StoryGAN: A Sequential Conditional GAN for Story Visualization

1. Model Architecture

StoryGAN is a sequential conditional GAN model designed to generate a coherent sequence of images from a multi-sentence story. The architecture ensures both local (sentence-level) and global (story-level) consistency. It comprises the following key components:

- **Story Encoder**: Encodes the entire story S into a latent vector h_0 using a stochastic process:

$$h_0 = \mu(S) + \sigma(S) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

  This vector initializes the hidden state of the Context Encoder.

- **Context Encoder**: A deep RNN composed of:
  - **GRU Layer**: Processes the current sentence s_t and random noise ϵ_t to produce an intermediate vector i_t.
  - **Text2Gist Cell**: Combines i_t with the previous context h_{t-1} to output the "gist" vector o_t, using dynamic filtering:

$$o_t = \text{Filter}(i_t) * h_t$$

    where Filter transforms i_t into a learned 1D convolutional filter.

- **Image Generator**: Uses o_t to generate the image $\hat{x}_t$ via convolutional layers and upsampling.

- **Discriminators**:
  - **Image Discriminator** D_I: Evaluates whether an image matches its corresponding sentence and initial context.
  - **Story Discriminator** D_S: Assesses the global coherence between the image sequence and the full story using element-wise feature multiplication:

$$D_S = \sigma(w^\top (E_{\text{img}}(X) \odot E_{\text{text}}(S)) + b)$$

2. Detailed Network Architecture

2.1 Story Encoder

Layer	Operation	Input Shape	Output Shape
1	Linear + BN + ReLU	$128 \times T$	128
2	Sampling from $\mathcal{N}(\mu(S), \sigma^2(S))$	128	128

2.2 Context Encoder

Input: Sentence vector s_t (128) + noise vector ϵ_t (dim depends on config)

Layer	Operation	Input Shape	Output Shape
1	Linear + BN + ReLU	$128 + \text{noise dim}$	128
2	GRU Layer	128	128
3	Text2Gist Cell	$(i_t : 128, h_{t-1} : 128)$	$o_t : 128$

Text2Gist performs:

- GRU-like update of hidden state h_t
- Computes $o_t = \text{Filter}(i_t) * h_t$ (filter has size $[C_{\text{out}}, 1, 1, 128]$)

2.3 Filter Network (Inside Text2Gist)

Layer	Operation	Input	Output Shape
1	Linear + BN + Tanh	128	1024
2	Reshape	1024	$[16, 1, 1, 64]$

2.4 Image Generator

Input: Gist vector o_t (128)

Layer	Operation	Output Shape	Notes
1	Conv2D (3x3, 512 channels) + BN + ReLU	$[512, 4, 4]$	
2	Upsample $\times 2$	$[512, 8, 8]$	
3	Conv2D (3x3, 256) + BN + ReLU	$[256, 8, 8]$	
4	Upsample $\times 2$	$[256, 16, 16]$	
5	Conv2D (3x3, 128) + BN + ReLU	$[128, 16, 16]$	
6	Upsample $\times 2$	$[128, 32, 32]$	
7	Conv2D (3x3, 64) + BN + ReLU	$[64, 32, 32]$	
8	Upsample $\times 2$	$[64, 64, 64]$	
9	Conv2D (3x3, 3) + Tanh	$[3, 64, 64]$	Output RGB image

2.5 Image Discriminator

Input: Image x_t, sentence vector s_t, story context h_0 (all concatenated)

Layer	Operation	Output Shape	Notes
1	Conv2D (4x4, 64) + BN + LeakyReLU	$[64, 32, 32]$	
2	Conv2D (4x4, 128) + BN + LeakyReLU	$[128, 16, 16]$	
3	Conv2D (4x4, 256) + BN + LeakyReLU	$[256, 8, 8]$	
4	Conv2D (4x4, 512) + BN + LeakyReLU	$[512, 4, 4]$	
5	Conv2D (3x3, 512) + BN + LeakyReLU	$[512, 4, 4]$	* Combines conditioning input
6	Conv2D (4x4, 1) + Sigmoid	$[1, 1, 1]$	Final score

2.6 Story Discriminator

2.6.1 Image Encoder

Layer	Operation	Output Shape
1–4	Conv2D Layers with BN + LeakyReLU	downsample to $[512, H, W]$
5	Conv2D (4x4, 32) + BN	$[32, 1, 1]$
6	Reshape and Concatenate over T	$[1, 32 \times 4 \times T]$

2.6.2 Text Encoder

Layer	Operation	Input Shape	Output Shape
1	Linear + BN	$128 \times T$	$32 \times 4 \times T$

2.6.3 Final Layer

$$D_S = \sigma(w^\top (E_{\text{img}}(X) \odot E_{\text{text}}(S)) + b)$$

3. Output

For a story of T sentences, the model produces an image sequence:

$$\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T], \quad \hat{x}_t \in \mathbb{R}^{3\times 64\times 64}$$

3. Loss Functions

The total loss for StoryGAN is:

$$\min_{\theta} \max_{\psi_I, \psi_S} \alpha \mathcal{L}_{\text{Image}} + \beta \mathcal{L}_{\text{Story}} + \mathcal{L}_{\text{KL}}$$

- **KL Divergence Loss**:

$$\mathcal{L}_{\text{KL}} = \text{KL}\left(\mathcal{N}(\mu(S), \text{diag}(\sigma^2(S))) \| \mathcal{N}(0, I)\right)$$

- **Image Discriminator Loss**:

$$\mathcal{L}_{\text{Image}} = \sum_{t=1}^{T} [\log D_I(x_t, s_t, h_0) + \log(1 - D_I(\hat{x}_t, s_t, h_0))]$$

- **Story Discriminator Loss**:

$$\mathcal{L}_{\text{Story}} = \log D_S(X, S) + \log(1 - D_S(\hat{X}, S))$$

Purpose of each loss:

- $\mathcal{L}_{\textbf{KL}}$ ensures a smooth latent space for story embeddings.
- $\mathcal{L}_{\textbf{Image}}$ promotes local consistency between each sentence and its image.
- $\mathcal{L}_{\textbf{Story}}$ enforces global coherence of the image sequence with the entire story.

4. Training Methodology

Inputs:

- A story $S = [s_1, s_2, \ldots, s_T]$
- Ground-truth images $X = [x_1, x_2, \ldots, x_T]$

Procedure:

1. **Sentence Encoding**: Each s_t is encoded into a 128-dimensional vector using a pretrained sentence encoder.
2. **Story Initialization**: Story encoder produces h_0, used to initialize the Context Encoder.
3. **Sequential Generation** (for each t):

$$i_t = \text{GRU}(s_t, \epsilon_t), \quad o_t = \text{Text2Gist}(i_t, h_{t-1}), \quad \hat{x}_t = \text{Generator}(o_t)$$

4. **Discriminator Updates**:
   - D_I is updated using real/fake sentence-image pairs.
   - D_S is updated using real/fake story-image sequences.
5. **Optimization**: Use the Adam optimizer; alternate updates to generator and discriminators with possibly different mini-batches.

5. Key Innovations

- **Text2Gist Cell**: Combines local sentence features and global story context using dynamic filtering.
- **Two-Level Discriminators**: Simultaneously enforce sentence-image and story-image sequence consistency.
- **Stochastic Story Encoding**: Improves robustness and diversity in the initial context representation.
- **Effective training schedule**: Alternating discriminator/generator updates and using KL regularization enables stable and effective training.