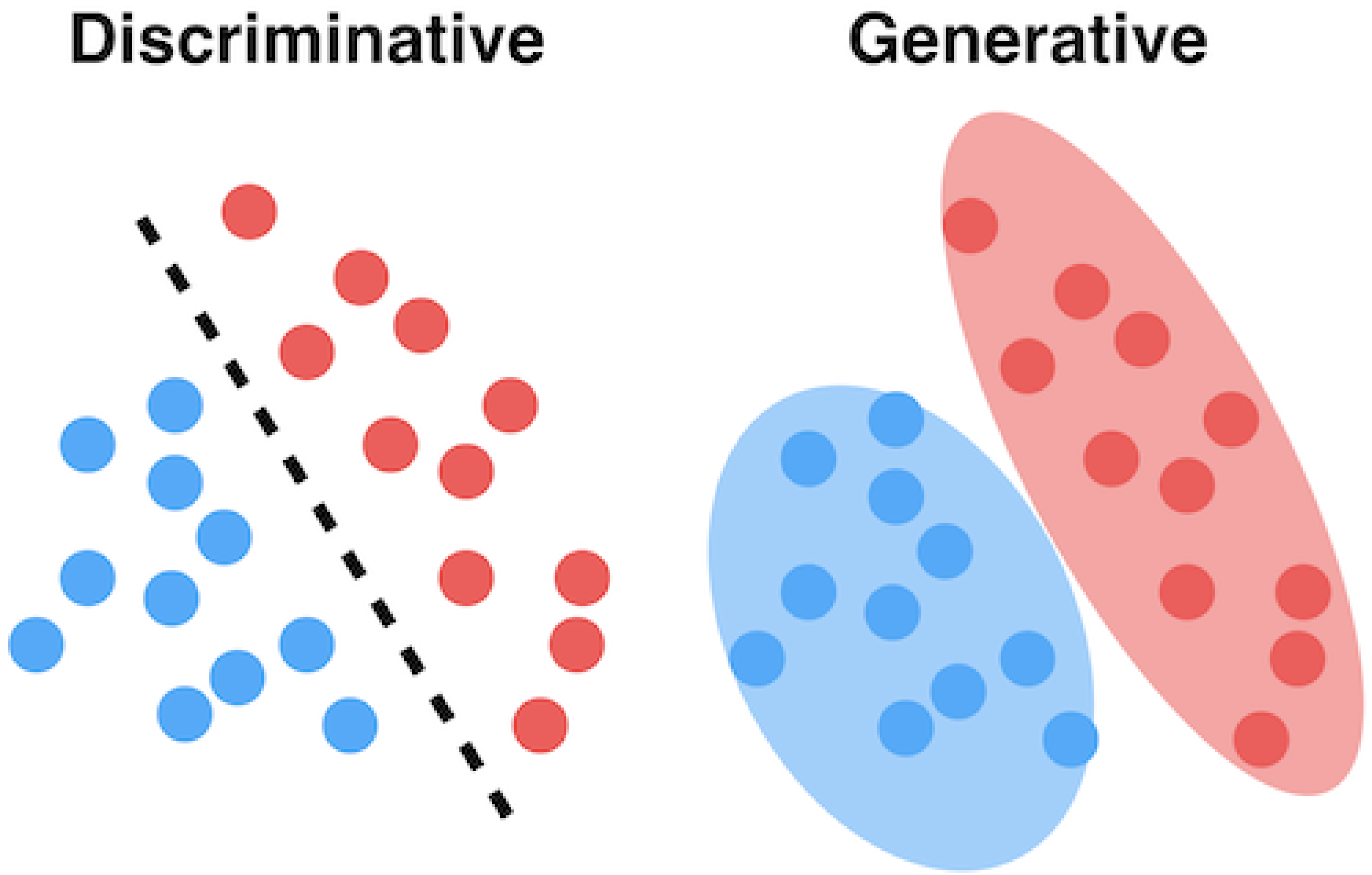
Generative Models

BITS F312: Neural Networks and Fuzzy Logic, Lab 09

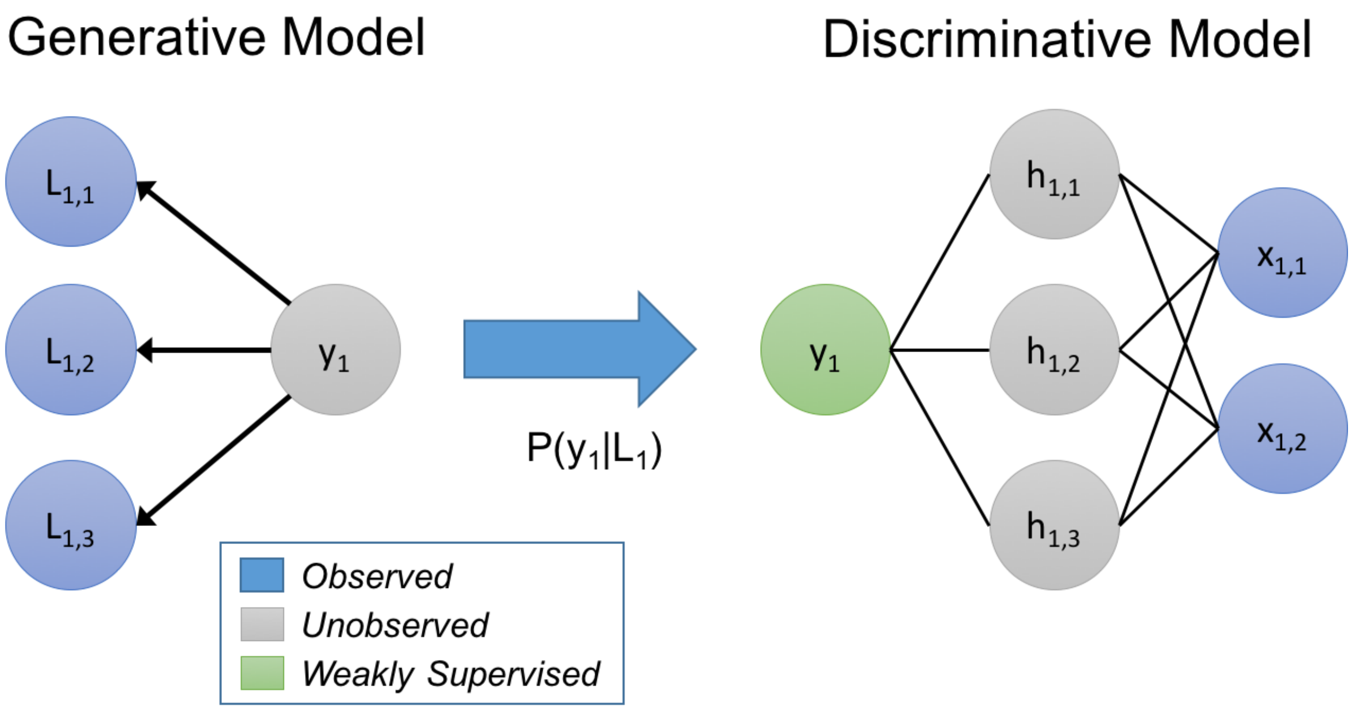
Generative Models

- Two types of classifiers – Generative and Discriminative.
- Discriminative models learn to estimate the conditional probability distribution, P(Y|X) for the data.
- Generative classifiers learn to estimate the joint probability distribution of X and Y, i.e. P(X, Y) .



Generative Models

- Intuitively, one can imagine that generative models model a probabilistic relationship between X and Y and hence provide detailed insights into what values of X generate particular values of Y and vice-versa.
- We can hence use generative models to "generate" a random value for X given a value for Y.



Example 1

“Generative Adversarial Style Transfer Networks for Face Aging”, Palsson et al. (ETH Zurich)

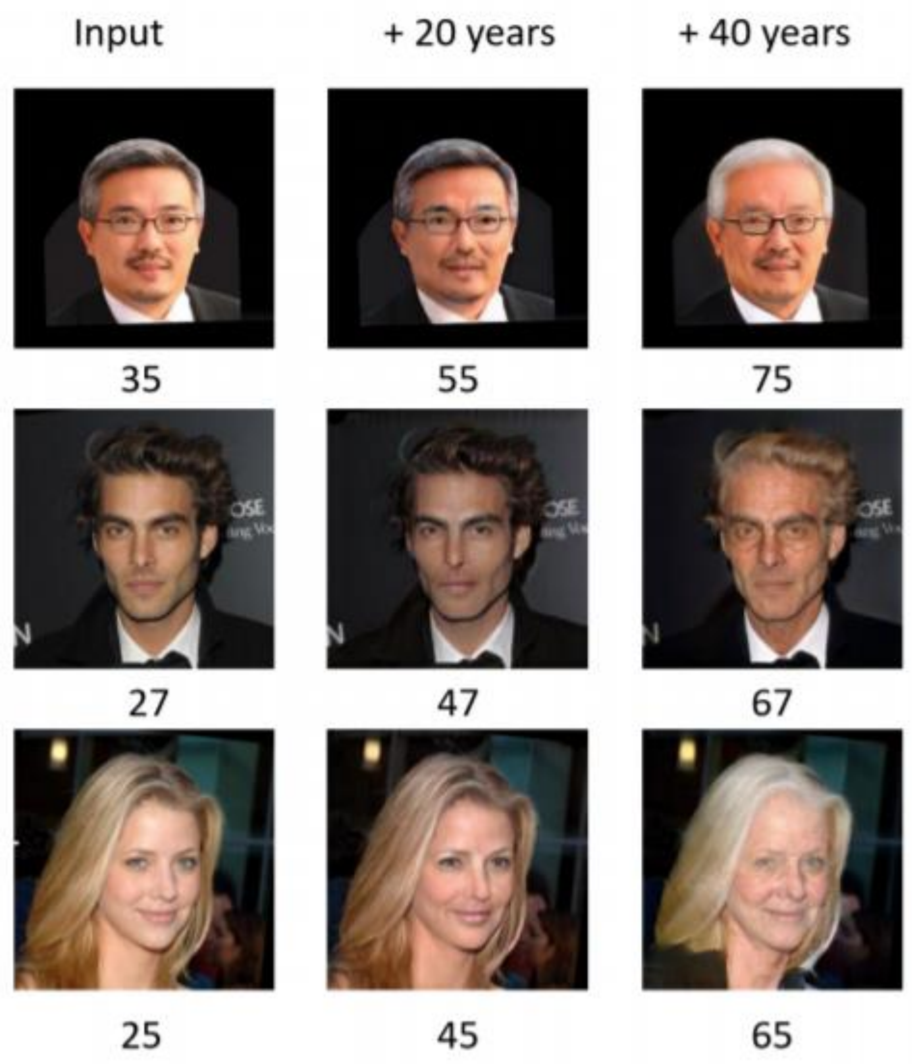


Figure 1. Face aging results of our methods.

Example 2

"Towards the Automatic Anime Characters Creation with Generative Adversarial Networks", Jin et al. (Fudan University)

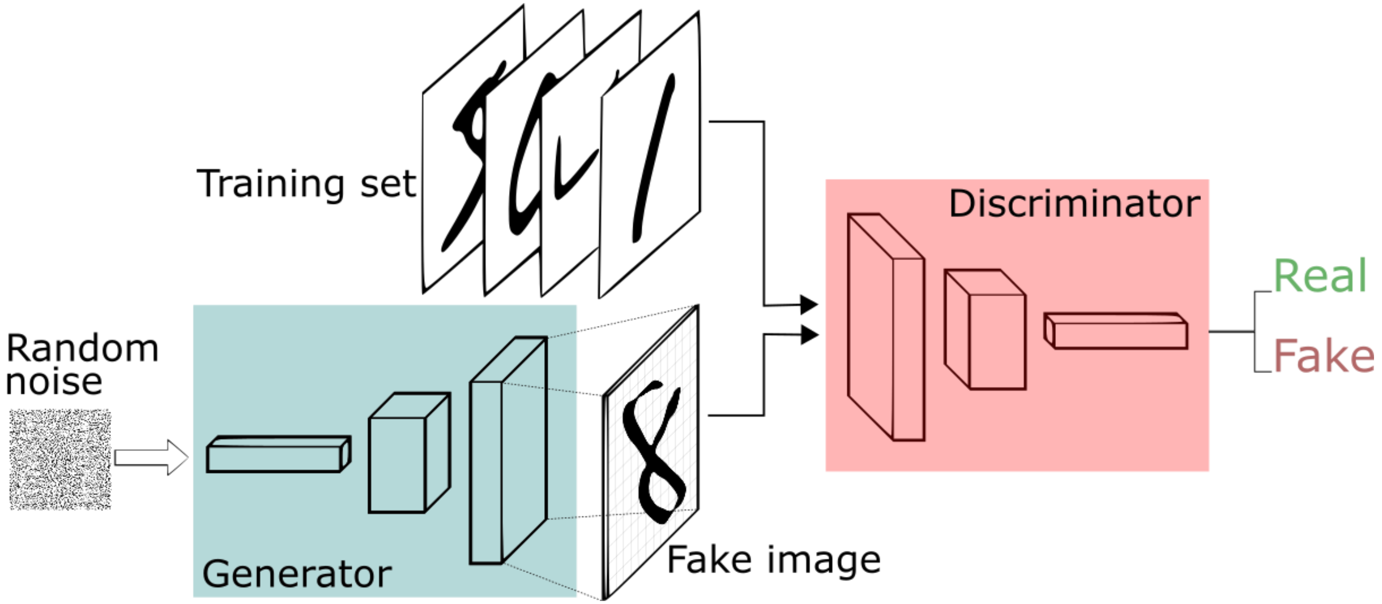
Figure 7: Generated samples

Different Generative Models

Model	Paper
Naïve Bayes	Reference
Hidden Markov Model	Reference
Latent Dirichlet Allocation	Blei et al.
Gaussian Mixture Model	Reference
Deep Boltzmann Machines	Salakhutdinov et al.
Deep Belief Nets	Hinton et al.
Variational Autoencoders	Pu et al.
Generative Adversarial Networks	Goodfellow et al.
Generative Moment-Matching Networks	Li et al.
Neural Autoregressive Distribution Estimator (NADE)	Larochelle et al.
Adversarial Autoencoders	Makhzani et al.

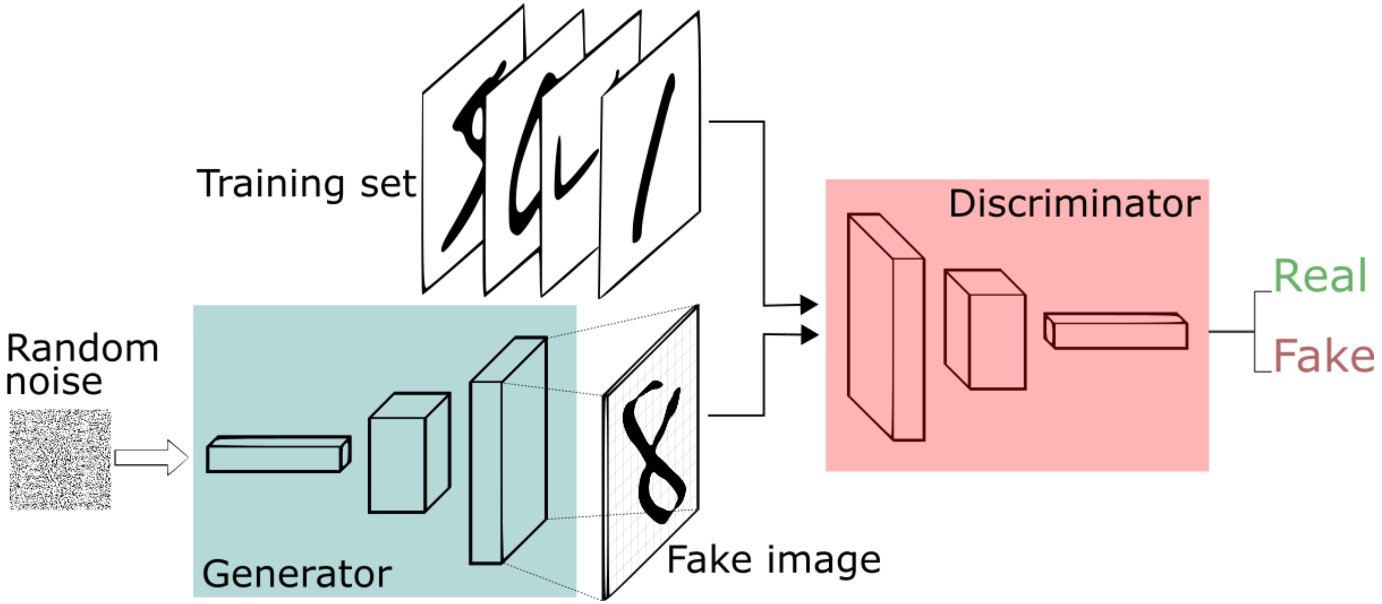
Generative Adversarial Networks (for images)

- A generative model based on a two-player minimax game, where the two players are a Generator (G) and a Discriminator (D).
- The Generator creates an image (the fake image) from a random distribution, $P_z(z)$ initially and the discriminator compares it with an instance from the data, *x* (the real image), which has distribution p_{data}.



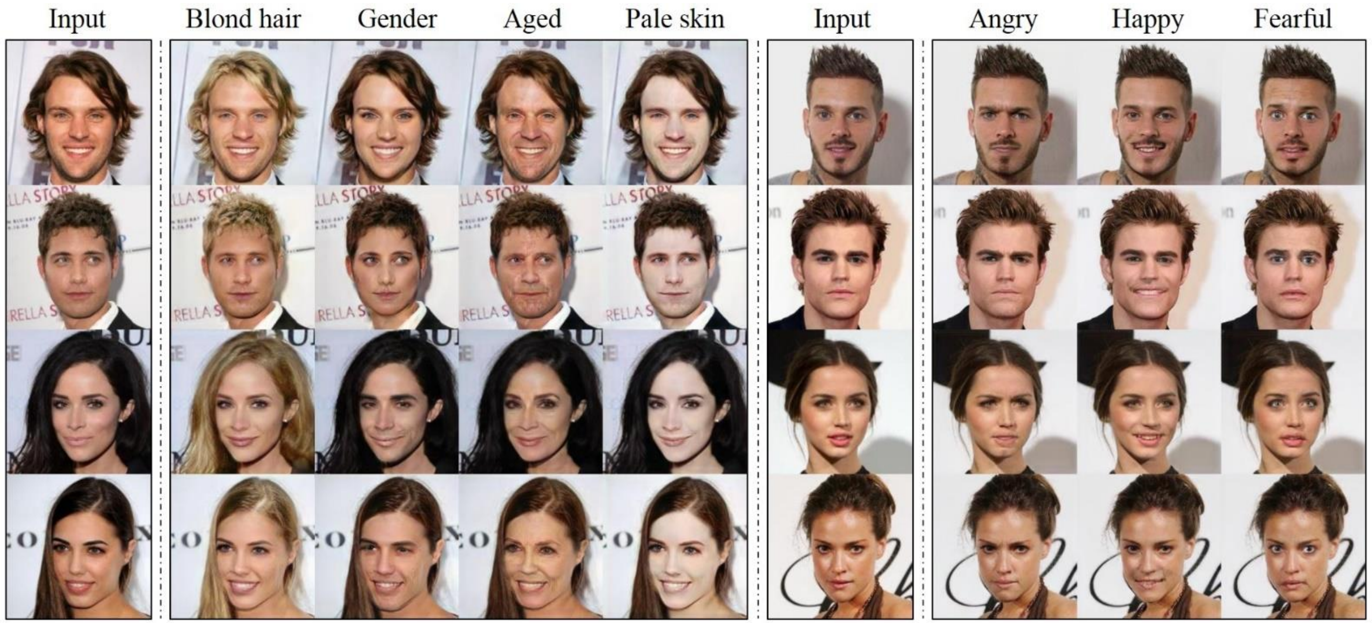
Generative Adversarial Networks (for images)

- If the discriminator is able to tell which image is fake correctly, the parameters, θ_g of the generator are updated so that the image produced next time is more like those from the data, that is, p_g becomes more like p_{data}.
- Eventually, the generator learns to generate images which are very similar to those from the data, and we retrieve these images which are the newly created samples.



An Example…

Results of StarGan on the CelebA dataset.

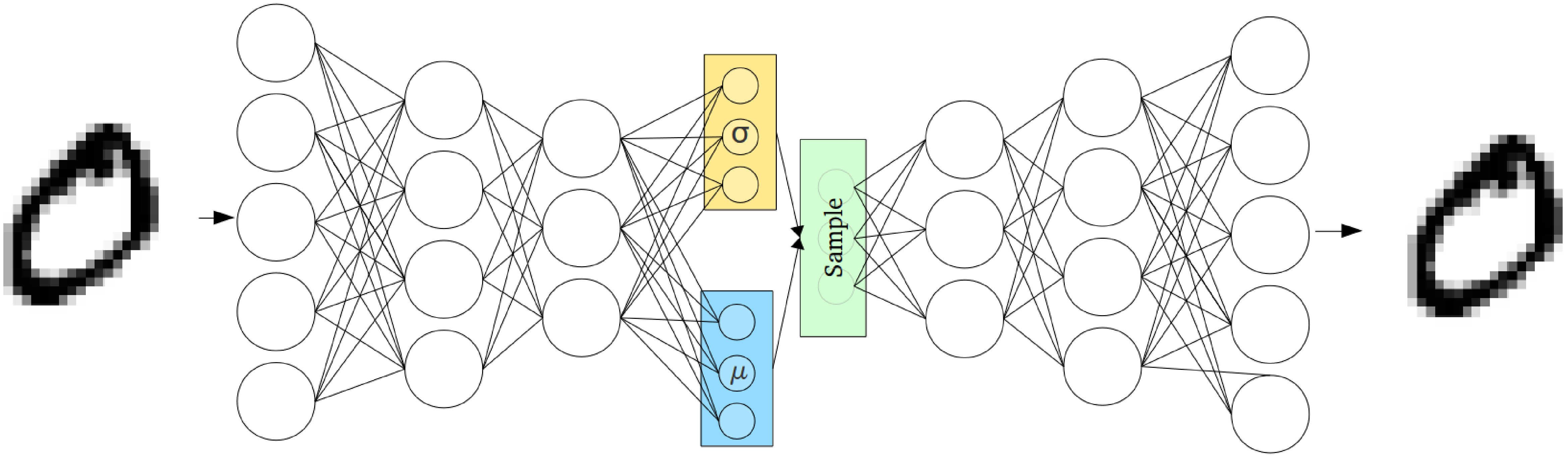


Value Function

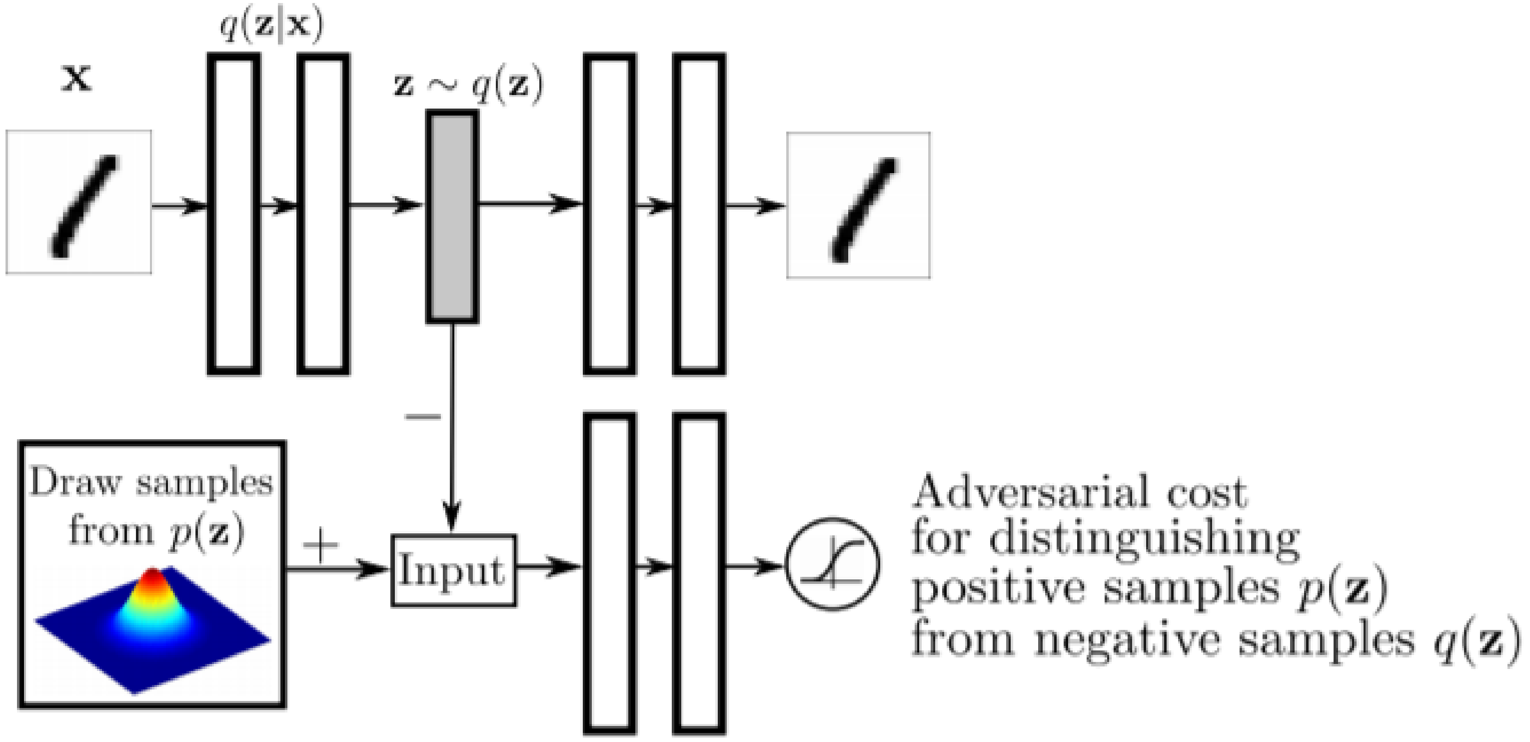
- Represents the loss function between of the GAN.
- Represents a two-player minimax game between the Generator (G) and the Discriminator (D)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

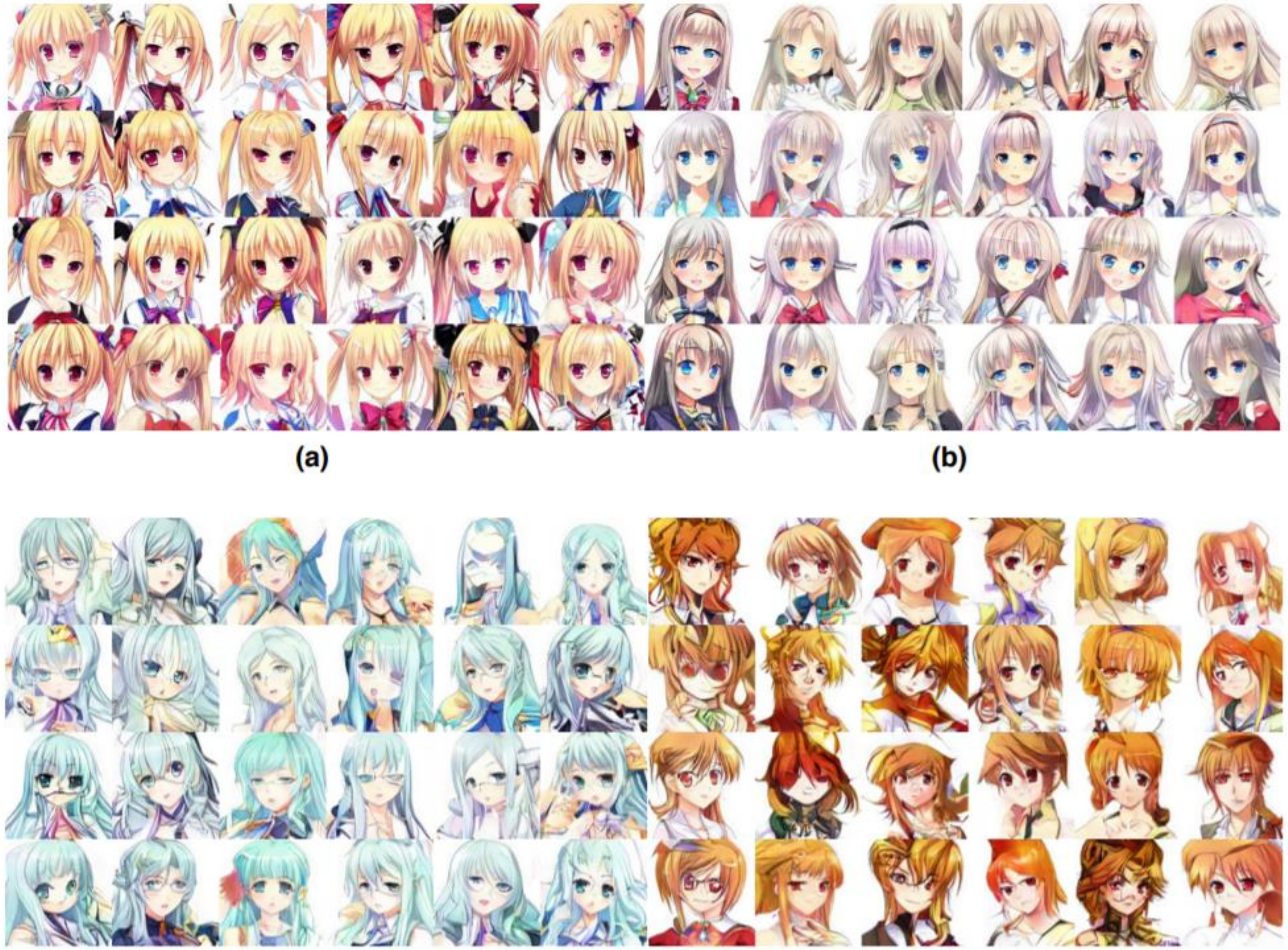
Variational Autoencoders



Adversarial Autoencoders



Conditional Generation



(a) (b) (c) (d)