Evaluation

Q: Why evaluation of GANs is hard?

A: There is no universal discriminator that can be used to compare two GANs.

Desired Properties of GANs

* Fidelity aka quality of image
* Diversity aka variety of images

Features-based Distance between images

1. Use pretrained models to extract features of (generated) images
2.

$$\text{dist}\left(\phi(X_{fake}), \phi(X_{real})\right),$$

where $X_{fake} = G_\theta(z)$, $z \sim P_{noise}$

Frechet Inception Distance (FID)

Let $X_1, X_2, \ldots, X_n \sim_{iid} \mathcal{N}(\mu_X, \Sigma_X)$ — features of n fake images extracted by Inception model.

We assume that $X_i \in \mathbb{R}^d$ comes from multivariate Normal distribution.

Same for real images $Y_1, Y_2, \dots, Y_m \sim_{iid} \mathcal{N}(\mu_y, \Sigma_y)$

To compute FID, given $\{X_i\}$ and $\{Y_i\}$ find μ_x, μ_y, Σ_x and Σ_y

$$FID := \|\mu_x - \mu_y\|_2^2 + Tr\left(\Sigma_x + \Sigma_y - 2\sqrt{\Sigma_x \Sigma_y}\right)$$

Remark

* lower FID is better (ideal = 0)
* $n, m \sim 50 \cdot 10^3$ – need large sample size
* FID is very popular

Inception Score (widely used previously)

Given x – image consider output of Inception classifier – $p(y|x)$

We have high fidelity if $p(y|x)$ has high peaks (low entropy)

Also we want $p(y)$ to have high entropy, i.e. G generates different classes – high diversity

$$p(y) = \int p(y,x)\,dx = \int p(y|x)\,p(x)\,dx$$

if assume $x \sim \mathcal{U}$

$$P(y) = \frac{1}{m} \sum p(y|x)$$

$$D_{KL}(p(y|x) \| p(y)) \longrightarrow \max$$

$$IS = \exp\left(\mathbb{E}_{x \sim p_G} D_{KL}(p(y|x) \| p(y))\right) \longrightarrow \max$$

Remark

* $P(y)$ remains const, only $P(y|x)$ changes in $\mathbb{E}_{x \sim p_G}$
* If IS is low then eigther
  - low diversity, or
  - low fidelity
* IS looks only at fakes

Remark

Since we use $x_1, x_2, \dots, x_n \sim P_g$ sample of fakes, and each $x_i = G(z_i)$, where $z_i \sim \mathcal{N}(0, \Sigma)$, we might use Truncation trick to control diversity / fidelity of $\{x_i\}$

Precision and Recall for GANs

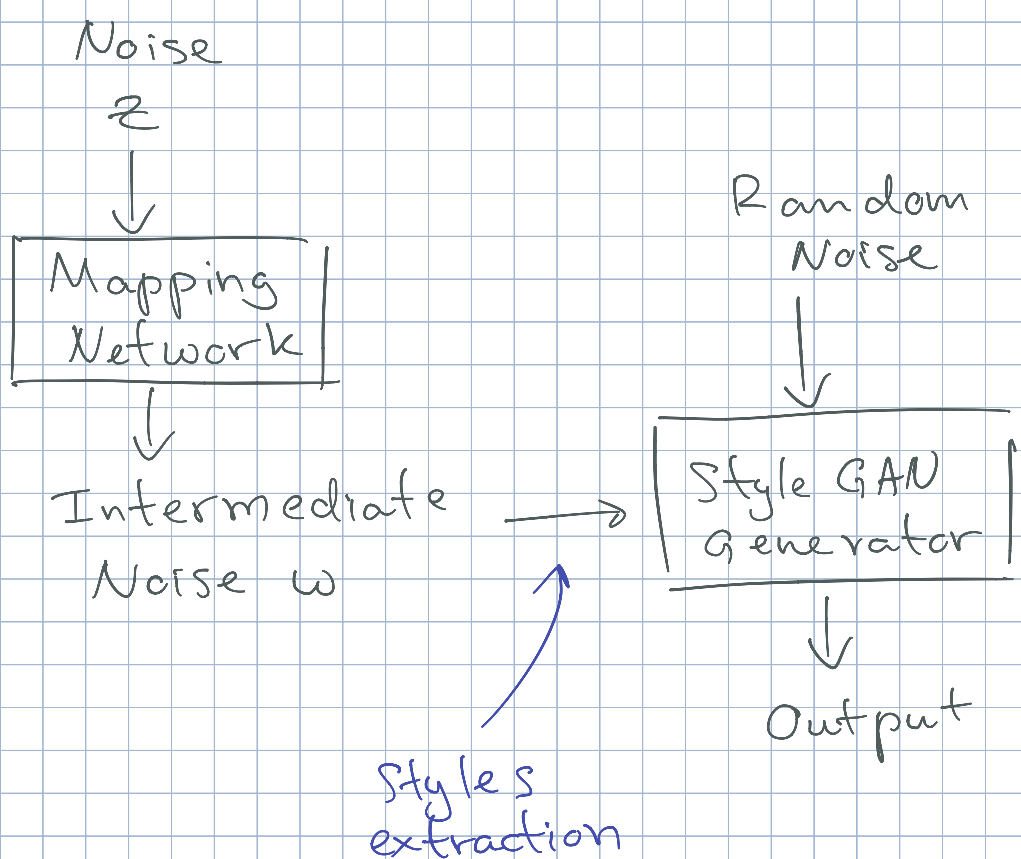
(See Notes on paper in the repo)

Disadvantages of GANs

* Hard evaluation
* Unstable training
* No density estimation

StyleGAN Overview

ProGAN - predecessor of StyleGAN



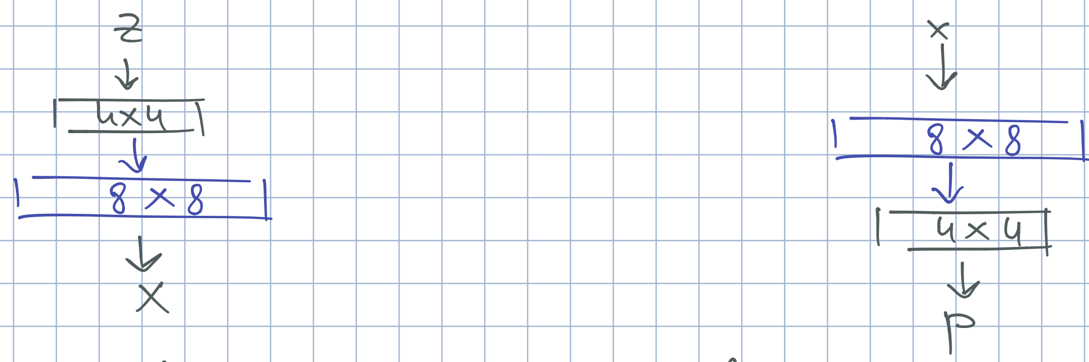
Progressive Growing (ProGAN)

Increasing resolution of images (fakes and reals) during training
(Helps for learning to generate high res images in a faster and more stable way)

Details:

Start with images 4x4

After some time - double the size



This procedure is continued untill we reach desired resolution

Remark

Note that we do not switch to new conv layers right away. Insted we use simple upsamling/downsampling and conv layer and build interpolation

$conv * \alpha + simple * (1-\alpha)$

And we increase α gradualy.

Noise Mapping Network

$z \sim \mathcal{N}$

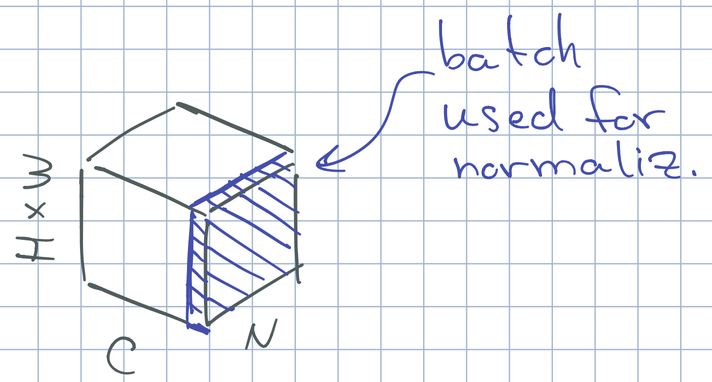
$w = MLP(z)$ — intermediate noise vector

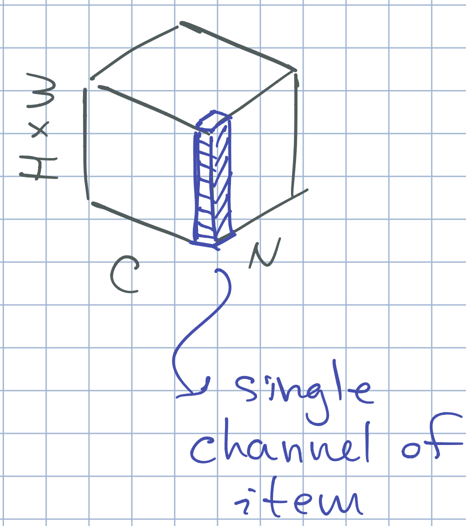
Intuition: MLP helps to build more disentangled noise represent.

Unlike in regular GANs, w is not used as simple input for G_θ.

Adaptive Instance Normalization (AdaIN)

Instance Normalization vs Batch Norm



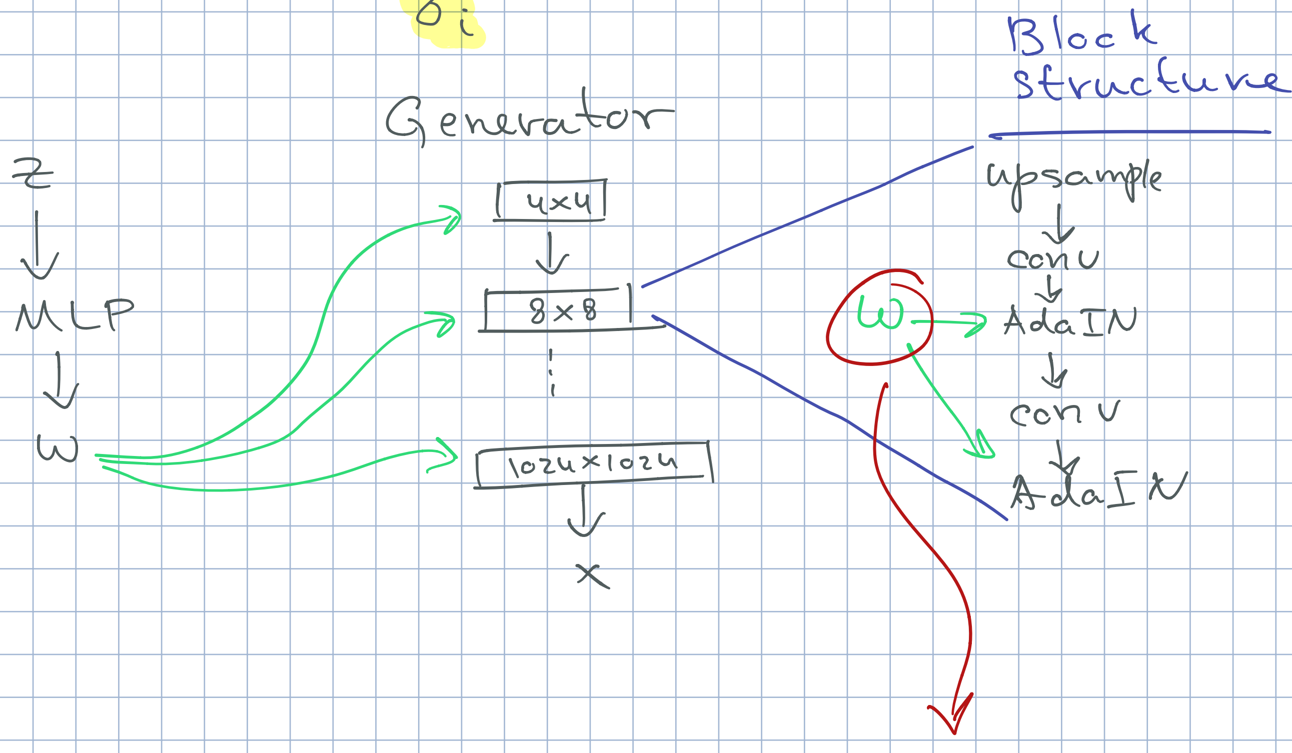


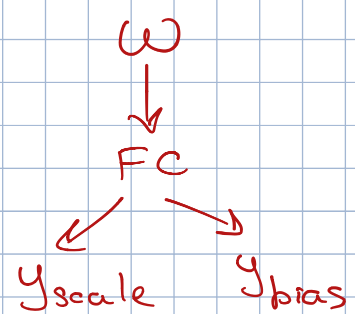
N – batch size

C – # of channels

HxW – img size (flattened)

$$\hat{x}_i \leftarrow \frac{x_i - \mu_i}{\sigma_i}$$





$$\hat{x}_i \leftarrow y_{scale} \cdot \frac{x_i - \mu_i}{\sigma_i} + y_{bias}$$

This is what controls style of generated images

Remark

* Earlier blocks of generator controls coarse style features, and later blocks responsible for more fine-grained style features.

* We can adjust w noise vector and fed it to later blocks in order to

slightly change some minor details in generated images.

Styles Mixing (Increase diversity)

Instead of using same ω as input for all blocks in G we can mix styles of two images.

$$z_1 \sim \mathcal{N}, \quad \omega_1 = MLP(z_1) \quad \text{and} \quad z_2, \omega_2$$

! We can use ω_1 and ω_2 as inputs for different blocks.

For example, if we will take ω_1 as input for first layers and ω_2 for remaining, we will take coarse style from first image and fine-grained style from second image.

Style Variation

Add some perturbations in generated images to futher increase diversity.

upsample
↓
conv
↓
AdaIN
↓
conv
↓
AdaIN

add noize vector (with λ mult.) to conv layer output