Synthetic Data Generation using GANs via Architectural Reconfiguration

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

with specialization in

INFORMATION TECHNOLOGY

Submitted by

Hemant Choudhary	IIT2022053
Mohit Yadav	IIT2022057
Anushka Dwivedi	IIT2022062
Sahul Kumar	IIT2022081
Harshini Bhukya	IIT2022084

Under the Supervision of

DR. ANJALI GAUTAM

to the

DEPARTMENT OF INFORMATION TECHNOLOGY

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD

भारतीय सूचना प्रौद्योगिकी संस्थान इलाहाबाद
Indian Institute of Information Technology Allahabad
An Institute of National Importance by Act of Parliament
Deoghat Jhalwa, Prayagraj 211015 (U.P.) India
Ph: 0532-2922025, 2922067; Web: www.iiita.ac.in; Email: contact@iiita.ac.in

CERTIFICATE

It is certified that the work contained in the thesis titled **"Synthetic Data Generation using GANs via Architectural Reconfiguration"** has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Anjali Gautam
Department of Information Technology
IIIT Allahabad

भारतीय सूचना प्रौद्योगिकी संस्थान इलाहाबाद
Indian Institute of Information Technology Allahabad
An Institute of National Importance by Act of Parliament
Deoghat Jhalwa, Prayagraj 211015 (U.P.) India
Ph: 0532-2922025, 2922067; Web: www.iiita.ac.in; Email: contact@iiita.ac.in

CANDIDATE DECLARATION

We, Hemant Choudhary (IIT2022053) Mohit Yadav (IIT2022057) Anushka Dwivedi (IIT2022062) Sahul Kumar (IIT2022081) Harshini Bhukya (IIT2022084) certify that this thesis work titled **"Synthetic Data Generation using GANs via Architectural Reconfiguration"** is submitted by us towards partial fulfillment of the requirement of the Degree of Bachelor of Technology in the Department of Information Technology, Indian Institute of Information Technology, Allahabad.

We understand that plagiarism includes:

1. Reproducing someone else's work (fully or partially) or ideas and claiming it as one's own.
2. Reproducing someone else's work (verbatim copying or paraphrasing) without crediting.
3. Committing literary theft (copying some unique literary construct).

We have given due credit to the original authors/sources through proper citation for all the words, ideas, diagrams, graphics, computer programs, experiments, results, and websites that are not my original contributions. We have used quotation marks to identify verbatim sentences and given due credit to the original authors/sources.

We affirm that no portion of my work is plagiarized. In the event of a complaint of plagiarism, We shall be fully responsible. understand that my supervisor may not be in a position to verify that this work is not plagiarized.

Date:

Hemant Choudhary (IIT2022053)
Mohit Yadav (IIT2022057)
Anushka Dwivedi (IIT2022062)
Sahul Kumar (IIT2022081)
Harshini Bhukya (IIT2022084)
Department of Information Technology
IIIT Allahabad
Prayagraj - 211015, U.P.

ABSTRACT

Training Generative Adversarial Networks (GANs) on high-fidelity images typically requires vast datasets and computational resources, posing challenges in scenarios with limited data availability. This research presents Re-GAN, a novel framework that introduces dynamic sparse-dense training to address these challenges. Re-GAN leverages a mask-based parameterization to selectively prune and regrow network connections during training. This adaptive strategy reduces computational overhead, enhances training stability, and allows the network to explore diverse architectures in real time. Re-GANs adaptive architecture leverages mask-based parameterization to prune low-magnitude weights, reducing computational overhead and strategically regrow connections to retain representational capacity. Evaluations on datasets such as CIFAR-10, Tiny-ImageNet, and FFHQ demonstrate Re-GANs superiority in generating high-quality images while significantly reducing floating point operations (FLOPs) and training time. Key findings underscore Re-GANs versatility as a general-purpose training methodology, capable of improving state-of-the-art GAN architectures, including StyleGAN2 and ProGAN. This research highlights the potential of dynamic architectural reconfiguration in advancing the scalability and applicability of GANs in diverse real-world scenarios.

Acknowledgements

We would like to express our heartfelt gratitude to **Dr. Anjali Gautam** for her invaluable guidance, constant encouragement, and insightful advice throughout the course of our mini project titled **"Synthetic Data Generation using GANs via Architectural Reconfiguration"** Her expertise and suggestions were pivotal in helping us navigate the challenges we encountered, and we deeply appreciate her support.

She served as our guide during the project. Her guidance played a crucial role in keeping us on the right path whenever we faced difficulties. Her timely advice and direction were instrumental in helping us understand complex concepts and tackle problems effectively. We are grateful for her patience and mentorship throughout this journey.

As 5th semester B.Tech students, we found the process both challenging and rewarding. The project allowed us to enhance our knowledge in machine learning and its applications in Synthetic Data Generation using GANs via Architectural Reconfiguration, a field we had not previously explored. We thoroughly enjoyed working on this project and appreciate the learning experience it provided.

Once again, we extend our sincere thanks to Dr. Anjali Gautam for making this project a success.

Contents

Chapter 1

Introduction

ABSTRACT

Generative Adversarial Networks (GANs) have demonstrated exceptional capabilities in image synthesis, but their training remains challenging due to high computational costs, instability, and the need for large datasets. This research introduces Re-GAN, a novel approach to GAN training that dynamically reconfigures network architectures by alternating between sparse and dense phases. Re-GAN employs mask-based parameterization to prune low-magnitude weights during training, reducing computational overhead while retaining model capacity through iterative regrowth of pruned connections.

This study explores the limitations of traditional GANs, such as fixed architectures and inefficient resource utilization, and demonstrates how Re-GAN addresses these issues by incorporating adaptive architectural updates.

1.1 Introduction

In recent years, Generative Adversarial Networks (GANs) have gained prominence in the field of artificial intelligence for their ability to generate high-quality images.[2]-[5] GANs have been widely adopted in applications such as data augmentation, domain adaptation, and image-to-image translation. However, training GANs effectively poses several challenges, especially when working with limited datasets. Traditional GANs like **StyleGAN2** often require large and diverse datasets to achieve optimal performance. In scenarios with constrained data availability, their performance deteriorates significantly, limiting their applicability.

Existing methods to address this issue, such as dynamic data augmentation and the lottery ticket hypothesis (LTH)[8]-[10], offer partial solutions. LTH identifies sparse sub-networks ("winning tickets") within a trained GAN that can match the performance of dense networks. However, these methods involve a time-consuming train-prune-retrain process, which significantly increases computational requirements. Additionally, they rely on training full-scale models before pruning, followed by fine-tuning, further complicating the process.

This project introduces **Re-GAN[1]**, a novel framework that overcomes these challenges by dynamically pruning and regrowing network connections during training itself. Re-GAN leverages mask-based parameterization to deactivate less significant weights, allowing efficient training with reduced computational overhead. By simultaneously reactivating pruned connections in subsequent iterations, the model retains its representational capacity, ensuring stable and high-quality training even with limited data.

Unlike traditional approaches that involve pre-training dense models, Re-GAN eliminates the need for fine-tuning or pre-training, making it computationally efficient. Additionally, the framework applies pruning exclusively to the generator while maintaining the standard architecture for the discriminator, ensuring equilibrium between the two networks during training.

Re-Gan has been validated across multiple datasets, including CIFAR-10 and Flickr Faces HQ (FFHQ), and is compatible with state-of-the-art GAN architectures like ProGAN, StyleGAN2[8], and AutoGAN. Experiments demonstrate that Re-GAN achieves significant improvements in computational efficiency, reducing training time and FLOPs while improving image quality. This work represents a step forward in efficient GAN training, providing a scalable and robust alternative for resource-constrained environments.

By addressing the limitations of traditional GANs and introducing a dynamic approach to architecture optimization, Re-GAN contributes to advancing generative modeling techniques, enabling broader applications in artificial intelligence.

Chapter 2

Literature Review

The development of the Re-GAN framework is founded on a comprehensive review of the existing literature in the field of generative adversarial networks (GANs). This chapter discusses major advancements in GAN architectures, their limitations, and the methodologies adopted to address these challenges.

2.1 Evolution of GANs

Generative Adversarial Networks (GANs), first introduced by Goodfellow et al., have become a cornerstone in generative modeling, enabling high-quality image[**4**] synthesis and numerous applications in computer vision. However, traditional GANs often rely on large and diverse datasets, and their performance deteriorates under limited data conditions. To overcome these challenges, various innovative GAN architectures have been proposed, which are discussed below.

2.1.1 ProGAN: Progressive Growing of GANs

[**12**]ProGAN revolutionized GAN training by introducing a progressive growing strategy. This method starts training at a low resolution and gradually increases the resolution, stabilizing the learning process and improving image quality. Key contributions of ProGAN include:

- Fade-in layers for smooth resolution transitions.
- Mini-batch standard deviation layers to enhance image diversity.

While ProGAN demonstrated significant improvements, it remains computationally expensive for high-resolution image generation.

2.1.2 AutoGAN: Neural Architecture Search for GANs

[**15**]AutoGAN automated the design of GAN architectures using Neural Architecture Search (NAS). By employing reinforcement learning, AutoGAN identified optimal architectures for both the generator and discriminator. Notable advantages of AutoGAN include:

- Automated architecture optimization.
- Competitive performance without manual tuning.

However, the NAS process is computationally intensive, limiting its scalability.

2.1.3 StyleGAN2: State-of-the-Art GAN Architecture

StyleGAN2 introduced significant refinements over its predecessor, including:

- Removal of instance normalization in the generator for better feature preservation.
- Path length regularization to encourage smooth latent space interpolation.
- Weight demodulation to eliminate artifacts in generated images.

StyleGAN2 set new benchmarks for high-quality image synthesis but requires extensive computational resources and large datasets.

2.2 Challenges in GAN Training

Despite the advancements in GAN architectures, several challenges remain:

- **Limited Data Availability:** Most GANs require extensive datasets to achieve optimal performance, limiting their applicability in low-data regimes.
- **High Computational Costs:** Methods like ProGAN and StyleGAN2 involve significant training time and resource consumption.
- **Rigidity of Fixed Architectures:** Traditional GANs rely on static architectures, which may not adapt well to varying dataset complexities.

2.3 Re-GAN: A Novel Approach

[**1**]To address these challenges, Re-GAN introduces dynamic sparse-dense training, leveraging mask-based parameterization for network pruning and regrowth during training. Key innovations of Re-GAN include:

- **Dynamic Pruning:** Prunes low-magnitude weights to reduce computational overhead while maintaining model efficiency.
- **Connection Regrowth:** Regrows pruned connections during training to preserve representational capacity and prevent premature loss of critical connections.
- **Equilibrium Maintenance:** Prunes only the generator while expanding the discriminator, ensuring stability during training.

ReGAN is compatible with various architectures, including ProGAN, StyleGAN2, and AutoGAN, and demonstrates robustness across datasets like CIFAR-10, Tiny-ImageNet, and FFHQ.

2.4 Summary

This literature review highlights the evolution of GANs, their limitations, and the novel contributions of Re-GAN. By addressing issues such as data scarcity, high computational costs, and rigid architectures, Re-GAN sets a new benchmark for efficient and scalable GAN training. Its dynamic pruning and regrowth methodology opens new possibilities for advancing generative modeling techniques, especially in resource-constrained environments.

Chapter 3

Methodologies

3.1 Methodology

3.1.1 Design Motivation

The human brain's synaptic topology is a dynamic system that enables stable and efficient learning through continuous rewiring of connections.[**19**][**20**] This biological process of adapting synaptic connectivity inspires our approach to GAN training, where the network structure evolves dynamically instead of remaining static. Conventional GANs rely on predefined architectures throughout training, but we introduce a novel method that prunes less significant connections and regrows them adaptively to optimize performance.Through extensive experimentation, we demonstrate that this dynamic approach yields significant advantages:

- Improves stability across diverse datasets, resolutions, domains, and architectures.
- Produces samples of equal or higher quality compared to traditional methods.
- Enhances data and computational efficiency by eliminating redundant connections.

In the next sub-section, we review traditional GAN models and describe our proposed GAN training methodology via architectural reconfiguration.

3.1.2 Dataset

The following datasets were used to evaluate the performance of the ReGAN framework:

- **CIFAR-10**: The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. It is commonly used for benchmarking image classification and generative models. Pytorch automatically downloads the CIFAR-10 dataset if it is not detected locally, ensuring seamless integration into the training pipeline.
- **Tiny-ImageNet**: The Tiny-ImageNet dataset is a subset of the ImageNet dataset, containing 200 classes with 500 training images, 50 validation images, and 50 test images per class. This dataset provides a challenging

environment for image generation tasks, with higher-resolution images than CIFAR-10. [**16**].

- **FFHQ (Flickr-Faces-HQ)**: The FFHQ dataset contains high-quality images of human faces, with a focus on diversity in age, ethnicity, and background. It is widely used for evaluating generative models, particularly for image generation tasks in facial datasets. [**17**].
- **Few-shot Dataset**: The Few-shot dataset consists of a limited number of images per class, making it particularly suitable for testing the efficiency and effectiveness of GANs in low-data regimes.[**18**].

By using these datasets, we aim to demonstrate ReGAN's capability to handle a wide range of image generation tasks, from small-scale datasets like CIFAR-10 to more complex, high-resolution datasets like FFHQ. The datasets were chosen to test the model's robustness across varying levels of data availability and image complexity.

3.1.3 Generative Adversarial Networks

A GAN model consists of a discriminator D and a generator G. The training objectives of the D and G can be represented as $\mathcal{L}_D$ and $\mathcal{L}_G$, respectively. The GAN framework can be represented as follows:

$$\max_G \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] \tag{1}$$

$$\min_D \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] \tag{2}$$

where p_z is the prior distribution (e.g., $N(0, I)$) and p_{data} is the real training data used to approximate the data distribution. The notations $\mathcal{L}_D$, $\mathcal{L}_G$, and $\mathcal{L}_G$ in Eq. (1) and (2) represent the mapping functions from which various GAN losses can be derived[**10**].

3.1.4 GANs Training with Architectural Reconfiguration

A networks structure can be represented as a directed acyclic graph (DAG) with a defined sequence of nodes, where each node x represents an input feature and each edge corresponds to a computation unit parameterized by hyperparameters. To introduce flexibility during training, we associate a binary mask variable $m \in \{0, 1\}$ with each computation unit, allowing selective activation or deactivation of connections. This mechanism facilitates dynamic pruning ($m = 1 \to 0$) to remove less significant connections and regrowth ($m = 0 \to 1$)to restore network capacity as needed. Our approach operates within a single-level configuration space tailored for GAN architectures, enabling the model to dynamically adjust its width during training for optimized performance and resource efficiency.

Revisiting the training process of GANs, Re-GAN introduces a novel approach by dynamically reconfiguring the generators architecture to explore various sub-network configurations during training. Initially, the training begins with a fully dense network during a warm-up phase, allowing the model to learn connection weights and assess their importance. To transition to a sparse structure, the least significant weights are pruned based on a predefined criterion. Following this, the pruned connections are selectively reactivated, enabling the network to grow and regain capacity through iterative optimization cycles.

Once the architecture(Fig. 3.1) transitions back from sparse to dense, the newly updated topology is trained until the next connectivity adjustment. The Re-GAN training process is governed by several key factors, including:

- Sparsity distribution
- Update schedule
- Pruning
- Growth

This adaptive approach ensures both efficiency and performance by dynamically altering the networks structure during training.

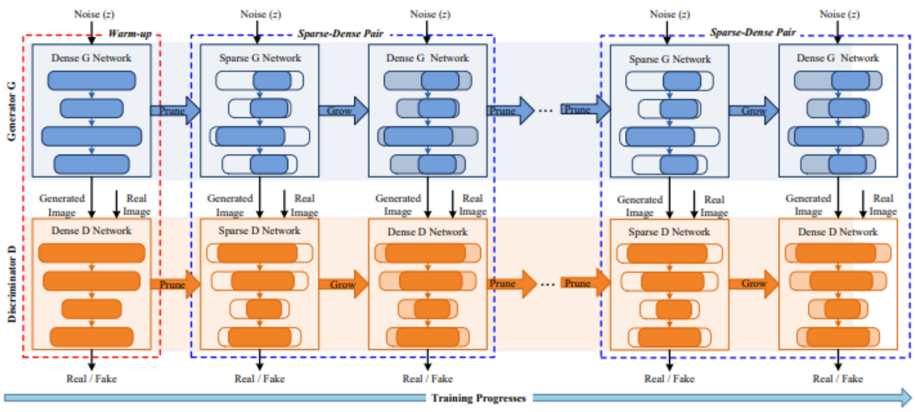

FIGURE 3.1: Architecture of Re-GAN

3.1.5 GAN Sub-networks Exploration

To enhance efficiency and flexibility during training, we dynamically prune low-weight connections using **unstructured magnitude pruning**, guided by binary masks m_G and m_D. The pruning ratio ρ determines the fraction of connections removed, set to $\rho = 10\%$ in most experiments and $\rho = 30\%$ for StyleGAN2 on FFHQ. A uniform sparsity distribution is applied across all layers to ensure consistency. Connections are pruned by sorting weights in ascending order and generating binary masks to exclude weights below a threshold λ. Once individual layer masks are computed, the complete parameter space mask M for both G (generator) and D (discriminator) is obtained. Consequently, the optimization equations for G and D are updated as follows:

$$\max_G \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x, m_G)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z), m_G))] \tag{3}$$

$$\min_D \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z), m_G))] \tag{4}$$

Here, m_G and m_D represent the masks for G and D, respectively, and $\odot$ denotes the Hadamard product applied to incorporate the masks.**[1]**

Update Schedule

The network undergoes iterative pruning and growth guided by two key factors:

- **Update Interval (g)**: Specifies the number of iterations between pruning and growing phases.
- **Learning Rate**: Reduced to $\frac{1}{10}$ of the initial rate during the growth phase to stabilize training dynamics.

Other hyperparameters are retained from the baseline GAN architectures for consistency.

Pruning and Growing Process

During each connectivity update, the weight magnitude is used as a pruning indicator:

- If the mask indicator changes from $1 \to 0$, the corresponding connection is removed from the computational graph.
- In the growing phase, pruned connections are reactivated with their weights initialized to zero, and training resumes.

This alternating process enhances the network's ability to adaptively allocate capacity, allowing it to explore optimal sub-network configurations. The growing phase increases the networks model capacity, aiding in reaching better local minima and avoiding stagnation in suboptimal configurations.

Efficiency Gains

By dynamically pruning less important connections, the training process achieves:

- Reduced memory and computational overhead.
- Faster convergence with minimal compromise on performance.
- Enhanced exploration of the networks architectural search space, improving the quality of generated samples.

Loss Functions

To further stabilize and optimize training, we evaluate three distinct loss functions for the discriminator:

- **WGAN-GP**: Improves stability and mitigates gradient-related issues.
- **Hinge Loss**: Encourages better adversarial dynamics (used in SNGAN).
- **Non-Saturating GAN Loss with 1-Sided Gradient Penalty**: Facilitates smooth convergence and higher-quality image generation (used in StyleGAN2).

This dynamic approach balances efficiency and performance, establishing a robust training methodology for diverse GAN applications.

Chapter 4

Results and Discussions

4.1 Model Comparison

The models used include StyleGAN2, Re-StyleGAN2, and Re-StyleGAN2 (optimized). The Re-StyleGAN2 model outperformed the StyleGAN2 model in terms of image generation and FID score. Additionally, the Re-StyleGAN2 (optimized) also outperformed StyleGAN2, providing somewhat similar yet better results.

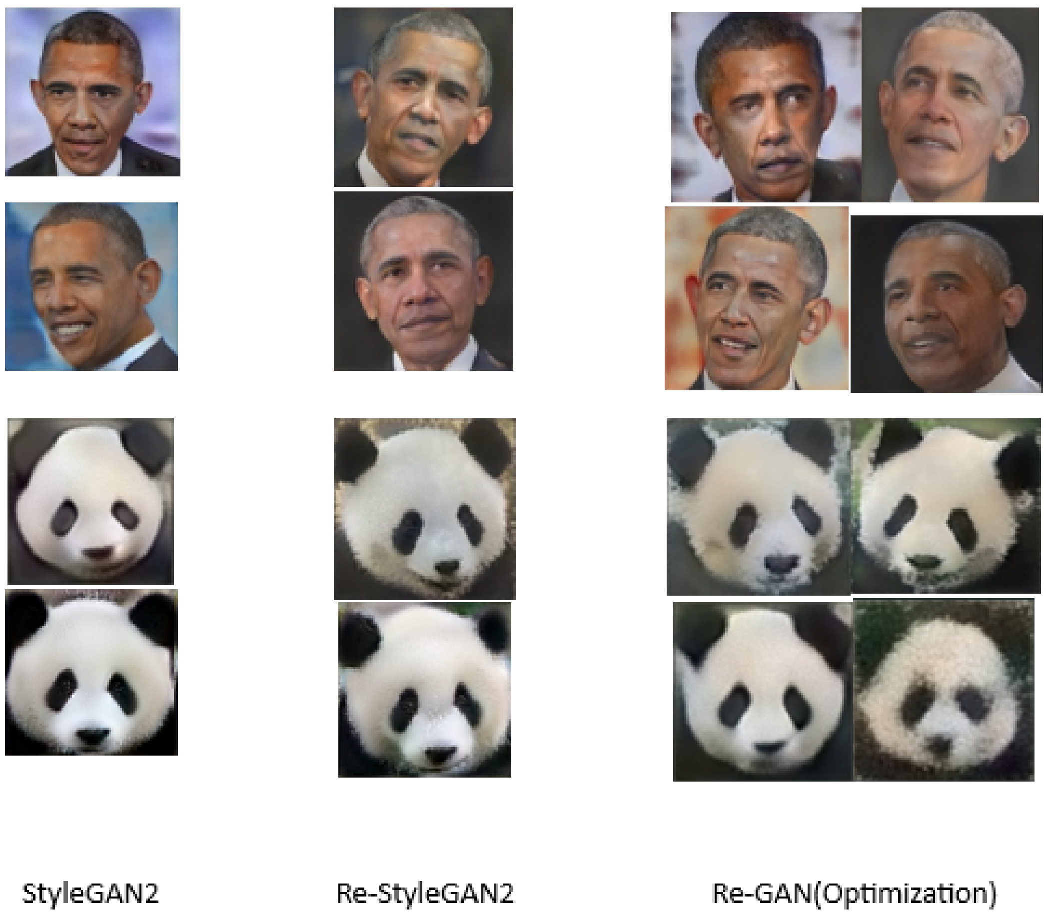

FIGURE 4.1: Output of different model StyleGAN2, Re-StyleGAN2, Re-StyleGAN2(optimized)

Methods	Obama	Grumpy Cat	Panda
StyleGAN2	86.67	51.3	95.08
Re-StyleGAN2	**72.92**	**38.23**	**78.45**
StyleGAN2+APA	68.74	33.72	18.24
Re-StyleGAN2+APA	64.28	31.74	16.33
StyleGAN2+DiffAug	46.31	28.6	12.89
Re-StyleGAN2+DiffAug	**45.7**	**27.36**	**12.6**
Re-StyleGAN2(Optimized)	**49.2**	**28.3**	**12.8**

FIGURE 4.2: FID comparison on few-shot datasets at 256Œ256 resolution.

4.2 Performance Metrics

4.2.1 Frechet Inception Distance (FID)

The **Frechet Inception Distance (FID)** is a widely used metric for evaluating the quality of images generated by GANs. It measures the similarity between the feature distributions of real and generated images. FID leverages a pre-trained Inception network to extract features and compares them using the Frechet distance (also called Wasserstein-2 distance).

The formula for FID is:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}),$$

where:

- μ_r, μ_g are the mean feature vectors of real and generated images.
- Σ_r, Σ_g are the covariance matrices of real and generated images.
- Tr represents the trace of a matrix.

FID score of model for Obama,Grumpy cat and Panda images were similar to Re-StyleGAN2 model for Obama images we got FID score of 49.2 for Panda we got FID score of 12.8 which is very close to Re-StyleGAN2 which is 12.6.

4.3 Data Preprocessing Insights

- **Image Resizing:** The images are resized to 32x32 pixels to maintain uniform input dimensions for the model. This step is crucial for effective training and inference.
- **Center Cropping:** A center crop of 32x32 pixels is applied to focus on the central part of the images. This ensures consistent input size while retaining important features.
- **Tensor Conversion**: Images are converted to PyTorch tensors, which are necessary for compatibility with deep learning models. This format facilitates efficient computation.

- **Normalization:** The images are normalized using specified mean and standard deviation values. This process helps the model to converge faster and enhances overall performance during training.

4.4 Challenges

Training GANs presents several challenges, particularly in ensuring stability and convergence. The dynamic pruning and growing of network connections introduced additional complexity in balancing sparsity and training efficiency. Achieving optimal hyperparameter tuning for pruning ratios and update intervals required extensive experimentation to prevent mode collapse and preserve model diversity. Moreover, the evaluation of generated images, such as maintaining consistent FID scores across datasets, demanded significant computational resources and careful architectural adjustments. Finally, adapting the Re-GAN framework to diverse datasets, including CIFAR-10, Tiny-ImageNet, and FFHQ, required tailored preprocessing and augmentation strategies to handle variations in data resolution and quality.

Chapter 5

Summary and Conclusions

5.1 Key Findings

The proposed Re-GAN framework demonstrated significant improvements in generating high-quality images with better FID scores across datasets like CIFAR-10, Tiny-ImageNet, and FFHQ. The ability to dynamically prune and grow network connections during training enhanced model efficiency while maintaining high image fidelity. Optimal hyperparameter tuning and careful dataset preparation played a critical role in achieving stable GAN training and avoiding mode collapse.

5.2 Contributions

This research introduces a novel approach to dynamic network reconfiguration in GAN training, showcasing its effectiveness in improving training efficiency and image quality. The work highlights the importance of architectural flexibility in GANs, providing a foundation for further exploration of sparse and dense network structures in generative modeling.

5.3 Future Scope

Future work includes extending the Re-GAN framework to larger-scale datasets and diverse domains, such as video generation and 3D modeling. Incorporating advanced pruning criteria and adaptive growing mechanisms will further optimize the network's performance. Additionally, integrating real-time evaluation metrics and deploying the framework in distributed training environments will enable more scalable and efficient GAN training.

5.4 Conclusion

This study establishes the effectiveness of dynamic network reconfiguration in improving the training efficiency and output quality of GANs. By addressing challenges such as stability and optimal hyperparameter tuning, the Re-GAN framework provides a robust approach to exploring sparse and dense sub-networks, paving the way for future advancements in generative modeling.

Chapter 6

References

[1] D. Saxena, J. Cao, J. Xu, and T. Kulshrestha, "Re-GAN: Data-efficient GANs training via architectural reconfiguration," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), June 2023.

[2] M. Arjovsky, S. Chintala, and L. Bottou, Wasserstein generative adversarial networks, in International conference on machine learning, Jan. 2017, pp. 214223, Accessed: Apr. 05, 2019. [Online]. Available: http://arxiv.org/abs/1701.07875.

[3] A. Brock, J. Donahue, and K. Simonyan, Large scale GaN training for high fidelity natural image synthesis, in Proceedings of International Conference on Learning Representations, 2019

[4] I. J. Goodfellow et al., Generative adversarial nets, in Advances in Neural Information Processing Systems, 2014, vol. 3, no. January, pp. 26722680, doi: 10.3156/jsoft.29.5_177_2.

[5] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, Least Squares Generative Adversarial Networks, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 28132821, doi: 10.1109/ICCV.2017.304.

[6] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, Analyzing and improving the image quality of stylegan, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 8110 8119.

[7] S. Zhao, Z. Liu, J. Lin, J. Y. Zhu, and S. Han, Differentiable augmentation for data-efficient GAN training, in Advances in Neural Information Processing Systems, 2020, pp. 75597570.

[8] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, Training generative adversarial networks with limited data, in Advances in Neural Information Processing Systems, 2020, vol. 33, pp. 1210412114.

[9] Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang, Image augmentations for gan training, arXiv Prepr. arXiv2006.02595, 2020.

[10]J. N. M. Kalibhat, Y. Balaji, and S. Feizi, Winning lottery tickets in deep generative models, in Proceedings of the AAAI Conference on Artificial Intelligence, 2021, vol. 35, no. 9, pp. 80388046

[11]X. Chen, Z. Zhang, Y. Sui, and T. Chen, GANs Can Play Lottery Tickets Too, arXiv Prepr. arXiv2106.00134, 2021, [Online]. Available: http://arxiv.org/abs/2106.00134.

[12]T. Karras, T. Aila, S. Laine, and J. Lehtinen, Progressive growing of GANs for improved quality, stability, and variation, in Proceedings of International Conference on Learning Representations, 2018.

[13]T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida,Spectral normalization for generative adversarial networks, in Proceedings of International Conference on Learning Representations, 2018.

[14]X. Gong, S. Chang, Y. Jiang, and Z. Wang, AutoGAN: Neural architecture search for generative adversarial networks, in Proceedings of the IEEE International Conference on Computer Vision, 2019, vol. 2019-Octob, pp. 32233233, doi: 10.1109/ICCV.2019.00332.

[15]AutoGAN. https://github.com/VITA-Group/AutoGAN.

[16]https://www.kaggle.com/c/tiny-imagenet

[17]https://github.com/NVlabs/ffhq-dataset

[18]https://github.com/odegeasslbc/FastGAN-pytorch

[19]A. J. G. D. Holtmaat et al., Transient and persistent dendritic spines in the neocortex in vivo, Neuron, vol. 45, no. 2, pp. 279291, 2005.

[20]D. D. Stettler, H. Yamahachi, W. Li, W. Denk, and C. D. Gilbert, Axons and synaptic boutons are highly dynamic in adult visual cortex, Neuron, vol. 49, no. 6, pp. 877887, 2006.