Generative AI for Text-to-Image Synthesis

Rajiv Chaudhary
IIT HYDERABAD
ai22btech11021@iith.ac.in

Sai Satwik
IIT HYDERABAD
ai22btech11025@iith.ac.in

Siddhesh Gholap
IIT HYDERABAD
ai22btech11007@iith.ac.in

Sudarshan Shivashankar
IIT HYDERABAD
ai22btech11027@iith.ac.in

1. Abstract

Text-to-image synthesis is a rapidly advancing field in generative AI, enabling the creation of realistic images from textual descriptions.

This project explores advancements in text-to-image synthesis using the SEA Attention GAN model, evaluated on the CUB-200 dataset. We conduct a series of experiments to enhance image generation quality by modifying the model architecture, loss functions, and text encoders.

The baseline experiment uses the small version of SEA Attention GAN with adversarial and matching-aware losses. In subsequent experiments, we incorporate a contrastive InfoNCE loss to encourage better alignment between image and text representations, and apply a WGAN-GP framework with a pretrained ResNet-18 discriminator to improve training stability and convergence. Additionally, we replace the traditional RNN-based text encoder with a pretrained CLIP encoder to leverage rich semantic embeddings. The effectiveness of each variation is quantitatively evaluated using the Fréchet Inception Distance (FID), demonstrating the impact of architectural and loss modifications on the quality of generated images.

The base implementation of code used in this work is available at: https://github.com/MingyuJ666/SEAttnGAN.

2. Literature Review

2.1. High-Resolution Image Synthesis with Latent Diffusion Models

The paper **"High-Resolution Image Synthesis with Latent Diffusion Models"** [1] by Rombach et al. (CVPR 2022) introduces **Latent Diffusion Models (LDMs)**, a novel approach to reduce the computational complexity of diffusion models while maintaining high-quality image synthesis. The authors address the high computational demands of traditional diffusion models, which operate directly in pixel space, by proposing a two-stage process that first compresses images into a lower-dimensional latent space and then applies diffusion models in this space.

The key contributions of the paper are:

- **Latent Space Compression**: LDMs use a pre-trained autoencoder to compress high-resolution images into a lower-dimensional latent space. This reduces the computational burden of training and inference while preserving perceptual quality.
- **Efficient Diffusion in Latent Space**: By applying diffusion models in the latent space, LDMs achieve state-of-the-art performance in tasks such as image inpainting, class-conditional image synthesis, and text-to-image generation, while significantly reducing computational costs.
- **Cross-Attention Conditioning**: The authors introduce a flexible conditioning mechanism based on cross-attention, enabling LDMs to handle various conditioning inputs such as text, semantic layouts, and class labels. This allows for high-resolution image synthesis with fine-grained control over the generated content.

2.2. DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis

The authors address three key limitations of existing models:

1. Entanglements in stacked architectures
2. Limited supervision for text-image consistency
3. Computational inefficiency of cross-modal attention.

DF-GAN proposes three main innovations:

- **One-Stage Backbone**: A single generator synthesizes high-resolution images directly, avoiding entanglements and improving coherence. It uses hinge loss and residual networks for stable training.
- **Target-Aware Discriminator**: Enhanced with:
  - **Matching-Aware Gradient Penalty (MA-GP)**: Ensures zero gradient for real and text-matching images, promoting smoother convergence.
  - **One-Way Output**: Simplifies the discriminator's out-

put to improve convergence and semantic consistency.

- **Deep Text-Image Fusion Block (DFBlock)**: Deepens text-image fusion using Affine Transformations and ReLU layers, enabling efficient fusion across all image scales without cross-modal attention.

DF-GAN achieves state-of-the-art performance on the **CUB** and **COCO** datasets, with an IS of 5.10 and FID of 14.81 on CUB, and an FID of 19.32 on COCO. It is also computationally efficient, with only 19 million parameters.

2.3. A Simple and Effective Baseline for Attentional Generative Adversarial Networks

The paper **"A Simple and Effective Baseline for Attentional Generative Adversarial Networks"**[2] by Jin et al. (2023) introduces **SEAttnGAN**, an optimized version of the original Attentional Generative Adversarial Network (AttnGAN). This work aims to reduce the complexity and computational demands of AttnGAN while maintaining, or even enhancing, image generation quality.

The key contributions of the paper are:

- **Model Simplification**: The authors streamline AttnGAN's architecture by removing redundant structures and refining the backbone network. This results in a significant reduction in model parameters—from AttnGAN's 230 million to SEAttnGAN's 26.37 million—leading to improved training efficiency without compromising performance.
- **Loss Function Integration**: SEAttnGAN integrates and reconstructs multiple losses from the Deep Attentional Multimodal Similarity Model (DAMSM), enhancing the model's ability to generate images that accurately reflect the input text descriptions.
- **Efficiency Gains**: SEAttnGAN demonstrates a substantial reduction in computational complexity and training time compared to its predecessor, making it more accessible for practical applications.
- **Image Quality**: Empirical results indicate that SEAttnGAN matches or surpasses AttnGAN in image generation quality, particularly on datasets like CUB.

The authors demonstrate that SEAttnGAN can generate high-quality images while being significantly more efficient. The model achieves competitive results with lower computational requirements, making it a viable alternative to traditional attention-based GANs.

2.4. Hierarchical Text-Conditional Image Generation with CLIP Latents

The paper **" Hierarchical Text-Conditional Image Generation with CLIP Latents"** [3] by Aditya Ramesh et al. (2022) introduces a novel approach to text-to-image synthesis using CLIP representations and diffusion models. The authors address three key challenges in existing text-to-image models:

1. Limited diversity due to direct text-to-image mapping.
2. Inefficiencies in conditioning mechanisms.
3. Computational costs in high-resolution image synthesis.

The proposed model, called **unCLIP**, consists of two main components:

- **CLIP Latent Prior:** A diffusion or autoregressive model generates a CLIP image embedding from a given text caption, enabling a more structured and semantically rich image representation.
- **Diffusion Decoder:** A diffusion model conditioned on CLIP image embeddings synthesizes high-quality images while preserving semantic consistency.

The approach offers several advantages:

- **Improved Diversity:** By generating images from learned embeddings rather than direct text conditioning, *unCLIP* increases variation in outputs while maintaining photorealism.
- **Zero-Shot Image Editing:** The model enables text-guided modifications by manipulating CLIP embeddings, allowing changes in style, attributes, and composition.
- **Interpolation and Variations:** The latent space structure allows smooth transitions between images and the creation of controlled variations of a given input.

Experiments on the MS-COCO dataset show that *unCLIP* outperforms previous models like GLIDE and DALL-E in diversity while maintaining competitive photorealism.

However, the model has limitations in attribute binding and fine-grained text rendering.

3. SEA-AttnGAN

3.1. Dataset

The model is trained on the CUB-200-2011 bird dataset, which consists of 11,788 images across 200 bird species. Each image is accompanied by 10 textual descriptions, providing detailed attribute information.

3.2. Model Architecture

The SEA-AttnGAN architecture consists of the following major components:

3.3. Text Encoder

For text encoding, we employ a recurrent neural network (RNN)-based approach:

- A word embedding layer maps tokens to 300-dimensional vectors.
- A bidirectional LSTM encodes the text into word-level and sentence-level embeddings.
- The sentence embedding is derived from the final hidden state of the LSTM, while word embeddings are used for attention-based alignment with image features.

3.4. Clip Encoder

- CLIP is a model that uses two encoders (one for text and one for images) to map both into a shared space, so related images and texts are close together.
- It is trained on a large dataset of image-text pairs using a contrastive loss, which brings correct pairs closer and pushes incorrect pairs apart.
- This approach allows CLIP to understand new images or descriptions without extra training (zero-shot learning), making it highly flexible for various tasks.
- In our project, we used the CLIP text encoder to get strong, general-purpose text features from captions.
- We added an adapter so the CLIP text embeddings matched the size needed by our GAN model.
- Using CLIP improved the alignment between text and generated images, helping our GAN create images that better matched the input descriptions.

3.5. SEA Attention Mechanism

The SEA block consists of two key components:

1. **Word-Level Attention:** This mechanism aligns image features with textual word embeddings through a dot-product-based attention operation. The attended features are concatenated with the image feature map to guide the generation process.
2. **Sentence-Level Affine Modulation:** A pair of multi-layer perceptrons (MLPs) learn scale and shift parameters to modulate intermediate image features based on sentence embeddings:

$$F' = \gamma(s) \cdot F + \beta(s)$$

where F represents the feature map, and $\gamma(s)$, $\beta(s)$ are the learned scale and shift parameters derived from the sentence embedding s.

3.6. Generator

The generator progressively synthesizes images while integrating text embeddings at multiple scales:

- The latent vector z is projected into an $8C \times 4 \times 4$ feature map.
- Four residual blocks refine features through affine modulation using sentence embeddings and progressive upsampling.
- Word-level attention is applied at a 64×64 resolution, followed by concatenation with attended features.
- A 1×1 convolution is used to refine the concatenated feature maps.
- Final upsampling layers generate a high-resolution 256×256 image.
- The output is a 3-channel RGB image generated using a Tanh activation function.

3.7. Discriminator

The discriminator evaluates both the realism of generated images and their alignment with the input text through a series of convolutional and residual layers:

- Convolutional layers progressively downsample the image while extracting hierarchical features.
- Feature maps are reduced from 256×256 to 4×4 through ResidualBlockD modules.
- Instead of spatially broadcasting the sentence embedding, it is concatenated with the final image feature vector.
- The concatenated tensor is passed through a fully connected classifier (`judge_net`) to produce a scalar output indicating the real/fake score.

Formally, the discriminator fusion is computed as:

$$\text{logits} = D(\text{concat}(F(I), S))$$

where $F(I)$ represents the extracted image feature tensor, and S denotes the sentence embedding.

This approach enables the discriminator to assess both the realism of the generated images and their semantic consistency with the input text, thereby ensuring stable adversarial training.

3.8. Loss Functions

The training procedure incorporates a combination of loss terms designed to produce images that are visually realistic, semantically aligned with the text, and diverse across prompts. The following loss components were implemented:

Adversarial Loss

A hinge-based adversarial loss is employed for both the generator and the discriminator. It promotes realism in generated images and discourages mismatches in image-text alignment.

- **Discriminator Loss(Matching-Aware Loss):**
  - Real image–text pairs,
  - Fake (generated) image–text pairs,
  - Mismatched image–text pairs (real images with incorrect captions).

  The total discriminator loss is defined as:

$$\mathcal{L}_D = \mathcal{L}_{\text{real}} + \frac{1}{2}\left(\mathcal{L}_{\text{fake}} + \mathcal{L}_{\text{mismatch}}\right)$$

  with components:
  - Real Loss:

$$\mathcal{L}_{\text{real}} = \mathbb{E}_{(x,t)}\left[\text{ReLU}(1 - D(x,t))\right]$$

  - Fake Loss:

$$\mathcal{L}_{\text{fake}} = \mathbb{E}_{(x_{\text{fake}},t)}\left[\text{ReLU}(1 + D(x_{\text{fake}},t))\right]$$

– Mismatched Loss:

$$\mathcal{L}_{\text{mismatch}} = \mathbb{E}_{(x,t_{\text{wrong}})}\left[\text{ReLU}(1 + D(x, t_{\text{wrong}}))\right]$$

This loss penalizes the discriminator when mismatched image-text pairs are incorrectly classified as real, reinforcing alignment between images and their respective descriptions.

- **Generator Loss:**

$$\mathcal{L}_G = -\mathbb{E}[D_{\text{fake}}]$$

We use the non-saturating GAN loss, defined as the negative mean of the discriminator's output on fake images. This encourages the generator to produce images that the discriminator classifies as real by maximizing the discriminator's response.

Contrastive Similarity Loss (InfoNCE)

To align image and text embeddings in a shared latent space, a contrastive loss (based on InfoNCE) is applied. It encourages high similarity for matched pairs and discourages similarity for mismatches:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(I, T^+))}{\sum_i \exp(\text{sim}(I, T_i))}$$

where $\text{sim}(I, T)$ is the cosine similarity between normalized image and text embeddings.

Discriminator Loss (WGAN-GP: used in Experiment-3)

Wasserstein GAN with Gradient Penalty stabilizes GAN training by using Wasserstein distance instead of binary cross-entropy. A gradient penalty term enforces the Lipschitz constraint, improving convergence and preventing mode collapse.

- Real loss: $\mathcal{L}_D^{\text{real}} = -\mathbb{E}[D(x_{\text{real}}, t)]$
- Fake loss: $\mathcal{L}_D^{\text{fake}} = \mathbb{E}[D(x_{\text{fake}}, t)]$
- Gradient Penalty: $\mathcal{L}_{\text{GP}} = \mathbb{E}[(\|\nabla_{\hat{x}} D(\hat{x}, t)\|_2 - 1)^2]$

The total discriminator loss is then:

$$\mathcal{L}_D = \mathcal{L}_D^{\text{real}} + \mathcal{L}_D^{\text{fake}} + \lambda_{\text{GP}} \cdot \mathcal{L}_{\text{GP}}$$

4. Experiments

Link to the Github repository with codes for following experiments.

Dataset and Training Details We train our model on the CUB-200-2011 dataset, which contains images along with 10 textual descriptions per image.

- **Optimizer:** Adam, learning rate 2×10^{-4}
- **Batch size:** 32
- **Text Encoder:** Pretrained and frozen during generator training

4.1. Experiment 1: Baseline Training

Objective: Train the SEA Attention GAN using the standard architecture and loss functions. **Configuration:**

- Epochs: 500
- Loss: Adversarial loss
  - Discriminator Loss (Matching Aware Loss)
  - Generator Loss ($\mathcal{L}_G$)
- Training Time: **70 hours (T4 GPU)**

4.2. Experiment 2: Adding InfoNCE Loss

Objective: Incorporate InfoNCE loss alongside Adversarial and DAMSM losses. **Configuration:**

- Epochs: 500
- Loss:
  - Adversarial
    - Discriminator Loss (Matching Aware Loss)
    - Generator Loss ($\mathcal{L}_G$)
  - Contrastive Loss (InfoNCE) ($\mathcal{L}_{contrastive}$)
  - $\mathcal{L}_{\text{total}} = \mathcal{L}_G + \lambda \cdot \mathcal{L}_{\text{contrastive}}$
    We used $\lambda = 0.1$ in our experiments.
- Training Time: **70 hours (T4 GPU)**

4.3. Experiment 3: Finetuning Resnet backbone Discriminator and WGAN Loss

- **Objective:** Trained a text-to-image GAN with a ResNet-based discriminator and WGAN-GP loss. Initially, the generator received more updates (5x) to stabilize learning; later, the focus shifted to the discriminator (3x updates).
- **Observation:** By epoch 50, the generator began exploiting frequent updates to fool the discriminator. Hence, we adjusted to favor discriminator training.
  Generated images were reasonable; further improvements could be made by strategies like freezing/unfreezing the discriminator.
- Epochs: 200
- Loss: WGAN-GP
  We used $\lambda_{\text{GP}} = 1.0$ in our experiments.
- Discriminator: Pretrained ResNet-18 backbone , fine-tuned on CUB-200 dataset
- Training Time: **40 hours (T4 GPU)**

4.4. Experiment 4: CLIP-based Text Encoder for SeattnGAN

- **Objective**: Use a CLIP-based text encoder with adapter to align text embeddings for text-to-image synthesis.
- Epochs: 389
- Loss: Adversarial loss with matching-aware and gradient penalty terms
- Text Encoder: CLIPTextEncoder (ViT-B/32 backbone, adapter trained to match RNN encoder output)
- Generator: Residual blocks with affine text conditioning and global attention

- Discriminator: Residual downsampling, text-image fusion at final stage
- Training Time: **53 hours (T4 GPU)**

5. Quantitative and Qualitative Analysis

5.1. Quantitative Results

We evaluate four variants of SEAttnGAN on the CUB-200 dataset:

1. **Baseline training**
2. **+InfoNCE loss**
3. **Fine-tuned ResNet backbone**
4. **CLIP-based text encoder (DF-GAN)**

For each, we note the FID over training epochs, and report the final metrics in tables.

Epoch	FID
150	280.60
160	300.59
180	218.05
200	**217.67**

Table 1. FID Scores across Epochs (ResNet)

Epoch	FID
220	241.49
288	302.38
317	283.16
338	286.19
368	290.92
388	**239.64**

Table 2. FID Scores across Epochs (Clip Encoder)

Epoch	FID
50	239.16
100	199.70
150	188.72
200	165.33
250	152.93
300	143.15
350	134.27
400	**118.76**
450	154.17
500	154.50

Table 3. FID Scores across Epochs (Sea Attan GAN with INFO NCE loss)

Epoch	FID
50	237.42
100	235.36
150	207.25
200	206.14
250	180.79
300	170.06
350	172.90
400	155.58
450	158.23
500	**139.79**

Table 4. FID Scores across Epochs (base line model)

5.2. Qualitative Analysis

Presents one exemplar generated image from each SEAttnGAN variant, along with the specific input caption used for conditioning.

Figure 1. **Baseline training: "this bird is a very bright red color with jet black wings and black outer and inner rectrices."**

Conclusion

We explored improvements to text-to-image synthesis using SEA-AttnGAN on the CUB-200 dataset. By adding InfoNCE loss, using WGAN-GP with a ResNet discriminator, and integrating a CLIP-based text encoder, we tried to achieve better FID scores and more accurate image-text alignment. Among all models, the InfoNCE-enhanced variant gave the best performance. Our results highlight the effectiveness of contrastive learning and pretrained encoders for generating realistic, semantically consistent images.

Figure 2. **Baseline training: this is a small bird with a black face and beak blue everywhere but on the underneath of the bird which is white and black and two white strips on each wing**

Figure 3. **+ InfoNCE loss: "this vibrant yellow bird has a black long thin beak, dark gray tail feathers and gray wings, as well as black coloring in the area between the bill and eye."**

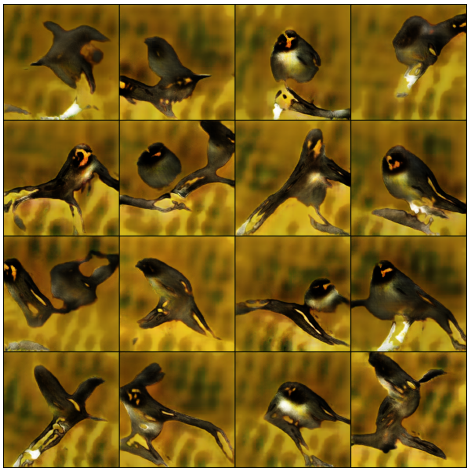

Figure 4. **+ InfoNce: this is a small bird with a black face and beak and blue everywhere on body but white on the underneath of the bird and two yellow strips on each wing"**

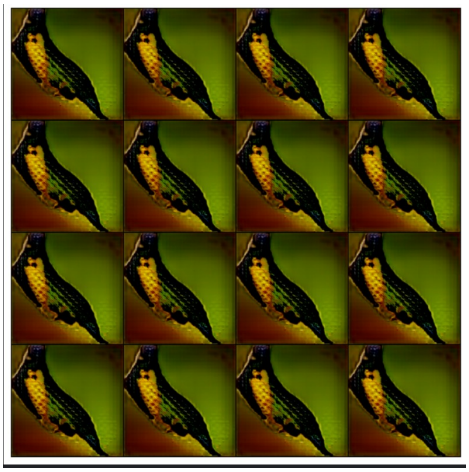

Figure 5. **+ Finetuned Resnet: "this vibrant yellow bird has a black long thin beak, dark gray tail feathers and gray wings, as well as black coloring in the area between the bill and eye."**

References

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. 1

[2] M. Jin, C. Zhang, Q. Yu, H. Xue, X. Jin, and X. Yang, "A simple and effective baseline for attentional generative adversarial networks," *arXiv preprint arXiv:2306.14708*, 2023. 2

[3] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022. 2