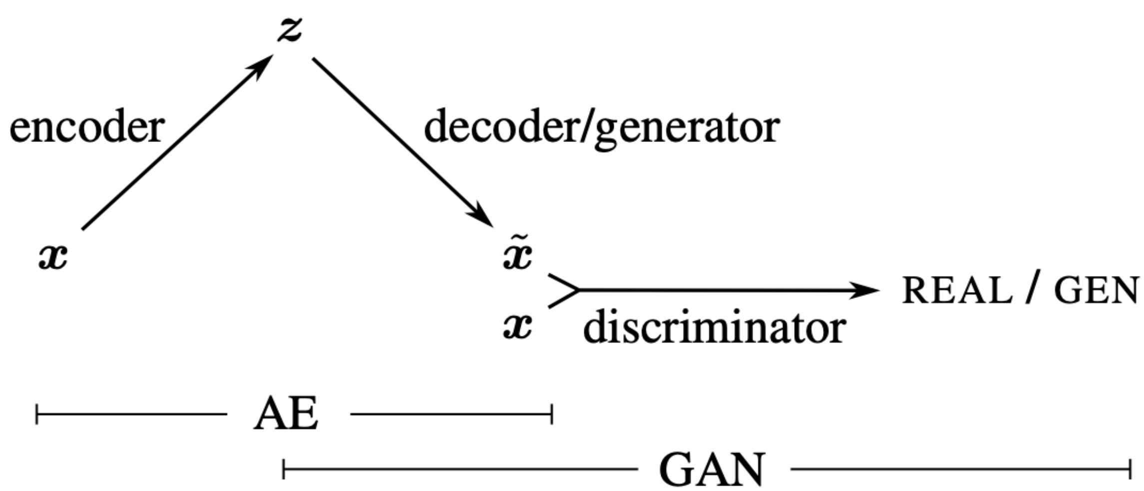
Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling

3D - VAE - GAN

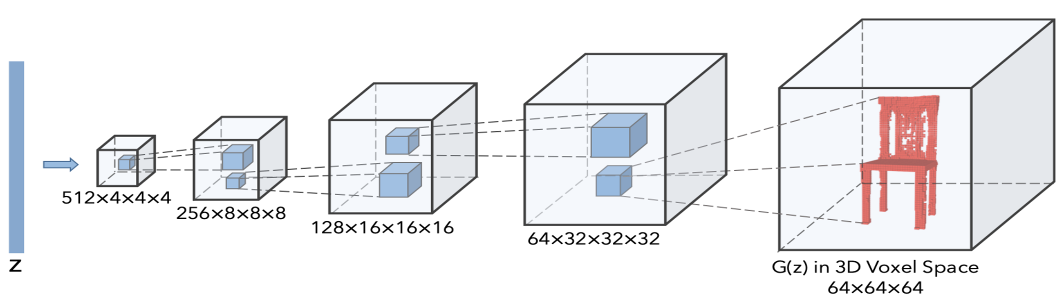
Report



Dataset: IKEA Dataset, SUN Database
Pytorch Code: https://github.com/rimchang/3DGAN-Pytorch
Paper: https://arxiv.org/pdf/1610.07584.pdf
Key points to note:

Encoder

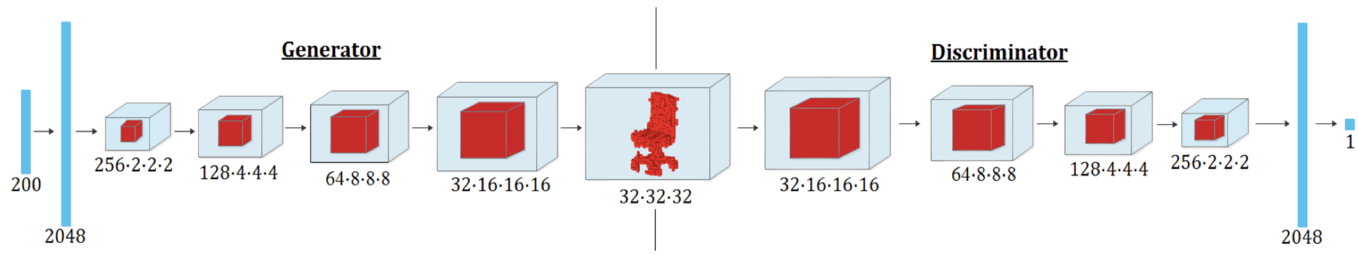
- The encoder is the part where we convert the image to a 200 D vector which is our latent space
- To do this we use 5 convolution layers with BN and ReLU in between



Generator

- The generator uses the latent space vector to generate the 3D model
- It contains 5 ConvTranspose3D layers with BN and ReLU in between the layers

Discriminator



- The discriminator takes the 3D Volume as input and predicts whether the 3D object is real or fake(generated model)
- The architecture of the Discriminator is the mirror of the generator model except there is a sigmoid layer attached at the end

Loss Function

$$L = L_{\text{3D-GAN}} + \alpha_1 L_{\text{KL}} + \alpha_2 L_{\text{recon}}$$

$$\begin{aligned} L_{\text{3D-GAN}} &= \log D(x) + \log(1 - D(G(z))) \\ L_{\text{KL}} &= D_{\text{KL}}(q(z|y) \mid\mid p(z)), \\ L_{\text{recon}} &= ||G(E(y)) - x||_2, \end{aligned}$$

- x is a 3D shape from the training set, y is its corresponding 2D image, and q(z|y) is the variational distribution of the latent representation z
- The loss function consists of three parts: an object reconstruction loss L_{Recon}, a cross-entropy loss $L_{3D\text{-}GAN}$ for 3D-GAN, and a KL divergence loss L_{KL} to restrict the distribution of the output of the encoder
- The Kullback-Leibler Divergence score, or KL divergence score, quantifies how much one probability distribution differs from another probability distribution
- So we use the KL Divergence score so that we can bring the q(z|y) as close to p(z). Basically, we want q(z|y) to represent a Gaussian distribution.