Comparative Analysis of conditional Generative Adversarial Network models

Xu Huirong 20074688D
Supervisor: Prof. Huang Jian

Hong Kong Polytechnic University
Department of Applied Mathematics

April 2024

Table of Contents

List of Figures

Abstract

Generative Adversarial Nets (GANs) [11] is one powerful tool of machine learning designed to solve generative problems, which achieves state-of-art performance in generating images. GANs have many advantages such as it can generate high-quality images without an incremental generation process. However, it suffers from problems such as there is no way to control on modes of the data that is generated [11]. To solve this issue, Mehdi and Simon proposed Conditional Generative Adversarial Nets[18], which feed the data $\mathbf{y}$ to both generator and discriminator to generate images according to specific labels or give multiple tags for pictures in multi-modal learning.

To further enhance the training of GANs, this project explores details of Auxiliary Classifier GANs (ACGANs), which shows that adding one auxiliary decoder network to the discriminator could greatly improve the quality of images and also produce high-resolution images[21], results and assessment of picture diversity would be discussed.

Apart from AC-GANs, this project investigates projection discriminators in cGANs. Instead of embedding conditional vectors to the feature vectors like the traditional cGANs, it used a parameterized discriminator with an inner product between condition $\mathbf{y}$ and the feature vector[19]. This project will describe the experimental details and comparison of both Projection GANs and ACGANs, with a thorough comparison providing important information for future research paths.

Acknowledgment

Throughout this project, I have received much support from the Department of Applied Mathematics. Firstly, I would like to express my gratitude and appreciation to my supervisor, Prof. Huang Jian, for his supervision, patient help, and insightful guidance throughout the Final Year Project.

Next, I'd like to thank my mentor, Ph.D. Niu Shengjie, for his regular support and directions offered for this project. His knowledge was really helpful in getting this project started.

At last, thanks to all the IT staff in PolyU and AMA for their continuous help with my server setup and debugging, without which I could not get the project done.

1 Introduction

1.1 Background

Generative learning and discriminative learning are two fundamental approaches in the field of machine learning. They aim to solve different problems and have various structures for future applications, such as GANs.

Discriminative models belong to the field of supervised learning. It sorts the data points into distinct classes and uses maximum likelihood and probability estimations to learn the boundaries. Its objective is to model, given the input features, the conditional probability distribution of the labels. Widely used discriminative models include logistic regression, decision trees, support vector machines, and so on.

However, for the generative models, it goes deep to learn the actual data distribution instead of the boundaries to do the distinguishing process. Through the process of capturing the fundamental structure of the data, generative models can produce new samples that align with the distribution of training data, which is essential for tasks like image synthesis.

Still, the traditional generative models, like Gaussian Mixture models, failed to capture the complex data distribution as well as the high-dimensional data distributions found in actual data. Goodfellow et al. (2014)[11] proposed Generative Adversarial Networks (GANs), which utilized generative models to avoid having to deal with the challenge of approximating a lot of complicated probabilistic computations[18]. GANs could generate convincing image samples on datasets with low variability and low resolution[8][22]. The discriminator $D(x)$, which assesses the divergence between the target distribution $q(x)$ and the current generative distribution $p_{\mathrm{G}}(x)$ is the most characteristic feature of GANs[20][2] and will be emphatically introduced in the later parts.

Still, GANs suffer from not being capable of generating high-resolution and coherent pictures. Mirza and Osindero proposed Conditional Generative Adversarial Nets (cGANs)[18], which tried to advance generative models and make them able to generate images or tags according to specific tasks. It has been proven to be a useful tool for tasks like class conditional image generation. Now that there are an increasing number of applications for cGANs. CGANs with auxiliary classifiers[21] introduced some modifications to the latent space to produce high-quality samples and came up with a new metric for accessing variability. CGANs with projection discriminator by

Miyato and Koyama[19]are also worth discussing for their parameterized discriminator and different structures from the normal cGANs.

Employing this goal-oriented study, we hope to expand on our knowledge of two models in conditional generative adversarial networks and offer an objective evaluation of their structure and performance. We also hope to fill in gaps in the literature and provide a strong groundwork for future advancements and uses of generative modeling.

1.2 Objective

The main goal of this research is to investigate two promising approaches in the field of Conditional Generative Adversarial Networks (cGANs) to clarify the various mechanisms and theoretical foundations that drive the capabilities and performance of these methods. These goals are as follows:

1. **Comprehensive Exploration:** Examine two cutting-edge cGAN approaches in-depth, paying particular attention to the distinct techniques they use to produce conditional images.
2. **Methodological Comparison:** Make a comparison between the chosen cGAN approaches to determine their respective advantages and disadvantages. This comparison will be based on qualitative and quantitative metrics.
3. **Experimental Interpretation:** Give an interpretation of the experimental results of each cGANs model and combine it with its methodology to have a deeper discussion.
4. **Identification of Advantages and Challenges:** Combine results to describe current benefits and difficulties with cGANs. This combination will help to suggest possible future research directions, such as model modifications and theoretical developments.

1.3 Limitations

The lack of experience with machine learning training on a server was one of the main obstacles this project faced. Because of this limitation, a large amount of the project schedule was devoted to making sure the models were running fine, instead of optimizing the parameters for best results. Consequently, it is possible that the ACGANs and Projection GANs' performance was not entirely optimized. Furthermore, the time limitations of the project made it more difficult to test and modify the models' parameters in order to perhaps produce better results.

Also, generating more images of different kinds would be of great help to test models' performance. However, due to time limitations, this goal is also not achieved.

The second limitation is the source of the code. For the Projection GANs, it provides the official code, but for the ACGANs there's no official code provided. Since there was no official implementation, I had to use code that was made available to the public and then modify it to fit my experimental framework. These community-contributed codebases are very helpful, but they might not be able to capture all the subtleties or hyperparameters used by the original paper. As such, it is possible that the generative capabilities and performance metrics we saw in our experiments do not accurately capture the potential of ACGANs as outlined in their seminal publication.

1.4 Source of Data

Our data source for this Final Year Project was the Imagenet dataset[6]. Developed and curated by researchers at Stanford University and Princeton University, the ImageNet dataset is one of the largest and most diverse image collections available for use in artificial intelligence research. It consists of over 14 million images, each labeled with one of 20,000 categories.

2 Related Work

2.1 Generative models

As a subclass of unsupervised learning, generative models seek to produce new data points with some variability by understanding the training set's true data distribution. These models are frequently utilized in many domains, including natural language processing, and computer vision.

Variational auto-encoders (VAEs), Generative adversarial Networks (GANs), and autoregressive models are the most widely used generative models. These models have demonstrated success in producing realistic, high-quality samples that are nearly identical to real data.

The two primary types of image-generative models that have been thoroughly examined are non-parametric and parametric. Non-parametric models use training image patches to replicate various functions such as texture generation[9] and super-resolution[10]. Generative parametric models are used in an extensive variety of deep learning techniques. For example, generative decoders are used in restricted Boltzmann machines, and denoising auto-encoders are used to rebuild the image from its latent form.[7]

2.2 Generative Adversarial Nets

Generative Adversarial Nets were first introduced by Goodfellow et al.[11] in 2014. It developed a deep implicit generative model that could produce real samples in a single generation step without relying on the incremental generation process or Markov chains.

The theory behind GANs is game theory. One network is called the *generator*; it has a prior distribution $p(z)$ over a vector z, which is the input to this generator. Another input to the *generator function* is the trainable parameter θ, which controls the criteria used in the game. Typically, the prior distribution $p(z)$ is an unstructured distribution, like a uniform distribution or a high-dimensional Gaussian distribution; samples z from this distribution are merely noise[11]. The main role of the generator

is to transform the mere noise z into realistic examples.

Another important component is the discriminator. It has a second multilayer perceptron $D(x, \theta)$, which will output a single scalar. The likelihood that x originated from the data instead of p_g is represented by $D(x)$. Theoretically, the GANs is just a min-max game between the generator and the discriminator. The generator tries to produce samples that the discriminator cannot tell if they're from the real world or the generator. On the other hand, the discriminator makes efforts to distinguish between generated samples and real samples; it is trained to assign both generated labels and real labels correctly to the two clusters, while the generator minimizes $\log(1 - D(G(z)))$. They play against each other to obtain the following min-max function:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \quad (1)$$

High-quality, realistic data is produced by the generator through this adversarial process. GANs have proven effective in several applications, such as image synthesis, super-resolution, and style transfer.

2.3 Conditional Generative Adversarial Nets

Conditional Generative Adversarial Nets were introduced by Mirza and Osindero in 2014[18]. It is constructed by feeding the data with condition y. y could be class labels or information from other modalities, and y is both constructed on the generator and discriminator. The min-max function of Conditional Generative Adversarial Nets can be expressed as:

$$\min_G \max_D \mathbb{E}_{(x,y) \sim p_{\text{data}}(x,y)}[\log D(x, y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z, y)))] \quad (2)$$

The mode collapse issue that frequently arises with GANs is resolved by this conditioning. It also enables the model to produce data according to specific tasks or labels. CGANs have been applied to several tasks, such as attribute-based face generation, text-to-image synthesis, and image-to-image translation.

3 Project Schedule

Stages	Tasks	Time Allocation
Research and topic choosing	– Discuss with supervisor, searching for the best topic. – Get fundamental knowledge about GANs and their application. – Identify necessary datasets and software that will be used for this topic.	2 months
Setting up software and accessing GPU	– Contact IT staff in AMA, require GPU access. – Install and configure the necessary packages and set up the environment.	1 month
Interim report writing	– Select and read papers about ACGANs and projection GANs, understanding their methodology. – Confirm the structure of the report and the parts that should be finished in the interim report. – Summarize and compare; write the interim report.	1-2 months
Code implementation	– Training and testing on existing code in ACGANs and Projection GANs. – Fine-tuning and modification of existing code. – Refine and optimize existing methods. – Compare and document the result.	1-2 months
Final reporting	– Finish the final report, summarizing all the works and outcomes. – Make use of more related materials to enrich the reports and dive deeper into the discussion.	1 month

Table 1: Project Schedule

4 Resource Utilization

4.1 Hardware Resources

The project used the GPUs from Prof. Huang Jian's GPU workstation, which has four A100s. Also, to access the server outside campus, I applied for a research VPN from AMA.

4.2 Software Resources

Python was the main language chosen for its relative easiness of use and its robust performance.

5 Methodology

5.1 ACGANs

Conditional GANs with Auxiliary Classifier(ACGANs) was introduced by Odena et al. in 2017[21]. Its biggest adjustment to existing GANs is to add more structure to the latent space and also utilize a specialized cost function. It could help to generate images with both high resolution and increased global coherence[21].

5.1.1 Architecture of the ACGANs

Different from the original GANs, except for the noise z to be put in the generator, each generated sample in ACGANs has a class label c, which is essential to generate images of certain classes. The generator uses both the noise and label to generate images $X_{\text{fake}} = G(c, z)$. A probability distribution over sources $P(S|X)$ and a probability distribution over the class labels $P(C|X)$ are both provided by the discriminator.

The generator and discriminator networks, along with their corresponding loss functions, are simultaneously optimized during the training phase of ACGANs. The generator is responsible for generating images that match certain class labels and seem realistic. The discriminator, on the other hand, is in charge of accurately categorizing the images into their appropriate classifications and differentiating between actual and fraudulent images.

The class log-likelihood L_c, penalizes the discriminator for making inaccurate class predictions, whereas the source log-likelihood L_s, penalizes it for improperly distinguishing between real and false images. The discriminator's accuracy in classifying the fictitious images and their given labels determines the generator's loss.

$$L_s = \mathbb{E}[\log P(S = \text{real}|X_{\text{real}})] + \mathbb{E}[\log(P(S = \text{fake}|X_{\text{fake}})] \tag{3}$$

$$L_c = \mathbb{E}[\log P(C = c|X_{\text{real}})] + \mathbb{E}[\log P(C = c|X_{\text{fake}})] \tag{4}$$

The goal of the discriminator D is to maximize the $L_s + L_c$, while the generator G aims to maximize the $L_c - L_s$. Also, different from the original GANs, the discrim-

inator of the ACGANs outputs the probability of each class label instead of merely probability distribution over sources.

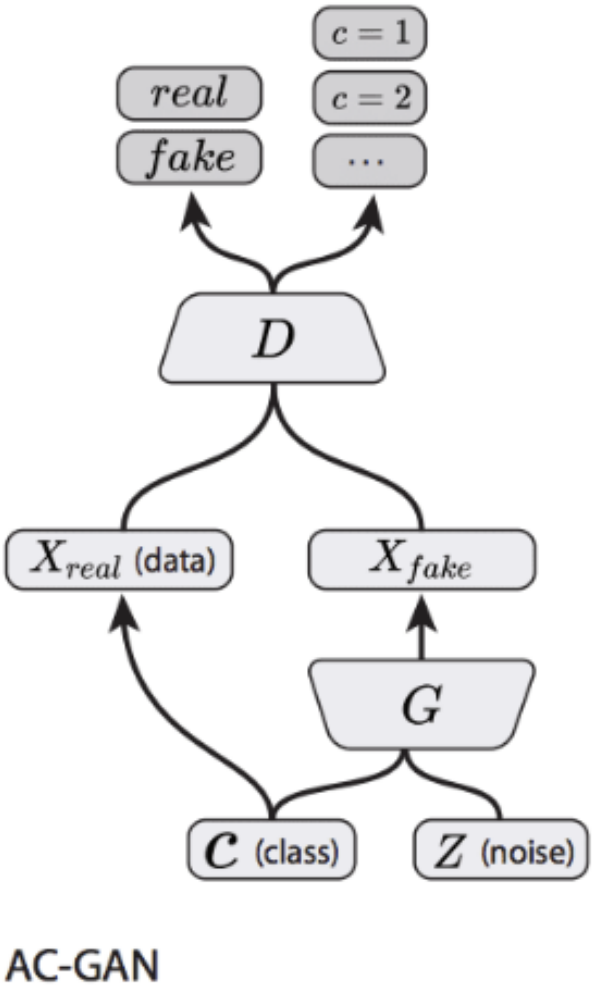


Figure 1: Schematic diagram of the ACGAN

Gradient descent and backpropagation techniques are used to update both networks. By producing realistic and precisely annotated images, the generator hopes to trick the discriminator and reduce its loss. On the other hand, by correctly classifying and differentiating between actual and fraudulent images, the discriminator aims to maximize its loss. Until the networks achieve an equilibrium—where the discriminator correctly detects and categorizes real and fake images and the generator generates high-quality, class-specific images—this adversarial process will continue.

5.1.2 Importance and application of ACGANs

This minor modification to the existing GANs can make training more stable and produce images with higher quality, due to the existence of class labels[21]. Also, ACGANs is able to produce images with more variety[26]. The biggest contribution of ACGANs lies in that it can produce images within a specific class, which is essential for the controlled generation of images.

Applications for ACGANs can be found in a variety of fields. They are employed in image-to-image translation, which transfers images between domains while maintaining properties unique to each class[15]. Another application of ACGANs is data augmentation, where they provide more training data for underrepresented classes[1]. Moreover, ACGANs are applied to domain adaptation in order to align the target and source domains' feature distributions[24].

5.2 Projection GANs

cGANs with Projection Discriminator (Projection GANs) was introduced by Miyato and Koyama in 2018[19]. Its biggest improvement to the conditional GANs is that it combines the conditional information into the discriminator in the cGANs by using the projection. This improvement leads to higher quality and only uses one pair of discriminator and generator.

5.2.1 Architecture of Projection GANs

Firstly, let's denote the input vector by $\mathbf{x}$ and the information given at first by $\mathbf{y}$. Also, let the p and q be the distribution of the generator model and the true distribution. The loss function of the discriminator can be expressed as

$$L_D = -\mathbb{E}_{x\sim q(x)}[\log D(x,y)] - \mathbb{E}_{\tilde{x}\sim p(x)}[\log(1 - D(\tilde{x},y))] \tag{5}$$

After some transformation, the likelihood function can be written as

$$f(x,y) := \mathbf{y}^\top \mathbf{V} \cdot \phi(x) + \psi(\phi(x)) \tag{6}$$

Where $\mathbf{V}$ is the embedding matrix of $\mathbf{y}$, and the $\phi(x)$ is the transformation of x to be put in the last layer of this network, $\psi(x)$ is a function containing some linear transformation of the $\phi(x)$. Since the proof is not the main focus of this Final Year Project, proof is omitted.

From equation (6) we can see that the Projection GANs combines the information with the input vector using the inner product. Its intricate design integrates class labels into the discriminator's decision-making process. Through a more direct and subtle interaction between the label information and picture attributes, this strategy improves the discriminator's capacity to verify the validity of generated images in a class-specific manner.

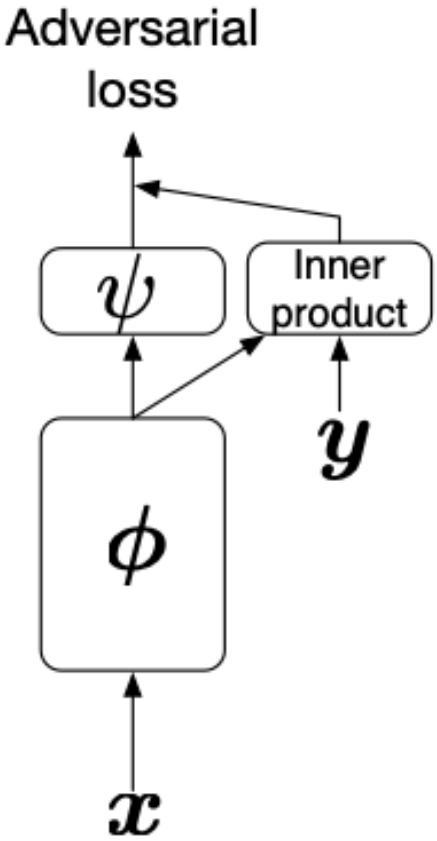


Figure 2: Schematic diagram of the Projection GANs

5.2.2 Importance and application of Projection GANs

The unconventional way that Projection GANs incorporate conditional information into the GAN architecture has attracted a lot of interest from the machine learning community. Projection GANs are unique in that they can directly project class labels onto the discriminator's feature representation of the input, which improves the model's ability to do conditional generation tasks[19]. A more expressive and adaptable model with increased efficacy in learning intricate and high-dimensional data distributions is the outcome of this projection process.

One of the important applications of the Projection GANs is to do the image synthesis. Since it can embed the information directly into the feature space. The model is especially useful for computer vision tasks where fine-grained detail is essential because of its ability to incorporate auxiliary information, which enables the creation of detailed and class-specific images[16]. Additionally, scientists are trying to utilize the Projection GANs and apply it to more semi-supervised learning scenarios[4].

Also, Projection GANs acts importantly in the field of domain adaptation, where models should be generalized across various data distributions. The innate capability of the Projection GANs to work with conditional information is an effective means to bridge domains. It allows for more smooth transitions and increases the adaptability of learned representations[24].

5.3 Comparison

5.3.1 Architectural Differences

Auxiliary Classifier By adding auxiliary categorization to the discriminator, Generative Adversarial Networks (ACGANs) offers a sophisticated design that expands upon the conventional GANs framework[21]. With the help of this extra classification network, the discriminator can categorize the images into the appropriate categories in addition to distinguishing between authentic and fraudulent images.

Also, the ACGANs can generate two outputs, one is the class prediction and the other is the authenticity signal. Each of them has a corresponding loss function that acts as a guide for the training process.

Different from the ACGANs, the Projection GANs present a novel strategy by directly projecting the class labels into the discriminator[19]. It eliminates the need for a separate classifier. The inner product between the feature space and the class is part of the discriminator's output, which is different from the ACGANs [3].

Comparing those two methods in their architecture, ACGANs is a straightforward addition to the GANs model, which makes it easier to adapt to the trending GANs architecture and implement [12]. But if the added auxiliary classifier is not handled correctly, it may lead to model collapse or over-fitting easily. For the Projection GANs, it may require more effort in the fine-tuning process to make sure the projection operates correctly.

5.3.2 Conditioning Mechanism

ACGANs provide class information to both the generator and the discriminator, its objective is to provide information such that it can generate images of a specific class. As for the projection GANs, it offers a more subtle combination of the class labels to the discriminator's feature space[19].

Since ACGANs explicitly uses the class labels alongside the generation process, also with the help of the auxiliary classifier, it is more adaptable when it comes to handling the conditional generation jobs[21]. At the same time, a more comprehensive

application of the class information is the result of Projection GAN's method. The projection process raises the fidelity and the diversity of the synthesis images[19].

5.3.3 Objective Functions

For ACGANs, it separates the objective function of the correct class and the real image. Its goal for the discriminator and generator can be clearly specified as below:

$$L_s = \mathbb{E}[\log P(S = \text{real}|X_{\text{real}})] + \mathbb{E}[\log(P(S = \text{fake}|X_{\text{fake}})] \tag{7}$$

$$L_c = \mathbb{E}[\log P(C = c|X_{\text{real}})] + \mathbb{E}[\log P(C = c|X_{\text{fake}})] \tag{8}$$

The goal of discriminator D is to maximize the $L_s + L_c$, while generator G aims to maximize the $L_c - L_s$. At the same time, Projection GANs provides an innovative

method of adding class information to the discriminator. It eliminates the need for a separate auxiliary classifier. Its objective function is represented as below:

$$L_D = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\min(0, -1 + D(x, y))] + \mathbb{E}_{z\sim p_z(z), y\sim p_{\text{data}}(y)}[\min(0, -1 - D(G(z, y), y))] \tag{9}$$

$$L_G = -\mathbb{E}_{z\sim p_z(z), y\sim p_{\text{data}}(y)}[D(G(z, y), y)] \tag{10}$$

For ACGANs, although it can generate images with high diversity, it suffers from the fact that the loss of the auxiliary classifier may outweigh the adversarial loss, which could result in a deterioration of the image quality.

6 Results and Findings

6.1 Implementation and Analysis of ACGANs

I adapted code from the GitHub [5] since the author did not provide its official code. I trained ACGANs on the imagenet dataset[6]. 50,000 epochs were used in the training process.

One big change made here is that I changed the objective of the code from producing bird pictures to producing dog pictures. The goal here is also to make it more comparable to Projection GANs. There are a few changed processes in the code:

6.1.1 Coding modification

- **ACGAN Architecture**: customizing the generator (_netG) and discriminator (_netD) networks to work with dog images.
- **Dataset preparation**: Leveraging the folder IDs of the dogs in the Projection GANs, I manually selected all the folders containing dog images, and then organized them for future processes.
- **DataLoader**: applied transformation such as image resizing, normalizing, and similar techniques on dog images
- **Training process**: Made it meet our objective to produce dog images
- **Visualization process:** Changed the main function to make it output generated images every 1000 epochs for visualization and future check.

6.1.2 Analysis of experiment result

In the experiment, the generator G is some layers of convolution, which change the noise and class labels to generated images. I trained the network on 128×128 resolutions. Leaky ReLU nonlinearity characterizes the discriminator D, which is a deep convolutional neural network[21].

Figure 3: Generated image from ACGANs

6.1.3 Model collapse problem

One serious problem encountered during the training of ACGANs was the model collapse problem. It refers to the problem that the generator starts to produce only a limited variety of generated images. This happens because the generator finds that certain kinds of images can fool the discriminator perfectly, so it will discard the variety in production and only focus on producing images that can perfectly fool the discriminator. In this case, GANs fails to capture the true distribution of the dataset images.

I did not encounter a model collapse problem during the training of Projection GANs, but I encountered this problem around the 4,000 epochs when training the ACGANs. The result when the model collapsed was like this:

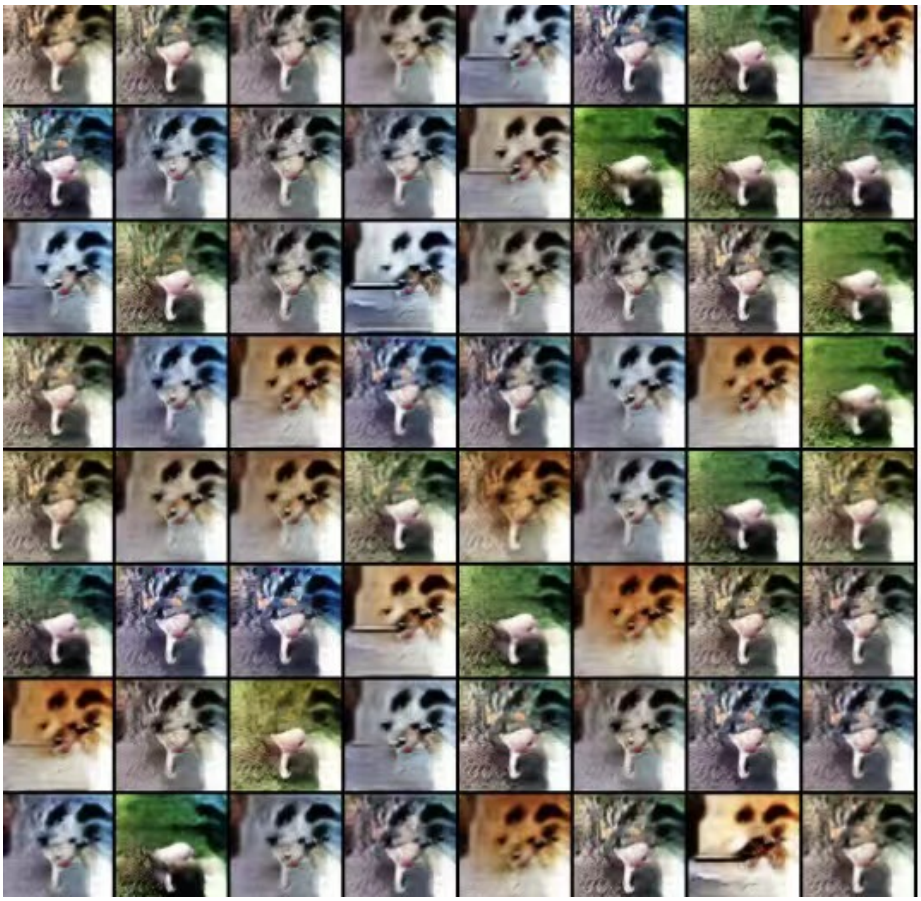


Figure 4: Generated images when model collapsed

As can be seen from the picture, the generator generated pictures in limited variation, which the discriminator may find hard to tell if it's fake or not. But lack of variety is obviously obeying our objective of generative models.

6.2 Implementation and Analysis of Projection GANs

The code of the Projection GANs was provided by the author of the paper. Firstly I followed the readme file to do the data pre-process. It contains steps like unzipping the dataset and grouping the images by their core features. After the experiment on pre-trained models to generate class-specific images, I used the network to do the training process.

As stated in the paper, the discriminator is based on the ResNet network[14] for its stability and flexibility in training. Also, the code adapts the generator used by Gulrajani et al.[13]. As the author did in the paper I used hyper-parameter $\alpha = 0.0002$, $\beta_1 = 0$, and $\beta_2 = 0.9$ for the Adam optimizer[17]. The discriminator was updated five times faster than the generator. Different from ACGANs, which used a pair of discriminators and generator for each 10-size class, the Projection GANs used a single pair of resnet for generator and discriminator[19].

6.2.1 Analysis of experiment result

Figure 5: Generated image from Projection GANs

The picture above is the result generated after the 250000 epochs from Projection GANs.

6.3 Comparison of the results from two methods

6.3.1 Model collapse

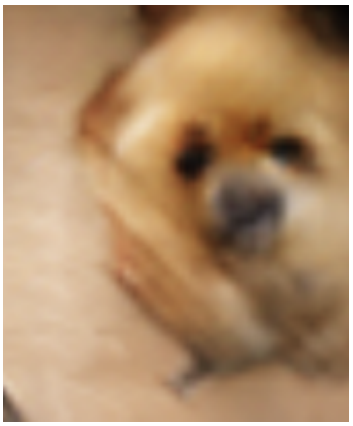
Firstly, one obvious advantage of Projection GANs is that it does not suffer from model collapse easily. It can perform image generation with high speed and high stability. However, the ACGANs will collapse when the number of epochs increases, this can be solved by techniques like early stopping. The model stability is not guaranteed in the ACGANs.

6.3.2 Visual comparison

Figure 6: Projection GANs(Left) and ACGANs(Right)

From the visual result of both dog images, we can see that, the Projection GANs has more vivid pictures with less abstract features.

(a) Abstract result of ACGANs



(b) Vivid result of ACGANs

(c) Abstract result of Projection GANs

(d) Vivid result of Projection GANs

Figure 7: Vivid and abstract pictures from two models

From the selected vivid and abstract pictures from both models we can see that the Projection GANs can generate pictures with more details such as the shade of the hair and background, while the ACGANs tend to use smooth color blocks to composite dog pictures, it may lack some details.

6.3.3 Variation across pictures

The variation across pictures stands for how the produced pictures are different from the original pictures in the dataset. This is a crucial metric for evaluating the performance of a generative model. We tend to use a model that can produce vivid pictures that vary from the original dataset.

Created by Wang et al., a technique for calculating how similar two photos are to each other is the Structural Similarity Index (SSIM)[25].

By taking changes in structural information, luminance, and contrast into account, SSIM is intended to provide a more perceptually relevant measure of image quality than conventional techniques like Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR), which may not correlate well with human visual perception. SSIM takes a value between -1 to 1, where 1 stands for perfect similarity[23].

I tested the SSIM of generated pictures from two models against the original dog pictures from the imagenet dataset. The result is presented in the table below:

Method	SSIM
Projection	0.1805
ACGANs	0.1791

Table 2: SSIM Comparison

From the table, we can see that their SSIMs are very close, which means their similarity with the original dog dataset is close. However, the SSIM of ACGANs is slightly lower than the Projection GANs's, which means that the ACGANs has slight variations in structural information, luminance, or contrast when compared to the original dataset. Nevertheless, this difference might not be noticeable to the naked eye because of the tiny variation in SSIM values.

6.4 Future work

Although GANs can generate images with high quality and fast speed without the use of backpropagation, there's still room for improvement. Given the theoretical analysis and experiment results, some topics might be worth discussing in the future.

1. **Improving Model Stability:** One big challenge faced by GANs models is that they suffer from model collapse easily. The future development for the ACGANs might be to adapt its method to a stable path. For example, it can be combined with the ideas in Wasserstein GANs [2], which uses the Wasserstein distance

as the loss function instead of traditional cross-entropy. This adaptation can greatly increase the model's stability.

2. **Higher Resolution Generation:** In the future, scientists could search for solutions to produce high-resolution images, which would be beneficial for applications requiring detailed image synthesis. This trend can be adapted to the future direction of technological product development, such as higher-definition electronic displays.

3. **Various Conditions:** Future research could focus on testing how the image synthesis could be done using different kinds of conditions instead of just conditioning on its class. For example, conditioning on background objects, weather, hair color, etc. Conditioning on tiny features could let the conditional GANs be more useful when it comes to some specific tasks only requiring some features on the images.

7 Reference

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. In *arXiv preprint arXiv:1711.04340*, 2017.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] Ting Chen, Bing Xu, Chong Zhang, and Carlos Guestrin. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12154–12163, 2019.

[5] clvrai. Acgan-pytorch. https://github.com/clvrai/ACGAN-PyTorch, 2022. Accessed: 2024-02-27.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[7] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751*, 2015.

[8] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Robert Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, 2015.

[9] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, volume 2, pages 1033–1038. IEEE, 1999.

[10] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *Computer Graphics and Applications, IEEE*, 22(2):56–65, 2002.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. 03 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[18] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[19] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *ICLR*, 2018.

[20] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, pages 271–279, 2016.

[21] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, pages 2642–2651, 2017.

[22] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, 2015.

[23] Hamid R Sheikh, Zhou Wang, Alan C Bovik, and Lawrence Cormack. Live image quality assessment database release 2. http://live.ece.utexas.edu/research/quality/, 2005.

[24] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[25] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[26] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2019.

8 Reflective Writing

I chose GANs model as the topic of my final year project because my research experience is in CUHK. my research in CUHK is mainly focused on deblur in computer vision. This topic is also related to images, so I would like to expand my knowledge of machine learning in images and utilize some of my existing ideas to complete my final-year project. also, GANs models have been a hot topic in machine learning, and I can learn a lot by going deeper into two of them this time. I learned about the main structure of the GANs model and the problems encountered with the model, and I got a deeper understanding of conditional GANs and its applications. For myself, the GANs model is very interesting because it is not only visually appealing, but it also serves as a good introductory model for machine learning. The training is not very difficult and there is a visible demonstration of progress at each step. This inspired me to learn more. Through this final-year project, I have been learning mainly by

reading papers and finding information on the internet. There is some knowledge that I have already stocked up in my original study: such as the test of SSIM index, load data of image model, etc. But how to apply this knowledge to new scenarios is what I need to learn. So I read a lot of articles, focusing on how the authors capture the key points of the model, and what aspects need to be seriously considered. And there are some new problems such as model collapse that I have never encountered before, which requires me to look for information and code solutions. I tried many ways to solve model collapse one by one, such as changing the learning rate, early stopping, changing the batch size, and so on. Eventually, I got better results.

I think the biggest change I've made since entering university is my ability to learn more. In high school filler education, the teacher explains each knowledge point very carefully, students do not need to use their brains, and memorization can do a good job. However, this is not the university case, as there are many knowledge points in each class, and no one will be able to explain them to you one by one, so it is especially important to be able to study and understand them on my own. Gradually, I began to learn to form my knowledge network and summarize. Some of the points I didn't understand in class can be found on web pages/YouTube, and programming questions can be asked on CSDN and ChatGPT. we are now more diversified in our learning, and knowledge can come from all over the world. I have gradually learned to study on my own and how to organize my time wisely.

And, because college doesn't have a very strict schedule telling us what we need to do each day and each time slot, time management will turn out to be a major

factor in determining everyone's grade. I've found a way to relax and stay efficient after adjusting time and time again: I study during the daytime, but I can't study after 9:00 p.m. It's my time to relax and recuperate. I believe that being able to study efficiently for a long period is the key to being able to win, and I am not in favor of not sleeping/staying up late to study before exams. How to stay physically and mentally healthy is also something I learned in college that is very important to me.

For my future study/life plans: I plan to study statistics at Duke University in the US and would like to learn more practical skills. Whether it is programming or modeling, I think it will help me a lot in my future employment. I hope to do some programming in the summer of 2024, like taking Introduction to Programming from Rice University at Coursera, and then some more targeted programming and modeling courses. This will help me in future interviews. I don't think it matters what path I choose, the most important thing is to be able to stick to it and do the best I can without regretting my choice!