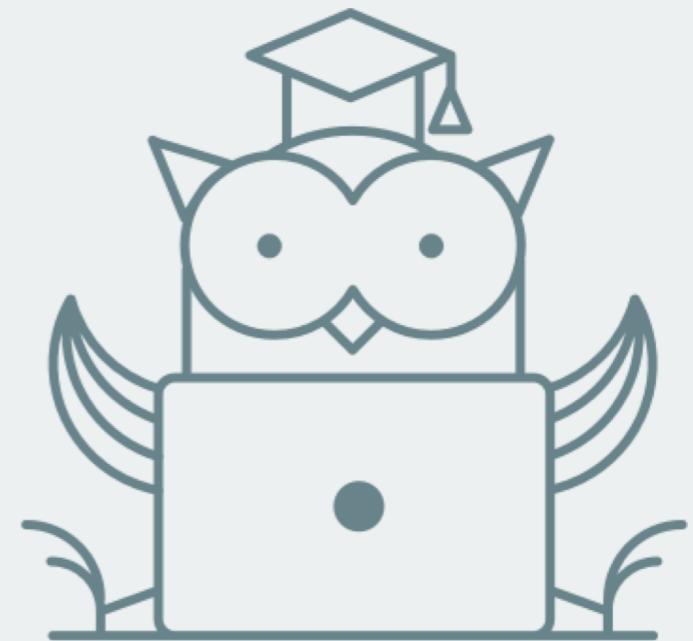


# Состязательные сети

«... и часто ложь, которой поверили,  
становится истиной.» Александр Македонский.

Артур Кадурин  
Insilico Medicine





# План на сегодня

1. Интро
2. Generative Adversarial Nets (GANs)
3. Информация и Адаптация
4. Честная условная генерация
5. Тексты, подкрепление и профессор
6. Другие трюки



# Обо мне

1. Mail.Ru Group. Data Scientist.

4 года анализа данных в рекламе,  
соц.сетях и поиске.





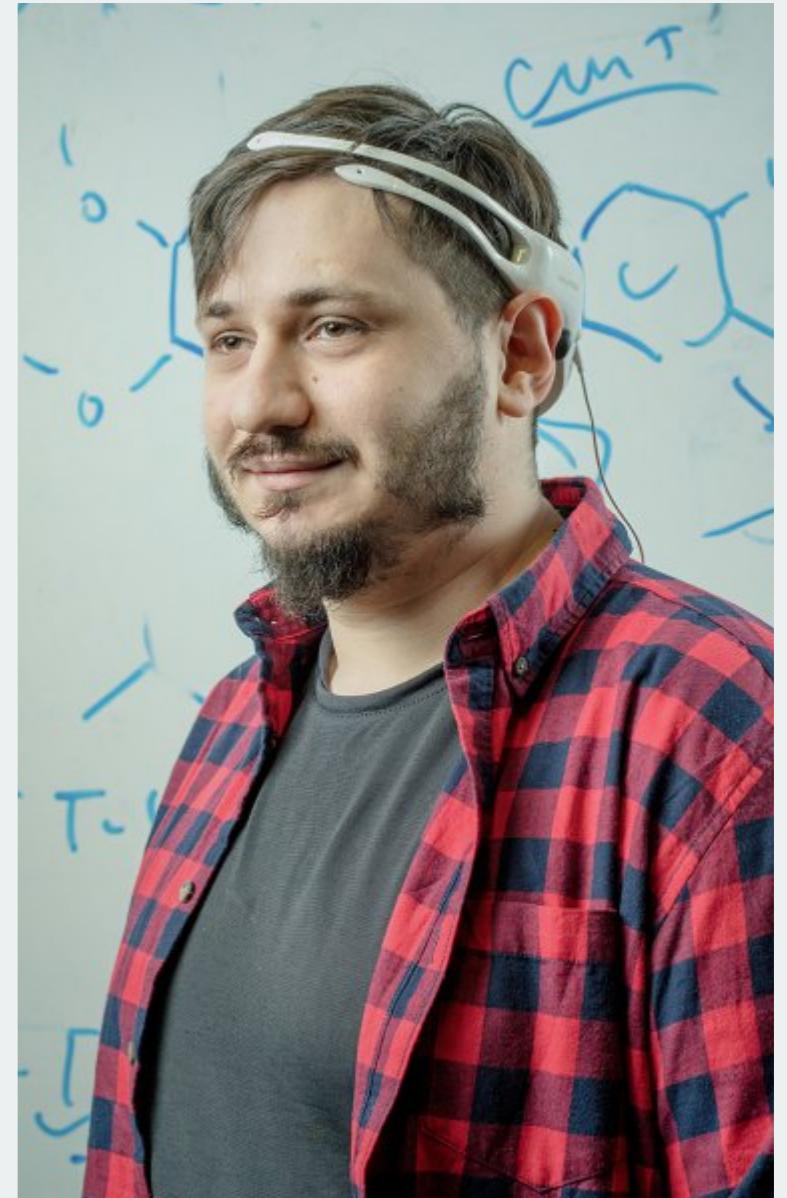
# Обо мне

## 1. Mail.Ru Group. Data Scientist.

4 года анализа данных в рекламе,  
соц.сетях и поиске.

## 2. Insilico Medicine. Chief AI Officer

2 года исследований в области  
применения глубокого обучения для  
разработки лекарственных препаратов





# Обо мне

## 1. Mail.Ru Group. Data Scientist.

4 года анализа данных в рекламе,  
соц.сетях и поиске.

## 2. Insilico Medicine. Chief AI Officer

2 года исследований в области  
применения глубокого обучения для  
разработки лекарственных препаратов

## 3. Книга «Глубокое обучение»





# Обо мне

## 1. Mail.Ru Group. Data Scientist.

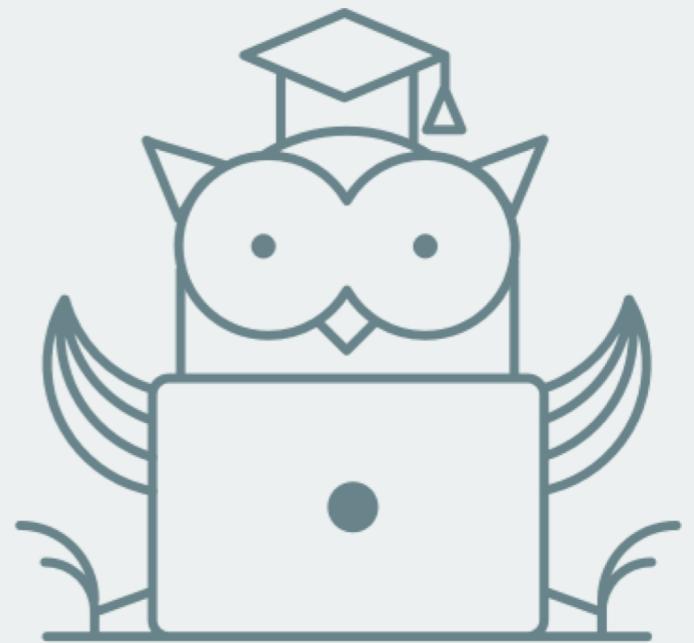
4 года анализа данных в рекламе,  
соц.сетях и поиске.

## 2. Insilico Medicine. Chief AI Officer

2 года исследований в области  
применения глубокого обучения для  
разработки лекарственных препаратов

## 3. Книга «Глубокое обучение»

## 4. Курс на платформе Отус



O T U S

# О вас

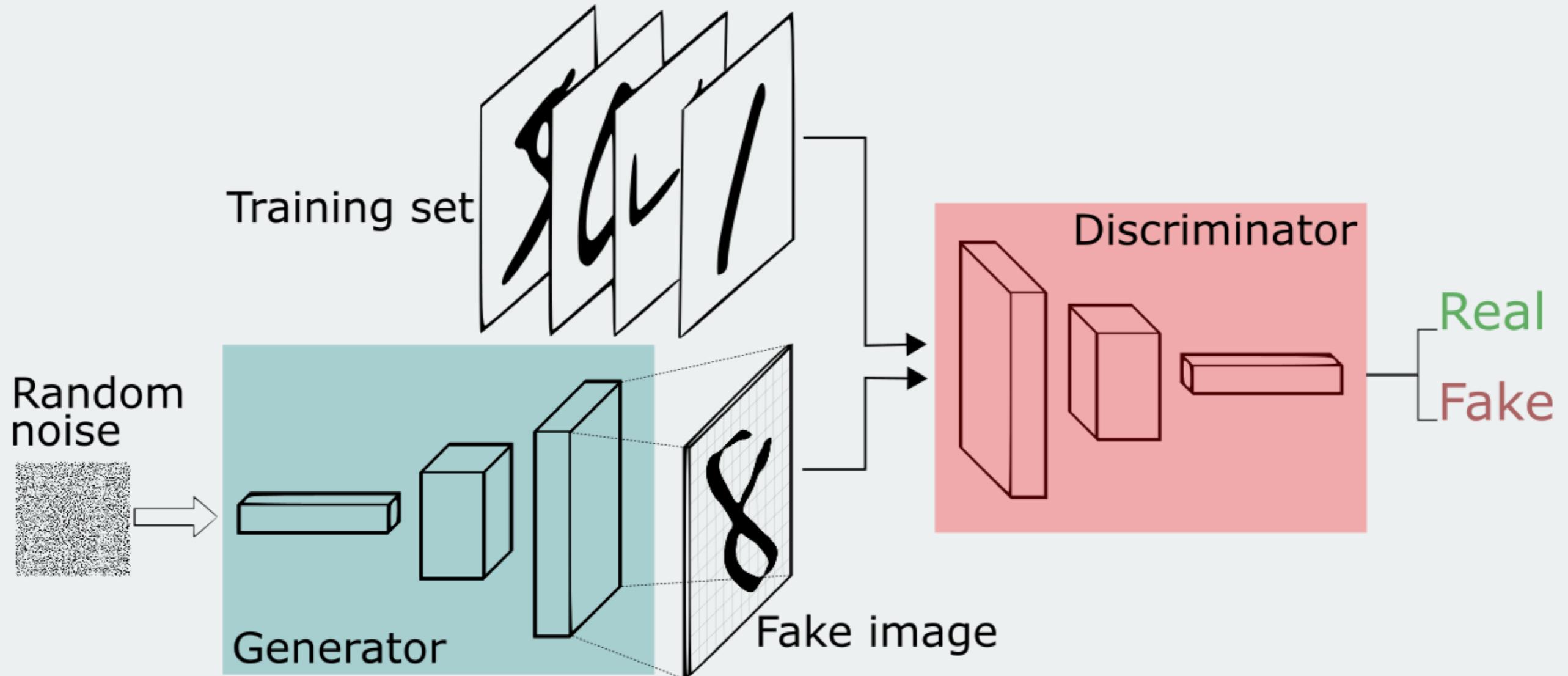




# План на сегодня

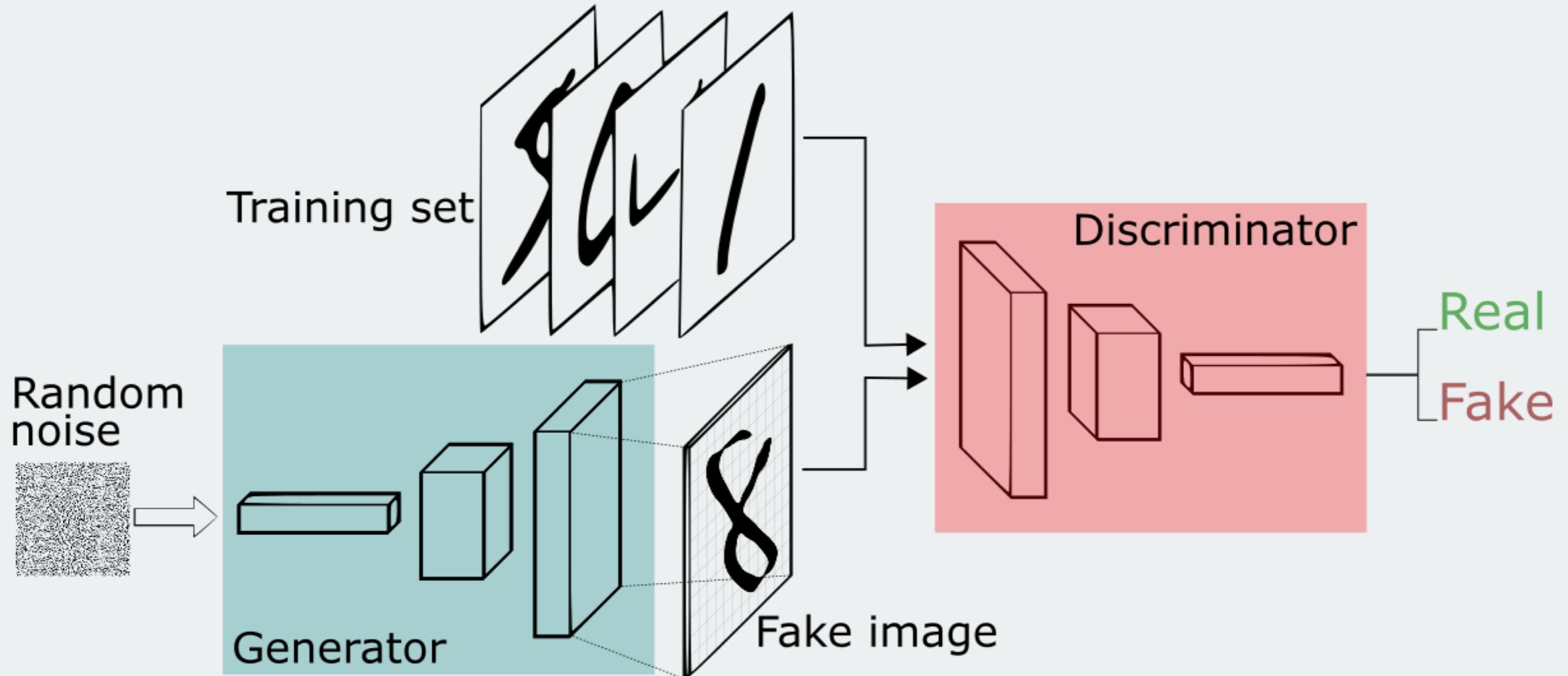
1. Интроверт
2. **Generative Adversarial Nets (GANs)**
3. Информация и Адаптация
4. Честная условная генерация
5. Тексты, подкрепление и профессор
6. Другие трюки

# Генеративные Состязательные Сети



<https://medium.com/ml-everything/generating-letters-using-generative-adversarial-networks-gans-161b0be3c229>

# Генеративные Состязательные Сети



$$\mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x \sim p_{gen}(x)}[\log(1 - D(x))]$$

<https://medium.com/ml-everything/generating-letters-using-generative-adversarial-networks-gans-161b0be3c229>

# Генеративные Состязательные Сети



$$\mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x \sim p_{gen}(x)}[\log(1 - D(x))]$$

$x \sim p_{gen}(x)$  – это то же самое, что и  $G(z), z \sim p_z$

# Генеративные Состязательные Сети



$$\mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x \sim p_{gen}(x)}[\log(1 - D(x))]$$

$x \sim p_{gen}(x)$  – это то же самое, что и  $G(z), z \sim p_z$

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

С одной стороны мы хотим максимизировать точность

Дискриминатора, с другой – максимизировать шанс Генератора «обмануть» Дискриминатор. Поэтому ВАЖНО обучать их по отдельности.

# Проблемы GANs



Устойчивость: минимаксная игра — поиск седловой точки

# Проблемы GANs



Устойчивость: минимаксная игра — поиск седловой точки

Mode collapse: часто Генератору оказывается достаточно генерировать только часть датасета

# Проблемы GANs



Устойчивость: минимаксная игра — поиск седловой точки

Mode collapse: часто Генератору оказывается достаточно генерировать только часть датасета

Насыщение Дискриминатора: градиенты от хорошего Дискриминатора могут быть нулевыми



# Проблемы GANs

Устойчивость: минимаксная игра — поиск седловой точки

Mode collapse: часто Генератору оказывается достаточно генерировать только часть датасета

Насыщение Дискриминатора: градиенты от хорошего Дискриминатора могут быть нулевыми

Обучить «ванильный» ГАН с нуля даже генерации MNIST'а задача нетривиальная, однако именно эта статья наделала очень много шума.



# Нейросети как искусство

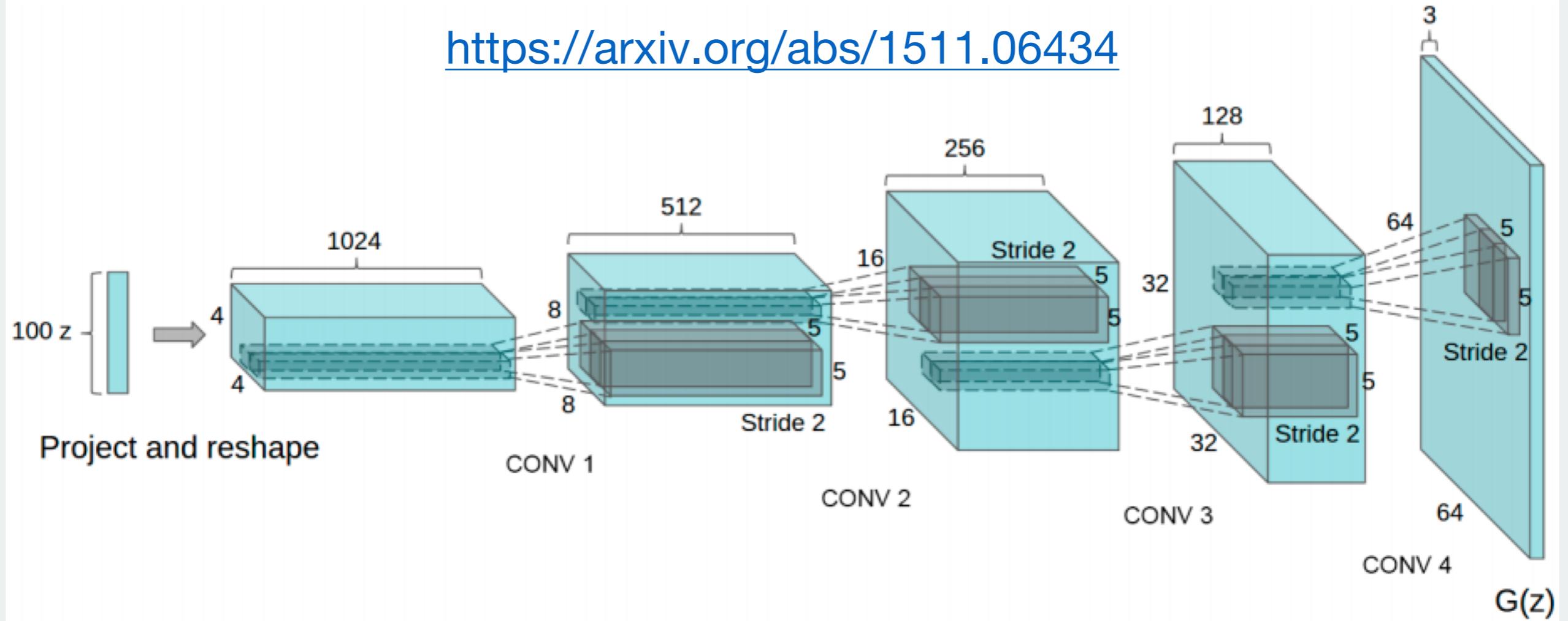
В 2016 году Ян ЛеКун назвал GAN'ы самой крутой идеей в машинном обучении за последние 20 лет. А в 2018 году портрет «написанный» с помощью GAN'ов был продан за \$432,500 на аукционе Christie's



# Deep Convolutional GANs



<https://arxiv.org/abs/1511.06434>



1. Ни каких пуллинг слоев как в Генераторе, так и в Дискриминаторе
2. Нет полно связных слоев
3. Batch Normalization на всех слоях кроме выхода  $G$  и входа  $D$
4. ReLU в Генераторе и LeakyReLU в Дискриминаторе

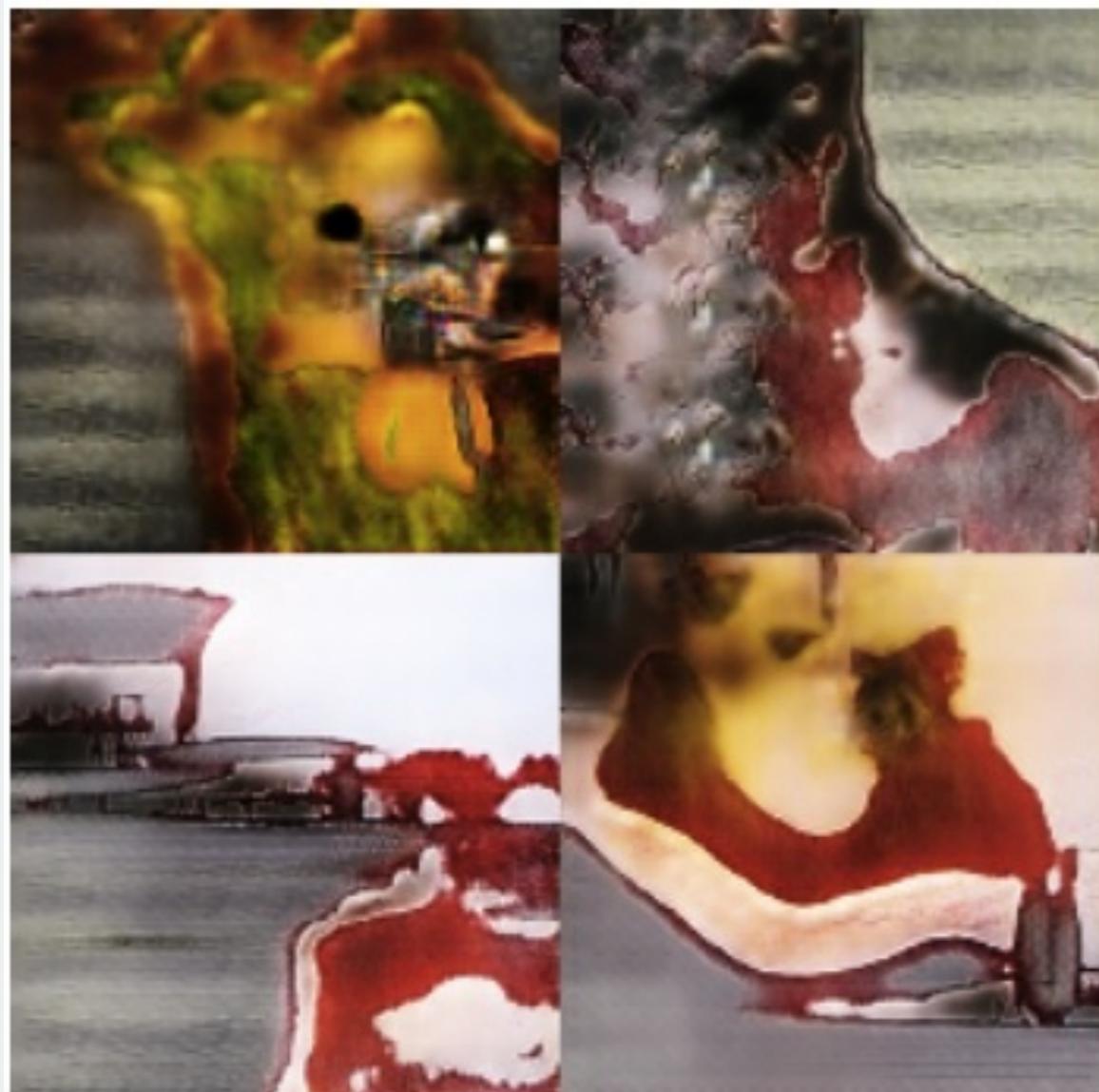


# Deep Convolutional GANs

<https://arxiv.org/abs/1511.06434>



# Deep Convolutional GANs



DCGAN



Improved DCGAN

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



# План на сегодня

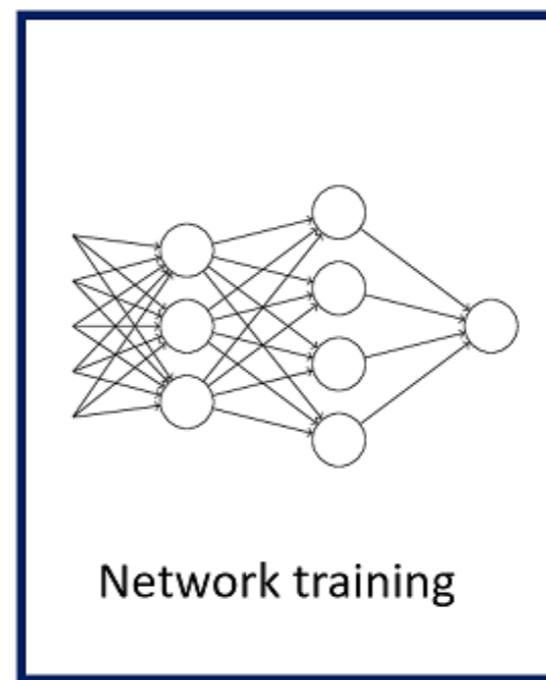
1. Интровер
2. Generative Adversarial Nets (GANs)
- 3. Информация и Адаптация**
4. Честная условная генерация
5. Тексты, подкрепление и профессор
6. Другие трюки



# Информация

0 0 0 0 0 0 0 0 0 0 0 0 0  
1 1 1 1 1 1 1 1 1 1 1 1 1  
2 2 2 2 2 2 2 2 2 2 2 2 2 0  
3 3 3 3 3 3 3 3 3 3 3 3 3 3  
4 4 4 4 4 4 4 4 4 4 4 4 4 4  
5 5 5 5 5 5 5 5 5 5 5 5 5 5  
6 6 6 6 6 6 6 6 6 6 6 6 6 6  
7 7 7 7 7 7 7 7 7 7 7 7 7 1  
8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9

Data & Labels



0  
1  
2  
3  
4  
5  
6  
7  
8  
9

A vertical column of large black numbers from 0 to 9, each enclosed in a small curly brace that groups them into two columns: {0,1,2,3} and {4,5,6,7,8,9}. An arrow points upwards from the bottom of the first brace towards the top of the second.

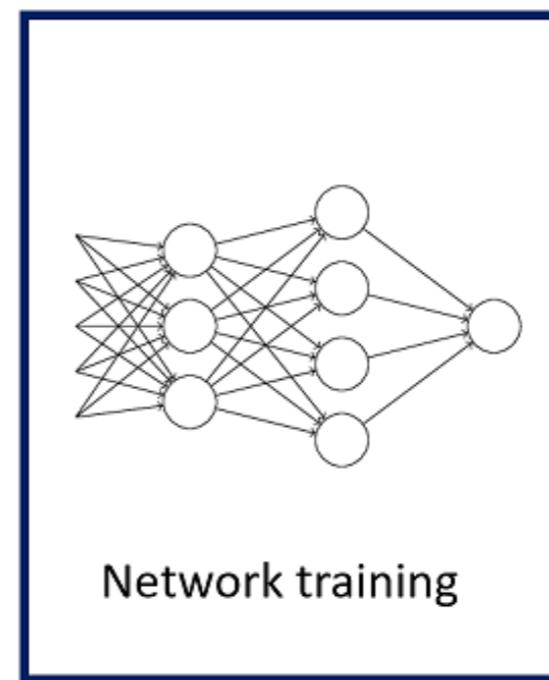


# Информация

$$D_{KL}(P||Q) = H(P, Q) - H(P) = \\ = \sum_i p_i \log p_i - \sum_i p_i \log q_i = \sum_i p_i \log \frac{p_i}{q_i}$$

0 0 0 0 0 0 0 0 0 0 0 0 0  
1 1 1 1 1 1 1 1 1 1 1 1 1  
2 2 2 2 2 2 2 2 2 2 2 2 2 0  
3 3 3 3 3 3 3 3 3 3 3 3 3 3  
4 4 4 4 4 4 4 4 4 4 4 4 4 4  
5 5 5 5 5 5 5 5 5 5 5 5 5 5  
6 6 6 6 6 6 6 6 6 6 6 6 6 6  
7 7 7 7 7 7 7 7 7 7 7 7 7 1  
8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9

Data & Labels



0  
1  
2  
3  
4  
5  
6  
7  
8  
9



# Доменная адаптация

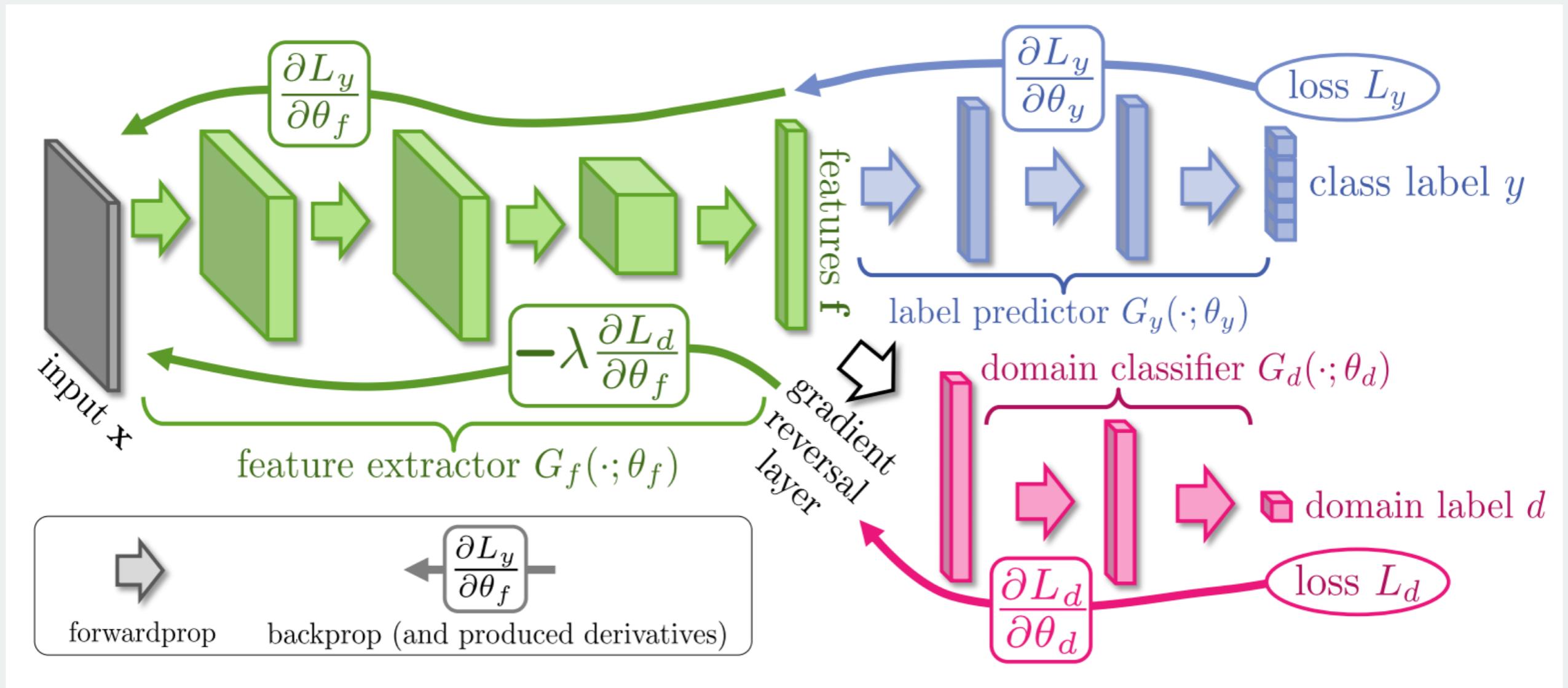
|        | MNIST   | SYN NUMBERS | SVHN  | SYN SIGNS |
|--------|---------|-------------|-------|-----------|
| SOURCE |         |             |       |           |
| TARGET |         |             |       |           |
|        | MNIST-M | SVHN        | MNIST | GTSRB     |

В реальной жизни часто возникает проблема с недостатком размеченных данных. В этом случае одним из подходов может быть адаптация сети обученной на большом публичном датасете к целевому. Спустя 3 месяца после появления в сети статьи Яна Гудфеллоу, Ярослав Ганин и Виктор Лемпицкий выложили на arXiv свою статью о доменной адаптации с аналогичной идеей.

Unsupervised Domain Adaptation by Backpropagation: <https://arxiv.org/abs/1409.7495>



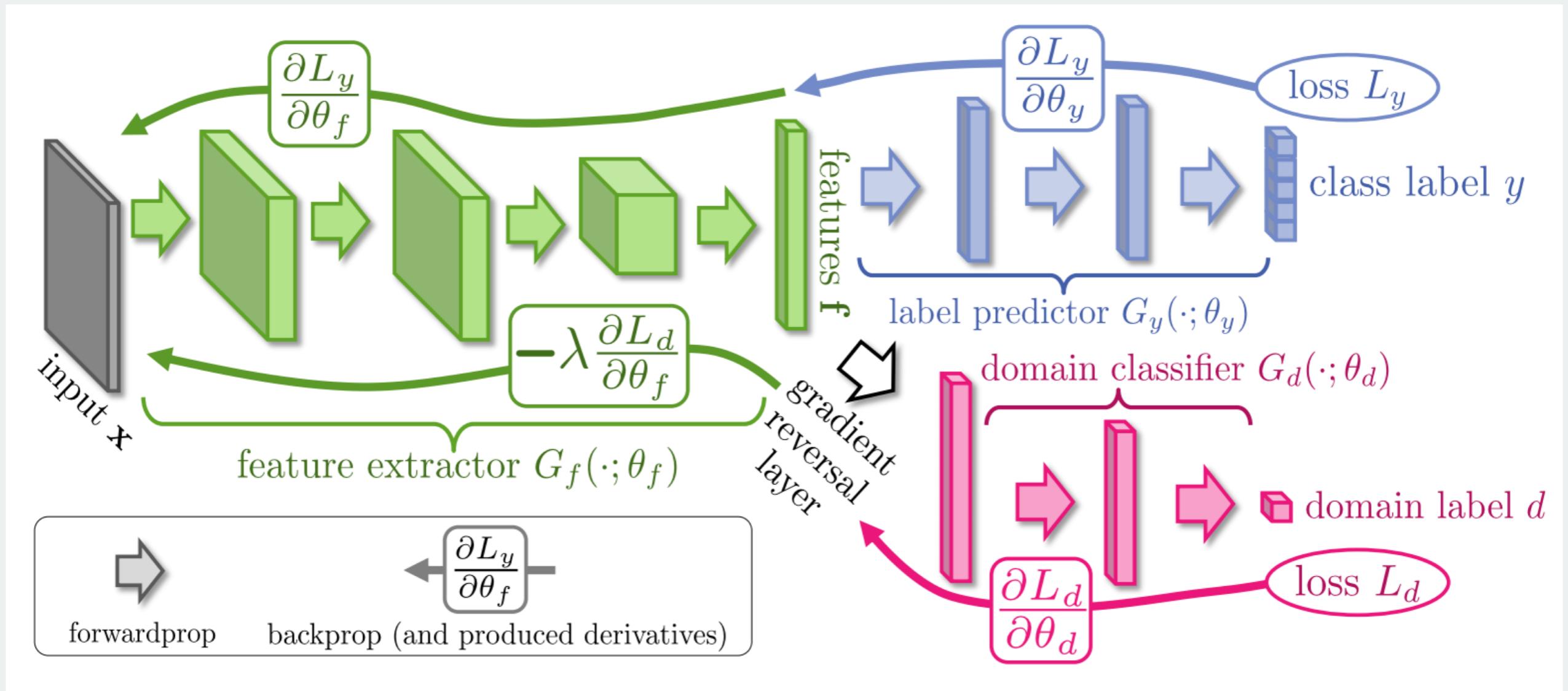
# Доменная адаптация



Unsupervised Domain Adaptation by Backpropagation: <https://arxiv.org/abs/1409.7495>



# Доменная адаптация

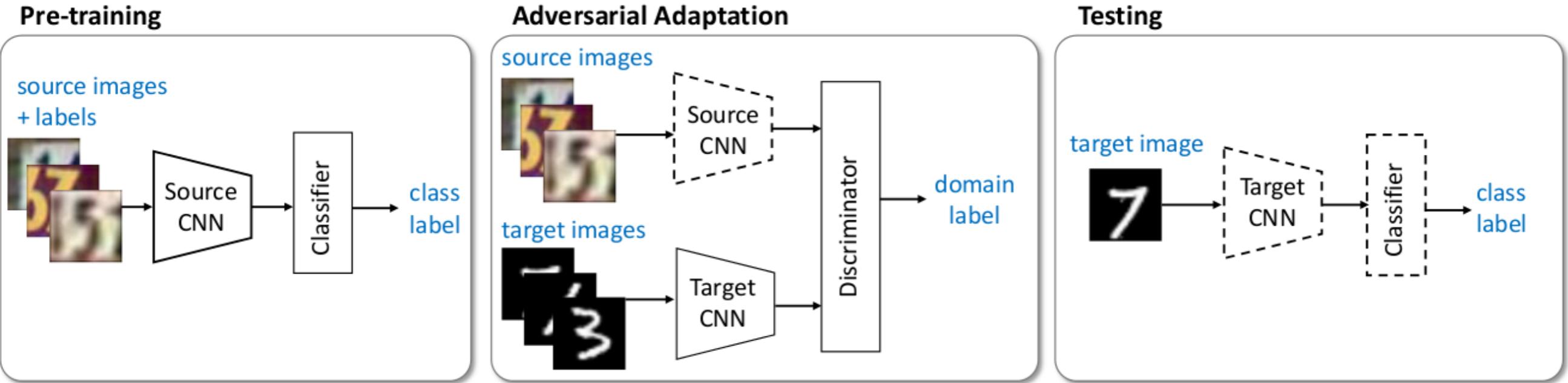


Впрочем, это не единственный способ решения задачи адаптации

Unsupervised Domain Adaptation by Backpropagation: <https://arxiv.org/abs/1409.7495>



# Доменная адаптация



В 2017 году появилась статья, в которой эта задача решается последовательно. Обучив нейронную сеть на датасете с большим количеством размеченных данных и «заставив» ее «не замечать» различия между разными датасетами мы можем ее использовать на целевых данных.

Adversarial Discriminative Domain Adaptation: <https://arxiv.org/abs/1702.05464>

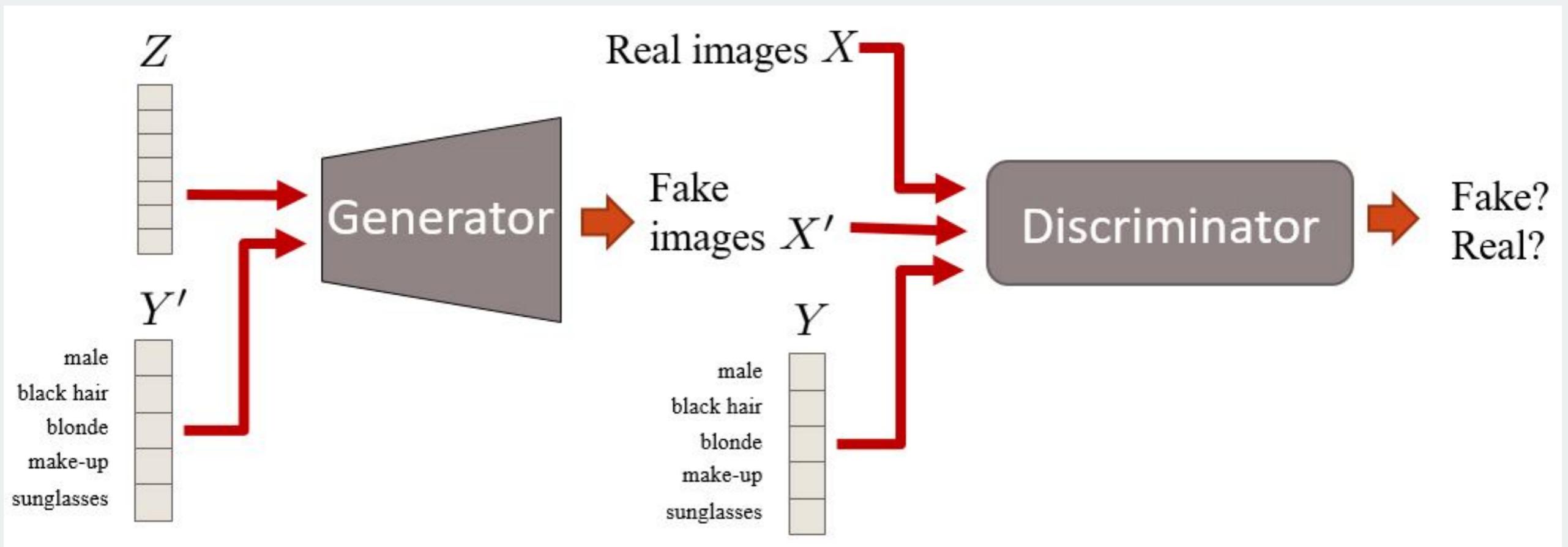


# План на сегодня

1. Интроверт
2. Generative Adversarial Nets (GANs)
3. Информация и Адаптация
- 4. Честная условная генерация**
5. Тексты, подкрепление и профессор
6. Другие трюки



# Условная генерация

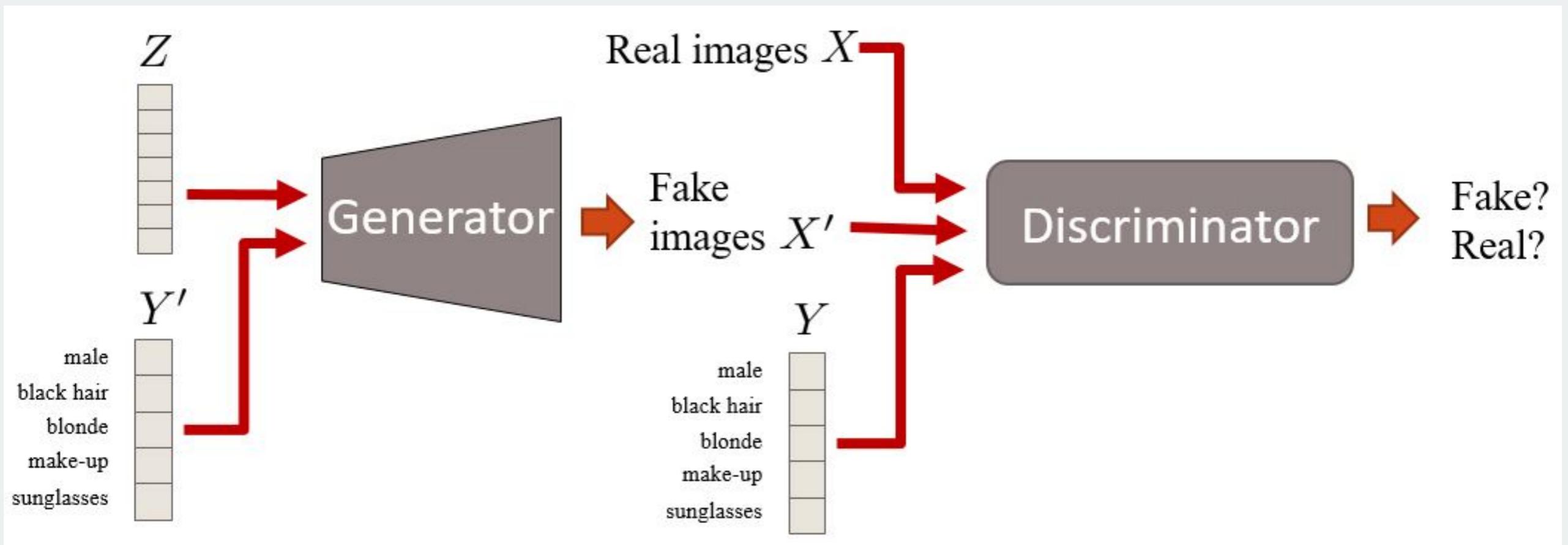


Конечно, генерировать совсем произвольное изображение, пусть даже и очень реалистичное, нам не интересно. Почти всегда хочется наложить какие-либо условия на то что должно получиться на выходе.

Conditional Generative Adversarial Nets: <https://arxiv.org/abs/1411.1784>  
<http://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them>



# Условная генерация



**Что будет если не подавать метку на вход Дискриминатору?**

Conditional Generative Adversarial Nets: <https://arxiv.org/abs/1411.1784>  
<http://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them>



# Условная генерация

**Что будет если не подавать метку на вход Дискриминатору?**

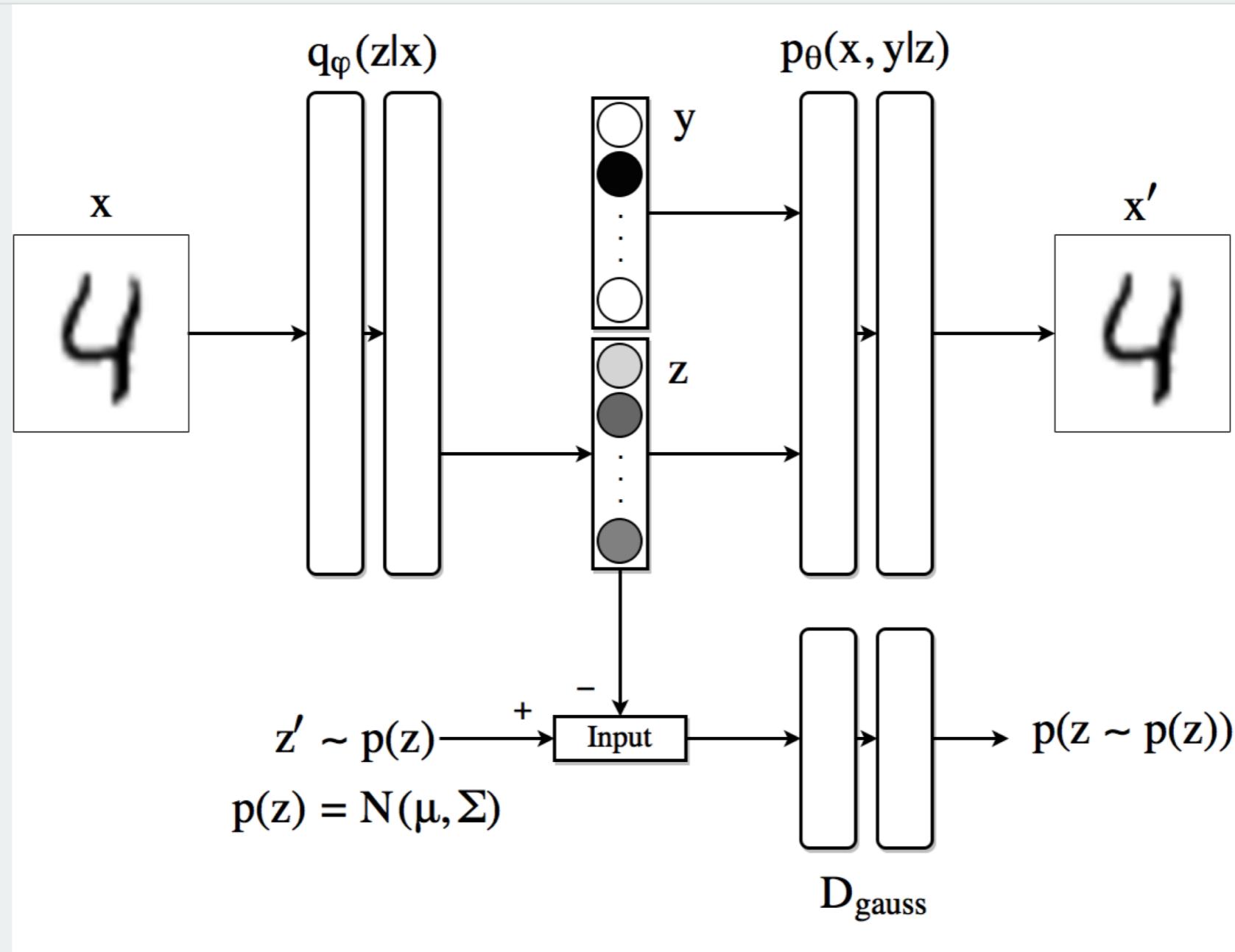
Есть тривиальное решение:

**Обнулить соответствующие метке веса первого слоя генератора и обучить обычный GANs!**

Т.е., если использовать метку при генерации сложнее чем не использовать, модель может не использовать ее. По факту, на MNIST этого не происходит, но на сложных датасетах вполне может произойти.



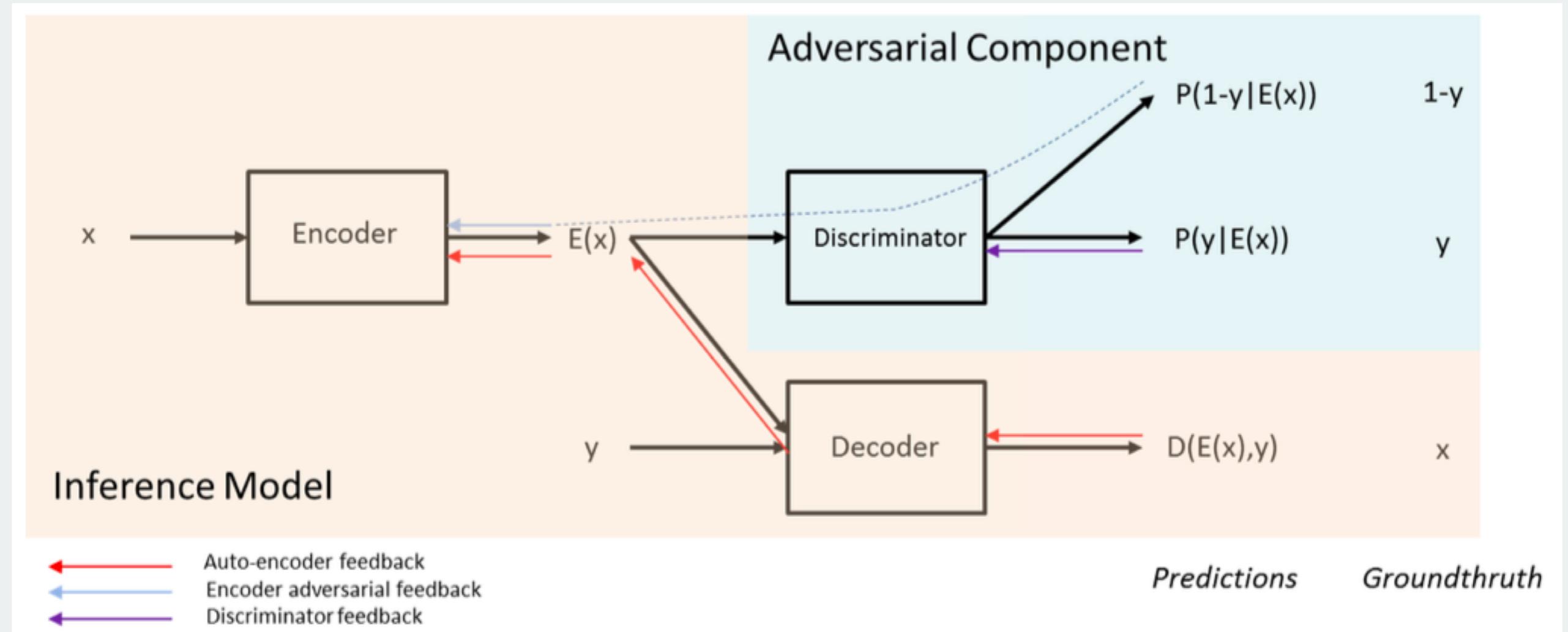
# Состязательный Автокодировщик



Adversarial Autoencoders: <https://arxiv.org/abs/1511.05644>



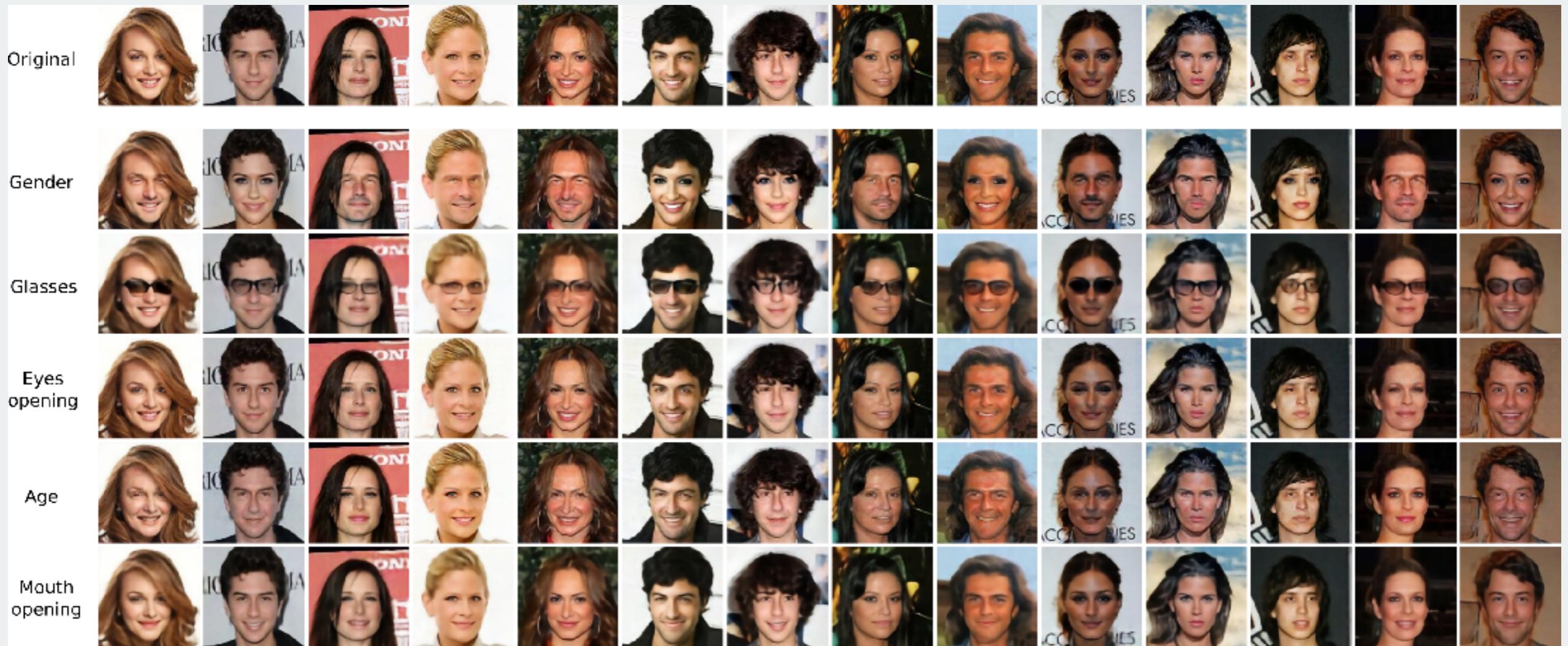
# FADER Networks



Для того чтобы исключить информацию о метке из латентного представления команда из Facebook использовала Дискриминатор. То есть, проверку на то, можно ли предсказать метку по латентному коду.

Fader Networks: Manipulating Images by Sliding Attributes: <https://arxiv.org/abs/1706.00409>

# FADER Networks

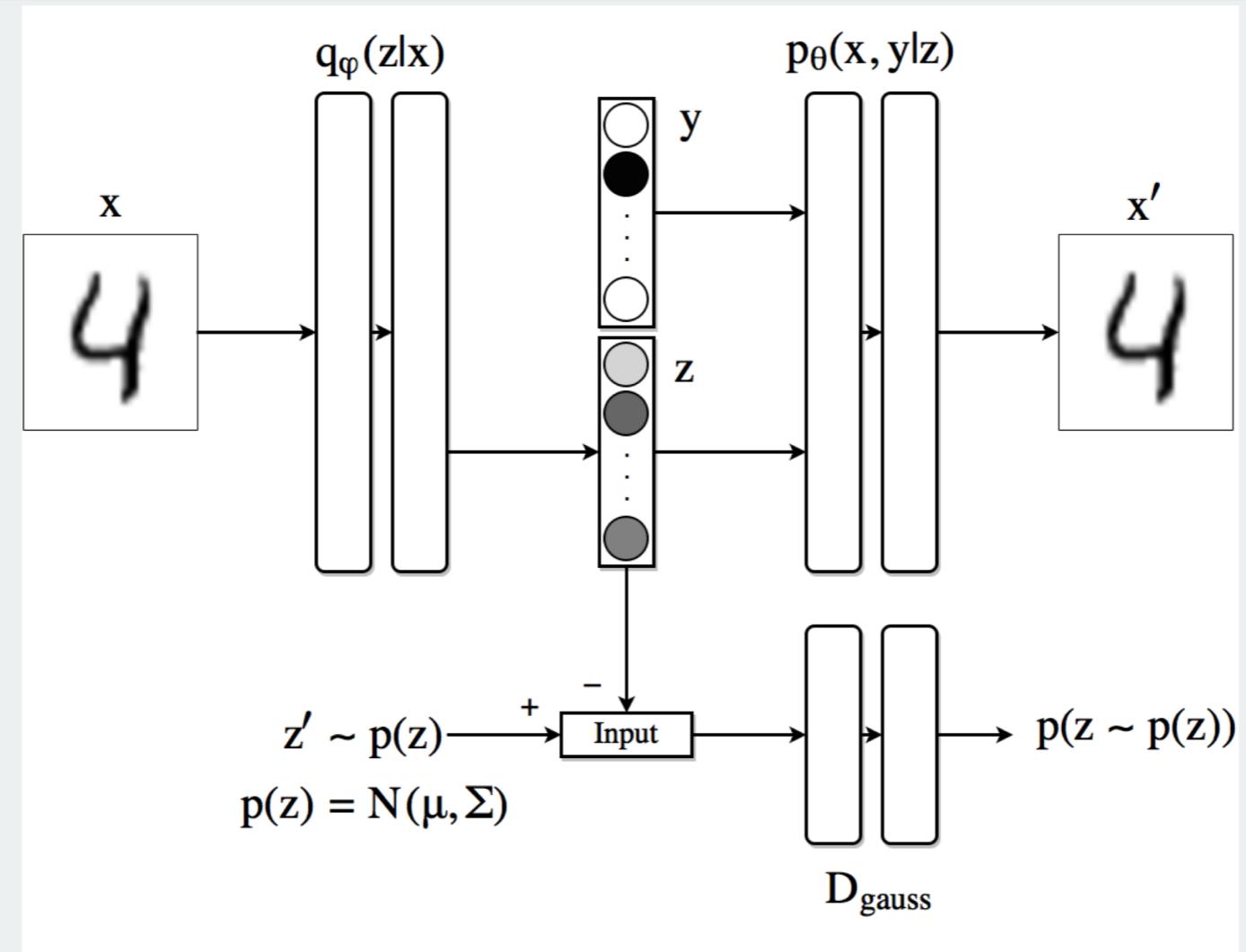


Для того чтобы исключить информацию о метке из латентного представления команда из Facebook использовала Дискриминатор. То есть, проверку на то, можно ли предсказать метку по латентному коду.

Fader Networks: Manipulating Images by Sliding Attributes: <https://arxiv.org/abs/1706.00409>

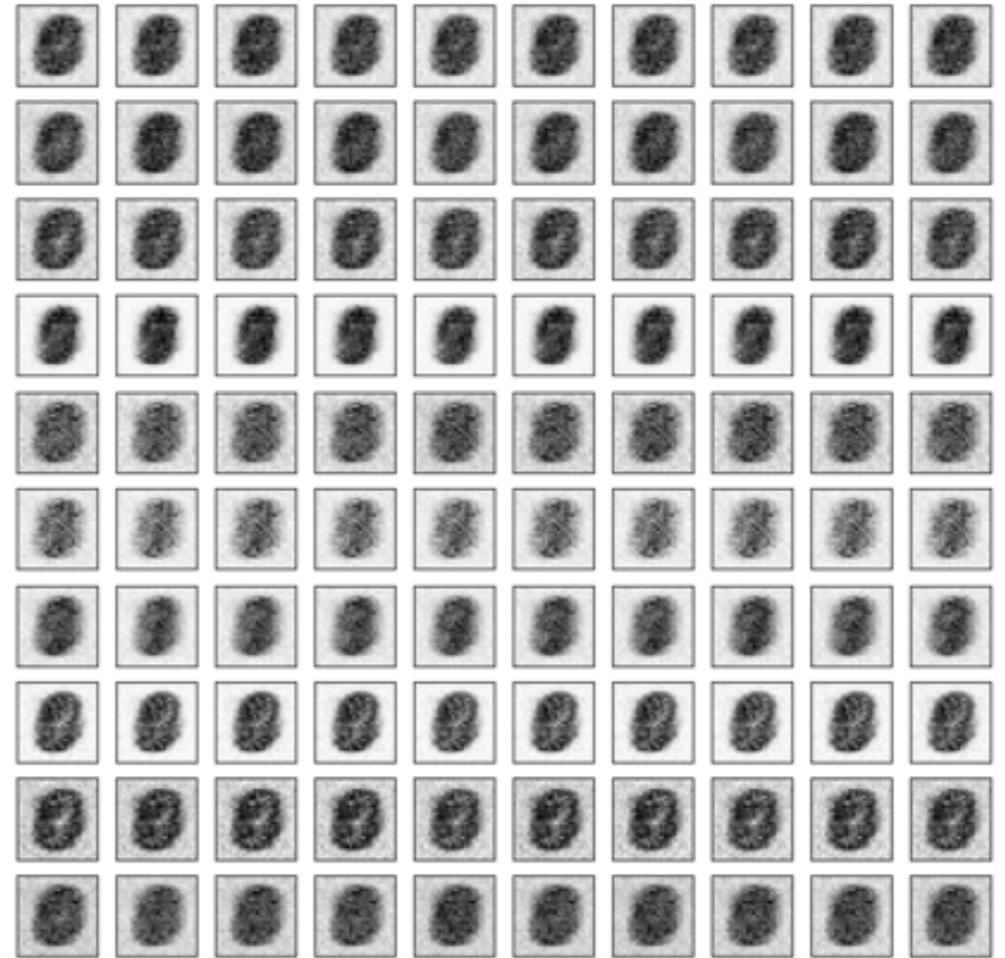
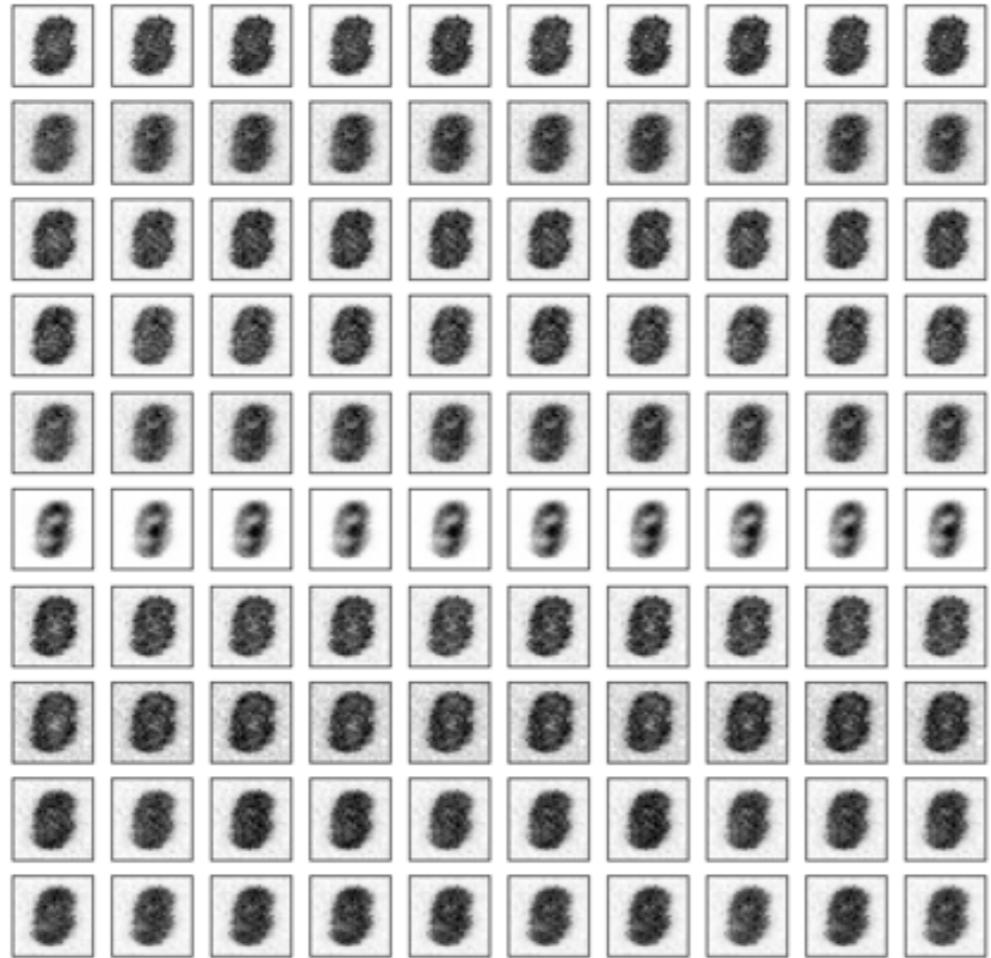


# Состязательный Автокодировщик



Альтернативным способом будет подавать метку на Дискриминатор. Для этого в качестве отрицательного примера надо использовать латентный код и метку входного изображения, а в качестве положительного независимо сэмплированные пары из приорных распределений.

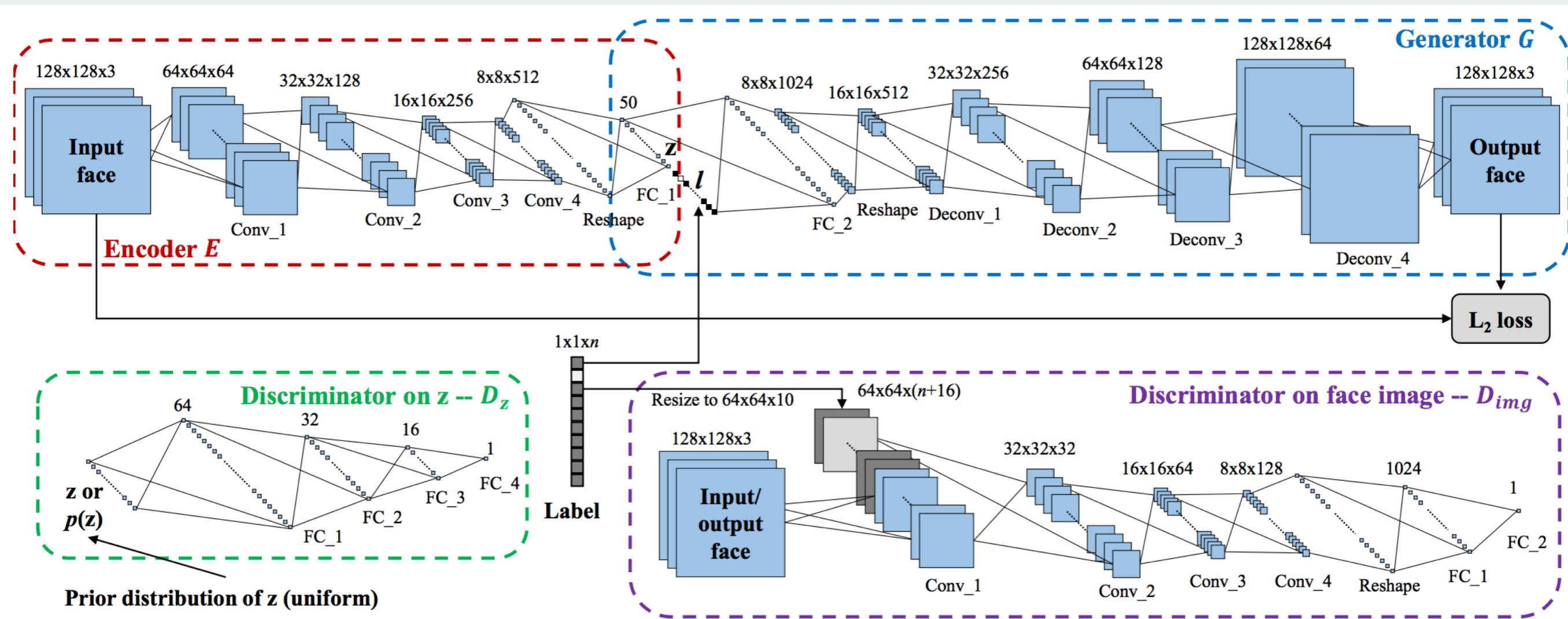
# Состязательный Автокодировщик





# Age progression/regression

Немного другой подход к «распутыванию» предложила команда из университета Теннеси.



Age Progression/Regression by Conditional Adversarial Autoencoder: <https://arxiv.org/abs/1702.08423>



# Age progression/regression



Age Progression/Regression by Conditional Adversarial Autoencoder: <https://arxiv.org/abs/1702.08423>



# План на сегодня

1. Интроверт
2. Generative Adversarial Nets (GANs)
3. Информация и Адаптация
4. Честная условная генерация
- 5. Тексты, подкрепление и профессор**
6. Другие трюки

# Генерация текстов



**Что будет если в DCGAN заменить свертки рекуррентками и начать генерировать тексты?**



# Генерация текстов

**Что будет если в DCGAN заменить свертки рекуррентками и начать генерировать тексты?**

Сделать так в лоб не получится, в связи с тем, что текст, в отличии от картинки дискретен. Это не даст пропустить градиент сквозь сгенерированные объекты между Генератором и Дискриминатором.

**Что делать?**



# Генерация текстов

**Что будет если в DCGAN заменить свертки рекуррентками и начать генерировать тексты?**

Сделать так в лоб не получится, в связи с тем, что текст, в отличии от картинки дискретен. Это не даст пропустить градиент сквозь сгенерированные объекты между Генератором и Дискриминатором.

**Что делать?**

1. Одним из простых решений будет переход от GANs к AAE. Выносим дискретные объекты наружу и все ок.



# Генерация текстов

**Что будет если в DCGAN заменить свертки рекуррентками и начать генерировать тексты?**

Сделать так в лоб не получится, в связи с тем, что текст, в отличии от картинки дискретен. Это не даст пропустить градиент сквозь сгенерированные объекты между Генератором и Дискриминатором.

**Что делать?**

1. Одним из простых решений будет переход от GANs к AAE. Выносим дискретные объекты наружу и все ок.
2. Второе и очень важное решение было предложено в 2016г в статье «Categorical Reparameterization with Gumbel-Softmax».



# Генерация текстов

**Что будет если в DCGAN заменить свертки рекуррентками и начать генерировать тексты?**

Сделать так в лоб не получится, в связи с тем, что текст, в отличии от картинки дискретен. Это не даст пропустить градиент сквозь сгенерированные объекты между Генератором и Дискриминатором.

**Что делать?**

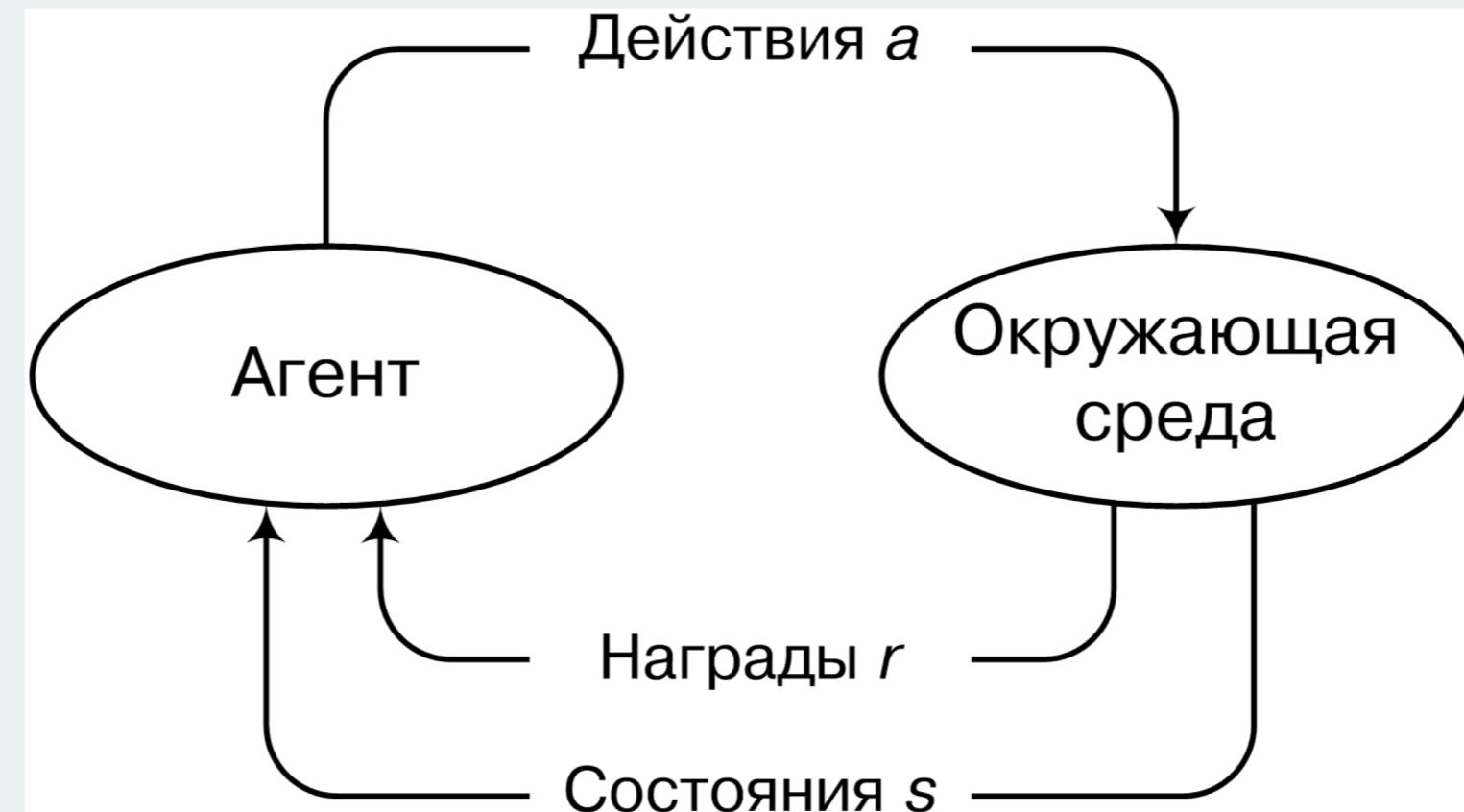
1. Одним из простых решений будет переход от GANs к AAE. Выносим дискретные объекты наружу и все ок.
2. Второе и очень важное решение было предложено в 2016г в статье «Categorical Reparameterization with Gumbel-Softmax».
3. И, наконец, давайте использовать RL!



# Обучение с подкреплением

Обучение с учителем: Для заранее заданного набора объектов известны «правильные» ответы. Нужно научиться предсказывать ответы для новых объектов.

Обучение с подкреплением: Можно совершить действие и узнать ответ. Нужно научиться совершать действия приводящие к «лучшим» ответам.





# Обучение с подкреплением

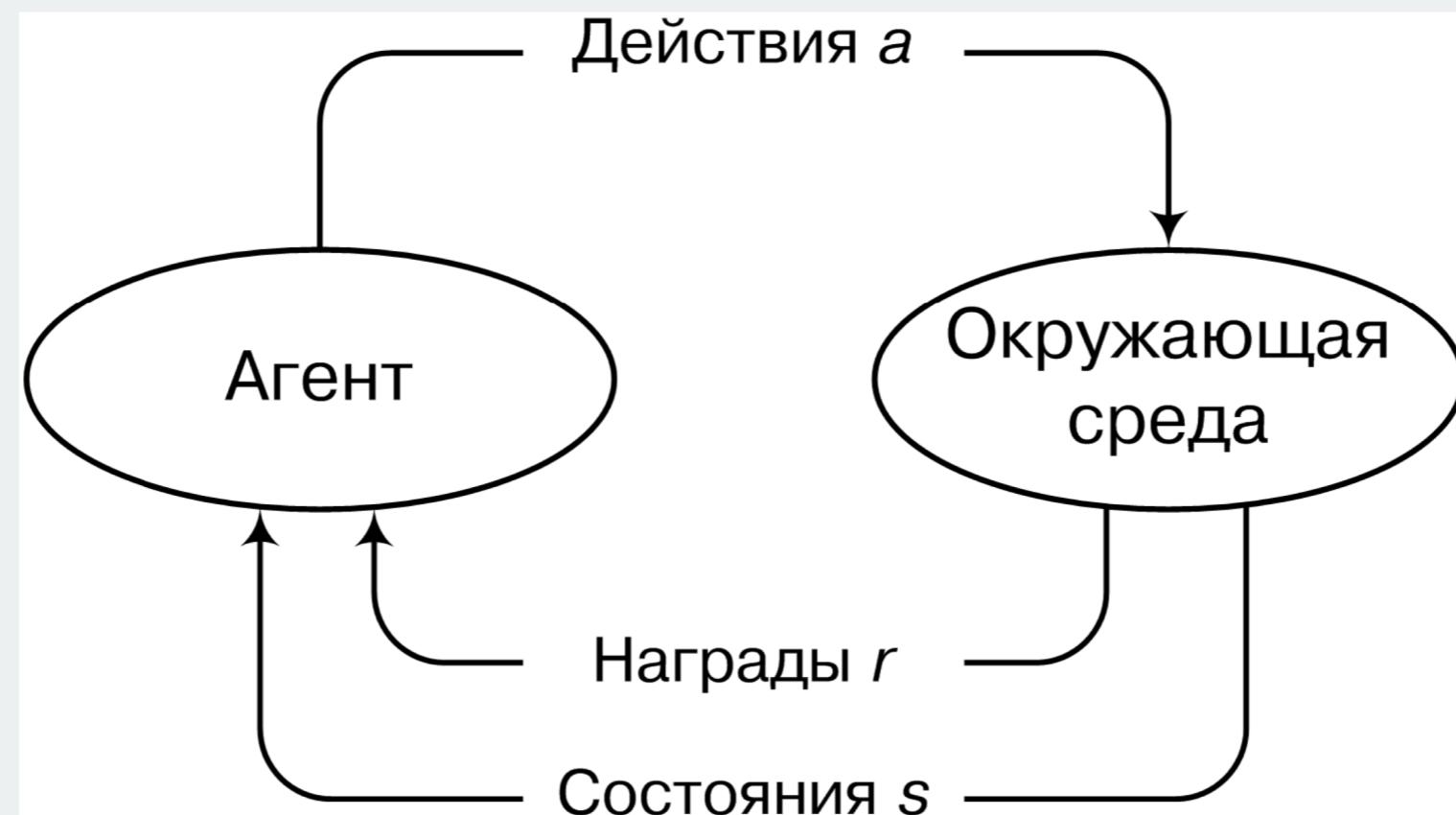
Марковский процесс принятия решений состоит из:

Множества состояний  $S$

Множества действий  $A$

Функции вознаграждения  $R: S \times A \rightarrow \mathbb{R}$ , которая задает  $R_{ss}^a$ ,

Функции перехода между состояниями  $p_s^a: S \times A \rightarrow \Pi(S)$ , которая задает  $P_{ss}^a$ ,

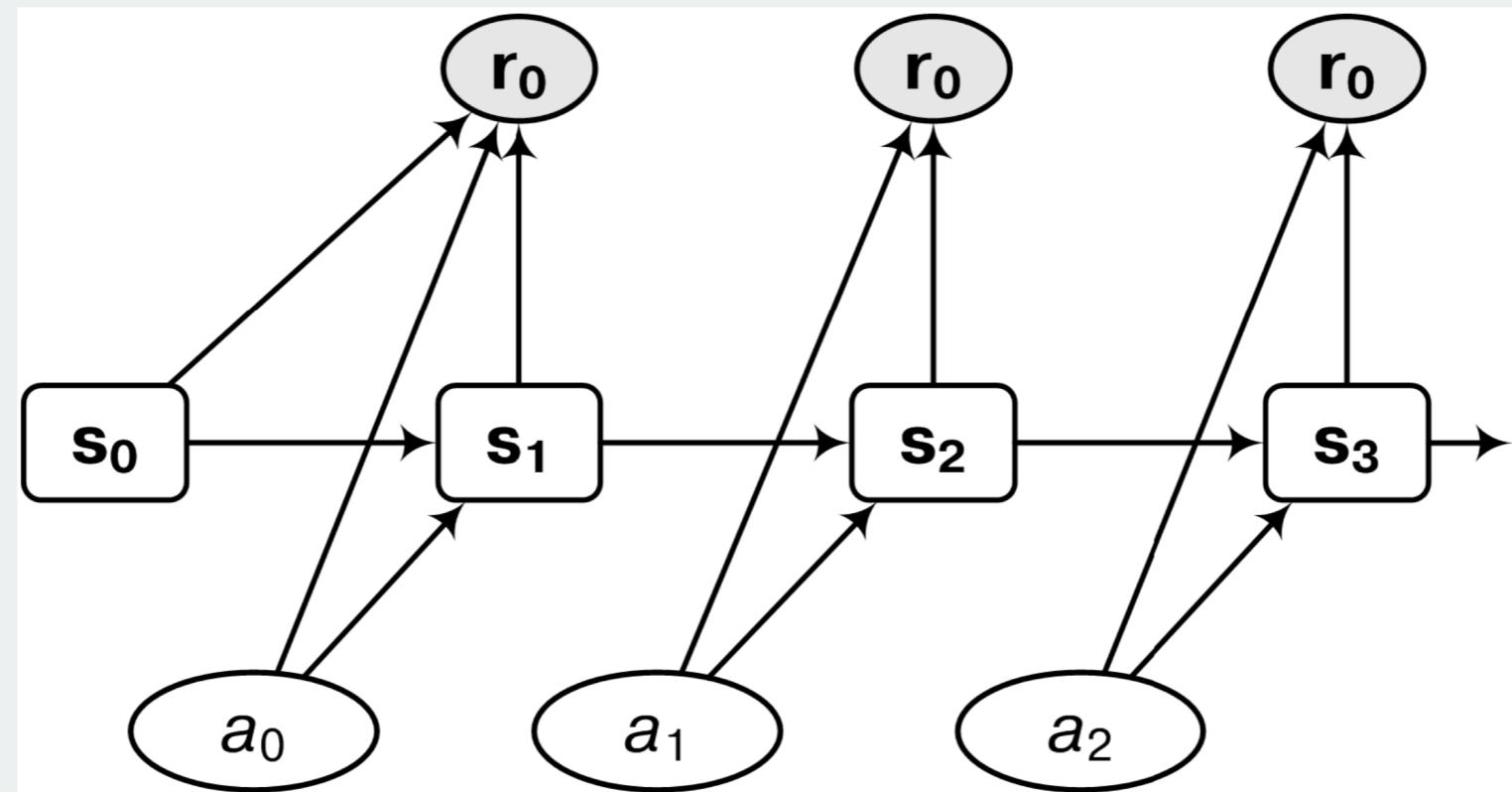




# Обучение с подкреплением

Марковское свойство заключается в том, что вероятности переходов между состояниями при выбранных действиях не зависят от истории предыдущих переходов.

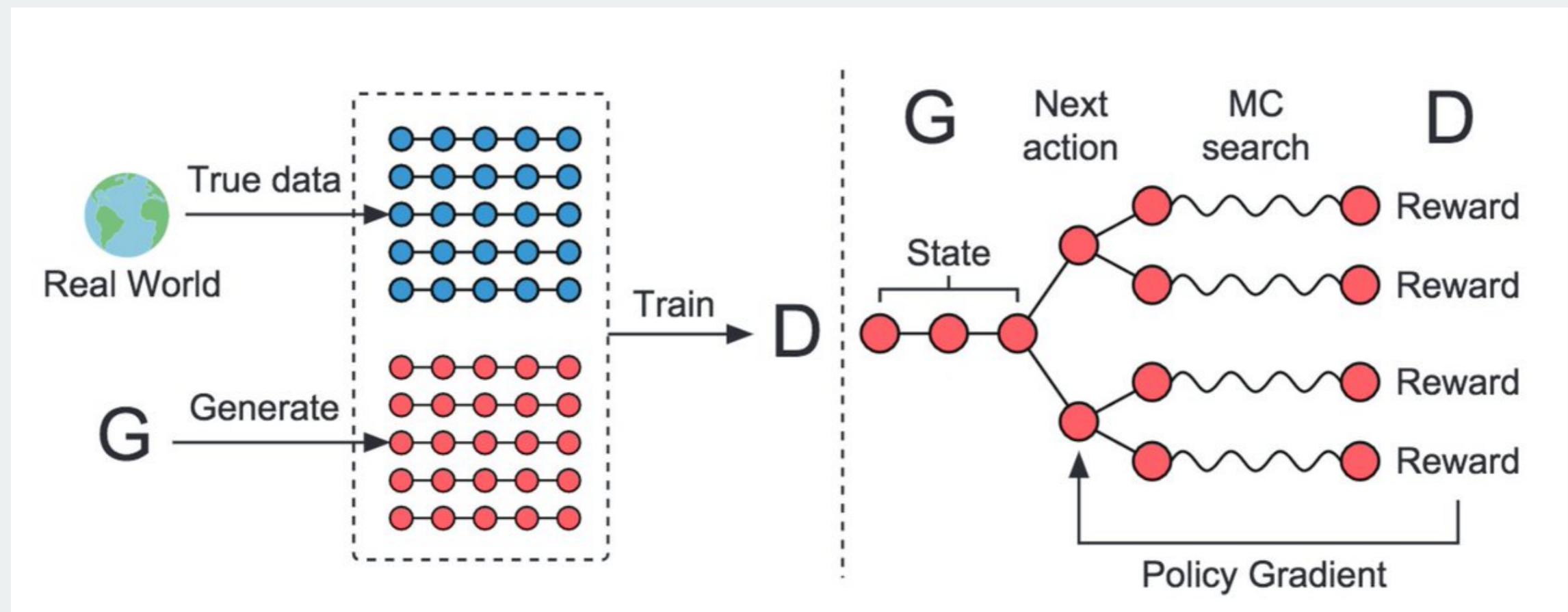
А учиться совершать лучшее действие можно, например, предсказывая ценность каждого из возможных действий в текущем состоянии.



# SeqGAN



Давайте тогда переформулируем генерацию последовательности букв в задачу выбора очередного действия в заданной ситуации и придумаем как оценивать «ценность» этого действия.



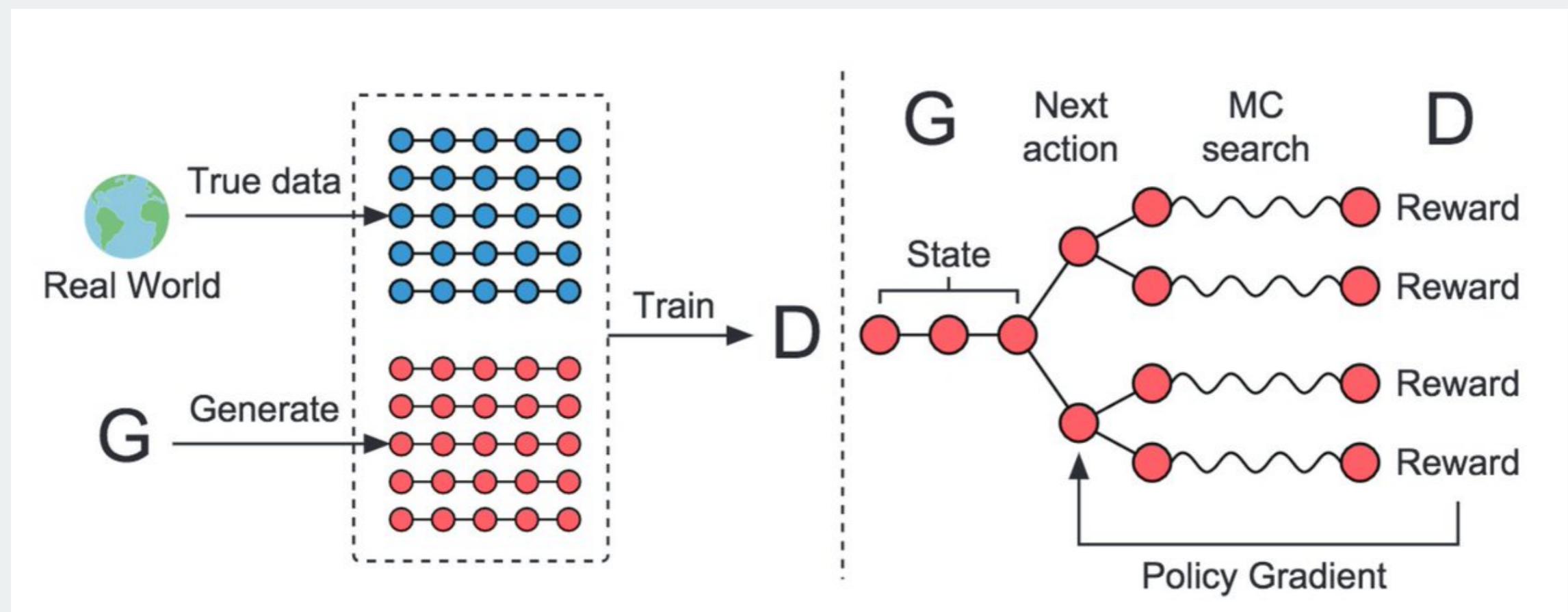
SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient: <https://arxiv.org/abs/1609.05473>



# SeqGAN

Давайте тогда переформулируем генерацию последовательности букв в задачу выбора очередного действия в заданной ситуации и придумаем как оценивать «ценность» этого действия.

В SeqGAN на каждом шаге генератор предсказывает распределение вероятностей выбора очередной буквы. Мы можем для каждой буквы посемплировать всю последовательность до конца и посчитать оценку.

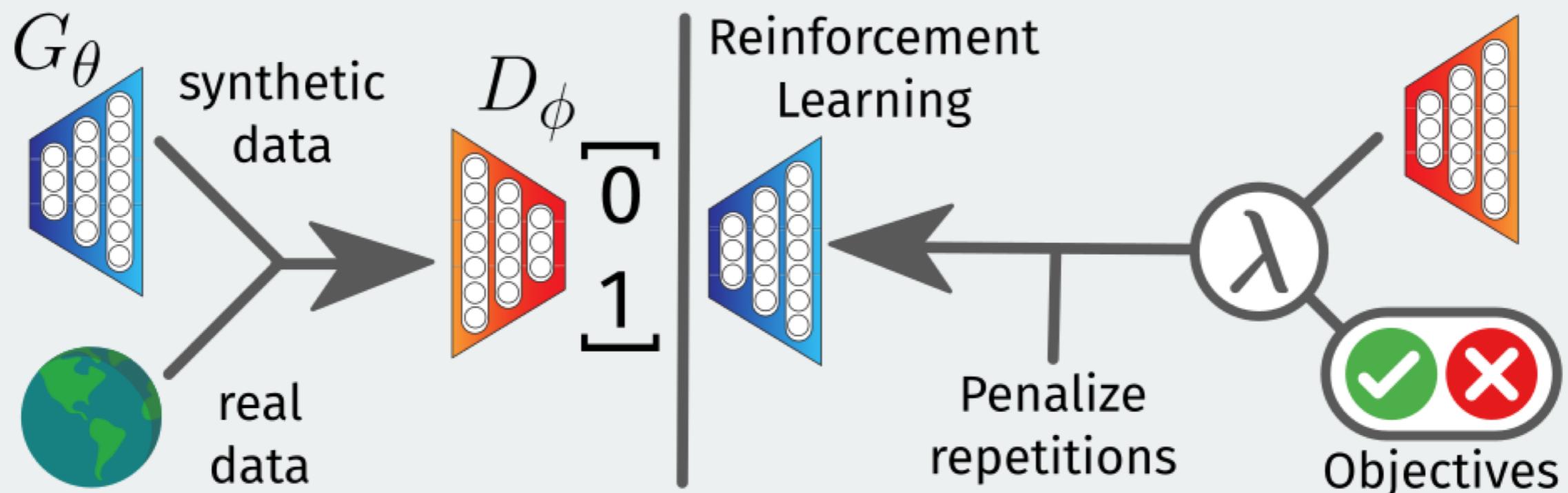


SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient: <https://arxiv.org/abs/1609.05473>



# ORGAN

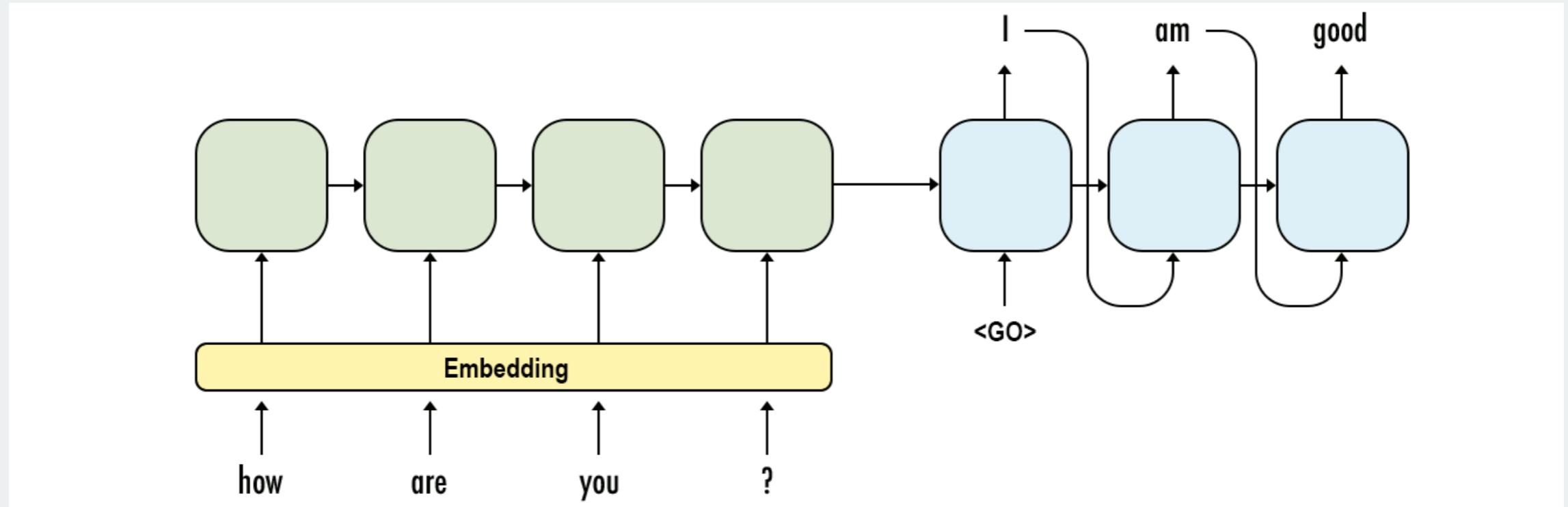
Интересно, что, раз уж ценность действия мы оцениваем с помощью какой-то внешней функции, то мы можем добавить к ней дополнительные слагаемые. Это может быть, например, автоматическая проверка валидности объекта, ассессорская оценка или, вообще, разнообразие сгенерированной выборки.



Objective-Reinforced Generative Adversarial Networks (ORGAN): <https://arxiv.org/abs/1705.10843>



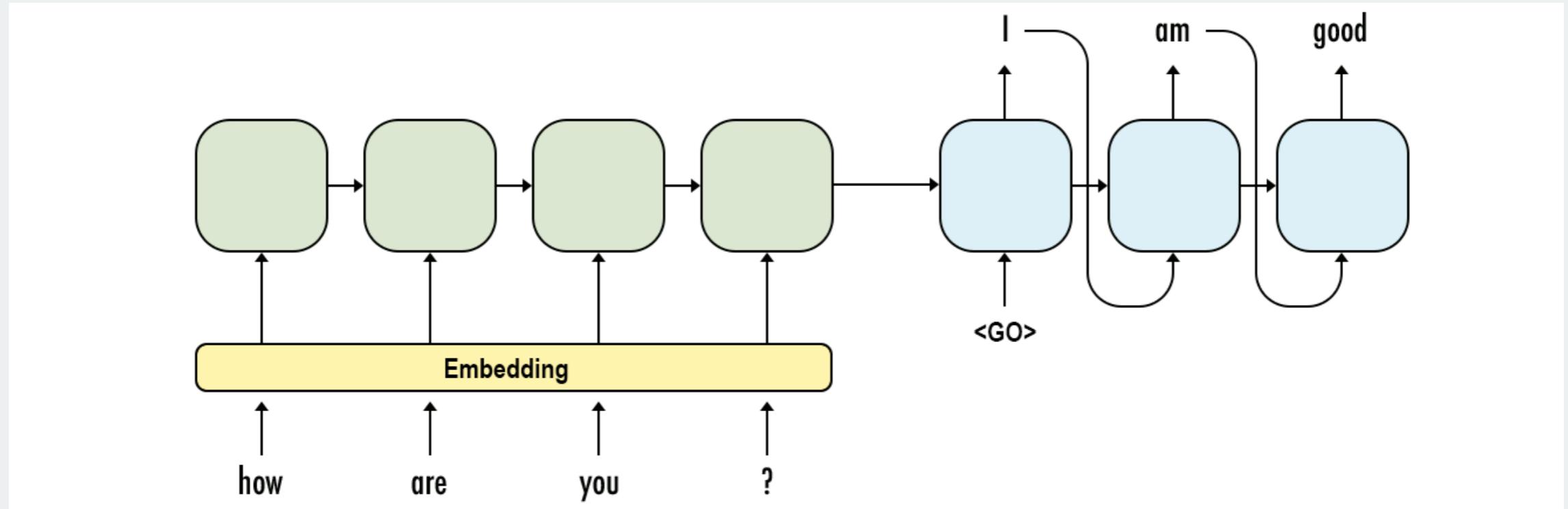
# Teacher Forcing



Когда мы учим обыкновенную рекуррентную нейронную сеть порождать символ за символом, выход на очередном шаге является входом для следующего. Однако это приводит к проблемам. **Каким?**



# Teacher Forcing

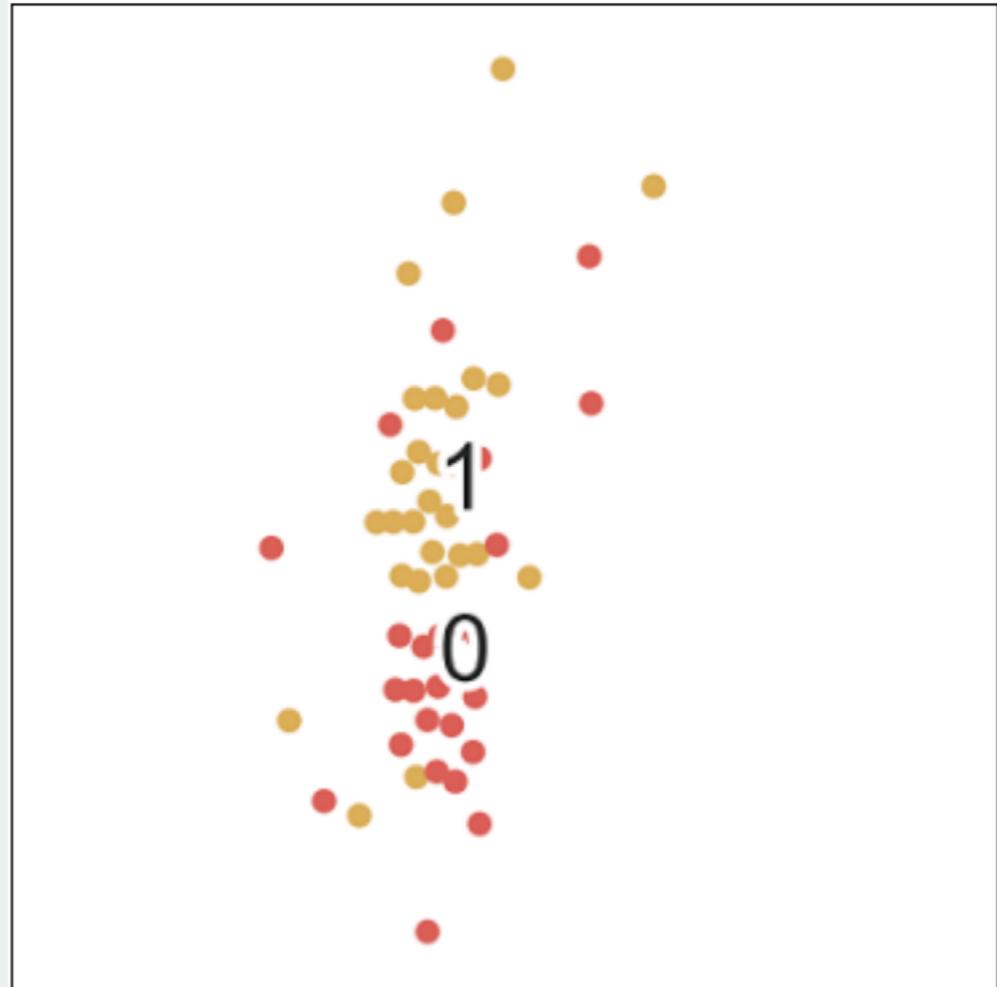


Когда мы учим обыкновенную рекуррентную нейронную сеть порождать символ за символом, выход на очередном шаге является входом для следующего. Однако это приводит к проблемам. **Каким?**

Поэтому, зачастую, вместо сгенерированного символа в качестве входа используется тот который должен был быть сгенерирован. Но и это приводит к проблемам. **Каким?**



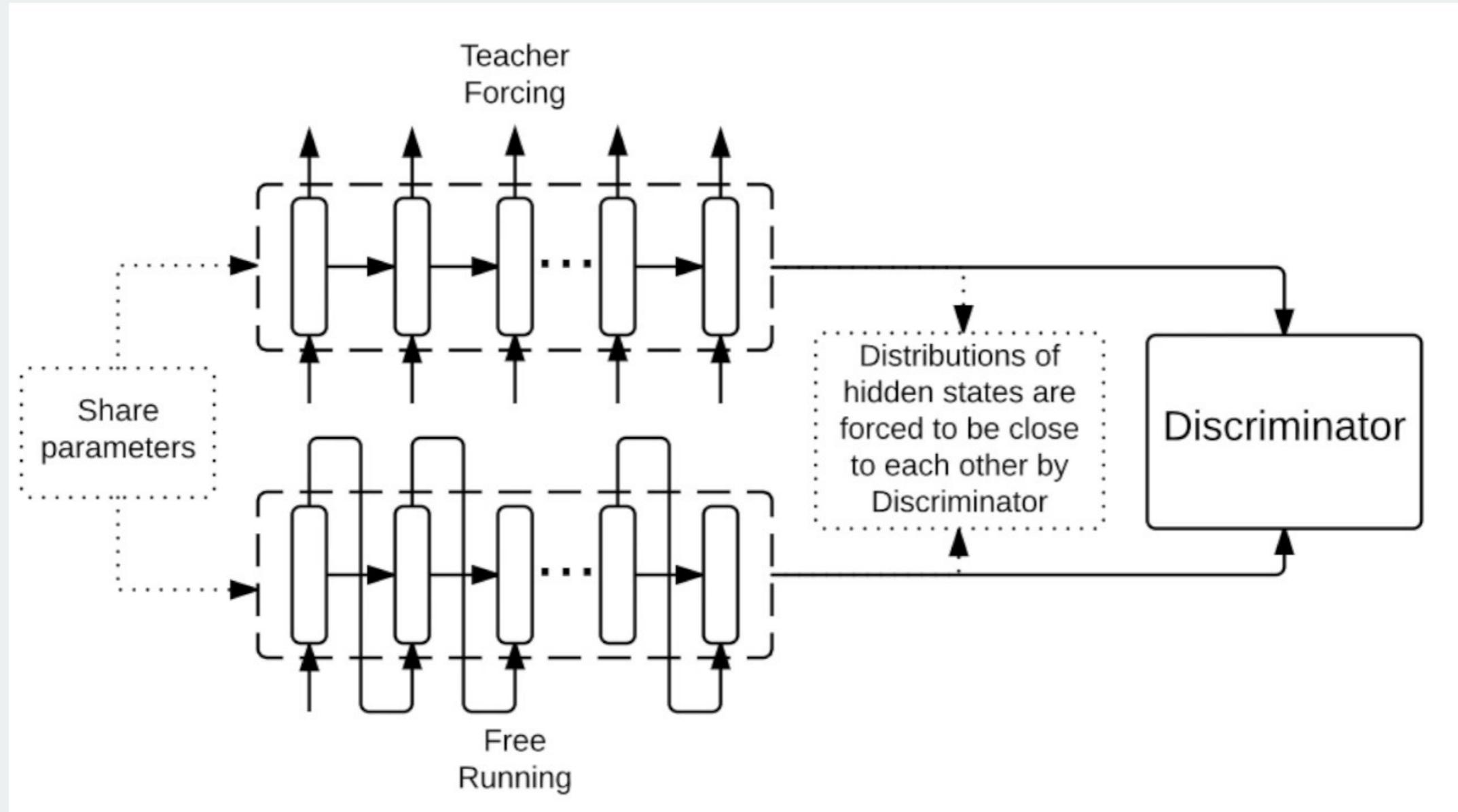
# Teacher Forcing



Для того чтобы проверить что состояния ячейки при freerun и при teacher forcing отличаются достаточно взять эти состояния и, например, визуализировать с помощью t-SNE.  
А раз мы уже умеем заставлять распределения быть похожими друг на друга...



# Professor Forcing



Professor Forcing: A New Algorithm for Training Recurrent Networks: <https://arxiv.org/abs/1610.09038>

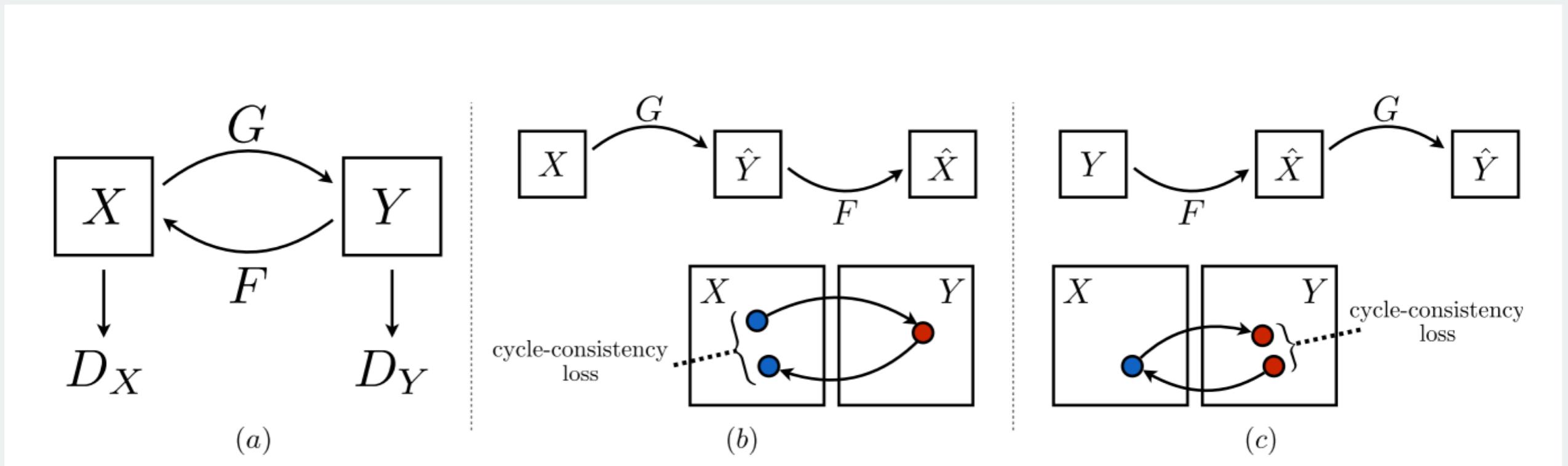


# План на сегодня

1. Иントро
2. Generative Adversarial Nets (GANs)
3. Информация и Адаптация
4. Честная условная генерация
5. Тексты, подкрепление и профессор
6. Другие трюки



# Cycle GANs



Вообще, собирать архитектуру нейронной сети можно кучей разных способов. Мы уже делали доменную адаптацию и дистилляцию метки прикладывая дискриминаторы к разным частям сети. А что, если теперь мы попробуем объекты из одной выборки преобразовать в объекты другой выборки и назад?

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks: <https://arxiv.org/abs/1703.10593>

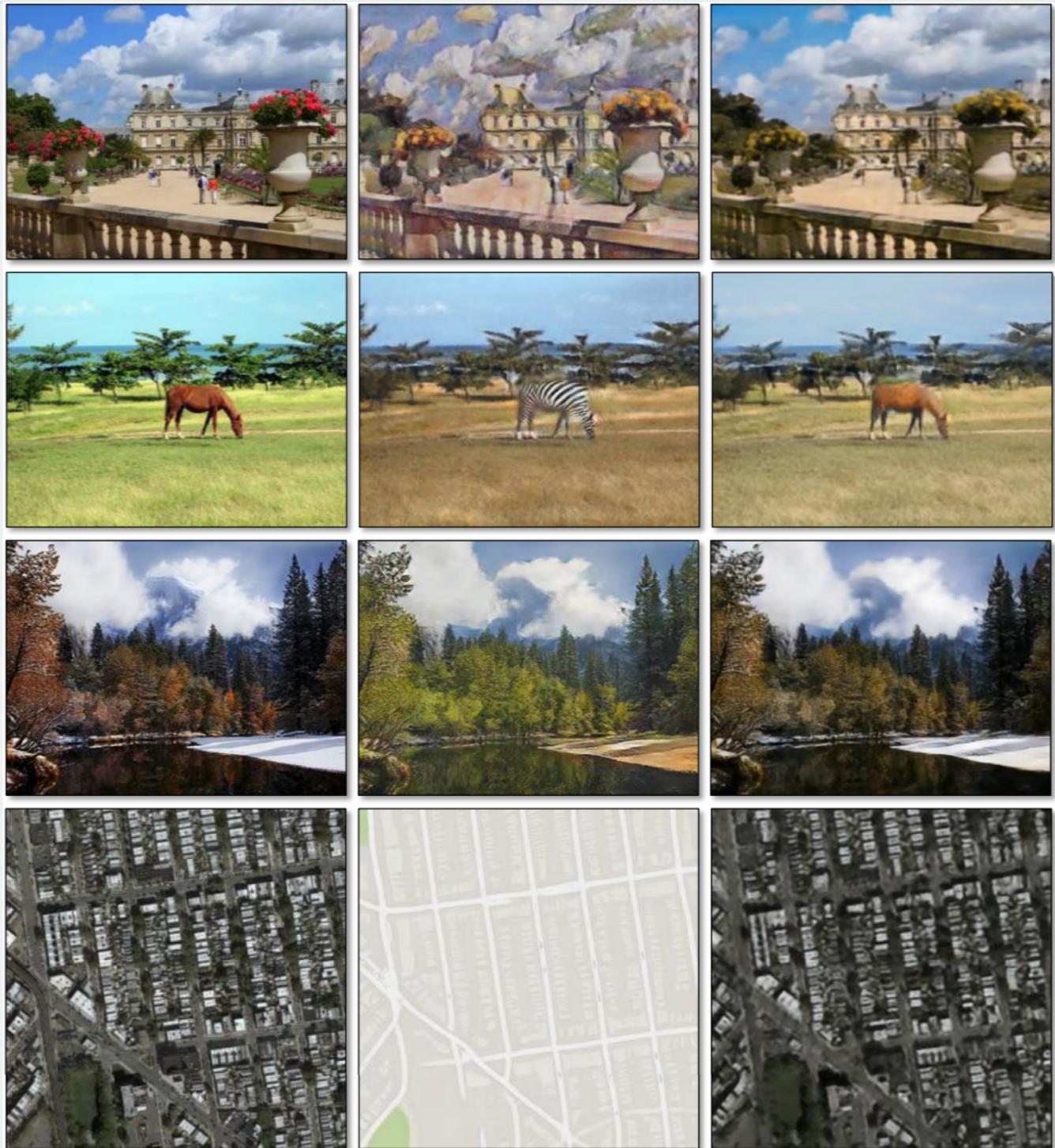


# Cycle GANs

Оказывается, что если обучать такую архитектуру, даже не имея изначально пар объектов из разных датасетов, то можно получить клевые картинки. Аналогичным образом можно попробовать сделать перевод!

<https://arxiv.org/abs/1611.00179>

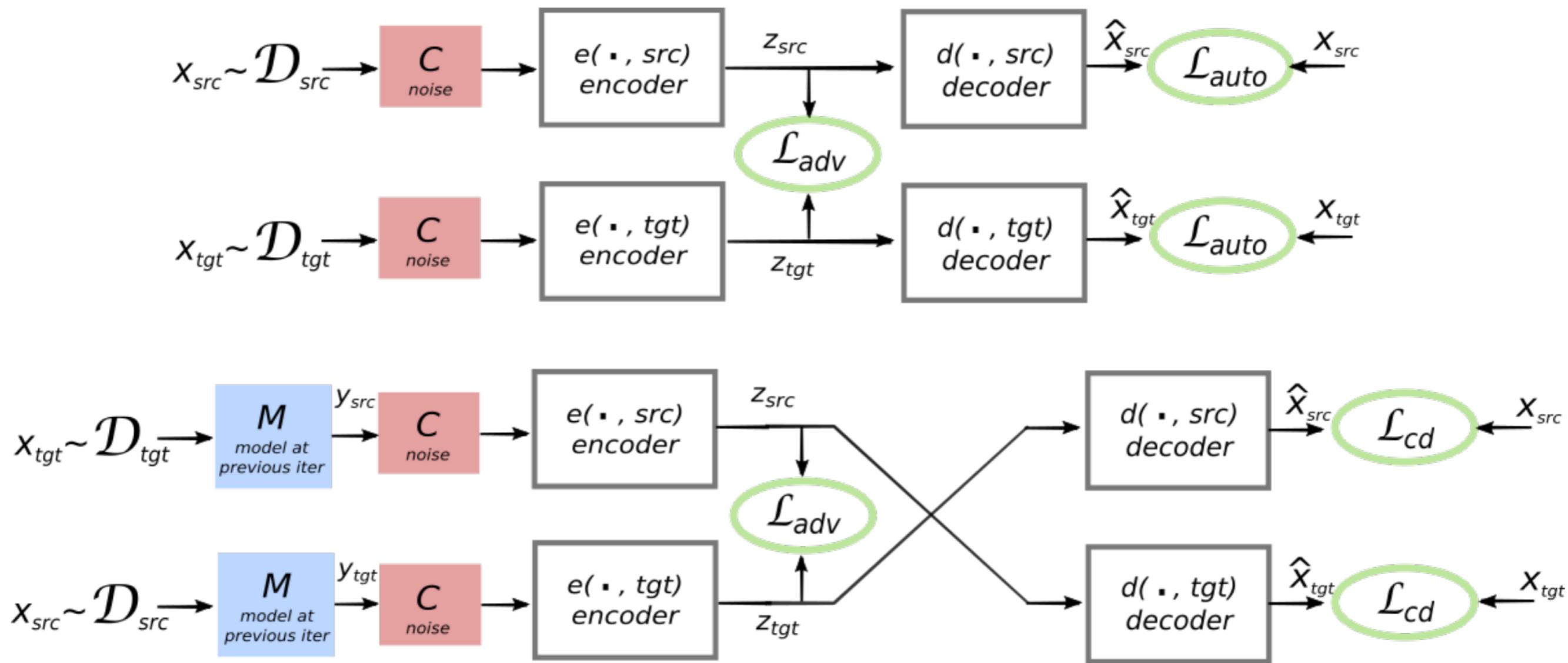
В этой статье, правда, пришлось использовать небольшой спаренный датасет для претрейна.



Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks: <https://arxiv.org/abs/1703.10593>



# Параллельные автокодировщики



Понятно, что похожий подход можно попробовать применить не только к обычным GAN'ам, но и к ААЕ.

Unsupervised Machine Translation Using Monolingual Corpora Only: <https://arxiv.org/abs/1711.00043>



# Катастрофическое забывание

Одна из известных проблем нейронных сетей заключается в том, что сеть обученная решать конкретную задачу, при дообучении на другую задачу забывает как решать первую. Одним из типичных способов обойти эту проблему является подмешивание примеров из первого множества при обучении второй задаче.

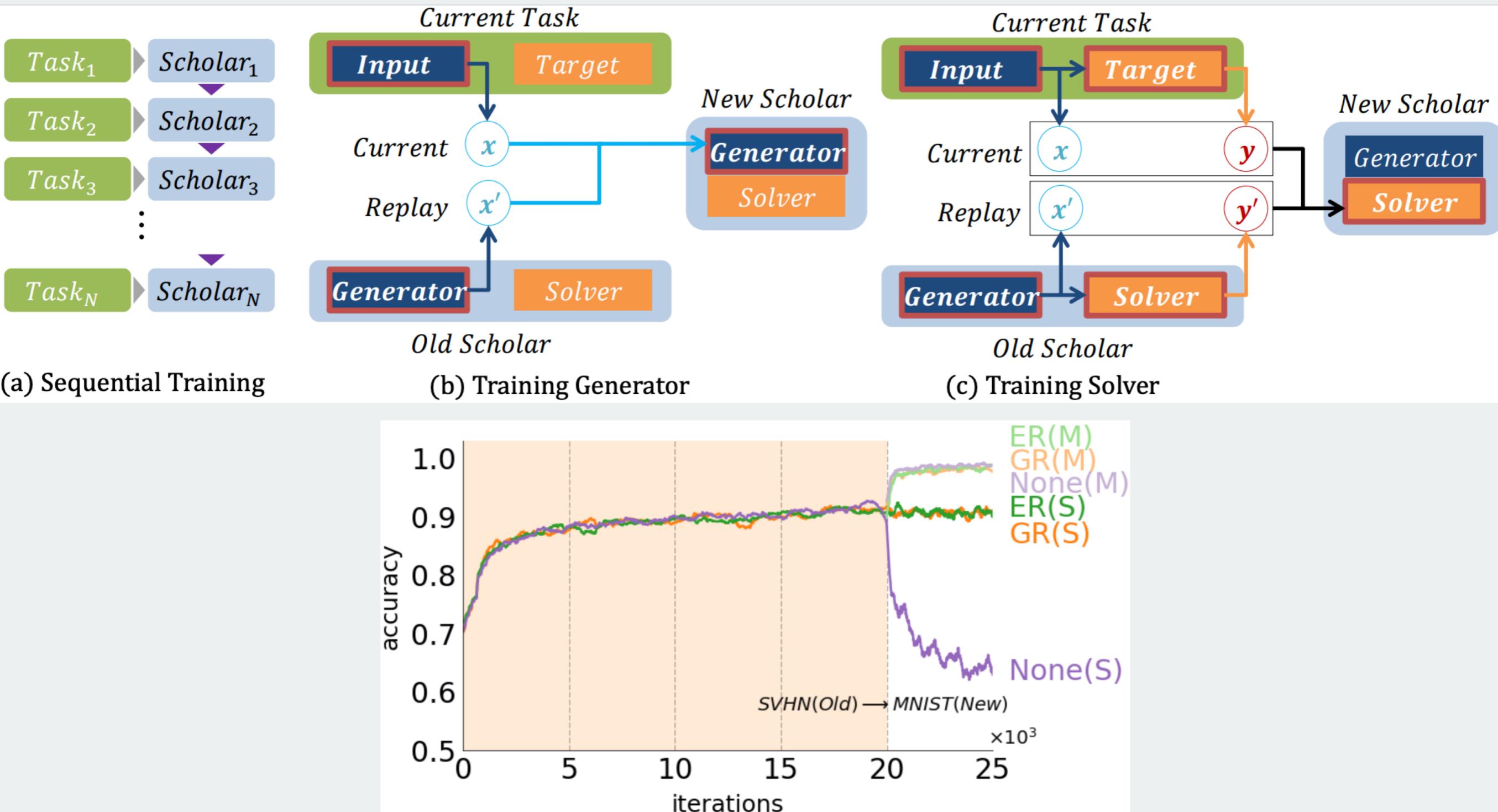
К счастью (или к сожалению), люди не сохраняют тренировочные примеры из прошлого для того чтобы не разучиться ходить в процессе обучения вождению. Современная теория в качестве возможного объяснения этой волшебной способности предлагает такой механизм: гиппокамп сохраняет новый опыт в течение дня, а во время сна, с помощью многократных повторов, эта информация объединяется с той что уже сохранена в коре.

Фактически, у нас есть две памяти и время от времени они синхронизируются.

Continual Learning with Deep Generative Replay: <https://arxiv.org/abs/1705.08690>



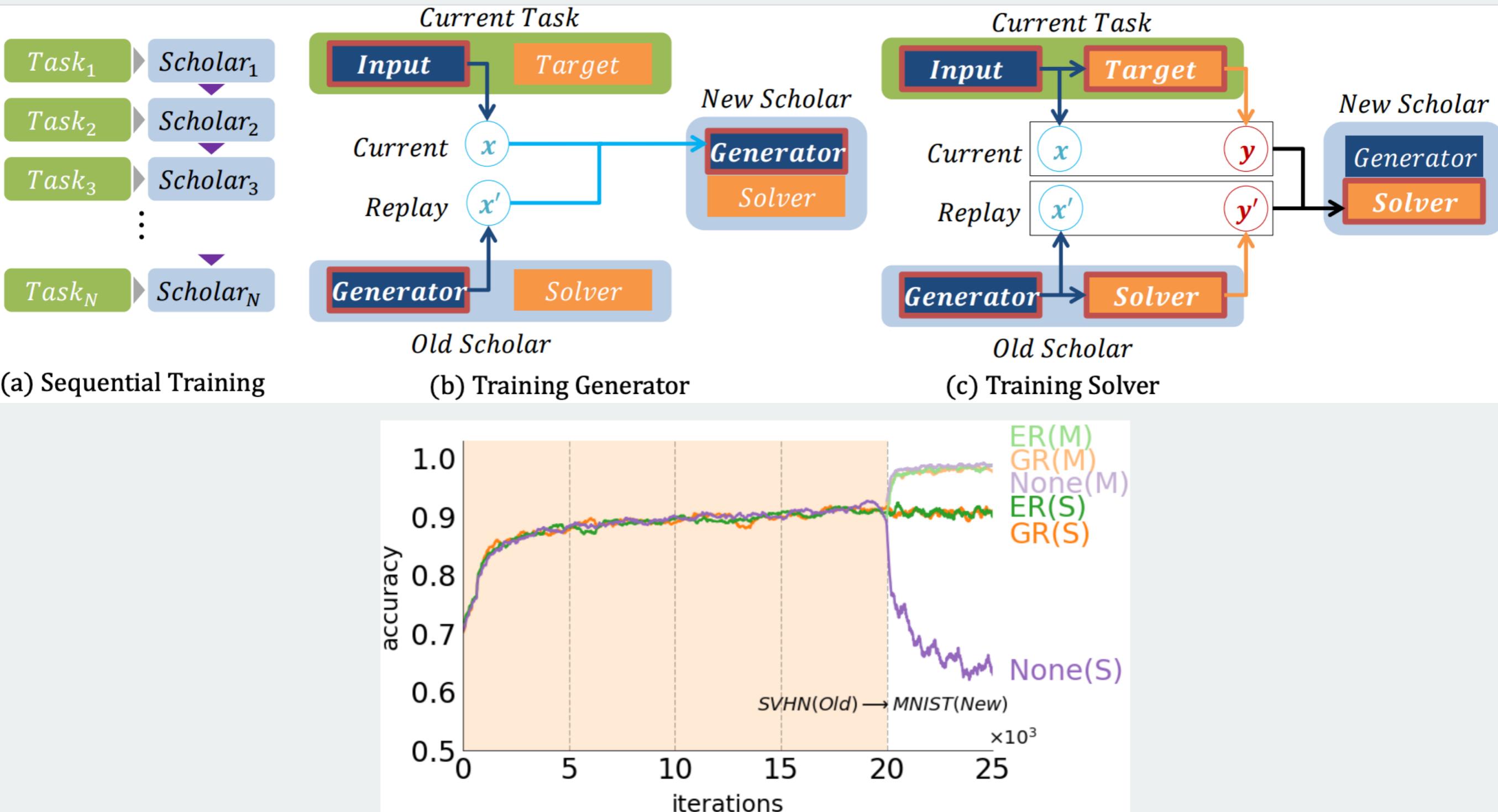
# Катастрофическое забывание



Continual Learning with Deep Generative Replay: <https://arxiv.org/abs/1705.08690>



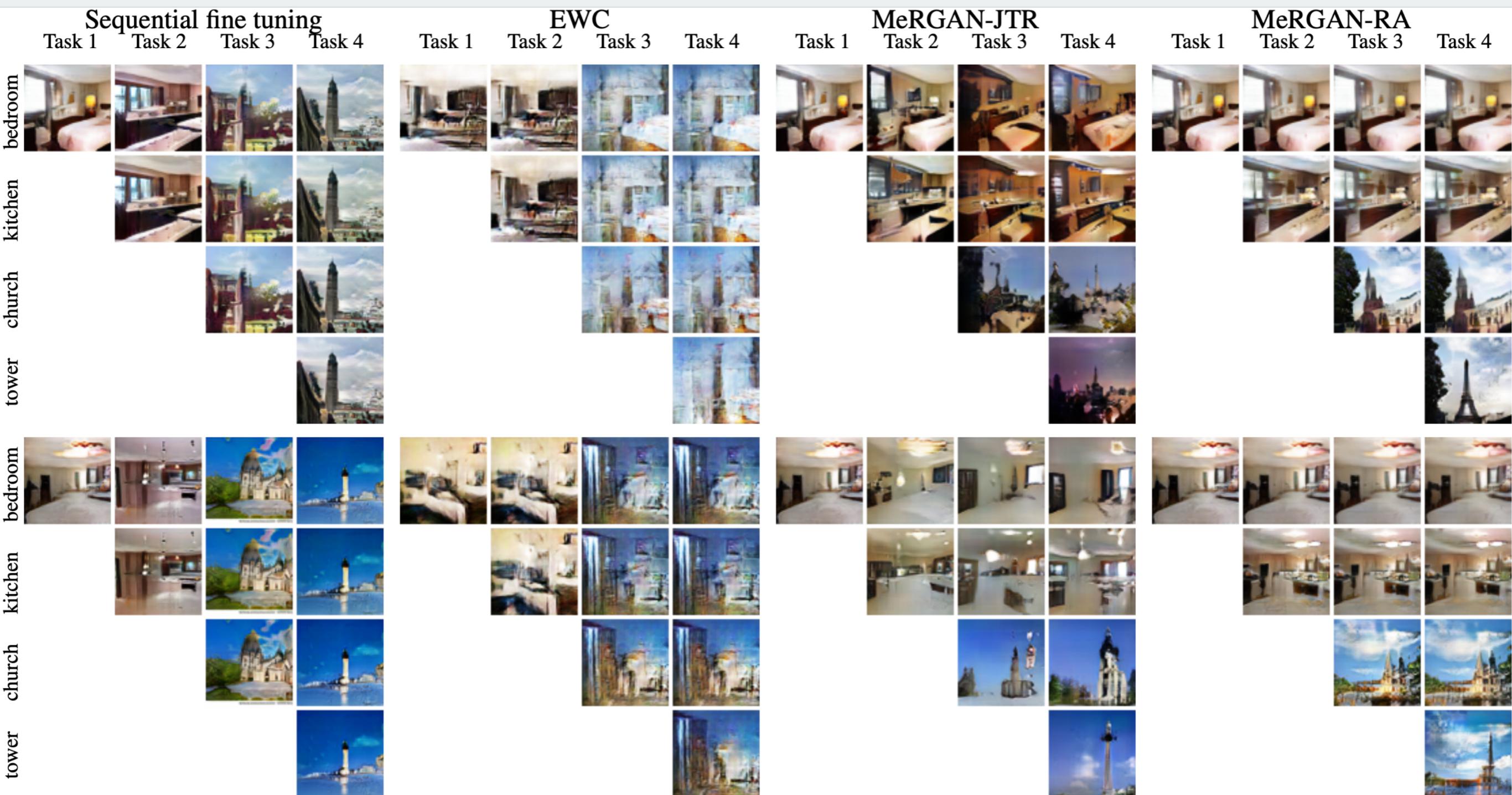
# Катастрофическое забывание



Continual Learning with Deep Generative Replay: <https://arxiv.org/abs/1705.08690>

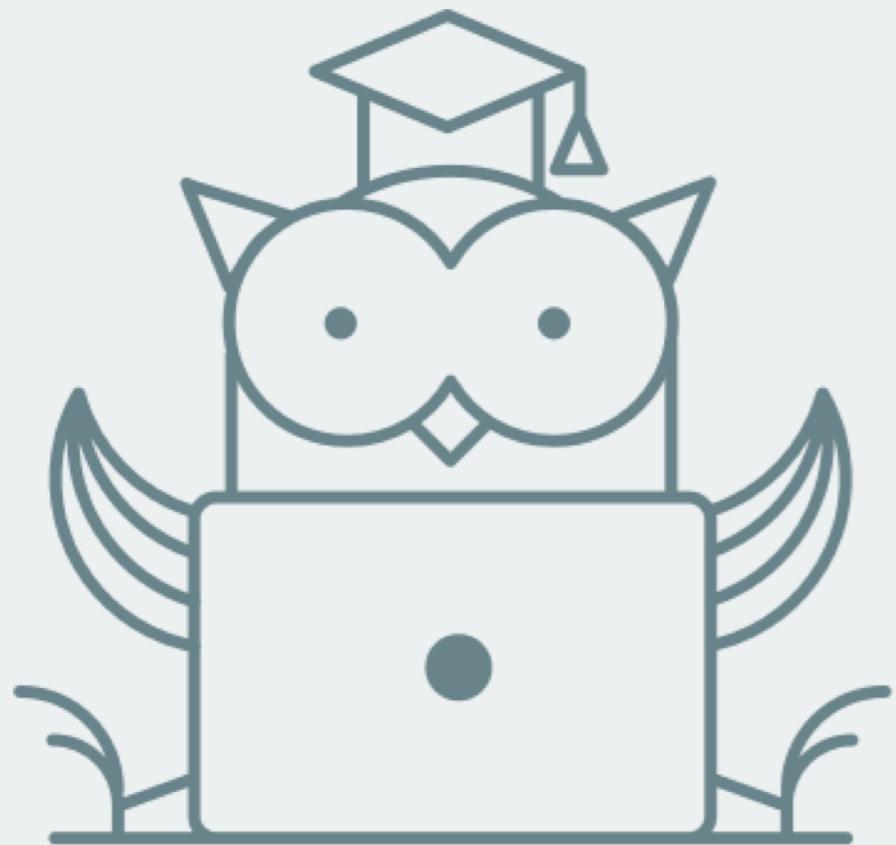


# Катастрофическое забывание



Memory Replay GANs: learning to generate images from new categories without forgetting: <https://arxiv.org/abs/1809.02058>

# Заключение



# Заключение



Спасибо  
за внимание!

