1 StoryGAN: Architecture, Loss Function, and Training Method

1.1 Model Architecture

StoryGAN is a sequential conditional Generative Adversarial Network (GAN) designed to generate a sequence of images that visualize a multi sentence story. The model is carefully constructed to ensure both local context (each image matches its corresponding sentence) and global context (the entire image sequence is coherent with the whole story). The architecture consists of the following key components:

- **Story Encoder:** The story encoder processes the entire story $S = [s_1, s_2, ..., s_T]$ and encodes it into a latent vector h_0 :
$$h_0 = \mu(S) + \sigma(S) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$
Here, $\mu(S)$ and $\sigma(S)$ are functions (typically neural networks) that output the mean and standard deviation for the story embedding, and $\odot$ denotes element wise multiplication. This encoding introduces diversity and robustness in the initial context for the story.

- **Context Encoder:** The context encoder is a deep recurrent neural network (RNN) that maintains the evolving context of the story as images are generated sequentially. It consists of:
  - **GRU Layer:** At each time step t, the GRU processes the current sentence embedding s_t and a random noise vector ϵ_t to produce an intermediate vector i_t.
  - **Text2Gist Cell:** This novel cell combines the intermediate vector i_t with the previous context h_{t-1} to produce a "gist" vector o_t:
$$o_t = \text{Filter}(i_t) * h_{t-1}$$
The Filter(i_t) operation transforms i_t into a learned 1D convolutional filter, which is then applied to h_{t-1}, allowing dynamic integration of new sentence information with the story context.

- **Image Generator:** The generator takes the gist vector o_t at each time step and produces the corresponding image $\hat{x}_t$ using a series of convolutional and upsampling layers. This process ensures that each generated image is constrained on both the current sentence and the evolving story context.

- **Discriminators:**
  - **Image Discriminator (D_I):** This discriminator evaluates whether a generated image $\hat{x}_t$ matches its corresponding sentence s_t and the initial story context h_0. It encourages the generator to produce images that are locally consistent with the input sentence.
  - **Story Discriminator (D_S):** This discriminator assesses the global relation of the entire generated image sequence $\hat{X} = [\hat{x}_1, ..., \hat{x}_T]$ with the full story S. It uses element wise feature multiplication between image and text embeddings:
$$D_S = \sigma(w^\top (E_{\text{img}}(X) \odot E_{\text{text}}(S)) + b)$$
where E_{img} and E_{text} are encoders for the image sequence and story, respectively, and σ is the sigmoid function.

1.2 Loss Functions

The training objective for StoryGAN combines several loss terms to enforce both local and global consistency:

- **KL Divergence Loss:** Regularizes the latent space of the story encoder, encouraging the distribution of story embeddings to be close to a standard normal distribution:
$$\mathcal{L}_{\text{KL}} = KL\left(\mathcal{N}(\mu(S), \text{diag}(\sigma^2(S))) \| \mathcal{N}(0, I)\right)$$

- **Image Discriminator Loss:** Encourages each generated image to match its sentence and context:
$$\mathcal{L}_{\text{Image}} = \sum_{t=1}^{T} [\log D_I(x_t, s_t, h_0) + \log(1 - D_I(\hat{x}_t, s_t, h_0))]$$
where x_t is the real image and $\hat{x}_t$ is the generated image for sentence s_t.

- **Story Discriminator Loss:** Encourages the entire image sequence to be coherent with the story:

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

  where X is the real image sequence and $\hat{X}$ is the generated sequence.

- **Total Loss:** The overall objective is:

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

  where α and β are hyperparameters balancing the contributions of each loss.

1.3 Training Methodology

The training process for StoryGAN proceeds as follows:

1. **Input Preparation:** Each story $S = [s_1, s_2, ..., s_T]$ and its corresponding ground-truth image sequence $X = [x_1, x_2, ..., x_T]$ are provided as input.
2. **Sentence Encoding:** Each sentence s_t is encoded into a fixed-length vector (e.g., 128-dimensional) using a pretrained sentence encoder.
3. **Story Initialization:** The story encoder produces the initial context vector h_0, which initializes the context encoder.
4. **Sequential Generation:** For each time step t:

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

5. **Discriminator Updates:**
   - The image discriminator D_I is updated using real and fake sentence-image pairs.
   - The story discriminator D_S is updated using real and fake story-image sequences.
6. **Optimization:** The generator and discriminators are optimized alternately using the Adam optimizer, possibly with different mini-batches for stability.
7. **Regularization:** KL divergence regularization is applied to the story encoder to ensure a smooth latent space and improve generalization.