



УНИВЕРСИТЕТ
ИСКУССТВЕННОГО
ИНТЕЛЛЕКТА

Генеративные сопоставительные сети Generative Adversarial Networks (GAN)





Генеративные состязательные сети (GAN)

Назначение генеративных сетей

Генеративные нейронные сети предназначены для создания новых реалистичных объектов из случайного распределения (шума).

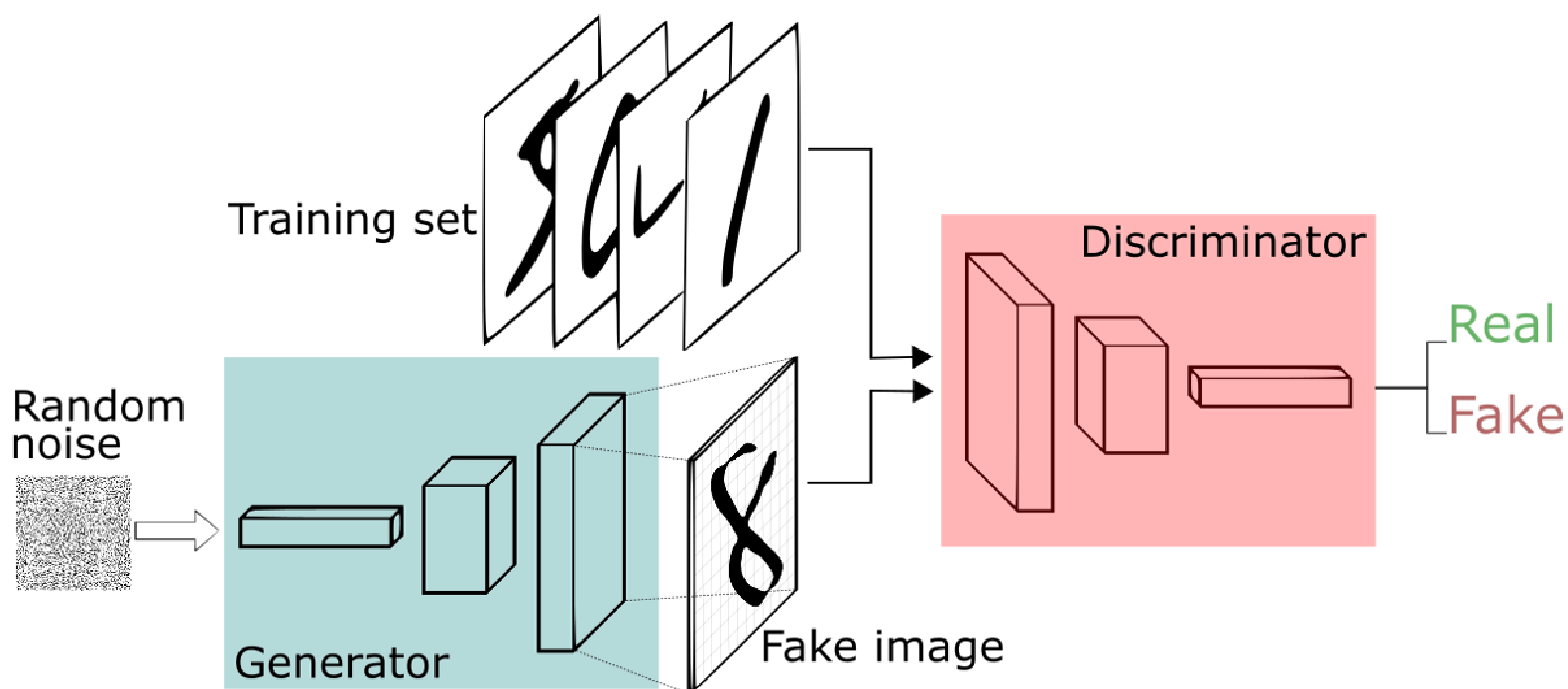
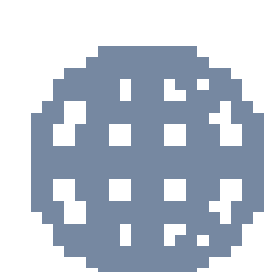


Рисунок 1. Принцип работы генеративно-состязательной нейронной сети



Генеративные состязательные сети (GAN)

Архитектура GAN

Архитектура состоит из двух нейронных сетей:

1. Генератор (Gen) набора числовых массивов размерности соответствующей размерности реальных объектов, получающий на вход случайные числа из заданного распределения $P(Z)$.
2. Дискриминатор (Dis) – классификатор реальных и сгенерированных объектов – необходим на этапе обучения сети.

Генератор учится генерировать объекты, похожие на реальные. Дискриминатор учится отличать сгенерированные объекты от реальных. В этом заключается принцип состязательности.

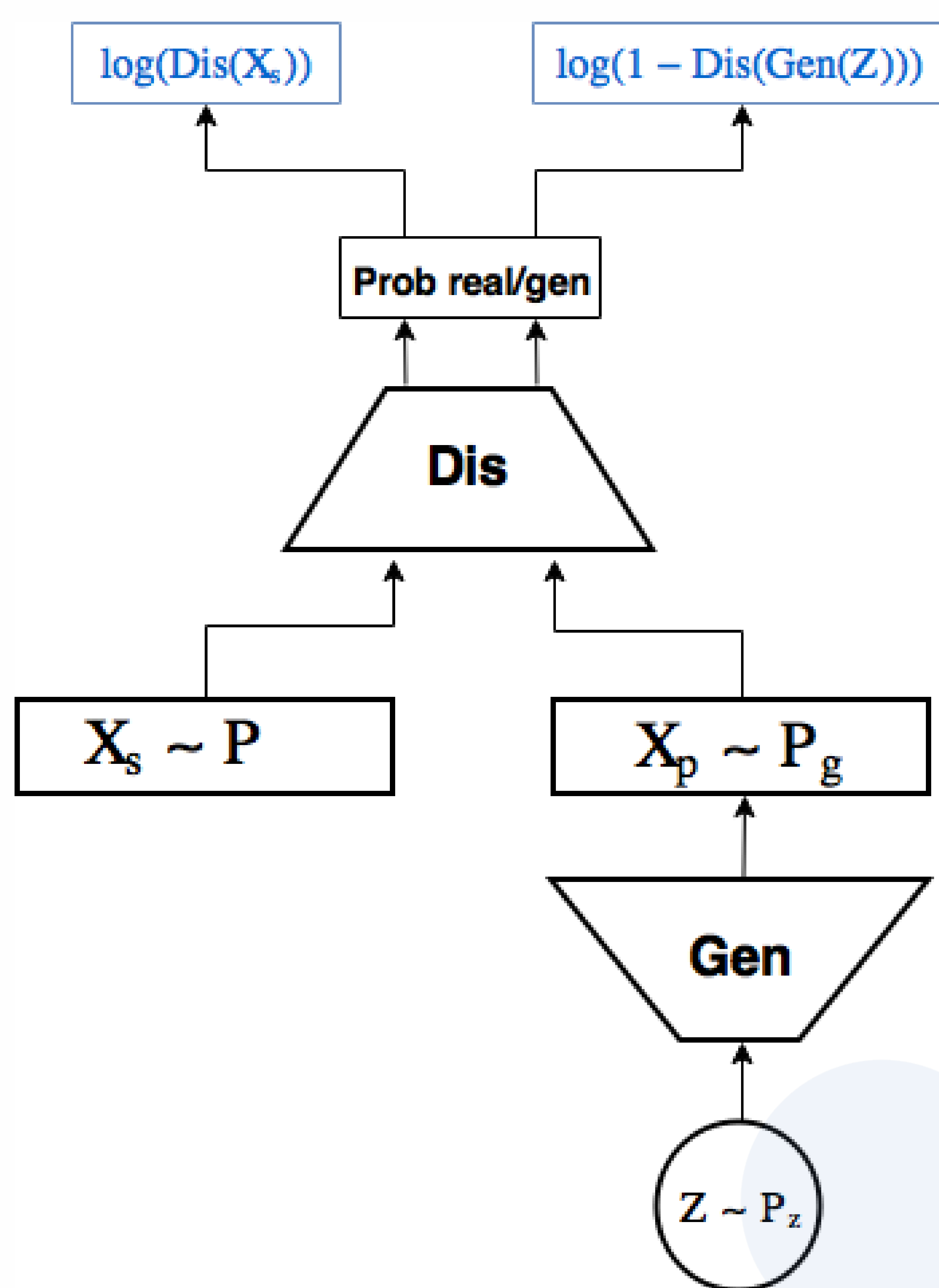


Рисунок 2. Принципиальная схема генеративно-состязательной нейронной сети

Генеративные состязательные сети (GAN)

Обучение GAN

Генератор и дискриминатор обучаются поочерёдно, как части одной сети. Математически эта задача формулируется так:

$$\min_G \max_D \mathbb{E}_{X \sim P} [\log(D(X))] + \mathbb{E}_{Z \sim P_z} [\log(1 - D(G(Z)))]$$

Обучение дискриминатора

Обучение дискриминатора заключается в минимизации функции перекрестной энтропии:

$$\theta_d = \theta_d - \nabla_{\theta_d} (\log(D(X_s)) + \log(1 - D(G(Z))))$$

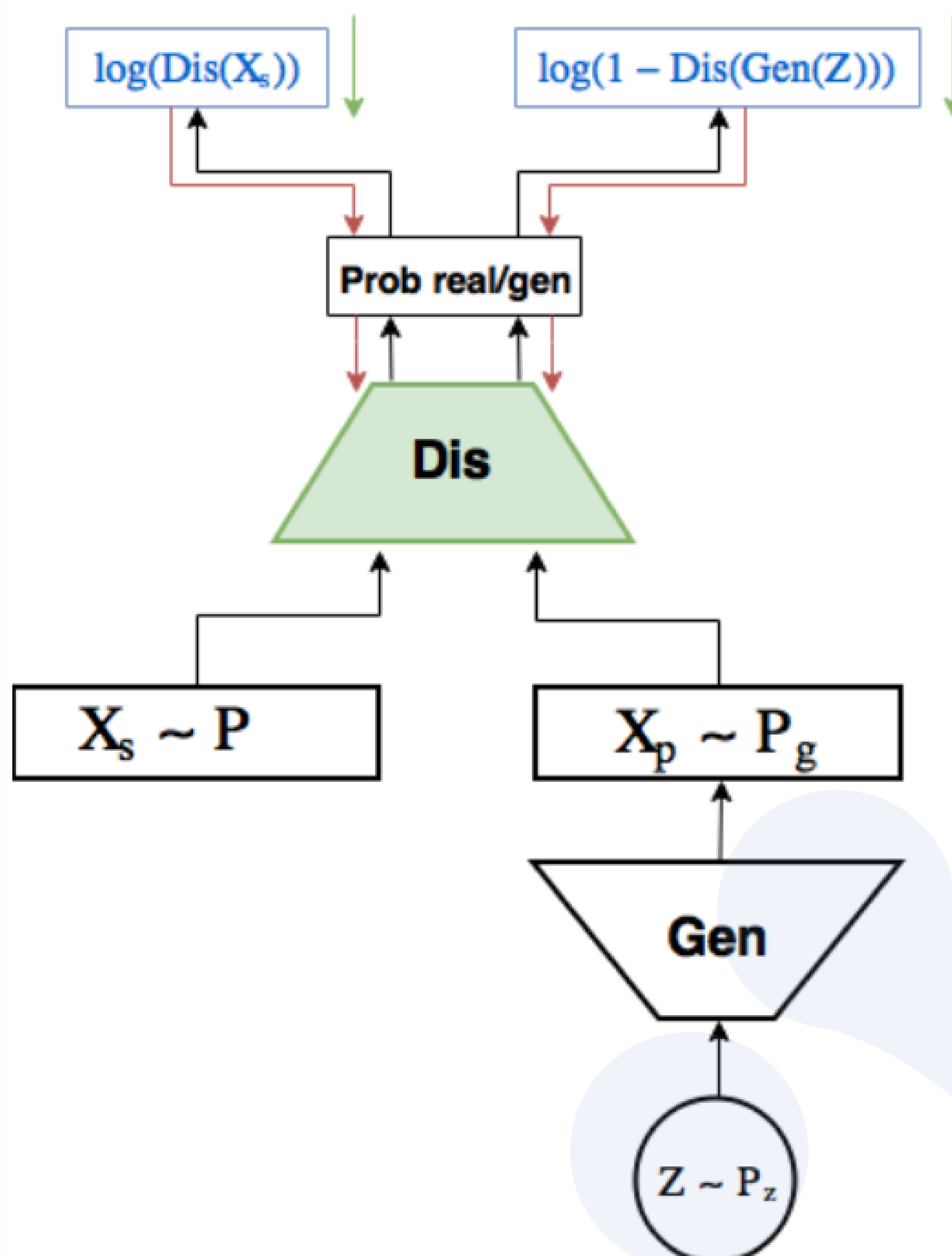


Рисунок 3. Обучение дискриминатора

Генеративные состязательные сети (GAN)

На этом этапе параметры генератора не изменяются, а на вход дискриминатора подается два набора данных: обучающий из реальных объектов X_s с метками 1 и набор сгенерированных объектов $X_p = G(Z)$ с метками 0.

Алгоритм обучения дискриминатора:

1. Генерация пакета X_p из случайного распределения.
2. Выбор пакета X_s из обучающего набора реальных объектов.
3. Градиентный шаг на X_p ([ссылка на документацию](#)).
4. Градиентный шаг на X_s .
5. Усреднение потерь для каждого шага.

Обучение генератора

Обучение генератора заключается в максимизации функции логарифма вероятности присвоения дискриминатором сгенерированному объекту лейбла реального объекта:

$$\theta_g = \theta_g + \nabla_{\theta_g} \log(1 - D(G(Z)))$$

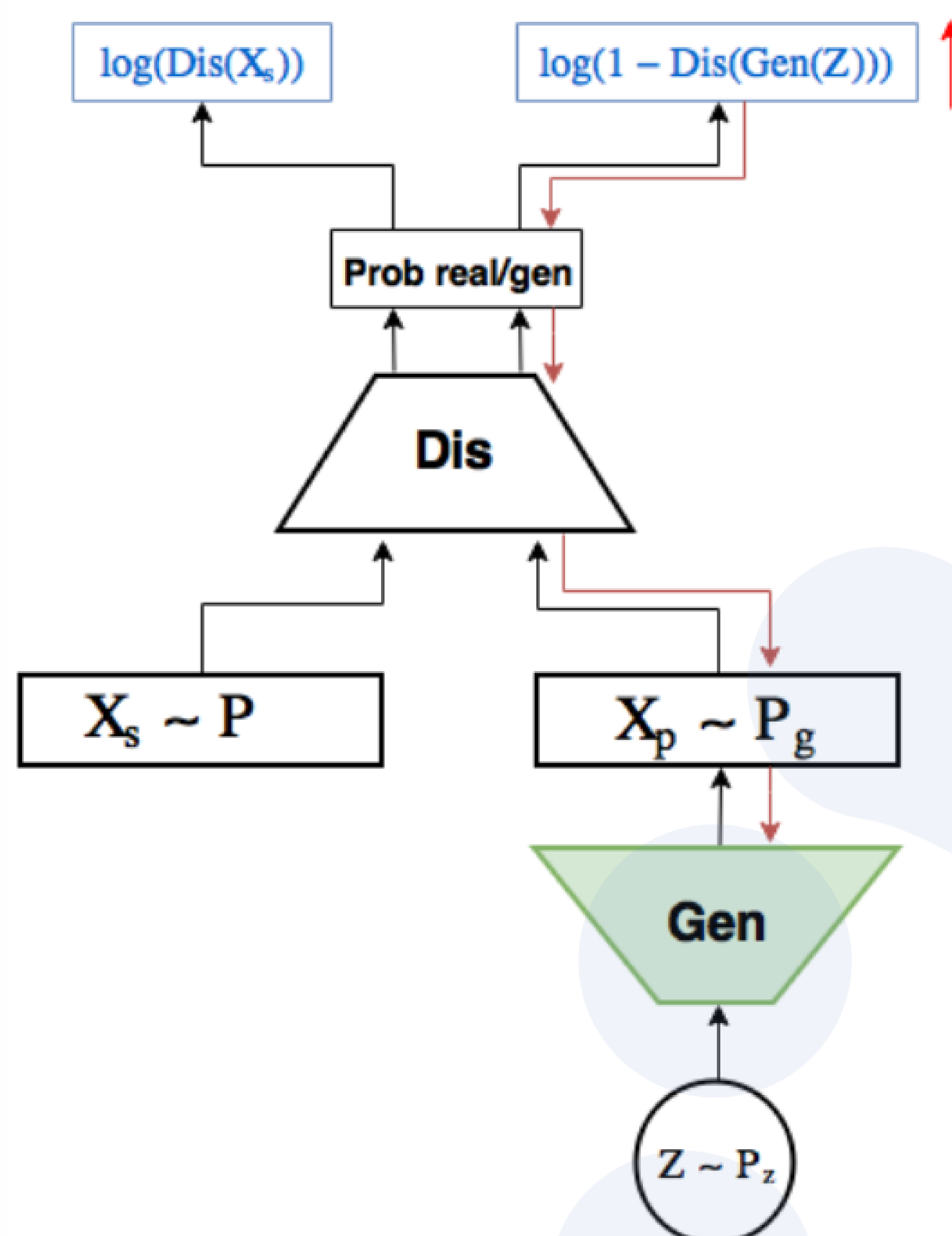


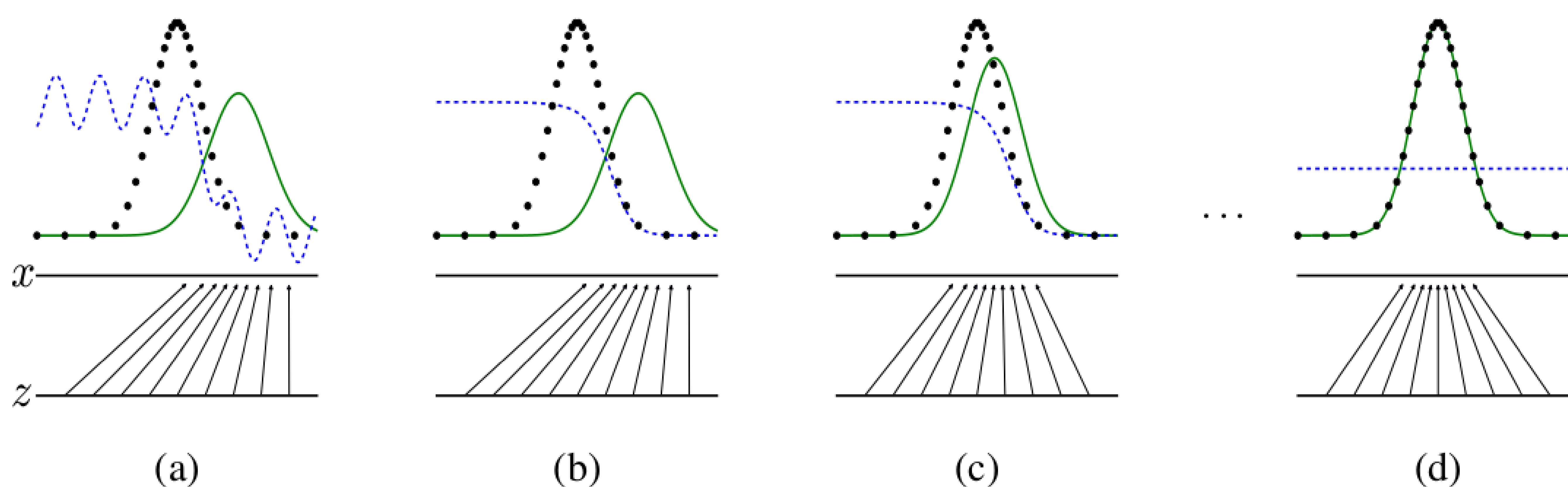
Рисунок 4. Обучение генератора

Генеративные состязательные сети (GAN)

Генератор обучается в объединении с дискриминатором, однако параметры дискриминатора на этом этапе обучения должны оставаться неизменными. Такого поведения можно добиться, если логическому атрибуту дискриминатора **trainable** задать значение **False** ([ссылка на документацию](#)).

На вход объединенной модели подаётся пакет из случайного распределения, а на выход – вектор из единиц – меток реальных объектов. Так функция генератора будет приближаться к распределению обучающей выборки.

Этапы обучения дискриминатора и генератора повторяются в идеальном случае до достижения оптимального состояния, при котором дискриминатор для всех X предсказывает 0.5, а распределение генерируемых объектов совпадает с распределением реальных объектов из обучающей выборки (рисунок 5).



..... - распределение реальных объектов обучающей выборки
——— - распределение сгенерированных объектов
..... - распределение вероятности распознавания дискриминатором реальных и сгенерированных объектов

Рисунок 5. Совпадение распределений сгенерированных и реальных объектов

Conditional GAN

Генеративно-состязательные сети с условием

Для генерации объектов с заданными параметрами архитектуру GAN можно дополнить входами для условия, которым может быть метка класса, маска и т. п. (рисунок 6).

Генеративные состязательные сети (GAN)

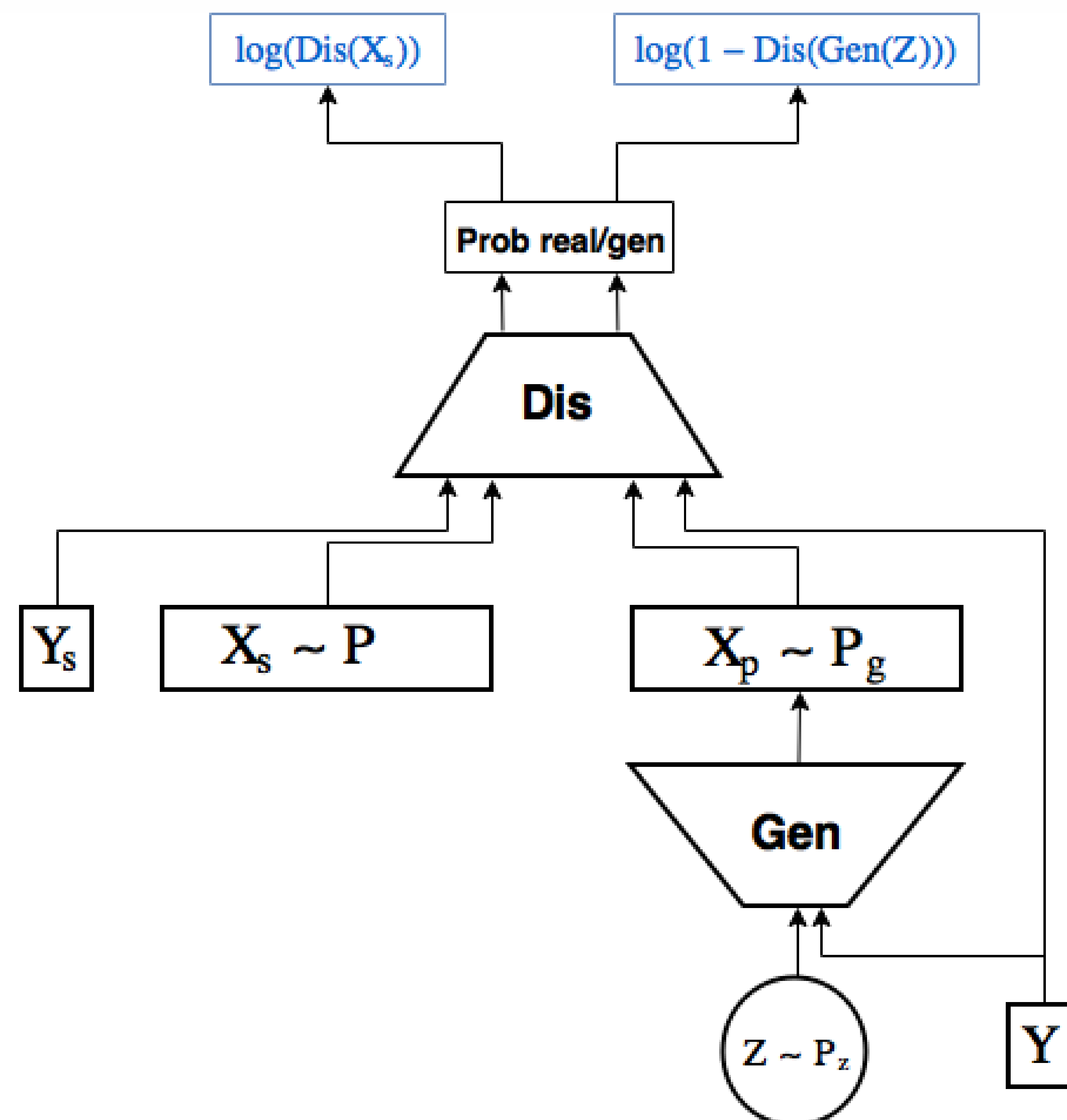


Рисунок 6. Принципиальная схема генеративно-состязательной нейронной сети с условием (CGAN)