Paper

- **Title:** Progressive Growing of GANs for Improved Quality, Stability, and Variation
- **Authors:** Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen
- **arXiv link:** https://arxiv.org/pdf/1710.10196.pdf

TL;DR

The paper shows a new method to train GANs. The main idea is to start with shallow Generator and Discriminator and then add layers to them slowly to increase details in the generated images. This also speeds up the training and stabilizes it. They also suggest a new evaluation method based on Wasserstein distance between the real and the generated distribution.

Progressive Growth method

- The paper uses Generator and Discriminator whose architectures are mirror images of each other.
- They start with Generator that generates a low resolution images and add layers to both Generator and Discriminator simultaneously.
- Such an increase makes the Generator learn the structure of the image first and then learn to add details to it to make it more realistic.
- While adding a new layer, it is faded in slowly
- During the transition, treat the layers that operate on the higher resolution like a residual block, whose weight α increases linearly from 0 to 1

Method to increase variation

In order to increase variation of generated images, the paper describes the following method:

- Compute standard deviation for each feature in each location over a minibatch.
- Average these estimates over all features and locations to get a scalar.
- Replicate the value and concatenate it to all spatial locations and over the minibatch, yielding one additional (constant) feature map.
- Results are better when this additional map is added towards the end of generation.

Normalization method

Equalized Learning Rate

They use a simple $\mathcal{N}(0, 1)$ initialization and then scale the weights explicitly at runtime by $\widehat{W}_i = \frac{w_i}{c}$ where c is the per-layer normalization constant from He's initializer (He et al. 2015).

Pixelwise normalization

To disallow the scenario where the magnitudes in the generator and discriminator spiral out of control, they normalize the feature vector in each pixel to unit length in the generator after each convolutional layer.

Multi Scale Similarity for Assessing GAN results

Intuition: a successful generator will produce samples whose local image structure is similar to the training set over all scales.
Method: Study this by considering the multi scale statistical similarity between distribution of local image patches drawn from a Laplacian pyramid. After normalizing the images with mean and standard deviation of each color channel, estimate similarity by using their sliced Wasserstein distance using Rabin et al. 2015