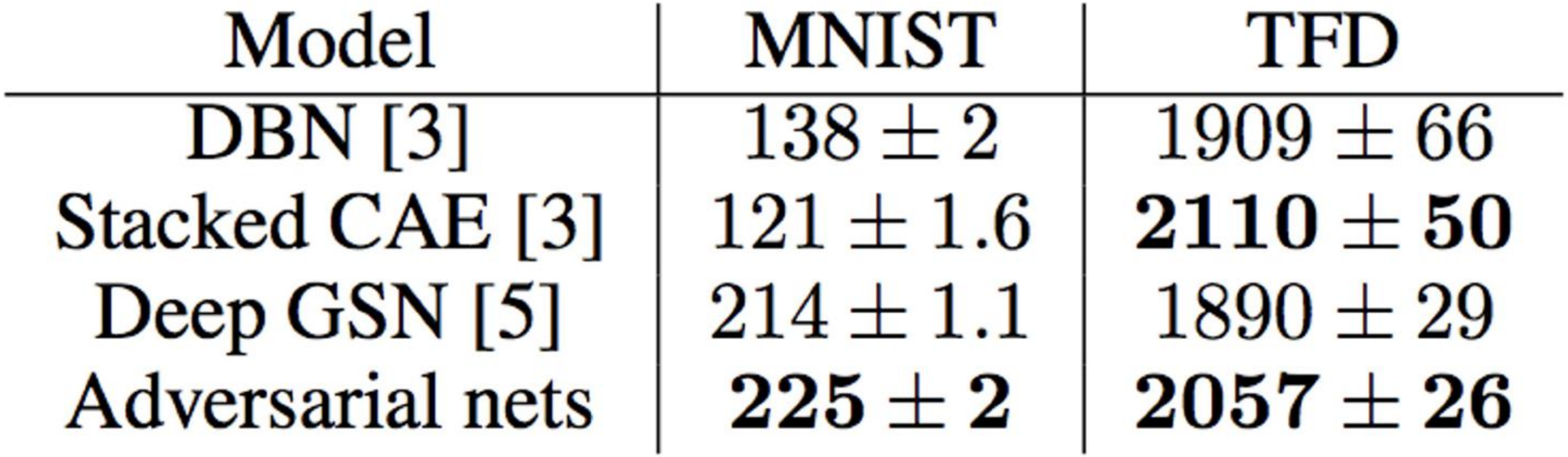
Parzen-Window density estimator

- Bandwidth h matters
- Bandwidth h chosen according to validation set



[Bishop 2006]

Evaluation

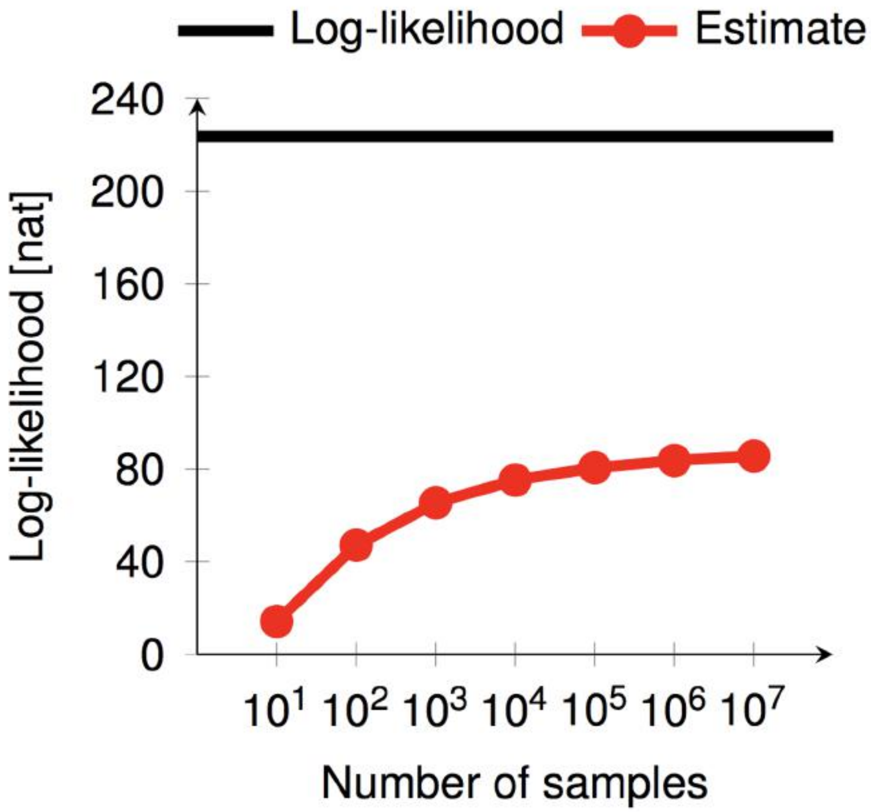


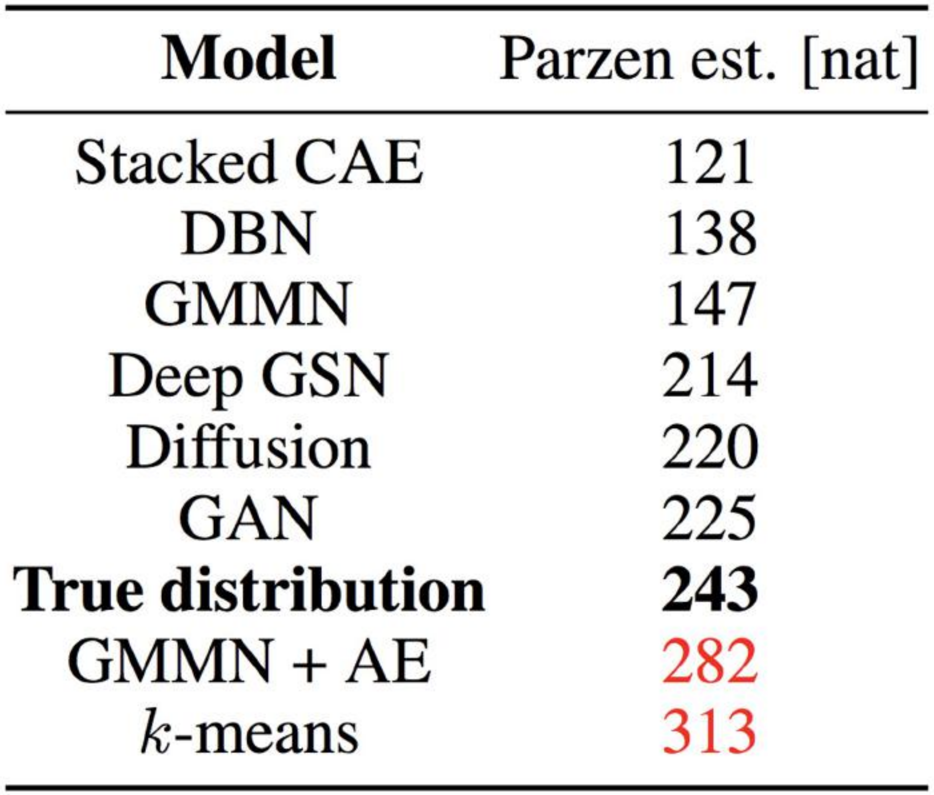
Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	**2110 ± 50**
Deep GSN [5]	214 ± 1.1	1890 ± 29
Adversarial nets	**225 ± 2**	**2057 ± 26**

Parzen Window density estimates (Goodfellow et al, 2014)

Parzen-Window density estimator

- Parzen Window estimator can be unreliable





Model	Parzen est. [nat]
Stacked CAE	121
DBN	138
GMMN	147
Deep GSN	214
Diffusion	220
GAN	225
True distribution	**243**
GMMN + AE	282
k-means	313

[A note on the evaluation of generative models (Theis, Van den Oord, Bethge 2015)]

Inception Score

- Can we side-step high-dim density estimation?
- **One idea:** good generators generate samples that are semantically diverse
- Semantics predictor: trained Inception Network v3
  - p(y|x), y is one of the 1000 ImageNet classes
- Considerations:
  - each image x should have distinctly recognizable object -> p(y|x) should have low entropy
  - there should be as many classes generated as possible -> p(y) should have high entropy

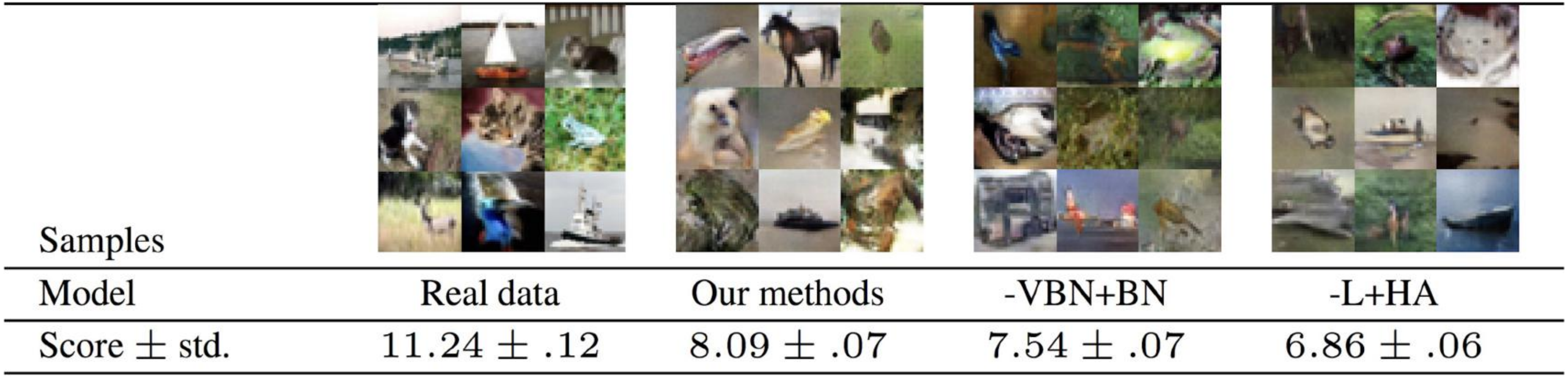
Inception Score

- Inception model: $p(y|x)$
- Marginal label distribution: $p(y) = \int_x p(y|x) p_g(x)$
- Inception Score:

$$\begin{aligned} \mathrm{IS}(x) &= \exp(\mathbb{E}_{x \sim p_g} \left[D_{\mathrm{KL}} \left[p(y|x) \parallel p(y) \right] \right]) \\ &= \exp(\mathbb{E}_{x \sim p_g, y \sim p(y|x)} \left[\log p(y|x) - \log p(y) \right]) \\ &= \exp(H(y) - H(y|x)) \end{aligned}$$

[Improved GAN: Salimans et al 2016]

Inception Score



Samples				
Model	Real data	Our methods	-VBN+BN	-L+HA
Score ± std.	$11.24 \pm .12$	$8.09 \pm .07$	$7.54 \pm .07$	$6.86 \pm .06$

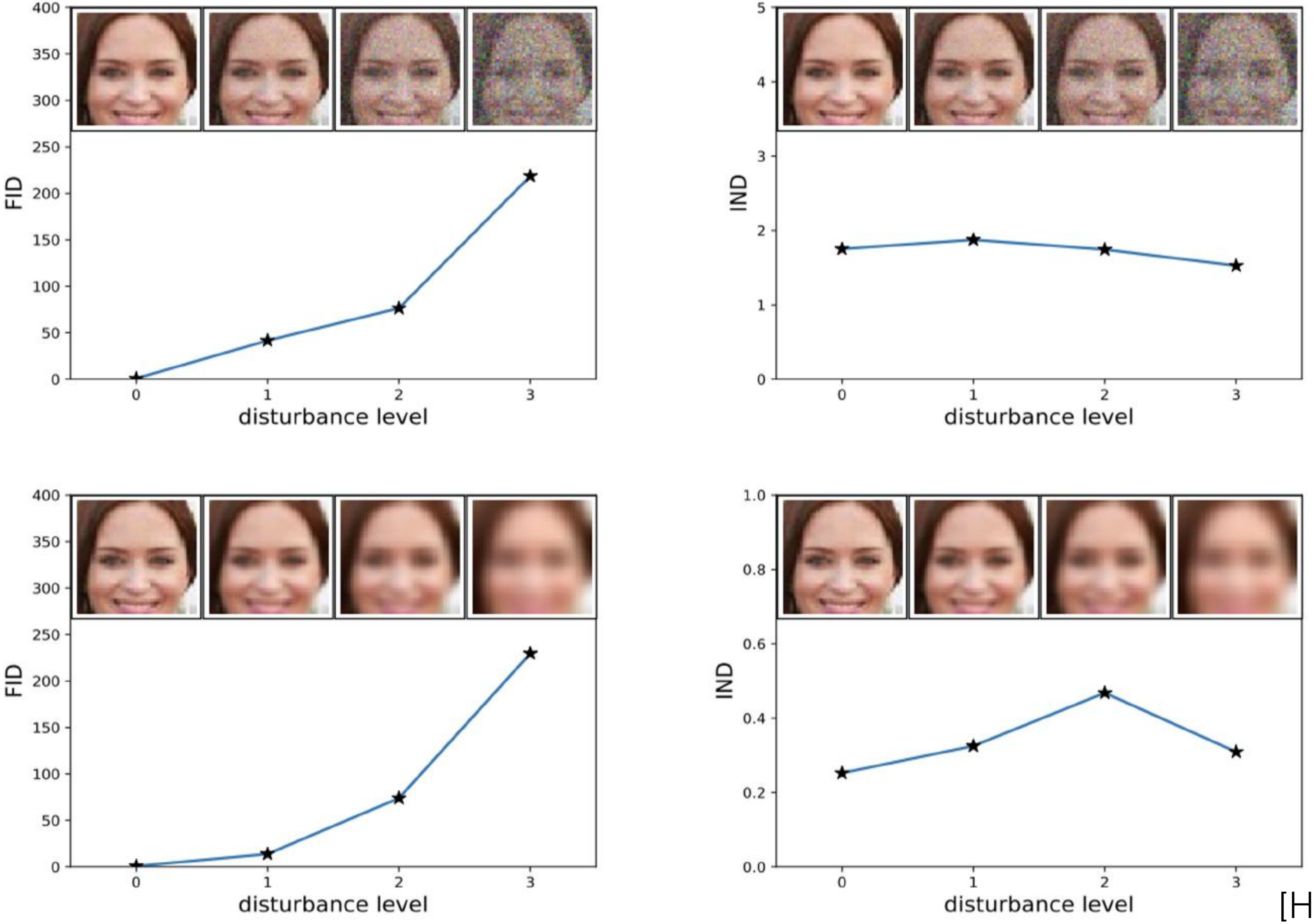
Fréchet Inception Distance

- Inception Score doesn’t sufficiently measure diversity: a list of 1000 images (one of each class) can obtain perfect Inception Score
- FID was proposed to capture more nuances
- Embed image x into some feature space (2048-dimensional activations of the Inception-v3 pool3 layer), then compare mean (m) & covariance (C) of those random features

$$d^2((\boldsymbol{m},\boldsymbol{C}),(\boldsymbol{m}_w,\boldsymbol{C}_w)) = \|\boldsymbol{m}-\boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\big(\boldsymbol{C}+\boldsymbol{C}_w-2\big(\boldsymbol{C}\boldsymbol{C}_w\big)^{1/2}\big)$$

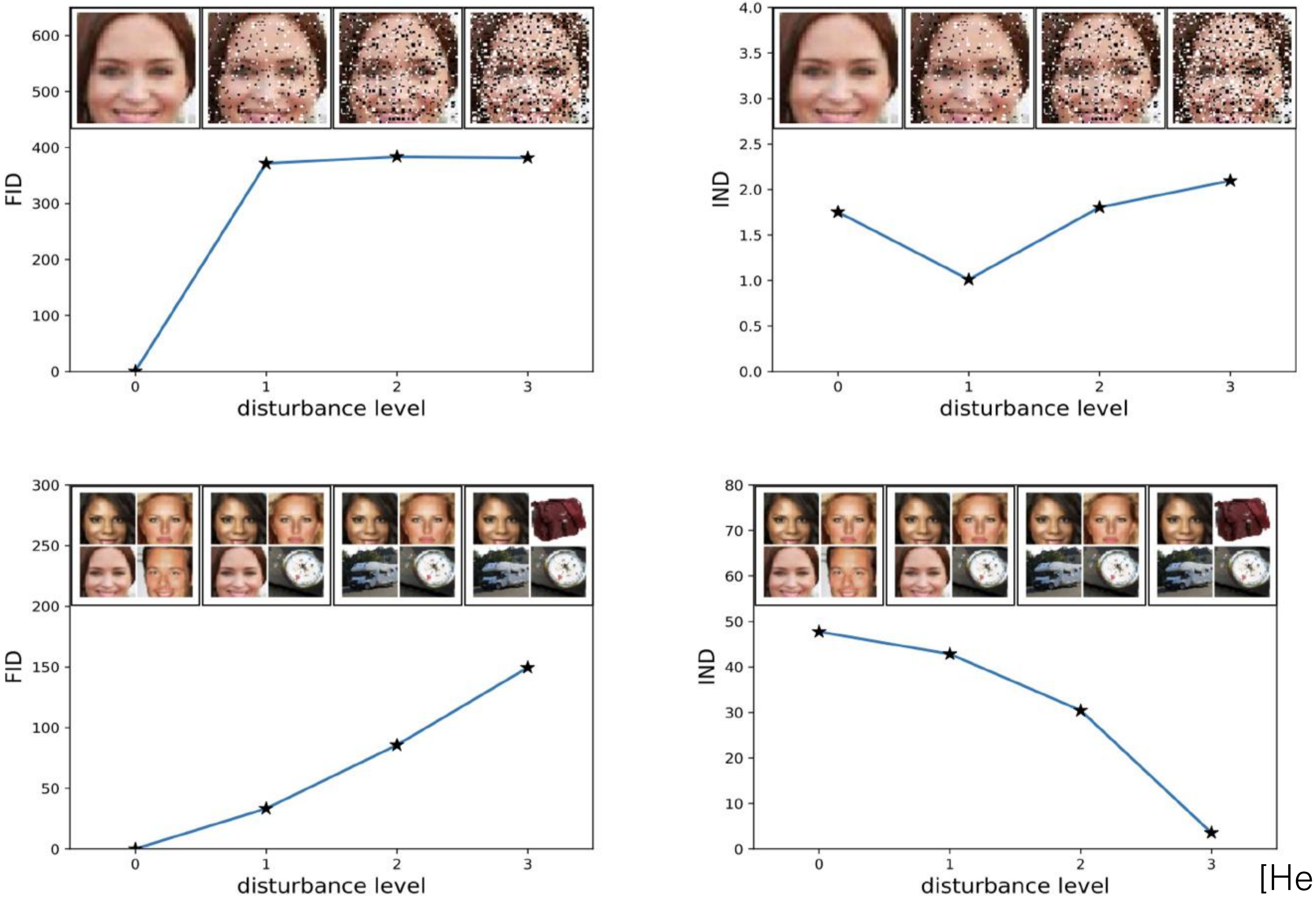
[Heusel et al, 2017]

Fréchet Inception Distance



[Heusel et al, 2017]

Fréchet Inception Distance



[Heusel et al, 2017]

Fréchet Inception Distance

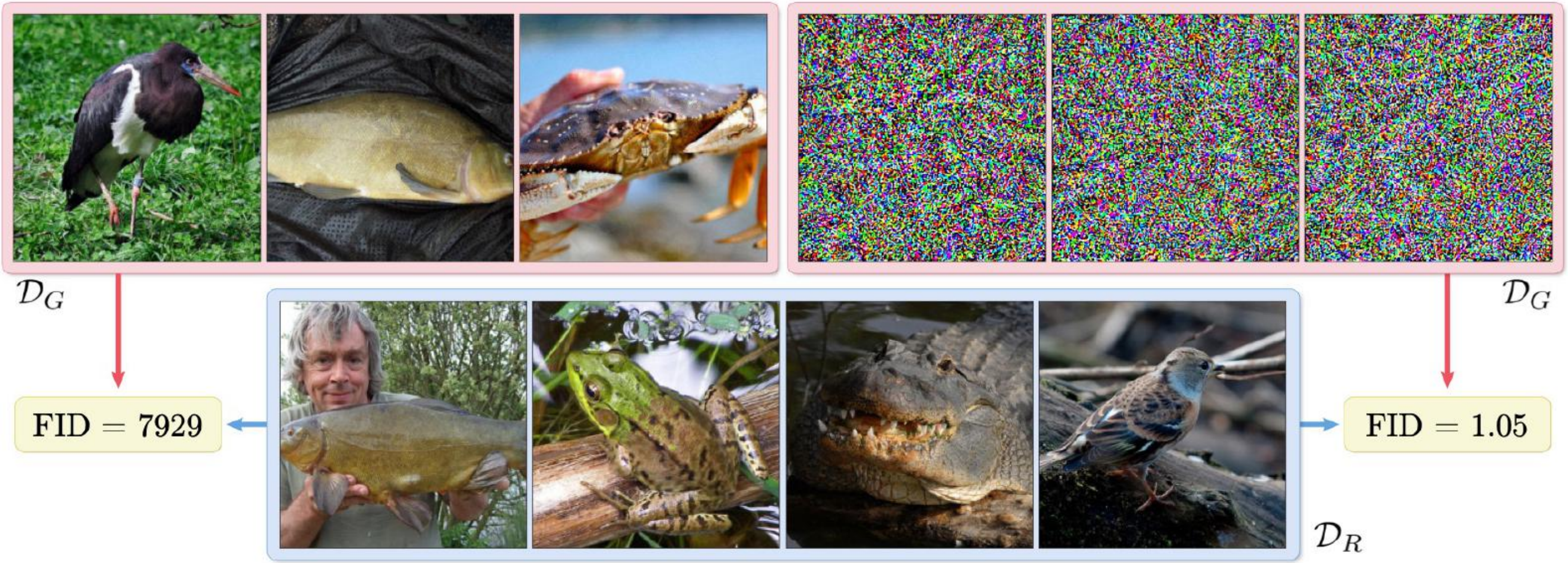


Figure 1. **Does the Fréchet Inception Distance (FID) accurately measure the distances between image distributions?** We generate datasets that demonstrate the unreliability of FID in judging perceptual (dis)similarities between image distributions. The top left box shows a sample of a dataset constructed by introducing imperceptible noise to each ImageNet image. Despite the remarkable visual similarity between this dataset and ImageNet (bottom box), an extremely large FID (almost 8000) between these two datasets showcases FID's failure to capture perceptual similarities. On the other hand, a remarkably low FID (almost 1.0) between a dataset of random noise images (samples shown in the top right box) and ImageNet illustrates FID's failure to capture perceptual dissimilarities.

One solution: Replace the Inception component of FID with a robustly trained counterpart!

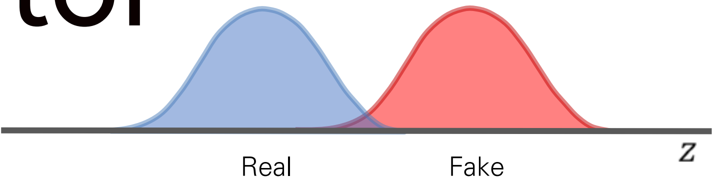
Generative Adversarial Networks

- Key pieces of GAN
  - Fast sampling
  - No inference
  - Notion of optimizing directly for what you care about – perceptual samples

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- **Theory of GANs**
- GAN Progression
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

GAN: Bayes-Optimal Discriminator



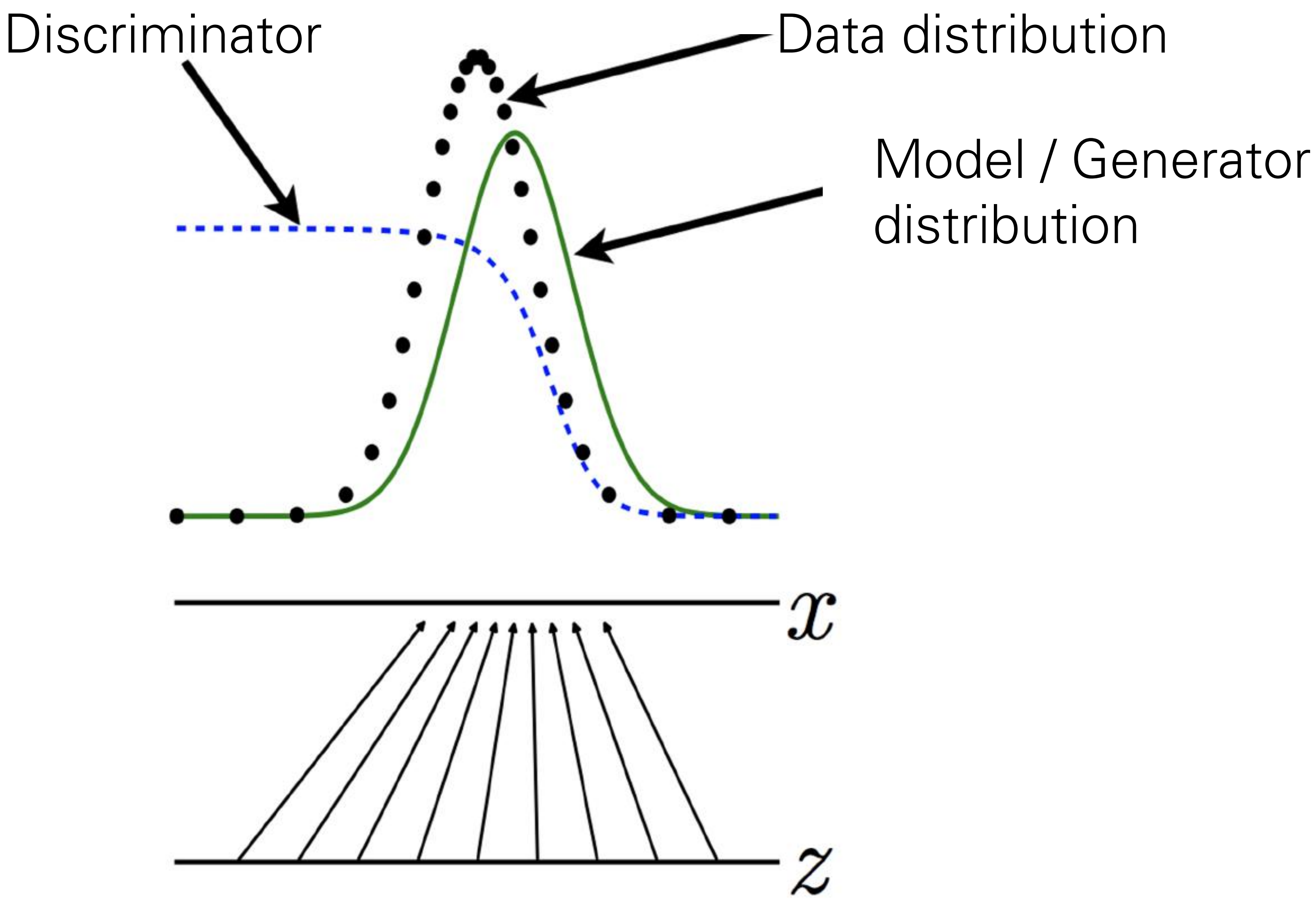
- What's the optimal discriminator given generated and true distributions?

$$
\begin{aligned}
V(G, D) &= \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \\
&= \int_x p_{\text{data}}(x)\log D(x)dx + \int_z p(z)\log(1 - D(G(z)))dz \\
&= \int_x p_{\text{data}}(x)\log D(x)dx + \int_x p_g(x)\log(1 - D(x))dx \\
&= \int_x \left[p_{\text{data}}(x)\log D(x) + p_g(x)\log(1 - D(x))\right]dx
\end{aligned}
$$

$$\nabla_y\left[a\log y + b\log(1-y)\right] = 0 \implies y^* = \frac{a}{a+b} \quad \forall \quad [a,b] \in \mathbb{R}^2 \backslash [0,0]$$

$$\implies D^*(x) = \frac{p_{\text{data}}(x)}{(p_{\text{data}}(x) + p_g(x))}$$

GAN: Bayes-Optimal Discriminator



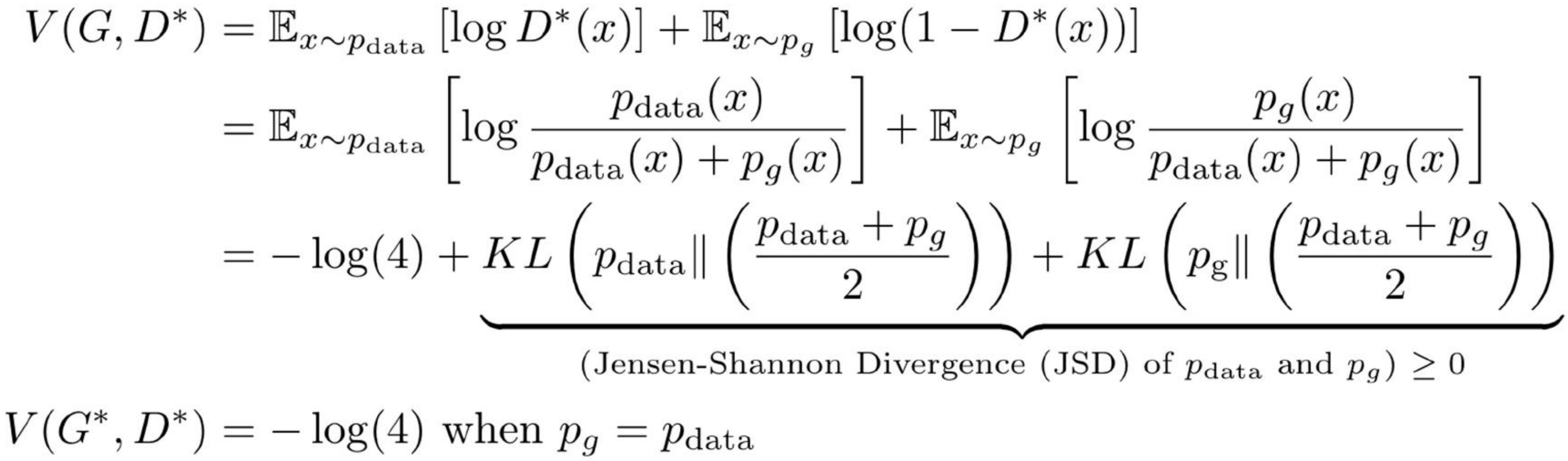
Discriminator

Data distribution

Model / Generator distribution

[Figure Source: Goodfellow NeurIPS 2016 Tutorial on GANs]

GAN: Generator Objective under Bayes-Optimal Discriminator D*?



$$
\begin{aligned}
V(G, D^*) &= \mathbb{E}_{x\sim p_{\text{data}}}\left[\log D^*(x)\right] + \mathbb{E}_{x\sim p_g}\left[\log(1 - D^*(x))\right] \\
&= \mathbb{E}_{x\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}\right] + \mathbb{E}_{x\sim p_g}\left[\log \frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)}\right] \\
&= -\log(4) + \underbrace{KL\left(p_{\text{data}} \| \left(\frac{p_{\text{data}} + p_g}{2}\right)\right) + KL\left(p_{\text{g}} \| \left(\frac{p_{\text{data}} + p_g}{2}\right)\right)}_{\text{(Jensen-Shannon Divergence (JSD) of } p_{\text{data}} \text{ and } p_g\text{)} \geq 0}
\end{aligned}
$$

$$V(G^*, D^*) = -\log(4) \text{ when } p_g = p_{\text{data}}$$

Compare this with ML objective: $KL(p_{data} \| p_g)$

Behaviors across divergence measures

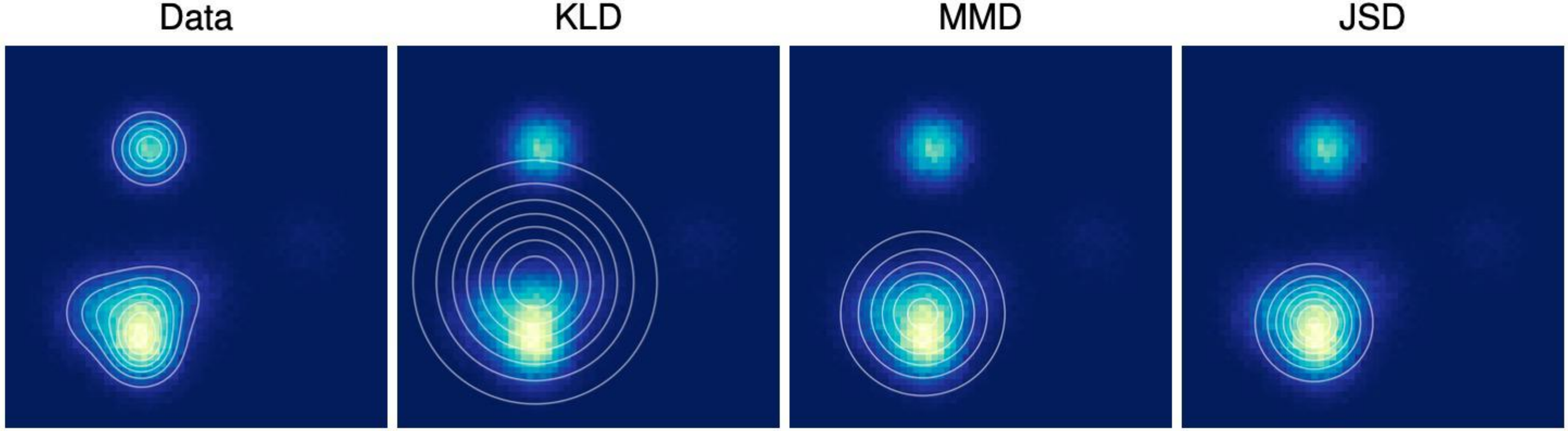
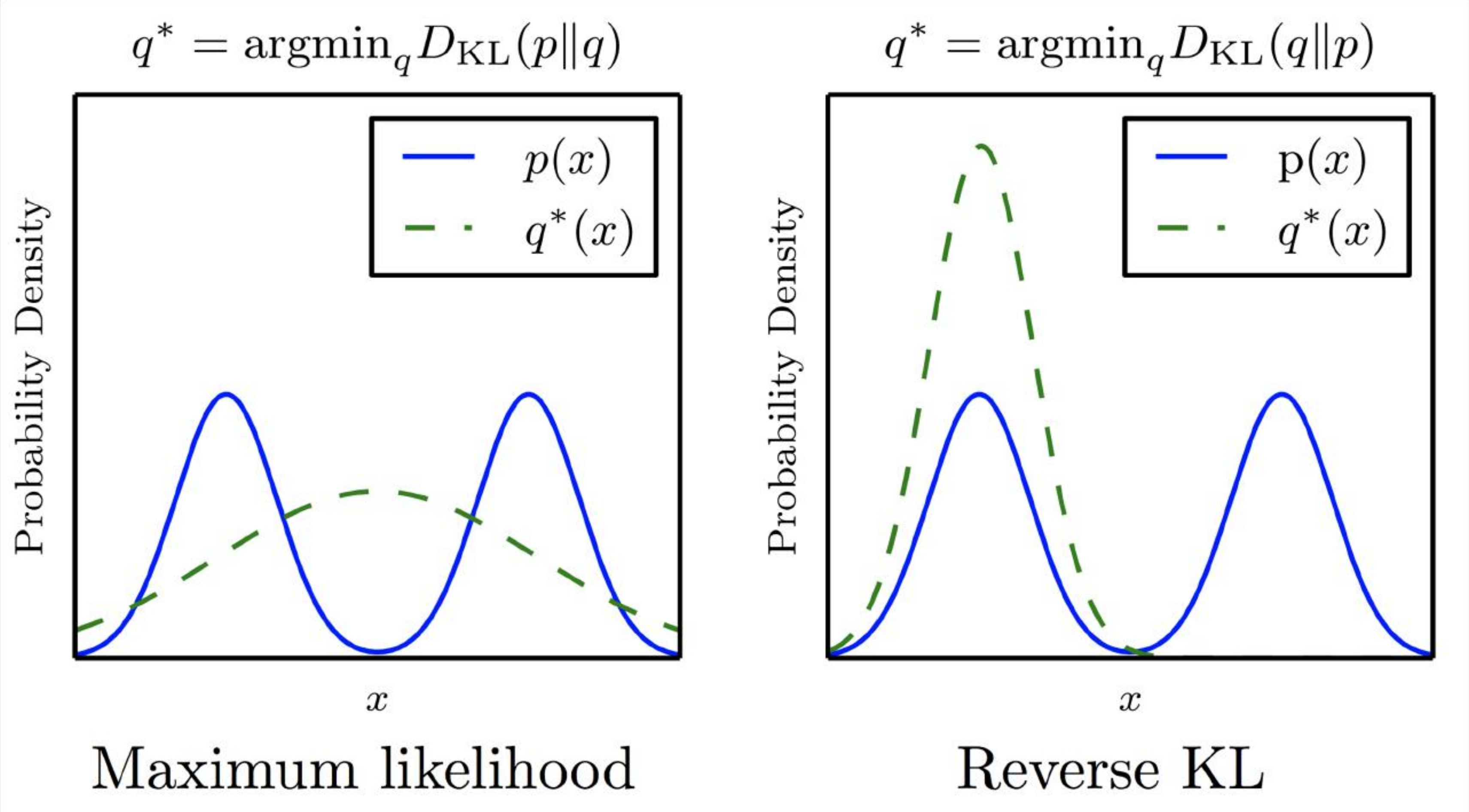


Figure 1: An isotropic Gaussian distribution was fit to data drawn from a mixture of Gaussians by either minimizing Kullback-Leibler divergence (KLD), maximum mean discrepancy (MMD), or Jensen-Shannon divergence (JSD). The different fits demonstrate different tradeoffs made by the three measures of distance between distributions.

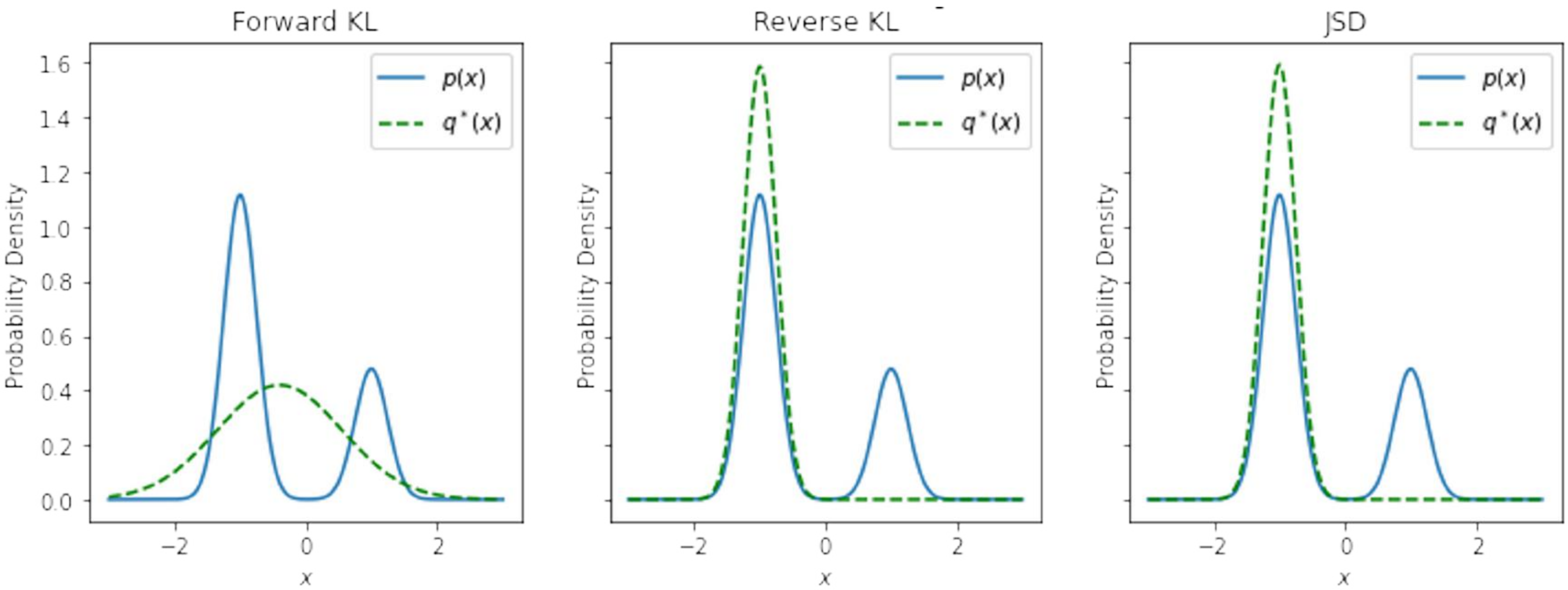
[“A note on the evaluation of generative models” -- Theis, Van den Oord, Bethge 2016]

Direction of KL divergence



Deep Learning Textbook (Goodfellow 2016) - Chapter 3

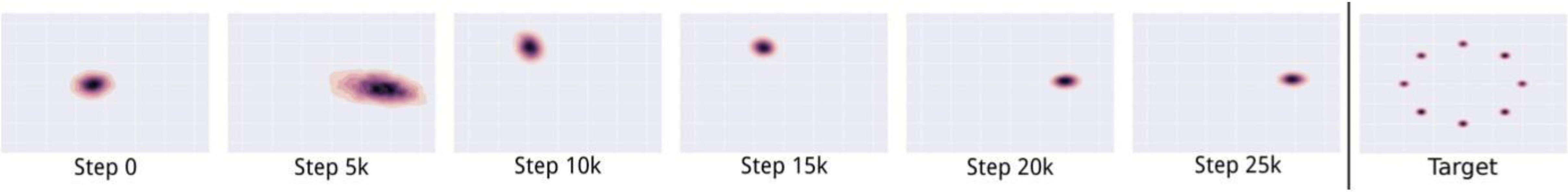
KL and JSD


Forward KL
Reverse KL
JSD
Probability Density
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−2
0
2
x
$p(x)$
$q^*(x)$

Mode covering vs Mode seeking: Tradeoffs

- For compression, one would prefer to ensure all points in the data distribution are assigned probability mass.
- For generating good samples, blurring across modes spoils perceptual quality because regions outside the data manifold are assigned non-zero probability mass.
- Picking one mode without assigning probability mass on points outside can produce “better-looking” samples.
- **Caveat:** More expressive density models can place probability mass more accurately. For example, using mixture of Gaussians as opposed to a single isotropic gaussian.

Mode Collapse



- Standard GAN training collapses when the true distribution is a mixture of gaussians!

(Figure from Metz et al. 2016)

Back to GANs

Recall

$$\min_G \underbrace{\max_D \mathbb{E}_{x\sim p_{\text{data}}}\left[\log D(x)\right] + \mathbb{E}_{z\sim p(z)}\left[\log(1 - D(G(z)))\right]}_{\text{Discriminator}}$$

Mini-Exercise

- Is it feasible to run the inner optimization to completion?
- For this specific objective, would it create problems if we were able to do so?

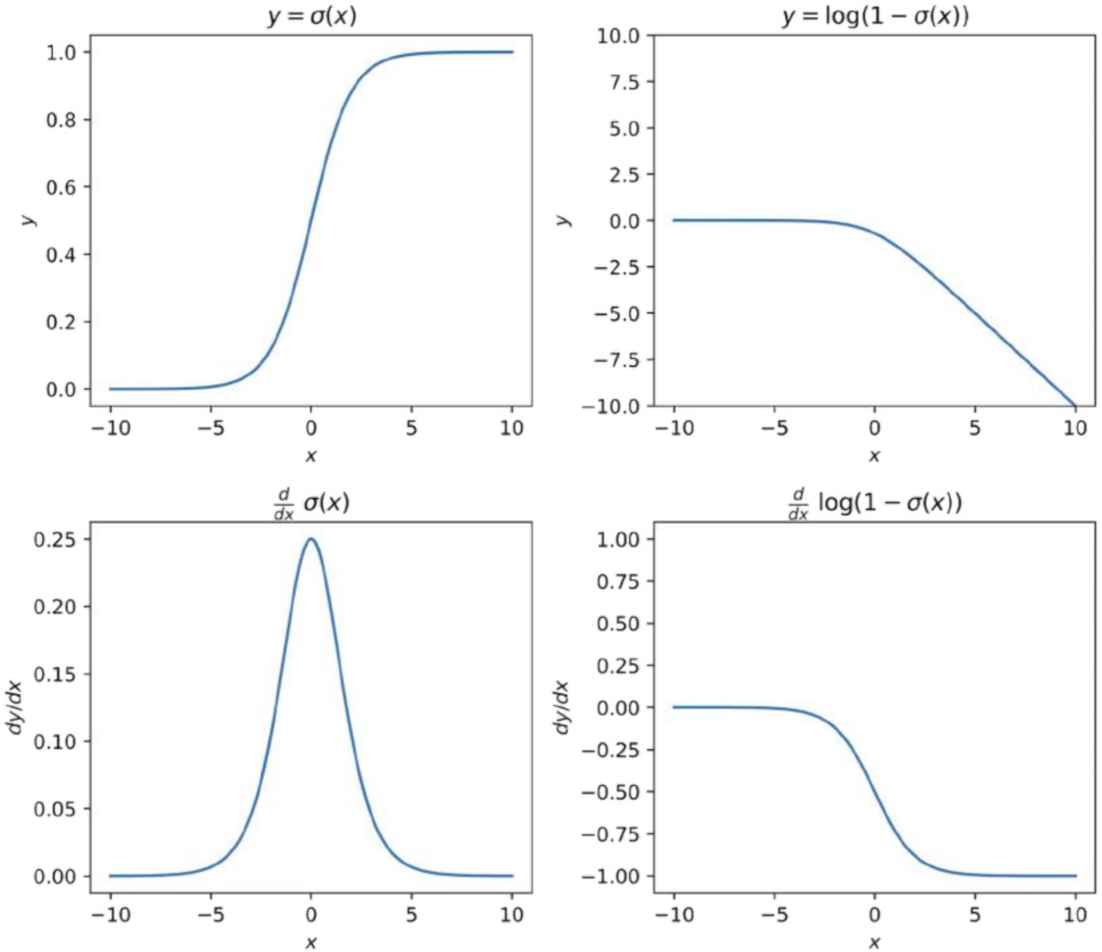
Discriminator Saturation

- Generator samples confidently classified as fake by the discriminator receive no gradient for the generator update.

$$\nabla_{G(z)} \log(1 - D(G(z)))$$ where

$$D(x) = \text{sigmoid}(x;\theta) = \sigma(x;\theta)$$

$$\nabla_x \sigma(x) = \sigma(x)(1 - \sigma(x))$$



Avoiding Discriminator Saturation: (1) Alternating Optimization

- Alternate gradient steps on discriminator and generator objectives

$$L^{(D)}(\theta_D, \theta_G) = -\mathbb{E}_{x \sim p_{\text{data}}} [\log D(x; \theta_D)] - \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z; \theta_G), \theta_D))]$$

$$L^{(G)}(\theta_D, \theta_G) = \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z; \theta_G), \theta_D))]$$

$$\theta_D := \theta_D - \alpha^{(D)} \nabla_{\theta_D} L^{(D)}(\theta_D, \theta_G)$$
$$\theta_G := \theta_G - \beta^{(G)} \nabla_{\theta_G} L^{(G)}(\theta_D, \theta_G)$$

- Balancing these two updates is hard for the zero-sum game

Avoiding Discriminator Saturation: (2) Non-Saturating Formulation

$$L^{(D)} = -\mathbb{E}_{x\sim p_{\text{data}}}\left[\log D(x)\right] - \mathbb{E}_{z\sim p(z)}\left[\log(1 - D(G(z)))\right]$$

$$L^{(G)} = -L^{D} \equiv \min_{G} \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z)))$$

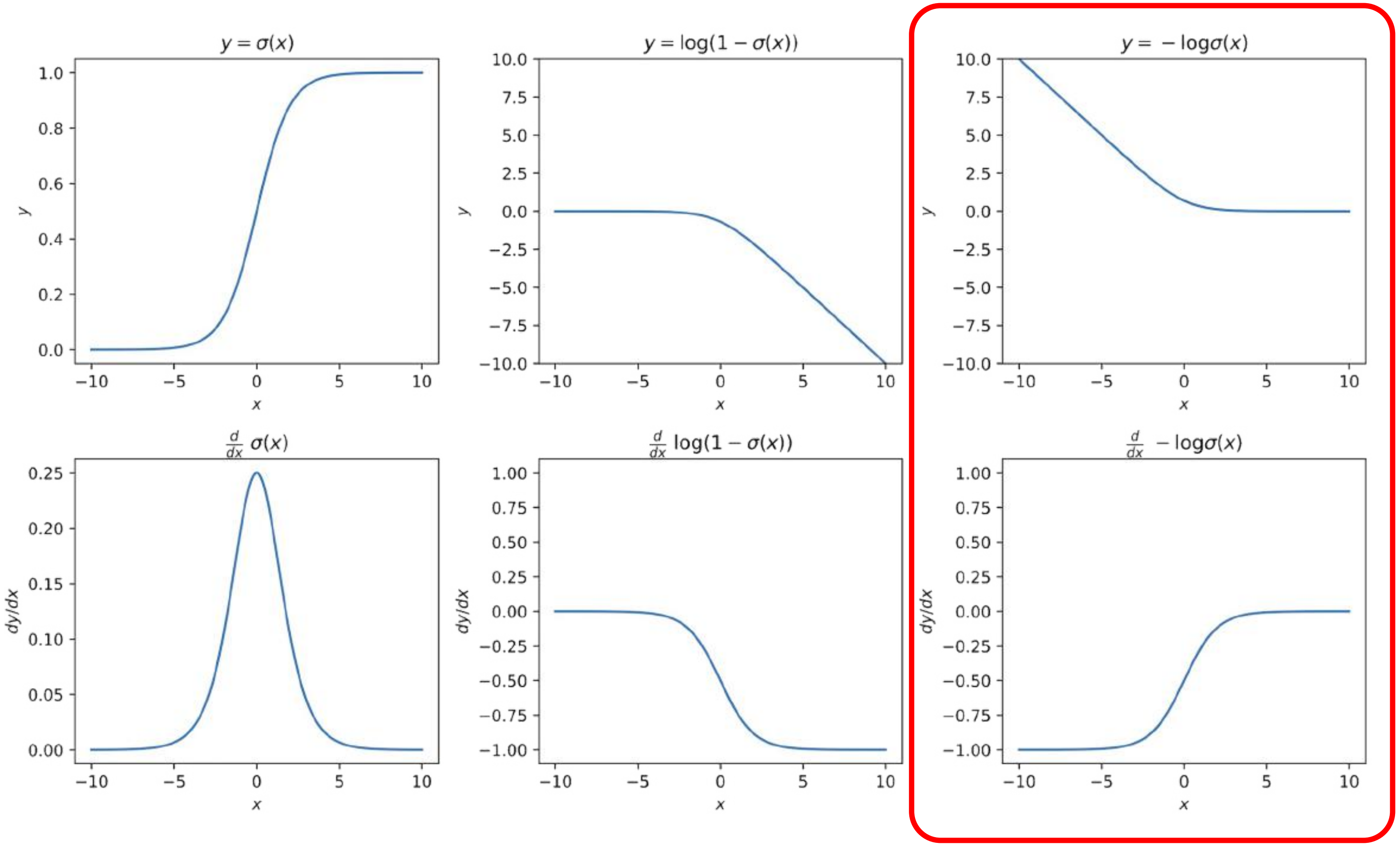
Not zero-sum

$$L^{(D)} = -\mathbb{E}_{x\sim p_{\text{data}}}\left[\log D(x)\right] - \mathbb{E}_{z\sim p(z)}\left[\log(1 - D(G(z)))\right]$$

$$L^{(G)} = -\mathbb{E}_{z\sim p(z)} \log(D(G(z)) \equiv \max_{G} \mathbb{E}_{z\sim p(z)} \log(D(G(z))$$

Avoiding Discriminator Saturation: (2) Non Saturating Formulation

- ORIGINAL ISSUE: Generator samples confidently classified as fake by the discriminator receive no gradient for the generator update.
- FIX: non-saturating loss for when discriminator confident about fake



Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - Improved Training of GANs (Salimans et al’16), Projected GAN (Sauer et al’21)
  - WGAN, WGAN-GP, Progressive GAN, SN-GAN, SAGAN
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN. VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

GAN Zoo

AN — Generative Adversarial Networks
3D-GAN — Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling acGAN
— Face Aging With Conditional Generative Adversarial Networks
AC-GAN — Conditional Image Synthesis With Auxiliary Classifier GANs
AdaGAN — AdaGAN: Boosting Generative Models
AEGAN — Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
AffGAN — Amortised MAP Inference for Image Super-resolution
AL-CGAN — Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
ALI — Adversarially Learned Inference
AMGAN — Generative Adversarial Nets with Labeled Data by Activation Maximization
AnoGAN — Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
ArtGAN — ArtGAN: Artwork Synthesis with Conditional Categorial GANs
b-GAN — b-GAN: Unified Framework of Generative Adversarial Networks
Bayesian GAN — Deep and Hierarchical Implicit Models
BEGAN — BEGAN: Boundary Equilibrium Generative Adversarial Networks
BiGAN — Adversarial Feature Learning
BS-GAN — Boundary-Seeking Generative Adversarial Networks
CGAN — Conditional Generative Adversarial Nets
CCGAN — Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks CatGAN —
Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks CoGAN —
Coupled Generative Adversarial Networks
Context-RNN-GAN — Contextual RNN-GANs for Abstract Reasoning Diagram Generation
C-RNN-GAN — C-RNN-GAN: Continuous recurrent neural networks with adversarial training
CS-GAN — Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets CVAE-
GAN — CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
CycleGAN — Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
DTN — Unsupervised Cross-Domain Image Generation
DCGAN — Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
DiscoGAN — Learning to Discover Cross-Domain Relations with Generative Adversarial Networks DR-GAN —
Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
DualGAN — DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
EBGAN — Energy-based Generative Adversarial Network
f-GAN — f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
GAWWN — Learning What and Where to Draw
GoGAN — Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
GP-GAN — GP-GAN: Towards Realistic High-Resolution Image Blending
IAN — Neural Photo Editing with Introspective Adversarial Networks
iGAN — Generative Visual Manipulation on the Natural Image Manifold
IcGAN — Invertible Conditional GANs for image editing
ID-CGAN- Image De-raining Using a Conditional Generative Adversarial Network
Improved GAN — Improved Techniques for Training GANs
InfoGAN — InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial
Nets LAGAN — Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for
Physics Synthesis LAPGAN — Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks
LR-GAN — LR-GAN: Layered Recursive Generative Adversarial Networks for Image Generation
LSGAN — Least Squares Generative Adversarial Networks
LS-GAN — Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities
MGAN — Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks
MAGAN — MAGAN: Margin Adaptation for Generative Adversarial Networks
MAD-GAN — Multi-Agent Diverse Generative Adversarial Networks
MalGAN — Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN
MaliGAN — Maximum-Likelihood Augmented Discrete Generative Adversarial Networks
MARTA-GAN — Deep Unsupervised Representation Learning for Remote Sensing Images
McGAN — McGan: Mean and Covariance Feature Matching GAN
MDGAN — Mode Regularized Generative Adversarial Networks
MedGAN — Generating Multi-label Discrete Electronic Health Records using Generative Adversarial Networks
MIX+GAN — Generalization and Equilibrium in Generative Adversarial Nets (GANs)
MPM-GAN — Message Passing Multi-Agent GANs
MV-BiGAN — Multi-view Generative Adversarial Networks
pix2pix — Image-to-Image Translation with Conditional Adversarial Networks
PPGN — Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space
PrGAN — 3D Shape Induction from 2D Views of Multiple Objects
RenderGAN — RenderGAN: Generating Realistic Labeled Data
RTT-GAN — Recurrent Topic-Transition GAN for Visual Paragraph Generation
SGAN — Stacked Generative Adversarial Networks
SGAN — Texture Synthesis with Spatial Generative Adversarial Networks
SAD-GAN — SAD-GAN: Synthetic Autonomous Driving using Generative Adversarial Networks
SalGAN — SalGAN: Visual Saliency Prediction with Generative Adversarial Networks
SEGAN — SEGAN: Speech Enhancement Generative Adversarial Network
SeGAN — SeGAN: Segmenting and Generating the Invisible
SeqGAN — SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient
SimGAN — Learning from Simulated and Unsupervised Images through Adversarial Training
SketchGAN — Adversarial Training For Sketch Retrieval
SL-GAN — Semi-Latent GAN: Learning to generate and modify facial images from attributes
Softmax-GAN — Softmax GAN
SRGAN — Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
S2GAN — Generative Image Modeling using Style and Structure Adversarial Networks
SSL-GAN — Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
StackGAN — StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks
TGAN — Temporal Generative Adversarial Nets
TAC-GAN — TAC-GAN — Text Conditioned Auxiliary Classifier Generative Adversarial Network
TP-GAN — Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving
Frontal View Synthesis Triple-GAN — Triple Generative Adversarial Nets
Unrolled GAN — Unrolled Generative Adversarial Networks
VGAN — Generating Videos with Scene Dynamics
VGAN — Generative Adversarial Networks as Variational Training of Energy Based Models
VAE-GAN — Autoencoding beyond pixels using a learned similarity metric
VariGAN — Multi-View Image Generation from a Single-View
ViGAN — Image Generation and Editing with Variational Info Generative AdversarialNetworks
WGAN — Wasserstein GAN
WGAN-GP — Improved Training of Wasserstein GANs
WaterGAN — WaterGAN: Unsupervised Generative Network to Enable Real-time Color Correction of Monocular
Underwater Images

Deep Hunt, blog by Avinash Hindupur
https://deephunt.in/the-gan-zoo-79597dc8c347

GAN Zoo

3D-ED-GAN - Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)

3D-IWGAN - Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)

3D-PhysNet - 3D-PhysNet: Learning the Intuitive Physics of Non-Rigid Object Deformations
3D-RecGAN - 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)

ABC-GAN - ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks (github)
ABC-GAN - GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference

AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs

acGAN - Face Aging With Conditional Generative Adversarial Networks
ACGAN - Coverless Information Hiding Based on Generative adversarial networks

acGAN - On-line Adaptative Curriculum Learning for GANs
ACtuAL - ACtuAL: Actor-Critic Under Adversarial Learning

AdaGAN - AdaGAN: Boosting Generative Models

Adaptive GAN - Customizing an Adversarial Example Generator with Class-Conditional GANs
AdvEntuRe - AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples

AdvGAN - Generating adversarial examples with adversarial networks
AE-GAN - AE-GAN: adversarial eliminating with GAN

AE-OT - Latent Space Optimal Transport for Generative Models

AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
AF-DCGAN - AF-DCGAN: Amplitude Feature Deep Convolutional GAN for Fingerprint Construction in Indoor Localization System

AffGAN - Amortised MAP Inference for Image Super-resolution
AIM - Generating Informative and Diverse Conversational Responses via Adversarial Information Maximization

AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts

ALI - Adversarially Learned Inference (github)
AlignGAN - AlignGAN: Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks

AlphaGAN - AlphaGAN: Generative adversarial networks for natural image matting
AM-GAN - Activation Maximization Generative Adversarial Nets

AmbientGAN - AmbientGAN: Generative models from lossy measurements (github)

AMC-GAN - Video Prediction with Appearance and Motion Conditions
AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery

APD - Adversarial Distillation of Bayesian Neural Network Posteriors
APE-GAN - APE-GAN: Adversarial Perturbation Elimination with GAN

ARAE - Adversarially Regularized Autoencoders for Generating Discrete Structures (github)

ARDA - Adversarial Representation Learning for Domain Adaptation
ARIGAN - ARIGAN: Synthetic Arabidopsis Plants using Generative Adversarial Network

ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
ASDL-GAN - Automatic Steganographic Distortion Learning Using a Generative Adversarial Network

ATA-GAN - Attention-Aware Generative Adversarial Networks (ATA-GANs)

Attention-GAN - Attention-GAN for Object Transfiguration in Wild Images
AttGAN - Arbitrary Facial Attribute Editing: Only Change What You Want (github)

AttnGAN - AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks (github)
AVID - AVID: Adversarial Visual Irregularity Detection

B-DCGAN - B-DCGAN:Evaluation of Binarized DCGAN for FPGA

b-GAN - Generative Adversarial Nets from a Density Ratio Estimation Perspective
BAGAN - BAGAN: Data Augmentation with Balancing GAN

Bayesian GAN - Deep and Hierarchical Implicit Models
Bayesian GAN - Bayesian GAN (github)

BCGAN - Bayesian Conditional Generative Adverserial Networks

BCGAN - Bidirectional Conditional Generative Adversarial networks
BEAM - Boltzmann Encoded Adversarial Machines

BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
BEGAN-CS - Escaping from Collapsing Modes in a Constrained Space

Bellman GAN - Distributional Multivariate Policy Evaluation and Exploration with the Bellman GAN

BGAN - Binary Generative Adversarial Networks for Image Retrieval (github)
Bi-GAN - Autonomously and Simultaneously Refining Deep Neural Network Parameters by a Bi-Generative Adversarial Network Aided Genetic Algorithm

BicycleGAN - Toward Multimodal Image-to-Image Translation (github)
BiGAN - Adversarial Feature Learning

BinGAN - BinGAN: Learning Compact Binary Descriptors with a Regularized GAN

BourGAN - BourGAN: Generative Networks with Metric Embeddings
BranchGAN - Branched Generative Adversarial Networks for Multi-Scale Image Manifold Learning

BRE - Improving GAN Training via Binarized Representation Entropy (BRE) Regularization (github)
BridgeGAN - Generative Adversarial Frontal View to Bird View Synthesis

BS-GAN - Boundary-Seeking Generative Adversarial Networks

BubGAN - BubGAN: Bubble Generative Adversarial Networks for Synthesizing Realistic Bubbly Flow Images
BWGAN - Banach Wasserstein GAN

C-GAN - Face Aging with Contextual Generative Adversarial Nets
C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training (github)

CA-GAN - Composition-aided Sketch-realistic Portrait Generation

CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks (github)
CAN - CAN: Creative Adversarial Networks, Generating Art by Learning About Styles and Deviating from Style Norms

CapsGAN - CapsGAN: Using Dynamic Routing for Generative Adversarial Networks
CapsuleGAN - CapsuleGAN: Generative Adversarial Capsule Network

CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks

CatGAN - CatGAN: Coupled Adversarial Transfer for Domain Generation
CausalGAN - CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training

CC-GAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks (github)
cd-GAN - Conditional Image-to-Image Translation

CDcGAN - Simultaneously Color-Depth Super-Resolution with Conditional Generative Adversarial Network

CE-GAN - Deep Learning for Imbalance Data Classification using Class Expert Generative Adversarial Network
CFG-GAN - Composite Functional Gradient Learning of Generative Adversarial Models

CGAN - Conditional Generative Adversarial Nets
CGAN - Controllable Generative Adversarial Network

Chekhov GAN - An Online Learning Approach to Generative Adversarial Networks

ciGAN - Conditional Infilling GANs for Data Augmentation in Mammogram Classification
CinCGAN - Unsupervised Image Super-Resolution using Cycle-in-Cycle Generative Adversarial Networks

CipherGAN - Unsupervised Cipher Cracking Using Discrete GANs
ClusterGAN - ClusterGAN : Latent Space Clustering in Generative Adversarial Networks

CM-GAN - CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning

CoAtt-GAN - Are You Talking to Me? Reasoned Visual Dialog Generation through Adversarial Learning
CoGAN - Coupled Generative Adversarial Networks

ComboGAN - ComboGAN: Unrestrained Scalability for Image Domain Translation (github)
ConceptGAN - Learning Compositional Visual Concepts with Mutual Consistency

Conditional cycleGAN - Conditional CycleGAN for Attribute Guided Face Image Generation

constrast-GAN - Generative Semantic Manipulation with Contrasting GAN
Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation

CorrGAN - Correlated discrete data generation using adversarial training
Coulomb GAN - Coulomb GANs: Provably Optimal Nash Equilibria via Potential Fields

Cover-GAN - Generative Steganography with Kerckhoffs' Principle based on Generative Adversarial Networks

cowboy - Defending Against Adversarial Attacks by Leveraging an Entire GAN
CR-GAN - CR-GAN: Learning Complete Representations for Multi-view Generation

Cramèr GAN - The Cramer Distance as a Solution to Biased Wasserstein Gradients
Cross-GAN - Crossing Generative Adversarial Networks for Cross-View Person Re-identification

crVAE-GAN - Channel-Recurrent Variational Autoencoders

CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
CSG - Speech-Driven Expressive Talking Lips with Conditional Sequential Generative Adversarial Networks

CT-GAN - CT-GAN: Conditional Transformation Generative Adversarial Network for Image Attribute Modification
CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training

CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (github)

D-GAN - Differential Generative Adversarial Networks: Synthesizing Non-linear Facial Variations with Limited Number of Training Data
D-WCGAN - I-vector Transformation Using Conditional Generative Adversarial Networks for Short Utterance Speaker Verification

D2GAN - Dual Discriminator Generative Adversarial Nets
D2IA-GAN - Tagging like Humans: Diverse and Distinct Image Annotation

DA-GAN - DA-GAN: Instance-level Image Translation by Deep Attention Generative Adversarial Networks (with Supplementary Materials)

DADA - DADA: Deep Adversarial Data Augmentation for Extremely Low Data Regime Classification
DAGAN - Data Augmentation Generative Adversarial Networks

DAN - Distributional Adversarial Networks
DBLRGAN - Adversarial Spatio-Temporal Learning for Video Deblurring

DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (github)

DE-GAN - Generative Adversarial Networks with Decoder-Encoder Output Noise
DeblurGAN - DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks (github)

DeepFD - Learning to Detect Fake Face Images in the Wild
Defense-GAN - Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models (github)

Defo-Net - Defo-Net: Learning Body Deformation using Generative Adversarial Networks

DeliGAN - DeLiGAN : Generative Adversarial Networks for Diverse and Limited Data (github)
DF-GAN - Learning Disentangling and Fusing Networks for Face Completion Under Structured Occlusions

DialogWAE - DialogWAE: Multimodal Response Generation with Conditional Wasserstein Auto-Encoder

DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
DistanceGAN - One-Sided Unsupervised Domain Mapping

DM-GAN - Dual Motion GAN for Future-Flow Embedded Video Prediction

DMGAN - Disconnected Manifold Learning for Generative Adversarial Networks
DNA-GAN - DNA-GAN: Learning Disentangled Representations from Multi-Attribute Images

DOPING - DOPING: Generative Data Augmentation for Unsupervised Anomaly Detection with GAN
dp-GAN - Differentially Private Releasing via Deep Generative Model

DP-GAN - DP-GAN: Diversity-Promoting Generative Adversarial Network for Generating Informative and Diversified Text

DPGAN - Differentially Private Generative Adversarial Network
DR-GAN - Representation Learning by Rotating Your Faces

DRAGAN - How to Train Your DRAGAN (github)
Dropout-GAN - Dropout-GAN: Learning from a Dynamic Ensemble of Discriminators

DRPAN - Discriminative Region Proposal Adversarial Networks for High-Quality Image-to-Image Translation

DSH-GAN - Deep Semantic Hashing with Generative Adversarial Networks
DSP-GAN - Depth Structure Preserving Scene Image Generation

DTLC-GAN - Generative Adversarial Image Synthesis with Decision Tree Latent Controller
DTN - Unsupervised Cross-Domain Image Generation

DTR-GAN - DTR-GAN: Dilated Temporal Relational Adversarial Network for Video Summarization

DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
Dualing GAN - Dualing GANs

DVGAN - Human Motion Modeling using DVGANs
Dynamics Transfer GAN - Dynamics Transfer GAN: Generating Video by Transferring Arbitrary Temporal Dynamics from a Source Video to a Single Target Image

E-GAN - Evolutionary Generative Adversarial Networks

EAR - Generative Model for Heterogeneous Inference
EBGAN - Energy-based Generative Adversarial Network

ecGAN - eCommerceGAN : A Generative Adversarial Network for E-commerce
ED//GAN - Stabilizing Training of Generative Adversarial Networks through Regularization

Editable GAN - Editable Generative Adversarial Networks: Generating and Editing Faces Simultaneously

EGAN - Enhanced Experience Replay Generation for Efficient Reinforcement Learning
EL-GAN - EL-GAN: Embedding Loss Driven Generative Adversarial Networks for Lane Detection

ELEGANT - ELEGANT: Exchanging Latent Encodings with GAN for Transferring Multiple Face Attributes
EnergyWGAN - Energy-relaxed Wassertein GANs (EnergyWGAN): Towards More Stable and High Resolution Image Generation

ESRGAN - ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks

ExGAN - Eye In-Painting with Exemplar Generative Adversarial Networks
ExposureGAN - Exposure: A White-Box Photo Post-Processing Framework (github)

ExprGAN - ExprGAN: Facial Expression Editing with Controllable Expression Intensity
f-CLSWGAN - Feature Generating Networks for Zero-Shot Learning

f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization

FairGAN - FairGAN: Fairness-aware Generative Adversarial Networks
Fairness GAN - Fairness GAN

FakeGAN - Detecting Deceptive Reviews using Generative Adversarial Networks
FBGAN - Feedback GAN (FBGAN) for DNA: a Novel Feedback-Loop Architecture for Optimizing Protein Functions

FBGAN - Featurized Bidirectional GAN: Adversarial Defense via Adversarially Learned Semantic Inference

FC-GAN - Fast-converging Conditional Generative Adversarial Networks for Image Synthesis
FF-GAN - Towards Large-Pose Face Frontalization in the Wild

FGGAN - Adversarial Learning for Fine-grained Image Search
Fictitious GAN - Fictitious GAN: Training GANs with Historical Models

FIGAN - Frame Interpolation with Multi-Scale Deep Loss Functions and Generative Adversarial Networks

Fila-GAN - Synthesizing Filamentary Structured Images with GANs
First Order GAN - First Order Generative Adversarial Networks (github)

Fisher GAN - Fisher GAN
Flow-GAN - Flow-GAN: Bridging implicit and prescribed learning in generative models

FrankenGAN - rankenGAN: Guided Detail Synthesis for Building Mass-Models Using Style-Synchonized GANs

FSEGAN - Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition
FTGAN - Hierarchical Video Generation from Orthogonal Information: Optical Flow and Texture

FusedGAN - Semi-supervised FusedGAN for Conditional Image Generation
FusionGAN - Learning to Fuse Music Genres with Generative Adversarial Dual Learning

FusionGAN - Generating a Fusion Image: One's Identity and Another's Shape

G2-GAN - Geometry Guided Adversarial Facial Expression Synthesis
GAAN - Generative Adversarial Autoencoder Networks

GAF - Generative Adversarial Forests for Better Conditioned Adversarial Learning
GAGAN - GAGAN: Geometry-Aware Generative Adverserial Networks

GAIA - Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions

GAIN - GAIN: Missing Data Imputation using Generative Adversarial Nets
GAMN - Generative Adversarial Mapping Networks

GAN - Generative Adversarial Networks (github)
GAN Lab - GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation

GAN Q-learning - GAN Q-learning

GAN-AD - Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series
GAN-ATV - A Novel Approach to Artistic Textual Visualization via GAN

GAN-CLS - Generative Adversarial Text to Image Synthesis (github)
GAN-RS - Towards Qualitative Advancement of Underwater Machine Vision with Generative Adversarial Networks

GAN-SD - Virtual-Taobao: Virtualizing Real-world Online Retail Environment for Reinforcement Learning

GAN-sep - GANs for Biological Image Synthesis (github)
GAN-VFS - Generative Adversarial Network-based Synthesis of Visible Faces from Polarimetric Thermal Faces

GAN-Word2Vec - Adversarial Training of Word2Vec for Basket Completion
GANAX - GANAX: A Unified MIMD-SIMD Acceleration for Generative Adversarial Networks

GANCS - Deep Generative Adversarial Networks for Compressed Sensing Automates MRI

GANDI - Guiding the search in continuous state-action spaces by learning an action sampling distribution from off-target samples
GANG - GANGs: Generative Adversarial Network Games

GANG - Beyond Local Nash Equilibria for Adversarial Networks
GANosaic - GANosaic: Mosaic Creation with Generative Texture Manifolds

GANVO - GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks

GAP - Context-Aware Generative Adversarial Privacy
GAP - Generative Adversarial Privacy

GATS - Sample-Efficient Deep RL with Generative Adversarial Tree Search
GAWWN - Learning What and Where to Draw (github)

GC-GAN - Geometry-Contrastive Generative Adversarial Network for Facial Expression Synthesis

GcGAN - Geometry-Consistent Adversarial Networks for One-Sided Unsupervised Domain Mapping
GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data (github)

GeoGAN - Generating Instance Segmentation Annotation by Geometry-guided GAN
Geometric GAN - Geometric GAN

GIN - Generative Invertible Networks (GIN): Pathophysiology-Interpretable Feature Mapping and Virtual Patient Generation

GLCA-GAN - Global and Local Consistent Age Generative Adversarial Networks
GM-GAN - Gaussian Mixture Generative Adversarial Networks for Diverse Datasets, and the Unsupervised Clustering of Images

GMAN - Generative Multi-Adversarial Networks
GMM-GAN - Towards Understanding the Dynamics of Generative Adversarial Networks

GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking

GONet - GONet: A Semi-Supervised Deep Learning Approach For Traversability Estimation
GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending (github)

GP-GAN - GP-GAN: Gender Preserving GAN for Synthesizing Faces from Landmarks
GPU - A generative adversarial framework for positive-unlabeled classification

GRAN - Generating images with recurrent adversarial networks (github)

Graphical-GAN - Graphical Generative Adversarial Networks
GraphSGAN - Semi-supervised Learning on Graphs with Generative Adversarial Nets

GraspGAN - Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping
GT-GAN - Deep Graph Translation

HAN - Chinese Typeface Transformation with Hierarchical Adversarial Network

HAN - Bidirectional Learning for Robust Neural Networks
HiGAN - Exploiting Images for Video Recognition with Hierarchical Generative Adversarial Networks

HP-GAN - HP-GAN: Probabilistic 3D human motion prediction via GAN
HR-DCGAN - High-Resolution Deep Convolutional Generative Adversarial Networks

hredGAN - Multi-turn Dialogue Response Generation in an Adversarial Learning framework

IAN - Neural Photo Editing with Introspective Adversarial Networks (github)
IcGAN - Invertible Conditional GANs for image editing (github)

ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
IdCycleGAN - Face Translation between Images and Videos using Identity-aware CycleGAN

IFcVAEGAN - Conditional Autoencoders with Adversarial Information Factorization

iGAN - Generative Visual Manipulation on the Natural Image Manifold (github)
IGMM-GAN - Coupled IGMM-GANs for deep multimodal anomaly detection in human mobility data

Improved GAN - Improved Techniques for Training GANs (github)
In2I - In2I : Unsupervised Multi-Image-to-Image Translation Using Generative Adversarial Networks

InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets (github)

IntroVAE - IntroVAE: Introspective Variational Autoencoders for Photographic Image Synthesis
IR2VI - IR2VI: Enhanced Night Environmental Perception by Unsupervised Thermal Image Translation

IRGAN - IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval models

IRGAN - Generative Adversarial Nets for Information Retrieval: Fundamentals and Advances
ISGAN - Invisible Steganography via Generative Adversarial Network

ISP-GPM - Inner Space Preserving Generative Pose Machine

Iterative-GAN - Two Birds with One Stone: Iteratively Learn Facial Attributes with GANs (github)
IterGAN - IterGANs: Iterative GANs to Learn and Control 3D Object Transformation

IVE-GAN - IVE-GAN: Invariant Encoding Generative Adversarial Networks
iVGAN - Towards an Understanding of Our World by GANing Videos in the Wild (github)

IWGAN - On Unifying Deep Generative Models

JointGAN - JointGAN: Multi-Domain Joint Distribution Learning with Generative Adversarial Nets
JR-GAN - JR-GAN: Jacobian Regularization for Generative Adversarial Networks

KBGAN - KBGAN: Adversarial Learning for Knowledge Graph Embeddings
KGAN - KGAN: How to Break The Minimax Game in GAN

l-GAN - Representation Learning and Adversarial Generation of 3D Point Clouds

LAC-GAN - Grounded Language Understanding for Manipulation Instructions Using GAN-Based Classification
LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis

LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks (github)
LB-GAN - Load Balanced GANs for Multi-view Face Image Synthesis

LBT - Learning Implicit Generative Models by Teaching Explicit Ones

LCC-GAN - Adversarial Learning with Local Coordinate Coding
LD-GAN - Linear Discriminant Generative Adversarial Networks

LDAN - Label Denoising Adversarial Network (LDAN) for Inverse Lighting of Face Images
LeakGAN - Long Text Generation via Adversarial Training with Leaked Information

LeGAN - Likelihood Estimation for Generative Adversarial Networks

LGAN - Global versus Localized Generative Adversarial Nets
Lipizzaner - Towards Distributed Coevolutionary GANs

LR-GAN - LR-GAN: Layered Recursive Generative Adversarial Networks for Image Generation
LS-GAN - Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities

LSGAN - Least Squares Generative Adversarial Networks

M-AAE - Mask-aware Photorealistic Face Attribute Manipulation
MAD-GAN - Multi-Agent Diverse Generative Adversarial Networks

MAGAN - MAGAN: Margin Adaptation for Generative Adversarial Networks
MAGAN - MAGAN: Aligning Biological Manifolds

MalGAN - Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN

MaliGAN - Maximum-Likelihood Augmented Discrete Generative Adversarial Networks
manifold-WGAN - Manifold-valued Image Generation with Wasserstein Adversarial Networks

MARTA-GAN - Deep Unsupervised Representation Learning for Remote Sensing Images
MaskGAN - MaskGAN: Better Text Generation via Filling in the ______

MC-GAN - Multi-Content GAN for Few-Shot Font Style Transfer (github)

MC-GAN - MC-GAN: Multi-conditional Generative Adversarial Network for Image Synthesis
McGAN - McGan: Mean and Covariance Feature Matching GAN

MD-GAN - Learning to Generate Time-Lapse Videos Using Multi-Stage Dynamic Generative Adversarial Networks
MDGAN - Mode Regularized Generative Adversarial Networks

MedGAN - Generating Multi-label Discrete Electronic Health Records using Generative Adversarial Networks

MedGAN - MedGAN: Medical Image Translation using GANs
MEGAN - MEGAN: Mixture of Experts of Generative Adversarial Networks for Multimodal Image Generation

MelanoGAN - MelanoGANs: High Resolution Skin Lesion Synthesis with GANs
memoryGAN - Memorization Precedes Generation: Learning Unsupervised GANs with Memory Networks

MeRGAN - Memory Replay GANs: learning to generate images from new categories without forgetting

MGAN - Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks (github)
MGGAN - Multi-Generator Generative Adversarial Nets

MGGAN - MGGAN: Solving Mode Collapse using Manifold Guided Training
MIL-GAN - Multimodal Storytelling via Generative Adversarial Imitation Learning

MinLGAN - Anomaly Detection via Minimum Likelihood Generative Adversarial Networks

MIX+GAN - Generalization and Equilibrium in Generative Adversarial Nets (GANs)
MIXGAN - MIXGAN: Learning Concepts from Different Domains for Mixture Generation

MLGAN - Metric Learning-based Generative Adversarial Network
MMC-GAN - A Multimodal Classifier Generative Adversarial Network for Carry and Place Tasks from Ambiguous Language Instructions

MMD-GAN - MMD GAN: Towards Deeper Understanding of Moment Matching Network (github)

MMGAN - MMGAN: Manifold Matching Generative Adversarial Network for Generating Images
MoCoGAN - MoCoGAN: Decomposing Motion and Content for Video Generation (github)

Modified GAN-CLS - Generate the corresponding Image from Text Description using Modified GAN-CLS Algorithm
ModularGAN - Modular Generative Adversarial Networks

MolGAN - MolGAN: An implicit generative model for small molecular graphs

MPM-GAN - Message Passing Multi-Agent GANs
MS-GAN - Temporal Coherency based Criteria for Predicting Video Frames using Deep Multi-stage Generative Adversarial Networks

MTGAN - MTGAN: Speaker Verification through Multitasking Triplet Generative Adversarial Networks
MuseGAN - MuseGAN: Symbolic-domain Music Generation and Accompaniment with Multi-track Sequential Generative Adversarial Networks

MV-BiGAN - Multi-view Generative Adversarial Networks

N2RPP - N2RPP: An Adversarial Network to Rebuild Plantar Pressure for ACLD Patients
NAN - Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing

NCE-GAN - Dihedral angle prediction using generative adversarial networks
ND-GAN - Novelty Detection with GAN

NetGAN - NetGAN: Generating Graphs via Random Walks

OCAN - One-Class Adversarial Nets for Fraud Detection
OptionGAN - OptionGAN: Learning Joint Reward-Policy Options using Generative Adversarial Inverse Reinforcement Learning

ORGAN - Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models
ORGAN - 3D Reconstruction of Incomplete Archaeological Objects Using a Generative Adversary Network

OT-GAN - Improving GANs Using Optimal Transport

PacGAN - PacGAN: The power of two samples in generative adversarial networks
PAN - Perceptual Adversarial Networks for Image-to-Image Transformation

PassGAN - PassGAN: A Deep Learning Approach for Password Guessing
PD-WGAN - Primal-Dual Wasserstein GAN

Perceptual GAN - Perceptual Generative Adversarial Networks for Small Object Detection

PGAN - Probabilistic Generative Adversarial Networks
PGD-GAN - Solving Linear Inverse Problems Using GAN Priors: An Algorithm with Provable Guarantees

PGGAN - Patch-Based Image Inpainting with Generative Adversarial Networks
PIONEER - Pioneer Networks: Progressively Growing Generative Autoencoder

Pip-GAN - Pipeline Generative Adversarial Networks for Facial Images Generation with Multiple Attributes

pix2pix - Image-to-Image Translation with Conditional Adversarial Networks (github)
pix2pixHD - High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs (github)

PixelGAN - PixelGAN Autoencoders
PM-GAN - PM-GANs: Discriminative Representation Learning for Action Recognition Using Partial-modalities

PN-GAN - Pose-Normalized Image Generation for Person Re-identification

POGAN - Perceptually Optimized Generative Adversarial Network for Single Image Dehazing
Pose-GAN - The Pose Knows: Video Forecasting by Generating Pose Futures

PP-GAN - Privacy-Protective-GAN for Face De-identification
PPAN - Privacy-Preserving Adversarial Networks

PPGN - Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space

PrGAN - 3D Shape Induction from 2D Views of Multiple Objects
ProGanSR - A Fully Progressive Approach to Single-Image Super-Resolution

Progressive GAN - Progressive Growing of GANs for Improved Quality, Stability, and Variation (github)
PS-GAN - Pedestrian-Synthesis-GAN: Generating Pedestrian Data in Real Scene and Beyond

PSGAN - Learning Texture Manifolds with the Periodic Spatial GAN

PSGAN - PSGAN: A Generative Adversarial Network for Remote Sensing Image Pan-Sharpening
PS²-GAN - High-Quality Facial Photo-Sketch Synthesis Using Multi-Adversarial Networks

RadialGAN - RadialGAN: Leveraging multiple datasets to improve target-specific predictive models using Generative Adversarial Networks
RaGAN - The relativistic discriminator: a key element missing from standard GAN

RAN - RAN4IQA: Restorative Adversarial Nets for No-Reference Image Quality Assessment (github)

RankGAN - Adversarial Ranking for Language Generation
RCGAN - Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs

ReConNN - Reconstruction of Simulation-Based Physical Field with Limited Samples by Reconstruction Neural Network
Recycle-GAN - Recycle-GAN: Unsupervised Video Retargeting

RefineGAN - Compressed Sensing MRI Reconstruction with Cyclic Loss in Generative Adversarial Networks

ReGAN - ReGAN: RE[LAX|BAR|INFORCE] based Sequence Generation using GANs (github)
RegCGAN - Unpaired Multi-Domain Image Generation via Regularized Conditional GANs

RenderGAN - RenderGAN: Generating Realistic Labeled Data
Resembled GAN - Resembled Generative Adversarial Networks: Two Domains with Similar Attributes

ResGAN - Generative Adversarial Network based on Resnet for Conditional Image Restoration

RNN-WGAN - Language Generation with Recurrent Generative Adversarial Networks without Pre-training (github)
RoCGAN - Robust Conditional Generative Adversarial Networks

RPGAN - Stabilizing GAN Training with Multiple Random Projections (github)
RTT-GAN - Recurrent Topic-Transition GAN for Visual Paragraph Generation

RWGAN - Relaxed Wasserstein with Applications to GANs

SAD-GAN - SAD-GAN: Synthetic Autonomous Driving using Generative Adversarial Networks
SAGA - Generative Adversarial Learning for Spectrum Sensing

SAGAN - Self-Attention Generative Adversarial Networks

SalGAN - SalGAN: Visual Saliency Prediction with Generative Adversarial Networks (github)
SAM - Sample-Efficient Imitation Learning via Generative Adversarial Nets

sAOG - Deep Structured Generative Models

SAR-GAN - Generating High Quality Visible Images from SAR Images Using CNNs
SBADA-GAN - From source to target and back: symmetric bi-directional adaptive GAN

ScarGAN - ScarGAN: Chained Generative Adversarial Networks to Simulate Pathological Tissue on Cardiovascular MR Scans
SCH-GAN - SCH-GAN: Semi-supervised Cross-modal Hashing by Generative Adversarial Network

SD-GAN - Semantically Decomposing the Latent Spaces of Generative Adversarial Networks

Sdf-GAN - Sdf-GAN: Semi-supervised Depth Fusion with Multi-scale Adversarial Networks
SEGAN - SEGAN: Speech Enhancement Generative Adversarial Network

SeGAN - SeGAN: Segmenting and Generating the Invisible
SegAN - SegAN: Adversarial Network with Multi-scale L1 Loss for Medical Image Segmentation

Sem-GAN - Sem-GAN: Semantically-Consistent Image-to-Image Translation

SeqGAN - SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient (github)
SeUDA - Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation in Chest X-ray Segmentation

SG-GAN - Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption (github)
SG-GAN - Sparsely Grouped Multi-task Generative Adversarial Networks for Facial Attribute Manipulation

SGAN - Texture Synthesis with Spatial Generative Adversarial Networks

SGAN - Stacked Generative Adversarial Networks (github)
SGAN - Steganographic Generative Adversarial Networks

SGAN - SGAN: An Alternative Training of Generative Adversarial Networks
SGAN - CT Image Enhancement Using Stacked Generative Adversarial Networks and Transfer Learning for Lesion Segmentation Improvement

sGAN - Generative Adversarial Training for MRA Image Synthesis Using Multi-Contrast MRI

SiftingGAN - SiftingGAN: Generating and Sifting Labeled Samples to Improve the Remote Sensing Image Scene Classification Baseline in vitro
SiGAN - SiGAN: Siamese Generative Adversarial Network for Identity-Preserving Face Hallucination

SimGAN - Learning from Simulated and Unsupervised Images through Adversarial Training
SisGAN - Semantic Image Synthesis via Adversarial Learning

Sketcher-Refiner GAN - Learning Myelin Content in Multiple Sclerosis from Multimodal MRI through Adversarial Training

SketchGAN - Adversarial Training For Sketch Retrieval
SketchyGAN - SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis

Skip-Thought GAN - Generating Text through Adversarial Training using Skip-Thought Vectors
SL-GAN - Semi-Latent GAN: Learning to generate and modify facial images from attributes

SLSR - Sparse Label Smoothing for Semi-supervised Person Re-Identification

SN-DCGAN - Generative Adversarial Networks for Unsupervised Object Co-localization
SN-GAN - Spectral Normalization for Generative Adversarial Networks (github)

SN-PatchGAN - Free-Form Image Inpainting with Gated Convolution
Sobolev GAN - Sobolev GAN

Social GAN - Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks

Softmax GAN - Softmax GAN
SoPhie - SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints

speech-driven animation GAN - End-to-End Speech-Driven Facial Animation with Temporal GANs
Spike-GAN - Synthesizing realistic neural population activity patterns using Generative Adversarial Networks

Splitting GAN - Class-Splitting Generative Adversarial Networks

SR-CNN-VAE-GAN - Semi-Recurrent CNN-based VAE-GAN for Sequential Data Generation (github)
SRGAN - Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

SRPGAN - SRPGAN: Perceptual Generative Adversarial Network for Single Image Super Resolution
SS-GAN - Semi-supervised Conditional GANs

ss-InfoGAN - Guiding InfoGAN with Semi-Supervision

SSGAN - SSGAN: Secure Steganography Based on Generative Adversarial Networks
SSL-GAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks

ST-CGAN - Stacked Conditional Generative Adversarial Networks for Jointly Learning Shadow Detection and Shadow Removal
ST-GAN - Style Transfer Generative Adversarial Networks: Learning to Play Chess Differently

ST-GAN - ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing

StackGAN - StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks (github)
StainGAN - StainGAN: Stain Style Transfer for Digital Histological Images

StarGAN - StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation (github)
StarGAN-VC - StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks

SteinGAN - Learning Deep Energy Models: Contrastive Divergence vs. Amortized MLE

StepGAN - Improving Conditional Sequence Generative Adversarial Networks by Stepwise Evaluation
Super-FAN - Super-FAN: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with GANs

SVSGAN - SVSGAN: Singing Voice Separation via Generative Adversarial Network
SWGAN - Solving Approximate Wasserstein GANs to Stationarity

SyncGAN - SyncGAN: Synchronize the Latent Space of Cross-modal Generative Adversarial Networks

S^2GAN - Generative Image Modeling using Style and Structure Adversarial Networks
T2Net - T2Net: Synthetic-to-Realistic Translation for Solving Single-Image Depth Estimation Tasks

table-GAN - Data Synthesis based on Generative Adversarial Networks
TAC-GAN - TAC-GAN - Text Conditioned Auxiliary Classifier Generative Adversarial Network (github)

TAN - Outline Colorization through Tandem Adversarial Networks

tcGAN - Cross-modal Hallucination for Few-shot Fine-grained Recognition
TD-GAN - Task Driven Generative Modeling for Unsupervised Domain Adaptation: Application to X-ray Image Segmentation

tempCycleGAN - Improving Surgical Training Phantoms by Hyperrealism: Deep Unpaired Image-to-Image Translation from Real Surgeries
tempoGAN - tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow

TequilaGAN - TequilaGAN: How to easily identify GAN samples

Text2Shape - Text2Shape: Generating Shapes from Natural Language by Learning Joint Embeddings
textGAN - Generating Text via Adversarial Training

TextureGAN - TextureGAN: Controlling Deep Image Synthesis with Texture Patches
TGAN - Temporal Generative Adversarial Nets

TGAN - Tensorizing Generative Adversarial Nets

TGAN - Tensor-Generative Adversarial Network with Two-dimensional Sparse Coding: Application to Real-time Indoor Localization
TGANs-C - To Create What You Tell: Generating Videos from Captions

tiny-GAN - Analysis of Nonautonomous Adversarial Systems
TP-GAN - Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis

TreeGAN - TreeGAN: Syntax-Aware Sequence Generation with Generative Adversarial Networks

Triple-GAN - Triple Generative Adversarial Nets
tripletGAN - TripletGAN: Training Generative Model with Triplet Loss

TV-GAN - TV-GAN: Generative Adversarial Network Based Thermal to Visible Face Recognition
Twin-GAN - Twin-GAN -- Unpaired Cross-Domain Image Translation with Weight-Sharing GANs

UGACH - Unsupervised Generative Adversarial Cross-modal Hashing

UGAN - Enhancing Underwater Imagery using Generative Adversarial Networks
Unim2im - Unsupervised Image-to-Image Translation with Generative Adversarial Networks (github)

UNIT - Unsupervised Image-to-image Translation Networks (github)
Unrolled GAN - Unrolled Generative Adversarial Networks (github)

UT-SCA-GAN - Spatial Image Steganography Based on Generative Adversarial Network

UV-GAN - UV-GAN: Adversarial Facial UV Map Completion for Pose-invariant Face Recognition
VA-GAN - Visual Feature Attribution using Wasserstein GANs

VAC+GAN - Versatile Auxiliary Classifier with Generative Adversarial Network (VAC+GAN), Multi Class Scenarios
VAE-GAN - Autoencoding beyond pixels using a learned similarity metric

VariGAN - Multi-View Image Generation from a Single-View

VAW-GAN - Voice Conversion from Unaligned Corpora using Variational Autoencoding Wasserstein Generative Adversarial Networks
VEEGAN - VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning (github)

VGAN - Generating Videos with Scene Dynamics (github)
VGAN - Generative Adversarial Networks as Variational Training of Energy Based Models (github)

VGAN - Text Generation Based on Generative Adversarial Nets with Latent Variable

ViGAN - Image Generation and Editing with Variational Info Generative Adversarial Networks
VIGAN - VIGAN: Missing View Imputation with Generative Adversarial Networks

VoiceGAN - Voice Impersonation using Generative Adversarial Networks
VOS-GAN - VOS-GAN: Adversarial Learning of Visual-Temporal Dynamics for Unsupervised Dense Prediction in Videos

VRAL - Variance Regularizing Adversarial Learning

WaterGAN - WaterGAN: Unsupervised Generative Network to Enable Real-time Color Correction of Monocular Underwater Images
WaveGAN - Synthesizing Audio with Generative Adversarial Networks

WaveletGLCA-GAN - Global and Local Consistent Wavelet-domain Age Synthesis
weGAN - Generative Adversarial Nets for Multiple Text Corpora

WGAN - Wasserstein GAN (github)

WGAN-CLS - Text to Image Synthesis Using Generative Adversarial Networks
WGAN-GP - Improved Training of Wasserstein GANs (github)

WGAN-L1 - Subsampled Turbulence Removal Network
WS-GAN - Weakly Supervised Generative Adversarial Networks for 3D Reconstruction

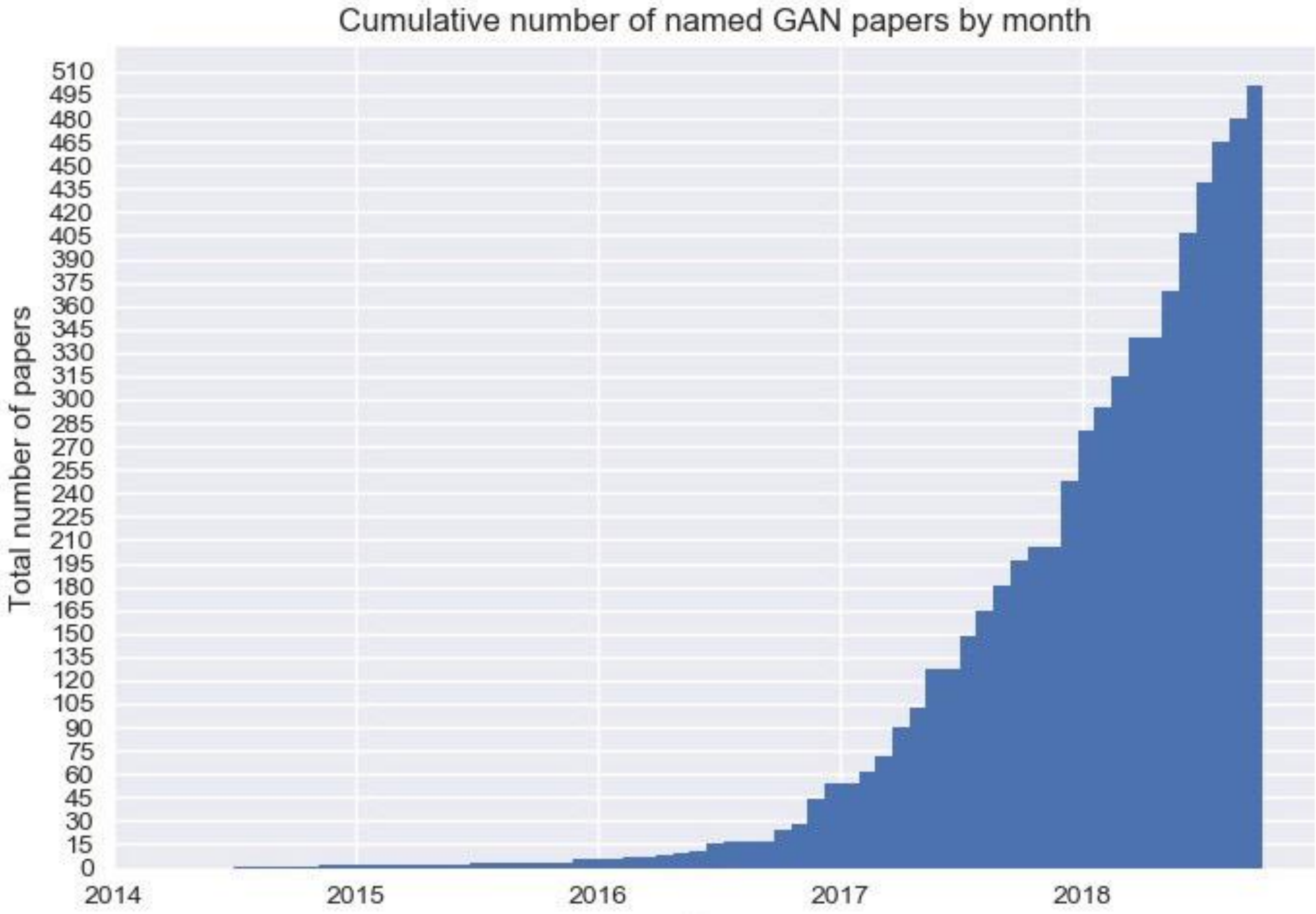
X-GANs - X-GANs: Image Reconstruction Made Easy for Extreme Cases

XGAN - XGAN: Unsupervised Image-to-Image Translation for many-to-many Mappings
ZipNet-GAN - ZipNet-GAN: Inferring Fine-grained Mobile Traffic Patterns via a Generative Adversarial Neural Network

α-GAN - Variational Approaches for Auto-Encoding Generative Adversarial Networks (github)
β-GAN - Annealed Generative Adversarial Networks

Δ-GAN - Triangle Generative Adversarial Networks

An explo-GAN of papers



Explosive growth — All the named GAN variants cumulatively since 2014.

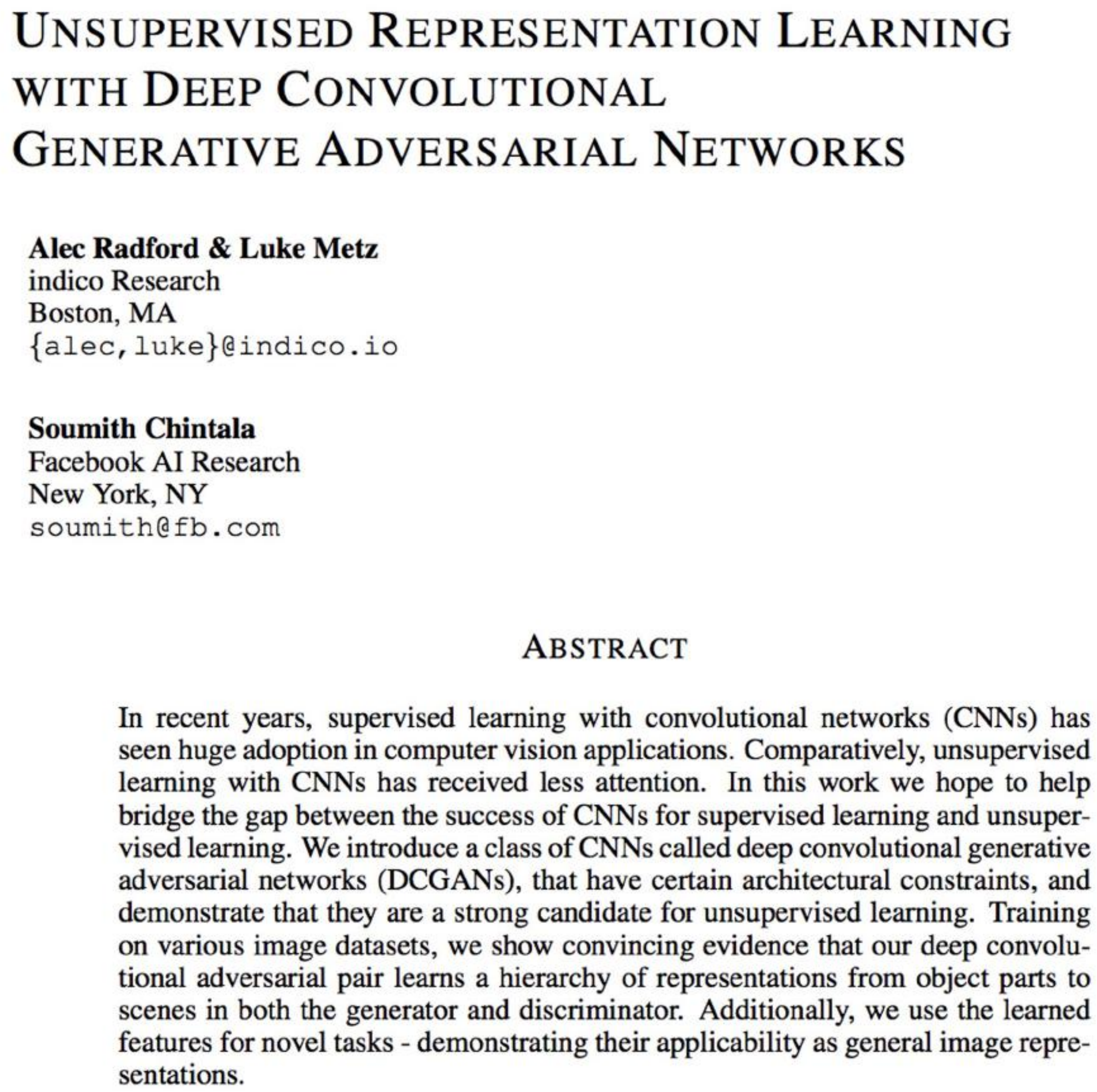
Credit: Bruno Gavranović

Deep Hunt, blog by Avinash Hindupur

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - **DC GAN (Radford et al, 2016)**
  - Improved Training of GANs (Salimans et al'16), Projected GAN (Sauer et al'21)
  - WGAN, WGAN-GP, Progressive GAN, SN-GAN, SAGAN
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

Deep Convolutional GAN (DCGAN)



Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

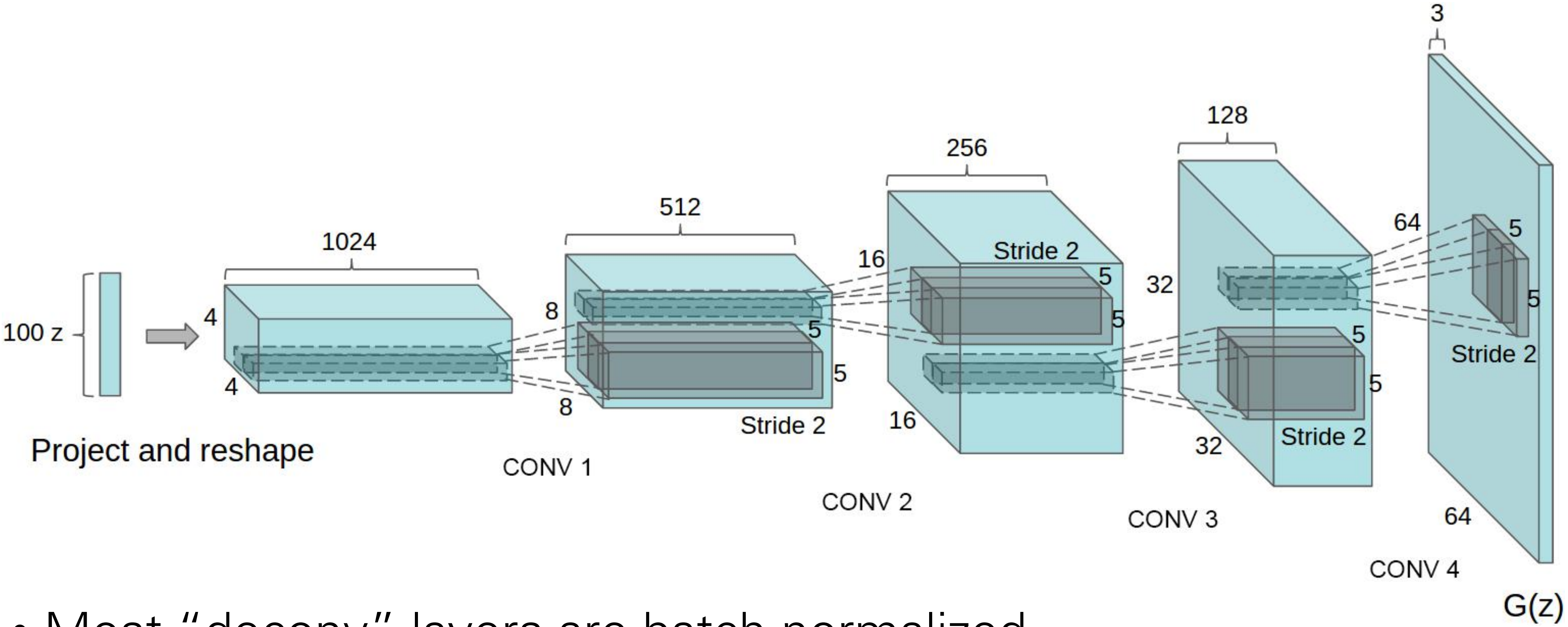
Alec Radford & Luke Metz
indico Research
Boston, MA
{alec,luke}@indico.io

Soumith Chintala
Facebook AI Research
New York, NY
soumith@fb.com

Abstract

In recent years, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. In this work we hope to help bridge the gap between the success of CNNs for supervised learning and unsupervised learning. We introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning. Training on various image datasets, we show convincing evidence that our deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator. Additionally, we use the learned features for novel tasks - demonstrating their applicability as general image representations.

Deep Convolutional GAN (DCGAN)



- Most “deconv” layers are batch normalized

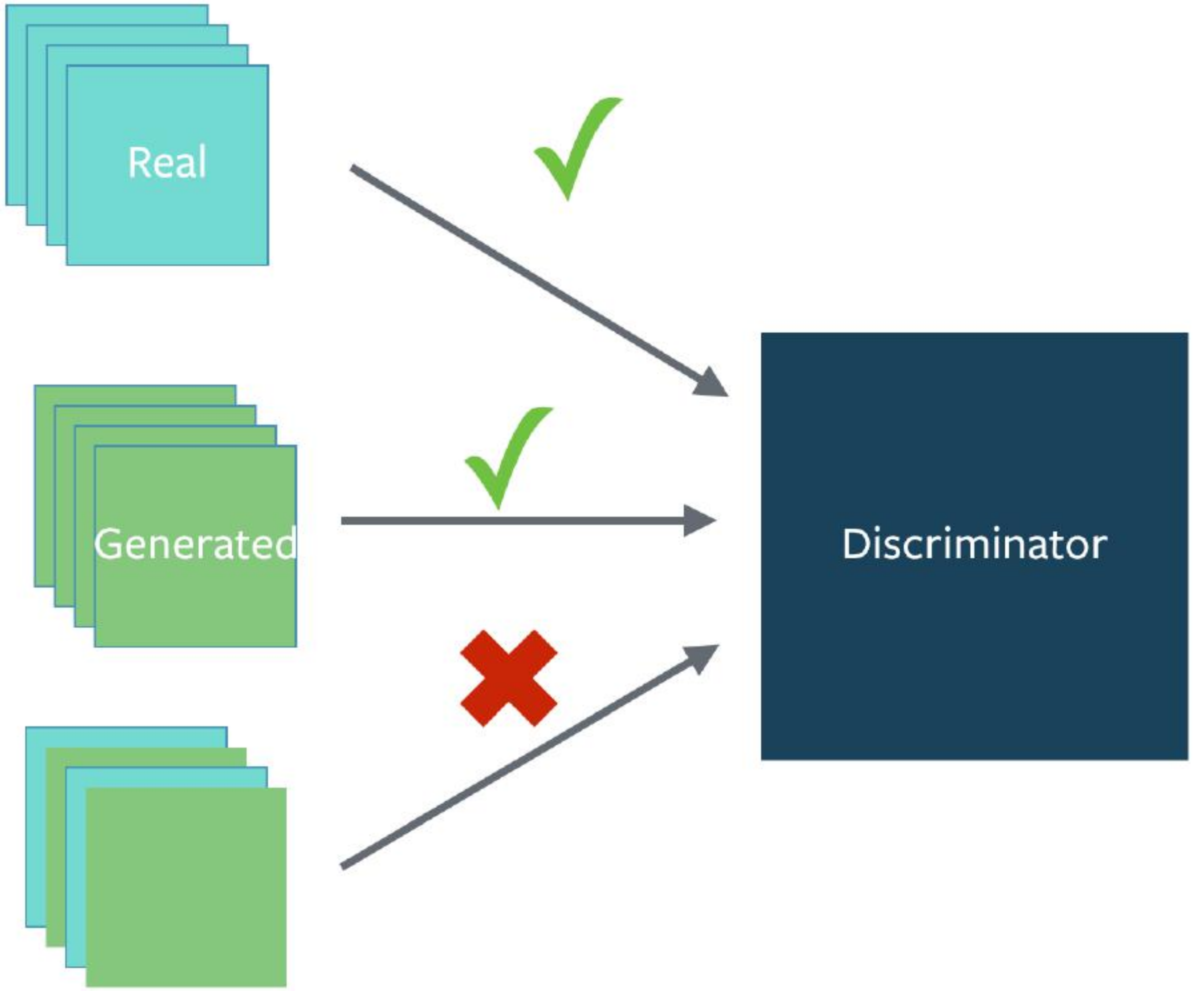
[Radford et al 2016]

DCGAN - Architecture Design

- Supervised Learning CNNs not directly usable
  - Remove max-pooling and mean-pooling
  - Upsample using transposed convolutions in the generator
  - Downsample with strided convolutions and average pooling
  - Non-Linearity: ReLU for generator, Leaky-ReLU (0.2) for discriminator
  - Output Non-Linearity: tanh for Generator, sigmoid for discriminator
  - Batch Normalization used to prevent mode collapse
  - Batch Normalization is not applied at the output of G and input of D
- Optimization details
  - Adam: small LR - 2e-4; small momentum: 0.5, batch-size: 128

[Radford et al 2016]

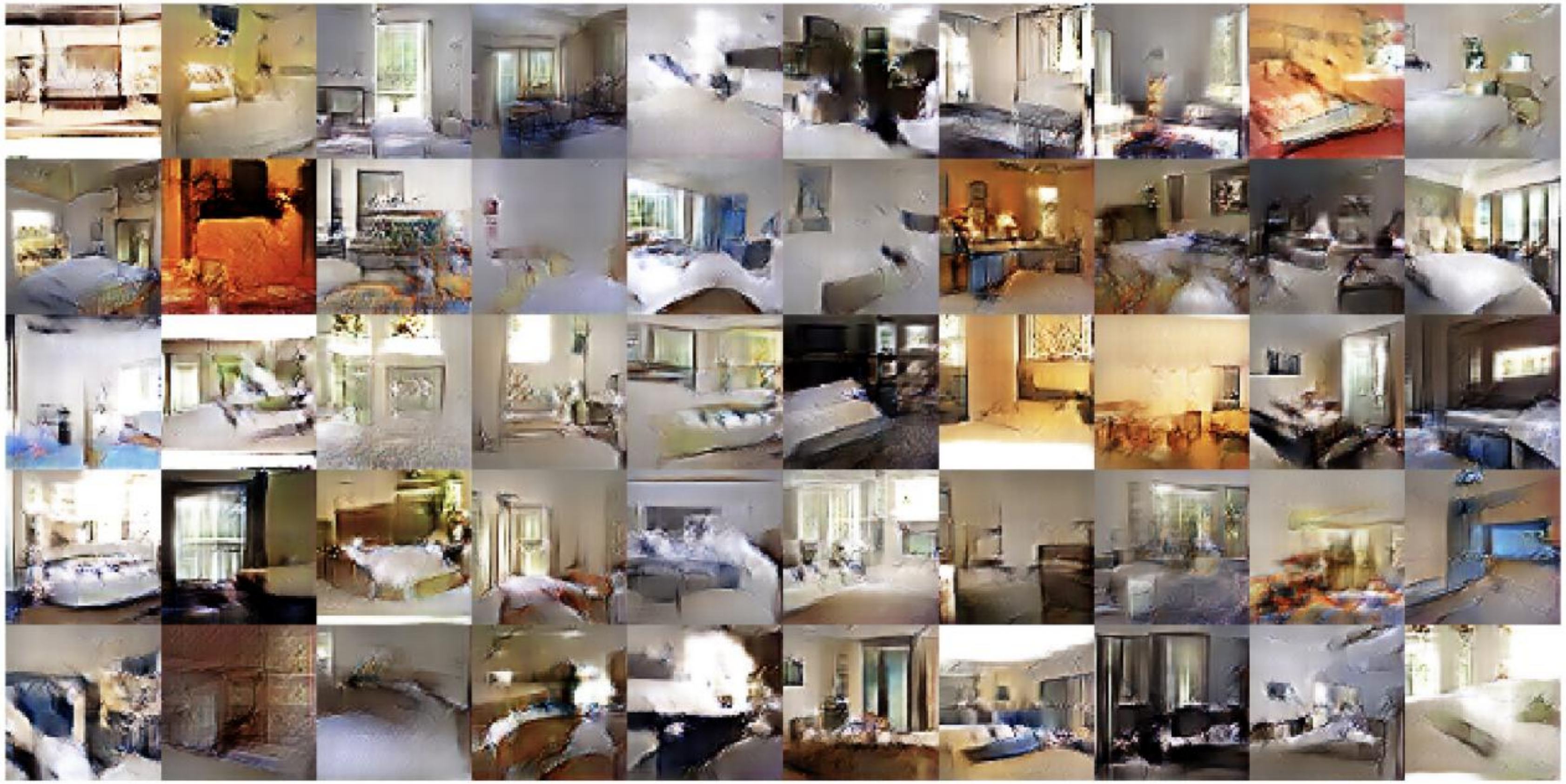
DCGAN Batch Norm



[Chintala 2016]

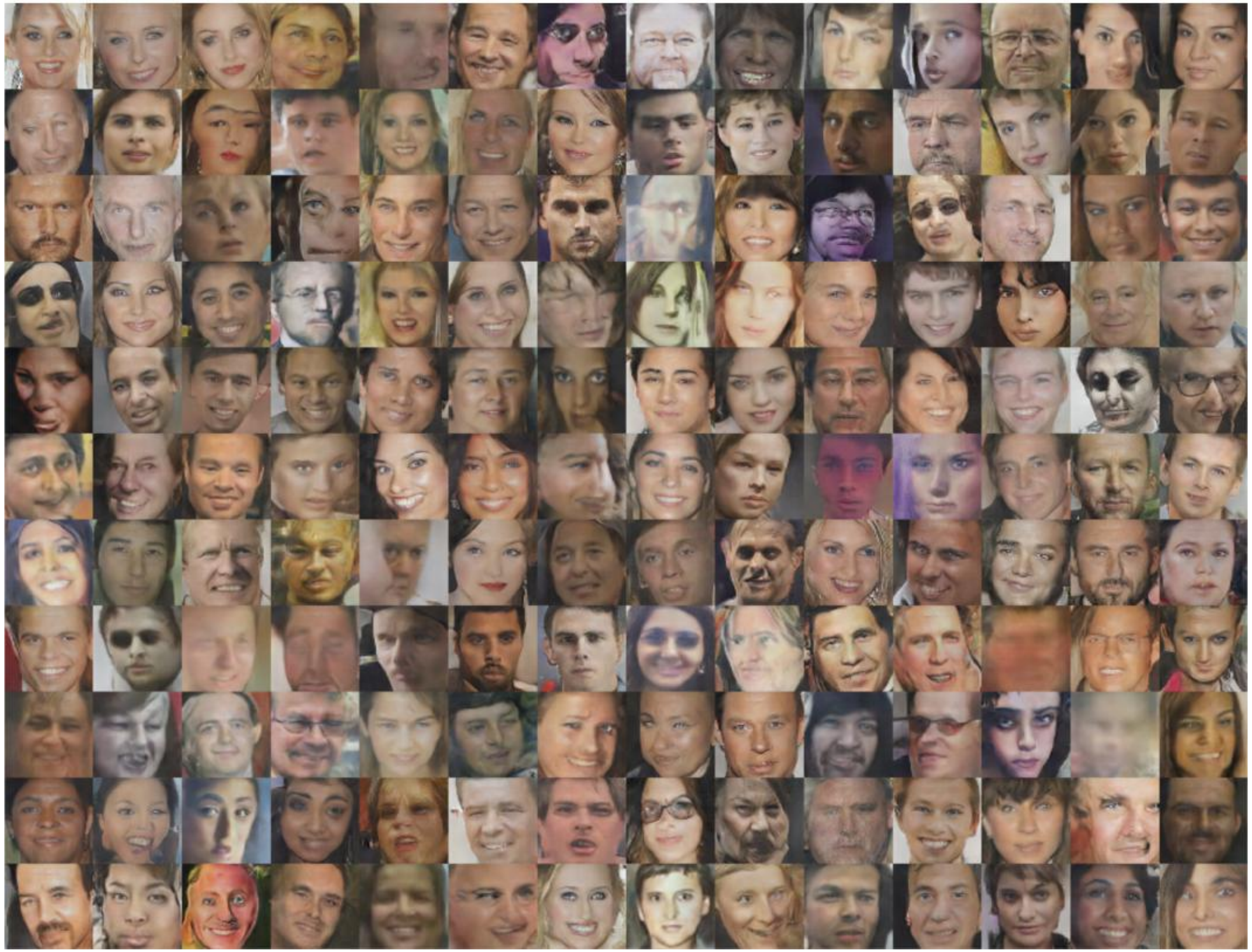
DCGAN - Key Results

- Good samples on datasets with 3M images (Faces, Bedrooms) for the first time



[Radford et al 2016]

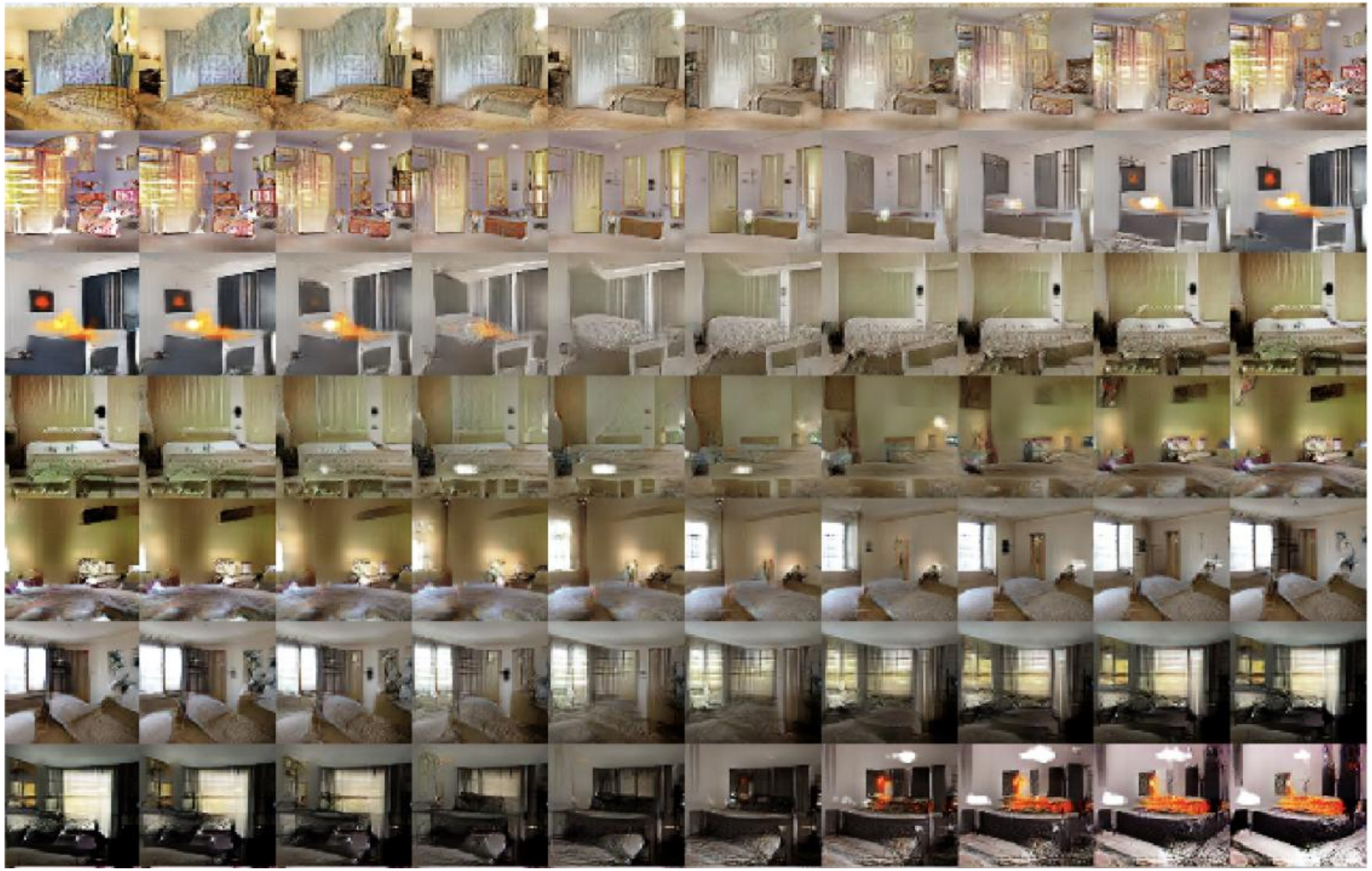
DCGAN - Key Results



[Radford et al 2016]

DCGAN - Key Results

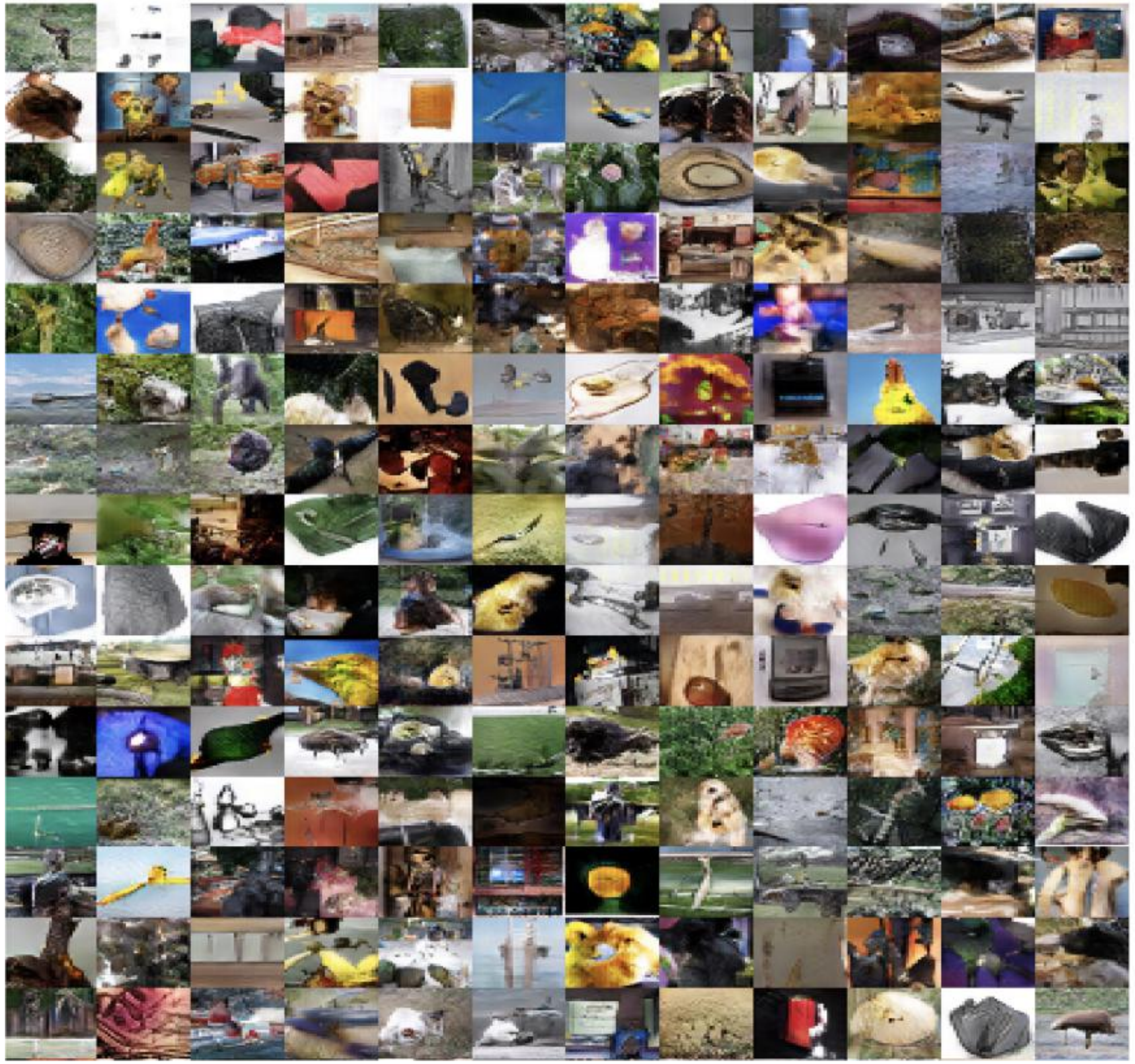
- Smooth interpolations in high dimensions



[Radford et al 2016]

DCGAN - Key Results

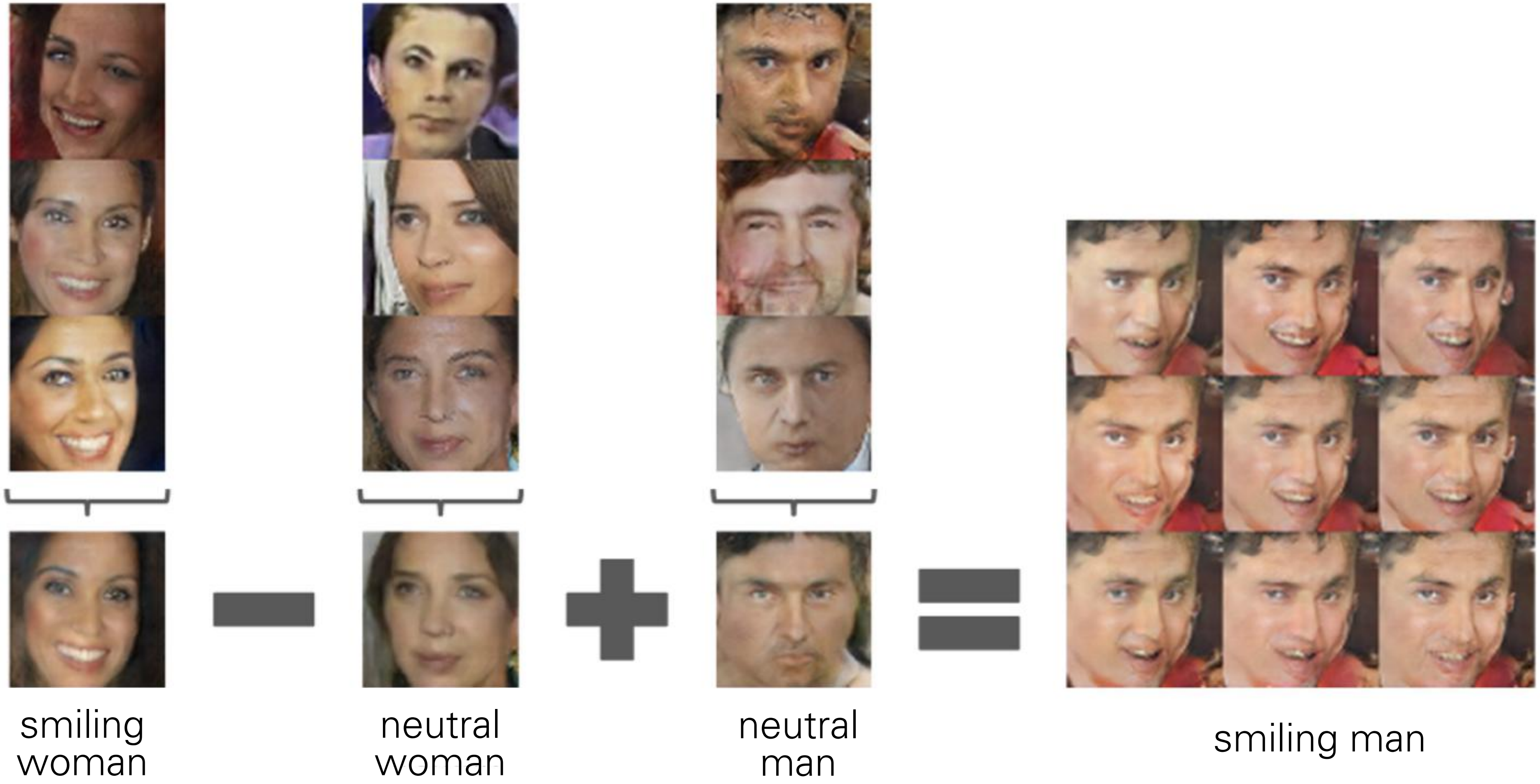
- Imagenet samples



[Radford et al 2016]

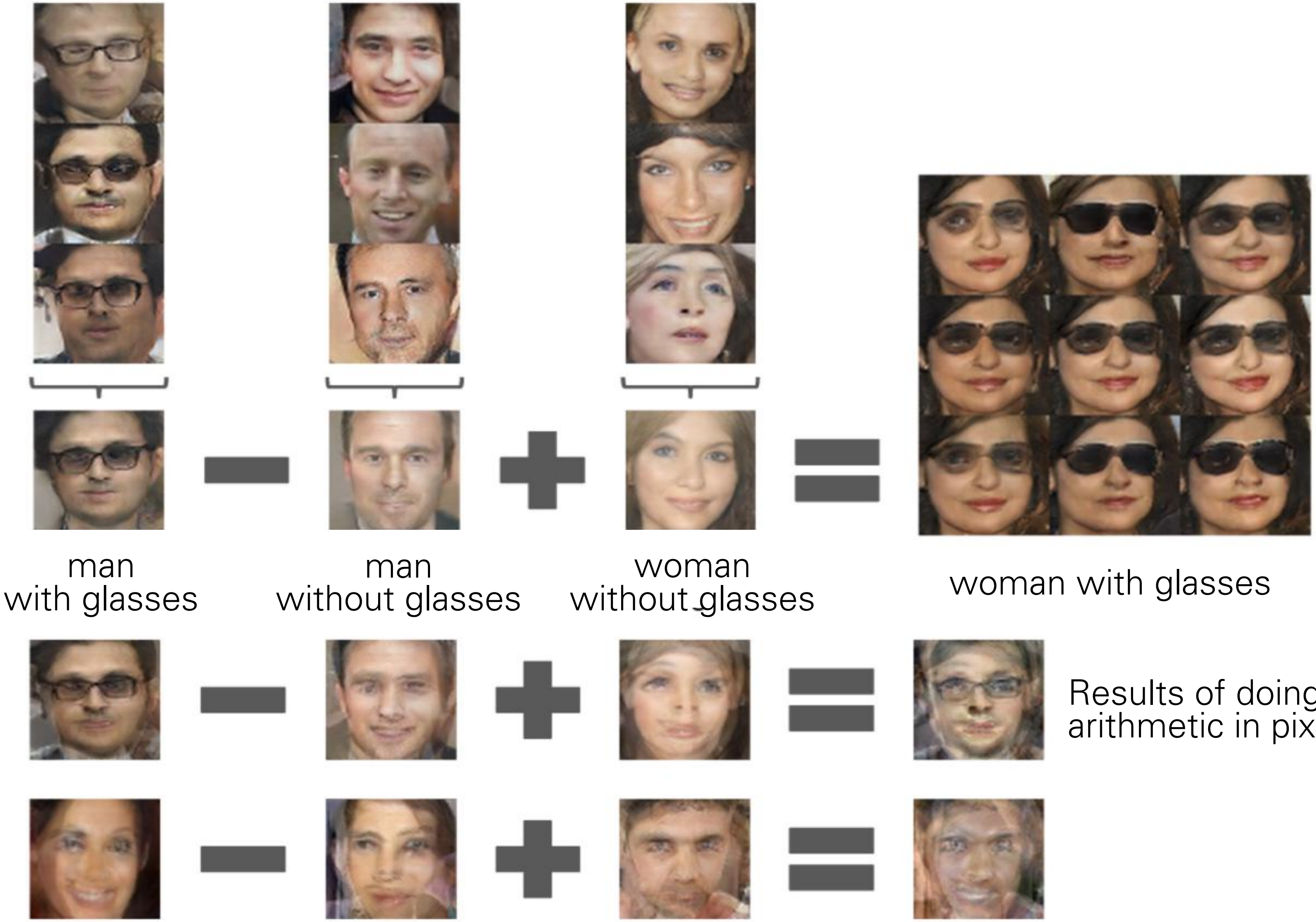
DCGAN - Key Results

- Vector Arithmetic



[Radford et al 2016]

DCGAN - Key Results



man with glasses

man without glasses

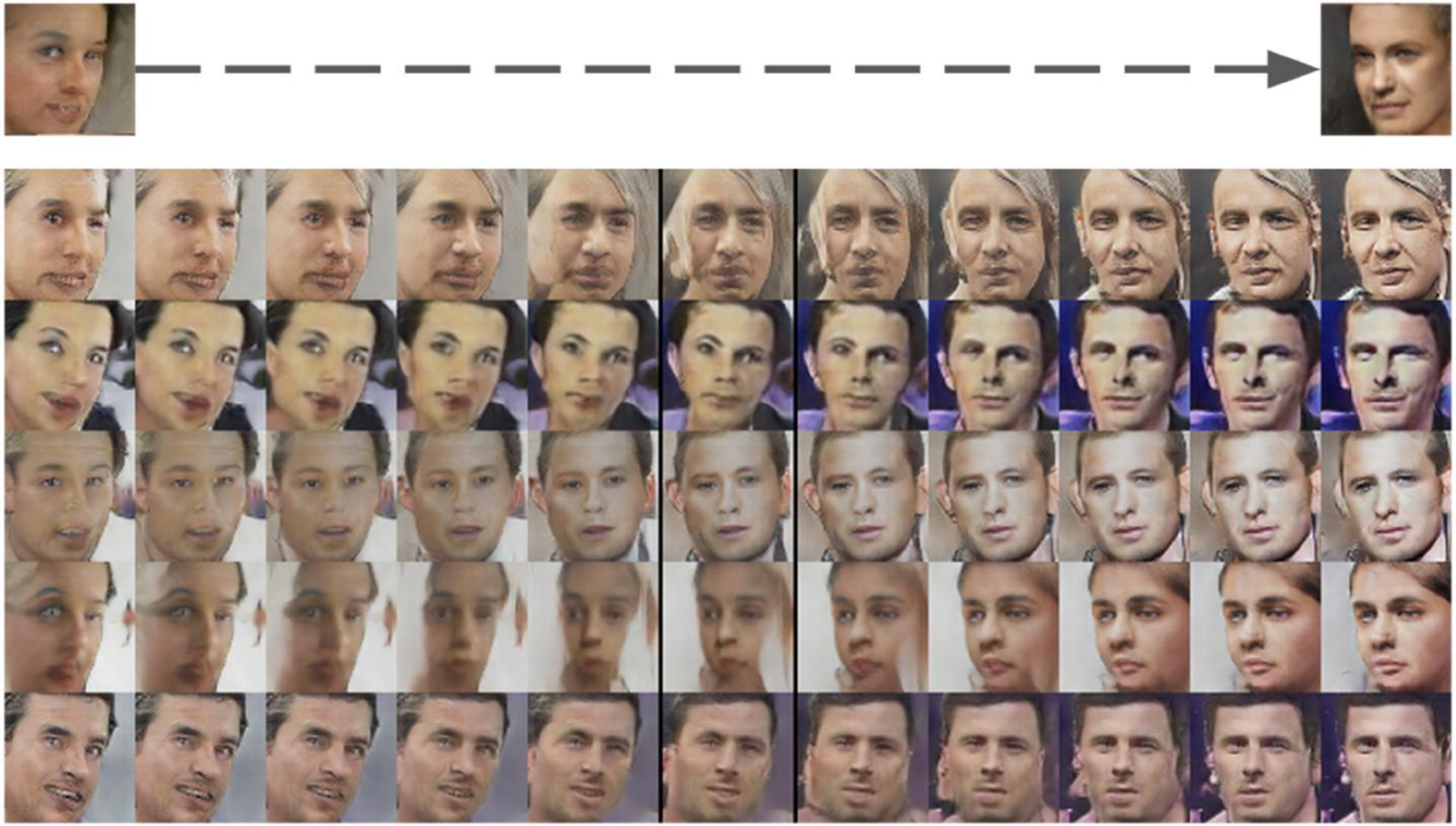
woman without glasses

woman with glasses

Results of doing the same arithmetic in pixel space

[Radford et al 2016]

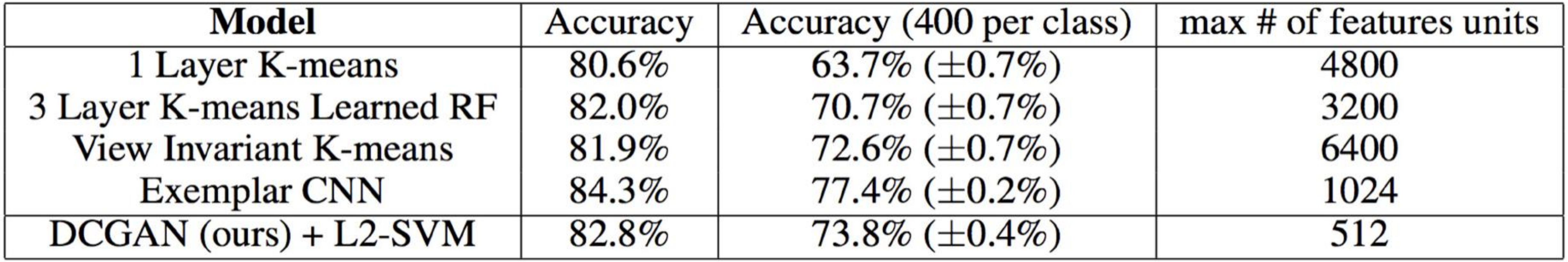
DCGAN - Key Results



[Radford et al 2016]

DCGAN - Key Results

- Representation Learning



Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% (±0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	84.3%	77.4% (±0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% (±0.4%)	512

[Radford et al 2016]

DCGAN - Conclusions

- Incredible samples for any generative model
- GANs could be made to work well with architecture details
- Perceptually good samples and interpolations
- Representation Learning
- **Problems to address:**
  - Unstable training
  - Brittle architecture / hyperparameters

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - **Improved Training of GANs (Salimans et al'16)**, Projected GAN (Sauer et al'21)
  - WGAN, WGAN-GP, Progressive GAN, SN-GAN, SAGAN
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

Improved training of GANs

- Feature Matching
- Minibatch discrimination
- Historical Averaging
- Virtual batch normalization
- One-sided label smoothing

Improved Techniques for Training GANs

Tim Salimans
tim@openai.com

Ian Goodfellow
ian@openai.com

Wojciech Zaremba
woj@openai.com

Vicki Cheung
vicki@openai.com

Alec Radford
alec.radford@gmail.com

Xi Chen
peter@openai.com

[Salimans 2016]

Improved training of GANs

- Feature Matching

$$||\mathbb{E}_{x\sim p_{\text{data}}} f(x) - \mathbb{E}_{z\sim p(z)} f(G(z))||^2$$

Generator objective

[Salimans 2016]

Improved training of GANs

- Minibatch discrimination

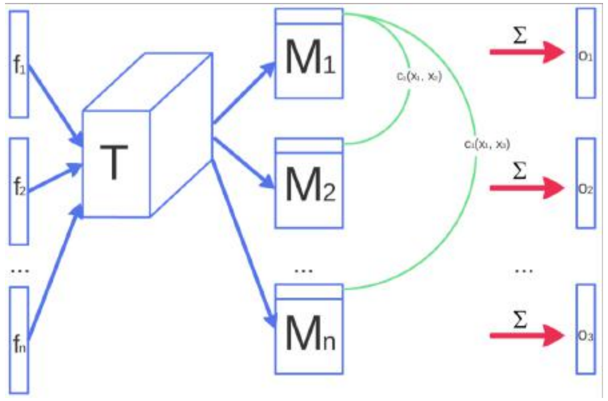
$$\mathbf{f}(\boldsymbol{x}_i) \in \mathbb{R}^A \qquad T \in \mathbb{R}^{A\times B\times C} \qquad M_i \in \mathbb{R}^{B\times C}$$

$$c_b(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||M_{i,b} - M_{j,b}||_{L_1}) \in \mathbb{R}$$

$$o(\boldsymbol{x}_i)_b = \sum_{j=1}^{n} c_b(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{R}$$

$$o(\boldsymbol{x}_i) = \Big[o(\boldsymbol{x}_i)_1, o(\boldsymbol{x}_i)_2, \ldots, o(\boldsymbol{x}_i)_B\Big] \in \mathbb{R}^B$$

$$o(\mathbf{X}) \in \mathbb{R}^{n\times B}$$



[Salimans 2016]

Allows to incorporate side information from other samples and is superior to feature matching in the unconditional setting. Helps addressing mode collapse by allowing discriminator to detect if the generated samples are too close to each other.

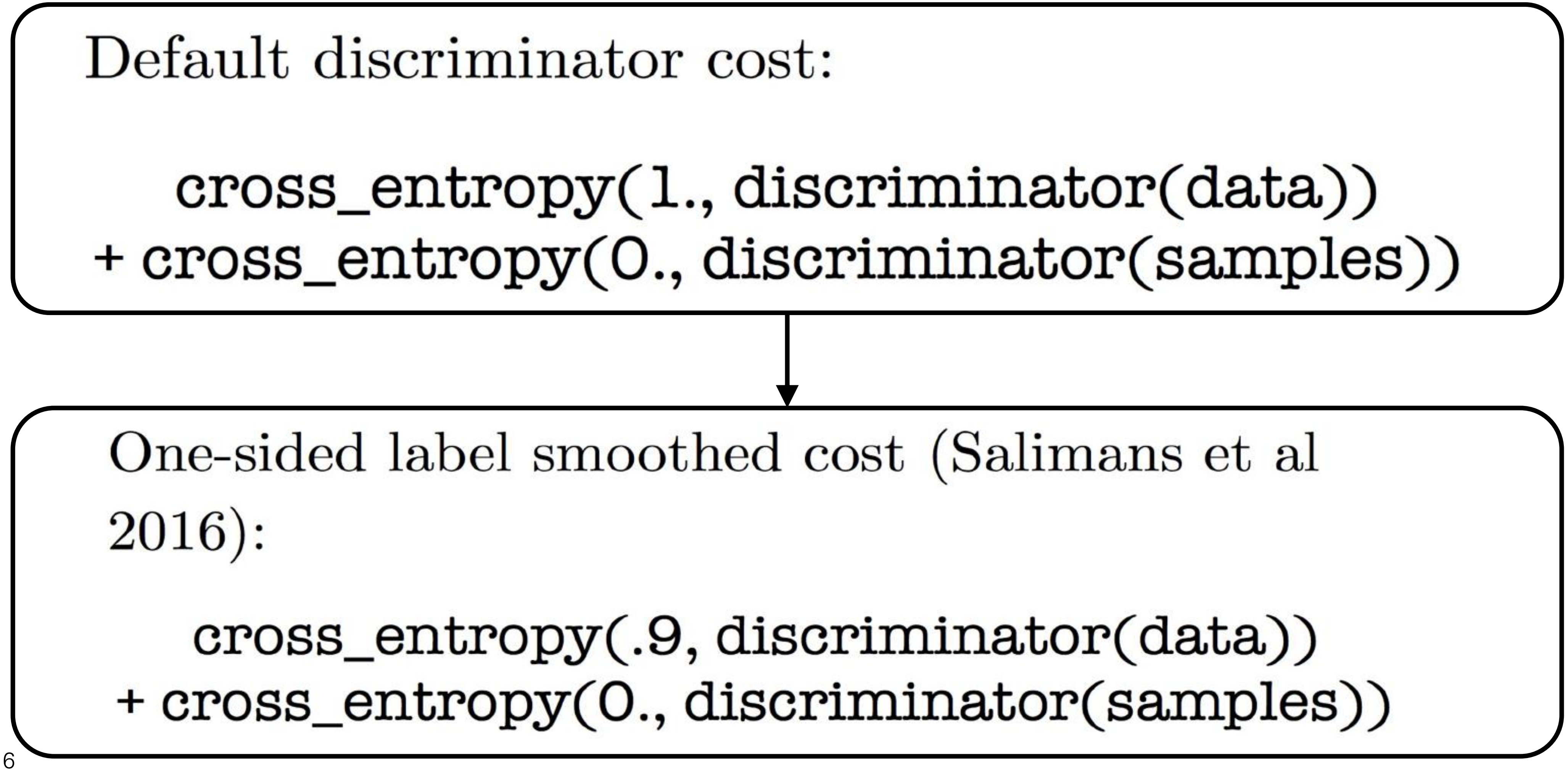
Improved training of GANs

- Historical Averaging

$$||\boldsymbol{\theta} - \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{\theta}[i]||^2$$

Improved training of GANs

- One-sided label smoothing



Default discriminator cost:

```
cross_entropy(1., discriminator(data))
+ cross_entropy(0., discriminator(samples))
```

One-sided label smoothed cost (Salimans et al 2016):

```
cross_entropy(.9, discriminator(data))
+ cross_entropy(0., discriminator(samples))
```

Figure source:
NeurIPS tutorial
Goodfellow 2016

Improved training of GANs

- Why one-sided?

Reinforces current generator behavior

$$D(\boldsymbol{x}) = \frac{(1-\alpha)p_{\text{data}}(\boldsymbol{x}) + \beta p_{\text{model}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_{\text{model}}(\boldsymbol{x})}$$

Figure source:
NeurIPS tutorial
Goodfellow 2016

Improved training of GANs

- Virtual Batch Normalization
  - Use a reference batch (fixed) to compute normalization statistics
  - Construct a batch containing the sample and reference batch

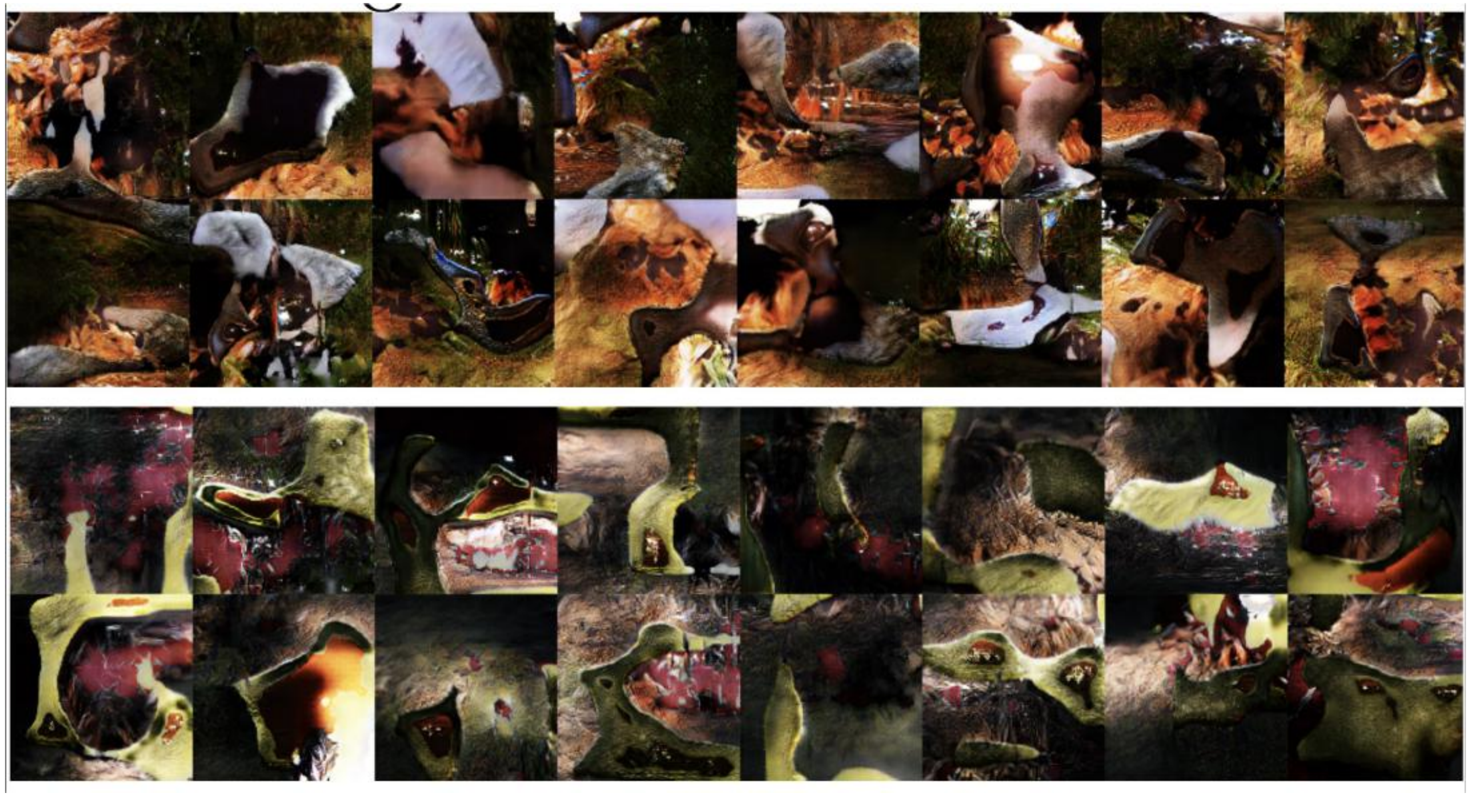


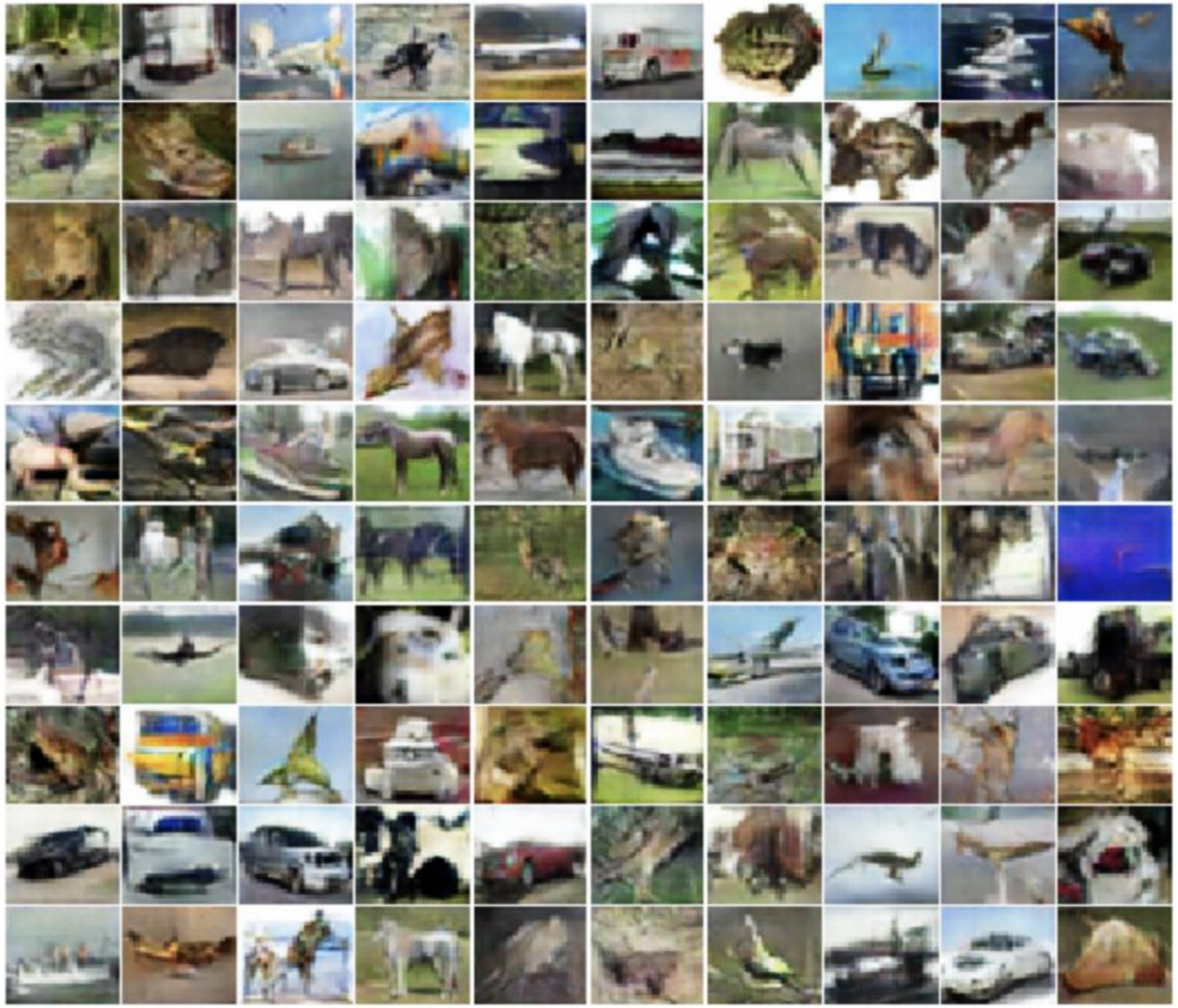
Figure source: NeurIPS tutorial Goodfellow 2016

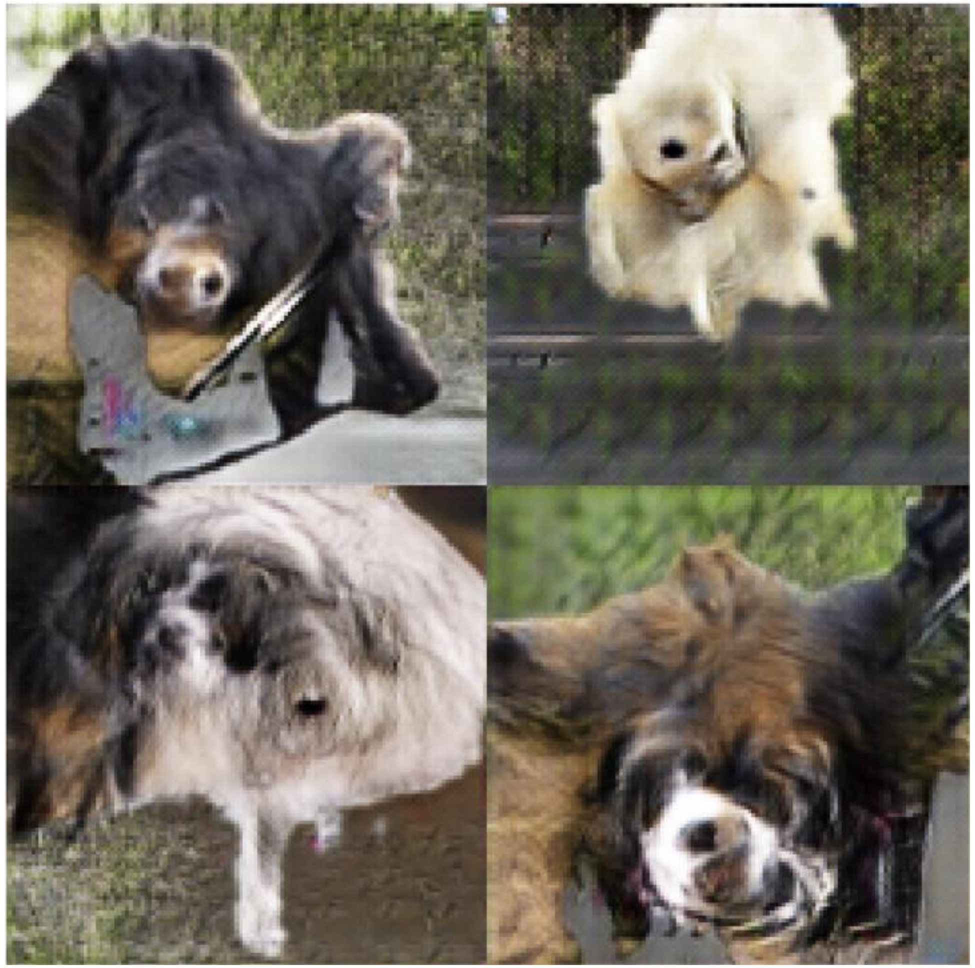
Improved training of GANs

- Semi-Supervised Learning
  - Predict labels in addition to fake/real in the discriminator
  - Approximate way of modeling p(x,y)
  - Generator doesn’t have to be made conditional p(x|y)
  - Use a deeper architecture for the discriminator compared to generator

$$
\begin{aligned}
L &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\text{model}}(y|\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{\text{model}}(y=K+1|\boldsymbol{x})] \\
&= L_{\text{supervised}} + L_{\text{unsupervised}}, \text{ where} \\
L_{\text{supervised}} &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)} \log p_{\text{model}}(y|\boldsymbol{x}, y<K+1) \\
L_{\text{unsupervised}} &= -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log[1 - p_{\text{model}}(y=K+1|\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim G} \log[p_{\text{model}}(y=K+1|\boldsymbol{x})]\}
\end{aligned}
$$

Improved training of GANs





Salimans 2016

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - Improved Training of GANs (Salimans et al'16), **Projected GAN (Sauer et al'21)**
  - WGAN, WGAN-GP, Progressive GAN, SN-GAN, SAGAN
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

Projected GAN

- Training GANs in pretrained feature spaces improves image quality, training speed, and sample efficiency.

$$\min_{G} \max_{\{D_l\}} \sum_{l\in\mathcal{L}} \left(\mathbb{E}_{\mathbf{x}}\left[\log D_l\left(P_l(\mathbf{x})\right)\right] + \mathbb{E}_{\mathbf{z}}\left[\log\left(1 - D_l\left(P_l(G(\mathbf{z}))\right)\right)\right]\right)$$

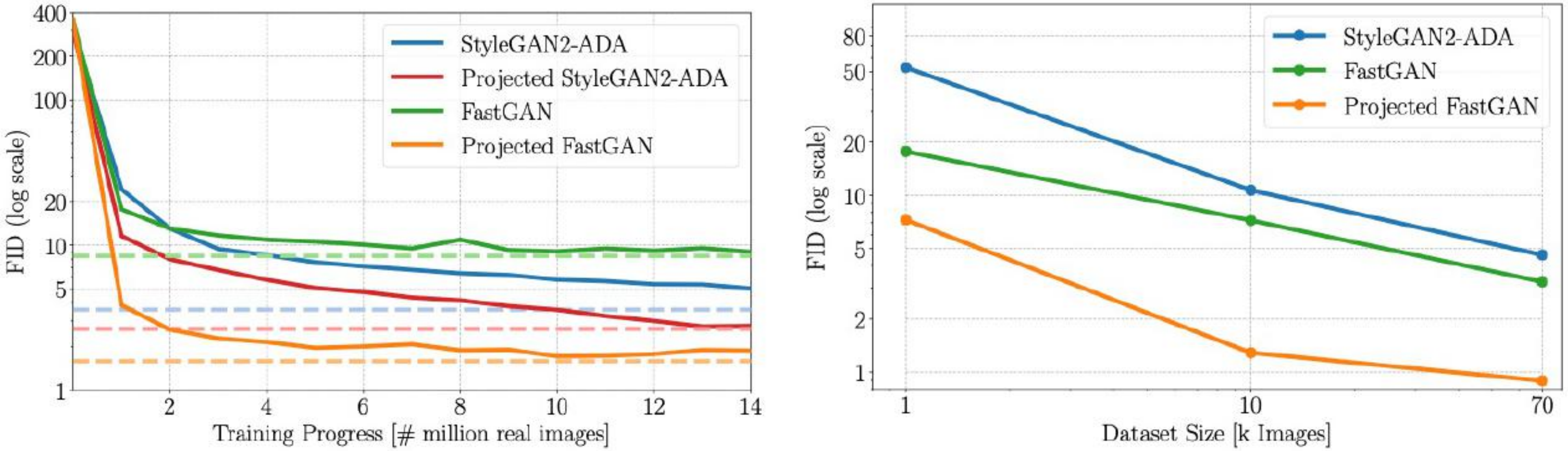


Figure 4: **Training Properties.** Left: Projected FastGAN surpasses the best FID of StyleGAN2 (at 88 M images) after just 1.1 M images on LSUN-Church. Right: Projected FastGAN yields significantly improved FID scores, even when using subsets of CLEVR with 1k and 10k samples.

Projected GAN

- Training GANs in pretrained feature spaces improves image quality, training speed, and sample efficiency.

$$\min_{G} \max_{\{D_l\}} \sum_{l \in \mathcal{L}} \left(\mathbb{E}_{\mathbf{x}} \left[\log D_l \left(P_l(\mathbf{x}) \right) \right] + \mathbb{E}_{\mathbf{z}} \left[\log \left(1 - D_l \left(P_l(G(\mathbf{z})) \right) \right) \right] \right)$$

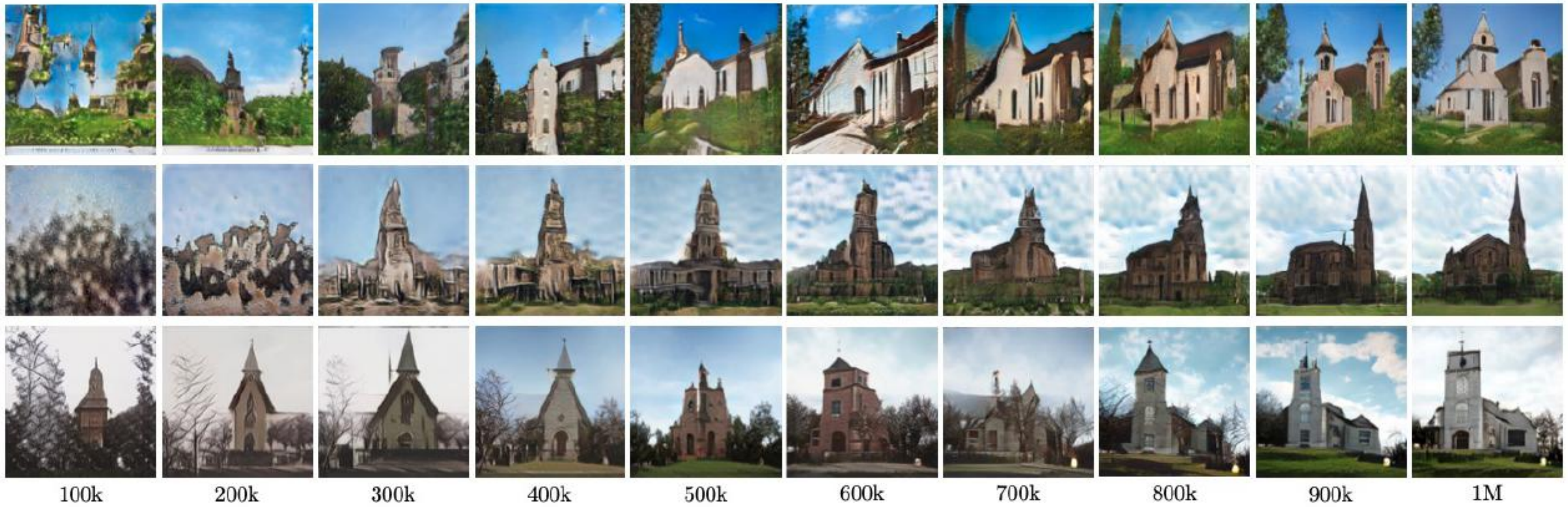
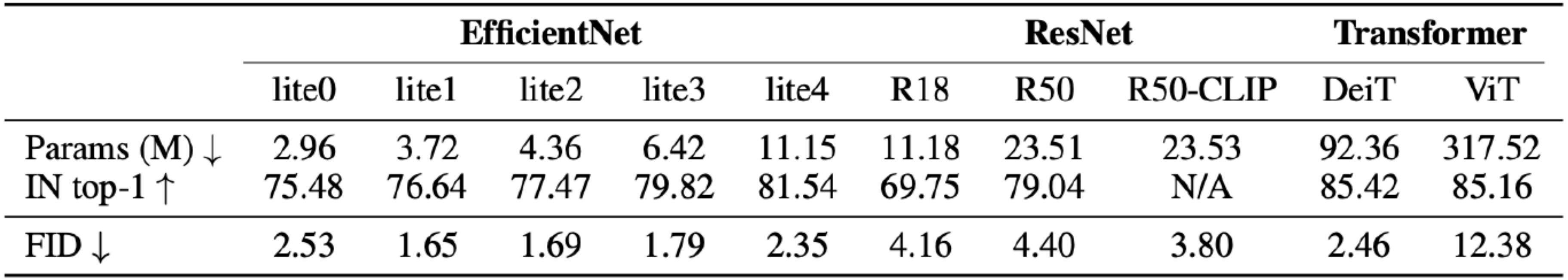


Figure 5: **Training progress on LSUN church at 256^2 pixels.** Shown are samples for a fixed noise vector $\mathbf{z}$ over k images. From top to bottom: FastGAN, StyleGAN2-ADA, Projected GAN.

Projected GAN

- The choice of pretrained network is important!



	EfficientNet					ResNet			Transformer	
	lite0	lite1	lite2	lite3	lite4	R18	R50	R50-CLIP	DeiT	ViT
Params (M) ↓	2.96	3.72	4.36	6.42	11.15	11.18	23.51	23.53	92.36	317.52
IN top-1 ↑	75.48	76.64	77.47	79.82	81.54	69.75	79.04	N/A	85.42	85.16
FID ↓	2.53	1.65	1.69	1.79	2.35	4.16	4.40	3.80	2.46	12.38

Table 2: **Pretrained Feature Networks Study**. We train the projected GAN with different pretrained feature networks. We find that compact EfficientNets outperform both ResNets and Transformers.

- More details:
  - Multi-Scale Discriminators
  - Random Projections
  - Cross-Channel Mixing (CCM)
  - Cross-Scale Mixing (CSM)

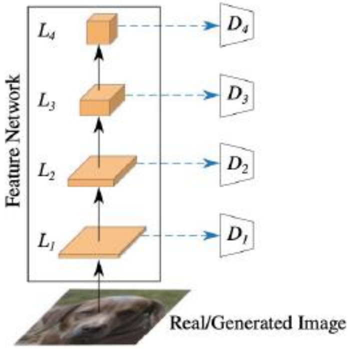


Figure 2: **CCM** (dashed blue arrows) employs 1×1 convolutions with random weights.

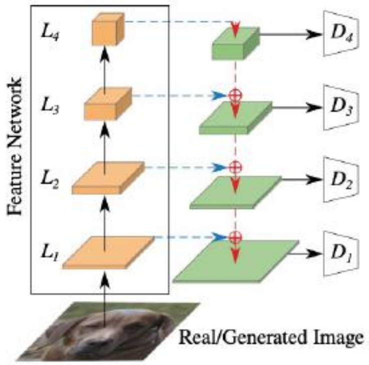
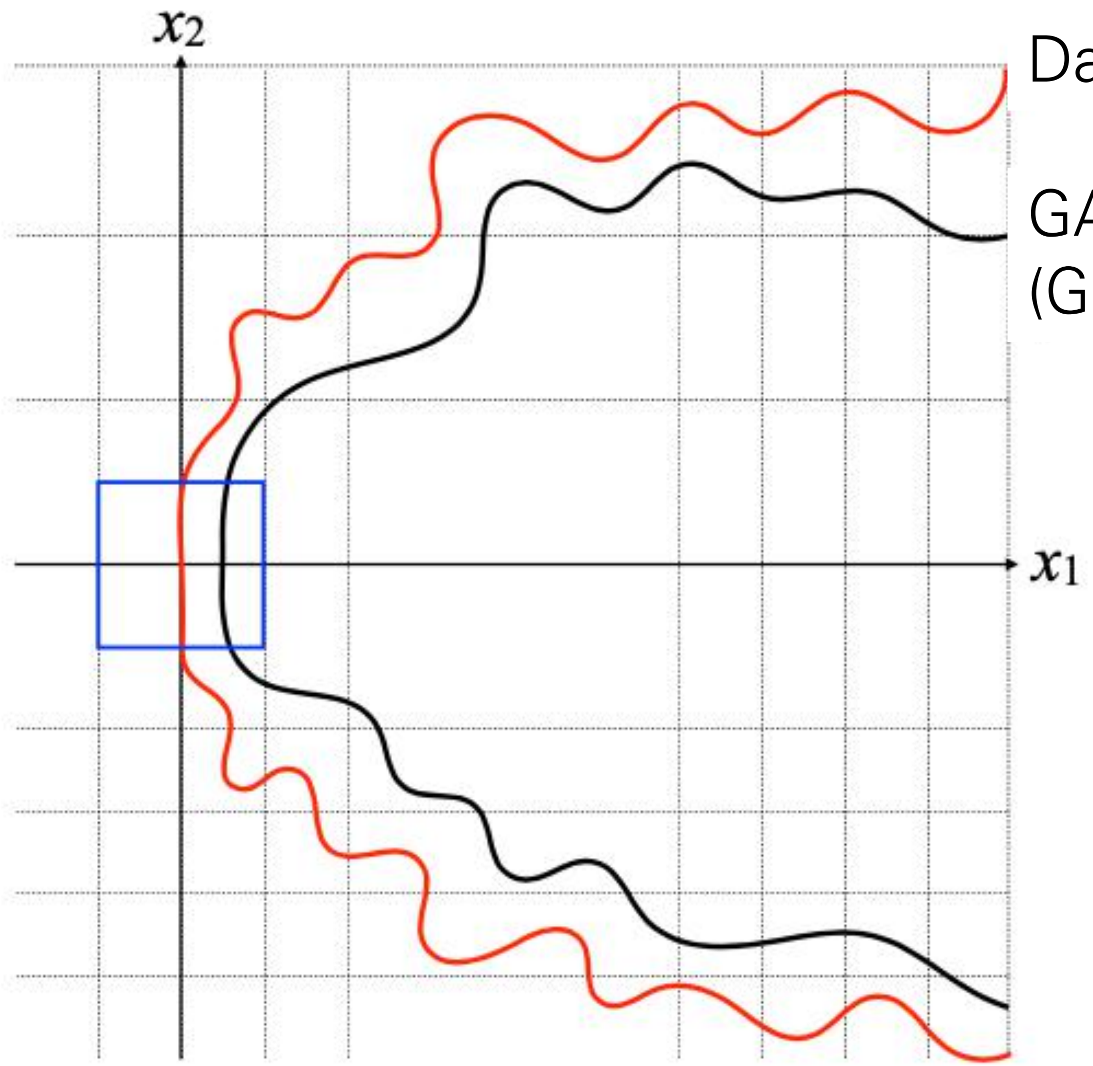


Figure 3: **CSM** (dashed red arrows) adds random 3×3 convolutions and bilinear upsampling, yielding a U-Network.

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - Improved Training of GANs (Salimans et al'16), Projected GAN (Sauer et al'21)
  - **WGAN**, WGAN-GP, Progressive GAN, SN-GAN, SAGAN, Projected GAN
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

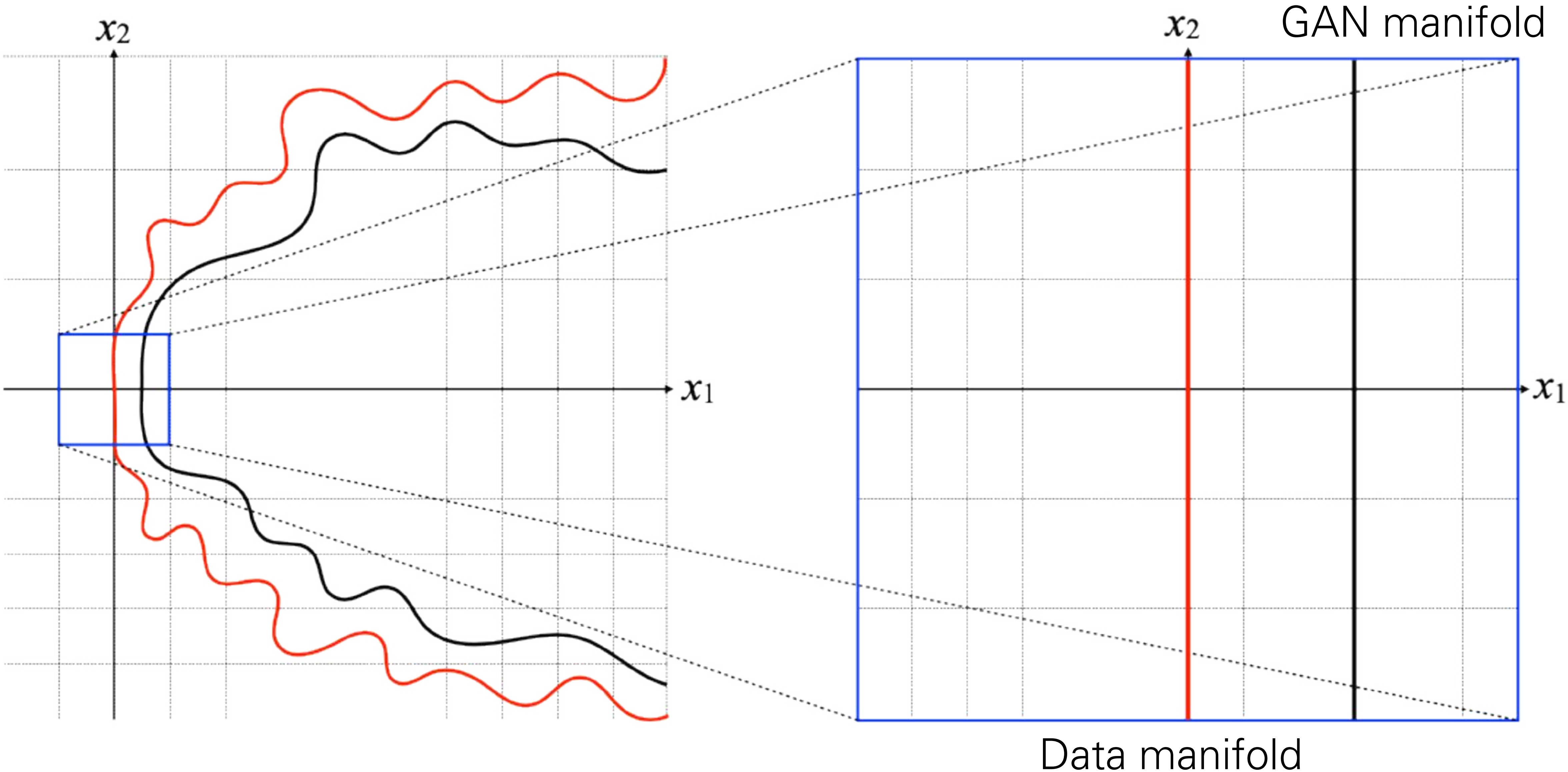
Training a GAN: Distances between Manifolds



Data manifold

GAN manifold
(Generative model)

Training a GAN: Distances between Manifolds

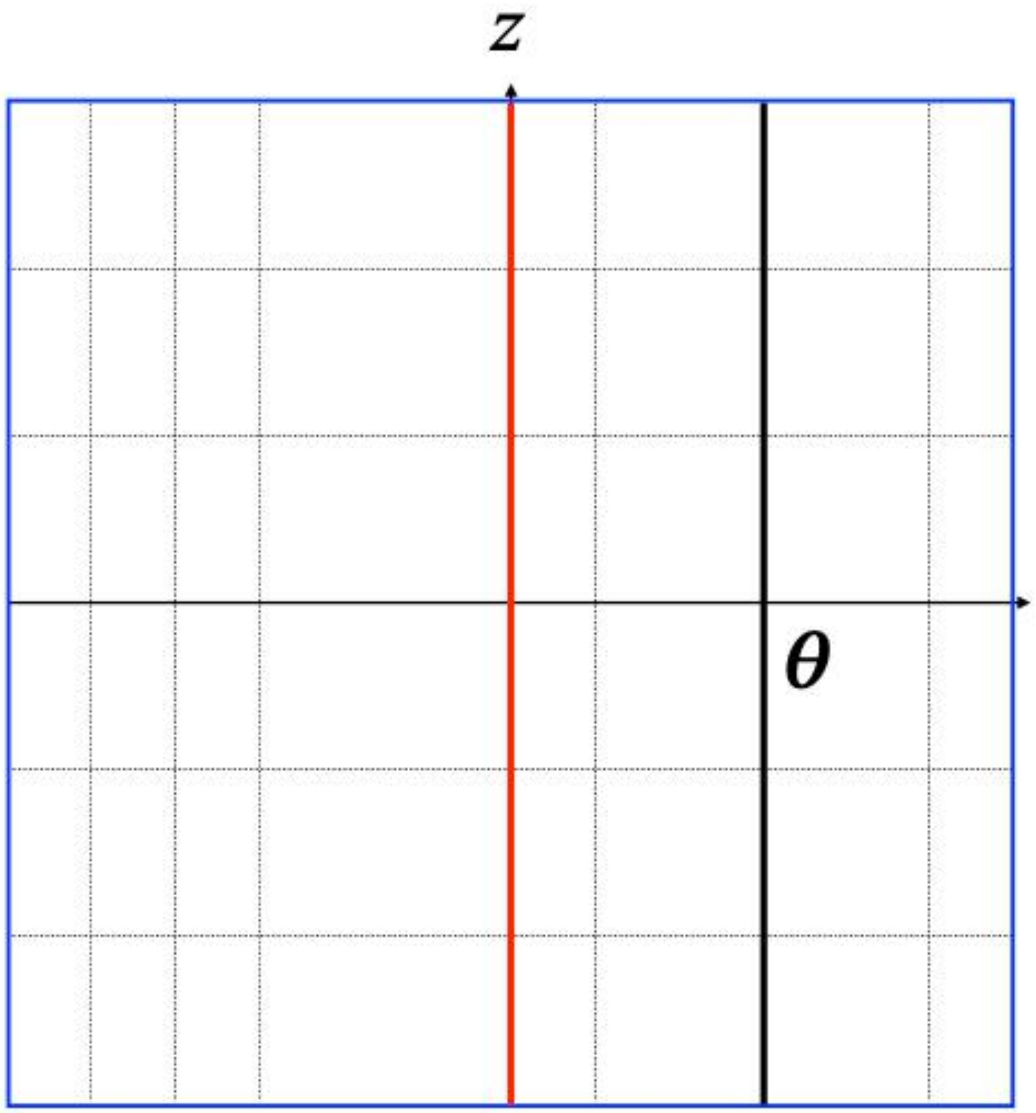


Jensen-Shannon Divergence

$$\mathrm{JS}\left(\mathbb{P}_r \| \mathbb{P}_g\right) = \mathrm{KL}\left(\mathbb{P}_r \| \frac{\mathbb{P}_r + \mathbb{P}_g}{2}\right) + \mathrm{KL}\left(\mathbb{P}_g \| \frac{\mathbb{P}_r + \mathbb{P}_g}{2}\right)$$

- What is the JS divergence in this simple case?

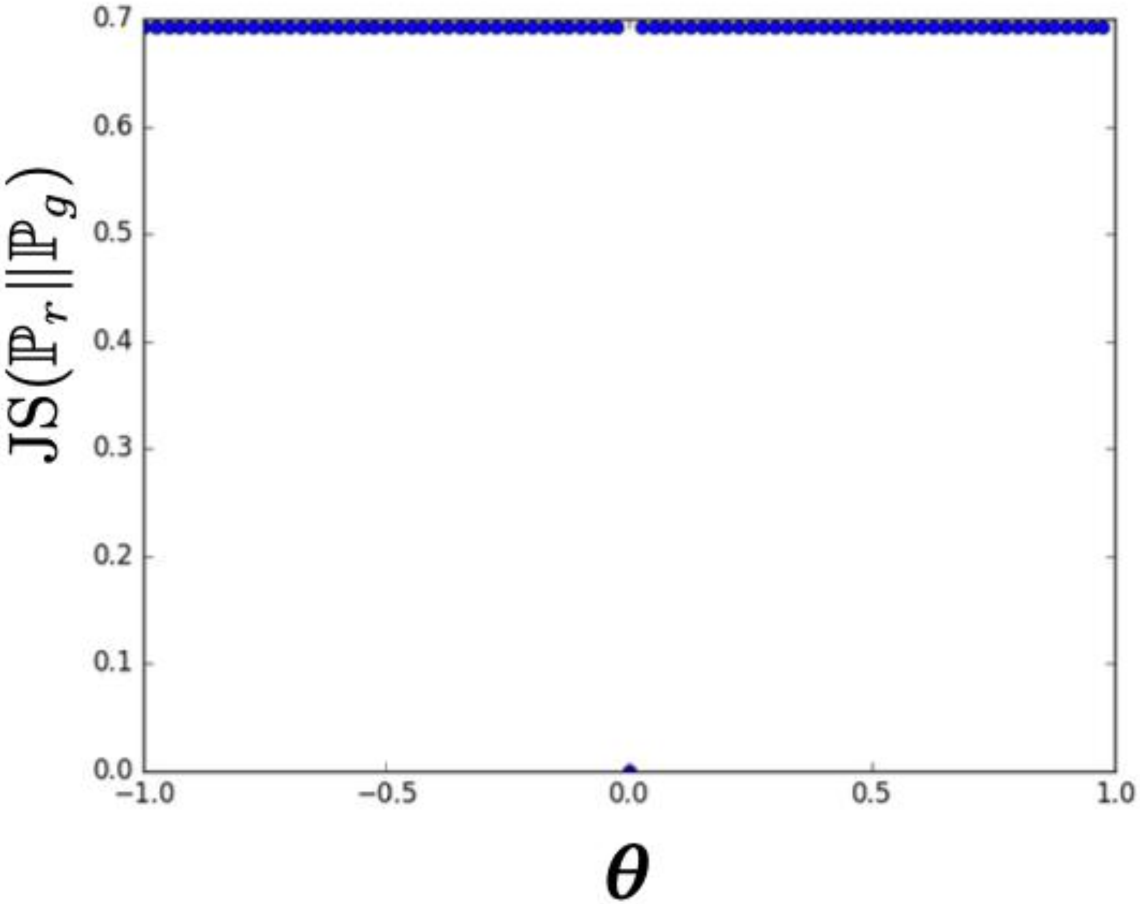
$$\mathrm{JS}\left(\mathbb{P}_r \| \mathbb{P}_g\right) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$$

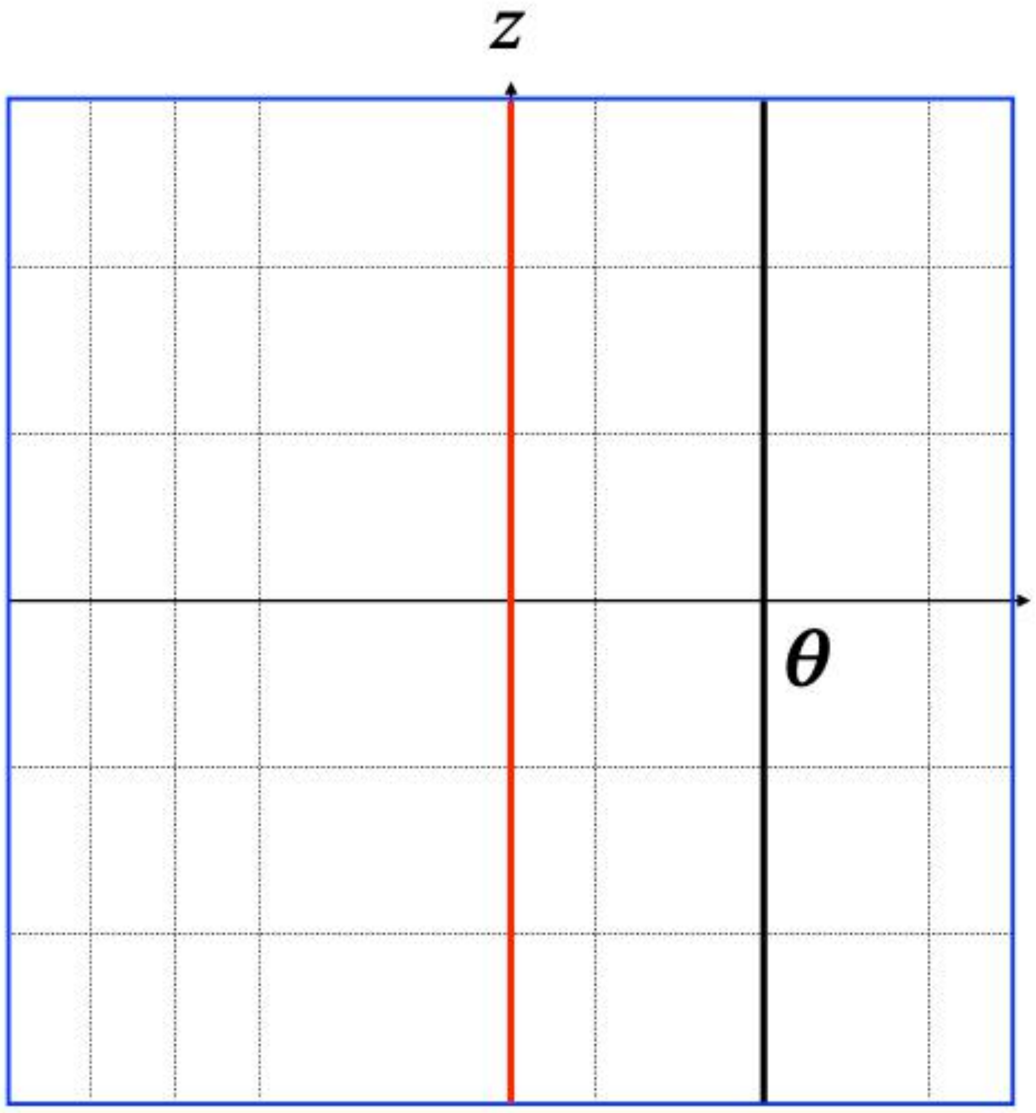


Example from (Arjovsky et al. 2017)

Jensen-Shannon Divergence

$$\mathrm{JS}\left(\mathbb{P}_r \| \mathbb{P}_g\right) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$$





Example from (Arjovsky et al. 2017)

Wasserstein Distance

- JS divergence is not a useful learning signal to train GANs.
- Another distance measure inspired from Optimal Transport is the Earth Mover (EM) (also called Wassertein-1 Distance) distance

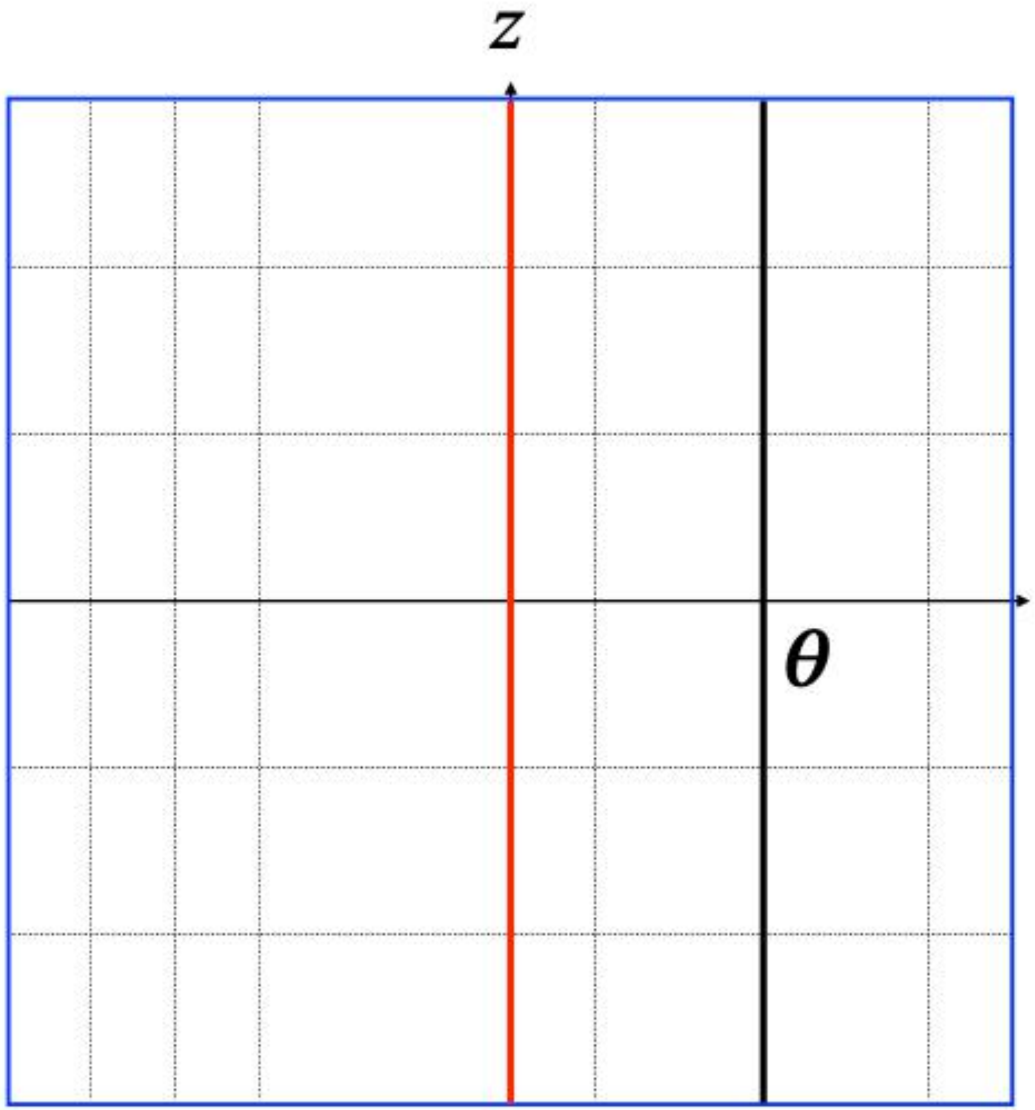
$$W\left(\mathbb{P}_r, \mathbb{P}_g\right) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|]$$

- The EM distance is continuous everywhere and differentiable almost everywhere (under mild assumptions).

Wasserstein Distance

$$W\left(\mathbb{P}_r, \mathbb{P}_g\right) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

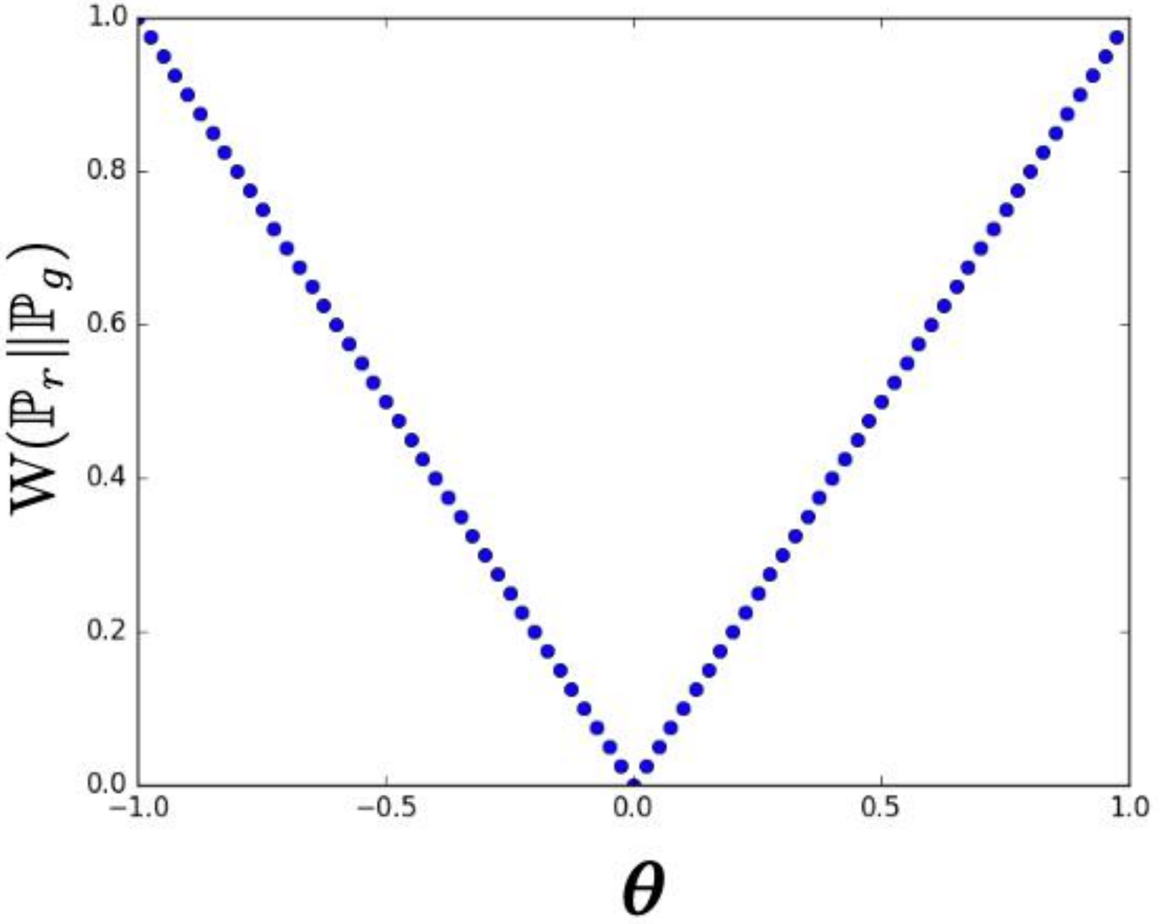
- What is the EM (or Wassertein) distance in this simple case?

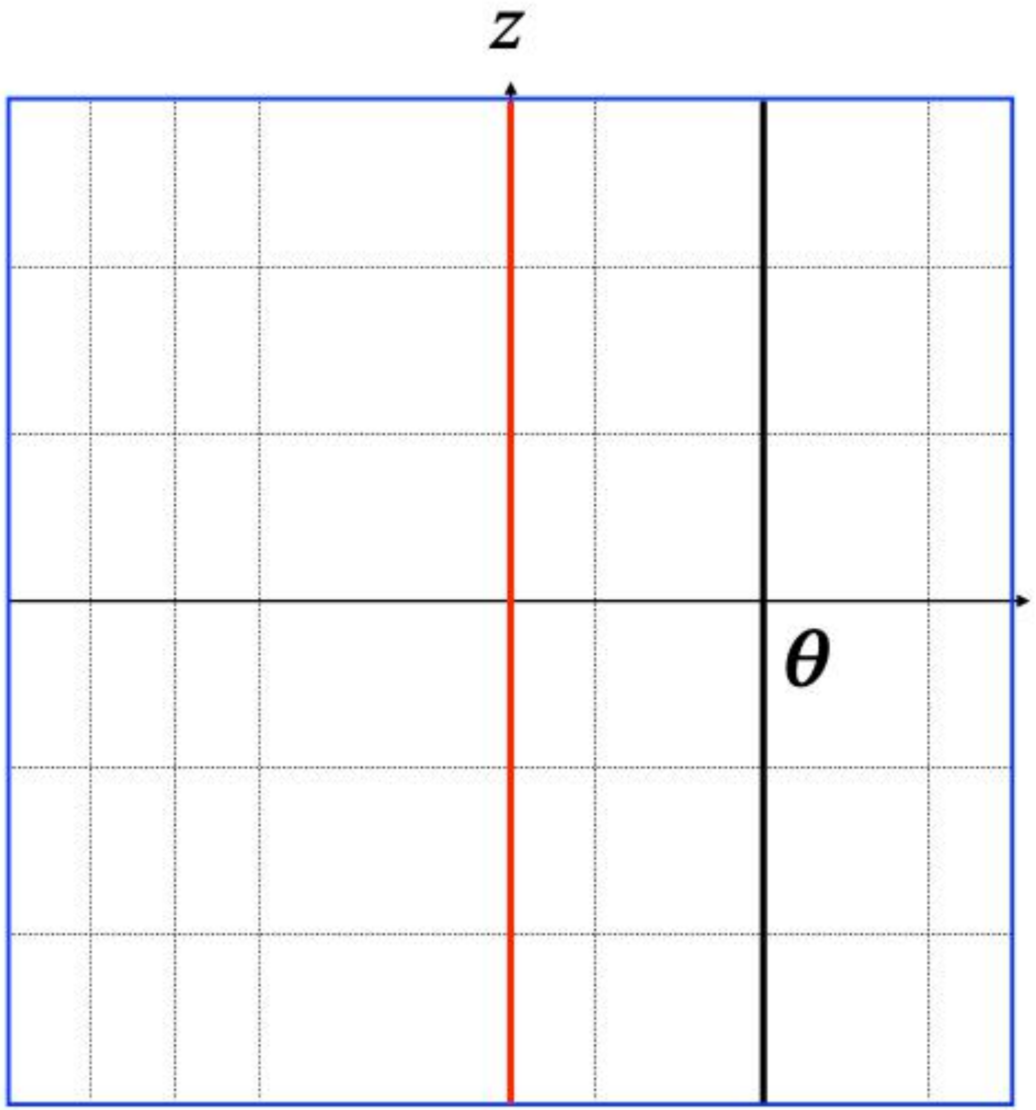


Example from (Arjovsky et al. 2017)

Wasserstein Distance

$$\mathrm{W}\left(\mathbb{P}_r \| \mathbb{P}_g\right) = |\theta|$$

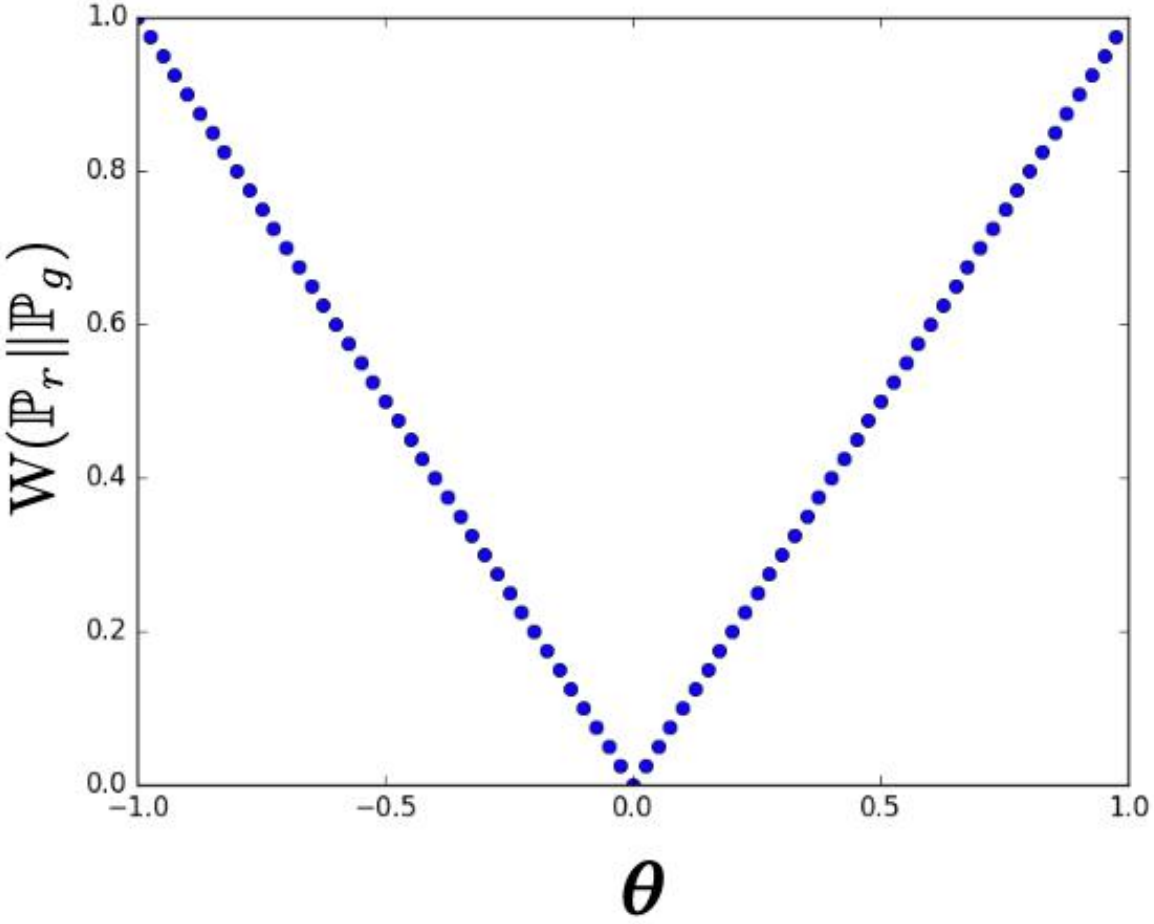


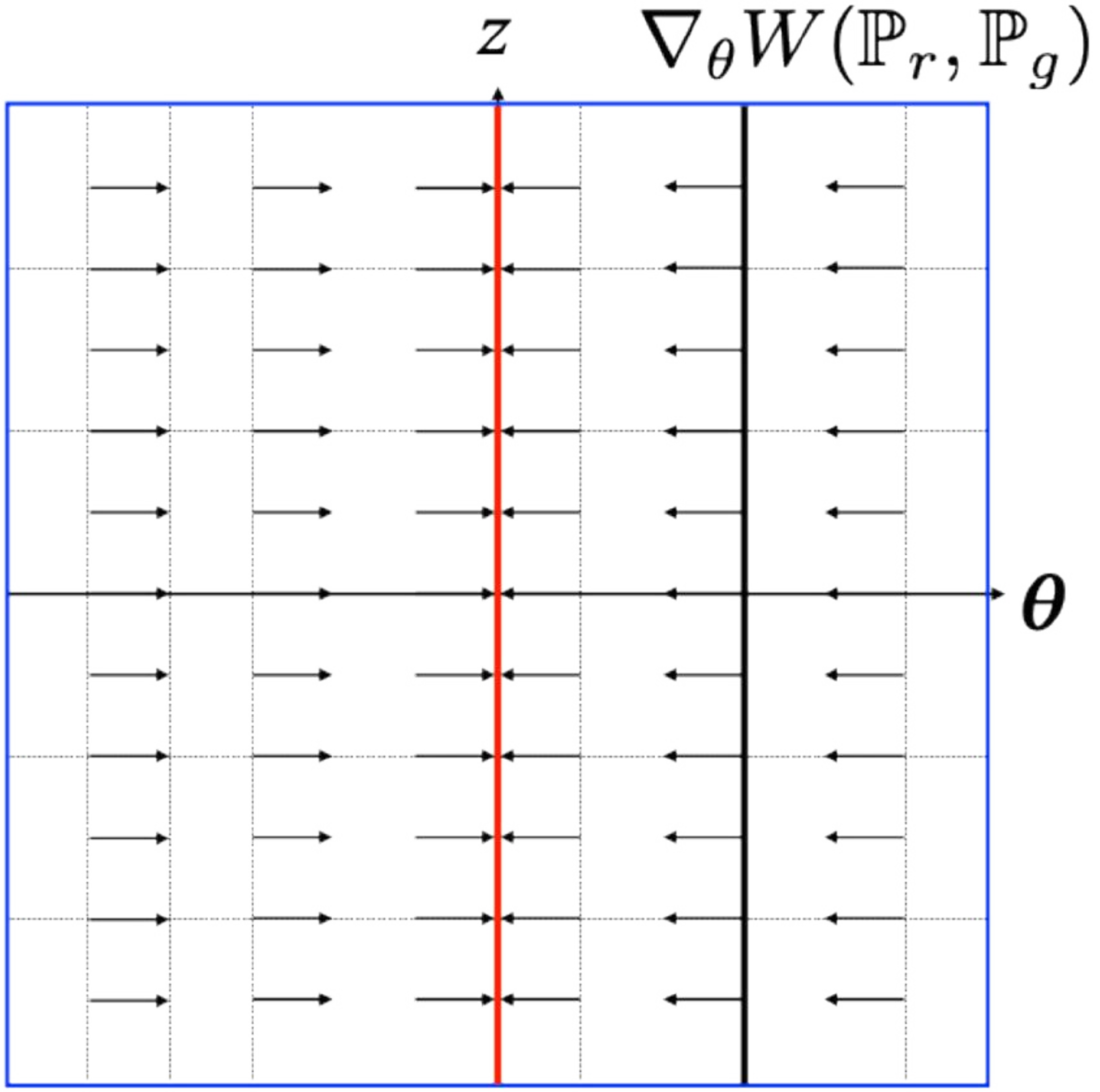


Example from (Arjovsky et al. 2017)

Wasserstein Distance

$$\mathrm{W}\left(\mathbb{P}_r \| \mathbb{P}_g\right) = |\theta|$$





Example from (Arjovsky et al. 2017)

Wasserstein GAN

A real-valued function *f*: X→Y is called *K*-Lipschitz continuous if there exists a real constant $K \geq 0$ such that, for all x_1,x_2.

$$|f(x_1) - f(x_2)| \leq K\,|x_1 - x_2|$$

Here *K* is known as a Lipschitz constant for function $f(\cdot)$

- $\mathrm{W}\left(\mathbb{P}_r \| \mathbb{P}_g\right)$ might have nice properties compared to $\mathrm{JS}\left(\mathbb{P}_r \| \mathbb{P}_g\right)$
- However, the infimum is intractable in:

$$W\left(\mathbb{P}_r, \mathbb{P}_g\right) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

- Can exploit Kantorovich-Rubinstein duality:

$$W\left(\mathbb{P}_r, \mathbb{P}_g\right) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)]$$

where the supremum is over all the 1-Lipschitz functions $f : \mathcal{X} \to \mathbb{R}$

Wasserstein GAN

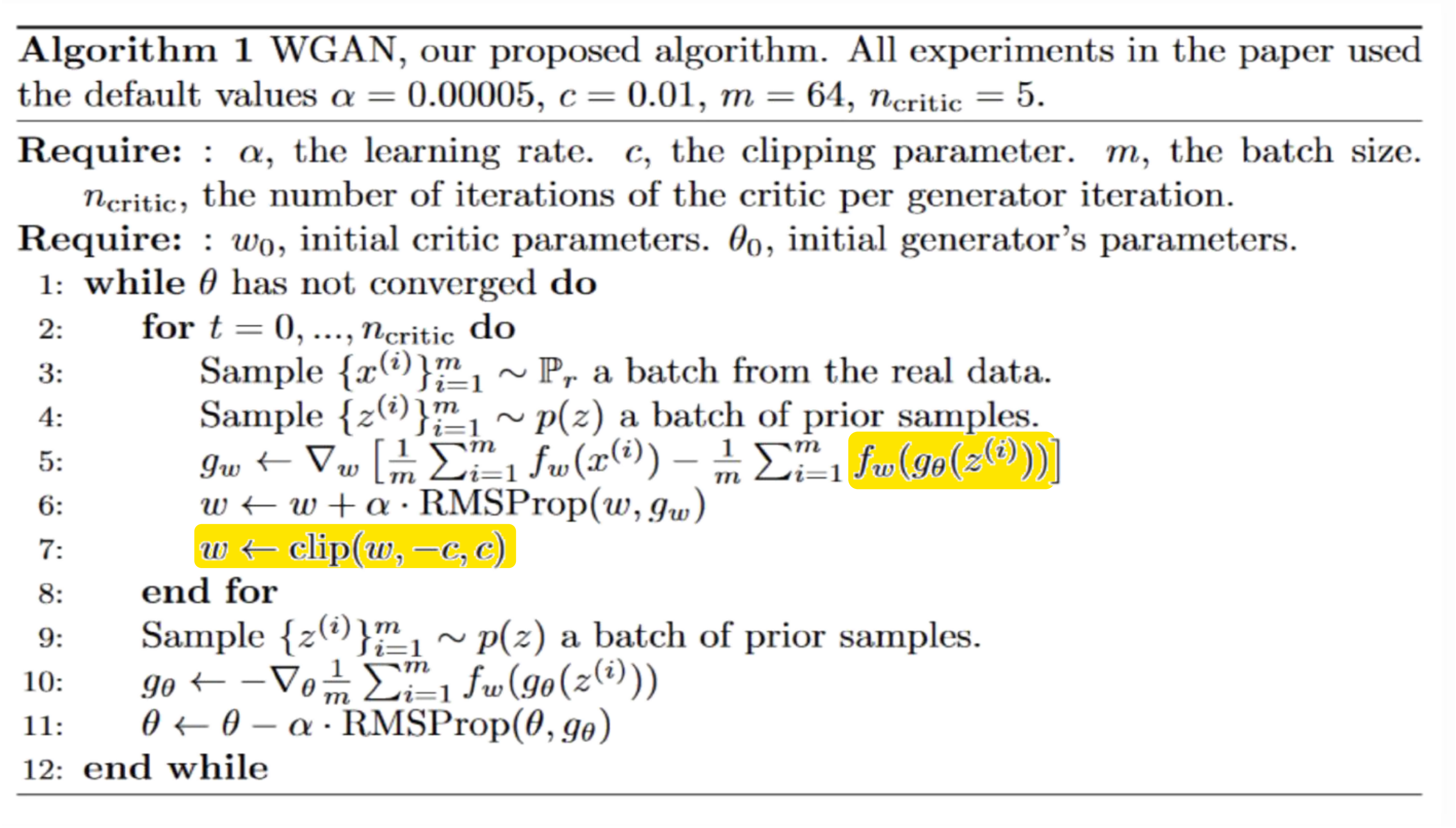
- The WGAN Objective function:

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [D(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [D(\tilde{\boldsymbol{x}}))\Bigg]$$

  where $\mathcal{D}$ is the set of 1-Lipschitz functions.

- Open question: how to effectively enforce the Lipschitz constraint on the critic D?
  - Arjovsky et al. (2017) propose to clip the weights of the critic to lie within a compact space [-c, c].
  - Results in a subset of the k -Lipschitz functions (k is a function of c).

Wasserstein GAN - Pseudocode

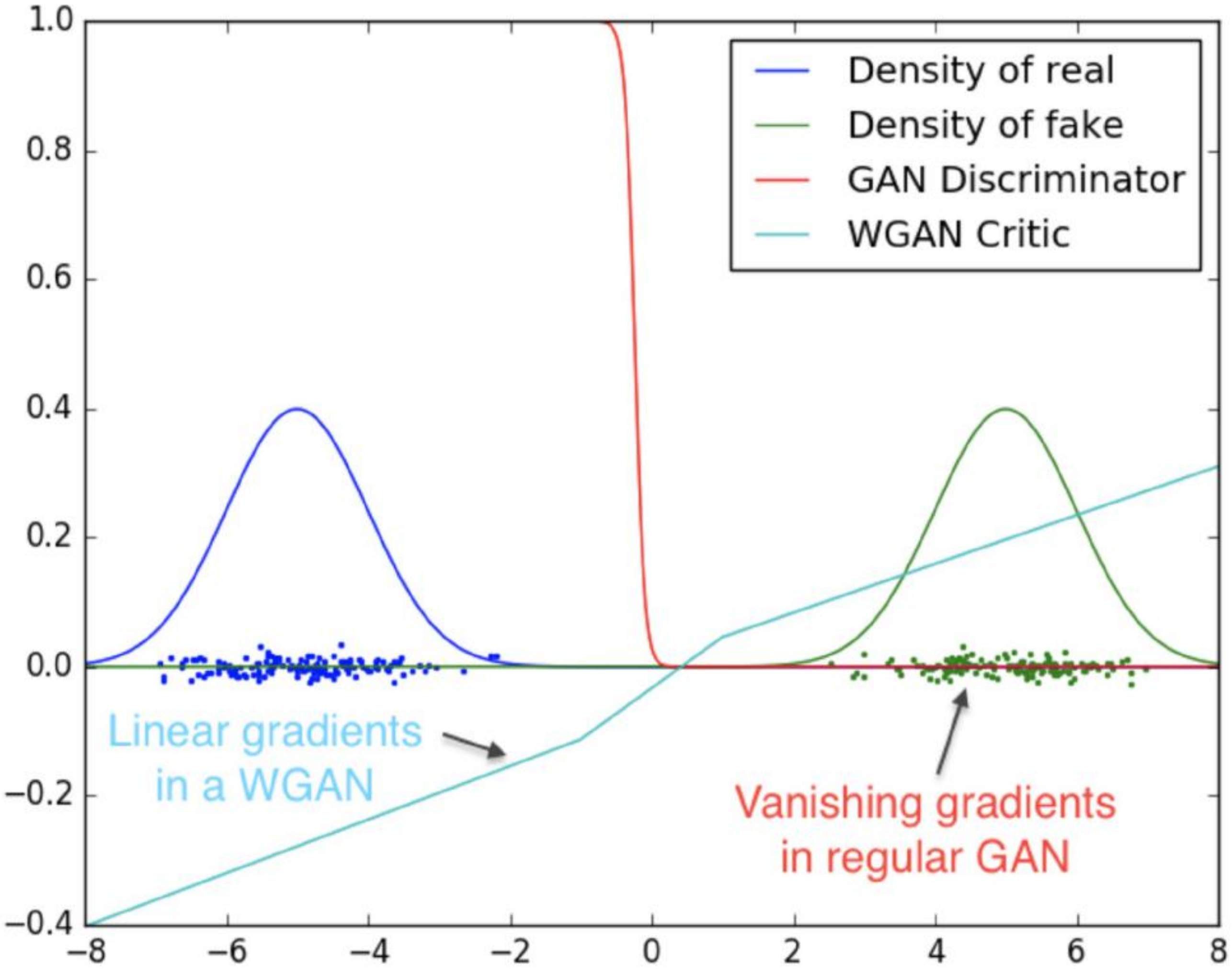


Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

```
Require: : $\alpha$, the learning rate. $c$, the clipping parameter. $m$, the batch size.
    $n_{\text{critic}}$, the number of iterations of the critic per generator iteration.
Require: : $w_0$, initial critic parameters. $\theta_0$, initial generator's parameters.
while $\theta$ has not converged do
    for $t = 0, ..., n_{\text{critic}}$ do
        Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
        Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
        $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
        $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
        $w \leftarrow \text{clip}(w, -c, c)$
    end for
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
    $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while
```

[Arjovsky et al 2017]

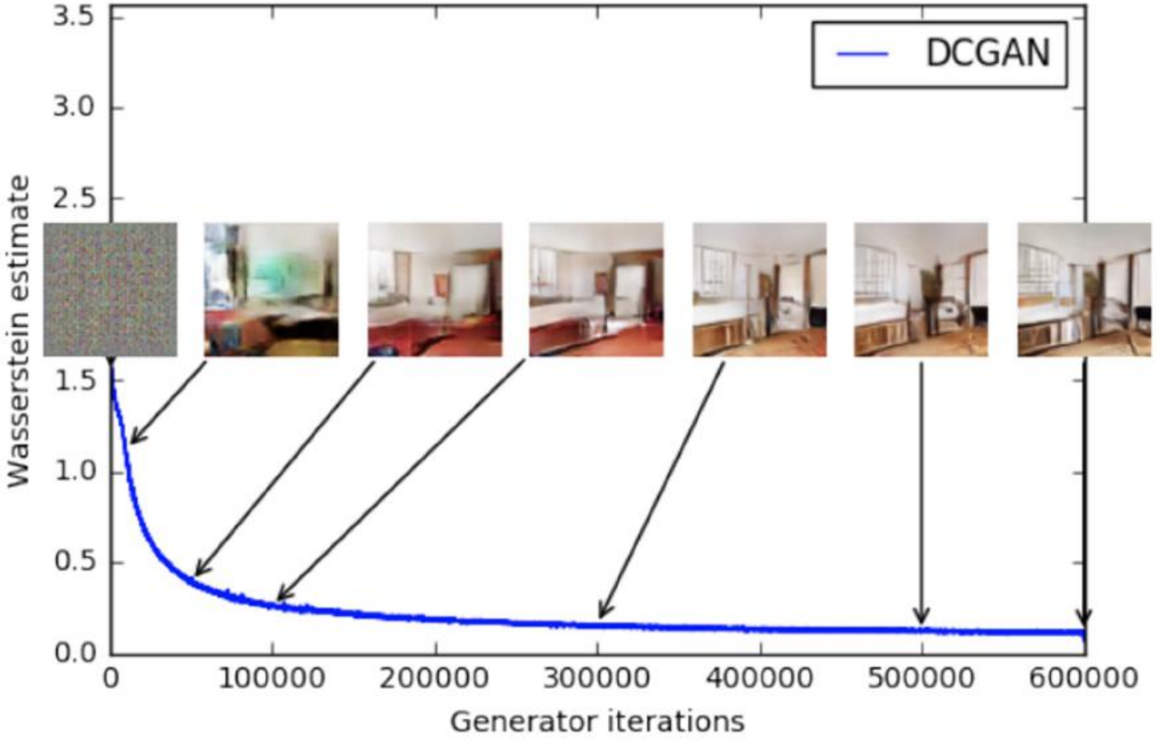
Wasserstein GAN - Training critic to converge



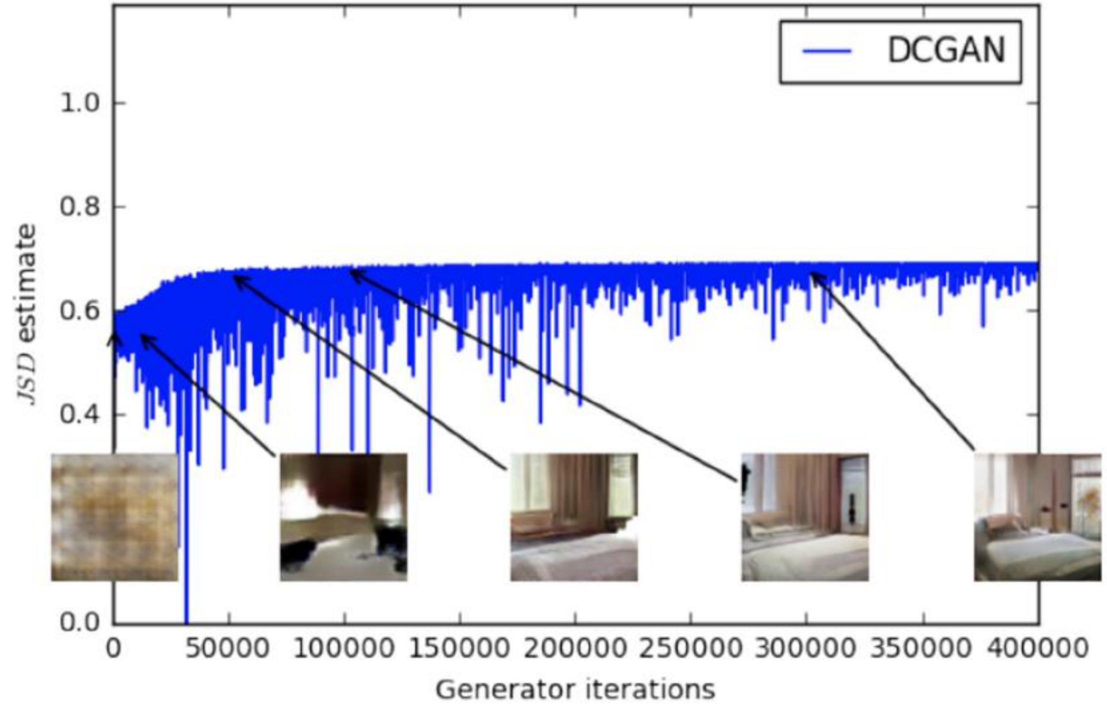
[Arjovsky et al 2017]

Wasserstein distance correlates with sample quality

Wasserstein Estimate

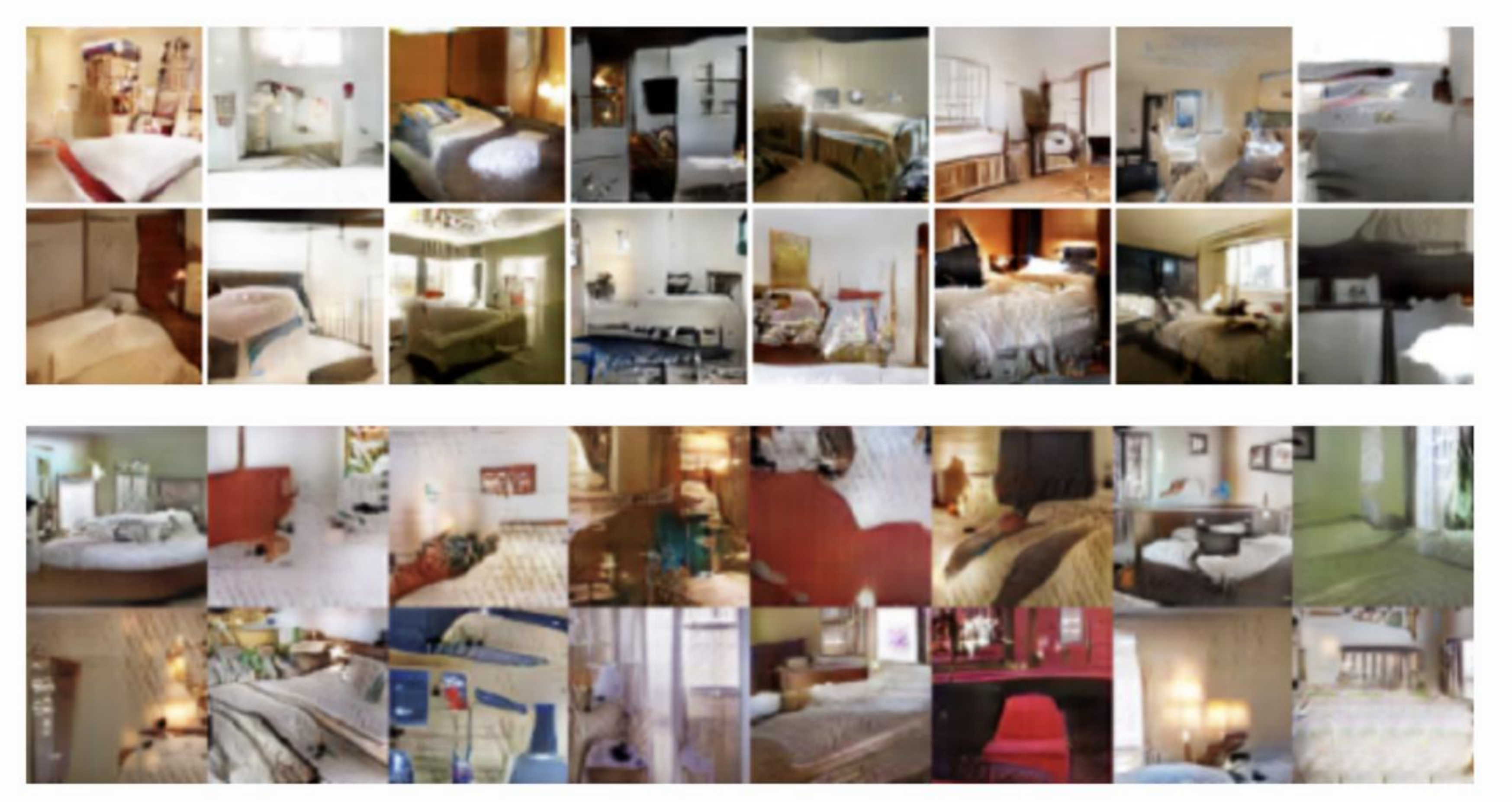


JSD Estimate



[Arjovsky et al 2017]

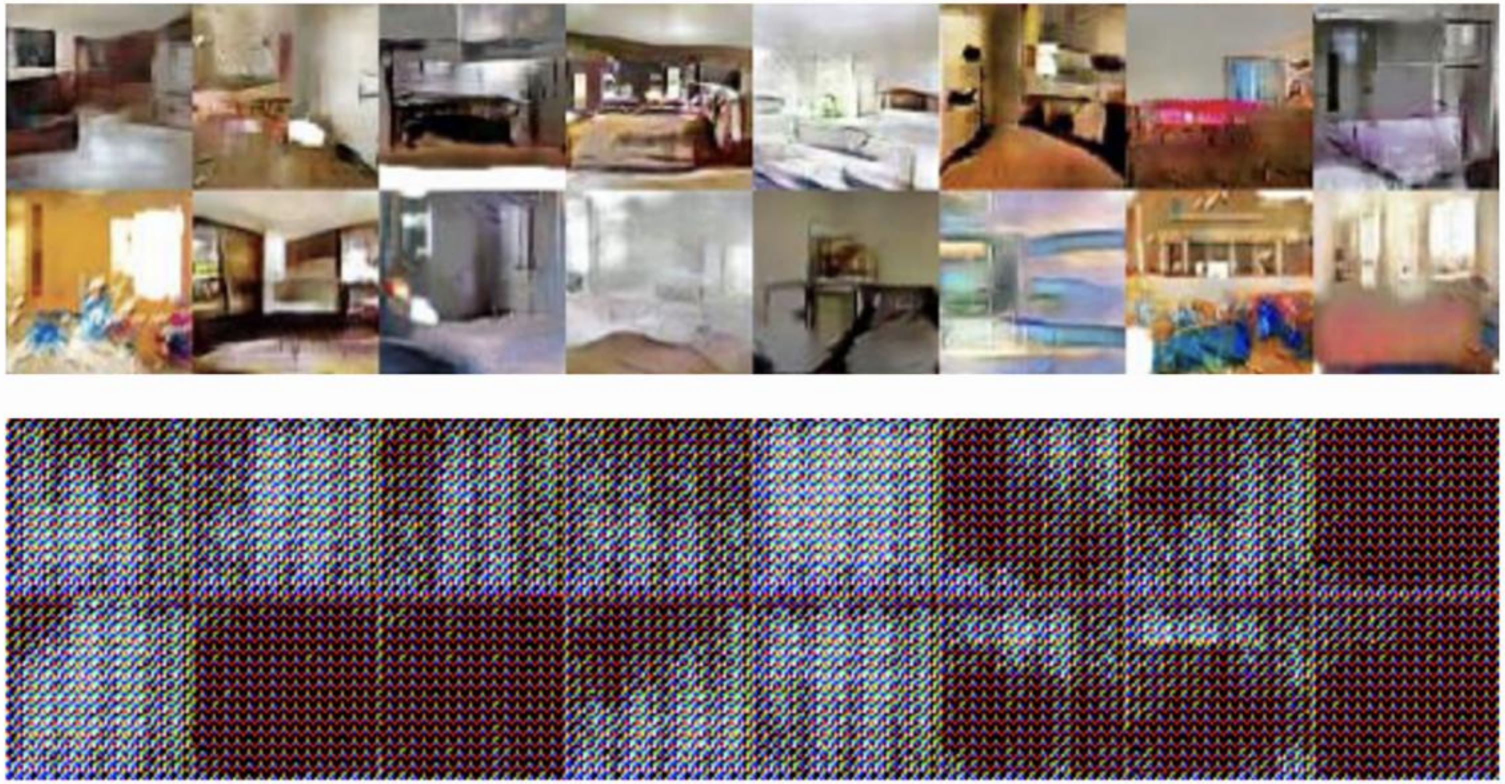
WGAN Samples on par with DCGAN



Top: WGAN with the same DCGAN architecture. Bottom: DCGAN

[Arjovsky et al 2017]

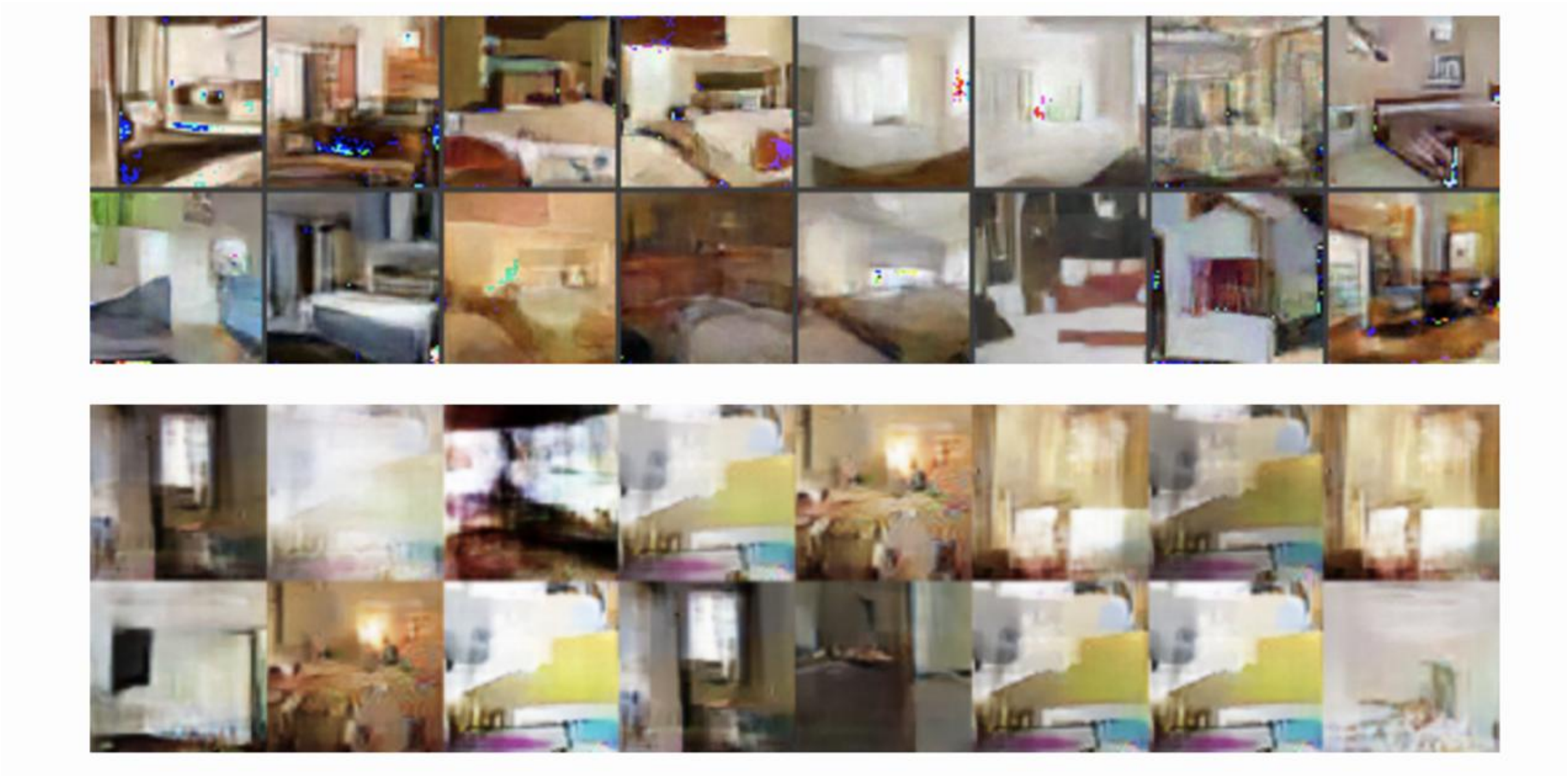
WGAN robust to architecture choices



Top: WGAN with DCGAN architecture, no batch norm. Bottom: DCGAN, no batch norm

[Arjovsky et al 2017]

WGAN robust to architecture choices



Top: WGAN with MLP architecture. Bottom: Standard GAN, same architecture

[Arjovsky et al 2017]

WGAN Summary

Standard GAN

$$\min_{G} \max_{D} \mathbb{E}_{x \sim P_r} \left[\log D(x)\right] + \mathbb{E}_{\tilde{x} \sim P_g} \left[\log(1 - D(\tilde{x}))\right]$$

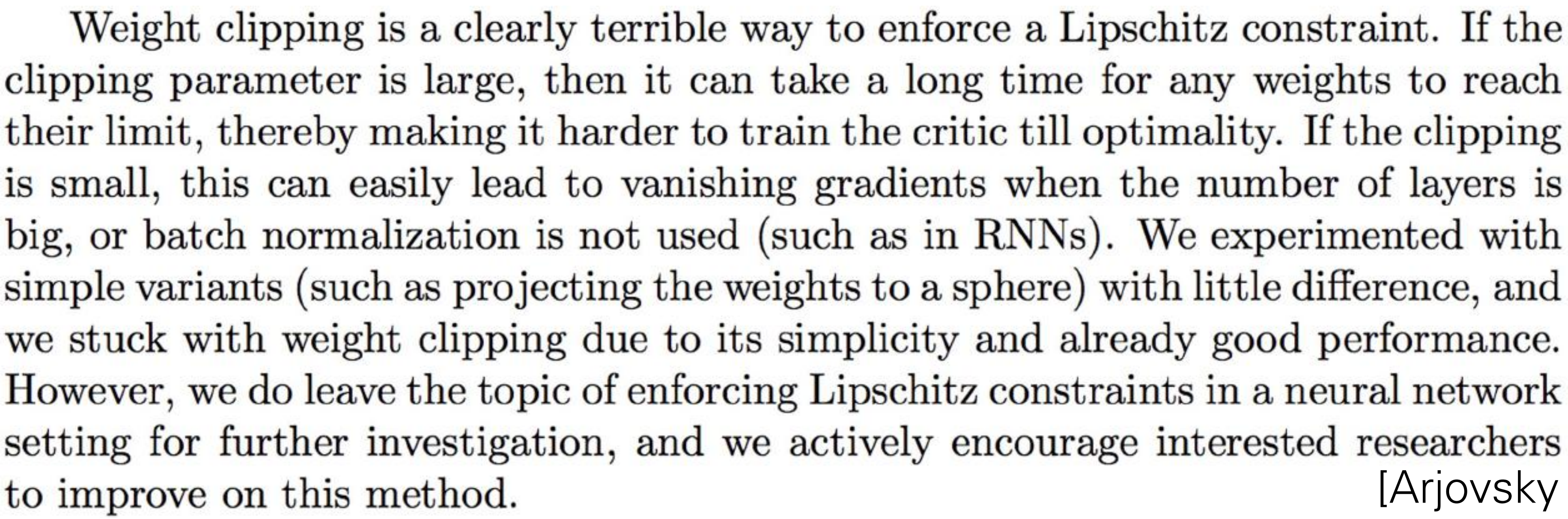
Wasserstein GAN

$$\min_{G} \max_{D \in \mathscr{D}} \mathbb{E}_{x \sim P_r} \left[D(x)\right] - \mathbb{E}_{\tilde{x} \sim P_g} \left[D(\tilde{x})\right]$$

[Arjovsky et al 2017]

WGAN Summary

- New divergence measure for optimizing the generator
- Addresses instabilities with JSD version (sigmoid cross entropy)
- Robust to architectural choices
- Progress on mode collapse and stability of derivative wrt input
- Introduces the idea of using lipschitzness to stabilize GAN training
- Negative:



Weight clipping is a clearly terrible way to enforce a Lipschitz constraint. If the clipping parameter is large, then it can take a long time for any weights to reach their limit, thereby making it harder to train the critic till optimality. If the clipping is small, this can easily lead to vanishing gradients when the number of layers is big, or batch normalization is not used (such as in RNNs). We experimented with simple variants (such as projecting the weights to a sphere) with little difference, and we stuck with weight clipping due to its simplicity and already good performance. However, we do leave the topic of enforcing Lipschitz constraints in a neural network setting for further investigation, and we actively encourage interested researchers to improve on this method.

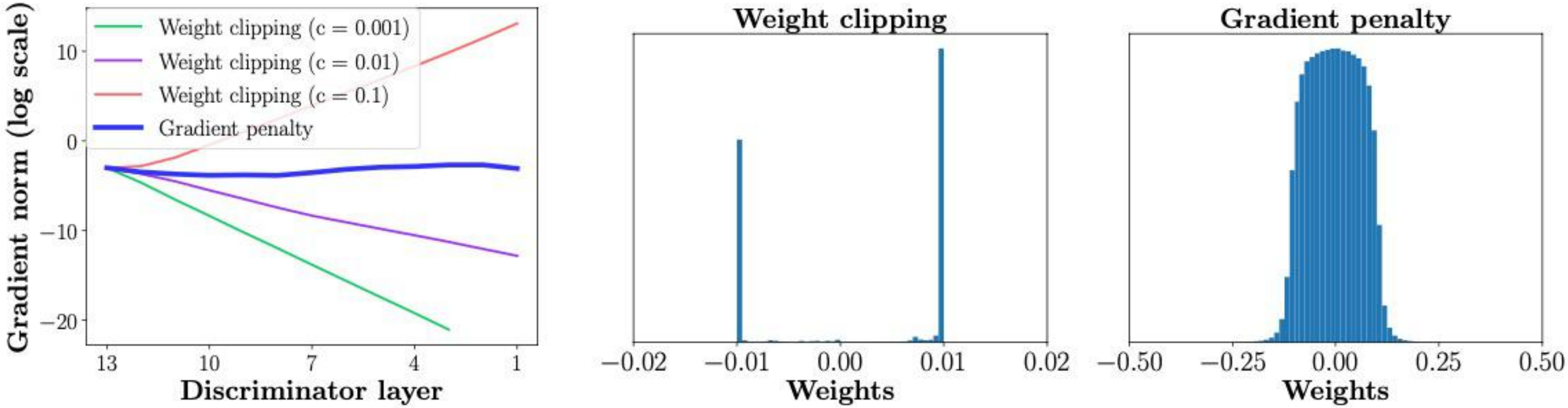
[Arjovsky et al 2017]

Lecture overview

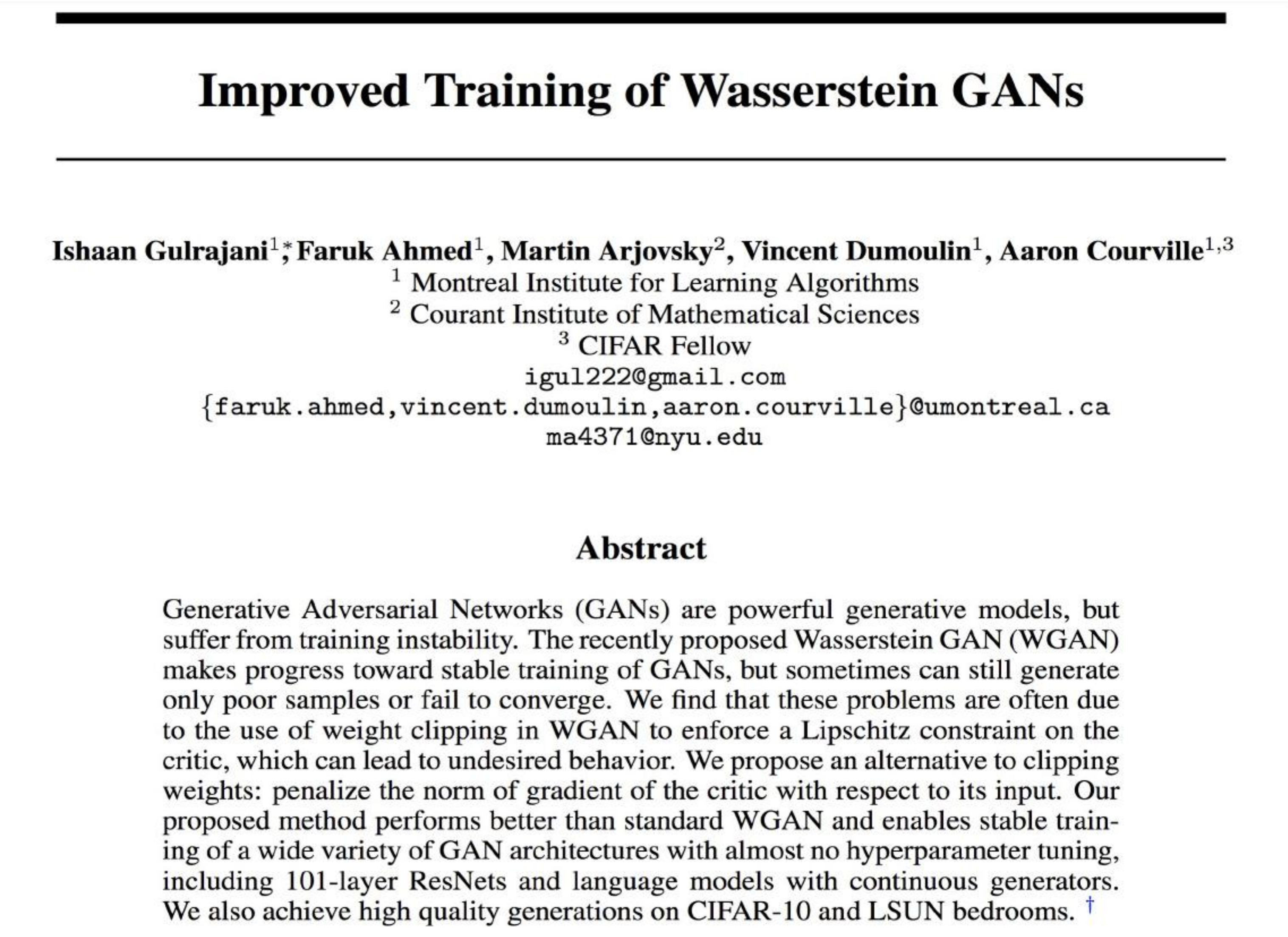
- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - Improved Training of GANs (Salimans et al'16), Projected GAN (Sauer et al'21) WGAN, **WGAN-GP**, Progressive GAN, SN-GAN, SAGAN
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

Issues with Weight Clipping

1. Underuse capacity
2. Exploding and vanishing gradients



WGAN-GP: Gradient Penalty Approach



Improved Training of Wasserstein GANs

Ishaan Gulrajani[1,*], Faruk Ahmed[1], Martin Arjovsky[2], Vincent Dumoulin[1], Aaron Courville[1,3]
[1] Montreal Institute for Learning Algorithms
[2] Courant Institute of Mathematical Sciences
[3] CIFAR Fellow
igul222@gmail.com
{faruk.ahmed,vincent.dumoulin,aaron.courville}@umontreal.ca
ma4371@nyu.edu

Abstract

Generative Adversarial Networks (GANs) are powerful generative models, but suffer from training instability. The recently proposed Wasserstein GAN (WGAN) makes progress toward stable training of GANs, but sometimes can still generate only poor samples or fail to converge. We find that these problems are often due to the use of weight clipping in WGAN to enforce a Lipschitz constraint on the critic, which can lead to undesired behavior. We propose an alternative to clipping weights: penalize the norm of gradient of the critic with respect to its input. Our proposed method performs better than standard WGAN and enables stable training of a wide variety of GAN architectures with almost no hyperparameter tuning, including 101-layer ResNets and language models with continuous generators. We also achieve high quality generations on CIFAR-10 and LSUN bedrooms. †

[Gulrajani et al 2017]

WGAN-GP: Gradient Penalty Approach

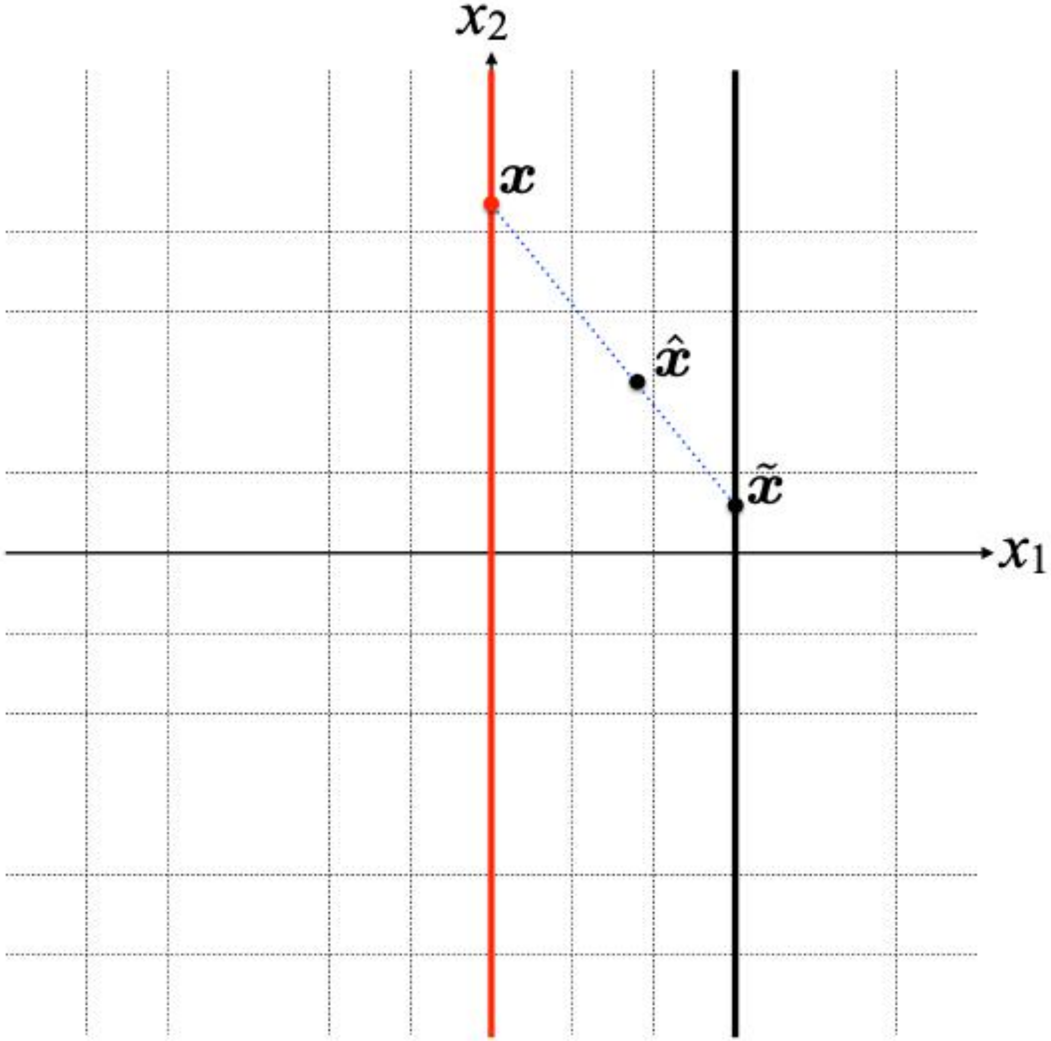
- **A property of the optimal WGAN critic:** If $\tilde{\boldsymbol{x}} \sim \mathbb{P}_g$ then there is a point $\boldsymbol{x} \sim \mathbb{P}_r$, such that for all points $\boldsymbol{x}_t = t\boldsymbol{x} + (1-t)\tilde{\boldsymbol{x}}$ (on a straight line between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$) then:

$$\nabla D^*(\boldsymbol{x}_t) = \frac{\boldsymbol{x} - \boldsymbol{x}_t}{\|\boldsymbol{x} - \boldsymbol{x}_t\|}$$

- This implies the optimal WGAN critic has gradient norm 1 at $\boldsymbol{x}_t$
- Gradient Penalty version of WGAN (i.e. WGAN-GP) objective

$$L = \underbrace{\underset{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g}{\mathbb{E}}\left[D(\tilde{\boldsymbol{x}})\right] - \underset{\boldsymbol{x} \sim \mathbb{P}_r}{\mathbb{E}}\left[D(\boldsymbol{x})\right]}_{\text{Original critic loss}} + \underbrace{\lambda \underset{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}}{\mathbb{E}}\left[\left(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1\right)^2\right]}_{\text{Our gradient penalty}}$$

WGAN-GP: Gradient Penalty Approach



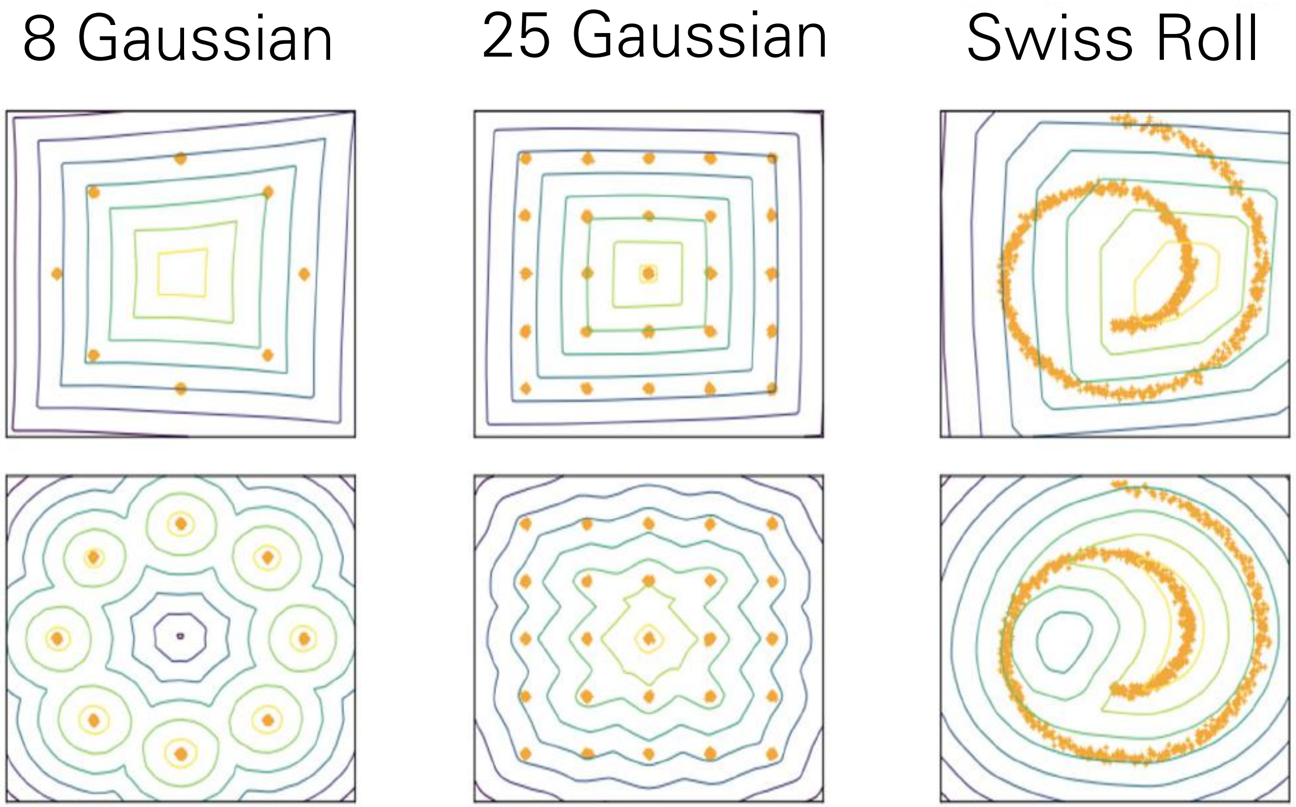
- Gradient penalty:

$$\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[\left(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1 \right)^2 \right]$$

Sample along straight lines:

$$\epsilon \sim U[0,1], \boldsymbol{x} \sim \mathbb{P}_r, \tilde{\boldsymbol{x}} \sim \mathbb{P}_g$$
$$\hat{\boldsymbol{x}} = \epsilon \boldsymbol{x} + (1-\epsilon)\tilde{\boldsymbol{x}}$$

WGAN-GP: Gradient Penalty for Lipschitzness

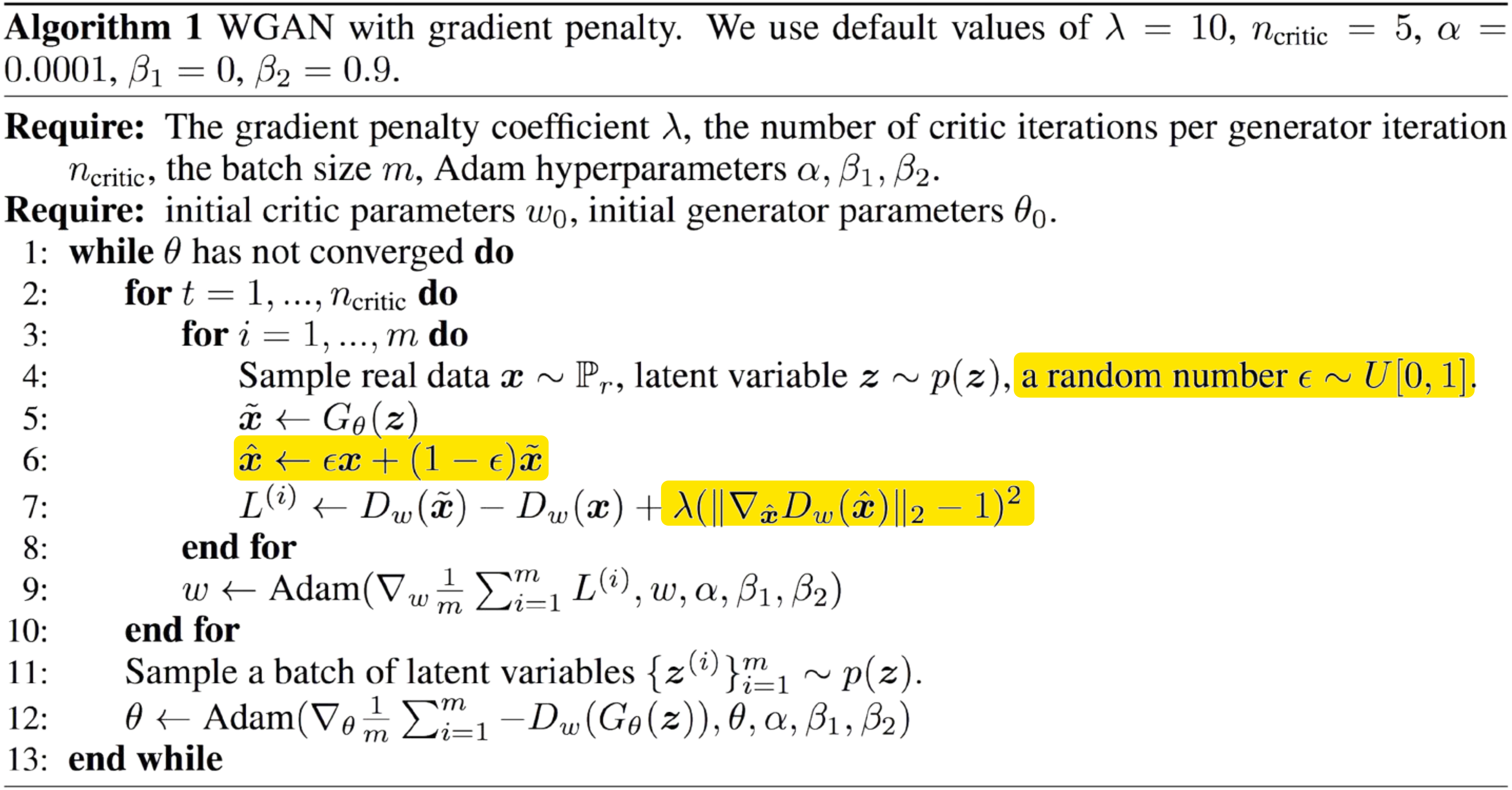


$$\max_{D} \underbrace{\mathbb{E}_{x\sim P_r}\left[D(x)\right] - \mathbb{E}_{\tilde{x}\sim P_g}\left[D(\tilde{x})\right]}_{\text{Wasserstein critic objective}} + \lambda \underbrace{\mathbb{E}_{\hat{x}\sim P_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1\right)^2\right]}_{\text{Gradient Penalty for Lipschitzness}}$$

$$\hat{x} \leftarrow \epsilon x + (1-\epsilon)\tilde{x}$$

[Gulrajani et al 2017]

WGAN-GP: Pseudocode

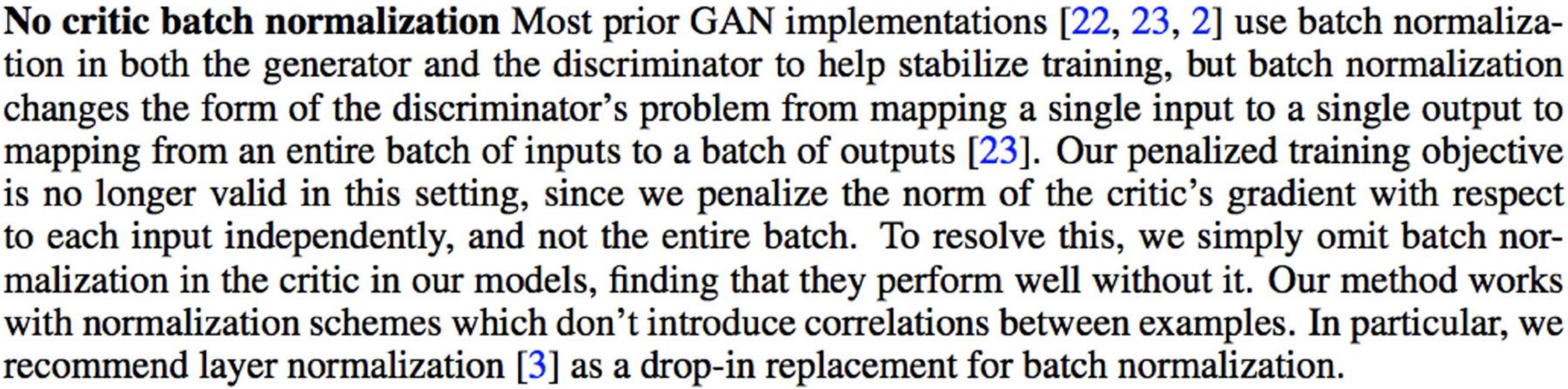


Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.

```
Require: The gradient penalty coefficient $\lambda$, the number of critic iterations per generator iteration
    $n_{\text{critic}}$, the batch size $m$, Adam hyperparameters $\alpha, \beta_1, \beta_2$.
Require: initial critic parameters $w_0$, initial generator parameters $\theta_0$.
while $\theta$ has not converged do
    for $t = 1, ..., n_{\text{critic}}$ do
        for $i = 1, ..., m$ do
            Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0, 1]$.
            $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
            $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$
            $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
        end for
        $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$
    end for
    Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m} \sim p(\boldsymbol{z})$.
    $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
end while
```

[Gulrajani et al 2017]

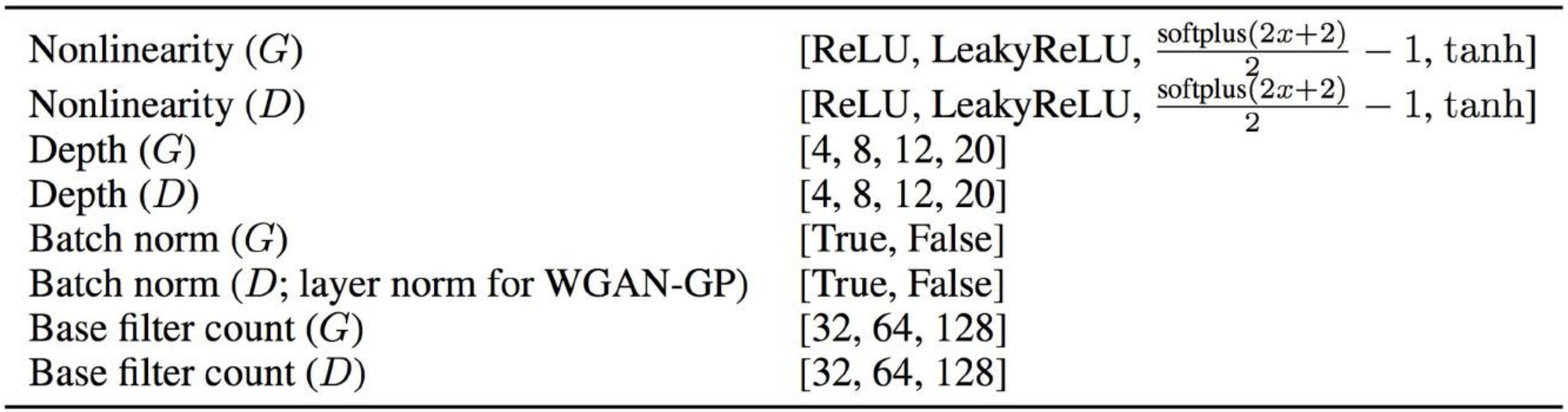
WGAN-GP: BatchNorm



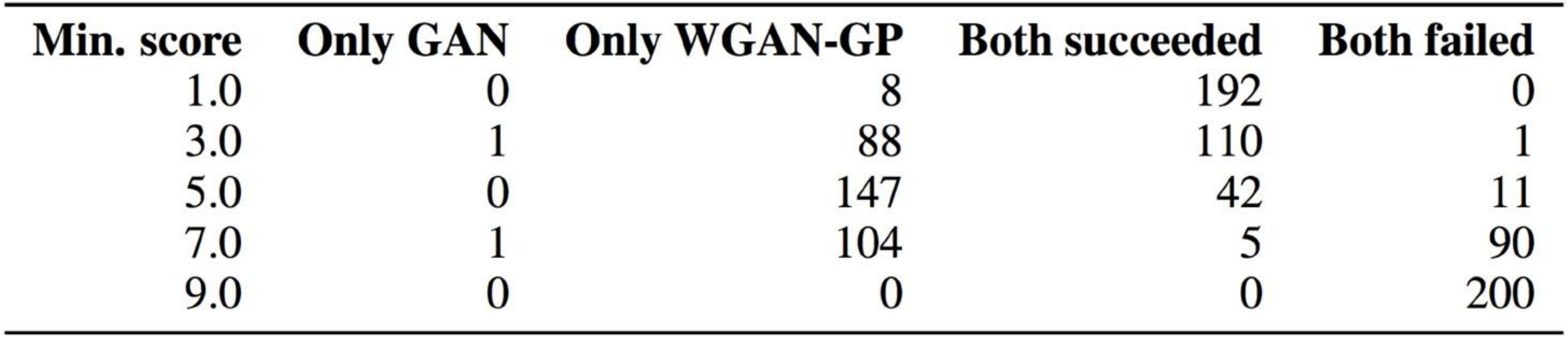
No critic batch normalization Most prior GAN implementations [22, 23, 2] use batch normalization in both the generator and the discriminator to help stabilize training, but batch normalization changes the form of the discriminator's problem from mapping a single input to a single output to mapping from an entire batch of inputs to a batch of outputs [23]. Our penalized training objective is no longer valid in this setting, since we penalize the norm of the critic's gradient with respect to each input independently, and not the entire batch. To resolve this, we simply omit batch normalization in the critic in our models, finding that they perform well without it. Our method works with normalization schemes which don't introduce correlations between examples. In particular, we recommend layer normalization [3] as a drop-in replacement for batch normalization.

[Gulrajani et al 2017]

WGAN-GP: Robustness to architectures



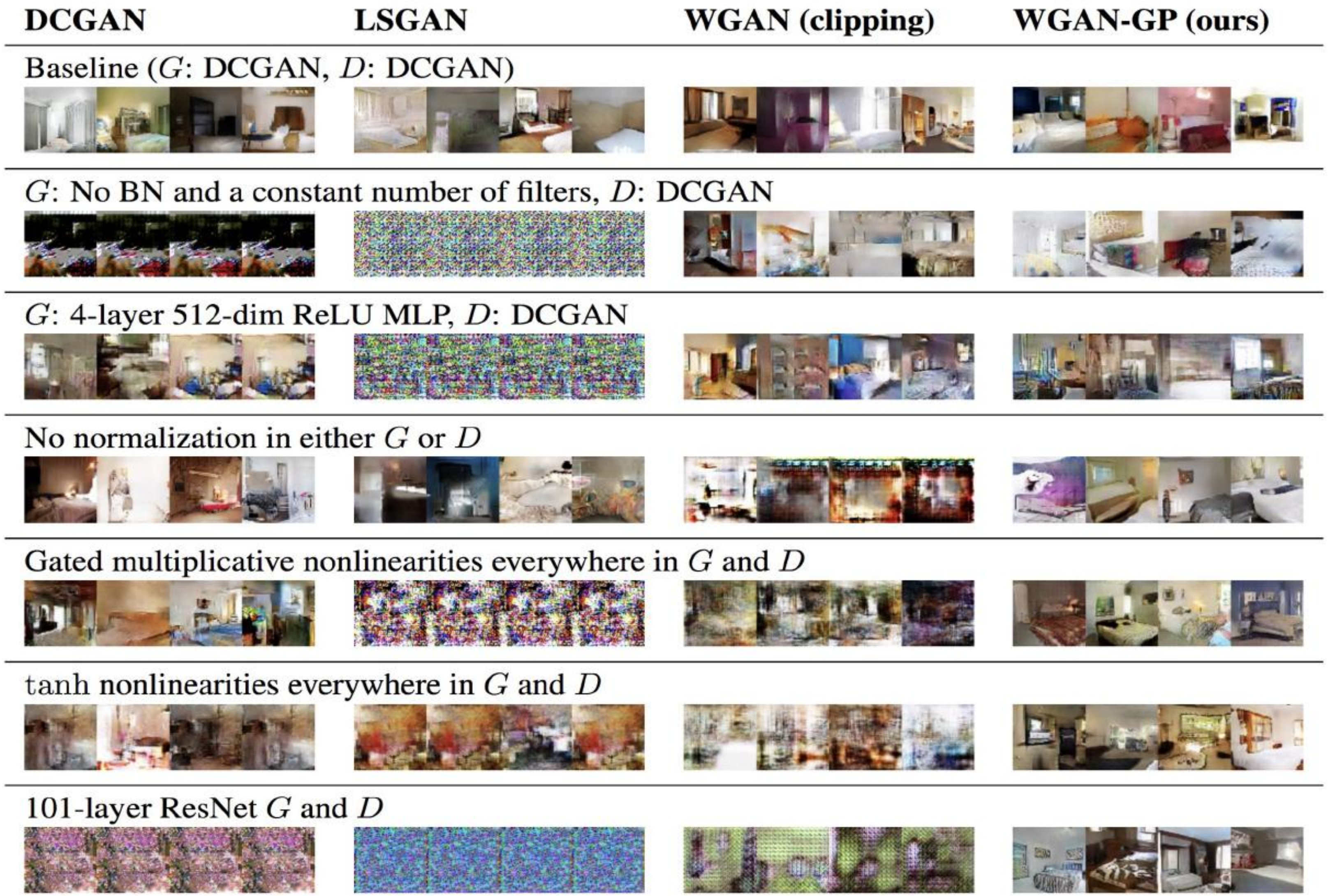
Nonlinearity (G)	[ReLU, LeakyReLU, $\frac{\text{softplus}(2x+2)}{2} - 1$, tanh]
Nonlinearity (D)	[ReLU, LeakyReLU, $\frac{\text{softplus}(2x+2)}{2} - 1$, tanh]
Depth (G)	[4, 8, 12, 20]
Depth (D)	[4, 8, 12, 20]
Batch norm (G)	[True, False]
Batch norm (D; layer norm for WGAN-GP)	[True, False]
Base filter count (G)	[32, 64, 128]
Base filter count (D)	[32, 64, 128]



Min. score	Only GAN	Only WGAN-GP	Both succeeded	Both failed
1.0	0	8	192	0
3.0	1	88	110	1
5.0	0	147	42	11
7.0	1	104	5	90
9.0	0	0	0	200

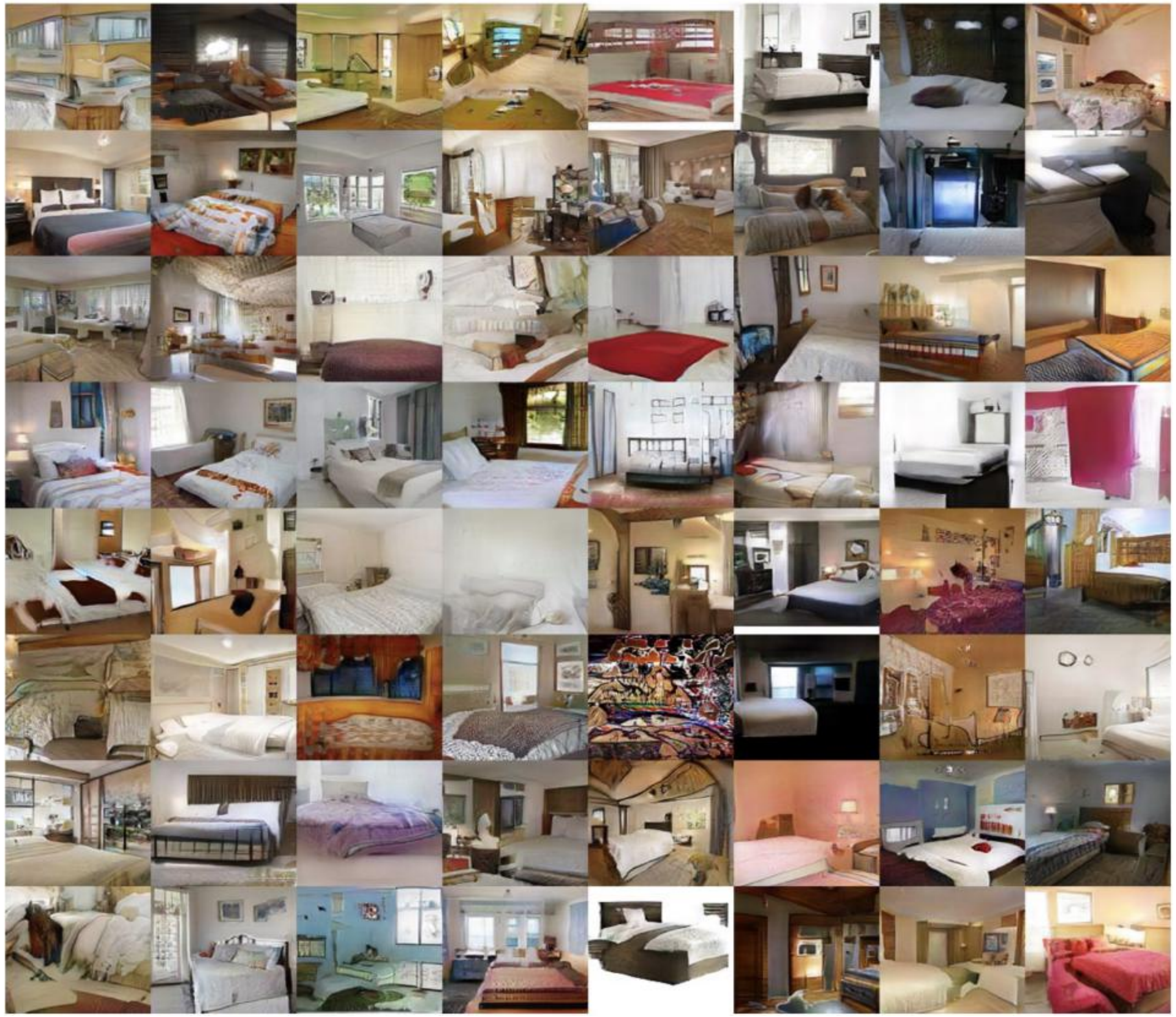
[Gulrajani et al 2017]

WGAN-GP: Robustness to architectures


DCGAN
LSGAN
WGAN (clipping)
WGAN-GP (ours)
Baseline (G: DCGAN, D: DCGAN)
G: No BN and a constant number of filters, D: DCGAN
G: 4-layer 512-dim ReLU MLP, D: DCGAN
No normalization in either G or D
Gated multiplicative nonlinearities everywhere in G and D
tanh nonlinearities everywhere in G and D
101-layer ResNet G and D

[Gulrajani et al 2017]

WGAN-GP: High quality samples



[Gulrajani et al 2017]

WGAN-GP: High quality samples

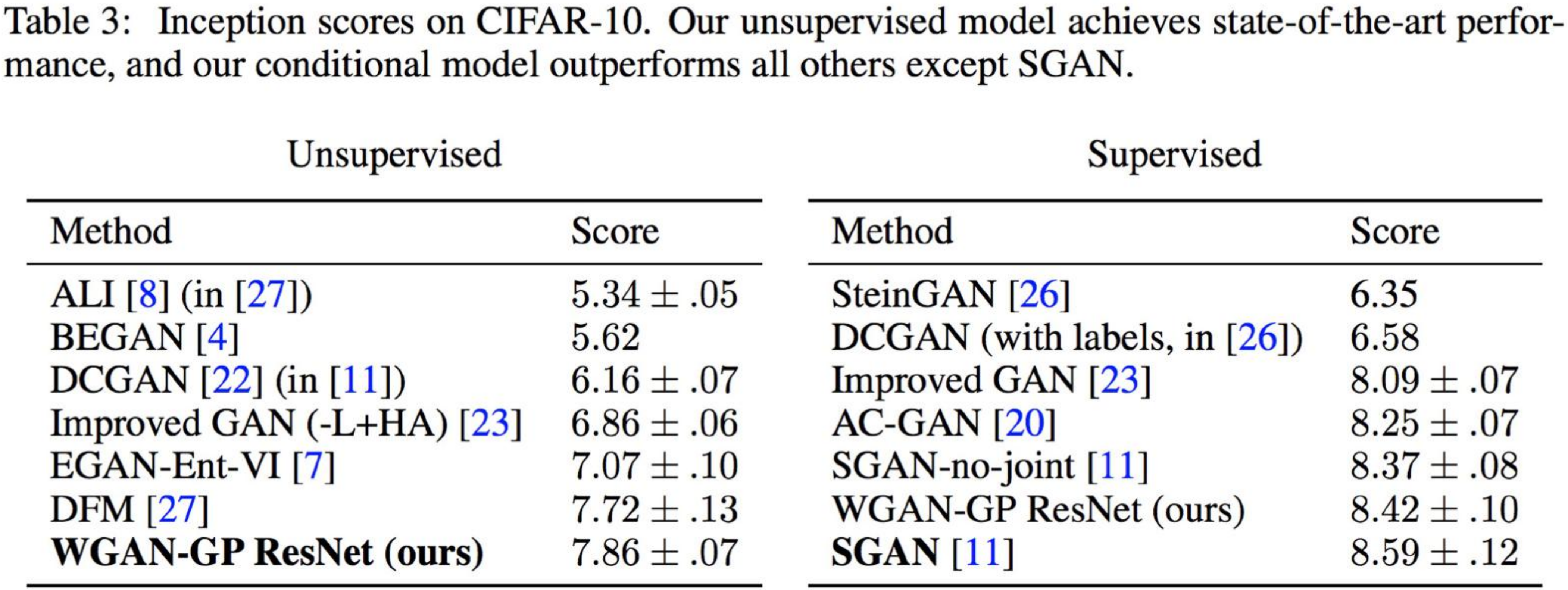


Table 3: Inception scores on CIFAR-10. Our unsupervised model achieves state-of-the-art performance, and our conditional model outperforms all others except SGAN.

Unsupervised

Method	Score
ALI [8] (in [27])	5.34 ± .05
BEGAN [4]	5.62
DCGAN [22] (in [11])	6.16 ± .07
Improved GAN (-L+HA) [23]	6.86 ± .06
EGAN-Ent-VI [7]	7.07 ± .10
DFM [27]	7.72 ± .13
WGAN-GP ResNet (ours)	7.86 ± .07

Supervised

Method	Score
SteinGAN [26]	6.35
DCGAN (with labels, in [26])	6.58
Improved GAN [23]	8.09 ± .07
AC-GAN [20]	8.25 ± .07
SGAN-no-joint [11]	8.37 ± .08
WGAN-GP ResNet (ours)	8.42 ± .10
SGAN [11]	8.59 ± .12

[Gulrajani et al 2017]

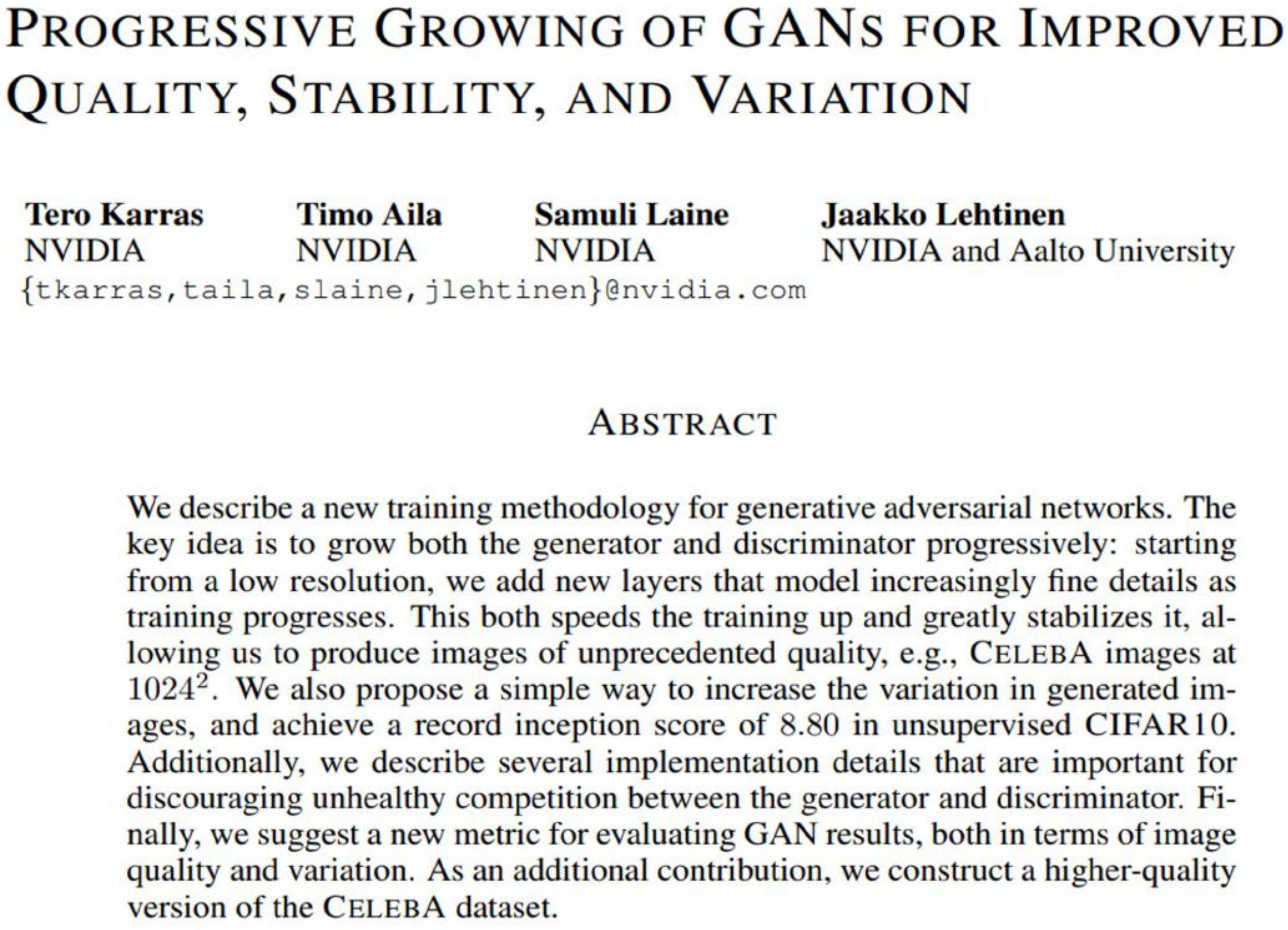
WGAN-GP: Summary

- Robustness to architectural choices
- Became a very popular GAN model - 2000+ citations, has been used in NVIDIA's Progressive GANs, StyleGAN, etc - biggest GAN successes
- Residual architecture widely adopted.
- Possible negative- slow wall clock time due to gradient penalty.
- Gradient penalty applied on a heuristic distribution of samples from current generator. Could be unstable when learning rates are high.

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - Improved Training of GANs (Salimans et al’16), Projected GAN (Sauer et al’21) WGAN, WGAN-GP, **Progressive GAN**, SN-GAN, SAGAN
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

Progressive growing of GANs



PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION

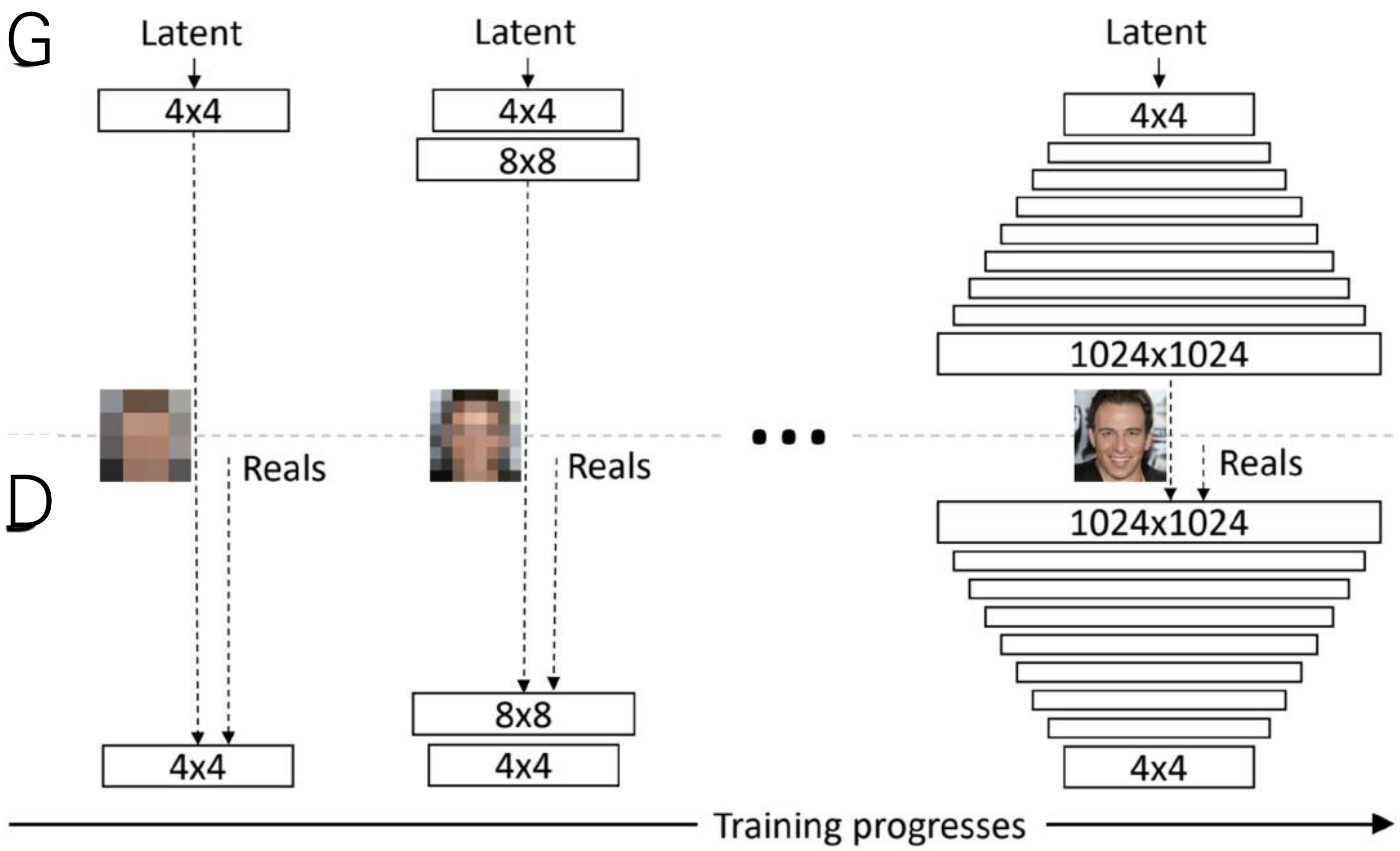
Tero Karras NVIDIA
Timo Aila NVIDIA
Samuli Laine NVIDIA
Jaakko Lehtinen NVIDIA and Aalto University
{tkarras,taila,slaine,jlehtinen}@nvidia.com

ABSTRACT

We describe a new training methodology for generative adversarial networks. The key idea is to grow both the generator and discriminator progressively: starting from a low resolution, we add new layers that model increasingly fine details as training progresses. This both speeds the training up and greatly stabilizes it, allowing us to produce images of unprecedented quality, e.g., CELEBA images at 1024^2. We also propose a simple way to increase the variation in generated images, and achieve a record inception score of 8.80 in unsupervised CIFAR10. Additionally, we describe several implementation details that are important for discouraging unhealthy competition between the generator and discriminator. Finally, we suggest a new metric for evaluating GAN results, both in terms of image quality and variation. As an additional contribution, we construct a higher-quality version of the CELEBA dataset.

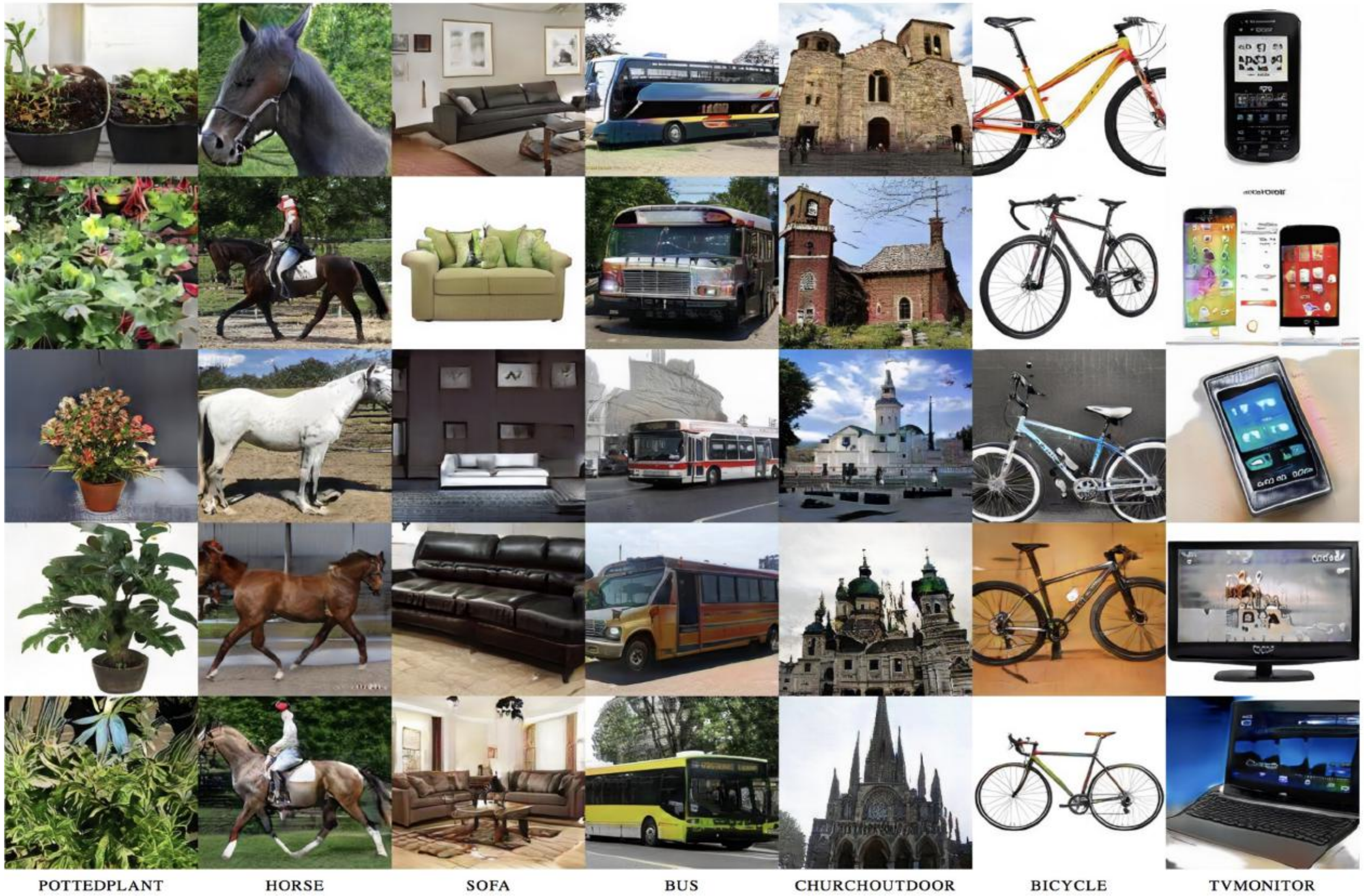
[Karras et al. 2017]

Progressive growing of GANs



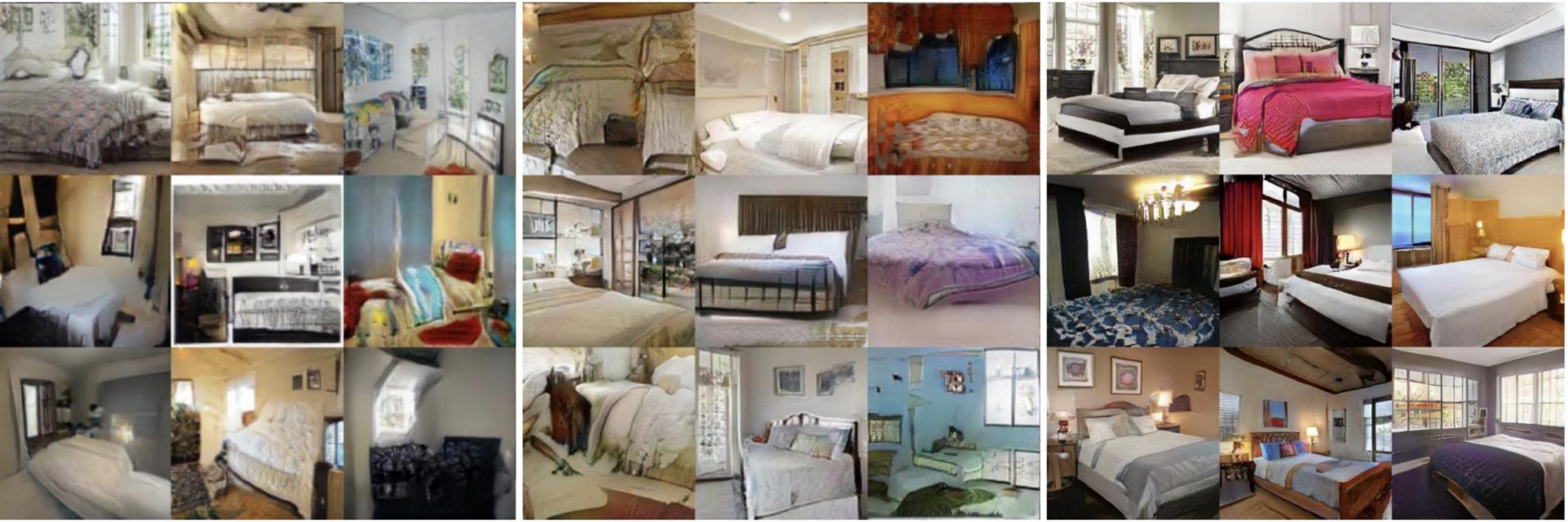
[Karras et al. 2017]

Progressive growing of GANs



[Karras et al. 2017]

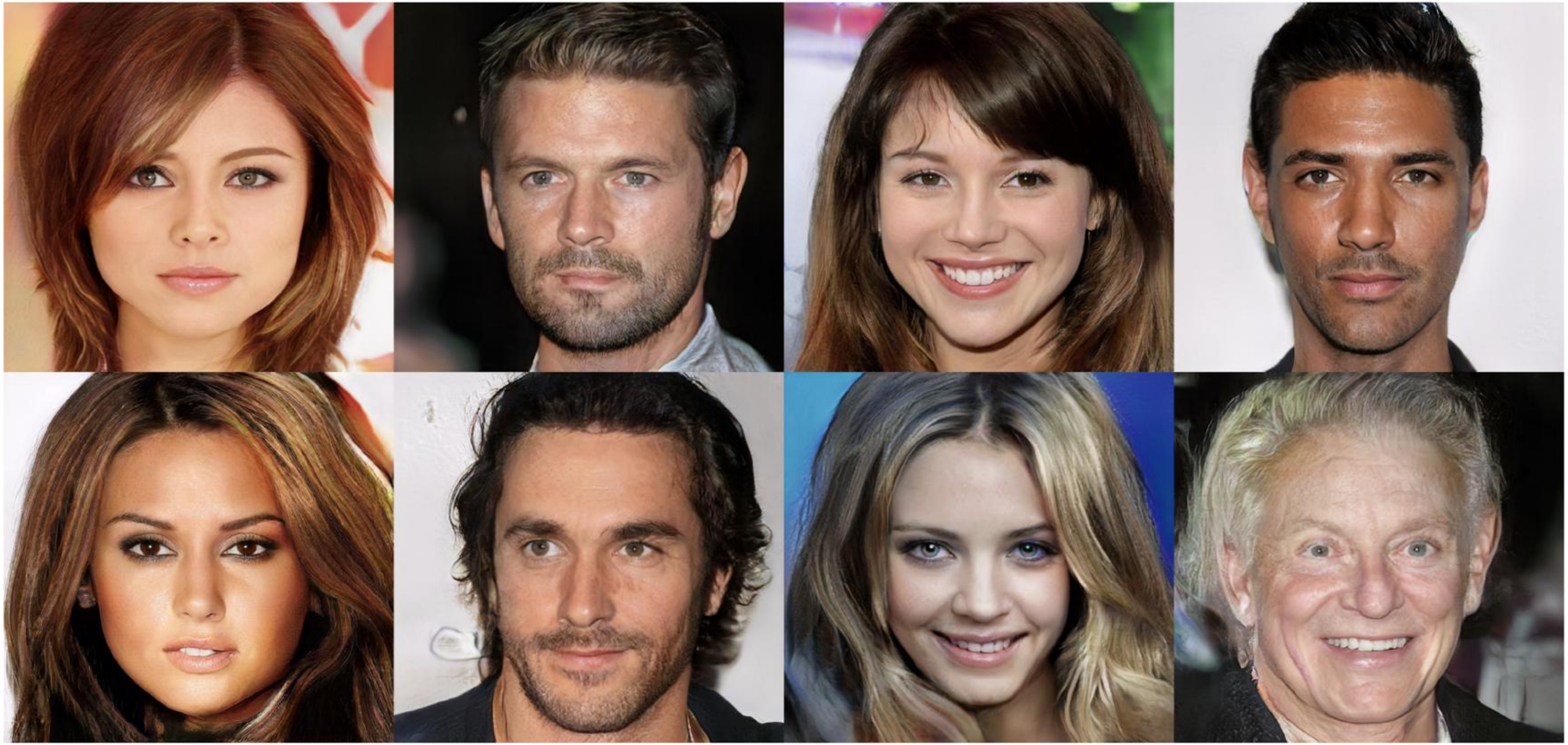
Progressive growing of GANs



Mao et al. (2016b) (128 × 128)　Gulrajani et al. (2017) (128 × 128)　Our (256 × 256)

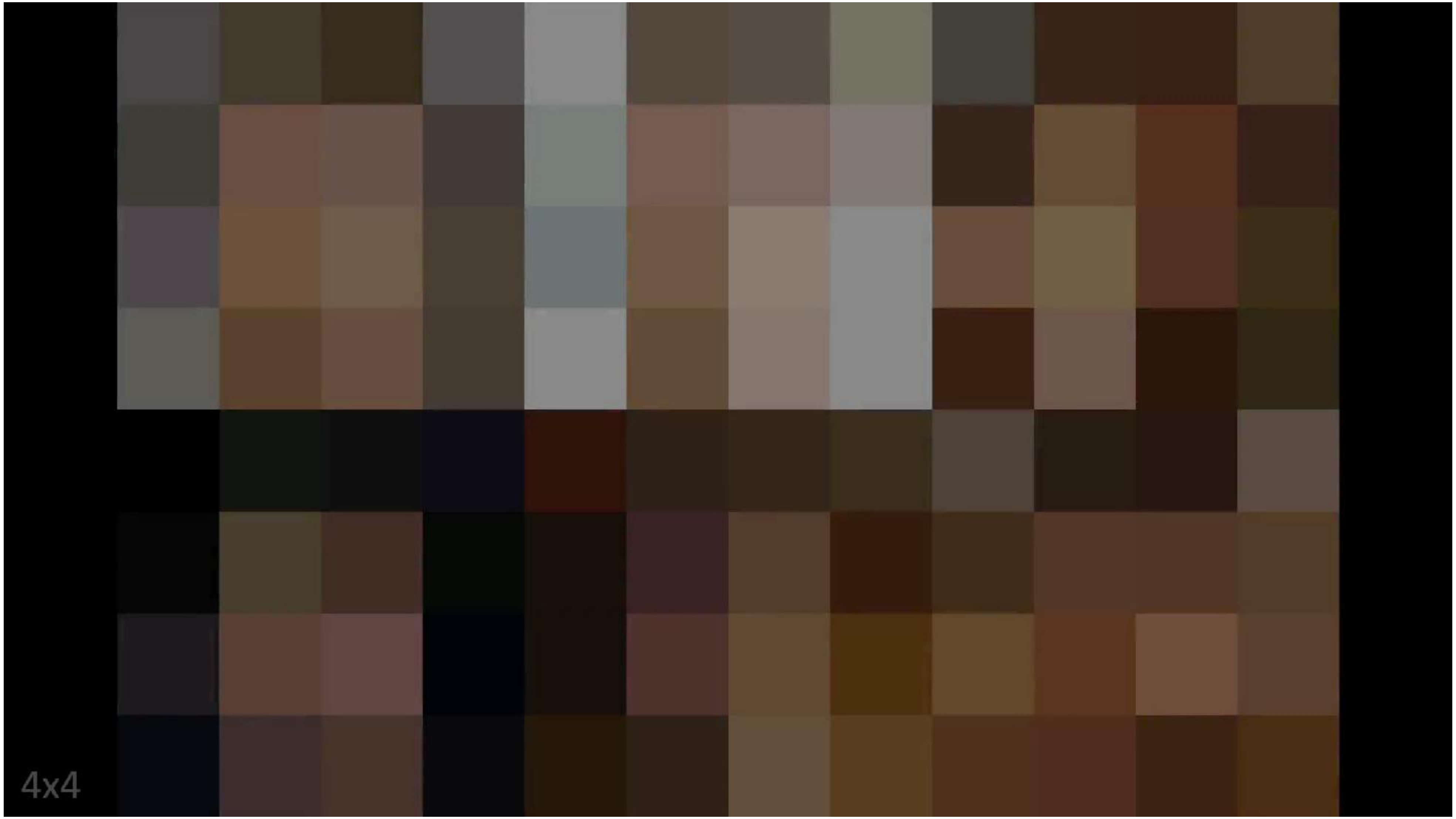
[Karras et al. 2017]

Progressive growing of GANs



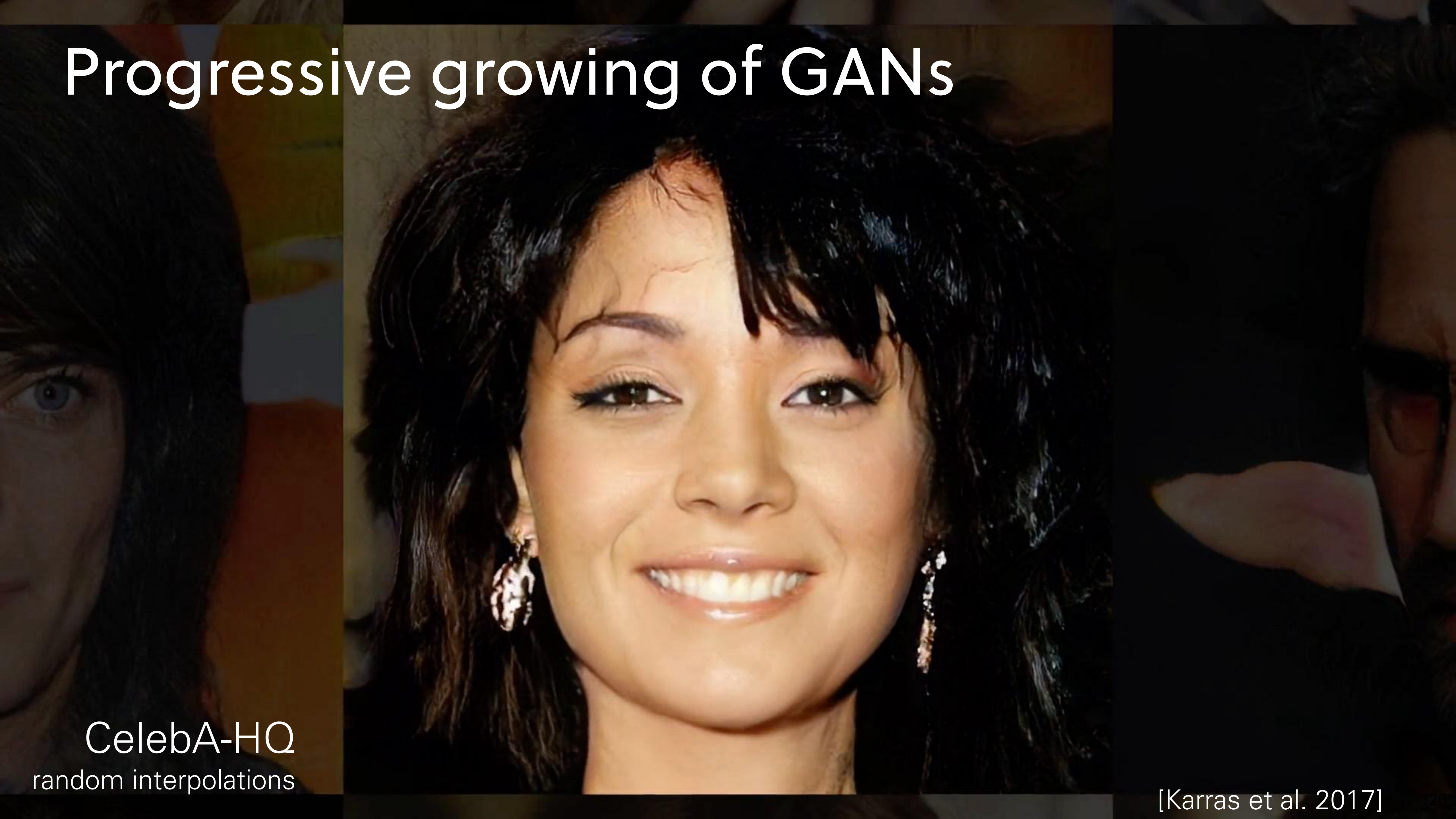
[Karras et al. 2017]

Progressive growing of GANs



[Karras et al. 2017]

Progressive growing of GANs



CelebA-HQ

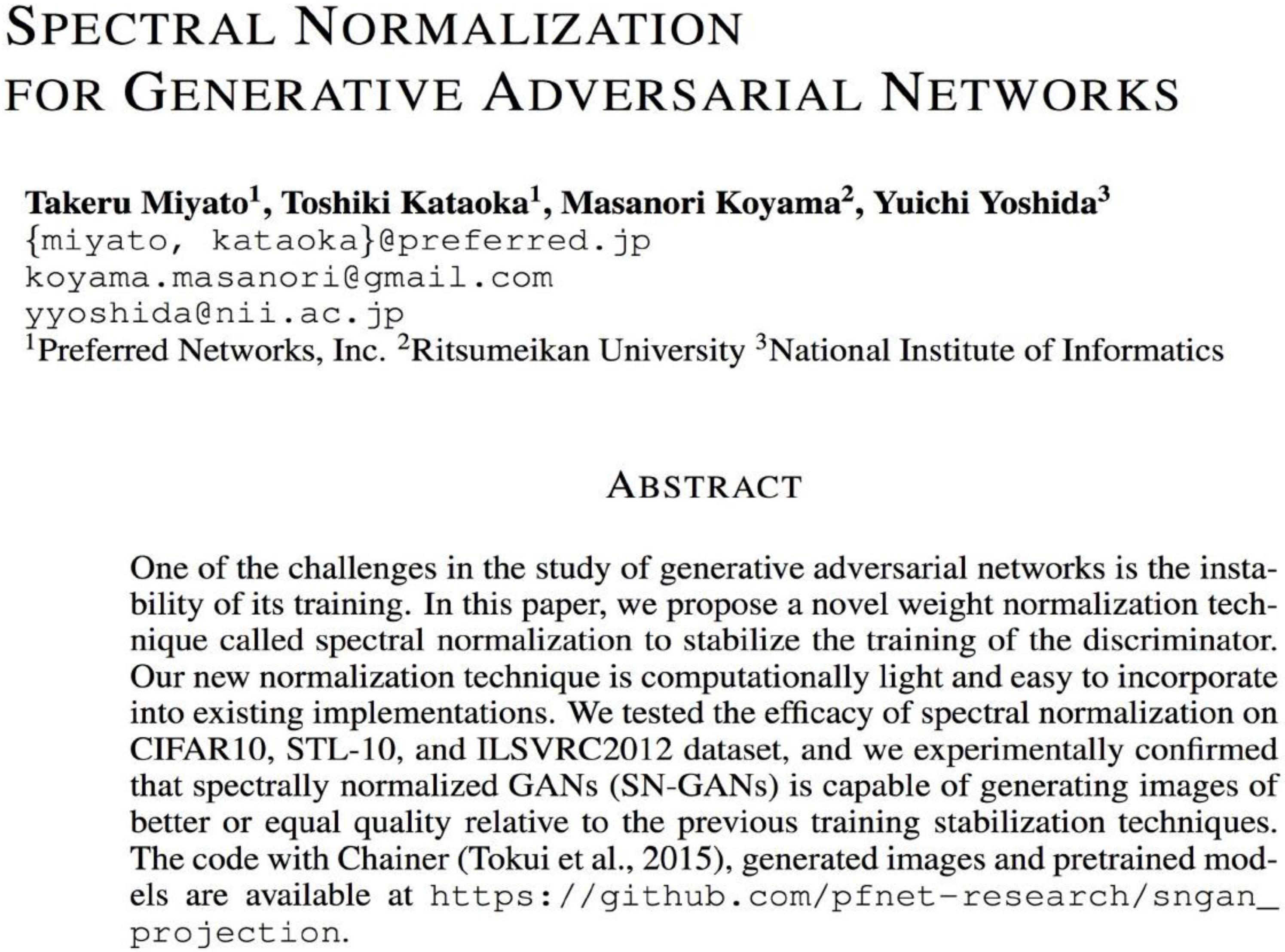
random interpolations

[Karras et al. 2017]

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - Improved Training of GANs (Salimans et al’16), Projected GAN (Sauer et al’21) WGAN, WGAN-GP, Progressive GAN, **SN-GAN**, SAGAN
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

Spectral Normalization GAN (SNGAN)



SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS

Takeru Miyato[1], Toshiki Kataoka[1], Masanori Koyama[2], Yuichi Yoshida[3]
{miyato, kataoka}@preferred.jp
koyama.masanori@gmail.com
yyoshida@nii.ac.jp
[1]Preferred Networks, Inc. [2]Ritsumeikan University [3]National Institute of Informatics

ABSTRACT

One of the challenges in the study of generative adversarial networks is the instability of its training. In this paper, we propose a novel weight normalization technique called spectral normalization to stabilize the training of the discriminator. Our new normalization technique is computationally light and easy to incorporate into existing implementations. We tested the efficacy of spectral normalization on CIFAR10, STL-10, and ILSVRC2012 dataset, and we experimentally confirmed that spectrally normalized GANs (SN-GANs) is capable of generating images of better or equal quality relative to the previous training stabilization techniques. The code with Chainer (Tokui et al., 2015), generated images and pretrained models are available at `https://github.com/pfnet-research/sngan_projection`.

[Miyato et al. 2017]

Spectral Normalization GAN (SNGAN)

(original) GAN formulation: $\min_G \max_D V(G, D)$

where $V(G, D) = \mathrm{E}_{\boldsymbol{x} \sim q_{\text{data}}}[\log D(\boldsymbol{x})] + \mathrm{E}_{\boldsymbol{x}' \sim p_G}[\log(1 - D(\boldsymbol{x}'))]$

WGAN formulation: $\min_G \left[\underset{\|f\|_{\text{Lip}} \leq K}{\arg\max} V(G, D) \right]$

where $\|f\|_{\text{Lip}} \leq K \Rightarrow \|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| / \|\boldsymbol{x} - \boldsymbol{x}'\| \leq K$

- Idea: Use spectral normalization to enforce the Lipschitz constraint

Spectral Normalization GAN (SNGAN)

- **Spectral Normalization strategy:** enforce the Lipschitz contraint by constraining the spectral norm of each layer of the neural network.

$$\text{spectral norm of the matrix } A\text{:} \quad \sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \max_{\|\boldsymbol{h}\|_2\leq 1} \|A\boldsymbol{h}\|_2$$

- Let g be a layer of a network: $g : \boldsymbol{h}_{in} \mapsto \boldsymbol{h}_{out}$

$$\text{for a linear layer } g(\boldsymbol{h}) = W\boldsymbol{h}\text{ :} \quad \|g\|_{\text{Lip}} = \sup_{\boldsymbol{h}} \sigma(\nabla g(\boldsymbol{h})) = \sup_{\boldsymbol{h}} \sigma(W) = \sigma(W)$$

- For the network *f*, we assume the Lipschitz norm of the activation function (*a*) equals 1 (typically ok) and use the inequality:

$$\|g_1 \circ g_2\|_{\text{Lip}} \leq \|g_1\|_{\text{Lip}} \cdot \|g_2\|_{\text{Lip}}$$

Spectral Normalization GAN (SNGAN)

- The Lipschitz norm for the network is:

activation function for layer L

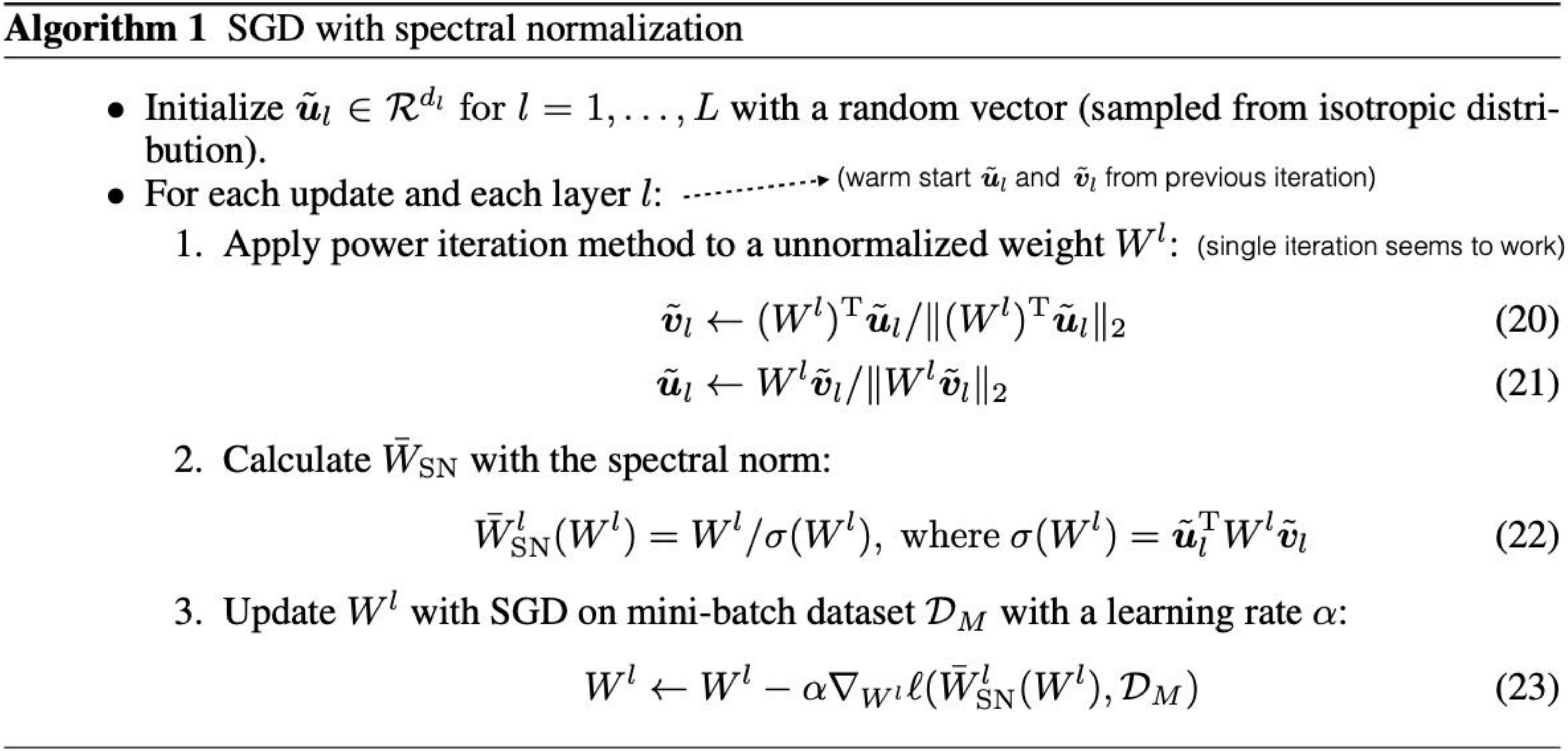
$$\|f\|_{\mathrm{Lip}} \leq \|(\boldsymbol{h}_L \mapsto W^{L+1}\boldsymbol{h}_L)\|_{\mathrm{Lip}} \cdot \|a_L\|_{\mathrm{Lip}} \cdot \|(\boldsymbol{h}_{L-1} \mapsto W^{L}\boldsymbol{h}_{L-1})\|_{\mathrm{Lip}} \cdots \|a_1\|_{\mathrm{Lip}} \cdot \|(\boldsymbol{h}_0 \mapsto W^{1}\boldsymbol{h}_0)\|_{\mathrm{Lip}} = \prod_{l=1}^{L+1} \|(\boldsymbol{h}_{l-1} \mapsto W^{l}\boldsymbol{h}_{l-1})\|_{\mathrm{Lip}} = \prod_{l=1}^{L+1} \sigma(W^l)$$

- Spectral Normalize the weights at each layer: $\bar{W}_{\mathrm{SN}}(W) := W/\sigma(W)$

where $\sigma(W)$ is efficiently approximated using the power method.

(as described on the next slide)

Spectral Normalization GAN (SNGAN)



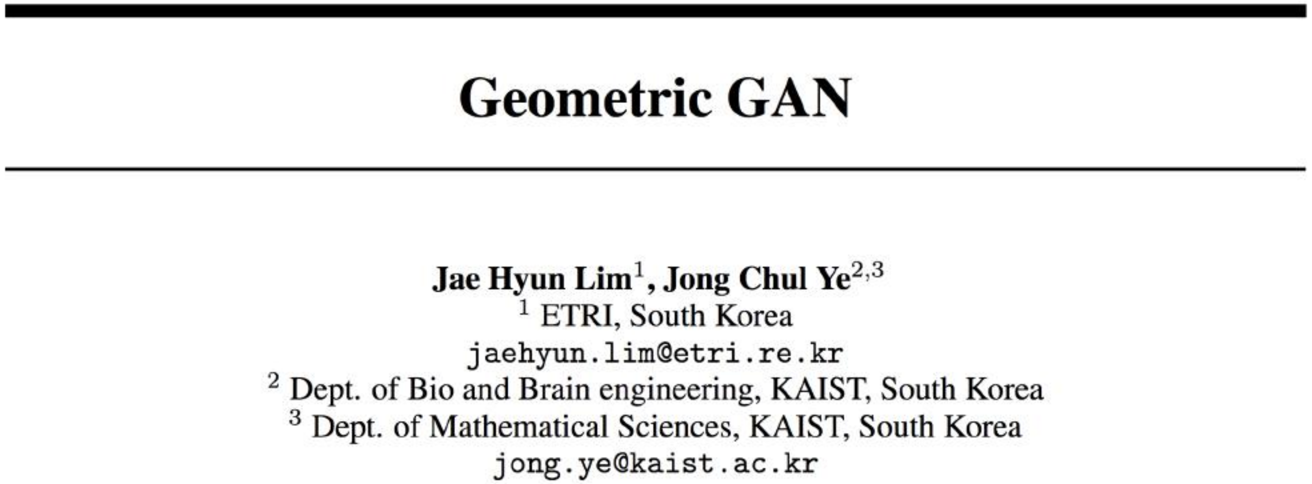
Algorithm 1 SGD with spectral normalization

- Initialize $\tilde{\boldsymbol{u}}_l \in \mathcal{R}^{d_l}$ for $l = 1, \ldots, L$ with a random vector (sampled from isotropic distribution).
- For each update and each layer l: ---→ (warm start $\tilde{\boldsymbol{u}}_l$ and $\tilde{\boldsymbol{v}}_l$ from previous iteration)
  1. Apply power iteration method to a unnormalized weight W^l: (single iteration seems to work)
  $$\tilde{\boldsymbol{v}}_l \leftarrow (W^l)^{\mathrm{T}} \tilde{\boldsymbol{u}}_l / \|(W^l)^{\mathrm{T}} \tilde{\boldsymbol{u}}_l\|_2 \tag{20}$$
  $$\tilde{\boldsymbol{u}}_l \leftarrow W^l \tilde{\boldsymbol{v}}_l / \|W^l \tilde{\boldsymbol{v}}_l\|_2 \tag{21}$$
  2. Calculate $\bar{W}_{\mathrm{SN}}$ with the spectral norm:
  $$\bar{W}^l_{\mathrm{SN}}(W^l) = W^l / \sigma(W^l), \text{ where } \sigma(W^l) = \tilde{\boldsymbol{u}}_l^{\mathrm{T}} W^l \tilde{\boldsymbol{v}}_l \tag{22}$$
  3. Update W^l with SGD on mini-batch dataset $\mathcal{D}_M$ with a learning rate α:
  $$W^l \leftarrow W^l - \alpha \nabla_{W^l} \ell(\bar{W}^l_{\mathrm{SN}}(W^l), \mathcal{D}_M) \tag{23}$$

Spectral Normalization GAN (SNGAN)

$$V_D(\hat{G}, D) = \underset{\boldsymbol{x}\sim q_{\mathrm{data}}(\boldsymbol{x})}{\mathrm{E}}\left[\min\left(0, -1 + D(\boldsymbol{x})\right)\right] + \underset{\boldsymbol{z}\sim p(\boldsymbol{z})}{\mathrm{E}}\left[\min\left(0, -1 - D\left(\hat{G}(\boldsymbol{z})\right)\right)\right]$$

$$V_G(G, \hat{D}) = - \underset{\boldsymbol{z}\sim p(\boldsymbol{z})}{\mathrm{E}}\left[\hat{D}\left(G(\boldsymbol{z})\right)\right],$$

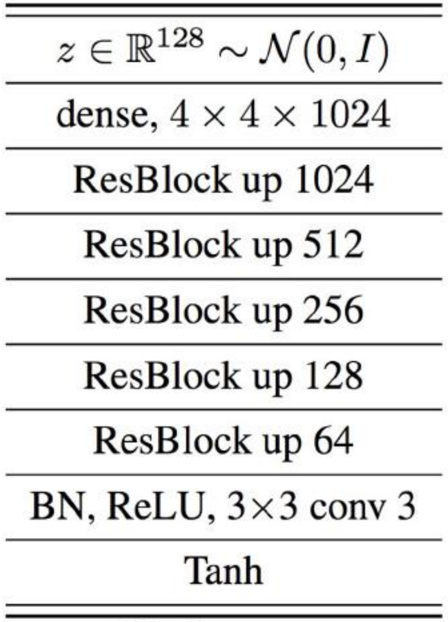


Geometric GAN

Jae Hyun Lim[1]**, Jong Chul Ye**[2,3]
[1] ETRI, South Korea
jaehyun.lim@etri.re.kr
[2] Dept. of Bio and Brain engineering, KAIST, South Korea
[3] Dept. of Mathematical Sciences, KAIST, South Korea
jong.ye@kaist.ac.kr

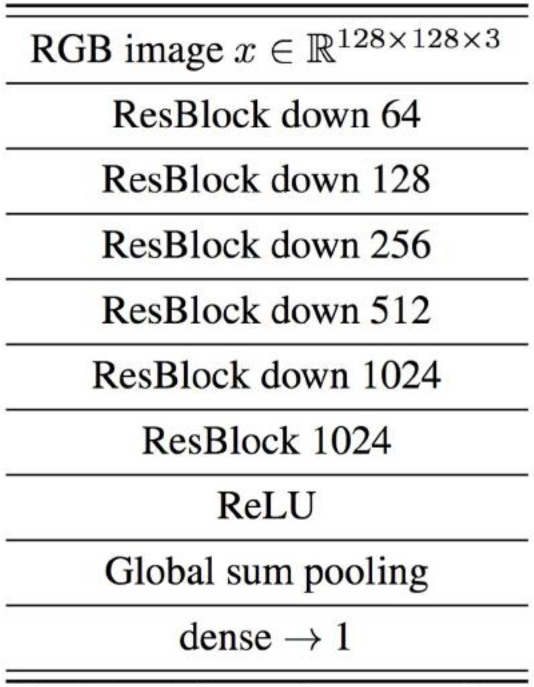
[Miyato et al. 2017]

Spectral Normalization GAN (SNGAN)



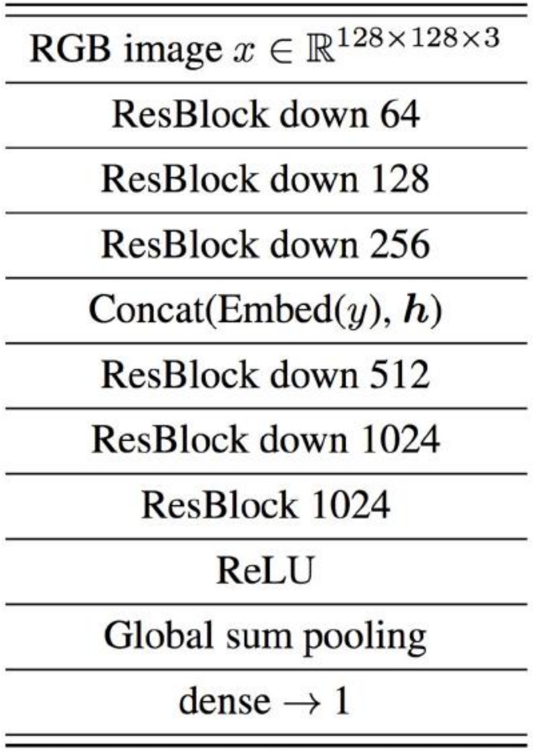
$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
dense, $4 \times 4 \times 1024$
ResBlock up 1024
ResBlock up 512
ResBlock up 256
ResBlock up 128
ResBlock up 64
BN, ReLU, 3×3 conv 3
Tanh

(a) Generator



RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512
ResBlock down 1024
ResBlock 1024
ReLU
Global sum pooling
dense → 1

(b) Discriminator for unconditional GANs.



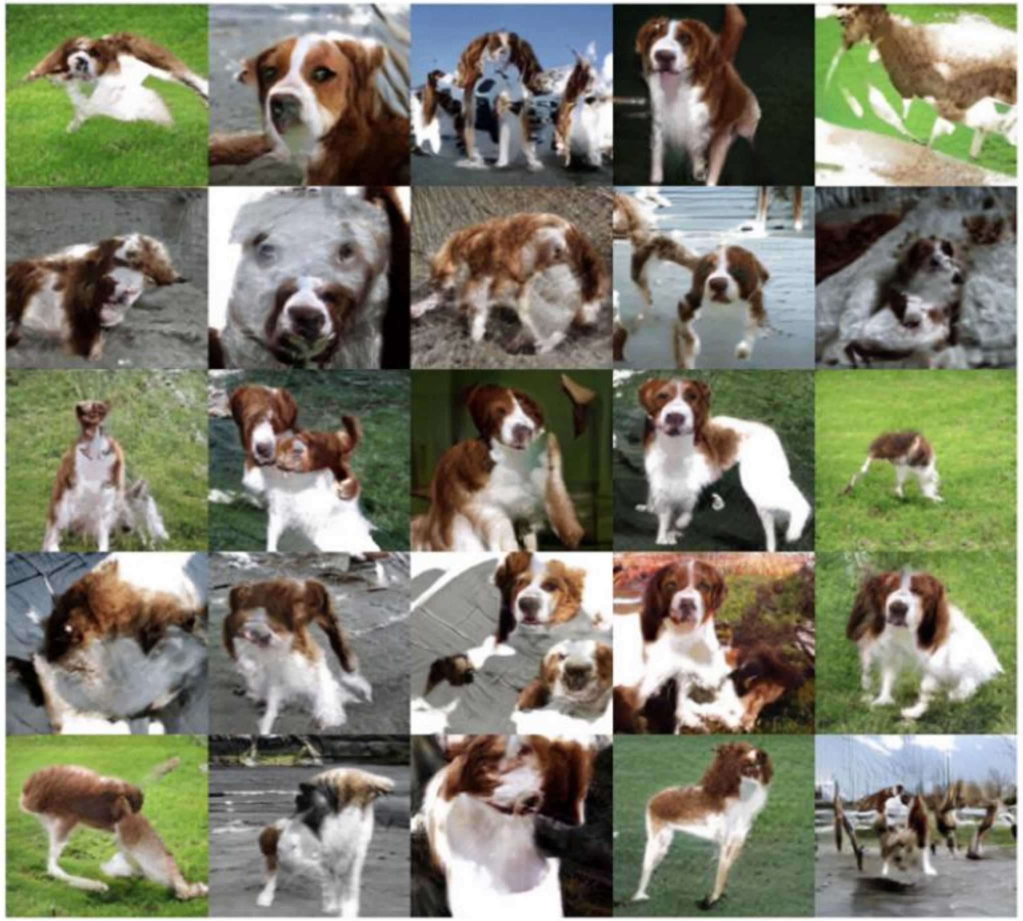
RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
Concat(Embed(y), $\boldsymbol{h}$)
ResBlock down 512
ResBlock down 1024
ResBlock 1024
ReLU
Global sum pooling
dense → 1

(c) Discriminator for conditional GANs. For computational ease, we embedded the integer label $y \in \{0, \ldots, 1000\}$ into 128 dimension before concatenating the vector to the output of the intermediate layer.

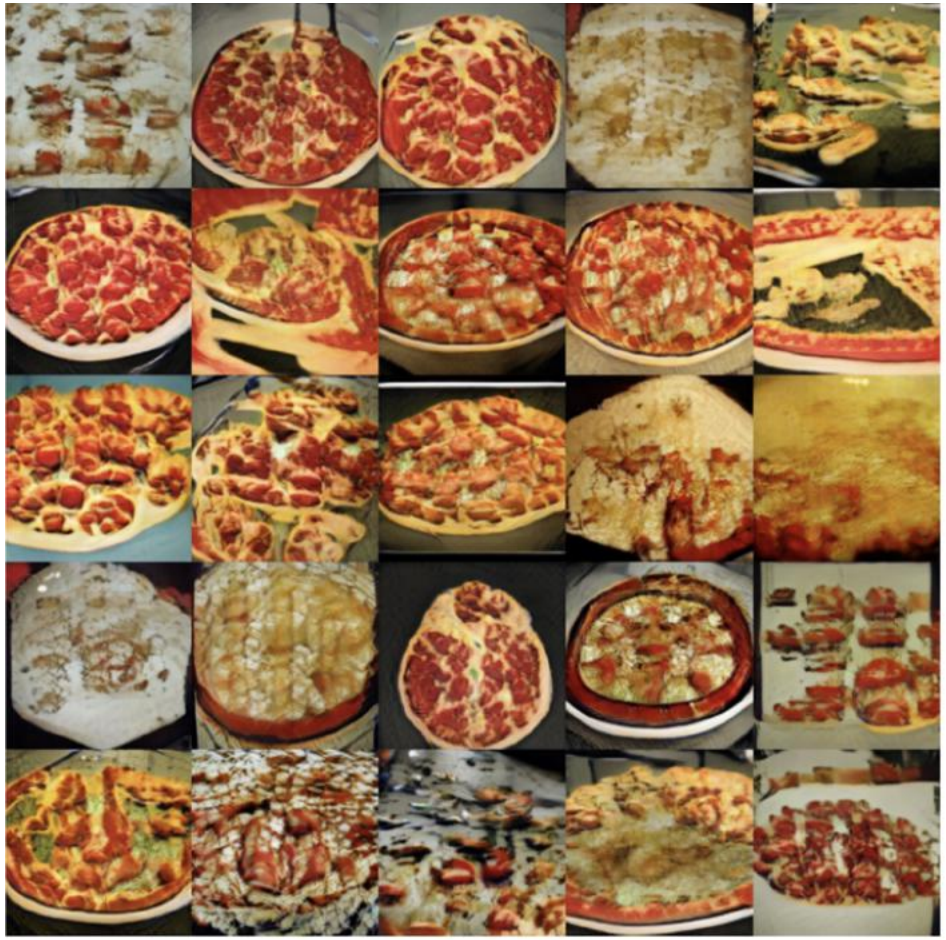
[Miyato et al. 2017]

Spectral Normalization GAN (SNGAN)

Welsh springer spaniel



Pizza



[Miyato et al. 2017]

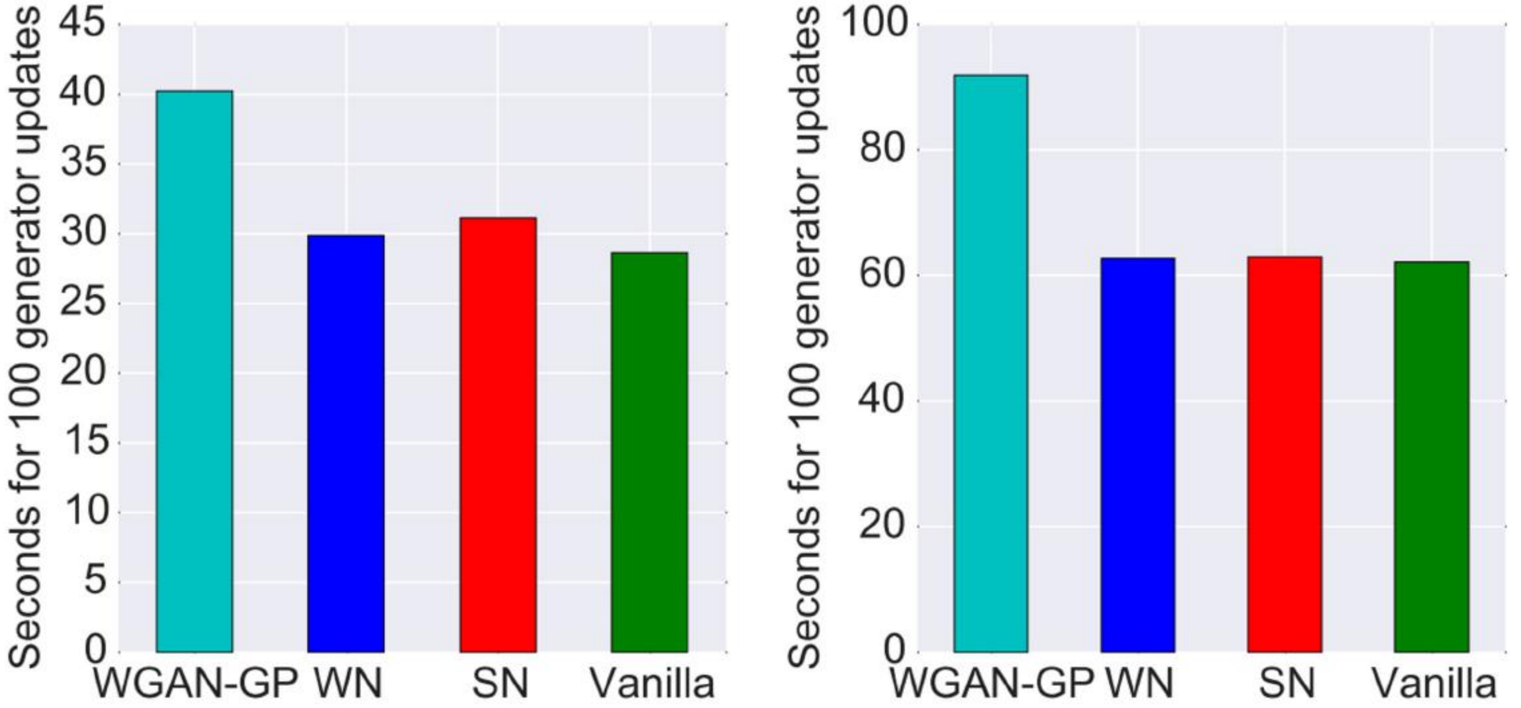
SNGAN: Summary

- High quality class conditional samples at Imagenet scale

- First GAN to work on full Imagenet (million image dataset)

- Computational benefits over WGAN-GP (single power iteration and no need of a backward pass)

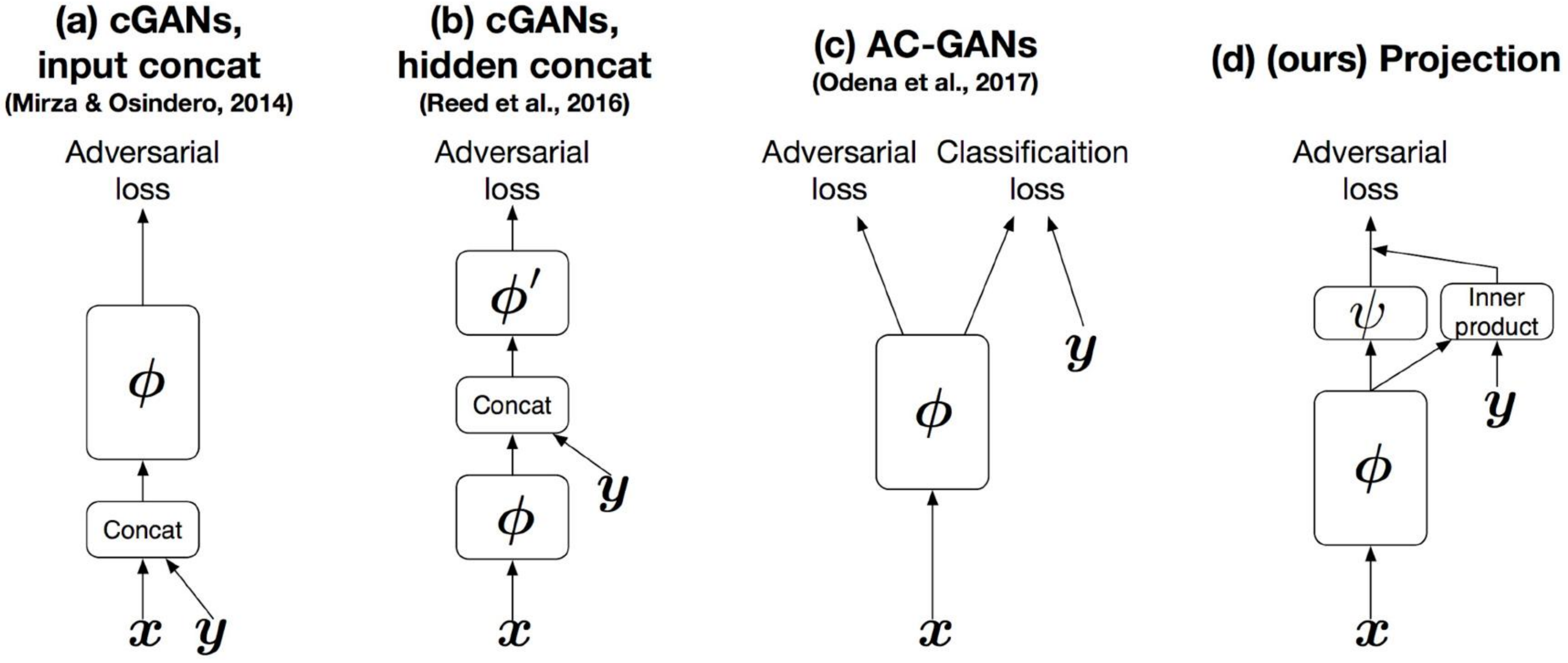
SNGAN: Computational Benefits



(a) CIFAR-10 (image size:32 × 32 × 3)

(b) STL-10 (images size:48 × 48 × 3)

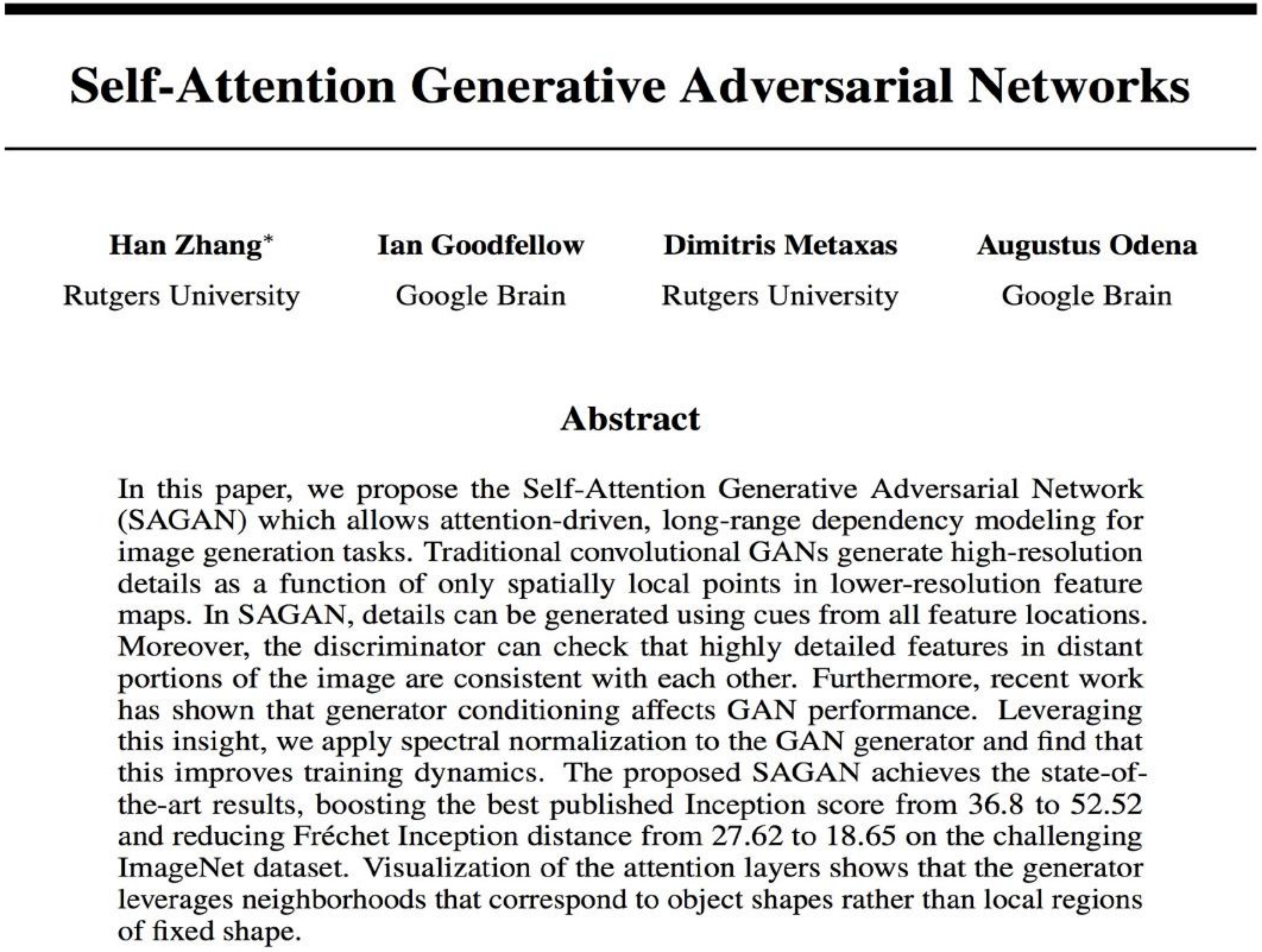
Projection Discriminator



Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - Improved Training of GANs (Salimans et al'16), Projected GAN (Sauer et al'21) WGAN, WGAN-GP, Progressive GAN, SN-GAN, **SAGAN**
  - BigGAN, BigGAN-Deep, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

Self Attention GAN (SAGAN)



Self-Attention Generative Adversarial Networks

Han Zhang[*] Rutgers University

Ian Goodfellow Google Brain

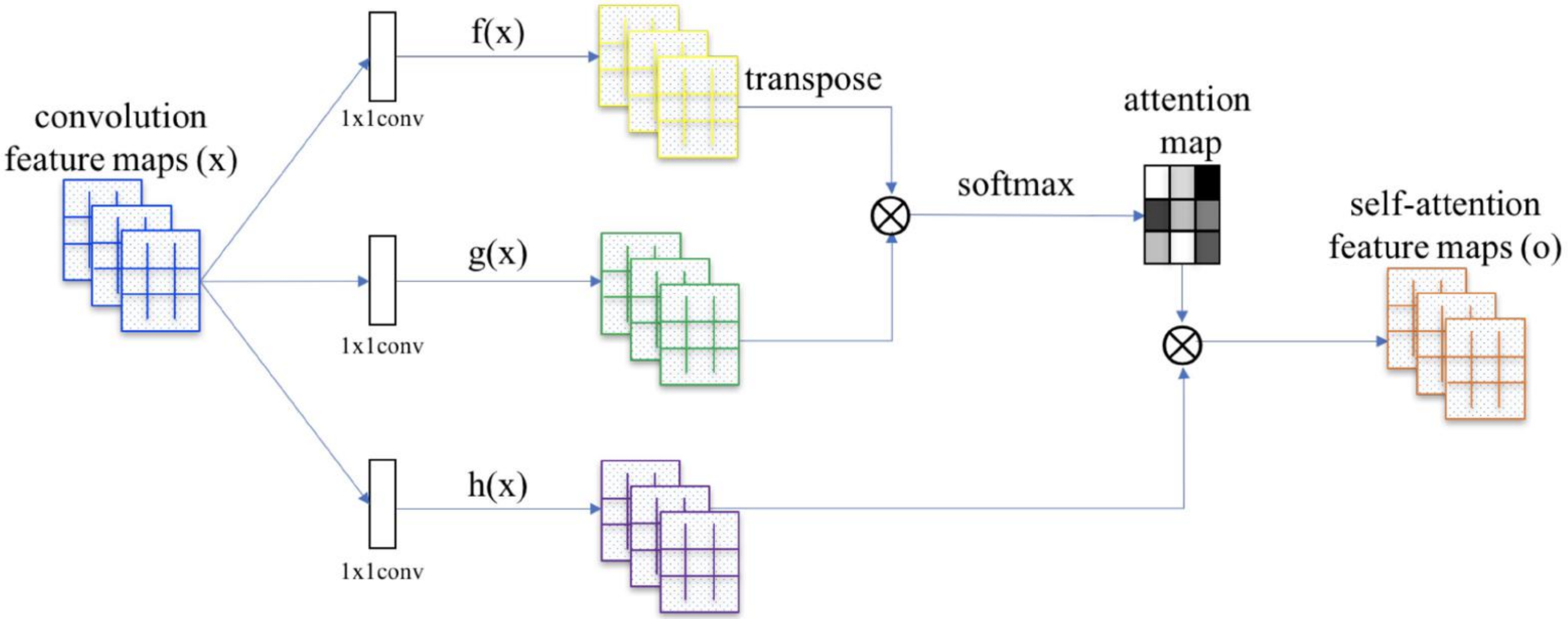
Dimitris Metaxas Rutgers University

Augustus Odena Google Brain

Abstract

In this paper, we propose the Self-Attention Generative Adversarial Network (SAGAN) which allows attention-driven, long-range dependency modeling for image generation tasks. Traditional convolutional GANs generate high-resolution details as a function of only spatially local points in lower-resolution feature maps. In SAGAN, details can be generated using cues from all feature locations. Moreover, the discriminator can check that highly detailed features in distant portions of the image are consistent with each other. Furthermore, recent work has shown that generator conditioning affects GAN performance. Leveraging this insight, we apply spectral normalization to the GAN generator and find that this improves training dynamics. The proposed SAGAN achieves the state-of-the-art results, boosting the best published Inception score from 36.8 to 52.52 and reducing Fréchet Inception distance from 27.62 to 18.65 on the challenging ImageNet dataset. Visualization of the attention layers shows that the generator leverages neighborhoods that correspond to object shapes rather than local regions of fixed shape.

Self Attention GAN (SAGAN)



[Zhang et al. 2018]

Self Attention GAN (SAGAN)

$$\boldsymbol{f(x)} = \boldsymbol{W_f x}, \ \boldsymbol{g(x)} = \boldsymbol{W_g x}$$

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp(s_{ij})}$$

$$s_{ij} = \boldsymbol{f(x_i)}^T \boldsymbol{g(x_j)}$$

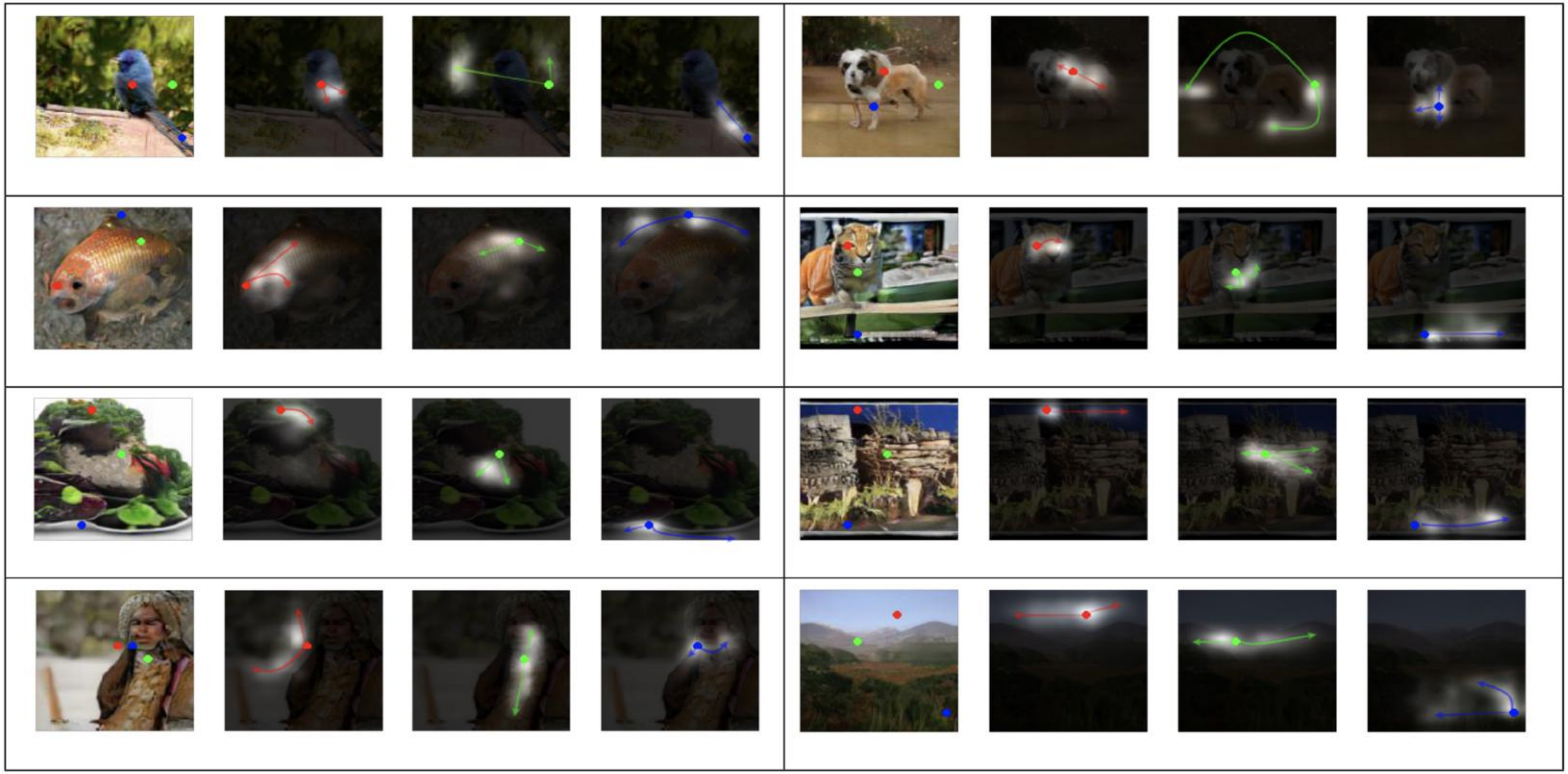
$$\boldsymbol{y_i} = \gamma \boldsymbol{o_i} + \boldsymbol{x_i}$$

[Zhang et al. 2018]

Self Attention GAN (SAGAN)

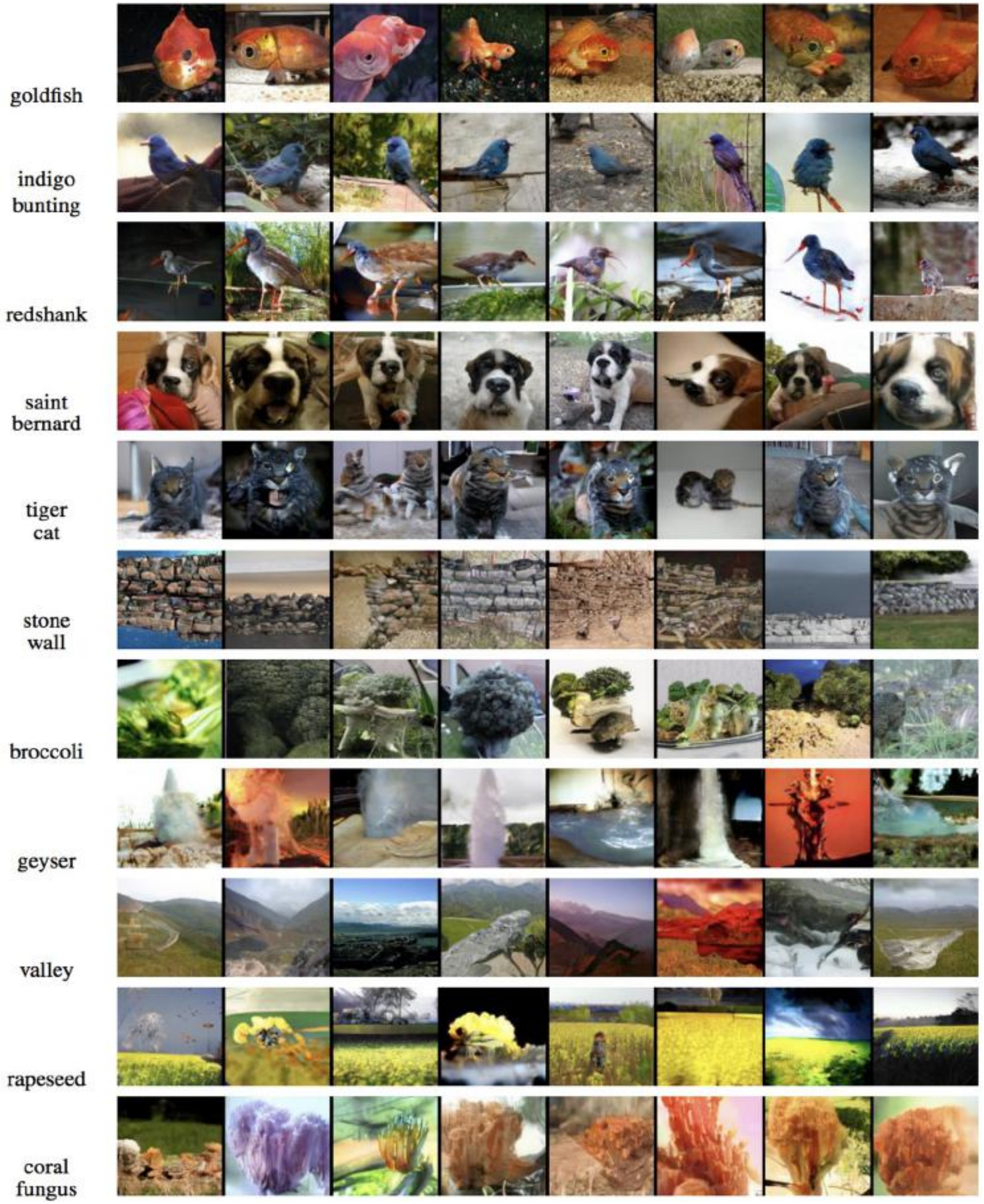
- Applies spectral normalization to both the generator and discriminator weight matrices
  - This is counter-intuitive to popular belief that you only have to mathematically condition the discriminator
- Uses self-attention in both the generator and discriminator
- Hinge Loss
- First GAN to produce “good” unconditional full ImageNet samples
- Conditional models
  - Conditional BN for G, Projection Discriminator for D

Self Attention GAN (SAGAN)



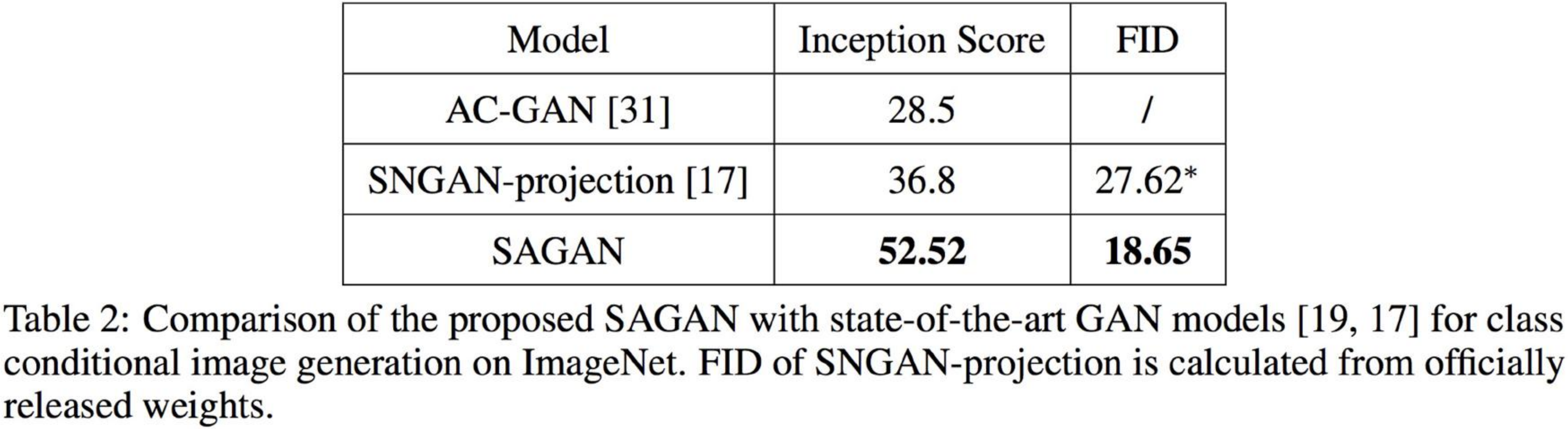
[Zhang et al. 2018]

Self Attention GAN (SAGAN)



[Zhang et al. 2018]

Self Attention GAN (SAGAN)



Model	Inception Score	FID
AC-GAN [31]	28.5	/
SNGAN-projection [17]	36.8	27.62*
SAGAN	**52.52**	**18.65**

Table 2: Comparison of the proposed SAGAN with state-of-the-art GAN models [19, 17] for class conditional image generation on ImageNet. FID of SNGAN-projection is calculated from officially released weights.

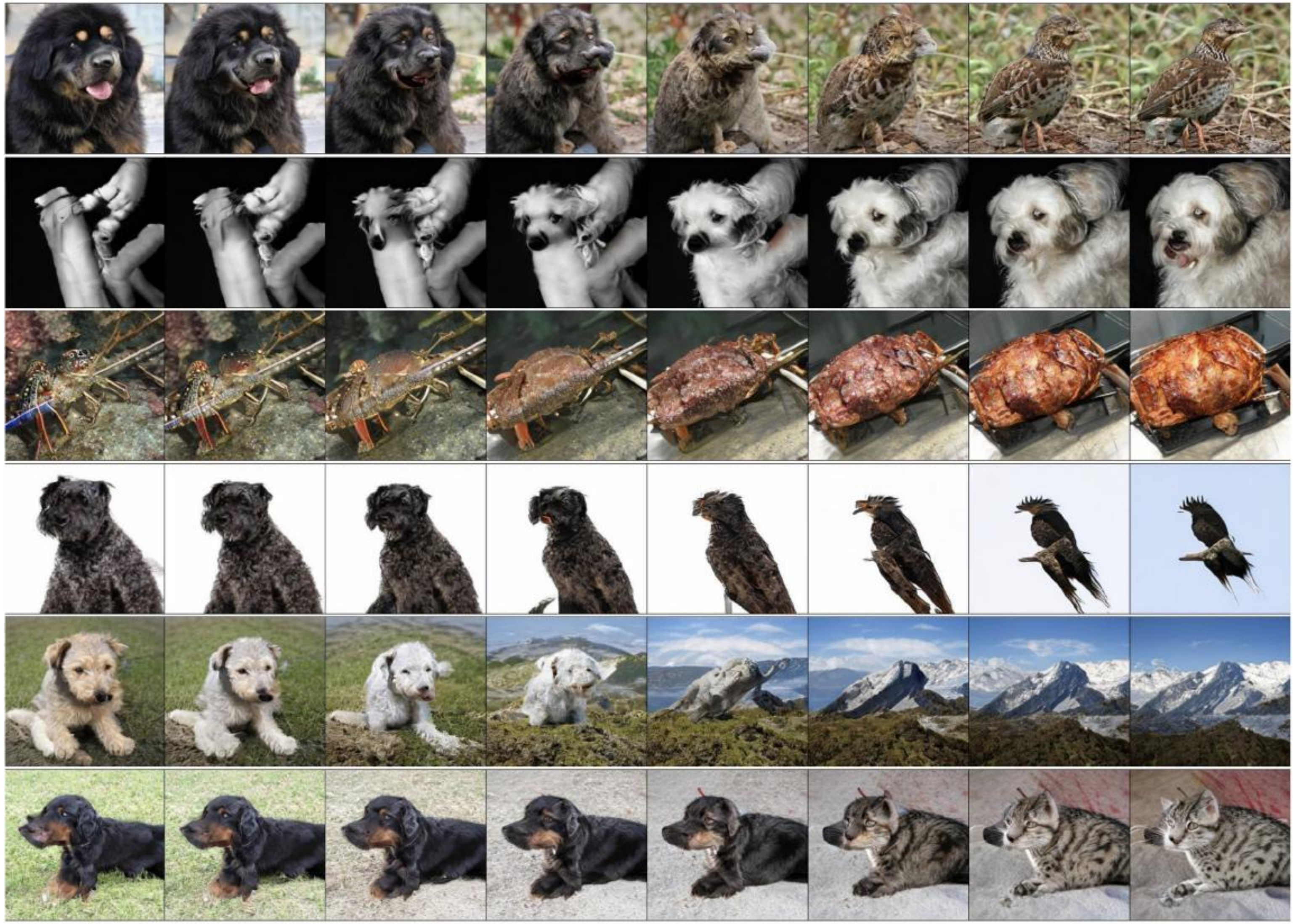
[Zhang et al. 2018]

Lecture overview

- Motivation and Definition of Implicit Models
- Original GAN (Goodfellow et al, 2014)
- Evaluation: Parzen, Inception, Frechet
- Theory of GANs
- **GAN Progression**
  - DC GAN (Radford et al, 2016)
  - Improved Training of GANs (Salimans et al’16), Projected GAN (Sauer et al’21) WGAN, WGAN-GP, Progressive GAN, SN-GAN, SAGAN
  - **BigGAN, BigGAN-Deep**, StyleGAN, StyleGAN2, StyleGAN3, StyleGAN-XL, Self-Distilled StyleGAN, VIB-GAN, VQ-GAN
- Conditional GANs, Cycle-Consistent Adversarial Networks
- GANs and Representations
- Applications

BigGAN

BigGAN



BigGAN

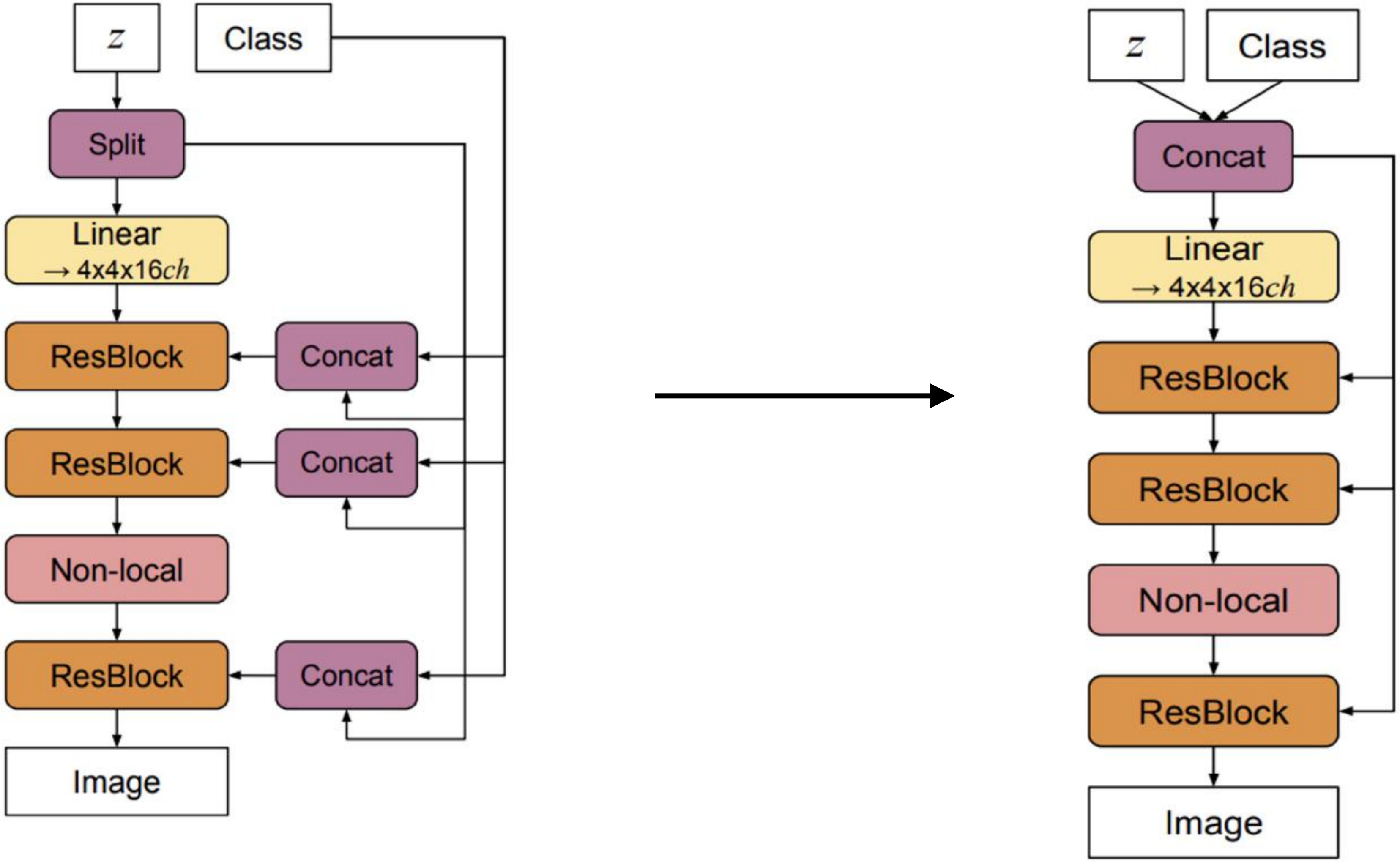
$$R_\beta(W) = \beta \|W^\top W - I\|_\mathrm{F}^2$$

↓

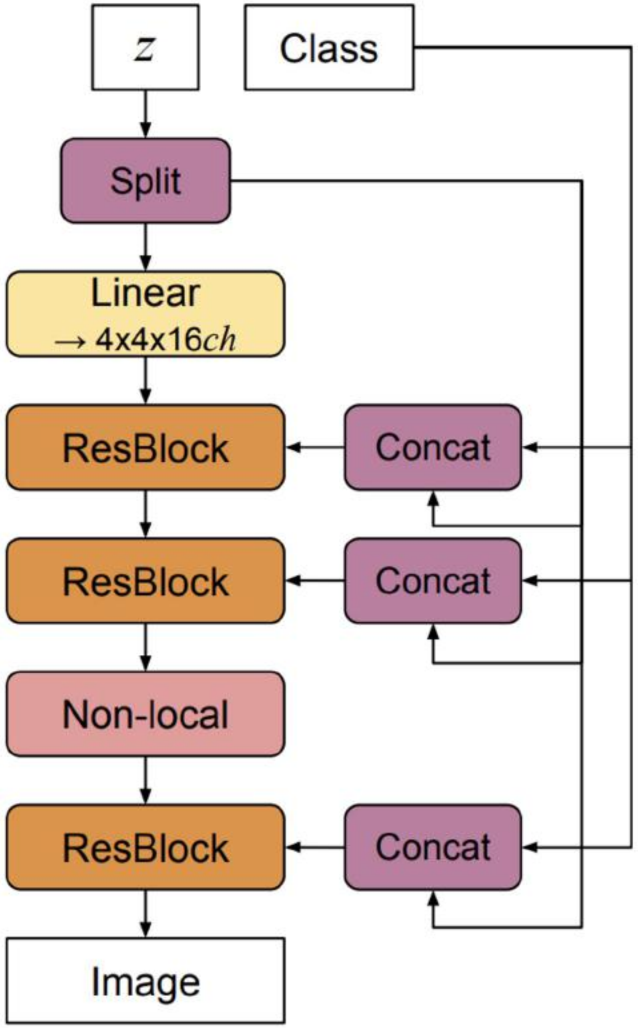
$$R_\beta(W) = \beta \|W^\top W \odot (\mathbf{1} - I)\|_\mathrm{F}^2,$$

Orthogonal Regularization

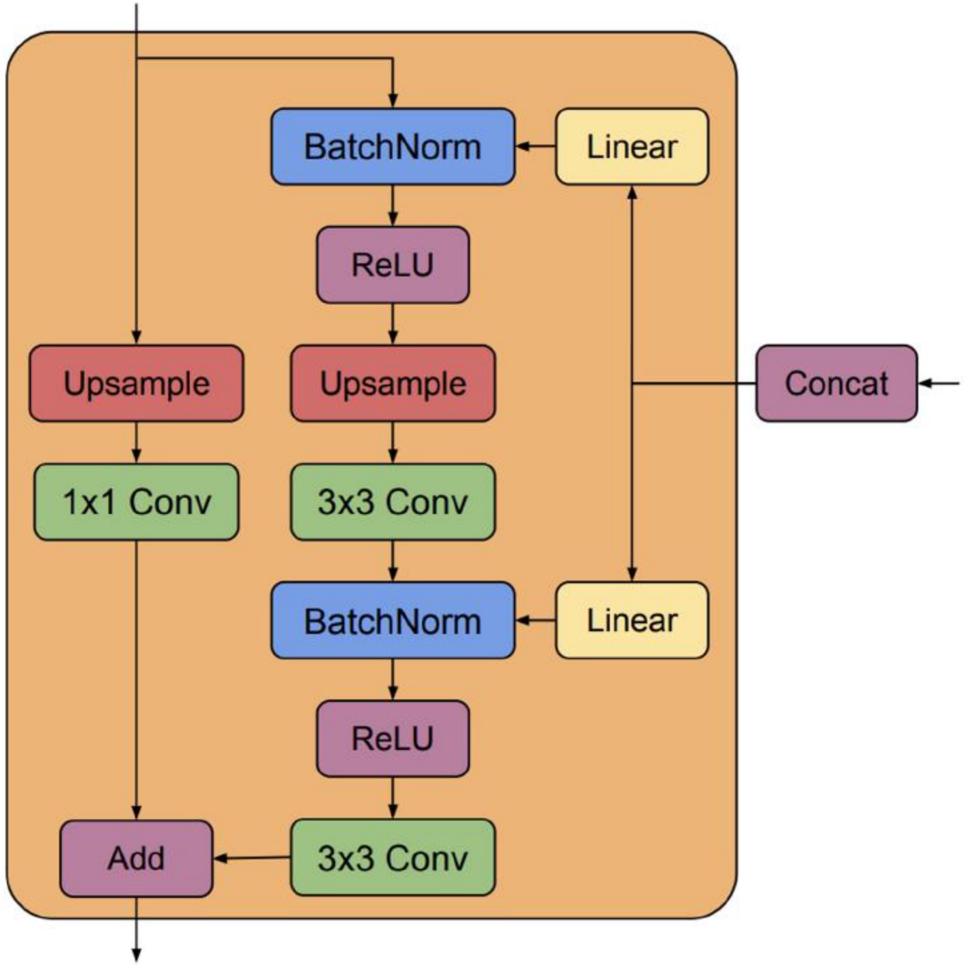
BigGAN and BigGAN-deep



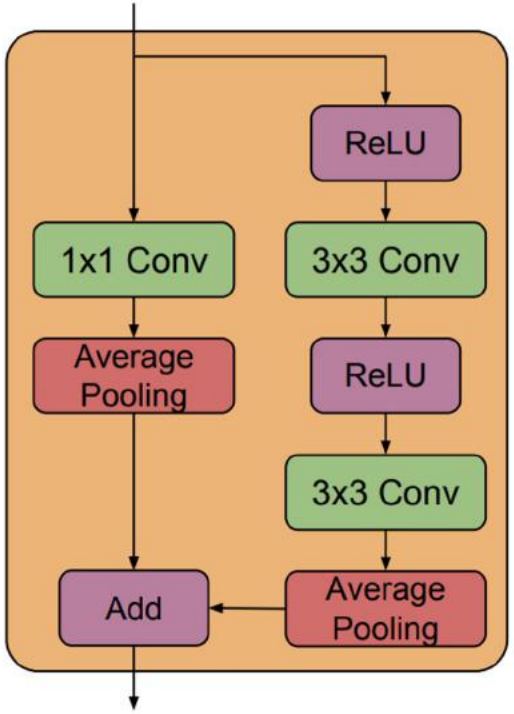
BigGAN



z
Class
Split
Linear
→ 4x4x16ch
ResBlock
Concat
ResBlock
Concat
Non-local
ResBlock
Concat
Image

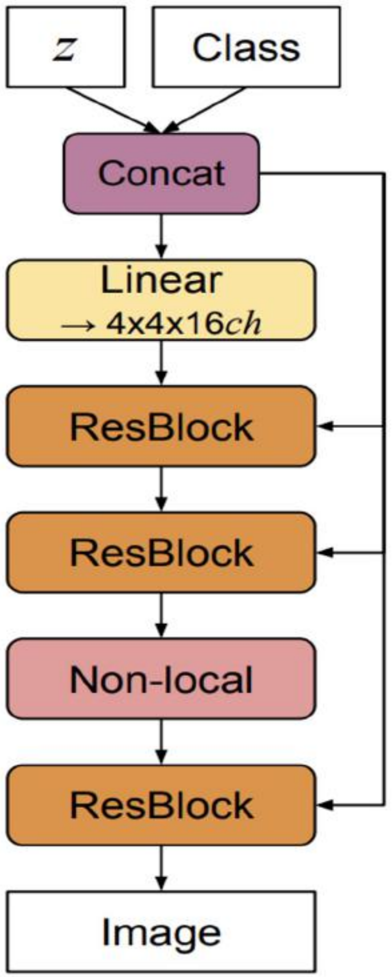


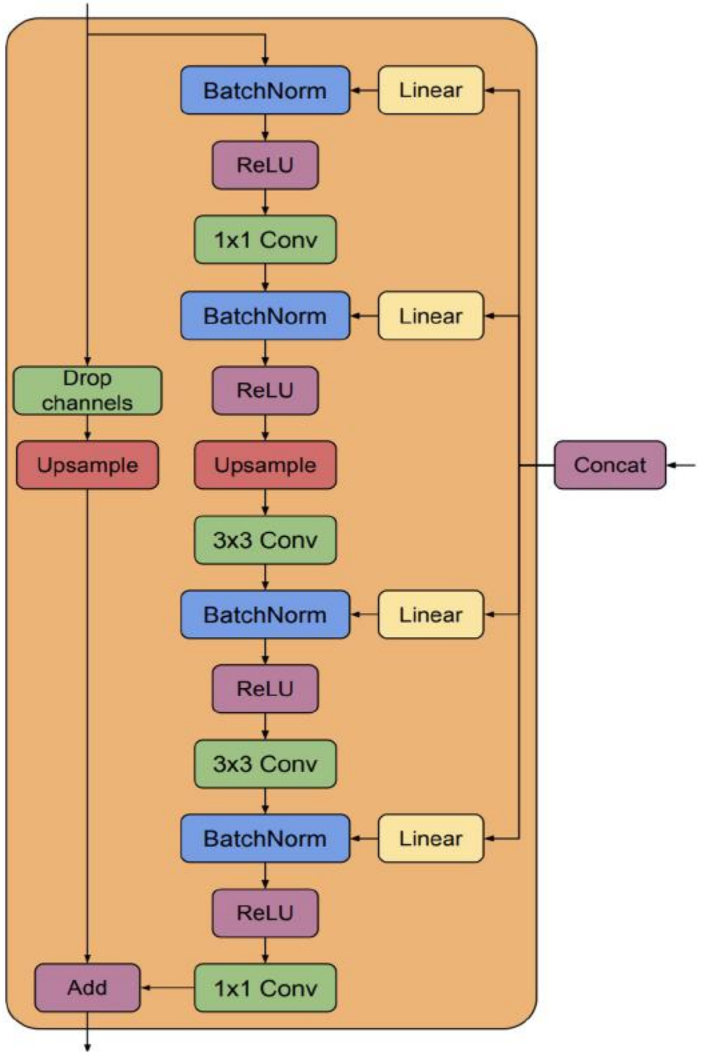
BatchNorm
Linear
ReLU
Upsample
Upsample
Concat
1x1 Conv
3x3 Conv
BatchNorm
Linear
ReLU
Add
3x3 Conv

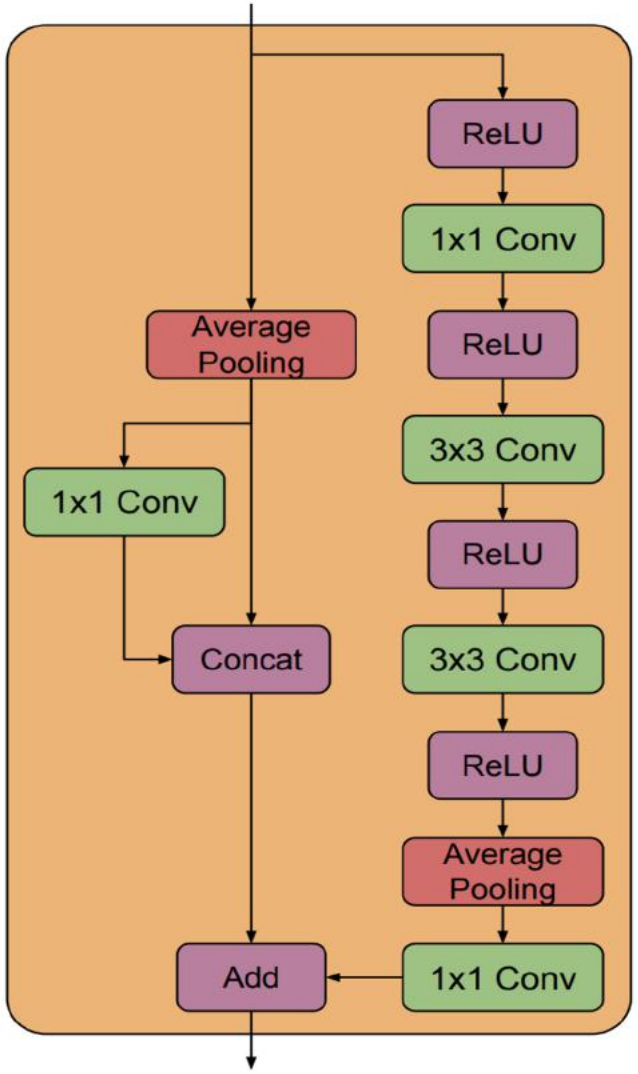


ReLU
1x1 Conv
3x3 Conv
Average Pooling
ReLU
3x3 Conv
Add
Average Pooling

BigGAN-deep


z
Class
Concat
Linear
→ 4x4x16ch
ResBlock
ResBlock
Non-local
ResBlock
Image


BatchNorm
Linear
ReLU
1x1 Conv
BatchNorm
Linear
Drop channels
ReLU
Upsample
Upsample
Concat
3x3 Conv
BatchNorm
Linear
ReLU
3x3 Conv
BatchNorm
Linear
ReLU
Add
1x1 Conv


ReLU
1x1 Conv
Average Pooling
ReLU
3x3 Conv
1x1 Conv
ReLU
Concat
3x3 Conv
ReLU
Average Pooling
Add
1x1 Conv

BigGAN

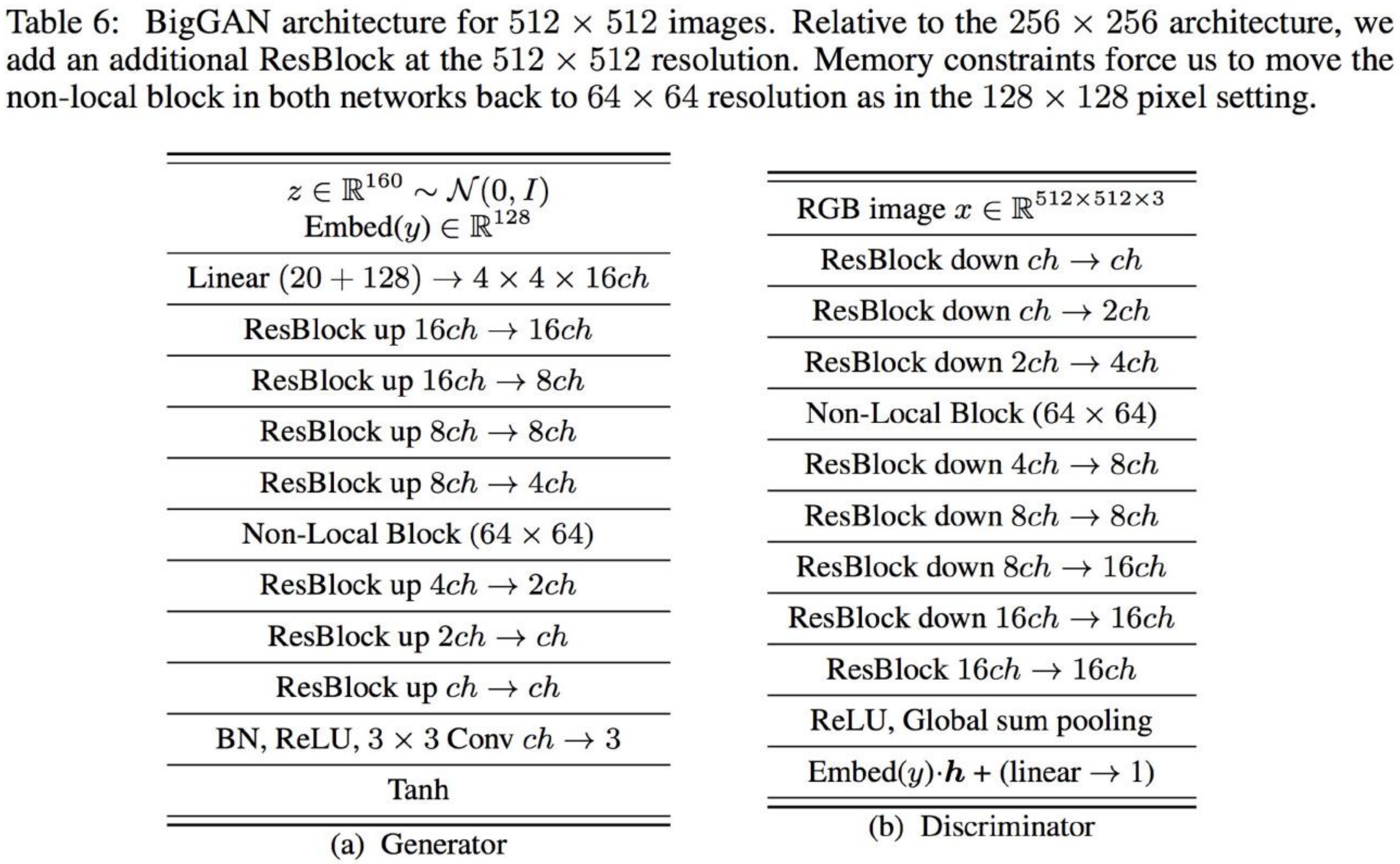


Table 6: BigGAN architecture for 512×512 images. Relative to the 256×256 architecture, we add an additional ResBlock at the 512×512 resolution. Memory constraints force us to move the non-local block in both networks back to 64×64 resolution as in the 128×128 pixel setting.

$z \in \mathbb{R}^{160} \sim \mathcal{N}(0, I)$ Embed(y) $\in \mathbb{R}^{128}$
Linear $(20 + 128) \rightarrow 4 \times 4 \times 16ch$
ResBlock up $16ch \rightarrow 16ch$
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 4ch$
Non-Local Block (64×64)
ResBlock up $4ch \rightarrow 2ch$
ResBlock up $2ch \rightarrow ch$
ResBlock up $ch \rightarrow ch$
BN, ReLU, 3×3 Conv $ch \rightarrow 3$
Tanh

(a) Generator

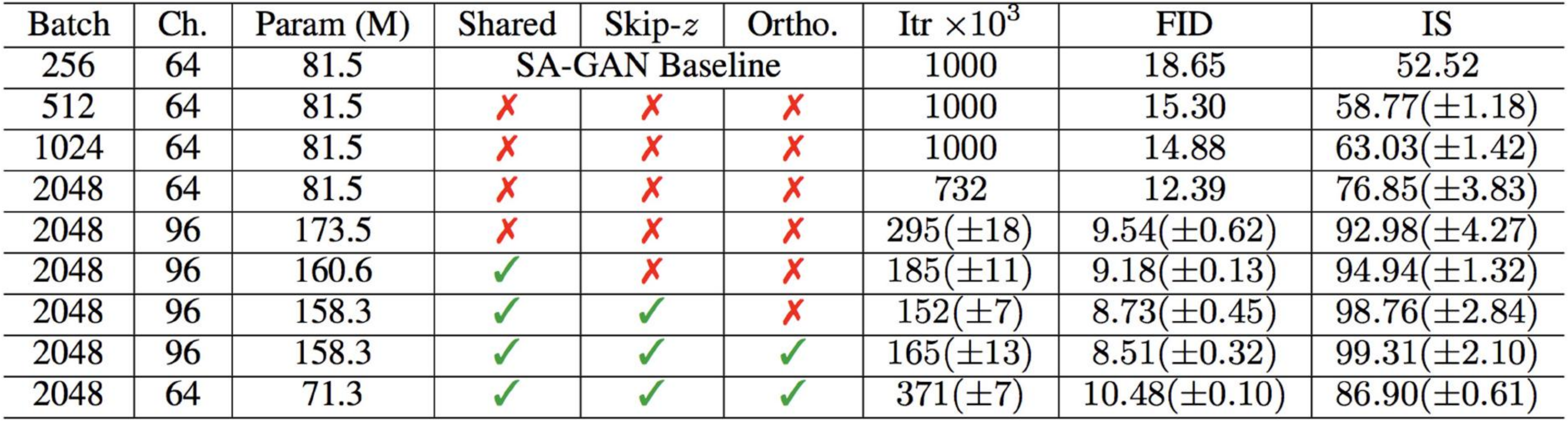
RGB image $x \in \mathbb{R}^{512 \times 512 \times 3}$
ResBlock down $ch \rightarrow ch$
ResBlock down $ch \rightarrow 2ch$
ResBlock down $2ch \rightarrow 4ch$
Non-Local Block (64×64)
ResBlock down $4ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$
ResBlock down $16ch \rightarrow 16ch$
ResBlock $16ch \rightarrow 16ch$
ReLU, Global sum pooling
Embed(y)$\cdot\boldsymbol{h}$ + (linear $\rightarrow 1$)

(b) Discriminator

BigGAN

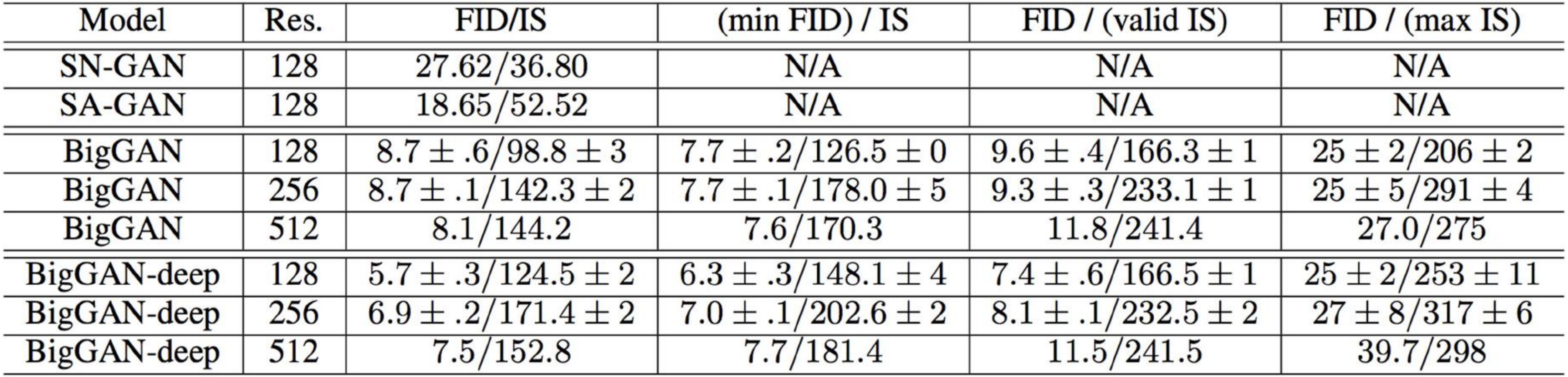
- Increase your batch size (as much as you can)
- Use Cross-Replica (Sync) Batch Norm
- Increase your model size
- Wider helps as much as deeper
- Fuse class information at all levels
- Hinge Loss
- Orthonormal regularization & Truncation Trick

BigGAN



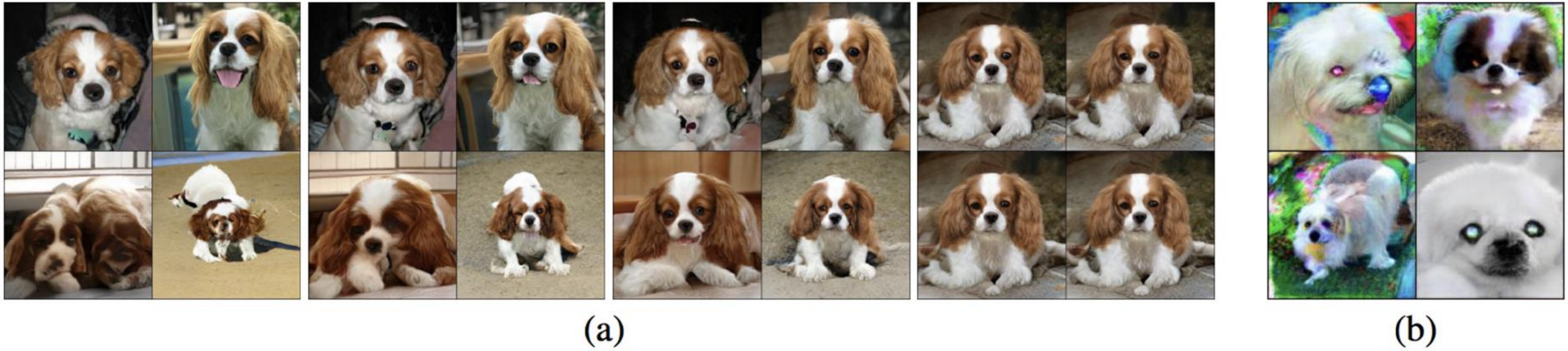
Batch	Ch.	Param (M)	Shared	Skip-z	Ortho.	Itr $\times 10^3$	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	✗	✗	✗	1000	15.30	58.77(±1.18)
1024	64	81.5	✗	✗	✗	1000	14.88	63.03(±1.42)
2048	64	81.5	✗	✗	✗	732	12.39	76.85(±3.83)
2048	96	173.5	✗	✗	✗	295(±18)	9.54(±0.62)	92.98(±4.27)
2048	96	160.6	✓	✗	✗	185(±11)	9.18(±0.13)	94.94(±1.32)
2048	96	158.3	✓	✓	✗	152(±7)	8.73(±0.45)	98.76(±2.84)
2048	96	158.3	✓	✓	✓	165(±13)	8.51(±0.32)	99.31(±2.10)
2048	64	71.3	✓	✓	✓	371(±7)	10.48(±0.10)	86.90(±0.61)

BigGAN



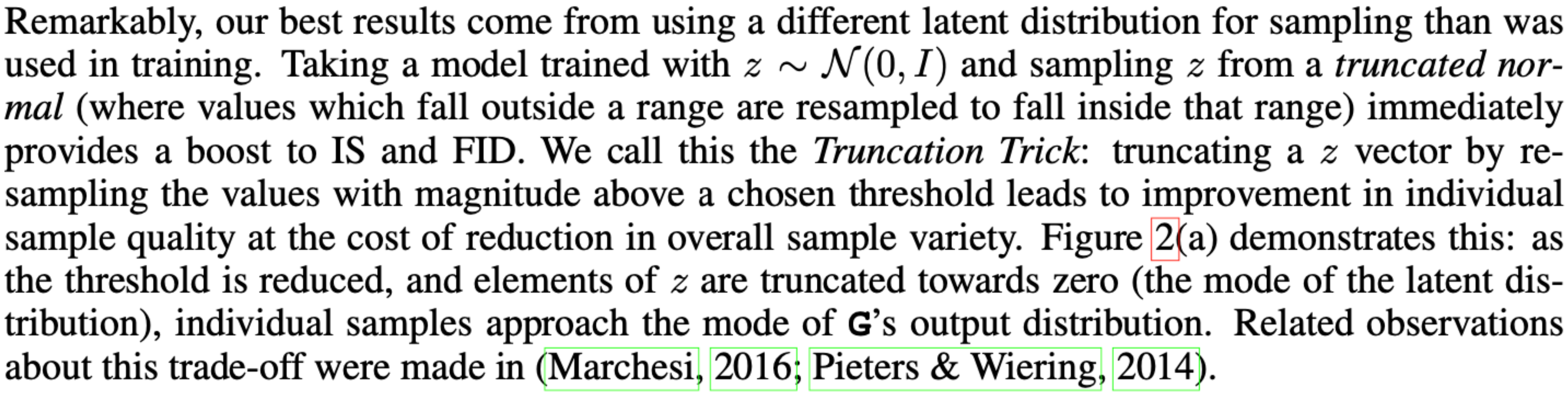
Model	Res.	FID/IS	(min FID) / IS	FID / (valid IS)	FID / (max IS)
SN-GAN	128	$27.62/36.80$	N/A	N/A	N/A
SA-GAN	128	$18.65/52.52$	N/A	N/A	N/A
BigGAN	128	$8.7 \pm .6/98.8 \pm 3$	$7.7 \pm .2/126.5 \pm 0$	$9.6 \pm .4/166.3 \pm 1$	$25 \pm 2/206 \pm 2$
BigGAN	256	$8.7 \pm .1/142.3 \pm 2$	$7.7 \pm .1/178.0 \pm 5$	$9.3 \pm .3/233.1 \pm 1$	$25 \pm 5/291 \pm 4$
BigGAN	512	$8.1/144.2$	$7.6/170.3$	$11.8/241.4$	$27.0/275$
BigGAN-deep	128	$5.7 \pm .3/124.5 \pm 2$	$6.3 \pm .3/148.1 \pm 4$	$7.4 \pm .6/166.5 \pm 1$	$25 \pm 2/253 \pm 11$
BigGAN-deep	256	$6.9 \pm .2/171.4 \pm 2$	$7.0 \pm .1/202.6 \pm 2$	$8.1 \pm .1/232.5 \pm 2$	$27 \pm 8/317 \pm 6$
BigGAN-deep	512	$7.5/152.8$	$7.7/181.4$	$11.5/241.5$	$39.7/298$

BigGAN - Truncation Trick



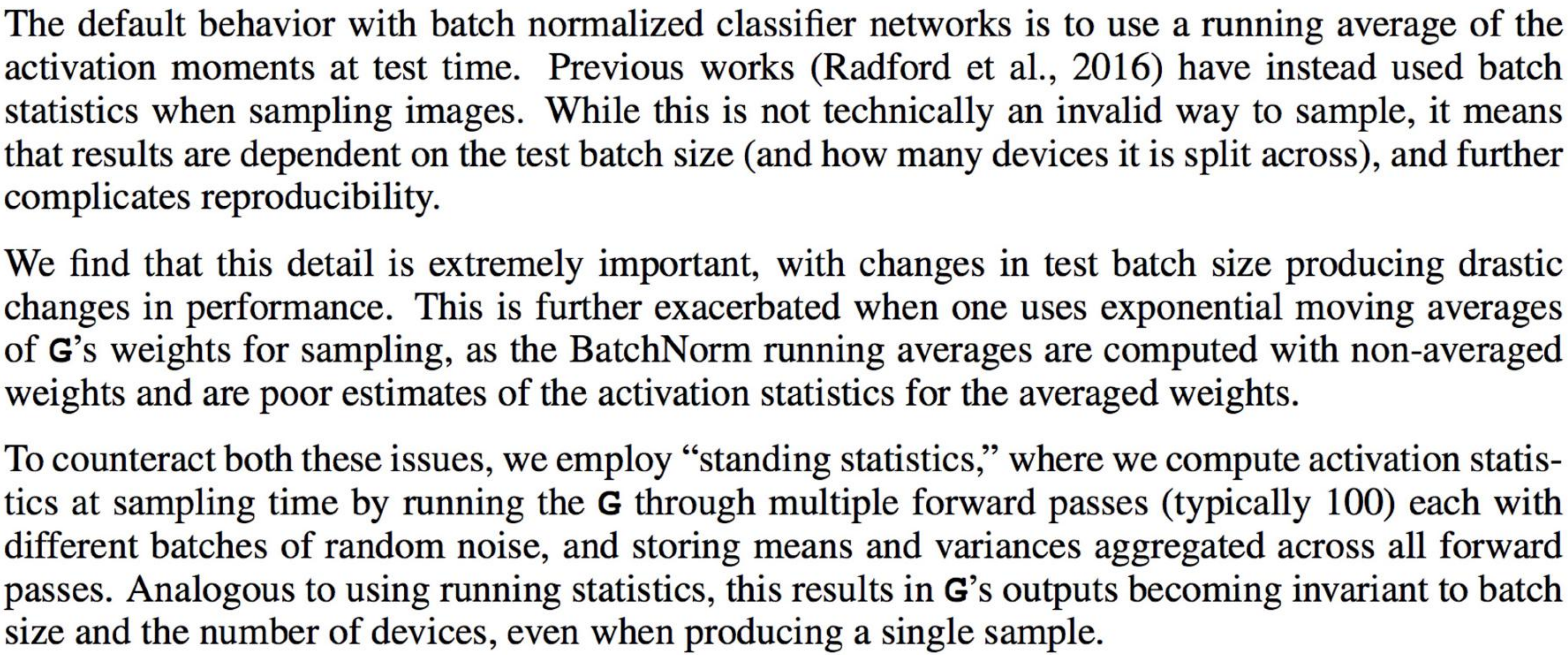
(a)

(b)



Remarkably, our best results come from using a different latent distribution for sampling than was used in training. Taking a model trained with $z \sim \mathcal{N}(0, I)$ and sampling z from a *truncated normal* (where values which fall outside a range are resampled to fall inside that range) immediately provides a boost to IS and FID. We call this the *Truncation Trick*: truncating a z vector by resampling the values with magnitude above a chosen threshold leads to improvement in individual sample quality at the cost of reduction in overall sample variety. Figure 2(a) demonstrates this: as the threshold is reduced, and elements of z are truncated towards zero (the mode of the latent distribution), individual samples approach the mode of **G**'s output distribution. Related observations about this trade-off were made in (Marchesi, 2016; Pieters & Wiering, 2014).

BigGAN - Sampling

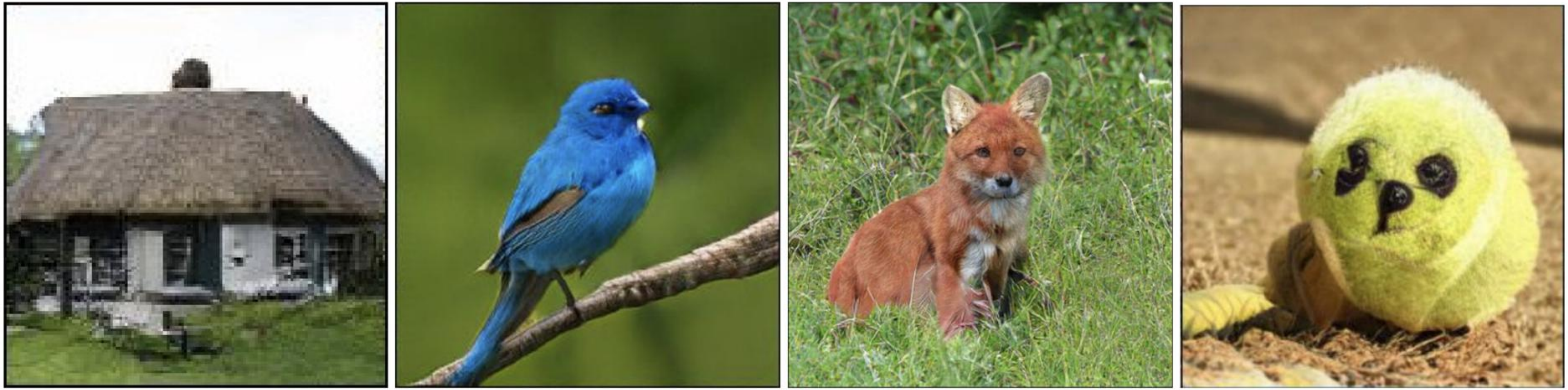


The default behavior with batch normalized classifier networks is to use a running average of the activation moments at test time. Previous works (Radford et al., 2016) have instead used batch statistics when sampling images. While this is not technically an invalid way to sample, it means that results are dependent on the test batch size (and how many devices it is split across), and further complicates reproducibility.

We find that this detail is extremely important, with changes in test batch size producing drastic changes in performance. This is further exacerbated when one uses exponential moving averages of **G**'s weights for sampling, as the BatchNorm running averages are computed with non-averaged weights and are poor estimates of the activation statistics for the averaged weights.

To counteract both these issues, we employ "standing statistics," where we compute activation statistics at sampling time by running the **G** through multiple forward passes (typically 100) each with different batches of random noise, and storing means and variances aggregated across all forward passes. Analogous to using running statistics, this results in **G**'s outputs becoming invariant to batch size and the number of devices, even when producing a single sample.

BigGAN



(a) 128×128 (b) 256×256 (c) 512×512 (d)

BigGAN