Investigation of Generative Adversarial Networks and their Comparative Parametric Analysis

by

Anahit Khachatryan

A thesis submitted in partial satisfaction of

the requirements for the degree of

Master of Science

in

Computer & Information Science

in the

COLLEGE OF SCIENCE AND ENGINEERING

of the

AMERICAN UNIVERSITY OF ARMENIA

Supervisor: Elen Vardanyan

Signature: ____________________ Date:______________

Committee Member: ____________________

Signature: ____________________ Date:______________

Committee Member: ______________________

Signature: ____________________ Date:______________

Committee Member: ______________________

Signature: ____________________ Date:______________

Abstract

Investigation of Generative Adversarial Networks and their Comparative Parametric Analysis

Author: Anahit Khachatryan

Generative Adversarial Networks (GANs) have emerged as a powerful tool for generating synthetic data in various domains such as image, text, and speech. However, the performance of GANs is highly dependent on their hyperparameters and architecture, and selecting the best set of parameters can be challenging. This paper investigates the performance of different GAN architectures and their corresponding hyperparameters through a comprehensive comparative parametric analysis. We evaluate the performance of GANs using multiple quantitative metrics such as Fréchet Inception Distance (FID), F1 score, precision and recall. Additionally, we conducted a user study to perform a qualitative evaluation of the generated samples. Our results show that the performance of GANs is susceptible to the choice of hyperparameters in quantitative and qualitative metrics. This study comprehensively evaluates GANs and their parameters, enabling researchers and practitioners to make informed decisions when using GANs for synthetic data generation.

Keywords: Generative Adversarial Networks, Synthetic Data Generation, Hyperparameters Optimization, Comparative Analysis, Quantitative Metrics, Qualitative Evaluation.

Licenses for Software and Content

Software Copyright License (to be distributed with software developed for masters project)

Content Copyright License (to be included with Technical Report)

Table of Contents

1 Introduction

1.1 Background and Motivation

The well-known generative adversarial networks (GANs) are used in supervised and unsupervised learning, including image and video generation and data augmentation.

The GANs architecture, in general (**Radford et al., 2015**; **Zhang et al., 2017**), consists of two blocks. The first part is the generator that creates new images, and the second is the discriminator that evaluates whether the generated data is real or fake. The generator aims to create a realistic image such that the discriminator cannot differentiate between accurate and generated data. The discriminator and generator training is a two-player min-max game where players want to optimize their cost function respectively (**Goodfellow et al., 2014**).

Each type of GANs has its strengths and weaknesses. There is still a big question about which GAN performs better. Today's challenge is finding optimal architectures and configurations for those components. The main problem is the lack of consistent evaluation metrics and limited comparisons done previously, which include putting all the architectures on the same page. This research work aims to do a comparative parametric analysis. This technique analyzes how different parametric changes can impact the model performance and the ability to generate high-quality data. By changing and analyzing the key parameters or hyperparameters, the researchers can gain insights into the strengths and weaknesses of different GAN architectures and improve their performance. The comparative analysis will help us to understand the complexities of GANs and improve upon them. This approach can improve their architecture, leading to better synthetic data generation.

1.2 Our contributions

- We conducted a comprehensive literature review on existing GANs, their main types and architectures, and selected four types.
- To evaluate the performance of different GAN architectures and training methods, measuring various metrics, including the Fréchet Inception Distance (FID), F1 score, precision, and recall.
- After selecting the four types of GANs to evaluate, modifying various hyperparameters and analyzing their performance is essential. This involves making parametric changes to the GANs, such as changing the learning rate, Adam optimizer β parameter, clip value, discriminator iterations, and batch normalization settings. Modifying these hyperparameters makes it possible to identify the optimal values for each that lead to the best FID score, F1 score, precision, and recall performance.
- To ensure a fair comparison between the different GAN architectures and training methods, it is vital to keep the architecture of each GAN constant, changing only the corresponding loss values. This makes sure that any differences in performance can be completely ascribed to the training and hyperparameter changes. This makes sure that any variations in performance can be completely attributable to the variations in hyperparameters and training strategies. It is essential to keep in mind that the optimization process is stochastic and that every run it does throughout the training period can result in a different set of results.
- We summarize the results of the analysis and provide a comparison between the different GAN architectures and training methods.
- The proposed hypothesis should be either accepted or rejected based on the capabilities of the results.

1.4 Thesis Outline

The remainder of this report is organized as follows:

Chapter 2 gives a thorough study of the pertinent literature on the subject of the investigation. It assesses the body of literature critically and pinpoints the research holes, obstacles, and opportunities in the area. The theoretical foundation and justification for the research topics and objectives are presented in the chapter on the literature review.

Chapter 3 outlines the research design, instruments, data collection techniques, and data analysis strategies used in the study. It describes the procedures followed in carrying out the study's data analysis. The methodology chapter ensures that the study is comprehensive, trustworthy, and valid.

Chapter 4 presents the research findings and interprets the results. The results and analysis chapter provide evidence to support or reject the research hypotheses.

Chapter 5 summarizes the research findings, interprets the results, and draws conclusions based on the research objectives and questions. It also provides recommendations for future research and practical implications of the research.

Appendix A shows the research findings, interprets the results based on the research objectives and questions. It contains supplementary information such as plots, graphs, codes, and other relevant data. It provides additional details that support the research study but are optional to the main body of the thesis.

Appendix B provides the link of the codes done throughout this thesis.

2 Literature Review

This chapter outlines the fundamental theoretical concepts behind the generative adversarial networks and presents an overview of the related work in the literature.

2.1 Introduction to Generative adversarial networks (GANs)

Deep learning offers the potential for discovering complicated models that describe probability distributions across artificial intelligence, including speech, audio, and picture processing. Many discriminative models have been used in deep learning and are so far successful applications. These models are based on piecewise linear units and backpropagation methods (**Jarrett et al., 2009**). Due to the difficulties of approximating several probabilistic calculations in maximum likelihood estimation, deep generative models have less of an influence. The capacity of GANs to produce high-quality data that is frequently indistinguishable from actual data is one of their key advantages (**Ian Goodfellow et al., 2014**).

The adversarial modeling method is the simplest when both models are multilayer perceptrons. We define a before input noise variable $p_z(z)$, then a data space where G is a differentiable function represented by a multilayer perceptron, to learn the generator's distribution p_g over data x. A second multilayer perceptron that produces a single scalar is also defined. D(x) denotes the likelihood that x originated from the data instead of p_g. We train D to maximize the likelihood that training examples and samples from G will receive the correct label. In addition, we train G to reduce $log(1 - D(G(z)))$.

$$min_G \, max_D \, L_{GAN}(D, G) = E_{(x\sim p_{data(x)})}[log(D(x))] + E_{z\sim p_z(z)}[log(1 - D(G(z)))] \quad (1)$$

G is the generator network

D is the discriminator network

$x \sim P_{data}$ represents a real data sample from the training dataset

$z \sim p_z$ represents a noise vector sampled from a prior distribution p_z

log is the natural logarithm function

E is the expected value operator

Optimizing D to completion in the inner loop of training on finite datasets would be computationally costly and lead to overfitting. Instead, we switch back and forth between the k stages of D optimization and one step of G optimization. **Algorithm 1** presents the procedure.

For G to learn effectively in practice, equation 1 offers more gradient. D can reject samples with high confidence because they deviate from the training data early in learning when G is inadequate. $log(1 - D(G(z)))$ saturates in this situation. We can train G to maximize log D(G(z)) rather than to minimize $log(1 - D(G(z)))$.

Following proper training of the generator and discriminator, generated handwritten image examples are shown in **Figure 1**.

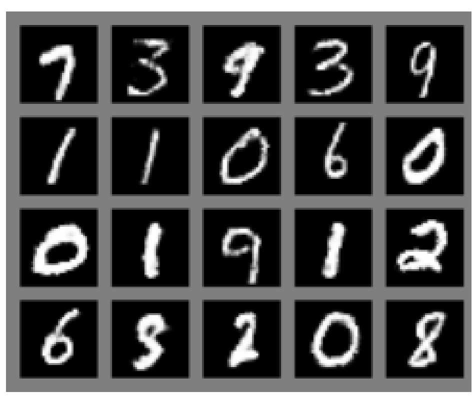

Figure 1: Generated handwritten digits after training a GAN model.

for number of training iterations **do**
for k steps **do**
- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right].$$

end for
- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$$

end for

Algorithm 1: Gradient descent training of GANs.

In the next section, we discuss the various modifications proposed to the generative adversarial networks, their architectures for generative and discriminative models, gradient calculations, experimental performance usage, and their advantages and disadvantages.

2.2 The general architecture of Generative adversarial networks (GANs)

The general architecture of the GANs is discussed in this section and visualized in **Figure 2**.

2.2.1 The generator model

The generator creates a sample in the domain using a fixed-length random vector as input. The generative process is seeded with a randomly selected vector from a Gaussian distribution. A compressed representation of the data distribution will be formed after training when points in this multidimensional vector space match points in the domain. This particular vector space is known as latent space, or vector spaces made up of latent variables. Latent variables, often hidden variables, are significant for a domain but cannot be observed directly. Latent variables and latent spaces are frequently referred to as a projection or compression of a data distribution. In other words, a latent space offers a compression or high-level idea of the observed raw data, such as the input data distribution. In the case of GANs, the generator model assigns meaning to points in a predetermined latent space, allowing for the generation of new and distinctive output instances from the generator model by adding new points selected from the latent space as input. The generator model is preserved and utilized to produce new samples after training.

2.2.2 The discriminator model

The discriminator predicts a binary class label of real or fake (created) based on the input of an actual or generated sample from the domain.

The training dataset contains the actual example. The generator model outputs the created examples. The discriminator is a typical classification paradigm that is widely known. The discriminator model is abandoned after training since we are more interested in the generator. The generator can occasionally be put to new uses since it has mastered the art of successfully extracting

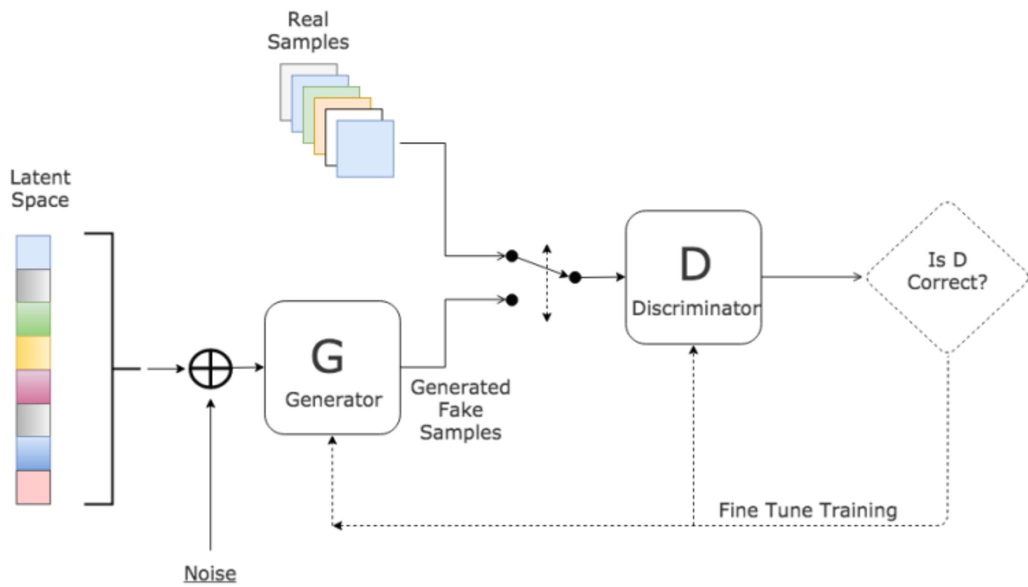

Figure 2: The structure of a simple Generative Adversarial Network (GAN)

features from examples in the issue area. Some or all feature extraction layers can be used in transfer learning applications with the same or similar input data. Convolutional Neural Networks, or CNNs, are frequently used by GANs as the generator and discriminator models when working with picture data.

This may be due to the fact that the initial description of the method was based on computer vision and used CNNs and image data, as well as the remarkable developments made in recent years using CNNs more generally to achieve state-of-the-art outcomes on a variety of computer vision tasks, including object detection and face recognition.When modeling image data, the latent space, the generator's input, provides a compressed representation of the images or photographs used to train the model. Additionally, it implies that the generator creates new images or photos, producing a result that model creators or users can quickly view and evaluate. The focus on computer vision applications with CNNs and the enormous improvements in the performance of GANs over other generative models, whether or not they are deep learning-based, may both be due to this aspect, the ability to assess the quality of the generated output visually. This fact may be more important than others.

2.3 Types of Generative Adversarial Networks

Many types of GANs variations try to improve stability. The leading architecture remains the same, but the loss functions can differ. The most popular types of GANs are Vanilla GAN (**2.3.1**), LSGAN (**2.3.2**), WGAN (**2.3.3**), WGAN GP (**2.3.4**), Conditional GANs (**Isola et al., 2017**), Cycle GANs (**Zhu et al., 2017**), Progressive GANs (**Karras et al., 2018**), Style GANs (**Karras et al., 2019**), and Self-attention GANs (**Zhang et al., 2019**), and many more. In this chapter, we will introduce four of them in detail.

2.3.1 Vanilla Generative Adversarial Networks (GANs)

The architecture of Vanilla GANs consists of two networks, a generator (G) and a discriminator (D), trained in an adversarial manner. The generator takes a random noise vector (z) as input and generates a fake sample which is fed to the discriminator along with real samples (x) from the dataset. The discriminator aims to distinguish between real and fake samples, while the generator aims to generate samples that can fool the discriminator. The loss function of Vanilla GANs is introduced in **(1)**.

One of the limitations of Vanilla GANs is the instability during training, which is due to the non-convex nature of the loss function. The generator and discriminator networks play a minimax game, where the generator tries to minimize the discriminator's ability to distinguish between real and fake samples. In contrast, the discriminator tries to maximize its ability to distinguish between them. This leads to oscillations in the training process, making it challenging to achieve convergence.

Another area for improvement of Vanilla GANs is the mode collapse problem, where the generator produces only a limited set of outputs, failing to capture the entire distribution of the actual data. This is often attributed to the instability of the training process and the difficulty of accurately estimating the accurate data distribution.

Despite these drawbacks, Vanilla GANs have been used to generate realistic images from noise (**Radford et al., 2015**), super-resolution (**Ledig et al., 2017**), and image-to-image translation (**Zhu et al., 2017**), among other image generation tasks. They have also been improved to overcome some drawbacks, such as Wasserstein GANs (**Arjovsky et al., 2017**) and Progressive GANs (**Karras et al., 2018**).

2.3.2 Least Squares Generative Adversarial Networks (LSGANs)

In recent years, a new variant of Generative Adversarial Networks (GANs) has emerged, known as Least Squares GANs (LSGANs) (**Tao et al., 2017**). LSGANs have gained attention for their ability to generate high-quality images with better stability and faster convergence than traditional GANs. This section will overview LSGANs, including their architecture, loss functions, and advantages over traditional GANs.

LSGANs use a different loss function than traditional GANs, which helps to address the problem of vanishing gradients during training. The loss function of LSGANs is given by:

$$min_{G} max_{D} L(D, G) = \frac{1}{2} E_{x \sim p_{data}} [(D(x) - 1)^2] + \frac{1}{2} E_{z \sim p_{z}} [(D(G(z)))^2] \quad (2)$$

The LSGAN's goal is to reduce this loss function. The architecture of LSGANs is similar to that of traditional GANs, with the discriminator and generator networks being trained in an adversarial manner. However, using the LSGAN loss function results in more stable training and better-quality generated images. In addition, it has been demonstrated that LSGANs are less likely to have mode collapse, a typical issue with standard GANs (**Arjovsky et al., 2017**).

Among other image production tasks, LSGANs have been used to produce high-resolution images (**Yu et al., 2018**) and images from textual descriptions (**Reed et al., 2016**). In both cases, LSGANs could generate images with higher fidelity and diversity than traditional GANs. One of the advantages of LSGANs over conventional GANs is their capacity to produce clearer, more lifelike images. This is so that training is more stable and gradients are better since the LSGAN loss function encourages the discriminator to output a continuous value rather than a binary one. The stability of the loss function has also been proven to allow LSGANs to converge more quickly than conventional GANs (**Tao et al., 2017**).

Despite the advantages of LSGANs, there are still some limitations and challenges associated with this approach. One of the main limitations is the need to manually tune the constants a and b in the loss function, which can be time-consuming and require expertise. Additionally, LSGANs may still suffer from mode collapse in certain situations, although this is less common than traditional GANs.

LSGANs are a promising approach for generating high-quality images with better stability and faster convergence than traditional GANs. The LSGAN loss function provides better gradients and more stable training, resulting in sharper and more realistic images.

2.3.3 Wasserstein Generative Adversarial Networks (WGAN)

The Generative Adversarial Network (GAN) version known as Wasserstein GANs (WGANs) has attracted attention for its capacity to produce high-quality images with better stability and convergence than conventional GANs (**Arjovsky et al., 2017**). Some of the drawbacks of conventional GANs are addressed by WGANs, which use the Wasserstein distance metric to calculate the separation between the proper and generated distributions. An overview of WGANs, including their design, loss function, and benefits over conventional GANs, will be given in this chapter. The loss function of WGANs is given by:

$$min_G max_D L_{WGAN}(D, G) = E_{x \sim p_{data}}[D(x)] - E_{z \sim p_z}[D(G(z))] \qquad (3)$$

The goal of the WGAN is to maximize this loss function, which calculates the discrepancy between the output of the discriminator's predicted value on generated samples and real samples. The architecture of WGANs is similar to that of traditional GANs, with the discriminator and generator networks being trained in an adversarial manner. However, using the Wasserstein distance metric results in more stable training and better-generated image quality. Additionally, it has been discovered that WGANs are less likely to experience mode collapse, a common issue with traditional GANs (**Arjovsky et al., 2017**).

One advantage of WGANs over traditional GANs is their ability to produce sharper and more diverse images. This is done using the Wasserstein distance metric, which delivers a more significant and continuous gradient signal to prevent some instability and vanishing gradient difficulties that could occur in standard GANs. Additionally, WGANs are more stable during training and less sensitive to hyperparameters, which may make them easier to use and apply in practice (**Gulrajani et al., 2017**).

WGANs have been used to generate high-resolution images (**Gulrajani et al., 2017**) and images from textual descriptions (**Zhang et al., 2017**), among other image-generation tasks. In both cases, WGANs could generate high-quality images with better fidelity and diversity than traditional GANs.

2.3.4 Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN GP)

The WGAN loss function can be further improved by adding a gradient penalty term. The WGAN-GP loss function is given by:

$$min_G max_D L_{WGAN\,GP}(D, G) = E_{x\sim p_{data}}[D(x)] - E_{z\sim p_z}[D(G(z))] + \lambda GP(E_{\sim x}[(||\nabla_{\sim x} D(\sim x)||_2 - 1)^2]) \quad (4)$$

E_x is the expected value over interpolated points between real data samples and generated data samples

$\nabla_{\sim x} D(\sim x)$ is the gradient of the discriminator with respect to the interpolated points

$||.||_2$ represents the L2 norm

λ is a hyperparameter that controls the strength of the gradient penalty

The gradient penalty term encourages the discriminator to have a Lipschitz continuous function, which helps to stabilize training and improve the quality of generated images (**Gulrajani et al., 2017**). Where D and G are the sets of potential discriminators and generators, respectively, ~x is a sample drawn uniformly along straight lines between pairs of data and generated samples, and λ is a penalty coefficient. x is an actual sample from the dataset, z is a random noise vector, and D and G are the sets of possible discriminators and generators, respectively.

This loss function, which evaluates the difference between the expected value of the discriminator's output on natural and produced samples and incentivizes the discriminator to have a Lipschitz continuous function, is what the WGAN-GP aims to maximize. The architecture of WGAN-GP is similar to that of WGANs and traditional GANs, with the discriminator and generator networks being trained in an adversarial manner. However, adding the gradient penalty term helps stabilize training and improve the quality of generated images. The gradient penalty term encourages the discriminator to have a Lipschitz continuous function, which means that a constant bounds the gradient of the discriminator concerning its input. This helps to avoid some of the instability and vanishing gradient problems that can occur in traditional GANs and other variants of WGANs (**Gulrajani et al., 2017**).

One of the advantages of WGAN-GP over other variants of GANs is its ability to generate high-quality images with better stability and convergence. WGAN-GP is less prone to mode collapse and better at preserving the diversity of the proper data distribution than other variants of GANs (**Gulrajani et al., 2017**). Additionally, WGAN-GP has been used to generate high-resolution images and videos, such as 1024 x 1024-pixel images of faces and 256 x 256-pixel videos of dancing human figures **(Karras et al., 2018**).

WGAN-GP has also been used for other image generation tasks, such as image inpainting, where missing parts of an image are filled in with plausible content. In one study, WGAN-GP was used to fill missing regions in natural images, such as landscapes and animals, and was shown to generate realistic and visually pleasing content.

2.5 The Applications

2.5.1 Image and Video Synthesis

One of the primary uses of GANs is synthesizing images and videos. It can produce realistic images when it learns the patterns and properties of the input data. This method produces synthetic content for films, video games, and animations.
High-resolution images produced by GANs can also be used in computer vision and medical imaging applications

2.5.2 Data Augmentation

In machine learning, data augmentation is a strategy to expand the quantity of training data available for models. Synthetic data can be used as a supplement to the current data collection process. This method applies to object recognition, image categorization, and other computer vision applications.

2.5.3 Image Editing and Style Transfer

GANs can be used for style transfer and image altering. Transferring a style from one image to another is known as style transfer. Artistic photos and films can be produced using this method. By altering the image's texture, lighting, and other features, GANs can also modify pictures.

2.5.4 Text-to-Image Generation

Images can be produced from text descriptions using GANs. This method entails creating an image that corresponds to the given description. In the gaming industry, text-to-picture generation is helpful because it enables designers to create game assets quickly and easily.

2.5.5 Face Generation

Realistic faces can be created with GANs. This method has uses in the entertainment sector, where it can be applied to produce artificial actors and actresses. Law enforcement can also employ GANs to draw sketches of suspects.

2.5.6 Image Restoration

Images that are outdated or damaged can be restored using GANs. In this method, the GAN is trained to create high-resolution images from low-resolution ones. Old images and artworks can be brought back to their original quality with this technology, which has applications in historic preservation.

2.5.7 Drug Discovery

In the field of drug discovery, GANs are applicable. To create novel compounds with particular properties, GANs can be employed. This method may hasten the drug discovery process by eliminating the need for costly and time-consuming tests.

2.5.8 Speech Synthesis

It is possible to create lifelike speech with GANs. This method involves teaching the GAN to produce speech that sounds like a specific voice. Speech synthesis is used in the entertainment business to provide artificial voices for films, video games, and animations.

2.6 Challenges of Generative Adversarial Networks

Several challenges associated with GANs must be addressed to improve their overall performance. This chapter will discuss some of the challenges in GANs and the ongoing research to overcome them.

2.6.1 Mode Collapse

Mode collapse is a common problem in GANs, where the generator produces a limited variety of output images. Instead of generating diverse images, the generator produces the same or similar output for different inputs. This happens when the generator identifies a subset of the training data that produces good results and focuses only on that subset. Several techniques have been proposed to address mode collapse in GANs.

One approach is regularization techniques like weight decay or dropout to prevent the network from overfitting to a particular mode (**Gulrajani et al., 2017**). Another approach is to use a Wasserstein GAN (WGAN) or a WGAN with gradient penalty (WGAN-GP), which are more stable and less prone to mode collapse (**Gulrajani et al., 2017**).

2.6.2 Training Instability

Because the training process is unstable, GANs can be challenging to train. The adversarial training of the generator and discriminator in GANs can lead to a feedback loop in which the generator produces images that are overly similar to the discriminator's training set.. This leads to the discriminator being too good at distinguishing between real and fake images, which makes it difficult for the generator to improve.

Several techniques have been proposed to address training instability in GANs. One approach is to add noise to the inputs of the discriminator, which has been shown to improve the stability of the training process (**Arjovsky et al., 2017**). Another approach is to use alternative loss functions, such as hinge loss or least squares loss, which are more stable than the traditional binary cross-entropy loss (**Mao et al., 2017**). Progressive training is another technique that improves stability and reduces the likelihood of mode collapse (**Karras et al., 2018**).

2.6.3 Evaluation Metrics

Evaluating the quality of GAN-generated images is challenging, as there are no clear metrics for measuring output quality. Metrics like mean squared error (MSE) or peak signal-to-noise ratio (PSNR) are used to evaluate the quality of images. But they do not capture the complexity or diversity of the generated images. Perceptual metrics like the Inception Score (IS) or Fréchet Inception Distance (FID) have been proposed to evaluate GANs. The Inception Score measures the diversity and quality of the generated images based on the output of a pre-trained Inception network (**Salimans et al., 2016**). The Fréchet Inception Distance measures the similarity between the distributions of the real and generated images (**Heusel et al., 2017**). These measurements, however, have their own limits and might not offer a comprehensive assessment of the performance of GANs.

2.6.4 Limited Data Availability

GANs require large amounts of training data to learn the patterns and features of the input data. The available data is limited in many applications, such as medical imaging. In such cases, GANs may be unable to learn the input data's complex patterns and features.

Transfer learning techniques can address the challenge of limited data availability. Transfer learning involves training a GAN on a larger dataset and then fine-tuning it on a smaller dataset. Another approach is to use synthetic data generated by GANs to augment the existing dataset. This technique has shown improvement in the performance of machine learning models in image classification and object detection.

2.7 Related Studies, Research Gaps and Questions

Generative Adversarial Networks (GANs) have been the subject of numerous studies to examine their strengths and weaknesses. In this chapter, we will discuss some related studies in GANs.

A technique called BigGAN (**Brock et al., 2018**) uses a large-scale generator and discriminator network to generate high-resolution images. The authors showed that BigGAN outperformed previous state-of-the-art models on several image generation tasks.

The progressive growing GAN (**Karras et al., 2018**) gradually increases the resolution of the generated images during training. The authors showed that progressive growing GANs produce higher-quality images than traditional GANs and are more stable during training.

Wasserstein GAN with Gradient Penalty (WGAN-GP) (**Tao et al., 2017**) uses a Lipschitz constraint to improve the stability of the GAN training process. The authors showed that WGAN-GP produces higher-quality images than traditional GANs and is less prone to mode collapse.

Another technique is AutoGAN (**Gong et al., 2019**), which uses an autoencoder to learn the features of the input data and generate high-quality images. The authors showed that AutoGAN outperformed several state-of-the-art GAN models on benchmark datasets.

2.7.1 Generalization

GANs may generalize poorly to new datasets because they are frequently trained on limited datasets. There is a significant research gap that has to be filled regarding the generalization performance of GANs. Transfer learning and domain adaptation are two methods that have been suggested to enhance the generalization capabilities of GANs. Additional research is necessary to create GAN models that can adapt to new datasets.

2.7.2 Interpretability

GANs are frequently called "black-box" models since it is challenging to comprehend how the generator creates the output images. As it can assist researchers and practitioners in understanding the model's operation and making intelligent usage decisions, Interpretability is an essential field of study for GANs. Attention mechanisms and visualization strategies are just two of the methods that have been suggested to increase the Interpretability of GANs (**Zhang et al., 2018**).

2.7.3 Robustness

Small changes in the input data can cause significant changes in the output image, making GANs prone to adversarial attacks. Being robust ensures that adversarial examples, a significant research gap in GANs, do not easily fool the model. Adversarial training and robust optimization are two methods that have been suggested to increase the adaptability of GANs (**Madry et al., 2017**).

2.7.4 Ethical Considerations

GANs have several ethical implications, such as their potential use for creating fake images or videos for malicious purposes. Ethical considerations are an important research gap in GANs, as they can help ensure that the technology is used responsibly and ethically. Several studies have explored the ethical implications of GANs, such as their potential use for creating fake news or deep fakes. The ability of GANs to generate realistic images and videos raises concerns about their potential misuse, such as creating fake identities or manipulating public opinion.

2.7.5 Research Questions

Many questions still need to be addressed in investigating GANs. Some of the fundamental research problems in this area are how they can be optimized for various data types or applications.

As they have limitations when they generate high-quality data, and the need for further research is the generated data quality evaluation method as the GANs output. The solution is to use perceptual-based evaluation metrics such as Inception Score or Fréchet Inception Distance. These metrics are based on the pre-trained neural network. They classify or compare the generated data with real data. They also can capture the quality of the data, which is highly correlated with human evaluation. The human evaluation involves asking humans to rate the data's quality based on some criteria (realism, novelty, or diversity). Human evaluations are more accurate. It is time-consuming and sometimes subjective.

How the parametric adjustments might impact the effectiveness of GANs and the caliber of their output is a further area for investigation. In order to determine each model's relative strengths and limitations, the research would require doing tests with various GAN models, each with a unique set of parameters.

The study should ensure that all the experiments are conducted in a fair manner to ensure the validity of the results.

Having the research problems, we can formulate questions, investigate the GANs and do their parametric comparative analysis.

2.8 Metrics of Evaluations

It is crucial to evaluate the performance of generative adversarial networks (GANs) in order to assess the range and quality of the generated images or videos. An overview of some of the most well-liked metrics used to evaluate the effectiveness of GANs is provided in this chapter.

One of the measures most frequently used when evaluating GANs is the **Inception Score (IS) (Salimans et al., 2016**). The IS measures the quality and diversity of the generated images by computing the entropy of the predicted class probabilities from an Inception model trained on the actual images. The higher the IS score, the better the quality and diversity of the generated images. However, the IS has been criticized for its sensitivity to dataset biases and lack of correlation with human perception (**Barratt & Sharma, 2018**).

Another commonly used metric for evaluating GANs is the Fréchet **Inception Distance (FID)** (**Heusel et al., 2017**). The FID measures the distance between the actual and generated distributions of feature vectors extracted from the Inception model. The lower the FID score, the better the quality and diversity of the generated images. The FID has been shown to be more robust to dataset biases and to correlate better with human perception than the IS (**Barratt & Sharma, 2018**).

The **Precision and Recall (PR)** metric has also been proposed for evaluating GANs (**Kynkaanniemi et al., 2019**). The PR metric measures the precision and recall of the generated images by computing the percentage of generated images close to the authentic images in terms of feature similarity. The higher the precision and recall scores, the better the quality and diversity of the generated images. The PR metric is more robust to dataset biases and correlates better with human perception than the IS (**Barratt & Sharma, 2018**).

The **Maximum Mean Discrepancy (MMD)** metric has recently been proposed for evaluating GANs (**Sajjadi et al., 2018**). The MMD measures the distance between the actual and generated distributions of feature vectors extracted from a pre-trained model. The lower the MMD score, the better the quality and diversity of the generated images. The MMD metric is more robust to dataset biases than the FID (Sajjadi et al., 2018). There are still specific difficulties in evaluating GANs despite the availability of numerous evaluation criteria. A lack of standard evaluation criteria that can efficiently evaluate the quality and variety of the generated images across various datasets and applications is one of the significant issues. Human perception is subjective, which makes it challenging to create measures that precisely measure the perceived worth of the created images.

3 Methodology

This chapter will cover the research design and methodology, experimental setup, and evaluation metrics to compare the performance of various Generative Adversarial Network (GAN) architectures .

3.1 Research Design and Methodology

The research design involved training and testing several GAN architectures MNIST datasets. The GAN architectures used in the study included Vanilla GAN, LSGAN, WGAN, and WGAN GP. The evaluation metrics used included Fréchet Inception Distance (FID), F1 score, precision, and recall.

3.2 Experimental Setup

The experiments were conducted on a large-scale computing cluster using Pytorch (**PyTorch, n.d.**) which is an open-source library.

All architectures are implemented the same using 3 convolutional layers (CNN), Leaky rectified linear units (RELU) as activation functions, and Linear layers in both discriminator and generator blocks. The architecture of the generator and discriminator are presented in **Table 1** and **Table 2**.

The dataset consists of around 50,000 handwritten images 28x28 pixel sized each. The input channel is 1, meaning that they are gray-scaled.

The optimizer is chosen as ADAM, as we observed that the distribution of RMSProp is similar to ADAM, and it is unclear which one is 'better'.

The training process for each GAN architecture was performed using a specific configuration of hyperparameters, such as the learning rate, optimizer parameter beta, discriminant iterations, clipping value, gradient penalty, and batch normalization.

Layer (type)	Output Shape	Param #
Linear-1	[-1, 8192]	909,312
ReLU-2	[-1, 8192]	0
ConvTranspose2d-3	[-1, 256, 7, 7]	1,179,648
BatchNorm2d-4	[-1, 256, 7, 7]	512
ReLU-5	[-1, 256, 7, 7]	0
ConvTranspose2d-6	[-1, 128, 14, 14]	524,288
BatchNorm2d-7	[-1, 128, 14, 14]	256
ReLU-8	[-1, 128, 14, 14]	0
ConvTranspose2d-9	[-1, 1, 28, 28]	2,048
Tanh-10	[-1, 1, 28, 28]	0

Total params: 2,616,064
Trainable params: 2,616,064
Non-trainable params: 0

Table 1: The generator architecture and the number of trainable parameters.

Layer (type)	Output Shape	Param #
Linear-1	[-1, 784]	623,280
LeakyReLU-2	[-1, 784]	0
Conv2d-3	[-1, 512, 14, 14]	4,608
BatchNorm2d-4	[-1, 512, 14, 14]	1,024
LeakyReLU-5	[-1, 512, 14, 14]	0
Conv2d-6	[-1, 256, 7, 7]	1,179,648
BatchNorm2d-7	[-1, 256, 7, 7]	512
LeakyReLU-8	[-1, 256, 7, 7]	0
Conv2d-9	[-1, 128, 4, 4]	294,912
BatchNorm2d-10	[-1, 128, 4, 4]	256
LeakyReLU-11	[-1, 128, 4, 4]	0
AvgPool2d-12	[-1, 128, 1, 1]	0
Linear-13	[-1, 1]	129

Total params: 2,104,369
Trainable params: 2,104,369
Non-trainable params: 0

Table 2: The discriminator architecture and the number of trainable parameters.

3.2.1 Range of Hyperparameters and Parameters

To find the best FID across the training run for each run, we do hyperparameter optimization. Every epoch, we compute the FID between the 1000 samples generated by the model and the 1000 samples from the test set in order to select the best model. Each data collection completed this costly computational search. **Figure 4** shows the models' sensitivity to the hyper-parameters. In two steps, we compute the best FID: We choose the optimal narrow range of the model's hyper-parameters after completing a large-scale search on a chosen wide range of them. The larger range, which we believe showed better results, is used to find the narrow range hyper-parameter ranges. Then, we re-run the training of the selected model with best hyper-parameters, to estimate the stability of the training and report the mean FID.

	Vanilla GAN	LSGAN	WGAN	WGAN GP
Learning rate	(1e-1, 1e-5)	(1e-1, 1e-5)	(1e-1, 1e-5)	(1e-1, 1e-5)
Adam's β parameter	U(0,1)	U(0,1)	U(0,1)	U(0,1)
Disc iterations	(1,2,5)	(1,2,5)	(1,2,5)	(1,2,5)
Clip value	N/A	N/A	U(0,1)	N/A
Gradient penalty	N/A	N/A	N/A	L(20,100)
Batchnorm	True/False	True/False	True/False	True/False

Table 3: The range of hyper-parameters used to do the evaluation and analysis. “U” denotes uniform sampling and “L” denotes sampling on a log scale. (a,b) indicates a range of parameters.

The range of hyper-parameters is presented in **Table 3**. U(a,b) means that the variables are sampled randomly from a uniform distribution in the range of (a,b). The L(a,b) means that the variables are sampled on a log scale. Learning rate (1e-a,1e-b) indicates the range of hyperparameters decreasing by a factor of 10 each time.

Learning rate - generator/discriminator learning rate

β - Adam optimizer's parameter

Disc iterations - number of discriminator updates per one generator update

Clip value - parameter of WGAN, the gradients will be clipped to this value

Gradient penalty - multiplier of gradient penalty for WGAN

Batchnorm - if True the batch normalization will be used

3.3 Fairness of Comparison

The models should be fairly compared, and even when the parameter used is fixed, a given algorithm may produce different results depending on the architecture, hyperparameters, network weights' random initialization, or the data set. The best score across all dimensions, the average score, or the median score are all reasonable objectives. These options can also be combined; for instance, one might train the model several times using the best hyperparameters and average the result over initializations using random initializations.

1. *Architecture* (**Table1**, **Table2**): We use the same architecture for all models. We note that this architecture performs well on considered data sets.
2. *Hyperparameters* (**Table 3**): For both training hyperparameters such as learning rate, batch normalization settings, number of discriminator iterations as well as model-specific ones such as gradient penalty multiplier, clip value, we perform the hyperparameter optimization on one data set and infer a good range of hyperparameters to use on the chosen data set.
3. *Random seed*: Even with everything else being fixed, varying the random seed may influence the results. We initialize the weights of the network on the same seed to have a complete fair comparison across the other parameters.

4. *Computational budget*: Different algorithms can achieve the best results depending on the budget to optimize the parameters. In practice, one can use hyperparameter values suggested by respective authors or try to optimize them and, in particular, we optimize the hyperparameters for each model and data set by performing a random search. In order to narrow down the range of possible configurations we made different choices for each of these parameters in order to reduce the variety of potential combinations while still researching the best solutions.

4 Results and Analysis

4.1 Interpretation of Results

The results showed that Vanilla GAN and LSGAN had the worst performance in terms of achieved F1 score, precision, and recall. This is likely due to their simple objective function, which only tries to minimize the Jensen-Shannon divergence between the real and generated data distributions. This objective function can be difficult to optimize and may lead to mode collapse or instability during training. As a result, the generated images may be of lower quality and less diverse.

On the other hand, WGAN and WGAN-GP showed better performance in terms of F1 score, precision, and recall (**Figure 3**). WGAN uses the Wasserstein distance as the distance metric instead of Jensen-Shannon divergence, which has been shown to be more stable and less prone to mode collapse. WGAN-GP further improves the stability of the GAN training process by adding a gradient penalty term to the Wasserstein distance objective function, which encourages the discriminator to have a Lipschitz continuous gradient. This helps to prevent the discriminator from becoming too powerful and dominating the generator during training, which can lead to mode collapse.

In addition, the results showed that all the GANs were sensitive towards hyperparameter changes in terms of FID scores(**Figure 4**, **Figure 5**). Specifically, all of them were very sensitive to the learning rate.There is no model that is stable when changing any hyperparameter (**Table 4**).

Overall, the results of this study suggest that researched GANs were able to generate high-quality images when choosing the best parameter.

The studies also showed that all the models are sensitive towards the parametric changes according to FID scores and the models are the most sensitive to learning rate changes.

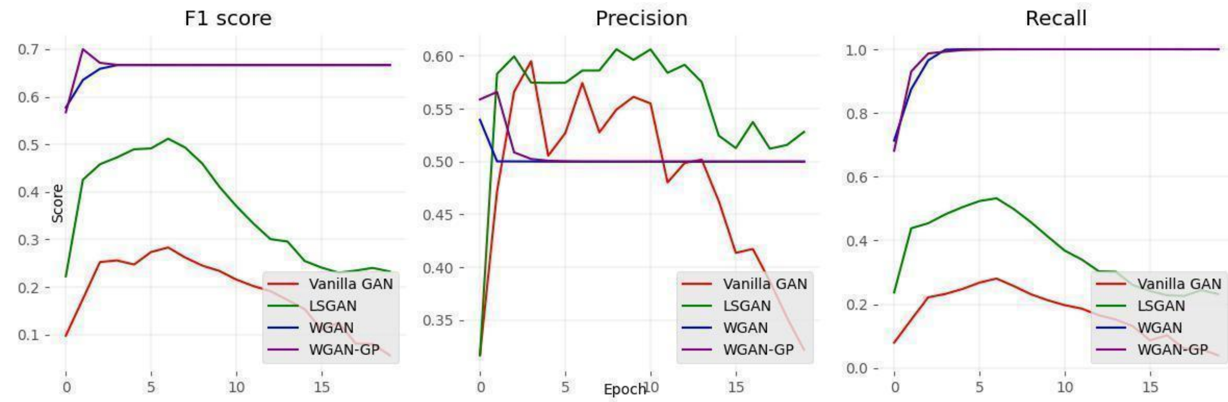

Figure 3: The plot shows the average F1 score, precision and recall across all hyperparameter changes. Both WGAN and WGAN GP have high precision and recall, and the underwhelming performance of Vanilla GAN and LSGAN should be a subject of further investigation.

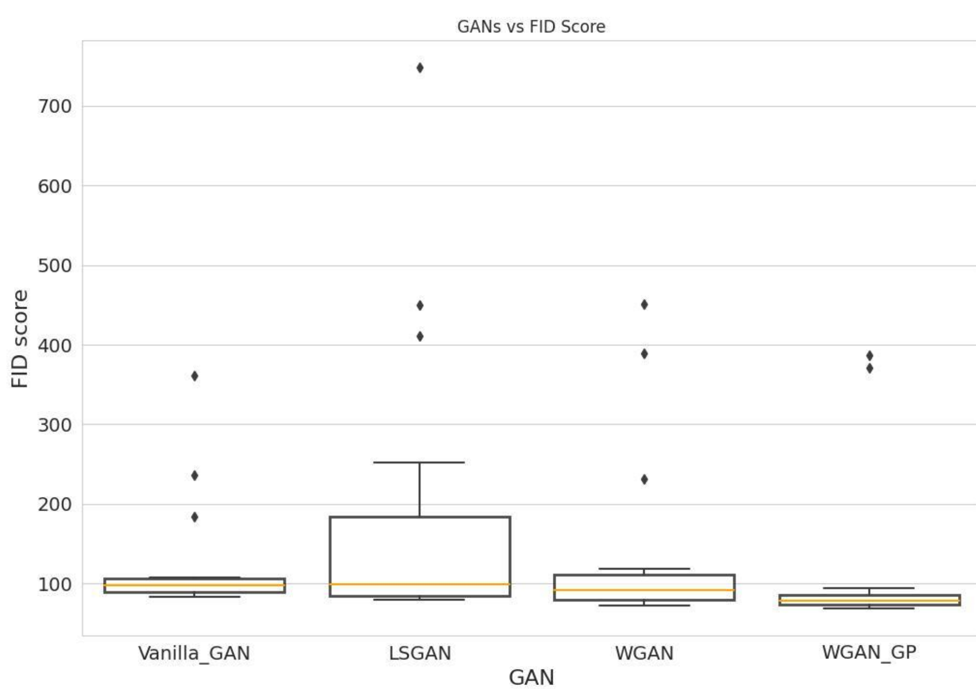

Figure 4: The FID scores across all parametric changes on four types of GANs.. We observe that GAN training is extremely sensitive to hyperparameter settings.

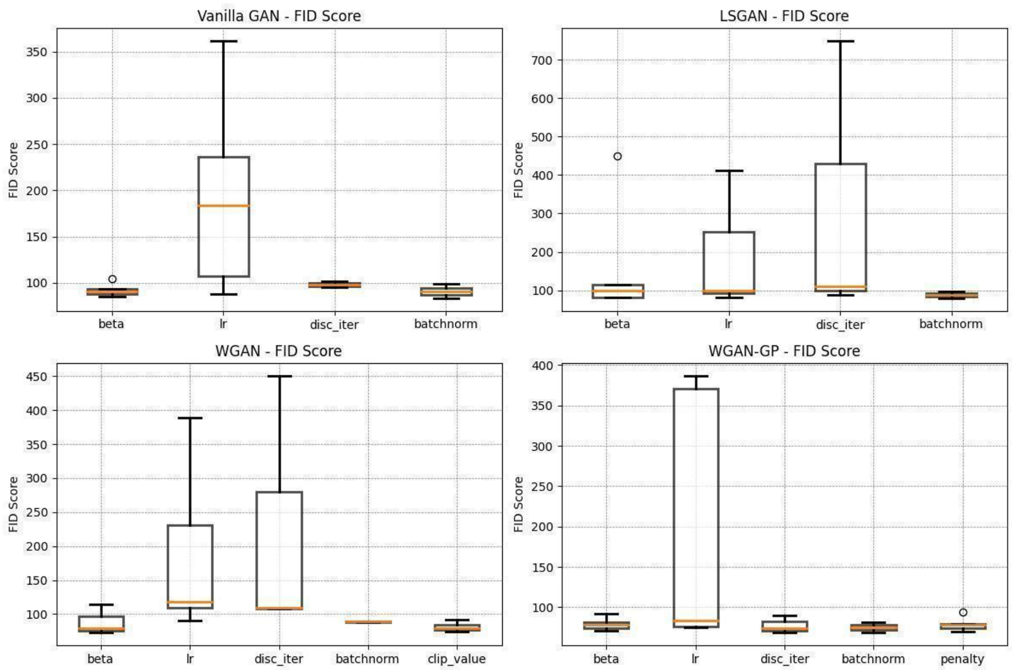

Figure 5: This plot shows how each parametric change affects the FID score per each type of GAN. It is obvious that the most impact has the learning rate change.

	Best FID	Mean FID	Worst FID
Vanilla GAN	83.73	127.65	361.19
LSGAN	80.13	192.67	748.37
WGAN	72.86	131.48	450.62
WGAN GP	68.67	108.31	386.74

Table 4: Best, worst and mean FID scores achieved by different types of GANs.

5 Conclusion and Recommendations

5.1 Summary of Findings

In this research work, we have presented a discussion on how to neutrally and fairly compare Generative Adversarial Networks (GANs). We have focused on two sets of evaluation metrics which are the Fréchet Inception Distance (FID), and precision, recall, and F1 score.

We provide empirical evidence that FID is a reasonable metric due to its robustness.

We highlight that algorithmic differences in state-of-the-art GANs become less relevant as the computational budget increases. Furthermore, we point out that given a limited budget, a "good" algorithm might be outperformed by a "bad" algorithm.

We also acknowledge that there are many dimensions to consider for model comparison, and this work only explores a subset of the options. We cannot exclude the possibility that some models significantly outperform others under currently unexplored conditions.

Overall, this work strongly suggests that future GAN research should take into account the computational resources required for training and use a more comprehensive approach for model comparison that includes multiple evaluation metrics.

5.2 Recommendations for Future Research

Based on our findings, there are several recommendations for future research:

1. Explore additional evaluation metrics: While FID and precision/recall/F1 are useful metrics, there may be other metrics that are relevant for evaluating GANs. Future research could explore other metrics that may provide additional insights into the performance of GANs.
2. Increase the diversity of datasets: The study primarily focused on evaluating GANs using the MNIST dataset. Future research could expand on this by including a wider range of datasets to test the robustness of FID and other evaluation metrics across different domains.

3. Think about other architectures: The study concentrated on a particular group of GAN architectures. Other architectures, like variational autoencoder-based GANs or autoencoder-based GANs, could be studied in the future..
4. Explore the impact of more hyperparameters: Future research could investigate the impact of hyperparameters such as , batch size, and number of epochs, regularization on GANs performance.
5. Investigate the impact of different optimization techniques: The study focused on the Adam optimizer. Future research could explore the performance of GANs using other optimization techniques such as SGD or RMSprop.
6. Analyze the effects of various types of noise: The study concentrated on Gaussian noise. Future studies might examine how different types of noise, including uniform noise or Poisson noise, affect GAN performance.
7. Examine the effect of conditioning: The study did not look at the effect of conditioning on the performance of GANs. Future studies might examine how well GANs function with various types of conditioning, such text or image conditioning.

In general, there are numerous potential directions for future study in the area of GANs. We can learn more about the advantages and disadvantages of GANs and continue to enhance their performance by investigating these areas.

6 References

6.1 List of sources cited in the thesis

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 5907-5915).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139-144.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Jarrett, K., Kavukcuoglu, K., & LeCun, Y. (2009). Computer Vision, 2009 IEEE 12th International Conference on.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139-144.

Ledig, Christian, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken et al. "Photo-realistic single image super-resolution using a generative adversarial network." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681-4690. 2017.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214-223). PMLR.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Tao, D., Huang, L., Lang, B., Liu, X., & Liu, Y. (2017). Centered Weight Normalization in Accelerating Training of Deep Neural Networks.

Yang, Y., Zhong, Z., Shen, T., & Lin, Z. (2018). Convolutional neural networks with alternately updated clique. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2413-2422).

Akata, Z., Malinowski, M., Fritz, M., & Schiele, B. (2016). Multi-cue zero-shot learning with strong supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 59-68).

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, *30*.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Lee, C. W., Fang, W., Yeh, C. K., & Wang, Y. C. F. (2018). Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1576-1585).

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, *30*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, *29*.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards deep learning models resistant to adversarial attacks." *arXiv preprint arXiv:1706.06083* (2017).

Barratt, S., & Sharma, R. (2018). A note on the inception score. *arXiv preprint arXiv:1801.01973*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, *30*.

Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., & Gelly, S. (2018). Assessing generative models via precision and recall. *Advances in neural information processing systems*, *31*.

Gong, X., Chang, S., Jiang, Y., & Wang, Z. (2019). Autogan: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 3224-3234).

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., & Aila, T. (2019). Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, *32*.

PyTorch. (n.d.). PyTorch. GitHub Repository. Retrieved from https://github.com/pytorch/pytorch

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1125-1134).

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 2223-2232).

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. In Proceedings of the 6th International Conference on Learning Representations (ICLR).

Karras, T., Laine, S., & Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 4401-4410).

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-Attention Generative Adversarial Networks. In Proceedings of the International Conference on Machine Learning (ICML) (pp. 7354-7363).

Appendix A

Additional plots

In this appendix, we provide the detailed results of the comparative analysis.

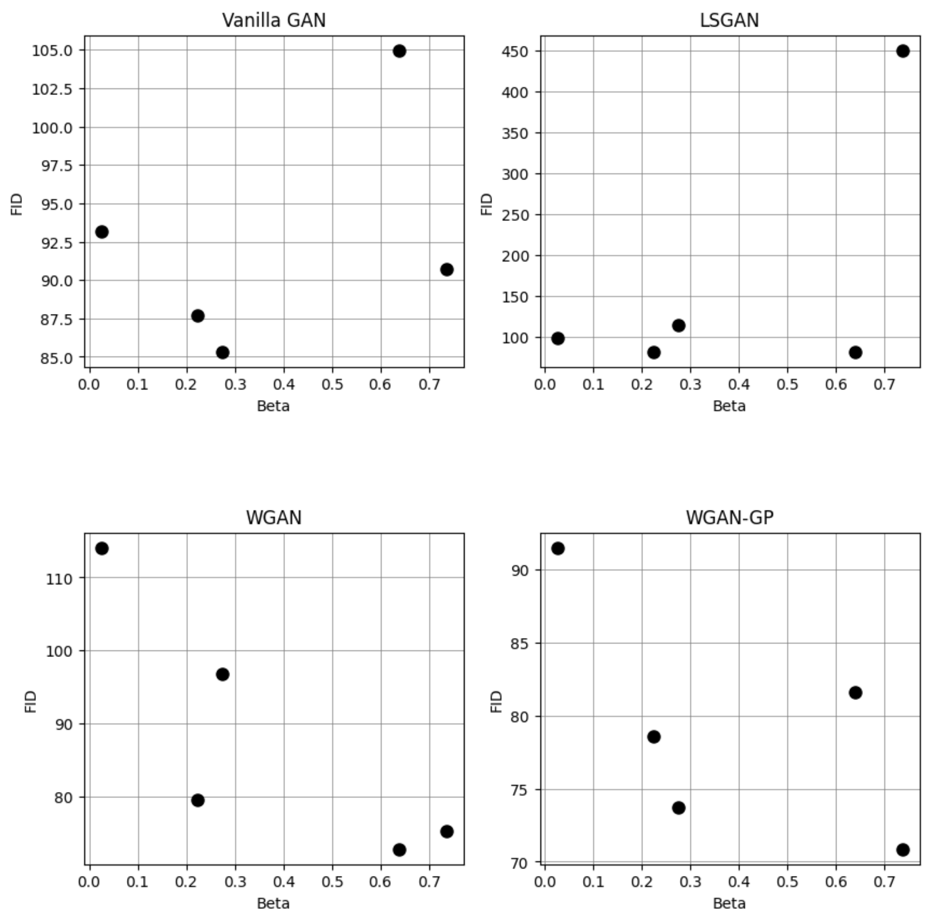

Figure 6: Scatter plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. Adam's β1 parameter.

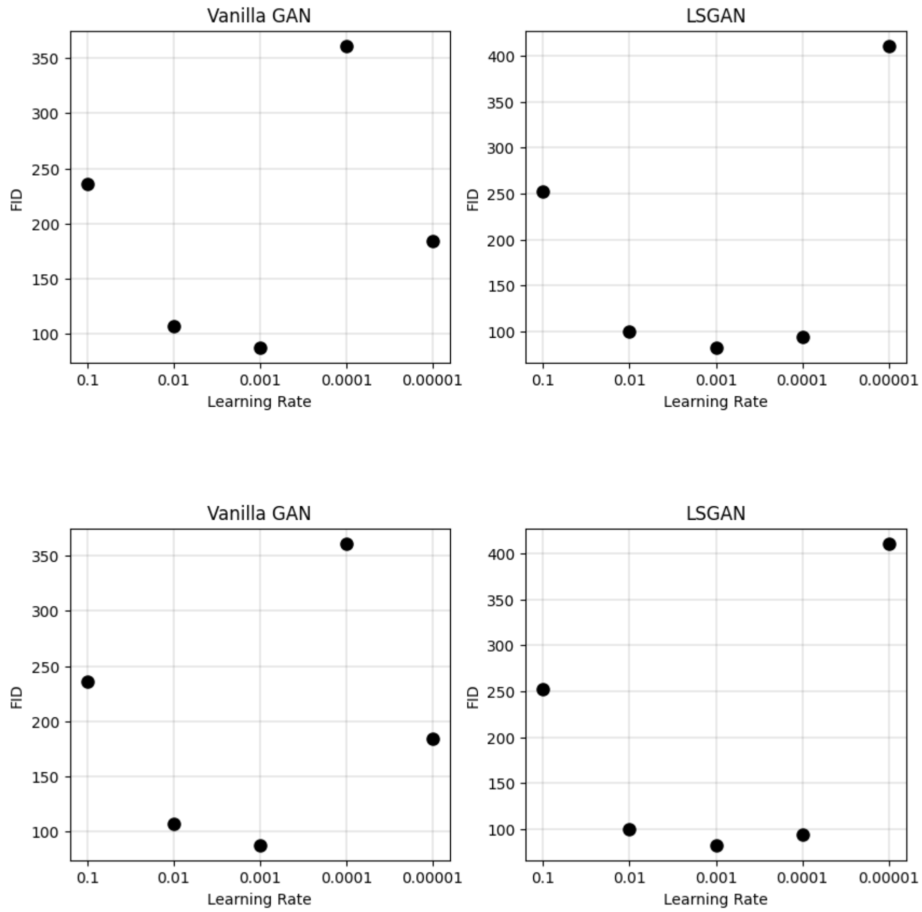

Figure 7: Scatter plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. learning rate.

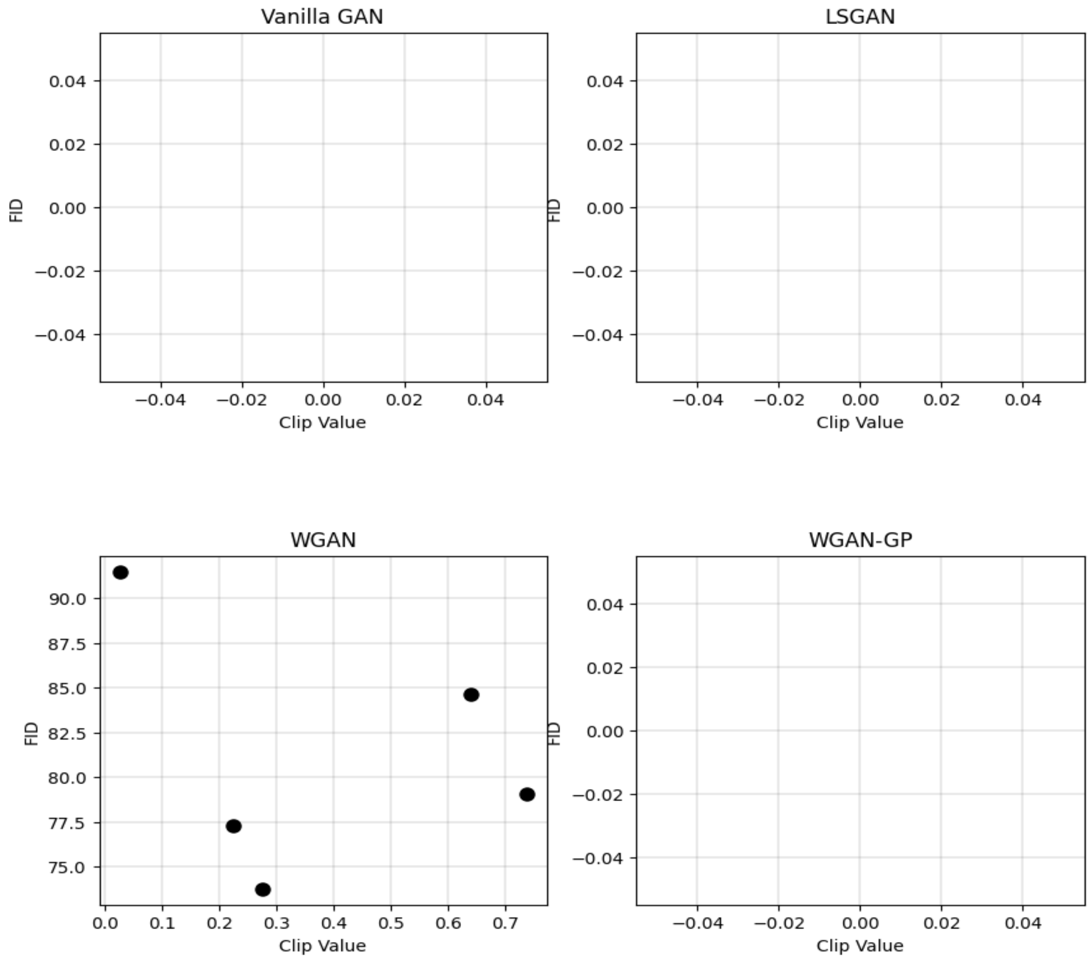

Figure 8: Scatter plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. clip value.

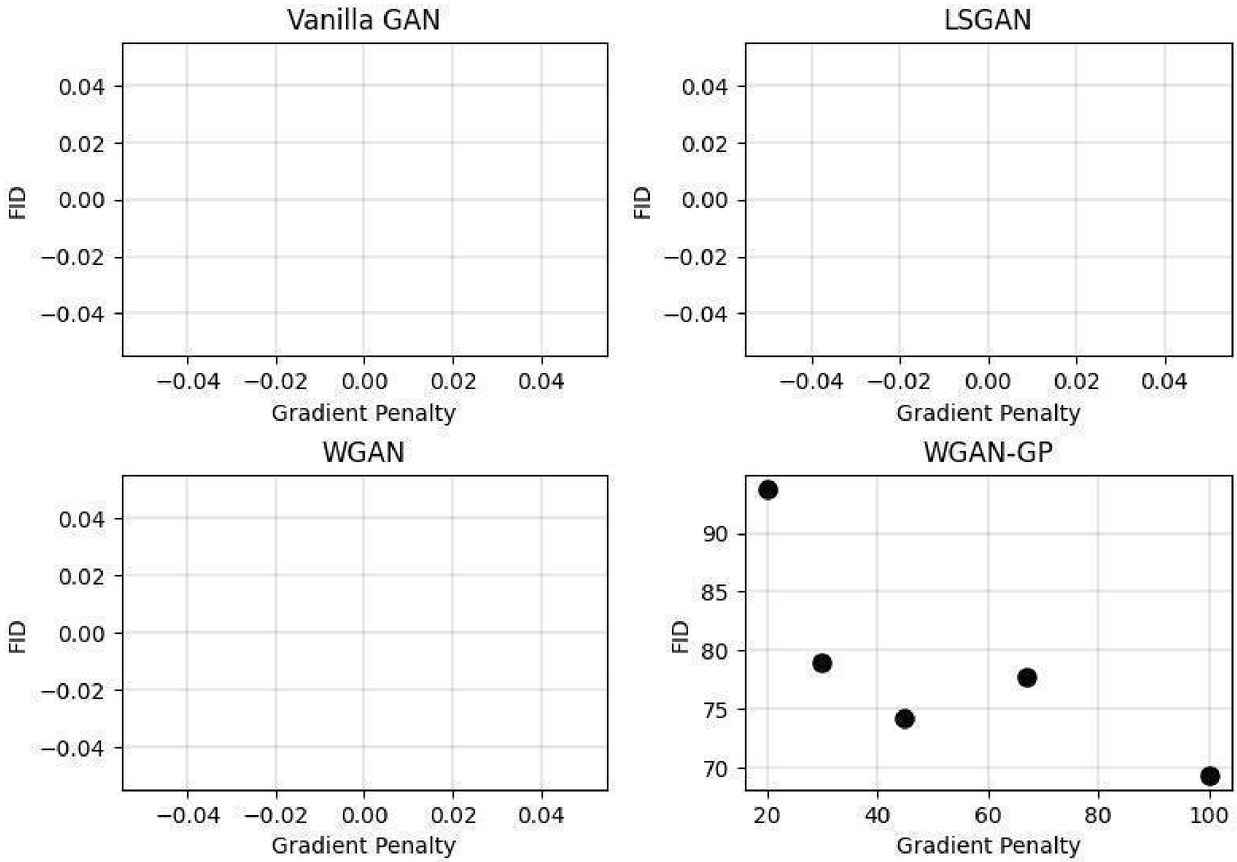

Figure 9: Scatter plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. gradient penalty.

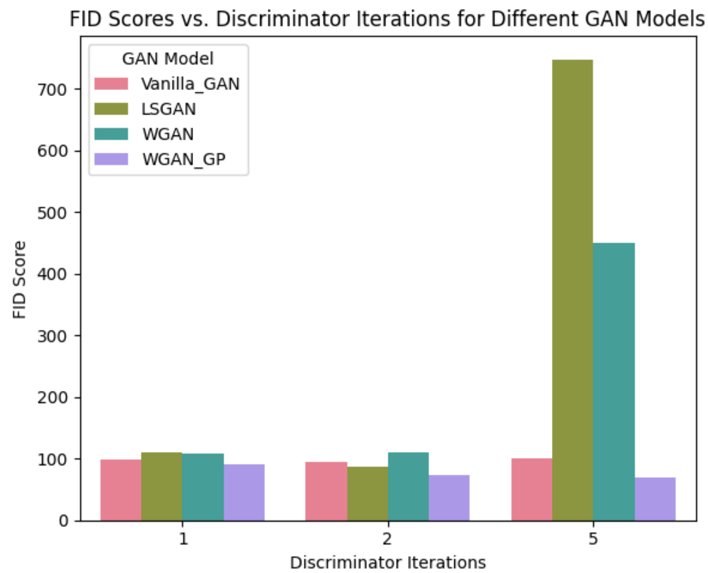

Figure 10: Bar plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. discriminator iterations.

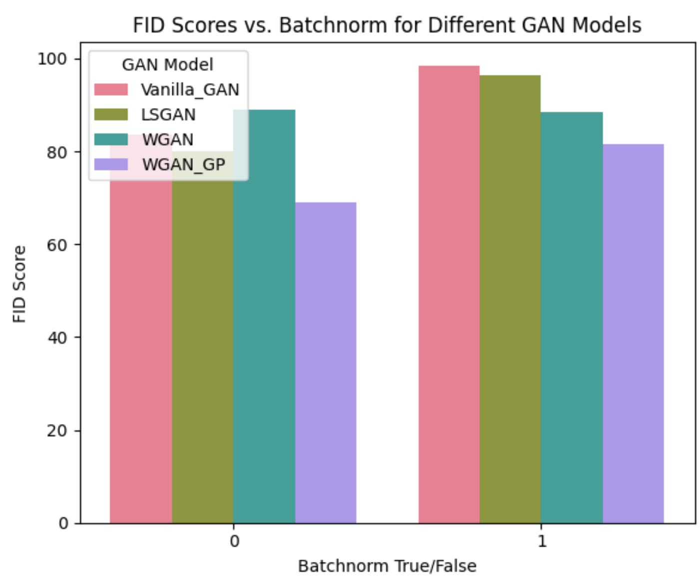

Figure 11:Bar plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. batch normalization.

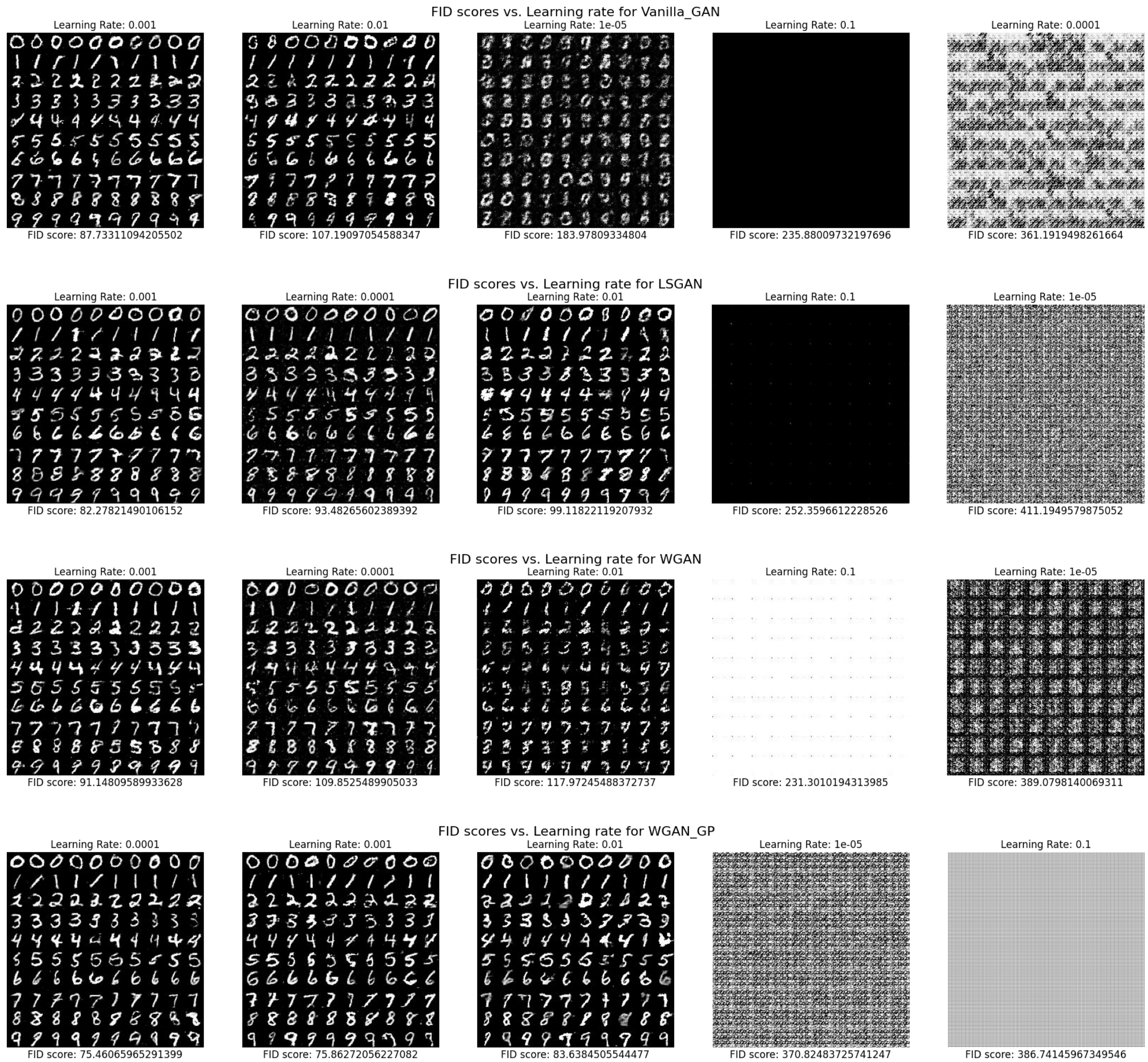

Figure 12: Generated images and their corresponding FID scores when changing the learning rate. The FID scores are plotted in ascending order. It illustrates that FID score is correlated with human evaluation as those images with high FID scores have less understandable generated digits. This also shows that the model is highly sensitive to the learning rate changes.

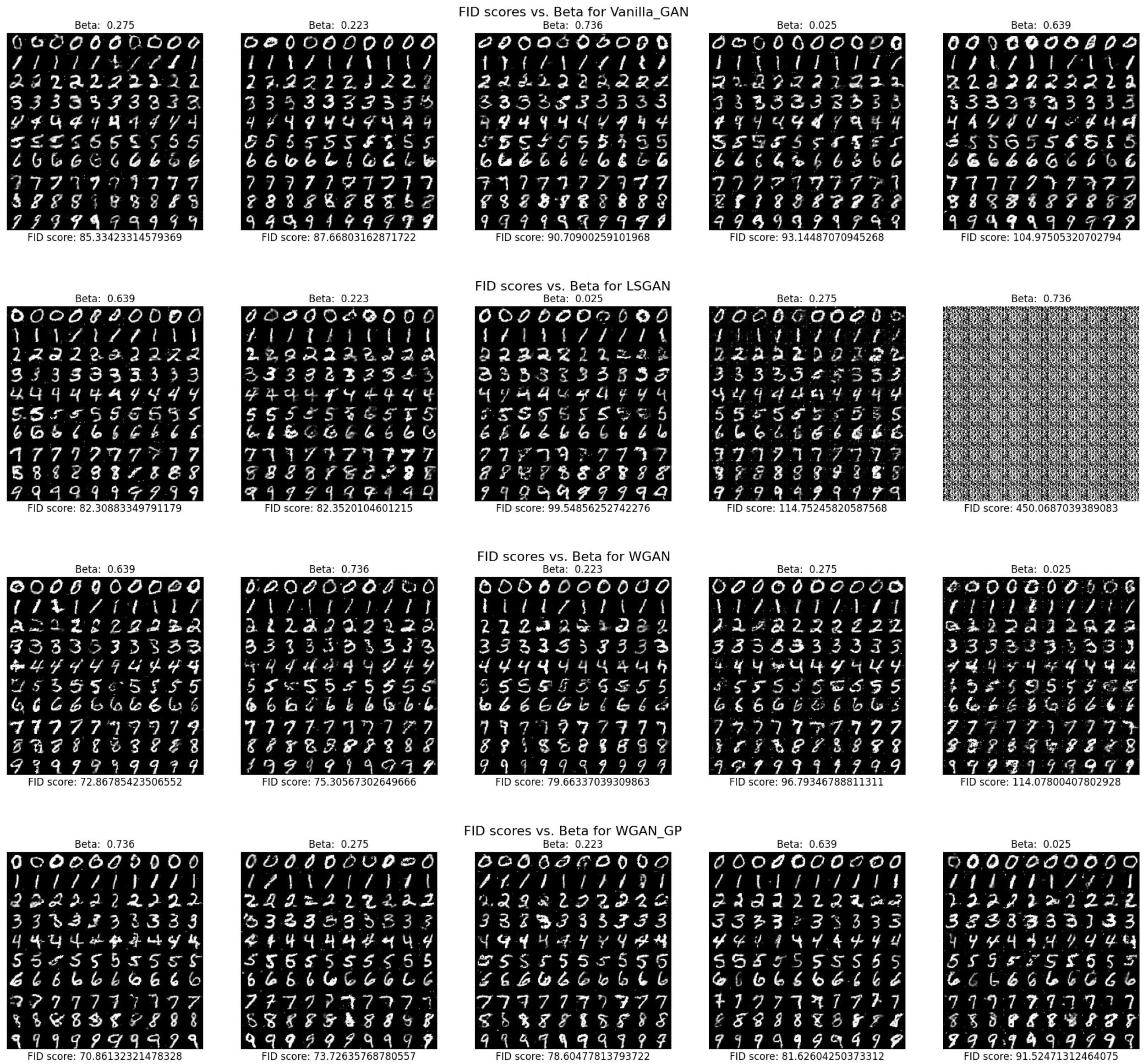

Figure 13: Generated images and their corresponding FID scores when changing the Adam's β. The FID scores are plotted in ascending order. It illustrates that FID score is correlated with human evaluation as those images with high FID scores have less understandable generated digits. This also shows that the model is highly sensitive to the β changes.

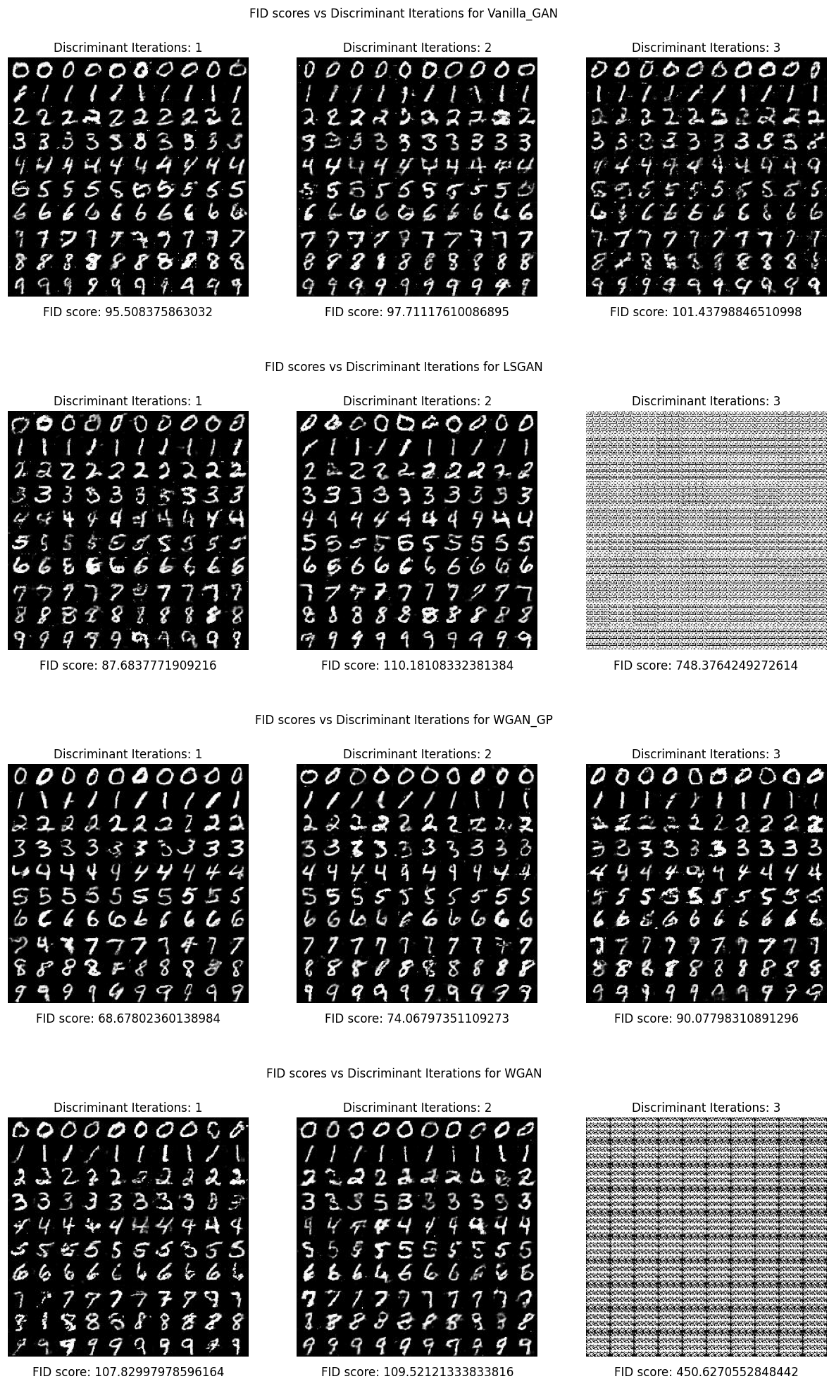

Figure 14: Generated images and their corresponding FID scores when changing the number of discriminator iterations. The FID scores are plotted in ascending order. It illustrates that FID score is correlated with human evaluation as those images with high FID scores have less understandable

generated digits. This also shows that the model is highly sensitive to the number of discriminator iteration changes.

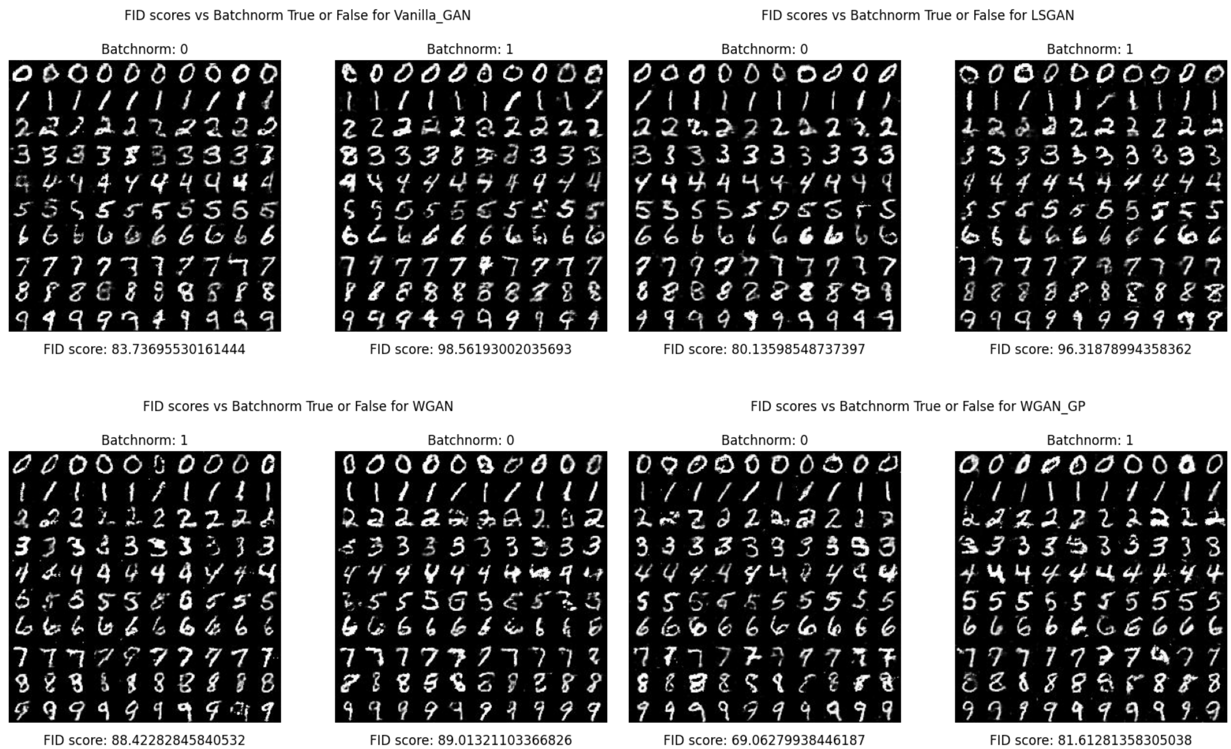

Figure 15: Generated images and their corresponding FID scores when changing the batch normalization to True or False. The FID scores are plotted in ascending order. It illustrates that FID score is correlated with human evaluation as those images with high FID scores have less understandable generated digits. This also shows that the model is highly sensitive to the batch normalization setting changes.

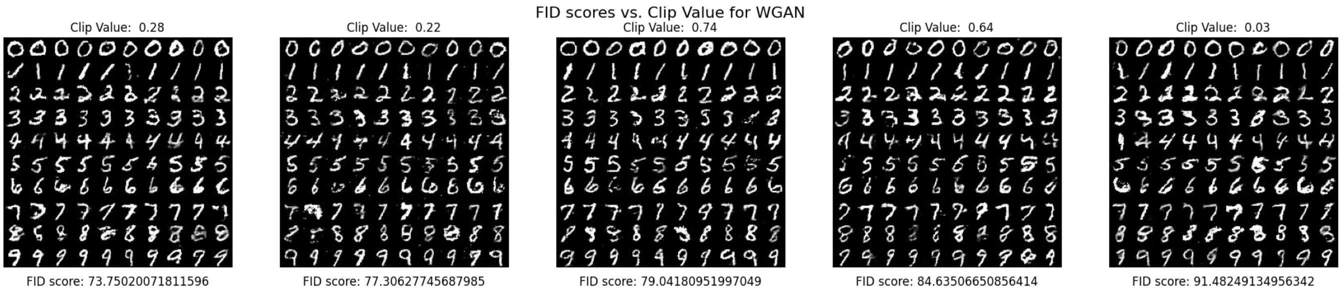

Figure 16: Generated images and their corresponding FID scores when changing the clip value for WGAN GP. The FID scores are plotted in ascending order. It illustrates that FID score is correlated with human evaluation as those images with high FID scores have less understandable generated digits. This also shows that the model is highly sensitive to the clip value changes.

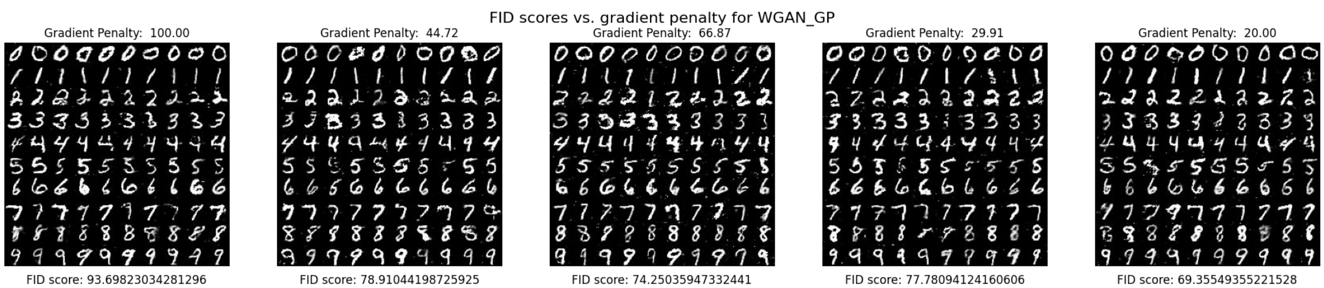

Figure 17: Generated images and their corresponding FID scores when changing the gradient penalty for WGAN. The FID scores are plotted in ascending order. It illustrates that FID score is correlated with human evaluation as those images with high FID scores have less understandable generated digits This also shows that the model is highly sensitive to the clip value changes.

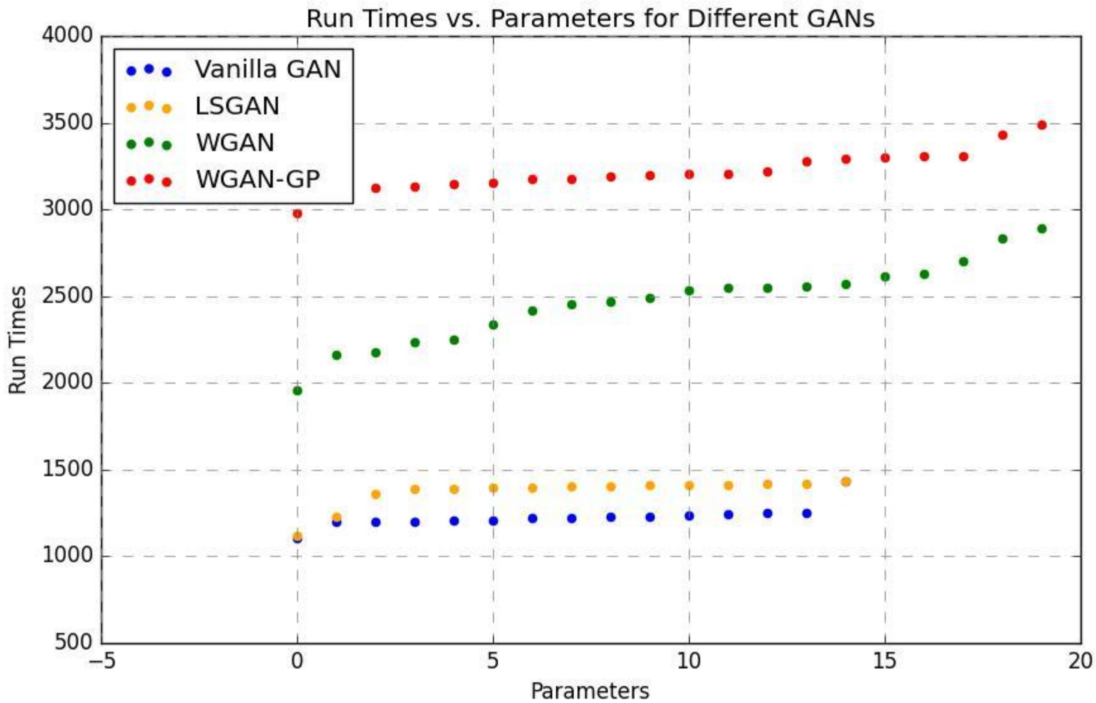

Figure 18: The plot shows the distribution of running times when changing the parameters.

Appendix B

Code Availability

A public GitHub repository contains all of the code used in this thesis for the GANs implementations, evaluation, results visualization, and multiple additional tasks. The repository is organized to facilitate easy navigation and understanding of the implemented algorithms and experiments. Researchers and practitioners are welcome to freely access, use, and alter the code for their own experiment..

The link of the repository:

https://github.com/anahitxachatryan/ComparativeAnalysisOnGANs-Thesis