Quantitative Comparison of DCGAN with DDPM

Anagha H.C.
Information Technology
National Institute of Technology Karnataka
Surathkal, Karnataka
hcanagha.211it008@nitk.edu.in

Abhayjit Singh Gulati
Information Technology
National Institute of Technology Karnataka
Surathkal, Karnataka
abhayjitsinghgulati.211ee002@nitk.edu.in

Sachin Prasanna
Information Technology
National Institute of Technology Karnataka
Surathkal, Karnataka
sachinprasanna.211it058@nitk.edu.in

***Abstract*—This paper presents a quantitative evaluation of Deep Convolutional Generative Adversarial Network (DCGAN) and Denoising Diffusion Probabilistic Models (DDPM) using the Fashion MNIST dataset. Fréchet Inception Distance (FID) and Inception Score (IS) metrics are employed to assess the fidelity and diversity of generated samples. Results reveal nuanced performance differences, providing insights for researchers and practitioners in the field of generative modeling and computer vision.**

***Index Terms*—Deep Convolutional Generative Adversarial Network, Denoising Diffusion Probabilistic Models, Gaussian Distribution, Fréchet Inception Distance**

I. Introduction

Generative models have emerged as powerful tools for synthesizing realistic and diverse data, finding applications in various domains, including computer vision and image generation. In this context, the Deep Convolutional Generative Adversarial Network (DCGAN) and Diffusion Probabilistic Models (DDPM) stand out as influential architectures with distinct approaches to generating high-quality images. The focus of this paper is to provide a quantitative comparison of these two models using the Fashion MNIST dataset, a widely adopted benchmark for evaluating generative models in the realm of fashion-related image synthesis.

The evaluation metrics employed for this comparative analysis are the Fréchet Inception Distance (FID) and Inception Score (IS). FID quantifies the dissimilarity between the distributions of real and generated data, offering insights into how well the models capture the underlying data distribution. Meanwhile, IS assesses the quality and diversity of the generated samples, providing a holistic measure of generative performance.

By conducting a rigorous analysis of DCGAN and DDPM on the Fashion MNIST dataset, we aim to unveil performance nuances that can inform the selection of these models for specific applications. This study contributes to the growing body of research aimed at advancing our understanding of generative models' capabilities, with implications for their practical use in fashion image synthesis and beyond. The subsequent sections delve into the experimental setup, methodology, and results, offering a comprehensive assessment of DCGAN and DDPM in the context of fashion-related generative tasks.

II. Literature Review

A. Generative Models

Generative models are a class of artificial intelligence algorithms that focus on creating data rather than making predictions. They learn the underlying patterns and structures within datasets, enabling them to generate new, synthetic data samples. These models have diverse applications, including generating realistic images, natural language text and audio. The ability to generate data with high fidelity and diversity has significant implications for fields like computer vision, natural language processing, and data augmentation. DDPMs also have a similar aim of learning the distribution of a training data sample and then generating a new sample that closely resembles it.

B. Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models, often abbreviated as DDPMs, are a noteworthy category within the realm of generative models. They do so by iteratively introducing controlled noise into an initial input signal. The underlying concept is to acquire a deep understanding of the noise removal process, enabling the generation of entirely fresh and coherent data samples. One notable achievement of DDPMs is their ability to produce high-quality images through a process inspired by nonequilibrium thermodynamics. This process involves iteratively applying noise and denoising to create new samples with impressive image synthesis results [1]. Further, Improved Denoising Diffusion Probabilistic Models have shown promising results in various applications, demonstrating enhanced capabilities in generating high-quality images with reduced noise levels [2]. This methodology stands in contrast to traditional generative models, which typically focus on modeling data distribution and sampling from it directly. In DDPMs, the noise injection and subsequent denoising steps are central to the creative process. This approach draws inspiration from the principles of diffusion, mimicking how noise dissipates in physical systems, such as in nonequilibrium thermodynamics

C. Generative Adversarial Networks

Generative Adversarial Networks (GANs), introduced by Ian Goodfellow and colleagues in 2014, comprise two crucial

neural networks: a generator and a discriminator. The generator and discriminator (aka critic).The generator produces a sample, such as an image, from a latent code. Ideally, the distribution of these images should be indistinguishable from the training distribution[3], while the discriminator's role is to distinguish real data from generated data. They engage in a competitive game, with the generator refining its output to resemble genuine data and the discriminator enhancing its ability to differentiate. Typically, a GAN consists of two networks: GAN training strikes a balance with a dynamic feedback loop; as the generator improves, the discriminator adapts, fostering ongoing competition. GANs excel in producing highly realistic data for computer vision, art, and data augmentation. In a related context, adversarial nets establish a competitive framework, pitting a generative model against a discriminative model. The generative model aims to craft "counterfeit" samples indistinguishable from genuine data, while the discriminative model detects discrepancies [4].

D. Deep Convolutional Generative Adversarial Networks

Deep Convolutional Generative Adversarial Networks (DCGAN) have become integral in unsupervised representation learning, as demonstrated by Radford et al. [5]

The architecture of DCGAN introduces key elements such as convolutional layers without fully connected counterparts and batch normalization. These design choices enhance training stability and enable the model to capture complex spatial patterns, making it particularly effective for image synthesis tasks.

The referenced study establishes DCGAN's capability in unsupervised learning, showcasing its proficiency in learning meaningful representations. In this work, we extend this exploration by quantitatively comparing DCGAN with Diffusion Probabilistic Models (DDPM) on the Fashion MNIST dataset. Building upon the insights from the foundational DCGAN paper, our analysis scrutinizes its performance, providing a comprehensive evaluation of its practical implications and contributions in the realm of generative modeling.

E. Perfomance Metrics

1) ***Fréchet Inception Distance (FID) Score****:* The Fréchet Inception Distance (FID) has emerged as a prominent metric for evaluating the quality of generated images in the domain of generative models. Introduced by Martin Heusel et al. in 2017, FID provides a robust measure of the similarity between the distribution of real and generated images. By utilizing feature representations from an Inception-v3 neural network, FID captures both the quality and diversity of generated samples. A lower FID score signifies a closer match between the generated and real image distributions, indicating superior performance in terms of image fidelity. [6]

2) ***Inception Score (IS)****:* The Inception Score (IS) is another widely utilized metric in the evaluation of generative models, introduced by Tim Salimans et al. in 2016. This metric combines both image quality and diversity, offering a quantitative measure of the overall performance of a generative model. IS is computed based on the output probabilities of an image classifier (Inception-v3) and provides a mean and standard deviation of these probabilities across generated samples. Higher IS scores, indicative of higher mean probabilities and lower standard deviations, suggest that the model produces diverse and high-quality images. [7]

III. DATASET

The Fashion MNIST dataset has emerged as a widely utilized benchmark for evaluating and comparing various machine learning and computer vision models, particularly in the context of image classification and generative modeling. Introduced as an alternative to the traditional MNIST dataset, Fashion MNIST was curated by Zalando Research to provide a more challenging and realistic dataset while maintaining the simplicity of MNIST.

Comprising a collection of grayscale images, each of 28x28 pixels, the Fashion MNIST dataset consists of 10 fashion-related classes, representing different articles of clothing and accessories. The classes include items such as t-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and ankle boots. This diversity makes Fashion MNIST a suitable choice for assessing the generalization capabilities of models across a range of fashion-related image recognition tasks.

The dataset's accessibility and straightforward structure make it an attractive option for researchers and practitioners seeking to experiment with and benchmark their algorithms in the domain of computer vision. Moreover, Fashion MNIST's manageable size facilitates rapid prototyping and experimentation, making it a preferred choice for tasks like generative modeling, where models like DCGAN and DDPM can be trained and evaluated efficiently.

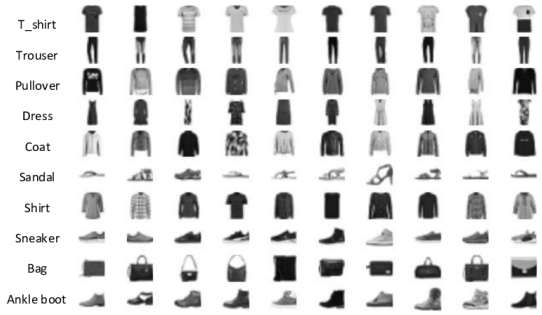

Fig. 1. Fashion MNIST Dataset

IV. METHODOLOGY

We coded out our own DCGAN and DDPM for the comparison purpose. Their architectures are as described in the subsequent subsections.

A. Deep Convolutional Generative Adversarial Network

1) ***Generator****:* The generator model for the Deep Convolutional Generative Adversarial Network (DCGAN) is designed to transform a latent noise vector of dimensionality 100 into realistic 28x28 pixel grayscale images. This model, implemented

using TensorFlow and Keras, consists of a sequential stack of layers, starting with a densely connected layer mapping the input noise to a high-dimensional representation. Batch normalization and LeakyReLU activation are applied to enhance training stability and introduce non-linearity. The subsequent reshaping layer transforms the representation into a 3D tensor, serving as the basis for the transposed convolutional layers that progressively upsample the data. The generator architecture employs Conv2DTranspose layers with appropriate strides and padding to increase spatial resolution, ultimately producing a synthetic image with the desired dimensions. The final activation function is set to tanh, ensuring the pixel values are within the range [-1, 1]. This generator architecture adheres to DCGAN principles, contributing to the model's ability to generate realistic fashion images in the context of our study.

*2) **Discriminator**:* The discriminator model is a crucial component designed to discern between real and generated images. Constructed using TensorFlow and Keras, this model follows a sequential architecture, beginning with Conv2D layers that perform convolutional operations on the input images. The use of LeakyReLU activation functions introduces non-linearity, enhancing the model's ability to capture complex patterns. To prevent overfitting, Dropout layers with a dropout rate of 0.3 are strategically placed after each convolutional layer.

The discriminator's architecture employs two convolutional layers with increasing filter sizes, effectively learning hierarchical features from the input images. The subsequent Flatten layer collapses the spatial dimensions, preparing the data for the final classification via a dense layer with a single output unit. This output represents the discriminator's decision on the authenticity of the input image – a value closer to 1 indicating a real image and closer to 0 indicating a generated image.

The discriminator's role in the adversarial training process is pivotal, as it provides feedback to the generator, guiding it to produce more realistic images. The architecture's design ensures that the discriminator can effectively distinguish between real and generated samples, contributing to the overall training stability and success of the DCGAN in generating high-quality fashion images.

*3) **Training**:* The training procedure for the Deep Convolutional Generative Adversarial Network (DCGAN) is orchestrated through a carefully crafted set of components and functions. The loss functions are defined using binary cross-entropy, with the discriminator loss evaluating the ability of the discriminator to distinguish between real and generated images, and the generator loss encouraging the generation of realistic samples. The model was ran for 100 epochs.

Two Adam optimizers, each with a learning rate of 1e-4, independently optimize the parameters of the generator and discriminator. This dual optimization scheme enables the models to evolve in response to their specific objectives during training.

A checkpoint mechanism is implemented to ensure model recovery and training continuity. The TensorFlow Checkpoint API is utilized to manage the saving and restoring of model parameters, optimizer states, and other critical training variables.

The training process itself is encapsulated within a TensorFlow function decorated with '@tf.function'. This function, named 'train_step', executes a training iteration. It involves generating images from random noise using the generator, computing the discriminator's response to both real and generated images, and subsequently calculating the generator and discriminator losses. The gradients of these losses with respect to the trainable variables are then computed using gradient tape, and the optimizer applies these gradients to update the model parameters.

This cohesive structure of loss definitions, optimizers, checkpointing, and the training step function forms the backbone of the adversarial training process for the DCGAN. This setup ensures the iterative refinement of the generator and discriminator, ultimately leading to the generation of compelling and realistic fashion images as part of the study.

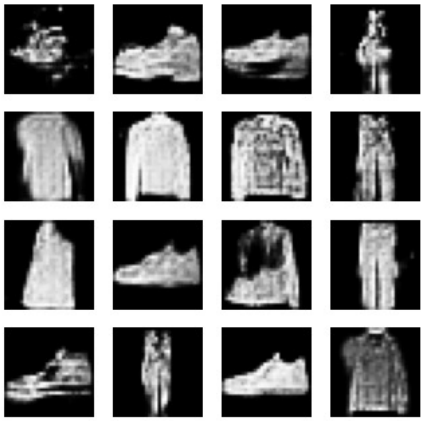

Fig. 2. Images generated by the DCGAN

B. Denoising Diffusion Probabilistic Model

*1) **Forward and Backward Processes**:* The implementation of the Diffusion Probabilistic Model (DDPM) is encapsulated in the a python class, designed as a PyTorch module. This class integrates a diffusion process, which progressively adds noise to an input image over a specified number of time steps. The model utilizes a neural network to estimate the added noise, allowing for the generation of diverse and realistic samples.

Key parameters include n_steps, representing the number of diffusion steps, and the dynamic beta values ranging from a minimum beta value to a maximum beta value. The class includes methods for both the forward and backward diffusion processes. During the forward process, noise is added to the input image at a given time step, and the method returns the noisy image. The backward process involves running each noisy image through the neural network for each timestep, providing an estimation of the noise that was added.

The class incorporates essential components such as the diffusion parameters (alphas, betas, alpha_bars) and handles device placement for efficient computation. The PyTorch tensors and operations enable seamless integration with deep learning workflows, facilitating the incorporation of DDPM into broader applications, such as generative modeling and image synthesis.

*2) **U-Net Architecture for the Neural Network**:* The proposed U-Net architecture, referred to as MyUNet, is a deep neural network designed for diffusion probabilistic modeling, integrating both positional and temporal embeddings to capture complex spatiotemporal dependencies in the input data. This architecture is implemented using PyTorch and comprises sinusoidal embeddings, time-based encoding, and a series of convolutional and transposed convolutional blocks.

The architecture begins with sinusoidal embedding of the temporal dimension, creating a representation of time that is then embedded into the network through a learned temporal embedding layer. The network consists of multiple blocks, each composed of convolutional layers, normalization, activation functions, and down-sampling operations. The blocks are organized to progressively capture hierarchical features from the input image.

The bottleneck of the U-Net introduces a mid-level temporal embedding and a set of blocks that maintain the spatial resolution. The subsequent transposed convolutional layers then upsample the data, merging information from lower-resolution feature maps through skip connections. This design allows the network to maintain both local and global information, enhancing its ability to generate high-quality samples.

The detailed architecture involves three main sections: the first half, bottleneck, and second half. Each section contains multiple blocks responsible for processing and refining the input data. The final output is a reconstructed image with a single channel, achieved through a convolutional layer.

The temporal embeddings play a crucial role in conditioning the model to capture temporal dependencies throughout the diffusion process. By incorporating sinusoidal embeddings and learned temporal embeddings, the network can effectively model the progression of diffusion steps and generate diverse and realistic samples.

This U-Net architecture, tailored for diffusion probabilistic modeling, exhibits a balance between capturing intricate temporal dependencies and spatial details, making it a suitable candidate for generative tasks involving sequential data. The design choices, such as skip connections and embedding strategies, contribute to the overall effectiveness of the network in generating coherent and diverse samples.

*3) **Training**:* The training loop for the Diffusion Probabilistic Model (DDPM) is a comprehensive iterative process designed for optimizing model parameters based on the Mean Squared Error (MSE) loss between predicted noise and true noise in the generated images. This loop operates over a specified number of epochs and processes batches of data from a given data loader. The model was ran for 20 epochs.

In each epoch, the loop iterates through the data loader, loading batches of images. For each image in the batch, random noise is generated, and a random time step is selected from the predefined number of diffusion steps. The forward process of the DDPM is then applied, generating noisy images. Subsequently, the model estimates the noise based on these generated images and the selected time step using the backward process.

The optimization objective is to minimize the MSE loss between the predicted noise and the true noise. The model parameters are updated using the backpropagation algorithm through the optimization step. The epoch loss is computed by accumulating the losses over all batches, and the average loss per image is logged. The training loop also includes a mechanism to store the model's state dictionary if the current epoch's loss is the best encountered so far. This ensures that the best-performing model is saved for future use.

The training progress is monitored through informative log strings, displaying the current loss at each epoch. If the model achieves the best loss so far, a corresponding log string indicates that the model has been stored.

Overall, this training loop serves as the core engine for optimizing the DDPM, iteratively improving its performance in generating images that faithfully capture the distribution of the input data. The inclusion of model storing and optional image display enhances the usability and interpretability of the training process, contributing to a comprehensive and insightful training routine for the DDPM.

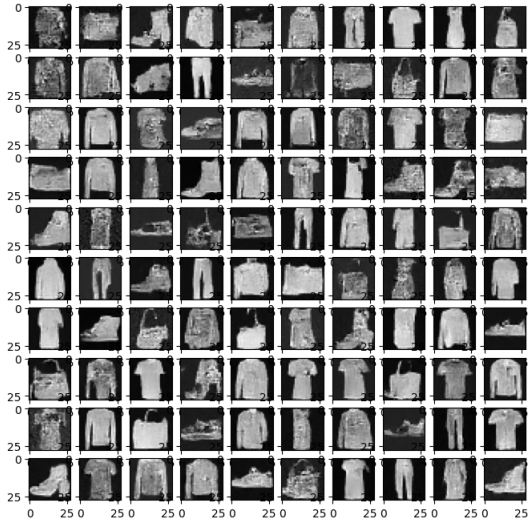

Fig. 3. Images generated by the DDPM

V. Result and Conclusion

In conclusion, the comprehensive evaluation of our models, with FID and IS scores as performance metrics, provides valuable insights into their respective generative capabilities. The DDPM model demonstrated a significantly lower FID score of 48.45, indicating a closer resemblance of its generated

images to real samples in comparison to the DCGAN, which yielded a higher FID score of 162.5. Moreover, the IS scores further emphasize the superior performance of the DDPM, with a mean of 2.5 and a standard deviation of 0.8, compared to the DCGAN's mean of 2.1 and standard deviation of 0.6.

These results collectively suggest that the DDPM outperforms the DCGAN in terms of both image quality and diversity. The lower FID score indicates a more faithful reproduction of the dataset's distribution, while the higher mean and standard deviation of the IS scores underscore the DDPM's ability to generate images with increased variety and realism.

The success of the DDPM can be attributed to its unique diffusion process, which effectively models the progressive addition of noise to images, allowing for the generation of diverse and high-quality samples. The findings presented in this evaluation underscore the potential of probabilistic models like DDPM in the domain of fashion image synthesis.

TABLE I
QUANTITATIVE COMPARISON OF DDPM AND DCGAN

Model	FID	IS
DDPM	48.45	2.5 +/- 0.8
DCGAN	162.5	2.1 +/- 0.6

VI. FUTURE SCOPE

The future scope of this project includes fine-tuning the parameters of our DDPM to align with those in the "Improved Denoising Diffusion Probabilistic Models" paper[2]. This adjustment could potentially enhance the model's generative performance. Additionally, exploring more advanced GAN variants like StyleGAN or BigGAN offers the potential for improved image quality and diversity.

ACKNOWLEDGMENT

We extend our heartfelt appreciation to Professor **Palla Parasuram Yadav** for his invaluable guidance and constant support throughout this project. His expertise and insightful feedback were instrumental in achieving our goals. We are deeply grateful to **Sujatha M** for her unwavering mentorship and dedication. Her guidance significantly contributed to the project's success.

REFERENCES

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in Proc. 34th Conf. Neural Inf. Process. Syst. (NeurIPS), Vancouver, Canada, 2020.

[2] A. Nichol and P. Dhariwal, "Improved Denoising Diffusion Probabilistic Models," in Proc. 38th Int. Conf. Mach. Learn. (ICML), PMLR 139, 2021.

[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," in Proc. Int. Conf. Learn. Representations, 2018. [Online]. Available: https://openreview.net/forum?id=Hk99zCeAb

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[5] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in Proc. Int. Conf. Learn. Representations, 2016. [Online]. Available: https://arxiv.org/abs/1511.06434

[6] A. Mathiasen and F. Hvilshøj, "Backpropagating through Fréchet Inception Distance," Under review by the International Conference on Machine Learning (ICML).

[7] M. Lee and J. Seok, "Score-Guided Generative Adversarial Networks," arXiv preprint arXiv:2004.04396, 2020.