GANs evolution

Aziz Temirkhanov

Laboratory for methods of big data analysis

LAMBDA • HSE

Fall 2023

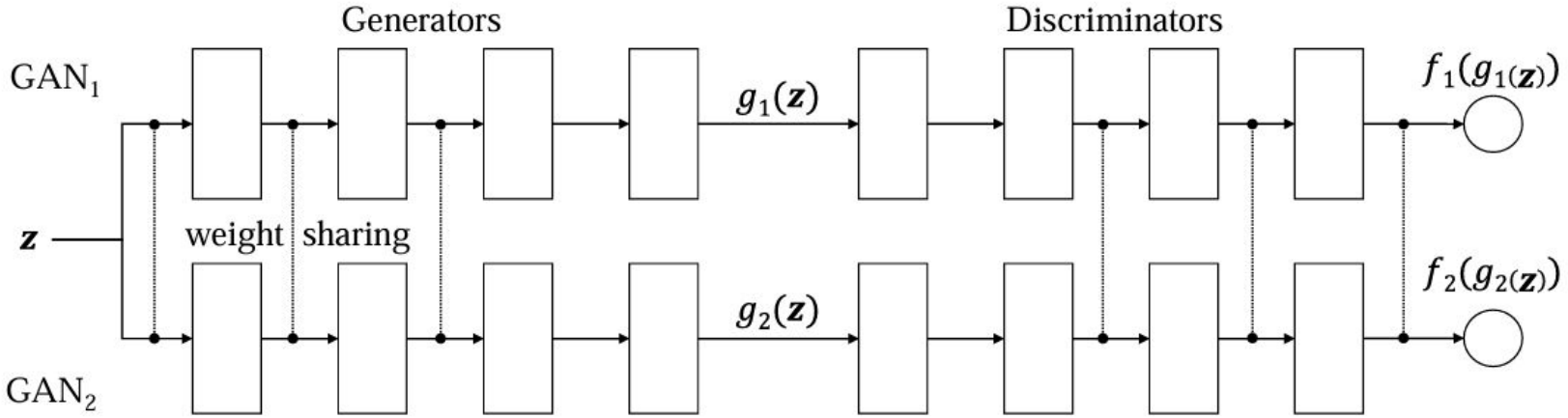
Evolution

- **Vanila GAN** https://arxiv.org/abs/1406.2661
- **DCGAN** https://arxiv.org/abs/1511.06434
- **CoGAN** https://arxiv.org/abs/1606.07536
- **ProGAN** https://arxiv.org/abs/1710.10196
- **StyleGAN** https://arxiv.org/abs/1812.04948

Deep Convolutional GAN (DCGAN)

- Introduced CNN into GANs framework
- Batch normalization
- ReLU and tanh for Generator and LeakyReLU for Discriminator

Coupled GAN



- Two generative models G1 and G2, with weight sharing for first layers
- Two discriminative models F1 and F2, with weight sharing for last layers

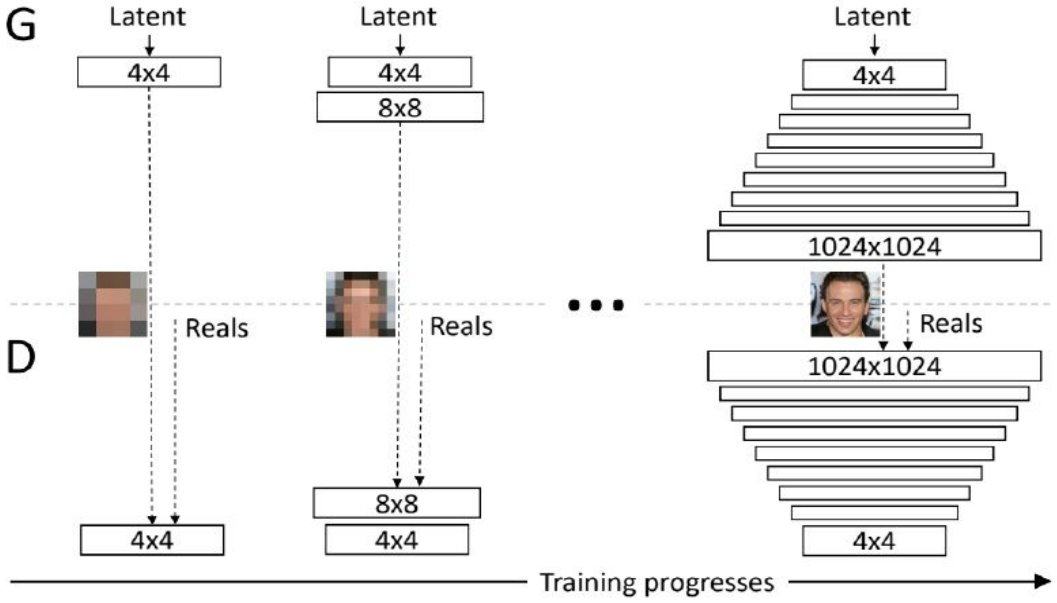
Progressive Growing of GANs (ProGAN)

Samples (1024x1024)

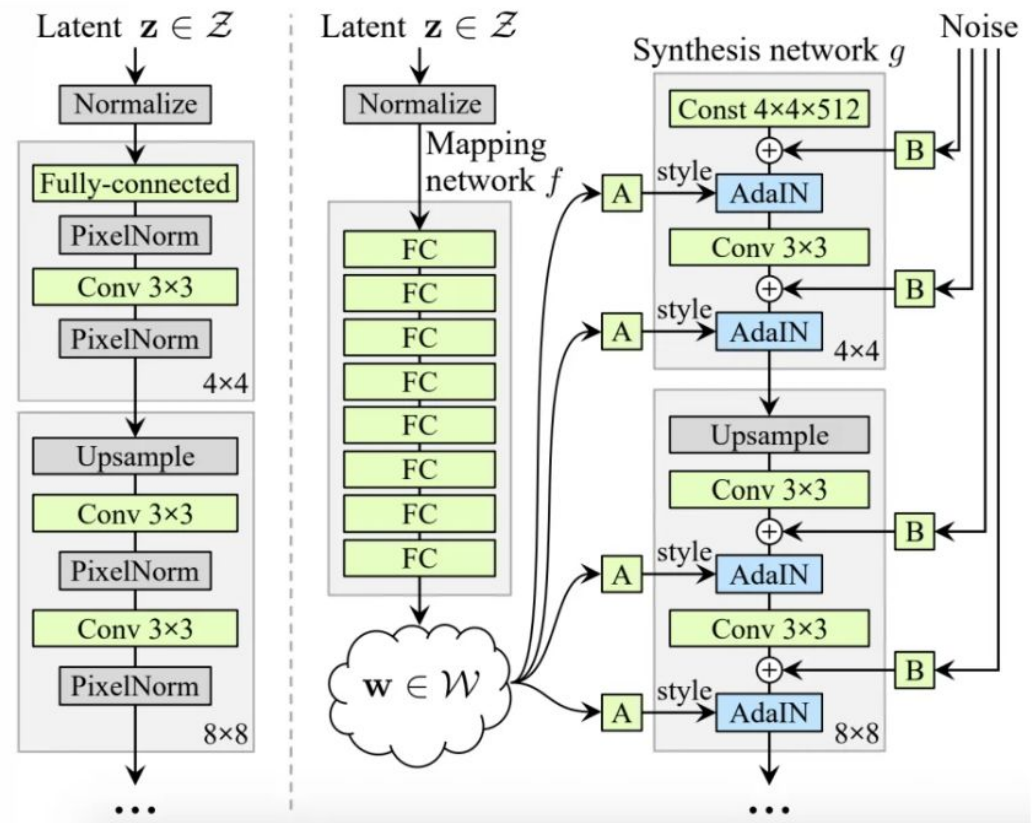
ProGAN

Grow both the generator and discriminator progressively, new layers will introduce higher-resolution details as the training progresses.

- Train GAN which generate 4x4 images (2 convs for G and D).
- Add upsampling layers to G, downsampling layers to D.
- Train GAN which generate 8x8 images.
- etc.



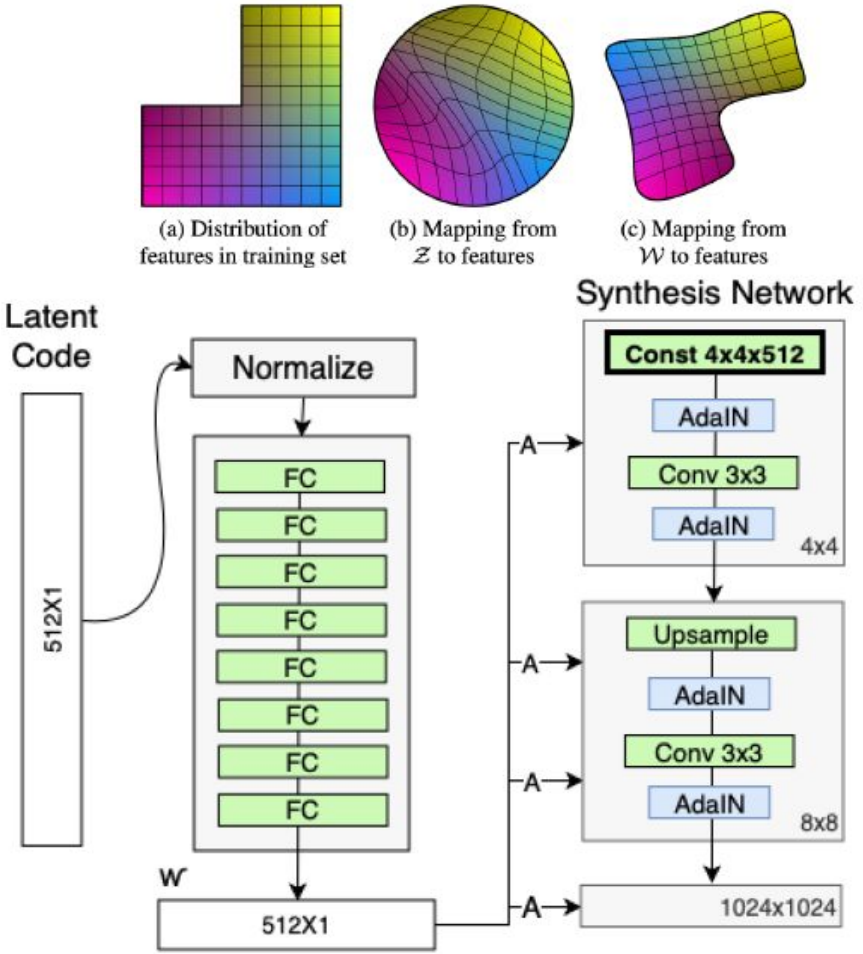
Style GAN



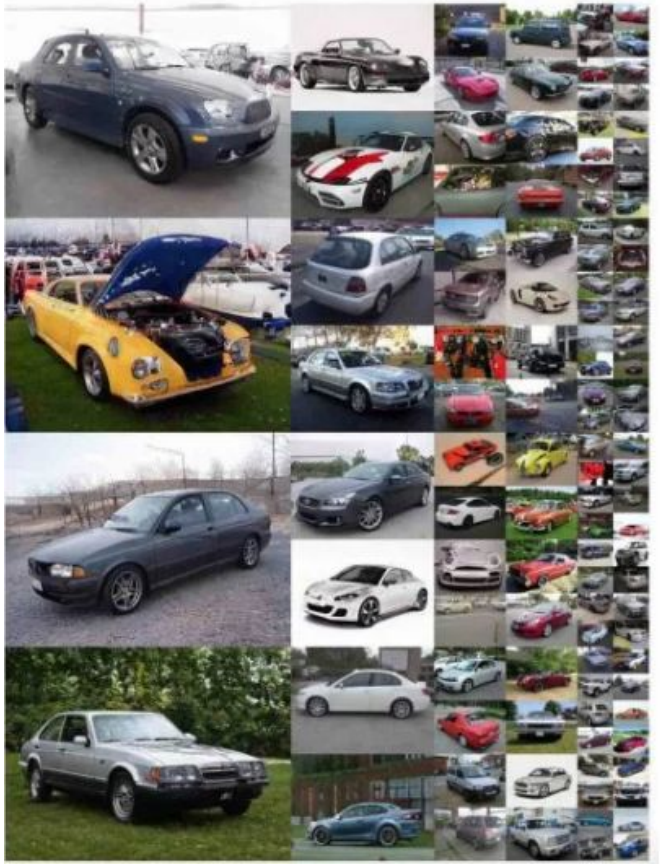
- Using the idea of **ProGAN** as base
- Improving Generator, keeping the same Discriminator
- Controlling specific features

StyleGAN

- Introduce a FC NN for map entangled features from input vector **z** to latent space ***W***
- Thus, reduce correlations between components of **z**
- Input vector **z** is actually is a learnable constant
- At each stage (recall the ProGAN), apply apping NN to a vector, at feed it to generative model. Thus, yields a different input at a different stage.
- Inject random noise to an input at each stage to add small details and increase variety
- Adaptive Istance Normalization of each input



StyleGAN



StyleGAN2

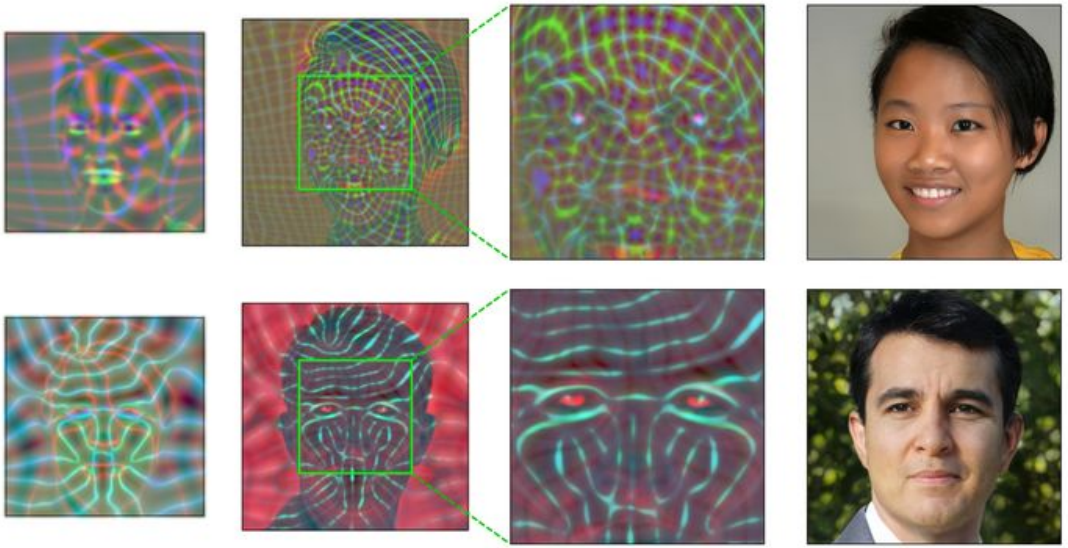
- Adaptive instance normalization allow to blend in styles at each stage effectively. Scale the input x and then force it to align channel-wise with statistics of style y

$$\text{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y)$$

- But in previous work, there were little artifacts (usually at hair, teeth area). And AdaIN was the troublemaker. After removing it, artefacts gone completely
- This, along with few small adjustments to generator, and a few for discriminator as well, yielded a better result.

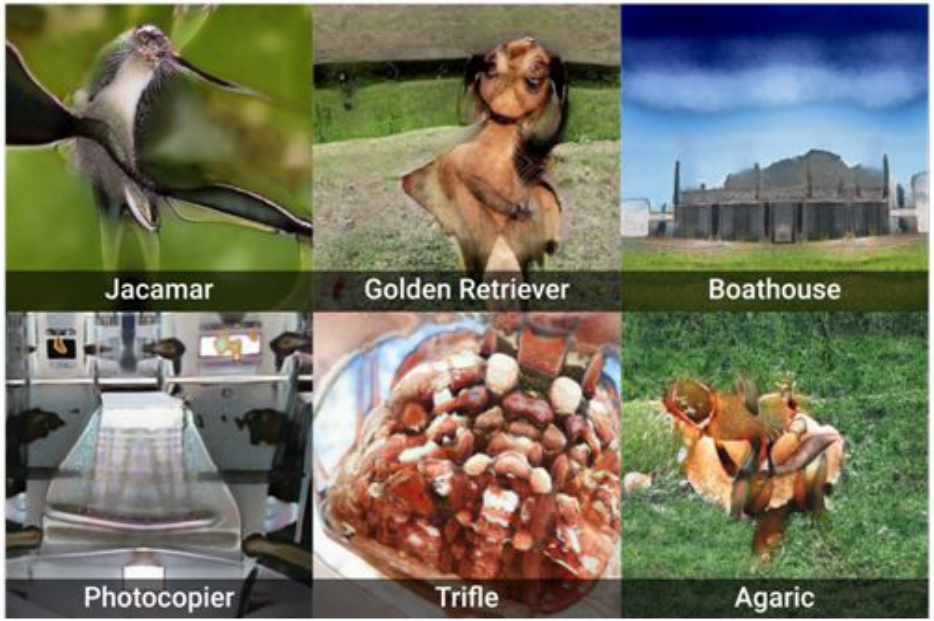
StyleGAN3

Use of fourier features, filtering, 1x1 convolution kernels and other modifications make the generator equivariant to translation and rotation



- Equivariance studies how transformations of the input image are encoded by the representation, invariance being a special case where a transformation has no effect.
- Standard convolutional architectures consist of stacked layers of operations that progressively downscale the image. Aliasing is a well-known side-effect of downsampling that may take place: it causes high-frequency components of the original signal to become indistinguishable from its low-frequency components.

StyleGAN-XL



StyleGAN3

StyleGAN-XL

StyleGAN-XL

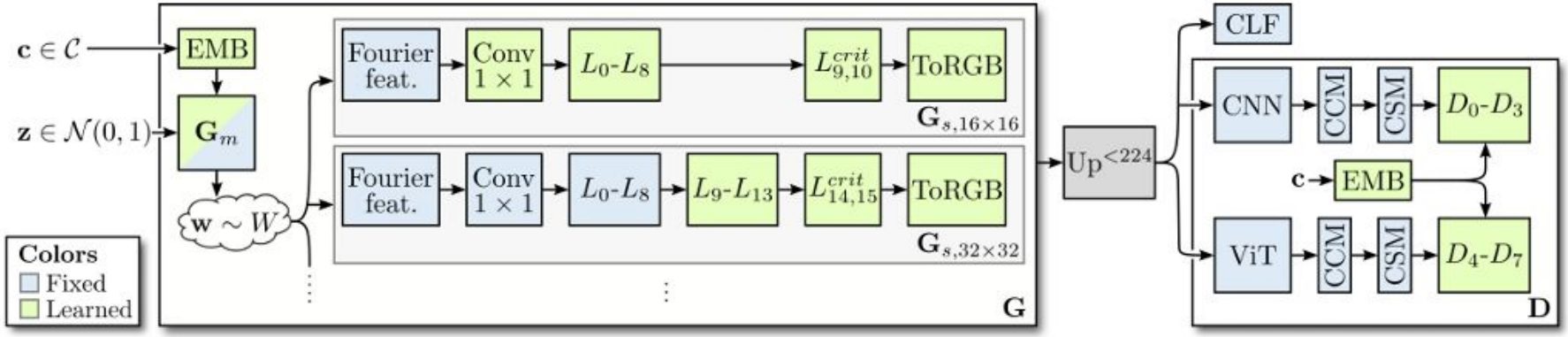
- Uses **StyleGAN3-T** and **StyleGAN3-R**, and **ProjectedGAN**:
- The original adversarial game between a generator **G** and a discriminator **D** can be extended by a set of feature projectors $\{P_l\}$. The projectors map real images x and images generated by G to the discriminator's input space.The ProjectedGAN objective is formulated as:

$$\min_{G} \max_{\{D_l\}} \sum_{l \in \mathcal{L}} \left(\mathbb{E}_{\mathbf{x}}[\log D_l(P_l(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}}[\log(1 - D_l(P_l(G(\mathbf{z}))))] \right)$$

StyleGAN-XL

- Using only StyleGAN3-T, as StyleGAN3-R yield overly symmetrical images
- Spectral normalization without gradient penalties
- Blur first 200k images before feeding them to discriminator
- Reducing StyleGAN latent code **z** to 64
- Conditioning of labels embedding (instead of raw labels), on modified feature network
- Using generalized embedding for a class from EfficientNet and ViT before applying to discriminator
- Using DeiT-small as a classifier, add cross-entropy to generator loss

StyleGAN-XL



- Labels *c* and code *z* forwarded to a feature mapping net (recall vanilla StyleGAN)
- Train a several generators G (as in ProGAN and in StyleGAN)
- Upsample, if necessary (should be larger then 224 x 224)
- Forward output to ViT and CCN model before applying discriminator stage

https://arxiv.org/pdf/2202.00273v2.pdf