Generative Image Models Overview

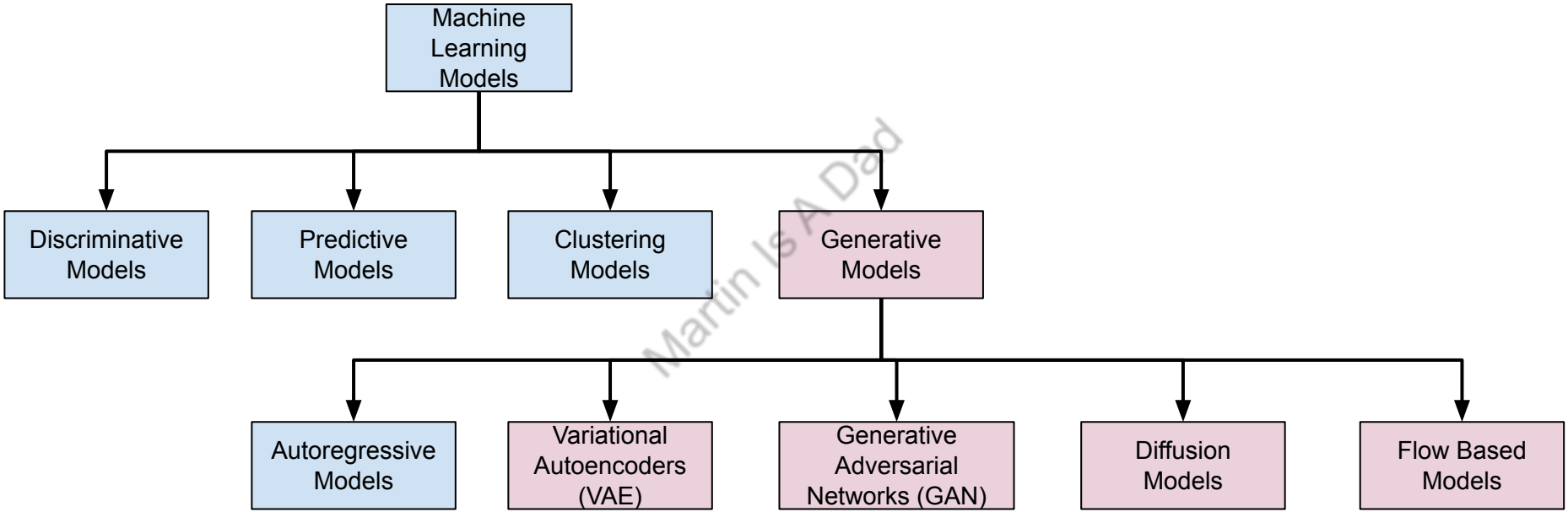
Goal

- Goes through all model types from a high level
  - Build intuition
  - Understand basic mechanism
  - Get familiar with high level architecture and essence
  - Get familiar with applications

- Deep dives on each model type will be **after** this overview
  - Architecture details
  - Training & sampling details
  - Math
  - More

Martin Is A Dad

Model Categories Overview

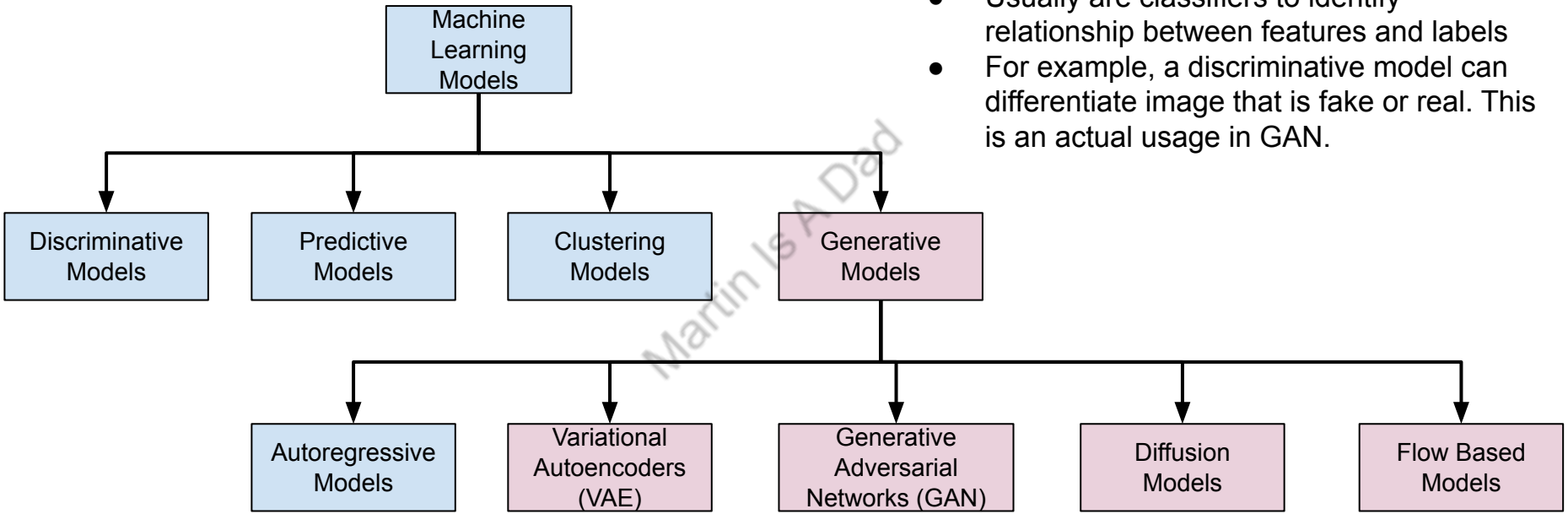


Martin Is A Dad

What We Won't Cover

Discriminative Model:

- Used in supervised learning with known labels.
- Usually are classifiers to identify relationship between features and labels
- For example, a discriminative model can differentiate image that is fake or real. This is an actual usage in GAN.

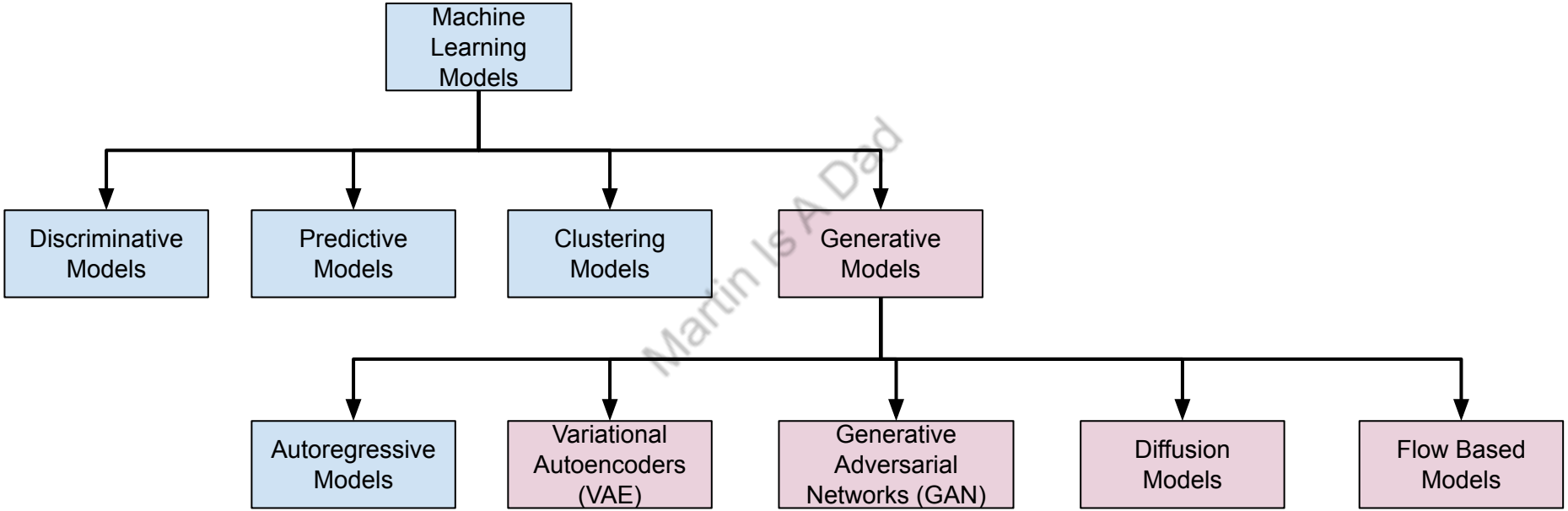


Martin Is A Dad

What We Won't Cover

Predictive Model:

- Use historical data to predict future data

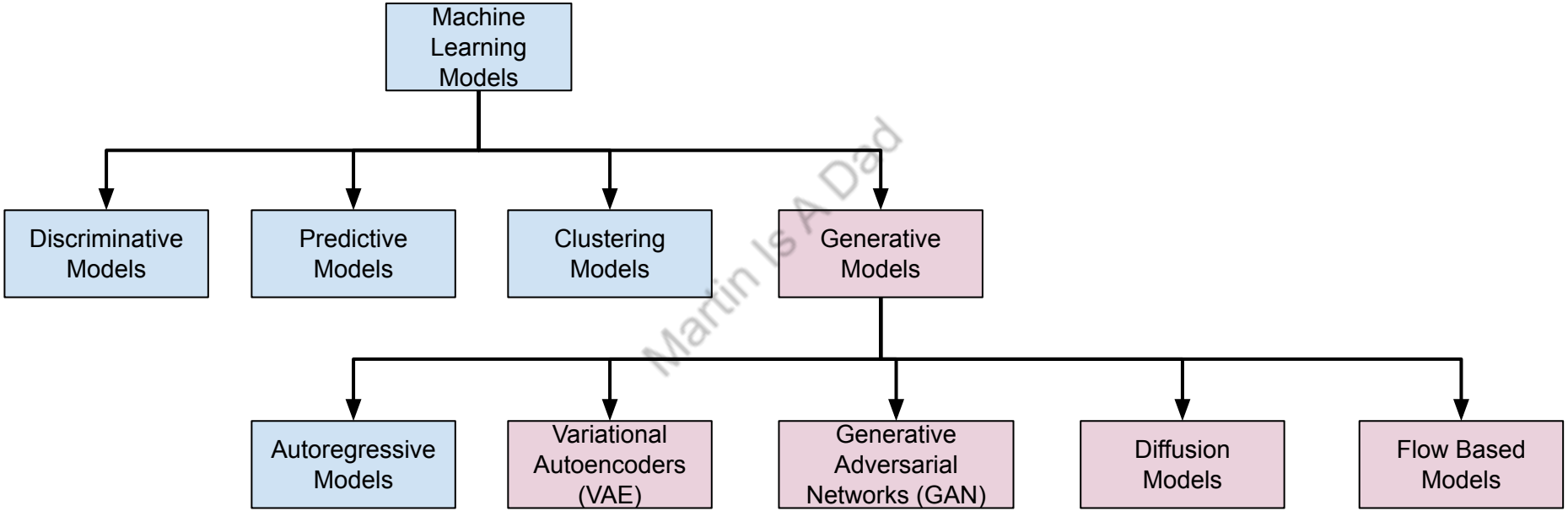


Martin Is A Dad

What We Won't Cover

Clustering Model:

- Used in unsupervised learning to group data set into clusters based on similarity

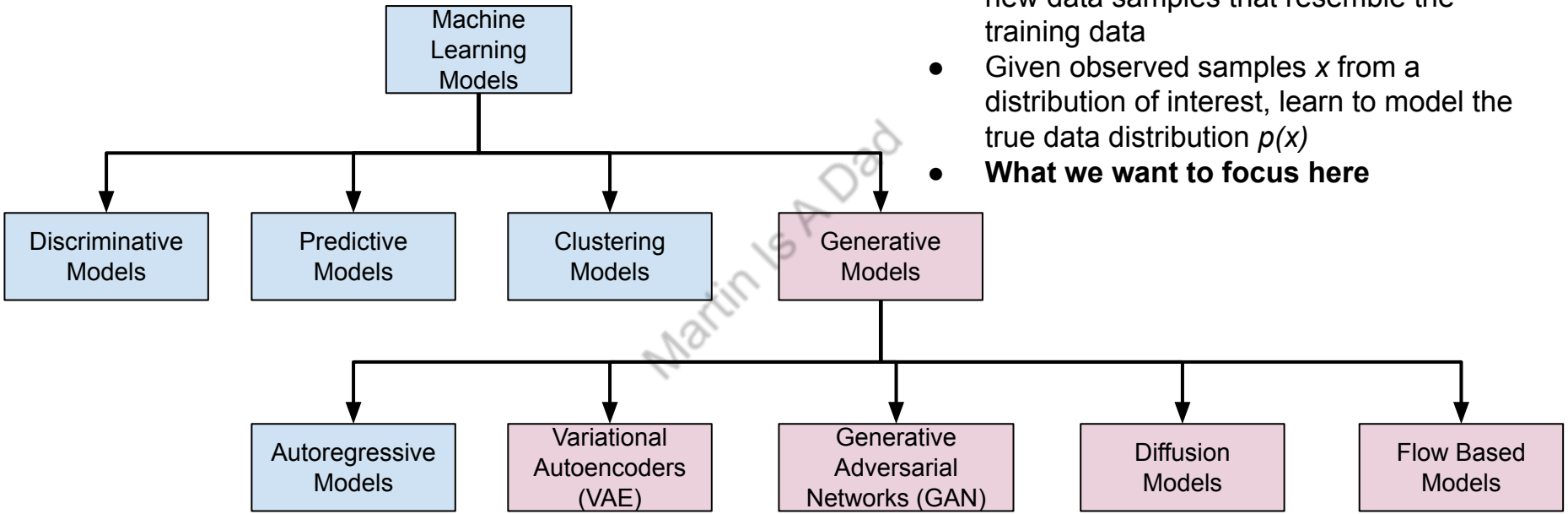


Martin Is A Dad

What We Won't Cover

Generative Model:

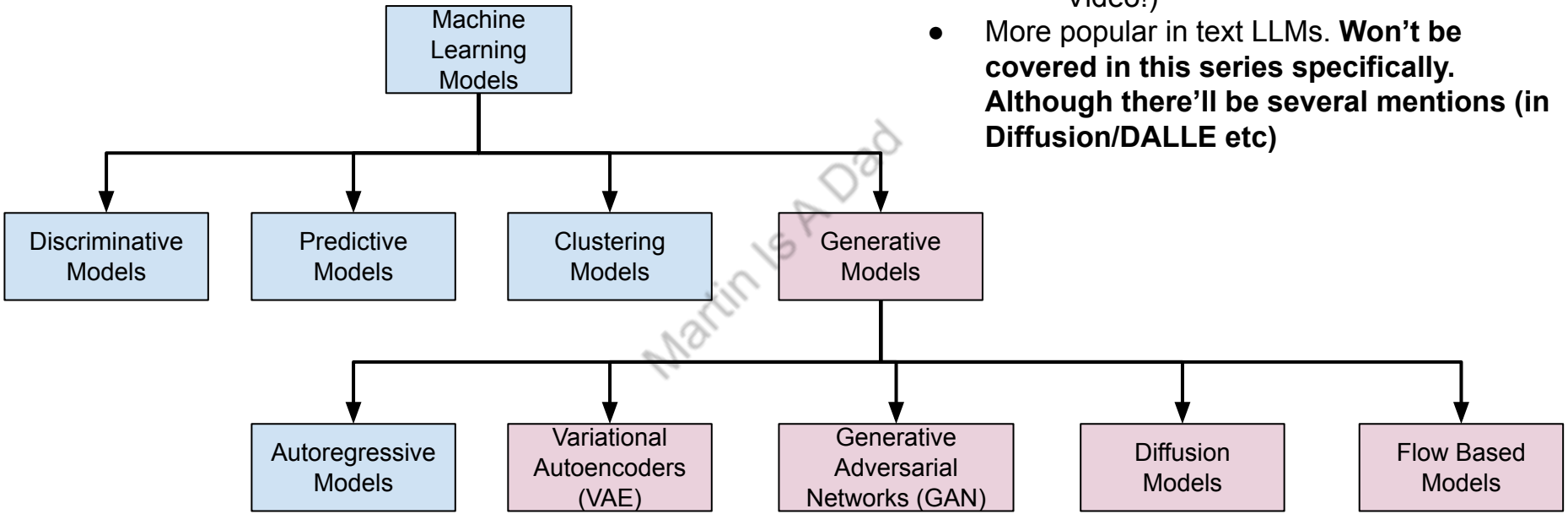
- Learns the underlying patterns and distribution of data, enabling it to create new data samples that resemble the training data
- Given observed samples *x* from a distribution of interest, learn to model the true data distribution *p(x)*
- **What we want to focus here**



Martin Is A Dad

What We Won't Cover

- Autoregressive Model
  - RNN
  - Transformer (check my Transformer Video!)
- More popular in text LLMs. **Won't be covered in this series specifically. Although there'll be several mentions (in Diffusion/DALLE etc)**



Martin Is A Dad

Autoencoder

- Autoencoder is composed of an encoder that learns to isolate the important latent variables from training data and a decoder that then uses those latent variables to reconstruct the input data.

- Latent variables are underlying variables of data that inform the way the data is distributed but are often not directly observable. Good example is ***Plato's Allegory of the Cave***. Group of people are chained inside a cave their entire life and can only see the 2-d shadows projected onto a wall in front of them, which are generated by unseen 3-d objects passed before a fire. To such people, everything they observe is actually determined by higher-dimensional abstract concepts that they can never behold. So the 3-d is latent space, and 2-d is observed. **However, in image, usually latent space is lower dimension.**

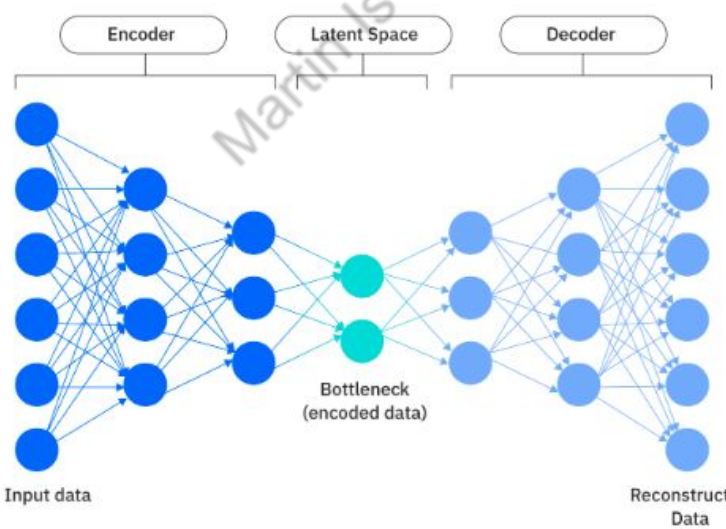
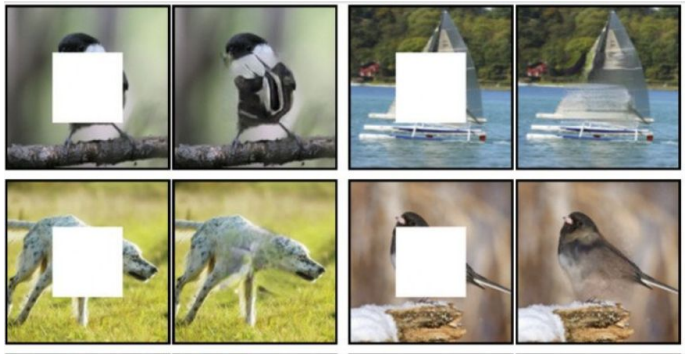
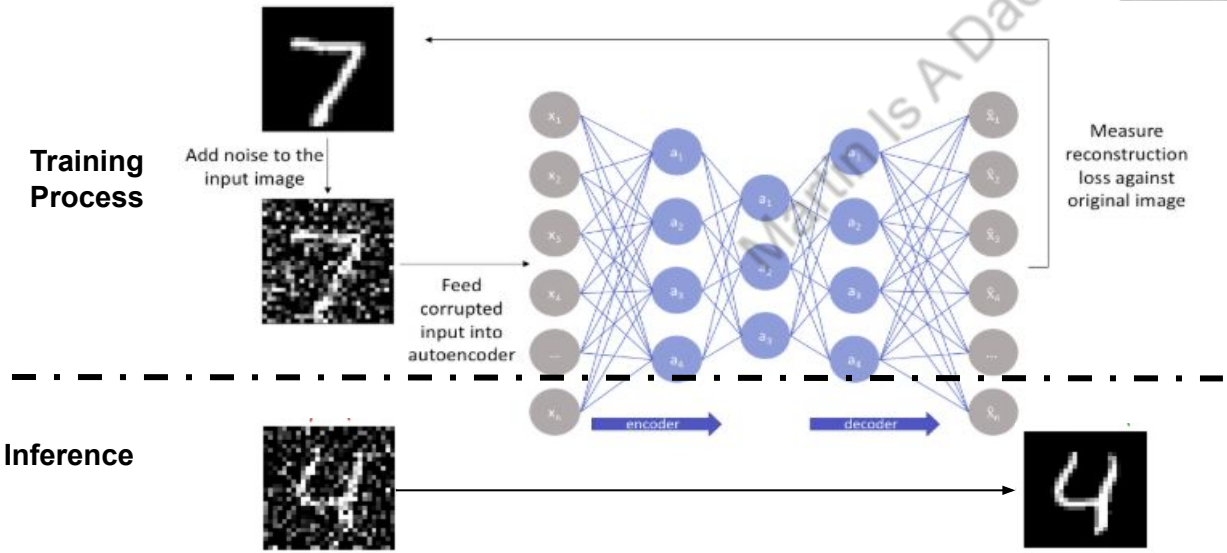


Image from IBM VAE blog

Autoencoder

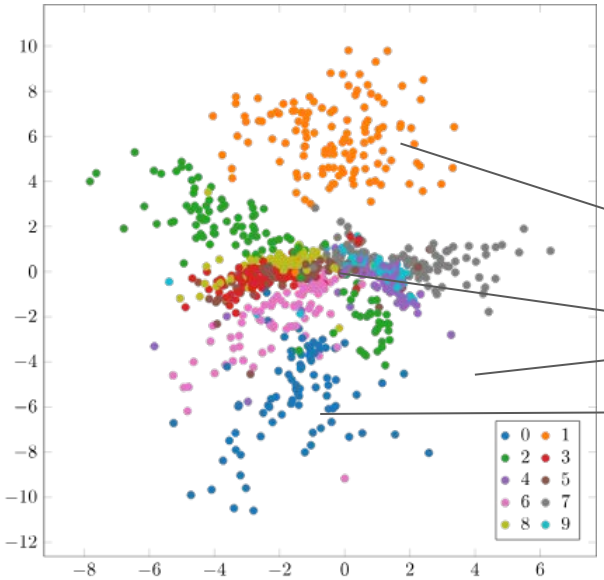
- Autoencoder have many applications:
  - Image compression
  - Extract latent embeddings from image
  - Image restorating
  - Magic Eraser
  - Denoise





Variational Autoencoders (VAE)

- Goal of autoencoder is to reconstruct the original image. However, for a generative model, what we want is `**Given observed samples *x* from a distribution of interest, learn to model the true data distribution *p(x)***`
- In plain words, we want to generate *new* samples that *resemble* the original input. That's why AE is not good for Generative task. Other reasons are:
  - Autoencoder are fixed, deterministic mapping and discrete on the latent space



If sample a random point from this 2-D space and feed it to the decoder, will it generate a new image that is "similar" to one of the 0-9 digit images in the training set?

Probably will generate a good `1`

???

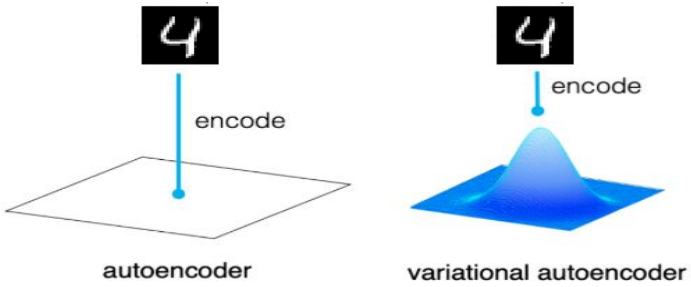
Probably will generate a good `0`

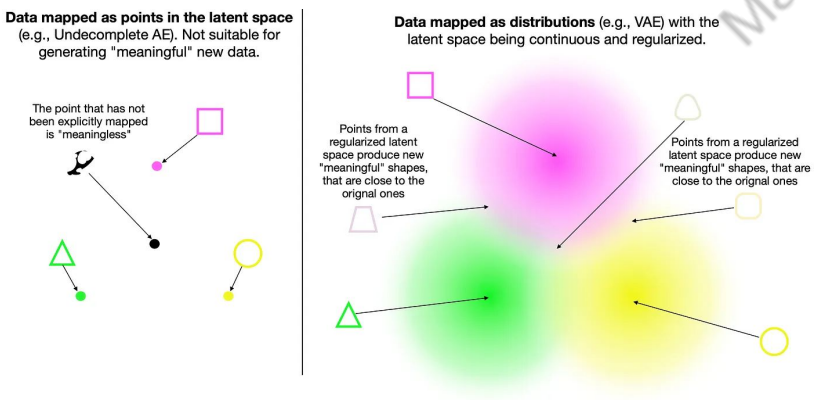
Martin for Dad

Latent space of MNIST data (number classification)

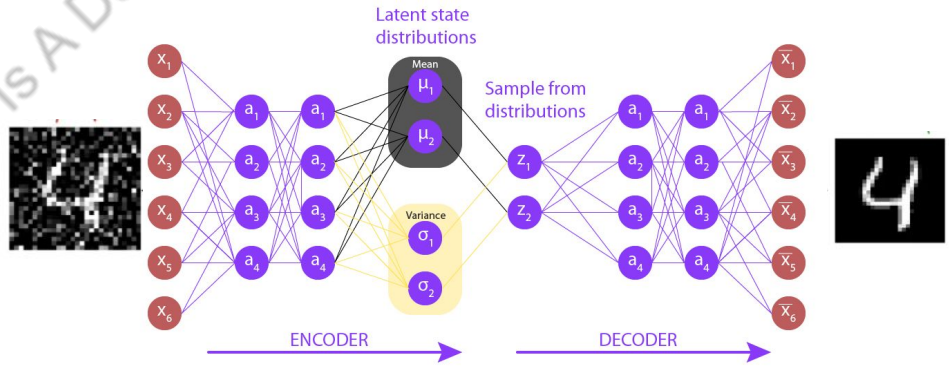
Variational Autoencoders (VAE)

- VAES are **probabilistic** models. VAEs encode latent variables of training data not as a fixed discrete value z, but as a continuous range of possibilities expressed as a probability distribution p(z), represented with mean vector (μ) and a std-deviation / variance vector (σ).



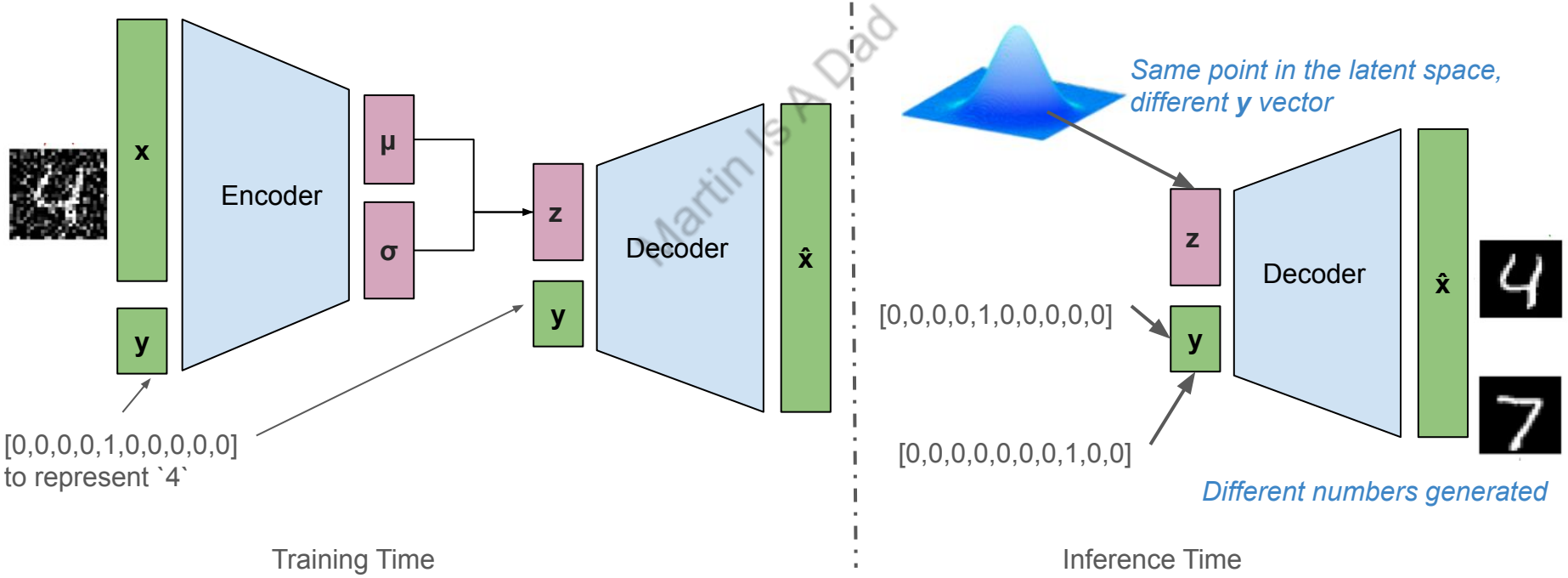


Martin Is A Dad



Conditional VAE

- With VAE, we can sample a random point in the latent space and generate new images similar to training set. However, we can't generate an new on demand (say generate a specific number with the MNIST example)
- With Conditional VAE, we will be able to achieve this
  - CVAE has an extra input to both the encoder and the decoder, usually an one-hot vector.

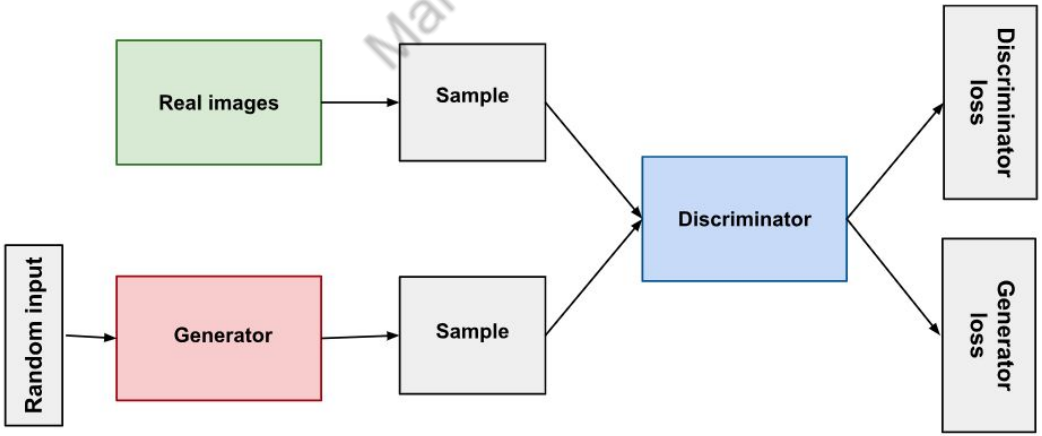


Generative Adversarial Network

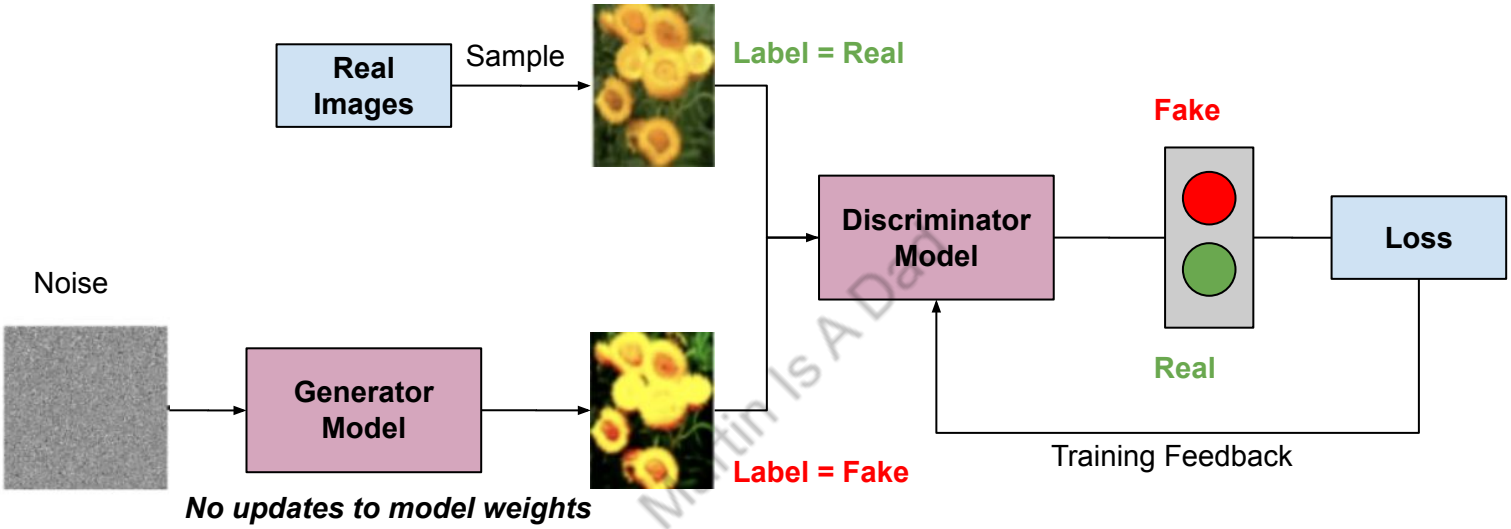


Generative Adversarial Network

- Consists of two models – a generator and a discriminator – that are trained in an adversarial process, where the generator tries to create data that looks realistic and the discriminator tries to distinguish between real and generated data.
- During Training
  - **Generator**: Takes random noise as input and generates data samples (e.g., images, text, or audio) that it hopes will fool the discriminator.
  - **Discriminator**: Takes both real data and the generated data from the generator as input and tries to determine whether the input is real or fake.
  - **Goal**: The goal is for the generator to reach a point where the discriminator can no longer reliably distinguish between real and generated data.
- Post Training: only generator is used to generate new real-like fake images

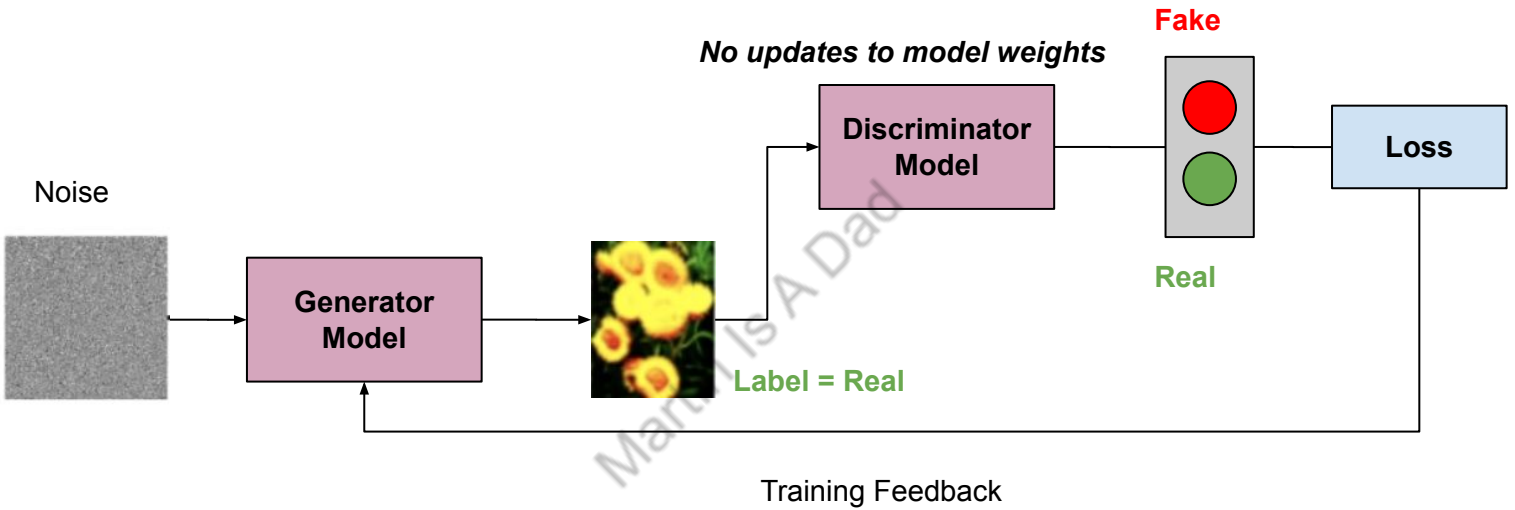


GAN Training Phase 1: Discriminator Training



- N real images, and N fake images produced by generator, are given to the discriminator to classify each image as fake or real.
- Discriminator's weights are updated based on its loss.
- Generator's weights are not updated

GAN Training Phase 2: Generator Training



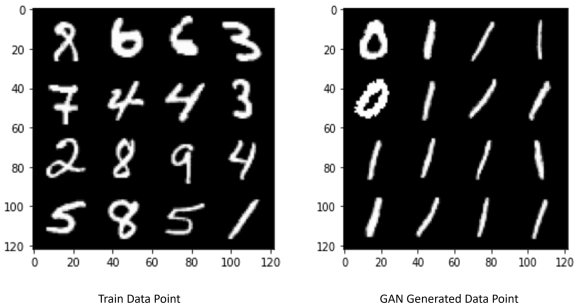
- Generator generates N fake images, feed it to the trained discriminator asking for classification
- **We label all the fake images with real when computing loss based on discriminator's output**
- If discriminator classifies as fake, this is a negative feedback; if classifies as real, this is a positive feedback

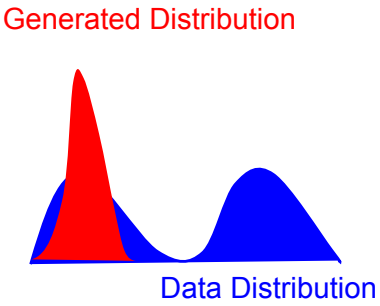
GAN Training

We then repeat phase 1 & phase 2 many many times until generator is good enough.

Challenges of GAN

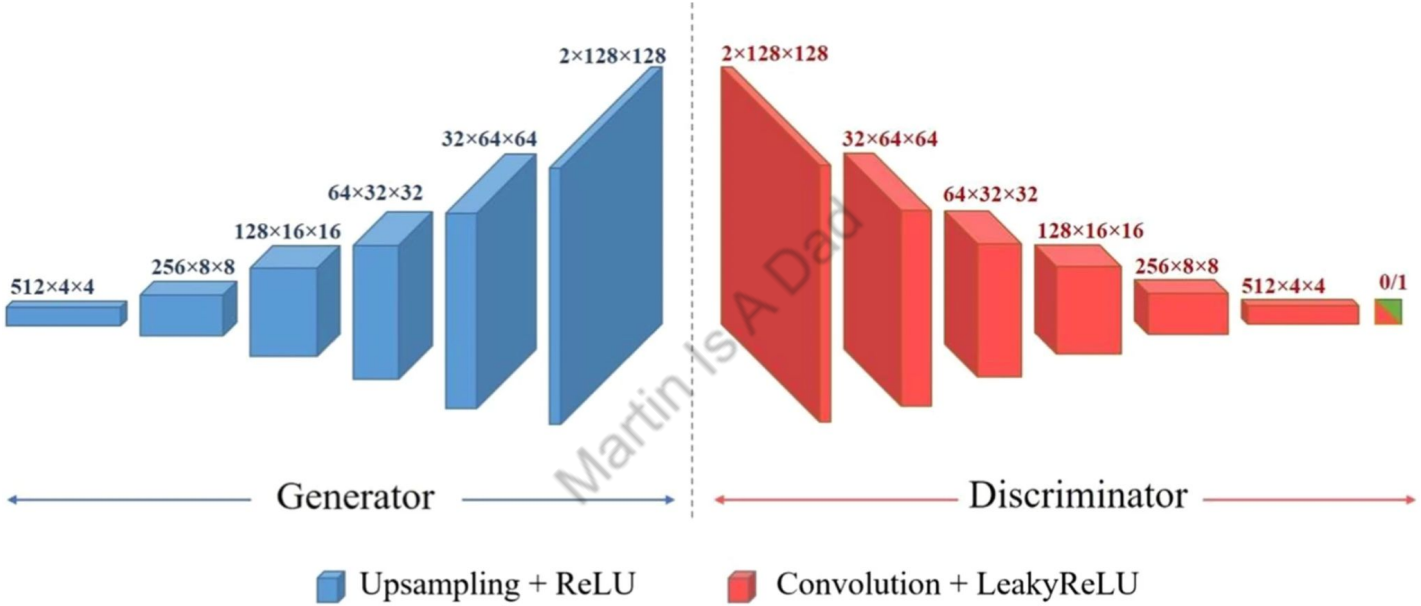
- **Data Quality and Quantity**: GANs often require large, high-quality datasets to train effectively.
- **Training unstable: generator** and discriminator failing to reach a stable equilibrium, leading to oscillating or diverging behavior
- **Generator hard to train**: Discriminator can have almost 0 loss. Even when generator is already generating realistic image, Discriminator can still figure it out. The gradients used to train the generator can become very small or zero. This makes it difficult for the generator to learn and improve, even if the discriminator is still improving.
    - We can add noise to discriminator input, or even make the labels noisy
- **Mode Collapse**:
    - Generator only produces a limited number of outputs instead of the variety that is desired





GAN Architecture

Generator: similar to the decoder in Autoencoder

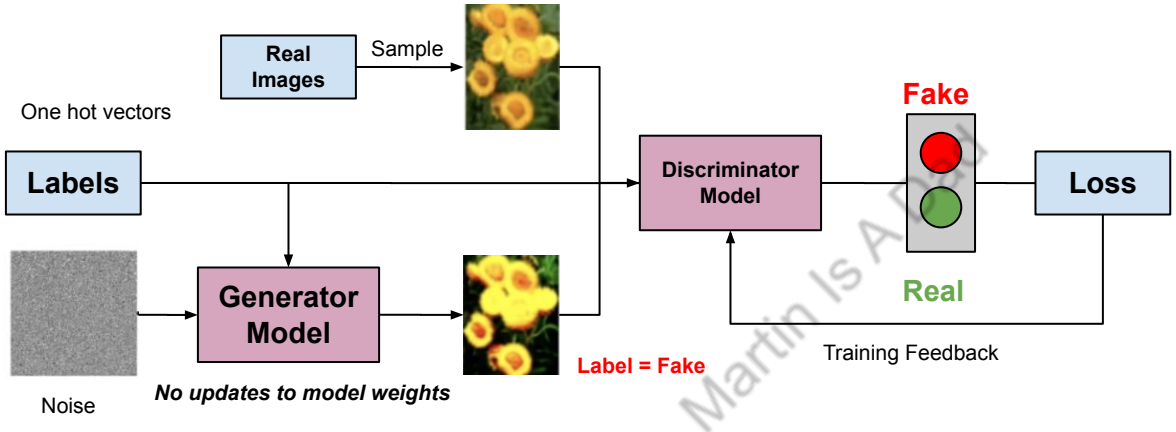


Discriminator: similar to the encoder in Autoencoder

Image from paper (Fully-automated deep-learning segmentation of pediatric...)

Conditional GAN

Similar to CVAE, we have conditional GAN to somewhat control the generated images



Martin Is A Dog

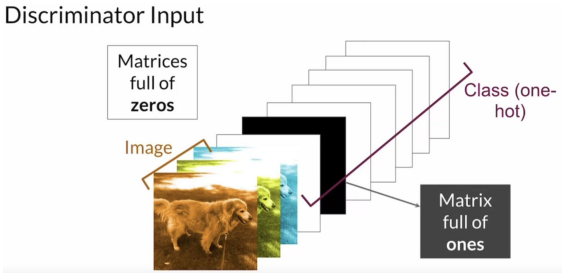
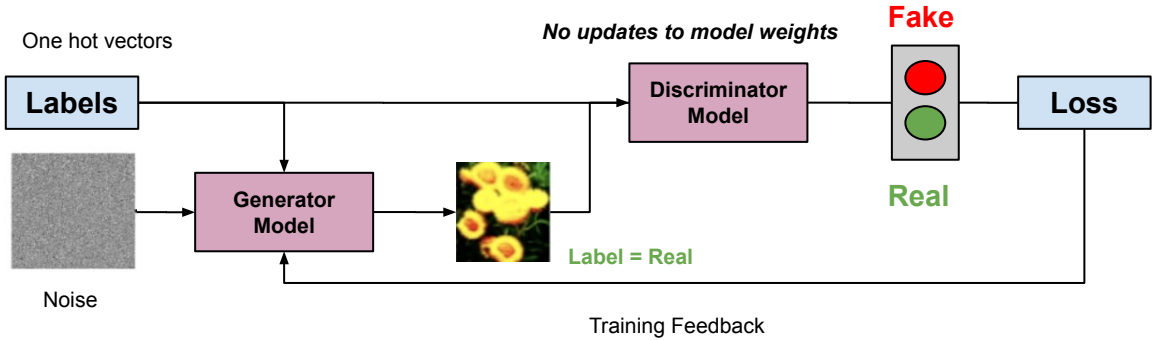
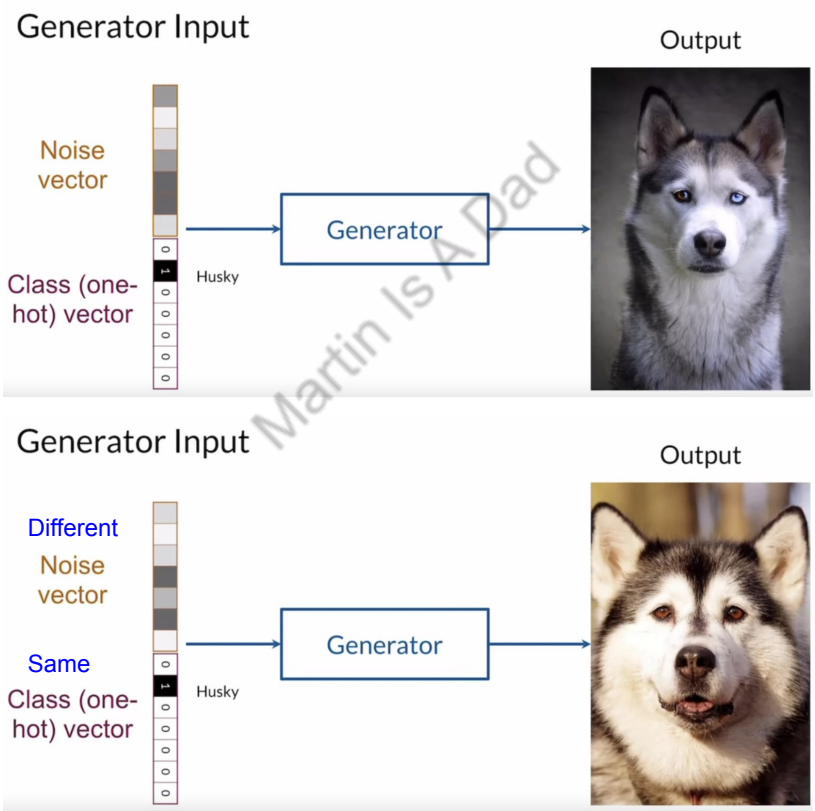


Image From DeepLearning.ai



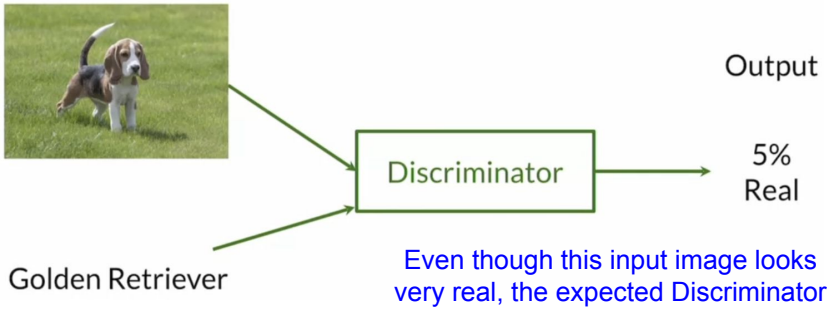
Condition input to the Generator

Similar to Conditional VAEs (input to decoder)



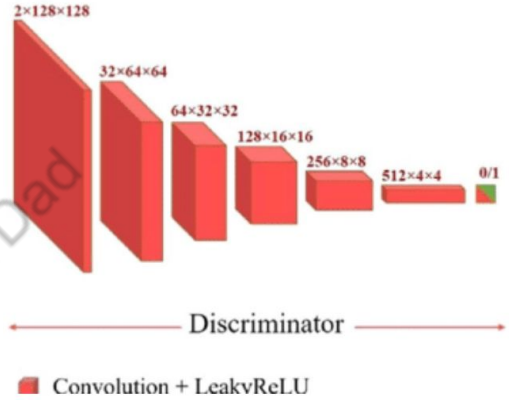
Generates different image but from the same class

Condition input to the Discriminator



Even though this input image looks very real, the expected Discriminator output will be “Fake” bcos the image does not match the class label

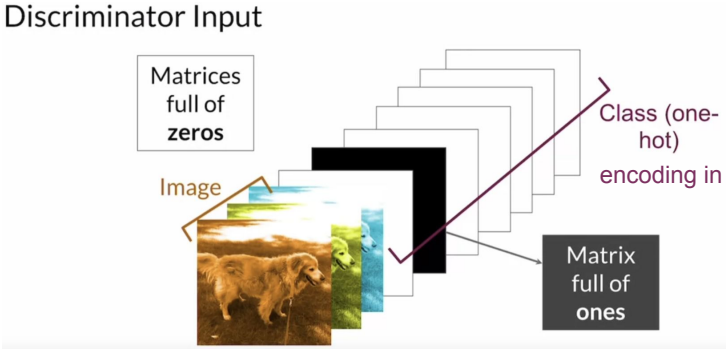




Discriminator

Input is a 3D image.

How will we add additional “class information” as input to the Discriminator?

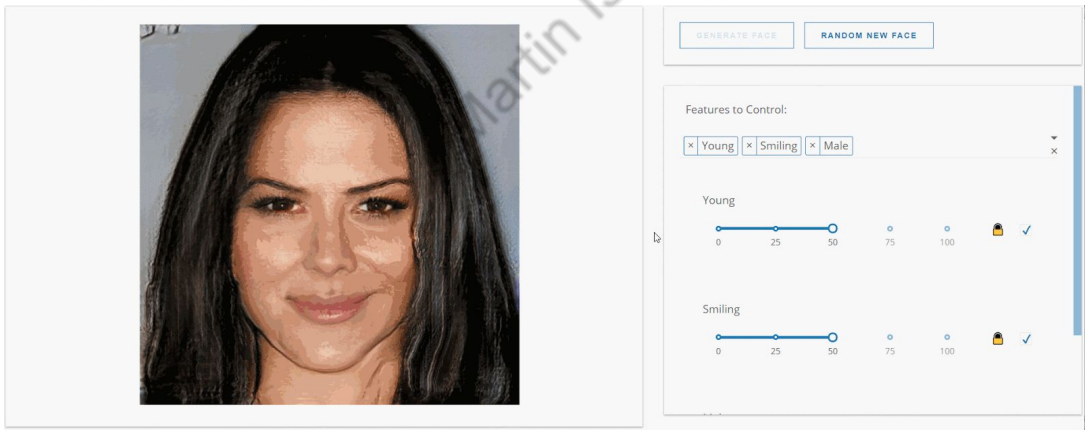


Martin Is A Dad

More GANs



Cycle GAN



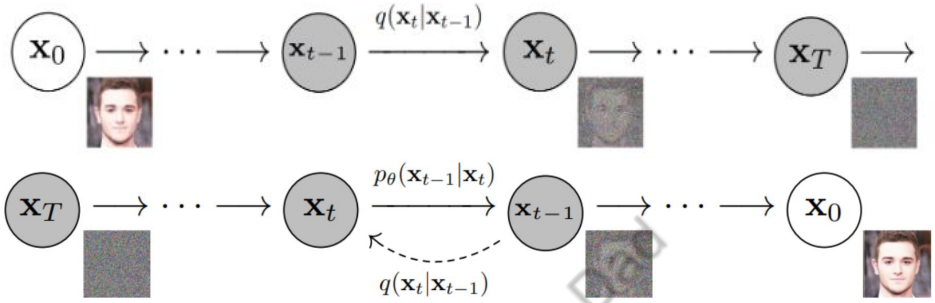
Controllable GAN

And MORE! Stay tuned in following episodes!

Diffusion Models



Diffusion Models



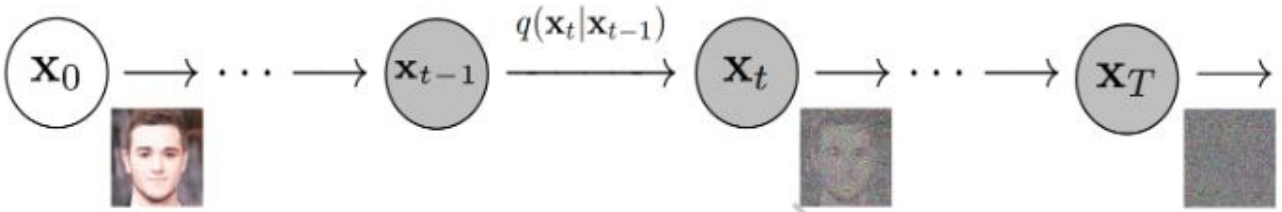
Intuition

- For images in training dataset, diffusion models gradually add noise to the original image, step-by-step, until it's destroyed, then learn to reverse that diffusion process and reconstruct the original data distribution.
- A trained diffusion model can generate new image that resemble the training data by simply denoising a random initial sample of pure noise. This is similar to a denoising autoencoder that we have covered before.

Method

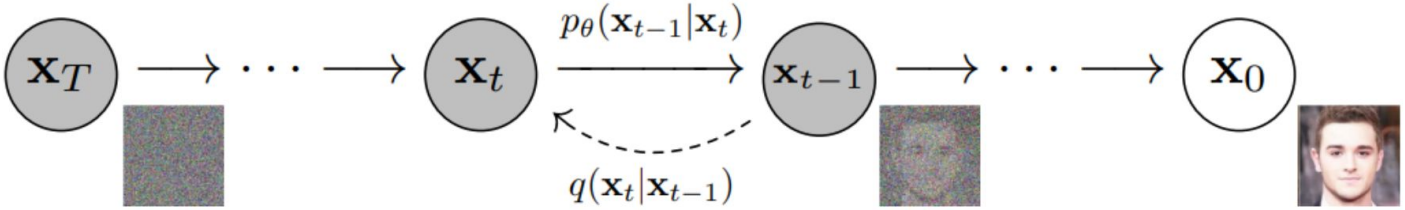
- Train an encoder-decoder model to: take a noisy image as input and predict the noise such that the original image can be reconstructed by subtracting the predicted noise from the input image
  - Not predicting the original image since our goal is not to reconstruct image, but to get the underlying actual distribution to generate new image. More details in following episodes.
- Loss can be computed based on the difference between the actual noise and the predicted noise

Diffusion Models: Forward Diffusion Process



- Image is transformed into pure noise (standard normal distribution) in T steps
- You can view it as a fixed Markov chain with T steps, where the image at timestep t maps to its subsequent state at timestep t+1. As such, each step depends only on the previous one.
- Gaussian noise added at each step in the Markov chain is not constant or arbitrary. Instead, the noise is derived from the structure of the original image and the rate at which it's added steadily increases with each consecutive step, all determined by a **scheduler**
- More details in following episodes.

Diffusion Models: Reverse Diffusion Process



Martin Is A Dad

- Where the actual machine learning happens.
- Directly transforming random noise into a new image is extremely difficult and complex, but transforming a noisy image into a slightly less noisy image is relatively easy. Diffusion models therefore formulate the **reverse diffusion process** as an incremental, iterative transformation of Gaussian noise to actual images
- You might think a simple reverse of forward diffusion process will work, but in practice computing $q(x_t|x_{t-1})$ is impossible. Instead the model predicts the entire noise present in x_t, then removes a fraction of that noise to get to x_{t-1}. More details in later episodes
- More details in following episodes.

Diffusion Models: Architecture

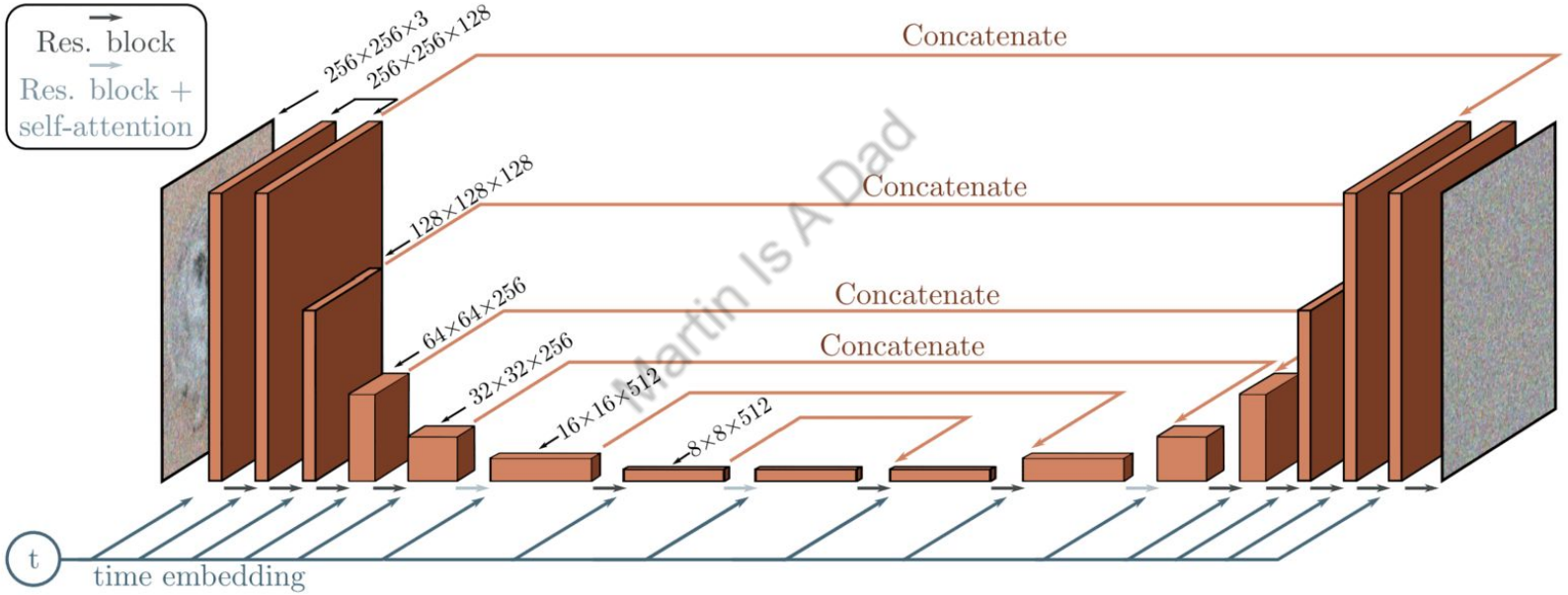
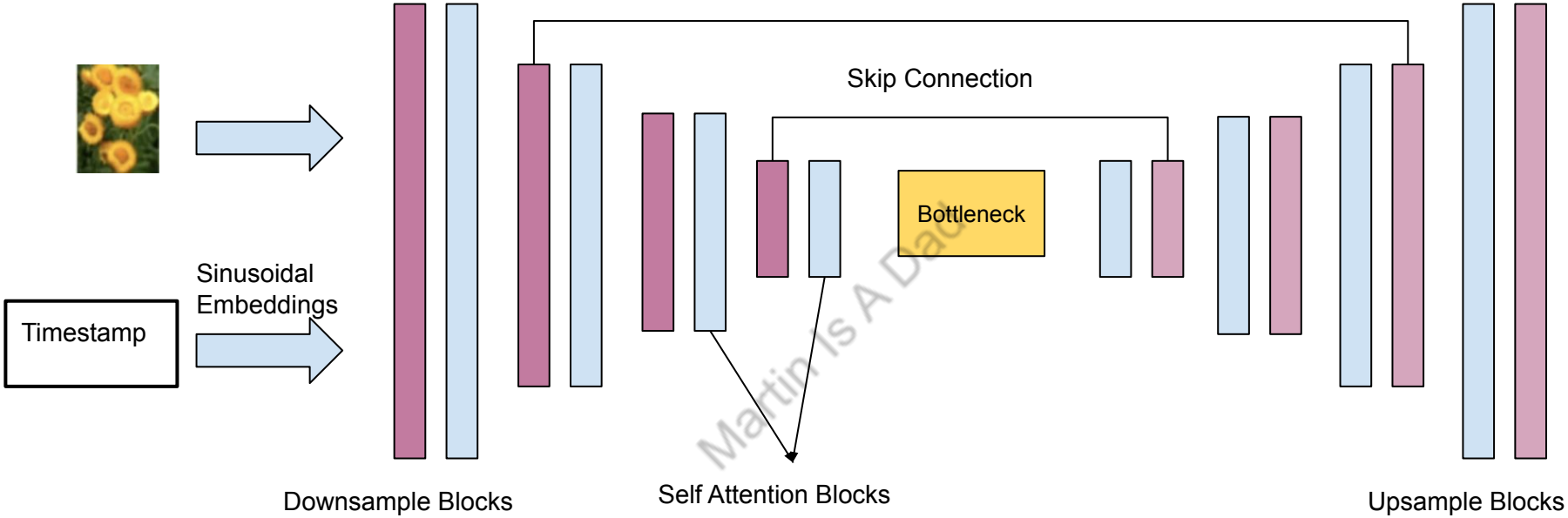


Image from [learnopencv.com blog]

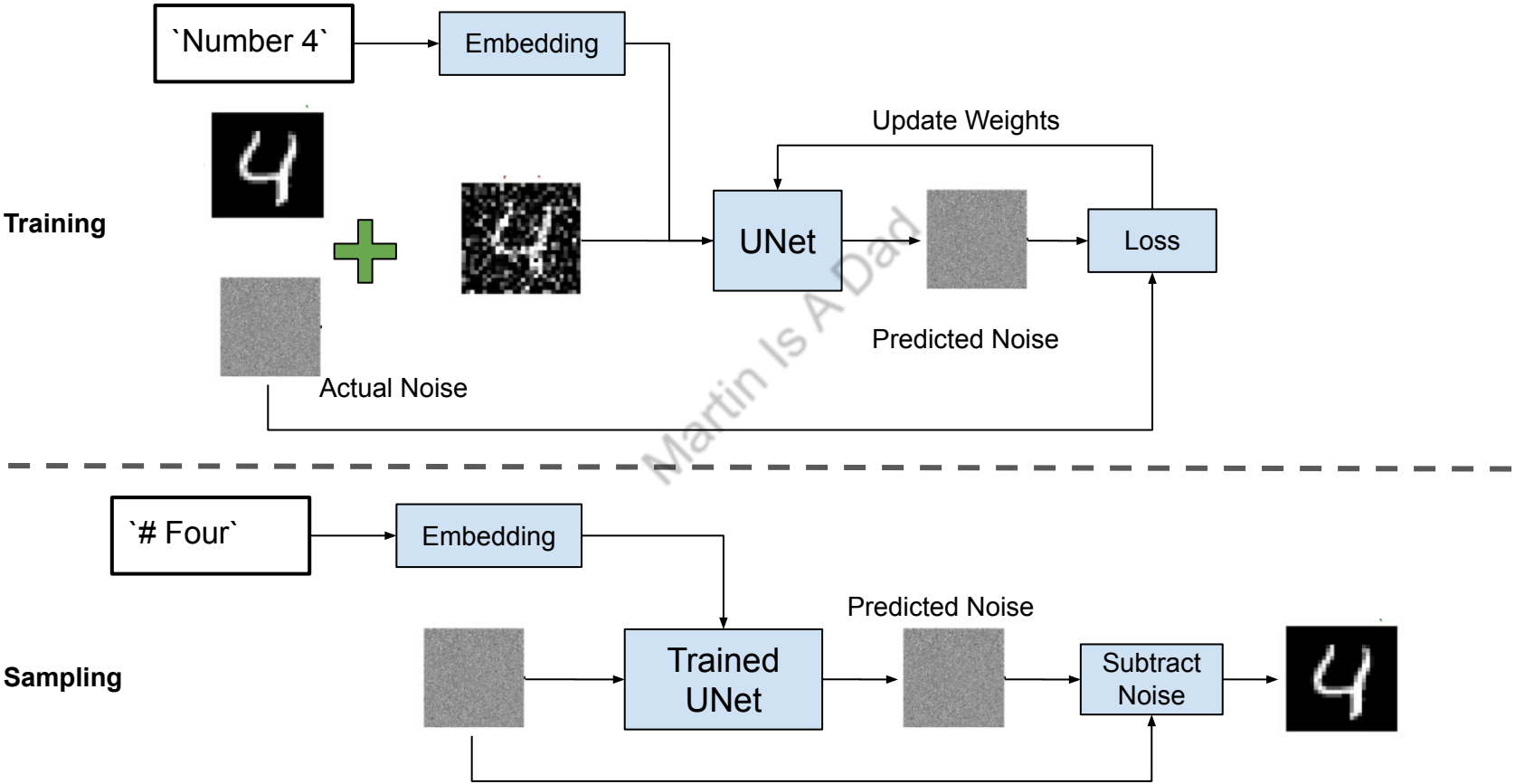
Diffusion Models: Architecture



- UNet-like architecture
- Input and Output size are identical
- Predicts noise of image

More details in following episodes!

Guided Diffusion Models

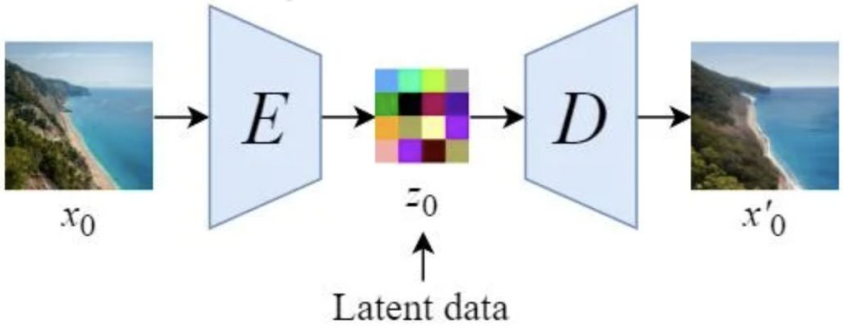


Latent Diffusion (Stable Diffusion)

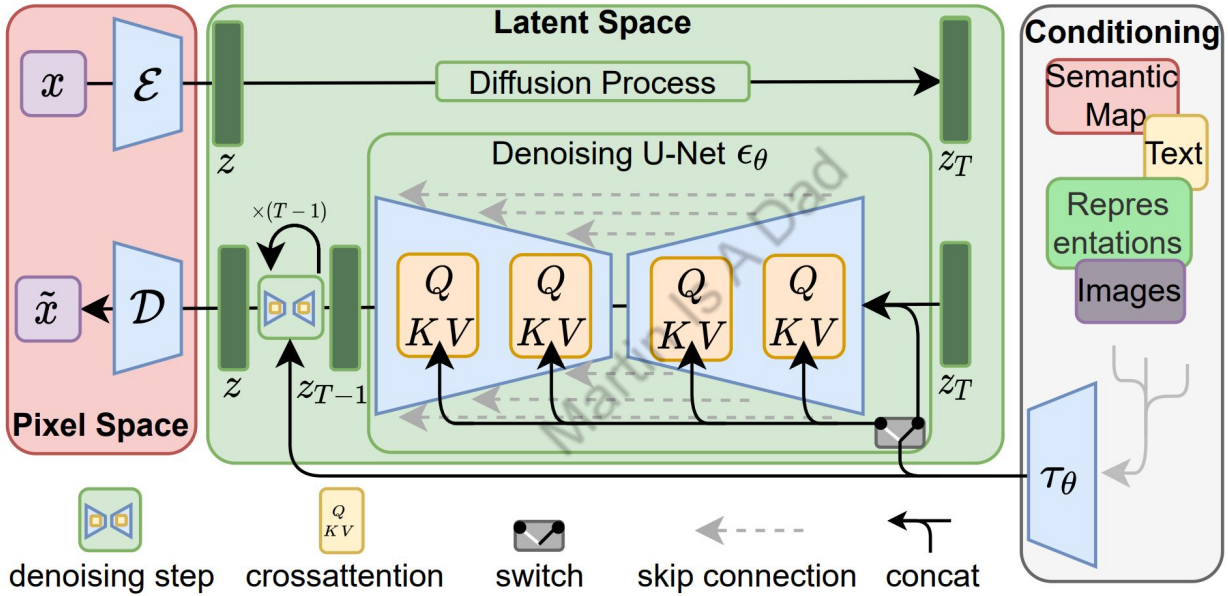
- Motivation
  - Training and sampling of diffusion model can be expensive, computation and memory wise, since it requires multiple denoising passes
  - Can we introduce a lower dimension latent space and perform diffusion there?
  - **Stable Diffusion** performs the diffusion process in the latent space

- TLDR
  - Trained Encoder for encoding a full-size image to a lower dimension representation (latent space).
  - Forward diffusion process and the reverse diffusion process within the latent space.
  - Trained Decoder can decode the image from its latent representation back to the pixel-space

Martin Is A Dad



Latent Diffusion (Stable Diffusion)



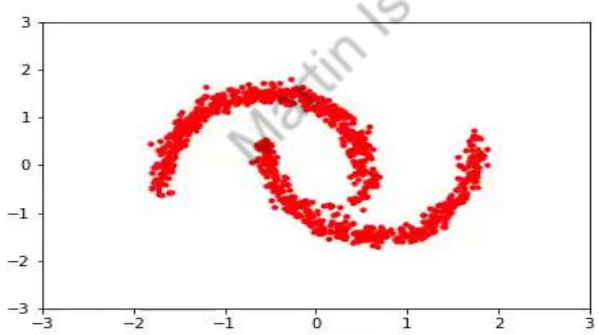
- Cross attention from Transformer appears again :)
  - K, V are from text prompts, Q is from the latent representation of image
- More details in following episodes!

Flow Based Model

Intuition

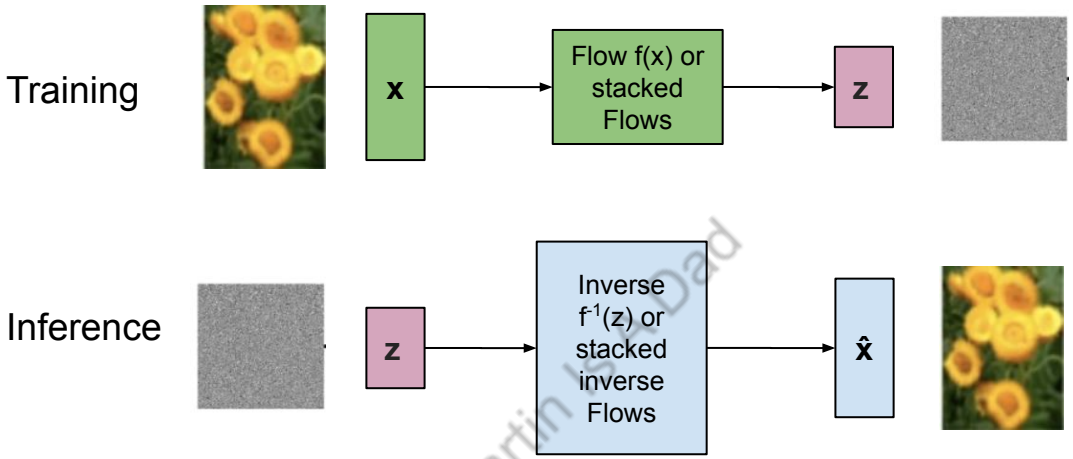
- Goal is to “generate new images” out of “pure noise” (samples from normal distribution)
- What if we:
  - Learn a function f(x) that maps image’s complex data distribution to “normal distribution” (same dim as input)
  - Then use f-inverse $f^{-1}(x)$ function to convert a random sample from the normal distribution space to generate a realistic image

Martin Is A Dad



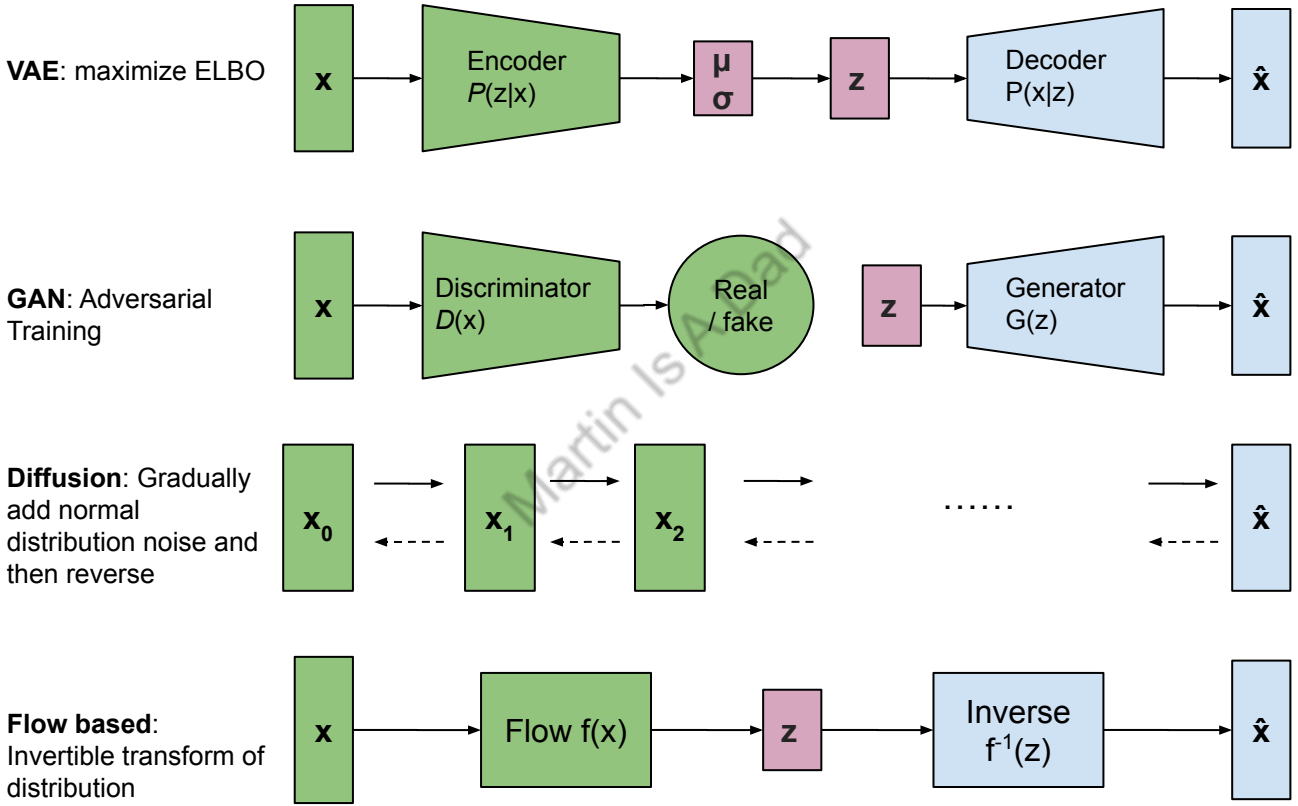
Map from complex input data distribution to a normal distribution

Flow Based Model



- Latent space “z” is same dimension as input “x”
- There should almost no data loss
  - f(x’) = z’, then $f^{-1}(z’) = x’$
- We can use f^{-1} to generate new real-like fake images, by samping random point in latent space
- More details about magical function f in following episodes

Different Generative Image Models



Martin Is A Dad

What's Next

- Autoencoders and VAE Deep Dive
- GAN and Variances Deep Dive
- Diffusion Models Deep Dive
- Flow Models Deep Dive
- CLIP, DALLE (not diffusion) and DALLE-2 (diffusion) Deep Dive

Martin Is A Dad

STAY TUNED :D