

Содержание

- ❑ Цель лекции
- ❑ Понятие генеративной модели
- ❑ Отличие дискриминативных и генеративных моделей
- ❑ Генеративные состязательные сети
 - Общая схема модели
 - Постановка задачи обучения
 - Алгоритм обучения модели
- ❑ Классификация генеративных состязательных сетей
- ❑ Примеры приложений генеративных состязательных сетей
- ❑ Заключение



Цель лекции

- **Цель** - изучить общую схему построения генеративных состязательных сетей и алгоритм их обучения, рассмотреть классификацию генеративных состязательных сетей и примеры их практического использования



ПОНЯТИЕ ГЕНЕРАТИВНОЙ МОДЕЛИ



Генеративная модель

- ❑ **Генеративная модель** описывает общие правила генерации набора данных в терминах вероятностной модели. Сэмплирование данных из построенной вероятностной модели позволяет генерировать новые данные



Схема генеративного моделирования (1)

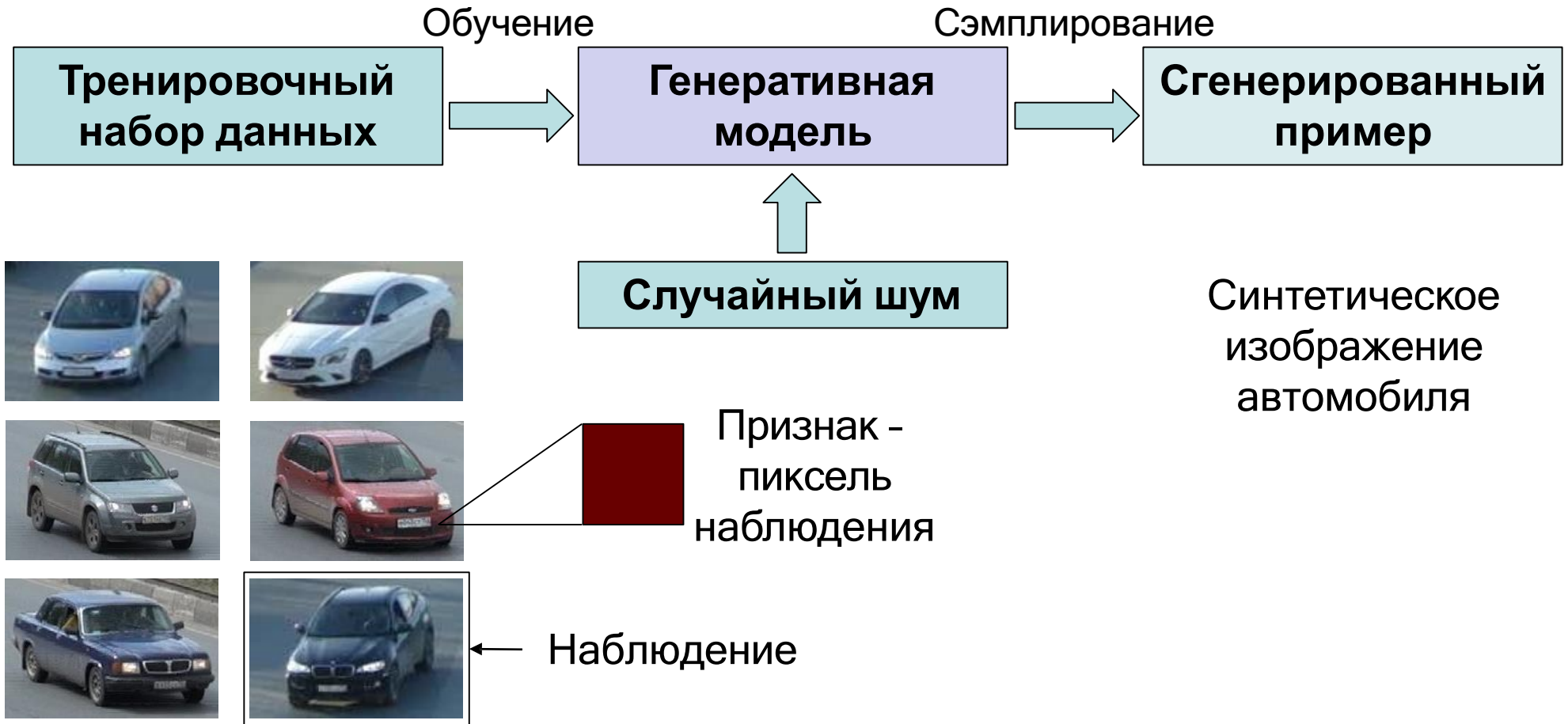


Схема генеративного моделирования (2)

- ❑ Предполагается, что имеется набор данных, содержащий множество примеров сущностей, которые требуется генерировать, - **тренировочный набор данных** (training dataset)
- ❑ **Наблюдение** (observation) - отдельный элемент тренировочного набора данных
- ❑ В задаче генерации изображений каждое наблюдение состоит из множества признаков, как правило, **признак** - значение интенсивности отдельного пикселя
- ❑ Модель обучается генерировать **данные** (изображения) по тем же правилам, что построен тренировочный набор данных
- ❑ В процессе генерации каждому пикселю назначается некоторое значение интенсивности



Схема генеративного моделирования (3)

- ❑ Генеративная модель должна быть вероятностной
- ❑ Если модель в процессе сэмплирования, например, в качестве значения интенсивности пикселя выдает среднее значение по тренировочной выборке, то модель не является генеративной, поскольку обеспечивает всегда одинаковый результат
- ❑ Модель должна включать элемент случайности, отражающий индивидуальные особенности сгенерированного примера
- ❑ Таким образом, ***должно существовать вероятностное распределение, которое объясняет, почему некоторые наблюдения (изображения) похожи в тренировочном наборе, а другие – нет***
- ❑ ***Цель – построить модель, наилучшим образом имитирующую это распределение***



ОТЛИЧИЕ ДИСКРИМИНАТИВНЫХ И ГЕНЕРАТИВНЫХ МОДЕЛЕЙ



Схема дискриминативного моделирования (1)

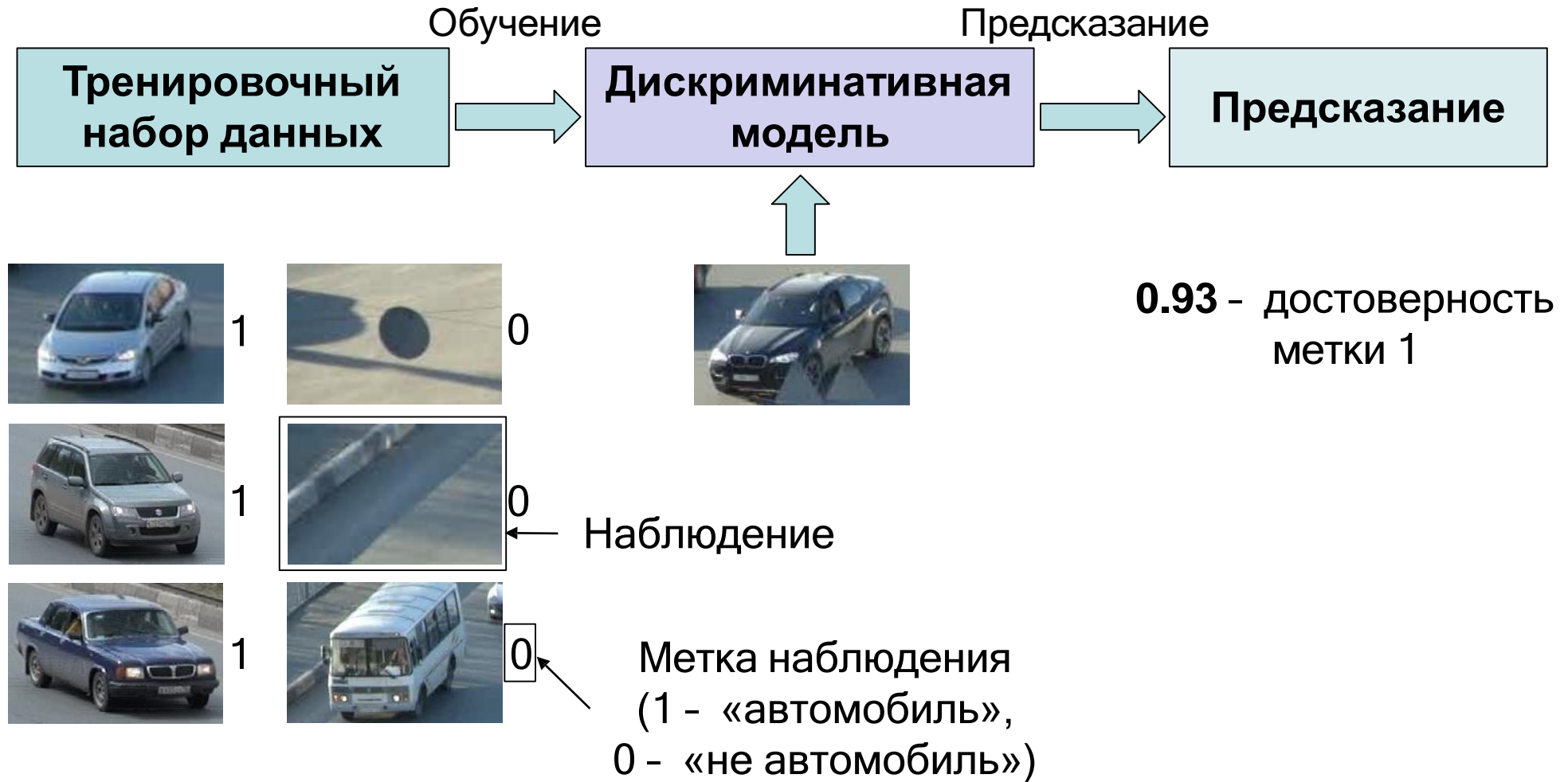


Схема дискриминативного моделирования (2)

- ❑ Предполагается, что имеется **тренировочный набор данных** (training dataset)
- ❑ **Наблюдение** (observation) - отдельный элемент тренировочного набора данных
- ❑ Каждое наблюдение имеет **метку** (label)
- ❑ **Цель – построить функцию на основании тренировочного набора данных, которая наилучшим образом отображает наблюдения на множество меток**



Отличие дискриминативного и генеративного моделирования (1)

□ *Разные цели моделирования*

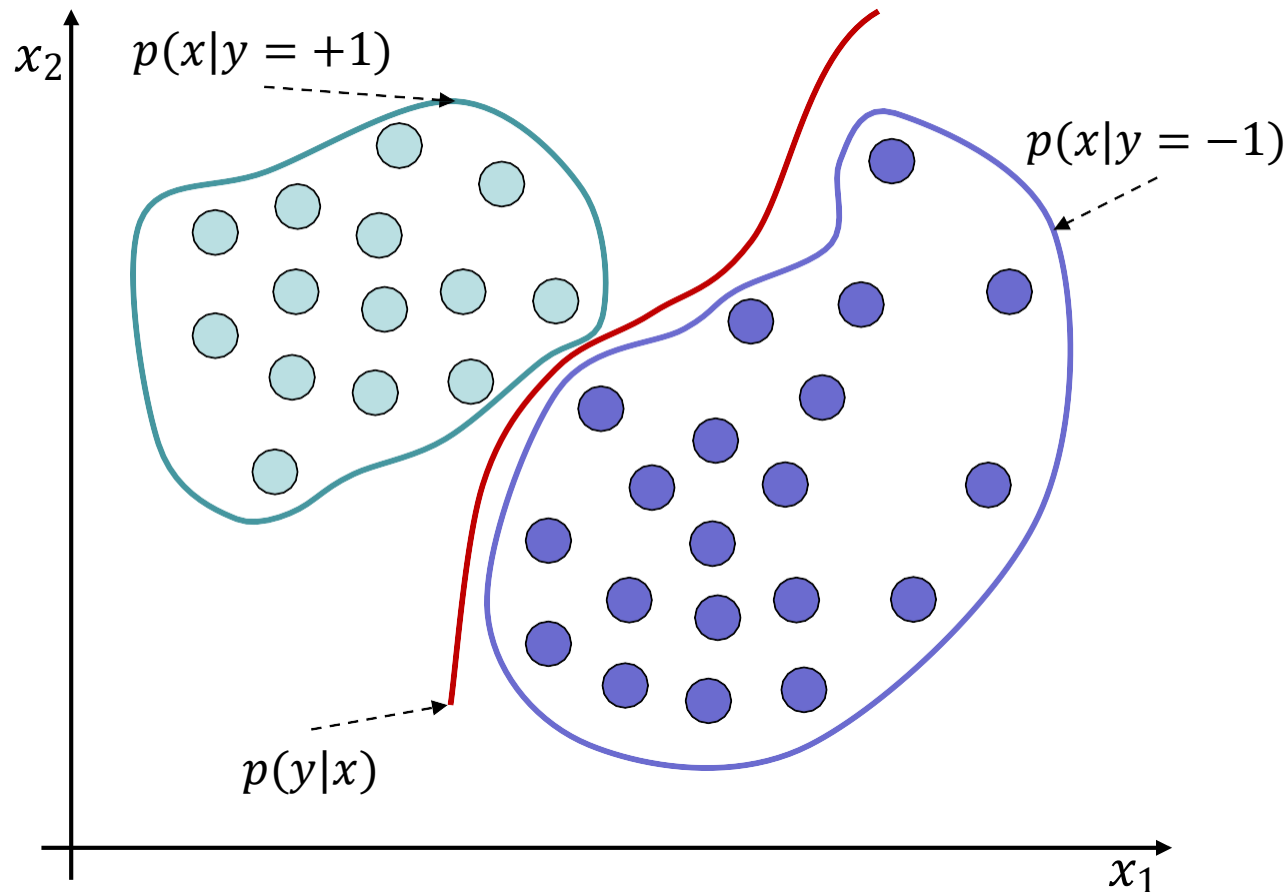
- **Генеративные модели** оценивают $p(x, y)$ - совместное распределение вероятностей (оценивают границы классов)
 - Если набор данных не размечен (решается задача обучения без учителя), то модели оценивают $p(x)$ - вероятность увидеть наблюдение x
 - Если тренировочный набор размечен, то генеративная модель может оценивать условную вероятность $p(x|y)$ (вычисляется по правилу Байеса) - вероятность увидеть наблюдение x при условии, что известна его принадлежность классу y
- **Дискриминативные модели** оценивают $p(y|x)$ - вероятность метки y при условии, что на входе имеется наблюдение x (оценивают границу между классами)

* Ng A.Y., Jordan M.I. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naïve Bayes // Advances in Neural Information Processing Systems. - 2002. - [\[http://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf\]](http://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf).



Отличие дискриминативного и генеративного моделирования (2)

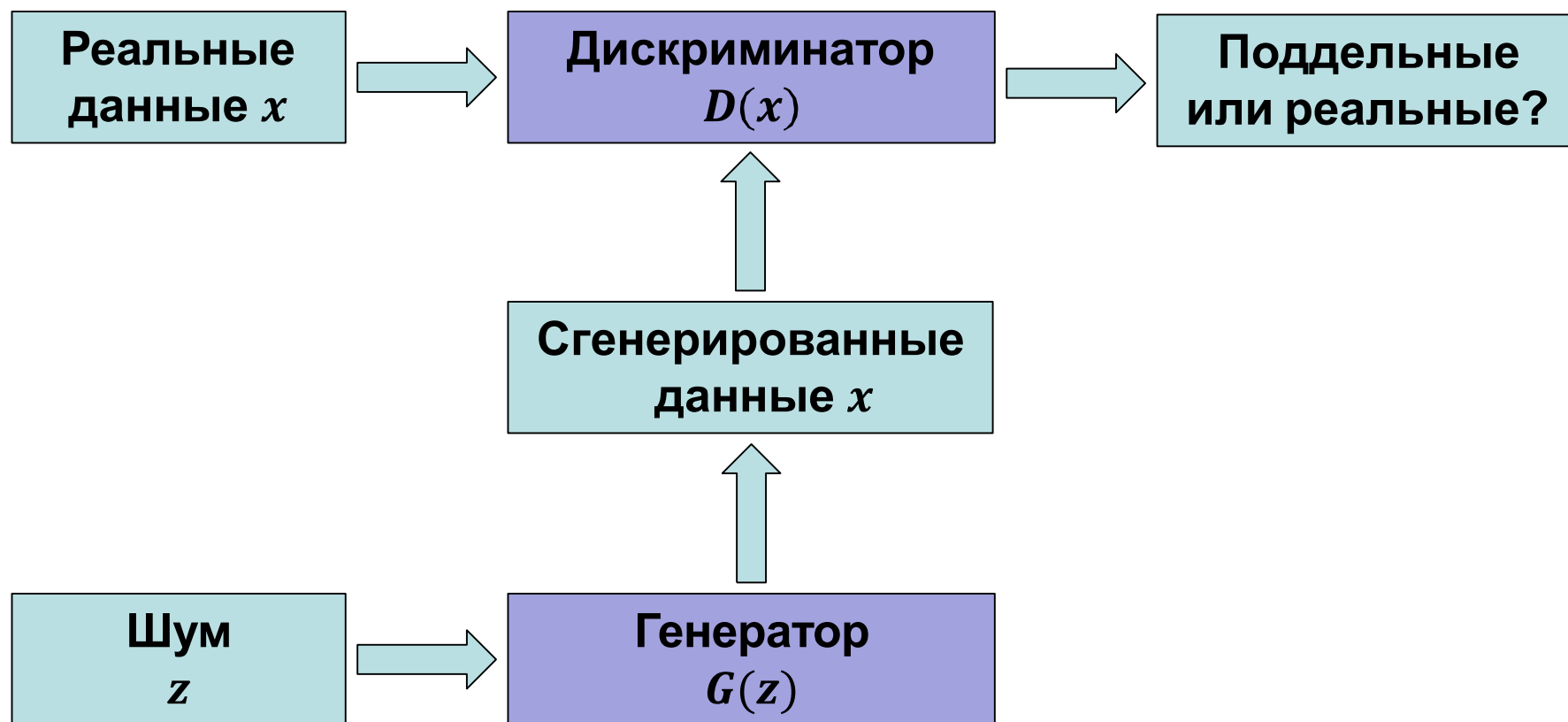
- Графическая интерпретация на примере двумерного наблюдения и двух классов объектов:



ГЕНЕРАТИВНЫЕ СОСТЯЗАТЕЛЬНЫЕ СЕТИ



Общая схема модели (1)



Общая схема модели (2)

- ❑ Генеративная состязательная сеть состоит из двух нейронных сетей:
 - **Генератор** - сеть, которая генерирует образцы. Цель генератора - научиться «обманывать» дискриминатор
 - **Дискриминатор** - сеть, которая пытается отличить реальные наблюдения от сгенерированных образцов. Цель дискриминатора - научиться наилучшим образом отличать «обман»
- ❑ Пример - генерация фотографий реалистичных лиц:
 - Вход генератора - многомерный случайный шум, выход генератора (вход дискриминатора) - сгенерированное RGB-изображение лица
 - Выход дискриминатора - достоверность того, что RGB-изображение - настоящее лицо (число от 0 до 1)



Постановка задачи обучения (1)

□ Обозначения:

- X - множество наблюдений из распределения p_{data}
- Z - пространство латентных факторов из распределения p_Z (например, множество случайных векторов из равномерного распределения на отрезке $[0; 1]$)

□ **Генератор**

- Отображение $G: Z \rightarrow \mathbb{R}^n$ с параметрами θ
- Цель G - сгенерировать образец, максимальное похожий на наблюдения из распределения p_{data}

□ **Дискриминатор**

- Отображение $D: \mathbb{R}^n \rightarrow [0; 1]$ с параметрами γ
- Цель D - выдавать максимальную оценку на наблюдениях из X и минимальную оценку на образцах, сгенерированных G



Постановка задачи обучения (2)

- Обозначения:
 - θ и γ - параметры нейронных сетей, соответствующих генератору и дискриминатору
 - p_{gen} - распределение образцов, порождаемое генератором
- **Задача** - построить распределение p_{gen} , которое наилучшим образом описывает распределение p_{data}



Постановка задачи обучения (3)

□ **Функция ошибки:**

$$E_{x \sim p_{data}} [\log D(x)] + E_{\tilde{x} \sim p_{gen}} [\log(1 - D(\tilde{x}))],$$

где $E_{\tilde{x} \sim p_{gen}} [\log(1 - D(\tilde{x}))] = E_{z \sim p_Z} [\log(1 - D(G(z)))]$

□ **Задача обучения** генеративной состязательной сети сводится к решению задачи оптимизации (минимаксная игра)

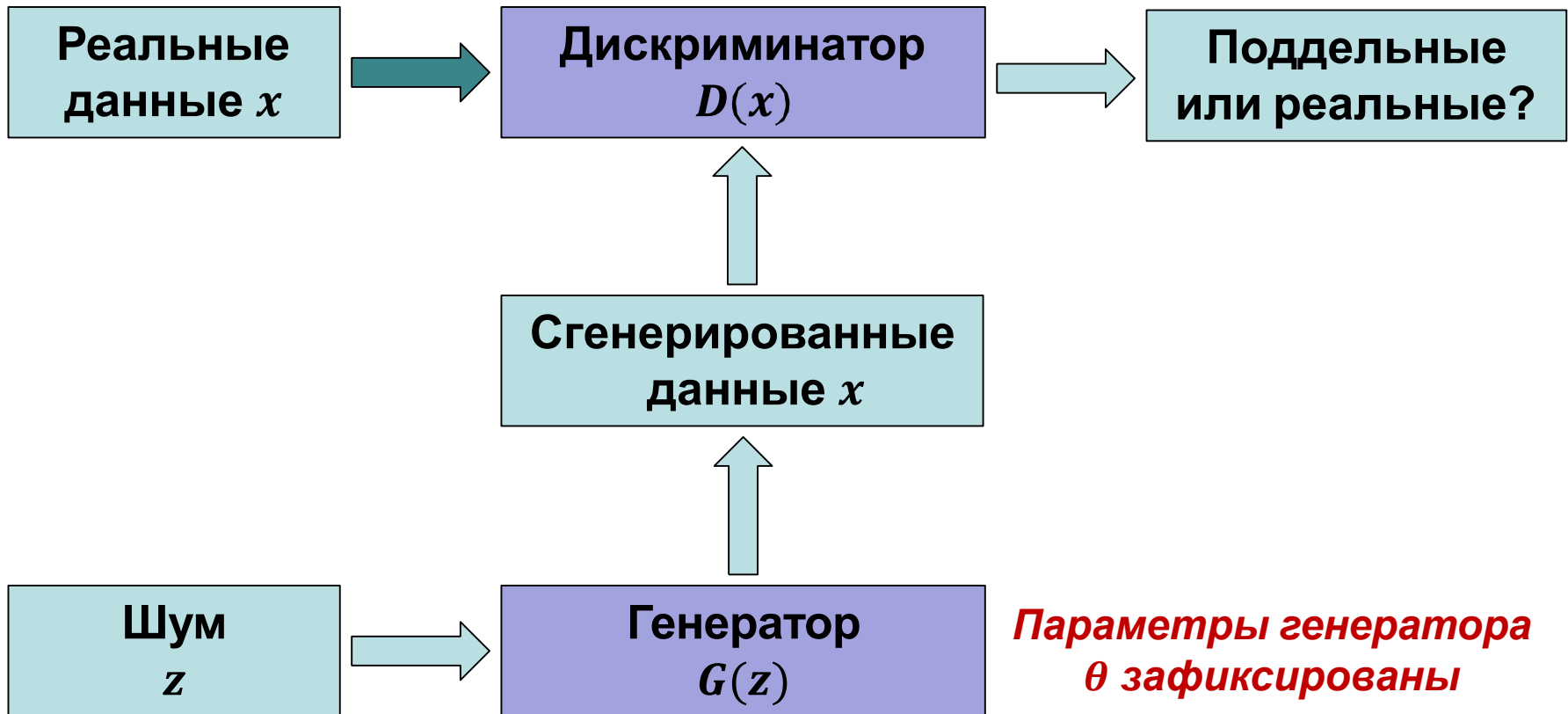
$$\min_G \max_D E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p_Z} [\log(1 - D(G(z)))]$$

□ **Примечание:** обоснование сходимости распределения p_{gen} к p_{data} описано в исходной статье*, где предложены модели

* Goodfellow I.J., et al. Generative Adversarial Nets. - 2014. - [<https://arxiv.org/pdf/1406.2661.pdf>].

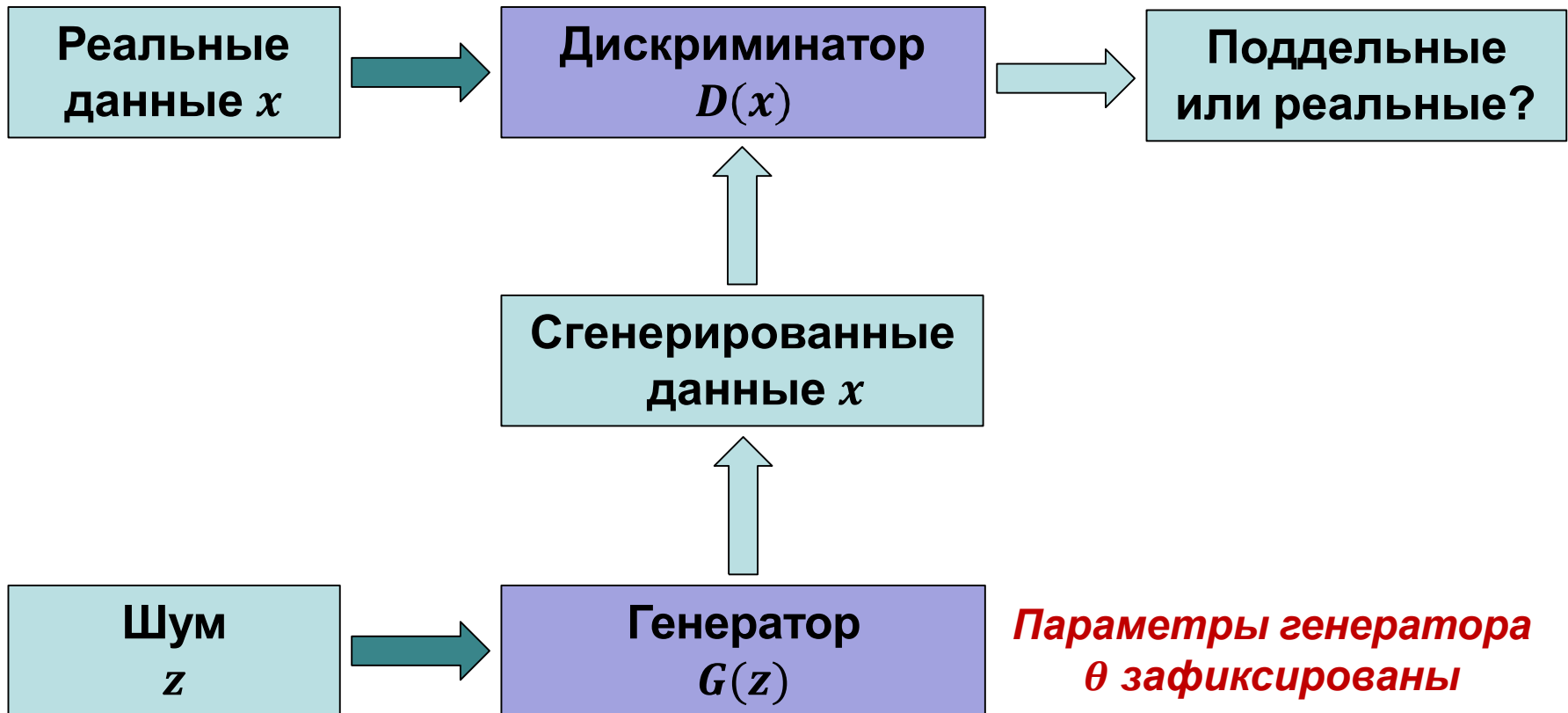
Алгоритм обучения. Этап 1 (1)

1. Сэмплировать $\{x_1, x_2, \dots, x_m\}$
из распределения p_{data}



Алгоритм обучения. Этап 1 (2)

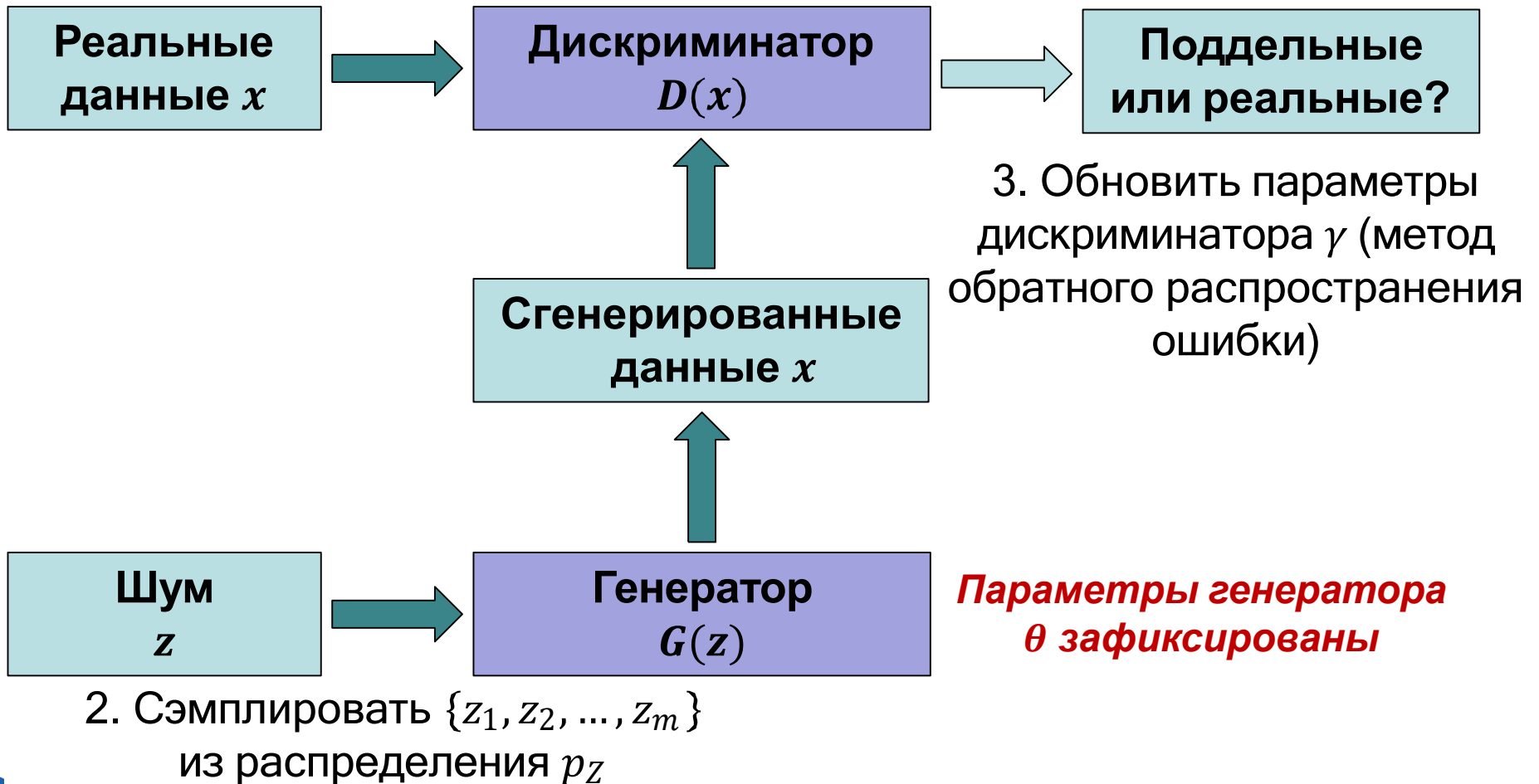
1. Сэмплировать $\{x_1, x_2, \dots, x_m\}$
из распределения p_{data}



2. Сэмплировать $\{z_1, z_2, \dots, z_m\}$
из распределения p_Z

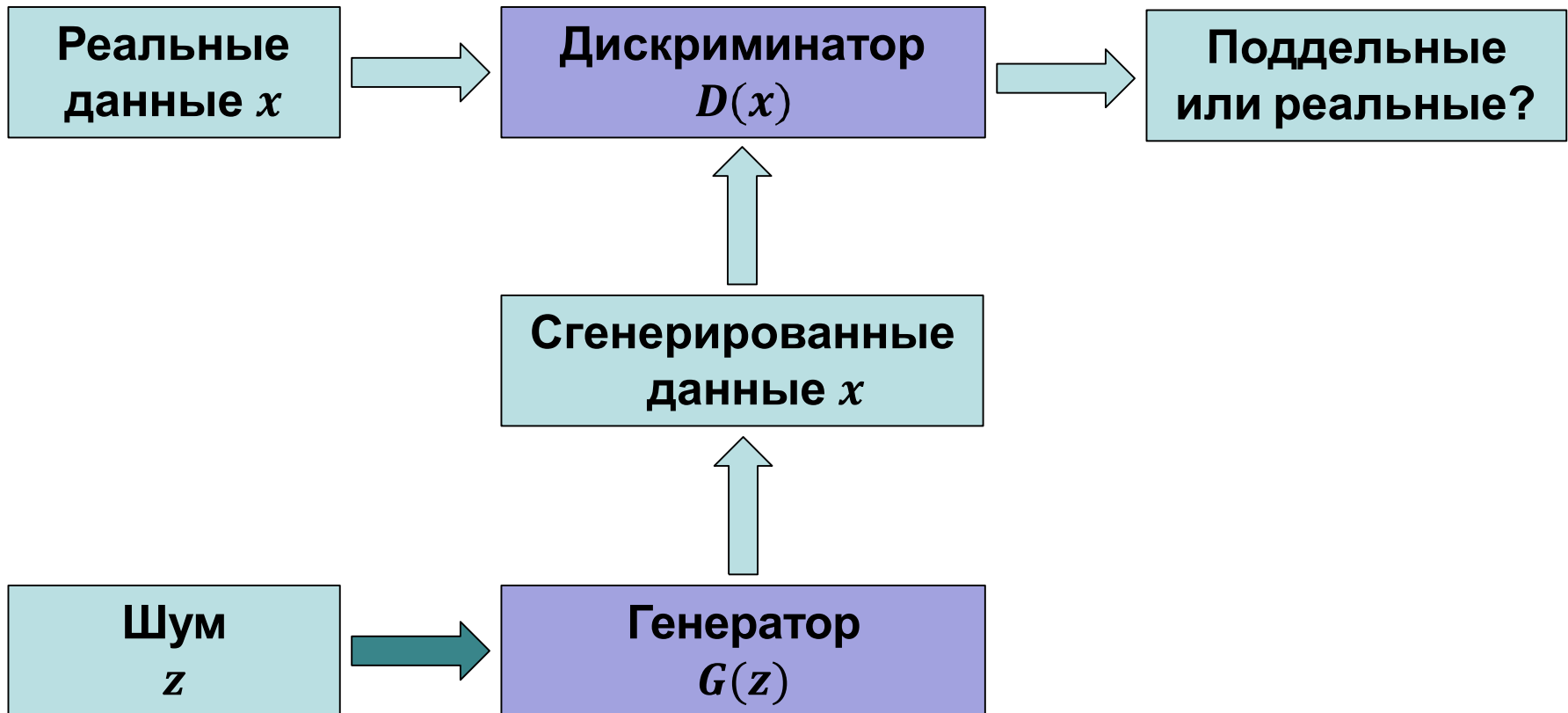
Алгоритм обучения. Этап 1 (3)

1. Сэмплировать $\{x_1, x_2, \dots, x_m\}$
из распределения p_{data}



Алгоритм обучения. Этап 2 (1)

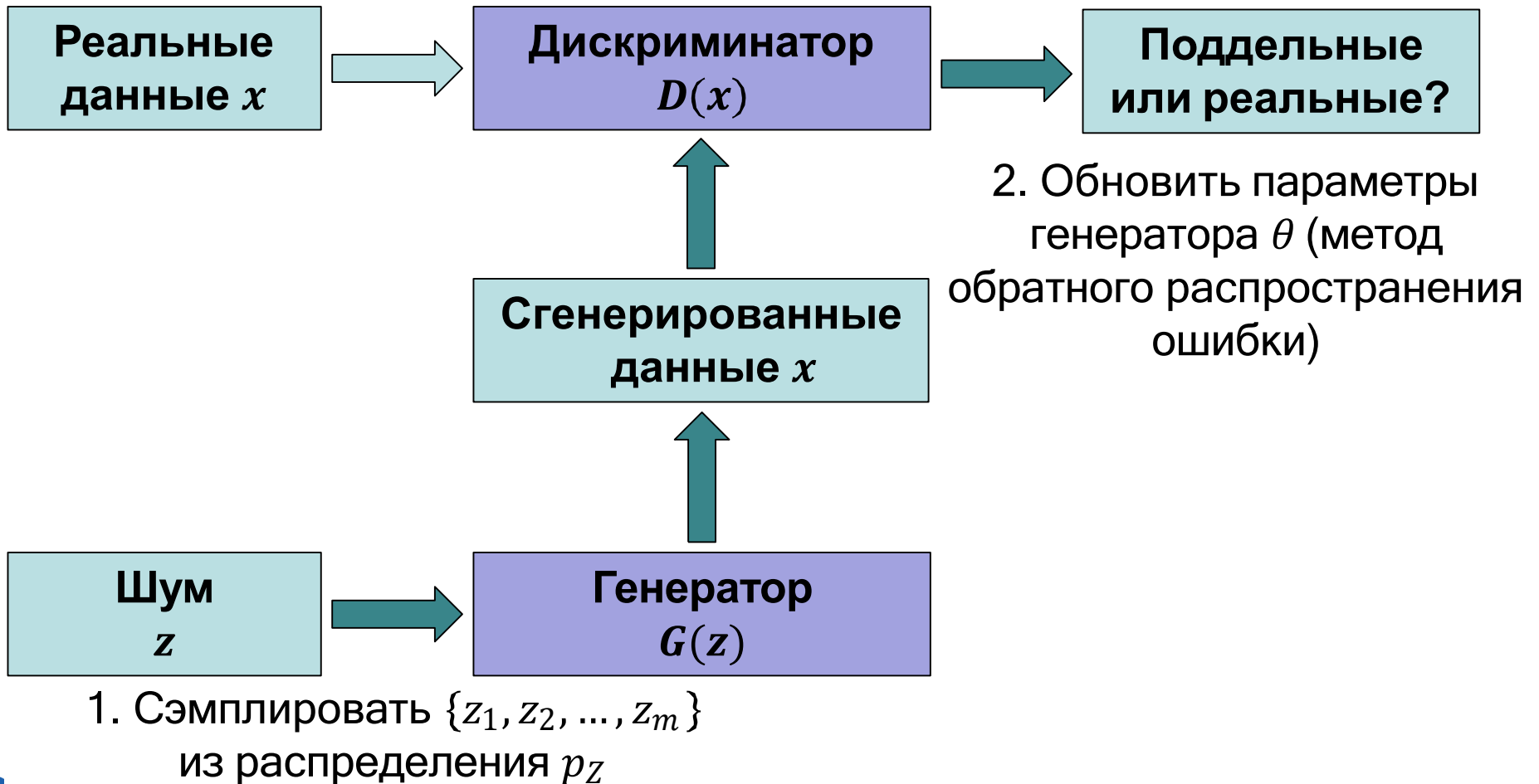
*Параметры дискриминатора
у зафиксированы*



1. Сэмплировать $\{z_1, z_2, \dots, z_m\}$
из распределения p_z

Алгоритм обучения. Этап 2 (2)

*Параметры дискриминатора
у зафиксированы*



Алгоритм обучения. Псевдокод

```
for i = 1..num_iteration do
```

```
  for j = 1..k do
```

```
    1. Сэмплировать  $\{x_1, \dots, x_m\}$  из распределения  $p_{data}$ 
```

```
    2. Сэмплировать  $\{z_1, \dots, z_m\}$  из распределения  $p_Z$ 
```

```
    1.3. Обновить параметры дискриминатора  $\gamma$ 
```

$$\Delta\gamma \leftarrow \nabla_{\gamma} \frac{1}{m} \sum_{t=1}^m [\log D(x_t)] + [\log(1 - D(G(z_t)))]$$

```
  end for
```

```
    1. Сэмплировать  $\{z_1, \dots, z_m\}$  из распределения  $p_Z$ 
```

```
    2. Обновить параметры генератора  $\theta$ 
```

$$\Delta\theta \leftarrow \nabla_{\theta} \frac{1}{m} \sum_{t=1}^m [\log(1 - D(G(z_t)))]$$

```
end for
```


Алгоритм обучения. Особенности реализации

- ❑ Количество итераций обновления параметров дискриминатора k и параметров генератора `num_iterations` являются параметрами алгоритма обучения
- ❑ При реализации обратного распространения ошибки в исходном методе обучения используется **стохастический градиентный спуск** (Stochastic Gradient Descend, SGD)



КЛАССИФИКАЦИЯ ГЕНЕРАТИВНЫХ СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ

Генерация синтетических данных с использованием генеративных состязательных сетей



Классификация генеративных состязательных сетей (1)

- ❑ **Полносвязные генеративные состязательные сети** (Fully Connected GANs)
- ❑ **Условные генеративные состязательные сети** (Conditional GANs, CGAN)
- ❑ **Пирамида Лапласа из состязательных сетей** (Laplacian Pyramid of Adversarial Networks, LAPGAN)
- ❑ **Глубокие сверточные генеративные состязательные сети** (Deep Convolutional GANs, DCGAN)
- ❑ **Генеративные рекуррентные состязательные сети** (Generative Recurrent Adversarial Networks, GRAN)
- ❑ ...

* Alqahtani H., Kavakli-Thorne M., Kumar G. Applications of Generative Adversarial Networks (GANs): An Updated Review // Archives of Computational Methods in Engineering. - 2019.

Классификация генеративных состязательных сетей (2)

- ❑ **Генеративные состязательные сети, максимизирующие информацию** (Information Maximizing GANs, InfoGAN)
- ❑ **Двунаправленные генеративные состязательные сети** (Bidirectional GANs, BiGAN)
- ❑ За пределами лекции останутся следующие виды генеративных состязательных сетей:
 - Генеративные автокодировщики (Adversarial Autoencoders, AAE)
 - Гибриды вариационных автокодировщиков и генеративных состязательных сетей (Variational Autoencoder-GAN, VAE-GAN)
 - Некоторые другие специализированные модели

* Alqahtani H., Kavakli-Thorne M., Kumar G. Applications of Generative Adversarial Networks (GANs): An Updated Review // Archives of Computational Methods in Engineering. - 2019.

Fully Connected GANs

- ❑ **Полносвязные генеративные состязательные сети** (Fully Connected GANs) - модели, в которых генератор и дискриминатор имеют вид полносвязных глубоких нейронных сетей
- ❑ Впервые полносвязные генеративные состязательные сети использованы для генерации реалистичных изображений, аналогичных содержащимся в следующих наборах данных*:
 - MNIST [<http://yann.lecun.com/exdb/mnist>]
 - CIFAR-10 [<https://www.cs.toronto.edu/~kriz/cifar.html>]
 - Toronto Face Dataset (TFD)**

* Goodfellow I.J., et al. Generative Adversarial Nets // Advances in neural information processing systems. - 2014. - P. 2672-2680. - [<https://arxiv.org/pdf/1406.2661.pdf>].

** Susskind J., Anderson A., Hinton G. E. The Toronto face dataset. Technical Report UTML TR 2010-001. - 2010.

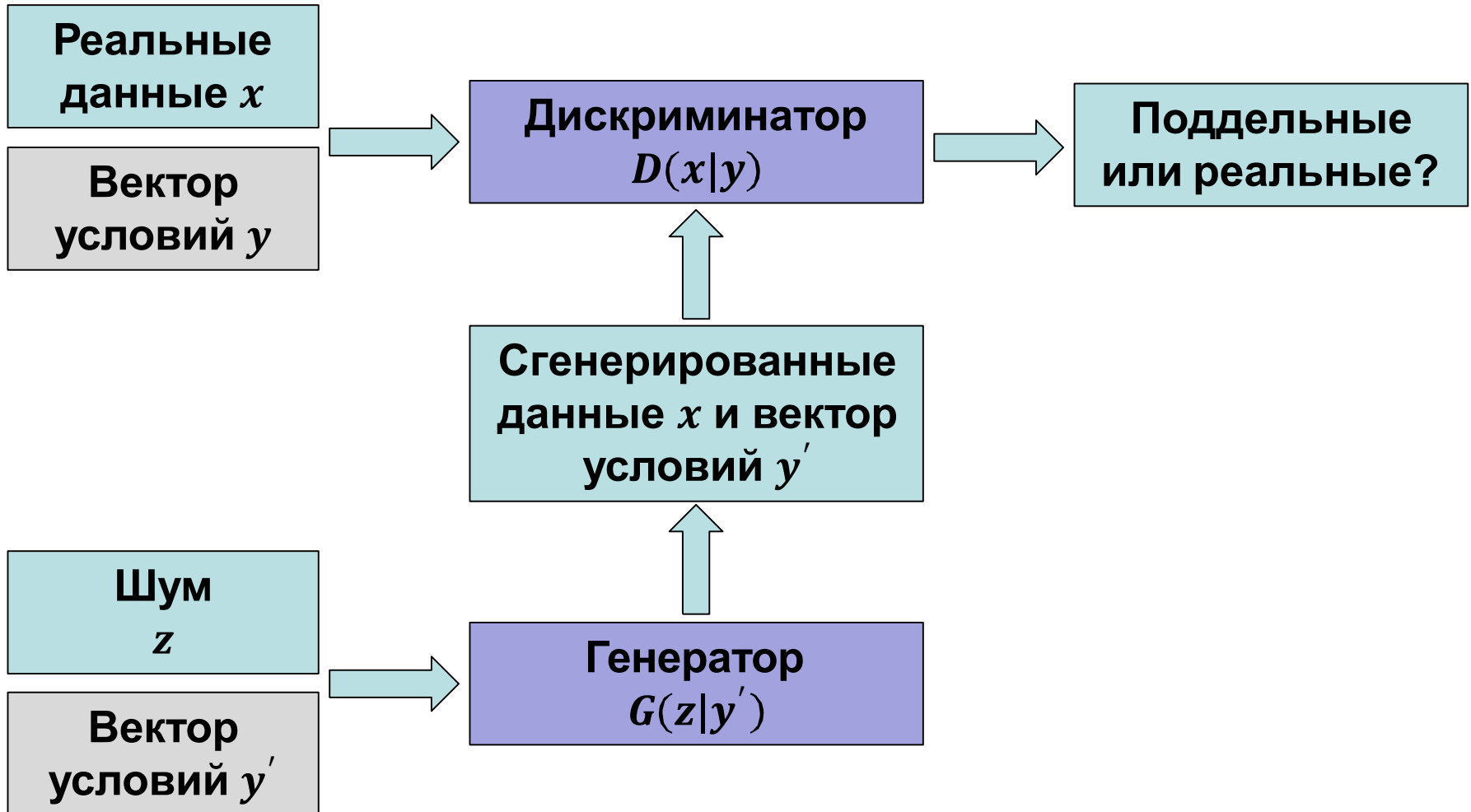
Conditional GANs (1)

- ❑ **Условные генеративные состязательные сети** (Conditional GANs, CGAN) - модели, позволяющие генерировать синтетические изображения, которые удовлетворяют некоторым условиям или обладают некоторыми свойствами (специфическими характеристиками)
- ❑ Генератор и дискриминатор получают дополнительную входную информацию
- ❑ В простейшем случае в качестве дополнительной информации используется класс изображения (вектор меток в представлении one-hot) или интересующие свойства

* Mirza M., Osindero S. Conditional generative adversarial nets. - 2014. - [<https://arxiv.org/pdf/1411.1784.pdf>].



Conditional GANs (2)



Conditional GANs (3)

❑ *Генератор*

- Вход генератора перестает быть абсолютно случайным в результате добавления вектора условий в качестве входной информации
- Добавление вектора условий помогает генератору понять, каким образом лучше сгенерировать данные

❑ *Дискриминатор*

- Дискриминатор принимает решение о реальности/поддельности данных, учитывая дополнительную информацию

- ❑ ***Общая схема алгоритма обучения аналогична ранее рассмотренной, разница состоит в функции ошибки, которая зависит от дополнительных векторов условий***

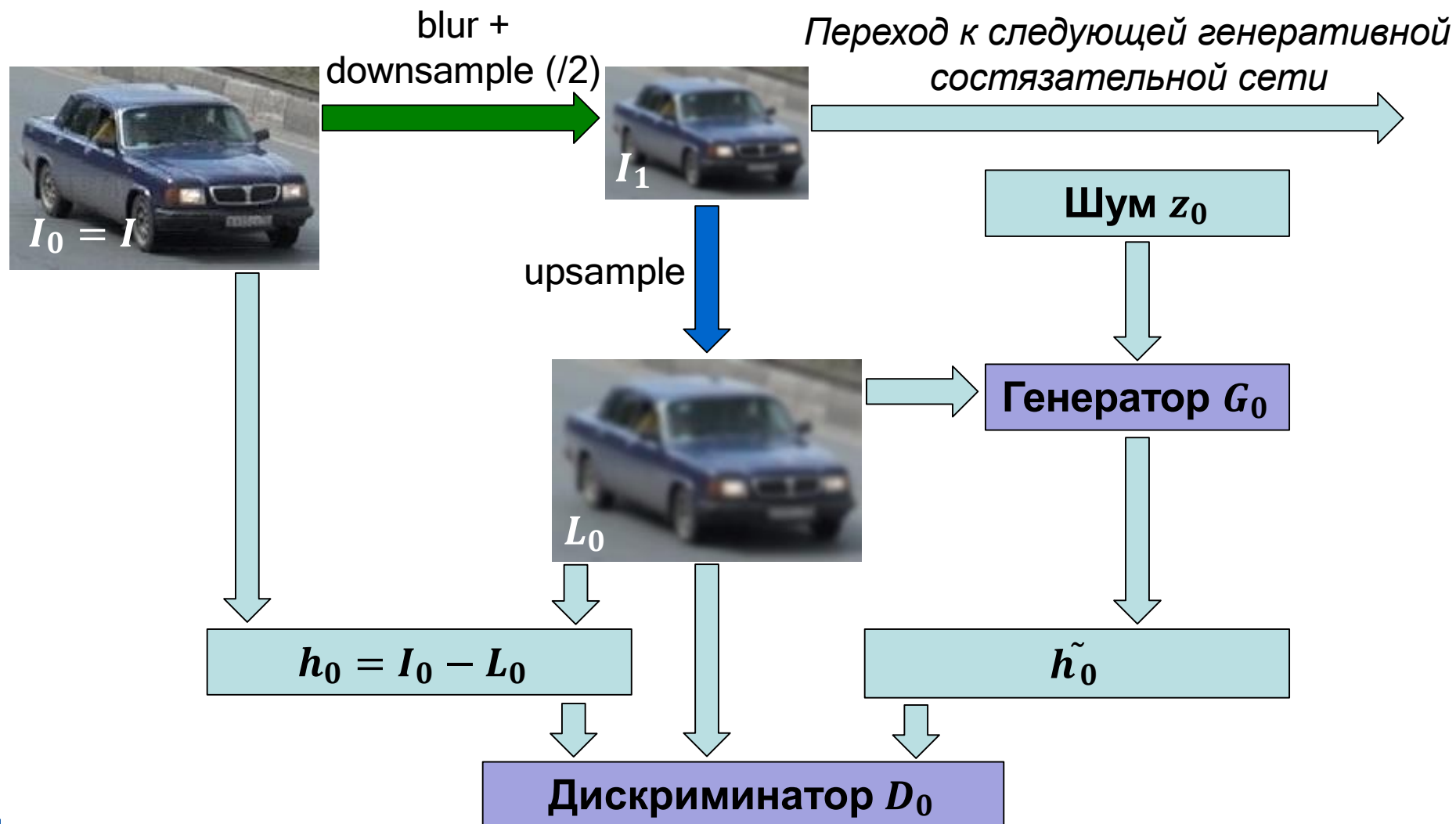


Laplacian Pyramid of Adversarial Networks (1)

- ❑ **Пирамида Лапласа из состязательных сетей** (Laplacian Pyramid of Adversarial Networks, LAPGAN) - модель, основанная на каскаде сверточных сетей, которые образуют пирамиду Лапласа для изображений разного масштаба
- ❑ Модель позволяет генерировать естественные изображения высокого разрешения за счет захвата особенностей на разных масштабах
- ❑ Пирамида Лапласа строится на базе пирамиды Гаусса с помощью операций повышающей (upsampling) и понижающей дискретизации (downsampling)
- ❑ Элемент пирамиды Лапласа задается разницей между соседними уровнями пирамиды Гаусса

* Denton E.L., et al. Deep generative image models using a Laplacian pyramid of adversarial networks // Advances in neural information processing systems. - 2015. - P. 1486-1494. - [<https://arxiv.org/pdf/1506.05751.pdf>].

Laplacian Pyramid of Adversarial Networks (2)



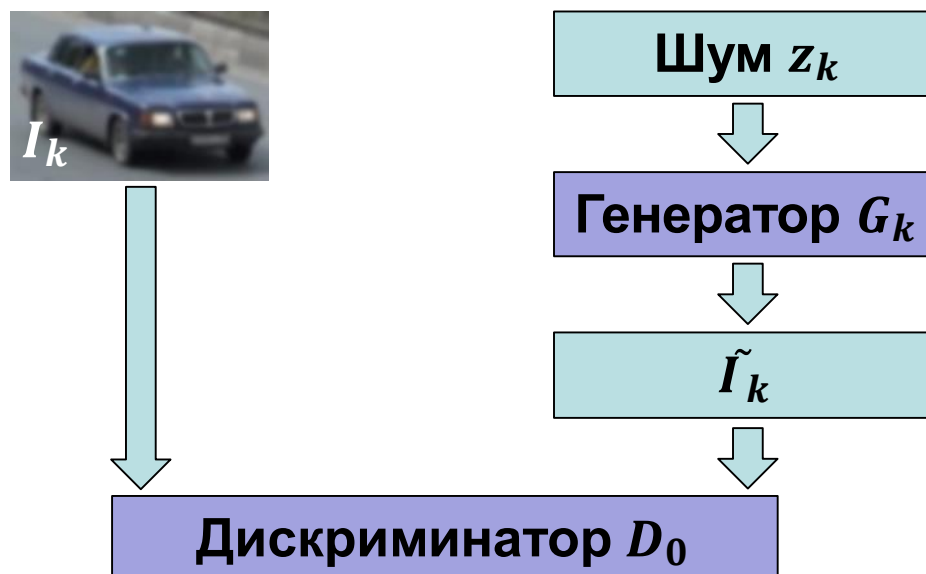
Laplacian Pyramid of Adversarial Networks (3)

- ❑ Первая состязательная сеть в каскаде в процессе обучения принимает на вход исходное изображение
- ❑ Изображение размывается и уменьшается вдвое средствами операции понижающей дискретизации
- ❑ Разрешение полученного изображения увеличивается до разрешения исходного изображения с использованием операции повышающей дискретизации
- ❑ **Генератор** принимает на вход увеличенное изображение и шум и пытается предсказать отличие увеличенного изображения относительно исходного
- ❑ **Дискриминатор** принимает решение, является ли полученное отличие реальным
- ❑ **Обучение модели – обучение условной генеративной состязательной сети**



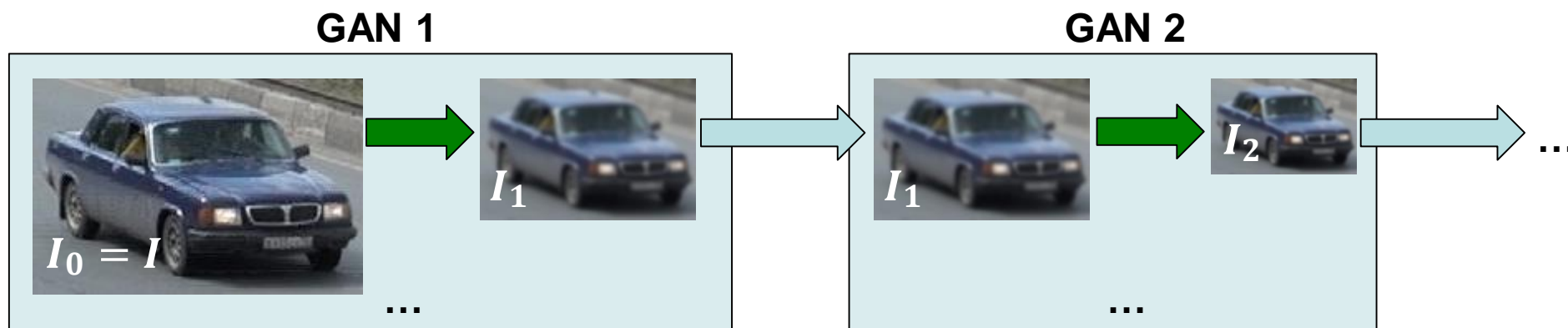
Laplacian Pyramid of Adversarial Networks (4)

- ❑ Схема построения каждой последующей генеративной состязательной сети в каскаде аналогична представленной, за исключением последней сети
- ❑ ***Последняя сеть обеспечивает восстановление изображения на основании шума***, т.е. она не является условной генеративной состязательной сетью



Laplacian Pyramid of Adversarial Networks (5)

- ❑ При обучении очередной модели на вход подается изображение, полученное в результате размытия и уменьшения разрешения для предшествующей сети



- ❑ Обучение отдельных генеративных состязательных сетей может осуществляться независимо
- ❑ Использование обученного каскада сводится к обратному проходу по построенной последовательности моделей, на входе последней модели - изображение низкого разрешения

Deep Convolutional GANs (1)

- **Глубокие сверточные генеративные состязательные сети** (Deep Convolutional GANs, DCGAN) - модели, в которых генератор и дискриминатор представляют собой глубокие сверточные сети, обладающие следующими ограничениями:
 - Отсутствуют полносвязные слои
 - Дискриминатор вместо слоев пространственного объединения (pooling) содержит разреженные свертки (strided convolutions), а генератор - разреженные свертки с дробным шагом (fractional-strided convolutions)
 - В генераторе и дискриминаторе используется нормализация по пачке (batch normalization)
 - ...

* Radford A., Metz L., Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. - 2015. - [<https://arxiv.org/pdf/1511.06434.pdf>].

Deep Convolutional GANs (2)

- ❑ **Глубокие сверточные генеративные состязательные сети** (Deep Convolutional GANs, DCGAN) - модели, в которых генератор и дискриминатор представляют собой глубокие сверточные сети, обладающие следующими ограничениями:
 - На всех слоях генератора, кроме последнего, используется функция активации «положительная срезка» (ReLU).
В исходной работе* используется гиперболический тангенс (Tanh)
 - На всех слоях дискриминатора используется «слабая положительная срезка» (Leaky ReLU)

* Radford A., Metz L., Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. - 2015. - [<https://arxiv.org/pdf/1511.06434.pdf>].

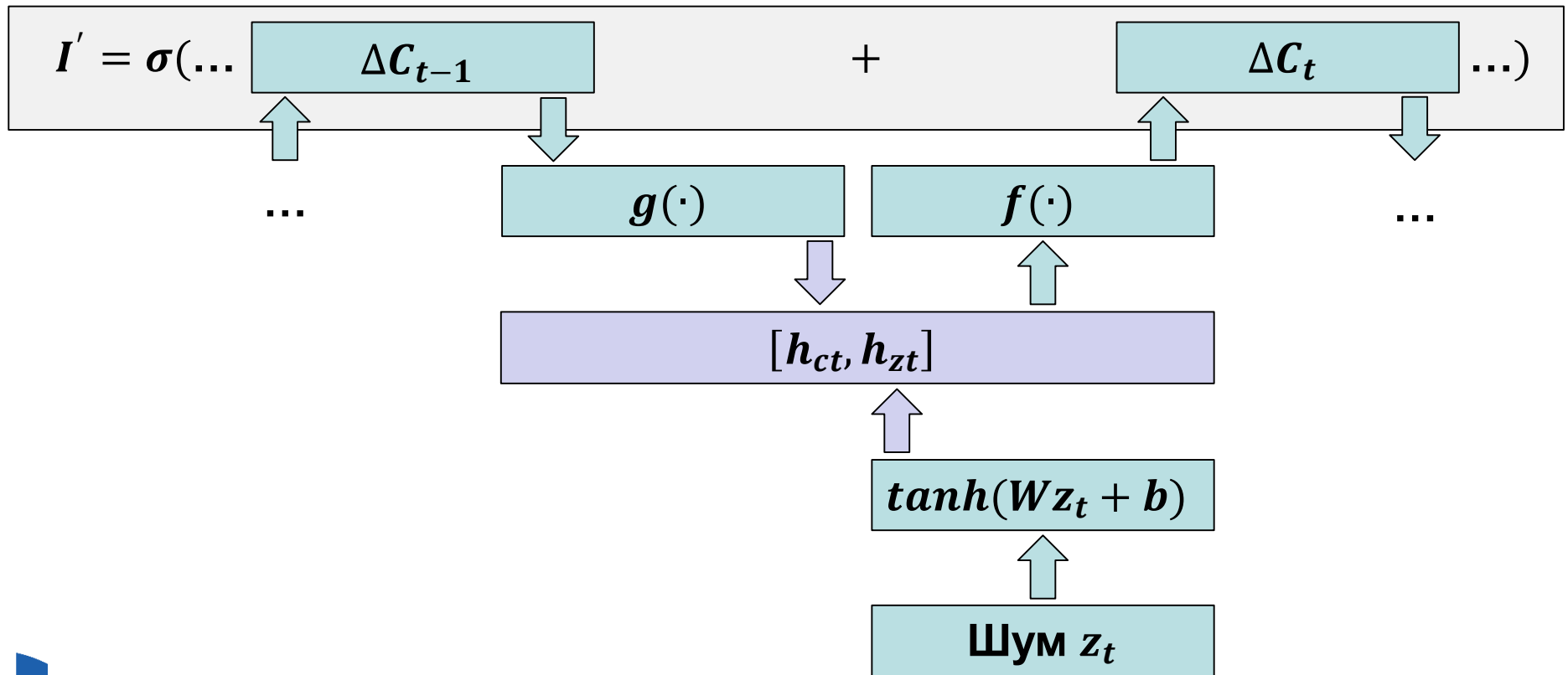
Generative Recurrent Adversarial Networks (1)

- ❑ **Генеративные рекуррентные состязательные сети** (Generative Recurrent Adversarial Networks, GRAN) - модели, в которых генератор содержит рекуррентную связь
- ❑ Генератор получает на вход последовательность примеров «шума» и создает последовательность изображений
- ❑ Очередное изображение построенной последовательности накапливает обновления, что приводит к получению финального образца
- ❑ Дискриминатор определяет, является ли полученное суммарное изображение реальным
- ❑ Далее приведена схема построения генератора

* Im D.J., Kim C.D., Jiang H., Memisevic R. Generating images with recurrent adversarial networks. - 2016. - [<https://arxiv.org/pdf/1602.05110.pdf>].

Generative Recurrent Adversarial Networks (2)

- $\{z_t, t = \overline{1, T}\}, z_t \sim p(Z)$ - последовательность примеров шума
- $\Delta C_1, \Delta C_2, \dots, \Delta C_T$ - сгенерированная последовательность изображений



Generative Recurrent Adversarial Networks (3)

- ❑ $g(\cdot)$ - сеть, содержащая последовательность сверточных слоев и полносвязный слой, работает как кодировщик
- ❑ $f(\cdot)$ - обратная копия сети $g(\cdot)$ (полносвязный слой и последовательность разверточных слоев), работает как декодировщик
- ❑ h_{ct} - закодированное представление изображения, сгенерированного на шаге $t - 1$
- ❑ h_{zt} - гипотеза о необходимых обновлениях
- ❑ $[h_{ct}, h_{zt}]$ - конкатенация
- ❑ Выход генератора представляет собой сумму по всем сгенерированным изображениям
- ❑ Обучение реализуется посредством **метода обратного распространения ошибки с разворачиванием во времени** (backpropagation through time)



Information Maximizing GANs (1)

- ❑ **Генеративные состязательные сети, максимизирующие информацию** (Information Maximizing GANs, InfoGAN) - теоретико-информационное расширение генеративных моделей, которые способны строить распутанные признаковые представления без учителя
- ❑ Распутанное признаковое представление - набор признаков, которые явно представляют характерные особенности экземпляра данных и могут быть полезны для широко круга задач
- ❑ Цель моделей - построить значимые представления посредством максимизации взаимной информации между небольшим подмножеством переменных шума и наблюдений

* Chen X., et al. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets // Advances in Neural Information Processing Systems. - 2016. - P. 2172-2180. - [<https://arxiv.org/pdf/1606.03657.pdf>].

Information Maximizing GANs (2)

