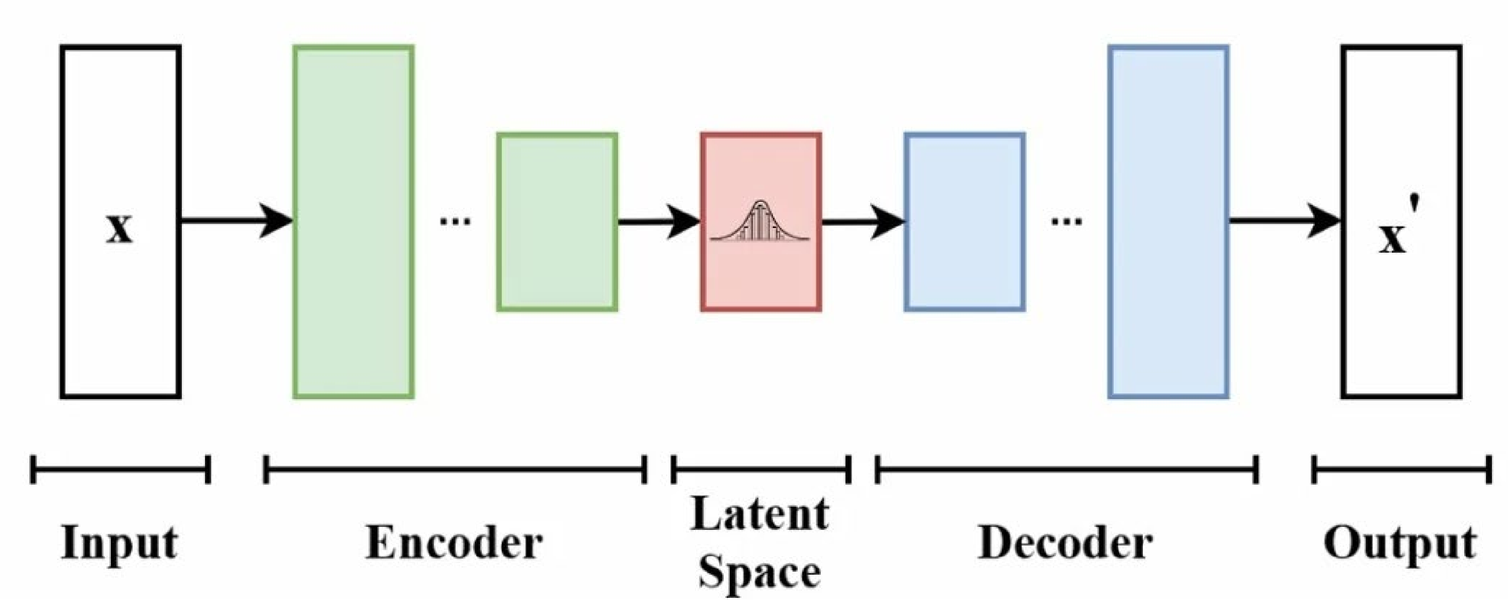
МИСИС
УНИВЕРСИТЕТ

Генеративные состязательные сети

Корнет Мария Евгеньевна
старший преподаватель кафедры
Инженерная кибернетика

Варианционный автокодировщик

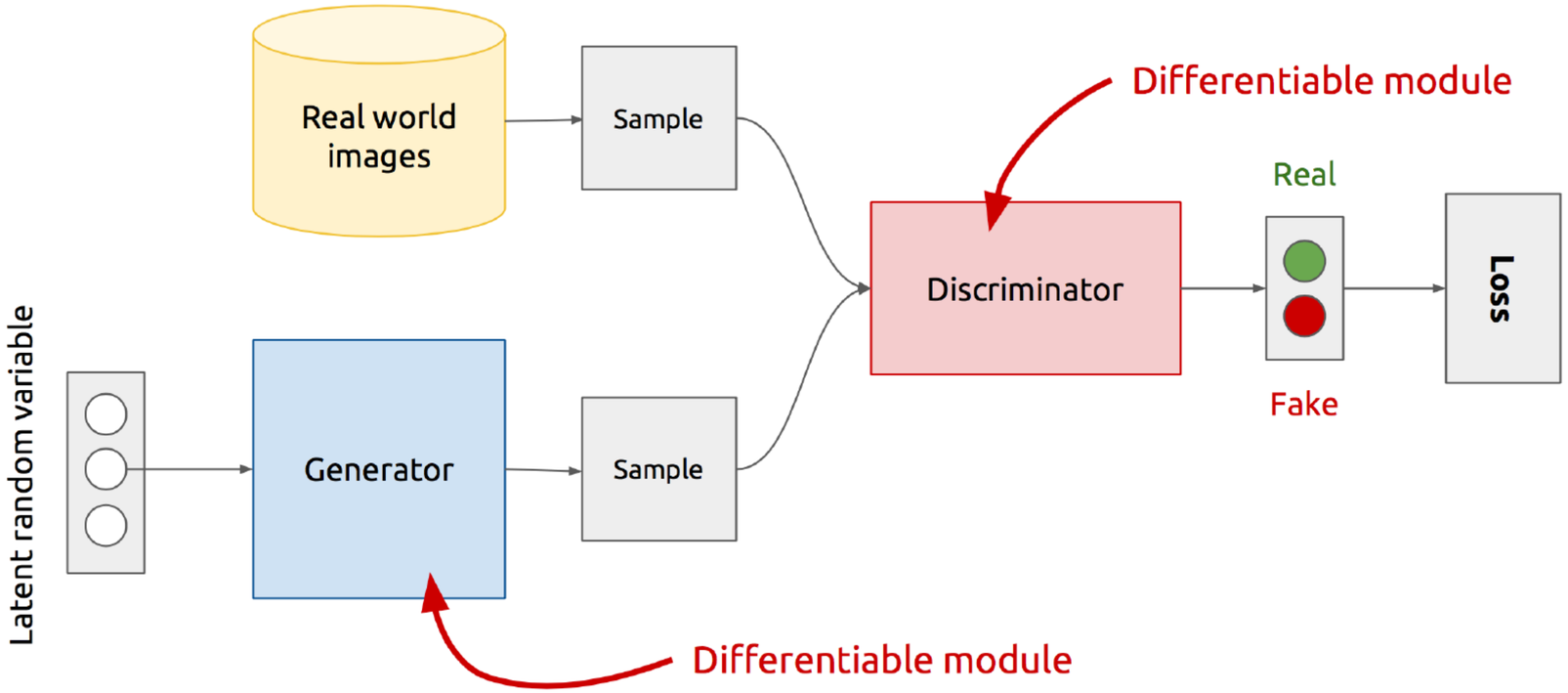


Проблемы VAE:

- Размытые изображения;
- Моделирование плотности в явном виде p(x|z);
- Компромисс между правдоподобием и качеством семплов.

Генеративные состязательные сети

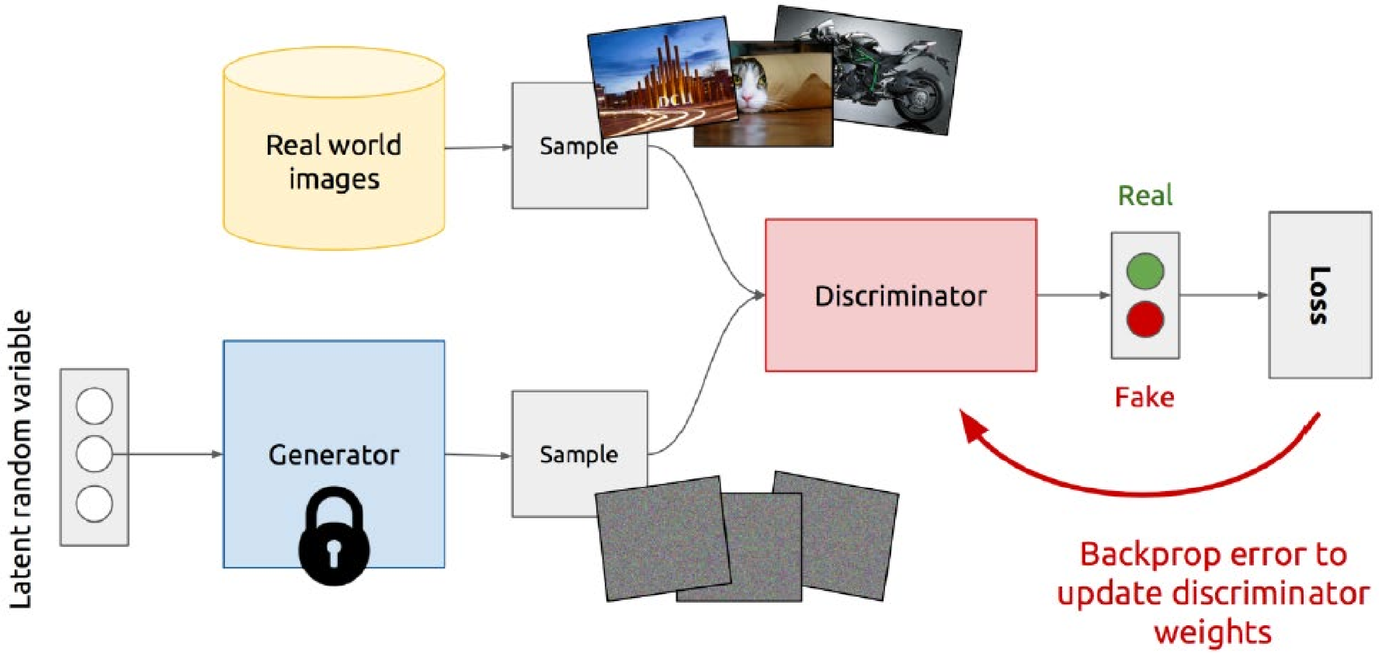
Генеративно-состязательные сети (Generative Adversarial Networks, GAN) – это класс генеративных моделей, общая черта которых заключается в том, что они обучаются одновременно с другой сетью, которая старается отличить сгенерированные объекты от настоящих.
2014г. **Ян Гудфеллоу** (Ian Goodfellow)



Генератор порождает модель из шума

Дискриминатор – сеть различающая настоящие и сгенерированные объекты.

Обучение дискриминатора



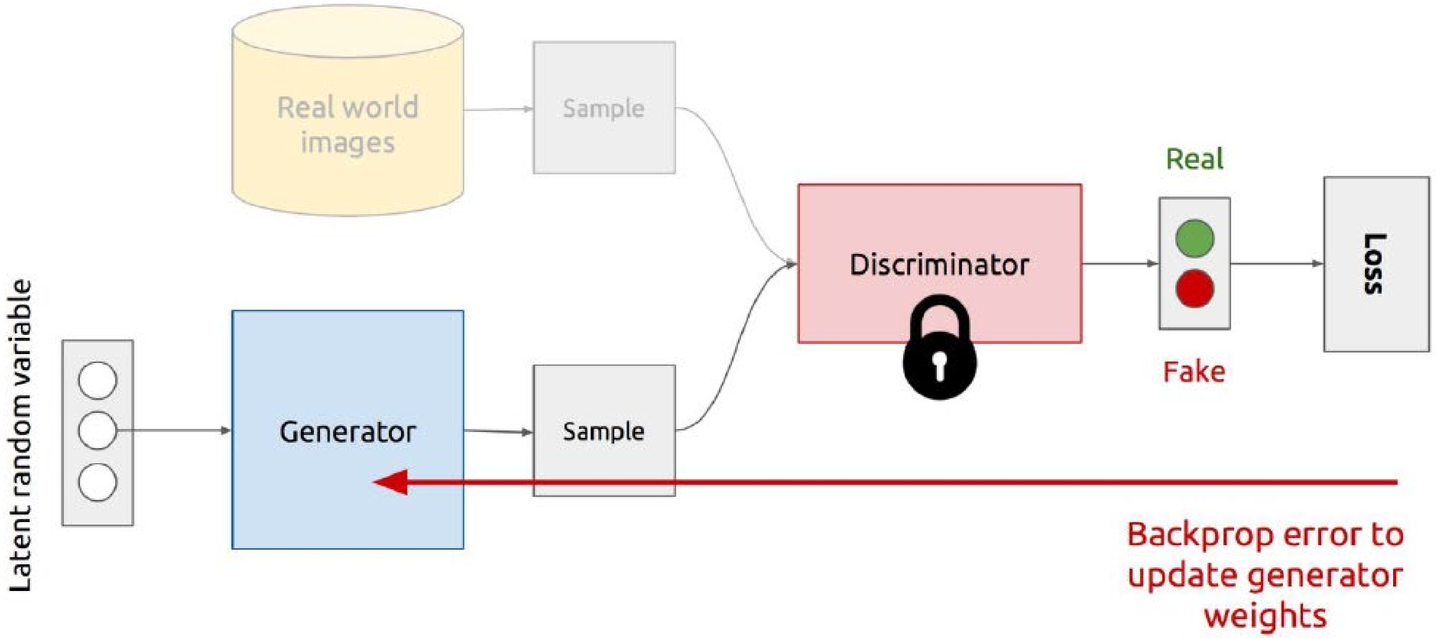
1. Берём **батч реальных изображений** из датасета.

2. Берём батч случайных векторов из латентного пространства и пропускаем их через **генератор** — получаем **фейковые картинки**.

3. Склеиваем реальный и фейковый батчи и подаём всё это на **дискриминатор.**

4. Считаем loss.

Обучение генератора



1. Снова берём случайный шум и пропускаем его через **генератор**, получая новые фейковые изображения.
2. Подаём эти фейковые картинки на дискриминатор, **но теперь метка-цель = 1 (Real)**

3. Считаем потерю генератора как ошибку дискриминатора на этих фейковых примерах.

4. Делаем backprop через дискриминатор к генератору, но обновляем только веса генератора

Обучение GAN

Обучение GAN формулируется как задача минимакс-оптимизации:

$$\min_{\theta} \max_{\phi} V(G_\theta, D_\phi) = \min_{\theta} \max_{\phi} \left[\mathbb{E}_{x \sim p_{\text{data}}} \log D_\phi(x) + \mathbb{E}_{z \sim p(z)} \log\big(1 - D_\phi(G_\theta(z))\big) \right]$$

$\mathbb{E}_{x \sim p_{\text{data}}} \log D_\phi(x)$: дискриминатор должен давать **высокое правдоподобие** (близко к 1) для настоящих выборок;

$\mathbb{E}_{z \sim p(z)} \log(1 - D_\phi(G_\theta(z)))$: **низкое правдоподобие** (близко к 0) для фейковых примеров от генератора.

Дискриминатор учится делать: $D_\phi(x) \to 1, \quad D_\phi(G_\theta(z)) \to 0.$

Генератор, наоборот хочет чтобы дискриминатор ошибался: $D_\phi(G_\theta(z)) \to 1$

То есть генератор подбирает такие изображения, которые дискриминатор считает реальными.

Особенности оптимизации

На практике разделяют оптимизацию:

- шаг по $\emptyset$: максимизация стандартной логистической функции потерь дискриминатора;
- шаг по θ: вместо минимизации $\log(1 - D_\phi(G_\theta(z)))$ обычно максимизируют $\log D_\phi(G_\theta(z))$

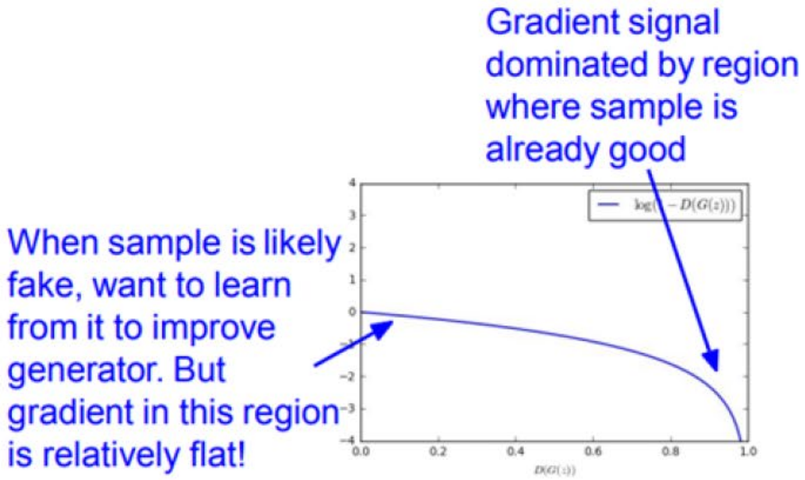
Если бы мы знали истинные распределения, то **идеальный дискриминатор**:

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

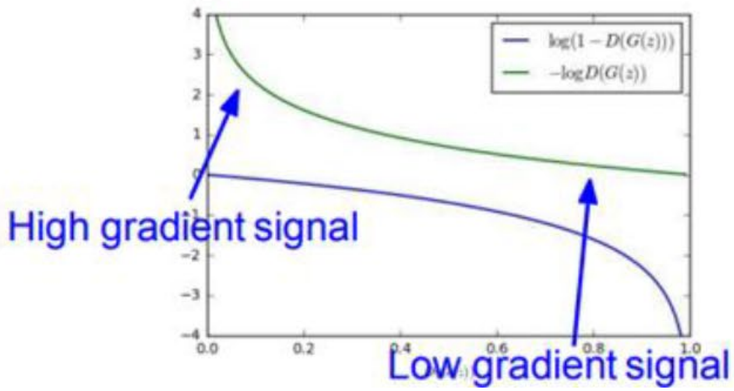
Он просто сравнивает, из какого распределения точка более вероятна — из настоящих данных или из генератор

Особенности оптимизации

$$\min_{\theta} \mathbf{E}_{z \sim p(z)} \log(1 - D_{\varphi}(G_{\theta}(x)))$$



$$\max_{\theta} \mathbf{E}_{z \sim p(z)} \log(D_{\varphi}(G_{\theta}(x)))$$



https://arxiv.org/abs/1406.2661

Goodfellow, Ian, et al. «Generative adversarial nets» Advances in neural information processing systems. 2014.

Примеры GAN

Проблемы GAN:

- Сложно обучать;
- Проблемы с Batchnorm;
- Mode collapse.

https://arxiv.org/abs/1406.2661
Goodfellow, Ian, et al. «Generative adversarial nets» Advances in neural information processing systems. 2014.

Mode collapse

Mode collapse – это ситуация, когда генератор фактически “забывает” большую часть распределения и **концентрируется на узком подмножестве мод**.

Основные признаки:

- при просмотре сэмплов мы видим почти одинаковые картинки;
- изменение латентного вектора даёт слабо отличимые вариации;
- генератор плохо реагирует на условную информацию (в conditional GAN выдаёт один класс вне зависимости от метки).

Способы преодоления mode collapse

Minibatch discrimination, minibatch features – дискриминатор смотрит не только на отдельные примеры, но и на разнообразие в батче, “наказывает” генератор за порождение слишком похожих образцов.
Feature matching – генератор оптимизирует не только “обман дискриминатора”, но и совпадение статистик скрытых признаков дискриминатора между реальными и фейковыми данными.
Unrolled GAN – при обновлении генератора учитывают несколько виртуальных шагов обновления дискриминатора, что делает игру более “дальновидной” и уменьшает склонность к коллапсу.
Модифицированные функции потерь и архитектуры – WGAN / WGAN-GP, спектральная нормализация, различные градиентные штрафы; они стабилизируют обучение и дают более информативные градиенты.
Дополнительные «награды» за разнообразие – например, InfoGAN (максимизация взаимной информации между частью латентного кода и наблюдаемыми факторами), mode-seeking потери и др.

Настройка GAN

1. Нормализация входа и выхода.

2. Выбор функция потерь генератора (hinge, Wasserstein) для стабильности градиентов.
3. Распределение шума (из гауссовского распределения).

4. Для downsampling использовать *Conv2d с шагом* или *average pooling*, а не MaxPooling – он даёт более “жёсткие” градиенты.

5. Для upsampling — ConvTranspose2d или upsample+Conv, а не сложные сверточные пирамиды, если вы не делаете StyleGAN-подобную модель.

6. В дискриминаторе лучше использовать LeakyReLU (или другие нелинейности с ненулевым градиентом слева), чтобы избежать больших областей нулевого градиента.

7. Регуляризация дискриминатора: Spectral Norm, Gradient Penalty.
8. Label smoothing и добавление шума.
9. В генераторе BatchNorm/InstanceNorm часто полезно. В дискриминаторе всё чаще **НЕ** используют BatchNorm.

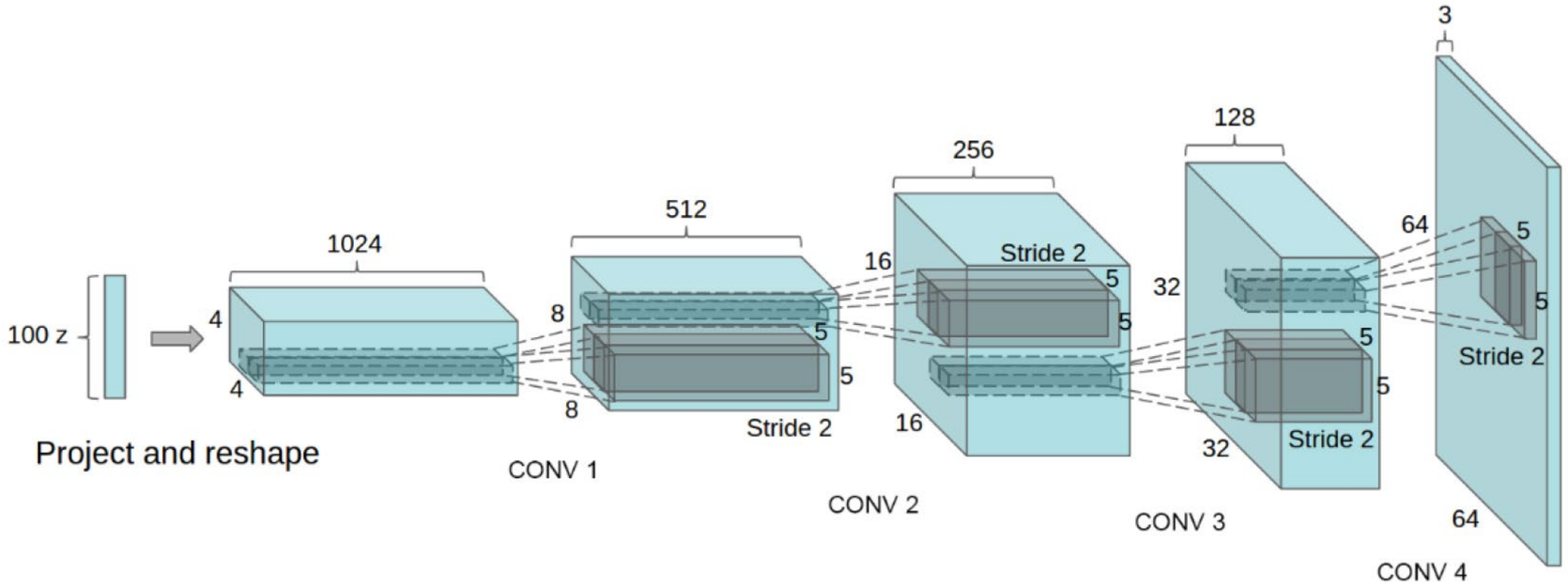
Выбор GAN

Для **низких разрешений** (MNIST/CIFAR-10, 64×64) — DCGAN / ResNet-GAN как сильная и простая база.
Для **фото высокого качества** — StyleGAN-подобные архитектуры (progressive growing, style-blocks).
Для **специфических задач** (image-to-image, суперразрешение, перенос стилей) — специализированные GAN (Pix2Pix, CycleGAN, SRGAN и др.).

Функция потерь / Метрика	Архитектура GAN	Год	Особенности
Binary Cross-Entropy	Vanilla GAN	2014	Оригинальный GAN, нестабильное обучение
Least Squares (L2)	LSGAN	2016	Более стабильно, избегает saturation
Wasserstein-1 Distance	WGAN	2017	Улучшенная стабильность, требует clipping
Wasserstein-1 + Gradient Penalty	WGAN-GP	2017	Лучшая стабильность, penalty вместо clipping
Hinge Loss	BigGAN, StyleGAN	2018-2019	Используется в SOTA моделях
Hinge Loss (двойной)	SAGAN	2018	Self-Attention GAN
Logistic Loss	StyleGAN2	2019	Улучшенная стабильность StyleGAN
Contrastive Loss	InfoGAN	2016	Контролируемая генерация
Maximum Mean Discrepancy (MMD)	MMD GAN	2017	Альтернатива JS-дивергенции
f-divergences	f-GAN	2016	Обобщение разных дивергенций
Relativistic Loss	RaGAN	2018	Относительные оценки
Energy-Based	EBGAN	2016	Energy-based подход
Boundary Equilibrium	BEGAN	2017	Автоэнкодер в дискриминаторе
Patch-based	PatchGAN	2016	Работает с патчами изображения
Projection Discriminator	cGAN, ProjGAN	2018	Улучшенный conditional GAN
Consistency Regularization	Diffusion GAN	2022	Комбинация диффузий и GAN
Progressive Growing	ProGAN	2017	Постепенное увеличение разрешения
Diffusion + Adversarial	DiGAN	2023	Гибрид диффузии и GAN
Latent Diffusion + GAN	LD-GAN	2022	GAN в латентном пространстве VQGAN
Transformer-based	GANsformer	2021	Трансформеры в GAN
Vision Transformer	ViTGAN	2021	ViT в качестве дискриминатора
Transformer Generator	TransGAN	2021	Полностью трансформерная архитектура

DGAN

Deep Convolutional GAN (DCGAN) — это классическая архитектура, которая показала, что **обычные сверточные сети** можно успешно использовать не только для распознавания, но и для генерации фотореалистичных изображений.



DGAN

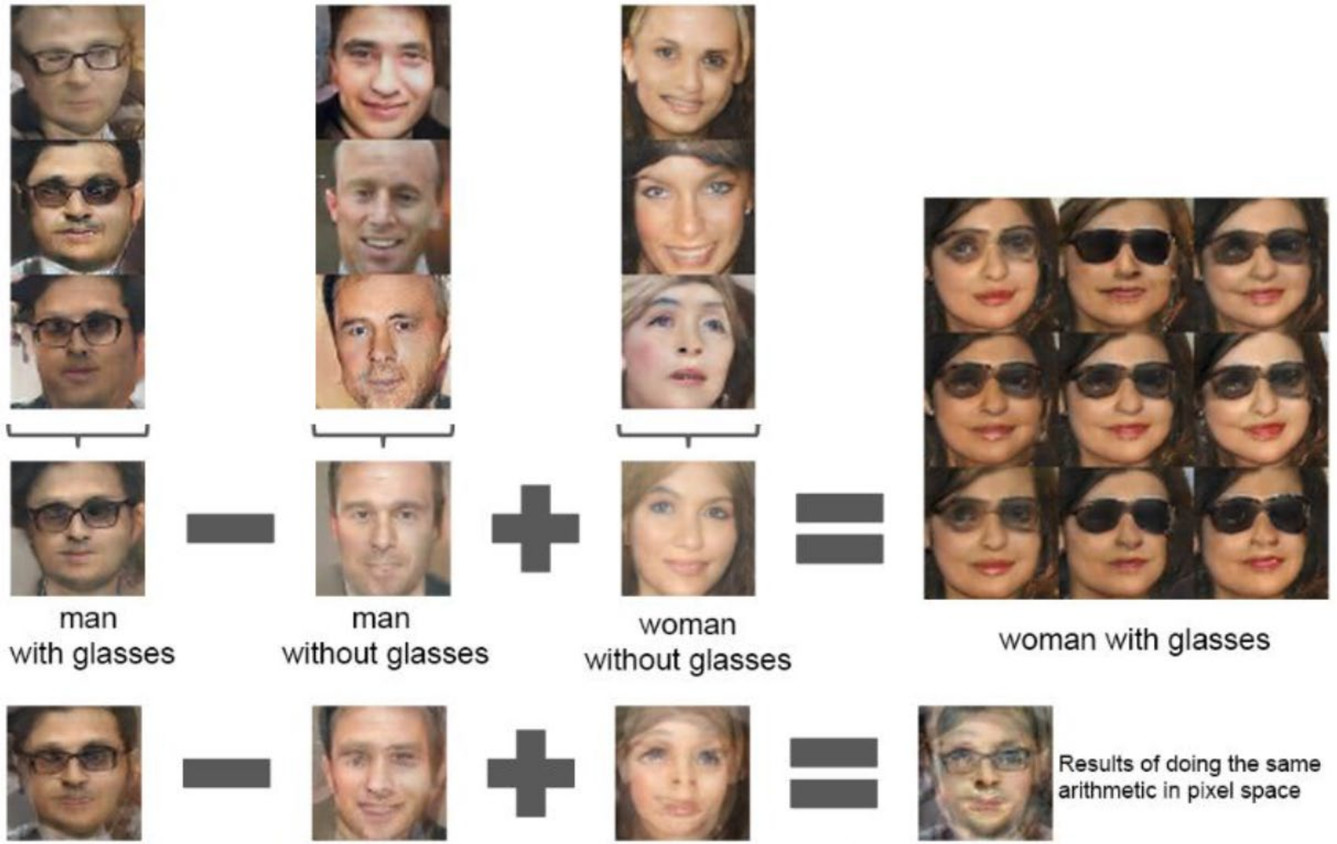
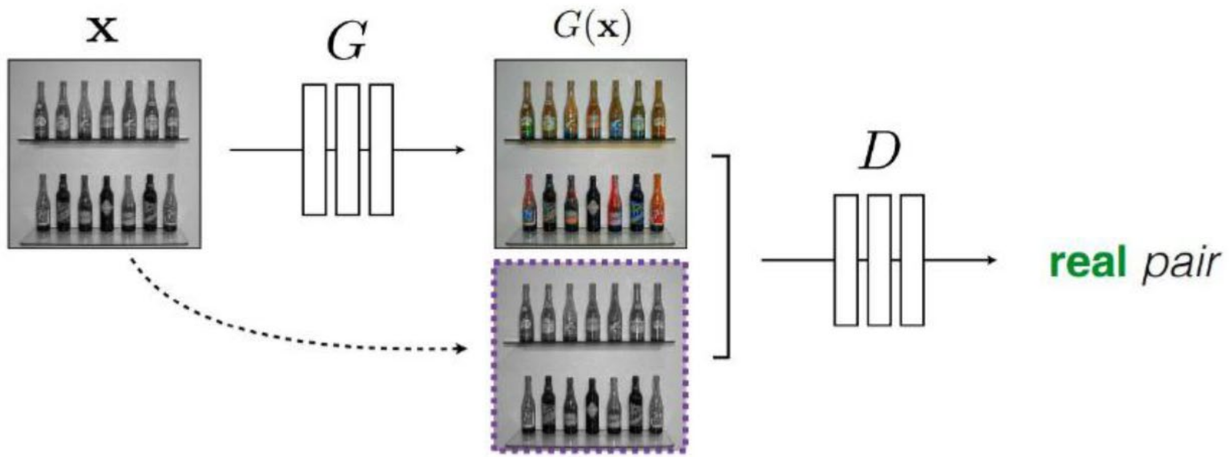


Figure 7: Vector arithmetic for visual concepts. For each column, the Z vectors of samples are averaged. Arithmetic was then performed on the mean vectors creating a new vector Y. The center sample on the right hand side is produce by feeding Y as input to the generator. To demonstrate the interpolation capabilities of the generator, uniform noise sampled with scale +-0.25 was added to Y to produce the 8 other samples. Applying arithmetic in the input space (bottom two examples) results in noisy overlap due to misalignment.

Условные GAN

Условные GAN (cGAN) учат генератор не просто «из шума рисовать картинку», а **генерировать изображение при заданном условии**.



Pix2Pix

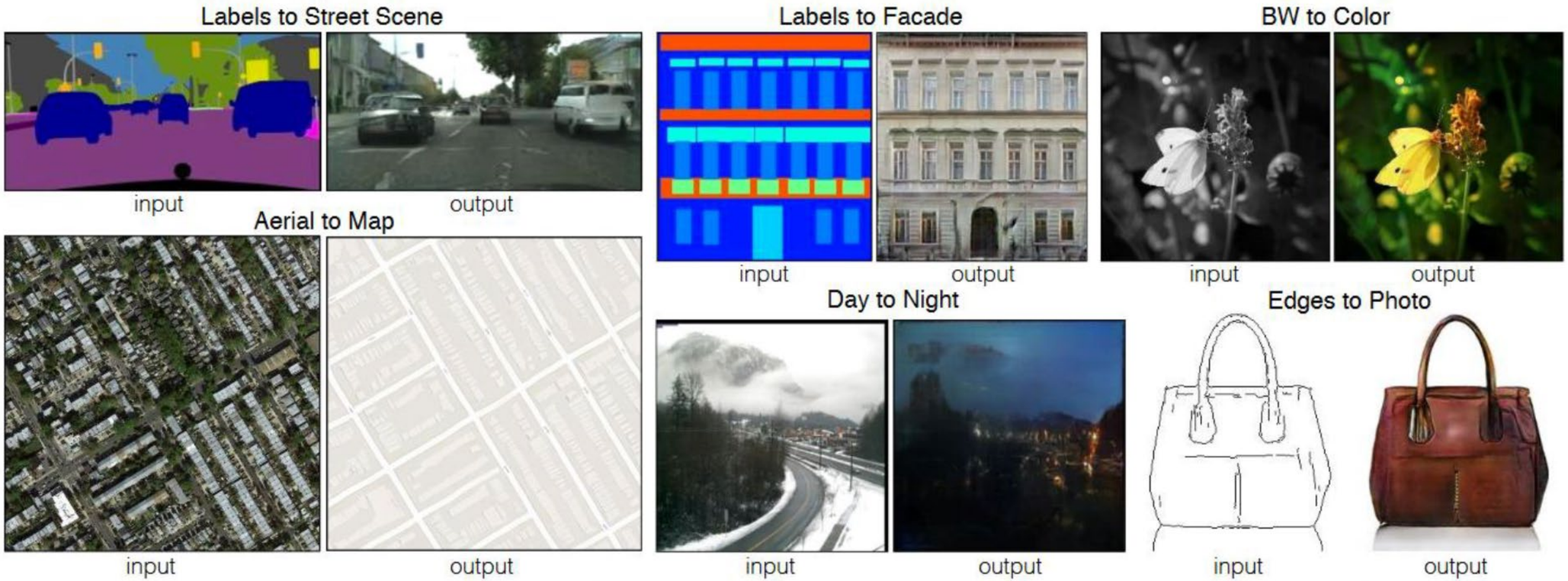
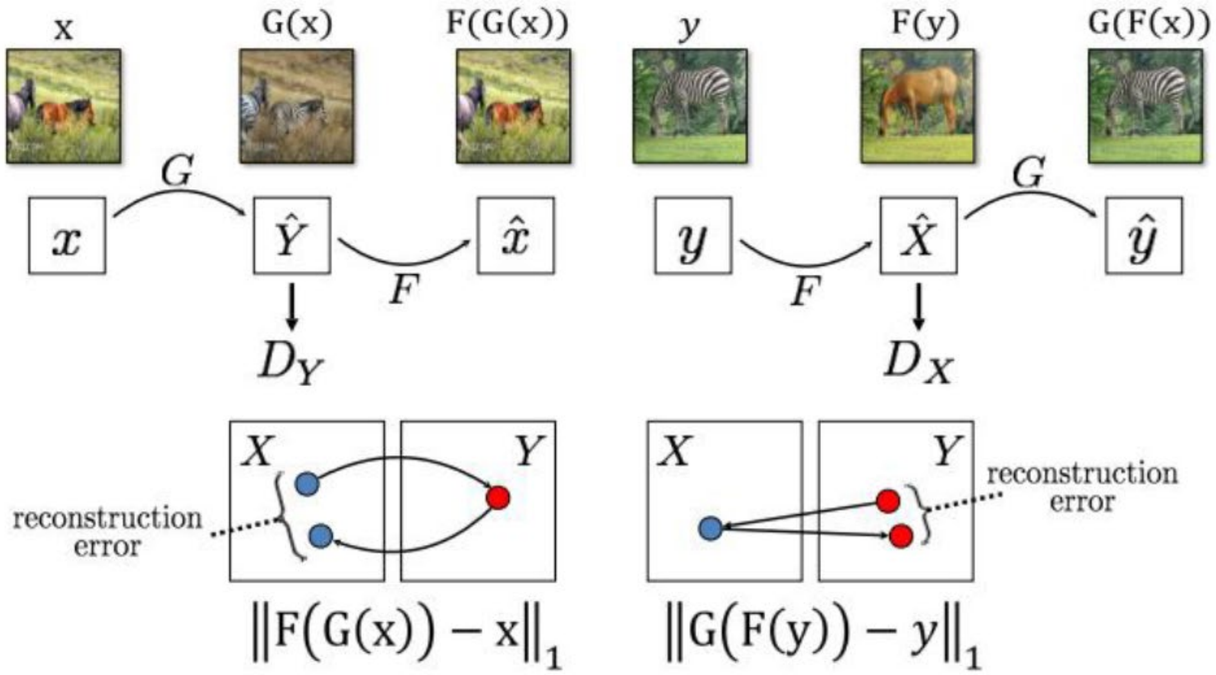


Figure 1: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

CycleGan

CycleGAN — это подход для **переноса стиля между двумя доменами без парных данных** (без соответствий “картинка A → картинка B”)



CycleGan

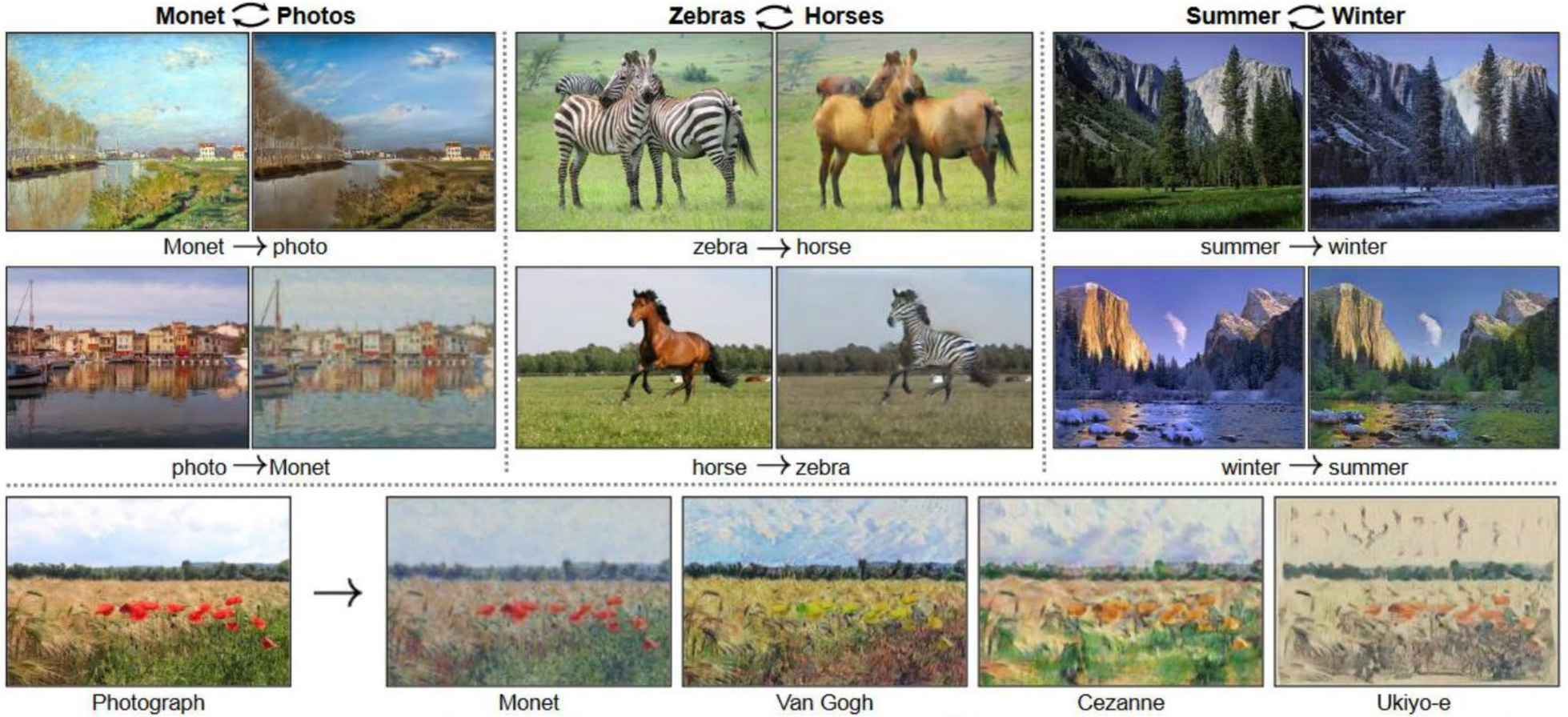


Figure 1: Given any two unordered image collections X and Y, our algorithm learns to automatically "translate" an image from one into the other and vice versa: *(left)* Monet paintings and landscape photos from Flickr; *(center)* zebras and horses from ImageNet; *(right)* summer and winter Yosemite photos from Flickr. Example application *(bottom)*: using a collection of paintings of famous artists, our method learns to render natural photographs into the respective styles.

ProGan

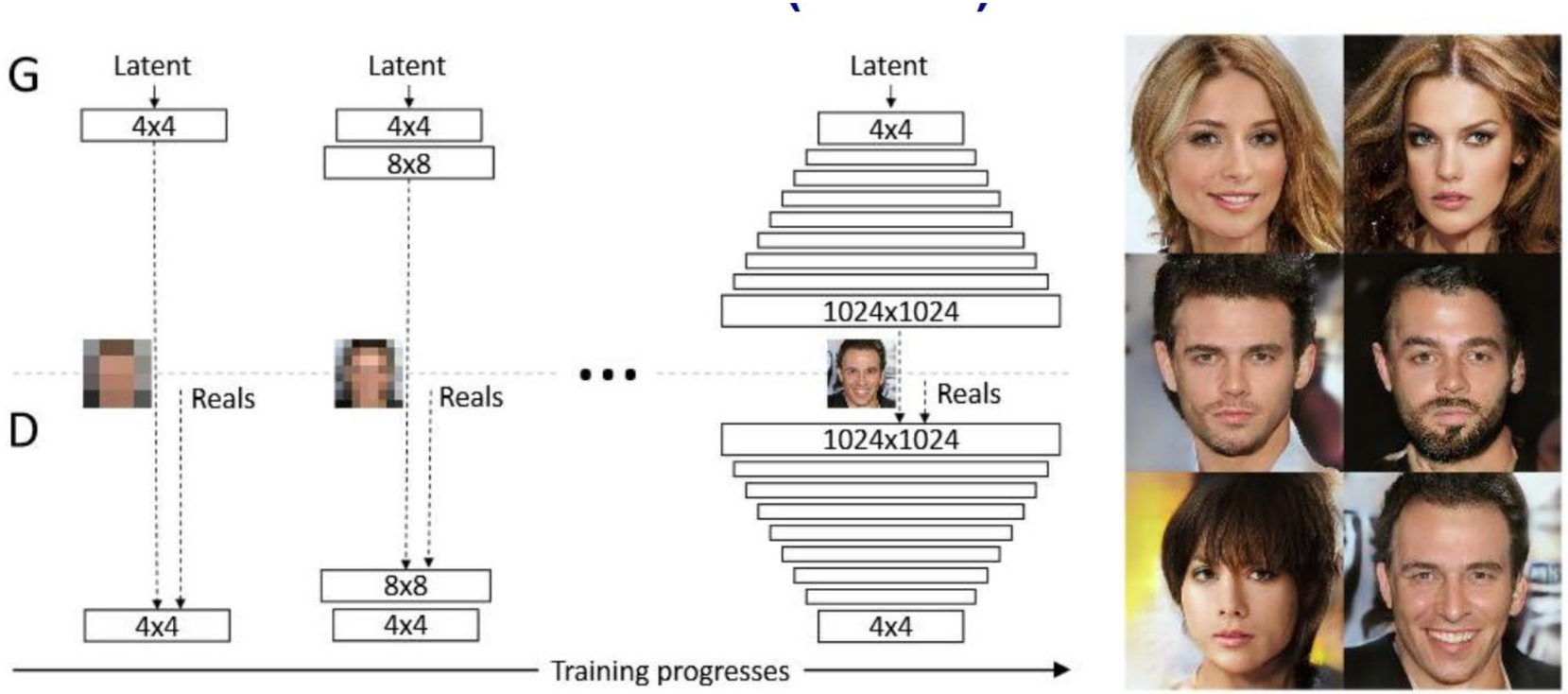


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $N \times N$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

МИСИС
УНИВЕРСИТЕТ

Спасибо
за внимание!