StyleAttnGAN: Style-Aware Attention Generative Adversarial Network for Text-to-Image Synthesis

Deep Fusion

***Abstract*—StyleAttnGAN (Style-Aware Attention Generative Adversarial Network) is a significant improvement over earlier models in the domain of text-to-image synthesis, such as StackGAN, ProGAN, and AttnGAN. The main goal of StyleAttnGAN is to generate realistic images from textual descriptions while preserving both fine-grained details and global image style. This paper introduces the architecture, mathematical transformations, and loss functions of StyleAttnGAN, demonstrating its advancements over previous models in generating high-quality images that accurately reflect textual descriptions and specified styles.**

I. StackGAN: Mathematical Transformations

A. Text Embedding and Generator Input

In StackGAN, the generator uses a text embedding vector as part of the input. The text is first processed into a fixed-length vector using a pre-trained text encoder (often an RNN, LSTM, or GRU). Denote this as:

$$\mathbf{e}_t = \text{TextEncoder}(T) \tag{1}$$

Where $\mathbf{e}_t$ is the text embedding vector, and T represents the input text description.

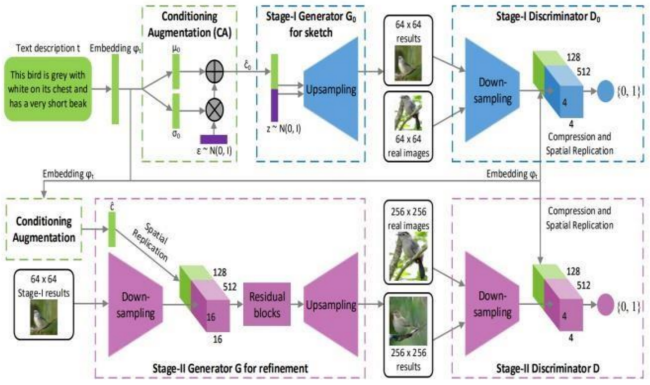

Fig. 1. Overview of the two-stage GAN structure with conditioning augmentation for text-to-image synthesis.

B. Generator Architecture (Stage I)

The first stage of StackGAN generates a coarse image from the text embedding vector $\mathbf{e}_t$ and random noise $\mathbf{z}_1$ sampled from a Gaussian distribution $p(\mathbf{z}_1)$:

$$I_1 = G_1(\mathbf{e}_t, \mathbf{z}_1) \tag{2}$$

Where G_1 is the generator for the coarse image and I_1 is the resulting low-resolution image.

C. Generator Architecture (Stage II)

In the second stage, the coarse image I_1 is refined with additional text information. The input to the second-stage generator is the concatenation of I_1 and $\mathbf{e}_t$, along with new random noise $\mathbf{z}_2$ sampled from $p(\mathbf{z}_2)$:

$$I_2 = G_2(\mathbf{e}_t, I_1, \mathbf{z}_2) \tag{3}$$

Where G_2 is the second-stage generator, and I_2 is the final high-resolution image.

D. Discriminator Losses

For both stages, the discriminator D evaluates the authenticity of the generated images by distinguishing between real and fake images:

$$L_{\text{dis}} = -\mathbb{E}_{I \sim p_{\text{data}}}[\log D(I)] - \mathbb{E}_{I_2 \sim G_2(\mathbf{e}_t, I_1, \mathbf{z}_2)}[\log(1 - D(I_2))] \tag{4}$$

Where p_{data} is the real image distribution, and $D(I_2)$ is the discriminator's output for the generated image I_2.

E. Generator Losses

The generator's loss function involves two components: the adversarial loss and the image reconstruction loss. The generator aims to minimize the discriminator's ability to distinguish fake images from real ones:

$$L_{\text{gen}} = \mathbb{E}_{I_2 \sim G_2(\mathbf{e}_t, I_1, \mathbf{z}_2)}[\log(1 - D(I_2))] + \lambda \cdot L_{\text{recon}}(I_2, I_{\text{real}}) \tag{5}$$

Where λ is a hyperparameter controlling the importance of the reconstruction loss L_{recon}, which measures the similarity between the generated image and the real image.

II. ProGAN: Mathematical Transformations

A. Progressive Training

ProGAN introduces progressive training, where both the generator and discriminator start with low-resolution images and gradually increase the image resolution during training. The progressive growth of the network is mathematically represented by scaling the generator's output from low to high resolution.

Let the resolution at each stage be denoted as r_n for the n-th stage. Initially, for stage 1 (low resolution), the generator produces $G_1(\mathbf{z})$, and the image resolution r_1 is small. As training progresses, higher resolutions are introduced:

$$I_n = G_n(\mathbf{z}_n) \tag{6}$$

Where I_n is the image at resolution r_n.

B. Loss Function

The loss functions in ProGAN are similar to those in traditional GANs but are adapted to handle progressively increasing resolutions:

$$L_{\text{dis}} = -\mathbb{E}_{I\sim p_{\text{data}}}[\log D(I)] - \mathbb{E}_{I_n\sim G_n(\mathbf{z}_n)}[\log(1-D(I_n))] \quad (7)$$

$$L_{\text{gen}} = \mathbb{E}_{I_n\sim G_n(\mathbf{z}_n)}[\log(1-D(I_n))] \quad (8)$$

ProGAN's key transformation here is the progressive training of both the generator and discriminator, scaling from low to high resolutions.

III. AttnGAN: Attention Mechanism

A. Attention Mechanism

In AttnGAN, the generator learns to focus on specific parts of the text description when generating corresponding parts of the image. Given a text embedding $\mathbf{e}_t$ and a noise vector $\mathbf{z}$, the generator computes an attention map α_i for each word w_i in the description:

$$\alpha_i = \text{Attention}(w_i, \mathbf{e}_t) \quad (9)$$

This attention map is then used to weight different parts of the image being generated, focusing more on the areas that correspond to the text descriptions.

B. Multi-Stage Generation

AttnGAN uses multiple stages to refine the image. At each stage n, the image I_n is generated based on the weighted text information from the attention mechanism:

$$I_n = G_n(\mathbf{e}_t, \alpha_n, \mathbf{z}) \quad (10)$$

Where α_n is the attention map at stage n, guiding the generator to focus on relevant regions.

C. Loss Function

AttnGAN introduces a multi-scale adversarial loss, where each stage contributes to the final image loss. The discriminator evaluates images at each stage:

$$L_{\text{dis}} = -\sum_n \mathbb{E}_{I_n\sim G_n(\mathbf{e}_t,\alpha_n,\mathbf{z})}[\log D(I_n)] \quad (11)$$

The generator's loss is the sum of adversarial losses at each stage:

$$L_{\text{gen}} = \sum_n \mathbb{E}_{I_n\sim G_n(\mathbf{e}_t,\alpha_n,\mathbf{z})}[\log(1-D(I_n))] \quad (12)$$

IV. StyleAttnGAN: A Detailed Explanation

A. *improvements on StyleAttnGAN over other*

StyleAttnGAN (Style-Aware Attention Generative Adversarial Network) is a significant improvement over earlier models like StackGAN, ProGAN, and AttnGAN in the domain of text-to-image synthesis. The main goal of StyleAttnGAN is to generate realistic images from textual descriptions while preserving both fine-grained details and global image style. It introduces a style-aware attention mechanism that guides the generation process, allowing for better control over both the fine details and the overall style of the image.While models like AttnGAN focus on the text-to-image alignment using attention mechanisms, StyleAttnGAN enhances this by adding a style vector that captures the global aesthetic of the image (such as texture, color palette, and overall composition). This enables StyleAttnGAN to generate more visually cohesive and aesthetically pleasing images while ensuring that they accurately reflect the textual descriptions.

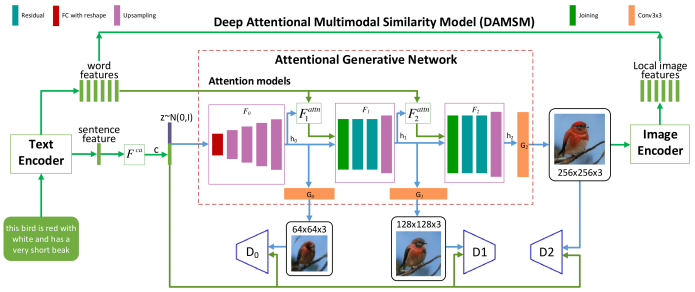

Fig. 2. Overview of the two-stage GAN structure with conditioning augmentation for text-to-image synthesis.

B. Key Components of StyleAttnGAN

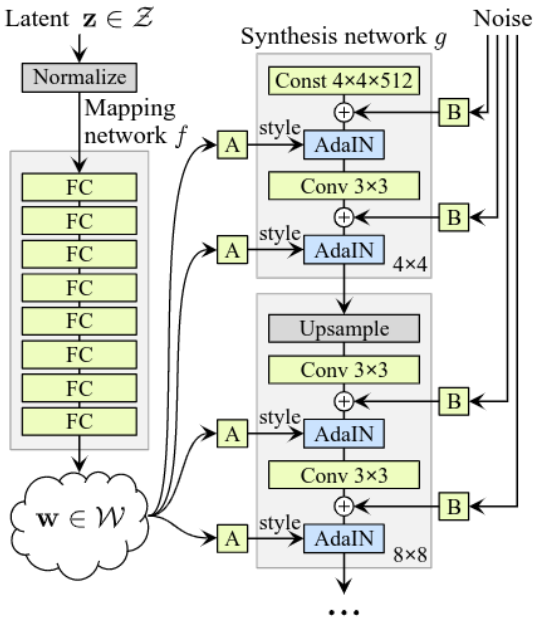

Fig. 3. Overview of the two-stage GAN structure with conditioning augmentation for text-to-image synthesis.

1) Text Embedding: StyleAttnGAN, like previous models (e.g., StackGAN and AttnGAN), relies on an embedding of the input text. The text embedding is usually obtained using a pre-trained text encoder (e.g., an RNN or LSTM), which processes the input text T into a vector representation $\mathbf{e}_t$:

$$\mathbf{e}_t = \text{TextEncoder}(T) \quad (13)$$

This embedding $\mathbf{e}_t$ contains semantic information about the text description and is used as an input to guide the image generation process.

2) Style-Aware Attention Mechanism: The key innovation in StyleAttnGAN is the style-aware attention mechanism. Unlike traditional attention mechanisms (like in AttnGAN), which focus on specific words or phrases in the text description, StyleAttnGAN uses a style vector $\mathbf{s}$ that encapsulates high-level, global image features. These include attributes such as the color scheme, texture, and overall aesthetic of the image. The style vector $\mathbf{s}$ is learned during training and helps the generator focus not only on the fine-grained details of the text but also on ensuring the overall style coherence of the generated image.

3) Generator with Style-Aware Attention: The generator in StyleAttnGAN produces images using both the text embedding $\mathbf{e}_t$ and the style vector $\mathbf{s}$. The generator applies the style vector to modulate the attention mechanism, allowing it to focus on specific areas of the image in a way that aligns with both the fine details described in the text and the desired global style.

The image generation process can be described as:

$$I_s = G_s(\mathbf{e}_t, \alpha_s, \mathbf{z}) \tag{14}$$

Where:

- I_s is the final image generated at the s-th stage.
- $\mathbf{z}$ is the random noise vector that provides variation.
- α_s is the style-aware attention map, which is calculated based on the style vector $\mathbf{s}$ and the text embedding $\mathbf{e}_t$.

4) Attention Maps: The attention maps in StyleAttnGAN are influenced by both the text and style vectors. In particular, the attention mechanism assigns different importance to different regions of the image depending on the relevance to the text description and the style characteristics. The attention mechanism can be represented as:

$$\alpha_s = \text{StyleAttention}(\mathbf{e}_t, \mathbf{s}) \tag{15}$$

This attention map α_s guides the generator in producing more realistic and consistent images by focusing on key areas of the image (local attention) while respecting the overall style (global attention).

V. Loss Function in StyleAttnGAN

A. Discriminator Loss

The discriminator loss function evaluates how well the discriminator can distinguish between real and generated images. This is the typical adversarial loss used in GANs:

$$L_{\text{dis}} = -\mathbb{E}_{I_s \sim p_{\text{data}}}[\log D(I_s)] - \mathbb{E}_{I_s \sim G_s(\mathbf{e}_t, \alpha_s, \mathbf{z})}[\log(1 - D(I_s))] \tag{16}$$

Where p_{data} is the distribution of real images, and $D(I_s)$ is the discriminator's classification of the generated image I_s.

B. Generator Loss

The generator's goal is to minimize the loss by fooling the discriminator. In StyleAttnGAN, the generator loss includes the adversarial loss as well as a style loss term. The style loss ensures that the generated image has a style that is consistent with the input style vector $\mathbf{s}$.

$$L_{\text{gen}} = \mathbb{E}_{I_s \sim G_s(\mathbf{e}_t, \alpha_s, \mathbf{z})}[\log(1 - D(I_s))] + \lambda \cdot L_{\text{style}}(I_s, \mathbf{s}) \tag{17}$$

Where λ is a hyperparameter that controls the relative importance of the style loss L_{style}, and $L_{\text{style}}(I_s, \mathbf{s})$ measures the difference between the global style of the generated image I_s and the target style vector $\mathbf{s}$.

C. Style Loss

The style loss term L_{style} is usually computed by comparing the feature representations of the generated image and the target style across different layers of a pre-trained convolutional neural network (such as VGG). This ensures that the global visual aesthetics (texture, color patterns, and overall composition) of the generated image align with the desired style.

VI. Summary of StyleAttnGAN's Strengths

- **Global Style Control**: By incorporating the style vector $\mathbf{s}$, StyleAttnGAN can generate images with a specific aesthetic or texture while maintaining coherence across the entire image.
- **Fine-Grained Detail Generation**: The style-aware attention mechanism allows the model to focus on specific parts of the image while aligning with the textual description, improving the fine-grained detail of generated images.
- **Text-Image Alignment**: StyleAttnGAN improves text-to-image alignment by learning to associate specific parts of the image with corresponding parts of the text description, using attention maps that are conditioned on both the text and style information.
- **Enhanced Robustness**: The combination of local attention and global style control makes StyleAttnGAN more robust than previous models in generating realistic images that adhere to both the text description and the desired visual style.

VII. Future Directions and Possible Enhancements

Possible future directions for StyleAttnGAN include:

- **Better Handling of Ambiguity in Text**: StyleAttnGAN could be enhanced to handle more ambiguous or incomplete text descriptions, improving its ability to generate plausible images even with limited or unclear textual input.
- **Style Transfer in Real-Time**: Future models could focus on enabling real-time style transfer for generating images with different visual styles dynamically, allowing users to adjust the image style post-generation.
- **Reduction of Computational Complexity**: The attention mechanism, while powerful, is computationally intensive. Future work could focus on reducing the computational cost of generating attention maps without sacrificing image quality.

VIII. Training Process

The training of Style-AttnGAN follows a multi-stage process where both style and content are refined through successive stages:

- **Stage-I Training:** A rough image is generated, capturing basic structure and style.
- **Stage-II and Beyond Training:** Each subsequent stage uses the attention mechanism to focus on specific parts of the text description, adding details relevant to different image regions. The style encoder guides the appearance of each stage, ensuring that the generated image maintains both text-aligned details and stylistic features throughout the refinement process.

This multi-stage approach not only improves image resolution and detail but also allows the model to produce images with high stylistic consistency.

IX. Data Requirements and Preprocessing

Like AttnGAN, Style-AttnGAN relies on datasets with paired images and text descriptions, but it also requires style references if a style transfer is to be applied. Common datasets include:

- **CUB (Caltech-UCSD Birds):** Provides high-quality, fine-grained bird images with descriptive captions, making it suitable for both content and style learning.
- **MS-COCO:** A larger, more diverse dataset that includes various object categories and scenes, suitable for testing the model's ability to generalize to different contexts.

Preprocessing steps include:

- **Text Processing:** Tokenizing and embedding descriptions to capture the semantic features of each word and sentence.
- **Style Preprocessing:** Extracting style features from reference images, if provided, or using predefined style embeddings.

X. Performance and Results

Style-AttnGAN has shown to outperform traditional text-to-image models by:

- **Producing High-Fidelity Images:** The attention mechanism and multi-stage generation contribute to higher detail and resolution.
- **Improving Style Consistency:** The style encoder enables Style-AttnGAN to generate images that are not only faithful to the text but also visually cohesive in terms of style.
- **Enhanced Content-Style Flexibility:** The separation of content and style allows for more flexibility, enabling users to generate images that capture both the essence of the description and the desired aesthetic.

XI. Conclusion

StyleAttnGAN represents a significant advancement in text-to-image synthesis by combining fine-grained text alignment with global style control. The introduction of a style-aware attention mechanism enables the model to generate not only high-resolution images that are semantically accurate but also aesthetically coherent. This makes StyleAttnGAN more robust than previous models like StackGAN, ProGAN, and AttnGAN in generating images that match the given text while maintaining a consistent visual style. As the field of generative modeling continues to advance, StyleAttnGAN sets a strong foundation for future improvements, especially in terms of efficiency and handling complex, ambiguous text inputs.

References

[1] Agarwal, Cheeku. *DEEP-FUSION*. GitHub Repository. Available at: https://github.com/cheekuag/DEEP-FUSION.

[2] Zhang, Han, et al. *StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks*. GitHub Repository. Available at: https://github.com/hanzhanggit/StackGAN.

[3] Karras, Tero, et al. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. arXiv preprint arXiv:1710.10196, 2017. Available at: https://arxiv.org/abs/1710.10196.

[4] Xu, Tao, et al. *AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks*. arXiv preprint arXiv:1711.10485, 2017. Available at: https://arxiv.org/abs/1711.10485.