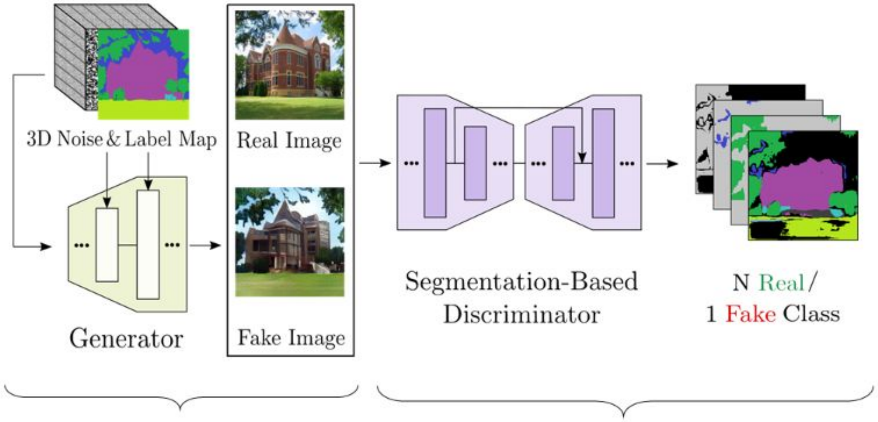
Generator Knows What Discriminator Should Learn in Unconditional GANs

Романов Игнат
Фридман Иван
Шешукова Марина

Conditional Generation

3 The OASIS model

In contrast to previous work, **OASIS** does *not* require a perceptual loss, it needs **O**nly **A**dversarial **S**upervision for **S**emantic **I**mage **S**ynthesis:



Generator (G)

- Noise is spatially replicated in 3D and concatenated with label map
- This tensor modulates all layers via a spatially-adaptive norm[4]

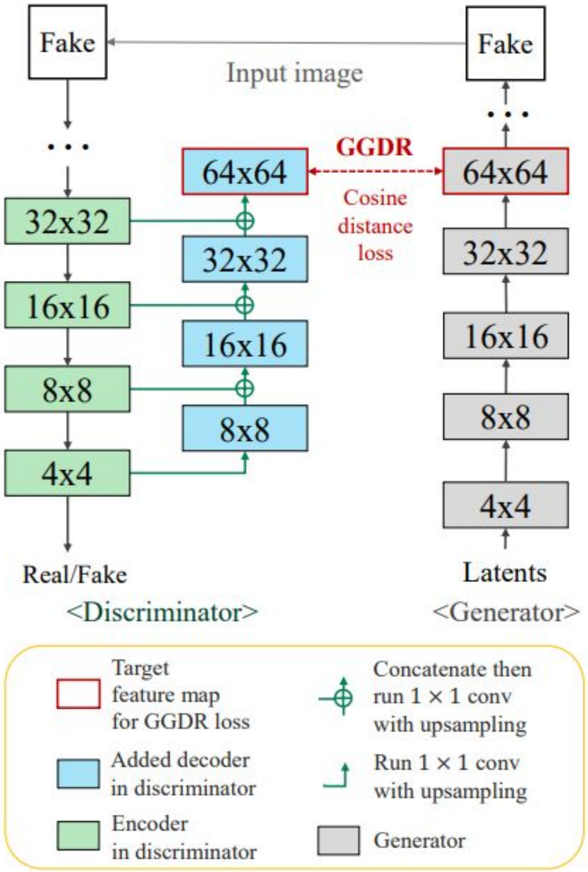
Discriminator (D)

- U-Net architecture, N real + 1 fake channel
- GAN loss = cross-entropy with N+1 classes
- Losses are weighted by inverse pixel-wise class frequency to achieve class-balance

Generator feature maps

(a) Visualization of the feature map of the decoder in our D

Модель

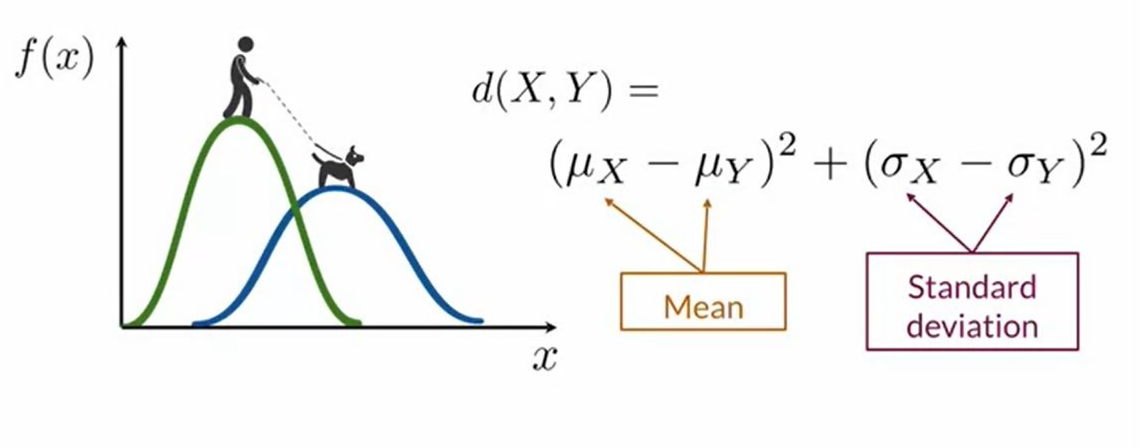

Fake
Input image
Fake
GGDR
Cosine
distance
loss
64x64
32x32
16x16
8x8
4x4
64x64
32x32
16x16
8x8
64x64
32x32
16x16
8x8
4x4
Real/Fake
Latents
<Discriminator>
<Generator>
Target
feature map
for GGDR loss
Concatenate then
run 1 × 1 conv
with upsampling
Added decoder
in discriminator
Run 1 × 1 conv
with upsampling
Encoder
in discriminator
Generator

Прикол

```python
def upfirdn2d(x, f, up=1, down=1, padding=0, flip_filter=False, gain=1, impl='cuda'):
    r"""Pad, upsample, filter, and downsample a batch of 2D images.

    Performs the following sequence of operations for each channel:

    1. Upsample the image by inserting N-1 zeros after each pixel (`up`).

    2. Pad the image with the specified number of zeros on each side (`padding`).
       Negative padding corresponds to cropping the image.

    3. Convolve the image with the specified 2D FIR filter (`f`), shrinking it
       so that the footprint of all output pixels lies within the input image.

    4. Downsample the image by keeping every Nth pixel (`down`).

    This sequence of operations bears close resemblance to scipy.signal.upfirdn().
    The fused op is considerably more efficient than performing the same calculation
    using standard PyTorch ops. It supports gradients of arbitrary order.

    Args:
        x:           Float32/float64/float16 input tensor of the shape
                     `[batch_size, num_channels, in_height, in_width]`.
        f:           Float32 FIR filter of the shape
                     `[filter_height, filter_width]` (non-separable),
                     `[filter_taps]` (separable), or
                     `None` (identity).
```

Fid

$$d_F(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma'))^2 = \|\mu-\mu'\|_2^2 + \operatorname{tr}\left(\Sigma+\Sigma'-2\left(\Sigma^{\frac{1}{2}}\cdot\Sigma'\cdot\Sigma^{\frac{1}{2}}\right)^{\frac{1}{2}}\right)$$



Результаты

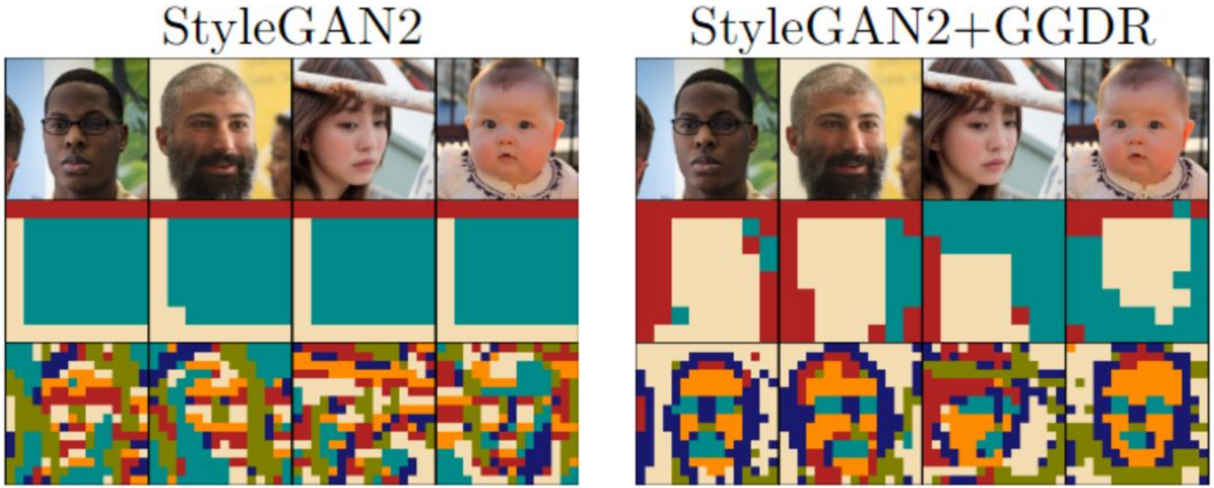
Table 1. FID scores of ours and comparison methods on FFHQ. We run three training for each data and show their means and standard deviations. The numbers are largely brought from ADA [26] and we follow their evaluation protocol.

FFHQ	2k	10k	140k
PA-GAN	56.49±7.28	27.71±2.77	3.78±0.06
WGAN-GP	79.19±6.30	35.68±1.27	6.43±0.37
zCR	71.61±9.64	23.02±2.09	3.45±0.19
AR	66.64±3.64	25.37±1.45	4.16±0.05
StyleGAN2	78.80±2.31	30.73±0.48	3.66±0.10
+GGDR	70.59±5.16	24.44±0.63	**3.14±0.03**
ADA	**16.49±0.65**	8.29±0.31	3.88±0.13
+GGDR	18.28±0.77	**6.11±0.15**	3.57±0.10

Table 2. FID scores of ours and comparison methods on CIFAR-10. We run three training for mean and standard deviations. We brought the numbers of diffusion models from [54].

CIFAR-10	FID	IS
ProGAN	15.52	8.56±0.06
AutoGAN	12.42	8.55±0.10
StyleGAN2	8.32±0.09	9.21±0.09
ADA	2.92±0.05	9.83±0.04
FSMR	2.90	9.68
DDPM	3.17±0.05	9.46±0.11
NCSN++	2.2	9.89
ADA+GGDR	**2.15±0.02**	**10.02±0.06**

Результаты



(b) K-means clustering of the feature maps of the encoder in the discriminators.

Результаты


FFHQ
LSUN Cat
LSUN Horse
LSUN Church
AFHQ Cat
AFHQ Dog
AFHQ Wild
Lands-scape