# EXPLORING HUMAN MOTION SYNTHESIS WITH LATENT-SPACE GANS

**Author**
Vinitra Muralikrishnan

September 11, 2024

## Contents

# 1 GANs

GANs (Generative Adversarial Networks) are a class of ML models that consist of 2 neural networks - a generator or a discriminator, that are trained simultaneously in a competitive setting.

## 1.1 Components of GAN

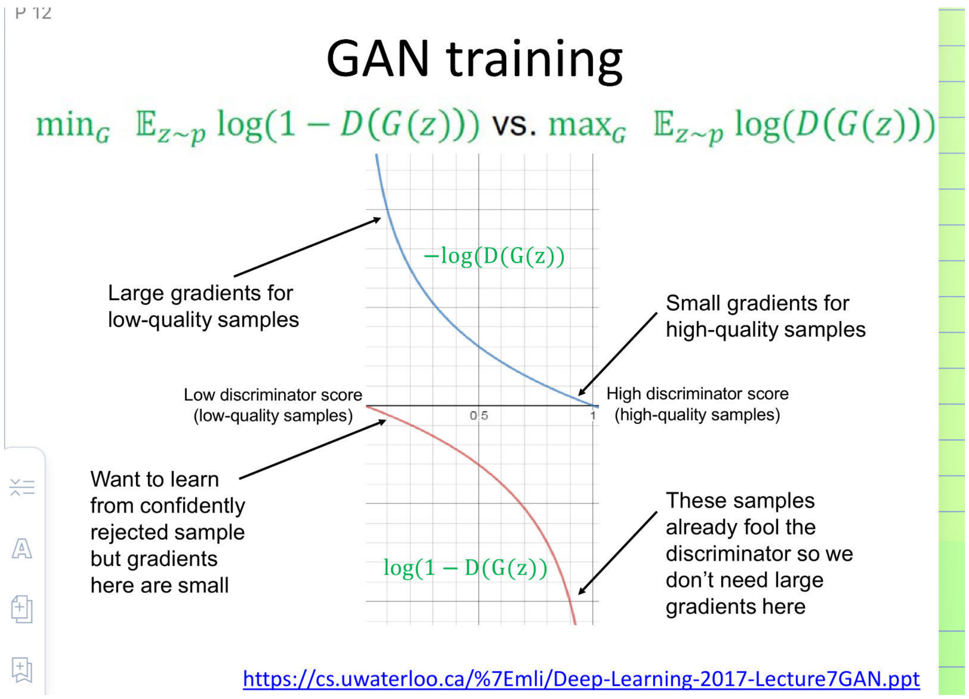
- Generator: Generator's job is to mimic real data. It takes a random noise vector and generates a sample that resembles the target data distribution. Over time, it learns to genrerate samples tahta re more realistic.
- Discriminator: It is a classifier that will classify between real data and fake data that is generated by the generator.
- Training: The generator tries to fool the discriminator by generating better data and discriminator tries to become better at distinguishing between real and fake. Both networks are trained in an adversarial manner. The loss function of the generator is designed to maximize the chance of fooling the discriminator and the discriminator minimizes the classification error betweeen real and fake data.
- Challenges: Mode collapse, training instability and vanishing gradients. Mode collapse happens when generator produces limited variations, failing to capture the full diversity of the real data (supposedly captured by mode).

## 1.2 GAN training

During the GAN training, the generator wants to fool the discriminator successfully and therefore the generator loss would involve minimizing the log likelihood of discriminator being right. The discriminator on the other hand, would work towards maximizing the log likelihood of discriminator being right.

$$V(G,D) = \mathop{\mathbb{E}}_{x\sim p_{data}} log\, D(x) + \mathop{\mathbb{E}}_{z\sim p} log(1 - D(G(z)))$$

In the above loss function, the generator loss would involve minimizing $\mathbb{E}_{z\sim p}\, log(1-D(G(z)))$. This is not a stable training process as the low quality samples wouldn't have large gradients with this equation. The high quality samples would have large gradients, leading to bigger corrections in the generator and throw off the training. Refer to the image below:



P 12

GAN training

$\min_G \; \mathbb{E}_{z\sim p} \log(1 - D(G(z)))$ vs. $\max_G \; \mathbb{E}_{z\sim p} \log(D(G(z)))$

https://cs.uwaterloo.ca/%7Emli/Deep-Learning-2017-Lecture7GAN.ppt

Therefore to correctly guide the generator, we instead choose to maximize the log likelihood of discriminator being wrong, i.e $\mathbb{E}_{z\sim p}\, log(D(G(z)))$

## 1.3 Variants

- DCGAN - uses convolutional layers instead of fully connected layers.
- WGAN - Uses wasserstein distance loss function instead of the original loss.
- StyleGAN: Allows more control over the generational process by separaing the high-level features and low-level features, resuling in impressive, fine-grained image generation.
- CycleGAN: Used for unpaired image-to-image translational tasks where images from one domain are transformed into another without needing paired samples.
- Condition GAN (cGAN): The generator and discriminator are conditioned on extra information such as text.

## 1.4 StyleGAN

Unlike traditional GANs where the generator takes a random noise vector as input, StyleGAN ues a style-based generator and maps the random input to

an intermediate latent space via a mapping network. This is done so the style can be controlled better at different layers of the network. AdaIN (Adaptive Instance Normalization) is employed to apply styles at different layers of the generator. This is done by adjusting mean and variance at each layer. This way the model can influce high-level features such as pose and shape in early layers, fine-grained features such as textures and colors in later layers. This separation allows for precise control over different aspects of the generated image. Next the StyleGAN generates image progressively from lower to higher resolutions. This ensures high fidelity outputs.
**Key takeaways:**

- Separation of coarse and fine features
- Latent space manipulation
- Style mixing

## 1.5 WGAN

The classical GAN uses Jensen-Shannon Divergence. However, this method has issues while working with gradients that can lead to unstable training. We therefore make use of Wasserstein distance to calculate the distance between 2 probability distributions, which is given as follows:

$$\underset{w \in W}{max} \underset{x \sim P_r}{\mathbb{E}}[f_w(x)] - \underset{z \in p(z)}{\mathbb{E}}[f_w(g_\theta(z))]$$

## 2 Variational Autoencoders

They are also generative models that learn a probabilistic mapping from a latent space to a data distribution.

### 2.1 Architecture

It comprises of a encoder that maps input data to a latent space and outputs a mean and variance, parametrizing a probability distribution. The VAE then samples from this distribution to a latent representation and introduces the stocahasticity that is key for generative tasks. The decoder then takes a sample from latent distribution and maps it back to the origina space, reconstructing the input. VAE optimizes a reconstrctuion loss and a KL divergence loss, which ensures that the latent space follows a known distribution (like a standard Gaussian). This regularization encourages, smoother latent space repsentations that can be sampled from.