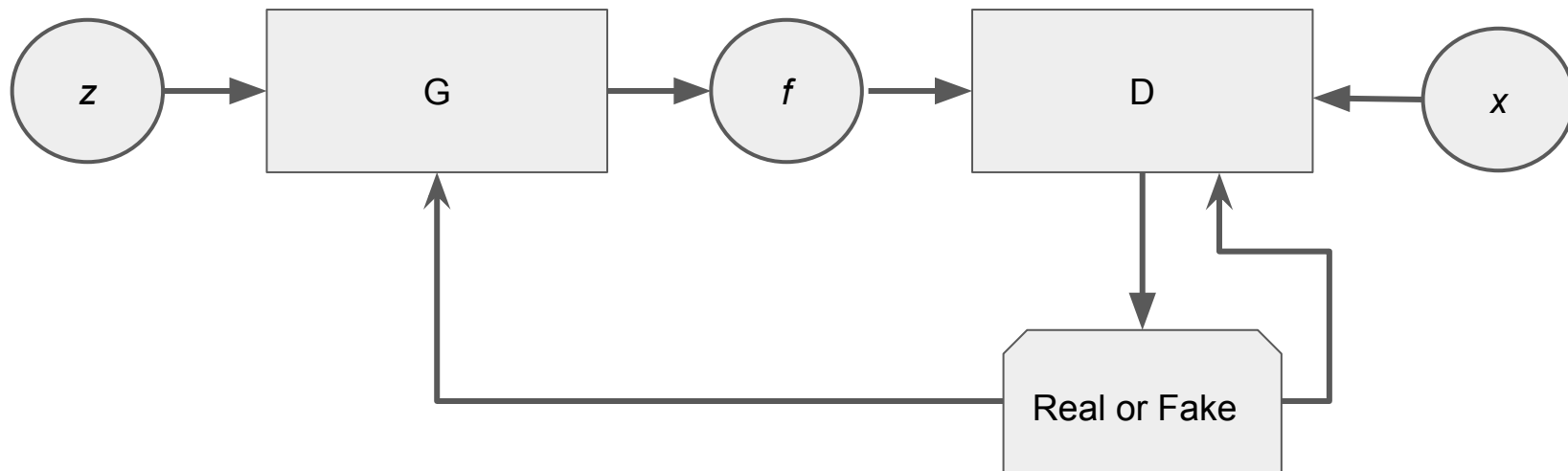


# Генеративно-состязательные сети

Концепция и трюки

# Концепция GAN



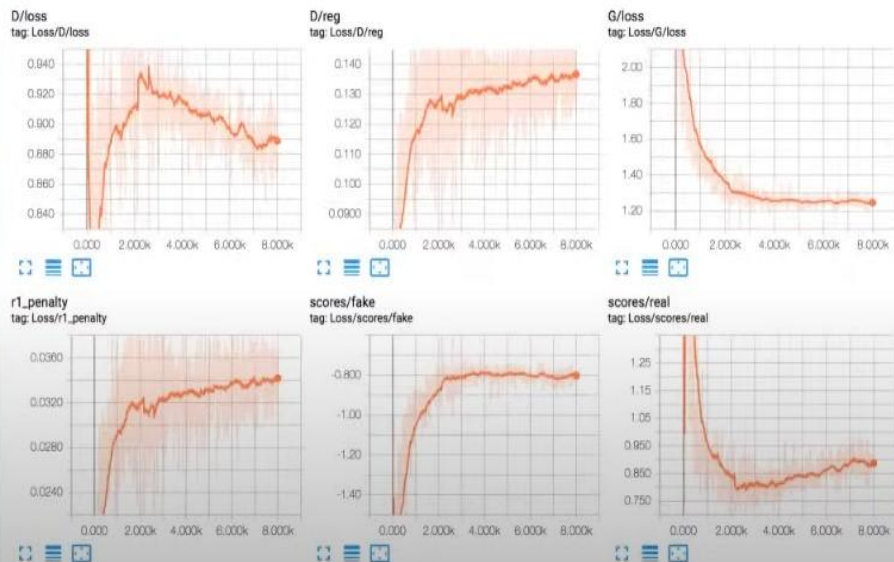
# GAN collaps

- mode collaps
- mode dropping
- divergence
- OOM



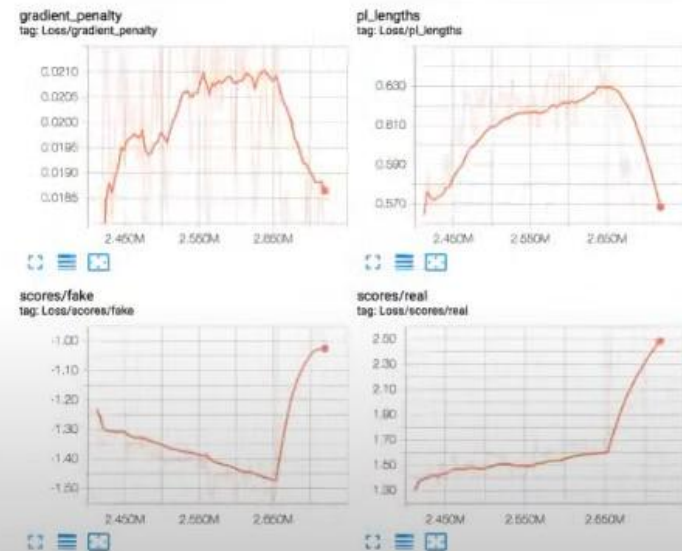
# GAN collaps

## (Almost) typical run



## Collapse

Loss



# GAN losses: standart

$$E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

1) D loss  $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[ \log D(\mathbf{x}^{(i)}) + \log(1 - D(G(\mathbf{z}^{(i)}))) \right]$

2) G loss  $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(\mathbf{z}^{(i)})))$

# GAN losses: non-saturating loss

$$\max \log(D(G(z)))$$

# GAN losses: Wasserstein loss

D loss 
$$\min_D - \left( \mathbb{E}_{x \sim p_X} [D(x)] - \mathbb{E}_{z \sim p_Z} [D(G(z))] \right).$$

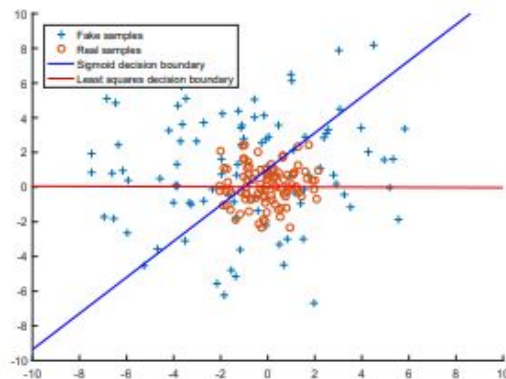
G loss 
$$\min_G - \left( \mathbb{E}_{z \sim p_Z} [D(G(z))] \right).$$

- вместо sigmoid у D липшецева непрерывная функция
- осмысленная функция потерь коррелирующая с качеством образцов и сходимостью генератора
- повышенная стабильность процесса оптимизации
- нужно обрезать веса
- не использовать BN

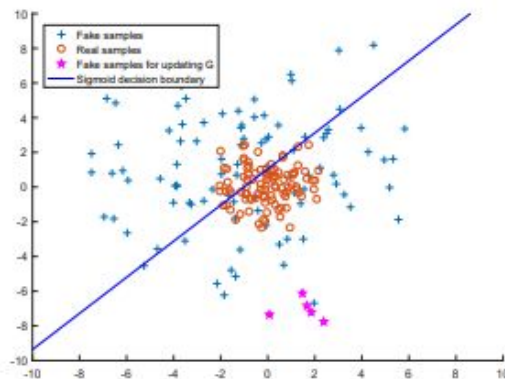
# GAN Least Squares Loss

$$\min_D V_{LS}(D) = \frac{1}{2} \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} \left[ (D(\mathbf{x}) - b)^2 \right] + \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p_{data}(\mathbf{z})} \left[ (D(G(\mathbf{z})) - a)^2 \right]$$

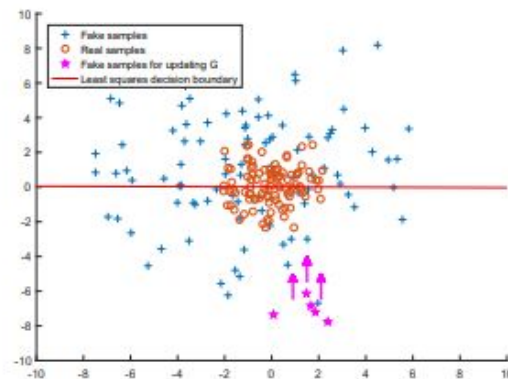
$$\min_G V_{LS}(G) = \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} \left[ (D(G(\mathbf{z})) - c)^2 \right]$$



(a)



(b)



(c)



# GAN Hinge Loss

$$L_D = -\mathbb{E}_{(x,y) \sim p_{data}} [\min(0, -1 + D(x, y))] - \mathbb{E}_{z \sim p_z, y \sim p_{data}} [\min(0, -1 - D(G(z), y))]$$

$$L_G = -\mathbb{E}_{z \sim p_z, y \sim p_{data}} D(G(z), y)$$

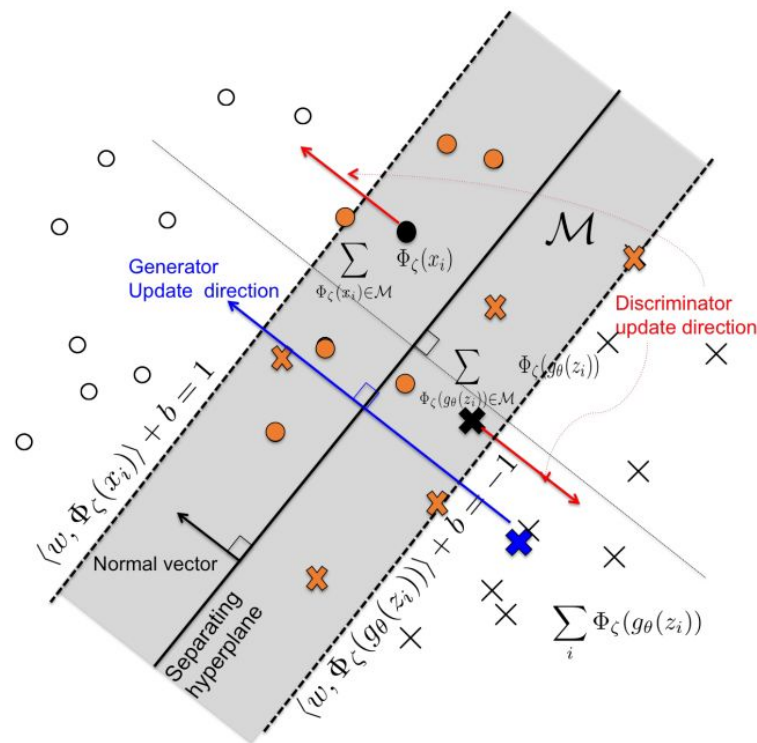
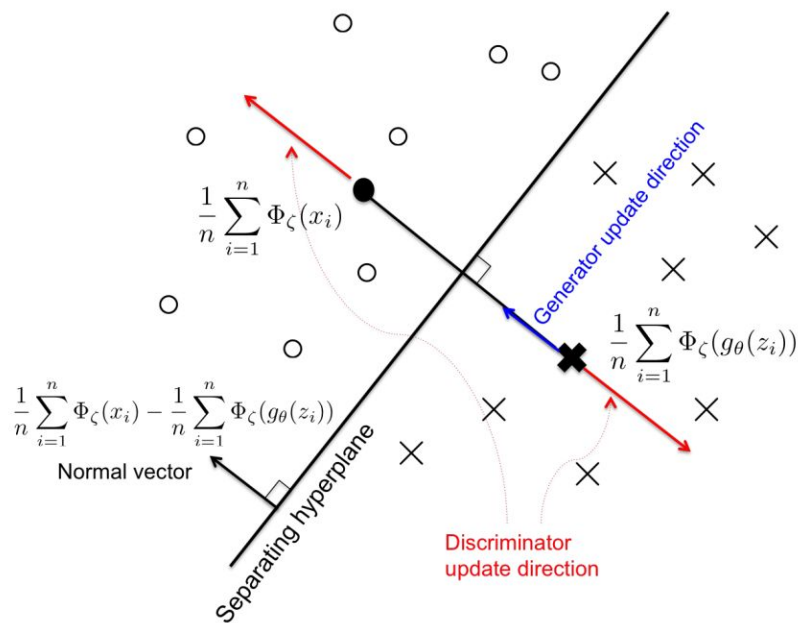


Figure 2: Geometry of the mean feature matching GAN.

# DRAGAN

$$\lambda \cdot \mathbb{E}_{x \sim P_{real}, \delta \sim N_d(0, cI)} [\|\nabla_{\mathbf{x}} D_{\theta}(x + \delta)\|^2]$$

$$\lambda \cdot \mathbb{E}_{x \sim P_{real}, \delta \sim N_d(0, cI)} [\max(0, \|\nabla_{\mathbf{x}} D_{\theta}(x + \delta)\|^2 - k)]$$

# История

- 1) 2014 64px Generative Adversarial Nets; Ian J. Goodfellow
- 2) 2016 128 px DCGAN
- 3) 2018 1024px ProGan
- 4) 2019 BigGan
- 5) StyleGAN 1-2-3
- 6) Stable Diffusion

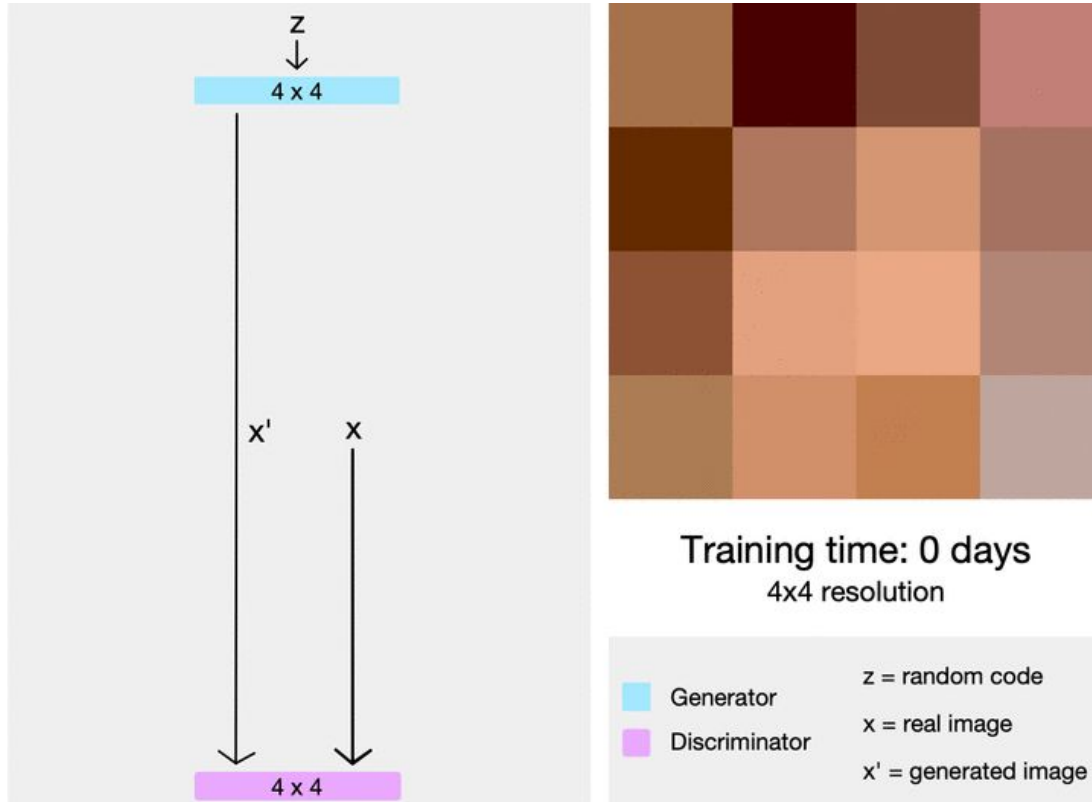
## Регуляризация

$$R_1 = \frac{\gamma}{2} \mathbb{E}_{x \sim p_d} [ \| \nabla D(x) \|^2 ]$$

$$R_2 = \frac{\gamma}{2} \mathbb{E}_{\hat{x} \sim p_g} [ \| \nabla D(x) \|^2 ]$$

$$R_{\text{LC}} = \mathbb{E}_{x \sim p_d} [ \| D(x) - \alpha_F \|^2 ] + \mathbb{E}_{\hat{x} \sim p_g} [ \| D(G(\hat{x})) - \alpha_R \|^2 ]$$

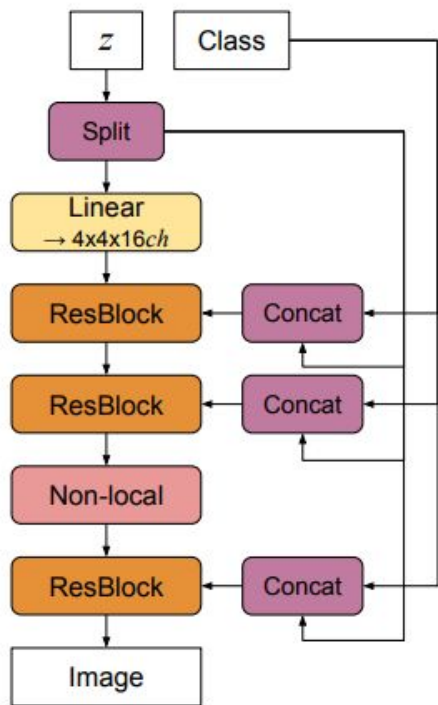
# Progressive Training



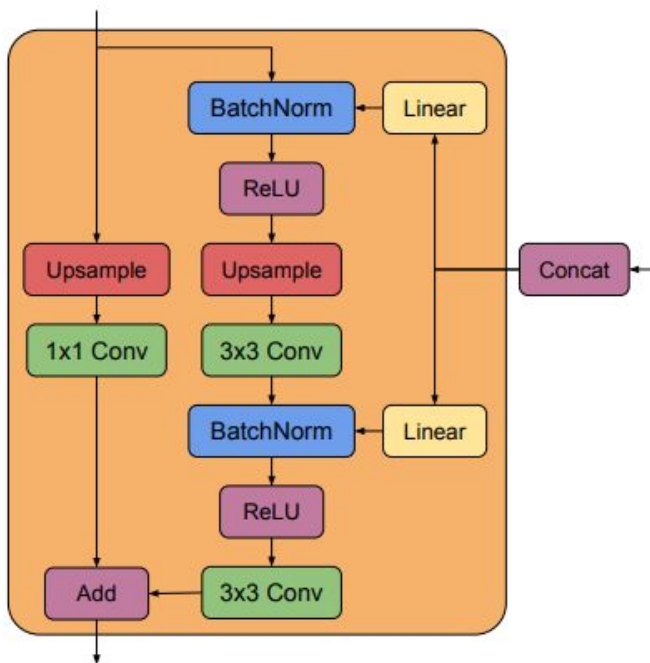
# BigGAN

- генерация изображений из разных доменов
- Self-attention блоки
- conditional batch norm
- большие батчи
- нет пирамиды

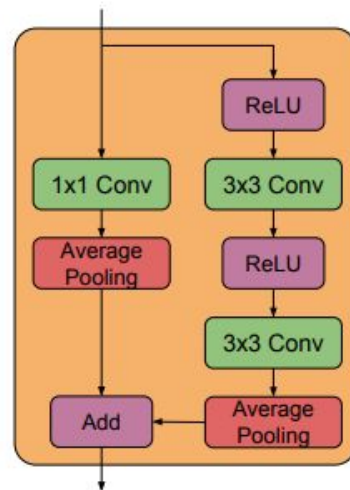
# BigGAN



(a)

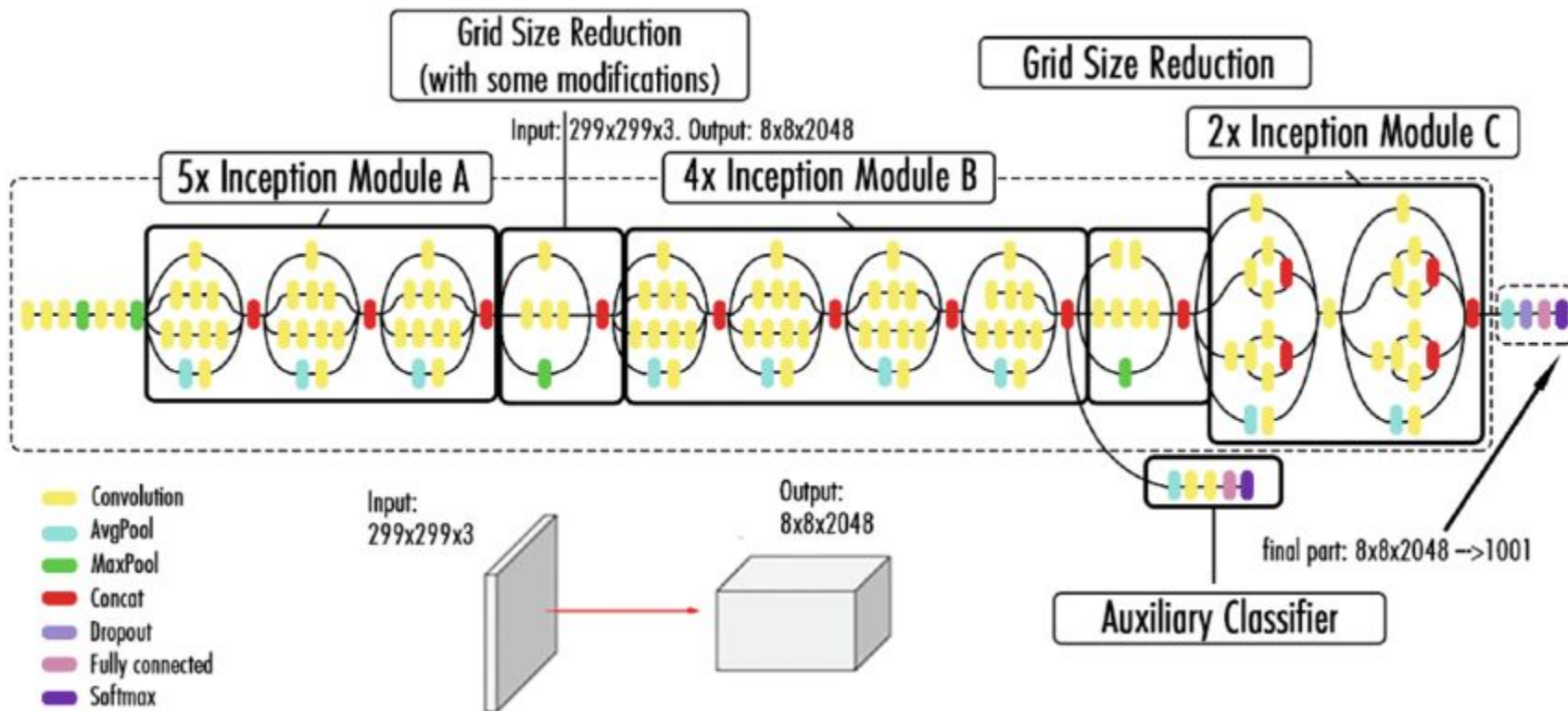


(b)



(c)

# InceptionV3





# Метрики

Inception score

$$IS(p_{gen}, p_{dis}) := \exp\left(\mathbb{E}_{x \sim p_{gen}} \left[ D_{KL} \left( p_{dis}(\cdot|x) \parallel \int p_{dis}(\cdot|x) p_{gen}(x) dx \right) \right]\right)$$

FID

$$d^2 = \|\mu_1 - \mu_2\|^2 + \text{Tr}(C_1 + C_2 - 2 \cdot \sqrt{C_1 C_2})$$

PSNR

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right)$$

# Почему FID не стабилен?

<https://arxiv.org/abs/2201.13019>

# Structural Similarity Index

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\sigma_x = \left( \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{1/2}$$

$$l(\mathbf{x}, \mathbf{y}) = \frac{2 \mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(\mathbf{x}, \mathbf{y}) = \frac{2 \sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}.$$

$$c_1 = K_1 L$$

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^\alpha \cdot [c(\mathbf{x}, \mathbf{y})]^\beta \cdot [s(\mathbf{x}, \mathbf{y})]^\gamma$$

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2 \mu_x \mu_y + C_1) (2 \sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1) (\sigma_x^2 + \sigma_y^2 + C_2)}$$

# Доп материалы

<https://www.youtube.com/watch?v=oEU-LNeA1Nc>

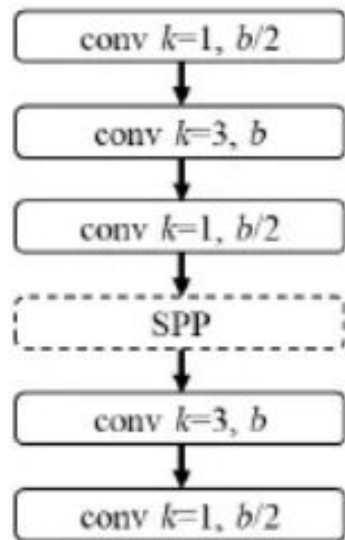
<https://github.com/Yangyangii/GAN-Tutorial>

<https://github.com/ChristophReich1996/Dirac-GAN>

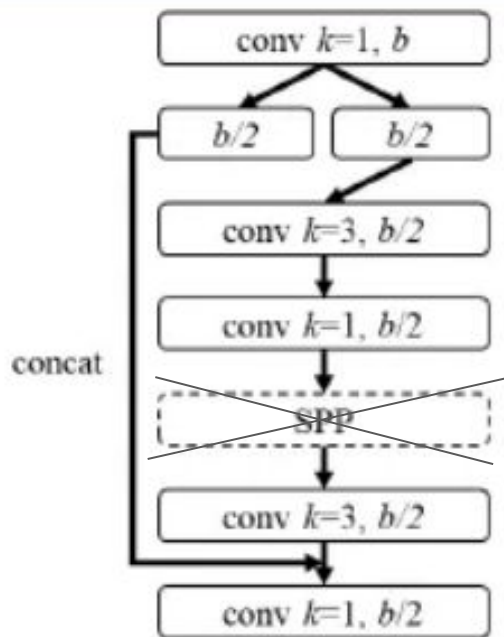
<https://arxiv.org/pdf/1606.03498v1.pdf>

[https://pytorch.org/hub/facebookresearch\\_pytorch-gan-zoo\\_pgan/](https://pytorch.org/hub/facebookresearch_pytorch-gan-zoo_pgan/)

# CSP block

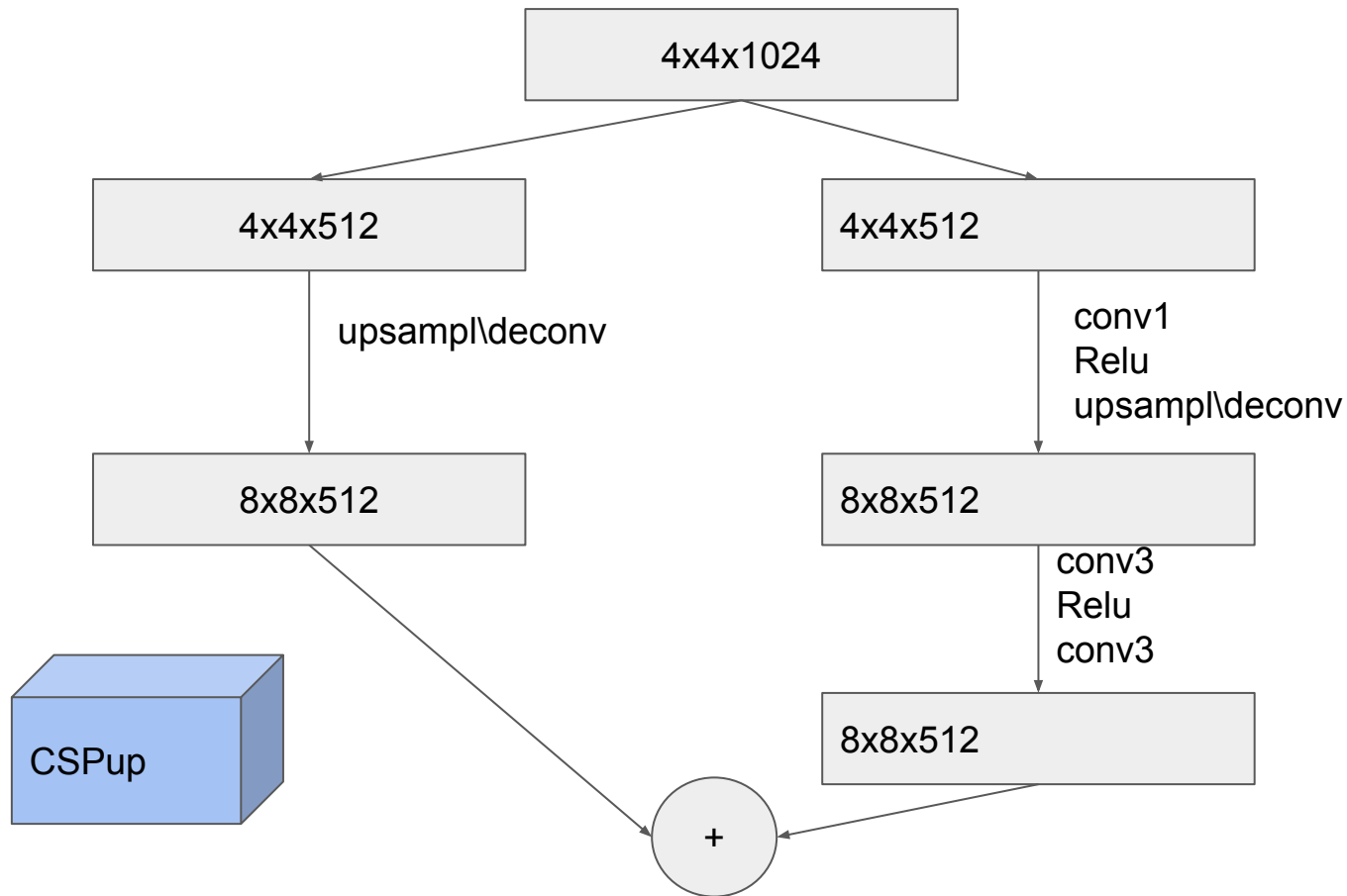


(a) reversed dark layers (SPP)

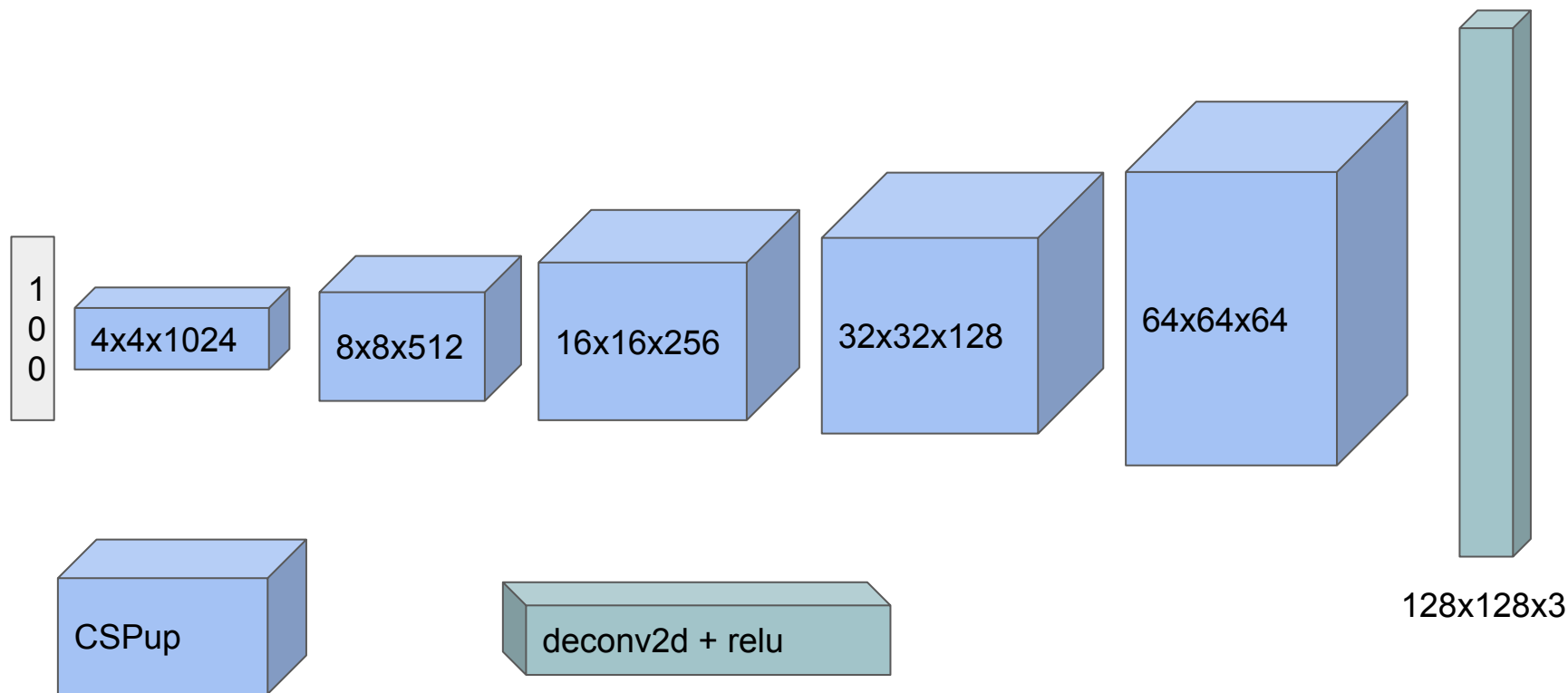


(b) reversed CSP dark layers (SPP)

# CSPup



# Генератор



# ДЗ

1. рисуем архитектуру:
  - на основе DCGAN
  - реснет блоки меняем на csp
2. запустить обучение на celeba
3. логировать лоссы и промежуточные картинки в clearml или tensorboard\weightandbias
4. добиться сходимости
5. попробовать разные варианты лоссов и регуляризаций
6. сравнить обучение при разных подходах
7. \* сравнить с сеткой на Resnet блоках классических





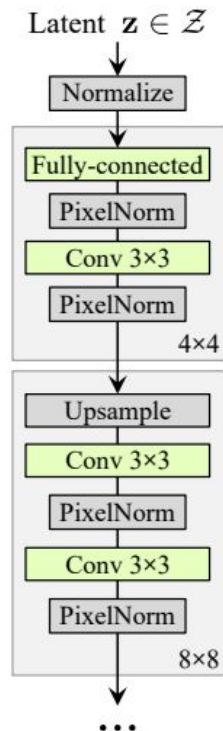


СТОП!СТАЙЛГАНЫ!

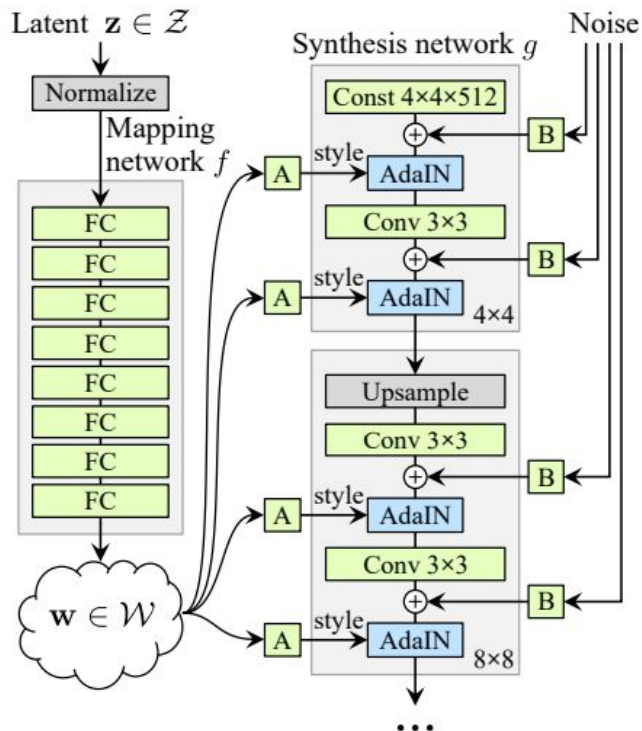
# StyleGAN

- на вход генератору латент + шум
- проекция 'запутанного' латента  $z$  в "распутанный"  $w$
- нестабильное обучение

# StyleGAN



(a) Traditional

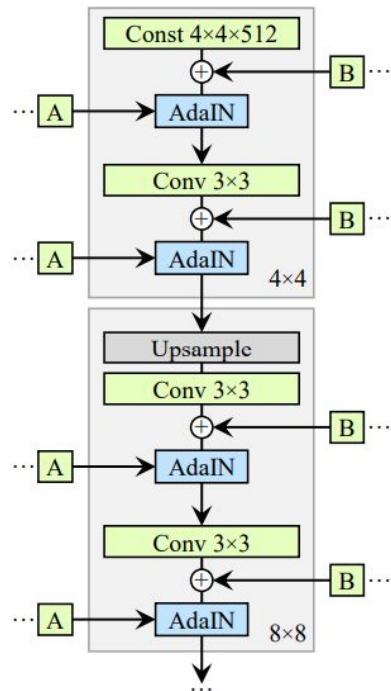


(b) Style-based generator

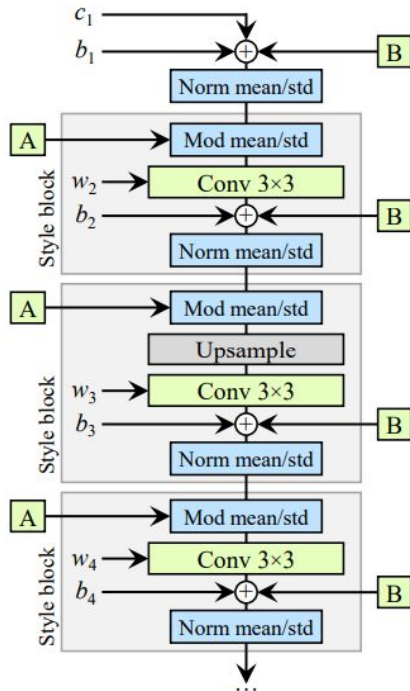
# StyleGAN 2

- строим изображение в несколько этапов постепенно увеличивая разрешение
- устранены артефакты из-за особенностей архитектуры G
- редизайн G и D (лучший D - ResNet)
- добавлены регуляризации

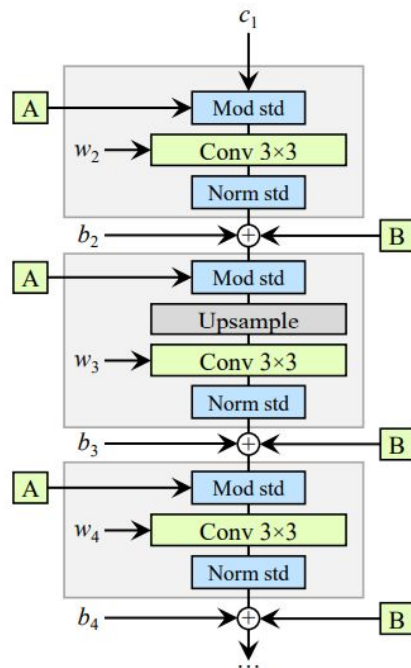
# StyleGAN 2



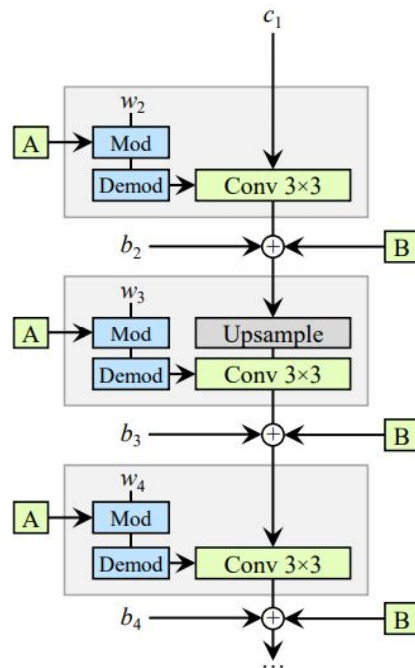
(a) StyleGAN



(b) StyleGAN (detailed)



(c) Revised architecture



(d) Weight demodulation

# StyleGAN 2 ADA

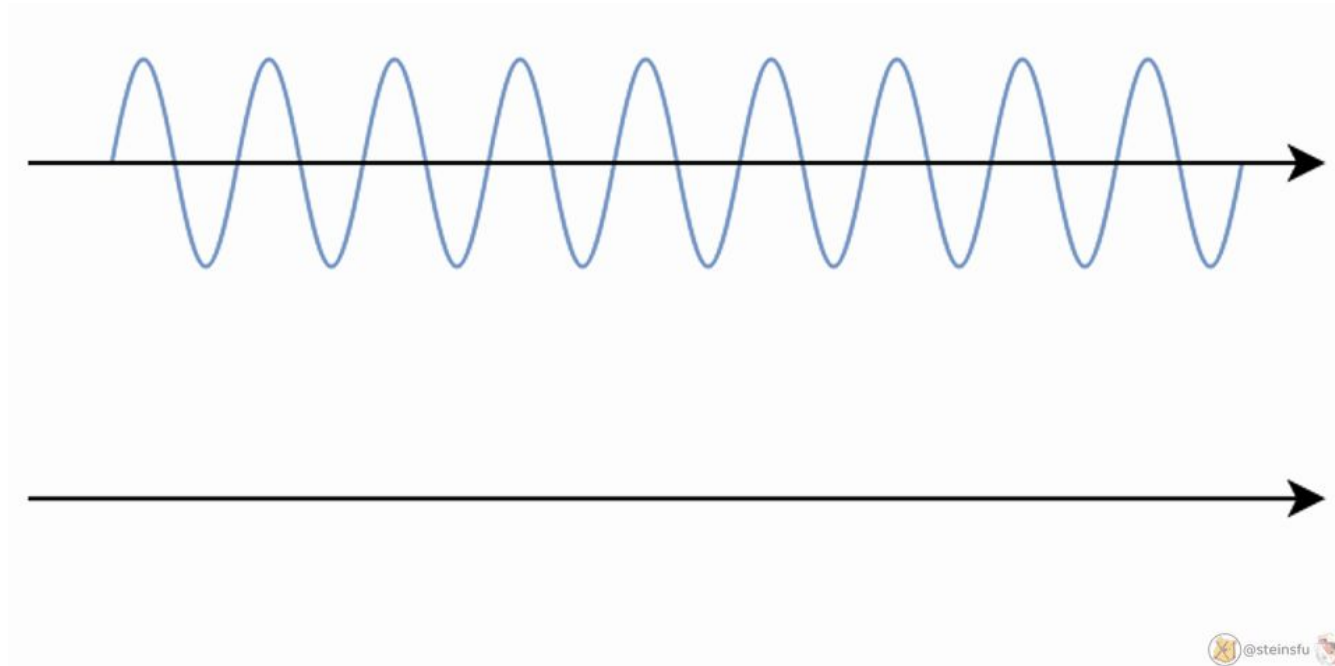
- устраняет mode collapse на ограниченном датасете с помощью дифф. аугментаций
- появился mix-precision training



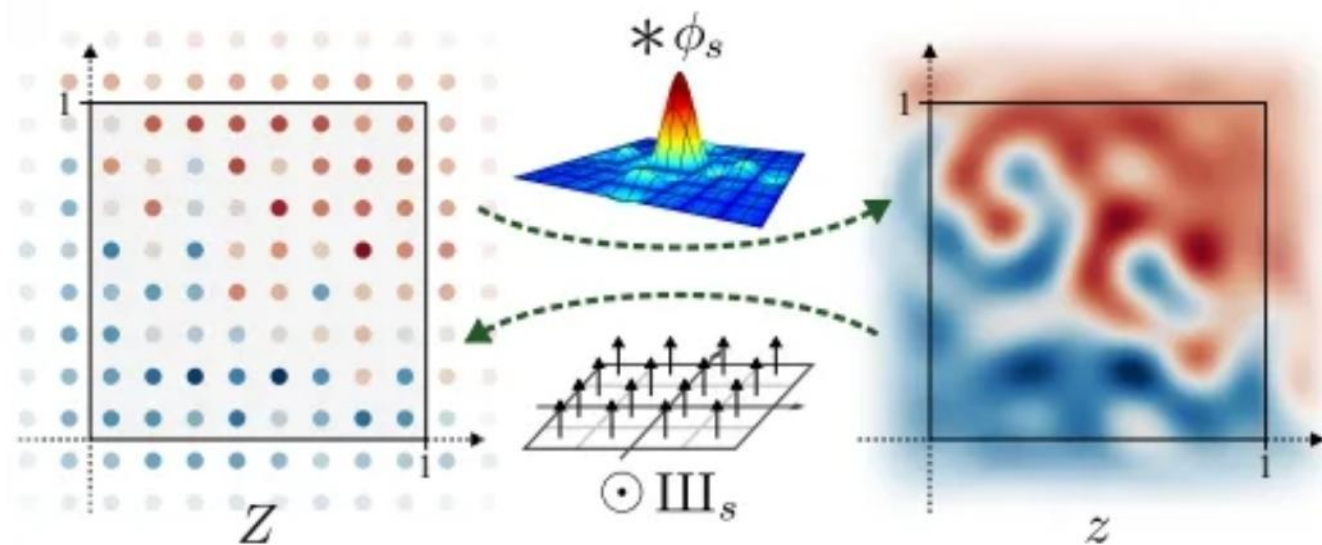
# StyleGAN 3

- уменьшили зависимости от локальных фич
- выбросили часть регуляризации, оставив сильную R1

# Aliasing



# Дискретное в непрерывное

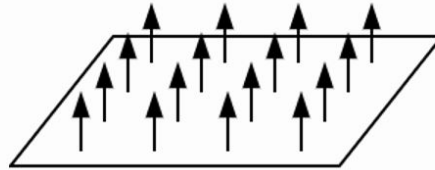
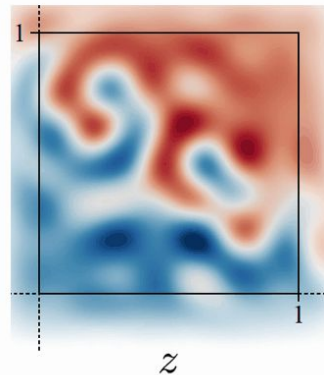


Conversion between continuous signal and discrete signal

# Непрерывное $\rightarrow$ дискретное

$$\text{III}_s(x) = \sum_{X \in \mathbb{Z}^2} \delta\left(x - \left(X + \frac{1}{2}\right)/s\right)$$

2D Dirac comb function



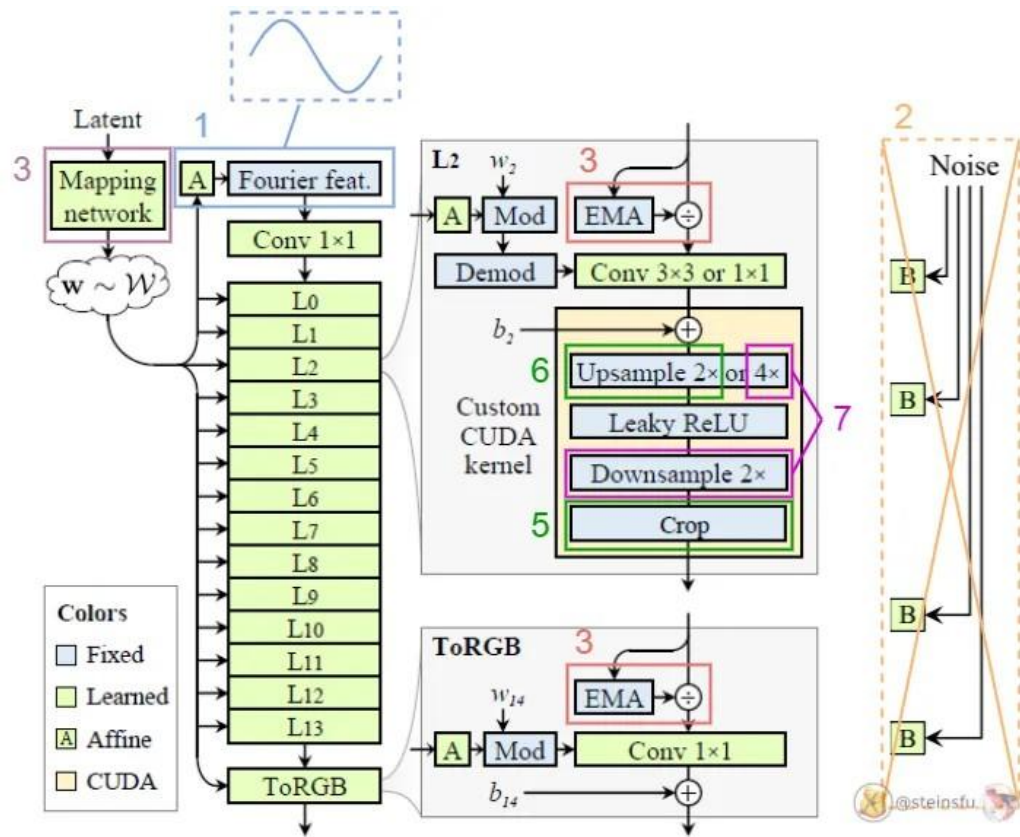
$\text{III}_s$

## Дискретное $\rightarrow$ непрерывное

*The Whittaker-Shannon interpolation formula states that the corresponding continuous representation  $z$  is obtained by convolving the discretely sampled Dirac grid  $Z$  with an ideal interpolation (low-pass) filter  $\Phi$ , where  $s$  is the sampling rate.*

*i.e.  $z(x) = (\Phi * Z)(x)$ , where  $*$  denotes the continuous convolution.*

# Изменения в архитектуре



# Translation Equivariance



# Training tips: dataset

- размер:  $10^5$  -  $10^6$  (можно меньше если в ограниченном домене)
- скрапинг и постпроцессинг (выравнивание, центрирование)
- лучше качество и больше разрешение (артефакты воспроизводятся, артефакты - первое куда начнет смотреть сеть)
- контрастность границ - важная фича

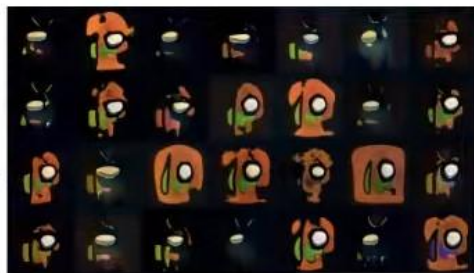


# Training tips: dataset

## Amogus



A



C



B

D



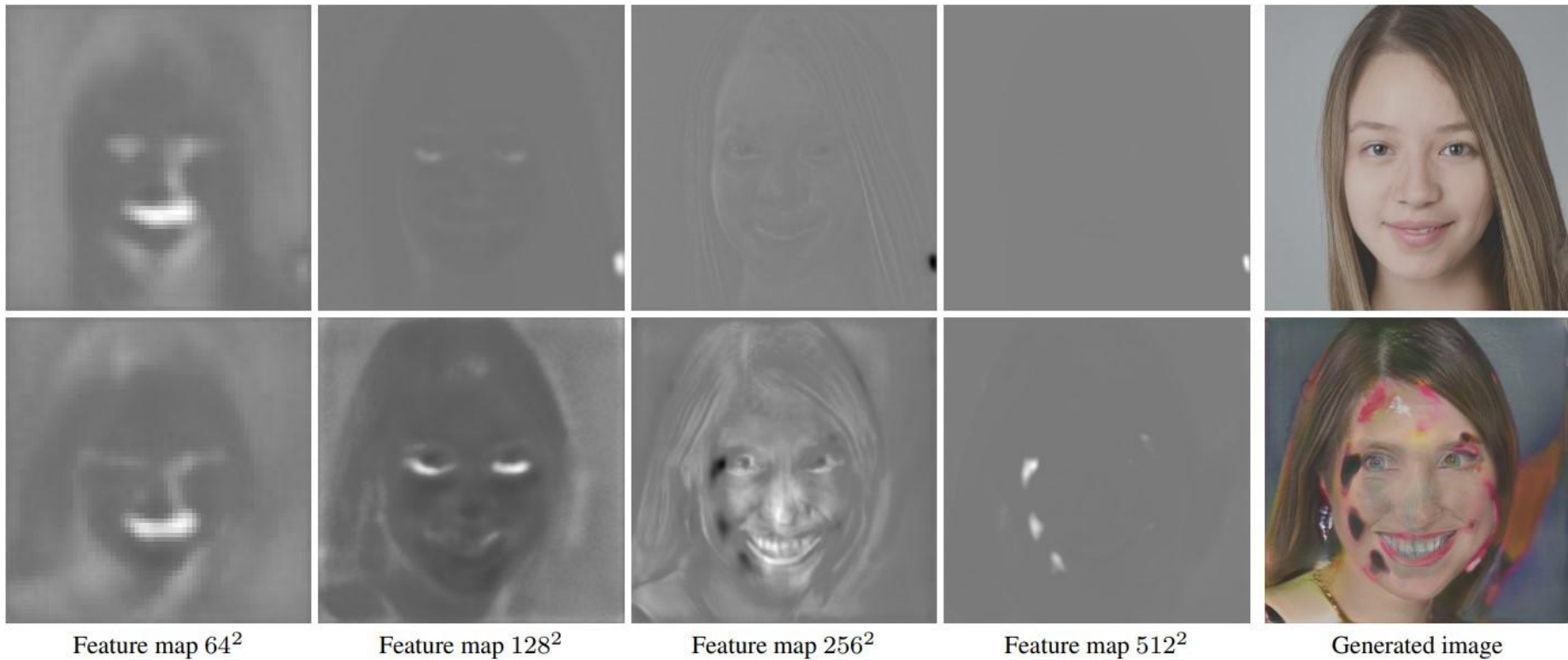
## Training tips:

- приемлемый объем вычислений (чем шире домен тем глубже сеть)
- выбор гиперпараметров (в StyleGAN ADA есть скрипт для хорошего подбора)
- регуляризации (style mixing, разные латенты, R1)
- R1 для гладкости латентного пространства
- $R1 >$  : больше фантазии сети
- $R1 <$  : меньше фантазии сети

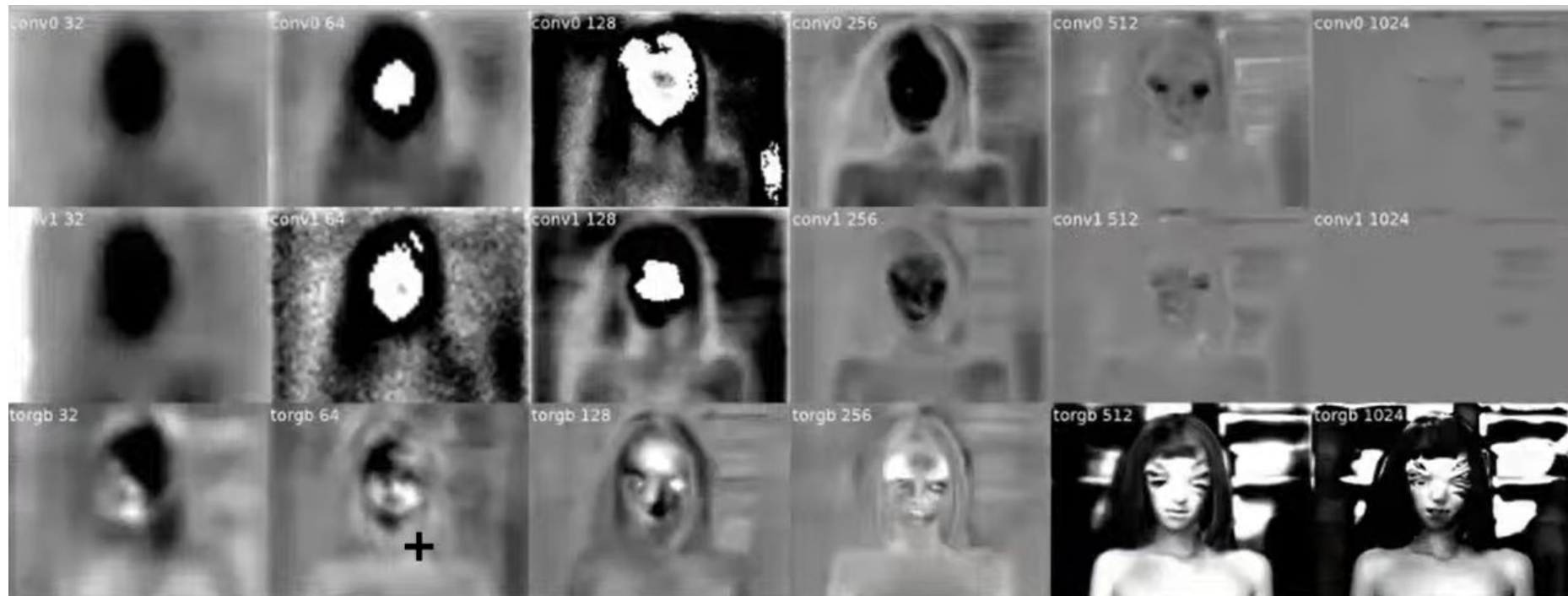
# Training tips: capacity

- если мало, упадет качество
- увеличение емкости слоев с малым разрешением закладывает основу генерации
- более верхние блоки отвечают за детали
- визуализируй активации слоев G

# Training tips



# Training tips



# Training tips: transfer learning

- $z \rightarrow \text{mapper} \rightarrow w$
- $\text{lr\_mapper} = \text{lr} * 0.01$  так что mapper практически не учится при TL (и не должен)
- заморозка нижних слоев если хотим сохранить домен (например были лица, теперь берем лица косплееров)
- лучше учить и G и D, но иногда D может выучиться быстрее и будет коллапс
-

# Sampling

- $Z$  пространство - знаем о нем все
- $W$  не знаем ничего
- если кластеризовать  $W$ , то центры кластеров могут быть хорошими стартовыми точками
- пропустить  $W$  через Clir и кластеризовать уже  $W_c$
- двигаться по пространству используя Clir

