

Прикладные задачи анализа данных

Лекция 3
Генеративно-состязательные сети

Михаил Гущин
mhushchyn@hse.ru

НИУ ВШЭ, 2022



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

АЗАМАТ
МУСАГАЛИЕВ

БРЮС
УИЛЛИС

НАГАРНИКИ

ПЕРВЫЙ СЕМЕЙНЫЙ БЛОКБАСТЕР

С 16 АВГУСТА



МЕГАФОН
представляет

План

- ▶ Приложения
- ▶ Генеративно-состязательные сети (GANы)
- ▶ Проблемы GANов
- ▶ Условный GAN
- ▶ GAN vs VAE

Приложения



This X Does Not Exist



This Person Does Not Exist

The site that started it all, with the name that says it all. Created using a style-based generative adversarial network (StyleGAN), this website had the tech community buzzing with excitement and intrigue and inspired many more sites.

Created by Phillip Wang.



This Cat Does Not Exist

These purr-fect GAN-made cats will freshen your feeline-gs and make you wish you could reach through your screen and cuddle them. Once in a while the cats have visual deformities due to imperfections in the model – beware, they can cause nightmares.

Created by Ryan Hoover.



This Rental Does Not Exist

Why bother trying to look for the perfect home when you can create one instead? Just find a listing you like, buy some land, build it, and then enjoy the rest of your life.

Created by Christopher Schmidt.

<https://thisxdoesnotexist.com>

Реалистичные фотографии



2014



2015



2016



2017



2018

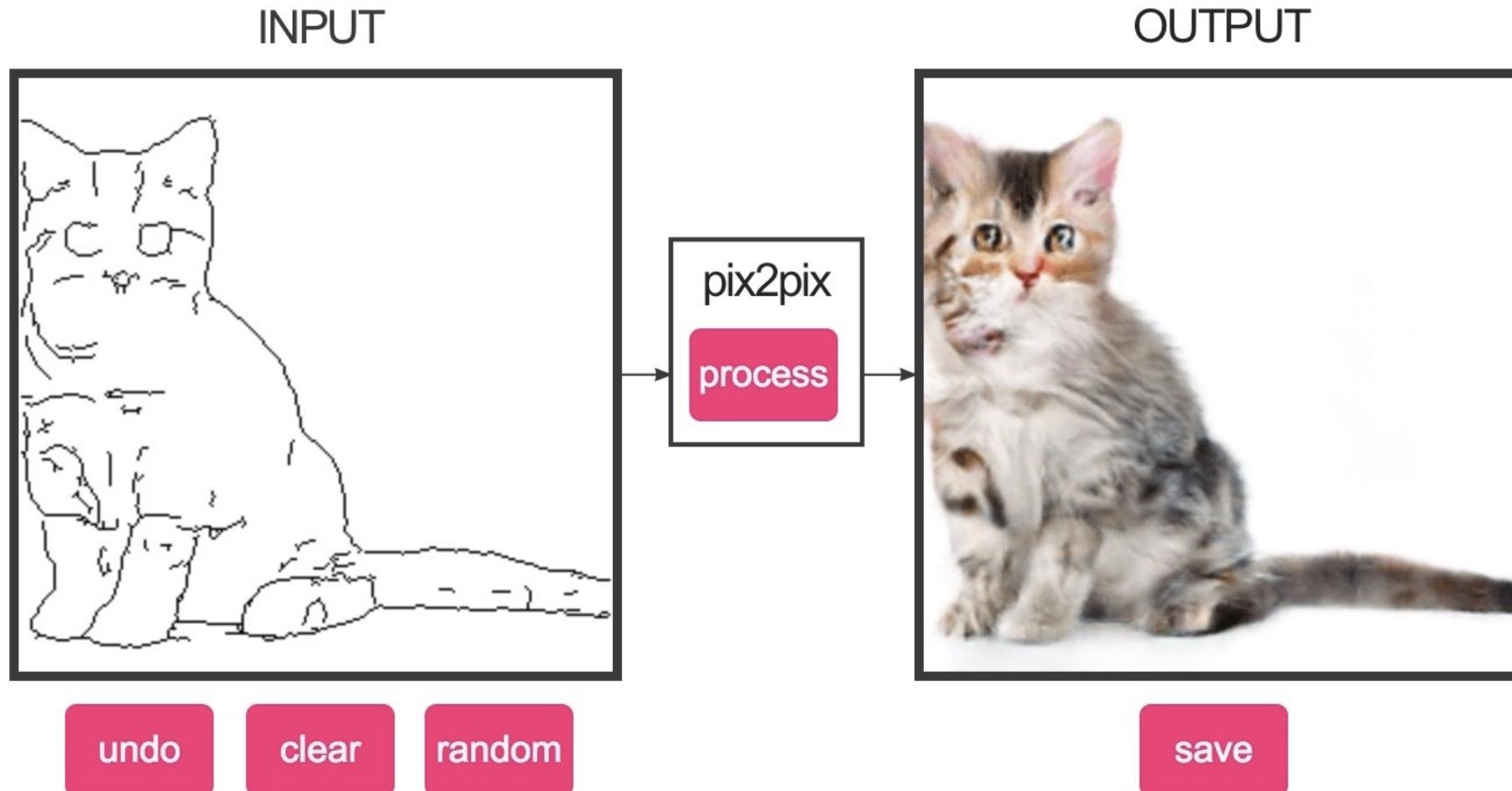
https://twitter.com/goodfellow_ian/status/1084973596236144640?s=20

Deepfake



<https://www.youtube.com/c/CtrlShiftFace>

Image-to-Image Translation



<https://affinelayer.com/pixsrv/>

Другие примеры

- ▶ Style Transfer
- ▶ Super Resolution
- ▶ Photo Inpainting
- ▶ 3D Object Generation
- ▶ Text-to-Image Translation
- ▶ и другие

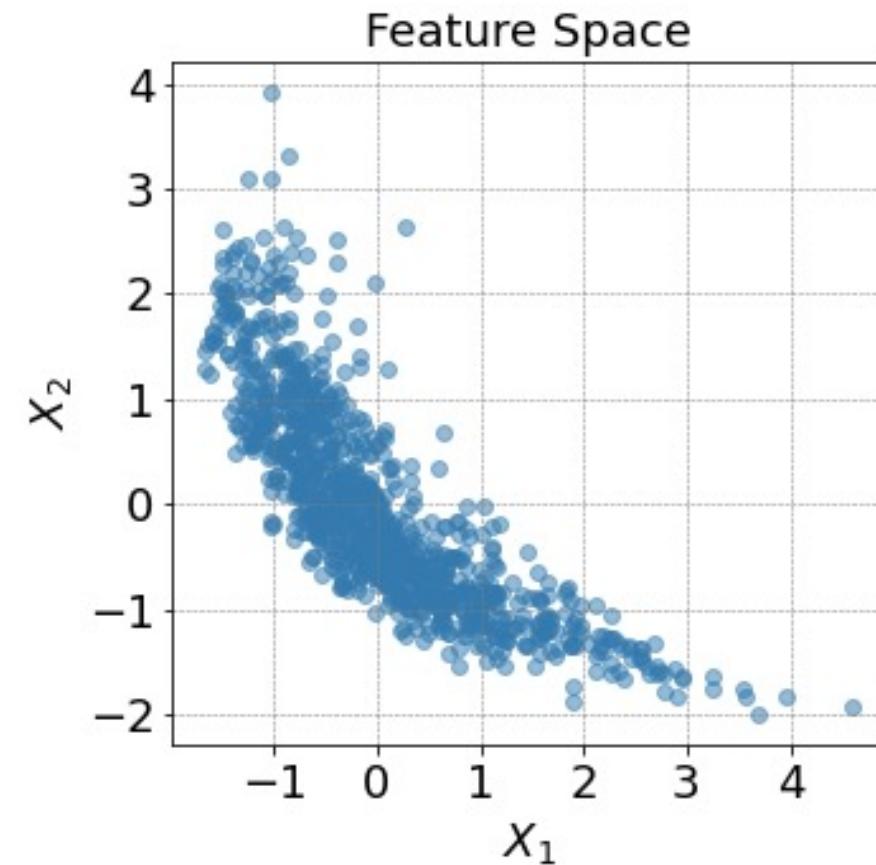
<https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/>

Генеративно-состязательные сети



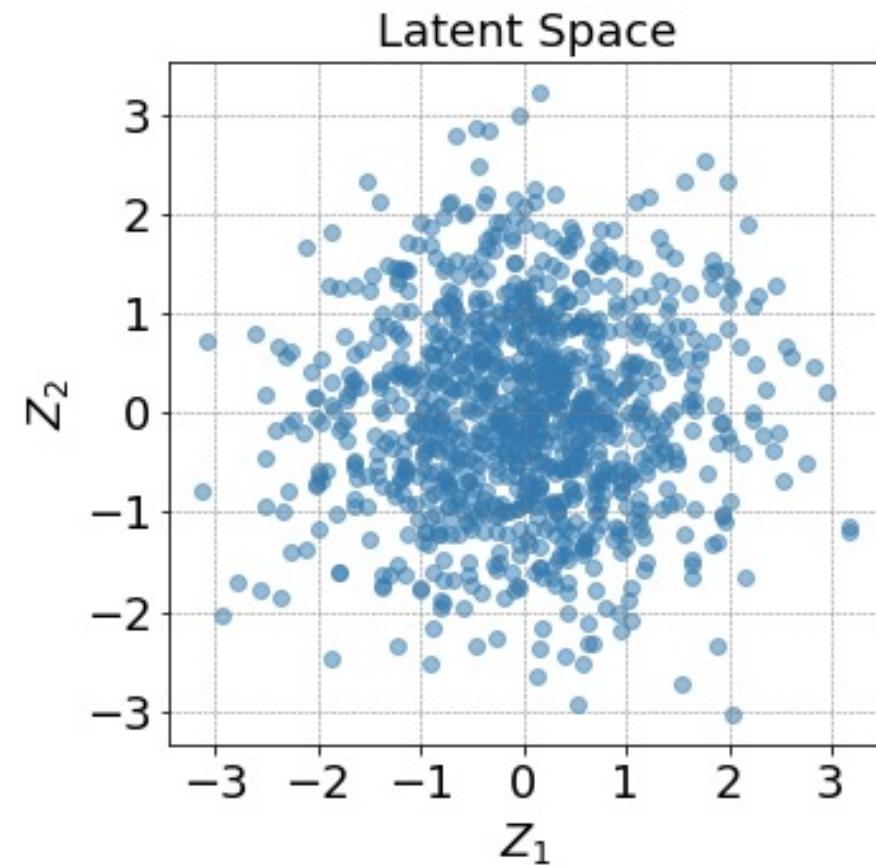
Постановка задачи

- ▶ **Дано:** выборка объектов $\mathbf{x}_i \sim p_x(\mathbf{x})$, где распределение $p_x(\mathbf{x})$ неизвестно.
- ▶ **Задача:** сгенерировать новые объекты $\hat{\mathbf{x}}_j \sim p_x(\mathbf{x})$ с таким же распределением, т.е. похожими на \mathbf{x}_i .

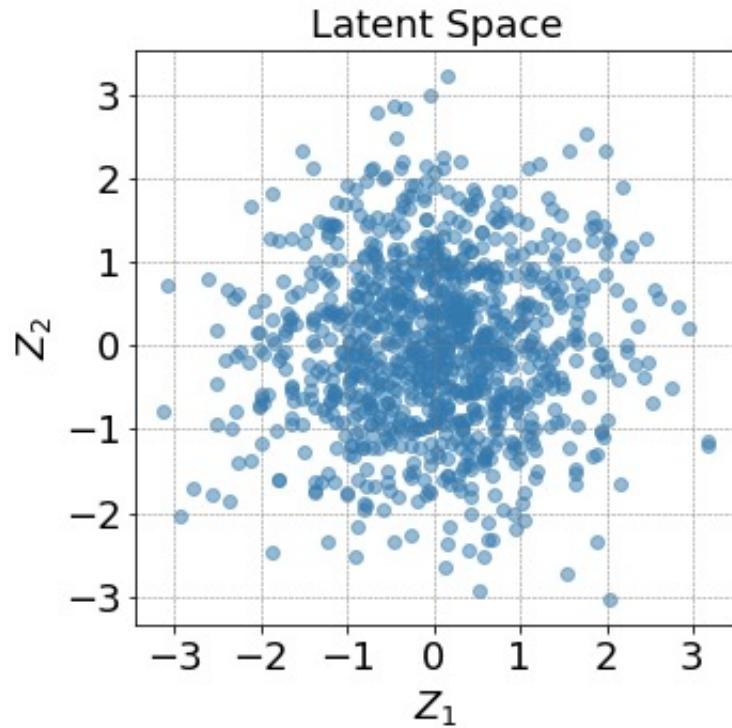


Известные распределения

- ▶ Мы умеем генерировать объекты из известных распределений.
- ▶ Генерируем выборку объектов $\mathbf{z}_i \sim p_z(\mathbf{z})$, где $p_z(z)$ - двумерное нормальное распределение.
- ▶ Пространство \mathbf{Z} называется **скрытым**.



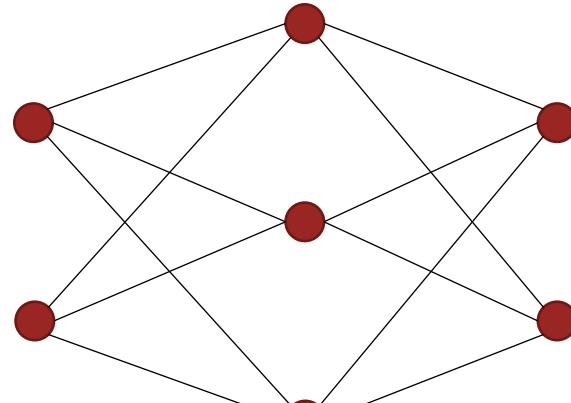
Генератор



Z

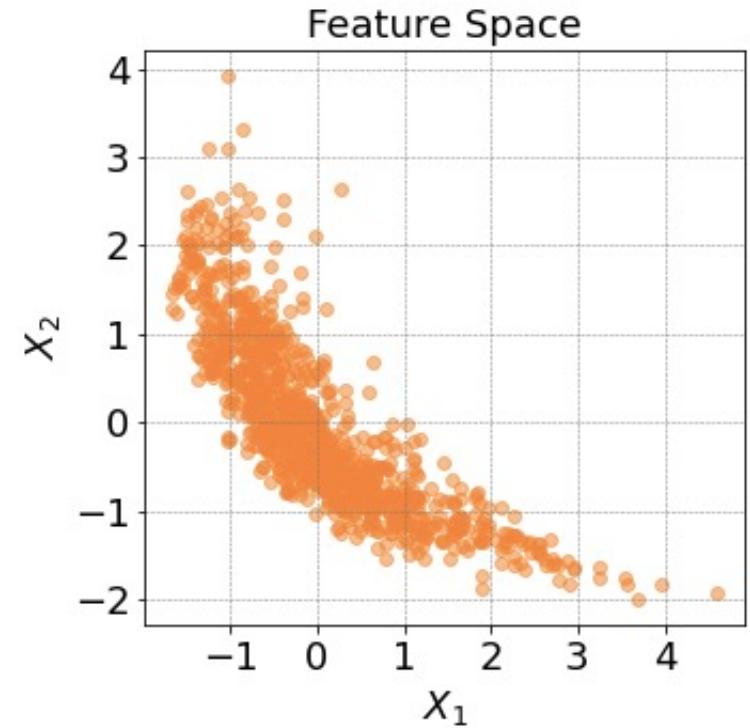
Случайный шум

Нейронная
сеть



$G(Z)$

Генератор



$\hat{X} = G(Z)$

Сгенерированные
объекты

Генератор

- ▶ X – матрица реальных объектов.
- ▶ $\hat{X} = G(Z)$ – матрица сгенерированных объектов.
- ▶ Хотим, чтобы объекты из \hat{X} были похожи на объекты из X .
- ▶ Как посчитать сходство между \hat{X} и X ?
- ▶ Как обучить генератор $G(Z)$?

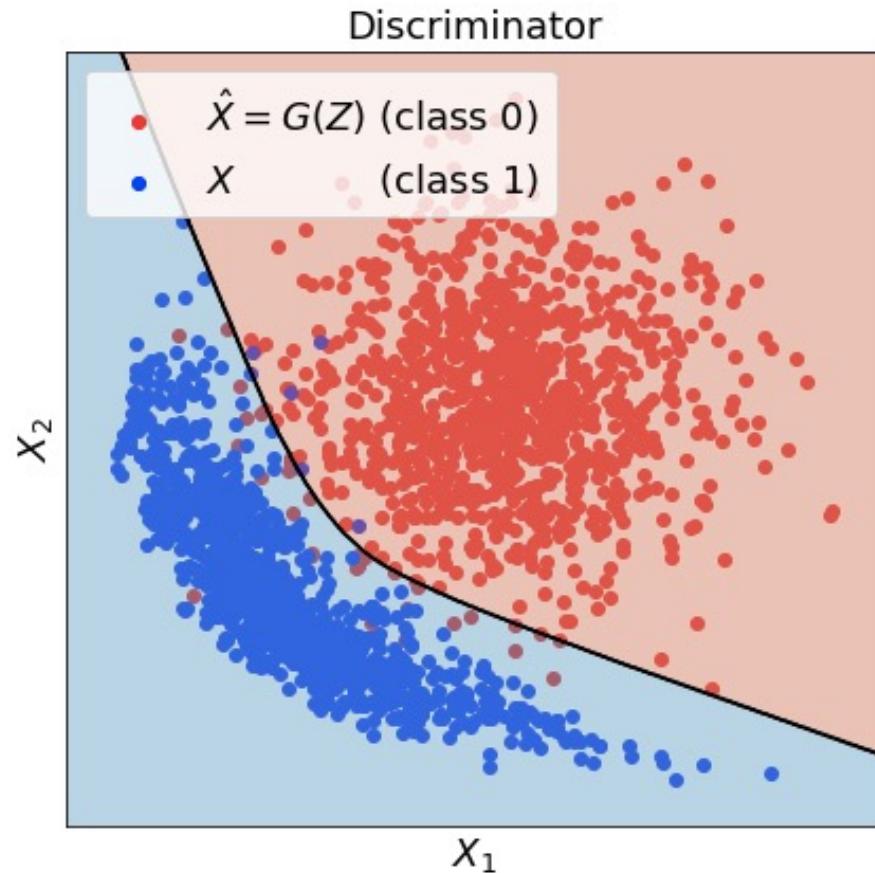
Дискриминатор

Давайте рассмотрим задачу **бинарной классификации**:

- ▶ пусть X – класс 1,
- ▶ пусть \hat{X} – класс 0.

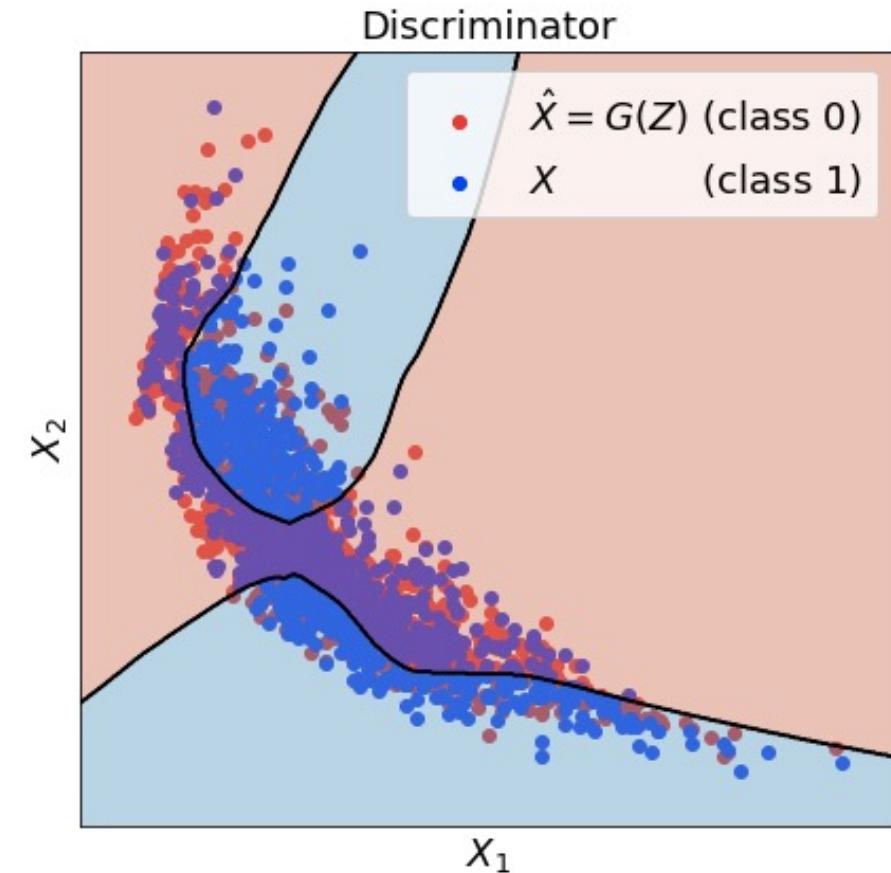
Задача классификатора – разделить два класса. Будем называть его **дискриминатором**.

Дискриминатор



ROC AUC = 0.995

Log Loss = 0.135



ROC AUC = 0.554

Log Loss = 0.689

Дискриминатор

В классификации используется **функция потерь L** :

$$L = -\frac{1}{n} \sum_{i=1}^n y_i \log D(x_i) + (1 - y_i) \log(1 - D(x_i))$$

где $D(x_i)$ – выход дискриминатора, $y_i \in \{0, 1\}$ – метка объекта.

Будем использовать значение этой функции как меру сходства объектов двух классов.

Дискриминатор

Перепишем функцию потерь:

$$L = -\frac{1}{n} \sum_{i:y_i=1} \log D(x_i) - \frac{1}{n} \sum_{i:y_i=0} \log(1 - D(x_i))$$

Здесь каждая сумма берется по своему классу.

Дискриминатор

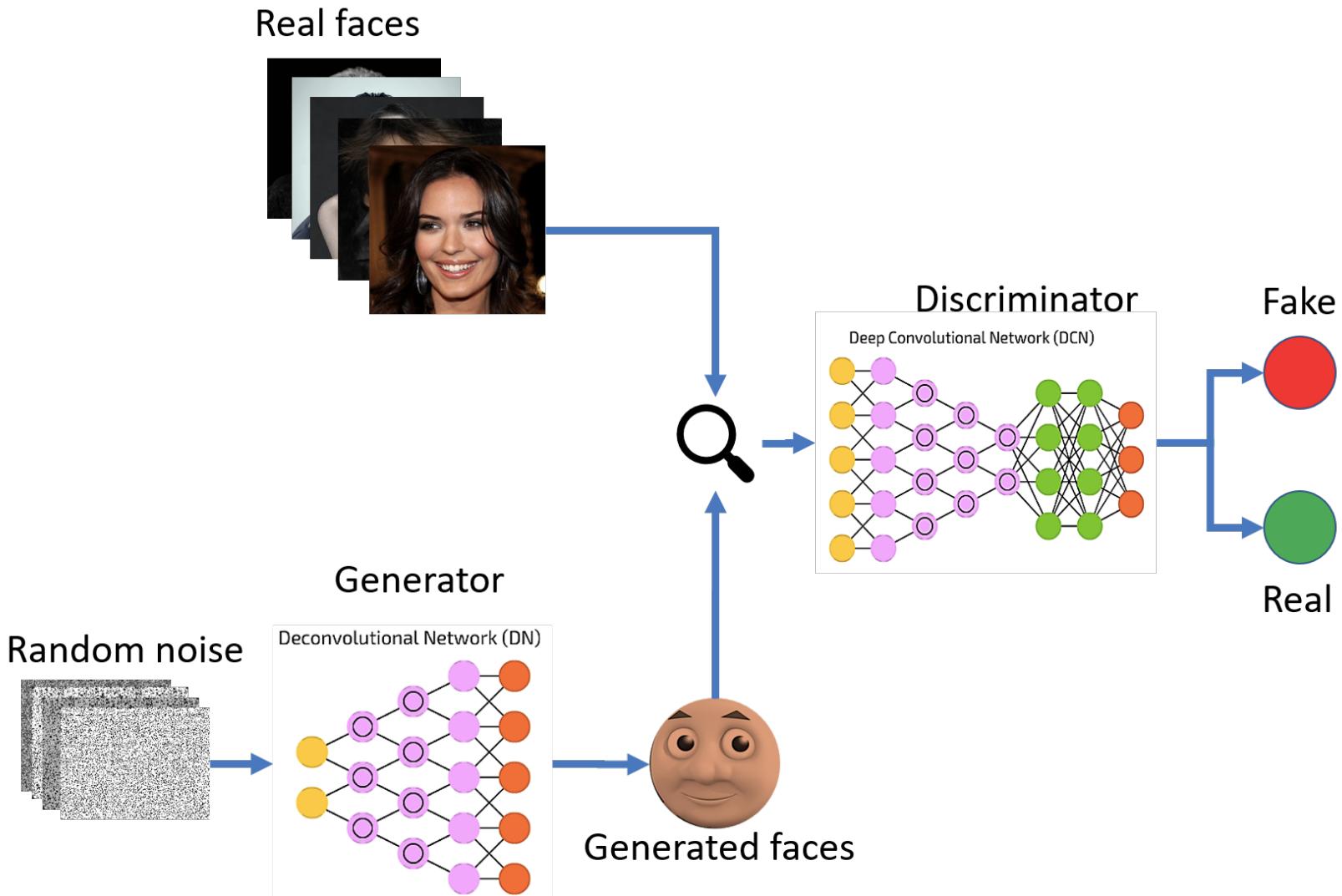
- ▶ Мы договорились, что X – класс 1, а $\hat{X} = G(Z)$ – класс 0.
- ▶ Тогда функция потерь имеет вид:

$$L(\mathbf{D}, \mathbf{G}) = -\frac{1}{n} \sum_{x_i \in X} \log \mathbf{D}(x_i) - \frac{1}{n} \sum_{z_i \in Z} \log(1 - \mathbf{D}(\mathbf{G}(z_i)))$$

- ▶ Обучение дискриминатора и генератора:

$$\max_{\mathbf{G}} \min_{\mathbf{D}} L(\mathbf{D}, \mathbf{G})$$

Алгоритм



Link: <https://www.spindox.it/en/blog/generative-adversarial-neural-networks/>

Алгоритм

- ▶ k раз делаем:

- Генерируем шум $\{z_1, z_2, \dots, z_m\}$
- Берем m реальных объектов $\{x_1, x_2, \dots, x_m\}$
- Обновляем веса дискриминатора W_D :

$$W_D = W_D - \alpha \nabla L(\mathbf{D}, \mathbf{G})$$

- ▶ Генерируем шум $\{z'_1, z'_2, \dots, z'_m\}$
- ▶ Обновляем веса генератора W_G :

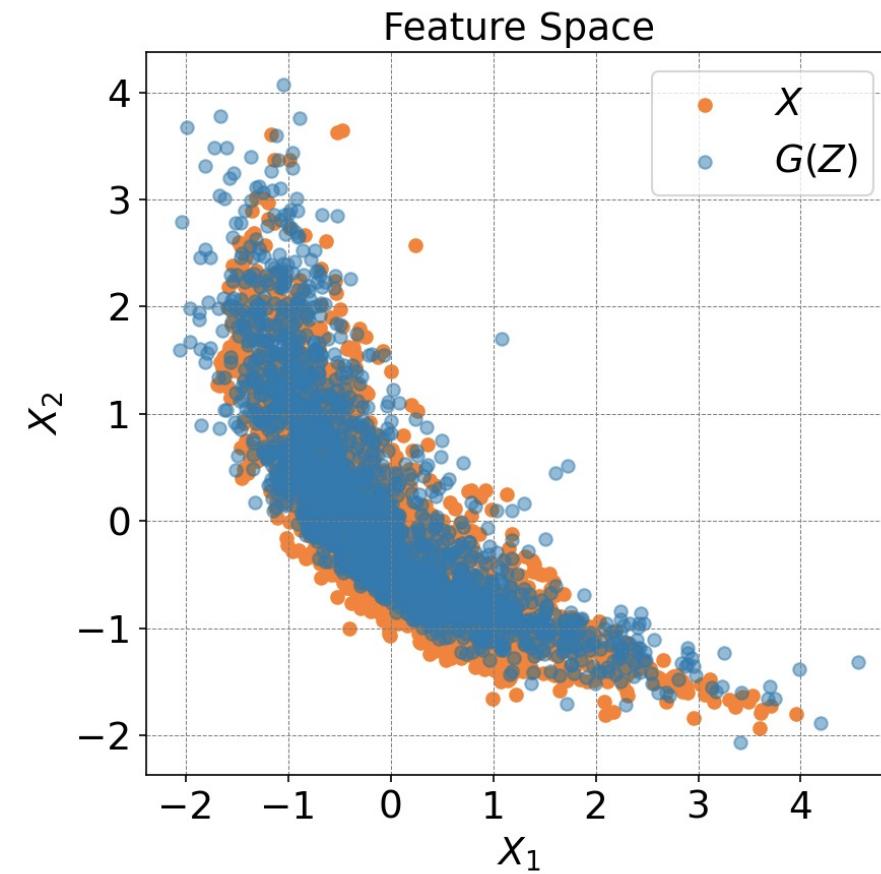
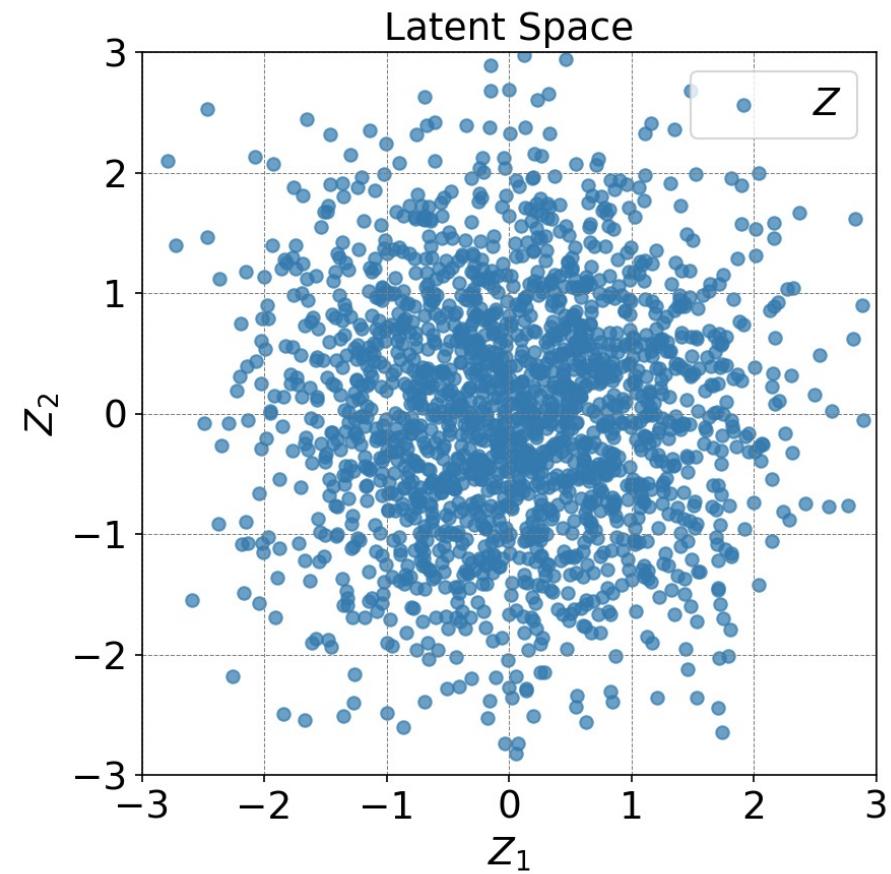
$$W_G = W_G + \alpha \nabla L(\mathbf{D}, \mathbf{G})$$

- ▶ Повторить шаги

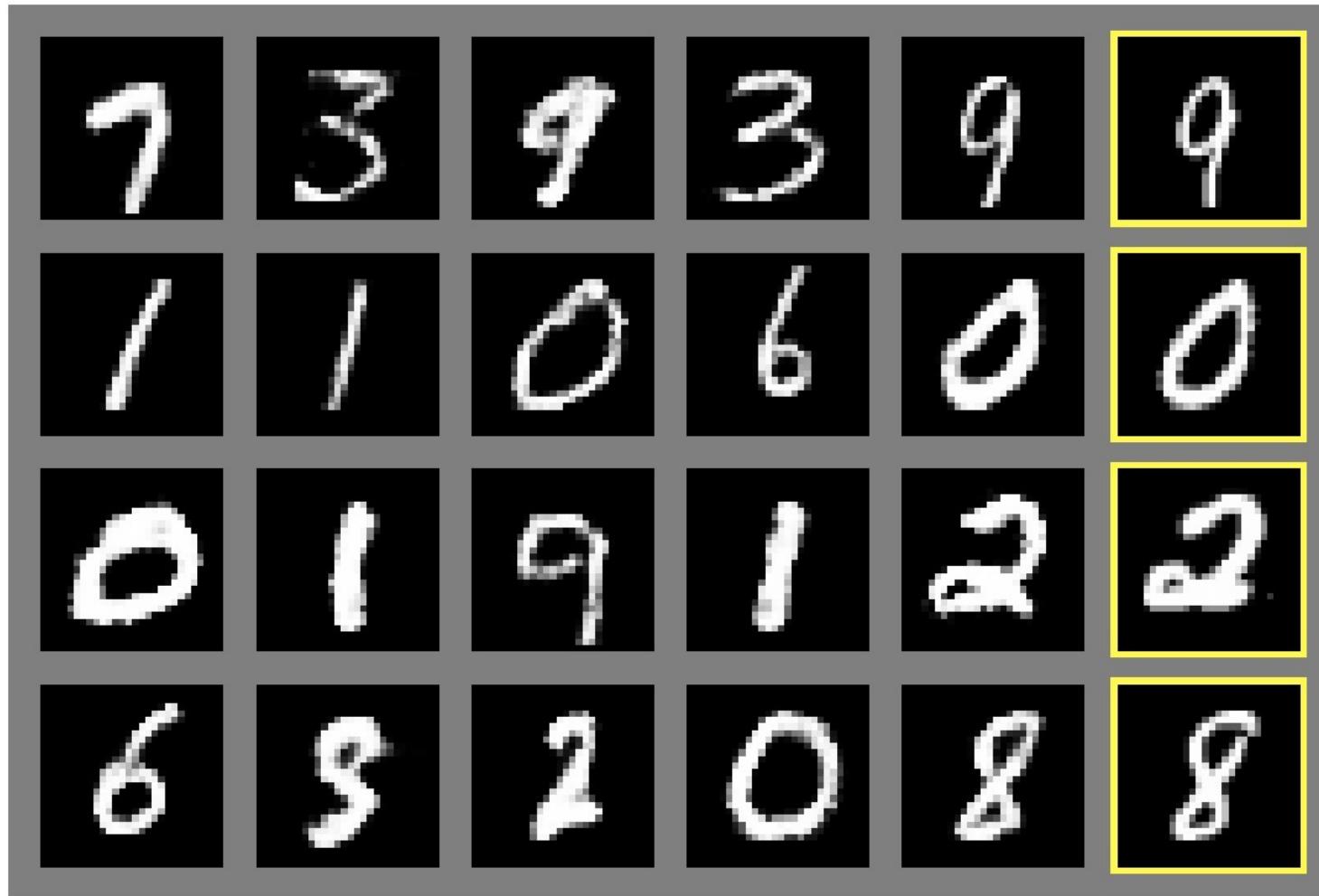


<https://arxiv.org/abs/1406.2661>

Пример



Пример



<https://arxiv.org/abs/1406.2661>

Пример



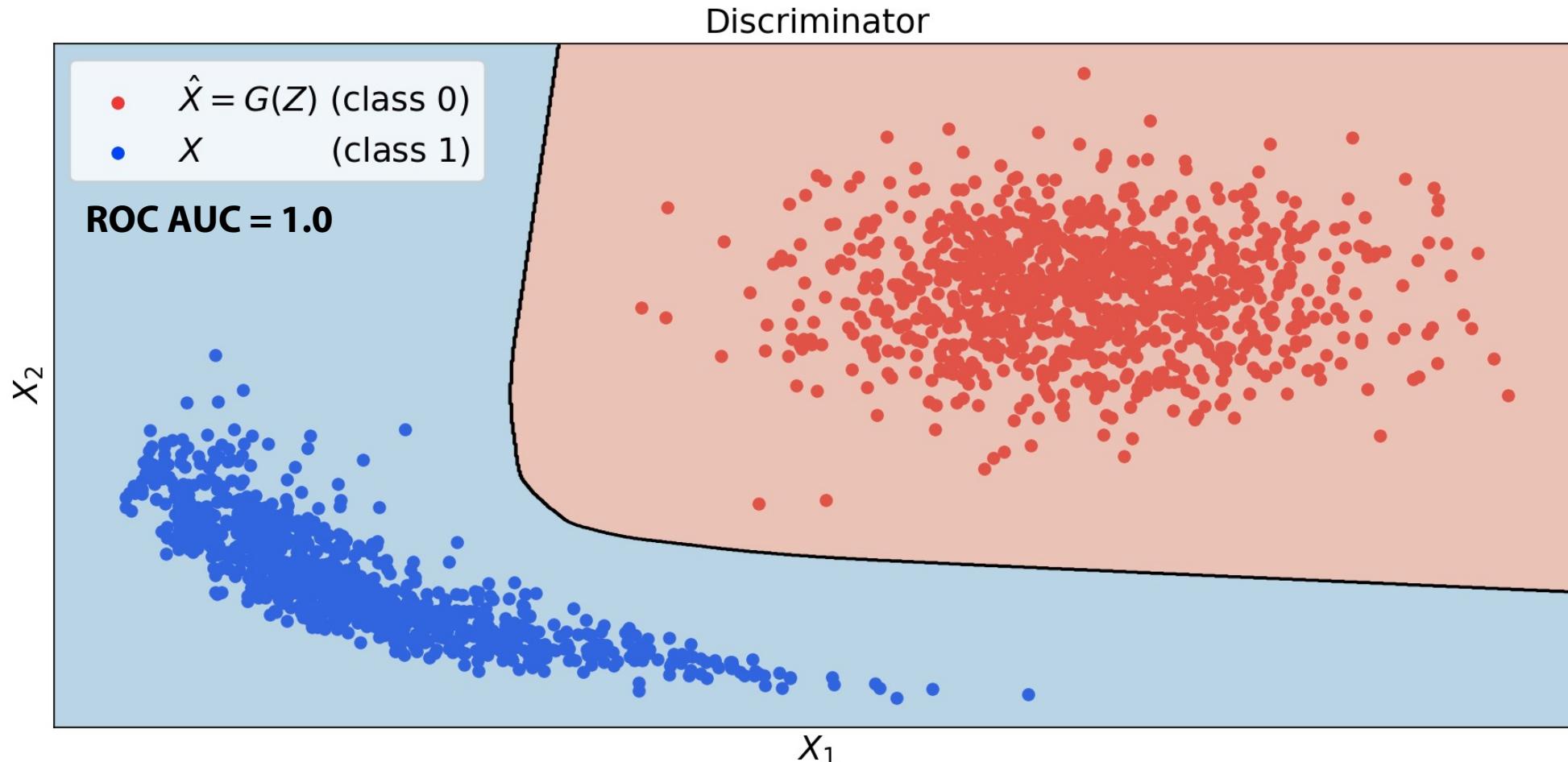
<https://arxiv.org/abs/1710.10196>

Проблемы GaNов



Затухание градиентов

Какая-то итерация обучения GAN:



Затухание градиентов

- ▶ Классы идеально разделяются дискриминатором =>
- ▶ $\mathbf{D}(x_i) = \mathbf{1}$ для любого x_i в X ;
- ▶ $\mathbf{D}(\hat{x}_j) = \mathbf{D}(\mathbf{G}(z_j)) = \mathbf{0}$ для любого z_j в Z .
- ▶ Подставим эти значения в функцию потерь:

$$L(\mathbf{D}, \mathbf{G}) = -\frac{1}{n} \sum_{x_i \in X} \log \mathbf{D}(x_i) - \frac{1}{n} \sum_{z_i \in Z} \log(1 - \mathbf{D}(\mathbf{G}(z_i)))$$

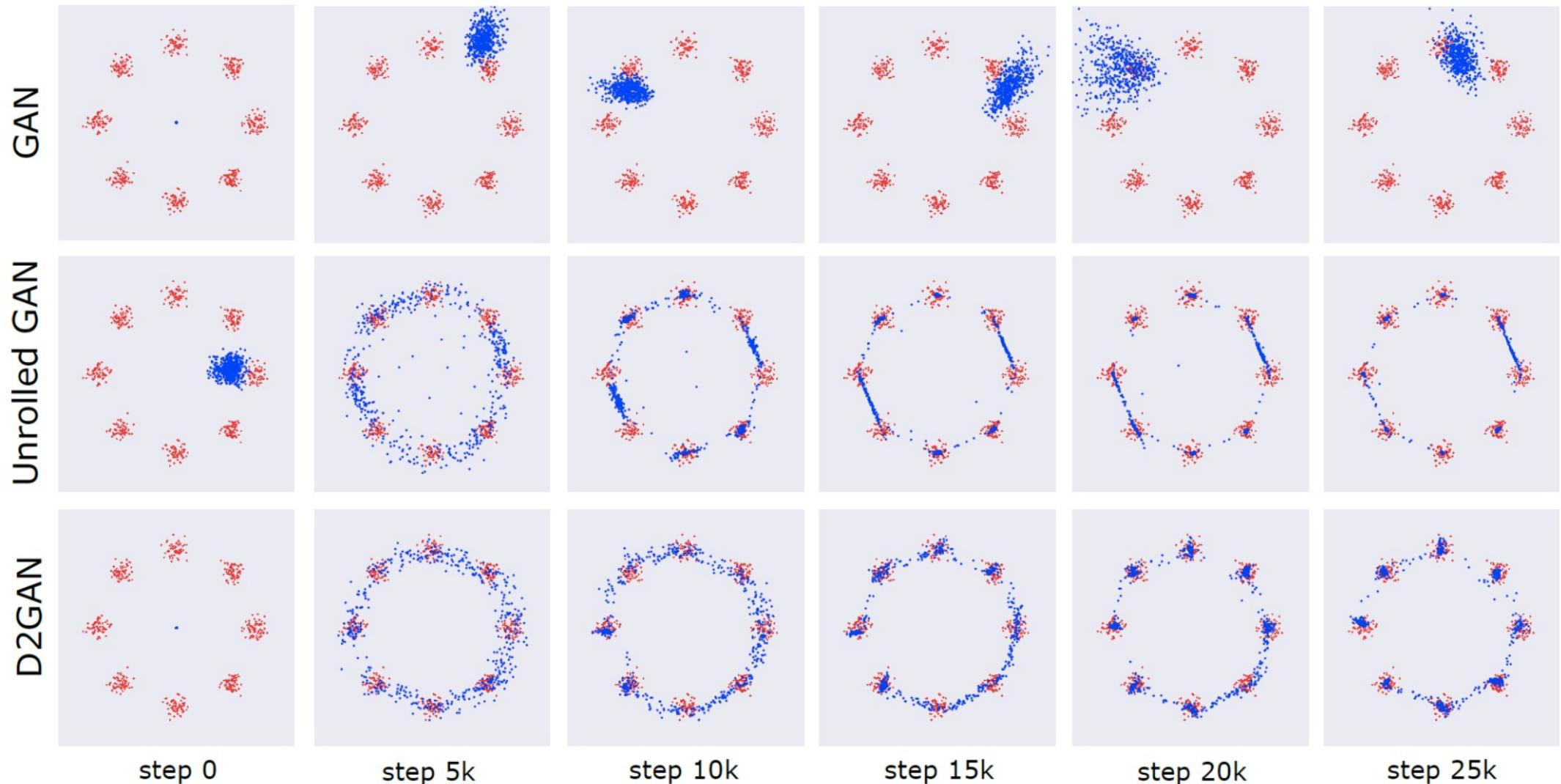
Затухание градиентов

- ▶ В результате получаем:

$$L(\mathbf{D}, \mathbf{G}) = -\frac{1}{n} \sum_{x_i \in X} \log \mathbf{1} - \frac{1}{n} \sum_{z_i \in Z} \log(1 - \mathbf{0}) = 0 = \text{const}$$

- ▶ $\nabla L(\mathbf{D}, \mathbf{G}) = 0 \Rightarrow$
- ▶ Дискриминатор и генератор больше **не обучаются!**

Схлопывание мод



<https://arxiv.org/abs/1709.03831>

Пример

1	1	1	0	0	0	0	0
0	0	1	1	0	0	0	1
1	1	1	0	1	0	0	1
0	0	1	1	1	1	0	1
0	0	1	1	1	1	1	0
0	0	0	0	0	0	1	1
0	1	0	1	0	0	0	1
1	0	1	1	1	1	0	1

Схлопывание мод

Причины:

- ▶ $D(x_i) = 1$ для x_i из некоторых мод =>
- ▶ $\nabla L(D, G) = 0$ в окрестности этих мод =>
- ▶ GAN не учится генерировать эти моды;
- ▶ Что-то еще, что все плохо понимают ☺

Условный GAN



Задача

Input

A yellow bird with brown wings and a pointed bill



This bird is blue and black in color, with a sharp beak and small eyes



A white bird with white feathers and gray wings



This small bird with a red belly, a pointed bill and red wings



StackGAN++



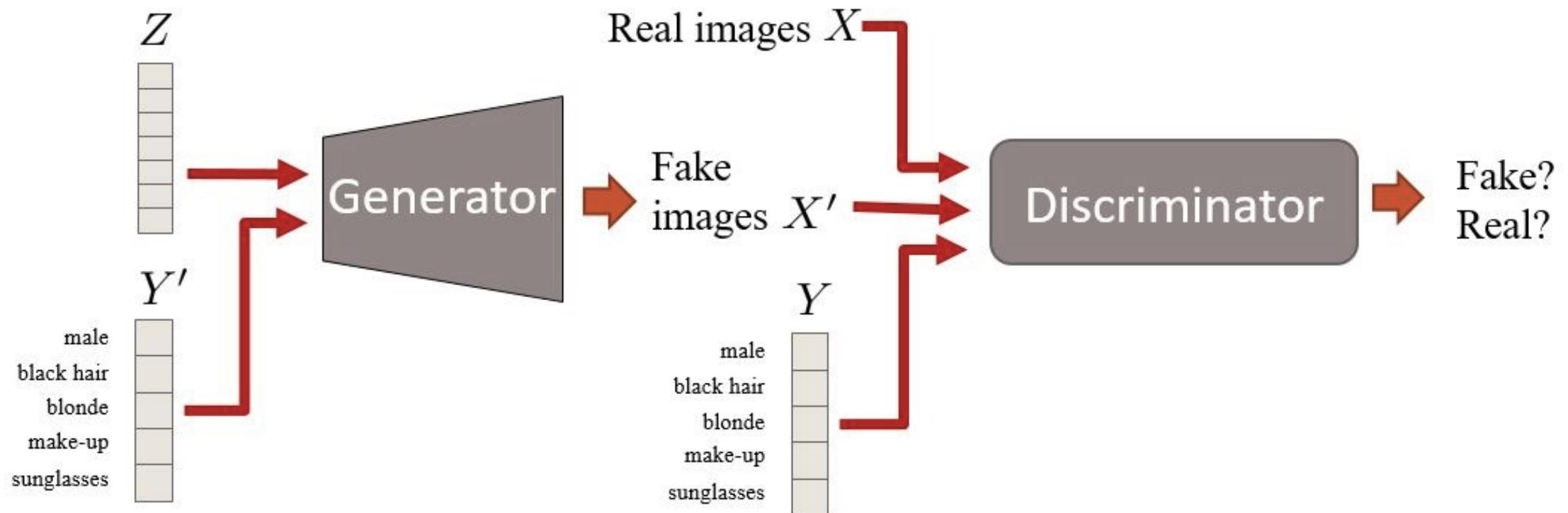
AttnGAN



ACGAN
(256 × 256)



Conditional GAN



<https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them>

ФУНКЦИЯ ПОТЕРЬ

$$L(\mathbf{D}, \mathbf{G}) = -\frac{1}{n} \sum_{x_i \in X} \log \mathbf{D}(x_i, \mathbf{c}_i) - \frac{1}{n} \sum_{z_i \in Z} \log(1 - \mathbf{D}(\mathbf{G}(z_i, \mathbf{c}_i)))$$

где \mathbf{c}_i - вектор некоторого условия

Conditional GAN

Условием Y может являться:

- ▶ Метка класса
 - «кот», «собака», «машина» и т.д.
- ▶ Дополнительные признаки
 - возраст, рост, освещенность и т.д.
- ▶ Другая картинка
 - эскиз рисунка

Пример



<https://arxiv.org/abs/1611.07004>

Пример

<https://proceedings.neurips.cc/paper/2017/file/dc6a6489640ca02b0d42dabeb8e46bb7-Paper.pdf>



Пример

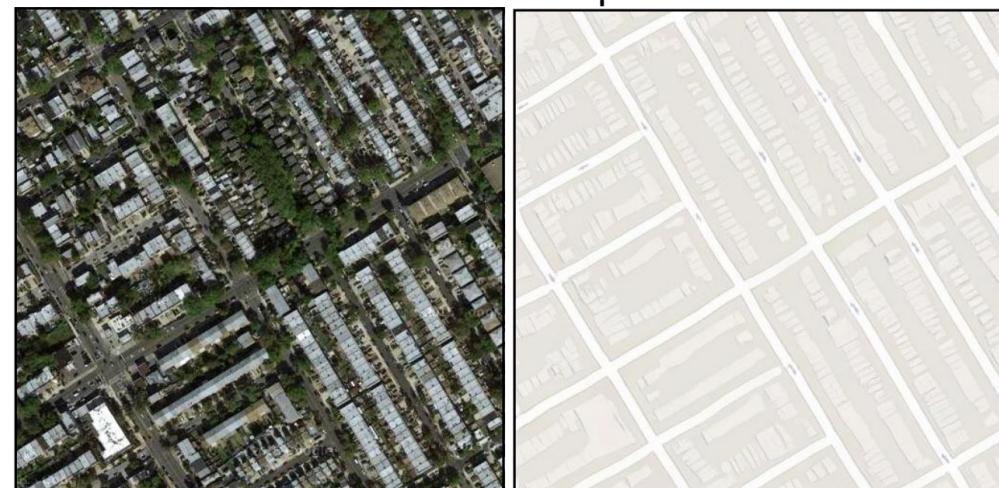
BW to Color



input

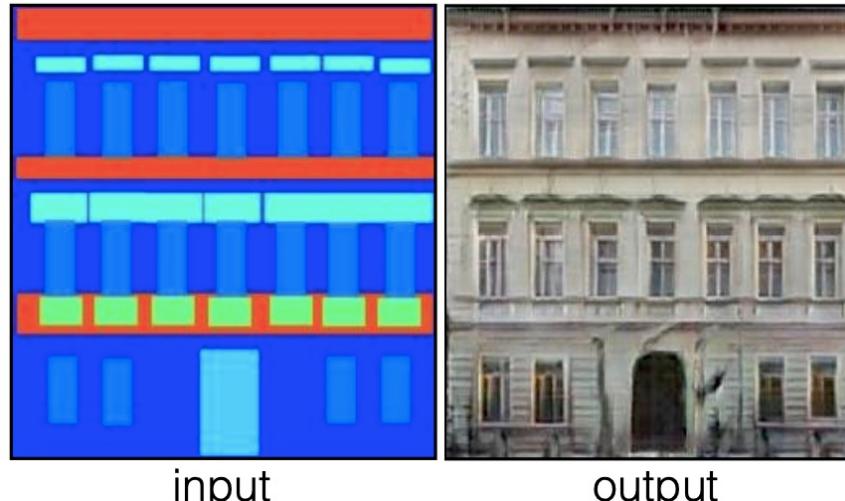
output

Aerial to Map



output

Labels to Facade



<https://arxiv.org/abs/1611.07004>

Другие модели

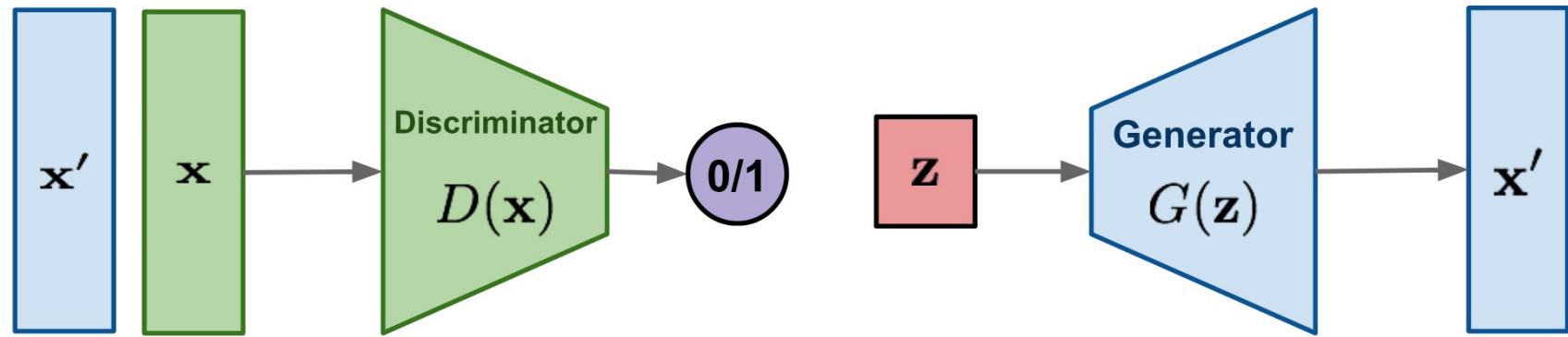
DCGAN	SGAN	AffGAN	RTT-GAN	SL-GAN
WGAN	SimGAN	TP-GAN	GANCS	Context-RNN-GAN
CGAN	VGAN	IcGAN	SSL-GAN	SketchGAN
LAPGAN	iGAN	ID-CGAN	MAD-GAN	GoGAN
SRGAN	3D-GAN	AnoGAN	PrGAN	RWGAN
CycleGAN	CoGAN	LS-GAN	AL-CGAN	MPM-GAN
WGAN-GP	CatGAN	Triple-GAN	ORGAN	MV-BiGAN
EBGAN	MGAN	TGAN	SD-GAN	
VAE-GAN	S ² GAN	BS-GAN	MedGAN	
BiGAN	LSGAN	MaIGAN	SGAN	

GAN vs VAE

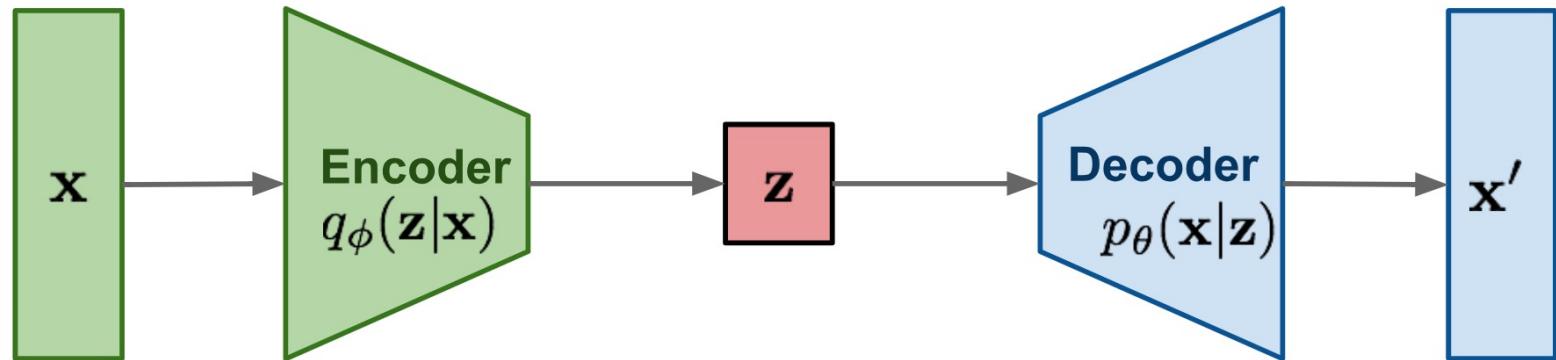


Сравнение моделей

GAN: minimize the classification error loss.



VAE: maximize ELBO.



<https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html>