CPSC 8430
Fall 2024
Homework 4
Generative Adversarial Networks (GANs)
Abdullah Al Mamun

GitHub repository: **https://github.com/abdullm/CPSC_8430_Fall24_AM_HW4**
Trained GAN Models Location:
https://drive.google.com/drive/folders/1unEdWbiIHaAp-G1Cl4BORA6lTVu2N51b?usp=sharing

1. Introduction and Objective

The objective of this homework task is to implement and compare various Generative Adversarial Networks (GANs), including DCGAN, WGAN, and ACGAN, for image generation tasks using the CIFAR-10 dataset. Each model is trained to generate realistic synthetic images conditioned on specific class labels. The task involves evaluating the models based on their ability to generate high-quality images, as well as analyzing their performance through metrics like FID (Frechet Inception Distance) and loss functions. This task demonstrates the application of deep learning techniques for generative modeling and explores the impact of architectural and training differences across GAN variants.Model

2. Comparison of GAN Architecture and Model Parameters:

The table below compares DCGAN, WGAN, and ACGAN based on their GAN architectures and model parameters used in the implementation.

Feature	DCGAN	WGAN	ACGAN
Generator Architecture	• Noise (100) and one-hot class vector (119) concatenated. • Fully connected layer with 4x4x512 output reshaped to (4, 4, 512). • ConvTranspose2D blocks (256 → 128 → 64 → 3 channels). • BatchNorm and ReLU after each layer except the output layer. • Output: Tanh activation for scaled image generation.	• Noise (100) and one-hot class vector (10) concatenated. • Fully connected layer with 2x2x256 output reshaped to (2, 2, 256). • ConvTranspose2D blocks (256 → 128 → 128 → 128 → 3 channels). • BatchNorm and ReLU after each layer except the output layer. • Output: Tanh activation for scaled image generation.	• Noise (100) and one-hot class vector (10) concatenated. • Fully connected layer with 4x4x512 output reshaped to (4, 4, 512). • ConvTranspose2D blocks (512 → 256 → 128 → 64 → 3 channels). • BatchNorm and ReLU after each layer except the output layer.

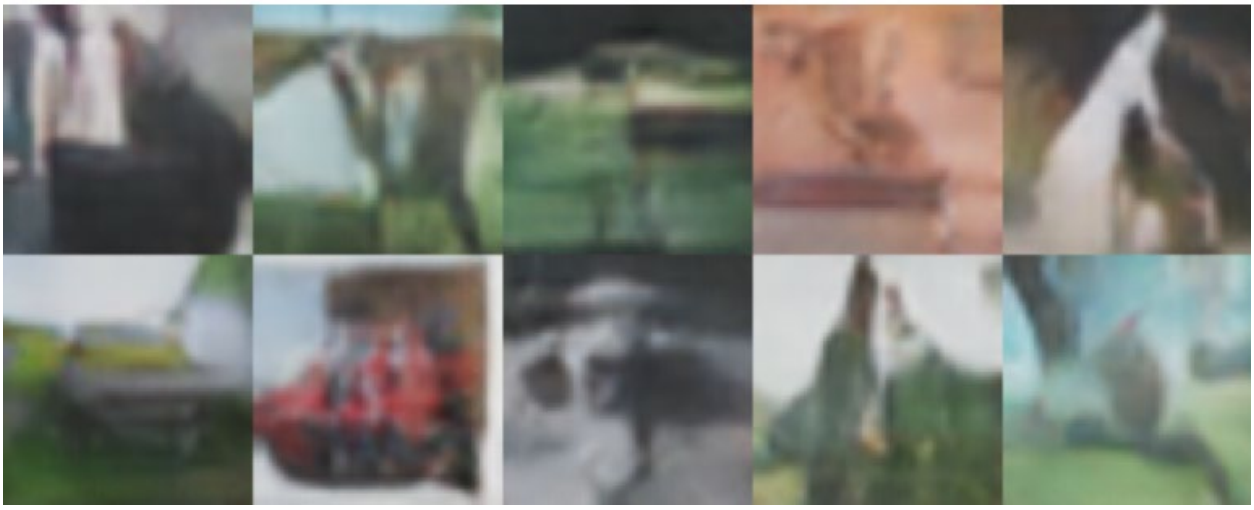
Feature	DCGAN	WGAN	ACGAN
			• Output: Tanh activation for scaled image generation.
Discriminator Architecture	• Image input (3 channels). • Embedding layer for class labels expanded to match image spatial dimensions. • Conv2D blocks (64 → 128 → 256 → 512). • Fully connected layer for real/fake classification. • Leaky ReLU activations. • Batch normalization for intermediate layers.	• Image input (3 channels). • Embedding layer for class labels expanded to match image spatial dimensions. • Conv2D blocks (64 → 128 → 256 → 1). • Fully connected layer for real/fake classification. • Leaky ReLU activations. • No BatchNorm in discriminator (critic).	• Image input (3 channels). • Embedding layer for class labels expanded to match image spatial dimensions. • Conv2D blocks (64 → 128 → 256 → 512). • Separate fully connected layers for real/fake classification and class prediction. • Leaky ReLU activations. • Batch normalization for intermediate layers.
Loss Functions	Binary Cross Entropy (BCE) for real/fake classification.	Wasserstein Loss with Gradient Penalty (Critic).	BCE for real/fake classification. Cross-Entropy Loss for class prediction.
Learning Rates	• Generator: 5E-4 • Discriminator: 5E-5	• Generator: 5E-5 • Discriminator: 2E-4	• Generator: 2E-4 • Discriminator: 2E-4
Epochs	100	500	100
Techniques Used	• Batch normalization in generator and discriminator layers.	• Gradient Penalty (lambda = 5) • Gradient Clipping for numerical stability.	• Softmax activation for class prediction in discriminator. • Separate loss terms for real/fake and class classification in discriminator.

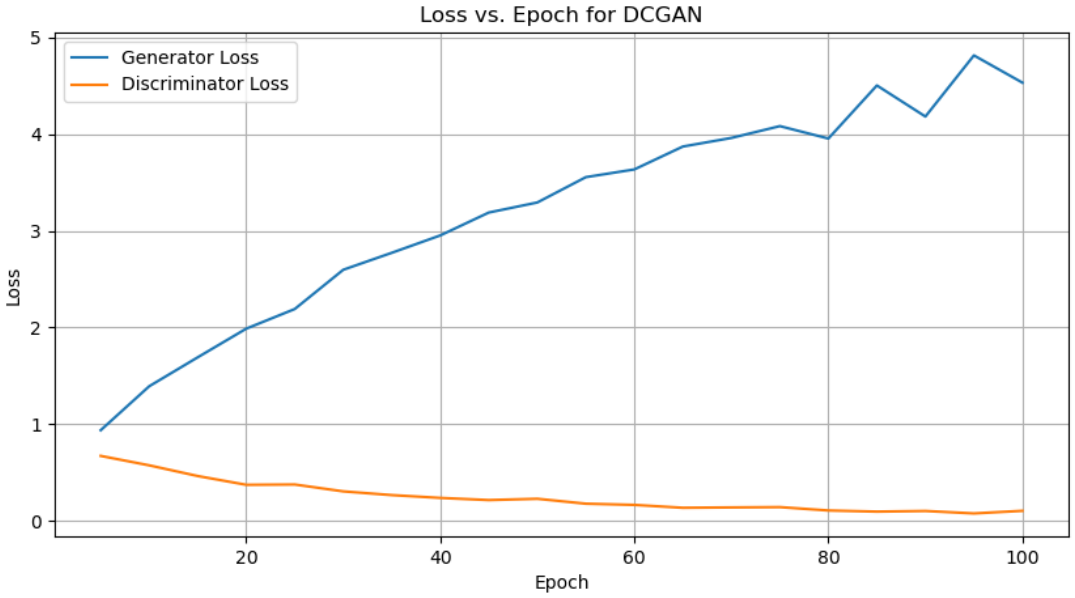
3. Results:

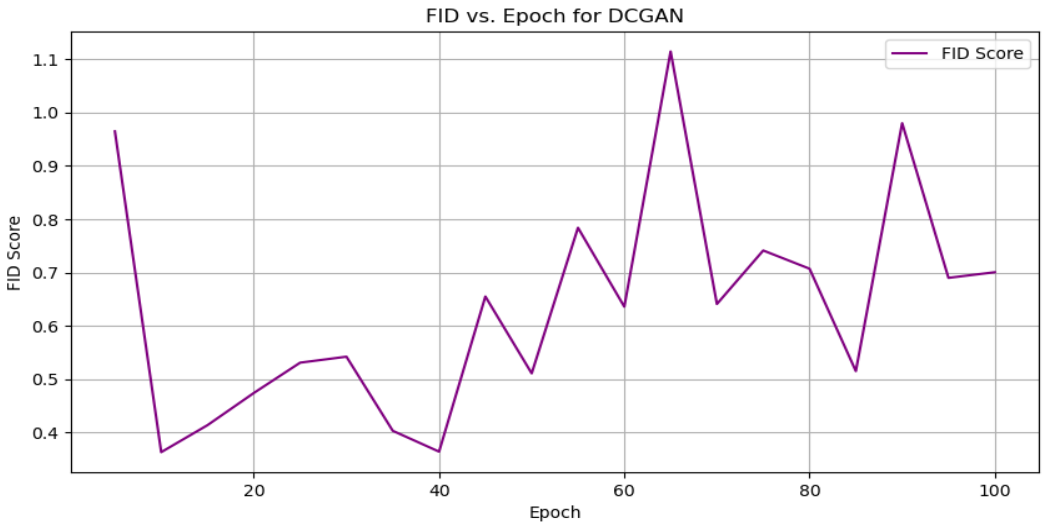
3.1. DCGAN

The following figures depict the 10 best generated images using the trained generator after 500 epochs, loss functions vs. epoch plot, and FID vs. epoch plot for DCGAN.

Generated Images for DCGAN

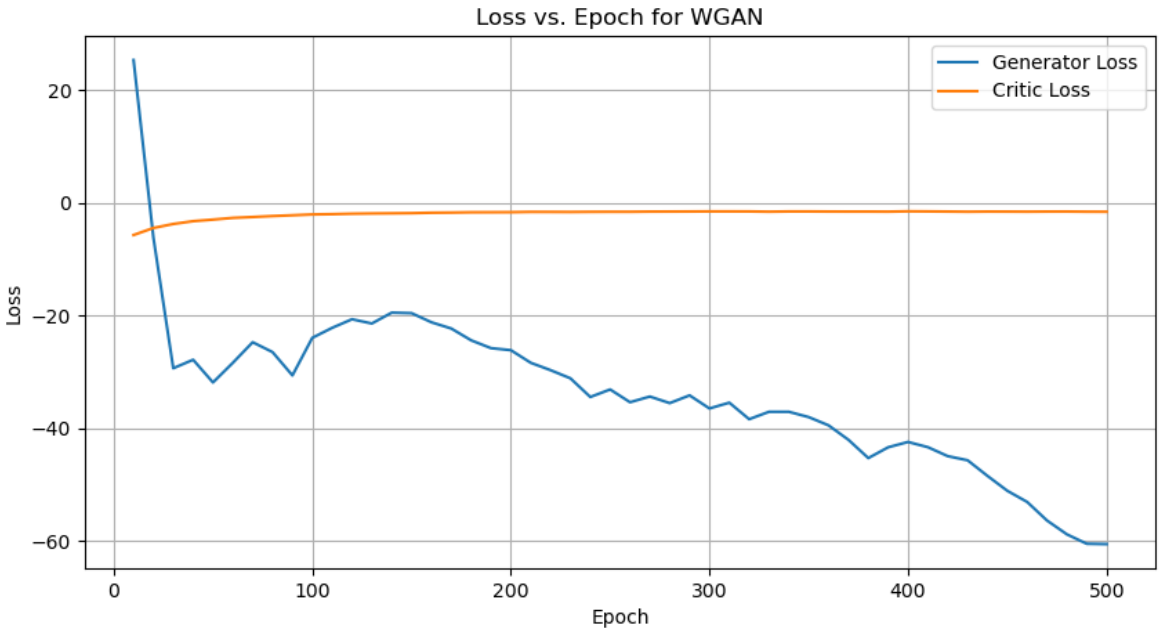


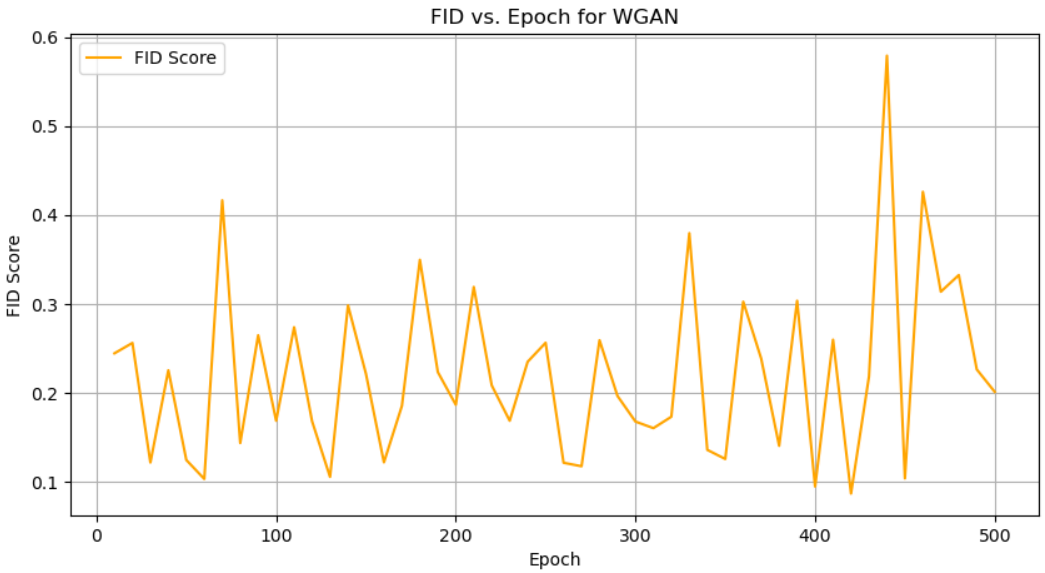




3.2. WGAN

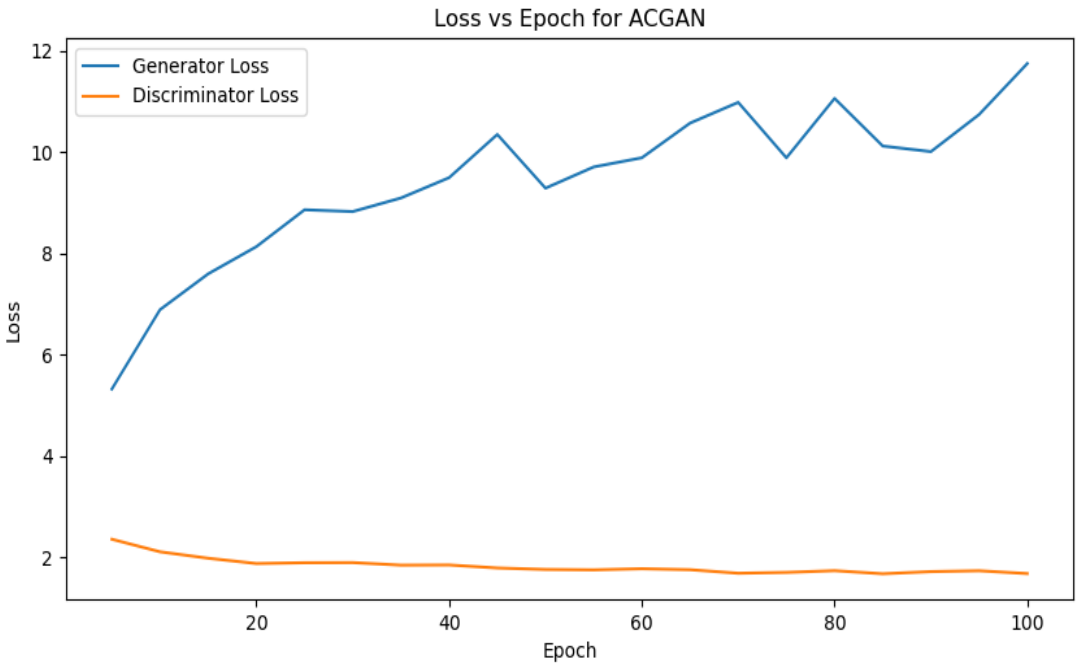
The following figures depict the 10 best generated images using the trained generator after 500 epochs, loss functions vs. epoch plot, and FID vs. epoch plot for WGAN.

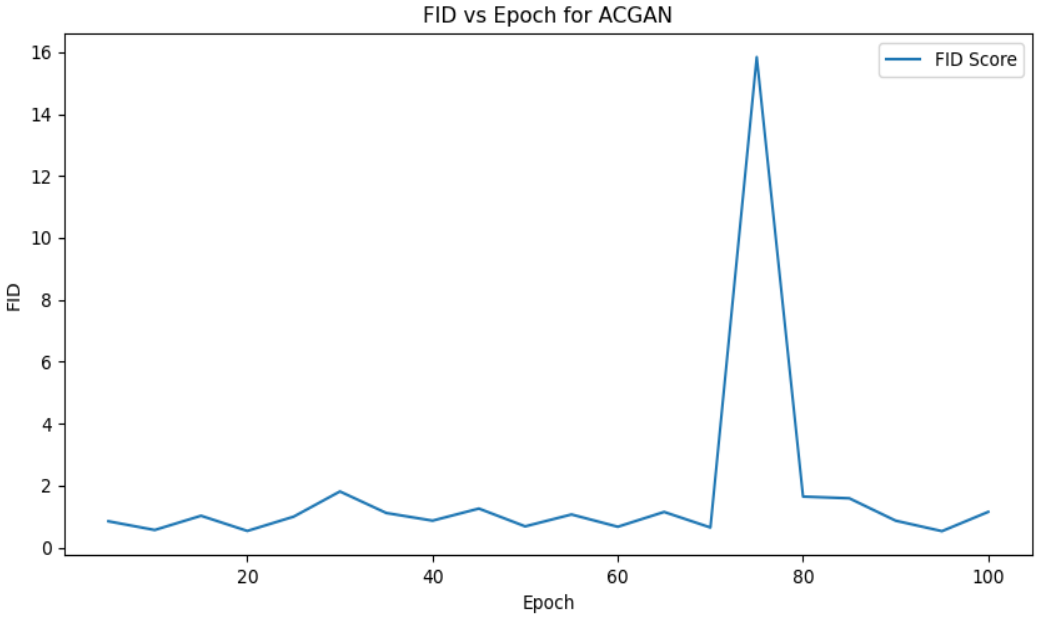




3.3. ACGAN

The following figures depict the 10 best generated images using the trained generator after 500 epochs, loss functions vs. epoch plot, and FID vs. epoch plot for ACGAN.


Loss vs Epoch for ACGAN
Generator Loss
Discriminator Loss
Loss
Epoch
12
10
8
6
4
2
20
40
60
80
100


FID vs Epoch for ACGAN
FID Score
FID
Epoch
16
14
12
10
8
6
4
2
0
20
40
60
80
100