# Teaching Notes: StackGAN

**Prepared by: Abhijit Singh Jowhari**

## Introduction

In these notes, we will discuss the key concepts and mechanisms behind the StackGAN model, which generates high-resolution images from text descriptions using a two-stage process. The focus will be on understanding the mathematical formulations, model architecture, and training procedures.

## 1 Stage-I GAN

### 1.1 Objective

The goal of Stage-I GAN is to generate a low-resolution image that captures the rough shape and correct colors of the object described by the text.

### 1.2 Text Embedding

- Let $\phi_t$ be the text embedding of the given description. - Gaussian conditioning variables $\hat{c}_0$ are sampled from $\mathcal{N}(\mu_0(\phi_t), \Sigma_0(\phi_t))$ to capture the variations in the meaning of $\phi_t$.

### 1.3 Training Process

- Conditioned on $\hat{c}_0$ and random variable $z$, Stage-I GAN trains the discriminator $D_0$ and the generator $G_0$. - The objective functions for the discriminator and generator are:

$$L_{D_0} = \mathbb{E}_{(I_0,t)\sim p_{\text{data}}}[\log D_0(I_0, \phi_t)] + \mathbb{E}_{z\sim p_z, t\sim p_{\text{data}}}[\log(1 - D_0(G_0(z, \hat{c}_0), \phi_t))], \tag{1}$$

$$L_{G_0} = \mathbb{E}_{z\sim p_z, t\sim p_{\text{data}}}[-\log D_0(G_0(z, \hat{c}_0), \phi_t)] + \lambda D_{\text{KL}}(\mathcal{N}(\mu_0(\phi_t), \Sigma_0(\phi_t)) \parallel \mathcal{N}(0, I)), \tag{2}$$

- $\lambda$ is a regularization parameter, set to 1 for all experiments.

### 1.4 Reparameterization Trick

- Using the reparameterization trick from [1], $\mu_0(\phi_t)$ and $\Sigma_0(\phi_t)$ are learned jointly with the network.

### 1.5 Text Encoder

- Follow the approach of Reed et al. [2] to pre-train a text encoder that maps text descriptions to the common feature space of images.

### 1.6 Model Architecture

#### 1.6.1 Generator $G_0$

- Text embedding $\phi_t$ is fed into a fully connected layer to generate $\mu_0$ and $\sigma_0$. - Conditioning vector $\hat{c}_0$ is computed by $\hat{c}_0 = \mu_0 + \sigma_0 \odot \epsilon$ where $\epsilon \sim \mathcal{N}(0, I)$. - $\hat{c}_0$ is concatenated with a noise vector to generate an image through up-sampling blocks.

#### 1.6.2 Discriminator $D_0$

- Text embedding $\phi_t$ is compressed and spatially replicated. - Image is down-sampled and concatenated with the text tensor. - The combined tensor is fed into a 1x1 convolutional layer and a fully connected layer to produce the decision score.

## 2 Stage-II GAN

### 2.1 Objective

The goal of Stage-II GAN is to generate high-resolution images by refining the low-resolution images from Stage-I, adding details and correcting defects.

### 2.2 Training Process

- Conditioned on low-resolution results $s_0 = G_0(z, \hat{c}_0)$ and Gaussian latent variables $\hat{c}$. - The objective functions for the discriminator and generator in Stage-II GAN are:

$$L_D = \mathbb{E}_{(I,t)\sim p_{\text{data}}}[\log D(I, \phi_t)] + \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{\text{data}}}[\log(1 - D(G(s_0, \hat{c}), \phi_t))], \quad (3)$$

$$L_G = \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{\text{data}}}[-\log D(G(s_0, \hat{c}), \phi_t)] + \lambda D_{\text{KL}}(\mathcal{N}(\mu(\phi_t), \Sigma(\phi_t)) \parallel \mathcal{N}(0, I)), \quad (4)$$

### 2.3 Differences from Stage-I

- Random noise $z$ is not used in this stage. - Gaussian conditioning variables $\hat{c}$ and $\hat{c}_0$ share the same pre-trained text encoder but have different fully connected layers for generating means and standard deviations.

### 2.4 Model Architecture

#### 2.4.1 Generator

- Designed as an encoder-decoder network with residual blocks [3]. - Text embedding $\phi_t$ generates the $N_g$ dimensional text conditioning vector $\hat{c}$. - Stage-I result $s_0$ is fed into down-sampling blocks. - Image features and text features are concatenated, processed by residual blocks, and up-sampled to generate high-resolution images.

#### 2.4.2 Discriminator

- Similar to Stage-I discriminator with additional down-sampling blocks for larger image size. - Uses matching-aware discriminator proposed by Reed et al. [2] to enforce better alignment between image and text.

## Conclusion

These notes provide an overview of the StackGAN model, emphasizing its two-stage process for generating high-resolution images from text descriptions. The model's architecture, training objectives, and the role of text embeddings are key points to understand its functionality and performance.

## References


[1] Kingma, D.P., Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114.

[2] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H. (2016). Generative Adversarial Text to Image Synthesis. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*.

[3] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*.