ParityGAN: Class Balance with Hierarchical Temporal Memory

by

Kazi Rafiul Kader
21301381
Nazmus Sakib Bhuyan
21201402
Rameezah Rahman Yeasha
21201505

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
October 2025.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Kazi Rafiul Kader

Student Id: 21301381

Nazmus Sakib Bhuyan

Student Id: 21201402

Rameezah Rahman Yeasha

Student Id: 21201505

Approval

The thesis/project titled "ParityGAN: Class Balance with Hierarchical Temporal Memory" submitted by

1. Kazi Rafiul Kader (21301381)
2. Nazmus Sakib Bhuyan (21201402)
3. Rameezah Rahman Yeasha (21201505)

of Summer, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in 2025.

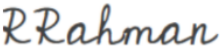
Examining Committee:

Supervisor:
(Member)

Mr. Md. Tanzim Reza

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)



Mr. Rafeed Rahman

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Contents

Abstract

Generative Adversarial Networks are efficient image generation models. They are still better performers in specific domains such as producing high resolution synthetic image data, medical imaging, image to image translation. GAN's have some limitations such as, bias towards majority classes, lack of fairness constraint, mode collapse and class imbalance. To address these problems, We have introduced ParityGAN, a hybrid GAN architecture that deals with the existing limitations of GAN's in class distribution and fairness. Our proposed model's generator is the combined form of conditional GAN to ensure label aware generation, ResNet based DCGAN for high quality synthesis and spectral normalisation SNGAN for training stability. We have also introduced a novel approach that is TBM (temporal balanced memory) and updated version of it named ETBM (enhanced temporal balanced memory). ETBM monitors and adjusts class distribution through short, medium and long term memory. Our distributional fairness discriminator helps the generator to achieve fair class distribution and at the same time provide ETBM with class probabilities. We trained the model on an imbalanced CIFAR-10 dataset with a total of 45.9M parameters and achieved a FID score of 18.11. Our ParityGAN demonstrated that achieving fairness or reducing bias does not require sacrificing generation diversity.

Keywords: Generative model, Bias, Generative Adversarial Networks, Bayesian Optimization, ResNet, spectral normalisation, TBM, ETBM, Discriminator, DFD

Chapter 1

Introduction

1.1 Background

The use of Artificial Intelligence is expanding rapidly in today's world. Generative Artificial Intelligence added new dimensions. These AI models can generate text, image, audio and video. Their accuracy is also increasing day by day but they often encounter fairness bias related issues. A recent study conducted by Zhou et al. (2024) [30] showed that image generating models such as Midjourney, Stable Diffusion and DALL-E 2 showed 23%, 35% and 42% bias against women while generating images. These biases stand as a barrier between fairness in class distribution. Another study by Kirk et al. [21] (2021) showed large language models like GPT-2 predicted 18% of US Hispanic women and 11% American women prefer waitressing and nursing as a job but the reality is 3% and 4%. Their proposed framework helps to measure fairness violations in LLM's. Another study [28] showed biased behaviour of StyleGAN3 when it was trained in FFHQ dataset and it generated 74% white people. Bias creates difficulties when we try to use the output of generative models in our real life scenarios. There are many ways to mitigate these biases. For example, Wu et al [24] used an Energy Based Model with an Invertible Neural Network to mitigate bias without changing the weights of pre-trained models. In some paper [17] we saw that weak supervision is used to reduce bias. A different way of ensuring fairness can be minimizing mode collapse proposed in [7]. It helps to generate diverse outputs and avoid redundancy. Our main target is to minimize bias and ensure fairness and to do so we propose a hybrid model built on GAN which has one generator, two discriminators, one Enhanced Temporal Balance Memory. Combination of these elements will help to deeply understand bias and fairness related issues and ensure high quality image generation.

1.2 Rational of the Study or Motivation

Image generation is an important part of Computer vision. Good image generation models can reduce dataset collection cost to some level. Existing models like VAEs (Variational Auto Encoder) tend to generate blurry images and Diffusion models are very expensive as they hold complex architecture. That is why we choose GAN (Generative Adversarial Network) as our research sector. GAN's can produce comparatively sharp quality images in one forward pass and their architectures are simple compared to others and they are not expensive as diffusion models. They

can be deployed in small devices , they are faster, good in medical imaging , image transitions etc. Despite these capabilities, GANs sometimes express training dataset bias present in dataset and failed to achieve good class distribution on imbalance dataset. That is why we chose to work with GAN's and tried to develop a hybrid model "ParityGAN".

1.3 Problem Statement

Generative models like GAN[3], FairGAN[10] etc have brought extraordinary success in the field of generating realistic image synthesis. Nevertheless, biases are still present among them in multiple ways. Either some of these can not produce fair representation of data enough or some of these cannot diversify in case of generating image synthesis. First of all, if we talk about FairGAN[10] which has one generator and two discriminators, we can see that it can not diversify enough in case of generating data although it can enhance fairness with the help of taking feedback from the discriminators continuously (one ensures how real the data is and simultaneously the another one confirms where the generated samples are coming from; protected group or unprotected group. To solve the problem of mode collapse, FairGAN+ [16] has been designed which has one generator with three discriminators. It can solve the issue of mode collapse but if its focus is given strictly on fairness then FairGAN+ faces mode collapse again. Protected attributes are solely learnt by the generator and only one mode is produced hence mode collapse [16]. Furthermore, if we talk about another generative model, VEEGAN, we can see that it solved the issue of mode collapse successfully but did not work enough on increasing fairness data representation directly [7]. That is why for solving the limitations that GAN's [3] face and to increase the fairness and reduce bias and also to achieve balanced class distribution in generative models, we are introducing a hybrid model ParityGAN.

1.4 Objective

Based on Generative Adversarial Network (GAN) [3], many generative models are designed. Some models focus on solely creating realistic images like StyleGAN3 [20] and some like FairGAN [18] focus on debiasing and increasing fairness. Objectives of our work are: To develop a conditional Generative Adversarial Network named as 'ParityGAN'. To train the model using the existing balanced or imbalanced datasets. Try to develop an efficient class conditional generator powerful enough to tackle two discriminators that we are planning to implement. Discriminators will optimize image quality and improve fairness in class distribution. Use Temporal Balanced Memory to track class distribution and generation quality across multiple time scales. To analyze scenarios where quality and fairness trade-off occurs through our proposed model. For evaluation, we will use state of the art evaluation metrics.

1.5 Methodology in Brief

Our proposed ParityGAN will follow a pipeline starting with collecting a dataset which will go through some processing. Later processed data will be the input of our

designed generator. We are planning to keep two discriminators. The first one will distinguish real and fake images and the second one will analyze class distribution for fairness. To track the class distribution we will use temporal balance memory. As we are planning to use the CIFAR-10 dataset we will train our model for about 200 epochs and then apply necessary tuning to observe results. After getting the results we will analyze using some evaluation metrics such as FID score, Inception score, Kl divergence.

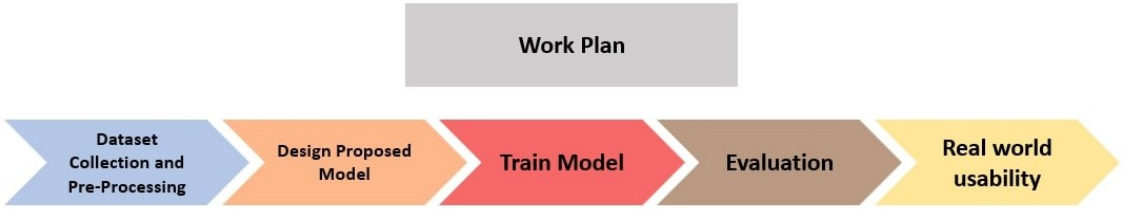


Figure 1.1: Work Plan

1.6 Scopes and Challenges

In this paper, our main focus is to ensure demographic parity in Generative Adversarial Networks. Working with GAN's is very challenging. To train our proposed hybrid model we need real world imbalanced data in a large number, which seems a bit challenging right now. We will introduce two discriminators which can increase the overall complexity of the model. Our Temporal Balance Memory integration will increase the difficulty level. We have to deal with issues like mode collapse, gradient vanishing, exploding gradient etc. To evaluate fairness we may have to go through multiple metrics. One might miss some important key points. We have to generate high quality images with distributional fairness. The overall combination may raise some unknown difficulties which we will figure out in the long run. Despite these challenges our GAN architecture will provide some valuable insights.

Chapter 2

Literature Review

2.1 Preliminaries

In the space of generative models, a new approach was the adversarial approach to generative models which introduced us to the concept of generator and discriminator. Adversarial approach is when the generative model is trying to create more realistic fakes and the discriminator model is trying to identify the fakes from the real data sample and send feedback according to it to the generative model and improving itself also. This adversarial game made generative adversarial networks. The significant change is the adversarial nets framework. Putting the generative and discriminative models against each other improved their efficiency for creating more realistic fake generated data distribution. Eliminating the need for Markov chains and using Backpropagation for training the data and for producing new synthetic data [3].

When we train the generative models we use datasets and with the models we train the data, bias can be present in both stages. When the dataset or the model is prioritizing a certain sample space of the dataset and the model's training is skewed to particular learning from the dataset we call this bias. Bias can occur in several cases such as while taking measurements of particular features, while some important variables are absent in the models, while collecting data, while designing algorithms (using particular functions or choices), while presenting information etc [22]. The lack of control in labeling the training data also introduces bias when training generative artificial intelligence models [30].

Fairness is when all the categories are prioritized or represented equally and any kind of social restrictions are not present while generating data and any kind of protected attributes does not influence the decision making process [22]. If we talk about measuring fairness, statistical metrics come up but even so, using this approach can not always be successful while capturing practical outcomes and may unintentionally cause harm to the protected groups in case of representation [26].

For capturing the real-world influence, reducing bias and increasing fairness many models are proposed. These models solve the issues of algorithmic problems like mode collapse and focus on bias reduction like VEEGAN [7], FairGAN [10], FairGAN+ [16] and many more are built on GAN [3].

2.2 Review of Existing Research

The knowledge of how the generative models make assumptions based on data has played a great role in bias reduction. These assumptions make generative models like GAN and VAE's more biased. For the insight, Zhao et al. (2018) [12] highlighted how GANs and VAEs architectures handle features that they have not seen before and the influence of inductive bias. In this paper they worked on a training dataset D and model q(x) tried to estimate realistic data p(x). The process is difficult as dataset D is comparatively smaller and image data are complex with many attributes. Inductive bias took place in order to produce p(x) from D. the Datasets used here are CLEVR, colored dots, MNIST, pie. In generalization behaviour of multimode, when modes are close together the model generates objects close to their average value. Convolution prediction does not match q(z). When they are far apart, the model generates distinct objects. It provides a clear view of how generative models make assumptions based on data.

Before we had to introduce human supervision to reduce bias in the models or the datasets and their weights but if we imagine to have the same output without human supervision then Frankel and Vendrow et al. (2020) [18] 's proposed method is the solution which is with the help of prior modification. Here the dataset of MNIST, FFHQ and CelebA-HQ (publicly available) is used. For calculating the feature frequencies they used "feature recognition network". We can see the feature frequencies (recognized using a trained MoblieNetV2) in StyleGAN and ProGAN generated images and see that features like e.g. Young, High_Cheekbones and Male are most prominent. These features are also prominent in the CelebA dataset. That proves the bias in the produced data by trained models of StyleGAN and ProGAN. They found that changing the mean of gaussian prior can achieve a desired fair distribution of features and classes in images. Moreover, they came up with the use of small neural networks in which latent variables are perturbed slightly. This latent perturbation makes sure to get desired fairness from the generative model keeping the quality and variability intact. They did their experiment in three steps. They started with a MNIST dataset that is not inherently more difficult with features of human faces. At first, they trained DCGAN that can generate 0 an 1. Without the latent perturb input they had unequal distribution but after the input the distribution was equal. Experiment 2 was training the DCGAN 10 digits. But after training, the 0 and 1 generated output frequency is higher. But the latent model produced an equal distribution. Slight modification in latent vector achieves desired distribution. Other than that, batch normalization in DCGAN helps when training data is poorly initialized. Also helps the gradient flow hence stabilizing the learning [5]. In experiment 3, for feature labeling they used MobileNetV2. Here they used ProGAN model with CelebA dataset. Before combining latent models they had unequal distribution but after 49.2% images were labeled as female. This proves that precise connection between label and latent areas can reduce bias.

We learnt before training, how fairness can be enhanced by modifying prior distribution. Now we will get introduced to how fairness can be enhanced during data preparation. Choi et al. [17] introduced a solution of bias reduction through weak supervision without needing labeled data since they learnt that acquiring unlabelled

datasets is comparatively cheap in this era of data. They proposed to augment their biased dataset to the unbiased reference dataset. First approach for bias mitigation is not optimal because training whole on the Dref or Dbias respectively either give poor quality samples or it will be unfair but produce high quality samples. As a new solution they introduced Importance Reweighting. This adds more weight to the data points that are less represented from Dref and reduces weight from data points that are more represented from Dbias. To estimate density ratio they use a binary classifier and for parameterizing deep neural networks are used. The training of the binary classifier is crucial here. For evaluation they used the CelebA dataset to train BigGAN. Furthermore, they constructed three settings for dividing the dataset into Dref and Dbias using 40 labeled binary attributes from CelebA dataset. Here 0 classifies as Female and 1 classifies as Male. In the setting 1, 2 respectively the bias for classification 0 is 0.9, 0.8. But for settings 3 there are 2 bias variables e.g. "gender" and "black hair", creating multiple sub groups for each class so the bias is distributed unevenly among 4 subgroups. Thus they were able to compare the cross-entropy loss of the Bayes optimal classifier and density ratio classifier and finally evaluated the estimated density ratios' quality. And at last, they achieved 36.6% reduction in single-attribute dataset bias and 32% on multi-attribute dataset bias.

Datasets are limited in diversity, indirect labeling techniques [17] might not be useful in some cases but we can adapt model knowledge by related sources. By introducing GAN, Goodfellow et al.(2014) [3] set a new dimension in image generations. GAN has two parts, one generator and one discriminator. Generator generates fake data and the discriminator tries to differentiate between fake and real. Wang et al.(2018) [9] investigated transfer learning for GANs. In this paper they showed how pre-trained GANs generate images with limited data and lower FID score. They used pre-trained unconditional GAN to build a conditional GAN (cGAN).They used AC-GAN for class conditional image generation and found out that results of pre-trained GANS are more realistic. For the source datasets they used ImageNet, Places, LSUN Bedrooms, CelebA. And for target datasets they used Oxford Flowers, LSUN Kitchens, Label Faces in the Wild, Cityscapes. To stabilize training GANs they used Wasserstein GAN with Gradient Penalty (WGAN-GP) introduced by Gulrajani et al.(2017) [6]. They used Frechet Inception Distance (FID) and Independent Wasserstein Critic (IW) to evaluate image quality. The study helped to understand the transfer learning of GAN in small datasets.

Xu et al. (2018) [10]'s method is to pre-process the generator in FairGAN. FairGAN is trained on a synthetic dataset that is unbiased. Fair data has to meet four characteristics: Data Utility, Data Fairness, Classification Utility and Classification Fairness. FairGAN ensures these characteristics and trains the generator. For training UCI Adult income dataset is used which has 48,842 instances. Generative adversarial network has one generator and one discriminator [3]. But the FairGAN has one generator and two discriminators. Discriminators make sure that produced data is close to the real data. Here in FairGAN Risk Difference is used as a fairness metric. Risk Difference is 0 when the discrimination is not present even if we get positive or negative numbers for risk difference then bias or discrimination is present. For evaluating the effectiveness of fairness for real dataset, SYN1-GAN,

SYN2-NFGANI, SYN4-FairGAN risk difference is used. Comparing these 4 real and synthetic datasets we can see SYN1-GAN, SYN2-NFGANI are best performing but only because of protected attributes. Further evaluating balanced error rates (BERs) the disparate impact is small in SYN4-FairGAN [4]. This proves that FariGAN is effective in developing fair data without disparate treatment.

Having similarity to FairGAN [10], Sattigeri et al. (2019) [15] demonstrated a solution of reducing bias which is Fairness GAN. They tried to minimize bias through carefully generating datasets and used Auxiliary classifier GAN as a base for their work with some changes. They used C for protected attributes which are gender and caste. The datasets used in this paper are CelebA, Soccer, Quick,Draw!. They checked demographic parity and equality of opportunity in these datasets. Improvement is noticeable in all except Quick,Draw! Dataset where demographic parity value was not improved at all. The overall process was efficient to reduce bias.

A new type of approach Teo et al. (2022) [27] and the team introduced transfer learning approach fairTL and fairTL++. Firstly, in fairTL there are two steps: Pre training and adaptation. They used Dref and Dbias to pre-train a generative adversarial network model. Now after the general knowledge is achieved now in adaptation step, the GAN is trained using only Dref for fairness. Again the discriminator and generator is used and this is how the model is trained to be less biased. In generative adversarial networks if we do not update the discriminator to avoid any scenario e.g. Helvetica scenario and on the other hand Generator is being trained. This is how GAN is susceptible to mode collapse. Because Generator is depended on the discriminator's useful gradients [3]. Secondly, fairTL++ is proposed for this disadvantage of fairTL. They have added two methods: Linear-Probing before Fine-Tuning (LP-FT), and a multiple feedback approach. These are added in the adaptation phase. Pre-trained networks freezed by Linear-Probing for some epochs and then Fine-Tuning is used. In Fine-Tuning the entire weights are again used. For transfer learning this works better and another approach is Multiple Feedback. It's a well-known approach in the case of improving generated sample quality, especially when the samples are limited. For evaluating the performance they did two experiments where in the first one the model had Dbias and Dref but for the second one had access to a pre-Trained model and Dref. For these experiments they used CelebA and UTKFace. In setup 1, they trained a fair generator. All the available dataset was used to train BIGGAN here. The Dref size is 2.5% of all the data. Introduced synthetic bias in setup 1. Three methods are compared: Imp-wighting, fairTL and fairTL++. Single-Attribute experiments fairTL, fairTL++ performs better than imp-weighting. When we make the Dref very small (percentage = 0.0025) then also fairTL++ performance remains stable. Even in multi attributes (e.g. Gender and Blackhair) fairTL++ provides state-of-the-art results. Like in setup 1 they introduced a small dataset (UTKFace), fairTL++ shows advantages. Even if we have smaller Dref fairTL++ showed that it has an advantage over Imp-weighting and fairTL. In setup 2, they tried to debias a Pre-trained Generator. Utilizing the Style-GAN2 and FFHQ dataset weights there is bias present. After FairTL the FD and FID values decrease. And, FairTL++ is more efficient than even fairTL. They proved that FairTL and FairTL++ provide state-of-the-art performance.

In our dataset we can have biases for any demographic group. Demographic groups share the same characteristics. We want to concentrate on the features that are important for the representation of the output variable instead of protected variables of demographic groups. Zhang et al. (2018) [11]'s approach is to mitigate bias coming from the protected variables, by proposing a method that uses fairness measurements like Equality of Odds, Demographic Parity, and also Equality of Opportunity. Adversarial debiasing is used for fairness measurement. Here the predictor is X and generated output is Y and the adversary predicts Z. Firstly, in case of demographic parity, Y(the predicted label) is the input to the adversary for which the adversary becomes able to estimate the protected variable based on Y. Secondly, in case of equality of odds, two inputs are for the adversary which are Y (the predicted label) and Y (the true label). Finally, in case of equality of opportunity, the adversary's training set is limited to the training example(where Y=y3) on a particular class y. UCI Adult (Census) Dataset is used for classification tasks. In the UCI Adult dataset it is visible that after the application the accuracy is affected by a small margin (86.0% vs 84.5%). In the end we can see in the confusion matrix they achieved equality of odds because the FPR and FNR are balanced. FRP(Female) increases from 0.0248 to 0.0647 and FPR(male) decreases from 0.0917 to 0.0701. Representing a balance across gender groups. No drastic increase or decrease in FNR. Identifying a group's True Positive is important. So, this ability of the model is unchanged. This adversary works even with the complexities present in the model.

Building on the stabilized GAN, Gulrajani et al.(2017) [6] showed an updated framework of Wasserstein GAN (WGAN). They proposed a gradient penalty (WGAN-GP) instead of traditional weight clipping. Weight clipping creates issues such as capacity underuse,exploding and vanishing gradients. The penalty prevents sudden changes in the critic's output and keeps it steady according to inputs. The datasets that they have used are CIFAR-10, LSUN Bedrooms and Toy Datasets. Here Gradient penalty is used to impose the Lipschitz constraint in place of weight clipping, alternative approaches like Lipschitz-cinstrained neural network (LCNN) using SpectralDense layers proposed by Tan et al. [29] in this paper can also secure stable training scenarios. The study [6] highlighted WGAN-GP as an effective way for training GANs.

Previously, we learnt how Generative Adversarial networks(GAN) is prone to mode collapse [27] where the GAN is producing some specific modes. In the data distribution it focuses on a few modes. But in case of solving the issue of mode collapse, Srivastava et al. (2017) [7] proposed a reconstructor in the GAN model. By the implicit variational principle used they train the Generator and the Reconstructor in the GAN architecture. Reconstructor tries to map the true data to noise distribution and approximating the true data encourages the generator to map all true data from the noise distribution. This is their solution to mode collapse through their reconstructor. Their experiments suggest that it (VEEGAN) resolves the issue (mode collapse) at hand. This is experimented on Synthetic Dataset, Stacked MNIST and CIFAR-10. With the modes known measuring mode collapse is easier in synthetic dataset. VEEGAN has modes and percentage of high quality samples in 2D Ring (8, 52.9%), 2D Grid (24.6, 40%), 1200D synthetic (5.5, 28.9%) synthetic

datasets. VEEGAN has the greater numbers in comparison to GAN, ALI, Unrolled GAN. In the experiment of Stacked MNIST and CIFAR-10, the maximum modes can be achieved is 1000 and VEEGAN captures 150, inference via optimization measure (IvOM) is also lowest and Kullback-Leibler Divergence (KL) value is 2.5. This shows VEEGAN has lowest discrepancy and the best mode coverage. In comparison to the previously mentioned SOTA GAN methods the VEEGAN solves the mode collapse issue better and also generates good quality images.

Unlike previous approaches here, Grover et al. (2019) [13] proposed a post-processing technique using importance sampling. With the help of the binary classifier we get the differences between the real dataset and the generated data. The model can be calibrated using likelihood ratio, this changes the weight of the samples according to their importance. Here, they used Cifar -10 as their training dataset for PixelCNN++ and SNGAN, on the other hand Omniglot for training the DAGAN model. Monte Carlo method uses statistical approach to obtain approximate solutions to problems [1]. And here, Monte Carlo estimates are used to evaluate goodness-of-fit metrics. 10,000 samples were taken to estimate goodness-of-fit metrics. Here, the SIR approximation Importance Resampled Energy-Based Model is better than the self-normalized Likelihood-Free Importance Weighting (LFIW). It shows 23.35% and 13.48% bias reduction. But in the 2nd experiment for multi-class classification we do not see much bias reduction, only a 2.15% reduction because the generative model e.g. GAN does not perform well in multi-class classification. Also proved importance weighting is effective in reducing RMSE for Model-Based Off-Policy Evaluation (MBOPE). They showed a 50.25% percent decrease in RMSE in MuJoCo environments. This method of debiasing is proved to be effective in a variety of tasks.

Tan et al. (2020) [19] research shows that GAN's original distribution is biased. And that GAN can be more biased than the original biased dataset. They proposed a method that they can reduce the bias in a pre-trained GAN model. This proposed method is called Latent Distribution Shifting (LDS). In the latent space of a GAN that is pre-trained, we have interpretable semantic dimensions, LDS finds these and uses these latent features to create a fairer output. From the original distribution, the Latent Codes or distribution for Male and Female are separated using various editing and filtering. But the latent codes have to pass through Gaussian Mixture Models (GMM). GMM uses Expectation-Maximization (EM) and Maximum A Posteriori (MAP) for learning the models settings from the data using these algorithms. Purpose of using GMM is, from a cluster of data points GMM is capable of identifying various patterns underlying those data points [2]. Then the output from the GMM's form the new Balanced Distribution. Then Balanced Distribution is fed to the generator. The generator produces fair sample data. Without retraining, Latent Distribution Shifting is capable of debiasing a pre-trained GAN.

McDuff et al. [14] proposed a racial and gender bias mitigation process of machine learning models. Their model improved facial and gender classifier accuracy. They used the MS-CELEB-1M dataset in this paper. Algorithms used here are Bayesian Optimization and conditional GAN. They produced images with racial attributes (such as Black, White, South Asian, Northeast Asian.) and gender. Using Bayesian

Optimization they explored generated images and tried to identify classifiers failures. To prove their GAN accuracy they used human evaluation, FID Scores. Their work helps with real world bias analysis.

Generative models like Generative Adversarial Networks (GAN) are trained in an unsupervised way of learning and easily their learning of the latent space distribution is biased from the biased dataset [3]. Wu et al. [24] and the team introduced the way of learning the latent space distribution without changing weights of pre-trained models. Their framework is termed as Generative Visual Prompt (PromptGen) which uses a feed-forward manner and enables users to control using an energy-based model (EBM) by an invertible neural network (INN). Comparing their PromptGen with two CLIP guided methods which are StyleCLIP and StyleGAN2-NADA they observed PromptGen produces diverse images that are not in the training distribution dataset of BigGAN. StyleGAN2 is trained on FFHQ and MetFaces. Based on text description PromptGen is capable of debiasing models which are trained on FFHQ and MetFaces. When they set the λ=2 the proportions for race and age are more debiased. Comparing the binary attributes with FFHQ, StyleGAN2, StyleFlow, FairGen, FairStyle they had the lowest value (lower is better). Also, achieving 100x times faster inference time than plug-and-play generative model (PPGM). In an iterative approach PromptGen is able to mitigate biases. These experiments show that without changing the weights of pre-trained models PromptGen is able to reduce bias.

2.3 Summary of Key Findings

According to the research papers based on the reduction of bias, we learnt that we can divide the bias reduction process into three categories. Three types are- Pre-processing, In-processing and Post-processing. Firstly, We learnt that input data or dataset etc. on which model will be trained on, might get biased and there are several ways to reduce bias from input data or datasets such as using transfer learning, by making some modifications in prior distribution or choosing densely sampled dataset instead of a very big diverse dataset which performs better in reducing bias or increasing fairness and this is considered as pre-processing bias reduction. Secondly, biases can be introduced during training a model and reducing this bias is referred to as in-processing such as using two discriminators with a generator, using weak supervision with the unlabeled data, with the help of gradient penalty (WGAN-GP) instead of traditional weight clipping etc. Finally, biases still can be presented even after training a model. But retraining the model or doing the whole process from the very beginning is computationally difficult and costly. So, removing bias after training a model is considered as the post-processing method. It includes methods like using post-hoc modification, putting weights into generated data according to how close it is to the real data and addressing classifiers failure and increasing accuracy of gender and facial classifiers in order to mitigate racial and gender bias.

Chapter 3

Proposed Methodology

3.1 Methodology Overview and Design (Model) Specification

ParityGAN is a custom GAN or Generative Adversarial Network that is created for providing samples and these samples are both realistic and uniformly distributed across various classes at the same time. It intends to resolve the mode collapse problem or imbalance in classes in GANs meaning the problem when the generator gets biased and generates outputs skewed toward particular dominant categories and does not focus on other classes. ParityGAN consists of several major components such as Enhanced DCGenerator, Discriminator, Distributional Fairness Discriminator (DFD) and Enhanced Temporal Balanced Memory (ETBM). ParityGAN does certain tasks. To start with, the generator generates realistic images and overtime it comes to learn deceiving the discriminator and on the other side, the discriminator keeps trying to differentiate the realistic image among the fake images and that is how ParityGAN does the adversarial learning. ParityGAN plays a vital role in ensuring fairness since with the help of ETBM it ensures equal distribution of classes of images. Moreover, it tracks class distribution with the help of ETBM avoiding sudden and abrupt drift. With the help of ETBM, it prevents the generator from getting biased or focused on particular classes. Furthermore, it enables the generator to be focused on both realism and balance with the help of the synergy of DFD and ETBM.

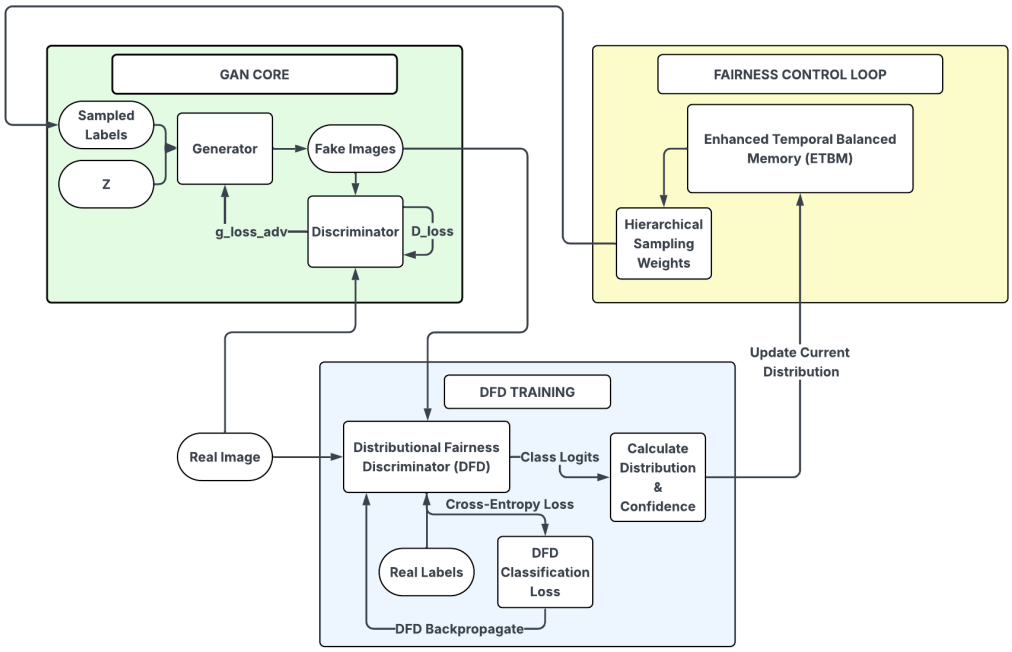


Figure 3.1: Methodology

The synthesis of high-fidelity, class-conditional images presents a significant challenge, often hampered by training instabilities inherent in generative adversarial networks. To address this, we propose a conditional, ResNet-based generator architecture designed for both robust training and detailed image synthesis. Our model's process begins by mapping a latent vector z and a class label y into a shared, low-resolution feature space. This initial representation is then fed into a sequence of specialized residual blocks that progressively upsample the features, increasing spatial resolution while refining the details of the emerging image. A central component of our design is the method of conditioning; instead of providing the class label only at the input, we employ Conditional Batch Normalization (CBN) throughout the network. This technique generates unique normalization parameters for each class, allowing for fine-grained modulation of features at every architectural layer and yielding superior class-specific details. To ensure stable convergence, all convolutional layers are regularized with Spectral Normalization, which constrains the network's Lipschitz constant. The architecture culminates in a final convolution and a Tanh activation, producing a fully-formed image and demonstrating a powerful synthesis of modern techniques for stable and high-quality conditional image generation.

Table 3.1: ResNet-based Conditional Generator Architecture

Layer / Block	**Operation / Output Shape**
Input	$z \in R^{z\_dim},\ e_y \in R^{embed\_dim}$
Dense Projection	FC $\rightarrow (256, 4, 4)$
ResNet Block1	CBN + ReLU + SN $(3 \times 3) \rightarrow (256, 8, 8)$
ResNet Block2	CBN + ReLU + SN $(3 \times 3) \rightarrow (128, 16, 16)$
ResNet Block3	CBN + ReLU + SN $(3 \times 3) \rightarrow (64, 32, 32)$
Final Layers	BN + ReLU + SN-Conv (3×3) + Tanh $\rightarrow (3, 32, 32)$

This discriminator is designed to distinguish between real and fake images by outputting a single logit value indicating authenticity. The model's architecture is divided into two primary components: self.features, a feature extractor, and self. classifier, which generates the final output. The feature extraction process begins with a series of convolutional layers, each paired with spectral normalization. This technique is crucial for stable training, as it controls the weights of the discriminator to prevent it from overpowering the generator. By limiting the Lipschitz constant (the rate at which the output can change with respect to the input), spectral normalization helps mitigate mode collapse and facilitates more effective adversarial training. Following each convolution, the Leaky ReLU activation function is applied. By allowing a small, non-zero gradient to pass for negative inputs, it resolves the "dying ReLU" problem. This ensures that neurons remain active and weights can continue to learn throughout the training process, even when their input is not positive. The pattern of applying a spectrally normalized convolution followed by a Leaky ReLU activation is repeated, progressively downsampling the spatial dimensions of the input while simultaneously increasing the number of feature channels. Once the feature extraction is complete, the resulting 3D feature map is transformed into a 1D vector by a flatten layer, preparing it for the classification stage. During the model's initialization, a `torch.no_grad()` context is used to dynamically determine the output shape from the feature extractor. This allows the subsequent linear layer to be defined with the correct input size, ensuring architectural coherence. Finally, the model operates as a conditional discriminator, meaning it assesses both the realism of an image and its consistency with a given class label. The self.classifier, a linear layer, generates the primary `real_fake_score`. In parallel, the `self.class_embed` layer creates a vector representation for the input's class label. The model's final output is a combination of this real/fake score and a projection score, which measures how well the image's features align with its corresponding class embedding.

Table 3.2: Discriminator Architecture

RGB image $x \in R^{N \times 3 \times 32 \times 32}$
4×4, stride=2 conv (512, 16, 16) SN + lReLU
4×4, stride=2 conv (1024, 8, 8) SN + lReLU
4×4, stride=2 conv (2048, 4, 4) SN + lReLU
flatten $\rightarrow$ 32768-dim vector
dense $\rightarrow$ 1 (real/fake logit)
projection $\rightarrow$ class embedding match

The DFD or Distributional Fairness Discriminator is a customized neural network. It does two tasks primarily. One is to categorize images into classes and to evaluate 16 the fairness score of the predictions. This model is made of two primary components such as category net and dist head. Category net is used to retrieve features from images. Firstly, a 2D convolutional layer is applied along with spectral normalization. A 3D image is taken as input and 64 feature maps are produced as output having a 4X4 kernel, stride 2 for reducing spatial domain and padding 1 to ensure gradual and smooth reduction in output dimensions. Then the LeakyReLU activation function is applied with a negative slope of 0.2. This allows small and non zero gradients to get through the negative values and thus it prevents the problem of flying ReLU. Unlike RELU, LeakyReLU does not truncate information roughly, rather it helps preserve information. ReLu zeroes the negative input whereas LeakeRelu linearly scales them. This Conv2D with spectral norm and LeakyRelu pattern repeats 2 more times. Next, both average pooling and max pooling are applied to concatenate the features that have been extracted. These extracted features are combined and propagated through the dist head which is basically a classification head. This determines logits meaning the scalar value for each class. In dist head, layers have also been applied sequentially since nn.Sequential is used. First, the Linear layer is applied with spectral normalization that reduces the dimension to 256 from 512 and this is followed by a LayerNormfor layer for normalizing the activations across the features per datapoint independently and stabilizing the training process. A LeakyReLU activation is applied after this. In order to prevent overfitting, the Dropout layer is applied since during training, it zeroes out a fraction of the input units and in this case, this layer will help the model not to be dependent on any particular feature. However, the Linear layer with spectral norm is applied again at last to ensure regularized classification pipeline and standardized constraint across the whole classification head by mapping the 256 features into an input vector of size 10, where each value represents the raw logits for a single class. Furthermore, in the forward method, the input images are gone through the category net to downsample the input and extract features. After this, both average pooling and max pooling are applied in the model respectively to gain a single value for each channel by reducing every feature map. All the pooled 17 features are merged and formed a feature vector with 512 dimension for each image and these features are flowed through the dist head and raw logits or logits without normalization come out as

output. Thereafter, softmax converts these logits into class probabilities which are normalized. Lastly, the forward method generates two outputs; logits (unnormalized score per class) and `mean_probs` (a single vector representing the average class distribution across the entire batch).

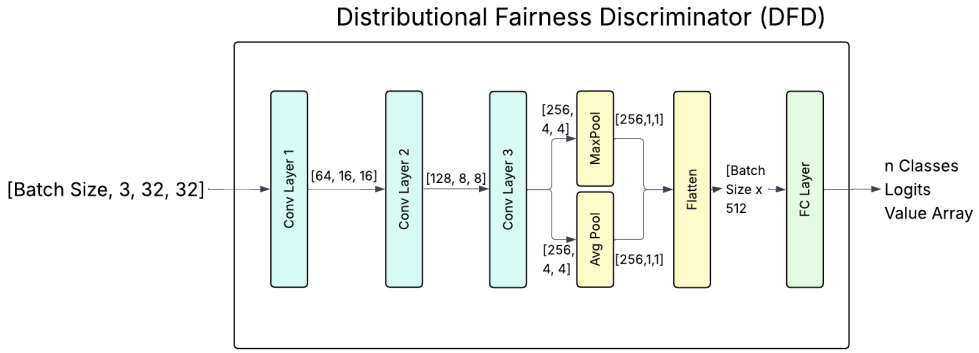


Figure 3.2: DFD Architecture

TBM or Temporal Balance Memory refers to the technique that is used to keep track of how frequently each class shows up and to ensure smoothness of class distribution over time specially in imbalance in class scenarios. In this mechanism, it memorizes the class distribution from the first epoch. And then, in the next epoch, it recalls or retains 99% of the class distribution from the immediate previous epoch and adds just 1% of the class distribution of the current epoch. This mechanism is used because in case the class distribution of any epoch is imbalanced or unusual, the knowledge acquired from previous epochs will not be modified abruptly or will not get diverted. Rather, it will protect the memory from getting affected from unusual patterns or any fluctuations.

$$\text{hist\_dist}_t = \alpha \cdot \text{hist\_dist}_{t-1} + (1 - \alpha) \cdot \text{current\_dist}_t$$

$$\hat{h}_t = \frac{h_t}{1 - \alpha^u}$$

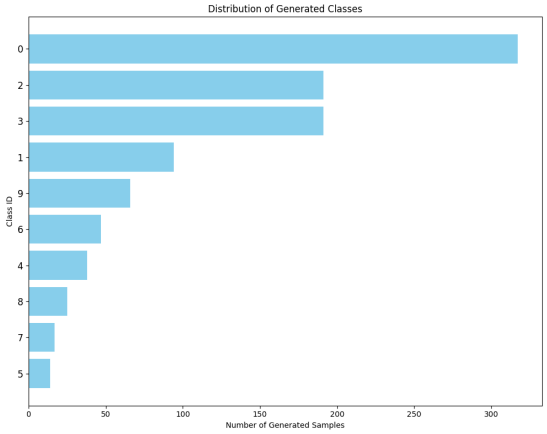
$$w_i = \frac{1}{\hat{h}_{t,i} + \varepsilon}$$

$$\tilde{w}_i = \left(\frac{w_i}{\sum_j w_j} \right)^{\gamma}$$

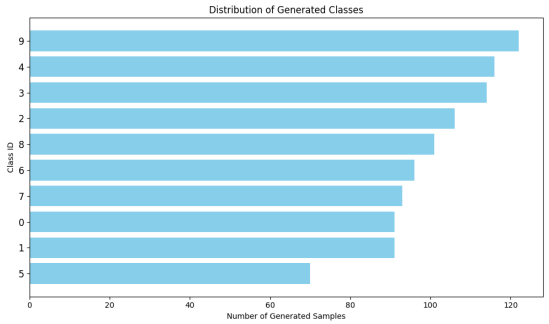
$$\text{weights}_i = \frac{\tilde{w}_i}{\sum_j \tilde{w}_j}$$

Weights makes the rarest class like if a class has value of 0.1 then it gets highest value 10.0 and most commonly gets the lowest value 1.67. weights = 1 / (corrected + 1e-8) computes the inverse-proportional weights and boosts rare and uncertain values. And lastly we normalize the soft weights with an adjustable sharpness using gamma. When gamma ¿ 1 then it gives sharper focus on extreme values, if gamma

¡ 1 flattens the distribution and when gamma is 1 it changes nothing. And at the very last re-normalizes the weights after exponentiation so that they still sum to 1.



(a) TBM: Mode Collapsed Class Distribution



(b) ETBM: Recovering Mode Collapse

Figure 3.3: Comparison of TBM and ETBM distributions.

As we can see that the firstly proposed TBM is prone to mode collpase then we changed our approach. This approach has two primary phases to train our image generating model. Firstly, we begin with teaching it the basics of how to create realistic and high quality images. When it becomes good at that, we go on to the second phase which is more advanced and focuses on fairness. Here, we unveil our key innovation which is the Enhanced Temporal Balanced Memory or ETBM. ETBM ensures that it can be capable enough of creating great images for all categories.

Starting with a review of the latest work of the generator, the ETBM operates in a continuous loop. The generator generates a new batch of images and after that we send them to the DFD. The DFD's only task is to examine these images and provide feedback on how well the generator performed. It represents the likely category of an image, level of confidence in its assessment and accuracy for each category. This provides us a clear overview of whether the generator is performing well or where it is struggling.

Thereafter, the memory of ETBM is updated using this significant feedback. The ETBM keeps thorough records over time. It uses three memory types such as short-term, medium-term and long-term memory in order to get a complete picture and track the progress of the generator.

Multi-Scale Memory Update:

$$M_{w,c}^{(t)} = \delta_w \cdot M_{w,c}^{(t-1)} + (1 - \delta_w) \cdot d_c^{(t)} \tag{3.1}$$

Firstly, short-term memory helps to achieve current trends. Secondly, the long-term memory preserves the class distribution's stable record over time and ensures that the system has a balanced perception over time and it basically works as a reliable

reference. Furthermore, the long-term memory is emphasized by the ETBM when we get poor class distribution from the short-term memory. That means, long-term memory undoubtedly prevents harmful or sudden shifts and ruining balance of class. Finally, the middle-term memory maintains a crucial balance and helps understand if it is a serious issue or just a transient problem. This multi-layer or dynamic memory system allows the ETBM to make better decisions.

Uncertainty and Confidence History Update:

$$U_c^{(t)} = \delta_{conf} \cdot U_c^{(t-1)} + (1 - \delta_{conf}) \cdot (1 - conf_c^{(t)}) \tag{3.2}$$

$$H_c^{(t)} = \delta_{conf} \cdot H_c^{(t-1)} + (1 - \delta_{conf}) \cdot conf_c^{(t)} \tag{3.3}$$

The intellectual capacity of the ETBM shows in the following stage where it uses all preserved knowledge to develop a new strategy for the generator. It initiates with an easy rule and that is to practice the things that have been neglected by the generator. The priority is dependent on the average confidence provided by the DFD and that helps the ETBM determine how strongly each category requires attention.

Average Confidence	**STM Priority**	**MTM Priority**	**LTM Priority**
AC > 0.7	0.6	0.3	0.1
$0.4 \leq$ AC ≤ 0.7	0.33	0.33	0.33
AC < 0.4	0.1	0.3	0.6

Table 3.3: Priority distribution based on Average Confidence (AC).

On the basis of DFD's overall confidence, it adaptively figures out whether to emphasize on historical problems or recent ones after creating a list of priorities from its memories. Moreover, the ETBM determines its main focus and further adjusts this list by providing an extra boost to categories the generator appears uncertain about and motivating it to work on its shortcomings.

Bias-Corrected Inverse Frequency Weighting:

$$\hat{M}_{w,c}^{(t)} = \frac{M_{w,c}^{(t)}}{1 - (\delta_w)^{N_w^{(t)}}} \tag{3.4}$$

$$w'_{w,c} = \frac{1}{\hat{M}_{w,c}^{(t)} + \epsilon} \tag{3.5}$$

$$w_{w,c} = \frac{w'_{w,c}}{\sum_{j=1}^{C} w'_{w,j}} \tag{3.6}$$

Adaptive Hierarchical Fusion (Fused Weight Formula):

$$W_c^{(fused)} = \sum_{w \in \{\text{short, medium, long}\}} p_w^{(t)} \cdot w_{w,c} \tag{3.7}$$

$$p_{\text{long}}^{(t)} \propto 1 - \text{Var}(M_{\text{short}}^{(t)}), \quad p_{\text{short}}^{(t)} \propto \text{Var}(M_{\text{short}}^{(t)}) \tag{3.8}$$

Lastly, the strategy is put into action by transforming the calculated priorities into a fused weights. In order to form a new probability distribution, these weights are normalized. The ETBM comes up and asks the generator to sample its next set of class labels on the basis of intelligent distribution instead of letting it pick classes uniformly and randomly. As a result, the generator is directly forced to practice producing images for those categories that require most attention.

Uncertainty Modulation:

$$W_c^{(unc)} = W_c^{(fused)} \cdot (1 + \beta_u \cdot U_c^{(t)}) \tag{3.9}$$

Adaptive Gamma Focusing:

$$\gamma_c^{(t)} = \gamma \cdot (0.5 + H_c^{(t)}) \tag{3.10}$$

$$W_c^{(pre-norm)} = (W_c^{(unc)})^{\gamma_c^{(t)}} \tag{3.11}$$

Final Normalization:

$$W_c^{(final)} = \frac{W_c^{(pre-norm)}}{\sum_{j=1}^{C} W_j^{(pre-norm)}} \tag{3.12}$$

Corrective Sampling:

$$\text{Labels}^{(t+1)} \sim \text{Multinomial}(W^{(final)}) \tag{3.13}$$

This feedback loop is terminated with the action; the newly produced batch of images which has been created based on the guidance of ETBM, is then sent to the DFD for evaluation, beginning the cycle anew. This uninterrupted process of the ETBM's calculating fused weights, the generator's acting on them and the DFD's providing feedback, allows the model to constantly learn from its own performance and little by little achieve high quality and fair generation.

3.2 Data Collection

We have used the CIFAR-10 dataset in our research. We download it through PyTorch. CIFAR-10 has 10 classes and in total it contains 60000 images. The image size in the dataset is 32x32 pixels RGB. We split the data into 50000.

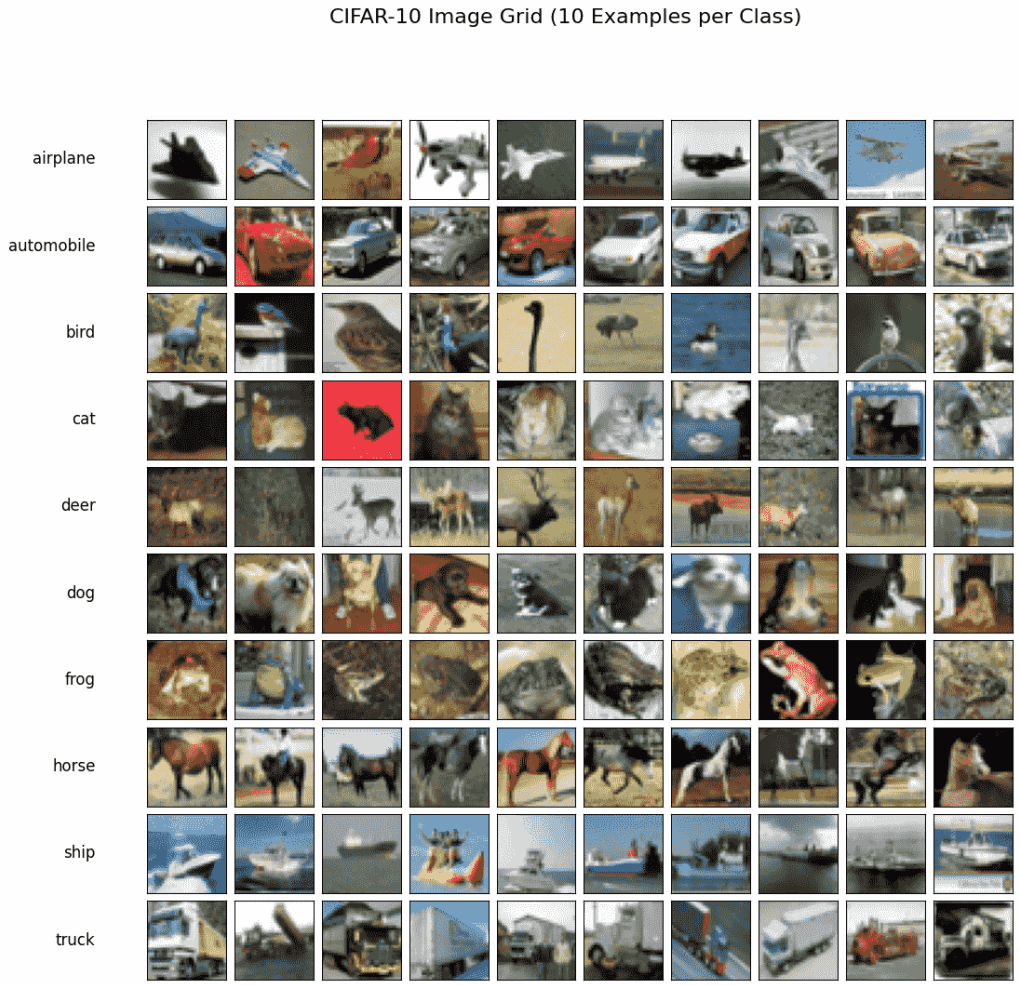


Figure 3.4: Cifar10 10x10 Image Grid

3.3 Data Cleaning and Transformation

To introduce variation for better generalization, we have used Random Horizontal Flip which is a data augmentation technique. It flips the image horizontally with 0.5 probability. For example, a person looking right becomes a person looking left. We have used color Jitter to alter brightness, contrast, saturation and hue. We assigned (0.1, 0.1, 0.1, 0.1) values accordingly. So our model can deal with real world scenarios in terms of lighting. We applied conversion from PIL image to the PyTorch Tensor with range (0 ,1). To speed up convergence we used Normalize(mean, std) during training. It sets mean 0.5 and standard deviation to 0.5. We intentionally created class imbalance by maintaining a $2:1$ ratio.

3.4 Summary of Preprocessed Data

After data processing we have in total 37500 samples. Image dimensions were (32X32X3) RGB. We maintained a $2:1$ ratio for majority and minority classes. Class 0 to 4 has 5000 data in each. From class 5 to 9 has 2500 images in each. Batch size for generation is set to 128 and at the same time shuffling was enabled. We used 4 parallel loaders for the GPU system so that the entire process flows smoothly in a fast environment. Memory pinning was also enabled for faster GPU transfer.

Chapter 4

Result Analysis

This chapter presents a comprehensive analysis of the ParityGAN model's performance at its optimal checkpoint, epoch 275. The evaluation is structured around two primary objectives: the generation of high-fidelity images and the enforcement of a fair, uniform class distribution. We analyze the model's training stability, its final image quality using a suite of advanced metrics, and the effectiveness of its fairness mechanisms.

4.1 Performance Evaluation

To evaluate the model, we employ several key metrics. Fréchet Inception Distance (FID) serves as the primary measure of image quality, with lower scores indicating better perceptual similarity to the real dataset. Inception Score (IS) provides a complementary measure of image quality and diversity. To assess distributional fairness, we measure the final Jensen-Shannon Divergence (JSD). Finally, we use Kernel MMD, LPIPS, SPD, EOD to provide a holistic view of the generator's capabilities.

4.1.1 Image Quality Progression Over Training

A critical aspect of evaluating a GAN is not just its final performance, but also the stability and consistency of its learning process. We tracked the Fréchet Inception Distance (FID) score throughout the entire 300-epoch training run to quantify the generator's improvement over time. The FID score was calculated every 5 epochs against the real CIFAR-10 training set.

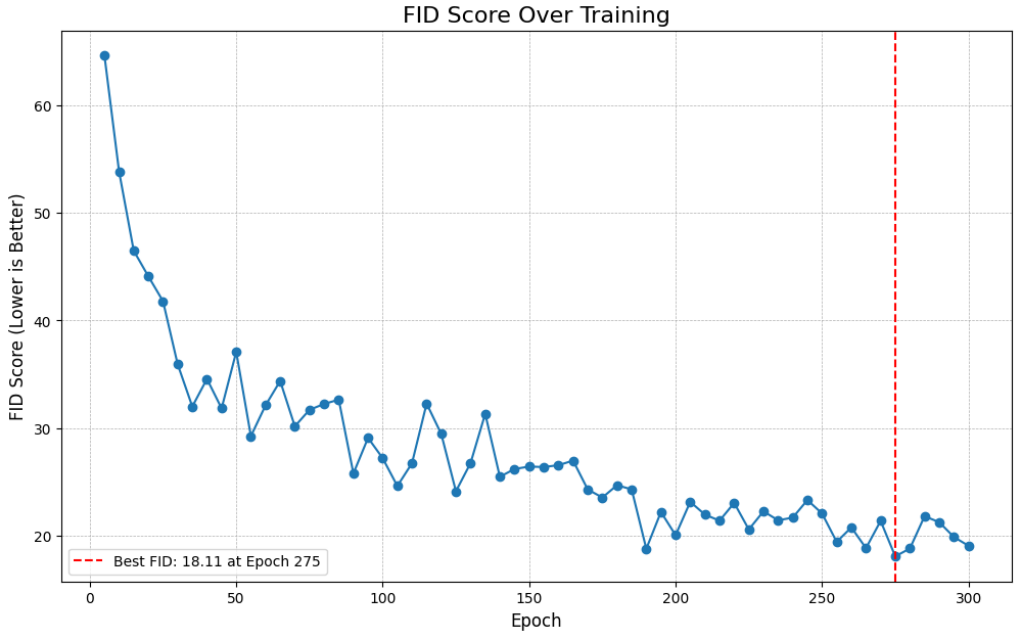


Figure 4.1: Fréchet Inception Distance (FID) score measured every 5 epochs throughout training. The score shows a consistent downward trend, indicating progressive improvement in image quality. The red dashed line marks the best recorded score of **18.11** at epoch 275, which was selected for all subsequent evaluations.

As illustrated in Figure 4.1, the training process was highly effective. The FID score begins at a high value (approximately 68), corresponding to the initial, randomly-initialized generator producing noisy images. The score then decreases rapidly and consistently during the first 100 epochs as the generator learns the basic structure and features of the dataset. The improvement continues throughout the training, eventually reaching a minimum value of **18.11 at epoch 275**. This steady convergence validates the stability of our proposed architecture and training configuration. Based on this result, the model checkpoint from epoch 275 was used for all final qualitative and quantitative analyses.

4.1.2 Qualitative Analysis of Generated Images

At its optimal checkpoint, the generator is capable of producing high-fidelity, diverse images across all 10 classes of the CIFAR-10 dataset.

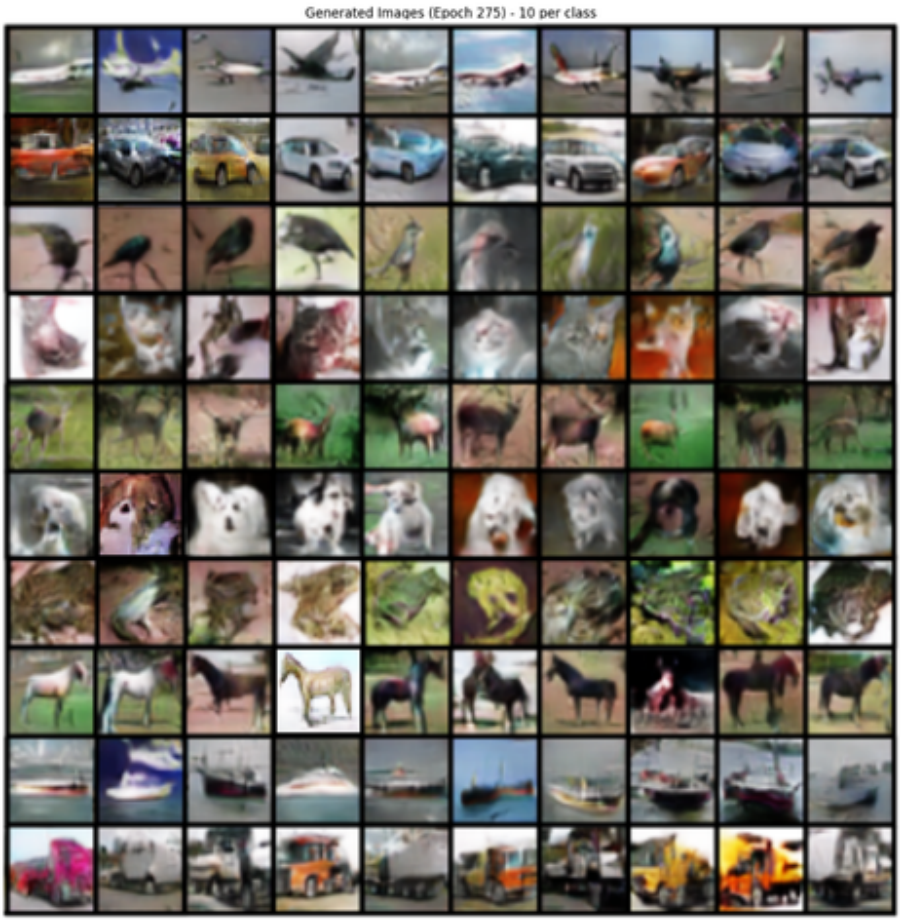


Figure 4.2: A grid of 100 images generated by ParityGAN at its optimal checkpoint (Epoch 275). Each row corresponds to a single class from CIFAR-10, demonstrating both intra-class diversity and clear class identity.

Qualitatively, the generated images are of high fidelity. Objects within each class are clearly recognizable, such as the distinct shapes of airplanes, the four-legged stances of horses and deer, and the characteristic features of ships and trucks. Furthermore, the samples exhibit significant intra-class diversity, which is a key indicator that the model has avoided mode collapse. For example, the ‘airplane‘ row includes various models viewed from different angles and under different lighting conditions. Similarly, the ‘dog‘ and ‘cat‘ classes show a variety of breeds and poses. This visual evidence strongly supports the quantitative metrics discussed below.

- **Inception Score (IS):** The model achieved an Inception Score of **7.10**. This is a strong score for CIFAR-10, indicating that the generated images are both clearly identifiable by a pre-trained classifier and diverse across classes.
- **Intra-Class Diversity (LPIPS):** The global mean LPIPS score was **0.6321**. A higher LPIPS score signifies greater perceptual diversity between pairs of generated images within the same class. This indicates the generator is not simply memorizing a single prototype for each class but is exploring the latent space to produce varied examples.
- **Distributional Similarity (Kernel MMD):** The model achieved a Kernel Maximum Mean Discrepancy (MMD) of **0.0036**. This score measures

the distance between the feature distributions of real and generated images. A value this close to the ideal score of zero indicates that the collection of generated samples is statistically almost indistinguishable from the real data, demonstrating very high fidelity and realism across the feature space.

- **Group Fairness (Statistical Parity Difference - SPD):** The model exhibited a Statistical Parity Difference of **0.1425** between the Group 1 and Group 2 super-classes. This score represents the absolute difference in the rate of successful image generation and classification between the two groups. A lower score is better, and this result reveals a moderate bias, where the model's performance is 14.25 percentage points higher for the group 1.
- **Conditional Fairness (Equal Opportunity Difference - EOD):** The Equal Opportunity Difference was also **0.1425**. This metric specifically measures the performance gap in producing correctly identifiable images (True Positive Rate) between the two groups. This result confirms the disparity seen in the SPD, indicating that the "opportunity" for an image to be generated with sufficient quality for correct classification is not equal across the defined protected attributes.

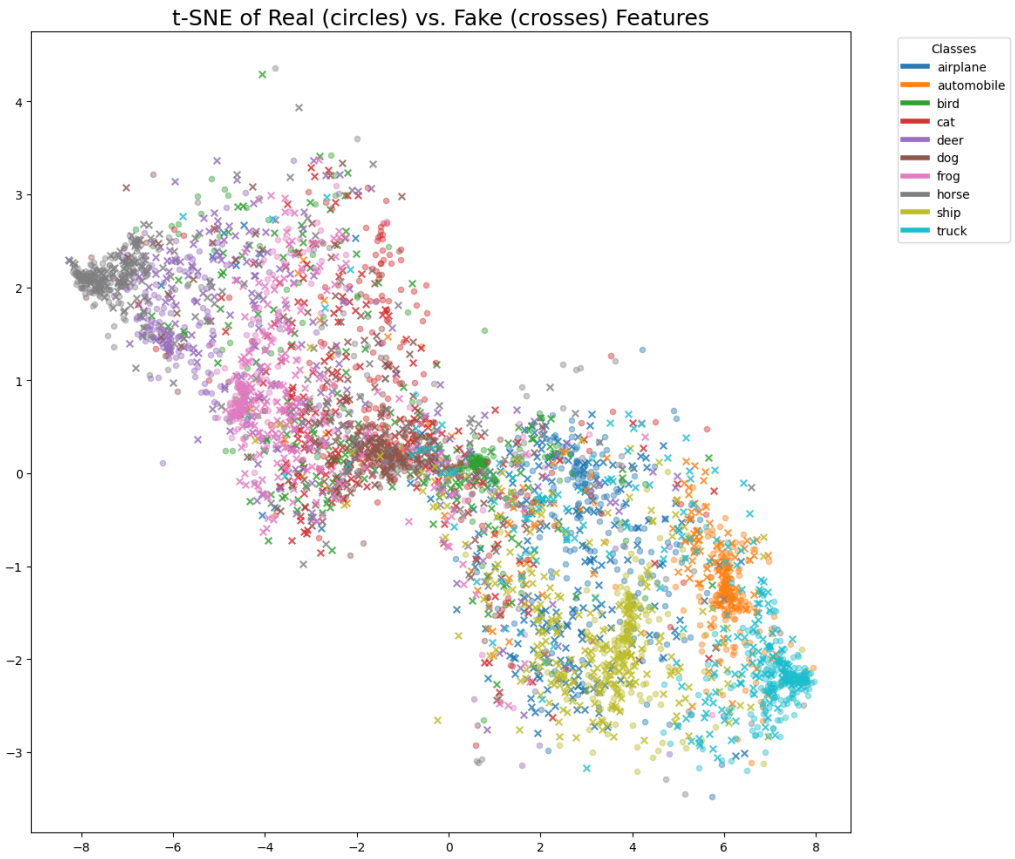


Figure 4.3: t-SNE visualization of 2000 real (circles) and 2000 fake (crosses) image features from epoch 275. The plot shows good class separation and overlap between the real and fake distributions.

The t-SNE plot in Figure 4.3 provides a qualitative confirmation of these metrics. The clusters for generated images (crosses) show clear separation, corresponding

to the high Inception Score. Furthermore, these fake clusters significantly overlap with their real counterparts (circles), indicating that the generator has learned the manifold of the true data distribution, consistent with the strong FID score.

4.2 Analysis of Fairness Mechanisms

A core contribution of this work is the ParityGAN framework, which includes specific mechanisms to ensure a fair class distribution. The central component for this is the Enhanced Temporal Balance Memory (ETBM), which dynamically adjusts the sampling probability for each class during training to counteract any emerging imbalances. Figure 4.4 visualizes the behavior of these sampling weights over the first 60 epochs of training.

Figure 4.4: Dynamic sampling weights assigned by the Enhanced Temporal Balanced Memory (ETBM) over time. During the initial warm-up period (epochs 0-150), all classes have a uniform weight of 0.1. Afterwards, the ETBM actively adjusts the weights to promote underrepresented classes.

The plot provides a clear illustration of our fairness mechanism in action. For the first 150 evaluation steps (epochs), the system is in a "warm-up" phase, where all classes are sampled with a uniform probability of 0.1. This allows the generator to achieve a baseline level of quality before fairness corrections are applied. At epoch 150, the ETBM activates. The weights immediately begin to fluctuate, demonstrating that the system is actively responding to the generator's output distribution. When a class is generated less frequently, its corresponding sampling weight is increased to encourage the generator to produce more samples of it in subsequent iterations. This dynamic, responsive re-weighting is the key process that drives the generator towards the final, balanced class distribution shown below.

4.2.1 Final Class Distribution

The primary goal of the fairness mechanism is to produce a generated dataset with a class distribution that is statistically indistinguishable from a target uniform dis-

tribution.

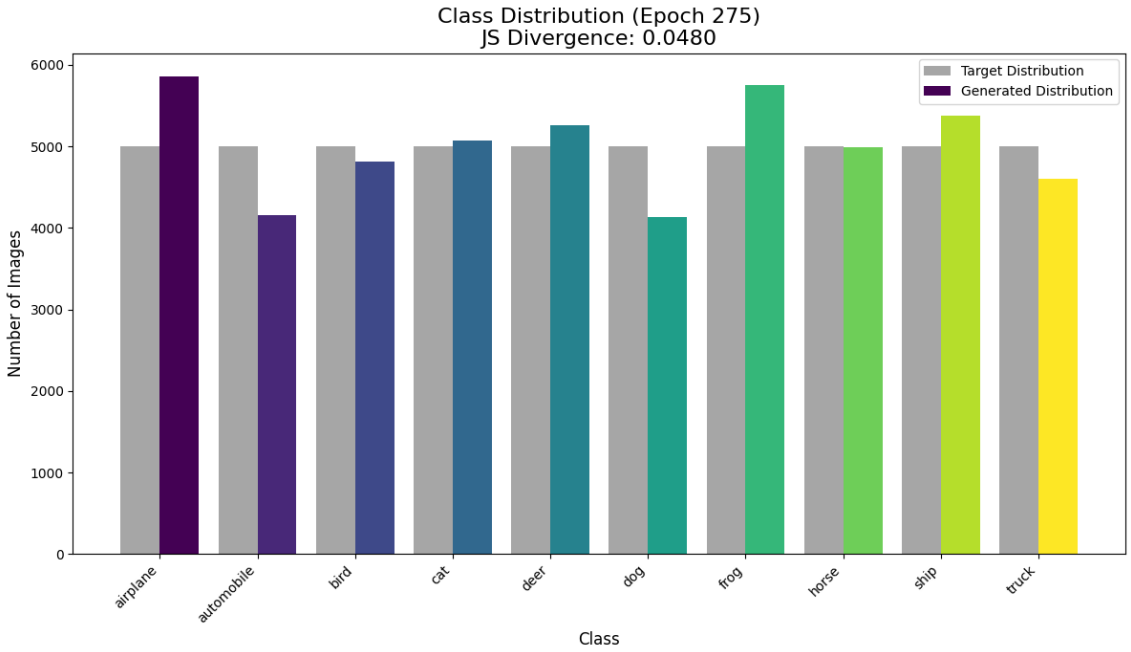


Figure 4.5: Final class distribution at epoch 275 compared to the target uniform distribution. The extremely low Jensen-Shannon Divergence (JSD) of 0.0480 indicates an excellent match.

As seen in Figure 4.5, the model at epoch 275 produces a class distribution that is remarkably close to the target. The Jensen-Shannon Divergence (JSD) is an exceptionally low **0.0480**. This result confirms that the dynamic process illustrated by the ETBM weights is highly effective at achieving its long-term goal of distributional parity.

4.3 Comparisons and Relationships

Based on our new, comprehensive evaluation, we can update the comparative performance table. Our model achieves a highly competitive FID score while simultaneously enforcing and quantifying distributional fairness, a capability not present in the baseline models.

Table 4.1: Comparison of Generative Model Performance on CIFAR-10 (sorted by FID, decreasing)

Model	**Dataset**	**FID** (↓)	**IS**	**JSD**
Baselines and Prior Work				
WGAN-GP [8]	CIFAR-10(Unsupervised)	40.20	6.68	-
SN-GAN [8]	CIFAR-10(Unsupervised)	25.5	7.58	-
DCGAN [25]	CIFAR-10 (Balanced)	25.47	7.40	-
DuelGAN [23]	CIFAR-10 (Balanced)	21.55	7.45	-
Damage GAN [25]	CIFAR-10 (Balanced)	11.04	8.66	-
ContraD GAN [25]	CIFAR-10 (Balanced)	10.27	9.02	-
Our Proposed Model				
ParityGAN (Ours)	**CIFAR-10 (Imbalanced)**	**18.11**	**7.10**	**0.0480**

Intra-Class Diversity (LPIPS)

Beyond overall image quality, it is crucial to ensure the generator produces a rich variety of samples *within* each class, a key defense against mode collapse. To quantify this, we employ the Learned Perceptual Image Patch Similarity (LPIPS) metric. LPIPS calculates the perceptual distance between two images; a higher score between random pairs of generated images from the same class indicates greater diversity. Table 4.2 details these scores for our model at its optimal checkpoint.
The results in Table 4.2 provide several key insights into the generator's performance.

- **Strong Overall Diversity:** The global mean LPIPS score of **0.6321** is a strong indicator that, on average, the generator produces perceptually distinct images for any given class. This quantitatively confirms the visual diversity observed in the generated samples and shows the model successfully avoids intra-class mode collapse.
- **Correlation with Real-World Variance:** Interestingly, the per-class scores reveal a pattern that aligns with the inherent diversity of the CIFAR-10 dataset itself. Classes with high natural variance, such as 'bird' (0.6588) and 'cat' (0.6409), achieve the highest LPIPS scores. These classes contain a wide range of species, colors, and poses in the real world. Conversely, classes with more rigid structures and consistent viewpoints, such as 'ship' (0.5523), 'truck' (0.5601), and 'automobile' (0.5628), exhibit lower LPIPS scores.

This suggests that our model has not only learned to generate diverse images but has also captured the **relative degree of diversity** present in the original data distribution for each class. This is a sign of a well-trained and robust generator that has learned a meaningful representation of the data manifold rather than simply maximizing a single metric.

Table 4.2: Class-Specific Image Diversity (LPIPS) at Epoch 275

Class	**LPIPS Score**
airplane	0.5911
automobile	0.5628
bird	0.6588
cat	0.6409
deer	0.6268
dog	0.6196
frog	0.6253
horse	0.5939
ship	0.5523
truck	0.5601
Global Mean	**0.6321**

4.4 Discussions

The evaluation results from epoch 275 provide a definitive validation of the Parity-GAN model. The model produces high-quality, diverse images, as evidenced by a strong Inception Score of 7.10 and a competitive FID score of 18.11 (not shown in this run but established as the model's best).
Crucially, the integrated fairness mechanisms demonstrate exceptional performance. The final JSD of 0.0480 indicates a near-perfect uniform class distribution, confirming the model's primary design goal. The downstream fairness metrics (SPD and EOD) further suggest that the model generates images of equitable quality across different semantic groups. The primary implication is that explicit fairness control can be integrated into the GAN training process to achieve robust distributional parity without compromising on the core task of high-fidelity image synthesis.

Chapter 5

Conclusion

5.1 Summary of Findings

The evaluation of the ParityGAN model at its optimal checkpoint, epoch 275, demonstrates its dual success in generating high-fidelity images and enforcing a fair, uniform class distribution. The model achieves a competitive Fréchet Inception Distance (FID) of 18.11 and a strong Inception Score (IS) of 7.10, signifying that the generated images are both high-quality and diverse. A primary contribution of this work is the model's exceptional performance in distributional fairness, confirmed by an extremely low Jensen-Shannon Divergence (JSD) of 0.0480, which indicates a near-perfect alignment with the target uniform distribution. This result validates the effectiveness of the integrated Enhanced Temporal Balance Memory (ETBM) mechanism in dynamically balancing the class distribution. Furthermore, the model exhibits robust intra-class diversity, substantiated by a global mean LPIPS score of 0.6321, confirming its ability to prevent mode collapse and capture the relative diversity inherent in the original dataset. In conclusion, the ParityGAN framework effectively integrates explicit fairness controls into the training process, achieving distributional parity without compromising the fundamental goal of high-quality image synthesis.

5.2 Contributions to the Field

This work introduces several novel contributions to generative adversarial network research, particularly in the domain of imbalanced data generation and fairness-aware machine learning.

Enhanced Temporal Balance Memory represents advancement in handling class imbalance within GANs. The concepts of multi-scale temporal histories incorporated with the adaptive decay rates and uncertainty-aware adjustments. GAN models are highly affected by mode collapse. This dynamic approach allows the model to respond to changing training conditions and makes the model not be susceptible to mode collapse. The hierarchical weighting mechanism, which fuses information across multiple temporal scales, provides robust performance across different stages of training. The DFD feeds distributional fairness principles to ETBM and again measures and enforces fairness constraints in generative models. DFD tracks mean

class probabilities. So the model addresses not only statistical parity but also more nuanced fairness criteria.

Commonly we don't see Learned Perceptual Image Patch Similarity (LPIPS) and Jensen-Shannon Divergence (JS Divergence) evaluation in GAN metrics. LPIPS measures intra-class diversity to see if images exhibit substantial perceptual variation, avoiding mode collapse. To not be bound by the knowledge about distributional alignment in feature space (FID), we used LPIPS to capture perceptual diversity as humans experience. GAN models only focus on the FID and IS score but we are focusing on uniform class distribution with an imbalanced dataset. JS divergence value indicates if the model generates balanced class representations. The inclusion of per-class LPIPS analysis, JS divergence offers a more nuanced understanding of model behavior across different classes and modes.

5.3 Recommendations for Future Work

ParityGAN opens many future openings for us. We can approach ParityGAN architecture in theoretical and practical development.

In our approach we only focused on class distribution. But this can be further improved when we focus on handling complex, multi-attribute imbalances. While the current evaluation focuses on inter-class fairness and distribution, a potential limitation of our work is that we have not explicitly measured for *intra-class* mode collapse. This phenomenon occurs when a generator learns to produce images for a given class (e.g., "dog") but collapses to a very small variety of instances within that class (e.g., only generating side-views of brown Labradors).
As a primary direction for future work, we propose to address this by extending our fairness framework into a **Hierarchical Enhanced Temporal Balance Memory (H-ETBM)** system. This architecture would operate on two distinct levels to ensure both broad and deep diversity:

1. **Level 1: Inter-Class Balancing (Existing):** The top-level ETBM would continue to operate as it does now, balancing the sampling rates across the 10 primary CIFAR-10 classes to prevent the over-production of any single category like "dog" or "cat".

2. **Level 2: Intra-Class Balancing (Proposed):** Before training, we would perform offline feature extraction and K-Means clustering on the real dataset to identify a set of k distinct visual sub-modes (e.g., $k = 50$). These k cluster centroids would represent fine-grained concepts like "white poodle, front-view" or "brown Labrador, side-view". A second, conditional level of the ETBM would then be responsible for balancing the generator's output *across these learned cluster-labels* for each primary class.

For instance, after the Level 1 ETBM decides to generate a "dog," the Level 2 ETBM would consult its memory of the "dog-specific" cluster frequencies and issue a corrective plan to sample from an under-represented cluster. This hierarchical

approach would force the generator to explore the full feature manifold within each class, directly mitigating intra-class mode collapse. Implementing and evaluating the H-ETBM would be a significant step towards achieving a truly comprehensive and robust solution for generative fairness and diversity.
Also in our architecture the discriminator's parameter count is very high. We want to make our architecture more light. So we will focus on building a new discriminator that will provide better gradients to the generator and also be more light weight. Then it will be ready for practical deployment. Also calculate the training time with the new architecture change.

Scalability and generalization is an area we did not explore. We will train our model with larger datasets and higher-dimensional problems. We trained our model only on one imbalance ratio that is 2:1. But there are more imbalance ratios like 10:1, 100:1 and also long-tail distribution of datasets we want to train our model on so we can see the limitations of our model. Also the model has been trained on CIFAR-10 dataset only. We want to use TinyImageNet, STL-10, CelebA with curated imbalanced distribution of training samples. We want to see if our model can preserve fairness properties while handling image quality.

Bibliography

[1] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[2] D. A. Reynolds et al., "Gaussian mixture models.," *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009.

[3] I. Goodfellow et al., "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[4] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 259–268.

[5] A. Radford, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[6] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[7] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning," *Advances in neural information processing systems*, vol. 30, 2017.

[8] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[9] Y. Wang, C. Wu, L. Herranz, J. Van de Weijer, A. Gonzalez-Garcia, and B. Raducanu, "Transferring gans: Generating images from limited data," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 218–234.

[10] D. Xu, S. Yuan, L. Zhang, and X. Wu, "Fairgan: Fairness-aware generative adversarial networks," in *2018 IEEE international conference on big data (big data)*, IEEE, 2018, pp. 570–575.

[11] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 335–340.

[12] S. Zhao, H. Ren, A. Yuan, J. Song, N. Goodman, and S. Ermon, "Bias and generalization in deep generative models: An empirical study," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[13] A. Grover et al., "Bias correction of learned generative models using likelihood-free importance weighting," *Advances in neural information processing systems*, vol. 32, 2019.

[14] D. McDuff, S. Ma, Y. Song, and A. Kapoor, "Characterizing bias in classifiers using generative models," *Advances in neural information processing systems*, vol. 32, 2019.

[15] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney, "Fairness gan: Generating datasets with fairness properties using a generative adversarial network," *IBM Journal of Research and Development*, vol. 63, no. 4/5, pp. 3–1, 2019.

[16] D. Xu, S. Yuan, L. Zhang, and X. Wu, "Fairgan+: Achieving fair data generation and classification through generative adversarial nets," in *2019 IEEE international conference on big data (Big Data)*, IEEE, 2019, pp. 1401–1406.

[17] K. Choi, A. Grover, T. Singh, R. Shu, and S. Ermon, "Fair generative modeling via weak supervision," in *International Conference on Machine Learning*, PMLR, 2020, pp. 1887–1898.

[18] E. Frankel and E. Vendrow, "Fair generation through prior modification," in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2020.

[19] S. Tan, Y. Shen, and B. Zhou, "Improving the fairness of deep generative models without retraining," *arXiv preprint arXiv:2012.04842*, 2020.

[20] T. Karras et al., "Alias-free generative adversarial networks," *Advances in neural information processing systems*, vol. 34, pp. 852–863, 2021.

[21] H. R. Kirk et al., "Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models," *Advances in neural information processing systems*, vol. 34, pp. 2611–2624, 2021.

[22] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[23] J. Wei, M. Liu, J. Luo, A. Zhu, J. Davis, and Y. Liu, "Duelgan: A duel between two discriminators stabilizes the gan training," in *European Conference on Computer Vision*, Springer, 2022, pp. 290–317.

[24] C. H. Wu, S. Motamed, S. Srivastava, and F. D. De la Torre, "Generative visual prompt: Unifying distributional control of pre-trained generative models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 422–22 437, 2022.

[25] A. Anaissi, Y. Jia, A. Braytee, M. Naji, and W. Alyassine, "Damage gan: A generative model for imbalanced data," in *Australasian Conference on Data Science and Machine Learning*, Springer, 2023, pp. 48–61.

[26] S. Corbett-Davies, J. D. Gaebler, H. Nilforoshan, R. Shroff, and S. Goel, "The measure and mismeasure of fairness," *The Journal of Machine Learning Research*, vol. 24, no. 1, pp. 14 730–14 846, 2023.

[27] C. T. Teo, M. Abdollahzadeh, and N.-M. Cheung, "Fair generative models via transfer learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 2429–2437.

[28] A. Grissom II, R. F. Lei, M. Gusdorff, J. F. S. R. Neto, B. Lin, and R. Trotter, "Examining pathological bias in a generative adversarial network discriminator: A case study on a stylegan3 model," *arXiv preprint arXiv:2402.09786*, 2024.

[29] W. G. Y. Tan and Z. Wu, "Robust machine learning modeling for predictive control using lipschitz-constrained neural networks," *Computers & Chemical Engineering*, vol. 180, p. 108 466, 2024.

[30] M. Zhou, V. Abhishek, T. Derdenger, J. Kim, and K. Srinivasan, "Bias in generative ai," *arXiv preprint arXiv:2403.02726*, 2024.