IMPROVING TEXT-TO-IMAGE GENERATION WITH MULTIMODAL SEMANTIC COHERENCE IN ADVERSARIAL TRAINING

By

Rahat Afza

Supervisor

Dr.Hammad Afzal

A thesis submitted to the faculty of Computer Software Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad,Pakistan, in partial fulfillment of the requirements for the degree of MS in Computer Software Engineering

June 2023

IMPROVING TEXT-TO-IMAGE GENERATION WITH MULTIMODAL SEMANTIC COHERENCE IN ADVERSARIAL TRAINING

By

Rahat Afza

00000359479

A thesis submitted to the faculty of Computer Software Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad,Pakistan, in partial fulfillment of the requirements for the degree of MS in Computer Software Engineering

June 2023

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS thesis entitled "**Improving Text-to-Image Generation with multi modal semantic coherence in Adversarial Training**" written by Maj. Rahat Afza, Registration No.**00000359479**, of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: _____

Name of Supervisor: **Professor Dr. Hammad Afzal**,

Date: 19/6/2023

Signature: (HoD) _____

Date: 19/6/2023

Signature: (Dean/Principle) _____

Date: 23/6/23

Brig
Dean MCS (NUST)
(Asif Masood, Phd)

Declaration

I, *Rahat Afza* declare that this thesis titled “Improving Text-to-Image Generation with multi modal semantic coherence in Adversarial Training” and the work presented in it are my own and has been generated by me as a result of my own original research.
I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
5. I have acknowledged all main sources of help
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself

Rahat Afza

Rahat Afza,

00000359479

Copyright Notice

Abstract

Research in the field of text to image generation has shown incredible momentum owing to the availability of more powerful natural language processing (NLP) models and generative networks. The quality of data representation learned by the generative models acts as a determining factor in the success of these models. Self-supervised learning augments the generative power of these networks by utilizing the underlying hidden structure of the data for providing supervisory signals. Contrastive learning (CL), a self-supervised technique, has been used in generative models to foster improvements in image to image and text to image (T2I) tasks. Use of generative adversarial networks (GAN) is not nascent in the field of text to image generation. But, GANs suffer from the problem of training instability. In T2I models, the existence of numerous mappings between the image and text captions adds more to this training instability and puts the adversarial loss under another constraint. Several T2I models have been proposed in the literature which have employed CL with the intent to stabilize the GAN training and improve semantic consistency of generated images and textual captions. But most of these models have used stacked architecture as baseline and attention computations for ensuring the semantic consistency of image and text. This setup becomes computationally more expensive as the resolution of the generated images increases. In this work we have employed CL in a single stage GAN for improving the convergence of generative model to a better learnt latent data representation. Comprehensive experiments upon benchmark datasets have shown remarkable improvement in the convergence rate of model when co-related with other similar state-of-the-art models.

This thesis is dedicated to *my beloved parents*

Acknowledgments

I am grateful to God Almighty, the Beneficiant and the most Merciful, who has bestowed me with the strength and the passion to accomplish this thesis.

I would like to convey my gratitude to my guide and support, Dr Hammad Afzal, CSE Department at MCS, who made me able to withstand the difficulties faced in this thesis. I would also extend my gratitude to my Thesis Committee Members for their support and guidance regarding the topic.

Last , but not the least, I am highly thankful to my husband, parents and family. They all have always stood by my dreams and have been a great source of inspiration for me. I would like to thank for all the care, love and support through my times of stress and excitement.

Contents

1 Introduction **1**

1.1 Overview 1

1.2 Motivation and Problem Statement 4

1.3 Objectives 5

1.4 Thesis Contribution 5

1.5 Area of Application 5

1.6 Thesis Organization 6

2 Background **8**

2.1 Deep Learning 8

2.1.1 Biological Intuition 9

2.2 Artificial Neural Networks 10

2.2.1 Activation Functions 11

2.2.2 Loss Function 12

2.2.3 Optimization 13

2.2.4 backpropagation 14

2.2.5 Batch Normalization 15

2.2.6 Residual connection 15

2.3 Convolutional Neural Networks 16

2.4 Generative Models 17

2.4.1 Generative Adversarial Networks (GAN) 18
2.4.2 Autoencoders 20
2.4.3 Diffusion Models 20
2.5 Text Encoders 21
2.5.1 Long Short Term Memory (LSTM) 22
2.5.2 Bidirectional Encoder Representations from Transformers(BERT) 23

3 Literature Review **24**
3.1 Generative Adversarial Network (GAN) 24
3.1.1 Preliminary Methods 24
3.1.2 Stacked Generative Adversarial Networks 27
3.1.3 Attention GAN 29
3.2 Auto-regressive Models 31
3.3 Diffusion Models 31

4 Method **32**
4.1 Introduction 32
4.1.1 DF-GAN 34
4.1.2 Deep Fusion Generator 35
4.1.3 Semantic-Aware Discriminator 36
4.2 Contrastive Learning 37
4.2.1 Architectures 38
4.2.2 Training 38
4.3 Model 39
4.3.1 Objective Functions 41
4.3.2 Contrast Aware Discriminator 42
4.3.3 Generator 43

5 Results and Evaluations **45**

5.1 Introduction 45

5.2 Evaluation Metrices 46

5.2.1 Fre chet Inception Distance (FID) 47

5.2.2 Inception Score (IS) 48

5.3 Model Evaluation 49

5.3.1 Baselines 49

5.3.2 Implementation Details 50

5.3.3 Comprehensive Performance Analysis 52

5.3.4 Performance Visualization 52

5.4 Ablation Study 53

6 CONCLUSION AND FUTURE WORK **56**

6.1 Conclusion 56

References **57**

List of Figures

2.1 GAN Framework.[1] 19

2.2 Diffusion process.[2] 21

2.3 Long Short Term Memory 22

3.1 Deep Structured Joint Embedding convolutional-recurrent net[3]..... 25

4.1 The architecture DF-GAN for text-to-image synthesis .[4] 34

4.2 Comparison of two-way output and one-way output. [4]..... 37

4.3 Illustration of the Semantically Coherent T2I-GAN framework..... 40

5.1 Pretrained classifier. 47

5.2 Generated images for selected examples from CUB dataset. These images are generated from the model trained on CUB dataset only for 1000 epochs but the generated images are even then are of comparable quality with the state of the art models. 53

5.3 Ablation study results demonstrating the impact of Contrastive loss in the training of generative model for the CUB dataset. These results are obtained after training the model for 300 epochs. 54

List of Tables

5.1 The results of evaluation metrices with state of the art methods on the test set for CUB and COCO datasets 52

5.2 Ablation study showing results on choosing different values of the hyper-parameters τ and λ_c 55

List of Abbreviations

Abbreviations

DL	Deep Learning
GAN	Generative Adversarial Networks
CL	Contrastive Learning
MPL	Multi Layer Perceptron
ANN	Artificial Neural Networks
VAE	Variational Auto Encoder
IS	Inception Score
FID	Freche Inception Distance
CV	Computer Vision
NLP	Natural Language Processing

CHAPTER 1

Introduction

1.1 Overview

Textual data has always been an efficient and effective way of communication and information sharing. It is often combined with visual content to make it more comprehensive, accurate, and intelligible means of conveying ideas and knowledge. Images and graphics are equitable part of communicating ideas and information in all fields of life whether it's the academic field, design industry, marketing industry or fashion industry. The application of graphical content can vary from augmenting the textual content for better comprehension to capturing attention with captivating illustrations, but importance of images is a fact. For this, there is dire need of devising novel models for the generation of plausible images satisfying various needs[5].

Remarkably plausible images satisfying various needs can already be generated using modern day computer graphics. But the caveat involved in the task is requirement of substantial human designers and developers' effort for translating high-level concepts to end product of pixel-level details. [2] Past few decades has witnessed an increase in the availability of computational resources. This has paved the way for data driven methods like deep learning (DL) to automate creative and complex human tasks such as hyper-natural image synthesis and natural language processing. Artificial Neural Networks (ANNs) or simply the Neural Networks (NNs) are the heart of DL. Taking intuition from human brain's

activity of passing signals through the biological neurons, ANNs recognize and learn the data pattern from extensive quantities of training data by propagating it through NN layers and thus avoid the need of explicit functional programming. In DL data distributions are learned using algorithms rather than pre-defining them. [3]

Generative Models (GM) are such powerful algorithms in the field of DL and Machine learning (ML) which learn the underlying hidden distribution from the training data and generate new data by sampling from the learned distribution. Some popular generative models include Neural Autoregressive Distribution Estimator, Variational autoencoders (VAE) and Generative Adversarial Networks (GAN). Computational advances and architectural innovations in the field of deep learning (DL) has enabled the deep generative models to show significant improvements in producing rich representations of the real-world data which are hard to distinguish from the real samples.

GANs are popular GMs which have data distribution matching capability which makes them excellent choice for various data synthesis and manipulation tasks. Generator and discriminator are two of its components that are trained in an adversarial fashion to learn the underlying data distribution, where the objective of the generator is to generate fake data that resembles real data and the role of discriminator is to distinguish the fake data from real data. Owing to their good conditional generative capability these are also extensively used for text to image (T2I) synthesis tasks.

Translation of text into image pixels involves two sub problems. First sub problem involves the learning of such a text representation which captures the visual details and second sub problem is concerned with generating plausible images from the captured visual description. Traditionally, detailed visual information about the object to be generated was captured in the attribute representations. But these attribute representations were cumbersome to obtain as domain specific knowledge was required. Automatic text representations learned directly from words and characters, using deep convolutional and recurrent networks has paved the way for translation of words and characters to image pixels[6].

Gap between textual modality and visual modality makes the task of image generation from visual details (captured from natural language descriptions) even more challenging.

The beginning of T2I synthesis was marked by GAN-INT-CLS, in which class conditional GAN was used to generate images from single sentences using sentence interpolations. Then, stacked architectures were introduced to increase the quality and resolution of the generated images and resolve GAN stability issues. It was used as the baseline architecture in following text to image generation tasks. T2I models based on stacked architecture obtained promising results. Another improvement in the fine-grained details was attained by attending to the relevant words in the input text. More recent methods, taking it as baseline backbone architecture, obtained further improvements by incorporating the formulation of semantic layout (object bounding boxes, segmentation masks or a combination) based on the input text. This generated layout was then used for leading towards more semantically consistent image generation for the input text. But, these multi step T2I generation processes require more fine-grained object labels for training such models.Another hitch involved in these multistage architectures is that the quality in the later stages of hierarchy highly depends on the results of initial stage. Furthermore, cross modal attention computation cost increases proportionally with the scale of generated imaged which makes these models hard to extend towards high resolution synthesis.

In T2I tasks, natural language adds another ambiguity by offering a general and flexible interface for describing objects in any space of visual categories. Contrastive Learning (CL) is a technique employed to deal with the problem of natural language in describing same visual content in various ways. It is employed to guide the generative model about the distinctiveness in captions of different images and analogy of the various captions for the same image[7]. CL has proved remarkable improvements in various computer vision and natural language processing tasks [8–11]. These improvements are attributed to the capability of contrastive learning to deal with GAN training issues including mode collapse and discriminator forgetting the learned data representation. In text to image generation tasks CL has also been used by [12, 13] for improving the semantic consistency of generated

images and textual descriptions.

1.2 Motivation and Problem Statement

GANs suffer from the problem of training instability due to the non stationary nature of the training environment.In T2I models, existence of multiple mappings between the image and text caption domains adds more to this training instability by putting the adversarial loss under another constraint.[13] In the field of self supervised learning, contrastive Learning among various views of data has lead to improved training stability[8, 14–16].Several T2I models[13, 17, 18] have been proposed in the literature which employed CL with the intent to stabilize the GAN training and improve semantic consistency of generated image and textual caption. But most of these models have used stacked architecture as baseline and attention computations for ensuring the semantic consistency of image and text. Which becomes computationally more expensive as the resolution of the generated images increases. Baseline staged architecture also introduces generator entanglements which badly effects the quality of generated image in the final stage. [4]suggested a simple backbone comprising of single generator and discriminator pair for generating high-resolution images. In this model, Hinge loss[19] was used for training the generative model and affine transformations for fusing the text and image features.

Given that there has been a lot of recent research and progress in the field of text to image generation models and that existing models contain contrastive learning, we focus to build upon a simple one stage GAN and add CL in the adversarial training to lead towards improved data convergence.

We will combine the semantic consistency of similar captions and uniqueness of captions for different images in the form of contrastive loss during the training phase of adversarial network to achieve highly consistent generated imaged for similar image captions.

1.3 Objectives

The main objectives of thesis are:

- To build GAN based T2I synthesis for generating high fidelity images semantically consistent with input text.
- Use of Hinge Loss and Contrastive learning to improve the convergence of model towards target distribution and inducing training stability in GAN
- Compare the model with state-of-the-art technique.

1.4 Thesis Contribution

The main contribution of this thesis is to propose a GAN based text to image model which uses contrastive learning in the training phase to generate more semantically consistent images for the input text. Our primary contributions are summarized below.

- Introduce a model to incorporate the contrastive learning in the training phase of generative model to learn an embedding space where related signals are pulled together and mismatching signals are pushed away.
- Generation of text encoded vector representations of image caption by employing pretrained BERT.
- Generation of image embedding by employing pretrained InceptionV3 model.
- Model evaluation and performance comparison with existing state-of-the-art models.

1.5 Area of Application

Visual content is an important part of communication in all fields of life whether it's the academic field, design industry, marketing industry or fashion industry. Continuous growth in the field of computer vision and natural language processing

has brought revolution in the multi model synthesis and manipulation tasks. One such active area of research is synthesis of images by taking guidance from text written in natural language.

The use of visual content generation conditioned on textual description is very beneficial in graphics designing industry. Effective graphic designing is backbone of many economy pillars of the country like marketing, print and fashion designing industry; owing to its ability of capturing attention of customers by presenting ideas more aesthetically. Thus, improvements in the field of graphic designing, image generation and manipulation tasks will boost not only computer aided designing but also positively impact other related fields by blending textual information with captivating illustrations.

This synthesis of realistic images conditioned on the textual input is extremely beneficial in a number of domains like art generation, image editing, video games and computer aided design. Although, high quality image synthesis has already been revolutionized remarkably, yet a substantial amount of effort is still required to narrow the gap between high-level concepts and end product of pixel-level details. High level control over the contents of the scene to be generated can be very beneficial in generating realistic images. Utilizing contrastive learning semantic consistency can be improved in the generated images conditioned on human written captions. This can be beneficial in a number of domains in Pakistan's graphics designing industry and improve their contributions in boosting country's economy.

- Art Generation
- Computer aided Design
- Image Editing and Manipulation
- Video Games

1.6 Thesis Organization

The thesis is structured as follows:

- Chapter 1: This chapter contains introduction, objectives and the contributions made in this thesis. It also contains brief overview of the proposed model.
- Chapter 2: In chapter 2, we have briefly covered the methods and concepts used in the succeeding chapters.
- Chapter 3: In this chapter, review of literature and background is given along with brief description of existing technique.
- Chapter 4: In this chapter, our proposed GAN is presented along with the introduction of the embedding technique and contrastive learning being used in the proposed model are also explained.
- Chapter 5: This chapter discusses the experiment detail and analysis of the results by comparing with baseline model along with the brief explanation of the evaluation metrics being used to evaluate the model.
- Chapter 6: This chapter concludes the report and proposes the future work.

CHAPTER 2

Background

In this chapter, we have briefly covered the methods and concepts used in the succeeding chapters. Section 2.1 describes how the biological theories lead towards the development of neural networks. A brief overview covering artificial neural networks and commonly used activation functions in provided in section 2.2. Other topics briefly described in this section include feed-forward neural networks, common loss functions, gradient descent, back-propagation, and popular improvements such as batch normalisation and residual connections.

Section 2.3 summaries Convolutional Neural Networks (CNNs). It covers convolutional layers, pooling layers, and gated linear units. It also introduces recurrent neural networks, their accompanying vanishing gradient problem.Section 2.4 introduces generative models. It covers generative adversarial networks and discusses some of GAN training instabilities including convergence, vanishing gradients, and mode collapse.This section also includes brief overview of auto-encoders and denoising auto-encoders. Lastly, section 2.5 covers salient's of the text encoders.

2.1 Deep Learning

Initially, artificial intelligence (AI) systems were developed using formal languages. These AI systems were based on hard-enciphered knowledge representing real world scenarios coded in formal languages. The difficulties faced in these hard-coded systems suggested that AI systems ought to have capability to attain their

own knowledge, by foraging patterns from real world data. Making systems artificially intelligent in this way is termed as machine learning. The higher the quality of learnt data representation, the better the performance of these machine learning algorithms will be. Designing of a successful artificial intelligence task is dependant on the design of appropriate feature set for that particular task. However, first difficulty faced in this endeavour is to arrive at a decision about the extraction of feature set which will be right for the task. In order to deal with this problem, researchers arrived at a finding that such algorithms should be designed which can not only learn the mapping from representation to accomplish that task but also the hidden data representation itself. This solution for the problem was names as representation learning. The quintessential illustration of a representation learning algorithm is the auto encoder. Two networks are combined together form an auto encoder; first component network called an encoder takes the input and translates it to different representation and the second member of auto encoder generates data resembling with the input for encoder. The second network of auto encoder is named as decoder.

Deep learning algorithms used simpler representations to express the complex representations or in other words it can be stated that in deep learning, simpler concepts are the building blocks used by computers to build complex concepts. For instance, a deep learning system uses simple concept of corners and contours to construct the concept of an image. These building blocks are also defined in terms of edges. The quintessential example of a deep learning model is the multilayer perceptron (MLP).[20]

2.1.1 Biological Intuition

Initial learning algorithms were developed based on the biological activity of the human brain. These computational models imitated the way learning happens or could happen in human brain. For the same reason these were named as artificial neural networks (ANNs). The basic idea of making these computational units intelligent through their interactions waas also inspired from the human brain activity.

Another inspiration from neuroscience was the hope of arriving at a single deep learning algorithm for various tasks. If we look at the initial research works in the field of machine learning, the most evident fact found would be the fragmented research; different communities were putting their effort in different directions like natural language processing, vision, motion planning, and speech recognition. But, the advancements in the field of deep learning changed the trend and research started for devising common practices for studying these diverse application areas. Neuroscience is not only the single foundational base of modern deep learning but other fields (linear algebra, probability, information theory, and numerical optimization) also have their contributions in it.[20]

Linear Models These models are the ancestors of modern deep learning. Neuroscience and applied mathematical concepts lead to the development of these models. These models were aimed to find the mapping for output y from a set of n input values $a_1, \ldots, a_n$. These models would learn a set of weights $w_1, \ldots, w_n$ and compute their output $f(a, w) = a_1a_1+,\ldots\ldots+x_nw_n$. Limitations associated with these simpler Linear models invited backlash from research communities. A famous inability of linear models was to learn the XOR function. In XOR function input values$f$ ([0, 1], w) and f ([1, 0], w) give 1 as output where as 0 is obtained as output for input values f ([1, 1], w) and f ([0, 0], w). In short, such flaws brought bad name to biologically inspired learning in general[21].

2.2 Artificial Neural Networks

A simple ANN is not much different from a linear classifier:

$$f = Wx \qquad x\epsilon R^D \mathbf{W}\epsilon R^{CxD} \tag{2.2.1}$$

Where x is input vector, W is learnable weight matrix of dimension D and C the number of categories. A 2 layer ANN is:

$$f = W_2max(0, W_1x) \qquad \mathbf{W_2}\epsilon R^{CxH}\, \mathbf{W_1}\epsilon R^{HxD} x\epsilon R^D \tag{2.2.2}$$

Where x is input vector, $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weight matrices H is number of hidden layer, c is number of output channels. It can be generalized to any number of layers. Max represents activation function (ReLU), covered in detail in following section. In practice a learnable bias term is added for each layer as well. In a fully connected neural network(multi layer perceptron) all elements of x effect all elements of H and all elements of H effect all elements of s (output layer).[20]

2.2.1 Activation Functions

These are the functions having input from the set of real numbers. The output from these functions is also a number but in a certain range. The output is achieved using a non-linear differentiable function. These are required to be differentiable because these are used in back propagation for training the neural networks and updating its parameters; these functions differentiate and provide gradients to the previous layer. Non-linearity is a requirement for computing complex features within the neural networks. If a non-linear activation function is not used, a neural network with multiple hidden layers and neurons could actually be collapsed into a simple linear regression [20].

Some common activation functions are:

- **ReLU (Recified linear Unit)**. It computes the maximum between input u and zero, or it squashes out all the negative values.

$$f^{[l]}(u^{[l]}) = max(0, u^{[l]}) \tag{2.2.3}$$

  Where u is the input from the layer l, f is activation (ReLU), that takes the maximum between the value zero and u. One problem with this activation is dying ReLU. For $u = 0$ it differentiates to zero which leads towards stopping the learning of ANN [20].

- **Leaky ReLU** is a variant of ReLU introduced to solve dying ReLU issue.

$$f^{[l]}(u^{[l]}) = max(az^{[l]}, u^{[l]}) \tag{2.2.4}$$

It maintains the same form as ReLU for the case when input value u is positive, it keeps the same positive value , but it adds a little value or a slope to the line when u is less than zero or when input u is negative. So, it has non-zero derivative when u is negative. This solves the dying ReLU problem by enlarge [20].

- **Sigmoid** The range for outputs values in this activation function is 0-1.

$$f^{[l]}(u^{[l]}) = \frac{1}{1 + e_{-u^{[l]}}} \tag{2.2.5}$$

It outputs values between 0.5 and 1 for $u^{[l]} >= 0$ and 0 and 0.5 for $u^{[l]} < 0$. This activation function is often used in binary classification models. It isn't used very often in hidden layers because the derivative of the function approaches zero at the boundary ranges of this function. It causes vanishing gradient problem, or saturated output [20].

- **tanh** It is similar to the sigmoid activation function known as hyperbolic tangent or tanh which outputs values in the range of $-1 - 1$.

$$f^{[l]}(u^{[l]}) = tanh(u^{[l]}) \tag{2.2.6}$$

One key difference from the sigmoid is that tanh keeps the sign of the input u. so, negatives input remains negative and vice versa. It can be useful in some applications. The same saturation and vanishing gradient issues do occur in this activation function also [20].

2.2.2 Loss Function

A loss function compares the output of the network to the target output. It gives information about how good a model is improving in learning the task. In case of classifier, it tells how good a classifier is; low loss means a good classifier. If we have a dataset of examples $(x_i, y_i)i^N_{=1}$ where x_i is the image and yi is the label. Loss for single example is $L_i = (f(x_i, W), y_i)$ and that of whole dataset is the average of per example losses [22]:

$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i) \tag{2.2.7}$$

2.2.3 Optimization

The process of finding weight matrix W which minimizes the loss function and fits the model to training data is called optimization.

$$w^* = argmin_w L(w) \tag{2.2.8}$$

For a function $f(x)$ which takes some scalar value as input and gives some scalar value as output, the derivative of the function gives the slope of function. In case when the input is a vector and output is scalar value, gradient at a point is a vector that gives the direction of greatest increase and the magnitude of the gradient gives information about the slope in the direction of greatest increase. In multidimension space, partial derivatives are computed along each dimension which are vectors along each dimension. The dot product of direction with the gradient computes the slope. Negative gradient is the direction of steepest descent. Iteratively stepping in the direction of the negative gradient is called gradient descent. When is performed for the whole batch it is known as the batch gradient descent:

$$\begin{aligned} L(w) &= \frac{1}{N}\sum_{i=1}^{N} L_i(x_i, y_i, W) + \lambda R(W) \\ \nabla_w L(w) &= \frac{1}{N}\sum_{i=1}^{N} \nabla_w L_i(x_i, y_i, W) + \lambda \nabla_w R(W) \end{aligned} \tag{2.2.9}$$

But taking full sum becomes expensive when the N is large. So a way around is to approximate the sum using the mini batch examples. It is known as the stochastic gradient descent (SGD). Some problems with the SGD are:

- Gradient descent faces slow progress due to varying rates of loss change in different directions. Slow progress rate in gradient descent for shallow dimensions and vice versa for steep directions.
- When the loss function has some saddle point or local minima. Here zero gradient is found but the gradient descent becomes stuck.
- Gradients computed for small batches can be noisy. These noisy gradients causes algorithms to linger around the objective as these are not the exact gradients; these are just the stochastic or approximate gradients.

$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(x_i, y_i, W)$$
$$\nabla_w L(W) = \frac{1}{N}\sum_{i=1}^{N} \nabla_w L_i(x_i, y_i, W) \tag{2.2.10}$$

To resolve these issues, SGD is not used in its vanilla version. Rather other versions of SGD are used. One such version is SGD combined with momentum. In simple SGD for every iteration steps are taken in the direction of gradient calculated for mini batches where as in SGD+Momemntum, at every point the gradients are integrated to compute a velocity vector. For every point in time this velocity vector is some weighted combination of current gradient and the historical velocity vector. Here step is taken in the direction of the velocity vector rather than the computed gradient.[23]

Adagrad is another category of optimization function which is based on adaptive learning rate. Here rather then keeping track of historical average of the gradients, we keep track of historical average of square of gradients. This function calculates the summation of element-wise gradients. These gradients are formalized by summing up the squares in every direction. In this way varying progress is obtained for steep directions and flat directions. An acceleration is observed for steep directions and damped effect for flatter directions. Adagrad is combined with RMS prop is a leaky version of adagrad. It adds friction term to deal with slowness of adagrad optimization process. Adam is adagrad+RMS prop combined with momentum.[20]

2.2.4 backpropagation

Backpropagation algorithms are used for efficiently computing gradients of complex functions. An effectual way of computing the gradients of a loss function with respect to the individual weights is backpropagation.Mathematically it can be expressed as:

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial w_{ij}} \tag{2.2.11}$$

Beginning from the last layer, it computes the gradients layer by layer. Efficiency is fostered in this algorithm by computation of partial derivative only once and

then its reuse in subsequent layers.[20]

2.2.5 Batch Normalization

Data distributions are normalized to solve the problem of covariate shift. In the data distribution with many inputs, if input is not normalized or it is skewed more towards higher values or lower value; it will effect the cost function in the training process. When such a model is tested on data skewed in opposite direction then cost function changes and the results of the model become unexpected. To deal with such issues, data distribution are normalized to have mean equal to 0 and a standard deviation equal to 1. It makes the cost function smoother and more balanced across all input dimensions. And as a result training would actually be much easier and potentially much faster.[24]

Normalization reduces the effect of covariate shift quite significantly. However, covariate shift is not a problem if the distribution of the data set is similar to the task required to be modeled.[24]

Neural networks are also susceptible to something called internal covariate shift also. Which just means covariate shift in the internal hidden layers of neural network. Batch normalization seeks to remedy the situation. It normalizes all the internal nodes based on statistics calculated for each input batch. And this is in order to reduce the internal covariate shift. And this has the added benefit of smoothing the cost function out and making the neural network easier to train and speeding up the whole training process.[20]

2.2.6 Residual connection

Deep Neural networks are incredibly powerful networks but are hard to train due to the depth of the networks. Enormous datasets are required for the training due to the presence of more and more parameters. So, training epochs takes more time to train and gradient loss decrease at slower rate at the initial epochs as compared to the shallow networks. Reason for the slow decrease is as the input arrives at the later layer of the network it becomes almost scrambled noise due multiplication with random weights at the earlier layers and it becomes less meaningful at the

later layers. Same is the case with gradient at the initial layers which becomes scrambled noise and less meaningful till the time it reaches at initial layers in back propagation after the update at the output layer. Update at the initial layers becomes less meaningful as gradients are not much meaningful.[25]

Skip connection is a way which is used to ensure input at later layers and gradients at initial layers becomes more meaningful. In skip connections, network layers are grouped in blocks and input to the blocks goes both forward through the block and around the block. At the end of each block output from that block is combined with the input to the block either through concatenation or element wise summation to keep the input more meaningful in the later layers.[25]

Residual network is built in this way out of these residual blocks which is hoped to accelerate the training process as here each block augments the data, makes the path for loss gradient to be shorter and its modularity.[20]

2.3 Convolutional Neural Networks

Computational primitives that respect the spatial structure of two dimensional image data are called as convolutional neural networks (CNN). The building blocks for fully connected neural networks are fully connected layers and activation functions whereas CNN are built from three components convolutional layers, pooling layer and normalization.

Convolution layers takes three dimensional tensors as input, and weight matrix(filter or kernel), which is also a three dimensional tensor. Depth of the input tensor and filter is required to be same. Filter slides over the image spatially and compute the dot products to construct another three dimensional tensor. Resultant matrix is known as the activation map. Number of filters is the hyperparameter, for n number of filters n activation maps are obtained. When a convolutional layer is stacked on top of another convolutional layer, it does not make any difference because it has same representational power as the single convolutional layer will have. So, activation functions are used in between the convolutional layers. [20]

For an input image i of dimensions C_{in} x H x W hyper-parameters will be kernel K_H x K_W , No of filters C_{out}, padding p and stride s. With the C_{out} filters of dimensions C_{in} x H x W the bias vector will be C_{out}. Output dimension will be C_{out} x H' x W' where H' will be $\frac{(H-K+2P)}{S+1}$ and W' will be $\frac{(W-K+2P)}{S+1}$.

In these networks, the task of pooling layer is to down sample the output of preceding convolutional layer. For an input with dimensions C x W x H and filter K, stride S and pooling function could be either max or avg, output will be C x W' x H'. Here W' is $\frac{(W-K)}{S+1}$ and H' is $\frac{(H-K)}{S+1}$.

Internal layers in deep neural networks face the issue of internal covariate shift. This issue pertains to the fact that input for a layer is the output from the previous layer which gets effected by the optimization applied between them. So, the input distribution received by the layer will not be from fixed input distribution. Batch normalization is applied to make the output of every layer to have zero mean and unit variance so that network can be trained towards the target distribution.[26]

2.4 Generative Models

A fundamental requirement for understanding generation of image guided by input text is to comprehend the concept of generative models. In contrast to the discriminative networks, which tries to predict the class of objects based on their features into classes, generative networks try to learn the generation of realistic representations of some class. The target of generative modelling is to learn the hidden data distribution from the training data. This learned data distribution is then sampled for synthesizing new data.[1]

These are partitioned as:

- Undirected generative models
- Directed generative models

This segregation is based on the fact whether the neural layers are interacting directly or indirectly. In recent years, incredible research has been done using directed generative models. Many popular models like Neural Autoregressive Dis-

tribution Estimator and Variational Encoders have been developed with basis on directed generative models.In generative models, the latent variables z are transformed to the observed samples x using feed forward neural networks. Simple distributions like Gaussian Distributions are used for sampling latent variables. Adversarially trained networks known as GANs are also a type of directed generative models which are actively been used for generating realistic images[1].

2.4.1 Generative Adversarial Networks (GAN)

GANs are neural network architecture for generative modeling. Since their inception in 2014, they have outperformed other generative networks in learning deep representations. The datasets used is this training are also not extensively annotated. After training, the generative model can then be used in a numerous applications including image generation, style transfer, image manipulations , enhancing image resolution and classification tasks. Properly trained GANs exhibit the generation of semantically meaningful data from standard distributions [27].

High fidelity images belonging to a variety of domains can be synthesized by utilizing GANs because proper training can lead them to generate semantically meaningful data from standard distributions. Framework of GAN is shown in figure 2.1 . It consists of two networks generator G and discriminator D working in rivalry. Generator is a typical feed forward network that maps the latent variable z to the observed samples x. Discriminator acts as inspector or critic. Its job is to classify the generated samples as fake or real [27].

GAN Architecture. In early GANs, fully connected architecture was used for both the generator and discriminator. These GANs were used for image generation but on simple datasets like Minist (Handwritten digits) and CIFAR-10 (natural images)[28]. A natural extension to these fully connected GANs was to convolutional (CNN) GANs because of their suitability for image data. Major hurdles faced in the GANs were the training instability of generator and discriminator networks due to non-convergence, mode collapse and diminished gradients. These difficulties were handled by the use of Laplacian pyramids for adversarial networks. In this scheme a real image is converted into a multi-scale pyramid image. CNN is

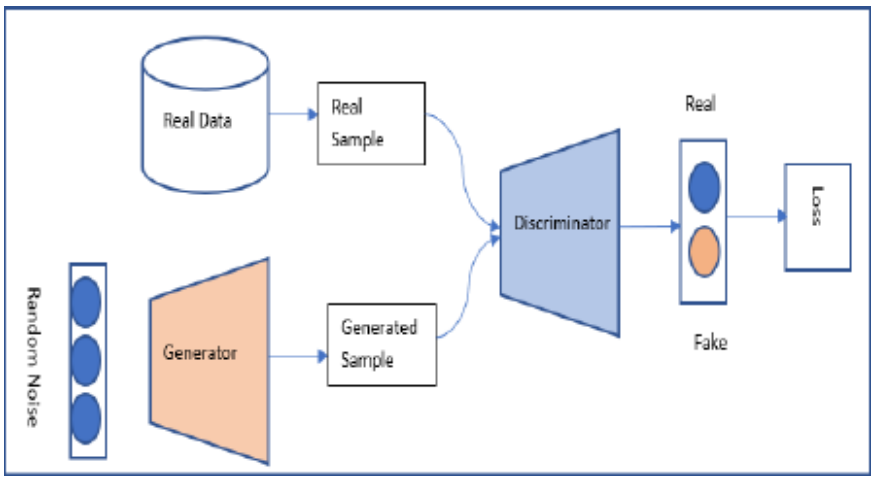

Figure 2.1: GAN Framework.[1]

trained for the purpose to generate multiscale and multi-level feature maps. The combination of all these maps lead to the generation of final feature map.[29]

Deep convolutional GAN(DCGAN) was proposed by [30]. In this architecture of GAN deep convolutional generator was trained in rivalry to a deep convolution discriminator network. Convolutions with strides or partial strides were used to learn both down sampling and up sampling spatial operations. A key requirement of mapping the image space to low dimensional latent space is to handle the sampling rate and locations. The use of convolutions with strides made this key requirement a reality.

Conditional GANs were proposed in 2014. These GANs have the ability to provide improved representation for multimodal data generation problems. This approach provided the prospect for the application of same generic method for a variety of problems and diminished the need of complex loss formulations. [31]. Another GAN architecture based on auto-encoder was proved extremely productive in various vision tasks. Two components of these networks are encoder and decoder. Here deterministic mapping is arrived at via utilizing the encoders and the decoders. [32].

2.4.2 Autoencoders

In these neural networks, the desired output is given as input. Network's first component called as the encoder represents it in the latent space and the task of other component of the network, known as the decoder, is to reproduce the target data from the latent representation in the lower space.[33]

Its leaning procedure involves representing the input into lower dimensional space. Real data is fed into the network as input and a latent representation is obtained for it which is then used by the decoder part of the network to reconstruct the target data.[33]

Variational Auto-encoder are trained to make them learn the latent probability distribution. This learned probability distribution models the input data rather than a function to generate output for the given input. Once the probability distribution is learnt, samples from this distribution are given to the decoder to reconstruct the target output. This auto encoders tries to maximize the likelihood of generating the real data more close to the input data.[33]

2.4.3 Diffusion Models

These generative models work by adding Gaussian noise to the training data, and then this noising process is reversed to recover the real data. After training, reverse noising process is used to generate data data from random noise. More formally a diffusion probabilistic model is a parameterized Markov chain. This chain gradually adds noise to the data in order to obtain the approximate posterior defined as$q(x_1 : T |a0)$. In this equation a_1, ..., a_T are the latent variables with the same dimensionality as a_0. Variational inference is used in the training of this model. After sufficient training these models generate samples matching the target data. In the training process data gets converted to pure noise. The objective of the diffusion model is to learn the reverse process $P_\vartheta(p_{t-1}|p_t)$, and it is achieved by traversing in the reverse direction of the Markov chain[2].

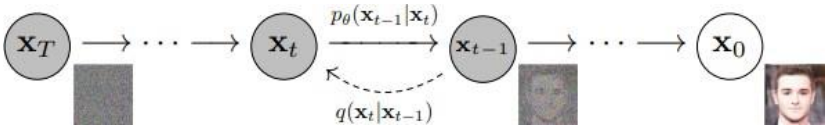

Figure 2.2: Diffusion process.[2]

2.5 Text Encoders

Growth in the field of deep learning has facilitated the use of neural networks for solving problem pertaining to the natural language. Attention mechanisms are combined with CNN, Recurrent networks or graph-based neural networks (GNNs) for various tasks in the domain of natural language. One advantage offered by these neural networks is the alleviation of feature engineering, a mechanism in which syntactic and semantic attributes of the natural language are captured in low-dimensional and dense vectors also known as the distributed representations.[34]. A good representation for a language task is the one which has captured language rules and common context from the training data. Language rules include lexical structures along with meanings, grammar or syntax with semantics and pragmatics of the language.[35]

Text to image generation models use textual embedding techniques for the representations of textual input into image vector space to facilitate further translation into images. With the inception of T2I task, different researchers used various text encoders to represent text into image vector space. A foundational work on this research area was performed by [6]. Text encodings were obtained by employing a pretrained character level convolutional recurrent neural network (char-CNN-RNN). This pretrained model learns the correlation function based on the class labels. Correlation of image and text are obtained in this way. Text representations like Word2Vec and Bag-of-Words did not prove to be effective for encoding task.

2.5.1 Long Short Term Memory (LSTM)

Artificial Neural Networks(ANN) are big composite functions which gets converted into multiplication in the derivative (chain rule). The more deep a function is the more number of multiplications are involved. Recurent neural networks (RNN) suffer vanishing gradient problem which gets worse with the increase in the depth of RNN. Simple RNN has problem learning long term dependencies.[36]

Recurrent network having both “long-term memory" and "short-term memory” is termed as LSTM. With every epoch of training the network model, connection weights and biases are updated. The architecture of LSTM is designed with the aim to make it having a short memory that can last longer timestamps. This is the reason it is called as long short-term memory. At a point in time, the hidden state of the network is obtained by getting the weighted sum of previous hidden states and current value. Components included in the LSTM unit are a cell, and three gates; input, output and forget. The duty of the cell is to keep the values over arbitrary time intervals. Three gates named as input, output and forget have the regulating duty on the information flowing into cell. [36]

Figure 2.3: Long Short Term Memory

Where x(t), h(t-1) and c(t-1) are inputs to the Network to get h(t) and optional c(t). Simple LSTM is comprised of:

- Forget gate: f(t) = Neuron (binary classifier)
- Input gate: i(t) = Neuron (binary classifier)
- Output gate: o(t) = Neuron (binary classifier)
- Cell: f(t) * c(t-1) + i(t) * Simple RNN

- h(t) = o(t) * tanh(c(t))

Bidirectional LSTM (BiLSTM) is a also a recurrent network. It is enhanced version of the LSTM and is also used for the processing of textual contents written in natural language. In this network input flows in two directions. This bidirectional flow of input enables to utilize the information from both sides. This capability makes it an excellent choice for modeling the sequential dependencies between words and phrases in both directions of the sequence. In short it can be stated that BiLSTM has one extra LSTM layer to reverse the flow of information. Input sequence backward flow is attributed to this additional layer of LSTM. For combining the output of the two layers average , sum , multiplication or concatenation can be used. [36]

2.5.2 Bidirectional Encoder Representations from Transformers(BERT)

It is a popular transformer-based language modeling network that uses attention in bidirectional training to capture the relationship of words and sub words in a textual content keeping the context in view. Training in both directions enables it to learn language context in better depth. For the training of this model a novel technique named Masked LM (MLM) was used. Encoder of this model scans the word sequence at once for capturing the context.

A sequence of tokens formalizes its input. This input is embedded in vectors and it is passed through the networks for processing. Output vector is also a sequence of vectors. vector for an input token can be found at same index. [37]

CHAPTER 3

Literature Review

An overview covering most common approaches of generative image modelling is included in this chapter. Section 3.1 introduces the generative models trained in adversarial fashion for text to image synthesis task. Section 3.2 illustrates the most common autoencoders, and autoregressive based models for generating images conditioned on text and section 3.3 provides brief overview of diffusion models.

3.1 Generative Adversarial Network (GAN)

3.1.1 Preliminary Methods

Generative adversarial image synthesis conditioned on text was proposed by [38] in 2016. This work laid the foundation of this challenging task of T2I synthesis to start with the translation of single human written sentence into image pixels. They aimed to learn direct mapping of image pixels from words and characters existing in the sentence. Building on the enormous progress laid in the fields of natural language representation and image synthesis, this T2I task was achieved by the authors. Deep symmetric structured joint embedding was utilized in this work which uses deep convolutional and recurrent text encoders. These encoders learnt the correspondence function for the textual content and images. This pre-trained text encoder approach was proposed by [3] in 2016.

This model was proposed to deal with the shortcoming of text models base on con-

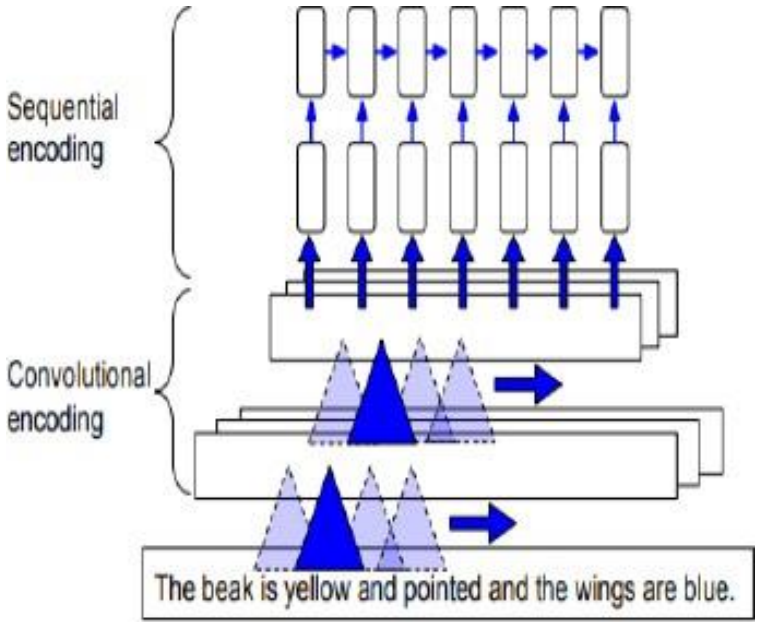

Figure 3.1: Deep Structured Joint Embedding convolutional-recurrent net[3].

volution. Initial models lacked the ability to capture the dependency This model was proposed to overcome the shortcoming found in convolution-only text models. These models were unable to capture the dependency structure for long input sequences. In this model researchers combined a recurrent network with temporal CNN hidden layer. This stack of recurrent layer with convolutional layer enabled the model to take advantage of both recurrent models and CNNs. In this way, this approach has exploited the CNN's capability of efficient learning to learn the low-level temporal features. Figure 4 shows the phenomenon of Deep Structured Joint Embedding. It was used in deep convolutional generative adversarial network (DC-GAN) conditioned on text features. [38] DC-GAN was proposed by [39] for stabilizing the training of GANs by proposing some architectural changes in the simple convolutional GAN. They applied a set of constraints for this purpose. During training of DC-GAN, added a third type of input to discriminator (matching aware discriminator GAN-CLS) consisting of real images with mismatching text in addition to the real (real image, matching text) and fake (fake image, right text) input pairs. Then, GAN INT learnt utilizing the manifold interpolation and GAN-INTCLS combines the both to produce plausible images.

This model was trained on two data sets CUB and Oxford 102. For CUB dataset GAN and GAN CLS get some color information right but produced images are not real whereas GAN-INT and GAN INT-CLS generates images which are better representations of the captions. For Oxford 102 Flower dataset all four GANs (GAN, GAN-CLS, GAN-INT, GAN-INT-CLS) performed good in generating plausible flower images as per the given text.

[6] demonstrated that usefulness of image generating system can be enhanced by executing more control over the contents of the scene to be generated. They proposed Generative Adversarial What-Where Network (GAWWN) which has the capability to synthesize images guided by the instructions about what to draw and where to draw. They also used structured joint embedding of visual descriptions and images for learning the correspondence function between images and text features. A minor modification in this approach was done by using CNN-GRU [40] instead of char-CNN-RNN [3]. GAWWN synthesized images by taking input of instructions describing the content to draw along with its locations. For this purpose bounding box or a set of keys describing parts.

[22] proposed a GAN based encoder-decoder architecture. This architecture focused on the semantics contained in both image and text and ignored the text irrelevant parts of the image. The generator was an encoder-decoder architecture and synthesized images guided text embeddings. The discriminator performed the discriminative task guided textual description. A pretrained text encoder was used for creating semantic representations. A text embedding augmentation method [41] which enabled the model to synthesize diverse images conditioned on text. [42] improved image feature representation by utilizing VGG in place of an image encoder in the generator. VGG is pretrained on imageNet. They performed human evaluation of the proposed method with baseline method [3] and found that their proposed method had ability to generate more plausible images.

Text to image generation methods considered the output image as single unit and tried to figure out the semantics of the input text in single unit of the image. Which led to lesser semantic consistency in generated image and input text. In reality, every natural images is viewed in context having of a foreground and back

ground. T2I generation methods could not generate a true representation of the semantics of the given text when they ignored this fact. Motivated by this reality, [43] proposed multi conditional GAN.This method synthesized a target image by drawing the background and fore ground of the image. For this purpose background of a source image was used and a text-descriptions were used to generate foreground object. A synthesis block was used to consider the background feature without nonlinear function and the foreground feature were the feature map resulting from the preceding layer.

3.1.2 Stacked Generative Adversarial Networks

Some difficulties associated with training of generative networks being trained in adversarial fashion, were their instability in training process, sensitivity to the choice of hyper parameters and model collapse. These challenges even turn to more severity when attempts are made to generate high-resolution images (256 x 256). Motivated by the work of many researchers to handle these challenges and stabilize the GAN training process [44] started their work to generate image of higher resolution by introducing multi stage GANs. If the images from real world are closely observed, it is evident that real world images specially, natural images can be modeled at different scales. Same continuous image signal with different sampling rates leads to the generation of images at different scales. Another interesting fact is the phenomenon that a relationship exists among the distribution of images at multiple discrete scales[45]. These facts and motivation of divide and conquer rule led [46] to the idea of breaking the complex task of image generation into sub problems.

Model proposed by [46] had three components. Conditional augmentation, two generative networks organised in a hierarchy of stages. Text description was encoded using a pretrained encoder. It generated images for given text in two stages: In stage-1, sketches comprising of basic shape and colors conditioned on the given text input were produced. For background layout a noise vector was used. Thus, an image of low resolution was generated. In stage-II, a high-resolution image was produced by correcting the defects of stage-1 image and reading the input text

again. This method generated higher resolution images (e.g., 256 x 256) having more photo-realistic details.

Very natural extension to this two staged network was to have multiple generative networks arranged in a tree like structure. Each branch of the tree was a GAN that used to produce image at scale lower than its succeeding branch GAN; in this way generation at higher resolution was achieved. Generators belonging to all branches are jointly trained. Whereas generators and discriminators are trained in alternating fashion. Authors performed comparison of the image generation capability of stackGAN-V1 and stackGAN-V2 by utilizing CUB, Oxford-102 and COCO for conditional generative tasks and LSUN (bedroom, church) and ImageNet (dog, cat) for unconditional generative tasks. They showed from the empirical results that StackGAN-V2 out performs not only stackGANV1 but also other state of the art T2I models at that time [44].

In an effort to generate high fidelity images [47] proposed fuseGAN built upon the stackGAN. Authors identified three important quality attributes for measuring the quality of the image generation models which include fidelity, diversity and controllable sampling. Instead of using multiple stages like stackGAN, they proposed a pipeline. This pipeline had a built-in stack. The first unconditional generator of the pipeline was tasked to generate unconditional structure prior. The second generator enhances the structure prior from first stage generator by adding style to it on the bases of input condition. Advantages claimed by the model include: 1) it enabled to control the image diversity. 2) semi-supervised data can be used in training and 3) is no reliance on additional intermediate supervision such as segmentation maps. They performed the training and evaluation of this image generation model using CUB dataset and claimed higher inception score and FID for the proposed model.

This challenging task of was handled by [48] by a method employing hierarchical-nested adversarial objectives inside the network hierarchies. In this way they regularized intermediate data representations. This regularization improved generator training for capturing the complex image statistics.

Another model for achieving high resolution images was perceptual pyramid ad-

versarial networks (PPAN) [49]. In this work they utilized a network architecture(Laplacian Pyramid Super-Resolution Network) which was proposed by [50]. This architecture progressively improved the image generation towards high-resolution. In this model, a pyramid framework was used for combining low-resolution and high resolution features. An auxiliary classification loss was also employed as a perceptual loss [51] in this model. Moving on the journey to arrive at high resolution images another model was proposed by utilizing hierarchically-fused architecture having only one discriminator. Global features for the image are extracted at varying scales utilizing different stages of hierarchy. Then these are fused together to generate the image at higher resolution.[52]

3.1.3 Attention GAN

Compelling sequence modeling and transduction model for various tasks had incorporated the attention mechanisms. Attention models had enables them to model dependencies without regard to their distance in the input or output sequence[53]. Attention is a mechanism by owing to which network were able to focus on specific parts of an input. It assigned more weight to important parts than insignificant parts of the input. Attention had proved to be an extremely powerful technique and led towards a major impact on improving language and vision applications[54]. AttenGAN [55] is another latest work that proposed attentional generative adversarial network for this multi modal task of T2I synthesis. It used attention-driven, hierarchy network for creating image conditioned on text description. It used Bidirectional LSTM for text encoding which concatenated hidden states for the forward and backward directions. Then, attention generative network was utilized to encode the text into global sentence vector and word vectors. An image at lower resolution was generated in first stage by utilizing global sentence vector. Then, high resolution images were generated in the later stages by combining regional image vector and corresponding word context vector. Finally, it computed the similarity of generated image and given text by a deep attentional multi modal similarity model.

An important contribution made by [55] was DAMSM network model. This extra

network takes the output of final stage GAN and computes its similarity with global sentence vector and word vector. This model was also used in SDGAN. In this model SCBN (Semantic conditioned batched normalization)was used reinforced the visual-semantic embedding in the feature maps. SD-GAN adopted a Siamese structure which extracted semantic information from the textual content.

Use of semantic layout to improve the consistency of generated image against input was proposed by [56]. They suggested the use of a network named as layout generator for synthesizing the semantic map for the image. Input text description is used as condition in this layout generation. This layout generator is further segregated as box generator and shape generator. Once this semantic layout is constructed , then attention module and similarity loss are used for infusing the fine grained details of the image.

BigGAN[57] architecture eas based on residual blocks. The use of Spectral Normalization and Non-local Blocks in both components of GAN was a novel feature of this model. BigGAN-Deep also introducesd conditioning information in the generator using Conditional Batch Normalization. Unquie feature of critic network the use of projection approach. For discriminator network, sentence vectors were linearly projected whereas in the generator network sentence vectors were concatenated with the noise vector z.

RiFeGAN suggested Rich Feature Generation in Text-to-Image Synthesis. Feature enrichment was performed utilizing Prior Knowledge. In this generative model, enrichment of the captions were from prior knowledge to tackle the problem of limited information. An extra network named attentional text-matching model was used to retrieve compatible captions from prior knowledge automatically. Then, multi captions attentional GANs were used to extract rich features and synthesizing high-quality images. [58]

MirrorGAN [59] was a model of learning Text-To-Image Generation by Redescription. In this model, first network was trained to generate images guided by text. Then a pretrained, image to text model is used to construct the input text from the generated image. The loss function computes the similarity of input text and generated text in addition to the adversarial loss and helps the generative model

in convergence towards the target distribution.

3.2 Auto-regressive Models

Rise of auto-regressive generative models has tremendously improved the natural language performance and few-shot language understanding. Use of auto-regressive models in CV is nothing new. PixelCNN,PixelRNN, Image transformer, ImageGPT, VQ-VAE have demonstrated incredible improvements in the field of image generation[60]. Conditional image generation models also witnessed remarkable improvements due to the existence of incredibly powerful image generative and natural language processing methods. DALL-E [61] based on the VQ-VAE(Vector Quantized Variational Encoder) brought a revolution in the field of text to image generation. Authors took intuition from the idea of increasing model size, compute and data for impressive results and trained a 12-billion parameter auto-regressive transformer on 250 million image- text pairs. They got successful in achieving zero shot text to image generation with high quality.

Authors of CogView [60] identified unstable large-scale text-to-image generative retraining and suggested Precision Bottleneck Relaxation and Sandwich Layernorm and were able to achieve improvement in the conditional image generation.

3.3 Diffusion Models

Another line of research with incredible momentum involves diffusion models[62] for text to image generation. Some of the popular models include GLIDE[63], DALL-E 2[64] and Imagen[65]. These models are diffusion based which use discrete image tokens with diffusion models and generate high fidelity images. These models have shown remarkable improvements in the FID scores for the generated images and zero shot image generation.

CHAPTER 4

Method

In this chapter we present reasoning behind the choice of baseline text-to-image architecture, contrastive learning techniques, several approaches of interpreting and combining them with synthesis models. Section 4.1 discusses our chosen backbone architecture of DF-GAN[66].This section is excursion DF-GAN structure and its novel components. In Section 4.2 we present contrastive learning approach along with a brief introduction of its architecture and training. Section 4.3 illustrates our model and describes the incorporation of constrastive learning technique in the adversarial training.

4.1 Introduction

This section provides acumens about our baseline architecture and text-encoder. For making a decision about the general approach for T2I synthesis we studied the existing work in the domain of T2I processing. We arrived at the general idea that most preexisting works in the context of generative image modelling falls into the categories of Auto-encoder, Auto-regressive Model, diffusion Models, or Generative Adversarial Network (GAN). The objective of an auto-encoders is to reconstruct the output from the latent representation of input. For text-to-image generation tasks, Auto-encoders have to learn the multi modal representation in the latent space. Recent research in this field has shown that more compute power, larger model size and larger data set leads to impressive results utilizing auto-regressive transformers[53]. And for multi modal tasks it goes even higher.

Where as GANs have excellent data matching capability and they produce more plausible results as compared to auto-regressive models of same size. Limitation of compute resources lead us to the choice of GAN coupled with GAN's excellent performance in recent image synthesis and manipulation tasks.

GANs suffer from the training instability issue which involves vanishing gradient problem in the generator training. Many techniques have been suggested to deal with this instability, and has made them more promising candidates for the image synthesis and manipulation tasks. Stacked architecture[46] was also introduced to deal with training instability for generating high scale images. Many SOTA GAN models have used stacked backbone architecture. But this architecture has its own limitations. It becomes computationally more expensive as the size of the image grows and image quality in later stages is determined by the quality of basic image structure generation in the initial stages.

To make generated images more consistent with input textual description some stack based architectures utilized extra networks and incorporated multi modal attention approach[55][59]. Other models proposed improvements through object driven hierarchical approaches to generate images from semantic layout(bounding boxes, segmentation masks) and text[52, 67–70]. But these models require more fine grained object labels for training. [66] suggested a single stage generator and discriminator architecture DF-GAN to deal with these problems. We have selected it as our baseline GAN architecture.

Text-to-image generation involves representation of text into visual space which is then utilized by the conditional generative networks to generate plausible images satisfying input text. It is not feasible to train a such a huge model from scratch. We shall use a pretrained text-encoder Bert[37] for obtaining text embeddings.

Self supervised learning is a field where the focus is to obtain good intermediate representations that can be used for downstream tasks without supervision. Self-supervised objectives are either auxiliary classification losses or contrastive losses. The objective in contrastive losses is to learn such an embedding space in which the contrasting input signals are closer and different data points are far apart. Auxiliary classification losses are concerned with making predictions for the rota-

tion of inputs and masked losses. The objective of such losses is to predict he true value of input masked out.[71] Natural language has the discrepancy of describing same visual content in various ways. As data sets used for text to image generation are annotated by humans, so same visual content can be annotated differently by different persons. To narrow this gap contrastive learning techniques are employed for pushing together the captions of same image and distinguishing the captions for different images in the learned representations.

Contrastive learning technique was employed by [13, 18] in cross modal attention based text-to-image models for improving the semantic consistency of image with captions described in different ways. We took intuition from it and applied the contrastive loss interpretation along with generator's conditional loss to improve generated image's semantic coherence with different captions of same image.

4.1.1 DF-GAN

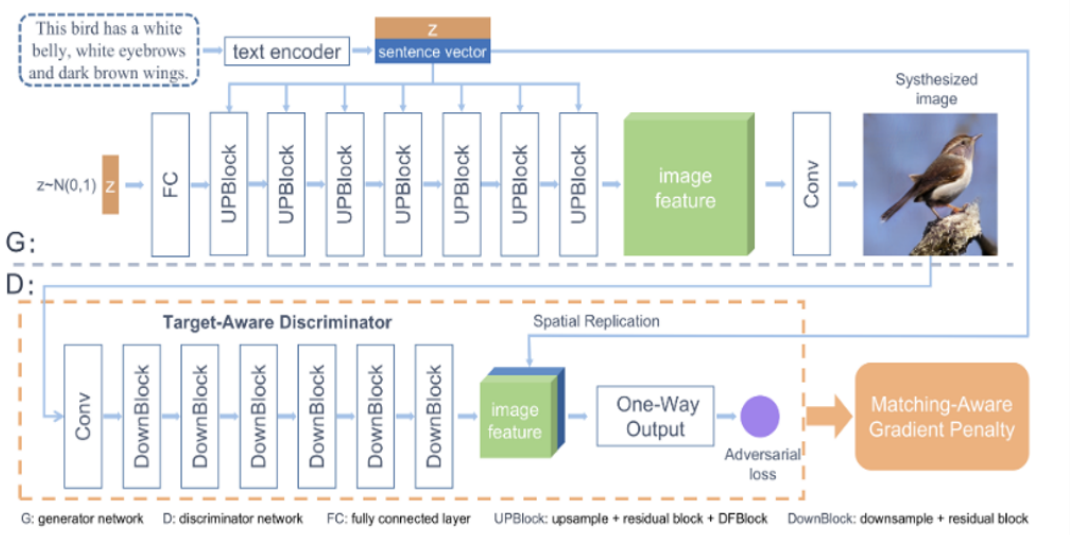

Figure 4.1: The architecture DF-GAN for text-to-image synthesis .[4]

Deep fusion generative network synthesizes images at improved resolution by utilizing a simple architecture comprising of one pair of generator and discriminator. For generating images conditioned on the input text, it fuses the text information and visual feature maps. Seven Deep text-image Fusion Blocks known as DF-Blocks are used for this fusion. Other important components of this generative

model are Matching-Aware Gradient Penalty (MA-GP) and One-Way Output. In stead of stacked architecture, hinge loss[19] along with residual networks is used for stabilising the GAN training issue. In stead of concatenating the textual embedding with the visual features along channel wise, it fuses text and image using affine transformations. Generator of this model has more layers as compared to the generators of most GANs[4].

4.1.2 Deep Fusion Generator

Deep fusion generator has more layers because it has to generate image from noise in single stage. It is formed of seven UpBlocks followed by a convolutional layer. Each UpBlock consists a upsample layer, a residual block and a DF-Block. Noise is reshaped by passing it through a fully connected layer and given as input to the generator along with the text embeddings. Pretrained bi directional Long Short-Term Memory (BiLSTM) [72]text encoder is used for extracting visual detail vectors from the text. Noise and text embedding are passed through the seven Upblocks for fusing the text and image features. Output from the seventh block is passed through a convolutional layer which generates an image from the input image features [66].

UpBlocks have residual networks to train layers more effectively in deeper networks. For stabilizing the GAN training, hinge loss is used along with residual networks. Cost function for this one stage method is described as:

$$\begin{aligned}\mathbf{CostFunc}_D = \mathrm{E}_{img\sim \imath_r}\Big[min(0, -1 + D(img, emb))\Big] - (\tfrac{1}{2})\mathrm{E}_{G(\text{⍰})\sim \mathrm{P}_g}\\ \Big[min(0, -1 - D(G((z), emb)))\Big] - (\tfrac{1}{2})\mathrm{E}_{img\sim \mathrm{P}_{mis}}\\ \Big[min(0, -1 - D(img, emb))\Big]\end{aligned} \tag{4.1.1}$$

Efficient text-image fusion is performed by the two blocks contained in each UpBlocks. Each deep fusion block (DF Block) has multiple Affine Transformation stacked on each other. In it two multi layer perceptron are used; one to learn text guided channel wise scaling parameter γ and other to predict shifting parameters

ϑ from sentence vector e [66].

$$\gamma = MultiLayerP\,erceptron_1(emb),\ \vartheta = MultiLayerP\,erceptron_2(emb) \quad (4.1.2)$$

For a given feature map $X \epsilon E^{BxCxHxW}$, affine transformation process is:

$$\mathbf{AFFINE}(img_i|emb) = \gamma.img_i + \vartheta_i \quad (4.1.3)$$

Affine layers widens the conditional latent representation space for the first network called generator. For adding more expansion to this conditional representation space , non linearity is added between the two MLP layer by ReLU layer. Due to this expansion, mapping of different images to different latent representation space becomes possible. This deep fusion process benefits the image synthesis process in two ways: at one hand generator takes full advantage of the textual information due to the fusion of text-image features and on the other hand enlarged representation space benefits the generator to generate more semantically consistent results.

4.1.3 Semantic-Aware Discriminator

Two important components of this semantic-aware discriminator are matching-aware zero-centered Gradient Penalty (MA-GP) and one way output.

[73] presented a analysis to lead the way towards the use of zero-centered gradient penalties for local convergence. This idea was used by the authors of DF-GAN to devise this cost function. They also employed gradient penalty on text matching real/target data to foster further improvement in conditional generation. The formulation of this model is described as:

$$\begin{aligned}
\mathbf{costFunc}_D = {} & \mathrm{E}_{img\sim \imath_r}\Big[min(0, -1 + D(img, emb))\Big] - (\tfrac{1}{2})\mathrm{E}^{G(\text{⍰})\sim \mathrm{P}_g} \\
& \Big[min(0, -1 - D(G((z), emb)))\Big] - (\tfrac{1}{2})\mathrm{E}\,img\sim \mathrm{P}_{mis} \\
& \Big[min(0, -1 - D(img, emb))\Big] + k\mathrm{E}_{img\sim \mathrm{P}_r} \\
& \Big[||(\square_{img} D(img, emb)|| + ||\square_e D(img, emb)||)^p\Big] \\
\mathbf{CostFunc}_G = {} & -\mathrm{E}_{G(\text{⍰})\sim \mathrm{P}_g}\ D(G(z), emb)
\end{aligned} \quad (4.1.4)$$

Here k and p are two hyper-parameters to balance the effectiveness of gradient penalty. MA-GP loss is the regularization on the discriminator due to which model converges to the target data conditioned on text. MA-GP calculations involve only gradient summation which made it computational economical.

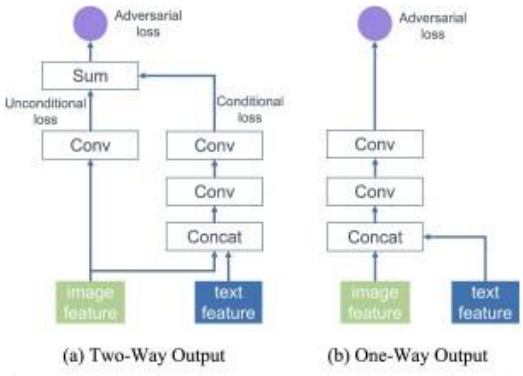

Figure 4.2: Comparison of two-way output and one-way output. [4]

Second improvement suggested by this simple backbone architecture is the one way out put. In contrast to previous text-to-image models, DF-GAN discriminator predicts the whole adversarial loss directly. One-way output devised in DF-GAN strengthened the MA-GP role for target aware convergence.

4.2 Contrastive Learning

Success of machine learning(ML) and deep learning(DL) methods is determined by the quality of datasets used to train these models. Self-supervised learning has proven its mark in the success of generative models because it evades the need of having large dataset and its annotations. Self-supervised learning utilizes the underlying structure of the data to obtain supervisory signals from the data. Contrastive leaning(CL) is a technique which falls under the umbrella of self-supervised learning. Self-supervised learning in CV, NLP domains, involves CL as a dominant component. The objective of CL is to bring close the embedding augmented versions of same sample and push away the embeddings of dissimilar samples.[74]

Contrastive Learning can be thought of as learning by comparing. Samples are compared to learn the underlying latent structure of the data. This comparison

involves the contrast of similar inputs (positive pairs) and also the dissimilar inputs(negative pairs). Contrastive learning aims to learn an embedding space. In this learned representation, similar samples are represented closer together where as dissimilar samples are pushed away. In this way, it arrives at a representation where positive pairs are pulled together and negative pairs are far away [12].

4.2.1 Architectures

In contrastive learning, negative samples are obtained by perform lookups from a dictionary. This dictionary is comprised from the complete training set. Depending on the lookups being performed, this dictionary could be a sub set of the complete training set. Categories of the CL techniques are based on the way negative samples are collected for a positive data point in training. [74]. These are categorized as:

- Use of two encoders which are trained End-to-End. One encoder is trained to generates mapping for positive samples. Second encoder to obtain representations for negative samples.
- Encodings of negative samples are obtained and stored in a memory bank. These encoding are then obtained from this memory bank.
- Use of momentum encoder which serves as dynamic dictionary lookup. Negative sample encodings are looked from this encoder in training.

4.2.2 Training

For fetching mapping of similar data signals together and dissimilar data signals far away in the learned representation, contrastive learning techniques employ similarity metrices. These metrices gauges the similarity of embeddings obtained for two samples. Cosine similarity is the most commonly used metric for this purpose. Different contrastive loss functions are devised based on this similarity metric. This metric measures the cosine of the angle existing between two vectors. Mathematically it is described as follows:

$$cos\_sim(\mathbf{V}_1, \mathbf{V}_2) = \frac{V_1.V_2}{||V_1||.||V_2||} \tag{4.2.1}$$

Noise Contrastive Estimation [75] is a metric used in several models which have utilized contrastive learning technique for improving the convergence of the model towards target data. It performs a comparison of sample embeddings: defined as :

$$L_{NCE} = -\log \frac{\exp(sim(img, k_+)/\tau)}{\exp(sim(img, k_+)/\tau) + \exp(sim(img, k\_)/\tau)} \tag{4.2.2}$$

For an original sample img , k+ is the positive sample, and k_denotes negative sample.τ is a hyper-parameter known as temperature coefficient. For computing the similarity, generally cosine similarity as defined in Equation 4.2.1 is used. NCE forges a nonlinear logistic regression to discriminates between observed data and some artificially synthesized noise.

InfoNCE is a variant of NCE. It is used in cases when there exists a large number of negative samples. Similarity metric l2 normalization and τ enables to assign different weights to different samples and facilitates the model to learn effectively

$$L_{infoNCE} = -\log \frac{\exp(sim(img, k_+)/\tau)}{\exp(sim(img, k_+)/\tau) + \Sigma_{i=0}^{k} \exp(sim(img, k_i)/\tau)} \tag{4.2.3}$$

Negative sample is represented as k_i.

Contrastive learning has been explored in a number of computer vision and natural language processing applications. In adversarial training it has also been used extensively [8–11]. It was employed as a measure to resolve two of training issues concerned with the GANs i.e mode collapse and discriminator forgetting the learned information. Contrastive learning was used along with mutual information maximization for long-term representation learning.

4.3 Model

In, this section, the proposed framework "Semantically Coherent Text to Image Generation" is illustrated. Proposed model is developed based upon the con-

trastive loss presented in Section 4.2.2 which is incorporated into the baseline model presented in Section 4.1.1. Proposed model architecture is illustrated in Figure 4.3.

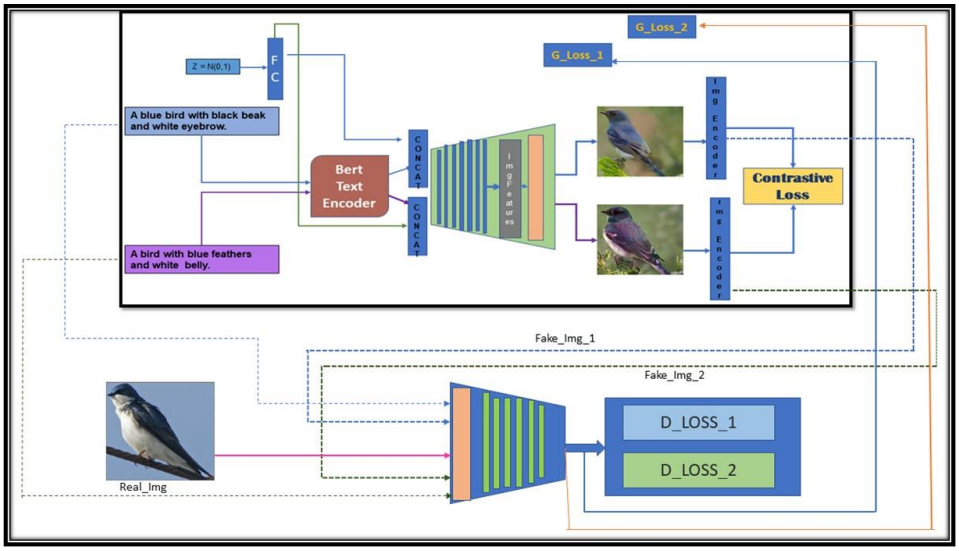

Figure 4.3: Illustration of the Semantically Coherent T2I-GAN framework

Proposed model has two main components, a generator network and a discriminator network. Textual caption for image generation are encoded using pretrained text encoder BERT[37]. Generator gets two inputs, one is the noise vector z sampled from the Gaussian distribution and the other is sentence embedding obtained by encoding the text with pretrained BERT. Which is then passed through the up sampling block of the generator to get the image features. These are then converted to a fake image by utilizing the convolutional layer.

Image generation conditioned on input text aims at producing realistic images and at the same time semantically consistent with the provided textual description. To arrive at this two fold objective we propose to incorporate contrastive learning. The aim here is to utilize the coherence of similar captions describing same image (positive pairs)to bring them closer in learned representation and push away dissimilar captions describing different images (negative pairs)far away in the representation. Since the image generation is conditioned on textual description, so this pre training approach will narrow the gap between the generated image and input text.

To achieve this contrastive learning objective we have followed the noisy contrastive estimation framework from[15]. Contrastive learning tends to correlate two signals in the representation. For a caption *c*, there exits some captions describing the same image (positives) and some describing other images (negatives). CL minimizes the distance among positives and pushes negatives far way in the learned representation. Normalized temperature scaled cross entropy loss is calculated for *i*th query as:

$$CL\text{-}Loss = -\log \frac{\exp(sim(c_i, c_j)/\tau)}{\Sigma_{k=1}^{2N} \mathbf{I}_{k=i} \exp(sim(c_i, c_k)/\tau)} \tag{4.3.1}$$

In the above equation positives are indicated as *i*th and *j*th c. When $k \neq i$ the value of $\mathbf{I}$ is 1. τ is temperature scale. We have sampled 2N captions for N images. At a time a mini batch is taken and CL loss is calculated across the mini batch.

4.3.1 Objective Functions

Discriminator Objective Following our baseline model, we used one-way discriminator with gradient penalty on the real images with two branches of matching captions. Loss function with MA-GP used for training is:

$$\begin{aligned} \mathbf{L}_D = & \mathrm{L}_{D|\neq} + \mathrm{L}_{D|\neq} \\ & + k\mathrm{E}_{img\sim \mathrm{P}_r} \left(||(\nabla_{img} D(img, c_1)|| + ||\nabla_e D(img, c_1)||)^p \right) \\ & + k\mathrm{E}_{img\sim \mathrm{P}_r} \left(||(\nabla_{img} D(img, c_2)|| + ||\nabla_e D(img, c_2)||)^p \right) \end{aligned} \tag{4.3.2}$$

Where $L_{D_{c1}}$ and $L_{D_{c2}}$ hinge wise adversarial loss for GANs [76].

Generator Objective The total loss for the generator is composed of hinge wise adversarial loss and normalized temperature scaled cross entropy loss (NT-Xent) [16].

4.3.2 Contrast Aware Discriminator

We augmented the matching aware zero-centered Gradient Penalty discriminator from our baseline model with contrastive learning to smooth the convergence of generated data points to the target data points. To fulfill desiderata we sampled a mini batch of images. For an image 'x' in the batch we loaded two captions describing the same image. Due to computational constraints of GPU we have to reduce the batch size to half to that of our baseline model because we are calculating the hinge loss [77] of an image x for two captions c_1 and c_2 and using it for the calculation for MA-GP on the target data to speed up the convergence towards the real data.

The hinge version of adversarial loss[77] for both the captions is calculated and semantic coherence of each caption for the image is obtained which is used for obtaining the gradient penalty on the real data. This MA-GP serves as a regularization on the critic and saves it from over fitting.

Following the single output intuition of the baseline method, we concatenated the c_1 features and x image features and obtained the adversarial loss L_{Dc1} by passing it through two convolutional layers. Then we concatenated the c_2 features with the image x and arrived at adversarial loss L_{Dc2}.

$$\begin{aligned} \mathbf{L}_{D1} = & \mathrm{E}_{img\sim \mathrm{P}_r}\left[min(0, -1 + D(img, c_1)) \right] \\ & - (\tfrac{1}{2})\mathrm{E}_{G(z)\sim \mathrm{P}_g}\left[min(0, -1 - D(G((z), c_1))) \right] \\ & - (\tfrac{1}{2})\mathrm{E}_{img\sim \mathrm{P}_{mis}}\left[min(0, -1 - D(img, c_1)) \right] \end{aligned} \tag{4.3.3}$$

$$\begin{aligned} \mathbf{L}_{D1} = & \mathrm{E}_{img\sim \mathrm{P}_r}\left[min(0, -1 + D(img, c_2)) \right] \\ & - (\tfrac{1}{2})\mathrm{E}_{G(z)\sim \mathrm{P}_g}\left[min(0, -1 - D(G((z), c_2))) \right] \\ & - (\tfrac{1}{2})\mathrm{E}_{img\sim \mathrm{P}_{mis}}\left[min(0, -1 - D(img, c_2)) \right] \end{aligned} \tag{4.3.4}$$

The hinge wise adversarial loss L_{Dc1} and L_{Dc2} are used to calculate the MA-GP

as:

$$\begin{aligned} \mathbf{L}_D &= \mathrm{L}_{\mathrm{D}\text{⊮}} + \mathrm{L}_{\mathrm{D}\text{⊭}} \\ &+ k\mathrm{E}_{img\sim \mathrm{P}_r}\left[(\|\square_{img} D(img, c_1)\| + \|\square_e D(img, c_1)\|)^p\right] \\ &+ k\mathrm{E}_{img\sim \mathrm{P}_r}\left[(\|\square_{img} D(img, c_2)\| + \|\square_e D(img, c_2)\|)^p\right] \end{aligned} \tag{4.3.5}$$

Here z is the noise vector sampled from the Guassian distribution N (0, 1); c_1 and c_2 are the caption embeddings; I_g, I_r and I_{mis} represent the generated distribution, real distribution and mis-matching data distribution, respectively.

4.3.3 Generator

Our model has single generator to generate image conditioned on the input text. Noise z sampled from Gaussian distribution and text embedding obtained from pretrained BERT are given as input to the generator. Noise z is reshaped by passing it through a fully connected layer and then it is fused with text embedding by utilizing the two Text-Image Fusion blocks same like the baseline model are used to fuse the text and noise vector.

We sampled mini batch of 15 images and captions for generating images. Adversarial loss for the generator is computed as

$$\begin{aligned} \mathbf{L}_{\mathrm{G}_{\mathbf{c1}}} &= -\mathrm{E}_{G(\text{⍰})\sim \mathrm{P}_g}\left[D(G(z), c1)\right] \\ \mathbf{L}_{\mathrm{G}_{\mathbf{c2}}} &= -\mathrm{E}_{G(\text{⍰})\sim \mathrm{P}_g}\left[D(G(z), c2)\right] \end{aligned} \tag{4.3.6}$$

To exploit the contrast of the positive sample we calculated the generator adversarial loss for the two captions and then arrived at the contrastive loss for the sampled image. For ith and jth contrasting samples the contrastive loss is calculated as

$$CL_{c1,c2} = -\log \frac{\exp(sim(c_i, c_j)/\tau)}{\Sigma_{k=1}^{2N} \mathbf{I}_{k\ i} \exp(sim(c_i, c_k)/\tau)} \tag{4.3.7}$$

Then the adversarial loss for the generator is calculated as

$$L_G = L_{Gc1} + L_{Gc2} + CL_{c1,c2} \tag{4.3.8}$$

CHAPTER 5

Results and Evaluations

Experimental results and discussion on evaluation metrices are included this chapter. This chapter also provides comparison of our model with other models. Section 5.1 covers experimental setup. Section 5.2 provides excursion of evaluation metrics such as the inception score, and the Fréchet inception distance. This section also includes discussion on the suitability of the selected evaluation metrices.

Section 5.3 outlines our experiment with contrastive loss, the strategy to combine it with the MA-GP loss and generator conditional loss and techniques for stabilizing GAN training. In the following section, hyperparameter tuning was performed. A visual analysis of the tuned model is also included in this section. Lastly, Section 5.4 compares our tuned model with other approaches in T2I generation.

5.1 Introduction

This section provides overview of model's experimental setup. We used the Caltech-UCSD Birds 200 (CUB) dataset [74]. This is a well-known dataset used in several state-of-the-art text-to-image generation models. This dataset has 8855 train and 2933 test images. Each image is described in ten different captions which are also part of this dataset. For the purpose of computing evaluation metrics, image is generated for the input captions from the test set. Therefore, the evaluation metrics are computed over 2933 images. We computed our evaluation metrics at every 10 epoch.

Images included in this dataset are real-world images belonging to 200 different classes. Train and test split is formed on the basis of image classes. 150 classes are included in the train set and 50 classes in the test set.

5.2 Evaluation Metrices

Evaluation of generative adversarial networks is similar to the evaluation of other models. Typically a check point is selected with weights frozen at a certain point. Then the output of generative model is compared against some metrices which are used across similar models for evaluation purpose.

Evaluation of GANs is challenging because of two factors:

- Generator of GAN generates a fake image starting from the noise. There does not exist any concrete way of telling how realistic the generated image is.
- Discriminator never gains perfection in classifying fake and real images and it often overfits in classifying fake and real images for a particular generator. So the discriminator can not be used to decide which generator is better then the other one.

Direct evaluation metrices like the likelihood are not applicable to the GANs, as GANs lack any overt representation. It is the reason behind the use of sampling based metrices for evaluation purposes. Two important considerations while evaluating the generated images are fidelity and diversity. Fidelity measures image quality and diversity measures the variety. For comparing the images on fidelity and diversity grounds, higher level features of the images are required to be compared. To perform such a comparison first task is to extract the features of both generated image and the real image. Typically a pretrained classifier, trained on large dataset like Imagenet is used for feature extraction. End task of the classifier is not required for this purpose. While training for this classification task, this network has learned valuable features which can be used for evaluation of images. This is the reason that final classification layer is lobbed off and the output from

an earlier layer is used. Most commonly the last pooling layer is used for feature extraction[78].

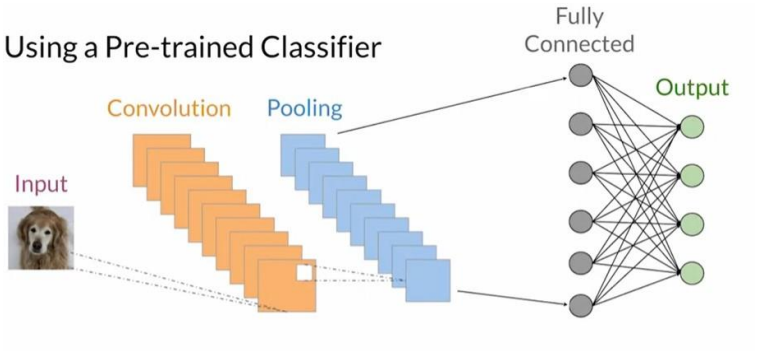

Figure 5.1: Pretrained classifier.

Imagenet is a huge dataset with 14,000,000 images belonging to 20,000 categories. Features extracted from the classifier trained on the Imagenet are often called Imagenet embeddings. [79] Inception-v3-Network is a 42 layer deep convolutional neural network classifier trained on Imagenet. It is used for feature extraction of real and fake images which are then compared for evaluations. [80] Two methods which are commonly used for feature comparison are discussed in the following sections .

5.2.1 Fre chet Inception Distance (FID)

It is the most popular metric for measuring the feature distance between the generated and real images. It is used for measuring the distance between curves that is further extended to measure the distance between distributions. FID between two single dimensional, normal distributions is defined as:

$$d(X, Y) = (\mu X - \mu Y)^2 + (\sigma X - \sigma Y)^2 \tag{5.2.1}$$

Where mean gives a sense of centre and standard deviation tells about the spread. Squares are taken to penalize the values farther away in the distributions. Multi variable normal distributions generalize the idea of normal distribution to higher dimension which are used to model complex distributions. In such distributions values in one dimension causes values in other dimension to become less or more

likely.Such a relationship is captured in the covariance matrix. Covariance measures the variance between two dimensions. Multivariate normal FID is:

$$FID = ||\mu X - \mu Y||^2 + Tr(\Sigma X + \Sigma Y - 2\sqrt{\Sigma X \Sigma Y}) \tag{5.2.2}$$

Where Σ denotes the covariance matrices. Real and fake image feature embeddings are treated as two multivariate distributions. Feature embeddings for both the fake and real images are extracted using Inception V3- network. FID evaluates the statistics obtained from the target multi-variable distribution and that of the multi-variable synthesized distribution. For the purpose of evaluation both the representations are normalized. In this evaluation, FID finds the difference in the mean and variance of the two distributions. Smaller the difference in the statistics of two distribution better the FID score is [78].

5.2.2 Inception Score (IS)

It is another widely used metric for calculating the distance between real and fake image distributions. It was developed before FID and is more relevant to the conditional generation. Inception score also uses Inception-V3 classifier pretrained on Imagenet. The difference here is IS uses it as a classifier, and not for feature extraction. A fake image is fed into the classifier and seen how it is been classified by the classifier. High value on one class suggests that image clearly resembles one class, means high fidelity is achieved. It is done for many samples to arrive at a judgment about the performance of the GAN. Second requirement from GAN is to produce diverse outputs[81].

Entropy can be thought of as randomness. While evaluating for the fidelity in the generated images the focus is on keeping entropy low. Where as on the diversity measures, high entropy is expected as more variety is required to be generated. These measurements are the used to make a decision about the performance of the model under consideration.IS uses KL Divergence as:

$$KLDivergenceD_{KL}\ (p(y|x)|p(y)) = p(y|x)log(\frac{p(y|x)}{p(y)}) \tag{5.2.3}$$

In this equation $p(y|x)$ defines conditional distribution and $p(y)$ describes marginal

distribution [81]. IS is calculated as:

$$IS = exp(\mathrm{E}_{x \sim p} D_{KL}(p(y|x)||p(y))) \tag{5.2.4}$$

5.3 Model Evaluation

This section illustrates the improvements in FID score and IS by the incorporation of contrastive loss and compare it with FID score and IS of the baseline models. We performed a comprehensive evaluation of our proposed model on two datasets which is summarized in the table 5.1.

Figure 5.2 shows the impact of incorporating the contrastive loss in the training phase of deep fusion generative adversarial network.

5.3.1 Baselines

For the purpose of comparison, we have considered following models.

- **AttnGAN**[55]: It is stack based architecture which enabled the generative network to formulize different regions of the image conditioned on most relevant words. An extra network DAMSM(Deep Attentional Multimodal Similarity Model) was used for ensuring semantic consistency of text and generated image.
- **MirrorGAN**[59]: This text-to-image generation model improved the consistency of generated image for input by redescription. It used stack based architecture to generate images from local word and global sentence attention. To ensure semantic consistency between the generated image and text, it also used encoder decoder-based image caption framework [82] for generating text from the fake image. Then computing loss of two text captions and augmenting the adversarial loss with this computed and value help the model towards better convergence.
- **DM-GAN**[83]: In this model a dynamic memory module was used to refine fuzzy image contents. Dynamic memory module comprised of a memory

writing gate for selecting the important text information based on the initial image content which enabled more consistent image generation conditioned on text.

- **XMC-GAN**[13]: This model used an attentional self-modulation generator for generating image from the text and used contrastive loss along with global and local attentional conditional losses for improving the role of discriminator in guiding the generator towards better convergence.
- DAE GAN [84]: This model was built upon the basic intuition of generating basic image from the sentence embedding and adding fine grained details from the words present in the sentence. It added aspect details information for further improving the quality of the generated image. Attended Global Refinement (AGR) module was used to employ fine-grained word-level features for global refinement, and a novel Aspect-aware Local Refinement (ALR) module to utilize aspect-level features for local enhancement. By alternately applying these two components in a dynamic way, image details were refined from both global and local perspectives.
- DF-GAN [66]: This model offered single generator and discriminator pair for generating high resolution images. For this purpose they utilized the affine transformation in the fusion blocks to fuse the text and image features. Scaling and shifting parameters were applied in the model in the MLPs to fuse text and image features and thus avoided the fuzzy features in the finally generated images.

5.3.2 Implementation Details

The proposed model is trained on google colab pro account having Tesla T4 GPU with 25GB RAM. We used the Adam optimiser [33] for both the generators and the discriminators with a learning rate of 0.0002, $\beta 1 = 0.0$, and $\beta 2 = 0.999$. We trained with a batch size of 15 because of the high memory consumption of images. We generated 256x256 images. Hyper parameter temperature is set to $\tau = 0.2$ and weight $\lambda)c = 0.5$ for training the generative model.

Algorithm 1 Generator Adversarial Network Model training algorithm with Contrastive loss

Input: Two components networks of GAN Generator G, Discriminator D. For training the generative model Batch size set to M, Hyper parameters temperature τ and coefficient λ_c are set, pretrained text encoder b for getting text embeddings and pretrained image encoder g for getting the image encoding.

for number of training iterations **do**

  for $t = 1, \ldots\ldots, M$ **do**

    Sample $\{Z_i\}_{i=1}^{M}$ from $P_{\square}$

    Sample image img and captions c_i, c_j

    $\{ (img_i, c_i)\}_{i=1}^{M} \sim P_{data}(img, c)$ and $\{ (img_i, c_j)\}_{j=1}^{M} \sim P_{data}(img, c)$

    $e_1 = b(c_i)$

    $e_2 = b(c_j)$

    $L_{D_{c1}} = E_{img\sim r}[min(0, -1 + D(img, c_1))] - \frac{1}{2}E_{\square\sim P_g}[min(0, -1 - D(G((\square), c_1)))] - \frac{1}{2}E_{img\sim P_{min}}[min(0, -1 - D(img, c_1))]$

    $L_{D_{c2}} = E_{img\sim r}[min(0, -1 + D(img, c_2))] - \frac{1}{2}E_{\square\sim P_g}[min(0, -1 - D(G((\square), C_2)))] - \frac{1}{2}E_{img\sim P_{min}}[min(0, -1 - D(img, C_2))]$

    $L_D = L_{D\ c1} + L_{D\ c2} + kE_{img\sim P}[||(\square_{(}img)D(img, C_1)\square_e D(img, c_1))^p||]$

    Update D to minimize L_D

  end for

  Sample $\{Z_i\}_{i=1}^{M} \sim P(Z), \{(img_i, c_i)\}_{i=1}^{M} \sim P_{data}(img, c), \{(img_j, c_j)\}_{i=1}^{M} \sim P_{data}(img, c)$

  $L_{G_{c1}} = -E_{G(\square)} \sim P_g[D(G(\square), c_1)]$

  $L_{G_{c2}} = -E_{G(\square)} \sim P_g[D(G(\square), c2)]$

  $img_1 = g(G(\square, c_1))$

  $img_2 = g(G(\square, c_2))$

  $L_c = NT - Xent(img_1, img_2)$

  $L_G = L_{Gc1} + L_{Gc2} + \lambda_c L_c$

  Update G to minimize L_G

end for

Table 5.1: The results of evaluation metrices with state of the art methods on the test set for CUB and COCO datasets

Model	CUB		COCO	
	IS ↑	FID ↓	IS ↑	FID ↓
AttnGAN [55]	4.36	23.98	-	35.49
MirrorGAN [59]	4.56	18.34	-	34.71
DM-GAN [83]	4.75	16.09	-	32.64
XMC-GAN [13]	-	-	-	9.30
DAE-GAN [84]	4.42	15.19	-	28.12
DF-GAN [66]	5.10	14.81	-	19.32
Ours	3.49	15.69	-	43.07

We trained our model for 1000 epochs on CUB dataset and 103 epochs on COCO dataset. Algorithm describing the training sequence of generative model is given as Algorithm 1.

5.3.3 Comprehensive Performance Analysis

Comparison of our model with GAN based state of the art models is presented in Table 5.1 . This table presents the optimized outcomes of each baseline models on benchmark datasets. The final row displays how the proposed model has performed on the benchmark datasets. Due to limitation of computational resources, we have trained our model on CUB dataset for 1000 epochs and on COCO dataset for only 103 epochs. Model has shown visible improvement on automated metric scales even after lesser training as compared to the other models under consideration.

5.3.4 Performance Visualization

Visual examination of selected images (Fig 5.2) convincingly shows the quality improvement achieved by employing the contrastive loss in the training phase of GAN.

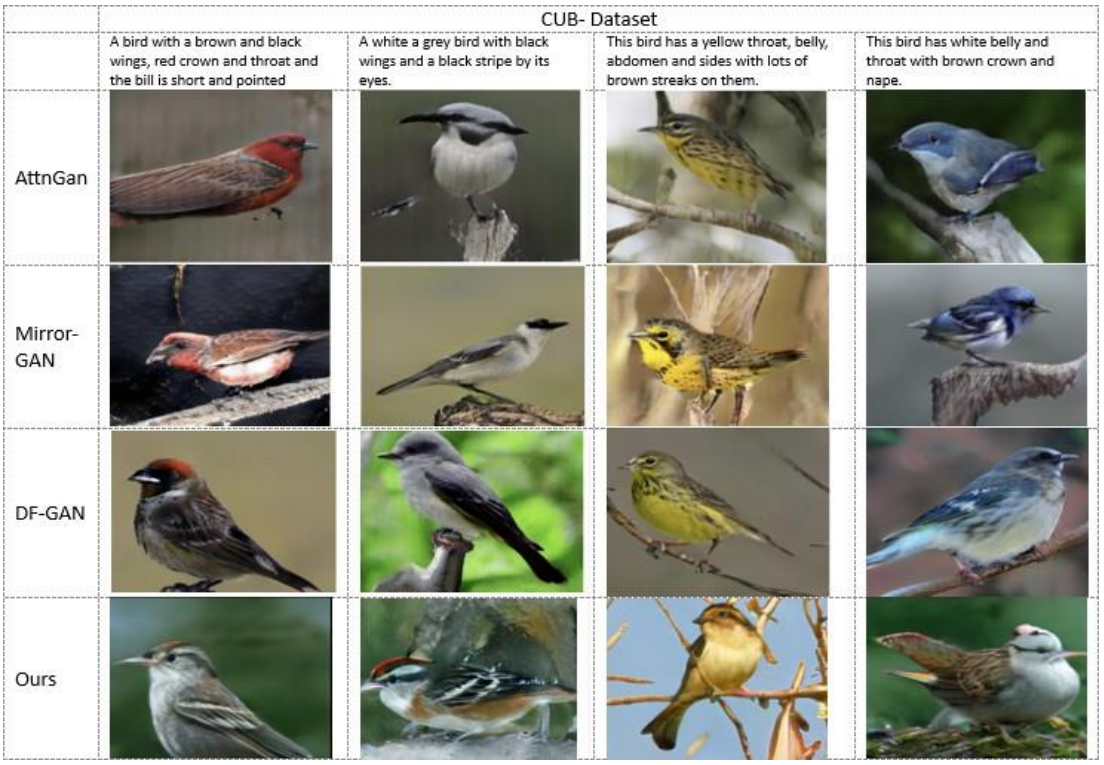

Figure 5.2: Generated images for selected examples from CUB dataset. These images are generated from the model trained on CUB dataset only for 1000 epochs but the generated images are even then are of comparable quality with the state of the art models.

5.4 Ablation Study

In our work we used contrastive learning during the training phase of the GAN for generating images guided by input text. We investigated the incorporation of contrastive loss with generator loss and discriminator loss respectively to arrive at the fair conclusion about the effect of the constrastive learning to augment the generative model in learning a better representation. With automated evaluations we established the opinion that constrastive learning adds a positive effect in making a generative model to learn a better data distribution and hence generate more consistent images for the text input. With the DF-GAN as the baseline model we show the results for (1) contrastive loss with MA-GP loss of critic (2) contrastive loss incorporation with the generator adversarial loss (3) both generator and discriminator getting contrastive loss along with adversarial losses.

Experimental results are shown in the Fig 5.3. It is evident from the results that contrastive learning technique improves the text to image generation capability of the baseline model. Generator adversarial loss augmented with contrastive loss make the generative model to converge towards hidden representation at improved rate as compared to the incorporation of contrastive loss along with the adversarial loss for the discriminator only.

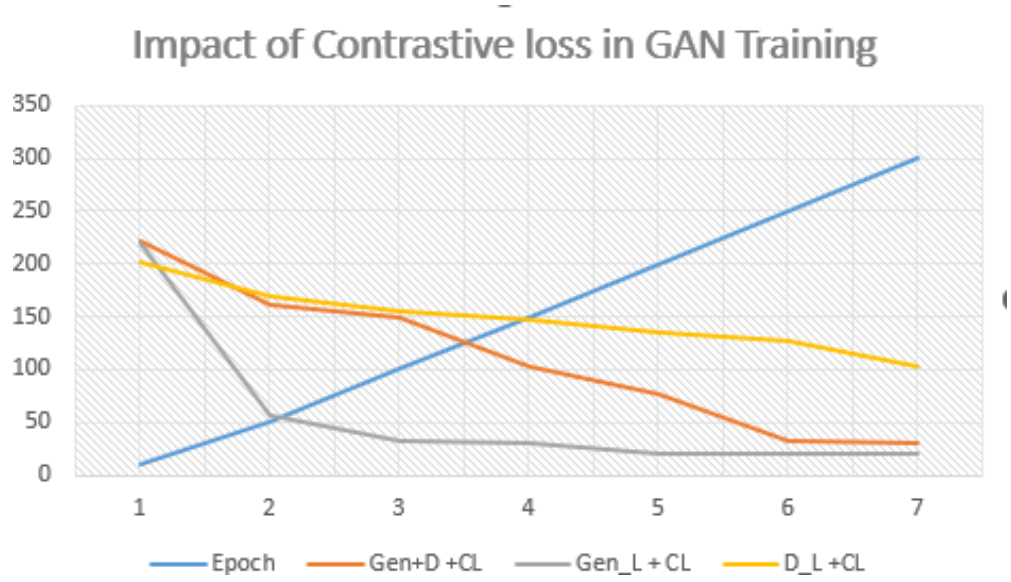

Figure 5.3: Ablation study results demonstrating the impact of Contrastive loss in the training of generative model for the CUB dataset. These results are obtained after training the model for 300 epochs.

We experimented with hyper-parameters to establish the impact of these values on the performance of the overall model. We performed these investigations for the CUB dataset. Table 5.2 presents the values of hyper-parameters used for the purpose. First we adjusted the values of temperature to get fair idea about its impact in the performance contrastive learning. we tuned its value at {0.1, 0.5, 1}. Form the experimental results it became evident that increasing the value from 0.5 leads lowering the FID score for the learned representation. Then we experimented with the second hyper-parameter λ_c and tuned its weights at values {0.1, 0.2, 0.5, 1.0}. Results of experiments lead to idea that changing the value of this hyper-parameter has very marginal impact on the training of generative model.

Table 5.2: Ablation study showing results on choosing different values of the hyper-parameters τ and λ_c

Model	Hyper-Parameter	IS ↑	FID ↓
CL in GAN Training	τ		
	0.1	3.90	16.11
	0.5	3.49	15.69
	1	2.99	19.99
CL in GAN Training	λ_c		
	0.1	4.42	16.22
	0.2	3.49	15.69
	0.5	3.77	15.89
	1.0	3.65	15.77

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, we used contrastive loss along with the adversarial loss of generative model, in the training phase of a simple one stage GAN. Contrastive learning, a self-supervised learning technique, improved the convergence of model in learning the underlying structure of the data. Comprehensive experiments on bench mark datasets showed remarkable improvements in convergence rate even with less training. However, due to constraint of computational resources, we have to stick to smaller batch size. Contrastive learning performs best with large batch size and more training. Use of larger batch size with enormous dataset can remarkably improve the results.

References

[1] Xudong Mao and Qing Li. *Generative adversarial networks for image generation*. Springer, 2021.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[3] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58, 2016.

[4] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16515–16525, 2022.

[5] James M Clark and Allan Paivio. Dual coding theory and education. *Educational psychology review*, 3(3):149–210, 1991.

[6] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. *Advances in neural information processing systems*, 29:217–225, 2016.

[7] Bo Dai and Dahua Lin. Contrastive learning for image captioning. *Advances in Neural Information Processing Systems*, 30, 2017.

[8] Kwot Sin Lee, Ngoc-Trung Tran, and Ngai-Man Cheung. Infomax-gan: Improved adversarial image generation via information maximization and con-

trastive learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3942–3952, 2021.

[9] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.

[10] Fengchun Qiao, Naiming Yao, Zirui Jiao, Zhihao Li, Hui Chen, and Hongan Wang. Geometry-contrastive gan for facial expression transfer. *arXiv preprint arXiv:1802.01822*, 2018.

[11] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5154–5163, 2020.

[12] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[13] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 833–842, 2021.

[14] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12154–12163, 2019.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[16] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[17] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. Semantics disentangling for text-to-image generation. In *Proceedings*

of the IEEE/CVF conference on computer vision and pattern recognition, pages 2327–2336, 2019.

[18] Hui Ye, Xiulong Yang, Martin Takac, Rajshekhar Sunderraman, and Shihao Ji. Improving text-to-image synthesis using contrastive learning. *arXiv preprint arXiv:2107.02423*, 2021.

[19] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[20] Eli Stevens, Luca Antiga, and Thomas Viehmann. *Deep learning with pytorch*. Manning, 2020.

[21] Shayle R Searle and Marvin HJ Gruber. *Linear models*. John Wiley & Sons, 2016.

[22] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659*, 2017.

[23] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

[24] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[25] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

[26] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. *Advances in neural information processing systems*, 31, 2018.

[27] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[29] 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.

[30] Yang Yu, Zhiqiang Gong, Ping Zhong, and Jiaxin Shan. Unsupervised representation learning with deep convolutional neural network for remote sensing images. In *International Conference on Image and Graphics*, pages 97–108. Springer, 2017.

[31] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. Face aging with conditional generative adversarial networks. In *2017 IEEE international conference on image processing (ICIP)*, pages 2089–2093. IEEE, 2017.

[32] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. It takes (only) two: Adversarial generator-encoder networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[33] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *arXiv preprint arXiv:2003.05991*, 2020.

[34] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26, 2020.

[35] Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Xu, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[36] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[38] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016.

[39] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[40] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[41] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in neural information processing systems*, pages 1099–1107, 2015.

[42] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5706–5714, 2017.

[43] Hyojin Park, Youngjoon Yoo, and Nojun Kwak. Mc-gan: Multi-conditional generative adversarial network for image synthesis. *arXiv preprint arXiv:1805.01123*, 2018.

[44] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.

[45] Bruno A Olshausen and David J Field. Natural image statistics and efficient coding. *Network: computation in neural systems*, 7(2):333, 1996.

[46] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[47] Navaneeth Bodla, Gang Hua, and Rama Chellappa. Semi-supervised fusedgan for conditional image generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 669–683, 2018.

[48] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6199–6208, 2018.

[49] Lianli Gao, Daiyuan Chen, Jingkuan Song, Xing Xu, Dongxiang Zhang, and Heng Tao Shen. Perceptual pyramid adversarial networks for text-to-image synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8312–8319, 2019.

[50] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018.

[51] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[52] Xin Huang, Mingjie Wang, and Minglun Gong. Hierarchically-fused generative adversarial network for text to realistic image synthesis. In *2019 16th Conference on Computer and Robot Vision (CRV)*, pages 73–80. IEEE, 2019.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[54] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[55] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[56] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7986–7994, 2018.

[57] Ting-Yun Chang and Chi-Jen Lu. Tinygan: Distilling biggan for conditional image generation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[58] Jun Cheng, Fuxiang Wu, Yanling Tian, Lei Wang, and Dapeng Tao. Rifegan: Rich feature generation for text-to-image synthesis from prior knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10911–10920, 2020.

[59] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019.

[60] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

[61] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[62] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[63] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[64] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[65] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[66] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020.

[67] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. *Advances in Neural Information Processing Systems*, 32, 2019.

[68] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12174–12182, 2019.

[69] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[70] Ayushman Dash, John Cristian Borges Gamboa, Sheraz Ahmed, Marcus Liwicki, and Muhammad Zeshan Afzal. Tac-gan-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017.

[71] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

[72] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[73] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[74] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[75] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[76] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[77] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

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

[79] 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.

[80] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[81] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[82] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[83] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5802–5810, 2019.

[84] Shulan Ruan, Yong Zhang, Kun Zhang, Yanbo Fan, Fan Tang, Qi Liu, and Enhong Chen. Dae-gan: Dynamic aspect-aware gan for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13960–13969, 2021.